

UNIVERSITY PARTNER



Project and Professionalism (6CS007)

Project Report On-Demand Queue Service App [QueueEase]

Student Id : 2227486

Student Name : Nayan Raj Khanal

Group : L6CG4

Supervisor : Ms. Erin Shakya

Cohort : 8

Submitted on : May 20th, 2024

Title and Declaration sheets

Declaration Sheet

Award Title: BSc(Hons) Computer Science

Declaration Sheet

(Presented in partial fulfillment of the assessment requirements for the above award.)

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work. I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free license to do all or any of those things referred to in section 16(I) of the Copyright Designs and Patents Act 1988. (viz. to copy work, to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make an adaptation of the work).

Student Name: Nayan Raj Khanal

Student ID Number: 2227486

Signature:



Date: 20/05/2024

Acknowledgements

I would like to extend my heartfelt gratitude to all those individuals who guided me throughout this journey. Ms. Erin Shakya my supervisor guided me from the start with her constructive feedbacks, knowledgeable insights and constant support. Mr. Bishal Khadka my reader was also present when I needed guidance and support. Mr. Biraj Dulal my module leader and all the faculty members of Project and Professionalism also have provided me with ideas and views that contributed to the success of this work. Last but not the least my family and friends stood by me and pushed me to reach that extra mile. I sincerely thank all because without you I wouldn't be here.

Abstract

Nobody enjoys going to a bureaucratic office early in the morning to avoid long queues just to spend the entire day standing in the same queue due to some government officials taking bribes or slacking off. The time waste and frustration that follow the day of government tasks make individuals pay ridiculous amounts of money without any scam protection to people who are willing to act as stand-ins. A mobile application developed using Flutter, Node JS and MongoDB and Firebase addresses inefficiencies in Nepal's official processes. Employing SCRUM methodology, the app connects users with verified workers, streamlining and revolutionizing the queue waiting experience. Through technology, security and user-centric design the app emerges as an innovative solution.

Table of Contents

1.	Introduction.....	1
1.1.	Aims.....	1
1.2.	Objectives	2
1.3.	Artefact Description	2
1.3.1.	Functional Decomposition Diagram (FDD):	2
1.3.2.	Legend:.....	3
1.4.	Academic Questions	4
1.5.	Background to the Project.....	5
1.1.	Scope and Limitations	5
1.2.	Report Structure.....	6
2.	Literature Review	7
2.1.	Content.....	7
2.2.	Conclusion:	17
3.	Full details of the Artefact	18
3.1.	Frontend.....	18
3.2.	Backend	18
3.3.	Database.....	19
3.4.	Other Tools	20
3.5.	Methodology.....	21
3.6.	Sub-Systems	21
3.6.1.	User Account Management subsystem (UAMSS)	22
3.6.2.	Verification and Security Subsystem (VSSS)	41
3.6.3.	Booking Subsystem (BSS)	55

3.6.4. Communication Subsystem (ComSS)	70
3.6.5. Payment Subsystem (PSS)	82
4. Conclusions	93
5. Critical Evaluation.....	94
6. Evidence of Project Management	95
6.1. Logs:	95
6.2. Full-Detail Gantt Chart.....	109
7. References and Bibliography.....	109
8. Appendices.....	112

Table of Figures

Figure 1. Functional Decomposition Diagram	2
Figure 2. Structure of ride-sharing application (Mishra, 2019).....	8
Figure 3. The Expert Review Development Process (Mohamed, et al., 2019)	10
Figure 4. Proposed model of risk-benefit-trust for m-payment (Park, et al., 2019)	11
Figure 5. MQTT mode. (Wu, et al., 2022).....	12
Figure 6. Overview of the mobile-augmented QMS (Soman, et al., 2020)	13
Figure 7. Proposed Research model for LBS (Kulkarni, et al., 2021).....	14
Figure 8. Proposed workflow architecture for EDQN (Chaudhry, et al., 2024)	15
Figure 9. Interface design flow chart (Jiao, 2022)	16
Figure 10. Block Diagram.....	20
Figure 11. Gantt Chart (Major Milestone)	21
Figure 12. User Registration Activity Diagram.....	24
Figure 13. User Authentication Activity Diagram	25
Figure 14. Profile Management Activity Diagram	26
Figure 15. User Registration Wireframe	27
Figure 16. User Authentication Wireframe	28
Figure 17. Profile Management Wireframe.....	29
Figure 18. User Account Management Subsystem Use-Case	30
Figure 19. User Account Management Subsystem ERD.....	30
Figure 20. User Account Management Subsystem Class Diagram.....	31
Figure 21. User Account Management Subsystem Sequence Diagram.....	32
Figure 22. Test Case 1 (UAMSS) Proof	34
Figure 23. Test Case 2 (UAMSS) Proof 1	36
Figure 24. Test Case 2 (UAMSS) Proof 2	37
Figure 25. Test Case 3 (UAMSS) Proof 1	39
Figure 26. Test Case 3 (UAMSS) Proof 2	40
Figure 27. Agent Registration Activity Diagram.....	42
Figure 28. Agent Mode Wireframe	43
Figure 29. Agent Registration Wireframe	44

Figure 30. Verification and Security Subsystem User-Case.....	45
Figure 31. Verification and Security Subsystem ERD	45
Figure 32. Verification and Security Class Diagram	46
Figure 33. Verification and Security Sequence Diagram	47
Figure 34. Test Case 1 (VSSS) Proof 1	49
Figure 35. Test Case 1 (VSSS) Proof 2	50
Figure 36. Test Case 1 (VSSS) Proof 3	51
Figure 37. Test Case 1 (VSSS) Proof 4	52
Figure 38. Test Case 2 (VSSS) Proof 2	54
Figure 39. User Booking Initiation Activity Diagram	56
Figure 40. Agent Notification Activity Diagram	57
Figure 41. Booking Initiation Wireframe	58
Figure 42. Agent Notification Wireframe.....	59
Figure 43. Booking Subsystem Use-Case.....	60
Figure 44. Booking Subsystem ERD	60
Figure 45. Booking Subsystem Class Diagram	61
Figure 46. Booking Subsystem Sequence Diagram.....	62
Figure 47. Test Case 1 (VSSS) Proof	64
Figure 48. Test Case 2 (VSSS) Proof	66
Figure 49. Test Case 3 (BSS) Proof 1	68
Figure 50. Test Case 3 (BSS) Proof 2.....	69
Figure 51. User-Agent Communication	71
Figure 52. Chatbox.....	72
Figure 53. Communication Subsystem Use-Case Diagram	73
Figure 54. Communication Subsystem ERD	73
Figure 55. Communication Subsystem Class Diagram	74
Figure 56 . Communication Subsystem Sequence Diagram.....	75
Figure 57. Test Case 1 (ComSS) Proof.....	77
Figure 58. Test Case 2 (ComSS) Proof.....	79
Figure 59. Test Case 3 (ComSS) Proof.....	81
Figure 60. Payment Activity Diagram	83

Figure 61. Payment Wireframe	84
Figure 62. Payment Subsystem Use-Case Diagram.....	85
Figure 63. Payment Subsystem ERD.....	85
Figure 64. Payment Subsystem Class Diagram.....	86
Figure 65. Payment Subsystem Sequence Diagram.....	87
Figure 66. Test Case 1 (PSS) Proof 1	89
Figure 67. Test Case 1 (PSS) Proof 2.....	90
Figure 68. Test Case 2 (PSS) Proof.....	92
Figure 75. Full-Detail Gantt Chart	109
Figure 76. App Flowchart	112
Figure 77. WBS.....	112

Table of Tables

Table 1. Conclusions of Literature Review	17
Table 2. User Account Management Subsystem SRS	23
Table 3. User Account Management Subsystem Data Dictionary	31
Table 4. Test Case 1 (UAMSS)	34
Table 5. Test Case 2 (UAMSS)	37
Table 6. Test Case 3 (UAMSS)	40
Table 7. Verification and Security Subsystem SRS	41
Table 8. Test Case 1 (VSSS)	52
Table 9. Test Case 2 (VSSS)	54
Table 10. Booking Subsystem SRS	55
Table 11. Booking Subsystem Data Dictionary	61
Table 12. Test Case 1 (BSS)	64
Table 13. Test Case 2 (BSS)	66
Table 14. Test Case 3 (BSS)	69
Table 15. Communication Subsystem SRS	70
Table 16. Communication Subsystem Data Dictionary	74
Table 17. Test Case 1 (ComSS)	77
Table 18. Test Case 2 (ComSS)	79
Table 19. Test Case 3 (ComSS)	81
Table 20. Payment Subsystem SRS	82
Table 21. Payment Subsystem Data Dictionary	86
Table 22. Test Case 1 (PSS)	90
Table 23. Test Case 2 (PSS)	92

1. Introduction

We all can agree that today's world revolves around technology. But still, even after all the accomplishments of our predecessors, there are some areas where the use of technology can significantly improve day-to-day standards. Bureaucratic systems are one such area where there exists a persistent challenge, notably in the context of Nepal. Long queues and wait times due to a lack of work ethics from government workers or various reasons plague individuals seeking to complete official tasks, leading to inefficiencies, wasted time and frustration. Some smart individuals see this as a business opportunity to collect quick cash by standing in the queue and people are willingly paying hefty sums for it.

And enter an "On-Demand Queue Assistance App" a mobile application that streamlines and revolutionizes the queue-waiting experience. The envisioned app aims to connect users with individuals willing to stand in queues on their behalf, providing a practical solution to optimize the cumbersome process of waiting for official tasks. Users can effortlessly hire and communicate with these queue stand-ins, ensuring a more time-efficient and user-friendly approach to navigating bureaucratic procedures.

1.1. Aims

The primary aim of QueueEase is to:

- Develop a mobile application to revolutionize the queue-waiting experience by reducing the time users spend waiting in queues during official processes.
- Create a platform that offers employment opportunities for individuals willing to stand in queues on behalf of others.
- Build a trustworthy platform ensuring the legitimacy of individuals offering queue-standing services to prevent scams and fraudulent activities.

1.2. Objectives

And accompanying these aims are several key objectives such as:

- Allow users to create accounts with necessary details with proper authentication.
- Allow users to register as agents providing necessary details with proper verification.
- Create a booking system that enables users to book agents by selecting the destination and its respective allocated price.
- Provide a secure chat system for communication between users and agents.
- Implement a fair transaction method between users and agents.
- Establish a feedback loop for continuous improvement.

1.3. Artefact Description

1.3.1. Functional Decomposition Diagram (FDD):

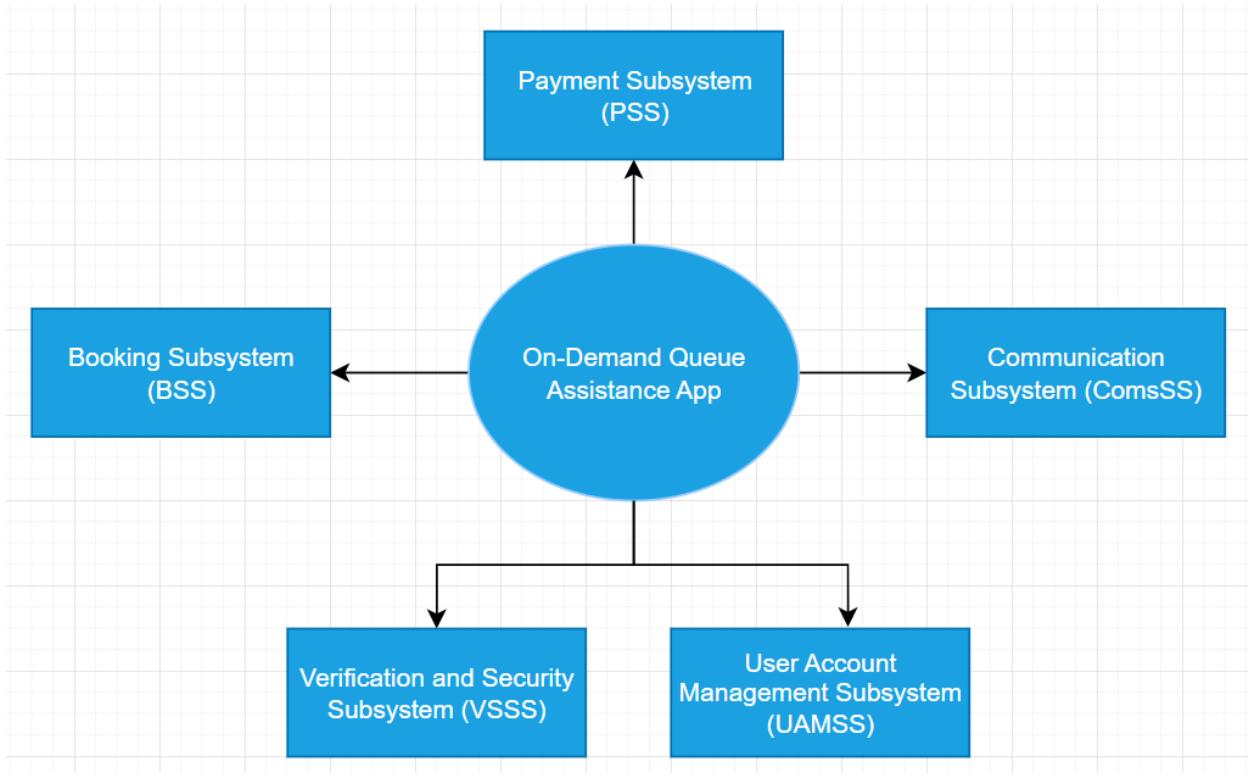
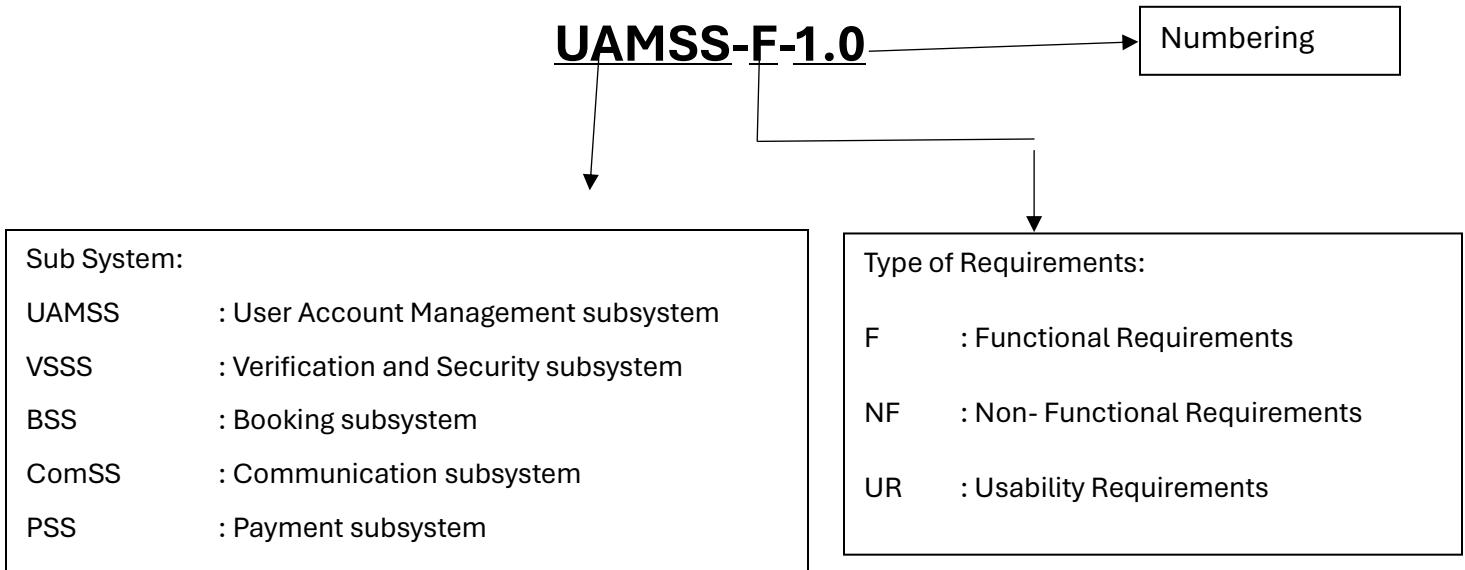


Figure 1. Functional Decomposition Diagram

1.3.2. Legend:



The artefact produced by this project is a mobile application designed to address the mentioned aims and objectives. Developed using Flutter, Node JS, backed by MongoDB and Firebase, the application consists of various sub-systems. These include:

- User Account Management which handles users account creation
- Verification handles verification of users and selection of agents
- Booking handles booking between user and agent
- Communication handles chat system between user and agents
- Payment handles transaction between user and agent.

Each sub-system plays a crucial role in tackling the problem and providing effective solution. GitHub as the version control system, VSCode source code editor, Postman and Thunderclient for API development and testing, Android Studio for emulator and Scrcpy for screen copy were also utilized for the completion of the app.

1.4. Academic Questions

The foundation of this project is based upon extensive academic inquiries that has been undertaken to ensure that this project understands the problem and produces results beneficial to all stakeholders.

- What are the best practices for user authentication in a mobile app?
→ As apps can be vulnerable to many security risks how can QueueEase improve the security to prevent breaches.
- How can the mobile app detect potentially fraudulent activities or scam users?
→ Being an on-demand service app, it requires interaction between strangers which increases the risk of fraudulent activities or scam users, how will QueueEase tackle that problem.
- How does a booking between user and agent take place?
→ For a service to be provided it must be issued by one party and accepted by other. How will QueueEase facilitate this interaction between two parties.
- What technologies can be used to implement real-time chat in a mobile app?
→ In order to make the app accessible and life easier for two parties it is necessary to provide a platform which supports call and chat when faced with issue, how will QueueEase handle chat rooms.
- What technologies can be used to implement payment in a mobile app?
→ The conclusion of any service is payment, how is QueueEase going to provide livelihood to its users.
- How can the system be designed to handle a growing number of users and agents efficiently?
→ The success of an app causes its downfall due to mismanagement. How will QueueEase handle servers, users, location when it grows.
- How to implement user interface to provide a smooth and intuitive experience for both users and agents?
→ The basis of love of the app is in its usability and visuals. How will QueueEase attract users with its UI and UX.

These various academic inquiries revolved around user authentication, user verification technology implementation, scalability and user interface design to make sure this app will be effective.

1.5. Background to the Project

In Nepal, if you ask anyone regarding official processes none of them will have a positive thing to say about it. People go to offices early in the morning to avoid long queue times just to realize that they must stand for the rest of the day due to bribery or sources or lack of work ethics from government workers. Valuable time is wasted which leads to frustration and conflict, which is common in today's state of Nepal. While some are willing to stand in queues on behalf of others, there is currently no organized and secure platform that facilitates this service. The charge for this service is also outrageous ranging anywhere from a couple thousand to multiple thousands of rupees including the risk of potential scams.

The proposed mobile application is developed to tackle common issues such as long queue times, corruption and inefficient processes to revolutionize the queue stand-in experience. The proposed solution is that verified individuals can be hired through the app to stand in queues on behalf of users, leading to potential job opportunities for individuals who are willing to step up as queue stand-ins and minimize the users' frustration, killing two birds with one stone. It comes with a secure verification system that ensures genuine agents, a booking system that allows users to book available agent for the desired location, an integrated secure communication system and a fair payment system. With a feedback loop for continuous improvement from both users and agent, this app aims to leave a positive impact on the citizens and hopefully bring realization to government offices regarding their management and ethics problems.

1.1. Scope and Limitations

The scope of this project extends beyond mere queue-standing assistance, in the future scope of QueueEase are:

- Expand to include additional government offices and locations beyond Kathmandu.

- Increase the types of services offered, expanding beyond queue-standing assistance only in official tasks.
- Collaborate with government entities to enhance the efficiency of official processes or potentially integrate directly with the government.
- Introduce community engagement features, such as forums, encouraging interaction among users and workers.
- Integrate with popular payment wallets, offering users additional payment options.

However, QueueEase also succumbs to limitations such as:

- Initially limited to specific government offices and locations, expansion requires coordination and time.
- Scarcity of individuals willing to stand in queues in certain locations limits the app's effectiveness drastically.
- Heavy reliance on internet connectivity for chat system, leading to difficulty in poor network areas.

1.2. Report Structure

This report will give an overview of the project, beginning with the introduction pointing out the project's aims, objectives, artefact description, background of the project, scopes and limitations. Following the introduction section, we investigate relevant key literatures and review them one by one. After the literature review, we then move onto the main body of the report which contains the full details of the artifacts including the development of the artefact, use of development cycle, justification of tools and technologies and testing. The body also answers the academic questions and provides supporting. Following the main body, the report includes conclusion where the initial aims of the project and academic question are referenced and discusses what has been discovered and what conclusions can be drawn. The report also includes critical evaluation section where the product, report,

software, findings, process-planning, management, quality of sources are discussed and it also contains a section of self-reflection. The final part of report contains evidence of project management including but not limited to logbooks, Gantt charts, commits etc. and references and bibliography.

2. Literature Review

2.1. Content

The key literatures reviewed in this section forms the foundation for understanding the relevant systems such as ride-sharing applications, user authentication model, mobile-payment service, mobile messaging system, queue management system, location based services, fraud detection and prevention using AI system and human-computer interface on user experience all which in turn help developing the subsystems of QueueEase.

The paper “**Risks and opportunities on adoption of ride-sharing applications - an exploratory study of Kathmandu**” talks about the rise of ride-sharing apps like Uber and Lyft as well as local platforms like Pathao, InDriver and Tootle in Nepal. This definitely represents a significant shift in the way people get around in cities. More and more people are slowly dropping bus rides and are opting for their personal rider. These apps create a mutually beneficial relationship between drivers and passengers, with smartphones in hand, internet connectivity alongside simple interface, it has taken the country with storm. The process of signing up for the apps is also user-friendly and straightforward with easy app installations, simple registrations and streamlined verification process. This gives users quick access to a range of ride services at ease. The technology behind these apps is governed by models like the Technology Acceptance Model (TAM) as well as the Theory of Reasoned Action (TRA) and the Theory of Planned Behavior (TPB). These models help us understand the factors that influence user behaviors such as perceived opportunities, behavioral control mechanisms, risk assessments, social influence dynamics and contextual elements. However, no matter the success the growth of ride-sharing apps in Nepal is hampered by regulatory challenges which prevent these services from expanding.

The policymakers need to create a more supportive and friendly environment to promote the growth of these apps. Despite the challenges this research offers a panoramic view of user adoption patterns and lays the foundation for further research to improve the ride-sharing ecosystem. (Mishra, 2019)

The basic structure of many ride-sharing app has been displayed below.

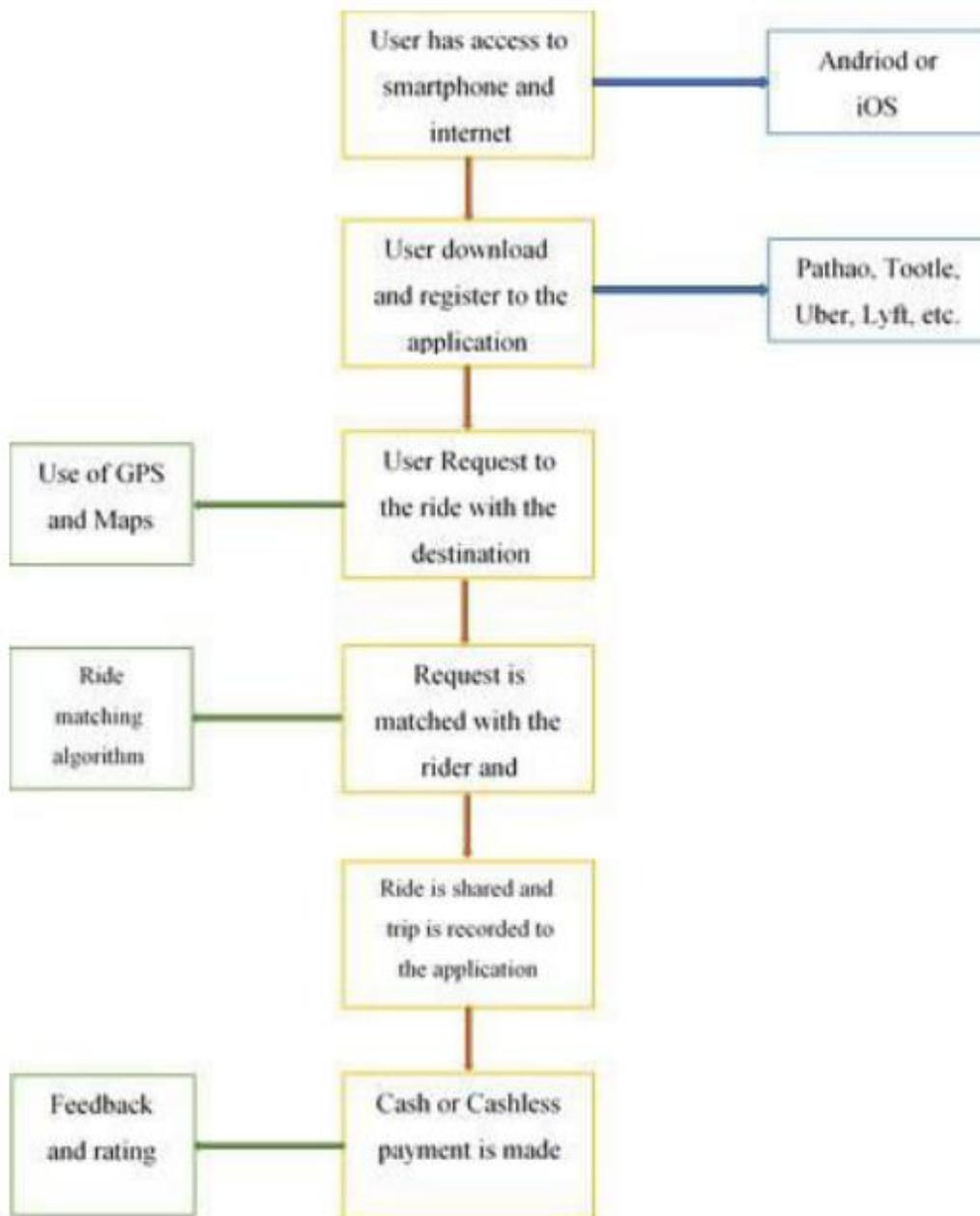


Figure 2. Structure of ride-sharing application (Mishra, 2019)

The paper “**An Improved User Authentication Model for Mobile Application Systems: An Expert Review Verification**” talks about the importance of mobile communication in today's world, it also states how vulnerable mobile communication is to hacking, making mobile apps susceptible to security breaches. Therefore, it is of upmost priority to have strong user authentication models that can protect sensitive information. The paper also talks about a recent study proposing an advanced model that improves on existing frameworks by combining multi-factor authentication, encryption and watermarking techniques. The model also keeps its focus on user acceptance, which is crucial for ensuring that users adopt and use the technology effectively. For the study rigorous methodology was used which involved expert reviews, pilot studies and surveys to evaluate the model's validity and feasibility. A panel of experts with at least three years of experience evaluated the model's feasibility through a well-designed questionnaire. Upon successful examination although the experts agreed that the model was feasible they pointed out some minor flaws which required adjustment. Before real-world application of the model the study pointed out that potential wording ambiguities in the questionnaire and the need for more experts in mobile app-related fields is required. Once these adjustments are made and new pilot study and survey is conducted this will provide further validation for the proposed user authentication model and ensure its success. (Mohamed, et al., 2019)

Below is the devised flow of the expert review development process.

