REQUIREMENT ENGINEERING PROCESS FOR MOBILE APPLICATION DEVELOPMENT: CHALLENGES AND RESOLUTIONS

By:
MAHRUKH TANVEER

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

January 2022

Requirement Engineering Process for Mobile Application Development: Challenges and Resolutions

By:
MAHRUKH TANVEER

MSSE, National University of Modern Languages, Islamabad, 2022

A THESIS SUBMIT IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTERS OF SCIENCE

In Software Engineering

To
FACULTY OF SOFTWARE ENGINEERING

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: Requirement Engineering Process for Mobile Application Development: Challenges and Resolutions

Submitted by: Mahrukh Tanveer **Registration #:** MSSE-S19-18634

Master of Science in Software Engineering
Degree name in full

Software Engineering
Name of Discipline

Dr. Huma Hayat _____
Name of Research Supervisor Signature of Research Supervisor

Dr. Basit Shahzad _____
Name of Dean (FE&CS) Signature of Dean (FE&CS)

Prof. Dr. Muhammad Safeer Awan _____
Name of Pro-Rector Academics Signature of Pro-Rector Academics

February 2022
Date

AUTHOR'S DECLARATION

I Mahrukh Tanveer Daughter of Tanveer Alam Registration # MSSE-S19-18634

Discipline Software Engineering

Candidate of **Master of Science in Software Engineering (MSSE)** at the National University of Modern Languages do hereby declare that the thesis **Requirement Engineering Process for Mobile Application Development: Challenges and Resolutions** submitted by me in partial fulfillment of MSSE degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in the future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be canceled and the degree revoked.

Signature of Candidate

Mahrukh Tanveer

Name of Candidate

February 2022

Date

ABSTRACT

Requirement Engineering Process for Mobile Application Development: Challenges & Resolutions

Due to the extensive usage of smartphones, more and more development firms are investing in mobile app development to leverage the growing demand. With this ongoing demand for mobile development, the presence and importance of web applications cannot be denied. Although both mobile and web development have pros and cons. Talking about the advancement in technologies, mobile applications are on top priority. But, it is still questionable what are and what type of challenges do the mobile developers face during the execution of the requirement engineering process while developing software applications for mobile platforms. So, for this purpose, research is conducted based on the entire software requirement engineering process that is determining the challenges for the execution of the entire requirement engineering process focusing the mobile development. The research has adopted the Systematic Literature Review for investigating the challenges, then an Expert Review is piloted for the validation of the list of challenges. Finally, an Industrial survey for the proposal of mitigating the challenges is accompanied. As a contribution to the research study, a validated and finalized list of 46 Challenges along with their Resolution Strategies is presented. This research may guide the practitioners and academicians towards the Requirement Engineering Process for Mobile Application Development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

R.E: Requirement Engineering

SLR: Systematic Literature Review

CPU: Central Processing Unit

GPS: Global Positioning System

MDL: Mobile Development Lifecycle

PRD: Product Requirement Document

UX: User Experience

QA: Quality Assurance

UWP: Universal Window Platform

DOC: Double Orientation Change

CO: Computation Offloading

MEC: Mobile Edge Computing

SEI: Software Engineering Institute

GT: Game Theory

RL: Reinforcement Learning

LSTM: Long Short Term Memory

CC: Cloud Computing

SMDP-RAS: Semi Markov Decision process-based resource allocation

MCC: Mobile Cloud Computing

CRM: Cyclic random movement

CRMGA: Cyclic random movement based on genetic algorithm

GAMCO: Genetic based Multi-site Computation Offloading

MCOSP: Mobile Code Offloading and Scheduling Problem

API: Application Programming Interface

CSOS: Context-Sensitive Offloading System

CUPUS: CloUd- based PUblish/Subscribe middleware

ICC: Inter Component Communication

NP: Non-deterministic Polynomial

DTNs: Delay Tolerant Networks

GUI: Graphical User Interface

OS: Operating System

LIST OF APPENDICES

ACKNOWLEDGEMENTS

In the name of Allah, the Most Merciful and Generous. I'd want to thank Allah Almighty for showers of blessings to accomplish my thesis during the COVID-19 pandemic.

I'd like to offer my sincere appreciation to everyone who has been a tower of strength for me throughout this time. Also, I would to special thanks to the Department of Software Engineering in general and my mentors Dr. Huma Hayat and Dr. Muhammad Noman Malik in particular, for their important constant supervision, encouragement, constructive recommendations, generous spending of time, and profound attention during the thesis.

DEDICATION

Alhamdulillah… All gratitude be to Allah Almighty for molding me into the person I am today and allowing me to realize my ambition. I dedicate this thesis work to my Supervisor, “Dr. Huma Hayat” for guiding me and giving me her precious time whenever, I needed it the most during my Research Study, Co-Supervisor, “Dr. Muhammad Noman Malik” and last but not the least my beloved brother “Dr.Sheraz Alam Khan” for supporting me the most and guiding me in writing my thesis report.

Thanks to my beloved sister Aisha Baji who has always been a supportive role and my parents who have always been pillars of strength in my life. My father, raised me to be THE STRONGEST AND PROUDEST DAUGHTER by loving me unconditionally and my mother who always remembers me in her special prayers.

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter starts with the introduction of software application development in the mobile platform context. Furthermore, the limitations of mobile devices along with other several challenges are discussed that make the development crucial for mobile developers to engineer mobile software in the context of the requirement engineering process. In addition, how mobile development is different than traditional or web app development in terms of the lifecycle is accompanied. The background of the proposed research study in this chapter shows the lack of research in literature that there is as such no research on the entire requirement engineering process for mobile development. This motivates us to investigate the challenges of implementing the requirement engineering process, particularly for mobile platforms. So, the research problem, aim of the research, and scope of research are presented along with the research questions. The research problem portrays the cause of conducting this research study.

1.2 Research Background

Mobile applications, typically referred to as mobile apps, are application software intended to run on smartphones and other portable devices. Mobile apps, as opposed to conventional and web applications, are designed to meet specific difficulties. Mobile applications, for example, must handle user input as well as input from rapidly evolving environments. Furthermore, when compared to modern personal computers and laptops, cell phones and mobile devices still have restricted resources. Furthermore, there is a wide range of mobile operating systems and the

The same operating system is continually upgraded in a short amount of time [1]. It's no wonder that the mobile app sector is surging, with over 3.2 billion smartphone users worldwide. App usage and smartphone penetration are continuing to rise significantly, with no signs of decreasing shortly. Add in the 1.14 billion tablet users around the world, a figure that has increased by 36% in the last six years [2].

Now, when talking about the development of simple desktop applications which is defined as a variety of web-based software that efficiently and effectively performs over the Internet using web browsers and web technology. Web applications use a blend of server-side scripts (PHP and ASP) to hold and retrieve data, as well as client-side scripts (JavaScript and HTML) displaying data to users. Users can use online forms, content management systems, shopping carts, and other tools to engage with the company. Employees may also use the apps to create documents, share information, collaborate on projects, and work on shared documents regardless of where they are or what device they are using [3]. In comparison with smartphone usage, the desktop contributes 42.9% of the traffic while mobile phones approximately cater to 54.25% of this [4].

As the research is typically based on the requirement engineering process, so, it is important to know what the requirement engineering process is? Before that, it is needed to understand the meaning of "requirements" which is according to IEEE: "The characteristic or condition that must be satisfied or exists in a system or component of a system for a functionally or practically applicable contract, specification, standard, or another document to be satisfied" [5]. Whereas on the other hand, the R.E. process is a practice of gathering software requirements which are the required functionalities from clients, interpreting them, and documenting them to engineer a quality product [6]. The major objective of the requirement engineering process is to achieve a high-quality software product while satisfying the user's needs [7]. The development of any software starts from the R.E. process that gains extreme importance in accomplishing the entire process smoothly in which each phase is equally important that from Requirement Gathering till the Requirement Documentation [8]. The smooth execution of this process not only leads to an efficient system but achieves cost-effectiveness as well [9].

There are many reasons for what makes mobile development different from traditional development. Based on limitations in mobile devices, challenges at the design level, testing of various types of apps, we can say that mobile apps operate differently than conventional ones. The reasons are discussed respectively below.

Firstly, a limited number of devices and screen sizes cannot be accommodated when designing and building an app. One of the biggest challenges is to create an app that runs well on as many different devices as possible, regardless of screen size [10]. One of the shortcomings of mobile phones is their reliance on battery power. Newly introduced models are more power-hungry. Furthermore, the average life of a mobile phone battery is only two to three years. Privacy has grown more difficult to maintain due to the availability of photo and video technology on mobile phones. Being photographed or filmed is possible almost everywhere. It's easy for that media to get up in the public domain via social media once it's been captured. There has been an increase in the use of mobile phones for tracking people's movements. So far, so good, but this is yet more step towards the security state when citizens have no privacy and don't even know when or who is following them, let alone have the power to fight back [11]. The above-mentioned reasons are related to the limitation of smartphone devices.

One of the major problems at design is that you should keep the number of form fields to a minimum and pre-select reasonable defaults when typing on a touch keyboard. You should think about how you will handle data problems. Because of the touch keyboard and lower screen size, forms filled out on touch devices will have significantly more errors. In addition, touch devices have smaller screens, which results in a smaller context. Thus, users have a harder time seeing what's going on, making decisions, and remembering what they've read before. Clicking links and buttons on a touch screen with your finger reduces the accuracy of the click. Fat finger problem is another name for this. All clickable components need to be large enough to reliably touch with a human finger and far enough apart so that users don't mistakenly click the wrong element, to achieve this goal. Although in contrast to desktop computers, the performance of touch devices is improving quickly, they are still slow devices. A lot of Java-script is executed on page load, thus start-up might be painfully delayed [12].

Disparities in OS versions lead to fragmentation. The number of versions of both Android and iOS is in the tens of thousands. Developers must ensure that their program runs on all different versions of operating systems before releasing it (whether it is Android, iOS, or any other OS). Regardless of the operating system installed on the device, the user experience should be smooth. A discrepancy in device availability leads to fragmentation. There are so many configurations that it is considered the most difficult part of testing. If the device is from the same product family, we can't guarantee that the same program will operate smoothly on another device because hardware specifications such as CPU speed, RAM size, screen

Resolutions or OS optimization could be different. In the background, a large number of apps are using CPU cycles, which drains battery life and slows down the device's operation. The power consumption of mobile apps should be kept to a minimum by following best practices. Before the final release, a thorough testing process is required to discover and rectify any leaks that may have occurred. When it comes to a product's future, it's crucial. It's a challenge to maintain the interaction between user and app simple and concise for the user, but while displaying all important information on a little screen in front of them. Factors like readability, font size, color combination, touch speed, processing time, etc. can have an impact on user experience. Globally, we can expect users to access apps via a variety of networks including cellular (3G+), Wi-Fi (4G), LTE (Long Term Evolution), and 2G (in select regions). In all of these networks, the app should be able to work flawlessly (in some cases there can dead zone also where there is no network available) [13]. One of the issues that must be dealt with testing various types of apps such as native (typically manufactured to run on a specific platform), web (mobile-optimized web pages that look like an app), and hybrid (blend of web and native both). This is a major worry because the testing of one app is very different from the other as their implementation differs greatly [14].

1.3 Research Problem

A seamless requirement engineering process in mobile development is extremely important, so we must be aware of the challenges that arise when developing mobile applications. To the best of our knowledge, there is as such no research conducted before, that includes the establishment of the challenges for the mobile applications while executing the entire requirement engineering process. Although, several studies have been conducted on mobile development but with different perspectives. Such as, some of the authors discussed requirement elicitation techniques challenges [9] and the same work but specific to elderly people facing UI challenges, while some showed their work on elicitation concepts but focusing on social networking or social media apps [15]. Multiple authors discussed requirement elicitation challenges but the main emphasis was on disabled people and visually challenges people [16], [17]. Similarly, some authors presented their work on requirement engineering process merged with agile methodologies [17] and some contributed their work in the development of mobile applications using agile-scrum [18]. Furthermore, testing issues in mobile application development [1] were also part of existing work in previous studies.

The aforementioned studies have clearly shown that as such no research is being conducted before, on the entire R.E. process particularly for mobile platforms, so, this research gap motivates us to discover all possible challenges that are overlooked for developing mobile applications in context with the R.E. process.

1.4 Research Questions

Based on the research aim, the following research questions are designed shown below:

RQ1: What are the key challenges confronted by the developers during the development of software applications for mobile platforms using the R.E. process?

RQ2: In how many possible classes, the discovered challenges can be categorized?

RQ3: What are the mitigation strategies that should be adopted to get over the investigated challenges?

1.5 Research Objective

As a result of the research study, the answer to the questions raised above are as follows:

Objective 1: The objective of RQ1 aims to identify the key challenges that come across for mobile applications development using the entire R.E. process.

Objective 2: The objective of RQ2 aims to classify the listed identified challenges obtained in phase 1

Objective 3: The objective of RQ3 aims to overcome the investigated challenges in the form of mitigation strategies by conducting an industrial survey.

1.6 Research Purpose

This research discusses an entire need to implement the R.E. process smoothly, so, for this purpose, it is compulsory to be must aware of all possible barriers that came across its way stopping to implement the R.E. process seamlessly in the development of mobile applications. Based on this need, the aim mainly focused on the investigation of challenges that the developers faced during the accomplishment of the R.E. process while developing software applications, particularly for mobile platforms. Moreover, this research is intended to generate

a list of challenges as a result of SLR conduction. The output of SLR is being analyzed and a coding technique is applied from Grounded Theory to eradicate the replication from SLR findings obtained as a result of the first phase. After the removal of replicated data, the identified challenges are gone through for evaluation by domain experts in phase 2 of Expert Review conduction. Finally, the final validated list is surveyed in the industry suggesting the resolutions to overcome these identified challenges.

1.7 Scope

The scope of the research study is to investigate all possible challenges confronted during the execution of the entire R.E. process for the development of mobile applications. The selection of primary studies is based on past ten years' papers that are from 2010-2020. The study research is restricted to mobile applications rather than that of web applications. The primary studies included are from high-quality and peer-reviewed journals, mature conferences, and accepted manuscripts. On top of that, the main focus was on the general "mobile applications" term, the applications specific to healthcare, game-development, virtual reality-based, augmented reality-based, graphical applications, location-based applications, and so on are not part of our research. Whereas, for Expert Review evaluation, the domain experts were chosen to have experience of at least 5 years and must be specialized in their domain is a mobile development and requirement engineering domains. For the industrial survey, people from the software industry were selected having experience of at least 2 years in mobile development and requirement engineering in both domains.

1.8 Research Contribution

The present conducted research contributes a list of key challenges that came across as hurdles in mobile development by implementing an entire R.E. process. Once the challenges are investigated in listed form, that list has gone through a technique of grounded theory, so that the redundant data can be removed. Then final core constructs are obtained in the form of challenges which are then passed through the next phase. Secondly, a classification of the investigated challenges is presented based on their nature that is either resource-related, requirement-related, security & privacy-related, communication-related, or stakeholder-related. Lastly, some resolutions are also proposed to overcome the identified challenges

1.9 Thesis Outline

Our research thesis outlines 6 chapters. The first chapter is about the introduction of the paper which consists of the overview of chapter one along with the research background, the research problem, research questions, objectives, aim, scope, contribution, and at the end is an outline of the thesis.

Chapter 2 is all about the overview of Requirement Engineering Process, Mobile Development, and existing studies discussed in detail. The research thesis comprises the relevant primary studies on mobile development challenges in the Requirement Engineering context. Chapter 3 addressed the multiple methodologies adopted to conduct the research study. Chapter 4 depicts the Systematic Literature Review findings along with Grounded Theory and Expert Review conduction. Chapter 5 of the proposed study explains the findings obtained from an Industrial Survey. The last chapter describes the conclusion of the whole research study along with the future work and threats of validity.

1.10 Summary of the Chapter

This chapter briefly portraits the introduction of the research gap by reporting the background, some of the reasons differentiating mobile application development from that of traditional or web development along with the research questions, research objectives, research purpose, scope, and research contribution respectively. At last, the thesis outline is mentioned.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the previous chapter, the brief introduction of our research thesis was discussed where the background, some of the major reasons with which mobile development can be distinguished from web development were reported. In addition, the research problem, research questions, objectives along purpose, scope, and contribution are described as well.

2.2 Requirement Engineering Process

Software Requirement Engineering commonly referred to as Requirement Engineering, is a systematic approach, rather it’s an entire process. It can be defined as a step-by-step procedure having different phases starting from the Requirement Elicitation phase (Requirement Gathering) to Requirement Specification (Requirement Documentation) [19]. Requirement Engineering Process evaluates software requirements based on the customer's needs. The goal of requirements engineering (RE) is to identify the system's objectives. Because of the insufficient, inaccurate, and vague requirements, 95% of software fails. A deep understanding of the Requirement Engineering process is crucial for the development of a successful software system [20]. This method supports the gathering of all feasible needs from multiple sources, which are then incorporated into software development. The purpose of the requirement engineering process is the entire satisfaction of the consumer to boost the software's productivity [21]. The major goal of this Requirement Engineering is to fulfill your end user's degree with the least amount of money and time possible [22].

Talking about the phases, the Requirement Elicitation and Analysis is the very first phase commonly known as Requirements Gathering. But, before Requirement Elicitation, it is important to know about the reasons of development that what is being developed? And Why?

What is its scope? Is the application acceptable to the users? Or not? Will the software be adaptable to changes and compliant with industry standards? The answer to all these questions comes under the Feasibility study, which is the additional step. Does it report that either the project is feasible or not? Then begins the first step of requirement gathering, which entails gathering requirements from all potential stakeholders or current procedures, if any are available. After the requirements have been collected, a thorough review is conducted to remove any anomalies, inaccuracies, or errors, as well as to resolve any disagreements that may have arisen.

Then come to the Requirement Specification phase defined as a type of document generated by a software analyst after the requirement elicitation phase from multiple sources while the customer's requirements are written in plain English. The analyst is responsible for writing the requirements in technical language for the ease and understanding of the development team. Once, the requirement specification phase is completed, then the phase of Requirement Validation is started. It simply means that all the mentioned requirements in the document are being validated. The user may demand an inappropriate or impossible solution, or experts may misunderstand the user's requirements. The last phase comes as the Requirement Management in which new changing requirements are being managed accordingly as the business need changes [23].

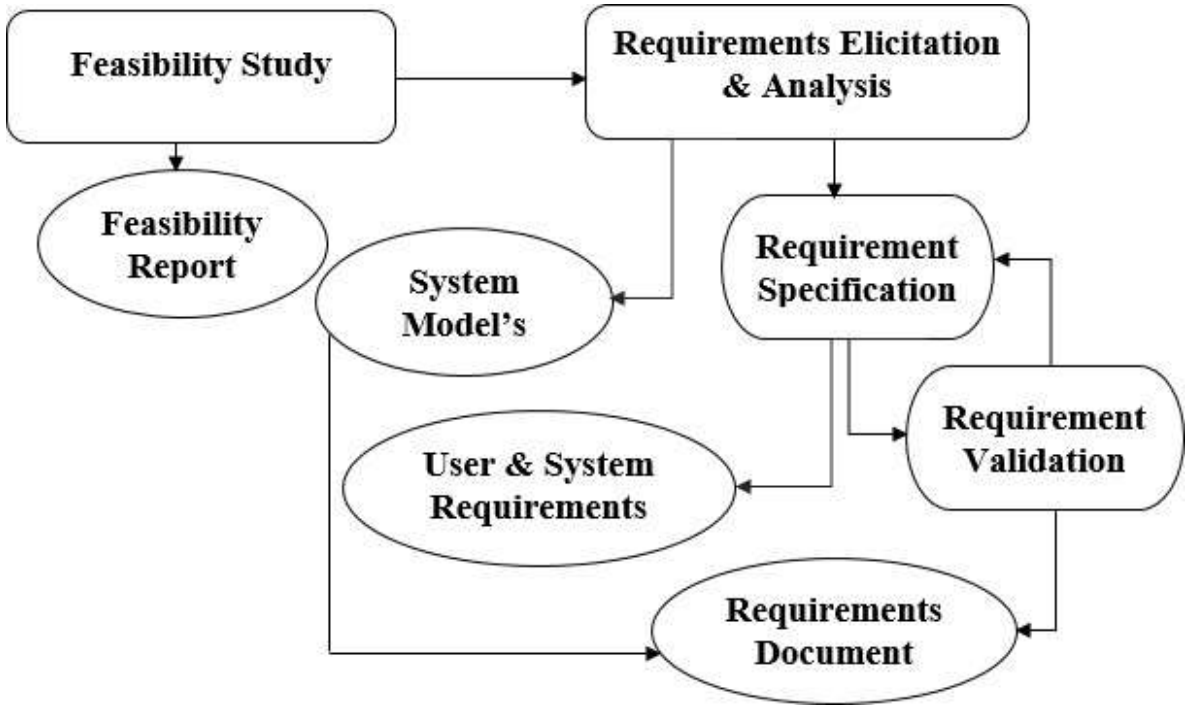

Figure 2.1: Requirement Engineering Process [23]

2.3 Mobile Application Development

Mobile applications, usually referred to as mobile apps, are software programmers designed to operate on smartphones and other mobile devices. Smartphones also referred to as smart mobile terminals are high-end mobile phones with advanced computing and connection that run on mobile operating systems. Stronger CPUs, expanding memories, high-resolution touch-screens, better sensors, GPS, high-speed internet access through Wi-Fi, and so on are all features of modern smartphones [24] [25]. Smartphones and mobile gadgets have grown increasingly popular in everyday life as considerably more computer power has been packed into them in recent years. Mobile applications have unique obstacles as compared to desktop and online applications. Mobile applications, for example, must process user input as well as input from constantly changing situations. Furthermore, when compared to modern personal computers and laptops, cell phones and mobile devices still have restricted resources. Furthermore, there is a wide range of mobile operating systems, and the same operating system is updated frequently and in a short amount of time [26].

As the number of new technologies grows at a rapid pace, software development becomes highly difficult. As a result of this rapid growth, working in such an environment becomes difficult for mobile developers confronting several obstacles during the development of mobile apps [27]. The short development lifespan of the mobile platform is one of its primary downsides, which will be covered briefly in the following chapter. Differences in the development process and device restrictions are another major contrast that distinguishes mobile apps from online apps. Because of the ambiguity of user requirements, the development process alters. Furthermore, mobile applications have a substantially shorter overall lifecycle than online or desktop applications. The next biggest noticeable distinction is the device's restricted capabilities, such as processor, battery, touch screens, and the user experience with touch screens, user context, and interactive behaviors [28].

Now, let’s talk about the Lifecycle of Mobile Development commonly referred MDL based on the principles of the software development lifecycle. It comprises multiple phases as Inception, Design, Development, Stabilization, Deployment, and Maintenance. The very first stage is Inception means planning. First of all, it is needed to convert the business vision into a prototype. Also, attempt to characterize the target consumers in terms of gender, age, occupation, position, social background, and geographic location to collect the relevant requirements. Furthermore, it also includes a properly defined scope, a chosen programming language, and PRD (Product Requirement Document) [29]. Then comes the design process

which encompasses defining the app's User Experience (UX), such as the overall layout, how it operates, and so on, as well as translating that UX into a good User Interface (UI) design, which is usually done with the assistance of a graphic designer. After done with the design phase, a development stage has been started which is an actual creation of the application and usually the most resource-intensive phase. Jumping towards the Stabilization phase, once the development is completed, in which a QA usually begins testing the application and fixing bugs. Frequently, an application will enter a limited beta phase in which a larger user community is allowed to use it and provide input, which will help to guide future modifications. Once, the testing or stabilization phase is fully completed, it's time to deploy the tested app or to distribute it among various platforms such as Android, iOS, and Universal Windows Platform (UWP). At last, Product Maintenance is a very important phase of MDL. Soon after the app is published on the App Store and Google Play, user input must be analyzed, make appropriate modifications and updates, give tech support for your product, and address issues raised by users [30].

2.4 Requirement Engineering Process for Mobile Application Development

Because of the extensive usage of smartphones, a growing number of development companies are investing in mobile app development to meet the growing demand. Given the ongoing demand for mobile development, the presence and importance of web applications cannot be overlooked. Both mobile and internet development have benefits and drawbacks when it comes to technological breakthroughs. Mobile apps, on the other hand, take precedence. The number of app downloads increased from 140.7 billion in 2016 to 218 billion in 2020, showing that the market would continue to grow [7]. In the year 2021, approximately 6.4 billion individuals own cell phones worldwide. Similarly, demand for software applications, particularly for mobile platforms, has increased significantly over the previous few decades. Nowadays, it has become a recent trend to follow a Requirement Engineering Process while developing a software application particularly for mobile platforms [2].

As mentioned above mobile development comprises many limitations so, to develop a successful software application for mobile phones, it becomes compulsory for mobile developers to follow the requirement engineering process. The reason behind this is that it is a systematic approach having multiple steps. Every step is equally important and to execute a successful mobile development process, a requirement engineering process is needed to be completely followed [19]. But, as far as we are aware, the entire requirement engineering process particularly for mobile development is in infancy. Very few

studies were reported only on a single phase of requirement elicitation or requirement prioritization or requirement engineering with agile [9] [31] [17]. But, none of the studies portray the entire requirement engineering process. So, based on the research gap, a study is being conducted on the entire Requirement Engineering process particularly for mobile applications.

2.5 Existing Studies on Requirement Engineering Process for Mobile Application Development

Because of the iPhone AppStore, mobile application development has exploded since it was initiated in July 2008. Different mobile devices like BlackBerry, Nokia Ovi, and Windows Phone have been created in the last while. Many applications are available for multiple devices and according to industry analysts, there are more than 250,000 applications in the various stores and marketplace [32]. More and more people are becoming interested in research into software development, which includes everything from system software to mobile applications. According to [33], mobile applications now account for 99 percent of the total Gross domestic product. The software development either in desktop or mobile starts from the requirement engineering process for obtaining a quality product in terms of software applications [19]. But, on the other hand, the requirement engineering typically in mobile development is still in infancy as it is quite different from traditional software development due to several obstacles. So, this chapter specifically addresses the related existing literature based on the aforementioned domain that is challenging for mobile development during the accomplishment of the R.E. process.

Table 2.1: Summary of Existing Literature relevant to the Research Studies

Title	Author/Year/ Ref. #	Type of Paper/Domain	Contribution	Limitation

A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective	Ali Shakarami/ 2020/ [34]	Mobile Edge Computing/Research article	The GT-based Computation Offloading approaches in MEC the environment is presented in the form of classical taxonomy with some open issues	The review is based on a single perspective that is Game Theory
LeakDoctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications	XIAOLEI WANG/ 2019/ [35]	Research Article/ Security & Privacy	A novel fine-grained approach named LeakDoctor is proposed	Some complex privacy disclosure cases through a network are not covered
Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing	RYUJI KOBAYASHI/ 2019/ [36]	Research Article/ Mobile Edge Computing	An efficient radio and computing resource allocation scheme is proposed	The only time factor is focused for a single scenario that is when multiple tasks are to be divided into local & offload tasks
GUILeak: Tracing Privacy Policy Claims on	Xiaoyin Wang/2018/ [37]	Software Engineering/ Research article	Proposed a novel approach to protect privacy policy violation	Inadequate validation sample

User Input Data for Android Applications			due to leak of user input data	
Why does the orientation change mess up my Android application? From GUI failures to code faults	Domenico Amalfitano/ 20 18/ [38]	Research Article/Software Testing	A framework named DOC is proposed for classifying GUI failures with the investigation of its key features along with possible faults causing them	Restricted access to the requirements of each app so, there was the risk that the GUI failures we detected could not be actual failures
Cordovaconfig: A tool for mobile hybrid apps' configuration	Abeer AlJarrah/20 18/ [39]	Conference Paper/Security & Privacy	Designed and built CORDOVACON FIG, a tool for configuring mobile hybrid apps	The sample size was small and most probably, the participants were students
A systematic study on software requirements elicitation techniques and its challenges in mobile application development	HAFSA DAR/2018/ [9]	Research Article/Requirement Engineering	Challenges on Elicitation techniques for mobile development are identified & guideline is proposed	No model or framework is proposed, data collection is manual and done by a single researcher

Why does data deletion fail? A study on deletion flaws and data remanence in android systems	JUNLIANG SHU/2017/ [40]	Research Article/Security & Privacy	Presented some flaws considering 3 typical scenarios of the android system Design & implement a framework named DataRaider to recover files from disc fragments, Also, some mitigation plans were proposed for data remanence issues along with some suggestions on data protection in android systems	Full disc encryption mechanism is not as effective as expected before all versions of android 5.0
Study and refactoring of android asynchronous programming	Yu Lin/2016/ [41]	Conference Paper/Computer Science	A refactoring tool named as AsyncDroid is proposed	Conversion of one mode of communication to another is not possible that is from shared memory based communication to distributed style

A systematic mapping study of mobile application testing techniques	Samer Zein/2016/ [1]	Accepted manuscript/Software Testing	Specific testing issues for practitioners are identified	Focused on testing issues only

A recent study was conducted in 2020 on mobile edge computing. The paper is based on computation offloading decision-making problems which state that it is critical to make the decisions about the offloading mechanism that which tasks are to be executed on the local device (mobile phones) and which tasks are to be migrated on a cloud for achieving best results in minimum time. The author has contributed CO approaches based on game theory in the form of classical taxonomy in the MEC environment [34]. The author has discussed privacy concerns in terms of privacy disclosures and for that, an analysis system is proposed to automatically diagnose privacy leaks in mobile applications. The main focus of the author is an analysis of data protection disclosures in internet traffic as well as a step towards automated confidentiality leak diagnosis. Whereas, private data disclosure is the act of sending one or more types of private data over the network [35].

Achieving an optimal solution in a mobile edge computing environment is difficult. This optimal solution in MEC refers to minimization of the total processing completion time of all tasks and to solve this optimized problem, a two-step scheme is proposed by the author in [36]. This paper particularly addressed the privacy threats or the threats or leakage of multiple private data which is given a name as a privacy disclosure. It refers to network requests sending out one or multiple types of private data. Android is a popular platform providing rich functionality to access personal sensitive data resulting in serious privacy threats. There are many mechanisms to overcome this issue but they all fail to achieve that the private information is either required for application functionality or simply it’s is transmitted to a third party. This work provided a new method for recognizing user-entered data privacy leaks in Android apps and deciding whether such leaks contradict the app's privacy policies [37]. The problem of changing orientation is a distinctive event in mobile platforms related to the graphical user interface which is commonly known as switching of the running application between portrait

and landscape layout configuration. The author of the paper proposed a framework to classify such GUI failures along with all possible faults causing those [38].

The author in [39] stated the issue of configuring mobile hybrid applications that smartphones are not secure in terms of doing this procedure as many programmers considered it as unimportant activity and non-functional as compared to the primary purpose of code, which only arises due to security breaches. A systematic literature review is being conducted in [9] and the requirement elicitation challenges particularly for mobile development are investigated along with the requirement gathering guideline. In the end, the identified challenges are classified as stakeholder-related, communication, scope, change, and so on. Similarly, a similar paper described the challenges based on testing techniques for mobile platforms and contributed testing issues for practitioners [1].

The mobile developers use async constructs from asynchronous programming which is the core part of mobile development. They do so to avoid unresponsiveness, but the issue addressed in the paper is that mobile developers are continuously unable to use these constructs resulting in memory leaks, lost results, and wasted energy. So, for its proper utilization, this paper presented a former study in which they showed how developers to retrofit asynchrony (a term in computer programming, refers to the occurrence of events independent of the main program flow and ways to deal with such events) and at the end, a refactoring tool is designed to transform the improper async constructs into the correct constructs [41].

Android platform has long been a target for privacy concerns and is most widely used popular due to the nature of open-source code. As a result, many threats arise from the Android operating system's tendency to delete sensitive data. There is a major issue with this operating system's data handling, such as when information is accessed, updated, or transferred. A data reappearance after an unintentional deletion could be a risk because Android does not provide adequate clarity on how third-party applications process user data stored on a mobile device. To solve this issue, a framework is proposed to recover a file from the disc fragments along with some mitigation strategies for data remanence [40].

As discussed above, there are numerous studies on Requirement Engineering particularly for Mobile App Development but, none of the studies have discussed the entire Requirement Engineering Process. Some showed their findings on Requirement Elicitation [9] [15], some have reported challenges particularly for disabled people [42] while some of the authors discussed the Requirement Engineering with Agile [18]. The research study being conducted is focusing on the entire Requirement Engineering process specifically for Mobile platforms.

2.6 Summary of the Chapter

A brief and thorough literature review has been conducted in this chapter. The purpose of conducting the literature review is basically to find out the research gap in already existing relevant studies. Approximately, a total of 120 studies are being reviewed, out of which 43 have been selected based on defined criteria discussed in the next chapter. The coming chapter will briefly discuss the methodologies which we have adopted during our research thesis.

CHAPTER 3

METHODOLOGY

3.1. Introduction

The previous chapter is chapter 2 was about the literature review that was conducted to find the research gap in the Requirement Engineering domain particularly for Mobile development or Mobile platforms. Now, this chapter 3 is basically about the set of methodologies adopted in our research thesis.

3.2 Overview

As mentioned above, this chapter is all about the set of methodologies or research designs that are chosen to conduct our research study. A set of multiple methodologies is adopted and the most relevant and popular protocols for each selected research design have been chosen. Firstly, a Systematic Literature Review is conducted to discover the challenges for mobile development executing the R.E. process. Then, to eliminate the similar data that is Implicit-Explicit, an approach is applied from Grounded Theory named Data Encoding Technique. Once, the results are found from the encoding technique that is the final challenge. Those challenges are passed through the next phase of Expert Review, in which the identified challenges have to be validated and evaluated by domain experts to avoid biasedness. The final validated list of challenges is then surveyed for obtaining the resolution strategies from academia.

3.3 Systematic Literature Review (SLR)

Systematic Literature Review (SLR) is the first adopted methodology, which is a well-known, formally approved, and extensively used protocol for conducting research. In addition to providing a detailed understanding of existing knowledge, it identifies the deficiencies and recent trends available for the research. The research study is conducted according to Kitchenham's guidelines [43]. The need for accompanying the Systematic Literature Review

is discovering challenges that the mobile developers face during the implementation of the entire Requirement Engineering Process. Every step given in the guide is being followed as shown below:

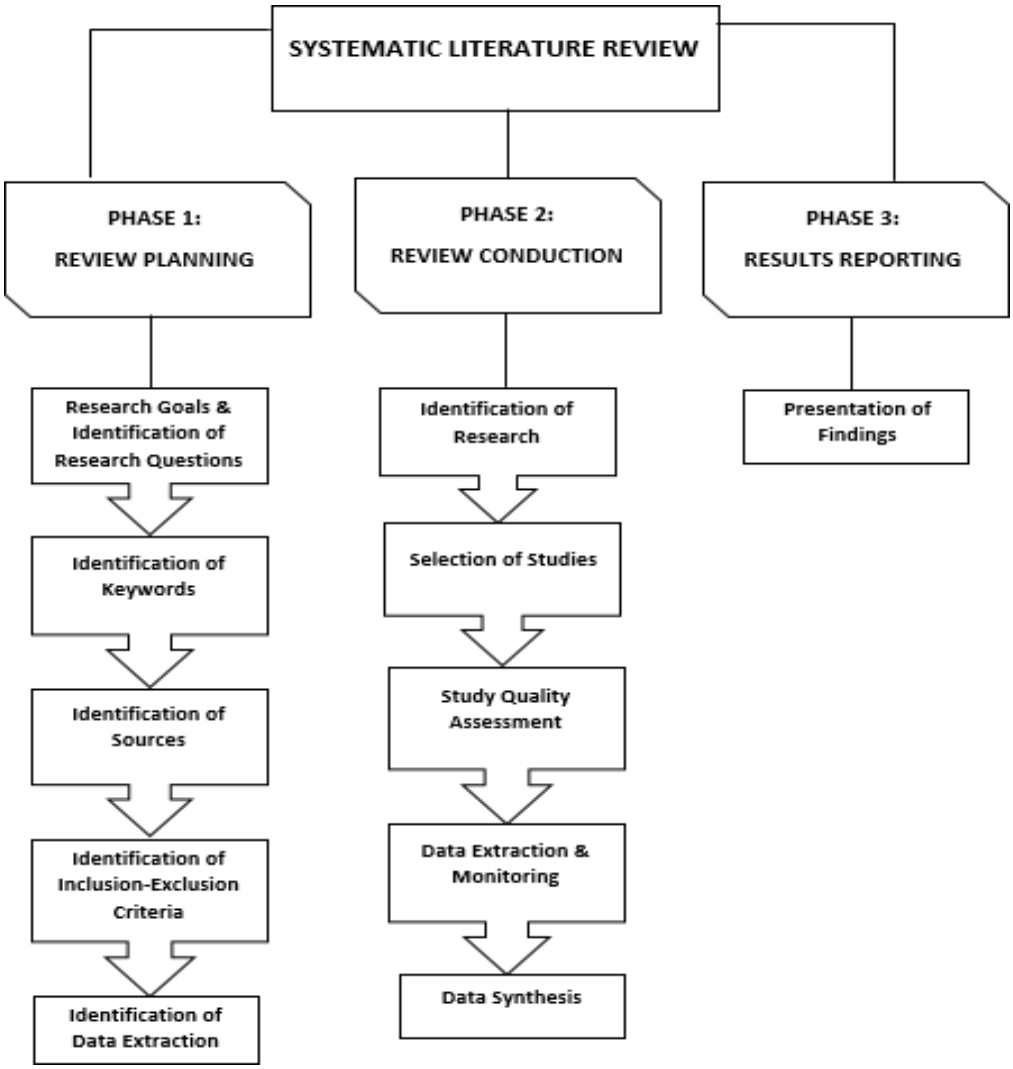

Figure 3.1: Steps of Systematic Literature Review [43]

3.2.1 Review Planning

It is the very first step for conducting a systematic literature review which starts with the planning strategy to initiate the research study including the following steps:

3.2.1.1 Background

The research study being piloted has a specific goal that is identifying the key concerns that cause problems for Mobile development during the R.E. process. Or to put

it another way, the study is designed to discover the challenges faced by software developers when implementing the R.E. process.

3.2.1.2 Research Questions

To achieve this goal, two research questions are developed for SLR conduction. Research questions and their rationales are presented in Table 3.1.

Table 3.1: Research Questions for SLR with their respective rationale

ID	Research Questions	Rationale
RQ1	What are the possible key challenges faced by the mobile developers for mobile application development during the R.E. process implementation?	This question helps us to find out the key challenges confronted by developers during the execution of the R.E. process in the mobile domain
RQ2	How many possible categories do these acknowledged challenges have?	This question aims to classify the discovered challenges based on their nature and occurrence

3.2.1.3 Strategy

To initiate the plan, firstly the resources are identified from where the primary studies along with the search terms are selected.

3.2.1.3.1 Resources

Finding a research study involves using search terms and resources. Electronic medium data sources are used to retrieve the majority of journal articles, accepted manuscripts, and conference proceedings. Books or printed sources were not chosen to gather information. A list of electronic databases from which SLR reviews publications can be found in Table 3.2 below: The majority of peer-reviewed literature on software engineering and computer science should be covered by these four electronic resources, according to our research.

Table 3.2: Electronic Data Sources

Electronic Database	URL
IEEE	https://ieeexplore.ieee.org
ScienceDirect	https://www.sciencedirect.com/
WileyOnlineLibrary	https://onlinelibrary.wiley.com/
ACM	https://dl.acm.org/

3.2.1.3.2 Search Terms

This was done by using search queries composed of keywords selected based on a set of strategies.

- The significant terms and keywords from the investigation's goal were derived.
- Mobile application development challenges are often discussed in conjunction with other domains, such as "Mobile Cloud Computing" (also known as mobile edge computing or mobile crowdsensing), the "Internet of Things" and networks (also known as network communication). There are, however, a large number of papers that focus on privacy and security. Based on the keywords and related synonyms, a total of 176 search strings was constructed. As can be seen in Table 3.3, there are several possible keywords as well as their synonyms.

Table 3.3: Major Keywords along with their Alternatives

Sr. No.	Main/ Actual Keywords	Related Synonyms
1	Requirement Engineering Process	Requirement Engineering, Requirements, Software Requirement Engineering, Software Requirement Engineering Process, Requirement Gathering, Requirement Inception, Requirement Elicitation, Requirement Prioritization, Requirement Validation, Requirement Specification
2	Mobile Application Development	Mobile Platform, Mobile Applications, Mobile Development
3	Challenges	Issues, Problems, Barriers

- This was done by using the Boolean operators "OR" and "AND".
- The following search string represents the generic search query used for the SLR:

Table 3.4: Search Strings piloted for SLR Conduction

#Try	Search Strings
Try1	((("requirement engineering") AND "mobile development") AND challenges)
Try2	((("requirement engineering") AND "mobile platform") AND challenges)
Try3	((("requirement engineering") AND "mobile applications") AND challenges)
. . .	. . .
Try100	((("requirement elicitation") AND "mobile application development") AND challenges)
Try101	((("requirement elicitation") AND "mobile development") AND issues)
Try102	((("requirement elicitation") AND "mobile platform") AND issues)
. . .	. . .
Try175	(((requirements) AND "mobile application") AND barriers)
Try176	(((requirements) AND "mobile application development") AND barriers)

The table of complete search strings has been attached in Appendix Section shown as Appendix A.

3.2.1.4 Selection Criteria (Inclusion-Exclusion Criteria)

To select and review high-quality papers most likely to be published from 2010 and 2020, a majority of the latest accepted manuscripts, mature conference proceedings, and journal articles will be used in the research study's analysis. As shown in Table 3.4, the following papers were selected for this review:

Table 3.5: Inclusion and Exclusion criteria for study selection

Inclusion Criteria	Exclusion Criteria
Material related to Requirement Engineering Process in context with Mobile Application Development published within the time frame of 10 years (2010-2020)	Studies before 2010 are not the part of Research
Published and comprehensive journal articles, conference proceedings (matured from 15th onwards), research, review articles, and (accepted manuscript)	Unaccepted and unauthenticated material, workshops, books, panels, special issue papers, special sections, editorials, discussions and tutorials are not included
The articles mainly focus on general applications, challenges, barriers, or issues related to mobile applications	Other than challenges and specific applications
Material published in the English language	Papers published other than the English Language

3.2.1.4 Study Selection Procedure

It can take a long time to conduct a systematic literature review (SLR). After a series of steps, the SLR was completed. The search is started based on the title and time frame, which was 2010-2020. When it comes to selecting the most relevant documents, a filter is used. This is followed by the categorization of selected content using keywords and inclusion criteria. The final studies after applying all filters are shown in Table 3.8 in the coming sections.

3.2.2 Review Conduction

The second phase of SLR in which the plan is being executed with the following steps shown below:

3.2.2.1 Study Quality Assessment Checklist & Procedure

Systematic Literature Review's quality assessment is a part or a phase that aims to evaluate the value of the selected studies to avoid biases. Accordingly, a set of questions in the form of questionnaires is being generated, and the respondents or candidates must answer each question for the chosen study. As shown in Table 3.6, the answers are given based on a certain scale and the questions asked. Each researcher receives the final selected studies. To collect responses from as many other researchers (Respondents or Candidates) as possible, you can choose how many studies you want to distribute.

Table 3.6: Quality Assessment Criteria adapted from [44]

No.:	Quality Assessment Questions:
1	Is there a clear statement (definition) of the aims (goals, purposes, problems, motivations, objectives, questions) of the research?
2	Is there an adequate description of the context in which the research was carried out?
3	Is the paper based on research?
4	Are references maintained accurately?
5	Does the study answer the research question defined or presents the results?
6	Is the reporting clear and coherent?
7	Are the metrics (methods, design, measures) used in the study clearly (fully) defined (description)?
8	Are the variables/metrics/methods/design used in the study adequately measured and validated (justified)?
9	Was the data analysis (collected) sufficiently rigorous?
10	Was there a control group with which to compare treatments?
11	Are the data collection methods adequately described (defined)?
12	Was the data collected in a way that addressed the research issue?
13	Was the research design appropriate to address the aims of the research?
14	Does the study provide a description and justification of the data analysis approaches?
15	Is the methodology (design) used suitable to address the stated research questions?
16	Is the study design stated clearly?
17	Are the metrics used in the study the most relevant ones for answering the research questions?
18	Is there a clear statement of findings (data) that relates to the aims of the research?
19	Do the researchers discuss any problems (limitations, threats) with the validity (reliability) of their results?
20	Is the study replicable?
21	Has sufficient data been presented to support the findings?
22	Are the findings credible?
23	Is the study of value for research or practice?
24	Are conclusions, implications for practice, and future research, reported suitably for its audience?
25	Has the approach been validated on a certain scale (either in academia or/ and industry)?

Table 3.7: Scale of Quality Assessment Checklist adapted from [43]

Answer:	Score:
Yes	1
Partially	0.5
No	0

The detail of the results of the Quality Assessment form distribution among various candidates is attached in Appendix Section shown in Appendix B. The following Table 3.8 shows the selected studies included in the research thesis to fill the research gap and to achieve the aim of the research study. The table comprises a total of 8 columns. Column 1 is defining the chosen Electronic Databases while the column next to it is explaining the total number of papers found from respective databases. Column 3 to Column 6 is showing the applied filter. At stage 1, the filter is applied based on title and keywords and a respective number of papers were found. At stage 2, the applied filter is based on abstract, which means we have selected the papers based on abstract. Stage 3 is about the repeated papers within the same databases that are included only once. At stage 4, the studies based on Abstract, Discussion, and Conclusion are selected. Finally, the second last column that is column 7 explains the studies selected based on the Quality Assessment Criteria mentioned in the previous section of the chapter. Finally, the last column is column 8 about the peer-reviewed.

Table 3.8: Study Selection Criteria

DB	Papers found	Stage 1= Title & Keywords	Stage 2= Abstract	Stage 3= Repeated	Stage 4= Abstract + Discussion + Conclusion	Quality Assessment	Peer-Reviewed
IEEE	526	230	85	70	20	10	7
Science Direct	31772	21929	2198	1361	33	15	12

ACM digital library	27531	2903	1473	94	45	28	16
Wiley Online library	14685	788	518	36	22	16	8
Total:	74514	25850	4274	1561	120	69	43

The search query was used to search four digital libraries: IEEE, ScienceDirect, Wiley Online Library, and the ACM Digital Library, all of which were searched systematically. A total of 526 papers were found based on the 2010-2020 time frame. Based on the title and keyword searches, we've reached the first stage of the process, with 230 papers in total. Then, after carefully reading the Abstracts of selected primary studies, 85 papers were selected from which 70 were repeated and only included once. There were then 20 papers selected for inclusion and exclusion based on inclusion-exclusion criteria. We've narrowed it down to about 10 papers as a result of this process. However, IEEE is only one source of information on electronics. For other electronic databases, the procedure is the same as described above. It is shown in Table 3.7 how to conduct the search and study selection for the research.

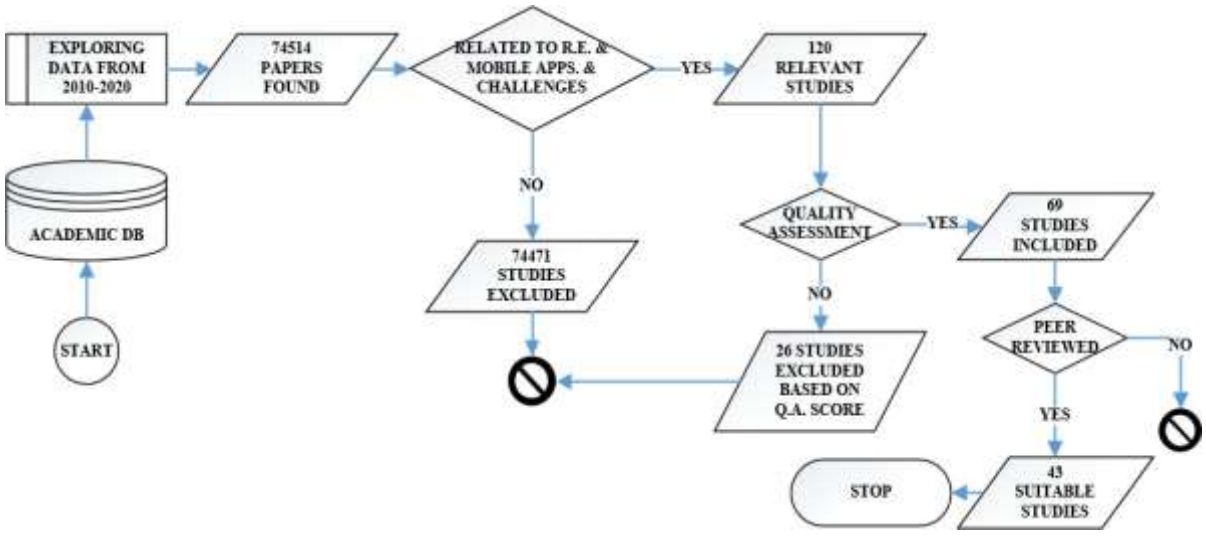

Figure 3.2: Flow Chart showing the selection criteria of the Research study

3.2.2.2 Data Extraction Strategy and Synthesis of Extracted Data

Accordingly, we have created a Data Extraction Form (Table 3.9) having 2 columns with a total of 12 entities and their associated information. The detail of extraction forms of all selected studies is mentioned in Appendix Section shown in Appendix C.

Table 3.9: Data Extraction Form

Entities	Relevant Information
Paper ID:	
Publisher:	
Title of Article:	
Type of Article:	
Year:	
Published in:	
Methodology:	
Contribution:	
Domain:	
Quality Assessment Score:	
Status of Exclusion/Inclusion:	
Answer to RQ1:	

3.2.2.3 Grounded Theory

An inductive and comparative technique for creating a theory, the grounded theory provides systematic instructions for gathering qualitative data, synthesizing it, and analyzing and conceptualizing the data. Barney G. Glaser and Anselm L. Strauss developed a grounded theory as the first formal and codified understanding of how to examine qualitative data [45]. Using the grounded theory research method of data encoding, general concepts (codes) can be extracted from specific data [46]. Aiming for a thorough and polished data extraction that is unbiased for the selected primary study is the goal of this technique in the research study being conducted. Thereafter, the fundamental constructs are decrypted. The extraction from the selected papers is performed by executing the Data Encoding techniques of Grounded Theory. The detail is mentioned in chapter 4 while, for a complete description, see Appendix Section is shown in Appendix D, E, and F respectively.

3.3 Expert Review

Results of this process include challenges for mobile development during the R.E. processes that have been identified. The challenges list needs to be evaluated by some experienced scholars who are experts in the mobile domain to validate the SLR findings. So, to achieve this goal, an expert review technique is adopted for expert review and validation [47].

The expert review is conducted to confirm the data gathering procedure and the list of challenges that can be identified. Listed below are the steps of the methodology described above:

3.4.1 Expert Identification

As a first, but crucial step, experts must be carefully identified and carefully selected before they can be confirmed to be involved in tasks (such as evaluating a challenge list and checking naming conventions of classification of those challenges). When it is applied to the research study being conducted it is needed to be very certain that the right expert must be chosen who can accurately assess the list of challenges, as well as classify the list based on the nature of the challenge and its occurrence as determined in phase 1, which is SLR conduction.

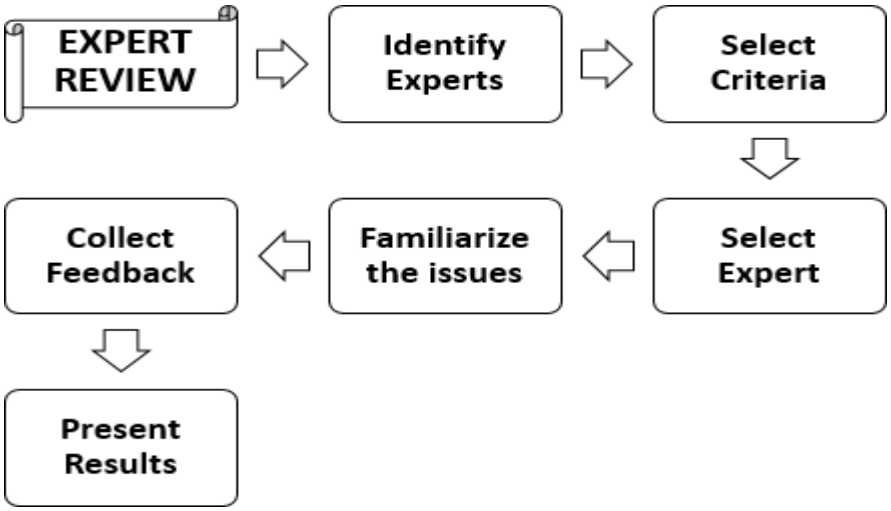

Figure 3.3: Expert Review Steps [46]

3.4.2 Selection Criteria

Under certain criteria, an expert is selected to review the tasks. Table 4.12 is showing the criteria for expert selection. The table has 2 columns. Column 1 is

showing the experience while column 2 is the required skill for evaluating the list of identified challenges.

Table 3.10: Showing the criteria for Expert selection

Selection Criteria	
Experience	**Expertise**
At least 10 Years, At most 15 Years	Must be specialized in the R.E. domain
At least 10 Years, At most 15 Years	Must be specialized in the Mobile domain

As it is clearly shown from the above table it would be necessary for the reviewer(s) in this scenario to have specialized skills in both mobile development and the R.E. domain. This means that they must have at least 10-15 years of technical and academic experience.

3.4.3 Expert Selection

To conduct an official expert opinion review, it is necessary to determine the uniqueness of a reviewer based on certain criteria mentioned in the previous step. A total of 9 experts were contacted. 4 among them showed their willingness to participate for expert opinion. Any research should have 1 to 4 experts for experts' validation [48]. The study fulfills the basic requirements of expert selection.

3.4.4 Issue Familiarization

To discuss the research problem, research purpose, and data collection with experts who are interested in reviewing and validating the SLR findings can be difficult. Move on to the next step once the problem is thoroughly explained to the interested experts

3.4.5 Collection of Responses

Collecting and quoting comments from specialists and experts is the basis of this step. The responses of experts are attached in Appendix Section shown in Appendix G.

3.4.6 Presentation of Results

Finally, once the feedback has been accumulated, the final results are presented in the form of a table or pie chart, whichever is most appropriate for the situation. The final evaluated list of challenges along with their categories is reported in a section of Expert Review in the next chapter that is Chapter 4.

Table 3.11: Personal details of Evaluators

Experts' No.	Organization' Name	Designation
Evaluator 1	National University of Modern Languages, H-9, Islamabad	Dean FE&CS/ Associate Professor
Evaluator 2	National University of Modern Languages, H-9, Islamabad	Assistant Professor/ HOD
Evaluator 3	National University of Modern Languages, H-9, Islamabad	Assistant Professor/ Mobile App Developer
Evaluator 4	University Technology Malaysia	Associate Professor

3.50 Industrial Survey

Another method that has been chosen is an industrial survey to resolve the RQ3. Since the primary goal of the research study is to propose resolution strategies for the discovered challenges in mobile application development, this method is being chosen as the research methodology. Software Engineering Institute published Kasunic [49] as a guideline for this (SEI). As it is the most widely-used guideline for conducting an effective survey in the field of software engineering, his work is followed for this reason. Figure 3.3 illustrates the steps involved in conducting a survey.

3.5.1 Research Question & Research Objective

Table 3.12: Research Question for Industrial Survey with its Respective Rationale

ID	Research Questions	Rationale
RQ3	What are the mitigation plans to overcome the discovered challenges?	This question helps in determining the most effective strategies for resolving issues in mobile platforms for the R.E. process accomplishment

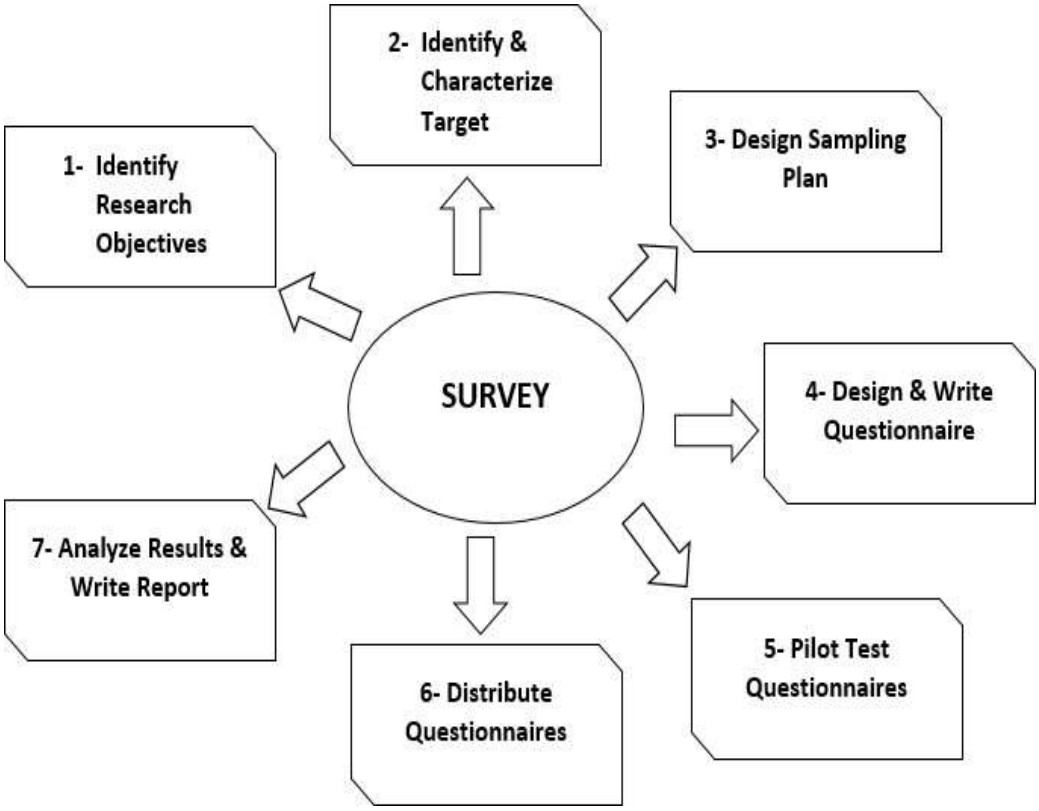

Figure 3.4: Steps Showing Survey Conduction [49]

3.5.1 Identification of the Research Objective

The very first step of conducting a survey is to recognize the problem statement of the research study along with the research aims. The main idea is to understand the problem statement that for which reason, a survey is being conducted. Once, it is clear then it will be easy to plan a strategy for achieving your aim of the research study. Based on the problem statement and research goals, one objective has to be achieved through an industrial survey that is to propose the solution for identified challenges faced by mobile developers during R.E. implementation.

3.5.2 Identification & Characterization of Target Audience

In this stage of surveying, it's crucial to identify the respondents to solve your research problem. Part of this process is making sure that the audience who are taking part in the survey must understand your questions and the terminology they acknowledge. It is again dependent on the research goal that the audience taking part in our survey are selected either based on their occupation, geography, demography, and other combination of these factors. Applying this to the research study being conducted, the respondents in this case,

chosen from industry specialized in both mobile development and R.E. domain having experience of at least 2 years or more.

3.5.3 Designing of Sampling Strategy

This step includes the sample size of the respondents taking part in the survey how small or big the size will be and is the sample size enough for responses. In other words, it is important to ensure the correctness of the sample size that must represent the entire population. A random sampling strategy is selected and a sample size of 99 people from the software industry. The sample size is calculated using Cochran's formula for sample calculation [50].

3.5.4 Designing of Questionnaires

To facilitate the analysis process of the survey results, the survey purpose and internal follow-up questions must be translated into carefully designed questionnaire items. A questionnaire is designed based on two sections. Section I is about the personal information regarding the person participating in the survey while Section II describes the list of discovered challenges to get proposed mitigation strategies from various mobile developers for each respective challenge along with its practicality level with pre-defined scale in designed survey form as shown below:

Table 3.13: Scale defining the Practicality Level for the Identified Challenge

Scale	Score
Very High	1
High	2
Moderate	3
Low	4
Very Low	5

The complete Questionnaire is attached in Appendix Section shown in Appendix H.

3.5.5 Pilot Test Questionnaires

It is basically to test the designed questionnaires from a small sample from the target audience. Its main purpose is to remove the bugs which leads to further improvement.

Based on the calculated sample size of 99 from the entire population of 200, the sample size for the pilot study was calculated as 41 [51].

3.5.6 Distribution of Questionnaires

The designed and corrected questionnaires are distributed to the audience for responses collection defined in a sampling plan stage. The survey questionnaire is being distributed online firstly via emails and LinkedIn Corporation. The online procedure was slow, then some software companies for responses collection were being visited.

3.5.7 Analyzing the Final Results & Writing a Report

Once the responses had been collected and finalized, then the appropriate method is chosen to represent the analyzed findings of the survey. The representation of survey findings can be reported and based on that, conclusions had been made. The detail is discussed in Chapter 5 and the complete survey form along with mitigation strategies is attached in Appendix Section shown in Appendix H.

3.6 Phases of Research Study:

The section of this chapter addresses the flow of the research study that has been conducted which shows a complete procedure of conducting the research study. A research study that has been conducted focused on three research questions that are challenges identification, classification of identified challenges, and solutions to overcome those identified challenges. Based on the research questions, a flow diagram is constructed shown below:

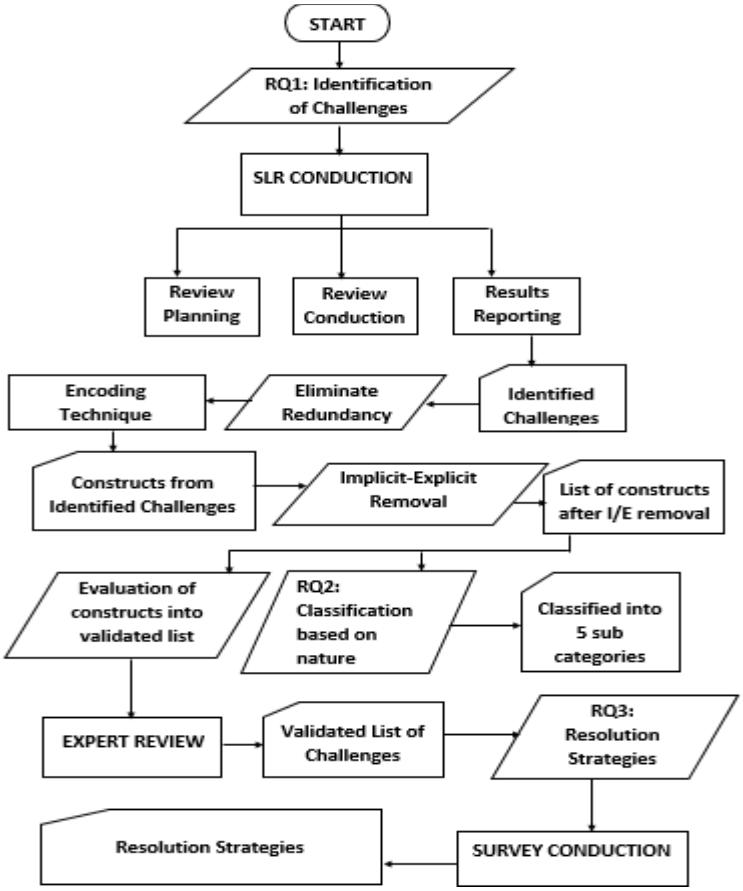

Figure 3.5: Flow Diagram of Research Study

The above figure is showing the flow of research study starting from the RQ1 that is to identify the challenges which the mobile developers face during the implementation of the Requirement Engineering Process. To answer the RQ1, a Systematic Literature Review [43] is conducted having three phases. The output of phase 3 is the results reporting is the identification of the challenges. The identified challenges are passed through a phase of Redundancy–Elimination. The duplication of challenges is being removed via using the Encoding Technique from Grounded Theory [45]. The output of the Redundancy phase is the relevant constructs from the identified challenges. The constructs are then passed through the phase of Implicit- Explicit Removal. The list after I/E removal is passed through two phases of Evaluation of constructs into validated list and RQ2 that is Classification of identified challenges based on nature respectively. The output from the RQ2 is the Classification of identified challenges into 5 Sub-Categories, while, the phases of evaluation are passed through Expert Opinion conduction where the list of constructs is being evaluated by multiple experts based on defined criteria of Expert Opinion [48]. The output from the Expert Review conduction is the validated list of challenges. The final list of validated challenges is passed through a phase of RQ3 that is to

propose Mitigation Strategies. For this purpose, Industrial Survey is being conducted to obtain mitigation strategies. The output of the Industrial Survey is the Resolution Strategies, the answer to the RQ3.

3.7 Summary of the Chapter

This chapter briefly explains the set of methodologies adopted during the research study that is a Systematic Literature Review [43] for challenges identification, Expert Review [47] to validate the identified list, and an Industrial Survey [49] for obtaining the mitigation plans to overcome the identified challenges. At last, the flow of the research study is explained diagrammatically. The next chapter is about the Requirement Engineering Challenges for Mobile Application Development.

CHAPTER 4

Challenges in Requirement Engineering Process for Mobile Application Development

4.1 Introduction

The following chapter discussed the findings of each research design that has been conducted to achieve the research goals. Three research questions have been designed based on the research study aim, so according to that, an SLR output leads to achieving discovered challenges. Then, the discovered challenges are transformed into meaningful constructs via the Grounded Theory Technique that is Data Encoding. The above output is passed through the next phase for validation of identified meaningful constructs into challenges that are in the listed form. When the responses are collected and improvements suggested by experts are completed, then an industrial survey is accompanied to achieve the last purpose of research which is to propose mitigation strategies to overcome the challenges.

4.1 SLR Findings

Phase 1 of the research study is the Systematic Literature Review conduction. To achieve the aim of RQ1 that is to discover the challenges faced by the mobile developers during the execution of the Requirement Engineering Process. So, the findings obtained in phase 1 are explained below.

4.1.1 Distribution of Studies based on Years

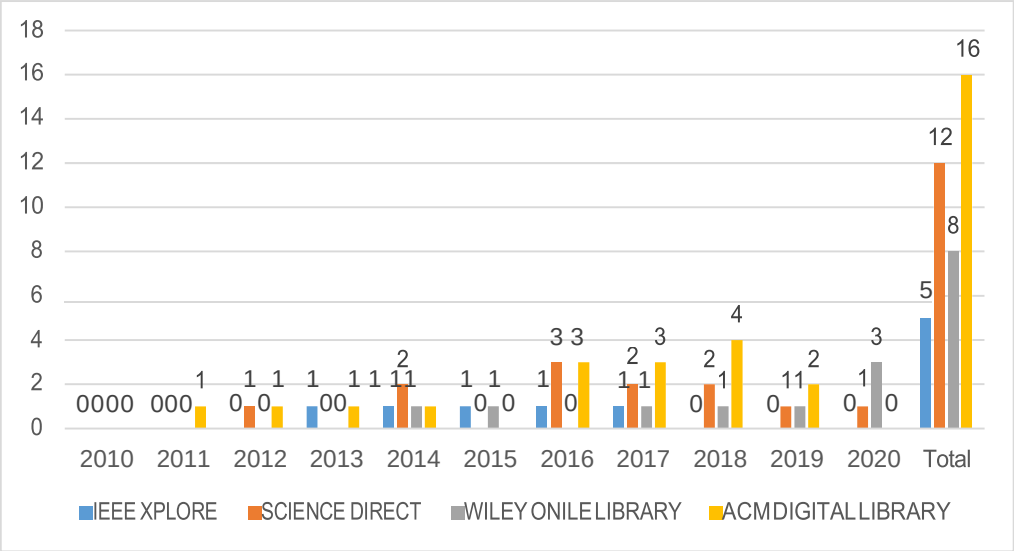

Figure 4.1: Graph Showing included studies per publication year

The above graph is showing the included studies per publication year from 2010 to 2020. On the x-axis, the year from 2010 to 2020 and total has been plotted while the y-axis is showing the number of studies. The sequence of electronic databases is as follows: IEEE with blue, Science Direct with orange, Wiley Online Library with grey, and ACM Digital Library with yellow. It is clearly shown that in the year 2010, no relevant paper is published from any database. In the year 2011, only 1 relevant study is published from ACM. 1 paper from Science Direct and 1 from ACM is published in 2012. Similarly, the sequence of a total number of related published studies is 5, 12, 8, and 16 from IEEE, Science Direct, Wiley Online Library, and ACM Digital Library respectively.

4.1.2 Distribution on basis of Type of Research Studies

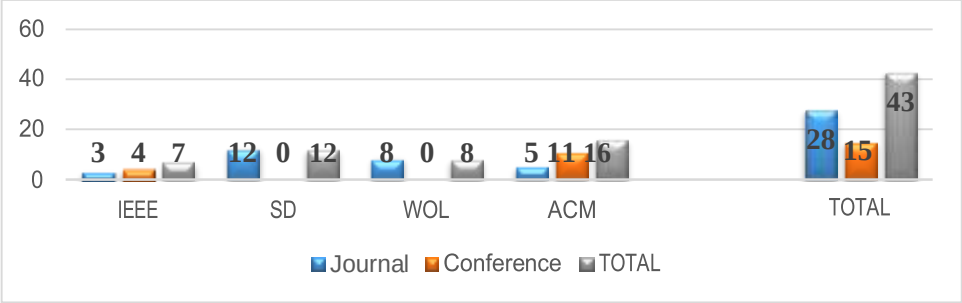

Figure 4.2: Graph showing the Distribution of Research Studies based on Type of Papers

The graph drawn above is the distribution of research studies based on the type of papers. On the x-axis, the type of papers is being plotted while, on the y-axis, the number of studies is being plotted. The blue shade is showing journal, orange shade is showing conference and grey is showing total. The sequence of databases is plotted as IEEE, Science Direct, Wiley Online Library, and ACM Digital Library respectively. After that, a total of found studies is plotted. A total of 7 papers are found from IEEE of which 3 are journals while 4 from the conference are there. From Science Direct, all papers found are a journal that is 12, same is the case with Wiley Online Library, and all papers published are a journal that is 8. From ACM Digital Library, 5 papers from journal type and 11 from conference-type are published. In total, 28 papers were published in a journal while 15 were published in a conference.

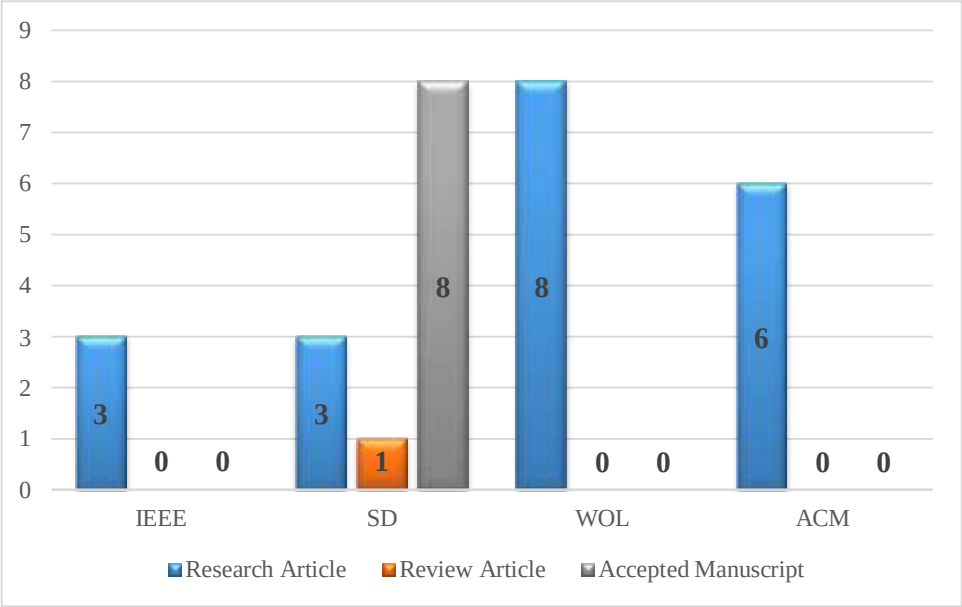

Figure 4.3: Graph showing the Distribution of Research Studies based on Journal Paper Type

The graph drawn above is showing the papers published based on the type of journal paper. On the x-axis, the type as Research article with blue shade, Review article with orange shade, and Accepted Manuscript with grey shade are plotted while, on the y-axis, the number of studies is plotted. A total of 3 papers published were research articles from IEEE. From Science Direct, 3 papers are research articles, 1 is a review article, and 8 are accepted manuscripts. 8 papers published were research category from Wiley Online Library while, from ACM Digital Library, 6 papers published were research articles.

The same graph is presented in the tabulated form below comprises 4 columns with DB, Review Article, Research Article, and Accepted Manuscript respectively.

Table 4.1: Distribution of Research Studies based on Type of Journal Papers

DB	Review Article	Research Article	Accepted Manuscript
IEEE	-	**X2, X4, X5**	-
SD	**S6**	**S7, S9, S12**	**S1-S5, S8, S10, S11**
WOL	-	**W1-W8**	-
ACM	-	**A1, A3-A5, A11, A16**	-

Only 3 research articles (X2, X4, X5) were published in IEEE, 1 (S6) review, 3 (S7, S9, S12) research articles, and 8 manuscripts (S1-S5, S8, S10-S11) published in Science Direct, 8 research papers (W1-W8) published in Wiley Online Library while, 6 research papers (A2, A3-A5, A11, A16) published in ACM Digital Library.

4.1.3 List of Conferences

Table 4.2 shown below is the distribution of included studies based on Conferences having 2 columns with entities titled Included Research Studies and Name of Conferences.

Table 4.2: Distribution of Included Studies based on Conferences

Included Research Studies	Name of Conferences
X1	Proceedings - International Conference on Network Protocols, ICNP
X3	Proceedings - Asia-Pacific Software Engineering Conference, APSEC
X6	Proceedings - International Conference on Computer Communications and Networks, ICCCN
X7	Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015
A2	Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017
A5	MobileHCI 2014 - Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services
A6	ACM International Conference Proceeding Series

A7	Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017
A8	Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2019
A9	Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM
A10	Proceedings of the 40th International Conference on Software Engineering, ICSE
A12	ACM International Conference Proceeding Series
A13	ASE 2016 - Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering
A14	Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015
A15	ASE 2018 - Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering

4.1.4 List of Journals

Table 4.3 is showing the distribution of included studies based on journals. The table comprises 2 columns with Included Research Studies and Name of Journals respectively.

Table 4.3: Distribution of Included Studies based on Journals

Included Research Studies	Name of Journals
X2	IEEE Journal on Selected Areas in Communications
X4	IEEE Access
X5	IEEE Access
S1	Future Generation Computer Systems
S2	Journal of Network and Computer Applications
S3	Future Generation Computer Systems
S4	Future Generation Computer Systems
S5	Computers and Security
S6	Computer Communications
S7	Computer Standards and Interfaces

S8	Journal of Systems and Software
S9	Computer Standards and Interfaces
S10	Pervasive and Mobile Computing
S11	Sustainable Computing: Informatics and Systems
S12	Applied Soft Computing Journal
W1	International Journal of Communication Systems
W2	Software - Practice and Experience
W3	Transactions on emerging telecommunications technologies
W4	Journal of Software: Evolution and Process
W5	Software - Practice and Experience
W6	Software - Practice and Experience
W7	Software - Practice and Experience
W8	Software Testing Verification and Reliability
A1	ACM SIGSOFT Software Engineering Notes
A3	ACM Transactions on Embedded Computing Systems
A4	ACM Transactions on Internet Technology
A11	Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Article No.: 28
A16	ACM Transactions on Embedded Computing Systems

4.1.5 Distribution based on Methodology

Table 4.3 shown below is the distribution of research studies based on methodology. The table is composed of 3 columns. Column 1 is Methodology Type, column 2 is showing Papers' ID while column 3 is the Reference Number.

Table 4.4: Distribution of Included Studies based on Type of Methodology

Methodology Type	Papers' ID	Reference Number
Simulation	**X1, X2, X5, X6, X7, S1, S2, S3, S4, S5, S9, S11, S12, W1, W3, W4, W5, W6, W7, W8, A1, A2, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15**	[52], [53], [36], [54], [41], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [38], [69], [70], [71], [72], [39], [73], [74], [75], [37], [35], [76], [77], [78], [79]

Case Study	**X3, W4, A3, A14**	[80], [65], [81], [78]
Systematic Literature Review	**X4,W2**	[9], [34]
Experimentation	**X5, X7, S1, S2, S3, S4, S5, S10, S11, S12, W1, W3, W5, W6, W7, W8, A3, A4, A5, A6, A9, A10, A11, A12, A13, A14, A16**	[36], [41], [55], [56], [57],[58], [59], [82], [61], [62], [63],[64], [66], [67], [68], [38], [81],[72], [39], [37], [35], [76], [77],[78], [40]
Empirical Study/ Empirical Evaluation	**X7, S10, W4, A13, A15**	[41], [82], [65], [77], [79]
Literature Review	**S6, S7, A16**	[83], [84], [40]
Systematic Mapping Study	**S8**	[1]
Survey	**W2**	[34]

A total of 34 studies were found under Simulation methodology, 4 from Case Study, 2 from SLR, 27 from Experimentation, 5 from Empirical Evaluation, 3 from Simple Literature Review, 1 from Systematic Mapping Study, and 1 from Survey is found. The Table of included studies along with their ID is attached in Appendix Section shown in Appendix J.

4.1.6 Contribution Facets:

Table 4.4 is showing the distribution of included studies based on the contribution given by the authors of respective studies. The table shown below is having 4 columns. Column 1 is the Type of Contribution Facet, column 2 is showing the Paper ID, column 3 is the Description against the respective Contribution made by the authors in the included studies, while, column 4 is showing the Reference Number.

Table 4.5: Distribution of Included Studies based on Contribution Facets

Type of Contribution Facet	Paper ID	Description	Reference Number
Framework	**X1**	We propose a framework of anonymous routing (FAR) for DTNs, which subsumes all the aforementioned protocols.	[52]

	X2	A decision-making framework for Resource allocation to mobile applications, revenue management, and co-operation among service providers	[53]
	S12	A framework is presented for generating models to make automatic decisions on the offloading of mobile applications using genetic programming (GP)	[62]
	S10	Evaluate and compare the current cross-platform framework for mobile applications based on energy consumption	[82]
	W5	A high-level extensible framework is presented for the evaluation of any CPDT	[66]
	W7	A framework named VAnDroid based on MDRE is presented, identifying the security risks & vulnerabilities related to android application communication model	[68]
	W8	A framework named DOC is proposed for classifying GUI failures with the investigation of its key features along with possible faults causing them	[38]
	A3	A framework is proposed that accurately estimates the remaining battery time of applications at runtime	[81]
	A5	An extension to the popular Calabash testing framework is proposed allowing for test automation for gesture-based mobile application	[72]
	A16	A framework named DataRaider to recover files from disc fragments are designed and implemented	[40]
Approach	**X3**	A model-driven approach for automated generation of mobile apps for multiple platforms	[80]
	S4	PATAS is a novel approach proposed for high performance Android system by using pre-cache technologies	[58]
	W2	The GT-based CO approaches in the MEC environment are presented in the form of classical taxonomy with some open issues	[34]
	W4	A model-based approach for mobile platforms to overcome the the problem of automation of performance testing	[65]

	W6	An edge server provisioning approach using LSTM prediction to estimate the future workload & RL technique to make the appropriate scaling decision	[67]
	A1	An approach for testing the software running on mobile terminals by using CC environment	[69]
	A10	Proposed a novel approach to protect privacy policy violations due to leak of user input data	[37]
	A11	A novel fine-grained approach named LeakDoctor is proposed which aims to detect each privacy disclosure automatically	[35]
	A14	A tooled approach named PAPRIKA is presented to assess the software quality of mobile application considering anti-patterns along with their detection	[78]
Challenges/ Issues	**X4**	Challenges on Elicitation techniques for mobile development	[9]
	S8	Specific testing issues for practitioners are identified	[84]
Scheme	**X5**	In this paper, a radio resource and computing resource allocation the scheme is proposed to minimize the total processing completion time of all the tasks.	[36]
Process	**X6**	Semi Markov Decision process-based resource allocation (SMDP-RAS) algorithm for secure MCC systems is proposed	[54]
Algorithm	**S2**	A set of two algorithms is proposed to find the optimal or the near-optimal partitioning of an application considering application's Size.	[56]
	S11	Proposed a multi-site application partitioning algorithm named Cyclic random movement (CRM) based on genetic algorithm (CRMGA)	[61]
	W1	This paper presented an optimized genetic-based decision algorithm for a multi-site CO problem called GAMCO, to find the best possible solution on time.	[63]
	A4	An online real-time scheduling algorithm for MCOSP on the basis of rent/buy problem.	[71]
Tool	**X7**	A refactoring tool named ASYNCDROID is built which enables Android developers to transform existing improperly-used async Constructs into correct constructs.	[41]

	A6	A tool named CORDOVACONFIG is designed and built for configuring mobile hybrid apps.	[39]
	A7	A practical tool named CRASHSCOPE is designed and implemented that automatically discovers, reports & reproduces crashes for Android applications	[73]
	A9	An energy emulation tool named WattsOn is built to estimate the energy consumption of app during development	[75]
	A15	A new tool named IctApiFinder is developed to detect the incompatible API usages in android applications	[79]
System	**S1**	Proposed context-sensitive offloading system (CSOS) that takes the advantage of ML reasoning techniques and robust profiling system to provide offloading decisions with a high level of accuracy.	[55]
	A8	A system providing reliable energy measurement for mobile applications without requiring a complex setup named EMaaS	[74]
	A12	A system is proposed to distribute and apply third-party security patches for android	[76]
Model	**S5**	A context-dependent computation-offloading model for MCC is proposed, which is based on application segments packed into autonomous agents	[59]
Ecosystem	**S3**	An ecosystem for mobile crowdsensing which relies on the CloUd- based PUblish/Subscribe middleware (CUPUS) to acquire sensor data from mobile devices in a flexible and energy-efficient manner and to perform near real-time processing of Big Data Streams.	[57]
Guideline	**S6**	A guideline on Computation Offloading in context of heterogeneous Cloud Computing is contributed	[83]
Standard	**S7**	Standard practice for developing secure mobile applications is presented	[84]
Platform	**S9**	A web based platform is proposed providing various services such as offline mode/ services, content adaptation service & synchronization services	[60]

Method	**W3**	A novel terminal energy-efficient scheduling method (AGILE for short) is presented to make decisions about mobile applications' tasks executed by mobile devices or cloud servers	[64]
Technique	**A2**	A technique enabling android users to protect their devices from multiple ICC vulnerabilities named SEALANT	[70]
	A13	Android	[77]
Flaws	**A16**	Some flaws considering 3 typical scenarios of the android system is presented	[40]
Strategies	**A16**	Also, some strategies or mitigation plans were proposed for data remanence issues along with some suggestions on data protection in android systems	[40]

10 studies proposed a Framework as contributions, 9 studies gave Approaches, 2 studies gave Challenges, and 2 studies, each contributed its work as Scheme and Process respectively. 4 studies presented an algorithm while 5 studies developed Tool as a contribution. 3 studies gave contributions based on System while only one study contributed as Model, Ecosystem, Guideline, Standard, Platform and Method respectively. 2 studies presented Techniques while only one study presented Flaws and Strategies respectively.