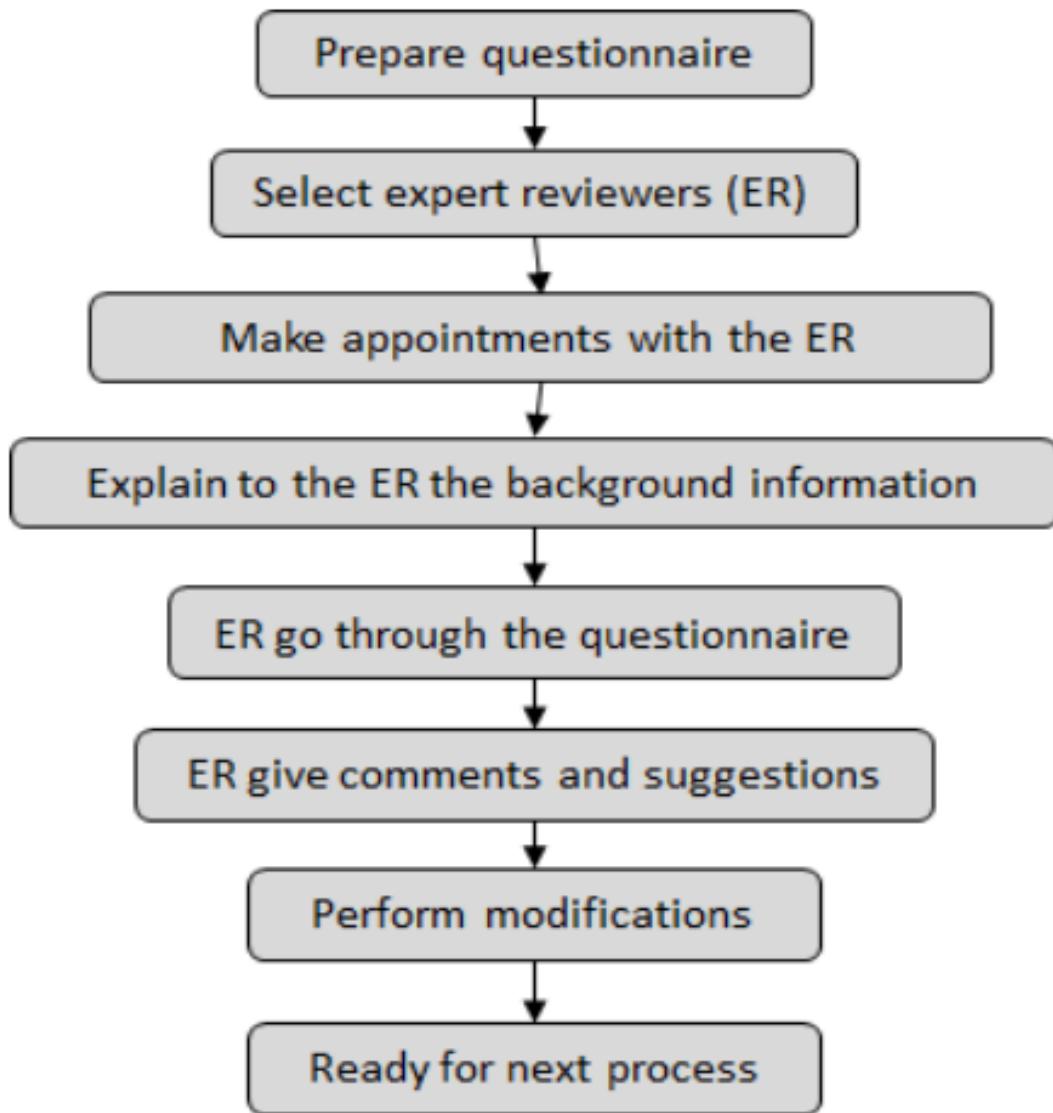


Figure 3. The Expert Review Development Process (Mohamed, et al., 2019)

The paper “**Mobile-payment service: Interplay of perceived risk, benefit, and trust in service adoption**” studied the factors which influences people's decision when it comes to using mobile payments. The factors listed out were perceived risk, perceived benefits, trust and demographic factors such as education. The research showed that trust played a crucial role in determining whether people will use mobile payments or not. On top of that education could also help people understand the relationship between risk and their willingness to use these payment systems. The study has significant implications for companies providing mobile payment services and policymakers. It highlights the need for increased consumer

education and trust-building measures. Risk is a significant barrier that people face which makes them hesitant when it comes to using mobile payments, but building trust can help overcome this. For starters, mobile payment providers must focus on developing trust and credibility in their services. Also, the policymakers can help by implementing educational programs that boost consumer knowledge and understanding of mobile payment functions. However, the research also points out some limitations such as relying on scenarios that may have influenced participants' familiarity with mobile payment functions. The paper concludes by saying in order to increase mobile payment adoption, it is essential to focus on enhancing trust and credibility and future research should prioritize these elements. (Park, et al., 2019)

Below is the proposed model for mobile payment services.

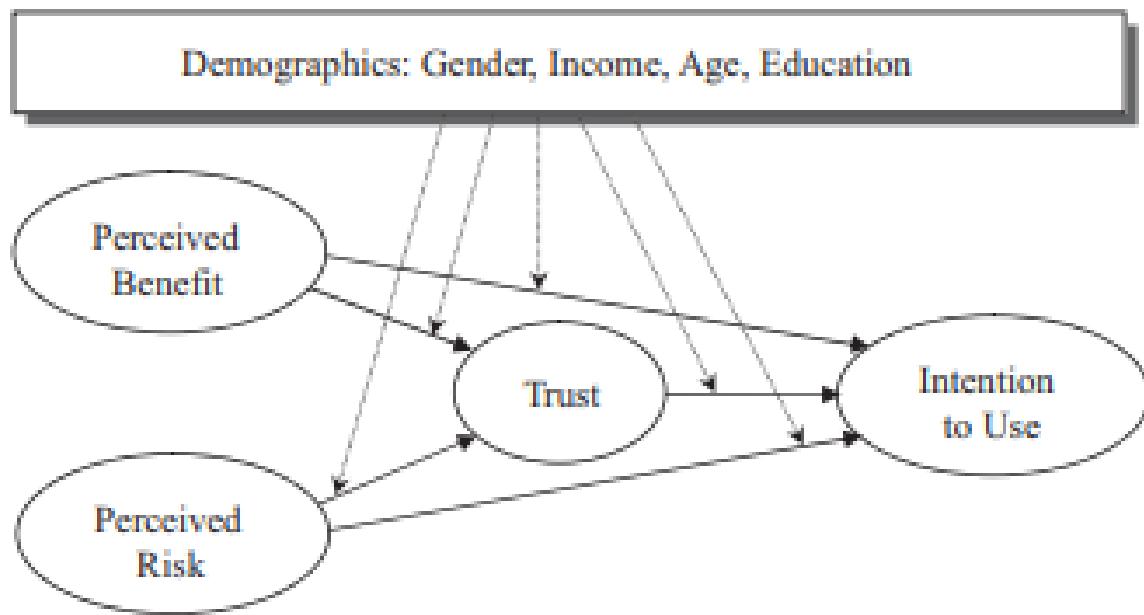


Figure 4. Proposed model of risk-benefit-trust for m-payment (Park, et al., 2019)

The paper “**A Novel Mobile Instant Messaging System**” proposes a new way to improve mobile instant messaging systems. Instead of using the current text-based protocols like IMS/SIP, it suggests the use of MQTT protocol. The reason being that these existing protocols have some limitations mainly when it comes to efficiency over wireless connections. The paper points out the advantages of MQTT has over other protocols like XMPP and SIP/SIMPLE

as MQTT is lightweight, simple and supports various services. The paper also draws the design and talks about the implementation of the MQTT-IM protocol stack which is made specifically for mobile devices with limited resources. The paper's methodology includes a mobile instant messaging system architecture comprising server and client components. The core module being the MQTT Broker and is responsible for message routing and management while the TCP transmission is used in the system architecture. Various performance evaluations were conducted in a controlled test environment, which showed that the MQTT-based system performs better than traditional protocols as it has lower standby power consumption, reduced traffic overhead, higher scalability in terms of concurrent connections and server processing capacity. However, at the end the paper talks about how it does not address the potential security vulnerabilities and scalability challenges of the proposed system. The paper also points out it is not ready to include real-world deployment scenarios and user feedback, which could have been helpful in validating the system's effectiveness in different IoT applications and user environments. (Wu, et al., 2022)

Below is the MQTT model proposed by the paper.

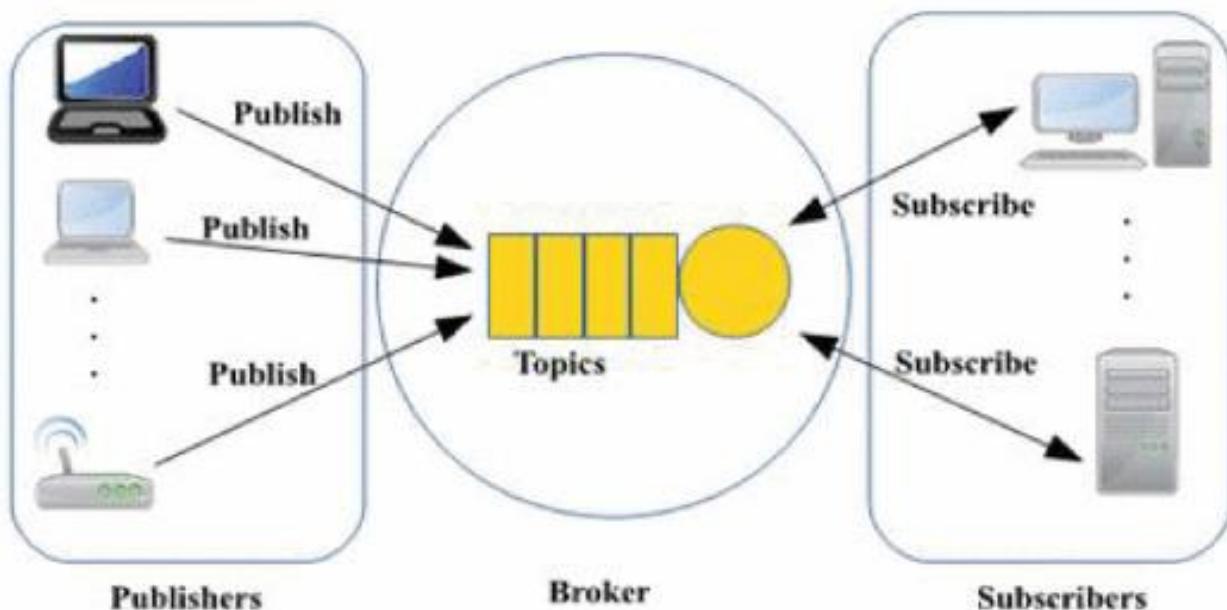


Figure 5. MQTT mode. (Wu, et al., 2022)

The paper “**Mobile-Augmented Smart Queue Management System for Hospitals**” proposes a smart queue management system for hospitals that helps manage high patient volumes effectively. The proposed solution integrates with existing Hospital Management Information Systems (HMIS) to provide an in-depth solution when it comes to managing the patient workflows across various service areas. Patients can use mobile apps or kiosks to generate tokens for different services and smart displays at service counters read that tokens to find out the service required. The proposed system’s dynamic queue generation algorithm helps control crowds during peak hours, which is particularly useful for government medical facilities. On top of that the proposed system supports family-based token generation and smooth queue management between service areas which enhances patient experience and optimizes hospital resources. On testing at a tertiary care hospital it was seen that the system has the potential to improve operational efficiency and patient satisfaction. But the paper points out that further validation and scalability testing across diverse hospital settings are necessary to ensure widespread adoption and effectiveness in real-world scenarios. (Soman, et al., 2020)

Below is the overview of the mobile-augmented QMS.



Figure 6. Overview of the mobile-augmented QMS (Soman, et al., 2020)

The paper "**Modelling Trust to Preserve End-User Privacy in Location Based Services**" tries to uncover the relationship between user trust and privacy concerns when it comes to Location Based Services (LBS). The paper proposes a research model that identifies the key to trust of end user in LBS, acknowledging the importance of end-user say when it comes to their privacy preservation. The model takes inspirations from existing literature on topics such as trust, risk perception and privacy concerns and identifies sources of trust that include end-user attributes, service provider attributes, service attributes and interaction experience. The paper firmly believes that these sources influence dimensions of end-user trust, including capabilities, integrity and goodwill exhibited by LBS and ultimately increasing the perception of trust. The paper's model aims to guide the design and development of user-centric Privacy Preserving Mechanisms (PPMs) while providing practical guidelines for LBS providers to improve end-user privacy perception. By bridging this gap of computational approaches and end-user perceptions the research aims to offer a framework to address the challenge of privacy preservation in LBS. In conclusion it hopes to advance efforts of similar notion and foster a more trustworthy environment when it involves information exchange. (Kulkarni, et al., 2021)

Below is the proposed research model for Location Based Services.

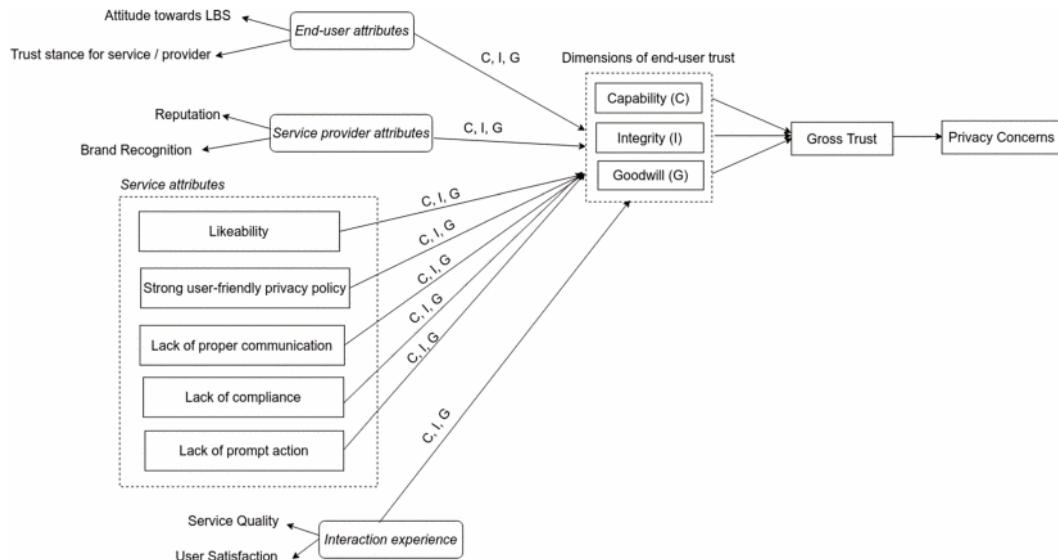


Figure 7. Proposed Research model for LBS (Kulkarni, et al., 2021)

The paper “**Fraud Detection and Prevention for a Secure Financial Future Using Artificial Intelligence**” proposes using advanced AI techniques to take a new approach for detecting and preventing fraud in financial datasets. The system uses innovative methods such as K-Means clustering, Principal Component Analysis (PCA) and an Enhanced Deep Q Network (EDQN). The main aim of proposed framework is to address the challenge of detecting fraudulent patterns in complex financial data. It hopes to achieve that by streamlining the dataset preparation process and identifying clear patterns more efficiently. With the help of PCA the paper aims to reduce dataset complexity while retaining much key information as possible for fraud detection. The integration of EDQN facilitates classification and prediction which enables the system to understand complex relationships in financial data and produce accurate forecasts. Upon testing the proposed approach has demonstrated superior performance in terms of accuracy, precision, recall and F-measure, making it an effective solution for combating financial fraud and safeguarding the integrity of financial systems. (Chaudhry, et al., 2024)

Below is the proposed workflow of the new approach.

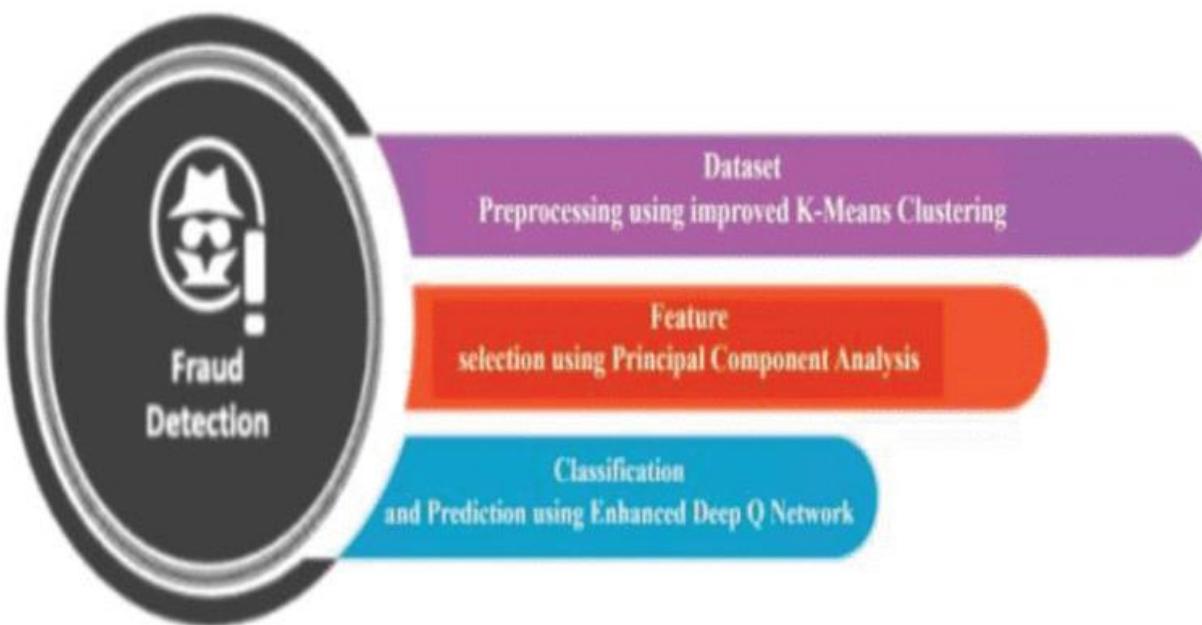


Figure 8. Proposed workflow architecture for EDQN (Chaudhry, et al., 2024)

The paper “**Research and Application of Human-Computer Interface Based on User Experience**” explores the design of human-computer interfaces (HCI) with a focus on user experience (UX). It seeks to create interfaces that prioritizes user satisfaction, functionality and enhance the relationship between users and computer interfaces. It emphasizes on the fact the importance of combining industrial design principles with UX theory is essential to achieve this goal. By adding emotional factors into the product design the paper aims to touch on the emotional side of users in hopes of enhancing market competitiveness of software platforms. The study calls for such HCI designs that not only meet functional requirements but also vibe with users on an emotional level. Through a structured design process alongside user-centered principles and technical research the paper is successful in showing that integrating emotional design into HCI is very effective. This paper paves the way for future advancements in interface design that prioritize user joy, efficiency and satisfaction. (Jiao, 2022)

Below is the basic interface design flow chart utilized in the paper.

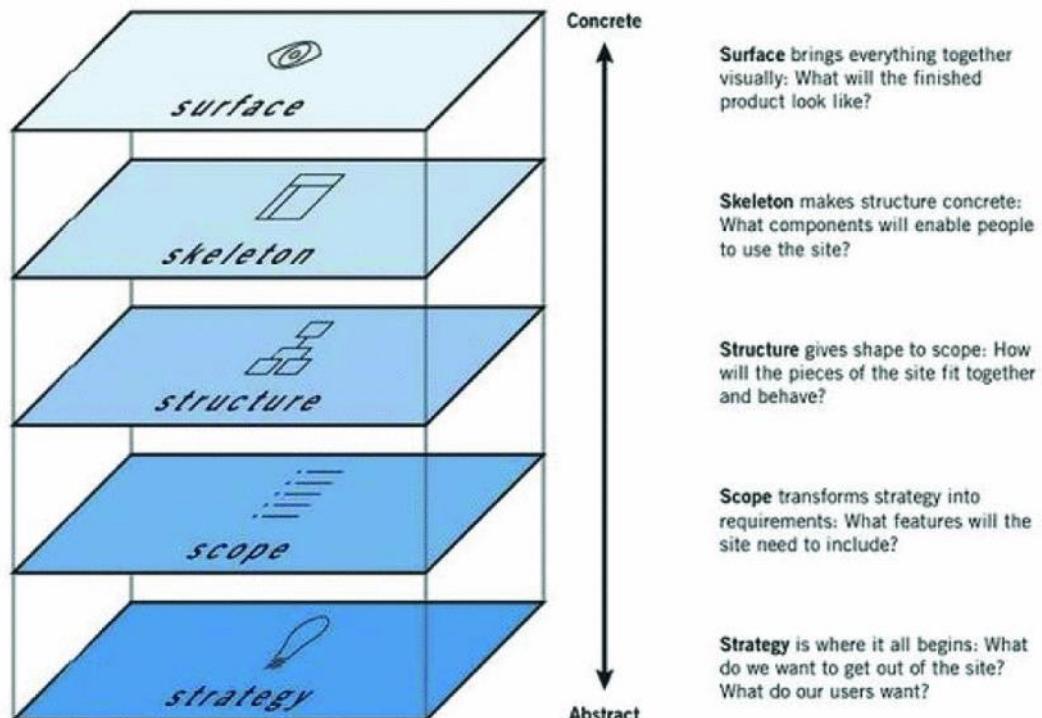


Figure 9. Interface design flow chart (Jiao, 2022)

2.2. Conclusion:

Paper Title	Relevance to QueueEase
Risks and opportunities on adoption of ride-sharing applications	QueueEase should comply with regulations provided by the country used. Also, it should prioritize user-friendly interfaces and understand what the user actually wants.
An Improved User Authentication Model for Mobile Application Systems	In order to improve security multi-factor authentication should be implemented. Also, QueueEase should prioritize security and user-friendliness in authentication.
Mobile-payment service	If QueueEase wants to make users adopt to mobile payment options it should build trust with its service and also educate how to mitigate the perceived risks associated alongside it.
A Novel Mobile Instant Messaging System	Even if it's not directly related to QueueEase researching different ways to implement chat and find out what suits best for the app is important.
Mobile-Augmented Smart Queue Management System	Even if it's not directly related to QueueEase researching different ways queues are managed in different sectors helps in developing efficient service app revolved around customer satisfaction.
Modelling Trust to Preserve End-User Privacy in Location Based Services	In order for QueueEase to make users trust their app and ensure they allow location services to be permitted in their smartphones it is necessary to build trust and ensure transparency to mitigate privacy concerns.
Fraud Detection and Prevention	Even if it's not directly related to QueueEase it is crucial to research ways to implement accurate and efficient fraud detection in the app.
Research and Application of Human-Computer Interface	If QueueEase wants to provide user satisfaction it must incorporate emotional design principles and user-centered approach in its interface design.

Table 1. Conclusions of Literature Review

3. Full details of the Artefact

The proposed artefact for this project will be a mobile app made with the help of Flutter for the front-end, Node JS for the back-end, MongoDB and Firebase for database, GitHub as the version control system, VSCode as the source code editor, Postman and Thunderclient for API development and testing, Android Studio for emulator services and Scrcpy for copying the screen. The app consists of several subsystems.

Key Components:

- Mobile application which is both easy to use and elegant.
- Backend infrastructure supporting the app's functionalities.
- Secure database to hold users' information

3.1. Frontend

Flutter serves as the frontend framework for developing the mobile applications user interface and interactions. Creating visually stunning and responsive UI components, handling user input events and managing stateful widgets are some of its functionalities. Flutter also provides a unique feature called hot reload feature which allows developers to see changes instantly speeding up the development process and refine the user experience.

The decision to use Flutter for this project was driven by several factors. Flutter provides a rich set of pre-built widgets and libraries making the development of complex UI with ease. Moreover, Flutter has a huge backing of developers that are willing to help and teach new developers. Last but not the least familiarity is also the reason.

(altexsoft, 2022)

3.2. Backend

Node JS serves as the backend infrastructure supporting the app's functionalities. The functionalities include providing various APIs (User, Agent, Booking, Communication, Payment) to handle requests and data processing, ensuring smooth communication between the mobile app and the database.

The reason to choose Node JS for this application was mainly because of the vast ecosystem of libraries and modules to choose from which makes it easy to develop and enhance functionality. Also, the community support as it has an active community who are ready to support and provide ample resources for troubleshooting and learning.

(Sufiyan, 2023)

3.3. Database

The database of the mobile application will store essential data related to users, agents, bookings, and communication logs using a NoSQL database.

- User Data: Stores information about registered users, including account details.
- Agent Data: Stores information about agents who are just registered users, including account details and verification.
- Booking Data: Records information about user bookings, such as destination, price and relevant details.
- Communication Logs: Stores logs of communication between users and agents, ensuring a record of interactions for transparency and dispute resolution.

Whereas for the database which will handle the data MongoDB was used as the primary database whereas Firebase was used for communication. A NoSQL database is chosen due to its flexible nature when it comes to handling diverse data types. Also, the schema-less nature allows for easy adaptation, crucial for an app with dynamic requirements.

(Özşahan, 2023) (Taylor, 2024)

Below is the block diagram of the mobile application incorporating all the systems and subsystems:

BLOCK DIAGRAM

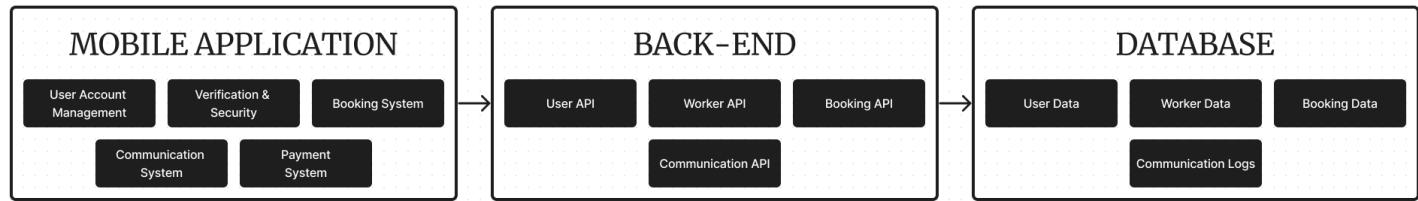


Figure 10. Block Diagram

3.4. Other Tools

Apart from the big three Flutter, NodeJS, MongoDB and Firebase the project utilizes GitHub as version control system, VSCode as the code editor, Postman and Thunder Client as the API platform, Android Studio for the emulator and Scrcpy for copying the screen.

Starting with the version control system GitHub, the main reason to use this system was because of its reputation and community backing. GitHub has been the go to choice for many developers for several years. Another reason was the familiarity with the system. (Juviler, 2024)

VSCode was chosen as the code editor for many of the same reasons. The familiarity, reputation and trust alongside with community backing and various helpful extensions makes it a no brainer choice. (GigoDev, 2023)

Postman and Thunder Client are both API platform. Postman allows building and using of APIs whereas Thunder Client allows for testing through the code base itself. Both were chosen because of simplicity, familiarity and ease of use. They are also trusted and used by many developers. (Demir, 2023) (Vadhineni, 2023)

Android Studio is official Integrated Development Environment (IDE) for Android app development. It provides numerous emulators to choose from to run and test the project on with ease. It also has a huge backing from developers allowing for quick Q&A when in a pickle. (Boog, 2022)

Finally, Scrcpy is an app that allows users to control their phone from their PC. It mirrors the screen connected via cable or TCP/IP. This helps in development and testing of application and is used by many app developers. (Genymobile, 2020)

3.5. Methodology

SCRUM is the most well-suited and practical way to build this mobile app. SCRUM is iterative and adaptable approach. Since the mobile app change constantly and SCRUM lets us adjust on the fly as user needs evolve it the perfect choice. On top of that, the other big plus is that since the work is to be shown to the client and since clients want to see progress, test things out and suggest improvements SCRUM is perfect as it allows the development cycle to be broken down into bite-sized chunks, called sprints. Each sprint delivers a working piece of the app which with a feedback loop keeps the app on track and makes sure that it is building the right thing as desired by the client. (Chandana, 2024)

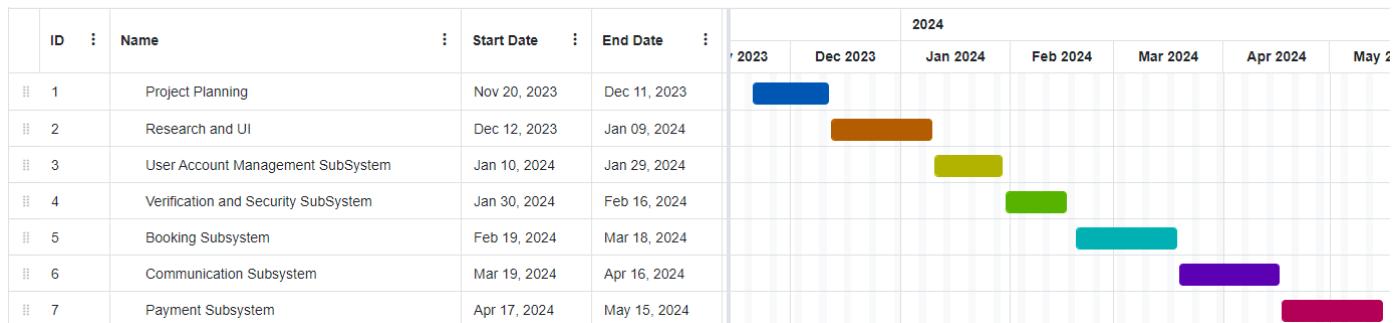


Figure 11. Gantt Chart (Major Milestone)

3.6. Sub-Systems

The mobile application consists of following sub-system:

1. The User Account Management subsystem handles user registration and authentication.
2. The Verification and Security handles users as agent registration and authentication.
3. The Booking subsystem allows users to view destination and its fare and book available agent for that specific government office.

4. The Communication subsystem allows secure and efficient communication between users and workers.
5. Payment subsystem ensures transparent and fair transactions and leverages Khalti.

Explained below are each sub-system with their purpose which explains its responsibility, scope explaining what it includes, table consisting of code, description and MoSCoW prioritization (The acronym MoSCoW represents four categories of initiatives: must-have, should-have, could-have, and won't-have, or will not have right now), activity diagram, wireframes, use-case diagram, ERD, Data Dictionary, Class Diagram and Sequence Diagram. (DevSamurai, 2023) Alongside all the design/modelling diagrams following them are the test cases for each subsystem.

3.6.1. User Account Management subsystem (UAMSS)

3.6.1.1. Purpose:

The User Account Management subsystem is responsible for handling user registration, authentication and profile viewing within the application.

3.6.1.2. Scope:

This subsystem covers the creation and maintenance of user accounts, ensuring secure access to the application.

3.6.1.3. SRS:

Req. Code	Req. Desc	MoSCoW
UAMSS-F-1.0	The system shall allow users to create an account by providing first and last name, unique email, and valid password	Must Have
UAMSS-NF-1.1	The system shall validate the first and last name, unique email, valid password and phone number during the registration process	Should Have
UAMSS-F-2.0	Registered users shall be able to log in with their email and password	Must Have

UAMSS-NF-2.1	The system shall implement secure password storage and encryption techniques	Should Have
UAMSS-NF-2.2	Secure communication (HTTPS) shall be implemented for data transmission	Should Have
UAMSS-F-3.0	Users shall be able to view their profiles	Must Have
UAMSS-NF-3.1	Users shall have the option to log out from their account from the settings menu	Should Have
UAMSS-NF-3.2	Users can add and change profile picture or edit password	Could Have

Table 2. User Account Management Subsystem SRS

3.6.1.4. Activity Diagram:

1. User Registration

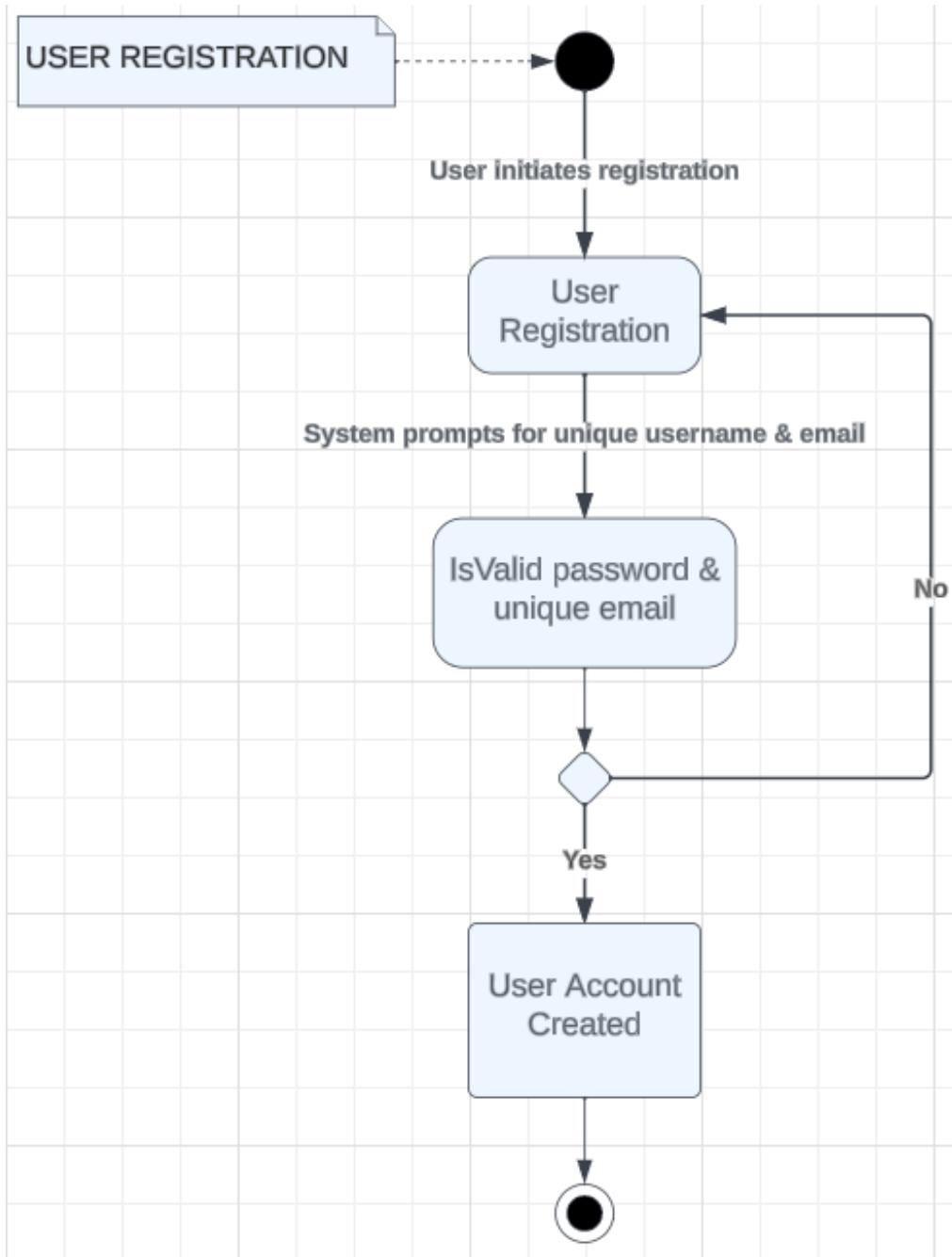


Figure 12. User Registration Activity Diagram

2. User Authentication

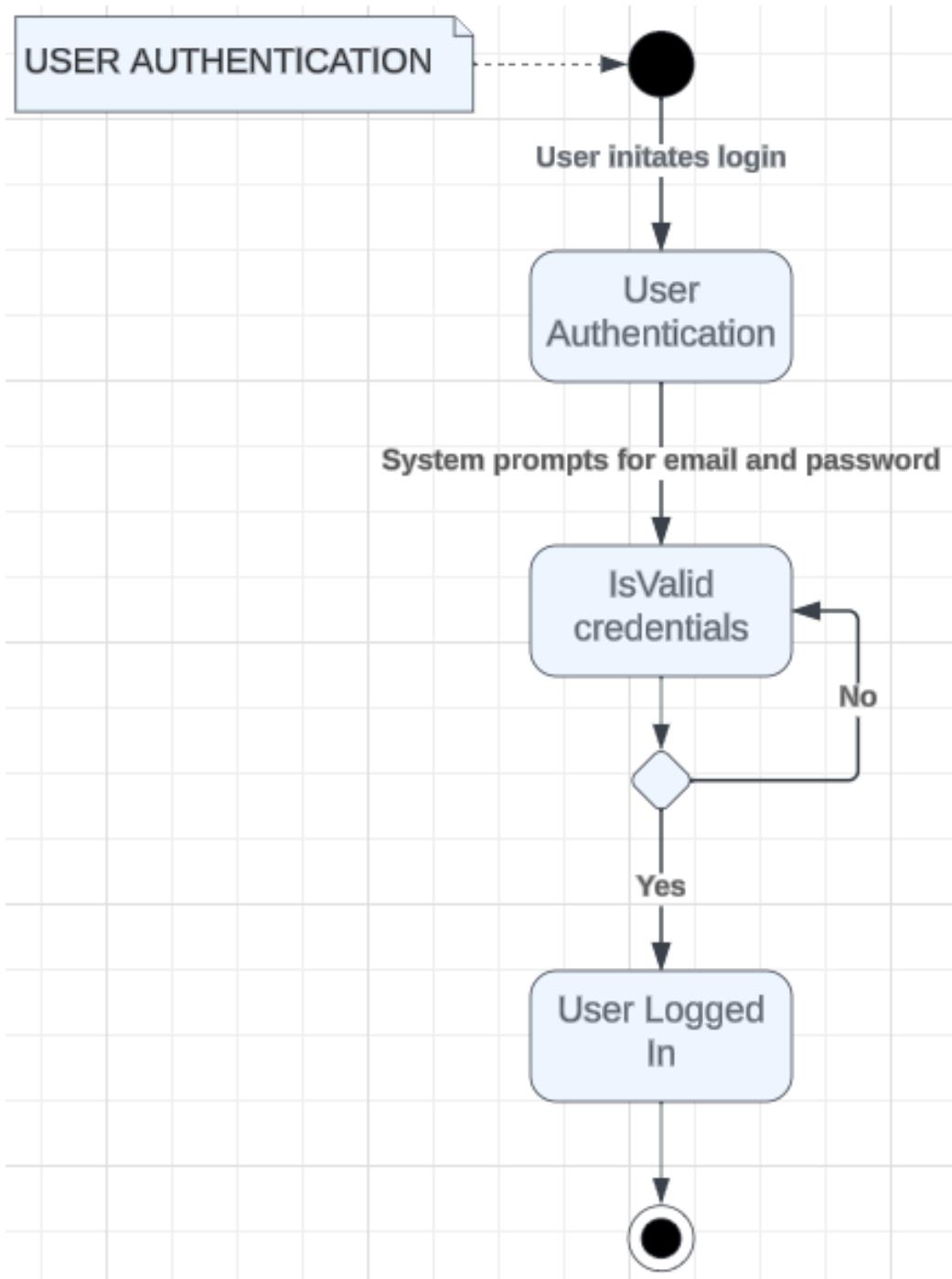


Figure 13. User Authentication Activity Diagram

3. Profile Management

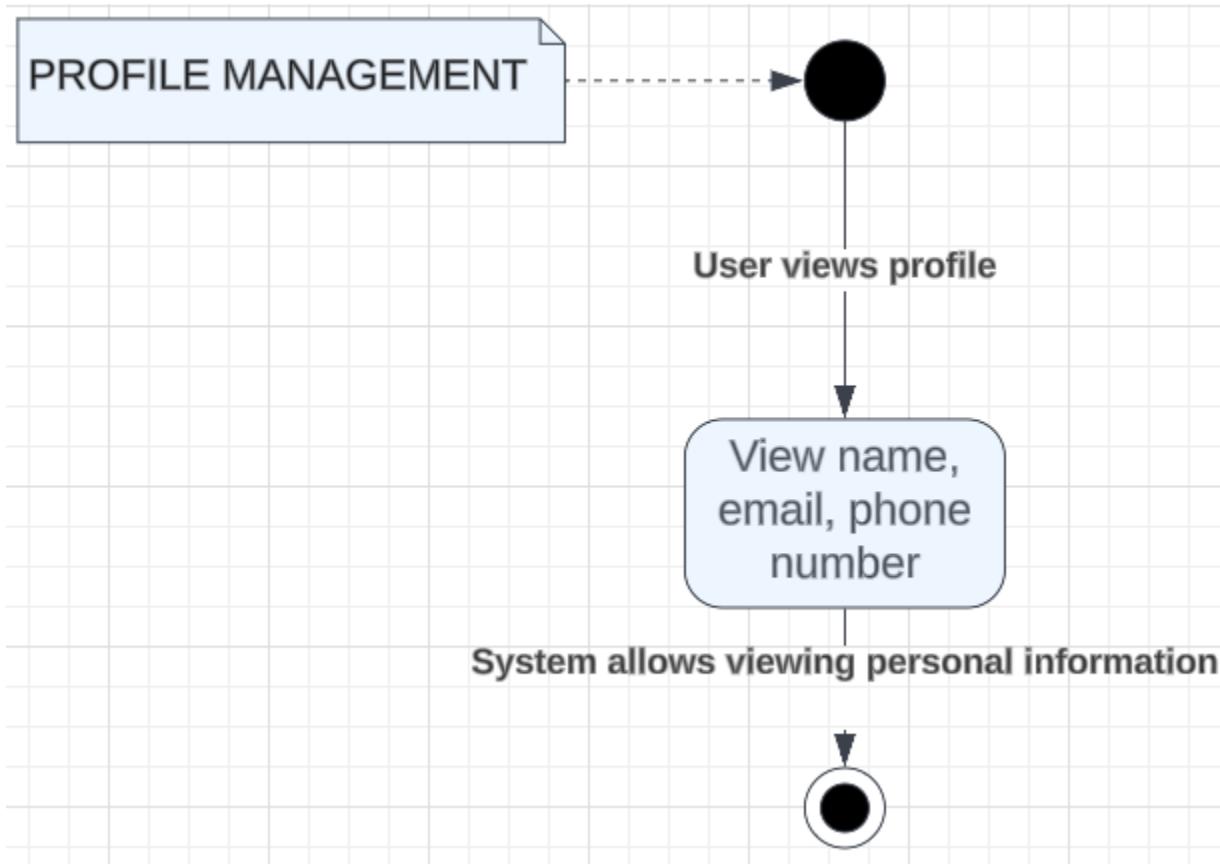
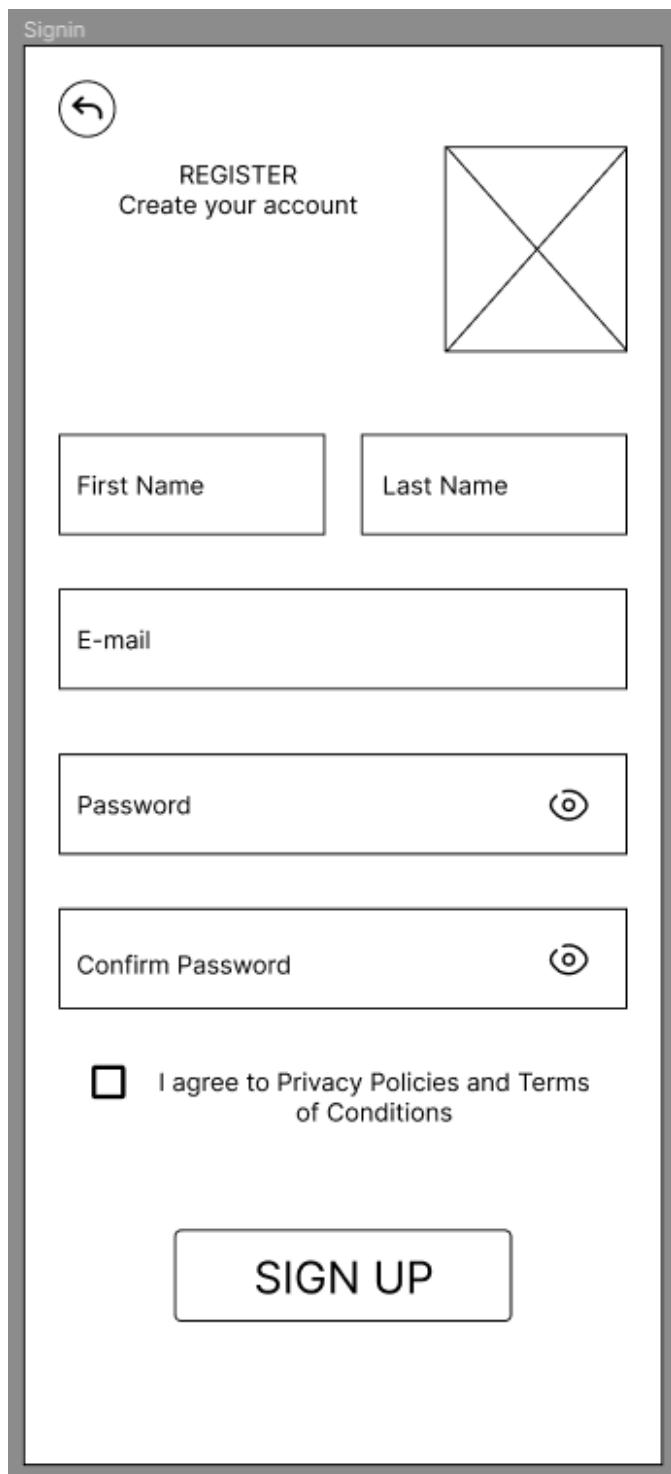


Figure 14. Profile Management Activity Diagram

3.6.1.5. Wireframe:

1. User Registration



The wireframe for the User Registration page is titled "Signin". It features a back arrow icon in the top-left corner. The main title "REGISTER" and subtitle "Create your account" are positioned at the top center. To the right of the text is a large square placeholder with a diagonal cross, likely for a profile picture or logo.

The form consists of several input fields:

- "First Name" and "Last Name" in separate input boxes.
- A large input box for "E-mail".
- A password field with a circular eye icon to its right.
- A confirm password field with a circular eye icon to its right.

Below the input fields is a checkbox labeled "I agree to Privacy Policies and Terms of Conditions".

A prominent "SIGN UP" button is located at the bottom center of the form.

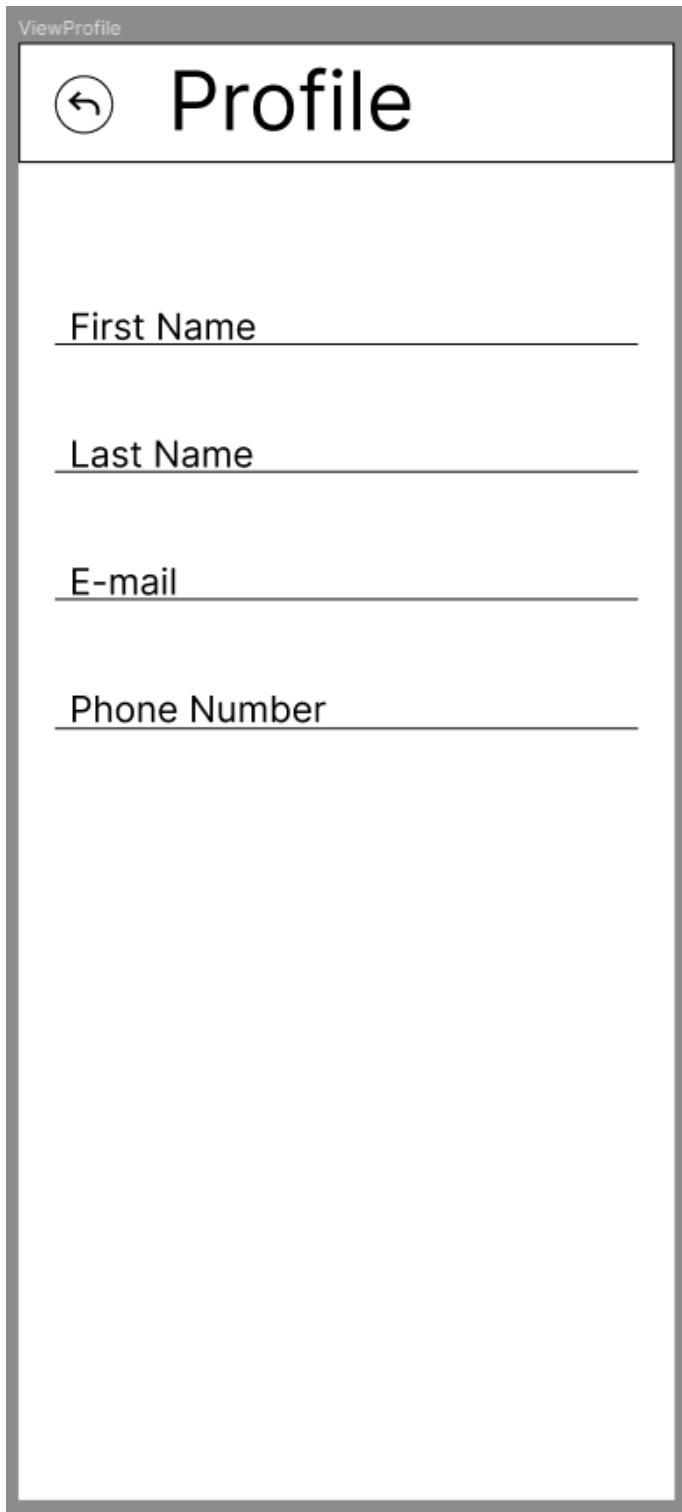
Figure 15. User Registration Wireframe

2. User Authentication



Figure 16. User Authentication Wireframe

3. Profile Management



A wireframe diagram of a mobile application interface titled "ViewProfile". The title bar contains a back arrow icon and the word "Profile". Below the title bar are four text input fields labeled "First Name", "Last Name", "E-mail", and "Phone Number", each preceded by a bold underlined label.

Label	Input Type
First Name	<input type="text"/>
Last Name	<input type="text"/>
E-mail	<input type="text"/>
Phone Number	<input type="text"/>

Figure 17. Profile Management Wireframe

3.6.1.6. Use-Case Diagram:

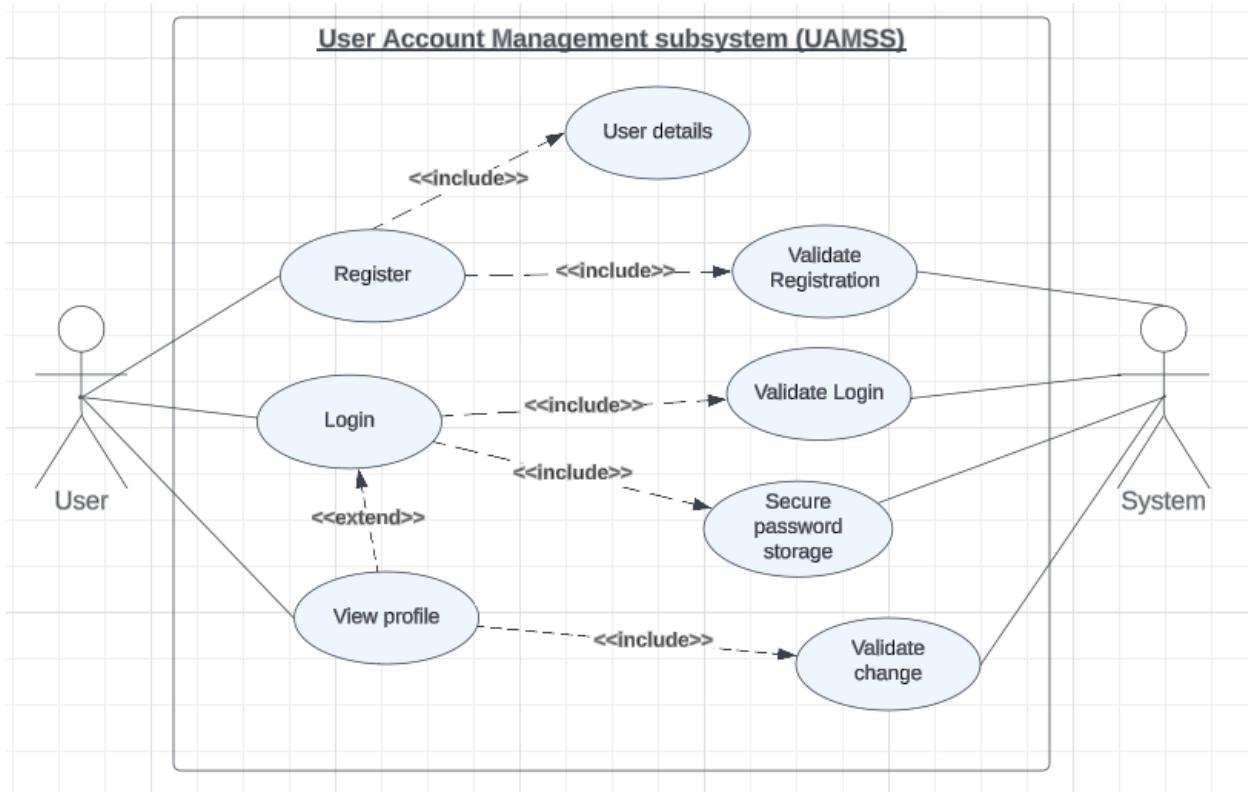


Figure 18. User Account Management Subsystem Use-Case

3.6.1.7. ERD

User Account Management subsystem (UAMSS)

User	
PK	userID
	firstName
	lastName
	email
	password
	phoneNumber
	isAgent

Figure 19. User Account Management Subsystem ERD

3.6.1.8. Data Dictionary:

Field Name	Data Type	Description
userID	String	Unique identifier for each user
firstName	String	User's first name
lastName	String	User's last name
email	String	User's email address
password	String	User's encrypted password
phoneNumber	String	User's phone number
isAgent	Boolean	User's agent status

Table 3. User Account Management Subsystem Data Dictionary

3.6.1.9. Class Diagram:

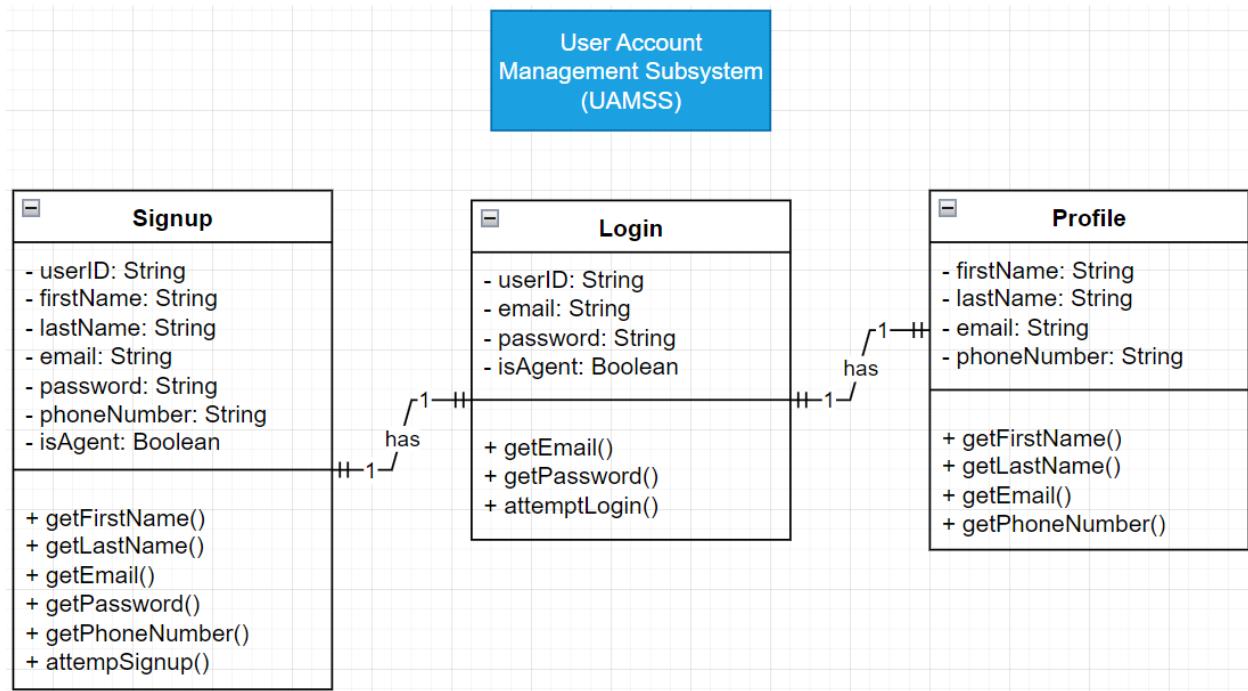


Figure 20. User Account Management Subsystem Class Diagram

3.6.1.10. Sequence Diagram:

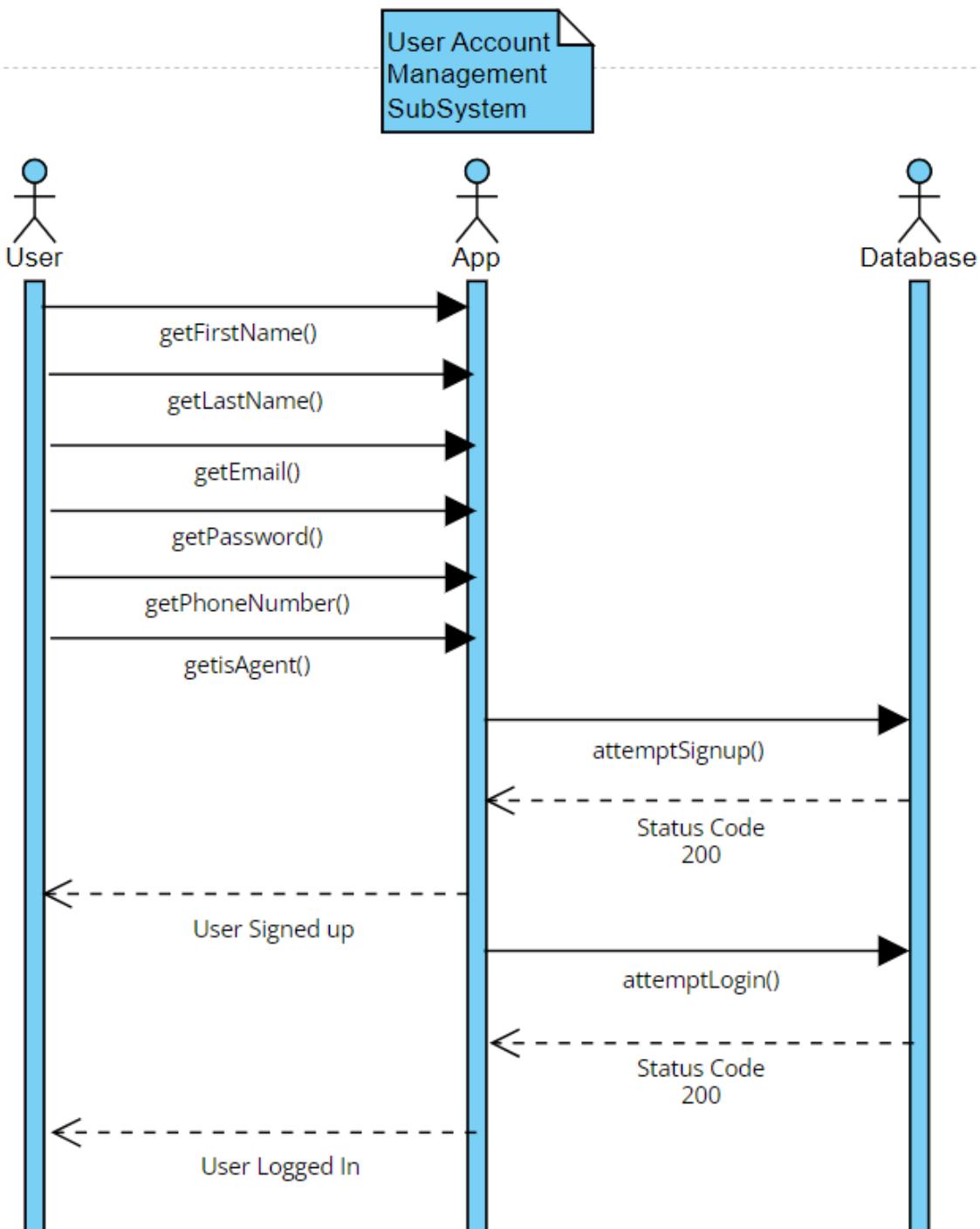


Figure 21. User Account Management Subsystem Sequence Diagram

3.6.1.11. Test Plan

Objective:

Verify the functionality and performance of the User Account Management subsystem.