4.1.7 Distribution of Challenges based on Sub-Categories:

Table 4.5 shown below is the distribution of challenges based on sub-categories. The table is composed of 5 columns with the following entities as No., Sub-Category, Paper ID, Challenge #, and Reference #.

Table 4.6: Distribution of Challenges based on Sub-Categories

R.E. Challenges faced by Mobile Developers:				
No.	**Sub-Category:**	**Paper ID:**	**Challenge #:**	**Reference #:**
1	Communication-related	X1, X4	1-4	[52], [9]

2	Requirement-related	X2, X4, S1, S2, S3, S11, S12, W1, W2, W8, A1, A4, A7, A14	5-19	[53], [9], [55], [56], [57], [61], [62], [63], [38], [69], [71], [73], [78]
3	Resource-related	X2, X3, X5, S9, S10, W3, W4, W5, W6, A1, A3, A4, A8, A9, A13, A15	20-29	[53], [80], [36], [60], [82], [64], [65], [66], [67], [69], [81], [71], [74], [75], [77], [79]
4	Security & Privacy-related	X7, S5, S6, W7, A2, A5, A6, A10, A11, A12, A16	30-38	[41], [59], [83], [68], [70], [72], [39], [37], [35], [76], [40]
5	Stakeholder-related	X2, X4, S4, S7, S8, A4	39-46	[53], [9], [58], [83], [84], [1], [71]

The sub-categories are sequence as Communication-related, Requirement-related, Resource-related, Security & Privacy-related, and Stakeholder-related, 4 challenges identified from 2 studies came under the communication category, 15 challenges from 13 studies came under the requirement category, 10 challenges from 16 studies came under the resource category, 9 challenges were identified from 11 studies came under security & privacy category while 9 challenges from 6 studies were identified which came under stakeholder category.

4.2 Findings from Grounded Theory

The tables shown below are an example of the discovered challenges identified from selected primary studies as a result of a systematic literature review. However, multiple authors presented the same idea with different names while few studies have the same meaning and same name. So, it is not needed to add the same studies twice or thrice and so on. For that purpose, the duplicated data has to be removed and eliminated. The concept from Grounded theory is going to help in eliminating the duplication from SLR conduction. The data encoding technique is applied shown in table 4.7 and selected the concept Implicit-Explicit removal so that useful constructs can be obtained as a result shown in Table 4.8. While Table 4.8 shows the encoded challenges other than Implicit-Explicit removal.

The table shown below comprises 4 columns titled Paper Id, Paper Statement, Respective Code, and Data Encoding respectively. Column 1 is showing the Paper ID given to included respective studies. Column 2 is showing the Paper Statement which is selected to be encoded

For construct extraction. Column 3 is the Respective Code given to the selected line from the study while column 4 is the Data Encoding or the name given to the extracted construct. Table 4.6 shown below is the implementation of the Encoding technique:

Table 4.7: Example of Data Encoding

Paper ID	Paper Statement	Respective Code	Data Encoding
X3	"In addition, mobile applications also have to support multiple platforms, as an application written for one platform (e.g., Android) cannot run on another platform (e.g., Windows Phone)."	X3L4	Multiplatform Support
X10	"However, developers can still use the inappropriate async constructs, which result in memory leaks, lost results, and wasted energy. Fortunately, refactoring tools can eliminate these problems by transforming async code to use the appropriate constructs."	X10Pa1L2, X10Pa1L4	Memory leaks & energy wastage **OR** Inappropriate async constructs **OR** Transformation of async code **OR** Use of appropriate constructs **OR** Asynchrony retrofitting
S9	"The abundance of mobile software applications (apps) has created a security challenge. The lack of development standards and best practices expose the Mobile device to potential attacks."	S9L1, S9L3	Security challenge in mobile apps **OR** Lack of development standards/ practices

Examples of encoded and named constructs from existing studies can be found in the examples provided above. Because Android applications can't be run on Apple or Windows phones, the first example with paper ID X3 illustrates the need for mobile applications to support multiple platforms through data encoding. "Multiplatform Support" is the term used for it. Asynchrony Retrofitting is a Data Encoding technique used to describe the incorrect use of async constructs in the second example. "Lack of development standards" is described in example 3 (S9), where the paper stated that the proliferation of software applications has posed a security risk because these apps are commonly available with little or no cost across multiple platforms which are developed by inexperienced programmers. The complete table showing the implementation of Data Encoding is attached in Appendix Section shown in Appendix K.

Explicit and implicit removal has been applied to the extracted data to make it more uniform and smooth. Multiple authors discussing the same issue with the same name convention are known as explicit removal, while multiple papers discussing the same issue but with different names, conventions are known as implicit removal. Hence, in this case, it's important to think about the issue only once, eliminating the need for repetitive thinking. The table shown below is composed of 3 columns in total. Column 1 is the Paper Id given to included studies. Column 2 is the extracted construct from included studies while column 3 is the name given to extracted construct after Implicit/explicit Removal. In Table 4.7, an example of this scenario is explained. The complete detail of Implicit-#explicit Removal is attached in Appendix Section shown in Appendix L.

Table 4.8: Example showing Implicit-Explicit Removal

Paper ID	Constructs	Implicit-Explicit Removal
X3, S12, S14, W6	Multiple platform support, Cross-platform incompatibility, Fragmentation of mobile devices, Fragmentation problems for developers, Availability of multiple platforms, Fragmentation, Restriction of using specific platforms & devices	Fragmentation
S1, S15, W2	The offloading decision, Inaccurate decisions, Dynamic nature of MCC environment,	Offloading Decision Problem

	CO decision Prolonging battery life, Reduction of energy consumption, Complexity in the adoption of MCC, Multisite partitioning, Offloading of computation-intensive tasks, The multisite offloading problem, Offloading decisions in multisite context, NP-complete problem	

Example 1: Multiple authors have focused on the same issue of platform compatibility and have used different names for a stated issue to differentiate it. As a result, we've eliminated the redundancy by using the implicit method. Similarly, in example 2, we used an explicit method to accomplish this, as other writers also followed this convention with the same problem.

The table shown below is the example showing other encoding unique challenges which have 3 columns with Paper Id, Identified Challenges and Final Selected name entities respectively. Column 1 is the Paper Id given to the included primary studies, column 2 is showing the Identified Challenges while column 3 is showing the Final Selected challenge from Identified Challenges column. The complete table is attached in the Appendix section shown in Appendix M.

Table 4.9: Example showing other Encoded Challenges

Paper ID	Identified Challenges	Final Selected
X1	Security & privacy in terms of DTNs or Anonymous communication	Anonymous communication
X10	Memory leaks & energy wastage or Inappropriate async constructs or Transformation of async code or Use of appropriate constructs or Lack of Asynchrony retrofitting	Lack of Asynchrony retrofitting
S5	Complexity in achieving efficiency for android application model or Lack of Requirement task efficiency & responsiveness	Lack of Requirement task efficiency & responsiveness

S7	The problem in adoption of MCC or Tampering during offloading data or Tamper attacks or Tamper Detection	Tampering during offloading data
W16	GUI failures or Impact of mobile events on GUI or Screen orientation changes problem or Changing orientation of app	Changing the orientation of the app
A26	Effect of design choices on software quality & performance or Tracking of antipatterns or Poor design choices or Frequent changing requirements	Frequent changing requirements
A28	Sensitive data deletion or Problem of data-erasure or Data-Erasure	Data-Erasure

4.4 Findings from Expert Review

Based on standard criteria [48], 4 experts were chosen for evaluating the identified challenges in the Expert Review phase. All of them gave some suggestions which are presented in a tabulated form below in Table 4.9. The suggestion table is composed of 4 columns. The columns are named Experts, Comments, Action Taken and Reference respectively.

Table 4.10: Showing Experts' Opinion for further Improvement

Experts	Comments	Action taken	Reference
Evaluator 1	The suggestion is to exclude some studies as these were only related to mobile development lacking R.E. process.	Excluded 26 studies	X5, X6, X7, S3, S6, S12, W3, W7, W8, W9, W10, W11, W12, W13, A4, A6, A7, A9, A14, A15, A18, A19, A20, A21,A23, A25
Evaluator 2	The suggestion is to consider only one category that is to only classify based on nature	Classified the challenges based on nature only having 5 sub-categories as Communication, Requirement, Resource, Privacy & Security, and Stakeholders	Column 1 is Category 1 is reconsider, redefined, and reclassified (based on nature) and column 2 that is Category 2 (based on occurrence) is eliminated

Evaluator 3	A suggestion is to reconsider the naming conventions of identified challenges	Resource Allocation Problem to Lack of appropriate Resource Allocation	X2 (study 2 from IEEE Xplore), X9 (study 9 from IEEE Xplore), W10 (study 10 from Wiley Online Library)
		Fragmentation to Platform Incompatibility	X3 (study 3 from IEEE Xplore), S11, S13 (study 11 and 13from Science Direct respectively), W6 (study 6 from Wiley Online Library)
		Asynchrony Retrofitting to Lack of Asynchrony retrofitting	X10 (study 10 from IEEE Xplore)
		Offloading Decision Problems to Lack of consideration of user & application requirements for offloading decision making	S1, S15 (study 1 & 15 from Science Direct respectively), W2 (study 2 from Wiley Online Library)
		Multi-site partitioning problem to Incorrect Requirement Partitioning	S2, S14 (study 2 & 14 from Science Direct respectively), W1 (study 1 from Wiley Online Library)
		Extraction of Useful Information to Lack of Useful Information Extraction	S4 (study 4 from Science Direct)
		Inefficient Execution Model Problem to Lack of Requirement task efficiency & responsiveness	S5 (study 5 from Science Direct)
		Tampering to Tampering during offloading data	S7 (study 7 from Science Direct)

		Lack of development standards to Lack of development standards & Practices Knowledge	S9 (Study 9 from Science Direct)
		Task Scheduling to Inaccurate Task Scheduling	W4 (study 4 from Wiley Online Library)
		Performance Variation to Limited Resources	W5 (study 5 from Wiley Online Library)
		Problems related to Android Communication Model to Lack of identification of risky actions & Vulnerabilities	W14 (study 14 from Wiley Online Library), A2 (study 2 from ACM Digital Library)
		Impact of Mobile Specific Events on GUI to change the orientation of App	W15 (study 15 from Wiley Online Library)
		Estimation of Battery life to Incorrect Estimation of Battery Life	A3 (study 3 from ACM Digital Library)
		Gesture-based Interaction Constraints to Inconsistent & Inefficient Testing	A8 (study 8 from ACM Digital Library)
		Problems with Configuration of Mobile Hybrid Apps to Lack of Configuration of Mobile Hybrid Apps	A10 (study 10 from ACM Digital Library)
		Energy Measurement to Lack of accurate quantification about the consumption of energy by the app	A12, A13 (study 12 & 13 from ACM Digital Library respectively)
		Privacy Disclosure to Unclear requirement for app functionality towards privacy threats	A16, A17 (study 16 & 17 from ACM Digital Library respectively)
		Patching to Patching for updation, correction or improvement	A22 (study 22 from ACM Digital Library)
		FIC Issues to Compatibility across various OS versions	A24, A27 (study 24 & 27 from ACM Digital Library respectively)
		Poor Design Choices to Frequent changing requirements	A26 (study 26 from ACM Digital Library)

		The Problem of Data Erasure to Data Erasure	A28 (study 28 from ACM Digital Library)
Evaluator 4	Suggestion is to decompose the single challenge to multiple	Requirement gathering challenges to: • Incomplete requirement gathering • Lack of accurate requirement prioritization • Lack of requirement effective articulation • Unawareness of need • Lack of effective verbal & presentation skills • Cultural & language barrier • Lack of domain knowledge • Unstable requirements • Change of user need & understanding • Requirement over-scoping or ill-defined scope • Ambiguities among stakeholders • Intra-group conflicts • Communication participation	X4 (study 4 from IEEE Xplore)
		Optimization problem to: • Lack of resource optimization • Inefficient requirement completion time • Inefficient response time • Energy inefficient	X8 (study 8 from IEEE Xplore) & A5 (study 5 from ACM)

		Emulator related constraints to: • Limited computational resources • Diversity of mobile Surroundings	A1 (study 1 from ACM)
		Challenges in mobile app testing to: • Event-driven structure • Complex contextual features	A11 (study 11 from ACM)

Evaluator 1 suggested excluding some of the studies as those were not related to the Requirement Engineering process. So, as per the experts' suggestion, studies with paper Id X5, X6, X7, S3, S6, S12, W3, W7, W8, W9, W10, W11, W12, W13, A4, A6, A7, A9, A14, A15, A18, A19, A20, A21, A23, A25 were being excluded. The suggestion from Expert 2 was to change the categorization from occurrence and nature to only from nature as Communication-related, Requirement-related, Resource-related, and Security & Privacy–related and Stakeholders-related. Expert 3 suggested to reconsider the naming convention so, as per suggestion, the naming conventions of the following challenges as Resource Allocation Problem to Lack of appropriate Resource Allocation, Fragmentation to Platform Incompatibility, Asynchrony Retrofitting to Lack of Asynchrony retrofitting, Offloading Decision Problems to Lack of consideration of user & application requirements for offloading decision making, Multi-site partitioning problem to Incorrect Requirement Partitioning, Extraction of Useful Information to Lack of Useful Information Extraction, Inefficient Execution Model Problem to Lack of Requirement task efficiency & responsiveness, Tampering to Tampering during offloading data, Lack of development standards to Lack of development standards & Practices Knowledge, Task Scheduling to Inaccurate Task Scheduling, Performance Variation to Limited Resources, Problems related to Android Communication Model to Lack of identification of risky actions & vulnerabilities, Impact of Mobile Specific Events on GUI to Changing orientation of App, Estimation of Battery life to Incorrect Estimation of Battery Life, Gesture-based Interaction Constraints to Inconsistent & Inefficient Testing, Problems with Configuration of Mobile Hybrid Apps to Lack of Configuration of Mobile Hybrid Apps, Energy Measurement to Lack of accurate quantification

about the consumption of energy by the app, Privacy Disclosure to Unclear requirement for app functionality towards privacy threats, Patching to Patching for updation, correction or improvement, FIC Issues to Compatibility across various OS versions, Poor Design Choices to Frequent changing requirements and The Problem of Data Erasure to Data Erasure. While, the last expert advised to decompose the single challenge to multiple as Requirement gathering challenges to Incomplete requirement gathering, Lack of accurate requirement prioritization, Lack of requirement effective articulation, Unawareness of need, Lack of effective verbal & presentation skills, Cultural & language barrier, lack of domain knowledge, Unstable requirements, Change of user need & understanding, Requirement over-scoping or ill-defined scope, Ambiguities among stakeholders, Intra-group conflicts and Communication participation. Optimization problem to Lack of resource optimization, inefficient requirement completion time, inefficient response time, and energy inefficiency. Emulator-related constraints to Limited computational resources and Diversity of mobile surroundings. Challenges in mobile app testing to Event-driven structure and Complex contextual features

So, considering suggestions given by each evaluator, we have finalized our list of Challenges as shown in Table 4.10. The final list of challenges is tabulated below comprises 4 columns as Sr. No., Paper ID, Sub-Category Name, and Respective Challenge(s).

Table 4.11: Final list of challenges along with categorization after Experts' Suggestions

Sr. No.	Paper ID	Sub-Category Name	Respective Challenge(s)
1	X1	Communication	Anonymous Communication
2	X4		Lack of requirement effective articulation
3	X4		Lack of Verbal & Presentation skills
4	X4		Lack of Communication participation
5	X4	Requirement	Incomplete requirement gathering
6	X4		Lack of accurate requirement prioritization
7	X4		Unstable requirements
8	X4		Change of user needs & understanding
9	X4		Requirement over scoping
10	X8, A5		Inefficient requirement completion time

11	S1, S15, W2		Lack of consideration of user & applications requirements for offloading decision making
12	S2, S14, W1		Incorrect requirement partitioning
13	S4		Lack of useful information extraction
14	W15		Changing the orientation of the app
15	A1		Diversity of mobile surroundings
16	A11		Event-driven structure
17	A11		Complex contextual features
18	A26		Frequent changing requirements
19	X2, X9, W10	Resource	Lack of appropriate resource allocation
20	X3, S11, S13, W6		Platform incompatibility
21	X8, A5		Lack of resource optimization
22	X8, A5		Energy inefficiency
23	W4		Inaccurate task scheduling
24	W5		Limited resources/resources lacking
25	A1		Lacking of / Limited computational resources
26	A3		Incorrect estimation of battery life
27	A12, A13		Lack of accurate quantification about the consumption of energy by the app
28	A24, A27		Compatibility across various OS versions
29	X10	Security & Privacy	Lack of asynchrony retrofitting
30	S7		Tampering during offloading data
31	S8		Limitation of profilers
32	W14, A2		Lack of identification of risky actions and Vulnerabilities
33	A8		Inconsistent and inefficient testing
34	A10		Lack of configuration of mobile hybrid apps
35	A16, A17		Unclear requirements for app functionality towards privacy threats
36	A22		Patching for updation, correction or improvement

37	A28		Data erasure
38	X4	Stakeholder	Unawareness of needs
39	X4		Cultural and language barrier
40	X4		Lack of domain knowledge
41	X4		Ambiguities among stakeholders
42	X4		Intragroup conflicts
43	X8, A5		Inefficient response time
44	S5		Lack of requirement task efficiency and Responsiveness
45	S9		Lack of development standards and practices Knowledge
46	S10		Testing issues for practitioners

The Communication sub-category comprises 4 challenges Anonymous Communication, Lack of requirement effective articulation, Lack of Verbal & Presentation skills, and Communication participation from studies having paper Id X1 and X4. The Requirement sub-category comprises challenges such as Incomplete requirement gathering, Lack of accurate requirement prioritization, Unstable requirements, Change of user needs & understanding, Requirement over scoping, Inefficient requirement completion time, Lack of consideration of user & applications requirements for offloading decision making, Incorrect requirement partitioning, Lack of useful information extraction, Changing the orientation of app, Diversity of mobile surroundings, Event-driven structure, Complex contextual features and Frequent changing requirements from studies with paper Id X4, X8, A5, S1, S15, W2, S2, S14, W1, S4, W15, A1, A11, and A26. From the Resource sub-category, a set of 10 challenges is listed as Lack of appropriate resource allocation, Platform incompatibility, Lack of resource optimization, Energy inefficiency, Inaccurate task scheduling, Limited resources/resource lacking, Lacking / Limited computational resources, Incorrect estimation of battery life, Lack of accurate quantification about the consumption of energy by the app and Compatibility across various OS versions with paper Id X2, X9, W10, X3, S11, S13, W6, X8, A5, W4, W5, A1, A3, A12, A13, A24, and A27. The fourth sub Category is Security & Privacy, a set of 9 challenges is listed as Lack of asynchrony retrofitting, Tampering during offloading data, Limitation of profilers, Lack of identification of risky actions and vulnerabilities, Inconsistent and inefficient testing, Lack of configuration of mobile hybrid apps, Unclear requirements for

app functionality towards privacy threats, Patching for updation, correction, or improvement, and Data erasure from X10, S7, S8, W14, A2, A8, A10, A16, A17, A22, and A28. While, the last sub-category of Stakeholder, a total of 9 challenges is presented as Unawareness of needs, Cultural and language barrier, lack of domain knowledge, Ambiguities among stakeholders, Intragroup conflicts, Inefficient response time, Lack of requirement task efficiency and responsiveness, Lack of development standards and practices knowledge and Testing issues for practitioners from studies with paper id as X4, X8, A5, S5, S9, and S10.

Table 4.11 shown below is showing the Requirement Engineering Challenges faced by Mobile Developers. The table is composed of 43 cells horizontally titled X1-X7, S1-S12, W1- W8, and A1-A16 while 47 cells are vertically titled as Challenge.

Table 4.12: Requirement Engineering Challenges for Mobile Application Development

Challenge Name	X1	X2	X3	X4	X5	X6	X7	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	W1	W2	W3	W4	W5	W6	W7	W8	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
Anonymous Communication	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of requirement effective articulation	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of Verbal & Presentation skills	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Communication participation	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Incomplete requirement gathering	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of accurate requirement prioritization	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Unstable requirements	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Change of user needs & understanding	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Requirement over scoping	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Inefficient requirement completion time	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
Lack of consideration of user & applications requirements for offloading decision making	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	✓		✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Incorrect requirement partitioning	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of useful information extraction	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Changing the orientation of the app	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Diversity of mobile surroundings	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Event-driven structure	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
Complex contextual features	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
Frequent changing requirements	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-

Lack of appropriate resource allocation	-	-	-	-	✓	✓		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Platform incompatibility	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of resource optimization	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
Energy inefficiency	-	✓		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
Inaccurate task scheduling	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Limited resources	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Limited computational resources	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Incorrect estimation of battery life	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of accurate quantification about the consumption of energy by the app	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	-	-
Compatibility across various OS versions	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓		✓	
Lack of asynchrony retrofitting	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tampering during offloading data	-	-	-	-	-	-	-	-	-	-	-	✓		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Limitation of profilers	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of identification of risky actions and vulnerabilities	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Inconsistent and inefficient testing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
Lack of configuration of mobile hybrid apps	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
Unclear requirements for app functionality towards privacy threats	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-
Patching for updating, correction, or improvement	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-
Data erasure	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
Unawareness of needs	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cultural and language barrier	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of domain knowledge	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Ambiguities among stakeholders	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Intragroup conflicts	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Inefficient response time	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-
Lack of requirement task efficiency and responsiveness	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lack of development standards and practices knowledge	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Testing issues for practitioners	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

The symbol "✓" shows the occurrence of challenges in respectively included stud (ies) while the symbol "-" is showing that the particular challenge does not occur in the respective paper. The challenge with the name "Anonymous Communication" occurred in study 1 that X1 published in IEEE Xplore. The challenges with names "Lack of requirement effective articulation", "Lack of Verbal & Presentation skills", "Communication participation", "Incomplete requirement gathering", "Lack of accurate requirement prioritization", "Unstable requirements", "Change of user needs & understanding" and "Requirement over scoping" is present in study X4. The challenge named "Inefficient requirement completion time" wasfound in studies X2 and A4. The challenge with "Lack of consideration of user & applications requirements for offloading decision making" occurred in S1, S12 and W1, "Incorrect requirement partitioning" is found in S3 and S11, "Lack of useful information extraction" occurred in study S3, "Changing the orientation of the app" found in W8, "Diversity of mobile surroundings" found in A1, "Event-driven structure" and "Complex contextual features" found in A7, "Frequent changing requirements" found in A14, "Lack of appropriate resource allocation" found in X5, X6 and W6, "Platform incompatibility" occurred in X3, S9, S10 and W5, "Lack of resource optimization" and "Energy inefficiency" found in X4 and A2, "Inaccurate task scheduling" found in W3,"Limited resources" found in A1, "Limited computational resources" found in A1,"Incorrect estimation of battery life" found in A3, "Lack of accurate quantification about the consumption of energy by the app" found in A8 and A9, "Compatibility across various OS versions" found in A13 and A15,"Lack of asynchrony retrofitting" found in X7, "Tampering during offloading data" found in S5, "Limitation of profilers" found in S6, "Lack of identification of risky actions and vulnerabilities" found in W7 and A2, "Inconsistent and inefficient testing" found in A5,"Lack of configuration of mobile hybrid apps" found in A6,"Unclear requirements for app functionality towards privacy threats" found in A10 and A11,"Patching for updating, correction, or improvement" found in A12,"Data erasure" found in A16,"Unawareness of needs" , "Cultural and language barrier", "Lack of domain knowledge", "Ambiguities among stakeholders", "Intragroup conflicts" found in X4, "Inefficient response time" found in X2 and A4, "Lack of requirement task efficiency and responsiveness" found in S4"Lack of development standards and practices knowledge" found in S7, and "Testing issues for practitioners" found in S8.

4.5 Description of discovered challenges

Anonymous Communication

An anonymous communications system is a piece of software that hides a user's IP address from the server that hosts the website they're visiting. The primary premise of this challenge is that the sender's and receiver's identities must not be leaked to a third party [85] [52]. When we talk about mobile development or in the mobile domain, the mobile developers face it as a challenge specifically when they are developing mobile applications like watts app, telegram, and similar to these applications. The functionalities these apps follow are end-to-end encryption in which the users' data is completely secure. But it's a challenge for those apps that don't implement encryption mechanisms.

Lack of requirement effective articulation

This challenge is basically about the conflicts between the users and developers. In the mobile development context, due to mobile environment diversity, users and developers both consider the requirements with their perspectives which leads to conflicts between them [9].

Lack of Verbal & Presentation skills

This challenge relates to presenting the views and thoughts making others difficult to understand the ideas. As understanding mobile development domain is difficult due to its diversification and complex nature, so, a person may not be skillful enough to share his/her ideas with other stakeholders or he/she may fail to present his/her views and thoughts to other stakeholders which may lead a great challenge in mobile development [9].

Lack of Communication participation

This challenge can be differentiated from above in a way that a person is skillful enough but he/she does not want to share his/her ideas with other stakeholders. He/she is reluctant to share his/her views with other stakeholders [9]. As the mobile domain is becoming advanced day by day, introducing new technical terms and technologies which may be difficult to express in front of other stakeholders or may be due to crowd fear which may cause anxiety and hypertension.

Incomplete requirement gathering

This challenge is basically about the requirement gathering phase. Incomplete requirement gathering means the absence of the necessary and compulsory requirements [9] As the mobile domain is vast it's obvious that compulsory requirements may lag or may be ignored sometimes.

Lack of accurate requirement prioritization

Requirement prioritization or software requirement prioritization is one of the phases in the requirement engineering process. This challenge is about the inaccurate prioritization of the requirements. The compulsory requirements could not be executed first or compulsory requirements given low priority [9]. As the mobile domain is complex to understand comprising multiple events and handling those events is quite difficult. A skillful person is needed to understand such handling of events. So, in this context, sometimes, it is confusing which requirement is to prioritize first or given the high priority.

Unstable requirements

Unstable requirements mean that the requirements are not stable. Stable in the sense that the requirements are not clear enough to achieve their goal. Or the requirements failed to capture their expectation [9]. As already mentioned above a mobile environment is needed to handle delicately. With its complex nature, the requirement analyst or mobile developer is unable to understand the mobile requirements.

Change of user needs & understanding

This challenge is about the user needs which change over time. Or we can say that the needs of users' are constantly changing over time which leads to a lack of understanding of user needs. The changes in the user needs may lead to the lacking of understanding of the actual needs of the users [9]. Due to the advance and introduction of new complex technologies, the user needs in context with mobile development changes create a barrier for mobile platforms.

Requirement over scoping or ill-defined scope

This challenge is about the inaccurate scope. The scope for the requirements or the project is either below the defined limit or above the defined limit. Or the scope for requirements is not properly defined which leads to conflicts and stakeholders dissatisfaction [9]. With the complexity of the mobile environment along with the new technologies, defining a scope for mobile requirements is difficult in such context.

Inefficient requirement completion time

Inefficient requirement completion time means that the requirement is not being completed within the estimated time. Or we can say that the lack of achieving optimal solution when the requirement could not be completed on estimated time [36] [71]. As mobile development is different as compared with traditional development which needed more computational resources such as memory or storage. So, due to limitations of resources, it is difficult for mobile requirements to be completed in the estimated time.

Lack of consideration of user requirements & application requirements for offloading decision making

This challenge is about the lacking in the consideration of either user or application or both types of requirements to offload decision making. Due to limited resources in a mobile environment, the requirements related to the user and application could not be considered or sometimes ignored to take decisions for migration of computation-intensive tasks [38] [39] [34].

Incorrect requirement partitioning

Incorrect requirement partitioning means that the requirements are not partitioned accurately or not given correct partition which may be problematic for mobile development due to lacking resources in the mobile domain. Or this problem is simply about making wrong partitions of tasks that are to be executed on the cloud and those that are executed on mobile devices for computation offloading [56] [61] [63].

Lack of Useful Information Extraction

This challenge particularly refers to mobile crowdsensing applications. Crowdsensing is a technique in which a large number of people using mobile devices with sensing and processing capabilities share data and extract information to measure, map, analyze, estimate, or infer processes of common interest. Because MCS applications run in dynamic environments that include sensors, mobile devices, and the cloud, it's critical to achieve energy-efficient and context-aware sensing process scheduling, which includes data transmission from sensors to mobile devices and from the cloud to mobile devices, which poses a problem in extracting useful information. To put it another way, both the sensing process and data transmission from mobile devices to the cloud must be managed to guarantee that user data is acquired only when an MCS application requires it [59].

Changing orientation app

The difficulty of changing orientation is a discrete event in mobile platforms that is generally referred to as switching between portrait and landscape layout configurations of the running app. When this happens, Android recommends that the app adapts to the new layout, preventing memory leaks and preserving the application's data as well as any outstanding essential message passing activity. However, putting this advice into practice is difficult, and Android programmers will confront programming challenges as a result [38].

Diversity of Mobile surroundings

This challenge is about the occurrence of a variety of mobile platforms that are programmed with their respective programming languages leading the mobile environment diverse creating a challenge for mobile platforms [69].

Event-driven structure

This challenge is basically about the multiple events occurring simultaneously in a mobile environment. The mobile environment is bounded in a complex structure having multiple events that could not be properly handled at the same time. So, it again creates a challenge for the mobile environment [73].

Complex contextual features

The contextual features such as notification and sensor handling in mobile development due to platform dependence are difficult to implement making it one of the major challenges in the mobile environment [73].

Frequent changing requirements

Software programmers must evolve overtime to deal with the introduction of new needs, adapt to changing situations, rectify defects, and improve software design. However, software quality may deteriorate as a result of aging, regardless of the type of changes made in mobile applications or mobile software. As a result of poor design and implementation choices, mobile software quality, as well as the performance of software applications, degrades over time. Antipatterns and code smell imply poor decisions [46].

Lack of appropriate resource allocation

Due to resource limitations in mobile development, the task that requires more resource allotment could not be allocated or there could be a possibility that the task needs less resource provided it with more allotment leading the resource wastage which results in lacking appropriate resource allocation creating a challenge in mobile application development [36] [54] [67].

Platform incompatibility

This challenge is about the platforms that lack compatibility. Due to scarcity of development resources, developers are compelled to support only a few platforms and devices. As a result of which mobile developers face platform incompatibility issues [80] [50] [51] [66].

Lack of resource optimization

The mobile environment lack occurrence of optimal resources as compared to the web environment or it is unable to provide the best resources for tasks execution which creates a challenge in the mobile environment [36] [71].

Energy inefficiency

The challenge of energy efficiency means that the computation-intensive tasks in a mobile environment consume more energy due to bounded resources [36] [71].

Inaccurate task scheduling

This problem is related to task scheduling, which is inefficient in a mobile setting. MCC is a high-performance data processing system, but it is difficult to forecast the precise scheduling of jobs, such as how to plan mobile application tasks in data centers to extend the battery life of mobile devices, in this case. As a result, precise assessment and prediction of mobile application task scheduling become a challenge on mobile platforms [53].

Resource lacking

The mobile environment, in comparison to traditional development, is constrained by limited resources. Due to the limited resources available, we put a priority on performance. Because there are multiple different mobile platforms, each with its operating system and hardware, native apps for these platforms are developed and maintained separately, resulting in varied performance [54].

Lack of computational resources

The mobile environment offers less memory and computing power than conventional PC systems creating a challenge for mobile platforms [69].

Incorrect estimation of battery life

This challenge is about the battery life of smartphones. Users of mobile devices are often concerned about energy alerts, and they frequently take actions to extend battery life. Commercial smartphone platforms, such as Android and iOS, do not, however, contain capabilities that show how much battery capacity is left. The problem is that the majority of smartphone users have no idea how long their battery will last. As a result, predicting the precise battery time available of running programmers is a challenging task [55].

Lack of accurate quantification about the consumption of energy by the app

It is quite difficult for mobile developers to quantify how much energy their apps consume and to investigate how that energy use changes as a result of factors outside their control, such as network congestion, mobile operator choice, and user screen brightness settings [56] [57].

Compatibility across various OS versions

The Android ecosystem is heavily fragmented. The occurrence of countless OS versions makes it impossible for mobile developers to test their apps. As a result of which, various compatibility issues arise leading to poor user experience [58] [59].

Lack of asynchrony retrofitting

Asynchronous programming is an important aspect of mobile development to avoid unresponsiveness. Developers can use several async structures in Android. However, developers can continue to utilize inefficient async techniques, resulting in memory leaks, lost results, and wasted resources. While asynchrony retrofitting is the adoption of proper utilization of async constructs to avoid the aforementioned issues. The Android OS provides developers with several async constructs which can be used by them. But, unfortunately, the mobile app developers are unable to utilize async constructs or techniques which results in memory leaks, lost results, and wasted resources [41].

Tampering during offloading data

Tampering refers to the intrusion of unknown risks that occur when code and data are offloaded to the public cloud, which is, by definition, an untrustworthy platform. As a result,

security issues arise, posing a problem for the MCC environment's ability to attain high performance that creates a major barrier in mobile development [43].

Limitations of profilers

Profilers are software programmers that keep track of not just the operating environment's (surrogate and network) settings, but also the available resources on the mobile device. In addition to the quality of the profiling data, the profilers must be lightweight to save money. However, the profiler's problem is that it's impossible to assess the offloading data execution possibilities, such as running duration, network availability, and communication cost. It's because of the unpredictability of mobile device behavior at runtime [60]. A profiler can help you better understand the timing of your code so you can optimize it for varied runtime situations or loads [61].

Lack of identification of risky actions and vulnerabilities

Android is widely used by mobile application developers all around the world. Android comes with a message transmission system that allows apps to communicate with one another. Because of the threats this system poses, identifying its dangerous actions and potential weaknesses is crucial. Because of Android's communication model, malicious apps can force other apps to perform unwanted actions and steal end-user data while appearing regular and innocuous [62] [63].

Inconsistent and inefficient testing

Touch and gesture-based interfaces are common in mobile applications. As smartphones and tablets became more mainstream, there was a larger requirement for specialized software engineering approaches. To secure high-quality solutions, software development requires consistent and effective testing. On the other side, testing mobile applications are still cumbersome, time-consuming, and error-prone. One factor is that smartphones emphasize touch-based interaction; gestures are difficult to integrate into automated application evaluations [64].

Lack of configuration of mobile hybrid app

Hybrid mobile applications are web apps wrapped in a native software shell that connects to whatever capabilities the mobile platform offers via a browser embedded in the app after it's downloaded and installed locally from an app store. Smartphone users are more

concerned about their privacy than laptop users, and they are hesitant to conduct privacy-sensitive and financial transactions on their phones due to their untrustworthy character. Smartphones, on the other hand, are not secure in terms of configuration, and many programmers overlook it as a non-functional and unimportant aspect of the code's primary purpose, forgetting that the impact of such defects may not always interfere with program logic, but only emerge after security breaches [27].

Unclear requirements for app functionality towards privacy threats

The term "privacy disclosure" refers to network requests that send out one or more forms of private information. Android is a well-known and widely used platform that gives users extensive capabilities for accessing personal sensitive data, posing major privacy risks. Several methods for detecting these dangers have been presented, however, they all fail to apply privacy policies or determine what is required for app functionality. In other words, establishing whether a privacy-sensitive data object, such as a user's location or identity, is required for the app's principal operation or just being transferred to a large number of third parties, is difficult. For example, Google Maps requires the user's location to provide driving instructions, whereas a weather app may request the user's location to provide weather services. [35][37].

Patching for updation, correction, or improvement

A patch is a set of changes to a computer program or its supporting data that are designed to update, rectify, or improve the program. It includes resolving security holes and other problems, and such upgrades are sometimes known as bug fixes. Patches are routinely released to improve the functionality, usability, or performance of an application. When a security breach is discovered, it is immediately repaired. The patching issue, on the other hand, is addressed by the fact that this patching method is limited to Google-connected or supported applications. The patching method does not update any other manufacturers or third-party software [65].

Data erasure

Because of its prominence as the most extensively used mobile operating system, Android has long been a subject of privacy concerns. In Android OS, data erasure is a prevalent concern. The most serious issue with this OS is how it and its apps deal with data, such as when and how it is viewed, modified, or moved. Data reappearance after improper erasure could be

Danger as a result of Android's lack of clarity regarding how third-party applications process user data saved on a mobile device [29].

Unawareness of needs

The mobile app requirement is typically different from the traditional development. Web development has its own needs and requirements while mobile development has its own. The unawareness of needs in this context may cause problems among stakeholders [9].

Cultural and language barrier

Every organization has its own culture and language to follow. The organization forces its employees to stick to the organizations' defined standards and practices. The differences in practices and development standards among different workplaces create one of the major challenges in mobile development [9]. In context with mobile platforms, this challenge occurs as when a client wants to build an iOS native app, then the organization must have a SWIFT (programming language) developer, and a similar case for Android native app, a company must have KOTLIN/Java developer. But, if the software company doesn't have any of the aforementioned developers, then it may occur a language barrier in a mobile environment.

Lack of domain knowledge

Mobile development is a diverse and vast domain comprised of multiple areas as compared with web development or traditional development. We need domain specialists for each area. Each developer must be a domain specialist. So, lack of domain knowledge is one of the major barriers faced by mobile developers [9].

Ambiguities among stakeholders

This challenge relates to the conflicts among the stakeholders. The participating stakeholders are unable to stick to a single point leading to the contradiction in views [9]. Let's say, the stakeholders are targeting complex business requirements that might not be possible for the development team to implement in such an environment. So, in this context, it may be a challenge or conflict.

Intragroup conflicts

This challenge occurs when people usually work in teams. So, it is obvious that disagreement or difference in opinions must be created which is a major barrier in mobile

Environment [9]. Let us say, a group of developers is working on the same project, this might cause code conflicts during the merging of code.

Inefficient response time

As mentioned above the mobile environment comprises an event-driven structure having multiple events or tasks occurring simultaneously. So, the execution of multiple tasks or events in a mobile environment is time taking process that leads to inefficient response time [36] [71].

Lack of requirement task efficiency and responsiveness

The Android operating system is critical in enabling mobile apps to provide users with a variety of benefits. Existing Android apps, on the other hand, face considerable challenges in terms of performance and quick response to user expectations due to inefficient execution strategies. In such a circumstance, achieving efficiency while still benefiting users is crucial [66].

Lack of development standards and practices knowledge

Mobile software programmers (apps) have become more popular, posing a security risk. These apps are typically offered for free or at a low cost across all platforms, and they are frequently developed by small firms and novice programmers. Due to a lack of development standards and best practices, the mobile device is vulnerable to attacks [67].

Testing issues for practitioners

The need for eliciting testing requirements early in the development process, conducting research in a real-world development environment, specific testing techniques targeting life-cycle conformance, mobile services testing, and a comparative study for security and usability testing are among the challenges [1].

4.5 Summary of the Chapter

This chapter was all about the SLR findings including the findings from Grounded Theory along with the findings from the Expert Review conduction. A total of 47 challenges were identified from SLR, then listed after constructs extraction from Grounded theory, and forwarded for expert opinion for validation and evaluation. The experts gave some suggestions and as per their suggestions, some changes were made. A complete list of

Challenges after Experts.' The suggestion was sent for Industrial Survey to obtain the resolutions. The complete survey findings will be explained in the next chapter.

CHAPTER 5

FINDINGS AND DISCUSSION

5.1 Introduction

The previous chapter was about the SLR findings including the findings from Grounded theory and Expert Review conduction. A list of 47 challenges was identified from phase 1 of the research study is Systematic Literature review, the relevant constructs were extracted via encoding techniques of Grounded theory. The final list was forwarded for Expert Opinion for validation. The experts gave some suggestions to make. The final list after experts' suggestions is forwarded to software houses for Industrial Survey conduction to propose some mitigation plans. This chapter will briefly discuss the findings from the Industrial Survey.