Scope:

This test plan covers user registration, authentication and profile management.

Test Case 1	
Objective	Validate that the registration form properly validates input fields (first name, last name, email, password).
Action	<ul style="list-style-type: none">• Open app• Click on sign up from welcome screen• Fill details• Click on Create Account
Expected Result	App should not allow empty fields also not allow invalid email address, simple password combination, invalid phone number.
Actual Result	On top of not allowing empty fields the app also did not allow invalid email address, simple password combination, invalid phone number.
RESULT	PASS

PROOF

First Name AH@.COM

Last Name AH@.COM

E-mail AH@.COM

Invalid email address.

Password 123hagsyajsd (eye icon)

Password must contain at least one uppercase letter.

Phone Number 0089675753

Invalid phone number format (10 digits required).

I agree to [Privacy Policy](#) and [Terms of use](#)

Create Account

Table 4. Test Case 1 (UAMSS)

Test Case 2	
Objective	Ensure that users can log in and sign in with valid email and password and combinations and correct name and phone number.
Action	<ul style="list-style-type: none"> • Open app • Click on sign up from welcome screen • Fill details • Click on Create Account • Click on Log in from welcome screen • Fill details • Click on Log In
Expected Result	Upon entering correct details the app should allow user to create account and log in with that account.
Actual Result	The app did allow creation of valid account and logging in of that same account.
RESULT	PASS

PROOF



REGISTER

Create your account



First Name



Last Name



E-mail



Password



Phone Number



I agree to [Privacy Policy](#) and [Terms of use](#)

Create Account

Signup successful

Figure 23. Test Case 2 (UAMSS) Proof 1

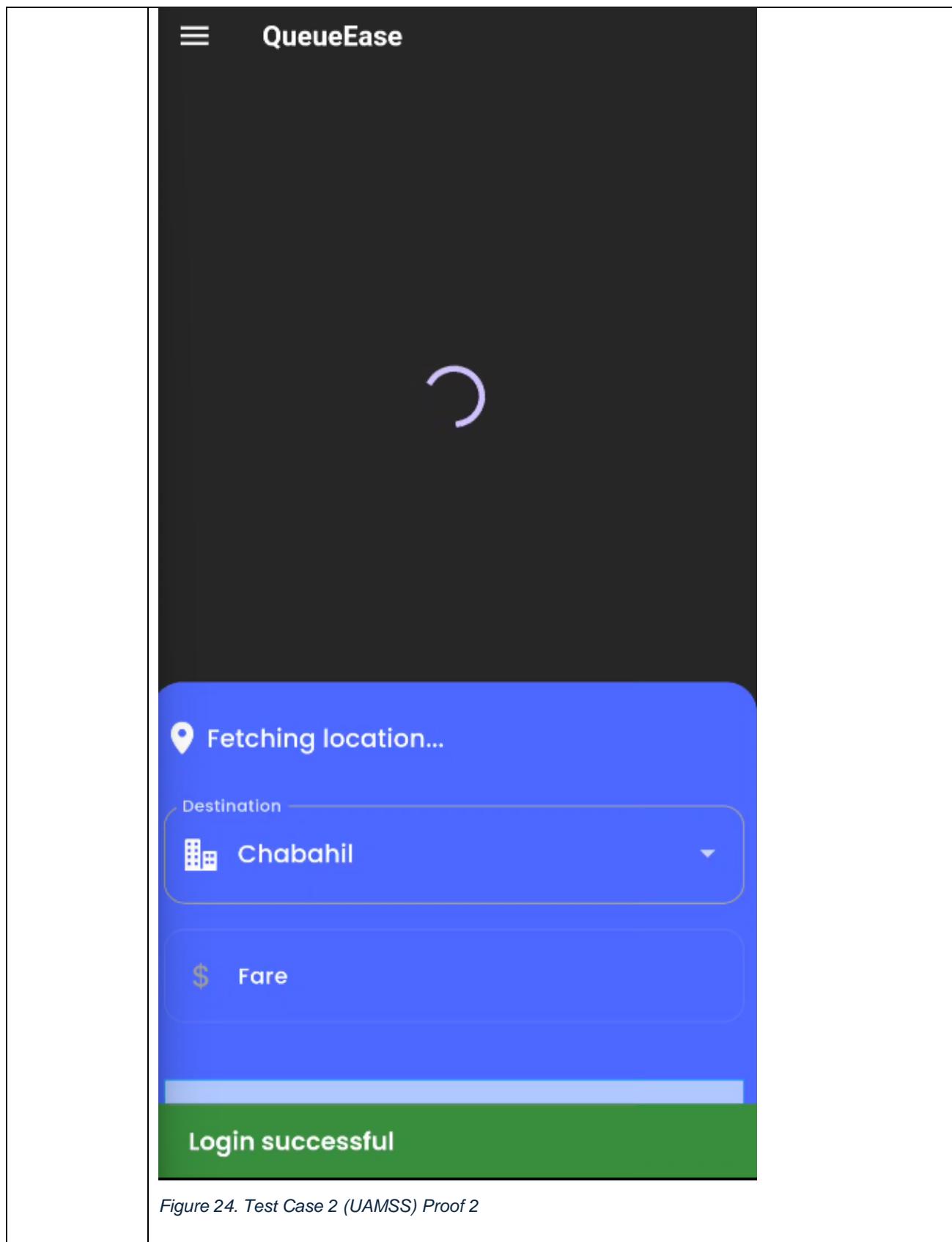


Figure 24. Test Case 2 (UAMSS) Proof 2

Table 5. Test Case 2 (UAMSS)

Test Case 3	
Objective	Ensure that users can view their profile details and log out from the session.
Action	<ul style="list-style-type: none"> • Open app • Log in • Click on drawer menu • Select profile • Click on Log Out
Expected Result	Upon navigating to the profile page, users should be able to see their details and when clicking log out button they must be logged out.
Actual Result	The profile page showed the users profile with their photo, name, email and number and when got logged out when clicked on logout button.
RESULT	PASS

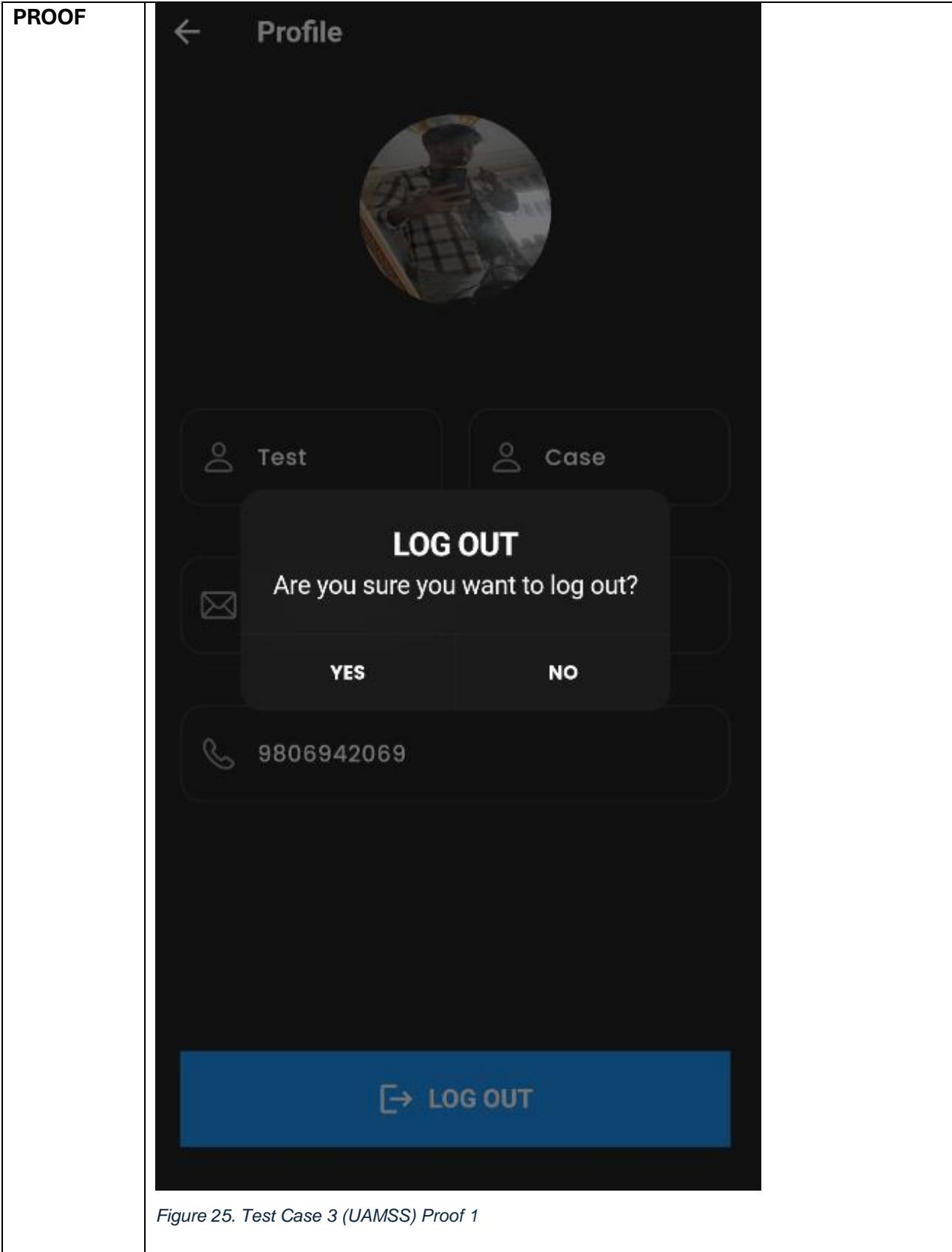


Figure 25. Test Case 3 (UAMSS) Proof 1

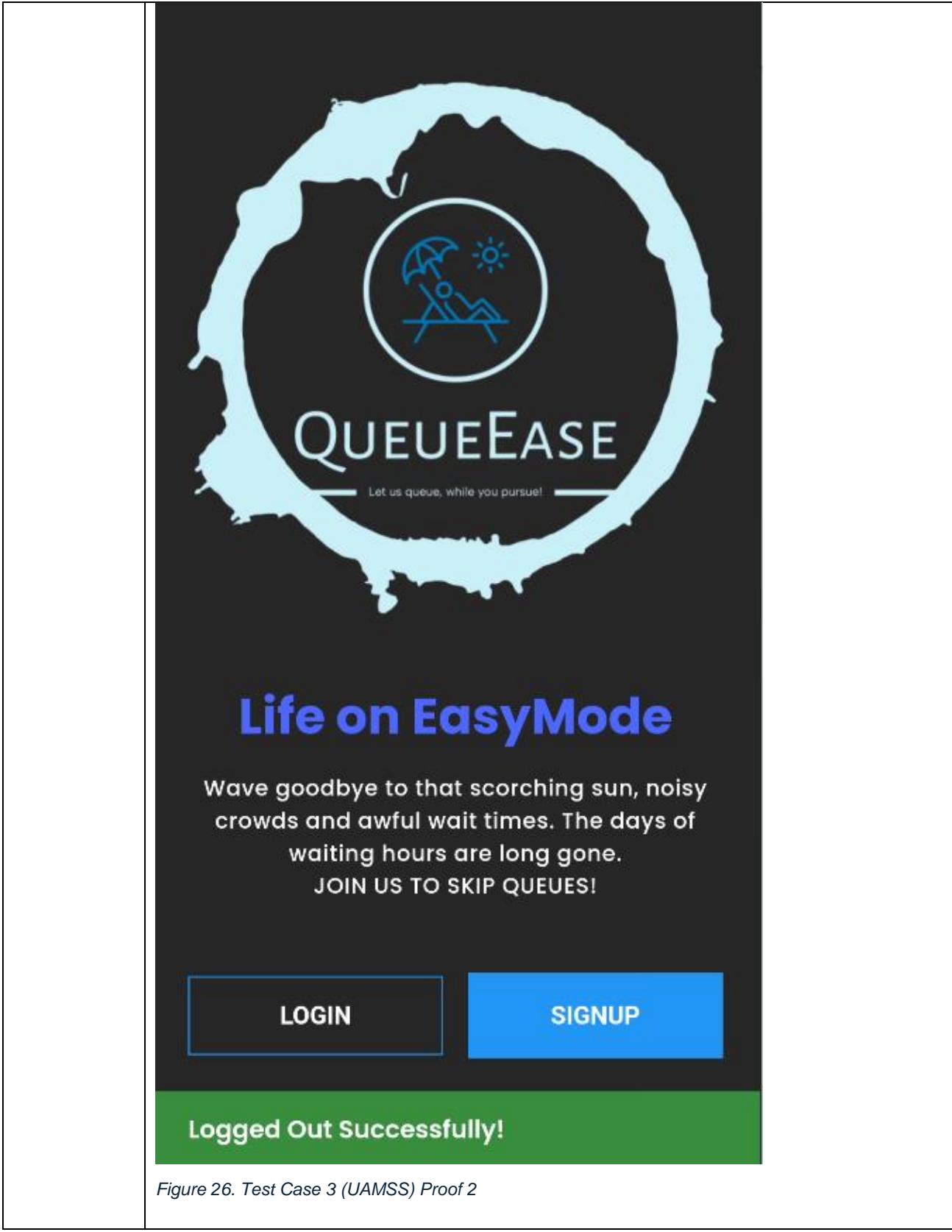


Figure 26. Test Case 3 (UAMSS) Proof 2

Table 6. Test Case 3 (UAMSS)

3.6.2. Verification and Security Subsystem (VSSS)

3.6.2.1. Purpose:

The Verification and Security subsystem aims to ensure the registration and authenticity of users as agents within the application by implementing a new registration and verification process.

3.6.2.2. Scope:

This subsystem contains functionalities related to agent registration and verification, including document submission, review, and approval.

3.6.2.3. SRS:

Req. Code	Req. Desc	MoSCoW
VSSS-F-1.0	Users, upon choosing the Agent mode from the menu, shall initiate the agent registration process	Must Have
AAMSS-NF-1.1	The system shall cross validate and securely store the provided information in the database	Should Have
AAMSS-UR-1.1	Acceptable documents may include government-issued ID, proof of residence for verification	Should Have
AAMSS-NF-1.2	Upon successful verification, users shall have access to the Agent mode functionalities	Should Have
AAMSS-UR-2.1	Agents can check out the ratings they received for their services	Could Have
AAMSS-UR-2.2	Agents shall be notified of the verification status	Should Have

Table 7. Verification and Security Subsystem SRS

3.6.2.4. Activity Diagram:

1. Agent Registration

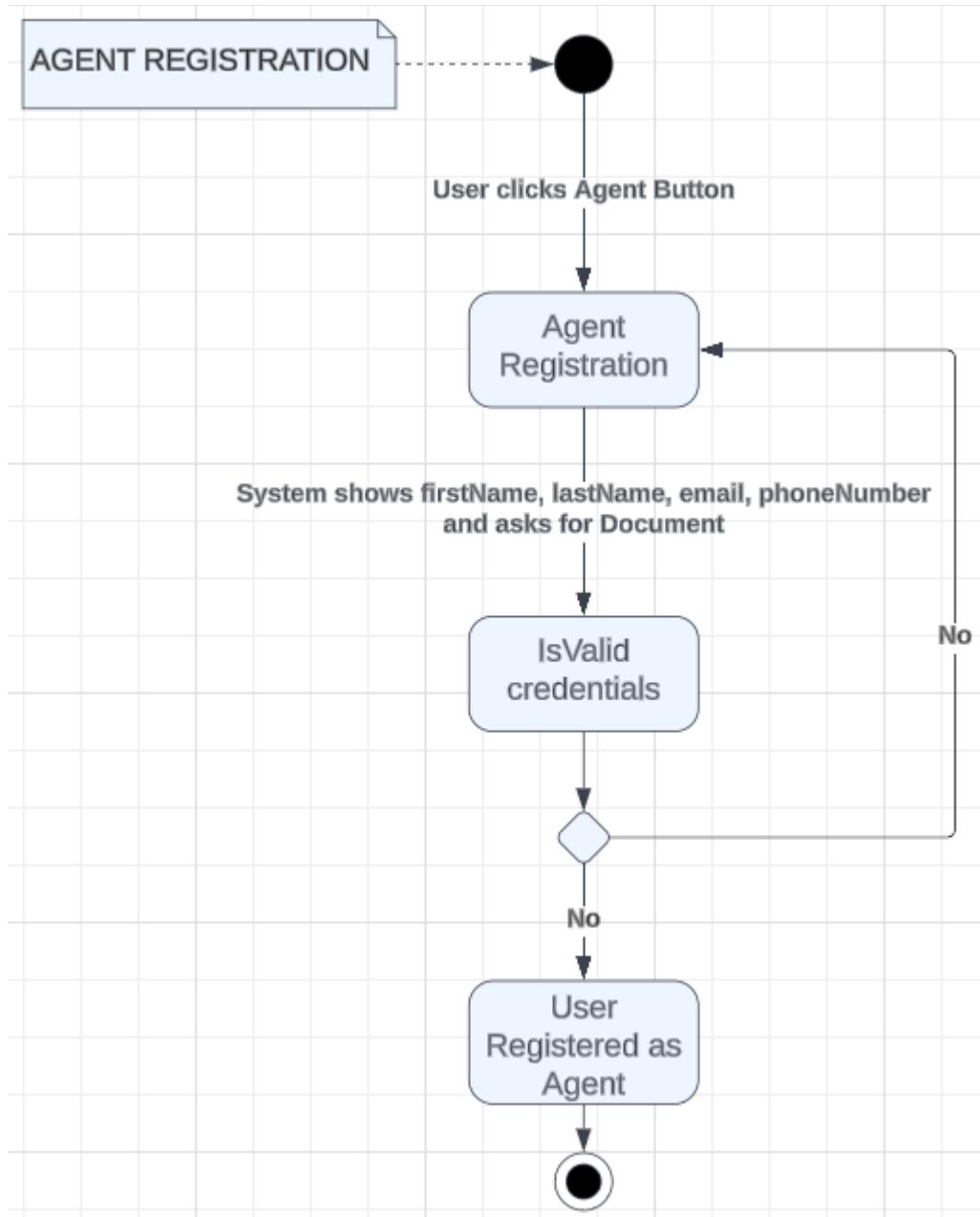


Figure 27. Agent Registration Activity Diagram

3.6.2.5. Wireframe:

1. Agent Mode

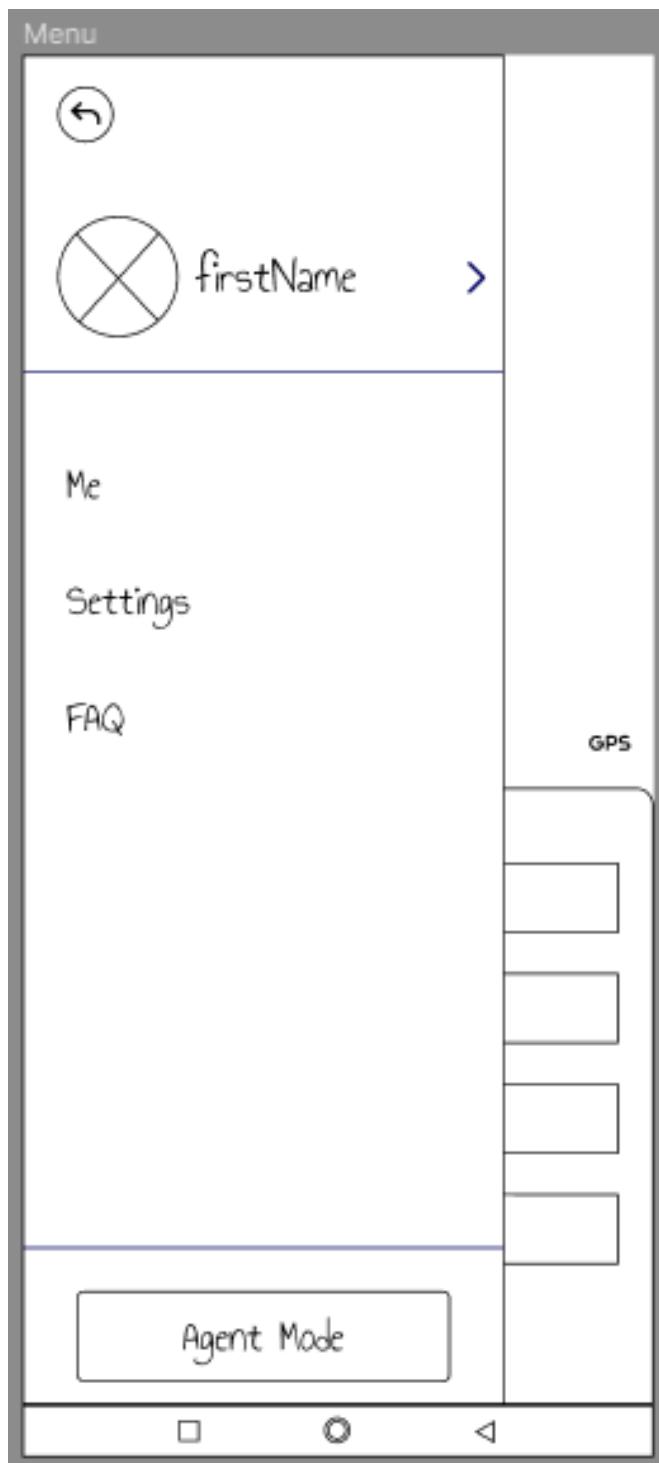
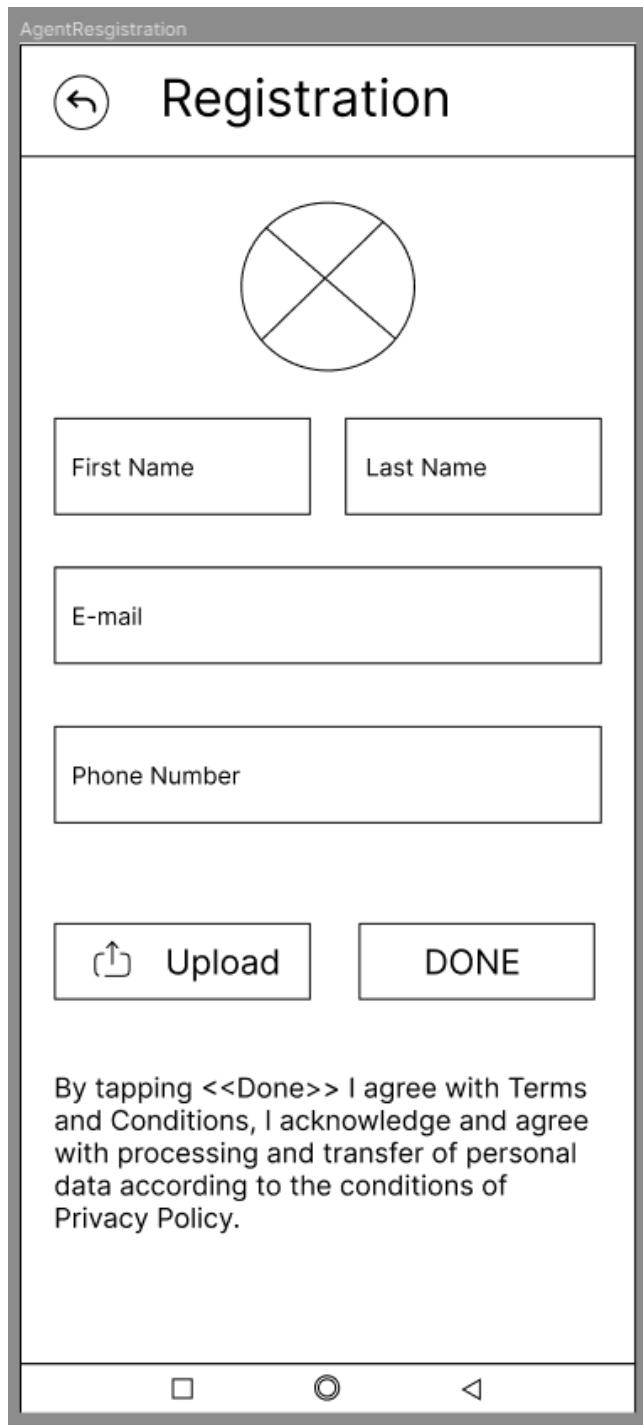


Figure 28. Agent Mode Wireframe

2. Agent Registration



The wireframe for the Agent Registration screen is titled "AgentRegistration" at the top. It features a header with a back arrow icon and the word "Registration". Below the header is a large circular placeholder with a diagonal cross. The form consists of several input fields: "First Name" and "Last Name" in separate boxes, followed by a larger box for "E-mail" and another for "Phone Number". At the bottom left is an "Upload" button with an upward arrow icon, and at the bottom right is a "DONE" button. A note below the input fields states: "By tapping <<Done>> I agree with Terms and Conditions, I acknowledge and agree with processing and transfer of personal data according to the conditions of Privacy Policy." The footer contains three navigation icons: a square, a circle, and a triangle.