5.2 Findings from Industrial Survey

An industrial survey was conducted so that a list of mitigation strategies can be obtained to overcome the discovered challenges that the mobile developers face during the accomplishment of the Requirement Engineering process. To achieve the goal, a protocol of Mark Kasunic [49] is followed which is the widely used guide for conducting an industrial survey. Each step is completely being followed from a standard guidelines and is reported in chapter 3. As the research area is particularly the mobile domain so, the target population for achieving the research goal was from software houses specifically implementing the Requirement Engineering process in a mobile environment. The questionnaire was created online in "Google forms" and is being sent to the target software houses in early September 2021. The questionnaire is composed of 2 sections. Sections 1 was designed to obtain the Respondent Personal Information as Respondent Name, Name of Organization, Designation, Size of Organization, Overall Experience and Experience in Mobile Development while Section 2 was designed to get the Mitigation Strategies for implementing Requirement Engineering Process specifically for mobile platforms along with the Practicality Level. Section 2 comprised a list of identified and validated challenges obtained from phase 1 of SLR conduction. The list follows a description and the respondents are asked to suggest the

Mitigation plans to overcome the respective challenges. The complete questionnaire is attached in Appendix Section shown in Appendix H. Online sources were used to contact various software companies for getting the responses. Among these contacts, 15 E-mails were there. The survey form was forwarded online to all 15 contacts. After 20 days, we get 2 responses. We wait for approximately 1.5 months but still, the process of getting responses was very slow. We get to achieve 10 responses in 1.5 months. We decided to contact more people via LinkedIn Corporation, so, 17 people among the required designation were selected and requested to fill out the survey form. This time, the response was better and we achieved 40 responses in 1 month and 3 weeks. To increase the sample size, we planned to visit some software companies in Islamabad. After getting an appointment, we visited the software organizations along with some printed survey forms. We conveyed our research problem to them and requested them to fill out the survey forms. We waited till the mid of December and after 3 months, we achieve our goal with 47 responses. After combining all responses, we can achieve a total of 97 responses. 17 among these 97 were incompletely filled, so, we decided to exclude them from our responses list. Finally, a combined result was a total of 80 responses that were used for data analysis both from the online and printed form.

As mentioned above Section 1 of the survey form was about the Respondents' Personal Information. So, the graph is showing the distribution of respondents based on their Overall Experience, Experience in Mobile Development, Designation, and Size of Organization respectively. The distribution of respondents based on the aforementioned criteria is represented in graphical form below:

5.2.1 Distribution of Respondents based on Overall Experience in Organization

The graph drawn below is showing the overall experience of respondents in this organization. 88.8% of the experience is more than 2 years while 11.3% is based on 2 years.

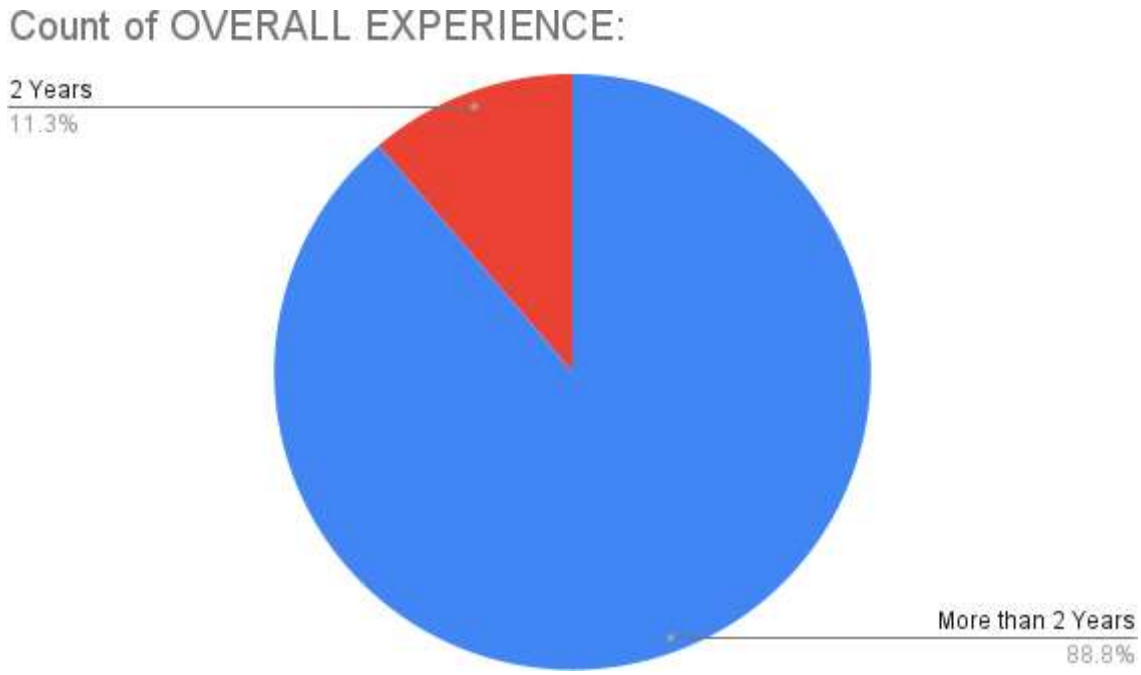

Figure 5.1: Distribution of Respondents based on their Overall Experience

5.2.2 Distribution of Respondents based on Experience in Mobile Development

The graph drawn below is the distribution of respondents based on their Experience in Mobile development. 57.5 5% of the respondents are having Experience of 2 years while 42.5 % of the respondents are having Experience of more than 2 years.

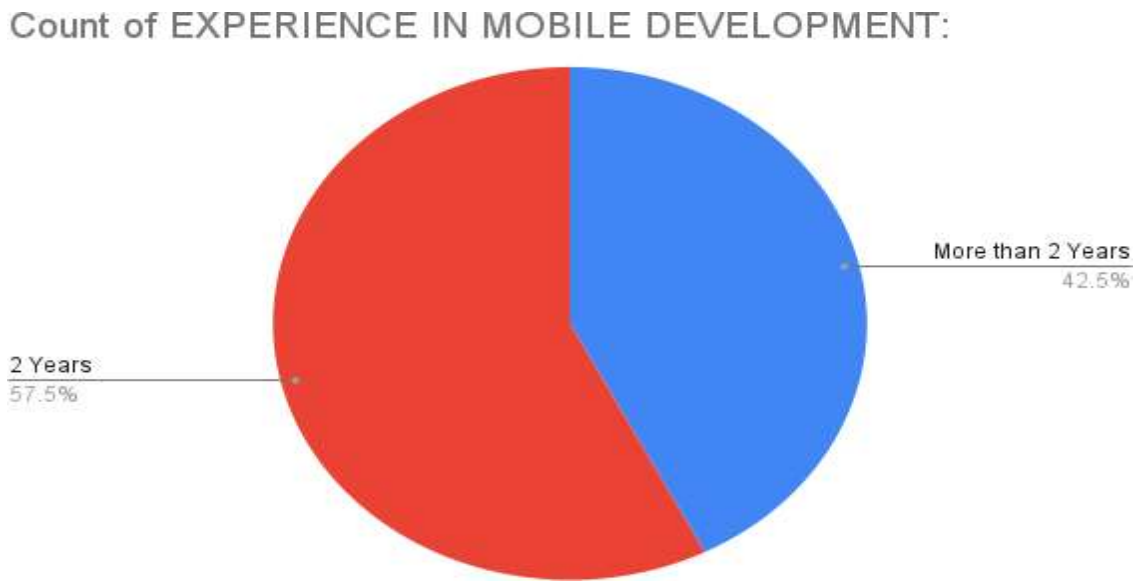

Figure 5.2: Showing the Distribution of Respondents based on their Experience in Mobile Development

5.2.3 Distribution of Respondents based on Designation

The graph shown below is the count of designation of the respondents who took part in an industrial survey. 97 people participated in filling our survey forms, out of which 80 responded properly. Out of 80 responses, the distribution goes like, 2 were full-stack developers, 15 were senior software developers, 12 were SQA engineers, 5 were team lead, 4 were IT engineers and PM. 11 were Android/ Mobile developers, 14 were software developers, 3 were Flutter developers and others were Junior software developers, Associate software engineers, CEO, Search Engine Optimization, Principle engineer, Lead mobile engineer Lead developer, Business analyst, and Associate consultant.

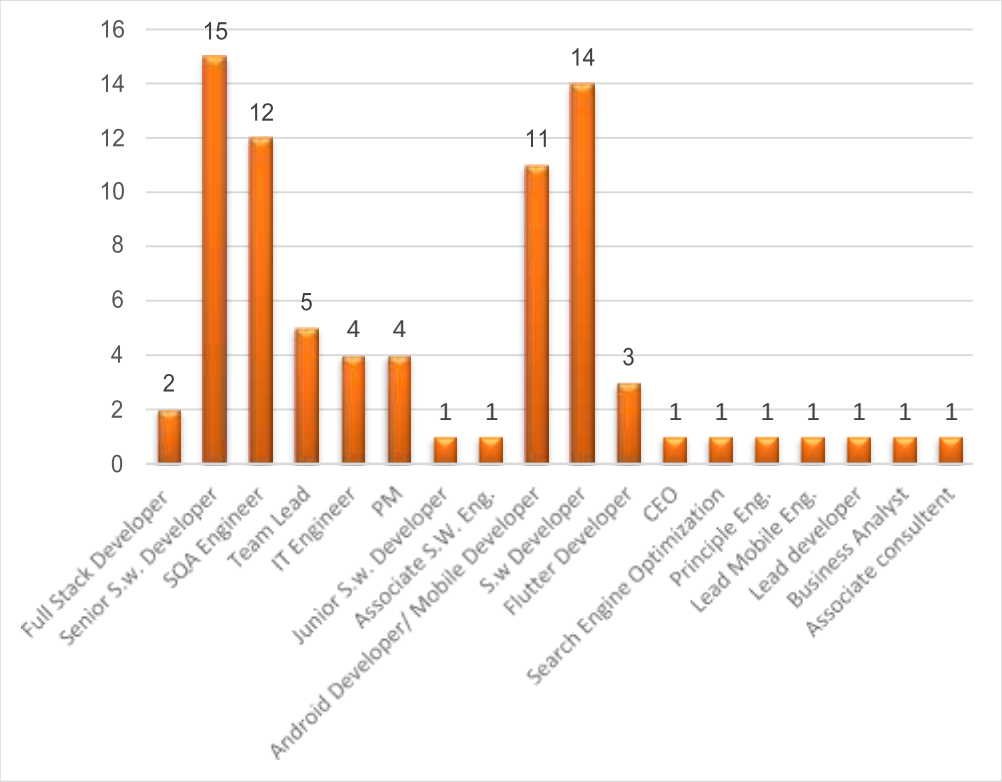

Figure 5.3: Distribution of Respondents based on their Designation

5.2.4 Distribution of Respondents based on the Size of Organization

The graph shown below is the distribution of respondents based on the size of an organization. The X-axis is showing the scale of distribution of organization as small, medium, large and medium-large respectively.

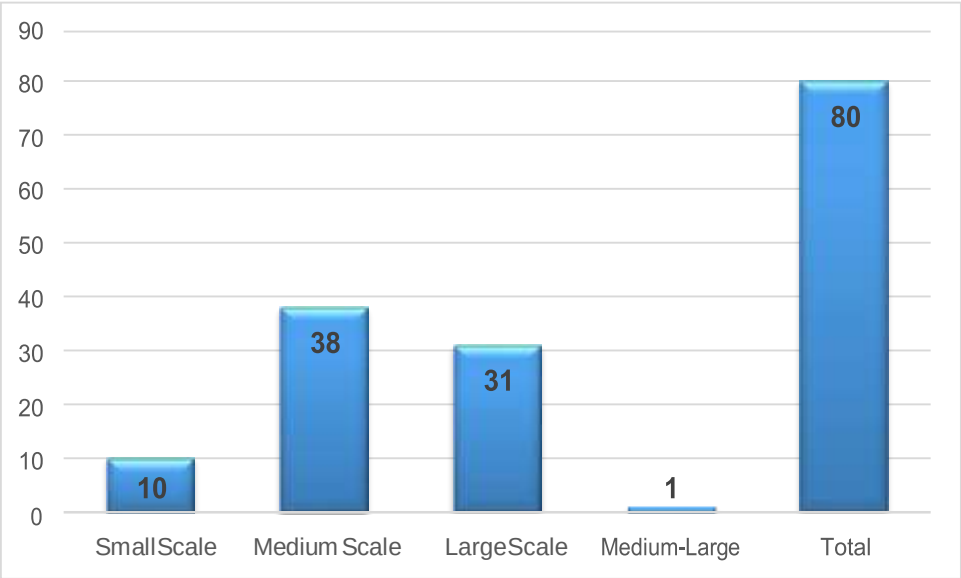

Figure 5.4(a): Graph showing Distribution of Respondents based on Organization Size

Out of 80 respondents, 10 were from small-scale organizations, 38 were from medium scale, 31 were from large scale while 1 was from a combination of both medium-large.

When talking about percent-wise, the graph presented below is showing the distribution based on percentage. 12.5% of respondents were from a small scale, 46.3% from medium, 40% from a large while, only 1.3% lies from the medium-large scale.

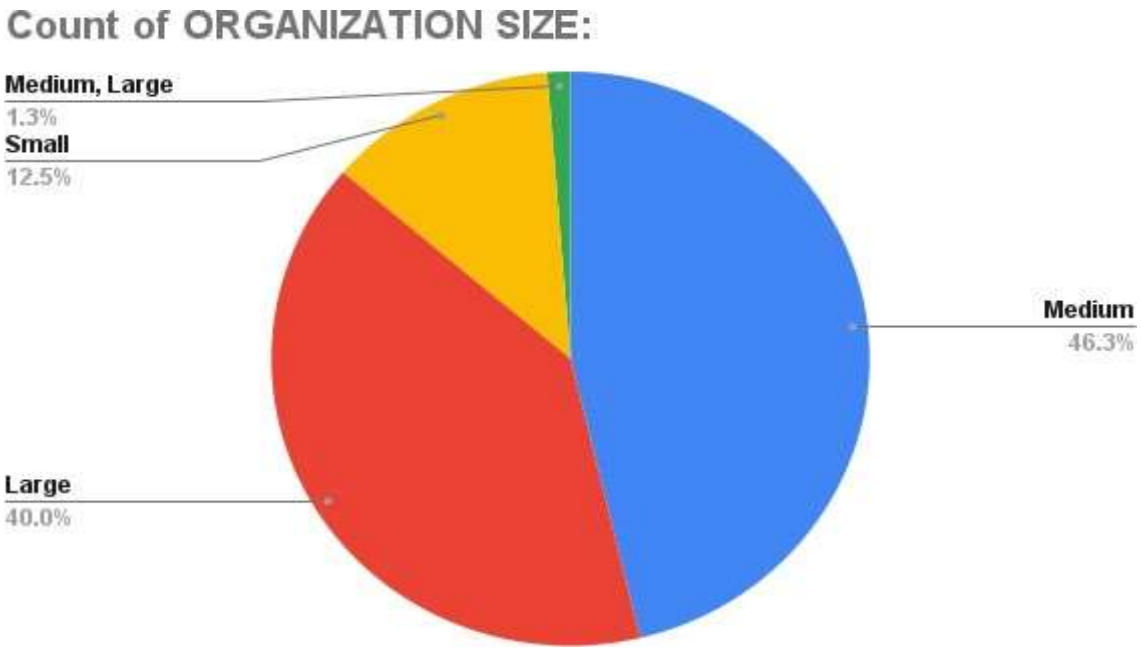

Figure 5.4(b): Distribution of Respondents based on Organization Size

Table 5.1 tabulated below is the list of challenges along with their solution to overcome the respective challenge and the practicality level.

Table 5.1: Showing the Challenges along with the Mitigation Strategies and Practicality Level

Challenge	Mitigation Strategies	Practicality Level of Challenges
Anonymous Communication	Encryption-Decryption mechanisms/ techniques/ Algorithms	96.4%
	Integrate encrypted channels/ do introduce encrypted Challenges	
	Use of APIs, tokens, or encrypted keys/ encrypted keys must be embedded/ encrypted keys and the token can resolve the issue	
	Kanban, Scrum, extreme programming	
	Both parties should use a platform that does not require their identities for communication. It should be made sure that the chosen platform provides end-to-end encryption because if the communication medium is secured, there is no chance for any intruder to get any information about the sender and receiver.	
	We can provide secure APIs using tokens or encrypted keys. So when the user is fetching the data or even store in the database, it will be based on token or Authorization keys. So user-ID will be secured and a third party can't access the information for any user.	

	Embed linguistics Translator/ use linguistics translator	
Lack of requirement effective articulation	In my opinion, this can be resolved by properly on-boarding of users in the app. There can be some video or detailed overview of the features on the first launch of app and this can be provided as a feature in the app so the user can even see this later on.	96.4%
	Conduct stand-up meetings/ stand-up meetings must be there/ must have daily meetings/ daily sprint Meetings	
	The requirement gathering should be done properly. All the tasks and scope should be defined in Scrum. So, no one has the conflicts/ check requirements carefully/ re-check requirement gathering phases/ The development of any project should start with the understanding and incorporation of user requirements in the development process.	
	Use JIRA for project management and PM should write detailed requirements about the feature	
	Communication between developer and client either verbally or graphically can overcome the challenge	
	Proper documentation/ specification document	

	make prototypes and have feedback from the users	
	There should always be some reference applications to keep an eye on existing features and after detailed analysis, some documents and a flow should be locked to act upon.	
	Never start work unless a listen to a go from both sides, one should define and the other one should note what he's getting, now he should define and the other one-note, practice this multiple times and at last you will get the final results	
	To minimize the conflicts, requirements should be revised several times. Also, it should be made sure that the developers are given a chance to communicate directly with stakeholders so that they could understand their perspective and both parties could agree upon a potential solution.	
	Developers make prototypes that are just a model of the requirement, like Designs on Figma (tool), that provides a basic knowledge of what this requirement will do. In which Figma provides the click functionality as well, you can click on buttons in the designs and it will move to the next screens as required in the functionality. Users can easily address developers to add something or remove and when he approves it, then Frontend developers can start work.	
Lack of Verbal & Presentation skills	Apply UML/ UML implementation/ UML concepts	97%
	Language barrier reduction	
	Generate models	

	Follow model-based communication	
	Model generation and implementation of UML Concepts	
	Effective visual presentation	
	Present using diagrams instead of long paragraphs	
	Generate models by applying UML diagrams	
	To present and understand the ideas effectively, discussion sessions should be arranged where both parties could try to explain their ideas in general terms (non-technical way) making it easy for others to grasp the ideas.	
	We propose them the ideas using Data Flow Diagram, Use Case Diagram, Sequence Flow Diagram, etc. So the user knows where we stand and then the user and developers will be on the same page. / introducing models/ use diagrammatic representation	
	Daily scrum meetings must be there	
Lack of Communication participation	Model based communication/ models/ diagrams/ communicate via models/ construct models/UML models	97.6%
	Requirement elicitation technique	
	Documentation must be there	

Incomplete requirement gathering	Interviews or questionnaires/ conduct interviews/ Interviews + questionnaires/ Interviews from relevant stakeholders	96.5%
	The proposed solution must be discussed carefully.	
	Proper implementation of V& V model on Requirement	
	Always take feedback from client-side when you complete your first module/ Try to take feedback from clients after completion of each module/chunk/ Take feedback from customers after completion of first Release	
	Meetings, online meetings or physical, what meeting on both side, from development to client, don't miss anyone, all will be stakeholders, so don't miss anyone. Gather requirements and pass to other one/ Brainstorming and meeting with the client many times during development.	
	The requirements should be revised between clients and developers. Stakeholders should be asked to explain their required system multiple times. The elicitor should have enough domain knowledge to understand stakeholders. The requirements elicitor should repeat what he understood in his terms, with added information - if needed, so that each requirement gets validated.	

	When a user has an innovative idea to add to the app, the developer team, mostly software engineers do the R&D to gather the requirement and make a list of challenges they will face during the development including third-party libraries or servers and their costs. They acknowledge the user about that, and if a user can afford those challenges then the team proceeds to development.	
	Requirements documents should be revised after taking comments from system architect, developers and quality assurance team/ User specifications document to be prepared by the developers and dually signed/agree/ SRS should be properly documented	
Lack of accurate requirement prioritization	Perform storyboard approach/ apply storyboard techniques/ Perform Storyboard approach	93.9%
	The ranking or Voting the Requirements	
	Product backlog, sprint backlog/ Use of scrum can minimize above risk	
	It is a part of the software engineers or analytics team to arrange the compulsory requirements in such an order that the main functionalities won't be missed out, and by the time any new important functionality comes into the app then they should manage it without disturbing other functionalities or requirements. Mostly we use JIRA (Atlassian account feature) to	

	manage the work and scrum master create tasks and assign the priorities for each task	
	The developer should have enough domain knowledge to prioritize the requirements in the correct order. If a developer has difficulty ranking them correctly, he should study the domain and understand with examples systems.	
	Do clarify your business need first then prioritize your requirements based on business need	
	Everything should be defined in Scrum with priority and deadline. So, they can follow everything. Team leads should help the developers where they are stuck or won't fulfill any requirements. So they won't switch the prioritized tasks.	
	It depends on the business team how they want to proceed with the product according to their targeted user. So prioritize features that are more demanding to the users.	
	This demonstrates the lack of understanding by the developer team. The compulsory requirements should be catered to in the first place. Subsequently, other Requirements may be fulfilled.	
	The requirement should be analyzed properly with the involvement of client/ Involve customer/ Conduct meeting with clients	
	follow mobile development lifecycle, prioritize the task stepwise	
Unstable requirements	Daily meetings must be conducted/ daily meetings should be conducted/ Must have daily meetings/ Daily meetings/ Daily scrums meetings must be Done	97.5%

	Requirements tested through TDD/ We should create the hierarchy where the developers should do unit testing and then they release the build with a document of what they achieve in this build and business logic steps should be mentioned in a document. So, they can give the build the TLs and then they test that build and if found anything then they must ask developers to fix them and pass this build to QA.	
	More time should be spent to understand the system, requirements should be discussed repeatedly between parties to obtain stable ones. Different gathering techniques should be used to get the correct requirements. For example, rough sketches for the flow of each required feature should be made repeatedly until improved to stability.	
	Sometimes the framework or OS versions don't support the functionality or requirement. And sometimes miscommunication between the teams leads to failing tasks. We should use daily scrum meetings to avoid that anomaly/ daily scrum meetings/ this issue can be resolved via daily scrum Meetings	
	Use of extreme programming and agile to include changing requirements	
	Change management, sprint backlog, product backlog, and kanban support change management so well.	
	Analyze the requirements	

Change of user needs & understanding	Take customer feedback/ user feedback/ release your product for customer feedback/ take feedback from clients	97.7%
	Release your product as a beta version to take feedback from users/ release your product for customer feedback/ take customer feedback after the release of your product	
	Take client feedback after each module or task	
	Prototyping before actual Development	
	This leads to scope creeping, you should stick to documented requirements and changes can be accommodated in next phase	
	Work on beta versions, do a QA properly	
	If the user needs changes are huge development should be stopped and quick requirements understanding sessions should be arranged between stakeholders and developers and it should be made sure that the user has agreed upon a new or updated set of requirements.	
	Frequent meetings with stakeholders should be conducted to ensure consistency	
	On-board customer as increase transparency	
	Product backlog	
Requirement over scoping or ill-defined scope	Re-consider the requirements and re-define the scope/ redefine requirements and scope/ consider your requirements and scope again/ define scope again/ re-define scope	96.3%
	Meetings with stakeholders/ pm & stakeholders role/ conduct meetings with project manager to discuss or re-define requirements & scope/ Involve your	

	stakeholders and discuss with project manager/ Stakeholders and PM involvement is necessary here	
	Create the prototypes while sharing the ideas and the solution architect should involve in that meeting so he can suggest which technology we might use and how far the possibilities where we can achieve the specific Requirement.	
	A standard form of documentation should be adopted to avoid conflicts.	
	A project scope definition document should be prepared to overcome this issue. It includes the outcome of a particular project and its associated benefits and also outlines any constraints imposed upon the project and the assumptions that have been made along the way/ follow the scope definition document	
	For any type of project, the definition of your project scope could very well change (it almost certainly will), during the life of the project but each change should be controlled and managed to avoid "scope creep" where the initial aims of the project are obscured by ongoing modifications.	
	Agile frameworks support continuous changes by users... I think it's kanban.	
	Work breakdown structure must be implemented/ WBS approach/ Follow WBS approach/ Implement	96.25%

Inefficient requirement completion time	WBS approach/ apply WBS concept/ Break your project into smaller modules/ project breakdown	
	Sprint backlog, stand up meetings	
	Analyze the requirements carefully	
	Split the bigger task in smaller chunks, that make easy to meet deadlines	
	The Schedule of the project should be followed religiously and some extra time should be reserved to compensate for the change in requirements	
	First of all, the user must be told that if the requirement is huge and must be completed on time then we will need several developers to work on. Secondly, make weekly tasks on JIRA Atlassian board that keep track of each task based on story points and estimated time. If organizations follow these rules les they won't miss the timeline.	
	Also, during the time estimation period, a fair time margin should be added to the total estimate of task completion time. This margin should be utilized to avoid delays. And it should be kept reserved, if not needed, for potential delays in upcoming tasks.	
	At first, a developer should prioritize all tasks correctly. Next, each task should be divided into small goals and it should be made sure that each goal is set using SMART method: Specific, Measurable, Attainable, Relevant, and Timely.	

	PM should prioritize the major functionality which is necessary and the other functionalities should be delivered in iterations/ Requirement prioritization, requirements with high priority should be given more time and vice versa	
Lack of consideration of user requirements & application requirements for offloading decision making	Face 2 face meetings between clients and developers/ conduct face to face meetings between clients and Developers	96.7%
	Rapid feedback through white box process	
	Must see the user requirement carefully/ We should understand the user needs and what is in trending/ Requirement gathering must be clear and brief	
	During the planning stage, software architecture should be also planned before implementation	
	Data engineers collect the data logics of each requirement that will be handled through the cloud. Not everything will be handled on the Frontend side of the app it will make the app overloaded and irresponsive.	
	User preferences should be given priority	
Incorrect requirement partitioning	Scrum/Follow Scrum/Implement Scrum	96.3%
	This will resist the team to follow agile methodology. The sprints should be made so that they will focus on specific features instead of touching every feature a bit.	
	Must be able to prioritize the requirement step by step/ The requirement prioritization phase should be completed properly	
	Implement WBA through JIRA platform normally use in scrum	

	To overcome this issue, existing systems should be studied to understand how partitioning is done in them and what is the ratio between success and failure.	
	Various partitioning methods need to be implemented and evaluated. And the best strategy to be implemented.	
Lack of useful information extraction	Automated sensors/ introduce automated/ embed automated sensors/ use automated sensors/ by deploying automated sensors	91.4%
	We should create a report that when for fewer mobile users and then run the cron jobs that schedule for the specific time when the public don't use their mobile phone so we can achieve heavy data transmissions easily like backup plans and updates.	
	Involve the usage of automated sensors that must be activated when required. In this way, we can overcome scheduling and energy issues and hence, extraction of useful information is there.	
	Use data processing tools to handle these scenarios/ tools usage	
	Secure data transmission should be followed	
	Threading in background service	
Changing the orientation of the app	Hire UI experts/ UI experts? Recruit UI specialists/ hire UI domain experts	93.2%
	Set the executable Prototype before live release	
	We should calculate the resolutions and size of the device whenever someone changes the orientation of the screen and then apply the calculated size with font text and widget size. So we can have better results on both layouts.	
	Bootstrap mobile can help reduce the above issue	

	Developer must save the instance of the application, that when the orientation changes, it doesn't affect any event	
	The app orientation should be managed for each feature separately. If a feature does not require orientation change or the user is unlikely to change device orientation for that particular feature, then it should be set fixed.	
	If orientation changes are required for a feature, then it should be made sure that the UI/UX does not break.	
	This might be a challenge in some apps features like when displaying a table that has many columns that we can't show it on portrait mode so we had to shift screen to landscape mode.	
	If you want to manually handle orientation changes in your app you must declare the "orientation", "screen Size", and "screen Layout" values in the android: configChanges attributes. You can declare multiple configuration values in the attribute by separating them with a pipe | character.	
Diversity of mobile surroundings	Use hybrid platforms/ implement hybrid frameworks	93.9%
	We should start using cross-platform in that case so we need to do coding in one language and run apps on multiple platforms like we can use Flutter, React Native, Ionic, Xamarin, etc	
	Testing through emulators	
	A unified development platform needs to be implemented to overcome this issue.	
	If it is required to target a single platform e.g. Android or iOS, then the diversity does not affect the developer.	

	If multiple platforms are being targeted, then it is needed to understand the components of each platform to provide a consistent user experience across all platforms.	
	We have now hybrid platforms to tackle such challenges. We have React-native, Flutter, and Ionic that allow the developer to create one app that will run on both iOS and Android OS.	
	Cross-platform language like reactive and node JS should be used	
Event driven structure	Use APIs & libraries	91.4%
	In that case, we can't give all the access in a single app. We can handle, for example, we have careem for clients and careem captain for drivers and on the other hand, we can create their admin panels on the web so we can manage everything from their admin panel by creating a bridge between apps and admin panel.	
	Each event must be given a priority number and handled accordingly	
	Trained and skillful developers must be there	
Complex contextual features	Use hybrid platforms	93.1%
	In that case, Firebase provides us the facility where we can implement notifications and other services easily. And the solution for sensor dependency we should select the development environment where we can do these things easily.	
	Specific professionals with strong skills need to be hired.	

	We can tackle such challenges by working on hybrid platforms like React-native, Flutter, and Ionic if these platforms don't give requirement OS compatibility issues. But if the requirement can't be done on hybrid platforms then we need to hire an android developer and iOS developers	
Frequent changing requirements	Implement agile methodologies	96.9%
	Flow the scrum through proper UAT	
	In that case, we should use flexible architecture where we can refactor things easily as we can have MVVM, MVC design pattern.	
	Current and future requirements need to be identified at the start of the project design. Moreover, proper software development should be applied to make applications scalable and easy to extend to cater any unforeseen requirements of the future.	
	It should be made sure during the requirements gathering phase that all the needs of stakeholders are understood correctly. Spending more time on better requirements engineering can reduce frequent changes during development.	
	Before development get started, the app must go through with design first, when the user approved all the designs then the development should start, later few things can be changed over time.	
Lack of appropriate resource allocation	Re-allocate resources/ re-define resource allotment	95.6%
	PM involvement	
	This is the problem of project management. All tasks may be allocated reasonable resources to complete the task within the forecasted time. Project managers should design and investigate project progress to the micro-level to assess the milestone achievement. This will help to procure more resources if initial estimates were not correct	

	That's where the project manager's job comes, he should acknowledge HR or C.T.O that we need this number of resources for app development. And later they can move in or out developers based on remaining tasks.	
Platform incompatibility	Use Hybrid platforms	93.1%
	It's necessary for project manager to understand the app requirements and then make decision whether to build that app on Hybrid platforms or Native platforms. If any requirement that only works on Native platform then development team must be hired that knows how to work on Native platform and vice versa.	
	Used Up to date technology	
	Teams must develop a solid architectural plan and should go for multi-platforms	
	Use cross platform development languages	
Lack of resource optimization	Re-allocate the resources/ re-define resources Allotment	17%
	Optimize the operational elements	
	Involve your PM/ conduct meetings with PM	
	We should take only those resources on-board with experience, based on complexity of app. If app is not complex like the idea is to build an app that is similar to the market apps then we should hire resources with average experience. But if the app is complex or have innovative ideas or requirements then we must take those resources on-board with high experience.	
	Use optimized website which can use minimum Resources	
	Execute energy consuming tasks on cloud/	23.5%

Energy inefficiency	transfer heavy tasks on cloud/execute heavy tasks on cloud/ transfer energy consuming tasks to cloud/ migrate heavy tasks and execute on the cloud migrate your heavy tasks on cloud/ Outsource the computation-intensive tasks to the cloud	
	This is a claim that has to be evaluated through proper research. Training for energy-efficient mobile computation needs to be conducted. Academia can play a major role to develop the skills of graduates by adopting advanced level courses in mobile application development.	
	Sometimes backend lacks to provide the computation of intensive tasks or sometimes the device OS. The backend team must create a structure for such intensive tasks so that they won't consume more energy on devices or overload the frontend side.	
Inaccurate task scheduling	Implement task scheduling algorithms/ use task scheduling algorithms/ by using task scheduling Algorithms	88%
	Schedule your tasks properly	
	Scrum meetings must be conducted	
	Task sheet would be maintained with the help of any tool, so we would easily check about the resources Scheduling	
	A research and development department has to be initiated in each company. A specialized solution to the estimation of scheduling in a mobile environment needs to be adopted. It would also be interesting to understand the performance of traditional estimation methods in mobile environments.	

	This challenge can be tackled by proper R&D and scrum meetings. So everyone must know information related to their tasks. Hence, they can provide a fair estimate/ conduct scrum meetings and use task scheduling algorithms that suit your requirements	
	Team lead/Project Manager should be experienced to assess this/ Project management responsibility	
	A proper project plan should be developed and followed	
Limited resources/resources lacking	Use best coding techniques/ best coding techniques and practices/ implement best coding guides and frameworks	21.1%
	PM involvement/ involve your project managers/ meeting with project manager/ project manager role	
	Use an optimized website with minimum resources	
Lack of computational resources	Use external memory chipsets/ embed extra memory silicon chips/ use external memory	91.3%
	We have now devices that provide the best performance chipsets and memory, still can't compete with the conventional PC but does the job easily for high intensive tasks.	
	Use an optimized website that can use minimum resources in terms of hardware requirements, content and other services	
Incorrect estimation of battery life	Embed energy consumption mechanisms to mobile devices	87.5%
	Include the feature that can track phone battery	
	mobile device developers are responsible to integrate such mechanisms to handle such issues	

	Don’t compromise on battery/ battery quality high	
	If your task is related to the battery you can resolve it by checking the estimated time of battery in your application if it is not suitable you can give an alert message to the user to connect their mobile to charge before performing a task.	
	It’s not the mobile developer's responsibility to estimate the battery life of a mobile. But in some scenarios, the app required a feature to display phone battery or track phone battery so we took battery life from the device.	
	With fast charging support and long battery life, this issue is not that alarming until there is a bug	
Lack of accurate quantification about the consumption of energy by the app	Implement standard guides & practice/ This is a research challenge and it would remain an open problem. A Research and Development team should generate guidelines for developers for overcoming this issue.	98.7%
	measure the app energy first	
	Include a feature that automatically tracks or predict the energy used by the app	
	allow notification or automatic prediction of energy consumed by app	
	can make algorithms to pre estimation	
	With fast charging support and long battery life, this the issue is not that alarming until there is a bug	
Compatibility across various OS versions	use of cross platforms/ cross-platform usage/ by using cross platforms	93.8%
	Programmed according to different O.S. systems.	
	Tested on the targeted device and targeted OS versions	

	Google provides a good platform to test and optimize its apps. However, more popular android versions may be tested for efficient operation of apps.	
	Proper testing on different OS versions to be done before releasing the app	
	Compatibility testing	
Lack of asynchrony retrofitting	Use proper defined async tasks/ use of advanced technologies and platforms with defined async tasks/ by using advanced technologies and platforms with proper defined async tasks/ use of latest platforms having defined async tasks	92.7%
	Use of advanced technology like react-native	
	Train your mobile developers, conduct workshops and educate them to become domain specialists/ train your mobile developers and make them skillful and domain experts/ train your mobile developers so that they can properly utilize async constructs/	
	Hire well-experienced and expert mobile developers/ hire expert mobile developers	
	This is a developer weakness and it must be dealt with proper training in the field or educational institutes.	
Tampering during offloading data	security and encryption mechanisms/ embed security and encryption mechanism/ implement security and encryption mechanisms	94.8%
	Tested through different security protocols via different OS versions	
	Encryption techniques or tokens are used	
	Secure and trusted cloud platforms should be selected.	
	The data engineers or backend developers make the data stored securely by using tokens or encrypting them. Many people don't want their info like credit card info in the wrong hands. So we use secured	

	methods like stripe to implement a secure payment.	
	Use Owasp coding practices	
Limitation of profilers	Efficient servers/ use properly managed servers/ use efficient servers/ efficient servers should be used for query execution and fast response/ Use of properly managed servers to execute each query efficiently	93.4%
	New offloading mechanisms need to develop to overcome these issues. These problems will remain to be a challenge for a while. However, R&D can develop improved algorithms.	
	Performances testing required	
Lack of identification of risky actions and vulnerabilities	Risk management techniques/ implement RM techniques/ risk management techniques should be implemented/ RM techniques must be followed/ follow RM techniques	92.5%
	Only apps downloaded from certified or trusted platforms should be allowed to communicate by the OS	
	Security testing	
Inconsistent and inefficient testing	Release beta version of your product and take customer feedback/ beta version product	96.25%
	Tested through higher-order test cases, set the proper mobile test plan and implement V&V model through manual and automated testing	
	Tested through higher-order test cases, set the proper mobile test plan, and implement V&V model through manual and automated testing	

	This is a known constraint and with the improvement in developing methodologies and technology, it will be reduced/ improving the development methods	
	hire a good SQA resource	
	End to end testing	
Lack of configuration of mobile hybrid apps	Review code/ Code review approach/ By reviewing the code/ Review your code thoroughly	92.7%
	Mobile development needs to develop proper SE processes to identify defects and bugs by rigorous testing before launch. The urgency to release products without proper testing is the main culprit.	
	Revert code	
	This challenge can be tackled by code review. The senior frontend developer review the code with every code pushed to the cloud.	
	A thorough test plan should be developed to ensure secure cross-platform app testing	
	Configuration testing	
Unclear requirements for app functionality towards privacy threats	Security and privacy mechanisms/ Implement security and privacy mechanisms	93.3%
	Set the privacy work scheme and implement V&V model for test the privacy work scheme	
	Data security mechanisms should be followed Properly	
	It is always difficult to handle privacy challenges. Mobile development is not so different. Rigorous implementation of privacy policies needs to incorporate at the OS level. Moreover, users need to educate in terms of these policies and their implications.	
	We need to gather privacy policies from the user. And list some important ones based on app requirements.	
	Secure coding practices	

Patching for updation, correction or improvement	Use of API can fix your patches or patching issues/ Use APIs/ Patch Management	86.9%
	This challenge can be tackled by using third-party apps or libraries that provide support and fix issues, you can track it by watching them on Git Hub.	
Data erasure	Embed encryption mechanisms/ Encryption techniques/ Use encryption techniques and algorithms	18.5%
	Yes, this is a major issue faced in the performance of android. However, the problem is only solvable through standardization and adaption of unified a platform for Android OS development.	
	API testing	
Unawareness of needs	Meetings with stakeholders/ Conduct meetings with stakeholders/ Involve your Stakeholders	94.7%
	We need to gather information for the app requirements before start development.	
	With new development platforms and tools, this the challenge is slowly reducing	
Cultural and language barrier	Use of official language/ Communicate using official language	88%
	Software ethics should be followed for the development of an app	
	Cultural diversity understanding/ Must understand the diversity of culture	
Lack of domain knowledge	Training sessions to mobile developers/ Train your mobile developers/ Conduct training for hired developers	93%
	Project manager experience plays an important role. He knows what is the best approach to follow to make the app efficient and user friendly/ Involvement of PM	
	Domain knowledge would be given first	

	Consult domain expert before implementing task/ Do involve domain experts or allow training sessions to train your mobile developers	
Ambiguities among stakeholders	Meetings with stakeholders/ Conduct meetings with stakeholders/ Frequent meetings with stakeholders	95.1%
	Business analytics communicates with stakeholders and tries to gather their views and solve their ambiguities by providing the best solutions.	
	Documentation/ Once the User Specification the document is finalized, it has to be followed	
Intragroup conflicts	Involve your project manager/ PM involvement/ Meetings with PM	96.9%
	Set the proper UAT for resolving intragroup conflicts	
	Follow the decision and instructions of the project Managers	
	All stakeholders of the system should be on board while deciding on some important issue	
	A project manager does meeting with senior developers and other developers. And provide the best approach for each person.	
Inefficient response time	Use async programming concepts/ Apply async programming concepts/ Implement async programming concepts	96.3%
	That's why we use asynchronous tasks so that we can wait for the execution to work properly and then shows the correct data. We can optimize code by doing multi-lines of code into short codes.	
	Performance testing tools	
	Team lead/Project manager involvement	
Lack of requirement task	Asynchronous programming concepts implementation/ Use of async programming concepts	91.7%
	Responsive design testing, performance testing	

efficiency and responsiveness	We use promises and asynchronous awaits to have a proper response from the servers or internal executions to avoid memory leaks & responsiveness issues.	
Lack of development standards and practices knowledge	Latest and updated document of development platforms must be used/	93.8%
	Implement CMMI protocol/ CMMI guide/ Follow the protocols of CMMI/ Protocols of CMMI must be followed to mitigate such a challenge	
	Use best coding practices and standards, CMMI Protocol	
	Best practices and standards should be followed	
	The best approach is to follow the latest documentation of development platforms. We will face enormous issues and lags if we stick to old approaches because with time the development is getting towards more optimized techniques.	
	Common standard should be followed for all Employees	
Testing issues for practitioners	QA team involvement/ QA team/ Involve QA team/ Conduct meeting with QA team	94.2%
	Implement V& V model	
	The Quality Assurance team tests the app in every aspect. They have tools like selenium where they test the app and find out bugs and security issues. And then they can address developers to fix them.	
	A proper test plan should be developed before the start of the development process	

The mitigation plans tabulated above in table 5.1 are the solutions against the discovered Requirement Engineering challenges faced by mobile developers during phase 1 of the research study is SLR conduction. It is clearly shown from the table with practicality level that the most frequent challenge faced by the mobile developers is “Lack of accurate quantification

about the consumption of energy by the app" with 98.7% practicality level. The next challenge faced by mobile developers less than "Lack of accurate quantification about the consumption of energy by the app" is "Change of user need and understanding" with 97.7% of the practicality level. Similarly, the order of the challenges along with the practicality level with ascending order from higher to lower is shown below in table 5.2. The table is composed of 3 columns named Serial Number, Respective Challenge, and Practicality Level of Challenge respectively.