Figure 29. Agent Registration Wireframe

3.6.2.6. Use-Case Diagram:

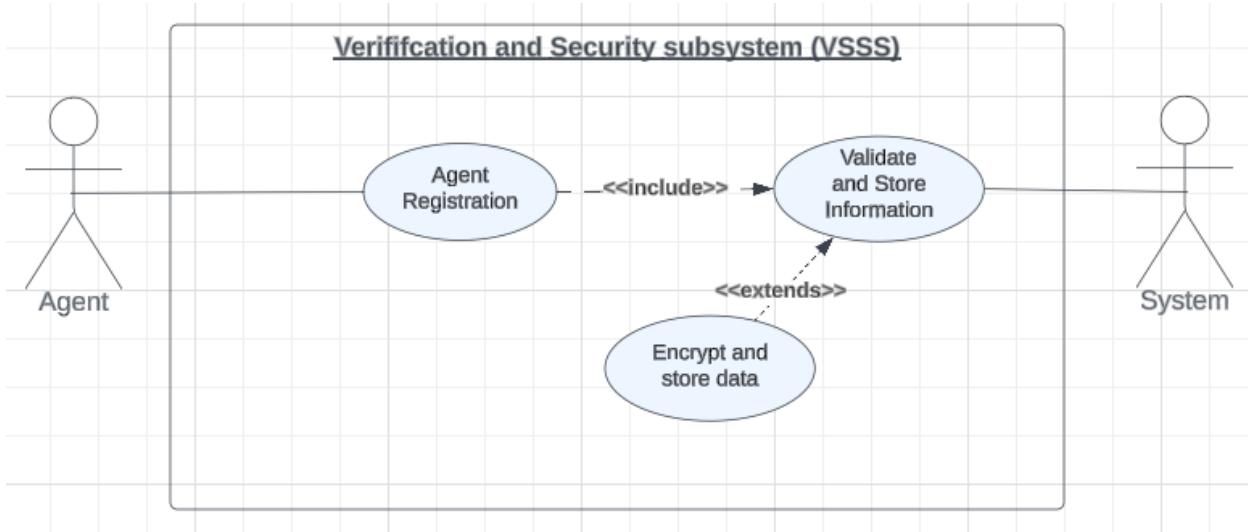


Figure 30. Verification and Security Subsystem User-Case

3.6.2.7. ERD:

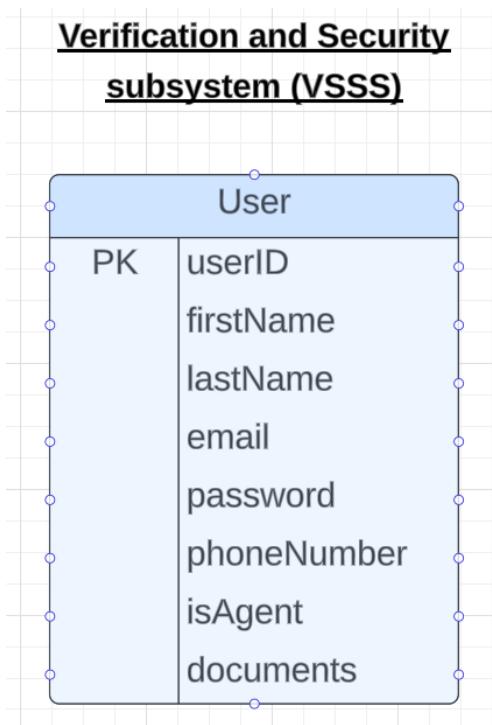


Figure 31. Verification and Security Subsystem ERD

3.6.2.8. Data Dictionary:

Field Name	Data Type	Description
userID	String	Unique identifier for each user
firstName	String	User's first name
lastName	String	User's last name
email	String	User's email address
password	String	User's encrypted password
phoneNumber	String	User's phone number
isAgent	Boolean	User's agent status
documents	String	Document's image url

3.6.2.9. Class Diagram:

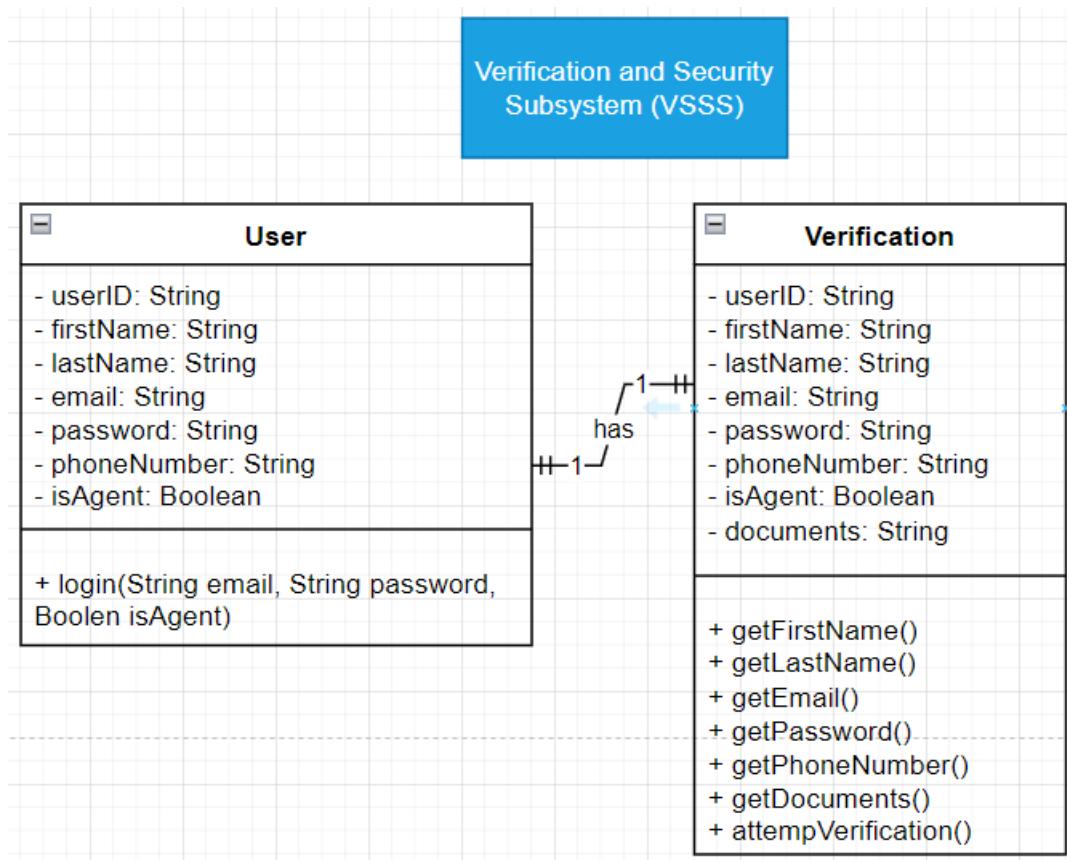


Figure 32. Verification and Security Class Diagram

3.6.2.10. Sequence Diagram:

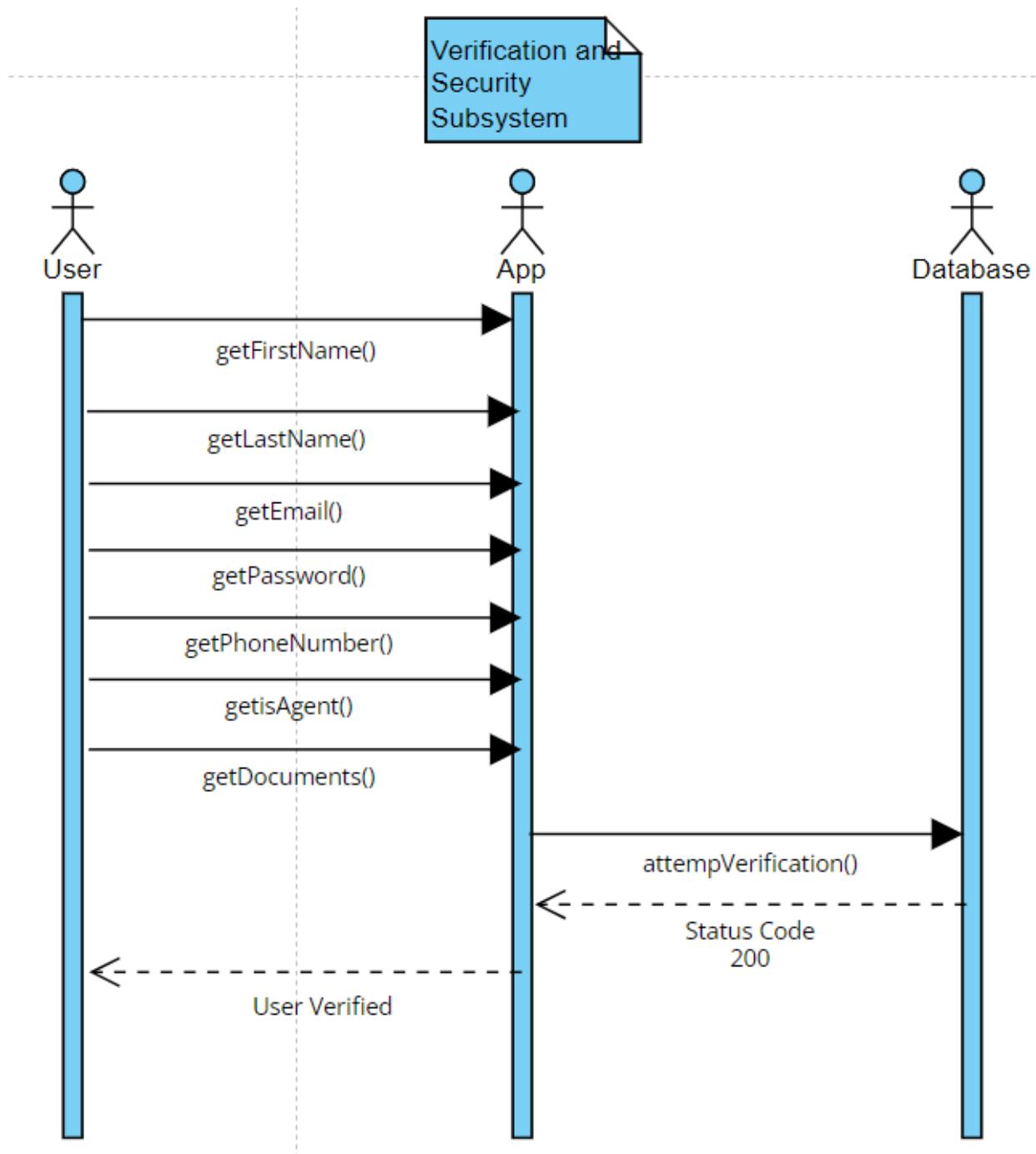


Figure 33. Verification and Security Sequence Diagram

3.6.2.11. Test Plan

Objective:

Verify the functionality and security of the Verification and Security Management subsystem.

Scope:

This test plan covers agent registration and document submission.

Test Case 1	
Objective	Validate that the user drawer has a agent mode button which transitions users to agent registration page and allows them to submit the form.
Action	<ul style="list-style-type: none">• Open app• Log in• Click on drawer menu• Click on Agent Mode Button• Click yes in the dialog box• Click on document button to upload document• Click on DONE to submit form
Expected Result	App should seamlessly navigate users to registration page and allow picking of a document image and submitting the form.
Actual Result	When users clicked on the agent button they were navigated to agent registration page where they could see their details and could upload a document as a proof and submit the form
RESULT	PASS

PROOF

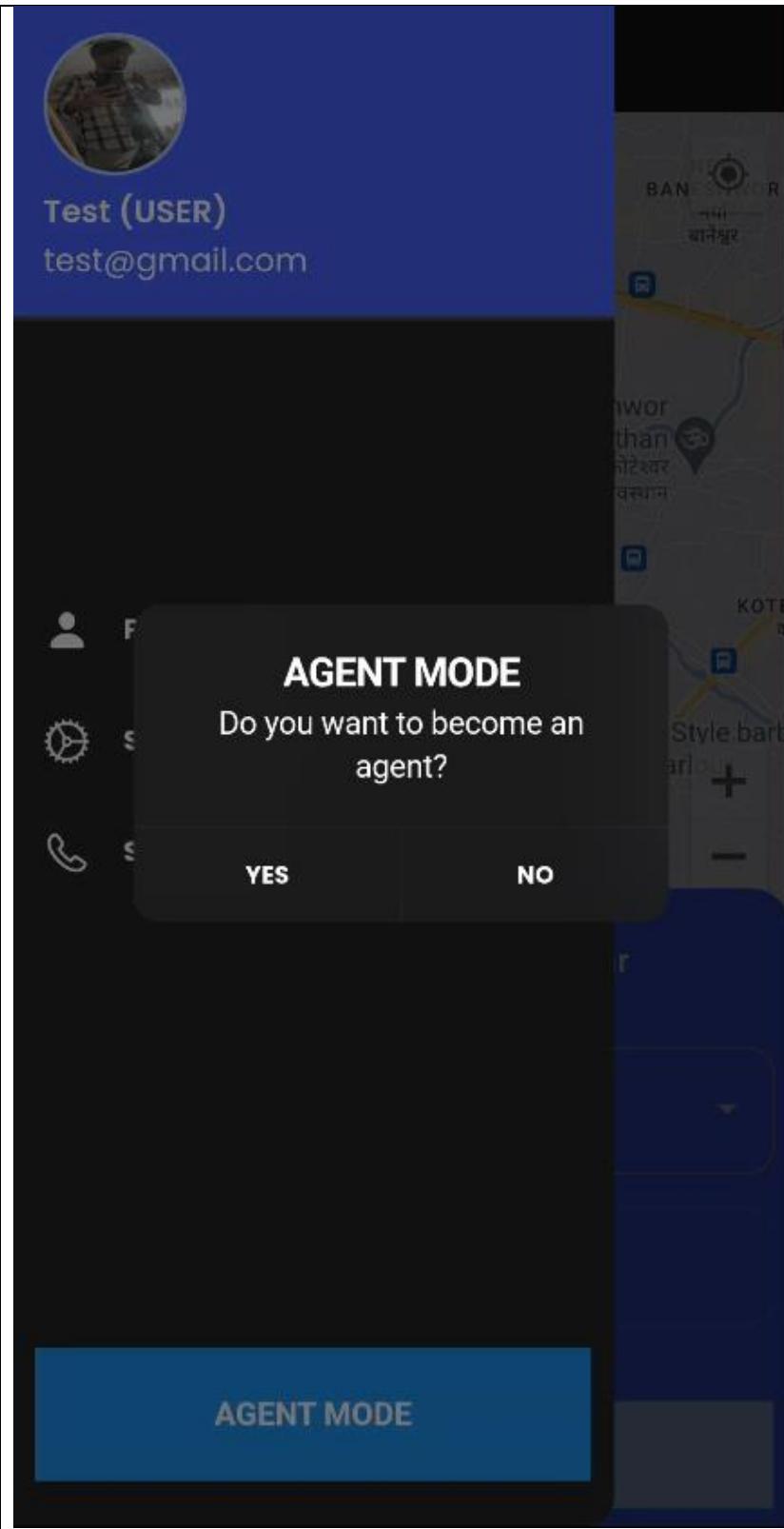


Figure 34. Test Case 1 (VSSS) Proof 1

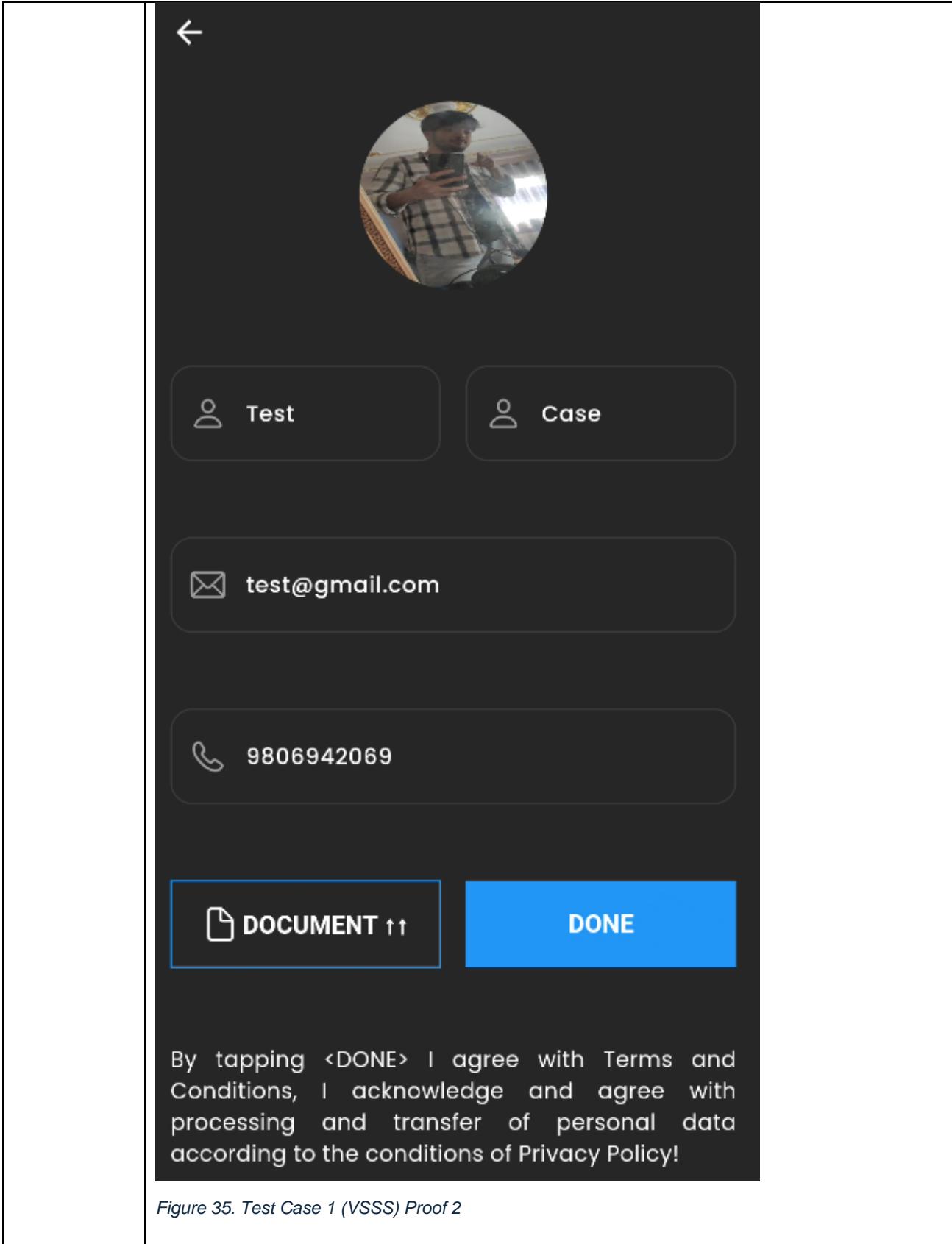


Figure 35. Test Case 1 (VSSS) Proof 2

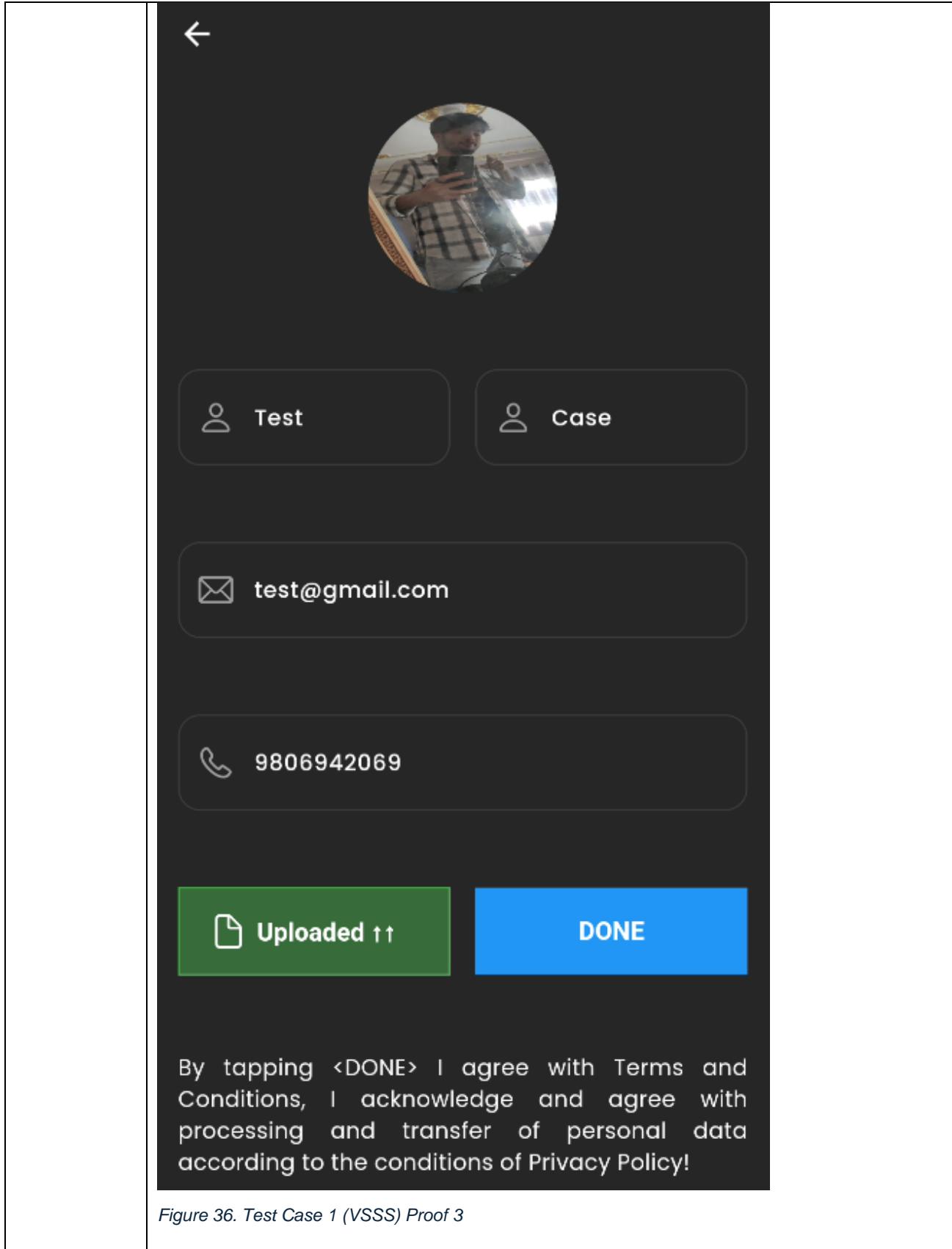


Figure 36. Test Case 1 (VSSS) Proof 3

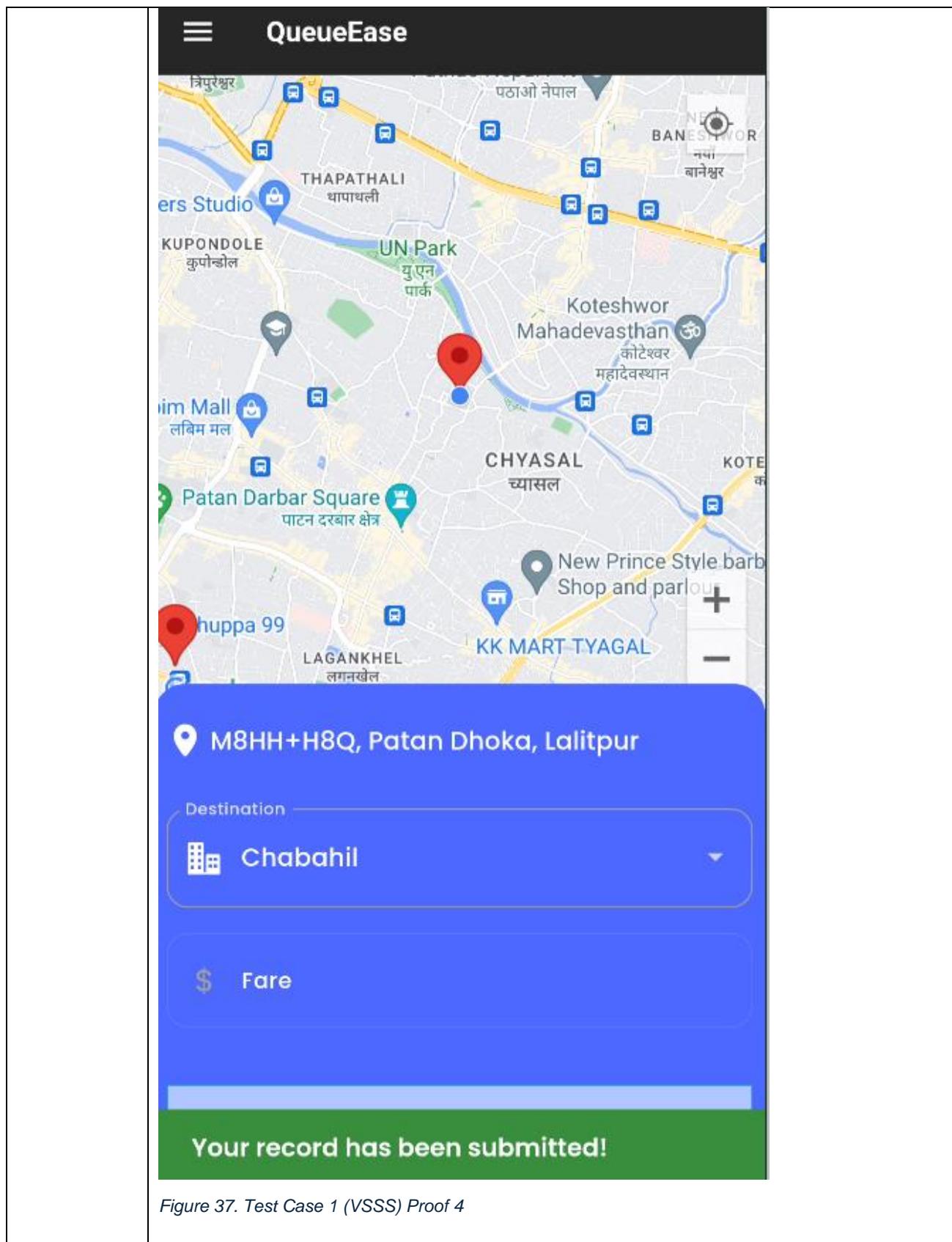


Figure 37. Test Case 1 (VSSS) Proof 4

Table 8. Test Case 1 (VSSS)

Test Case 2	
Objective	Validate that the registered user has agent access throughout the app.
Action	<ul style="list-style-type: none"> • Register as agent • Click on drawer
Expected Result	After submitting the form the app should redirect users to home page and when again clicking the drawer they must have agent role.
Actual Result	After submission of the form users were able to tell from the drawer role tag beside their name and the addition of request in drawer and removal of agent mode button.
RESULT	PASS

PROOF

The image shows a mobile application interface. At the top left, there is a circular profile picture of a person. Below the picture, the text "Test (AGENT)" is displayed, followed by the email address "test@gmail.com". To the right of the profile section is a map showing a route from "BANESTHOR" to "KOTE". The map includes labels for "IWOR than", "नोटेश्वर", "वस्यान", "Prince Style barb", and "and parlour". Below the map, there is a dark blue navigation bar with four items: "PROFILE" (with a user icon), "SETTINGS" (with a gear icon), "SUPPORT" (with a phone icon), and "REQUESTS" (with a clock icon). The "REQUESTS" item has a small red notification badge with the number "1" above it.

Figure 38. Test Case 2 (VSSS) Proof 2

Table 9. Test Case 2 (VSSS)

3.6.3. Booking Subsystem (BSS)

3.6.3.1. Purpose:

The Booking subsystem is designed to facilitate the process of users requesting and agents accepting bookings within the system.

3.6.3.2. Scope:

This subsystem contains functionalities related to user request submission, agent response handling and the initiation of the booking process

3.6.3.3. SRS:

Req. Code	Req. Desc	MoSCoW
BSS-F-1.0	Users shall access the booking menu on the dashboard and can submit a new booking request	Must Have
BSS-NF-1.1	Users shall provide details like their destination and price	Should Have
BSS-F-2.0	Agents registered within the requested destination shall receive real-time notifications of new booking requests	Must Have
BSS-NF-2.1	The system shall provide agents with relevant details such as requesters name, destination and price	Should Have
BSS-NF-2.2	Agents shall have the option to accept or decline booking requests	Should Have
BSS-NF-2.3	Upon accepting a booking request, the system will initiate the chat subsystem for communication	Must Have
BSS-NF-2.4	The system shall maintain a booking history for both users and agents, accessible from their respective dashboards.	Could Have

Table 10. Booking Subsystem SRS

3.6.3.4. Activity Diagram:

1. User Booking Initiation

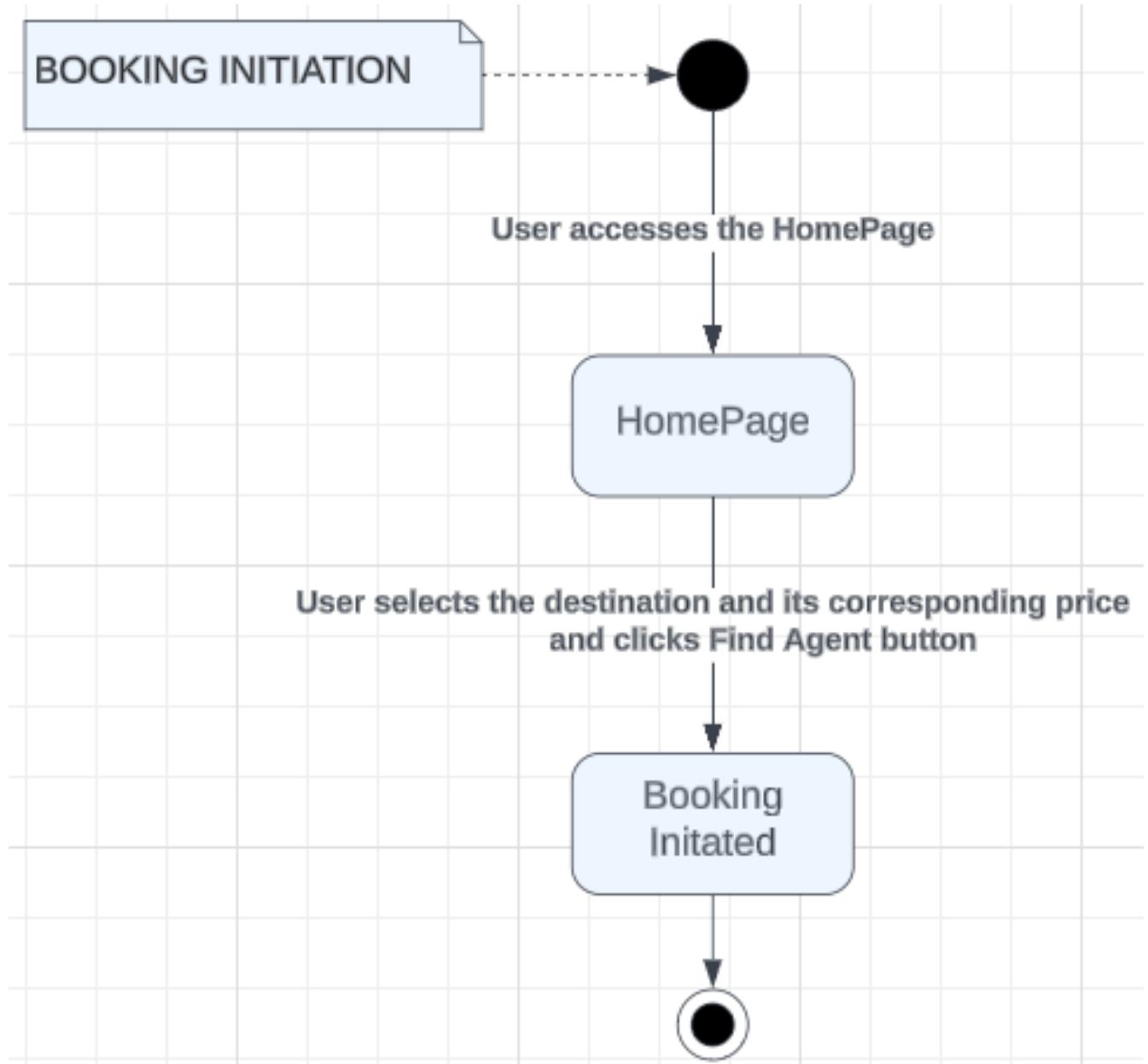


Figure 39. User Booking Initiation Activity Diagram

2. Agent Notification

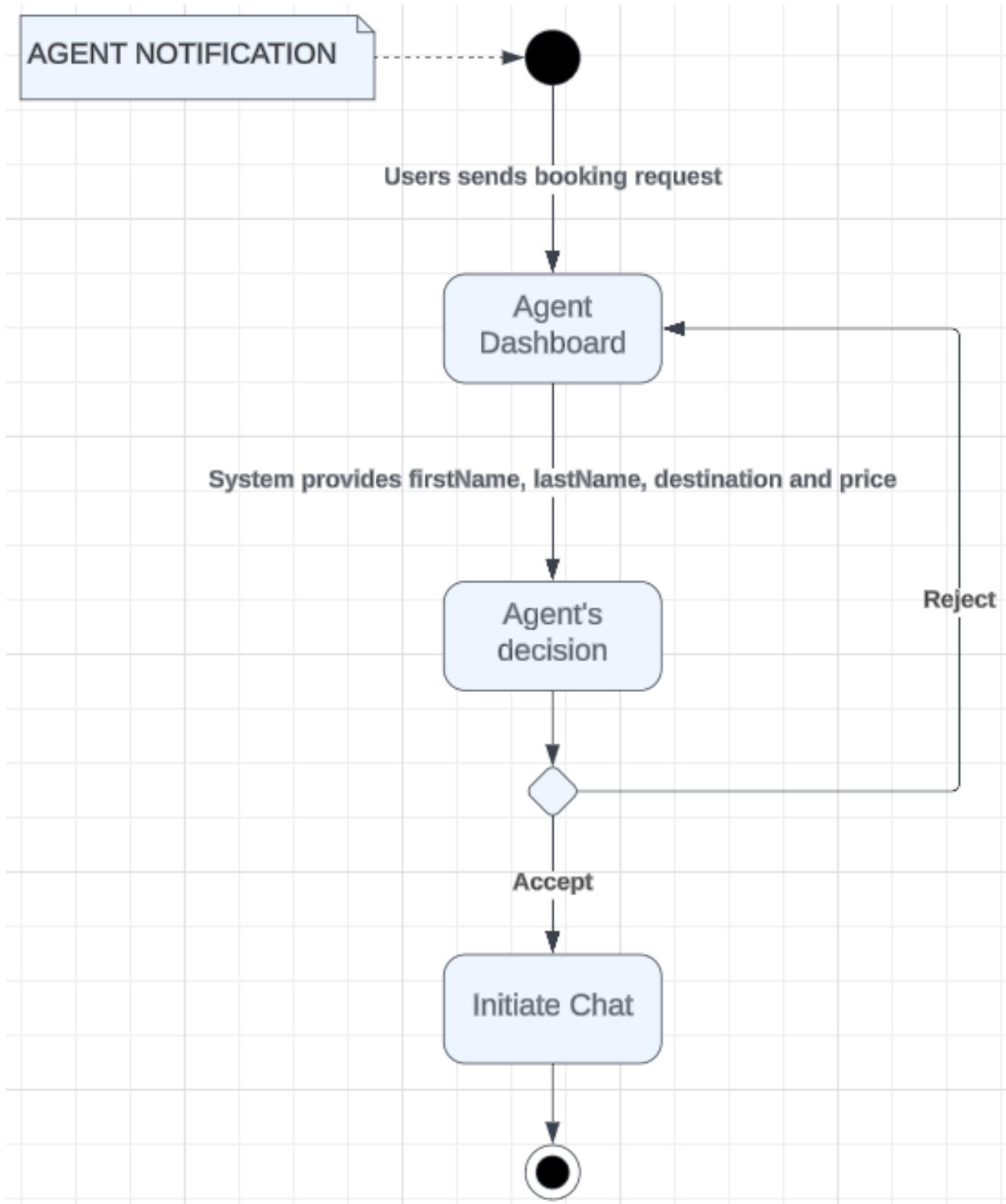


Figure 40. Agent Notification Activity Diagram

3.6.3.5. Wireframe:

1. User Dashboard

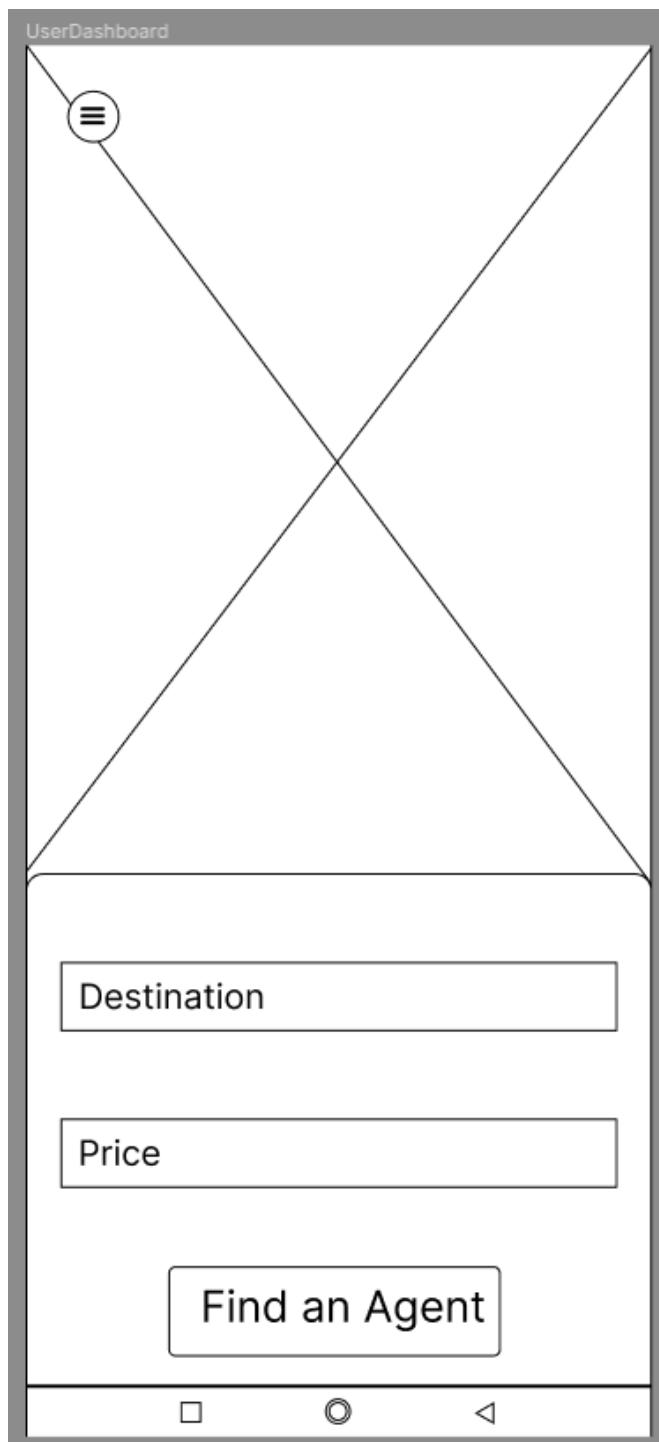


Figure 41. Booking Initiation Wireframe

2. Agent Dashboard

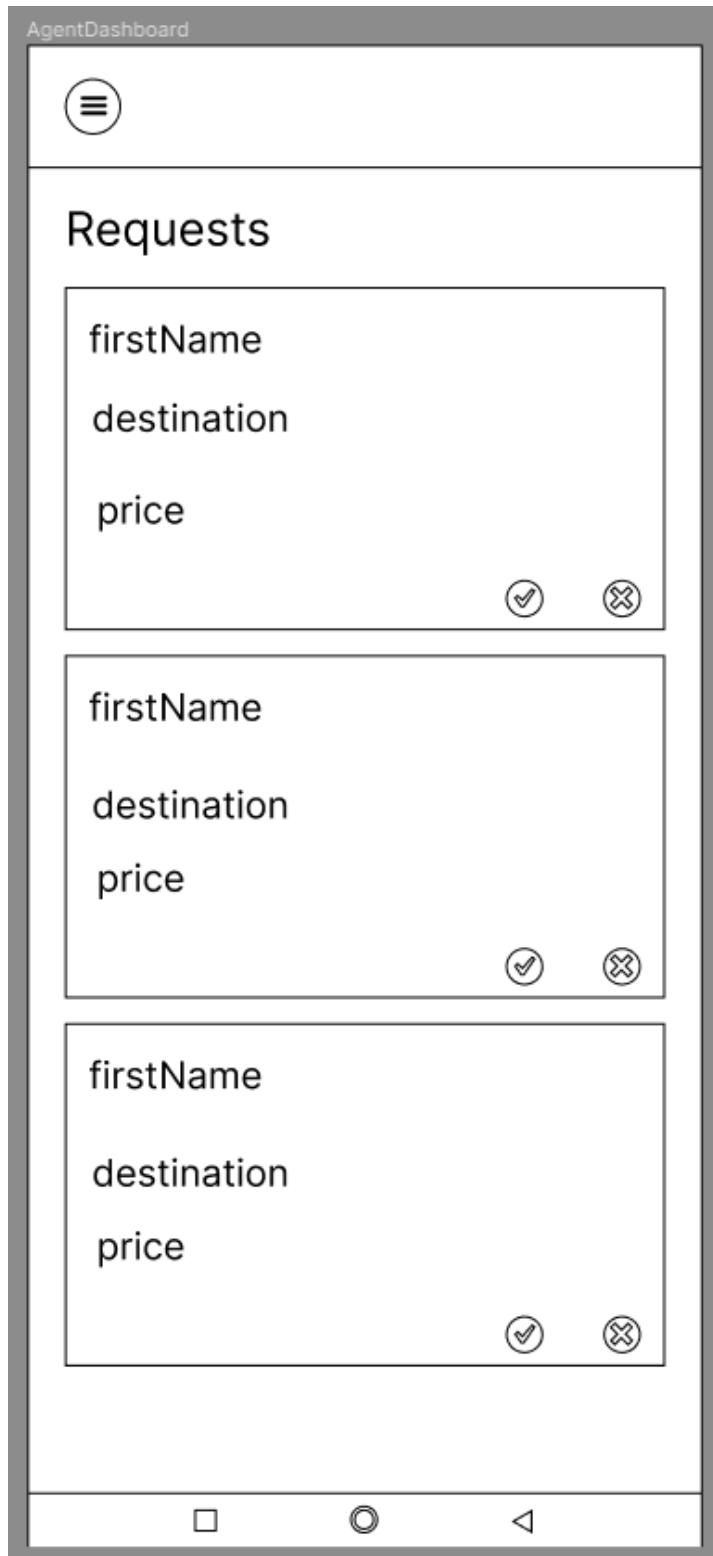


Figure 42. Agent Notification Wireframe

3.6.3.6. Use-Case Diagram:

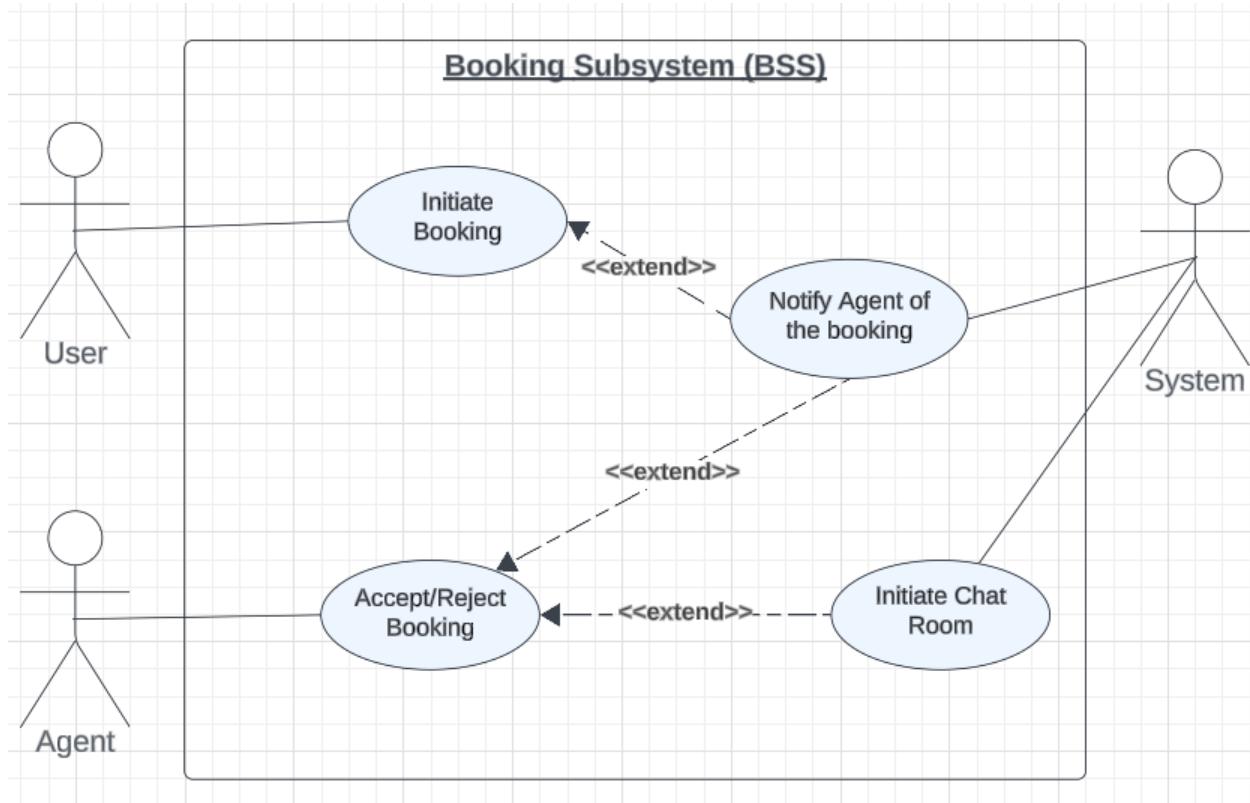


Figure 43. Booking Subsystem Use-Case

3.6.3.7. ERD:

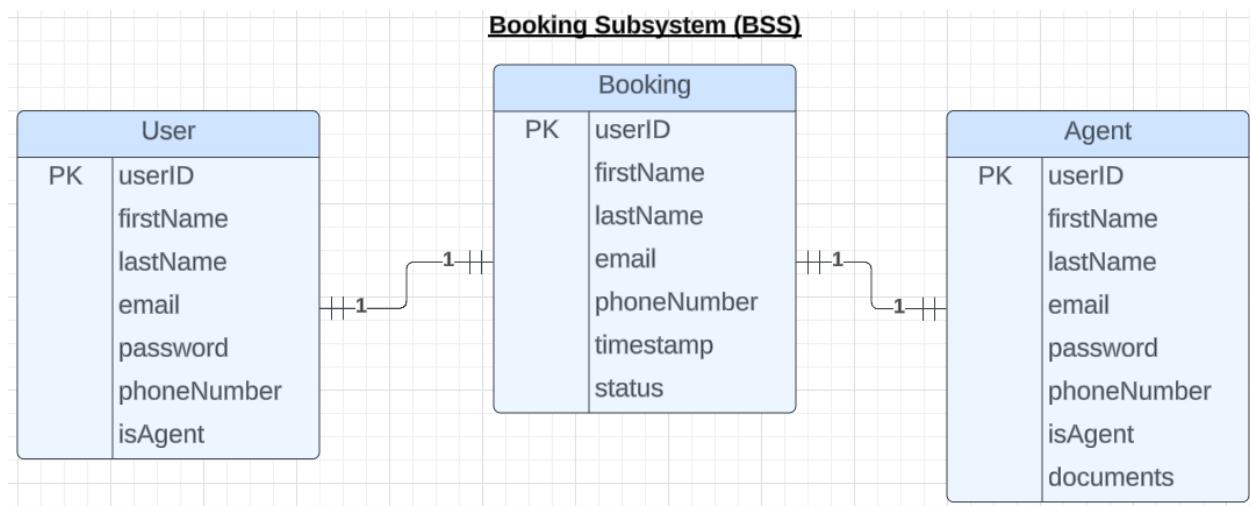


Figure 44. Booking Subsystem ERD

3.6.3.8. Data Dictionary:

Field Name	Data Type	Description
userID	String	Foreign Key referencing the User entity who initiated the booking
agentID	String	Foreign Key referencing the Agent entity who accepted the booking
destination	String	Location where the user wants the agent to stand in line
price	Double	Amount the user is willing to pay for the service
status	String	Booking status (e.g., Pending, Accepted, Completed, Canceled)
timestamp	Date	Date and time when the booking was created

Table 11. Booking Subsystem Data Dictionary.

3.6.3.9. Class Diagram:

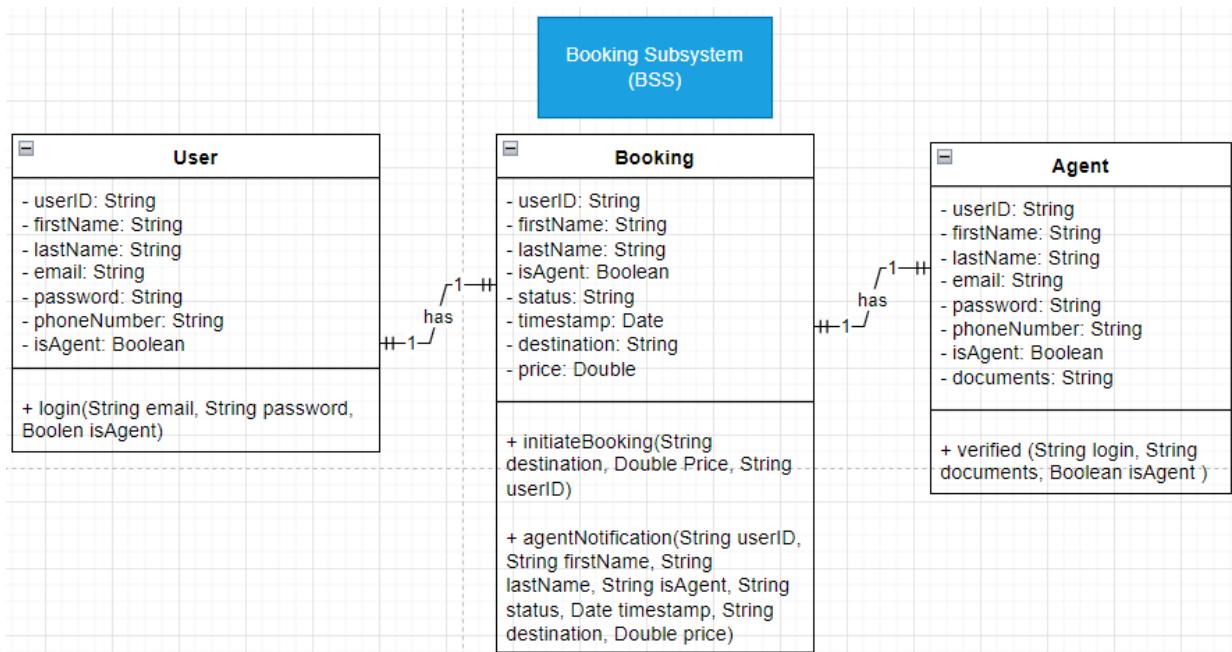


Figure 45. Booking Subsystem Class Diagram

3.6.3.10. Sequence Diagram:

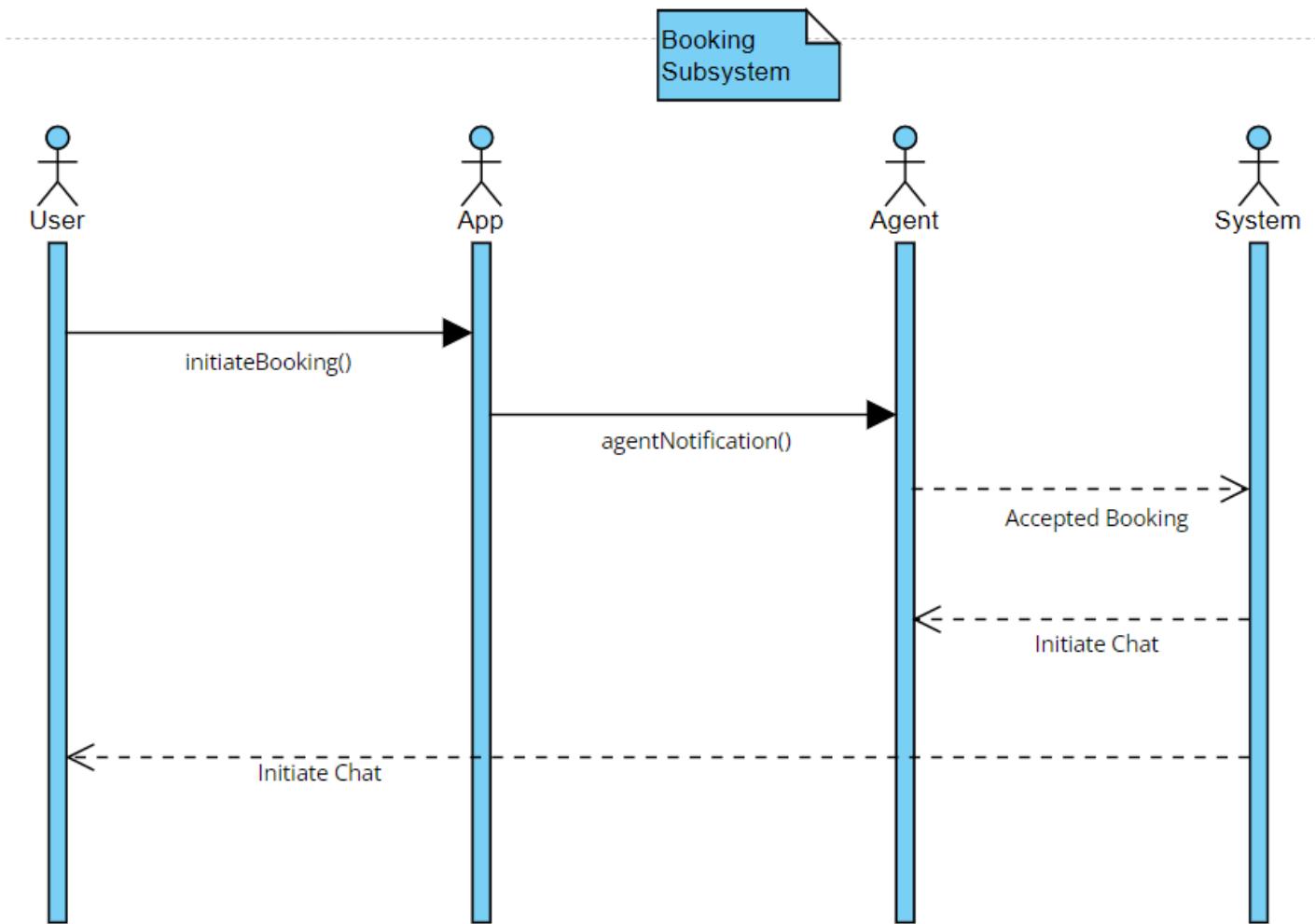


Figure 46. Booking Subsystem Sequence Diagram

3.6.3.11. Test Plan

Objective:

Verify the functionality and reliability of the Booking subsystem.

Scope:

This test plan covers user request submission, agent notification, and booking acceptance.