Table 5.2: Challenges with Practicality level shown in ascending order from higher to lower

Sr. No.:	Respective Challenge	Practicality Level
1	Lack of accurate quantification of the energy consumption by the App	98.7%
2	Change of user needs & understanding	97.7%
3	Lack of communication participation	97.6%
4	Unstable requirements	97.5%
5	Lack of verbal & presentation skills	97%
6	Frequent changing requirements	96.9%
7	Intra group conflicts	96.9%
8	Lack of consideration of user & application requirements for offloading decision making	96.7%
9	Incomplete requirement gathering	96.5%
10	Anonymous communication	96.4%
11	Lack of requirement effective articulation	96.4%
12	Incorrect requirement partitioning	96.3%
13	Inefficient response time	96.3%
14	Requirement over-scoping	96.3%
15	Inconsistent & inefficient testing	96.25%
16	Insufficient requirement completion time	96.25%
17	Lack of appropriate resource allocation	95.6%
18	Ambiguities among stakeholders	95.1%
19	Tampering during offloading data	94.8%

20	Unawareness of needs	94.7%
21	Testing issues for practitioners	94.2%
22	Diversity of mobile surroundings	93.9%
23	Lack of accurate requirement prioritization	93.9%
24	Compatibility across various platforms	93.8%
25	Lack of development standards & practices knowledge	93.8%
26	Limitation of profilers	93.4%
27	Unclear requirements for app functionality towards privacy threats	93.3%
28	Changing the orientation of an app	93.2%
29	Platform incompatibility	93.1%
30	Complex contextual features	93.1%
31	Lack of domain knowledge	93%
32	Lack of asynchrony retrofitting	92.7%
33	Lack of configuration of hybrid mobile apps	92.7%
34	Lack of identification of risky actions & vulnerabilities	92.5%
35	Lack of requirement task efficiency & responsiveness	91.7%
36	Event driven structure	91.4%
37	Lack of useful information extraction	91.4%
38	Lacking of computational resources	91.3%
39	Inaccurate task scheduling	88%
40	Culture & language barrier	88%
41	Incorrect estimation of battery life	87%
42	Patching for updation, correction & improvement	86.9%
43	Energy inefficiency	23.5%
44	Limited resources/ resource lacking	21.1%
45	Data erasure	18.5%
46	Lack of resource optimization	17%

The above table is clearly showing that a total of 38 challenges are lying in the range of 91%-98%. This means that the 38 listed challenges from top to bottom that is from “Lack of accurate quantification of the energy consumption by the app” to “Lack of computational resources” are the most occurring and frequent challenges faced by the mobile developer during

requirement engineering process execution. 4 challenges lie under the range of 88% to 86% that is from “Inaccurate task scheduling” (Challenge # 39) to “Patching for updation, correction or improvement” (Challenge # 42). Whereas, the challenges with low probability are “Energy inefficiency” (23.5%), “Limited resources” (21.1%), “Data erasure” (18.5%), and “Lack of resource optimization” (17%).

5.3 Summary of the Chapter:

This chapter is all about the Industrial survey findings. Based on the research study, the RQ3 was to propose the resolutions or solution strategies for the challenges faced during the accomplishment of the Requirement Engineering process, particularly in the mobile app development domain. Along with the solution strategies, the distribution of respondents taking part in the industrial survey is presented in pie-chart and bar graph form.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Overview

This chapter will conclude all findings of the research study along with the brief contributions made in the research thesis in the form of RQ1, RQ2, and RQ3. The research study was conducted to acknowledge the Requirement Engineering Process challenges during its implementation, particularly in mobile application development.

6.2 Summary of the Contribution:

RQ1: What are the possible key challenges faced by the mobile developers for mobile application development during the R.E. process implementation?

A total of 43 primary studies is being selected during the conduction of phase 1 is SLR by exploring 4 electronic databases like IEEE Xplore, Science Direct, Wiley Online Library, and ACM Digital Library. As a result of SLR conduction, a list of 46 challenges has been discovered after extracting the relevant constructs from the Encoding Technique of Grounded Theory and validation from phase 2 of Expert Opinion or Expert Review conduction. Based on standard criteria, 4 Experts were chosen to evaluate the identified list. As per the suggestions given by the Experts', the final list is obtained as an output of phase 2.

RQ2: How many possible categories do these acknowledged challenges have?

The output obtained as a result of Grounded Theory, a list of extracted constructs are being categorized into 5 sub-categories as Communication-related, Requirement-related, Resource-related, Security & Privacy-related, and Stakeholder-related based on the main category that is Nature of Challenge. The complete list of categorized challenges is described in Appendix Section shown in Appendix N.

RQ3: What are the mitigation plans to overcome the discovered challenges?

The validated list from phase 2 of Expert opinion is passed through the next phase which is phase 3 of the Industrial survey. The main purpose of conducting an Industrial Survey was to come up with the Resolutions against each identified Requirement Engineering challenge faced by mobile developers in mobile application development. As a result of the Industrial Survey, a total of 259 solution strategies with overall 97 responses were achieved. But, due to some incomplete responses, 17 results were being eliminated from the survey findings, hence achieving 80 responses in total. A total of 7 solutions with 84 responses for the challenge "Anonymous Communication" is obtained. The most suggested strategy was to "Integration of Encryption- Decryption Mechanisms". A challenge with the name "Lack of Requirement Effective Articulation" with 12 strategies and 83 responses was obtained. The most probable answer was "Conduct Stand-up Meetings or Daily Scrum Meetings". 11 mitigation plans with 90 responses were obtained for the challenge named "Lack of Verbal & Presentation Skills". The most occurring solution given was "Implementation of UML". The challenge with the title "Lack of Communication Participation" is found with 3 solutions and 82 responses. The most given answer to this issue was "Communicate via Models or Construct Models". The challenges discussed above wholly come under the Communication-related category. The summary of complete challenges along with the most occurring mitigation plan as explained in the above format is tabulated below in Table 6.1:

6.3 Threats of Validity

In conducting this research study, we need to be focused on several factors while generalizing the results. To begin, we looked at publications that were published and unpublished both during the process of finding relevant material. But, we only considered accepted manuscripts for unpublished material. However, studies that may have appeared in journals or mature conference proceedings that were not published may have been overlooked.

Secondly, most of the studies found were special issue papers in the Wiley Online Library, which were excluded. Because including these papers means to go out of the box from our research scope.

We have only focused on general applications excluded specific ones. For example; applications related to medical health care, AR and VR-based applications, context-aware applications, location-based applications and graphics applications, and so on.

The primary studies with no validation or testing on some scale were simply excluded.

As we have generated 176 search strings in total based on keywords and related synonyms, the found literature was volumetric. For this purpose, we reduced our time frame range that is from 2005-2020 to 2010-2020.

6.4 Future Work

In our future work, we are planning to include specific applications in our research study. As we have focused on general mobile applications and we have discussed challenges related to these applications only throughout- our research study. We will increase our scope size in terms of adding more different types of publications as Special Issue papers that were found in large volume from literature.

6.5 Conclusion

The development of software applications particularly for mobile devices is a prevalent phenomenon. But, mobile application development is still in its infancy when it is integrated with the Requirement Engineering process. So, that is what our research is all about, basically the merger of 2 domains based on the Requirement Engineering process with Mobile App Development, as it is different from that of traditional or web development. The research study is focused on 3 phases. Phase 1 based on SLR conduction, is about the Challenges Identification for Requirement Engineering Process particularly for Mobile App Development, phase 2 is about the Validation of Challenges via Expert Review conduction while, phase 3 is all about proposing the Resolution Strategies to overcome the discovered challenges from SLR. A total of 43 primary studies have been explored and as a result, a total of 47 challenges are being acknowledged with 259 Resolution Strategies in total along with the practicality level of the respective challenge. In addition to the aforementioned contribution, the identified challenges are also categorized based on the Nature of the Challenge as Communication-related, Requirement-related, Resource-related, Security & Privacy-related, and Stakeholder-related. This research study may guide the practitioners and the academicians towards the Requirement Engineering Process for Mobile Application Development.

References

[1] S. Zein, N. Salleh, and J. Grundy, "A systematic mapping study of mobile application testing techniques," *J. Syst. Softw.*, vol. 117, pp. 334–356, 2016, doi: 10.1016/j.jss.2016.03.065.

[2] B. Ian, "Mobile App Download and Usage Statistics (2021)," *Buildfire*, 2021. [Online]. Available: https://buildfire.com/app-statistics/.

[3] R. Gibb, "What is a Web Application?," 2016.

[4] C. Petrov, "52 Mobile vs. Desktop Usage Statistics For 2019," *Tech Jury*, 2019. [Online]. Available: https://techjury.net/stats-about/mobile-vs-desktop-usage/.

[5] F. Schneider, "What is your favorite definition of the term 'requirement'?," *researchgate*, 2013.

[6] Geeksforgeeks, "Software Engineering | Requirements Engineering Process," *Geeksforgeeks*, 2020. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/.

[7] A. goleman, daniel; boyatzis, Richard; Mckee, "Requirement Engineering," *Journal of Chemical Information and Modeling*, 2019.

[8] D. Zanutto, E. C. Lorenzini, R. Mantellato, G. Colombatti, and A. Sanchez-Torres, "Software Engineering Issues for Mobile Application Development," in *Proceedings of the International Astronautical Congress, IAC*, 2012, vol. 4, pp. 2577–2585.

[9] H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, and B. Shahzad, "A systematic study on software requirements elicitation techniques and its challenges in mobile application development," *IEEE Access*, vol. 6, pp. 63859–63867, 2018, doi: 10.1109/ACCESS.2018.2874981.

[10] Bit Mascot, "Top 10 Challenges Faced by Mobile App Developers - Bit Mascot," *Bit Mascot*, 2016. [Online]. Available: https://www.bitmascot.com/top-10-challenges-faced-mobile-app-developers/.

[11] Paul Goodman, "18 Disadvantages of Mobile Phones | TurboFuture," 2018. [Online]. Available: https://turbofuture.com/cell-phones/Disadvantages-of-Mobile-Phones.

[12] Jamie Appleseed, "8 Limitations When Designing For Mobile - Articles - Baymard Institute," 2012. [Online]. Available: http://baymard.com/blog/mobile-design-limitations.

[13] D. Loomba, "5 major challenges of mobile app testing," *taraspan*, 2021. .

[14] N. Levy, "Mobile Development Challenges," *hackernoon*, 2020. [Online]. Available: http://www.slideshare.net/levynir/mobile-development-challenges-10288414.

[15] B. Dolly and M. A. Khanum, “Requirement Elicitation in Mobile Apps : A Review,” 2016, pp. 242–247.

[16] A. Ashraf, M. N. Tahir, S. Khan, and A. Raza, “Challenges in Requirement Elicitation of Cellular Applications : For Visually Challenged People,” *J. Ind. Intell. Inf.*, vol. 1, no. 2, pp. 226–230, 2013, doi: 10.12720/jiii.1.4.226-230.

[17] H. K. Flora and D. S. V. Chande, “A Review and Analysis on Mobile Application Development Processes using Agile Methodologies,” *Int. J. Res. Comput. Sci.*, vol. 3, no. 4, pp. 8–18, 2013, doi: 10.7815/ijorcs.34.2013.068.

[18] S. Kaleel and S. Harishankar, “Applying Agile Methodology in Mobile Software Engineering: Android Application Development and Its Challenges,” *Comput. Sci. Tech. Reports*, p. 11, 2013.

[19] M. UsmanMalik, N. Majeed Chaudhry, and K. Shahzad Malik, “Evaluation of Efficient Requirement Engineering Techniques in Agile Software Development,” *Int. J. Comput. Appl.*, vol. 83, no. 3, pp. 24–29, 2013, doi: 10.5120/14429-2574.

[20] M. Sadiq and S. K. Jain, “An insight into requirements engineering processes,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2012, vol. 108 LNICST, pp. 313–318, doi: 10.1007/978-3-642-35615-5_48.

[21] D. Pandey, U. Suman, and A. K. Ramani, “An effective requirement engineering process model for software development and requirements management,” in *Proceedings - 2nd International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom 2010*, 2010, pp. 287–291, doi: 10.1109/ARTCom.2010.24.

[22] E. By, “International Journal of Software Engineering (Ijse),” 2011.

[23] Z. Jin, “Requirement Engineering Methodologies,”.

[24] G. Canfora, F. Mercaldo, C. A. Visaggio, M. D’Angelo, A. Furno, and C. Manganelli, “A case study of automating user experience-oriented performance testing on smartphones,” in *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation, ICST 2013*, 2013, pp. 66–69, doi: 10.1109/ICST.2013.16.

[25] L. Lu, Y. Hong, Y. Huang, K. Su, and Y. Yan, “Activity page based functional test automation for android application,” in *Proceedings of the 2012 3rd World Congress on Software Engineering, WCSE 2012*, 2012, pp. 37–40, doi: 10.1109/WCSE.2012.15.

[26] Z. Liu, X. Gao, and X. Long, “Adaptive random testing of mobile application,” in *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, 2010, vol. 2, doi: 10.1109/ICCET.2010.5485442.

[27] V. E. S. Souza, “Requirements-based Software System Adaptation,” 2012.

[28] B. Dolly and M. A. Khanum, “Requirement Elicitation in Mobile Apps : A Review,” pp. 242–247, 2016.

[29] J. McWherter and S. Gowell, *Professional mobile Application Development*, vol. 53, no. 9. 2012.

[30] M. Qiu, W. Dai, and K. Gai, *Mobile Applications Development with Android*. 2016.

[31] “Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique.”

[32] S. Agrawal and A. I. Wasserman, “Mobile Application Development: A Developer Survey,” *Submitt. Publ.*, 2010.

[33] R. Solutions, “Mobile Usage Statistics and Trends 2016 ©,” 2016.

[34] A. Shakarami, A. Shahidinejad, and M. Ghobaei-Arani, “A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective,” *Softw. - Pract. Exp.*, vol. 50, no. 9, pp. 1719–1759, 2020, doi: 10.1002/spe.2839.

[35] X. Wang, A. Continella, Y. Yang, Y. He, and S. Zhu, “LeakDoctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications,” *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol. Artic. No. 28*, vol. 3, no. 1, p. page 1–25, 2019.

[36] R. Kobayashi and K. Adachi, “Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing,” *IEEE Access*, vol. 7, pp. 141119–141132, 2019, doi: 10.1109/ACCESS.2019.2944184.

[37] X. Wang, R. Slavin, and T. D. Breaux, “GUILeak : Tracing Privacy Policy Claims on User Input Data for Android Applications,” pp. 37–47, 2018, doi: 10.1145/3180155.3180196.

[38] D. Amalfitano, V. Riccio, A. C. R. Paiva, and A. R. Fasolino, “Why does the orientation change mess up my Android application? From GUI failures to code faults,” *Softw. Test. Verif. Reliab.*, vol. 28, no. 1, pp. 1–27, 2018, doi: 10.1002/stvr.1654.

[39] A. AlJarrah and M. Shehab, “Cordovaconfig: A tool for mobile hybrid apps’ configuration,” in *ACM International Conference Proceeding Series*, 2018, pp. 161–170, doi: 10.1145/3282894.3282931.

[40] J. Shu, Y. Zhang, J. Li, B. Li, and D. Gu, “Why data deletion fails? A study on deletion flaws and data remanence in android systems,” *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 2, pp. 1–22, 2017, doi: 10.1145/3007211.

[41] Y. Lin, S. Okur, and D. Dig, “Study and refactoring of android asynchronous

programming," in *Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*, 2016, pp. 224–235, doi: 10.1109/ASE.2015.50.

[42] M. N. Tahir, S. Khan, and A. Raza, "Challenges in Requirements Engineering for Mobile Applications for Disabled –Autism," *J. Ind. Intell. Inf.*, vol. 1, no. 4, pp. 226–230, 2013, doi: 10.12720/jiii.1.4.226-230.

[43] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," *Tech. the report, Ver. 2.3 EBSE Tech. Report. EBSE*, 2007.

[44] Y. Zhou, H. Zhang, X. Huang, S. Yang, M. A. Babar, and H. Tang, "Quality assessment of systematic reviews in software engineering: A tertiary study," *ACM Int. Conf. Proceeding Ser.*, vol. 27-29-Apri, 2015, doi: 10.1145/2745802.2745815.

[45] K. Charmaz, "Grounded Theory: Methodology and Theory Construction," *Int. Encycl. Soc. Behav. Sci. Second Ed.*, pp. 402–407, 2015, doi: 10.1016/B978-0-08-097086-8.44029-8.

[46] M. Vollsteedt and S. Rezat, *Compendium for Early Career Researchers in Mathematics Education*. 2019.

[47] B. Ayyub, "A practical guide on conducting expert-opinion elicitation of probabilities and consequences for corps facilities," *Inst. Water Resour. Alexandria, VA, USA*, no. January, 2001.

[48] Experienceux, "What is an expert review?," 2016. [Online]. Available: http://www.experienceux.co.uk/faqs/what-is-an-expert-review/.

[49] M. Kasunic, "Designing An Effective Survey.," *Softw. Eng. Inst.*, no. September, p. 140, 2005.

[50] statisticsfun, "How to calculate Sample Size," *Journal of thoracic disease*, 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22263004%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3256489%0Ahttp://www.youtube.com/watch?v=Z2dKK1xicgs&feature=youtube_gdata_player.

[51] "Calculation of sample size in pilot studies," *crutzen.net*. [Online]. Available: http://www.crutzen.net/n.htm.

[52] K. Sakai, M. Te Sun, W. S. Ku, and J. Wu, "A framework for anonymous routing in delay tolerant networks," in *Proceedings - International Conference on Network Protocols, ICNP*, 2017, pp. 1–10, doi: 10.1109/ICNP.2017.8117531.

[53] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative

resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, 2013, doi: 10.1109/JSAC.2013.131209.

[54] Y. Liu and M. J. Lee, "Security-aware resource allocation for mobile cloud computing systems," in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 2015, doi: 10.1109/ICCCN.2015.7288465.

[55] W. Junior, E. Oliveira, A. Santos, and K. Dias, "A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment," *Futur. Gener. Comput. Syst.*, vol. 90, pp. 503–520, 2019, doi: 10.1016/j.future.2018.08.026.

[56] M. Goudarzi, M. Zamani, and A. T. Haghighat, "A fast hybrid multi-site computation offloading for mobile cloud computing," *J. Netw. Comput. Appl.*, vol. 80, pp. 219–231, 2017, doi: 10.1016/j.jnca.2016.12.031.

[57] A. Antonić, M. Marjanović, K. Pripužić, and I. Podnar Žarko, "A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 607–622, 2016, doi: 10.1016/j.future.2015.08.005.

[58] H. Zhao, M. Chen, M. Qiu, K. Gai, and M. Liu, "A novel pre-cache schema for high-performance Android system," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 766–772, 2016, doi: 10.1016/j.future.2015.05.005.

[59] P. Angin, B. Bhargava, and R. Ranchal, "A self-protecting agent's based model for high-performance mobile-cloud computing," *Comput. Secur.*, vol. 77, pp. 380–396, 2018, doi: 10.1016/j.cose.2018.04.011.

[60] Y. W. Kao, C. Lin, K. A. Yang, and S. M. Yuan, "A web-based, offline-able, and personalized runtime environment for executing applications on mobile devices," *Comput. Stand. Interfaces*, vol. 34, no. 1, pp. 212–224, 2012, doi: 10.1016/j.csi.2011.08.006.

[61] W. L. Zhang, B. Guo, Y. Shen, D. G. Li, and J. K. Li, "An energy-efficient algorithm for multi-site application partitioning in MCC," *Sustain. Comput. Informatics Syst.*, vol. 18, pp. 45–53, 2018, doi: 10.1016/j.suscom.2018.02.008.

[62] G. Folino and F. S. Pisani, "Automatic offloading of mobile applications into the cloud by means of genetic programming," *Appl. Soft Comput. J.*, vol. 25, pp. 253–265, 2014, doi: 10.1016/j.asoc.2014.09.016.

[63] M. Goudarzi, M. Zamani, and A. Toroghi Haghighat, "A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing," *Int. J. Commun. Syst.*, vol. 30, no. 10, pp. 1–13, 2017, doi: 10.1002/dac.3241.

[64] G. A. Zhang, J. Y. Gu, Z. H. Bao, C. Xu, and S. B. Zhang, “AGILE: A terminal energy-efficient scheduling method in mobile cloud computing,” *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 3, pp. 294–307, 2014, doi: 10.1002/ett.

[65] M. Usman, M. Z. Iqbal, and M. U. Khan, “An automated model-based approach for unit-level performance test generation of mobile applications,” *J. Softw. Evol. Process*, vol. 32, no. 1, pp. 1–31, 2020, doi: 10.1002/smr.2215.

[66] Y. H. Wang and I. C. Wu, “An evaluation framework for cross-platform mobile application development tools,” *Softw. - Pract. Exp.*, vol. 39, no. 7, pp. 701–736, 2015, doi: 10.1002/spe.

[67] A. Shahidinejad and M. Ghobaei-Arani, “Joint computation offloading and resource provisioning for edge-cloud computing environment: A machine learning-based approach,” *Softw. - Pract. Exp.*, vol. 50, no. 12, pp. 2212–2230, 2020, doi: 10.1002/spe.2888.

[68] A. Nirumand, B. Zamani, and B. Tork Ladani, “VAnDroid: A framework for vulnerability analysis of Android applications using a model-driven reverse engineering technique,” *Softw. - Pract. Exp.*, vol. 49, no. 1, pp. 70–99, 2019, doi: 10.1002/spe.2643.

[69] S. Baride and K. Dutta, “A cloud based software testing paradigm for mobile applications,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 3, pp. 1–4, 2011, doi: 10.1145/1968587.1968601.

[70] Y. K. Lee, J. Y. Bang, G. Safi, A. Shahbazian, Y. Zhao, and N. Medvidovic, “A sealant for inter-app security holes in android,” *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017*, no. 4. pp. 312–323, 2017, doi: 10.1109/ICSE.2017.36.

[71] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, “An online algorithm for task offloading in heterogeneous mobile clouds,” *ACM Trans. Internet Technol.*, vol. 18, no. 2, 2018, doi: 10.1145/3122981.

[72] M. Hesenius, T. Griebe, S. Gries, and V. Gruhn, “Automating UI tests for mobile applications with formal gesture descriptions,” *MobileHCI 2014 - Proc. 16th ACM Int. Conf. Human-Computer Interact. with Mob. Devices Serv.*, pp. 213–222, 2014, doi: 10.1145/2628363.2628391.

[73] K. Moran, M. Linares-Vasquez, C. Bernal-Cardenas, C. Vendome, and D. Poshyvanyk, “CrashScope: A practical tool for automated testing of android applications,” in *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, 2017, pp. 15–18, doi: 10.1109/ICSE-C.2017.16.

[74] L. Cruz and R. Abreu, “EMaaS: Energy measurements as a service for mobile applications,” in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2019*, 2019, pp. 101–104, doi: 10.1109/ICSE-NIER.2019.00034.

[75] R. Mittal, A. Kansal, and R. Chandra, “Empowering developers to estimate app energy consumption,” in *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2012, pp. 317–327, doi: 10.1145/2348543.2348583.

[76] C. Mulliner, J. Oberheide, W. Robertson, and E. Kirda, “PatchDroid: Scalable third-party security patches for Android devices,” in *ACM International Conference Proceeding Series*, 2013, pp. 259–268, doi: 10.1145/2523649.2523679.

[77] L. Wei, Y. Liu, and S. C. Cheung, “Taming android fragmentation: Characterizing and detecting compatibility issues for android apps,” in *ASE 2016 - Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 2016, pp. 226–237, doi: 10.1145/2970276.2970312.

[78] G. Hecht, O. Benomar, R. Rouvoy, N. Moha, and L. Duchien, “Tracking the software quality of android applications along their evolution,” in *Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015*, 2016, pp. 236–247, doi: 10.1109/ASE.2015.46.

[79] D. He, H. Zheng, L. Li, G. Li, L. Wang, and J. Xue, “Understanding and detecting evolution-induced compatibility issues in android apps,” in *ASE 2018 - Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018, pp. 167–177, doi: 10.1145/3238147.3238185.

[80] M. Usman, M. Z. Iqbal, and M. U. Khan, “A model-driven approach to generate mobile applications for multiple platforms,” in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 2014, vol. 1, pp. 111–118, doi: 10.1109/APSEC.2014.26.

[81] D. Kim, Y. Chon, W. Jung, Y. Kim, and H. Cha, “Accurate prediction of available battery time for mobile applications,” *ACM Trans. Embed. Comput. Syst.*, vol. 15, no. 3, 2016, doi: 10.1145/2875423.

[82] M. Ciman and O. Gaggi, “An empirical analysis of energy consumption of cross-platform frameworks for mobile development,” *Pervasive Mob. Comput.*, vol. 39, pp. 214–230, 2017, doi: 10.1016/j.pmcj.2016.10.004.

[83] Q. H. Nguyen and F. Dressler, “A smartphone perspective on computation offloading—

A survey," *Comput. Commun.*, vol. 159, no. May, pp. 133–154, 2020, doi: 10.1016/j.comcom.2020.05.001.

[84] S. M. Dye and K. Scarfone, "A standard for developing secure mobile applications," *Comput. Stand. Interfaces*, vol. 36, no. 3, pp. 524–530, 2014, doi: 10.1016/j.csi.2013.09.005.

[85] "Anonymous communications," *Journal of the American Medical Association*, 1895.

APPENDICES

APPENDIX A

Table A: Showing a list of Search Strings designed for SLR conduction

Sr. No.:	Search Strings
1	((("requirement engineering") AND "mobile development") AND challenges)
2	((("requirement engineering") AND "mobile platform") AND challenges)
3	((("requirement engineering") AND "mobile applications") AND challenges)
4	((("requirement engineering") AND "mobile application development") AND challenges)
5	((("requirement engineering") AND "mobile development") AND issues)
6	((("requirement engineering") AND "mobile platform") AND issues)
7	((("requirement engineering") AND "mobile applications") AND issues)
8	((("requirement engineering") AND "mobile application development") AND issues)
9	((("requirement engineering") AND "mobile development") AND problems)
10	((("requirement engineering") AND "mobile platform") AND problems)
11	((("requirement engineering") AND "mobile applications") AND problems)
12	((("requirement engineering") AND "mobile application development") AND problems)
13	((("requirement engineering") AND "mobile development") AND barriers)
14	((("requirement engineering") AND "mobile platform") AND barriers)
15	((("requirement engineering") AND "mobile applications") AND barriers)
16	((("requirement engineering") AND "mobile application development") AND barriers)
17	((("requirement engineering process") AND "mobile development") AND challenges)
18	((("requirement engineering process") AND "mobile platform") AND challenges)
19	((("requirement engineering process") AND "mobile applications") AND challenges)

20	((("requirement engineering process") AND "mobile application development") AND challenges)
21	((("requirement engineering process") AND "mobile development") AND issues)
22	((("requirement engineering process") AND "mobile platform") AND issues)
23	((("requirement engineering process") AND "mobile applications") AND issues)
24	((("requirement engineering process") AND "mobile application development") AND issues)
25	((("requirement engineering process") AND "mobile development") AND problems)
26	((("requirement engineering process") AND "mobile platform") AND problems)
27	((("requirement engineering process") AND "mobile applications") AND problems)
28	((("requirement engineering process") AND "mobile application development") AND problems)
29	((("requirement engineering process") AND "mobile development") AND barriers)
30	((("requirement engineering process") AND "mobile platform") AND barriers)
31	((("requirement engineering process") AND "mobile applications") AND barriers)
32	((("requirement engineering process") AND "mobile application development") AND barriers)
33	((("software requirement engineering process") AND "mobile development") AND challenges)
34	((("software requirement engineering process") AND "mobile platform") AND challenges)
35	((("software requirement engineering process") AND "mobile applications") AND challenges)
36	((("software requirement engineering process") AND "mobile application development") AND challenges)
37	((("software requirement engineering process") AND "mobile development") AND issues)
38	((("software requirement engineering process") AND "mobile platform") AND issues)
39	((("software requirement engineering process") AND "mobile applications") AND issues)
40	((("software requirement engineering process") AND "mobile application development") AND issues)

41	((("software requirement engineering process") AND "mobile development") AND problems)
42	((("software requirement engineering process") AND "mobile platform") AND problems)
43	((("software requirement engineering process") AND "mobile applications”) AND problems)
44	((("software requirement engineering process") AND "mobile application development") AND problems)
45	((("software requirement engineering process") AND "mobile development") AND barriers)
46	((("software requirement engineering process") AND "mobile platform") AND barriers)
47	((("software requirement engineering process") AND "mobile applications") AND barriers)
48	((("software requirement engineering process") AND "mobile application development") AND barriers)
49	((("software requirement engineering") AND "mobile development") AND challenges)
50	((("software requirement engineering") AND "mobile platform") AND challenges)
51	((("software requirement engineering") AND "mobile applications") AND challenges)
52	((("software requirement engineering") AND "mobile application development") AND challenges)
53	((("software requirement engineering") AND "mobile development") AND issues)
54	((("software requirement engineering") AND "mobile platform") AND issues)
55	((("software requirement engineering") AND "mobile applications") AND issues)
56	((("software requirement engineering") AND "mobile application development") AND issues)
57	((("software requirement engineering") AND "mobile development") AND problems)
58	((("software requirement engineering") AND "mobile platform") AND problems)
59	((("software requirement engineering") AND "mobile application") AND problems)
60	((("software requirement engineering") AND "mobile application development") AND problems)
61	((("software requirement engineering") AND "mobile development") AND barriers)

62	((("software requirement engineering") AND "mobile platform") AND barriers)
63	((("software requirement engineering") AND "mobile applications") AND barriers)
64	((("software requirement engineering") AND "mobile application development") AND barriers)
65	((("requirement gathering") AND "mobile development") AND challenges)
66	((("requirement gathering") AND "mobile platform") AND challenges)
67	((("requirement gathering") AND "mobile applications") AND challenges)
68	((("requirement gathering") AND "mobile application development") AND challenges)
69	((("requirement gathering") AND "mobile development") AND issues)
70	((("requirement gathering") AND "mobile platform") AND issues)
71	((("requirement gathering") AND "mobile applications") AND issues)
72	((("requirement gathering") AND "mobile application development") AND issues)
73	((("requirement gathering") AND "mobile development") AND problems)
74	((("requirement gathering") AND "mobile platform") AND problems)
75	((("requirement gathering") AND "mobile applications") AND problems)
76	((("requirement gathering") AND "mobile application development") AND problems)
77	((("requirement gathering") AND "mobile development") AND barriers)
78	((("requirement gathering") AND "mobile platform") AND barriers)
79	((("requirement gathering") AND "mobile applications") AND barriers)
80	((("requirement gathering") AND "mobile application development") AND barriers)
81	((("requirement inception") AND "mobile development") AND challenges)
82	((("requirement inception") AND "mobile platform") AND challenges)
83	((("requirement inception") AND "mobile applications”) AND challenges)
84	((("requirement inception") AND "mobile application development") AND challenges)

85	((("requirement inception") AND "mobile development") AND issues)
86	((("requirement inception") AND "mobile 95platform") AND issues)
87	((("requirement inception") AND "mobile applications") AND issues)
88	((("requirement inception") AND "mobile application development") AND issues)
89	((("requirement inception") AND "mobile development") AND problems)
90	((("requirement inception") AND "mobile platform") AND problems)
91	((("requirement inception") AND "mobile applications") AND problems)
92	((("requirement inception") AND "mobile application development") AND problems)
93	((("requirement inception") AND "mobile development") AND barriers)
94	((("requirement inception") AND "mobile platform") AND barriers)
95	((("requirement inception") AND "mobile applications") AND barriers)
96	((("requirement inception") AND "mobile application development") AND barriers)
97	((("requirement elicitation") AND "mobile development") AND challenges)
98	((("requirement elicitation") AND "mobile platform") AND challenges)
99	((("requirement elicitation") AND "mobile applications") AND challenges)
100	((("requirement elicitation") AND "mobile application development") AND challenges)
101	((("requirement elicitation") AND "mobile development") AND issues)
102	((("requirement elicitation") AND "mobile platform") AND issues)
103	((("requirement elicitation") AND "mobile applications") AND issues)
104	((("requirement elicitation") AND "mobile application development") AND issues)
105	((("requirement elicitation") AND "mobile development") AND problems)
106	((("requirement elicitation") AND "mobile platform") AND problems)
107	((("requirement elicitation") AND "mobile applications") AND problems)

108	((("requirement elicitation") AND "mobile application development") AND problems)
109	((("requirement elicitation") AND "mobile development") AND barriers)
110	((("requirement elicitation") AND "mobile platform") AND barriers)
111	((("requirement elicitation") AND "mobile applications") AND barriers)
112	((("requirement elicitation") AND "mobile application development") AND barriers)
113	((("requirement prioritization") AND "mobile development") AND challenges)
114	((("requirement prioritization") AND "mobile platform") AND challenges)
115	((("requirement prioritization") AND "mobile applications") AND challenges)
116	((("requirement prioritization") AND "mobile application development") AND challenges)
117	((("requirement prioritization") AND "mobile development") AND issues)
118	((("requirement prioritization") AND "mobile platform") AND issues)
119	((("requirement prioritization") AND "mobile applications") AND issues)
120	((("requirement prioritization") AND "mobile application development") AND issues)
121	((("requirement prioritization") AND "mobile development") AND problems)
122	((("requirement prioritization") AND "mobile platform") AND problems)
123	((("requirement prioritization") AND "mobile applications") AND problems)
124	((("requirement prioritization") AND "mobile application development") AND problems)
125	((("requirement prioritization") AND "mobile development") AND barriers)
126	((("requirement prioritization") AND "mobile platform") AND barriers)
127	((("requirement prioritization") AND "mobile applications") AND barriers)
128	((("requirement prioritization") AND "mobile application development") AND barriers)
129	((("requirement validation") AND "mobile development") AND challenges)
130	((("requirement validation") AND "mobile platform") AND challenges)

131	((("requirement validation") AND "mobile applications") AND challenges)
132	((("requirement validation") AND "mobile application development") AND challenges)
133	((("requirement validation") AND "mobile development") AND issues)
134	((("requirement validation") AND "mobile platform”) AND issues)
135	((("requirement validation") AND "mobile applications") AND issues)
136	((("requirement validation") AND "mobile application development") AND issues)
137	((("requirement validation") AND "mobile development") AND problems)
138	((("requirement validation") AND "mobile platform") AND problems)
140	((("requirement validation") AND "mobile applications") AND problems)
141	((("requirement validation") AND "mobile application development") AND problems)
142	((("requirement validation") AND "mobile development") AND barriers)
143	((("requirement validation") AND "mobile platform") AND barriers)
144	((("requirement validation") AND "mobile applications") AND barriers)
145	((("requirement validation") AND "mobile application development") AND barriers)
146	((("requirement specification") AND "mobile development") AND challenges)
147	((("requirement specification") AND "mobile platform") AND challenges)
148	((("requirement specification") AND "mobile applications") AND challenges)
149	((("requirement specification") AND "mobile application development") AND challenges)
150	((("requirement specification") AND "mobile development") AND issues)
151	((("requirement specification") AND "mobile platform") AND issues)
152	((("requirement specification") AND "mobile applications") AND issues)
153	((("requirement specification") AND "mobile application development") AND issues)
154	((("requirement specification") AND "mobile development") AND problems)

155	((("requirement specification") AND "mobile platform") AND problems)
156	((("requirement specification") AND "mobile applications") AND problems)
157	((("requirement specification") AND "mobile application development") AND problems)
158	((("requirement specification") AND "mobile development") AND barriers)
159	((("requirement specification") AND "mobile platform") AND barriers)
160	((("requirement specification") AND "mobile applications") AND barriers)
161	((("requirement specification") AND "mobile application development") AND barriers)
162	(((requirements) AND "mobile development") AND challenges)
163	(((requirements) AND "mobile platform") AND challenges)
164	(((requirements) AND "mobile applications") AND challenges)
165	(((requirements) AND "mobile application development") AND challenges)
166	(((requirements) AND "mobile development") AND issues)
167	(((requirements) AND "mobile platform") AND issues)
168	(((requirements) AND "mobile applications") AND issues)
169	(((requirements) AND "mobile application development") AND issues)
170	(((requirements) AND "mobile development") AND problems)
171	(((requirements) AND "mobile platform") AND problems)
172	(((requirements) AND "mobile applications") AND problems)
173	(((requirements) AND "mobile application development") AND problems)
174	(((requirements) AND "mobile development") AND barriers)
175	(((requirements) AND "mobile platform") AND barriers)
176	(((requirements) AND "mobile application") AND barriers)

APPENDIX B

Table B: Showing the Distribution of Quality Assessment Form among various Candidates for Quality Evaluation

Candidates & Database	Title	Type of Paper & Year	QA average value	Status
C1 & IEEE Xplore	A Framework for Anonymous Routing in Delay Tolerant Networks	Conference paper (25th), 2017	0.78	Included
	A Framework for Cooperative Resource Management in Mobile Cloud Computing	Research paper (Journal), 2013	0.60	Included
	A Model-driven Approach to Generate Mobile Applications for Multiple Platforms	Conference paper, (21st), 2014	0.74	Included
	A Systematic Study on Software Requirements Elicitation Techniques and Its Challenges in Mobile Application Development	Research paper (Journal), 2018	0.88	Included
	Energy Efficient and Delay Aware Service Selection in Mobile Edge Computing	Conference paper, (20th), 2020	0.82	Included
	MAS: Mobile-Apps Assessment and Analysis System	Conference paper, (47th), 2017	0.94	Included
	Multidevice Collaborative Power Management Through Decentralized Knowledge Sharing	Research paper (Journal), 2020	0.86	Included

	Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing	Research paper (Journal), 2019	0.84	Included
	Security-Aware Resource Allocation for Mobile Cloud Computing Systems	Conference paper, (25th), 2015	0.84	Included
	Study and Refactoring of Android Asynchronous Programming	Conference paper, (30th), 2015	0.84	Included
C2 & ACM Digital Library	A cloud based software testing paradigm for mobile applications	Research paper (Journal), 2011	0.88	Included
	A sealant for inter-app security holes in android	Conference paper (39th), 2017	0.90	Included
	Accurate prediction of available battery time for mobile applications	Research paper (Journal), 2016	0.78	Included
	An online algorithm for task offloading in heterogeneous mobile clouds	Research paper (Journal), 2018	0.72	Included
	Cordovaconfig: A tool for mobile hybrid apps' configuration	Conference paper (17th), 2017	0.70	Included
C3 & ACM Digital Library	CrashScope: A practical tool for automated testing of android applications	Conference paper (39th), 2017	0.78	Included

	EMaaS: Energy measurements as a service for mobile applications	Conference paper (41th), 2019	0.60	Included
	Empowering developers to estimate app energy consumption	Conference paper (18th), 2012	0.98	Included
	GUILeak: tracing privacy policy claims on user input data for Android applications	Conference paper (40th), 2018	0.88	Included
	LeakDoctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications	Research paper (Journal), 2019	0.96	Included
C4 & ACM Digital Library	PatchDroid: Scalable third-party security patches for android devices	Conference paper (29th), 2013	0.80	Included
	Taming android fragmentation: Characterizing and detecting compatibility issues for android apps	Conference paper (31th), 2016	0.88	Included
	Tracking the software quality of android applications along their evolution	Conference paper (30th), 2015	0.94	Included
	Understanding and detecting evolution-induced compatibility issues in android apps	Conference paper (33th), 2018	0.88	Included
	Why data deletion fails? A study on deletion flaws and data remanence in android systems	Research paper (Journal), 2019	0.86	Included

C5 & Science Direct	A Context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment	Accepted Manuscript (Journal), 2019	0.90	Included
	A Fast Hybrid Multi-site Computation Offloading for Mobile Cloud Computing	Accepted Manuscript (Journal), 2017	0.92	Included
	A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things	Accepted Manuscript (Journal), 2015	0.88	Included
	A novel pre-cache schema for a high-performance Android system	Accepted Manuscript (Journal), 2016	0.74	Included
	A self-protecting agent's based model for high-performance mobile-cloud computing	Accepted Manuscript (Journal), 2018	0.84	Included
	A smartphone perspective on computation offloading—A survey	Review paper (Journal), 2020	0.72	Included
	A standard for developing secure mobile applications	Research paper (Journal), 2014	0.56	Included
C6 & Science Direct	A systematic mapping study of mobile application testing techniques	Accepted Manuscript (Journal), 2016	0.86	Included

	A web-based, offline-able, and personalized runtime environment for executing applications on mobile devices	Research paper (Journal), 2012	0.88	Included
	An empirical analysis of energy consumption of cross-platform frameworks for mobile development	Accepted Manuscript (Journal), 2017	0.90	Included
	An energy-efficient algorithm for multi-site application partitioning in MCC	Accepted Manuscript (Journal), 2018	0.88	Included
	Automatic offloading of mobile applications into the cloud by means of genetic programming	Research paper (Journal), 2014	0.86	Included
C7 & Wiley Online Library	A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing	Research paper (Journal), 2017	0.90	Included
	A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective	Research paper (Journal), 2020	0.82	Included
	AGILE: A terminal energy-efficient scheduling method in mobile cloud computing	Research paper (Journal), 2015	0.90	Included
	An automated model-based approach for unit-level performance test generation of mobile applications	Research paper (Journal), 2020	0.88	Included