Test Case 1	
Objective	Validate that the booking container captures required information and button works as intended
Action	<ul style="list-style-type: none">• Log in• Enter destination in the booking container• Click on Find an Agent button
Expected Result	After logging in in the homepage user should see their location select the destination of the government office and view price and find agent by clicking the button.
Actual Result	When users log in to the app they are greeted with a home screen containing their location and a dropdown box which allows them to select the destination and related price and when pressing the button request is being sent to agents.
RESULT	PASS

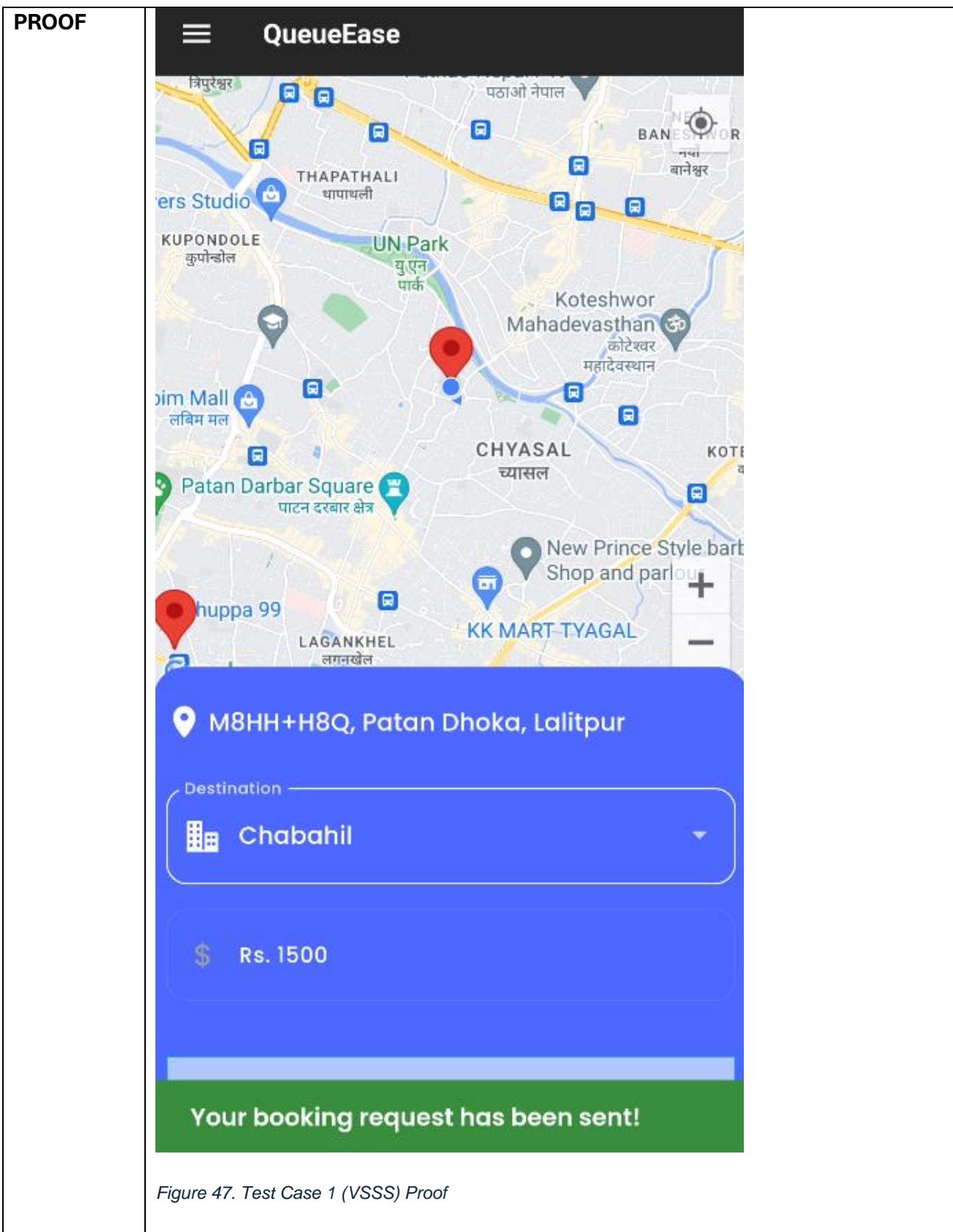


Figure 47. Test Case 1 (VSSS) Proof

Table 12. Test Case 1 (BSS)

Test Case 2	
Objective	Validate that the booking container captures required information and button works as intended
Action	<ul style="list-style-type: none"> • Log in • Enter destination in the booking container • Click on Find an Agent button
Expected Result	After logging in in the homepage user should see their location select the destination of the government office and view price and find agent by clicking the button.
Actual Result	When users log in to the app they are greeted with a home screen containing their location and a dropdown box which allows them to select the destination and related price and when pressing the button request is being sent to agents.
RESULT	PASS

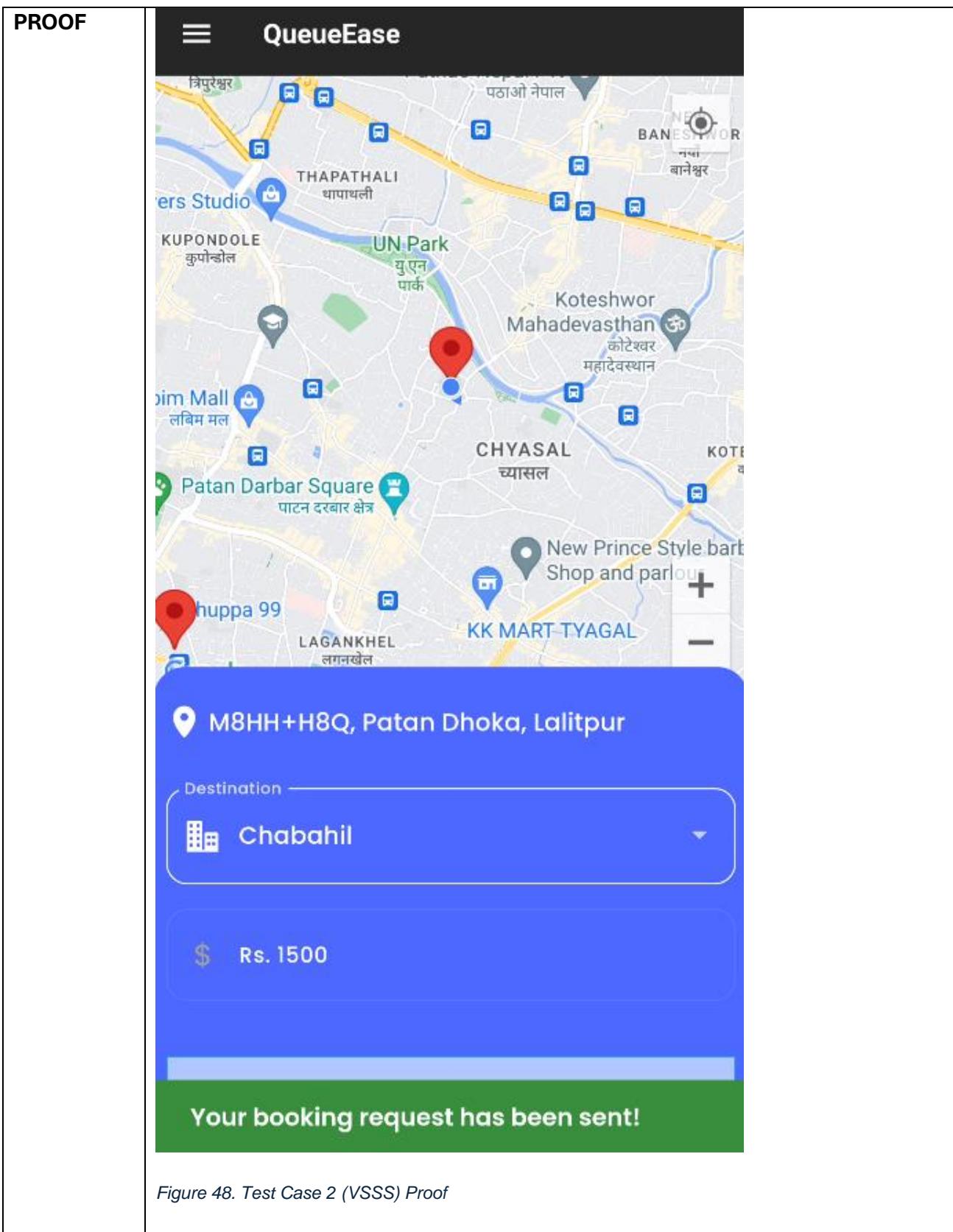


Figure 48. Test Case 2 (VSSS) Proof

Table 13. Test Case 2 (BSS)

Test Case 3	
Objective	Simulate a new booking request and confirm that agents receive real-time notifications with proper details (destination, name, price)
Action	<ul style="list-style-type: none"> • Log in as agent • Click on VIEW
Expected Result	After logging in as agent wait for request and when request arrives it should show in a pop up and can be viewed or closed. The pop up should contain relevant details of the requestor.
Actual Result	Agents got real-time notifications in alertbox format with relevant details of the requestor and had the option to view or close.
RESULT	PASS

PROOF

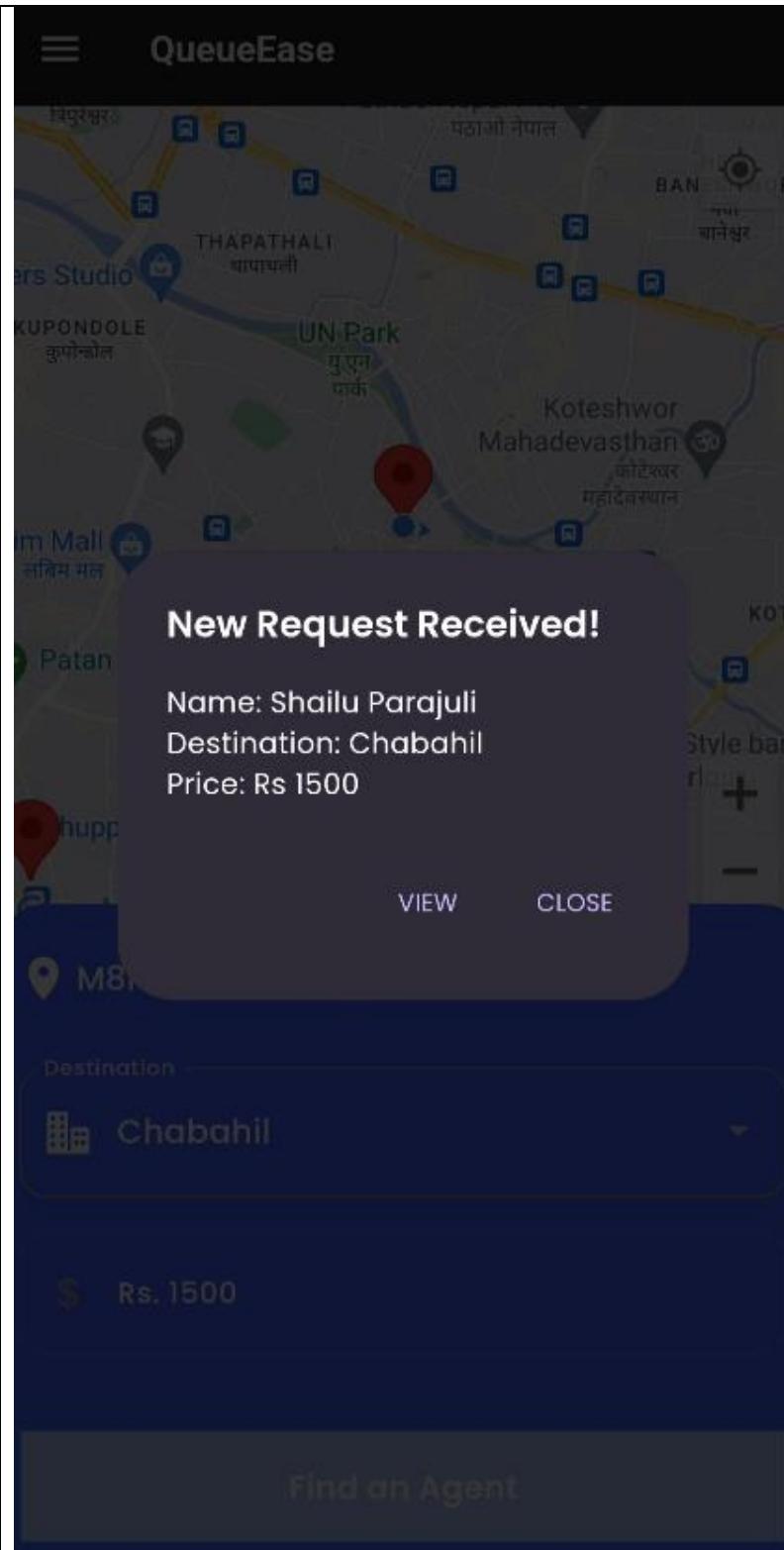


Figure 49. Test Case 3 (BSS) Proof 1

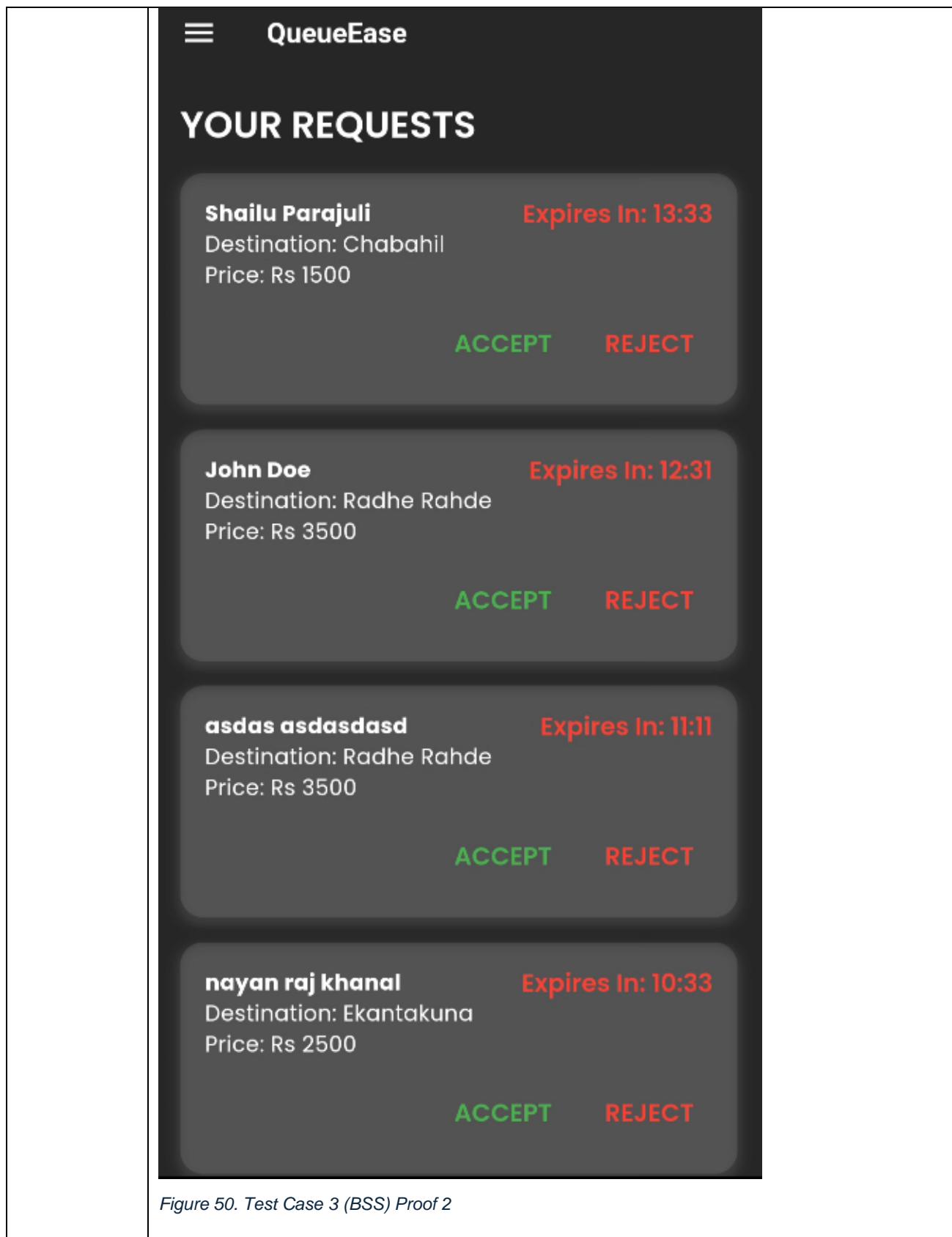


Table 14. Test Case 3 (BSS)

3.6.4. Communication Subsystem (ComSS)

3.6.4.1. Purpose:

The Communication subsystem is designed to enable real-time communication between users and agents within the system.

3.6.4.2. Scope:

This subsystem consists of functionalities related to initiating, maintaining and concluding chats between users and agents.

3.6.4.3. SRS:

Req. Code	Req. Desc	MoSCoW
ComSS-F-1.0	Upon acceptance of a booking request, the system shall initiate a chat interface for both the user and the agent.	Must Have
ComSS-NF-1.1	The chat interface shall allow basic real-time text messaging and option to call	Should Have
ComSS-NF-1.2	Users shall be able to confirm the completion of service through the chat interface.	Should Have
ComSS-NF-1.3	Upon completion of the service the user and agent can exit out of the chat interface	Should Have
ComSS-NF-1.4	The system shall store chat histories securely for each booking and can be accessible from their respective dashboards.	Could Have

Table 15. Communication Subsystem SRS

3.6.4.4. Activity Diagram:

1. User Agent Communication

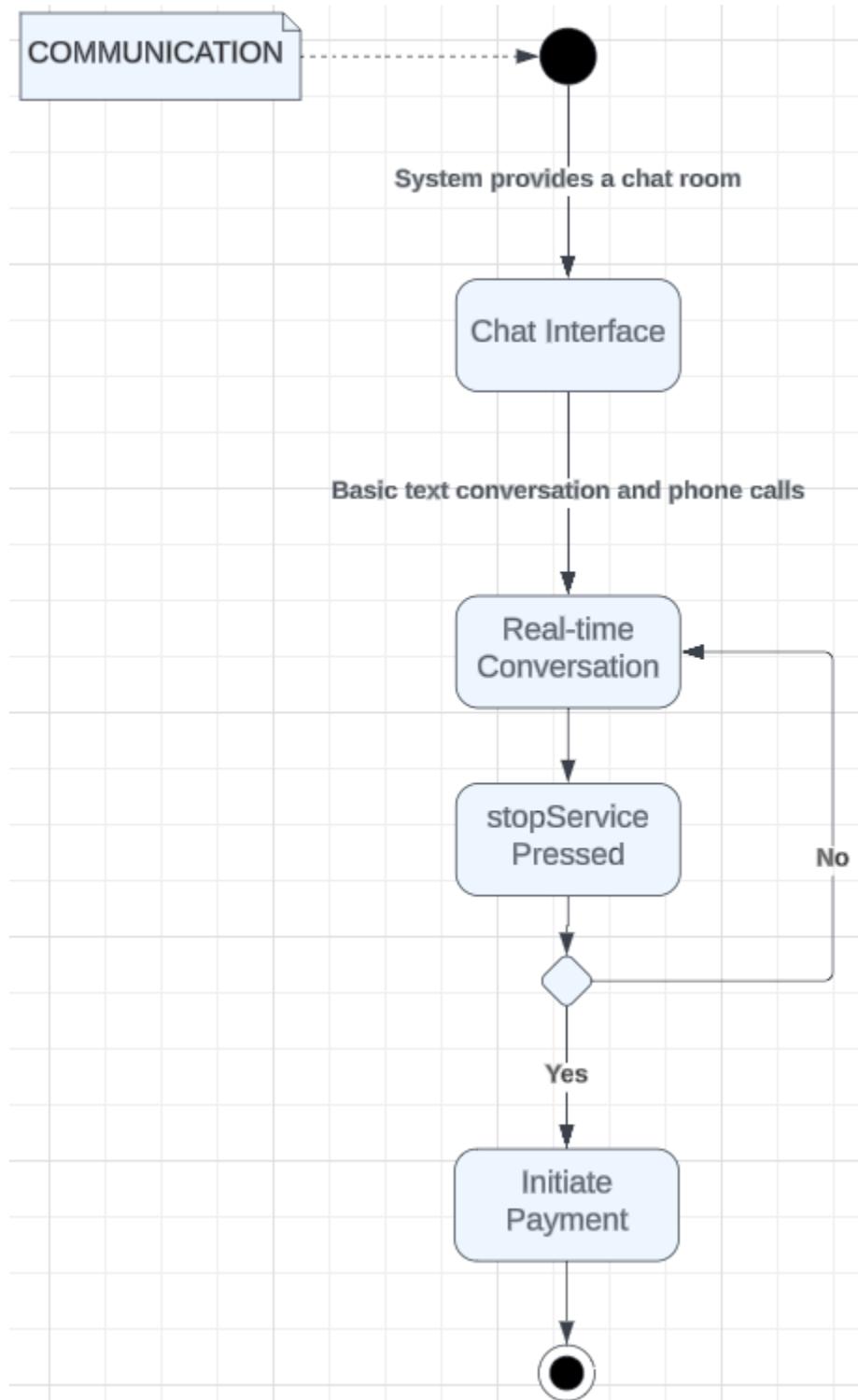


Figure 51. User-Agent Communication

3.6.4.5. Wireframe:

1. Chat box user-side

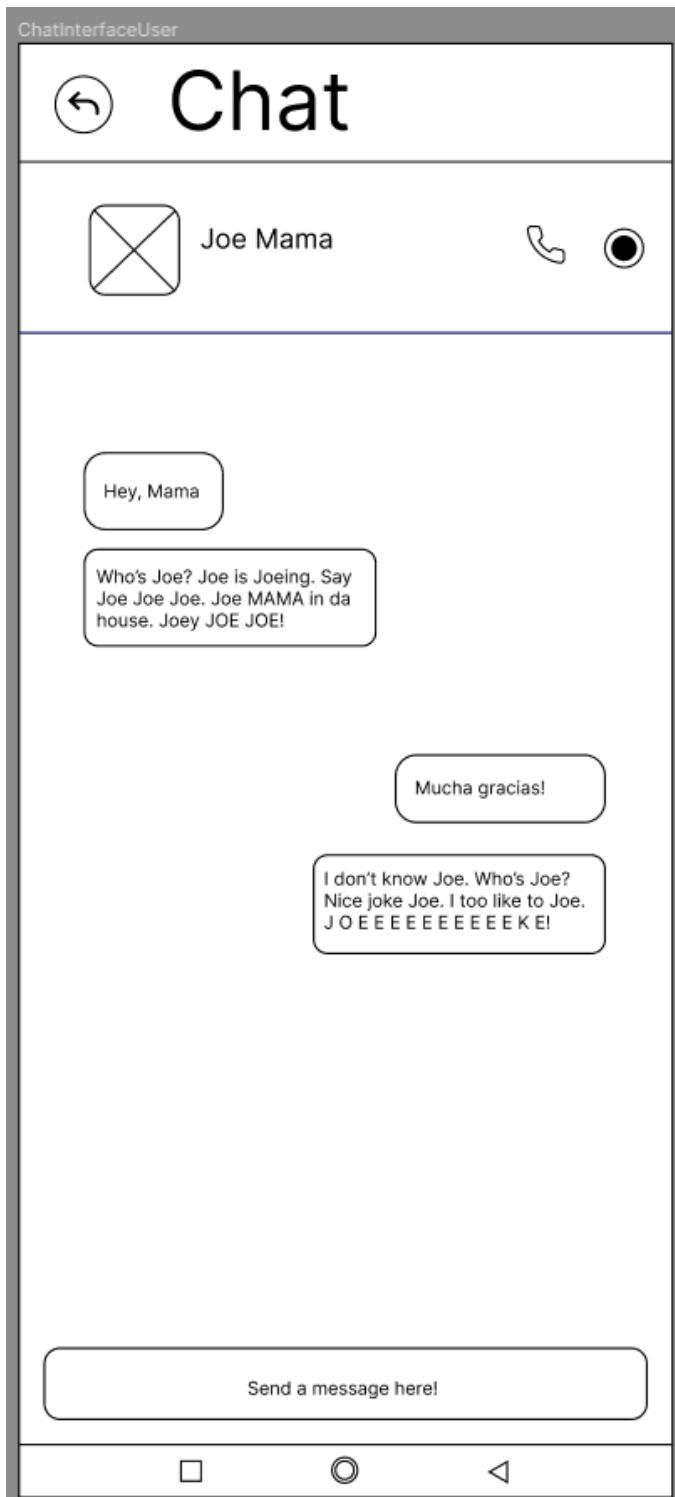


Figure 52. Chatbox

3.6.4.6. Use-Case Diagram:

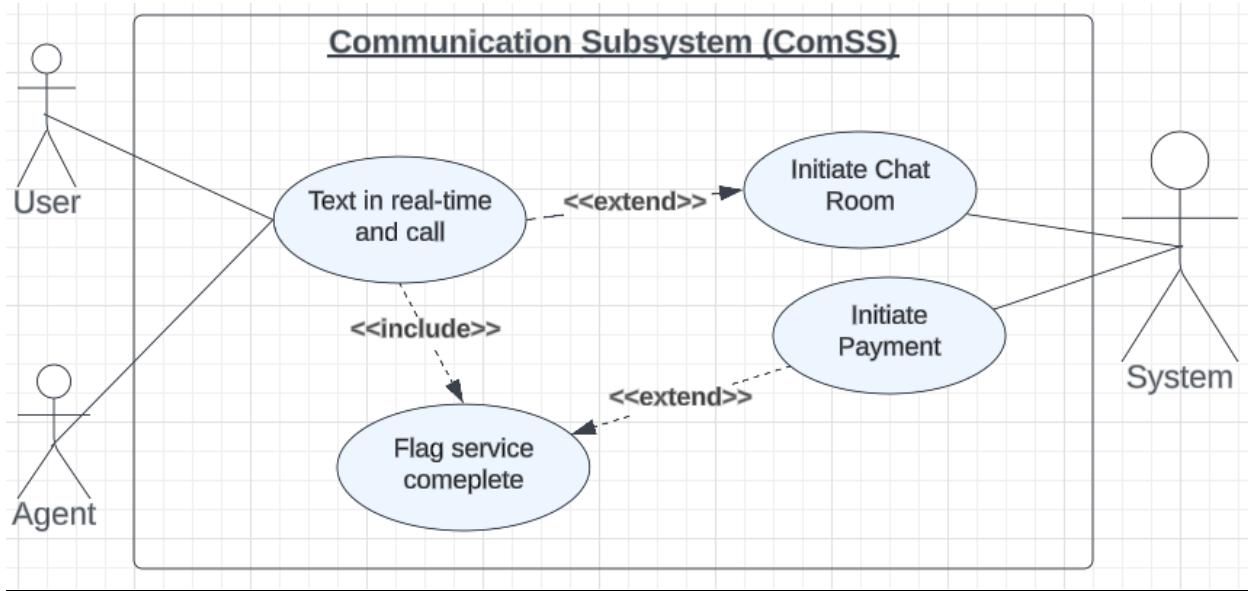


Figure 53. Communication Subsystem Use-Case Diagram

3.6.4.7. ERD:

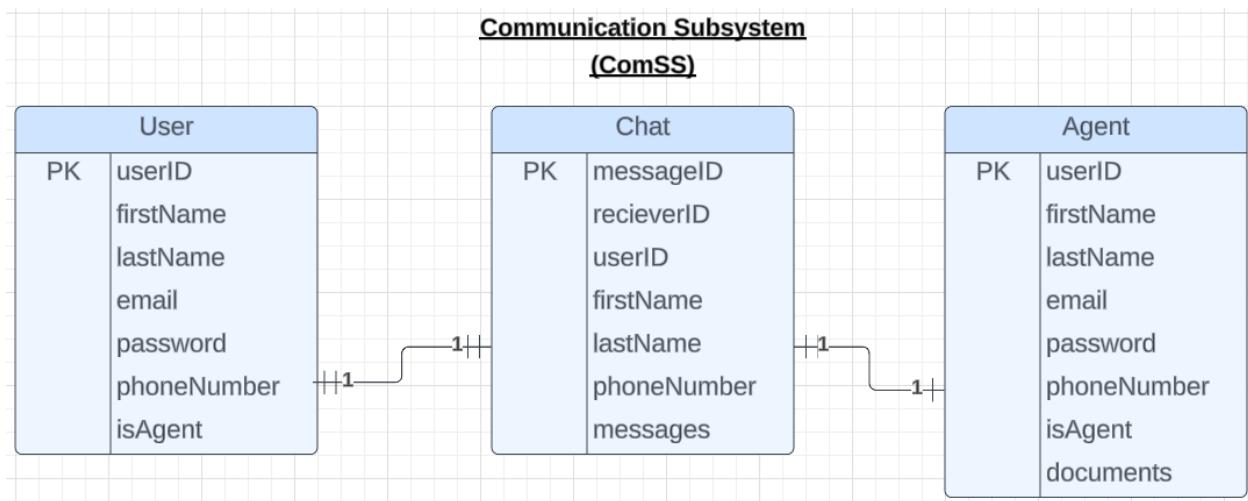


Figure 54. Communication Subsystem ERD

3.6.4.8. Data Dictionary:

Field Name	Data Type	Description
messageID	String	Unique identifier for each message
userID	String	Foreign Key referencing the User or Agent who sent the message
recieverID	String	User or Agent who received the message
phoneNumber	String	Phone number of the user
firstName	String	First name of the user
lastName	String	Last name of the user
messages	String	Messages between users and agents