	An evaluation framework for cross-platform mobile application development tools	Research paper (Journal), 2014	0.92	Included
C8 & Wiley Online Library	Joint computation offloading and resource provisioning for an edge-cloud computing environment: A machine learning-based approach	Research paper (Journal), 2020	0.88	Included
	VAnDroid: A framework for vulnerability analysis of Android applications using a model-driven reverse engineering technique	Research paper (Journal), 2019	0.84	Included
	Why does the orientation change mess up my Android application? From GUI failures to code faults	Research paper (Journal), 2018	0.82	Included

APPENDIX C

Table C: Data Extraction Forms obtained during SLR conduction

Entities	Relevant Information
ID:	X1
Publisher:	IEEE
Title of Article:	A framework for anonymous routing in delay tolerant networks
Type of Article:	Conference paper
Year:	2017
Published in:	2017 IEEE 25th International Conference on Network Protocols (ICNP)
Methodology:	Literature review, Simulation
Contribution:	Proposed a framework for Anonymous Routing
Domain:	Networks
Quality Assessment Score:	0.78
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Anonymous communication

DTNs: Delay Tolerant Networks

Entities	Relevant Information
ID:	X2
Publisher:	IEEE
Title of Article:	A Framework for Cooperative Resource Management in Mobile Cloud Computing
Type of Article:	Research paper
Year:	2013
Published in:	IEEE Journal on Selected Areas in Communications
Methodology:	Simulation

Contribution:	A decision-making framework for Resource allocation to mobile applications, revenue management, and co-operation among service providers
Domain:	Mobile Cloud Computing
Quality Assessment Score:	0.60
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Resource sharing problem

Entities	**Relevant Information**
ID:	X3
Publisher:	IEEE
Title of Article:	A Model-Driven Approach to Generate Mobile Applications for Multiple Platforms
Type of Article:	Conference paper
Year:	2014
Published in:	2014 21st Asia-Pacific Software Engineering Conference
Methodology:	Literature review and Case study
Contribution:	A model-driven approach for automated generation of mobile apps for multiple platforms
Domain:	Software Engineering
Quality Assessment Score:	0.74
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Multiple platform support

Entities	**Relevant Information**
ID:	X4
Publisher:	IEEE

Title of Article:	A Systematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development
Type of Article:	Research article
Year:	2018
Published in:	IEEE Access
Methodology:	Systematic Literature Review
Contribution:	Challenges on Elicitation techniques for mobile development
Domain:	Software Requirement Engineering
Quality Assessment Score:	0.88
Status of Exclusion/ Inclusion:	Included
Answer to RQ1:	Challenges in Requirement elicitation techniques, approaches & tools

Entities	**Relevant Information**
ID:	X5
Publisher:	IEEE
Title of Article:	Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing
Type of Article:	Research article
Year:	2019
Published in:	IEEE Access
Methodology:	Simulation, Experimentation
Contribution:	An efficient radio and computing resource allocation scheme is proposed
Domain:	Mobile Edge Computing

Quality Assessment Score:	0.84
Status of Exclusion/ Inclusion:	Included
Answer to RQ1:	Optimization problem

Entities	**Relevant Information**
ID:	X6
Publisher:	IEEE
Title of Article:	Security-Aware Resource Allocation for Mobile Cloud Computing Systems
Type of Article:	Conference paper
Year:	2015
Published in:	2015 24th International Conference on Computer Communication and Networks (ICCCN)
Methodology:	Simulation
Contribution:	Semi Markov Decision process-based resource allocation (SMDP-RAS) algorithm for secure MCC systems is proposed
Domain:	Mobile Cloud Computing
Quality Assessment Score:	0.84
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Resource allocation problem

Entities	**Relevant Information**
ID:	X7
Publisher:	IEEE
Title of Article:	Study and Refactoring of Android Asynchronous Programming (T)

Type of Article:	Conference paper
Year:	2015
Published in:	2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)
Methodology:	Literature review, Survey, Simulation, Experimentation
Contribution:	A refactoring tool named as AsyncDroid is proposed
Domain:	Computer Science
Quality Assessment Score:	0.84
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Asynchrony retrofitting

2. Science Direct

Entities	**Relevant Information**
ID:	S1
Publisher:	Elsevier
Title of Article:	A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment
Type of Article:	Accepted manuscript
Year:	2019
Published in:	Future Generation Computer Systems
Methodology:	Simulation, Experimentation
Contribution:	Proposed context-sensitive offloading system (CSOS) that takes the advantage of ML reasoning techniques and robust profiling system to provide offloading decisions with a high level of accuracy
Domain:	Mobile cloud computing
Quality Assessment Score:	0.90

Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Ignorance of contextual information

Entities	**Relevant Information**
ID:	S2
Publisher:	Elsevier
Title of Article:	A fast hybrid multi-site computation offloading for mobile cloud computing
Type of Article:	Accepted manuscript
Year:	2017
Published in:	Journal of Network and Computer Applications
Methodology:	Simulation, Testbed Experiments
Contribution:	Proposed a fast hybrid multisite computation offloading solution that finds an offloading solution promptly
Domain:	Mobile cloud computing
Quality Assessment Score:	0.92
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Partitioning problem

Entities	**Relevant Information**
ID:	S3
Publisher:	Elsevier
Title of Article:	A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things
Type of Article:	Accepted manuscript
Year:	2016

Published in:	Future Generation Computer Systems
Methodology:	Simulation, Experiment
Contribution:	The paper presents an ecosystem for mobile crowd sensing which relies on the CloUd- based PUblish/Subscribe middleware (CUPUS) to acquire sensor data from mobile devices in a flexible and energy-efficient manner and to perform near real-time processing of Big Data streams.
Domain:	Mobile Internet of Things
Quality Assessment Score:	0.88
Status of Exclusion/ Inclusion:	Included
Answer to RQ1:	Managing mobile sensor data

Entities	**Relevant Information**
ID:	S4
Publisher:	Elsevier
Title of Article:	A novel pre-cache schema for high performance Android system
Type of Article:	Accepted manuscript
Year:	2016
Published in:	Future Generation Computer Systems
Methodology:	Simulation, Experiment
Contribution:	PATAS is a novel approach proposed for high-performance Android systems by using pre-cache technologies
Domain:	Computer science
Quality Assessment Score:	0.74
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Inefficient execution model problem

Entities	Relevant Information
ID:	S5
Publisher:	Elsevier
Title of Article:	A self-protecting agent's based model for high-performance mobile-cloud computing
Type of Article:	Accepted manuscript
Year:	2018
Published in:	Computers and Security
Methodology:	Simulation, Experiment
Contribution:	Presented a context-dependent CO model for MCC based on application segments packed into autonomous agents
Domain:	Mobile cloud computing
Quality Assessment Score:	0.84
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Tampering problem

Entities	Relevant Information
ID:	S6
Publisher:	Elsevier
Title of Article:	A smartphone perspective on computation offloading—A survey
Type of Article:	Review paper
Year:	2020
Published in:	Computer Communications
Methodology:	Literature review
Contribution:	Guideline on CO in heterogeneous CC
Domain:	Mobile edge computing
Quality Assessment Score:	0.72
Status of Exclusion/Inclusion:	Included

Answer to RQ1:	Construction of Power energy models, Problem with profilers

Entities	**Relevant Information**
ID:	S7
Publisher:	Elsevier
Title of Article:	A standard for developing secure mobile applications
Type of Article:	Research article
Year:	2014
Published in:	Computer Standards and Interfaces
Methodology:	Literature review
Contribution:	Standard practice for developing secure mobile application
Domain:	Cyber Security
Quality Assessment Score:	0.56
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Lack of development standards

Entities	**Relevant Information**
ID:	S8
Publisher:	Elsevier
Title of Article:	A systematic mapping study of mobile application testing techniques
Type of Article:	Accepted manuscript
Year:	2016
Published in:	Journal of Systems and Software
Methodology:	Systematic mapping study
Contribution:	Specific testing issues for practitioners are identified
Domain:	Software testing

Quality Assessment Score:	0.86
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Key testing issues for practitioners

Entities	Relevant Information
ID:	S9
Publisher:	Elsevier
Title of Article:	A web-based, offline-able, and personalized runtime environment for executing applications on mobile devices
Type of Article:	Research article
Year:	2012
Published in:	Computer Standards and Interfaces
Methodology:	Simulation
Contribution:	A web based platform is proposed providing various services such as offline mode/ services, content adaptation service & synchronization services
Domain:	Computer science
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Cross-platform compatibility, Offline execution of web apps

Entities	Relevant Information
ID:	S10
Publisher:	Elsevier
Title of Article:	An empirical analysis of energy consumption of cross-platform frameworks for mobile development
Type of Article:	Accepted manuscript
Year:	2017
Published in:	Pervasive and Mobile Computing
Methodology:	Literature review, Empirical investigation, Experiment

Contribution:	Evaluate and compare the current cross-platform framework for mobile applications based on energy consumption
Domain:	Mobile computing
Quality Assessment Score:	0.90
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Fragmentation problems for developers

Entities	**Relevant Information**
ID:	S11
Publisher:	Elsevier
Title of Article:	An energy-efficient algorithm for multi-site application partitioning in MCC
Type of Article:	Accepted manuscript
Year:	2018
Published in:	Sustainable Computing: Informatics and Systems
Methodology:	Simulation, Experiment
Contribution:	Proposed a multi-site application partitioning algorithm named Cyclic random movement (CRM) based on genetic algorithm (CRMGA)
Domain:	Mobile cloud computing
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Multi-site partitioning

Entities	**Relevant Information**
ID:	S12
Publisher:	Elsevier

Title of Article:	Automatic offloading of mobile applications into the cloud by means of genetic programming
Type of Article:	Research article
Year:	2014
Published in:	Applied Soft Computing Journal
Methodology:	Simulation, Experiment
Contribution:	A framework is presented for generating models to make automatic decisions on the offloading of mobile applications using genetic programming (GP)
Domain:	Cloud computing. Mobile computing, Data mining
Quality Assessment Score:	0.86
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Offloading of computation tasks

3- Wiley Online Library

Entities	**Relevant Information**
ID:	W1
Publisher:	John Wiley & Sons
Title of Article:	A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing
Type of Article:	Research article
Year:	2017
Published in:	International Journal of Communication Systems
Methodology:	Experiment, Simulation
Contribution:	This paper presented an optimized genetic-based decision algorithm for multi-site CO problem called GAMCO, to find the best possible solution promptly
Domain:	Mobile cloud computing
Quality Assessment Score:	0.90

Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Offloading decision in multi-site context

CO: Computation offloading

GAMCO: Genetic Algorithm based Multi-site Computation Offloading

Entities	Relevant Information
ID:	W2
Publisher:	John Wiley & Sons
Title of Article:	A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective
Type of Article:	Research article
Year:	2020
Published in:	Software - Practice and Experience
Methodology:	Systematic literature review, Survey
Contribution:	The GT-based CO approaches in the MEC environment are presented in the form of classical taxonomy with some open issues
Domain:	Mobile edge computing
Quality Assessment Score:	0.82
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	CO problem

GT: Game Theory

CO: Computation Offloading

MEC: Mobile Edge Computing

Entities	Relevant Information
ID:	W3
Publisher:	John Wiley & Sons

Title of Article:	AGILE: A terminal energy-efficient scheduling method in a mobile cloud computing
Type of Article:	Research article
Year:	2014
Published in:	Transactions on emerging telecommunications technologies
Methodology:	Simulation, Experiment
Contribution:	A novel terminal energy-efficient scheduling method (AGILE for short) is presented to make decisions about mobile applications' tasks executed by mobile devices or cloud servers
Domain:	Mobile cloud computing
Quality Assessment Score:	0.90
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Tasks scheduling

Entities	**Relevant Information**
ID:	W4
Publisher:	John Wiley & Sons
Title of Article:	An automated model-based approach for unit-level performance test generation of mobile applications
Type of Article:	Research article
Year:	2020
Published in:	Journal of Software: Evolution and Process
Methodology:	Simulation, Case study, Evaluation
Contribution:	A model-based approach for mobile platforms to overcome the the problem of automation of performance testing
Domain:	Software testing
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included

Answer to RQ1:	Variation in performance

Entities	Relevant Information
ID:	W5
Publisher:	John Wiley & Sons
Title of Article:	An evaluation framework for cross-platform mobile application development tools
Type of Article:	Research article
Year:	2015
Published in:	Software - Practice and Experience
Methodology:	Simulation, Experiment
Contribution:	A high-level extensible framework is presented for the evaluation of any CPDT
Domain:	Computer science
Quality Assessment Score:	0.92
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Fragmentation

CPDT: Cross-Platform Development Tool

Entities	Relevant Information
ID:	W6
Publisher:	John Wiley & Sons
Title of Article:	Joint computation offloading and resource provisioning for an edge- cloud computing environment: A machine learning-based approach
Type of Article:	Research article
Year:	2020
Published in:	Software - Practice and Experience
Methodology:	Simulation, Experiment

Contribution:	An edge server provisioning approach using LSTM prediction to estimate the future workload & RL technique to make the appropriate scaling decision
Domain:	Edge cloud computing
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Handling of dynamic workloads

LSTM: Long Short Term Model

RL: Reinforcement Learning

Entities	**Relevant Information**
ID:	W7
Publisher:	John Wiley & Sons
Title of Article:	VAnDroid: A framework for vulnerability analysis of Android applications using a model-driven reverse engineering technique
Type of Article:	Research article
Year:	2019
Published in:	Software - Practice and Experience
Methodology:	Simulation, Experiment
Contribution:	A framework named VAnDroid based on MDRE is presented, identifying the security risks & vulnerabilities related to android application communication model
Domain:	Computer science
Quality Assessment Score:	0.84
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Android application communication problem

VAnDroid: Vulnerability Analysis of Android Application

MDRE: Model-Driven Reverse Engineering

Entities	Relevant Information
ID:	W8
Publisher:	John Wiley & Sons
Title of Article:	Why does the orientation change mess up my Android application? From GUI failures to code faults
Type of Article:	Research article
Year:	2018
Published in:	Software Testing Verification and Reliability
Methodology:	Simulation, Experiment
Contribution:	A framework named DOC is proposed for classifying GUI failures with the investigation of its key features along with possible faults causing them
Domain:	Software testing
Quality Assessment Score:	0.82
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	GUI failures or Screen orientation changes the problem

DOC: Double Orientation Change

GUI: Graphical User Interface

4- ACM Digital Library

Entities	Relevant Information
ID:	A1
Publisher:	Association for Computing Machinery
Title of Article:	A cloud based software testing paradigm for mobile applications
Type of Article:	Research article
Year:	2011

Published in:	ACM SIGSOFT Software Engineering Notes
Methodology:	Simulation
Contribution:	An approach for testing the software running on mobile terminals by using CC environment
Domain:	Software Engineering, Software Testing
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Emulator related constraints

CC: Cloud computing

Entities	**Relevant Information**
ID:	A2
Publisher:	IEEE press
Title of Article:	A SEALANT for inter-app security holes in android
Type of Article:	Conference paper
Year:	2017
Published in:	Proceedings of the 39th International Conference on Software Engineering
Methodology:	Simulation, Case study
Contribution:	A technique enabling android users to protect their devices from multiple ICC vulnerabilities named SEALANT
Domain:	Software Engineering
Quality Assessment Score:	0.90
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Android communication model problem

SEALENT: **S**ecurity of **E**nd-users of **A**ndroid via **L**ight weight **AN**alysis **T**echnique

Entities	Relevant Information
ID:	A3
Publisher:	Association for Computing Machinery
Title of Article:	Accurate prediction of available battery time for mobile applications
Type of Article:	Research article
Year:	2016
Published in:	ACM Transactions on Embedded Computing Systems
Methodology:	Experiment
Contribution:	A framework is proposed that accurately estimates the remaining battery time of applications at runtime
Domain:	Computer science
Quality Assessment Score:	0.78
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Estimation of battery time

Entities	Relevant Information
ID:	A4
Publisher:	Association for Computing Machinery
Title of Article:	An online algorithm for task offloading in heterogeneous mobile clouds
Type of Article:	Research paper
Year:	2018
Published in:	ACM Transactions on Internet Technology
Methodology:	Simulation, Experiment
Contribution:	An online real-time scheduling algorithm for MCOSP on the basis of rent/buy problem
Domain:	Human-Centered Computing, Computer science
Quality Assessment Score:	0.72

Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Optimization problem

MCOSP: Mobile Code Offloading Scheduling Problem

Entities	Relevant Information
ID:	A5
Publisher:	Association for Computing Machinery
Title of Article:	Automating UI tests for mobile applications with a formal gesture Descriptions
Type of Article:	Conference paper
Year:	2014
Published in:	Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services
Methodology:	Simulation, Experiment
Contribution:	An extension to the popular Calabash testing framework is proposed allowing for test automation for gesture-based mobile application
Domain:	Software Centered computing, Human-computer interaction
Quality Assessment Score:	0.62
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Gesture-based interaction constraint

RCA: Response Cache Approach

Entities	Relevant Information
ID:	A6
Publisher:	Association for Computing Machinery
Title of Article:	Cordovaconfig: A tool for mobile hybrid apps' configuration
Type of Article:	Conference paper
Year:	2018

Published in:	Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia
Methodology:	Simulation, Case study
Contribution:	Designed and built CORDOVACONFIG, a tool for configuring mobile hybrid apps.
Domain:	Security & Privacy
Quality Assessment Score:	0.70
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Configuration problems

Entities	**Relevant Information**
ID:	A7
Publisher:	IEEE Press
Title of Article:	CrashScope: A practical tool for automated testing of android Applications
Type of Article:	Conference paper
Year:	2017
Published in:	Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017
Methodology:	Simulation
Contribution:	Designed & implemented a practical tool named CRASHSCOPE, that automatically discovers, reports & reproduces crashes for Android applications
Domain:	Software engineering, Software testing
Quality Assessment Score:	0.78
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Implementation of contextual features

Entities	Relevant Information
ID:	A8
Publisher:	IEEE Press
Title of Article:	EMaaS: Energy measurements as a service for mobile applications
Type of Article:	Conference paper
Year:	2019
Published in:	Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2019
Methodology:	Simulation
Contribution:	A system providing reliable energy measurement for mobile applications without requiring a complex setup named EMaaS
Domain:	Software engineering
Quality Assessment Score:	0.60
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Measurement of energy consumption

EMaaS: Energy Measurement as a Service

Entities	Relevant Information
ID:	A9
Publisher:	Association for Computing Machinery
Title of Article:	Empowering developers to estimate app energy consumption
Type of Article:	Conference paper
Year:	2012
Published in:	Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM
Methodology:	Simulation, Experiment
Contribution:	An energy emulation tool named WattsOn estimates the energy consumption of app during development
Domain:	Computer science

Quality Assessment Score:	0.98
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Energy measurement

Entities	**Relevant Information**
ID:	A10
Publisher:	Association for Computing Machinery
Title of Article:	GUILeak: tracing privacy policy claims on user input data for Android applications
Type of Article:	Conference paper
Year:	2018
Published in:	Proceedings of the 40th International Conference on Software Engineering
Methodology:	Simulation, Experiment
Contribution:	Proposed a novel approach to protect privacy policy violations due to leak of user input data
Domain:	Software Engineering
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Disclosing of information

Entities	**Relevant Information**
ID:	A11
Publisher:	Association for Computing Machinery
Title of Article:	LeakDoctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications
Type of Article:	Research article
Year:	2019

Published in:	Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies
Methodology:	Simulation, Experiment
Contribution:	A novel fine-grained approach named LeakDoctor is proposed which aims to detect each privacy disclosure automatically
Domain:	Security & Privacy
Quality Assessment Score:	0.96
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Privacy disclosure

Entities	**Relevant Information**
ID:	A12
Publisher:	Association for Computing
Title of Article:	PatchDroid: Scalable third-party security patches for android Devices
Type of Article:	Conference paper
Year:	2013
Published in:	Proceedings of the 29th Annual Computer Security Applications Conference
Methodology:	Simulation, Experiment
Contribution:	A system is proposed to distribute and apply third-party security patches for android
Domain:	Security & Privacy, Applied computing
Quality Assessment Score:	0.80
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Patching problem

Entities	**Relevant Information**

ID:	A13
Publisher:	Association for Computing Machinery
Title of Article:	Taming android fragmentation: Characterizing and detecting compatibility issues for android apps
Type of Article:	Research article
Year:	2016
Published in:	Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering
Methodology:	Empirical study, Simulation, Experiment
Contribution:	A technique named FicFinder is presented to automatically detect compatibility issues in Android applications
Domain:	Software engineering, Human-Centered Computing
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	FIC issues

FicFinder: Fragmentation-induced compatibility issues Finder

Entities	**Relevant Information**
ID:	A14
Publisher:	IEEE Press
Title of Article:	Tracking the software quality of Android applications along their evolution
Type of Article:	Conference paper
Year:	2015
Published in:	Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering

Methodology:	Simulation, Experiment, Case study
Contribution:	A tooled approach named PAPRIKA is presented to assess the software quality of Mobile applications considering anti-patterns along with their detection
Domain:	Software Quality
Quality Assessment Score:	0.94
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Poor design choices problem

Entities	**Relevant Information**
ID:	A15
Publisher:	Association for Computing Machinery
Title of Article:	Understanding and detecting evolution-induced compatibility issues in Android apps
Type of Article:	Conference paper
Year:	2018
Published in:	Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering
Methodology:	An empirical investigation, Simulation
Contribution:	A new tool named IctApiFinder is developed to detect the incompatible API usages in android applications
Domain:	Software engineering
Quality Assessment Score:	0.88
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Evolution induced compatibility problem

Entities	**Relevant Information**
ID:	A16

Publisher:	Association for Computing Machinery
Title of Article:	Why Data Deletion Fails? A Study on Deletion Flaws and Data Remanence in Android Systems
Type of Article:	Research article
Year:	2017
Published in:	ACM Transactions on Embedded Computing Systems
Methodology:	Literature review, Experiment
Contribution:	1- Presented some flaws considering 3 typical scenarios of the Android system 2- Design & implement a framework named DataRaider to recover files from disc fragments 3- Also, some mitigation plans were proposed for data remanence issues along with some suggestions on data protection in android systems
Domain:	Security & Privacy
Quality Assessment Score:	0.86
Status of Exclusion/Inclusion:	Included
Answer to RQ1:	Problem of data-erasure

APPENDIX D:

Table D: Showing the Implementation of Data Encoding Technique

Paper ID	Paper Statement	Respective Code	Data Encoding
X1	"Security and privacy issues are considered to be two of the most significant concerns to organizations and individuals using mobile applications. In this paper, we seek to address anonymous communications in delay tolerant networks (DTNs)"	X1L1, X1L2	Security and Privacy in DTNs **OR** Anonymous Communication
X2	"In this paper, we consider the resource (i.e., radio and computing resources) sharing problem to support mobile applications in a mobile cloud computing environment"	X2L2	Resource Sharing problem
X3	"In addition, mobile applications also have to support multiple platforms, as an application written for one platform (e.g., Android) cannot run on another platform (e.g., Windows Phone)."	X3L4	Multiple platform support
X4	"This also refers to the emergence of challenges in Requirements Elicitation techniques, approaches, and tools while performing them. Particularly, in the area of Requirements Engineering for software development, several techniques and approaches have been observed in literature but for mobile application development, which is different from traditional software development, has not been discussed much in past studies. Short development cycle, device limitations, and less development time for mobile application development are some of the issues to which there is no 'silver bullet' available"	X4L2, X4L3, X4L4	Challenges in Requirement Elicitation Techniques

X5	"to minimize the total processing completion time of all tasks "	X5L5Ph2 Ph= phrase	Minimization of TPC time of all tasks **OR** Optimization problem
X6	"The mobile request for using cloud resource is classified according to its security requirement and the amount of required resource for remote computing"	X6L2	Security requirement in terms of resource allocation **OR** Request for using cloud resource **OR** Resource allocation problem (from security perspective)
X7	"However, developers can still use the inappropriate async constructs, which result in memory leaks, lost results, and wasted energy. Fortunately, refactoring tools can eliminate these problems by transforming async code to use the appropriate constructs."	X7Pa1L2, X7Pa1L4 Pa = Paragraph	Memory leaks & energy wastage **OR** Inappropriate async constructs **OR** Transformation of async code **OR** Use of appropriate constructs **OR** Asynchrony retrofitting
S1	"There still remain challenges regarding the dynamic nature of the MCC environment. Most solutions design a single reasoner for the offloading decision and do not know how accurate and precise this technique is, so that when applied in real-world environments it can contribute to inaccurate	S1L3Ph2, S1L4	Offloading decision **OR** Inaccurate decisions **OR** Dynamic nature of MCC environment

	decisions and consequently the low performance of the overall system."		**OR** CO decision
S2	"Moreover, in the real world, different types of clouds/servers with heterogeneous processing speeds and access delays are available for offloading resulting in the time-consuming process of deciding for offloading"	S2L5	Heterogeneous processing speed & Access delays of clouds **OR** Multi-site CO **OR** Partitioning problem **OR** Optimal partitioning
S3	"The inherent device mobility and high sensing frequency can produce dense and rich spatiotemporal information about our environment, but also creates new challenges due to device dynamicity and energy constraints, as well as large volumes of generated raw sensor data which needs to be processed and analyzed to extract useful information for end-users."	S3L2	Processing & Analysis of raw sensor data **OR** Extraction of useful information **OR** Management of mobile sensor resources **OR** Managing mobile sensor data
S4	"However, current Android application model is not efficient by using current two common approaches, including Activity+XML Layout Files (AXLF) and HTML+WebKit (HWK) models."	S4L2	Complexity in achieving efficiency for the android application model **OR** Inefficient execution model problem
S5	"Moreover, the security risks arising from offloading data and code to an untrusted platform and the computational overhead introduced by complex security mechanisms stand as deterrents for adoption of MCC at large"	S5L3	The problem in the adoption of MCC **OR** Tampering

S6	"Constructing power/energy models is one of the most challenging phases in an offloading system." "As mentioned above, the application behavior is non-deterministic at runtime. This brings difficulties to estimate the execution data of offloading candidates, such as running time, available memory, and communication cost."	S6Sec3Sub4Pa1L1, S6Sec4Sub4Pa1L2, S6Sec4Sub4Pa1L3 Sec= Section Sub= Sub-section	Construction of power/ energy model **AND** Unpredictable app behaviour **OR** Complexity in estimating execution data
S7	"The abundance of mobile software applications (apps) has created a security challenge. The lack of development standards and best practices expose the mobile device to potential attacks."	S7L1, S7L3	Security challenge in mobile apps **OR** Lack of development standards/ practices
S8	"As far as we are aware, there are currently no available comprehensive systematic review studies in the area of a mobile and smart-phone application testing."	S8Sec1P3L1	Smart-phone application testing
S9	"When creating mobile applications, developers usually encounter the cross-platform incompatibility problem (for example, iPhone applications cannot be executed on the Android platform)."	S9L3	Cross-platform incompatibility
S10	"The increasing fragmentation of mobile devices market has created the problem of supporting all the possible mobile platforms to reach the highest number of potential users."	S10L1	Supporting mobile platforms **OR** Fragmentation of mobile devices
S11	"Prolonging battery life for a mobile device has been an urgent issue in mobile computing. Computation offloading is an effective approach to reduce energy consumption on a mobile device. However, most previous work focuses on offloading computation	S11L1, S11L2, S11L3	Prolonging battery life **OR** Reduction of energy consumption **OR**

	from mobile to a single server, which cannot adapt to mobile cloud computing (MCC) where data and computation are commonly distributed to multiple sites."		Complexity in the adoption of MCC **OR** Multisite partitioning
S12	"The limited battery life of modern mobile devices is one of the key problems limiting their use. Even if the offloading of computation onto cloud computing platforms can considerably extend battery duration, it is really hard not only to evaluate the cases where offloading guarantees real advantages based on the requirements of the application in terms of data transfer, the computing power needed, etc. but also to evaluate whether user requirements (i.e. the costs of using the cloud services, a determined QoS required, etc.) are satisfied"	S12L1, S12L2	Limited battery life **OR** Offloading computation problem **OR** Offloading problem
W1	"In this context, mobile devices can offload the computation-intensive parts of their applications to powerful cloud servers. However, they should decide what computation-intensive parts are appropriate for offloading to be beneficial instead of local execution on the mobile device. Moreover, in the real world, different types of clouds/servers with heterogeneous processing speeds are available that should be considered for offloading."	W1L2, W1L3, W1L4	Offloading of computation-intensive tasks **OR** Multisite offloading problem **OR** Offloading decisions in a multisite context **OR** NP-complete problem
W2	"On the other hand, due to the resource limitations, resource heterogeneity, dynamic nature, and unpredictable behavior of MEC environments, it is necessary to consider the computation offloading issues as the challenging problem in the MEC environment"	W2L4	Resource limitations in the MEC environment **OR** Resources Heterogeneity of MEC **OR**

			Unpredictable behavior of MEC **OR** Computation Offloading problem
W3	"However, little work has been performed about how to schedule mobile application tasks in data centers to extend battery life for mobile terminals"	W3L4	Battery life escalation **OR** Energy-saving of mobile devices **OR** Energy minimization of mobile devices **OR** Task scheduling
W4	"The overall performance of native applications may significantly vary across platforms. The current industrial practice is to manually test the performance for each variant, which is not a scalable or efficient approach."	W4L4, W4L5	Performance variation **OR** Inefficient approach usage **OR** Performance test generation
W5	"The mobile application market is becoming increasingly fragmented with the availability of multiple mobile platforms that differ in development procedures. Developers are forced to choose to support only some platforms and specific devices because of limited development resources."	W5L1, W5L2	Availability of multiple mobile platforms **OR** Fragmentation **OR** Restriction of using specific platform and devices
W6	"Since the submit-ted workloads to the smart mobile applications changes over the time, decision making about offloading and edge server provisioning to handle the dynamic workloads of mobile	W6L3	Resource management **OR** Offloading decision making

	applications are one of the challenging issues into the resource management scope."		**OR** Handling the dynamic workloads
W7	"Android provides applications with a message passing system to communicate within and between them. Due to the risks associated with this system, it is vital to detect its unsafe operations and potential vulnerabilities."	W7L1, W7L2	Unsafe operations and vulnerabilities **OR** Android application communication model problem
W8	"This paper investigates the failures exposed in mobile apps by the mobile-specific event of changing the screen orientation."	W8L1	GUI failures **OR** Impact of mobile events on GUI **OR** Screen orientation changes problem
A1	"But these emulators typically cannot emulate actual network speed and availability, actual device-specific content-rendering speed, memory limitation, cache size, CPU speed, and stack size. These emulators are designed for specific platforms, which lack testing of applications on heterogeneous mobile platforms"	A1L3, A1L4	Problems with emulators **OR** Emulators related constraints **OR** Construction of emulators for specific platforms
A2	"Android's communication model has a major security weakness: malicious apps can manipulate other apps into performing unintended operations and can steal end-user data while appearing ordinary and harmless."	A2L1	Security issues w.r.t. android communication model **OR** Android communication model problem
A3	"Energy consumption in mobile devices is an important issue for both system developers and users. Users are aware of the battery-related	A3L1, A3L2	Energy consumption of mobile devices **OR** Estimation of battery time

	information of their mobile devices and tend to take appropriate actions to increase the battery life."		**OR** Extension of battery life
A4	"The computation offloading decision making and tasks scheduling among heterogeneous shared resources in mobile clouds are becoming challenging problems in terms of providing global optimal task response time and energy efficiency."	A4L2	CO decision making **OR** Task scheduling in mobile clouds **OR** Optimization problems
A5	"However, testing mobile applications is still cumbersome, time-consuming, and error-prone. One reason is the devices' focus on touch-based interaction – gestures cannot be easily incorporated into automated application tests"	A5L4, A5L5	Restrictions in touch and gesture-based interactions **OR** Problems with touch-based interaction-gestures **OR** Gesture-based interaction constraints
A6	"Configuring mobile hybrid apps properly is an important but often neglected activity. Coarse-grained configurations and risky default settings result in several privacy and security breaches. Moreover, middleware libraries provide a basic interface to the developers which may drive them off from changing the default settings"	A6L5, A6L6	A problem in configuring mobile hybrid apps **OR** Security & Privacy breaches in terms of configuration **OR** Configuration problems
A7	"Unique challenges arise when testing mobile applications due to their prevailing event-driven nature and complex contextual features (e.g. sensors, notifications). Current automated input generation approaches for Android apps are typically not practical for developers to use due to required instrumentation or platform dependence and	A7L1, A7L2	Challenges in Mobile app testing **OR** Event-driven nature of mobile applications **OR**

	generally do not effectively exercise contextual features"		Applicability / implementation of contextual features **OR** A complexity of contextual features
A8	"Measuring energy consumption is a challenging the task faced by developers when building mobile apps."	A8L1	Measurement of energy consumption
A9	"Battery life is a critical performance and user experience metric on mobile devices. However, it is difficult for app developers to measure the energy used by their apps, and to explore how energy use might change with conditions that vary outside of the developer's control such as network congestion, choice of mobile operator, and user settings for screen brightness."	A9L1, A9L2	Impact of various conditions on energy measurement **OR** Energy measurement estimation problem
A10	"This popularity coupled with user data collection by Android apps has made privacy protection a well-known challenge in the Android ecosystem. In practice, app producers provide privacy policies disclosing what information is collected and processed by the app."	A10L2, A10L3	Privacy protection in Android ecosystem **OR** Disclosing of information
A11	"With the enormous popularity of smartphones, millions of mobile apps are developed to provide rich functionalities for users by accessing certain personal data, leading to great privacy concerns. To address this problem, many approaches have been proposed to detect privacy disclosures in mobile apps, but they largely fail to automatically determine whether the privacy disclosures are necessary for the functionality of apps."	A11L1, A11L2	Privacy concerns in terms of accessing personal data **OR** Privacy Disclosures

A12	"Unfortunately, more than 30% of all devices contain publicly known security vulnerabilities and, in practice, cannot be updated through normal mechanisms since they are no longer supported by the manufacturer and mobile operator. This failure of traditional patch distribution systems has resulted in the creation of a large population of vulnerable mobile devices."	A12P1L2, A12P1L3	Security vulnerabilities in terms of patching **OR** Patching **OR** Patch distribution **OR** Vulnerabilities in mobile devices
A13	"Android ecosystem is heavily fragmented. The numerous combinations of different device models and operating system versions make it impossible for Android app developers to exhaustively test their apps. As a result, various compatibility issues arise, causing a poor user experience. However, little is known on the characteristics of such fragmentation induced compatibility issues and no mature tools exist to help developers quickly diagnose and fix these issues."	A13L1, A13L2, A13L3, A13L4	Fragmentation of Android ecosystem **OR** Fragmentation induced compatibility issues **OR** FIC issues impact on UX
A14	"However, addressing these requirements may result in poor design choices, also known as antipatterns, which may incidentally degrade software quality and performance."	A14L2	Effect of design choices on software quality & performance **OR** Tracking of antipatterns **OR** Poor design choices
A15	"The frequent release of Android OS and its various versions bring many compatibility issues to Android Apps. This paper studies and addresses such evolution-induced compatibility problems."	A15L1, A15L2	Compatibility issues **OR** Evolution induced compatibility problems **OR** Limitations in Fragmented Android O.S.

A16	"While they carry valuable information, data erasure is somehow much more vulnerable than was predicted. The security mechanisms provided by the Android system are not flexible enough to thoroughly delete sensitive data. In addition to the weakness among several provided data-erasing and file deleting mechanisms, we also target the Android OS design flaws in data erasure, and unveil that the design of the Android OS contradicts some secure data-erasure demands"	A16L2, A16L3, A16L4	Sensitive data deletion **OR** Problem of data-erasure **OR** Data erasing problem **OR** Inflexible android security mechanisms **OR** Limitations of data erasing and file deleting Mechanisms

APPENDIX E

Table E: Showing the Application of Implicit-Explicit Removal

Serial Number	Paper ID	Constructs	After I/E removal
1	X5, X6, W6	Resource sharing, Resource management, Resource allocation problem, Handling of dynamic workloads, Security requirement in terms of resource allocation, Request for using cloud resource	Resource allocation problem
2	X3, S9, S10, W5	Multiple platform support, Cross-platform incompatibility, Fragmentation of mobile devices, Fragmentation problems for developers, Availability of multiple platforms, Fragmentation, Restriction of using specific platforms & devices	Fragmentation
3	X2, A4	Minimization of TPC time of all tasks, Optimization problem, Task scheduling in mobile clouds	Optimization problem
4	S1, S12, W2	Offloading decision, Inaccurate decisions, Dynamic nature of MCC environment, CO decision, CO problem, Resource limitations in MEC environment, Resource heterogeneity of MEC, Unpredictable behavior of MEC	Offloading decisions problem
5	S2, S11, W1	Partitioning problems, Multi-site partitioning, Multi-site CO, Optimal partitioning, Reduction of energy consumption, Prolonging battery life, Complexity in the adoption of MCC, Offloading of computation-intensive tasks, Multi-site offloading problem, offloading decision in multi-site context, NP-complete problem	Multi-site Partitioning problem
6	W7, A2	Unsafe operations & vulnerabilities in android communication model, Android application communication model, Android communication model problem, Security issues w.r.t. android communication model	Problems related to Android communication model

7	A8, A9	Measurement of energy consumption, Impact of various conditions on Energy measurement, Energy measurement estimation problem	Energy measurement
8	A10, A11	Privacy protection of the android ecosystem, Disclosing of information, Privacy disclosures, Privacy concerns in terms of Personal	Privacy disclosures
9	A13, A15	Fragmentation of Android ecosystem, Fragmentation induced compatibility issues, FIC issues impact on UX, Evolution induced compatibility problems, Compatibility issues, Limitations in fragmented android O.S.	FIC issues

APPENDIX F

Table F: Showing the Other Encoded Challenges

Serial number	Paper ID	Identified Challenges	Final Selected
1	X1	Security & privacy in terms of DTNs or Anonymous communication	Anonymous communication
2	X4	Challenges in Requirement Elicitation techniques (Detail below after table)	
3	X7	Memory leaks & energy wastage or Inappropriate async constructs or Transformation of async code or Use of appropriate constructs or Asynchrony retrofitting	Asynchrony retrofitting
4	S3	Processing & Analysis of raw sensor data or Extraction of useful information in the MCS domain or Management of mobile sensor resources or Managing mobile sensor data	Extraction of useful information in MCS domain
5	S4	Complexity in achieving efficiency for android application model or Inefficient execution model problem	Inefficient execution model problem
6	S5	A problem in adoption of MCC or Tampering or Tamper attacks or Tamper Detection	Tampering
7	S6	Unpredictable app behavior or Complexity in estimating execution data or Limitation in profilers	Limitation in profilers
8	S7	Security challenges in mobile apps or Lack of development standards/ Practices	Lack of development standards/ practices
9	S8	Key testing issues for practitioners	
10	W3	Battery life escalation or Energy saving of mobile devices or Energy minimization of mobile devices or Task scheduling	Task scheduling
11	W4	Performance variation or Inefficient approach usage or Performance test generation	Performance variation
12	W8	GUI failures or Impact of mobile events on GUI or Screen orientation changes problem	Screen orientation changes problem
13	A1	Problems with emulators or Emulators related constraints or Construction of emulators for specific platforms	Construction of emulators for specific platforms

14	A3	Energy consumption of mobile devices or Estimation of battery time or Extension of battery life	Estimation of battery time
15	A5	Restrictions in touch and gesture-based interactions or Problems with touch-based interaction- gestures or Gesture-based interaction constraints	Gesture-based interaction constraints
16	A6	The problem in configuring mobile hybrid apps or Security & Privacy breaches in terms of configuration or Configuration problems	The problem in configuring mobile hybrid apps
17	A7	Challenges in Mobile app testing or Event-driven nature of mobile applications or Applicability/implementation of contextual features or Complexity of contextual features	Event-driven nature of mobile applications
18	A12	Security vulnerabilities in terms of patching or Patching or Patch distribution or Vulnerabilities in mobile devices	Patching
19	A14	Effect of design choices on software quality & performance or Tracking of antipatterns or Poor design choices	Effect of design choices on software quality
20	A16	Sensitive data deletion or Problem of data-erasure or Data erasing problem or Inflexible android security mechanisms or Limitations of data erasing and file deleting mechanisms	The problem of data- erasure

APPENDIX G

EXPERT REVIEW EVALUATION FORM:

SECTION I:

INVITATION LETTER

Respected Sir,

It is stated that my name is Mahrukh Tanveer and I am doing Masters in Software Engineering from the National University of Modern Languages, H-9, Islamabad. I am a research student and my research is based on a topic titled: "Requirement Engineering Process for Mobile Application Development". My first research question is about the identification of challenges for mobile development during the execution of the requirement engineering process. So, for that purpose, I conducted a systematic literature review and identified a list of approximately 53 challenges. Now, I want my list of challenges to be evaluated to complete phase 2 of my research study and move towards my next phase. For this reason, Expert Review is being conducted. So, please kindly, spare some time for validating my research problem. I shall be very thankful to you.

Yours' sincerely;

Mahrukh Tanveer

Department: Software Engineering

SECTION II:

PERSONAL INFORMATION OF EXPERT REVIEW:

Name:

Designation:

Year of Experience:

Expertise:

Domain:

Educational Qualification:

Additional Skills:

SECTION III:

TASKS TO BE PERFORMED BY REVIEWEE:

Task 1: To check the naming conventions given to a particular identified challenge

Task 2: To confirm the correctness of the classification (based on occurrence & nature of challenges both) for each identified challenge

Category 1: Classification of identified challenges based on the occurrence

Category 2: Classification of identified challenges based on nature

Acronyms:

DTNs: Delay Tolerant Networks

O.S.: Operating System

CC: Cloud Computing

QoS: Quality of Service

MCC: Mobile Cloud Computing

MEC: Mobile Edge Computing

ECC: Edge Cloud Computing

MCS: Mobile Crowd Sensing

IoT: Internet of Things

MAS: Mobile Analysis System

DVFS: Dynamic Voltage and Frequency Scaling

GUI: Graphical User Interface

HTTP: Hypertext Transfer Protocol

APs: Application Processors

CO: Computation Offloading

GPS: Global Positioning System

MD: Mobile Device

VM: Virtual Machine

CPU: Central Processing Unit

OTP: One Time Password

SMS: Short Message Service

MMS: Multimedia Messaging Service

PAP: Password Authentication Protocol

FIC-issues: Fragmentation Induced Compatibility Issues

TABLE G: Showing Expert Review Evaluation Form

No	Paper ID	Category 1	Category 2	Challenge	Description
1	X1	Network communication-related (communication in DTNs), Security & Privacy related	Domain-specific (Network), App-specific	Anonymous Communication	Delay tolerant networks (DTNs) are designed to handle data communications in networks with network latency. Anonymous communication challenge is that the identity of sender and receiver must not be revealed to a third party.
2	X2, X9, W10	Resource related (radio & computing resources, cloud resources), Requirement related (Security)	Domain-specific (MCC,ECC)	Resource Allocation Problem	Resources are the important aspect that must be considered while developing in any platform either mobile or simply traditional development. When we talk about a simple cloud environment that is limited in resources, the efficient utilization of resources in such a scenario is quite difficult, so, we have to assign the resources for the execution of the tasks very carefully. While on the other hand, in MCC or MEC where the tasks are volumetric & handling such a bundle of tasks is complex as it changes over time. So, resources must be allocated very efficiently so that better utilization of resources & workloads can be managed properly.
3	X3, S11a,	Requirement change management (fragmentation),	Platform/O. S. specific	Fragmentation	When developing mobile apps, developers frequently run across the issue of cross-platform

	S13, W6	Resource related (resource limitation)			incompatibility (for example, iPhone applications cannot be executed on the Android platform). Due to a scarcity of development resources, developers are compelled to support only a few platforms and devices. As a result of this situation, the fragmentation problem arises.
4	X4	Stakeholder related Requirement related, Communication-related, Knowledge related, Change-related, Scope related, Human-Factor related Social Organization related	Domain-specific (R.E.)	Requirement Gathering Challenges	a. **Stakeholder** (User Participation, Staffing, Stakeholder) b. **Requirement** (Prioritization, Schedule, Skill, Traceability) c. **Communication** (Articulation related, Unawareness of needs, Verbal & Presentation Skills, Culture & Language Barrier) d. **Knowledge** (Domain Related, Problem Analysis) **e. Change** (Management & Political Rules, Acceptance Criteria Changes, Unstable Requirements, Change in User

					Needs & Understanding) f. **Scope** (Over-Scoping, Ill-Defined Scope) g. **Human-Factor** (Conflicts, Ambiguities among Stakeholders, Intra-Group Conflicts, Communication Participation) h. **Social-Organization** (Policy & Structure, Complexity, Cultural & Time-Zone Differences
5	X5	Requirement related (interaction delay & energy consumption), Resources related (service resources)	Domain-specific (MEC)	Service Selection Problem	MEC provides high-quality services to users by executing tasks on network edge but due to limited resources and complexity of service requests, it's a major challenge how to select the appropriate services to minimize the interaction delay with users and energy consumption of mobile devices.
6	X6	Requirement (Security) related	App-specific	Guideline Compliance	Many security guidelines are generated for mobile development but having no technical feasibility and compliance with this security requirement.

7	X7	Requirement (energy consumption) related	Device-specific	DVFS-Control Problem	DVFS is a modification of power and speed settings on a computing device's different CPUs, controller chips, and peripheral devices to optimize resource allotment for activities and optimum power savings when those resources are not needed. Furthermore, the tendency of embedding several cores in recent mobile processors compounds the issue. Fine-grained DVFS control is available in multicore processors with on-chip voltage regulators, allowing separate cores to run at different operating points. In this instance, we need to coordinate the power mode of all cores for global optimization, which increases the DVFS control complexity exponentially and makes it difficult to implement power management policy in such a situation.
8	X8, A5	Resource related (radio & computing), Requirement related (processing completion time, response time, energy efficiency)	Domain specific (MEC, MCC)	Optimization Problem	Optimization refers to the best possible; when we talk about optimization problems then it simply means problems in achieving the best possible solutions. Whether it's simply handling multiple tasks simultaneously in any environment (MEC, MCC) or providing global tasks responsiveness, in both cases it is problematic to achieve the

					the optimal solution in terms of the execution time of all tasks.
9	X10	Requirement related (wastage of energy), Security & Privacy related (lost results, memory leaks)	Platform/ O.S. specific	Asynchrony Retrofitting	Asynchronous programming is an important aspect of mobile development to avoid unresponsiveness. Developers can use several async structures in Android. However, developers can continue to utilize inefficient async techniques, resulting in memory leaks, lost results, and wasted resources. While asynchrony retrofitting is the adoption of proper utilization of async constructs to avoid the aforementioned issues.
10	S1, S15, W2	Requirement related (energy saving, performance, app requirement, user requirement), Resource related (resource limitations, resource heterogeneity)	Domain-specific (MEC, MCC), App-specific	Offloading Decisions Problem	Computation offloading is meant for migrating the computation-intensive tasks to cloud servers. Whether it MCC or MEC environment or simple CO on the cloud, it is critical to make decisions about offloading the tasks that which tasks are to be offloaded and which must be executed locally (that is on a mobile device) to maximize the benefits of CO. This decision making is difficult due to dynamic nature, unpredictable behavior of MCC & MEC environment, resource limitations as well as the user and applications' both types of requirements must be taken into

					account for offloading decision making.
11	S2, S14, W1	Requirement related (energy consumption, computation power, execution time)	Domain-specific (MCC, MEC)	Multi-site Partitioning Problem	This problem is about the making partitions of tasks that are to be executed on the cloud and those which are executed on mobile devices for CO. Because CO is beneficial only when it benefits us with minimum energy consumption and minimal execution time. So, partitioning whether it's on a single site or multi-site, it's difficult in any situation to decide about the partitioning of offloaded and local tasks.
12	S3, A6, A15, A19	Security-related (android security, android vulnerability, malware threats), Privacy-related (privacy leakage)	Domain-specific (MCS/IoT), Platform/O. S. specific, App-specific, Device specific	Malware Detection	Mobile app security is becoming increasingly important since the software has become an integral part of billions of people's daily lives. As the most popular operating system, android is a prime target for malicious programmers looking to exploit flaws and propagate malware. Moreover, Android applications are not secure due to the open-source code of the Android platform.
13	S4	Requirement related (wastage of energy), Security & Privacy related (lost results, memory leaks)	Domain-specific (MCS/IoT)	Extraction of Useful Information	Crowdsensing refers to a technique in which a large number of people with mobile devices capable of sensing and computing share data and extract information to measure, map, analyze, estimate, or infer

		Platform/ O.S. specific Requirement related (energy constraints), Resource related (sensor data resources)			processes of common interest. Because MCS applications run in dynamic environments that comprise sensors, mobile devices, and the cloud so, it's critical to achieve energy-efficient and context-aware scheduling of the sensing process including data transmission from sensors to mobile devices and from the cloud to mobile devices. In other words, both the sensing process and data transmission from mobile devices to the cloud must be regulated to ensure that user data is only collected when an MCS application requires it.
14	S5	Requirement related (performance)	Platform/O. S. specific, App-specific	Inefficient Execution Model Problem	Android O.S. plays an important role in supporting mobile apps to deliver users with multiple benefits. But, due to inefficient execution techniques, existing Android apps have a significant challenge in terms of efficiency and quick responsiveness to user expectations. So, achieving efficiency and benefiting users in such a scenario is critical.
15	S6	Security-related (security increase data transmission), Requirement related (performance)	Domain-specific (MCC)	Timing Attacks	Timing attacks are a type of side-channel attack in which the attacker analyses the time it takes for a system to answer various queries to corrupt it. Because offloading

					sometimes demands multiple sending/receiving, it is particularly sensitive to timing assaults.
16	S7	Requirement related (performance), Security-related (from offloading data & code)	Domain-specific (MCC)	Tampering	Tampering usually refers to interference. In this context, it is the interference of unknown threats that arise when the code and data are offloaded to the public cloud, which is obviously, in every case an untrusted platform. So, it results in security risks which are problematic for the MCC environment to achieve high performance.
17	S8a, S12	Requirement related (power modeling)	Device-specific	Power modeling	Constructing power/energy models is the most complicated task for offloading systems. Firstly, it is due to accuracy dependence on various factors, different power co-efficient, model generation for one or specific smartphones, limitations on system accessibility. Secondly, models are built close to reality but it is no potential to fulfill all possible operating conditions. Moreover, The existing power models and their associated devices are being obsolete by a recent hardware developments.
18	S8b	Requirement related (power modeling)	Device-specific	Limitation of Profilers	Profilers are the software programs that are not only involved in monitoring the parameters of operating environments but also to control the available resources of

					mobile devices. The problem with the profiler is that it is difficult to calculate the possibilities of offloading data execution i.e. running time, network availability & communication cost. It is due to the unpredictable behaviors of mobile devices at runtime.
19	S9	Knowledge related (less experienced programmers, lack of best development practices), Security-related (less experienced programmers, lack of best development practices)	Device-specific	Lack of Development Standards	The proliferation of mobile software applications (apps) has posed a security risk. These apps are commonly available for little or no cost across all platforms and are frequently built by tiny businesses and inexperienced programmers. The mobile device is vulnerable to assaults due to a lack of development standards and best practices.
20	S10	Requirement related (eliciting testing requirements)	Domain-specific (Software testing)	Testing Issues for Practitioners	The challenges are the Need for eliciting testing requirements early in the development process, conduction of research in a real-world development environment, specific testing techniques targeting life-cycle conformance, mobile services testing, comparative study for security & usability testing.
21	S11b	Requirement change related (Fragmentation)	Platform/O. S. specific	Offline Execution of Web Apps	The mobile Web browsers are commonly developed using Web-related standards, so, web apps are more likely to be executed on several platforms than mobile

					applications. But, the issue with Web applications are that they cannot be used in offline mode.
22	W3	Requirement related (energy consumption & power capabilities), Security & Privacy related (protecting devices from threats, protecting piracy, leakage of private information), Network related (network disconnection, bandwidth, network traffic), Resource related (radio resource)	Domain-specific (MCC)	Constraints in Mobile Computing	Issues at communication side include low bandwidth which is due to radio resource availability than another wired network, service availability in which mobile users are unable to connect to the cloud due to network traffic, and so on, heterogeneity in terms of different networks involvement so the issue is that how to handle this wireless connectivity while satisfying MCC requirements. On other hand, we have issues regarding the computing side involving computation offloading in static & dynamic environments. When talking about static environment energy saving is not always effective, as in the case of code compilation, more energy is consumed a.c.t. local processing when data size is small. While in dynamic situation issues in terms of a network (network connection status & bandwidth) arises in the case when the data transmitted is not received by the destination node or lost on the server-side. The second issue on the computing side is related to security discussed in two contexts: 1) security in terms of

					mobile users which involves difficulty in protecting mobile devices for threats due to constraints of processing & power in mobile devices and while using location-based services such as GPS, more chances of private data leakage are there i.e. current location. 2) data security on the cloud further comprises the integrity of users data on the cloud in which the energy consumption of mobile users isn't taken into account, authentication in terms of complexity of using difficult to learn passwords, and digital right management in which protecting piracy of digital content such as audio, video, images and so on from illegal access is of critical importance for content providers of MCC like traditional CC and peer-to-peer networks. The third problem on the computing side is about enhancing the efficiency of data access because in this situation handling data resources on the cloud is not easy b/c of low bandwidth, mobility & resource capacity. The last limitation on the computing side is composed of context-aware cloud services in which achieving QoS at an acceptable level is not

					problematic with the disconnection issues.
23	W4	Requirement related	Domain-specific (MCC)	Task Scheduling	MCC constitutes high-performance data processing system but in this situation, it is complex to predict the exact scheduling of tasks that how to schedule mobile application tasks in data centers to extend the battery life of mobile devices.
24	W5	Requirement related	Domain-specific (Software testing), App-specific	Performance Variation	Due to the limited resources of mobile devices, performance is an important aspect. As several different mobile platforms along with unlike O.S. & hardware, native apps developed & maintained separately for these platforms resulting in varying performance.
25	W7	Requirement related	Domain-specific (GUI testing), App-specific	The complexity of GUI Testing	Graphical user interface (GUI) testing is a sort of mobile application testing that ensures that the GUI components work properly. Whether manual or automated, GUI testing often takes a lot of time and effort.
26	W8	Requirement related	Domain-specific (MCC)	Limited Battery Life	Mobile cloud computing (MCC) is a new technology that makes it easier to run complicated applications on mobile devices. For greater flexibility and mobility, mobile users are encouraged to carry out a variety of tasks utilizing their mobile devices. However, the limited battery life of mobile

					devices puts such advantages to the test.
27	W9	Requirement related (Performance)	App-specific	Complexity of Performance Evaluation	Given the limited resources available on mobile devices, a thorough performance study of a mobile app is essential. However, in the mobile sector, performance evaluation is still a manual and time-consuming process. The variety of mobile devices merely adds to the task's difficulty.
28	W11	Requirement related (Time & energy consumption), Resource related (limitation in resources of cloudlets)	Domain-specific (MEC)	Multi-Constraint Problem in MEC	Cloudlets in MEC have limited processing capabilities, making it impossible for them to provide on-demand resources for heavy activities. In this context, it's critical to look into the resource schedule issue. So both mobile devices as well as cloudlets are important in this concern. Furthermore, the time consumption and energy consumption of MDs, as well as the load balancing of cloudlets jointly should be considered. These three aforementioned concerns jointly are known as the Multi-constraint a problem in MEC.
29	W12	Resource related (Computation& Communication resources)	Domain-specific (MEC)	VM Migration Problem	Virtual machine (VM) migration is a critical issue in a MEC system which refers to the process of shifting a VM from one edge node to another. Virtual machines (VMs) are used to divide and distribute

					physical resources including processing power, storage, and network bandwidth. On the other hand, user mobility in terms of their free movement in a running application environment is the other concern. To achieve optimal performance in this context, it is crucial to decide how best to migrate VMs between the nodes in the MEC environment.
30	W13	Network related (bandwidth, disconnected mode), Resource related (cache size, data updates), Requirement (Power battery)	Domain-specific (Networks)	Restriction to Information Access	There are various drawbacks to the development of mobile information systems for accessing the information which on the top priority includes limited resources of mobile devices in terms of storage capacity and power. Furthermore, in wireless contexts, mobile client users experience frequent server disconnections, and accessing the server for data is costly in the wireless network. As a result, researchers have looked into caching solutions for better data management on mobile platforms. However, keeping the cached items in the cache up to date with the source data at the server is a huge difficulty.
31	W14	Security and Privacy related (attack to steal victims'	Device-specific	Side Glance Attacks	A side-glance attack and phishing assault are possible while browsing the mobile page and employing

		identity & relevant security information)			image code on a mobile device. It refers to a tangible attack in which an attacker keeps looking straight at the screen of a victim's mobile device to steal the victim's identity and relevant security information is known as a side-glance attack. When a victim types his or her identity (ID) and password to access a mobile bank or financial service in a mobile environment, the victim's entire ID and password may be exposed on the screen of the mobile device or through a touching sequence.
32	W15, A2	Network communication & Security related (message passing, android communication model has security weakness/ vulnerability associated with android model)	Platform/O. S. specific	Problems related to Android communication model	Android is extensively employed by mobile app developers all around the world. Android includes a message transmission system that allows apps to connect with one other and with each other. Because of the dangers that this system poses, it is critical to identify its risky actions and potential vulnerabilities. Malicious apps can force other apps into executing undesired activities and stealing end-user data while appearing normal and benign, because of Android's communication paradigm.

33	W16	Requirement related (GUI failures, app quality, user experience)	App-specific, Domain-specific (GUI testing)	Impact of Mobile Specific Event on GUI (orientation change event)	The problem of changing orientation is the distinctive event in mobile platforms which is commonly known as switching of the running app between portrait and landscape layout configurations. When this particular event occurs, Android guidelines recommend that the application adapts to the new layout, preventing memory leaks and retaining its state as well as any pending important message passing activity. But, unfortunately, putting this advice into implementation is not simple, and Android programmers will face programming hurdles as a result.
34	A1	Network related (network speed), Requirement related (testing is time consuming), Resource related (cache size, stack size, memory limits)	Domain-specific (Software testing)	Emulators related Constraints	Because of the limited computational resources and diversity of mobile surroundings, evaluating apps for mobile computing devices is time-consuming. There have been many different types of emulators suggested and deployed for this purpose in recent years but, often are unable to replicate the following: actual network speed and availability, actual device-specific content-rendering speed, memory limits, cache size, CPU speed, and stack size. These emulators are made for specific platforms and do

					not allow for application testing across a variety of mobile platforms which becomes a considerable challenge in the software testing domain.
35	A3	Requirement related (energy consumption of mobile device)	App-specific	Estimation of Battery Life	Mobile users are generally concerned about energy alarms with their devices, and they often take steps to extend battery life. Commercial smartphone platforms, such as Android and iOS, do not, however, include features that provide information regarding the remaining battery capacity. The issue is that most smartphone users are unaware of how long their battery will survive. So, estimating the accurate battery time availability of running applications is thought-provoking.
36	A4	Security-related (user password leakage)	App-specific	OTP – Vulnerability	A large number of user passwords have been exposed as a result of security breaches in user accounts. To improve the security of the Password Authentication Protocol (PAP) in such situations, Android app developers frequently use the short message service to provide a supplemental One-Time Password (OTP) authentication (SMS). But, SMS is not designed to be a secure service, so an SMS One-Time

					Password is subject to a variety of assaults.
37	A7	Requirement related (energy-consuming operation), Network communication-related (HTTP requests)	Domain-specific (Network), App-specific	Complexity in Bundling of HTTP Requests	For apps that operate on mobile devices, energy is a valuable resource. Making HTTP requests is one of the most energy-intensive tasks. Previous research has shown that combining smaller HTTP requests into a single bigger HTTP request can enhance network communication energy efficiency, but no automated method for detecting when apps can be bundled or transforming them to do so has been developed. As a result, it is complex for executing the bundling process.
38	A8	Requirement related (gesture & touch-based interfaces)	Domain-specific (UI testing), App-specific	Gesture-based Interaction Constraints	In mobile applications, touch and gesture-based interfaces are widespread. Smartphones and tablets produced a greater demand for specialist software engineering methodologies as they evolved into mass-market products. Consistent and effective testing is critical in software development to ensure high-quality solutions. Testing mobile applications, on the other hand, is still inconvenient, time-consuming, and error-prone. The smartphones' emphasis on touch-based interaction is one factor — gestures are difficult to incorporate

					into automated application assessments.
39	A9	Network Communication-related (data communication), Requirement related (energy overheads)	Device-specific	Data Communication Overheads	Mobile applications use device energy to operate, and the rapid depletion of battery power on mobile devices is a key usability issue. Data communication is the second-largest consumer of mobile device energy after the display. Advancements in battery and power management technology for mobile devices have not remained consistent with end-user needs or mobile application requirements, and given the increased energy consumption that is likely to be associated with future networking developments, the situation will only worsen over time. Furthermore, the device will use less energy if there is less data flowing through the network interface but unfortunately it doesn't happen to make it problematic on the data communication side.
40	A10	Privacy & Security related (privacy & security breaches)	App-specific	Problems with Configuration of Mobile Hybrid Apps	Hybrid mobile apps are web apps wrapped in a native software shell, connected to whatever features the mobile platform gives through a browser embedded in the app once it's downloaded and installed locally from an app store. Users using smartphones are more concerned

					about their privacy a.c.t. laptops & are hesitant for doing privacy-sensitive and financial actions on their smartphones due to their untrustworthy nature. On the other hand, smartphones are not secure in terms of configuration as well as many programmers consider them as non-functional and unimportant a.c.t. the code's primary purpose, forgetting that the impact of such defects may not always interfere with program logic but only arise after security breaches.
41	A11	Requirement related (event-driven & contextual features)	Domain-specific (Software testing), App-specific	Challenges in Mobile App Testing	Because of the event-driven structure and complex contextual features of cellular phones, testing these applications presents unique challenges (e.g. sensors, notifications). Due to the required instrumentation or platform dependence, current automated input generation options for Android apps are frequently not supportable for developers to work and do not effectively implement contextual features.
42	A12, A13	Requirement related (energy consumption)	App-specific	Energy measurement	When it comes to developing mobile applications, battery life is an important performance and user experience measure that must be taken into consideration. App developers, on the other hand, find it

					difficult to quantify how much energy their apps consume and to investigate how that energy use changes as a result of factors outside their control, such as network congestion, mobile operator choice, and user screen brightness settings.
43	A14	Requirement related (power consumption & surface overheating)	Device-specific	Power & Thermal Analysis	Recently, mobile applications have become more performance and resource-heavy, resulting in a significant battery drain and high surface temperature, degrading the user experience even more. As a result, high power consumption and surface overheating have been viewed as serious design challenges for smartphones.
44	A16, A17	Privacy related (privacy leakage)	App specific	Privacy disclosures	Privacy disclosure refers to the network requests sending out one or multiple types of private data. Android is a well-known and popular platform that provides rich functionality to users for accessing personal sensitive data resulting in serious privacy threats. Several methods were proposed to detect these threats but these all either fail to implement the privacy policies or to decide its requirement for app functionality. In other words, it is problematic in determining if a privacy-sensitive data object, such as a user's location or identity, is

					required for the app's main functionality or is simply being transmitted to numerous third parties. For example; Google Maps requires the user's position information to provide driving directions while in the case of providing weather services, a weather app may ask for the user's location.
45	A18	Requirement related (energy savings & delay)	App-specific	Energy Minimization	Continuously running mobile applications, such as those for health and context monitoring, must be energy efficient. Postponing the execution of delay-tolerant activities until a time when they would spend less energy is an appealing way to save energy in such systems. But, adding delays to preserve power, on the other hand, may have a negative influence on the user experience.
46	A20	Requirement related (network performance)	App-specific	Limitation of Network Architectures	To access multimedia content, mobile applications are heavily reliant on Internet services allowing users to exchange and/or download photos, audio, and video from the Internet using their mobile devices. As a result, the inclusion of multimedia content causes app developers, users, and manufacturers to prioritize energy conservation. Different limits, such as battery and storage capacity, broadcast constraints, user

					interferences, disconnections, noise, limited bandwidths, and network delays, constantly confront mobile devices and their supporting wireless networks and communications affecting the overall network performance.
47	A21	Requirement related (face authentication & finger-print scanning)	Device-specific	Environmental Factors' Impact on Biometric Authentication Method	Using the current biometric methods is highly reliant on a variety of environmental conditions. Face authentication, for example, is dependent on illumination, camera shake, and picture framing, whereas fingerprint scanning is dependenton finger placement. Because of all of these factors, it becomes difficult for users to use such systems and making it time-consuming as well.
48	A22	Security-related (security vulnerabilities)	Platform/O. S. specific, Device specific	Patching	A patch refers to a sequence of adaptations in a computer program or its supporting data intended to update, correct, or improve it. It involves addressing security flaws and other defects, and such updates are sometimes referred to as bug fixes. Patches are frequently made to improve a program's functionality, usability, or performance. Every time, the security breach when found, it is fixed. The patching problem on the other hand addressed is that, this patching mechanism is only for

					those applications that are directly connected or supported by Google. No other manufacturers or third-party applications are updated through the patching process.
49	A23	Privacy-related (protection of private data)	Domain-specific (MCS)	Data Reconstruction Attacks	Data reconstruction attack in general is an approach for partially recreating a private dataset using public aggregate information. Considering the MCS environment where comprises a cloud server architecture commonly used to combine information from participating users. As, local sensor data contains or can be used to deduce users' private information, uploading the data to the cloud and allowing third parties access to the data puts participants at risk of privacy leakage. This privacy leakage, as a result, gives rise to data reconstruction attacks which is a a major barrier in the MCS domain.
50	A24, A27	Requirement change management (fragmentation)	Platform/O. S. specific, App-specific	FIC issues	With over 80% market share, Android is the most popular smartphone operating system. The number of Android apps is experiencing exponential growth, with over 35,000 new apps being released on Google Play each month. However, Android OS is updated often, and dealing with compatibility issues across different

					OS versions is a well-known concern for app developers.
51	A25	Requirement related (power demands)	Device-specific	Power Demand Extension for Battery	Today's mobile device technology is fast evolving, and more cutting-edge technologies (e.g., augmented reality, voice control, and holographic) are being introducedto the devices, increasing the power demands on the battery exponentially. Furthermore, to increase usability, mobile device manufacturers are aiming towards lighter and thinner devices, which makes battery design even more difficult because it must offer a considerable quantity of energy under very tight limits. Unfortunately, the latest configuration provided by the battery of mobile devices, namely Li-ion batteries, was unable to keep up, and battery life was sometimes compromised to satisfy those rising power needs and design constraints. As a result, one of the most common complaints about current mobile devices are battery life.
52	A26	Requirement related (performance & quality)	App-specific, Domain-specific (Software quality)	Poor Design Choices	The software programs must change through time to deal with the introduction of new needs, adapt to new settings, correct errors, and improve software architecture. However, software quality may

					decline as a result of software aging, independent of the type of changes made. Because of the injection of poor design and implementation choices into software, software quality deteriorates over time. Antipatterns and code smells are signs of poor decisions.
53	A28	Security & Privacy related (data deletion is vulnerable)	Device-specific, Platform/O. S. specific	The problem of Data Erasure	Android has long been a target of privacy concerns due to its popularity as the most widely used mobile operating system. The problem of data erasure is common in android O.S. The foremost problem in this O.S. is how this system & its applications handle data, such as when and when data is accessed, updated, or transferred. As, android does not provide adequate clarity regarding how third-party applications process user data stored on a mobile device, data reappearance after unsafe erasure could be a risk as a result.

APPENDIX H

SURVEY FORM:

SECTION I:

INVITATION LETTER

Respected Sir,

It is stated that my name is Mahrukh Tanveer and I am a research student doing Masters in Software Engineering from the National University of Modern Languages, H-9, Islamabad. My research is based on the topic named "Requirement Engineering Process for Mobile Application Development: Challenges and Resolutions." My third and last research question is about the mitigation strategies that mean to propose the solutions to overcome the identified challenges that are covered in phase 1 (RQ1). For the discovery of challenges, a systematic literature review is conducted in phase 1 and briefly reviewed and improved by different experts in phase 2 of Expert Review. Now, I want my final list of reviewed and improved challenges to be looked upon to propose some mitigation plans for each respective challenge. For this purpose, I am conducting an Industrial survey. So, please spare some time to suggest mitigation strategies to overcome the identified challenges. I shall be very thankful to you.

Yours' Sincerely,

Mahrukh Tanveer

Department: Software Engineering

SECTION II:

PERSONAL INFORMATION:

Name of person:

Organization name:

Organization size:

Designation:

Overall Experience:

Experience in Mobile Development:

NOTE: The description of challenges are attached in ANNEX A

SECTION III:

Table H1: Showing the questionnaire form designed for industrial survey conduction to get mitigation strategies

Category Name	Respective Challenge(s)	Mitigation Strategies
Communication	Anonymous Communication	
	Lack of requirement effective articulation	
	Lack of Verbal & Presentation skills	
	Lack of Communication participation	
Requirement	Incomplete requirement gathering	
	Lack of accurate requirement prioritization	
	Unstable requirements	
	Change of user needs & understanding	
	Requirement over scoping	
	Inefficient requirement completion time	
	Lack of consideration of user & applications requirements for offloading decision making	
	Incorrect requirement partitioning	
	Lack of useful information extraction	
	Changing the orientation of the app	
	Diversity of mobile surroundings	
	Event-driven structure	
	Complex contextual features	
	Frequent changing requirements	
Resource	Lack of appropriate resource allocation	
	Platform incompatibility	

	Lack of resource optimization	
	Energy inefficiency	
	Inaccurate task scheduling	
	Limited resources/resources lacking	
	Lacking computational resources	
	Incorrect estimation of battery life	
	Lack of accurate quantification about the consumption of energy by the app	
	Compatibility across various OS versions	
Security & Privacy	Lack of asynchrony retrofitting	
	Tampering during offloading data	
	Limitation of profilers	
	Lack of identification of risky actions and vulnerabilities	
	Inconsistent and inefficient testing	
	Lack of configuration of mobile hybrid Apps	
	Unclear requirements for app functionality towards privacy threats	
	Patching for updation, correction or improvement	
	Data erasure	
Stakeholder	Unawareness of needs	
	Cultural and language barrier	
	Lack of domain knowledge	
	Ambiguities among stakeholders	
	Intragroup conflicts	
	Inefficient response time	
	Lack of requirement task efficiency and responsiveness	
	Lack of development standards and practices knowledge	

	Testing issues for practitioners	

SECTION IV:

Table H2: Showing the questionnaire form designed for industrial survey conduction to evaluate respective challenges based on practicality level

Category Name	Respective Challenge(s)	Practicality level of challenges				
		Very High	High	Moderate	Low	Very Low
Communication	Anonymous Communication					
	Lack of requirement effective Articulation					
	Lack of Verbal & Presentation Skills					
	Lack of Communication participation					
Requirement	Incomplete requirement Gathering					
	Lack of accurate requirement Prioritization					
	Unstable requirements					
	Change of user needs & understanding					
	Requirement over scoping					
	Inefficient requirement completion time					
	Lack of consideration of user & applications					
	Requirements for offloading decision making					
	Incorrect requirement partitioning					
	Lack of useful information Extraction					

	Changing the orientation of the app					
	Diversity of mobile surroundings					
	Event-driven structure					
	Complex contextual features					
	Frequent changing requirements					
Resource	Lack of appropriate resource Allocation					
	Platform incompatibility					
	Lack of resource optimization					
	Energy inefficiency					
	Inaccurate task scheduling					
	Limited resources/resources Lacking					
	Lacking of computational resources					
	Incorrect estimation of battery Life					
	Lack of accurate quantification about the consumption of energy by the app					
	Compatibility across various OS versions					
Security & Privacy	Lack of asynchrony retrofitting					
	Tampering during offloading Data					
	Limitation of profilers					
	Lack of identification of risky actions and vulnerabilities					
	Inconsistent and inefficient testing					

	Lack of configuration of mobile hybrid apps					
	Unclear requirements for an app functionality towards privacy threats					
	Patching for updation, correction or improvement					
	Data erasure					
Stakeholder	Unawareness of needs					
	Cultural and language barrier					
	Lack of domain knowledge					
	Ambiguities among Stakeholders					
	Intragroup conflicts					
	Inefficient response time					
	Lack of requirement task efficiency and responsiveness					
	Lack of development standards and practices knowledge					
	Testing issues for practitioners					

ANNEX A:

Category Name	Respective Challenge(s)	Description of Challenges
Communication	Anonymous Communication	This challenge is that the identity of sender and receiver must not be revealed to the third party.
	Lack of requirement effective articulation	Users and developers both consider the requirements with their perspectives which leads to conflicts between them.

	Lack of Verbal & Presentation skills	This challenge relates to presenting the views and thoughts making others difficult to understand the ideas.
	Lack of Communication participation	Is refers to unclear and incompleteness of representation of ideas and thoughts.
Requirement	Incomplete requirement gathering	The absence of the necessary and compulsory requirements.
	Lack of accurate requirement prioritization	The compulsory requirements could not be executed first or compulsory requirements given low priority.
	Unstable requirements	The requirements are not clear enough to achieve its goal. Or the requirements failed to capture their expectation.
	Change of user needs & understanding	The needs of users constantly change over time which leads to a lack of understanding of user needs.
	Requirement over scoping or ill-defined scope	The scope for requirements is not properly defined which lead to conflicts and stakeholders dissatisfaction.
	Inefficient requirement completion time	The lack of achieving optimal solution when the requirement could not be completed on estimated time.
	Lack of consideration of user requirements & application requirements for offloading decision making	The requirements related to the user and application could not be considered to make decisions for the migration of computation-intensive tasks.

	Incorrect requirement partitioning	This problem is about making wrong partitions of tasks that are to be executed on the cloud and those which are executed on mobile devices for computation offloading.
	Lack of useful information extraction	As MCS applications run in dynamic environments that comprise sensors, mobile devices, and the cloud so, it's critical to achieve energy-efficient and context-aware scheduling of the sensing process including data transmission from sensors to mobile devices and from the cloud to mobile devices which is difficult for extracting useful information.
	Changing the orientation of the app	The problem of changing orientation is the distinctive event in mobile platforms which is commonly known as switching of the running app between portrait and landscape layout configurations.
	Diversity of mobile surroundings	The occurrence of a variety of mobile platforms that are programmed with their respective programming languages leads the mobile environment to be diverse.
	Event-driven structure	The mobile environment is bounded in a complex structure having multiple events that could not be properly handled
	Complex contextual features	The contextual features such as notification and sensor handling in mobile development due to platform dependence are difficult to implement making it one of the major challenges in mobile environment.
	Frequent changing requirements	The constantly changing requirements may result in poor design choices degrade the software application performance and its quality.
Resource	Lack of appropriate	The task that requires more resource allotment could not be allocated or there could be a possibility that

	resource allocation	the task needs fewer resources provided it with more allotment leading the resource wastage.
	Platform incompatibility	Due to scarcity of development resources, developers are compelled to support only a few platforms and devices. As a result of which mobile developers face platform incompatibility issues.
	Lack of resource optimization	The mobile environment lack occurrence of optimal resources or it is unable to provide the best resources for tasks execution.
	Energy inefficiency	The computation intensive tasks in mobile environment consume more energy due to bounded resources.
	Inaccurate task scheduling	The exact estimation and prediction of the scheduling of mobile application tasks is challenging in mobile platforms.
	Limited resources/resource lacking	In comparison with traditional development, the mobile environment is bounded by limited resources.
	Lacking Computational resources	The mobile environment offers less memory and computing power than conventional PC systems.
	Incorrect estimation of battery life	Mostly, smartphones are unaware of how long their battery will survive. So, estimating the accurate battery time availability of running applications is thought-provoking creating a challenge in mobile development.
	Lack of accurate quantification about the consumption of energy by the app	It is quite difficult for mobile developers to quantify how much energy their apps consume and to investigate how that energy use changes as a result of factors outside their control, such as network congestion, mobile operator choice, and user screen brightness settings.

	Compatibility across various OS versions	The Android ecosystem is heavily fragmented. The occurrence of countless OS versions makes it impossible for mobile developers to test their apps. As a result of which, various compatibility issues arises leading to poor user experience.
Security & Privacy	Lack of asynchrony retrofitting	The mobile app developers are unable to utilize async constructs or techniques which results in memory leaks lost results and wasted resources.
	Tampering during offloading data	It is the interference of unknown threats that arise when the code and data are offloaded to the public cloud, which is obviously, in every case an untrusted platform. So, it results in security risks which are problematic for the MCC environment to achieve high performance.
	Limitation of profilers	The problem with the profiler is that it is difficult to calculate the possibilities of offloading data execution i.e. running time, network availability & communication cost. It is due to the unpredictable behaviors of mobile devices at runtime.
	Lack of identification of risky actions and vulnerabilities	The android platform enables apps to connect. Due to the hazards posed by this system, it is important to identify its risky actions and possible threats.
	Inconsistent and inefficient testing	Touch and Gesture-based interfaces make software testing inconsistent and inefficient in a mobile environment.
	Lack of configuration of mobile hybrid apps	Smartphones are not secure in terms of configuration as well as many programmers consider them as a non-functional and unimportant a.c.t. the code's primary purpose, forgetting that the impact of such defects may not always interfere with program logic but only arise after security breaches.

	Unclear requirements for app functionality towards privacy threats	It is difficult to detect the privacy threats in mobile development because these all either fail to implement the privacy policies or to decide its requirement for app functionality.
	Patching for updation, correction or improvement	The patching mechanism is only for those applications that are directly connected or supported by Google. No other manufacturers or third-party applications are updated through the patching process.
	Data erasure	The problem is particularly found in O.S. which refers that how this system & its applications handle data, such as when data is accessed, updated, or transferred. As, android does not provide adequate clarity regarding how third-party applications process user data stored on a mobile device, data reappearance after unsafe erasure could be a risk as a result.
Stakeholder	Unawareness of needs	The mobile app requirement is typically different from the traditional development. The unawareness of needs in this context may cause problems among stakeholders.
	Cultural and language barrier	The differences in practices and development standards among different workplaces create one of the major challenges in mobile development.
	Lack of domain knowledge	Mobile development is a diverse and vast domain so, lack of domain knowledge is one of the major barriers faced by mobile developers.
	Ambiguities among stakeholders	Contradiction in views among the participating stakeholders.

	Intragroup conflicts	Disagreement or difference in opinions creates conflicts among members in a group usually when they work in teams.
	Inefficient response time	The execution of tasks in a mobile environment is time taking process which leads to inefficient response time.
	Lack of requirement task efficiency and responsiveness	Due to inefficient execution techniques in Android O.S., the existing apps have a significant challenge in terms of efficiency and quick responsiveness to user expectations.
	Lack of development standards and practices knowledge	The proliferation of mobile software applications (apps) has posed a security risk. These apps are commonly available for little or no cost across all platforms and are frequently built by tiny businesses and inexperienced programmers. The mobile device is vulnerable to assaults due to a lack of development standards and best practices.
	Testing issues for practitioners	The challenges are the Need for eliciting testing requirements early in the development process, conduction of research in a real-world development environment, specific testing techniques targeting life-cycle conformance, mobile services testing, comparative study for security & usability testing.

APPENDIX I

Table I: Showing the Mitigation Strategies obtained in Survey Conduction

Challenge	Mitigation Strategies	Number of Responses	Practicality Level of Challenge
Anonymous Communication	Encryption-Decryption mechanisms/ techniques/ Algorithms	33 responses	V.H: 8 or 10.4% H: 10 or 13% M: 63 or 75.3% L: 2 or 2.6% V.L: 1 or 1.3%
	Integrate encrypted channels/ do introduce encrypted Challenges	24 responses	
	Use of APIs, tokens, or encrypted keys/ encrypted keys must be embedded/ encrypted keys and tokens can resolve the issue	16 responses	
	Kanban, Scrum, extreme programming	7 response	
	Both parties should use a platform that does not require their identities for communication. It should be made sure that the chosen platform provides end-to-end encryption because if the communication medium is secured, there is no chance for any intruder to get any information about the sender and receiver.	1 response	
	We can provide secure APIs using tokens or encrypted keys. So when a user is fetching the data or even store in a database, it will be based on token or Authorization keys. So user-ID will be secured and a third party can't access the information for any user.	1 response	

	Embed linguistics Translator/ use linguistics translator	2 responses	
Lack of requirement effective articulation	In my opinion, this can be resolved by properly onboarding users in the app. There can be some video or detailed overview of the features on the first launch of the app and this can be provided as a feature in the app so the user can even see this later on.	1 response	V.H: 13 or 16.9% H: 64 or 77.1% M: 4 or 5.2% L: 2 or 2.6% V.L: 0 or 0%
	Conduct stand-up meetings/ stand-up meetings must be there/ must have daily meetings/ daily sprint meetings	67 responses	
	The requirement gathering should be done properly. All the tasks and scope should be defined in Scrum. So no one has the conflicts/ check requirements carefully/ re-check requirement gathering phases/ The development of any project should start with the understanding and incorporation of user requirements in the development process.	3 responses	
	Use JIRA for project management and PM should write detailed requirements about the feature	1 response	
	Communication between developer and client either verbally or graphically can overcome the challenge	3 responses	
	Proper documentation/ specification document	2 responses	

	make prototypes and have feedback from the users	1 response	
	There should always be some reference applications to keep an eye on existing features and after detailed analysis, some documents and a flow should be locked to act upon.	1 response	
	Never start work unless a listen to a go from both sides, one should define and the other one should note what he's getting, now he should define and the other one-note, practice this multiple time and at last you will get a final results	1 response	
	To minimize the conflicts, requirements should be revised several times. Also, it should be made sure that the developers are given a chance to communicate directly with stakeholders so that they could understand their perspective and both parties could agree upon a potential solution.	1 response	
	Developers make prototypes that are just a model of the requirement, like Designs on Figma (tool), that provides a basic knowledge of what this requirement will do. In which Figma provides the click functionality as well, you can click on buttons in the designs and it will move to the next screens as required in the functionality. Users can easily address developers to add something or remove and when he approves it, then Frontend developers can start work.	1 response	
Lack of Verbal & Presentation skills	Apply UML/ UML implementation/ UML concepts	23 responses	V.H: 6 or 7.8% H: 15 or 19.5% M: 67 or 74.4%
	Language barrier reduction	2 responses	

	Generate models	7 responses	L: 1 or 1.3% V.L: 1 or 1.3%
	Follow model-based communication	9 responses	
	Model generation and implementation of UML Concepts	3 responses	
	Effective visual presentation	1 response	
	Present using diagrams instead of long paragraphs	1 response	
	Generate models by applying UML diagrams	19 responses	
	To present and understand the ideas effectively, discussion sessions should be arranged where both parties could try to explain their ideas in general terms (non-technical way) making it easy for others to grasp the ideas.	1 response	
	We propose them the ideas using Data Flow Diagram, Use Case Diagram, Sequence Flow Diagram, etc. So the user knows where we stand and then the user and developers will be on the same page. / introducing models/ use diagrammatic representation	23 responses	
	Daily scrum meetings must be there	1 response	
Lack of Communication participation	Model based communication/ models/ diagrams/ communicate via models/ construct models/UML models	79 responses	V.H: 8 or 10.8% H: 9 or 12.2% M: 65 or 79.2% L: 1 or 1.4% V.L: 1 or 1.4%
	Requirement elicitation technique	1 response	
	Documentation must be there	2 responses	

Incomplete requirement gathering	Interviews or questionnaires/ conduct interviews/ Interviews + questionnaires/ Interviews from relevant stakeholders	73 Responses	V.H: 69 or 78.4% H: 9 or 12.2% M: 7 or 9.5% L: 3 or 4.2% V.L: 0 or 0%
	The proposed solution must be discussed carefully.	1 response	
	Proper implementation of V& V model on Requirement	1 response	
	Always take feedback from client-side when you complete your first module/ Try to take feedback from clients after completion of each module/chunk/ Take feedback from customers after completion of first release	6 responses	
	Meetings, online meetings or physical, what meeting on both side, from development to client, don't miss anyone, all will be stakeholders, so don't miss anyone. Gather requirements and pass to other one/ Brainstorming and meeting with the client many times during development.	2 responses	
	The requirements should be revised between clients and developers. Stakeholders should be asked to explain their required system multiple times. The elicitor should have enough domain knowledge to understand stakeholders. The requirements elicitor should repeat what he understood in his terms, with added information - if needed, so that each requirement gets validated.	1 response	

	When the user has an innovative idea to add to the app, the developer team, mostly software engineers do the R&D to gather the requirement and make a list of challenges they will face during the development including third-party libraries or servers and their costs. They acknowledge user about that, and if a user can afford those challenges then the team proceed to development.	1 response	
	Requirements documents should be revised after taking comments from system architect, developers and quality assurance team/ User specifications document to be prepared by the developers and dually signed/agree/ SRS should be properly documented	3 responses	
Lack of accurate requirement prioritization	Perform storyboard approach/ apply storyboard techniques/ Perform Storyboard approach	69 Responses	V.H: 8 or 10.7% H: 8 or 10.7% M: 62 or 74.6% L: 2 or 2.7% V.L: 2 or 2.7%
	The ranking or Voting the Requirements	2 responses	
	Product backlog, sprint backlog/ Use of scrum can minimize above risk	3 responses	
	It is a part of the software engineers or analytics team to arrange the compulsory requirements in such an order that the main functionalities won't be missed out, and by the time any new important functionality comes into the app then they should manage it without disturbing other functionalities or requirements. Mostly we use JIRA (Atlassian account feature) to manage the work and scrum master creates tasks and assigns the priorities for each task.	1 response	

	The developer should have enough domain knowledge to prioritize the requirements incorrect order. If a developer has difficulty ranking them correctly, he should study the domain and understand it with examples systems.	1 response	
	Do clarify your business need first then prioritize your requirements on the basis of business need	1 response	
	Everything should be define in Scrum with priority and deadline. So, they can follow everything. Team leads should help the developers where they stuck or won't fulfil any requirement. So they won't switch the prioritize tasks.	1 response	
	It totally depends on the business team how they want to proceed with the product according to their targeted user. So prioritize features that are more demanding to the users.	1 response	
	This demonstrates the lack of understanding by the developer team. The compulsory requirements should be catered in the first place. Subsequently, other requirements may be fulfilled.	1 response	
	Requirement should be analyzed properly with the involvement of client/ Involve customer/ Conduct meeting with clients	3 responses	
	follow mobile development lifecycle, prioritize the task step wise	1 response	
Unstable requirements	Daily meetings must be conducted/ daily meetings should be conducted/ Must have daily meetings/ Daily meetings/ Daily scrums meetings must be Done	73 responses	V.H: 7 or 9.7% H: 64 or 77.1% M: 10 or 13.9%

	Requirements tested through TDD/ We should create the hierarchy where the developers should do unit testing and then they release the build with a document of what they achieve in this build and business logic steps should be mentioned in the document. So, they can give the build the TLs and then they test that build and if found anything then they must ask developers to fix them and pass this build to QA.	2 responses	L: 0 or 0% V.L: 2 or 2.8%
	More time should be spent to understand the system, requirements should be discussed repeatedly between parties to obtain stable ones. Different gathering techniques should be used to get the correct requirements. For example, rough sketches for the flow of each required feature should be made repeatedly until improved to stability.	2 responses	
	Sometimes the framework or OS versions don't support the functionality or requirement. And sometimes miscommunication between the teams leads to failing tasks. We should use daily scrum meetings to avoid that anomaly/ daily scrum meetings/ this issue can be resolved via daily scrum Meetings	3 responses	
	Use of extreme programming and agile to include changing requirements	1 response	
	Change management, sprint backlog, product backlog, and kanban support change management so well.	1 response	
	Analyze the requirements	1 response	

Change of user needs & understanding	Take customer feedback/ user feedback/ release your product for customer feedback/ take feedback from clients	33 responses	V.H: 5 or 6.8% H: 74 or 82.2% M: 9 or 12.2% L: 2 or 2.7% V.L: 0 or 0%
	Release your product as a beta version to take feedback from users/ release your product for customer feedback/ take customer feedback after the release of your product	21 responses	
	Take client feedback after each module or task	29 responses	
	Prototyping before actual Development	1 response	
	This leads to scope creeping, you should stick to documented requirements and changes can be accommodated in next phase	1 response	
	Work on beta versions, do a QA properly	1 response	
	If the user needs changes arehuge development should be stopped and quick requirements understanding sessions should be arranged between stakeholders and developers and it should be made sure that the user has agreed upon a new or updated set of requirements.	1 response	
	Frequent meetings with stakeholders should be conducted to ensure consistency	1 response	
	On-board customer as increase transparency	1 response	
	Product backlog	1 response	
Requirement over scoping or ill-defined scope	Re-consider the requirements and re-define the scope/ redefine requirements and scope/ consider your requirements and scope again/ define scope again/ re-define scope	39 responses	V.H: 9 or 12.3% H: 12 or 16.4% M: 58 or 70.7% L: 3 or 4.1% V.L: 0 or 0%
	Meetings with stakeholders/ pm & stakeholders role/ conduct meetings with a project manager to discuss or re-define requirements & scope/ Involve your	37 responses	

	stakeholders and discuss with project manager/ Stakeholders and PM involvement is necessary here		
	Create the prototypes while sharing the ideas and the solution architect should involve in that meeting so he can suggest which technology we might use and how far the possibilities where we can achieve the specific requirement.	1 response	
	A standard form of documentation should be adopted to avoid conflicts.	1 response	
	A project scope definition document should be prepared to overcome this issue. It includes the outcome of a particular project and its associated benefits and also outlines any constraints imposed upon the project and the assumptions that have been made along the way/ follow scope definition Document	2 responses	
	For any type of project, the definition of your project scope could very well change (it almost certainly will), during the life of the project but each change should be controlled and managed to avoid "scope creep" where the initial aims of the project are obscured by ongoing modifications.	1 response	
	Agile frameworks support continuous changes by users... I think it's kanban.	1 response	
	Work breakdown structure must be implemented/ WBS approach/ Follow WBS approach/ Implement	69 responses	V.H: 8 or 10.8% H: 10 or 13.5%

Inefficient requirement completion time	WBS approach/ apply WBS concept/ Break your project into smaller modules/ project breakdown		M: 59 or 71.6% L: 1 or 1.7% V.L: 2 or 2.4%
	Sprint backlog, stand up meetings	1 response	
	Analyze the requirements carefully	1 response	
	Split the bigger task in smaller chunks, that make easy to meet deadlines	1 response	
	The Schedule of a project should be followed religiously and some extra time should be reserved to compensate for the change in requirements	1 response	
	First of all, the user must be told that if the requirement is huge and must be completed on time then we will need the number of developers to work on. Secondly, make weekly tasks on JIRA Atlassian board that keep track of each task based on story points and estimated time. If organizations follow these rules they won't miss the timeline.	1 response	
	Also, during the time estimation period, a fair time margin should be added to the total estimate of task completion time. This margin should be utilized to avoid delays. And it should be kept reserved, if not needed, for potential delays in upcoming tasks.	1 response	
	At first, a developer should prioritize all tasks correctly. Next, each task should be divided into small goals and it should be made sure that each goal is set using SMART method: Specific, Measurable, Attainable, Relevant, and Timely.	1 response	

	PM should prioritize the major functionality which is necessary and the other functionalities should be delivered in iterations/ Requirement prioritization, requirements with high priority should be given more time and vice versa	4 responses	
Lack of consideration of user requirements & application requirements for offloading decision making	Face 2 face meetings between clients and developers/ conduct face to face meetings between clients and Developers	67 responses	V.H: 5 or 7% H: 76 or 82.6% M: 8 or 11.3% L: 3 or 4.2% V.L: 0 or 0%
	Rapid feedback through white box process	1 response	
	Must see the user requirement carefully/ We should understand the user needs and what is in trending/ Requirement gathering must be clear and brief	21 responses	
	During the planning stage, software architecture should be also planned before implementation	1 response	
	Data engineers collect the data logics of each requirement that will be handled through the cloud. Not everything will be handled on the Frontend side of an app it will make the app overloaded and irresponsive.	1 response	
	User preferences should be given priority	1 response	
Incorrect requirement partitioning	Scrum/Follow Scrum/Implement Scrum	77 response	V.H: 6 or 8.2% H: 64 or 77.1% M: 10 or 13.7% L: 2 or 2.7% V.L: 1 or 1.4%
	This will resist the team to follow agile methodology. The sprints should be made so that they will focus on specific features instead of touching every feature a bit.	1 response	
	Must be able to prioritize the requirement step by step/ The requirement prioritization phase should be completed properly	2 responses	
	Implement WBA through JIRA platform normally use in a scrum to overcome this issue, existing systems	1 response	

	should be studied to understand how partitioning is done in them and what is the ratio between success and failure.	1 response	
	Various partitioning methods need to be implemented and evaluated. And the best strategy to be implemented.	1 response	
Lack of useful information extraction	Automated sensors/ introduce automated/ embed automated sensors/ use automated sensors/ by deploying automated sensors	76 responses	V.H: 4 or 5.6% H: 8 or 11.1% M: 63 or 76.8% L: 5 or 6.9% V.L: 2 or 2.8%
	We should create a report that when for fewer mobile users and then run the cron jobs that schedule for the specific time when the public don't use their mobile phone so we can achieve heavy data transmissions easily like backup plans and updates.	1 response	
	Involve the usage of automated sensors that must be activated when required. In this way, we can overcome scheduling and energy issues and hence, extraction of useful information is there.	1 response	
	Use data processing tools to handle these scenarios/ tools usage	2 responses	
	Secure data transmission should be followed	1 response	
	Threading in background service	1 response	
Changing the orientation of the app	Hire UI experts/ UI experts? Recruit UI specialists/ hire UI domain experts	81 responses	V.H: 7 or 9.6% H: 7 or 9.6% M: 69 or 80.2% L: 4 or 5.5% V.L: 1 or 1.4%
	Set the executable Prototype before live release	1 response	
	We should calculate the resolutions and size of the device and whenever someone change the orientation of the screen and then apply the calculated size with font text and widget size. So we can have better results on both layouts.	1 response	
	Bootstrap mobile can help reducing the above issue	1 response	

	Developer must save the instance of the application, that when the orientation changes, it doesn't affect any event	1 response	
	The app orientation should be managed for each feature separately. If a feature does not require orientation change or the user is unlikely to change device orientation for that particular feature, then it should be set fixed.	1 response	
	If orientation changes are required for a feature, then it should be made sure that the UI/UX does not break.	1 response	
	This might be a challenge in some apps features like when displaying a table that has many columns that we can't show it on portrait mode so we had to shift screen to landscape mode.	1 response	
	If you want to manually handle orientation changes in your app you must declare the "orientation", "screen Size", and "screen Layout" values in the android: configChanges attributes. You can declare multiple configuration values in the attribute by separating them with a pipe | character.	1 response	
Diversity of mobile surroundings	Use hybrid platforms/ implement hybrid frameworks	75 responses	V.H: 4 or 5.5 % H: 65 or 79.2% M: 8 or 11% L: 3 or 4.1% V.L: 2 or 2.7%
	We should start using cross-platform in that case so we need to do coding in one language and run apps on multiple platforms like we can use Flutter, React Native, Ionic, Xamarin, etc	1 response	
	Testing through emulators	1 response	
	A unified development platform needs to be implemented to overcome this issue.	1 response	
	If it is required to target a single platform e.g. Android or iOS, then the diversity does not affect the developer.	1 response	

	If multiple platforms are being targeted, then it is needed to understand the components of each platform to provide a consistent user experience across all platforms.	1 response	
	We have now hybrid platforms to tackle such challenges. We have React-native, Flutter, and Ionic that allow a developer to create one app that will run on both iOS and Android OS.	1 response	
	Cross-platform language like reactive and node JS should be used	1 response	
Event driven structure	Use APIs & libraries	63 responses	V.H:4 or 5.5% H: 74 or 78.7% M: 8 or 11% L: 7 or 9.6% V.L: 1 or 1.4%
	In that case, we can't give all the access in a single app. We can have multiple applications, for example, we have careem for clients and careem captain for drivers and on the other hand, we can create their admin panels on the web so we can manage everything from their admin panel by creating a bridge between apps and admin panel.	1 response	
	Each event must be given a priority number and handled accordingly	1 response	
	Trained and skillful developers must be there	29 responses	
Complex contextual features	Use hybrid platforms	57 responses	V.H: 5 or 7.1% H: 70 or 79.5% M: 7 or 10% L: 6 or 8.6% V.L: 0 or 0%
	In that case, Firebase provides us the facility where we can implement notifications and other services easily. And the solution for sensor dependency we should	1 response	

	select the development environment where we can do these things easily.		
	Specific professionals with strong skills need to be hired.	29 response	
	We can tackle such challenges by working on hybrid platforms like React-native, Flutter, and Ionic if these platforms don't give requirement OS compatibility issues. But if the requirement can't be done on hybrid platforms then we need to hire an android developer and iOS developers	1 response	
Frequent changing requirements	Implement agile methodologies	74 responses	V.H: 7 or 9.7% H: 11 or 15.3% M: 76 or 78.3% L: 2 or 2.8% V.L: 1 or 1.4%
	Flow the scrum through proper UAT	19 responses	
	In that case, we should use flexible architecture where we can refactor things easily as we can have MVVM, MVC design pattern.	1 response	
	Current and future requirements need to be identified at the start of the project design. Moreover, proper software development should be applied to make applications scalable and easy to extend to cater any unforeseen requirements of the future.	1 response	
	It should be made sure during the requirements gathering phase that all the needs of stakeholders are understood correctly. Spending more time on better requirements engineering can reduce frequent changes during development.	1 response	
	Before development get started, the app must go through with design first, when user-approved all the designs then the development should start, later few things can be changed over time.	1 response	
Lack of appropriate	Re-allocate resources/ re-define resource allotment	68 responses	V.H: 7 or 9.9% H: 8 or 11.3%

resource allocation	PM involvement	21 responses	M: 72 or 79.1% L: 3 or 4.2% V.L: 1 or 1.4%
	This is the problem of project management. All tasks may be allocated reasonable resources to complete the task within the forecasted time. Project managers should design and investigate project progress to the micro level to assess the milestone achievement. This will help to procure more resources if initial estimates were not correct.	1 response	
	That's where the project manager's job comes, he should acknowledge HR or C.T.O that we need this number of resources for app development. And later they can move in or out developers based on remaining tasks.	1 response	
Platform incompatibility	Use Hybrid platforms	83 responses	V.H: 7 or 9.9% H: 67 or 77% M: 7 or 9.9% L: 5 or 7% V.L: 1 or 1.4%
	A project manager must understand the app requirements and then decide whether to build that app on Hybrid platforms or Native platforms. If any requirement only works on the Native platform then a development team must be hired that knows how to work on the Native platform and vice versa.	1 response	
	Used Up to date technology	1 response	
	Teams must develop a solid architectural plan and should go for multi-platforms	1 response	
	Use cross-platform development languages	1 response	
Lack of resource optimization	Re-allocate the resources/ re-define resources Allotment	56 responses	V.H: 5 or 7.1% H: 8 or 11.4% M: 2 or 2.9% L: 72 or 81.8% V.L: 1 or 1.4%
	Optimize the operational elements	1 response	
	Involve your PM/ conduct meetings with PM	29 responses	
	We should take only those resources on board with experience, based on the complexity of the app. If an app is not complex like the idea is to build an app that	1 response	

	is similar to the market apps then we should hire resources with average experience. But if the app is complex or has innovative ideas or requirements then we must take those resources on board with high experience.		
	Use optimized website which can use minimum Resources	1 response	
Energy inefficiency	execute energy-consuming tasks on cloud/ transfer heavy tasks on cloud/execute heavy tasks on cloud/ transfer energy consuming tasks to cloud/ migrate heavy tasks and execute on the cloud migrate your heavy tasks on cloud/ Outsource the computation-intensive tasks to the cloud	83 responses	V.H: 3 or 4.3% H: 10 or 14.5% M: 7 or 10.1% L: 64 or 75.2% V.L: 0 or 0%
	This is a claim that has to be evaluated through proper research. Training for energy-efficient mobile computation needs to be conducted. Academia can play a major role to develop the skills of graduates by adopting advanced level courses in mobile application development.	1 response	
	Sometimes backend lacks to provide the computation of intensive tasks or sometimes the device OS. The backend team must create a structure for such intensive tasks so that they won't consume more energy on devices or overload the frontend side.	1 response	
Inaccurate task scheduling	Implement task scheduling algorithms/ use task scheduling algorithms/ by using task scheduling algorithms	61 responses	V.H: 6 or 8.7% H: 70 or 76% M: 5 or 7.2% L: 10 or 14.7% V.L: 1 or 1.4%
	Schedule your tasks properly	1 response	
	Scrum meetings must be conducted	23 responses	

	Task sheet would be maintained with the help of any tool, so we would easily check about the resources scheduling	1 response	
	A research and development department has to be initiated in each company. The specialized solution to an estimation of scheduling in a mobile environment needs to be adopted. It would also be interesting to understand the performance of traditional estimation methods in mobile environments.	1 response	
	This challenge can be tackled by proper R&D and scrum meetings. So everyone must know information related to their tasks. Hence, they can provide a fair estimate/ conduct scrum meetings and use task scheduling algorithms that suit your requirements	2 responses	
	Team lead/Project Manager should be experienced to assess this/ Project management responsibility	2 responses	
	A proper project plan should be developed and followed	1 response	
Limited resources/resources lacking	Use best coding techniques/ best coding techniques and practices/ implement best coding guides and Frameworks	65 responses	V.H: 3 or 4.4% H: 8 or 11.8% M: 7 or 10.3% L: 66 or 77.6% V.L: 1 or 1.5%
	PM involvement/ involve your project managers/ meeting with project manager/ project manager role	19 responses	
	Use an optimized website with minimum resources	1 response	
Lack of computational resources	Use external memory chipsets/ embed extra memory silicon chips/ use external memory	79 responses	V.H: 2 or 2.9% H: 64 or 79% M: 8 or 11.*% L: 6 or 8.8% V.L: 1 or 1.5%
	We have now devices that provide the best performance chipsets and memory, still can't compete with the conventional PC but does the job easily for high intensive tasks.	1 response	

	Use optimized website which can use minimum resources in terms of hardware requirements, content, and other services	1 response	
Incorrect estimation of battery life	Embed energy consumption mechanisms to mobile Devices	73 responses	V.H: 3 or 4.4% H: 63 or 78% M: 4 or 5.9% L: 9 or 13.2% V.L: 1 or 1.5%
	Include the feature that can track phone battery	1 response	
	mobile device developers are responsible to integrate such mechanisms to handle such issues	1 response	
	Don't compromise on battery/ battery quality high	2 responses	
	If your task is related to the battery you can resolve it by checking the estimated time of battery in your application if it is not suitable you can give an alert message to the user to connect their mobile to charge before performing a task.	1 response	
	It's not the mobile developer's responsibility to estimate the battery life of a mobile. But in some scenarios, the app required a feature to display phone battery or track phone battery so we took battery life from the device.	1 response	
	With fast charging support and long battery life, this issue is not that alarming until there is a bug	1 response	
Lack of accurate quantification about the consumption of energy by the app	Implement standard guides & practice/ This is a research challenge and it would remain an open problem. A Research and Development team should generate guidelines for developers for overcoming this issue.	77 responses	V.H: 2 or 2.4% H: 3 or 3.6% M: 76 or % L: 1 or 1.2% V.L: 0 or 0%
	measure the app energy first	1 response	
	Include a feature that automatically tracks or predict the energy used by the app	1 response	
	allow notification or automatic prediction of energy consumed by app	1 response	

	can make algorithms to pre estimation	1 response	
	With fast charging support and long battery life, this the issue is not that alarming until there is a bug	1 response	
Compatibility across various OS versions	use of cross platforms/ cross-platform usage/ by using cross platforms	76 responses	V.H: 4 or 5.7% H: 65 or 80.2% M: 7 or 10% L: 4 or 5.7% V.L: 1 or 1.4%
	Programmed according to different O.S. systems.	1 response	
	Tested on the targeted device and targeted OS versions	1 response	
	Google provides a good platform to test and optimize its apps. However, more popular android versions may be tested for efficient operation of apps.	1 response	
	Proper testing on different OS versions to be done before releasing the app	1 response	
	Compatibility testing	1 response	
Lack of asynchrony retrofitting	Use proper defined async tasks/ use of advanced technologies and platforms with defined async tasks/ by using advanced technologies and platforms with proper defined async tasks/ use of latest platforms having defined async tasks	63 responses	V.H: 6 or 8.7% H: 7 or 10.1% M: 76 or 79.1% L: 6 or 8.7% V.L: 1 or 1.4%
	Use of advanced technology like react-native	2 responses	
	Train your mobile developers, conduct workshops and educate them to become domain specialists/ train your mobile developers and make them skillful and domain experts/ train your mobile developers so that they can properly utilize async constructs/	27 responses	
	Hire well-experienced and expert mobile developers/ hire expert mobile developers	3 responses	
	This is a developer weakness and it must be dealt with proper training in the field or educational institutes.	1 response	
Tampering during offloading data	security and encryption mechanisms/ embed security and encryption mechanism/ implement security and encryption mechanisms	66 responses	V.H: 78 or 80.4% H: 9 or 13.2% M: 5 or 7.4% L: 4 or 5.9%
	Tested through different security protocols via different OS versions	1 responses	

	Encryption techniques or tokens are used	27 responses	V.L: 1 or 1.5%
	Secure and trusted cloud platforms should be selected.	1 response	
	The data engineers or backend developers make the data stored securely by using tokens or encrypting them. Many people don't want their info like credit card info in the wrong hands. So we use secured methods like stripe to implement a secure payment.	1 response	
	Use Owasp coding practices	1 response	
Limitation of profilers	Efficient servers/ use properly managed servers/ use efficient servers/ efficient servers should be used for query execution and fast response/ Use of properly managed servers to execute each query efficiently	67 responses	V.H: 5 or 7.5% H: 77 or 84.6% M: 3 or 4.5% L: 5 or 7.5% V.L: 1 or 1.5%
	New offloading mechanisms need to develop to overcome these issues. This problem will remain to be a challenge for a while. However, R&D can develop improved algorithms.	23 responses	
	Performances testing required	1 response	
Lack of identification of risky actions and vulnerabilities	Risk management techniques/ implement RM techniques/ risk management techniques should be implemented/ RM techniques must be followed/ follow RM techniques	79 responses	V.H: 6 or 8.8% H: 5 or 7.4% M: 64 or 79% L: 4 or 5.9% V.L: 2 or 2.9%
	Only apps downloaded from certified, or trusted platforms should be allowed to communicate by the OS	1 response	
	Security testing	1 response	
Inconsistent and inefficient testing	Release beta version of your product and take customer feedback/ beta version product	71 responses	V.H: 4 or 5.9% H: 66 or 82.5%

	Tested through higher-order test cases, set the proper mobile test plan and implement V&V model through manual and automated testing	1 response	M: 7 or 7.3% L: 2 or 2.9% V.L: 1 or 1.5%
	This is a known constraint and with the improvement in developing methodologies and technology, it will be reduced/ improving the development methods	5 responses	
	hire a good SQA resource	1 response	
	End to end testing	1 response	
Lack of configuration of mobile hybrid apps	Review code/ Code review approach/ By reviewing the code/ Review your code thoroughly	78 responses	V.H: 67 or 80.7% H: 8 or 11.9% M: 2 or 3% L: 4 or 6% V.L: 2 or 3%
	Mobile development needs to develop proper SE processes to identify defects and bugs by rigorous testing before launch. The urgency to release products without proper testing is the main culprit.	1 response	
	Revert code	1 response	
	This challenge can be tackled by code review. The senior frontend developer reviews the code with every code pushed to the cloud.	1 response	
	A thorough test plan should be developed to ensure secure cross-platform app testing	1 response	
	Configuration testing	1 response	
Unclear requirements for app functionality towards privacy threats	Security and privacy mechanisms/ Implement security and privacy mechanisms	83 responses	V.H: 8 or 11.9% H: 73 or 81.1% M: 3 or 4.5% L: 5 or 7.5% V.L: 1 or 1.5%
	Set the privacy work scheme and implement V&V model for test the privacy work scheme	1 response	
	Data security mechanisms should be followed Properly	3 responses	
	It is always difficult to handle privacy challenges. Mobile development is not so different. Rigorous implementation of privacy policies needs to	1 response	

	incorporate at the OS level. Moreover, users need to educate in terms of these policies and their implications.		
	We need to gather privacy policies from the user. And list some important ones based on app requirements.	1 response	
	Secure coding practices	1 response	
Patching for updation, correction or improvement	Use of API can fix your patches or patching issues/ Use APIs/ Patch Management	73 responses	V.H: 3 or 4.5% H: 68 or 80.9% M: 2 or 3% L: 8 or 11.9% V.L: 3 or 4.5%
	This challenge can be tackled by using third-party apps or libraries that provide support and fix issues, you can track it by watching them on Git Hub.	11 responses	
Data erasure	Embed encryption mechanisms/ Encryption techniques/ Use encryption techniques and algorithms	74 responses	V.H: 5 or 7.6% H: 7 or 10.6% M: 3 or 4.5% L: 64 or 79% V.L: 2 or 3%
	Yes, this is a major issue faced in the performance of android. However, the problem is only solvable through standardization and adaption of unified a platform for Android OS development.	3 responses	
	API testing	1 response	
Unawareness of needs	Meetings with stakeholders/ Conduct meetings with stakeholders/ Involve your Stakeholders	81 responses	V.H: 5 or 7.5% H: 8 or 11.9% M: 77 or 81% L: 5 or 7.5% V.L: 0 or 0%
	We need to gather information for the app requirements before start development.	11 responses	
	With new development platforms and tools, this the challenge is slowly reducing	3 responses	
Cultural and language barrier	Use of official language/ Communicate using official Language	65 responses	V.H: 66 or 78.5% H: 2 or 3% M: 6 or 9%
	Software ethics should be followed for the development of an app	2 responses	

	Cultural diversity understanding/ Must understand the diversity of culture	17 responses	L: 8 or 11.3% V.L: 2 or 3%
Lack of domain knowledge	Training sessions to mobile developers/ Train your mobile developers/ Conduct training for hired developers	67 responses	V.H: 4 or 6% H: 73 or 84.8% M: 3 or 4.5% L: 4 or 6% V.L: 2 or 3%
	Project manager experience plays an important role. He knows what is the best approach to follow to make the app efficient and user friendly/ Involvement of PM	9 responses	
	Domain knowledge would be given first	3 responses	
	Consult domain expert before implementing task/ Do involve domain experts or allow training session to train your mobile developers	7 responses	
Ambiguities among stakeholders	Meetings with stakeholders/ Conduct meetings with stakeholders/ Frequent meetings with stakeholders	73 responses	V.H: 6 or 8.8% H: 7 or 10.3% M: 66 or 79.5% L: 3 or 4.4% V.L: 1 or 1.5%
	Business analytics communicates with stakeholders and tries to gather their views and solve their ambiguities by providing the best solutions.	1 response	
	Documentation/ Once the User Specification the document is finalized, it has to be followed	9 responses	
Intragroup conflicts	Involve your project manager/ PM involvement/ Meetings with PM	87 responses	V.H: 6 or 8.8% H: 8 or 11.8% M: 80 or 82.4% L: 3 or 4.4% V.L: 0 or 0%
	Set the proper UAT for resolving intragroup conflicts	1 response	
	Follow the decision and instructions of the project Managers	1 response	
	All stakeholders of the system should be on board while deciding on some important issue	6 responses	
	A project manager does meeting with senior developers and other developers. And provide the best approach for each person.	2 responses	

Inefficient response time	Use async programming concepts/ Apply async programming concepts/ Implement async programming concepts	80 responses	V.H: 6 or 9% H: 7 or 10.4% M: 67 or 80.7% L: 3 or 4.5% V.L: 0 or 0%
	That's why we use asynchronous tasks so that we can wait for the execution to work properly and then shows the correct data. We can optimize code by doing multi-lines of code into shortcodes.	1 response	
	Performance testing tools	1 response	
	Team lead/Project manager involvement	1 response	
Lack of requirement task efficiency and responsiveness	Asynchronous programming concepts implementation/ Use of async programming concepts	75 responses	V.H: 4 or 5.9% H: 8 or 11.8% M: 66 or 77.6% L: 7 or 10.3% V.L: 0 or 0%
	Responsive design testing, performance testing	7 responses	
	We use promises and asynchronous awaits to have a proper response from the servers or internal executions to avoid memory leaks & responsiveness issues.	6 responses	
Lack of development standards and practices knowledge	Latest and updated document of development platforms must be used/	67 responses	V.H: 6 or 8.8% H: 78 or 80.4% M: 7 or 10.3% L: 6 or 8.8% V.L: 0 or 0%
	Implement CMMI protocol/ CMMI guide/ Follow the protocols of CMMI/ Protocols of CMMI must be followed to mitigate such a challenge	19 responses	
	Use best coding practices and standards, CMMI Protocol	5 responses	
	Best practices and standards should be followed	2 responses	
	The best approach is to follow the latest documentation of development platforms. We will face enormous issues and lags if we stick to old approaches because with time the development is getting towards more optimized techniques.	3 responses	
	Common standard should be followed for all Employees	1 response	
Testing issues for practitioners	QA team involvement/ QA team/ Involve QA team/ Conduct meeting with QA team	71 responses	V.H: 69 or 79.3%

	Implement V& V model	1 response	H: 8 or 11.8% M: 5 or 7.4% L: 5 or 7.4% V.L: 0 or 0%
	The Quality Assurance team test the app in every aspect. They have tools like selenium where the test the app and find out bugs and security issues. And then they can address developers to fix them.	9 responses	
	Proper test plan should be developed before the start of development process	6 responses	

APPENDIX J

Table J: Summary of Challenges with Most Occurring Solution Strategies

Category	Respective Challenge	# of Strategies	# of Responses	Most Occurring Solution from Survey Findings
Communication-related	Anonymous Communication	7	84	Integrate Encryption –Decryption Mechanisms
	Lack of requirement effective articulation	12	83	Conduct Stand-up or Daily Scrum meetings
	Lack of Verbal & Presentation skills	11	90	Implementation of UML
	Lack of Communication participation	3	82	Construct Models and Diagrams/ Communicate via Models
Requirement-related	Incomplete requirement gathering	8	88	Conduct Interviews and Questionnaires
	Lack of accurate requirement prioritization	11	83	Apply / Follow Story Board Approach
	Unstable requirements	7	83	Daily Scrum Meetings must be conducted
	Change of user needs & understanding	10	90	Take Customer Feedback
	Requirement over scoping	7	82	Re-consider the requirements & re-define the scope

	Inefficient requirement completion time	9	80	Follow Work Breakdown Structure
	Lack of consideration of user & applications requirements for offloading decision making	6	92	Conduct face-to-face meetings between customers/ clients/users and developers
	Incorrect requirement partitioning	6	83	Implement Scrum
	Lack of useful information extraction	6	82	By embedding and deploying automated sensors
	Changing orientation of app	9	89	Hire UI domain specialists
	Diversity of mobile surroundings	8	82	Use hybrid platforms/ Implement hybrid framework
	Event-driven structure	4	94	Use APIs and libraries
	Complex contextual features	4	88	Use hybrid platforms
	Frequent changing requirements	6	97	Implement agile methodologies
Resource-related	Lack of appropriate resource allocation	4	91	Re-define the resources allotment

	Platform incompatibility	5	87	Use hybrid platforms
	Lack of resource optimization	5	88	Re-allocate the resources
	Energy inefficiency	3	85	Outsource the computation-intensive task on cloud for execution
	Inaccurate task scheduling	8	92	Implement task scheduling algorithms
	Limited resources/ resources Lacking	3	85	Implement best coding practices, techniques, and frameworks
	Lacking of / Limited/ computational resources	3	81	Embed external memory chipsets
	Incorrect estimation of battery life	7	80	Embed energy consumption mechanisms to mobile devices
	Lack of accurate quantification about the consumption of energy by the app	6	82	Implement standard guides and practices
	Compatibility across various OS versions	6	81	Use of cross platforms
Security & Privacy-related	Lack of asynchrony retrofitting	5	96	Use advanced technologies and platforms with properly defined async tasks
	Tampering during offloading data	6	97	Embed security and encryption mechanisms
	Limitation of profilers	3	91	Use efficient servers for query execution and fast response

	Lack of identification of risky actions and vulnerabilities	3	81	Implement risk management techniques
	Inconsistent and inefficient testing	6	80	Release beta version of your product for customer feedback
	Lack of configuration of mobile hybrid apps	6	83	Review your code thoroughly
	Unclear requirements for app functionality towards privacy threats	6	90	Implement security and privacy mechanisms
	Patching for updation, correction or improvement	2	84	Use of APIs can fix the patches
	Data erasure	3	81	Embed Encryption Techniques and algorithms
Stakeholder-related	Unawareness of needs	3	95	Conduct meetings with Stakeholders
	Cultural and language barrier	3	84	Use of Official Language
	Lack of domain knowledge	4	86	Conduct Training Sessions for Hired Developers
	Ambiguities among stakeholders	3	83	Conduct meetings with Stakeholders
	Intragroup conflicts	5	97	Involve the Project Manager

	Inefficient response time	4	83	Apply Asynchronous Programming concepts
	Lack of requirement task efficiency and responsiveness	3	85	Implement Asynchronous Programming concepts
	Lack of development standards and practices knowledge	6	87	The latest and updated document of development platforms must be used
	Testing issues for practitioners	4	87	Involve the Quality Assurance Team

APPENDIX K

Table K: Final List of Challenges after Experts' Opinion Along with Categorization

Sr. No.	Paper ID	Sub-Category Name	Respective Challenge(s)
1	X1	Communication	Anonymous Communication
2	X4		Lack of requirement effective articulation
3	X4		Lack of Verbal & Presentation skills
4	X4		Lack of Communication participation
5	X4	Requirement	Incomplete requirement gathering
6	X4		Lack of accurate requirement prioritization
7	X4		Unstable requirements
8	X4		Change of user needs & understanding
9	X4		Requirement over scoping
10	X8, A5		Inefficient requirement completion time
11	S1, S15, W2		Lack of consideration of user & applications requirements for offloading decision making
12	S2, S14, W1		Incorrect requirement partitioning
13	S4		Lack of useful information extraction
14	W15		Changing the orientation of an app
15	A1		Diversity of mobile surroundings
16	A11		Event-driven structure
17	A11		Complex contextual features
18	A26		Frequent changing requirements
19	X2, X9, W10	Resource	Lack of appropriate resource allocation
20	X3, S11, S13, W6		Platform incompatibility
21	X8, A5		Lack of resource optimization
22	X8, A5		Energy inefficiency
23	W4		Inaccurate task scheduling

24	W5		Limited resources/ resources lacking
25	A1		Lacking of / Limited computational resources
26	A3		Incorrect estimation of battery life
27	A12, A13		Lack of accurate quantification about the consumption of energy by the app
28	A24, A27		Compatibility across various OS versions
29	X10	Security & Privacy	Lack of asynchrony retrofitting
30	S7		Tampering during offloading data
31	S8		Limitation of profilers
32	W14, A2		Lack of identification of risky actions and vulnerabilities
33	A8		Inconsistent and inefficient testing
34	A10		Lack of configuration of mobile hybrid apps
35	A16, A17		Unclear requirements for app functionality towards privacy Threats
36	A22		Patching for updation, correction, or improvement
37	A28		Data erasure
38	X4	Stakeholder	Unawareness of needs
39	X4		Cultural and language barrier
40	X4		Lack of domain knowledge
41	X4		Ambiguities among stakeholders
42	X4		Intragroup conflicts
43	X8, A5		Inefficient response time
44	S5		Lack of requirement task efficiency and responsiveness
45	S9		Lack of development standards and practices knowledge
46	S10		Testing issues for practitioners

APPENDIX L:

Table L: Showing the Included Primary Studies along with IDs

Sr. No:	Paper ID:	Name of Primary Study
1	**X1**	A framework for anonymous routing in delay tolerant networks
2	**X2**	A Framework for Cooperative Resource Management in Mobile Cloud Computing
3	**X3**	A Model-Driven Approach to Generate Mobile Applications for Multiple Platforms
4	**X4**	A Systematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development
5	**X5**	Radio and Computing Resource Allocation for Minimizing Total Processing Completion Time in Mobile Edge Computing
6	**X6**	Security-Aware Resource Allocation for Mobile Cloud Computing Systems
7	**X7**	Study and Refactoring of Android Asynchronous Programming (T)
8	**S1**	A Context-sensitive offloading system using machine-learning
9	**S2**	A fast hybrid multi-site computation offloading for mobile cloud computing
10	**S3**	A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things
11	**S4**	A novel pre-cache schema for a high-performance Android system
12	**S5**	A self-protecting agent's based model for high-performance mobile-cloud computing
13	**S6**	A smartphone perspective on computation offloading—A survey
14	**S7**	A standard for developing secure mobile applications
15	**S8**	A systematic mapping study of mobile application testing techniques
16	**S9**	A web-based, offline-able, and personalized runtime environment for executing applications on mobile devices
17	**S10**	An empirical analysis of energy consumption of cross-platform frameworks for mobile Development
18	**S11**	An energy-efficient algorithm for multi-site application partitioning in MCC
19	**S12**	Automatic offloading of mobile applications into the cloud by means of genetic Programming
20	**W1**	A genetic-based decision algorithm for multisite computation offloading in mobile cloud Computing

21	**W2**	A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective
22	**W3**	AGILE: A terminal energy-efficient scheduling method in mobile cloud computing
23	**W4**	An automated model-based approach for unit-level performance test generation of mobile Applications
24	**W5**	An evaluation framework for cross-platform mobile application development tools
25	**W6**	Joint computation offloading and resource provisioning for edge-cloud computing environment: A machine learning-based approach
26	**W7**	VAnDroid: A framework for vulnerability analysis of Android applications using a model-driven reverse engineering technique
27	**W8**	Why does the orientation change mess up my Android application? From GUI failures to code faults
28	**A1**	A cloud based software testing paradigm for mobile applications
29	**A2**	A sealant for inter-app security holes in android
30	**A3**	Accurate prediction of available battery time for mobile applications
31	**A4**	An online algorithm for task offloading in heterogeneous mobile clouds
32	**A5**	Automating UI tests for mobile applications with formal gesture descriptions
33	**A6**	Cordovaconfig: A tool for mobile hybrid apps' configuration
34	**A7**	CrashScope: A practical tool for automated testing of android applications
35	**A8**	EMaaS: Energy measurements as a service for mobile applications
36	**A9**	Empowering developers to estimate app energy consumption
37	**A10**	GUI-Leak: Tracing Privacy Policy Claims on User Input Data for Android Applications
38	**A11**	Leak-Doctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications
39	**A12**	PatchDroid: Scalable third-party security patches for Android devices
40	**A13**	Taming Android fragmentation: Characterizing and detecting compatibility issues for android apps
41	**A14**	Tracking the software quality of android applications along with their evolution
42	**A15**	Understanding and detecting evolution-induced compatibility issues in android apps
43	**A16**	Why does data deletion fail? A study on deletion flaws and data remanence in android systems