Table 16. Communication Subsystem Data Dictionary

3.6.4.9. Class Diagram:

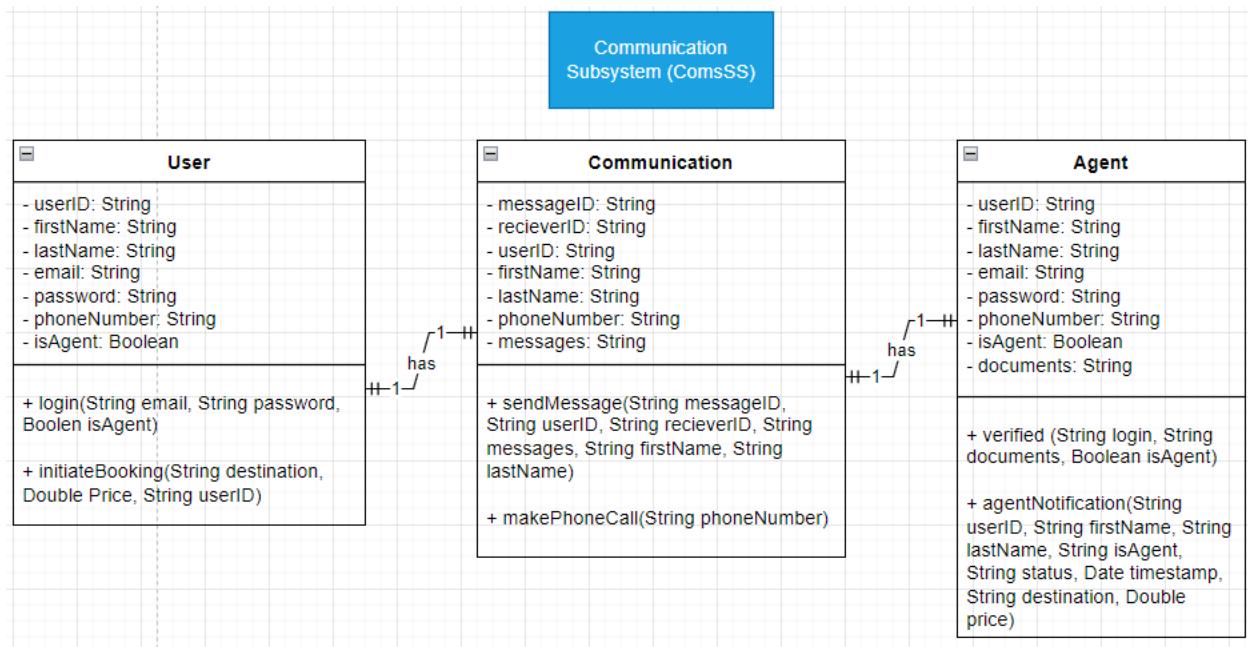


Figure 55. Communication Subsystem Class Diagram

3.6.4.10. Sequence Diagram:

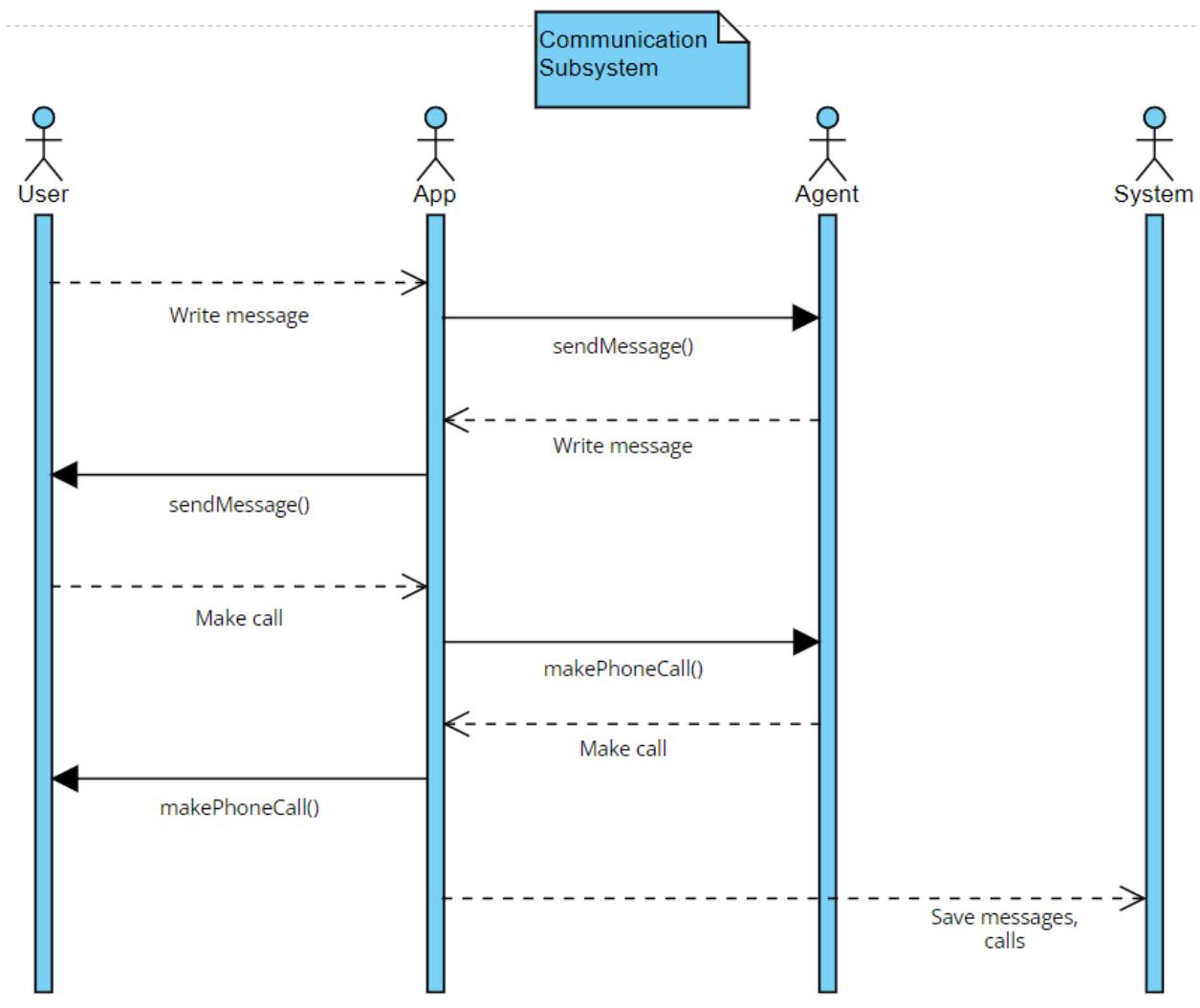


Figure 56 . Communication Subsystem Sequence Diagram

3.6.4.11. Test Plan

Test Case 1	
Objective	Verify that a chat room is established when agent accepts a request from the user and only that agent and agent are included.
Action	<ul style="list-style-type: none">• Log in as agent• Go to requests• Accept a request
Expected Result	After logging in as and navigating to the request tab. Agents when click on accept button should make the app create a room between that agent and the requestor.
Actual Result	Agents after logging in and getting to request page when clicking on accept button were redirected to the chat page.
RESULT	PASS

PROOF



Figure 57. Test Case 1 (ComSS) Proof

Table 17. Test Case 1 (ComSS)

Test Case 2	
Objective	Test the ability of users and agents to call each other via the app
Action	<ul style="list-style-type: none"> • Log in • ChatPage • Call
Expected Result	After logging in as and navigating to the chat page. Agents and users when click on call button should be able to call each other.
Actual Result	After logging in as and navigating to the chat page. Agents and users when click on call button were able to call each other.
RESULT	PASS

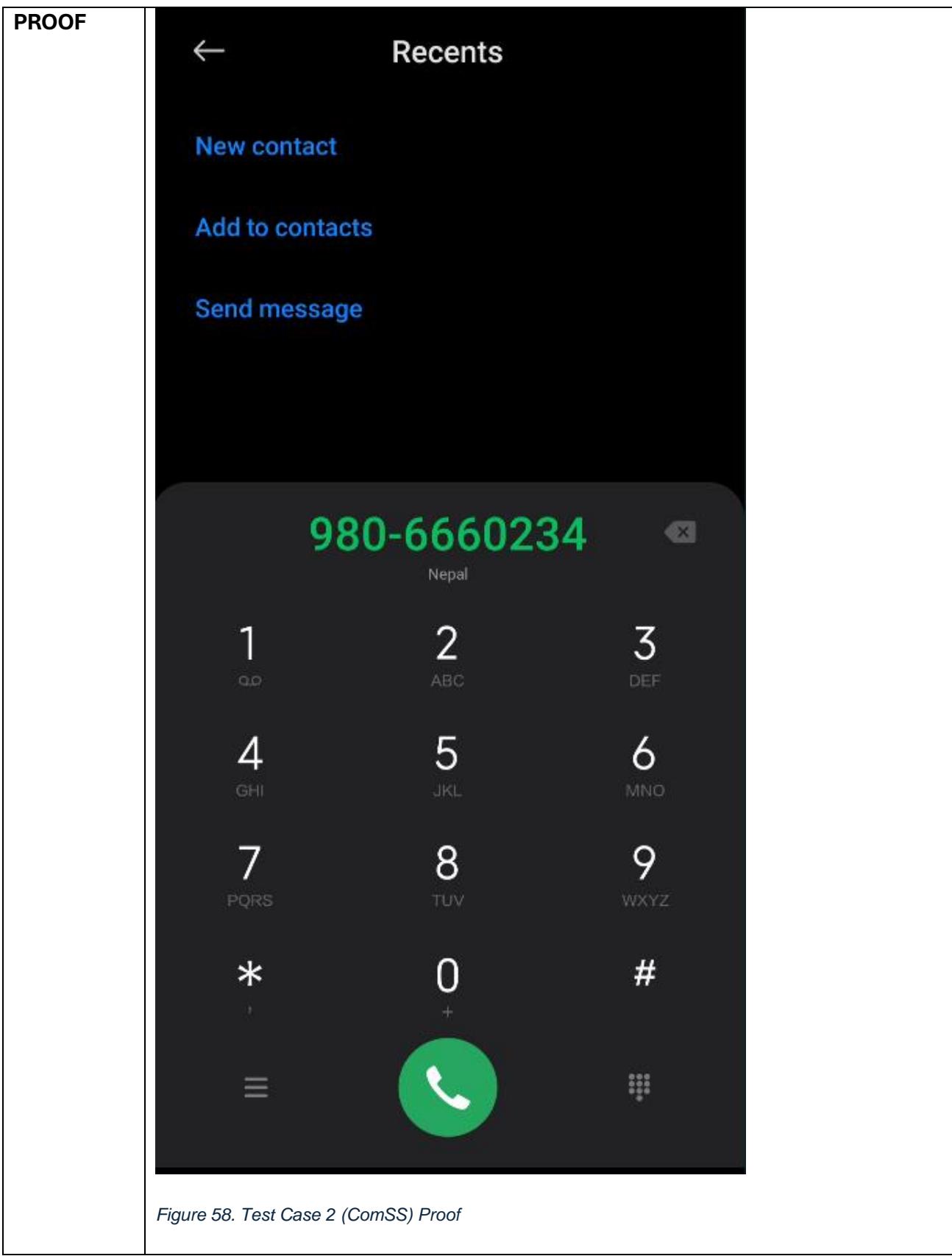


Table 18. Test Case 2 (ComSS)

Test Case 3	
Objective	Test the ability of users and agents to exchange text messages in real-time.
Action	<ul style="list-style-type: none"> • Log in • ChatPage • Text
Expected Result	After logging in as and navigating to the chat page. Agents and users should be able to exchange text messages in real-time.
Actual Result	After logging in as and navigating to the chat page. Agents and users were able to exchange text messages in real-time.
RESULT	PASS

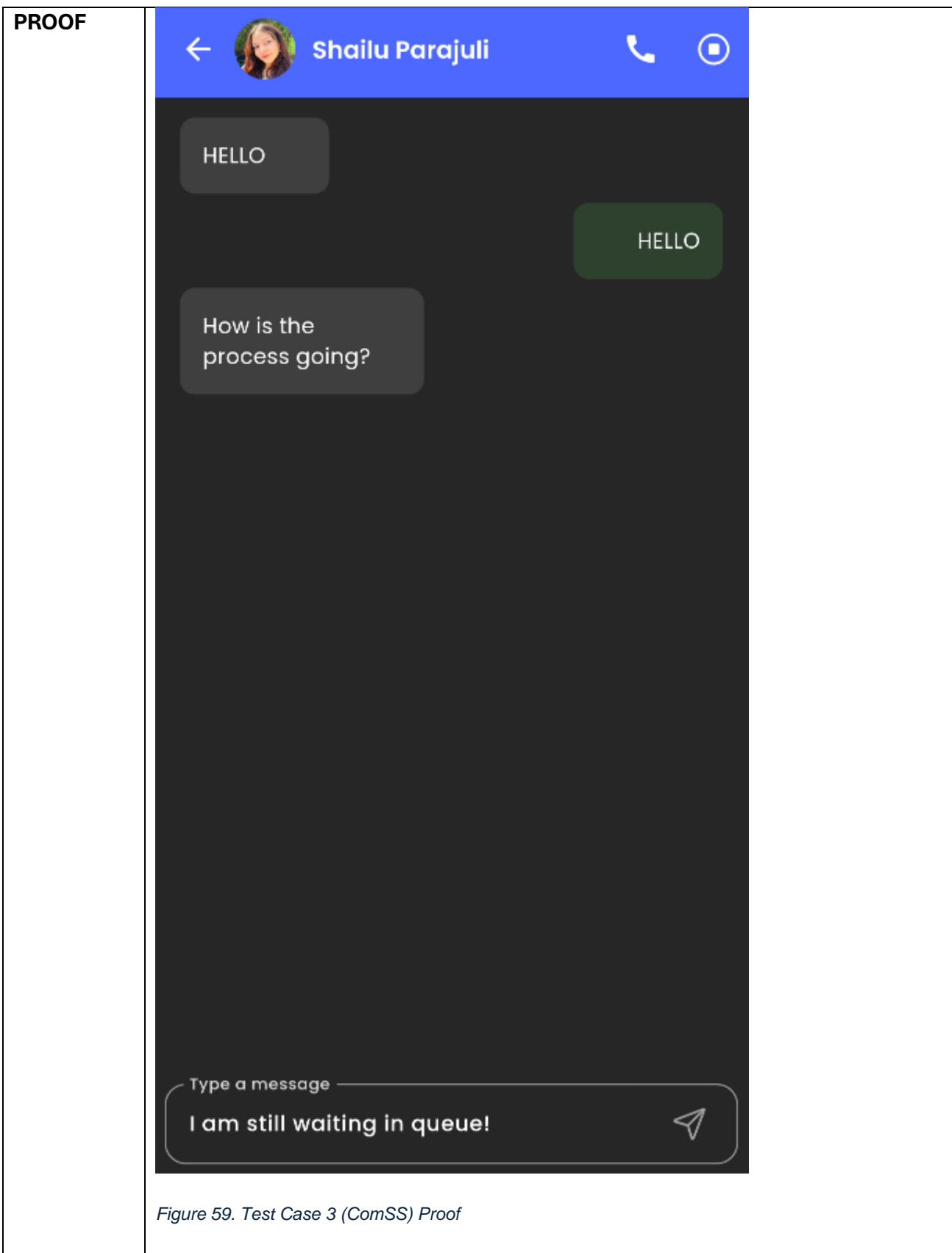


Figure 59. Test Case 3 (ComSS) Proof

Table 19. Test Case 3 (ComSS)

3.6.5. Payment Subsystem (PSS)

3.6.5.1. Purpose:

The Payment subsystem is designed to facilitate secure processing of payments within the system.

3.6.5.2. Scope:

This subsystem contains functionalities related to payment initiation and processing

3.6.5.3. SRS:

Req. Code	Req. Desc	MoSCoW
PSS-F-1.0	Users shall initiate payments after end of service through the app	Must Have
PSS-NF-1.1	The system shall support payment options like electronic wallets (Khalti)	Should Have
PSS-NF-1.2	For electronic wallet payments, the system shall integrate with the Khalti API	Should Have
PSS-NF-1.3	The system shall maintain a detailed transaction history for both users and agents which will be accessible from their respective dashboards.	Could Have
PSS-UR-1.1	Users shall confirm the payment amount before finalizing the transaction	Should Have

Table 20. Payment Subsystem SRS

3.6.5.4. Activity Diagram:

1. Payment Processing

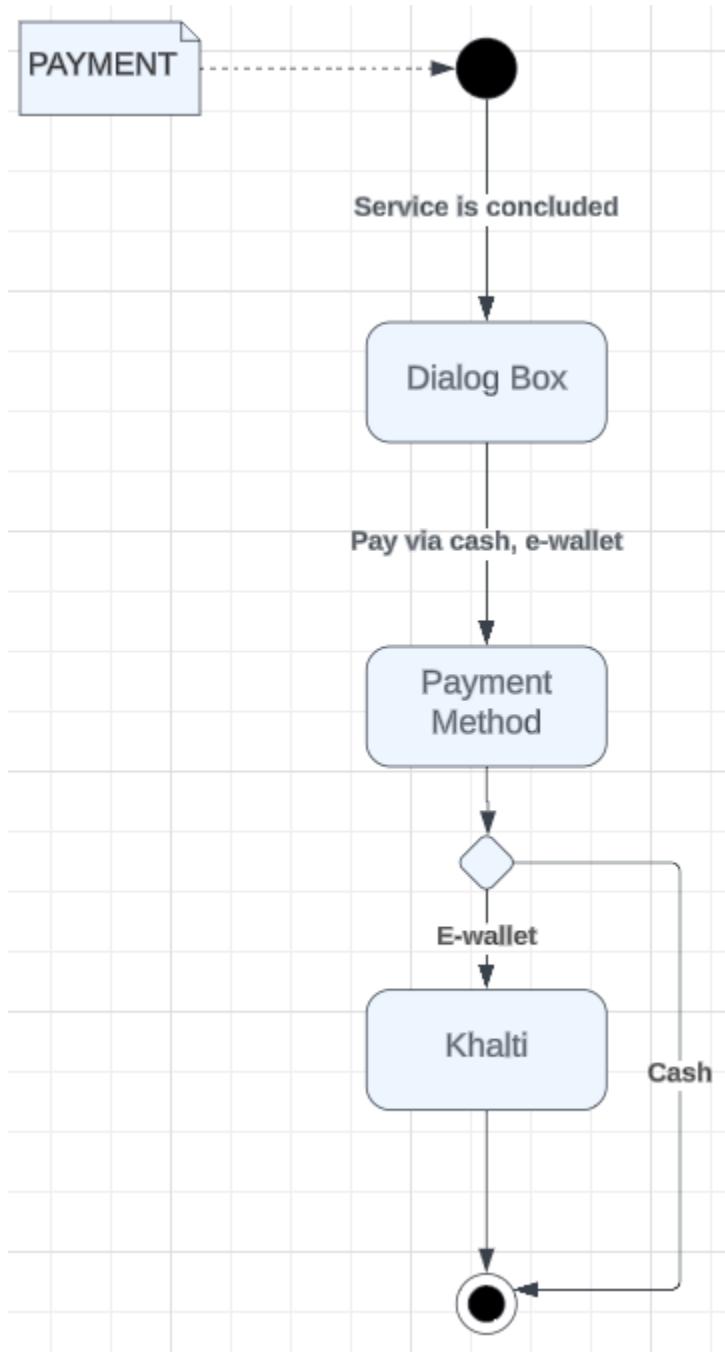


Figure 60. Payment Activity Diagram

3.6.5.5. Wireframe:

1. Payment Initiation

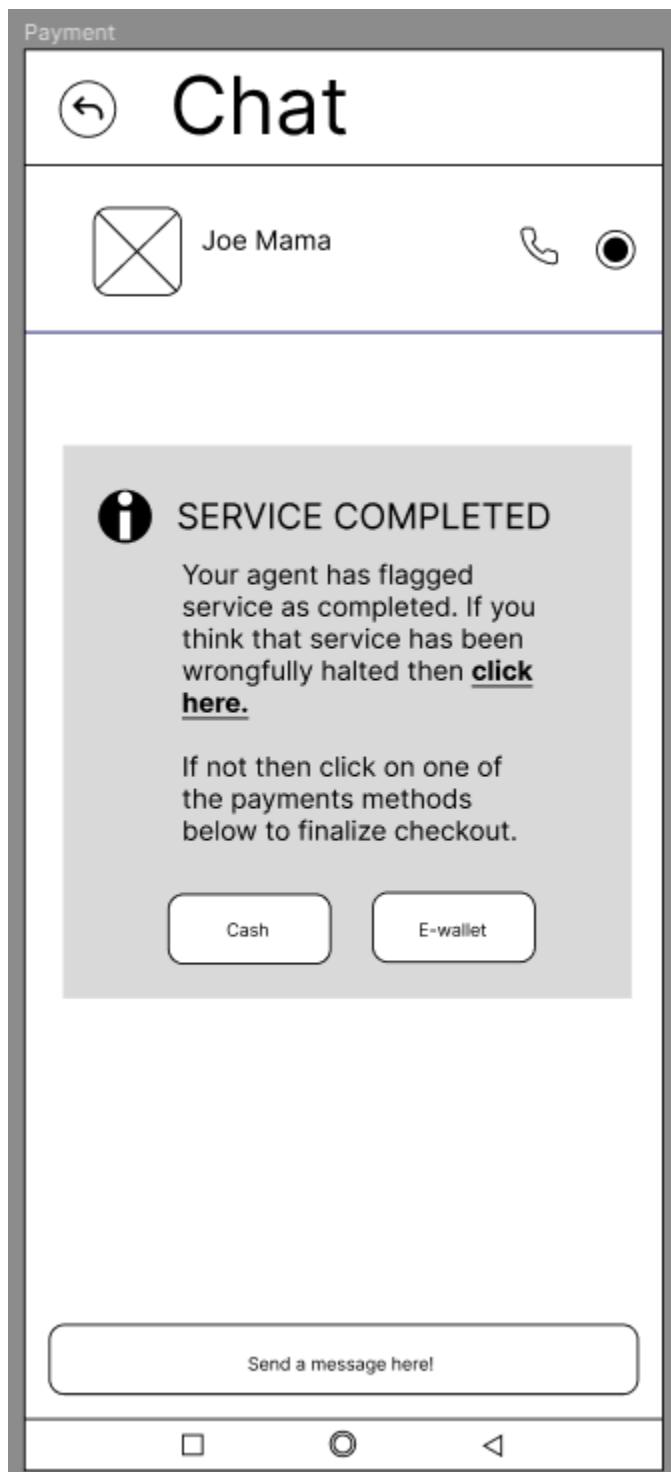


Figure 61. Payment Wireframe

3.6.5.6. Use-Case Diagram:

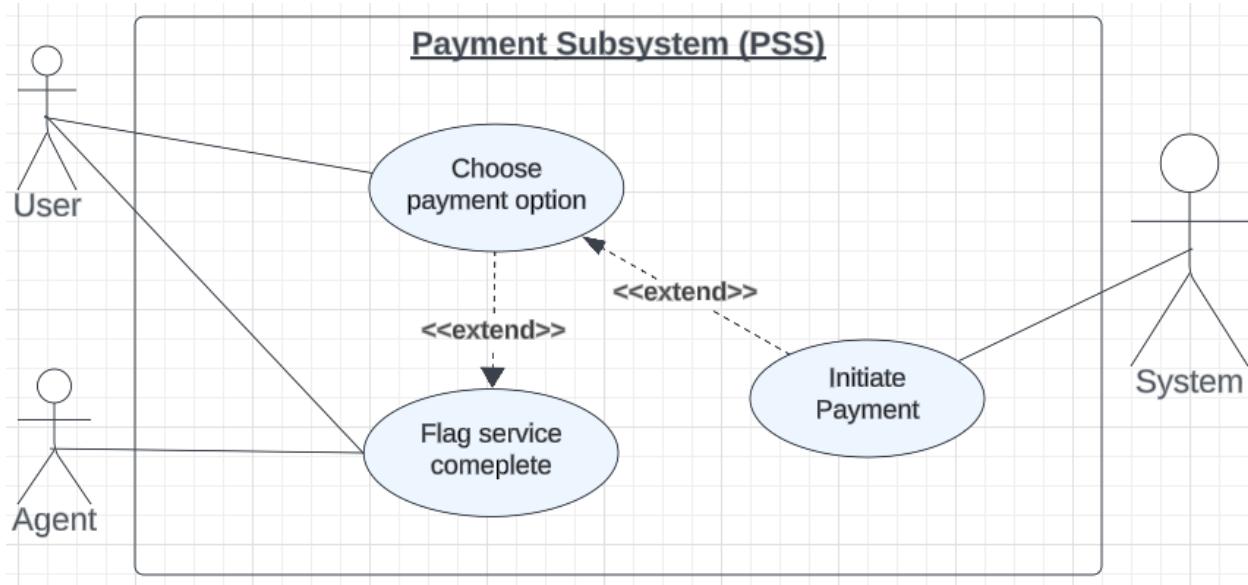


Figure 62. Payment Subsystem Use-Case Diagram

3.6.5.7. ERD:

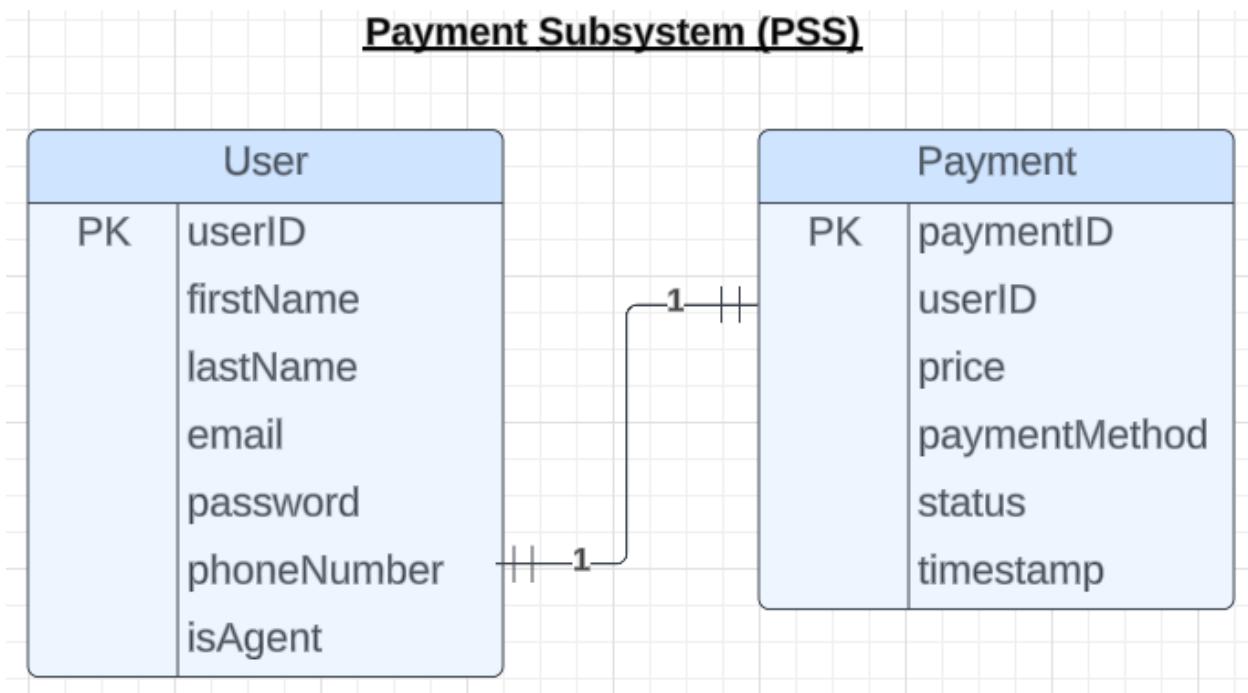


Figure 63. Payment Subsystem ERD

3.6.5.8. Data Dictionary:

Field Name	Data Type	Description
paymentID	String	Unique identifier for each payment
userID	String	User entity making the payment
price	Double	Amount paid for the service
paymentMethod	String	Method used for payment (e.g., cash, electronic wallet).
status	String	Payment status (e.g., Pending, Completed, Failed)
timestamp	Date	Date and time when the payment was made

Table 21. Payment Subsystem Data Dictionary

3.6.5.9. Class Diagram:

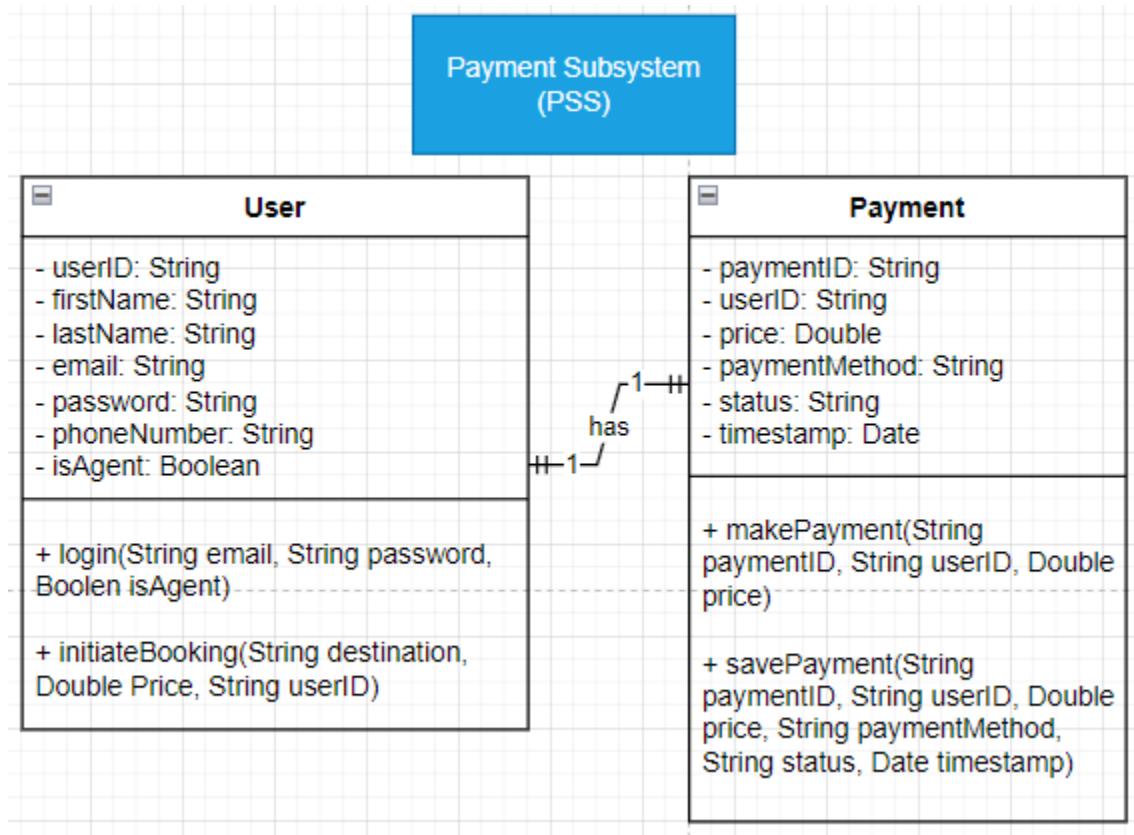


Figure 64. Payment Subsystem Class Diagram

3.6.5.10. Sequence Diagram:

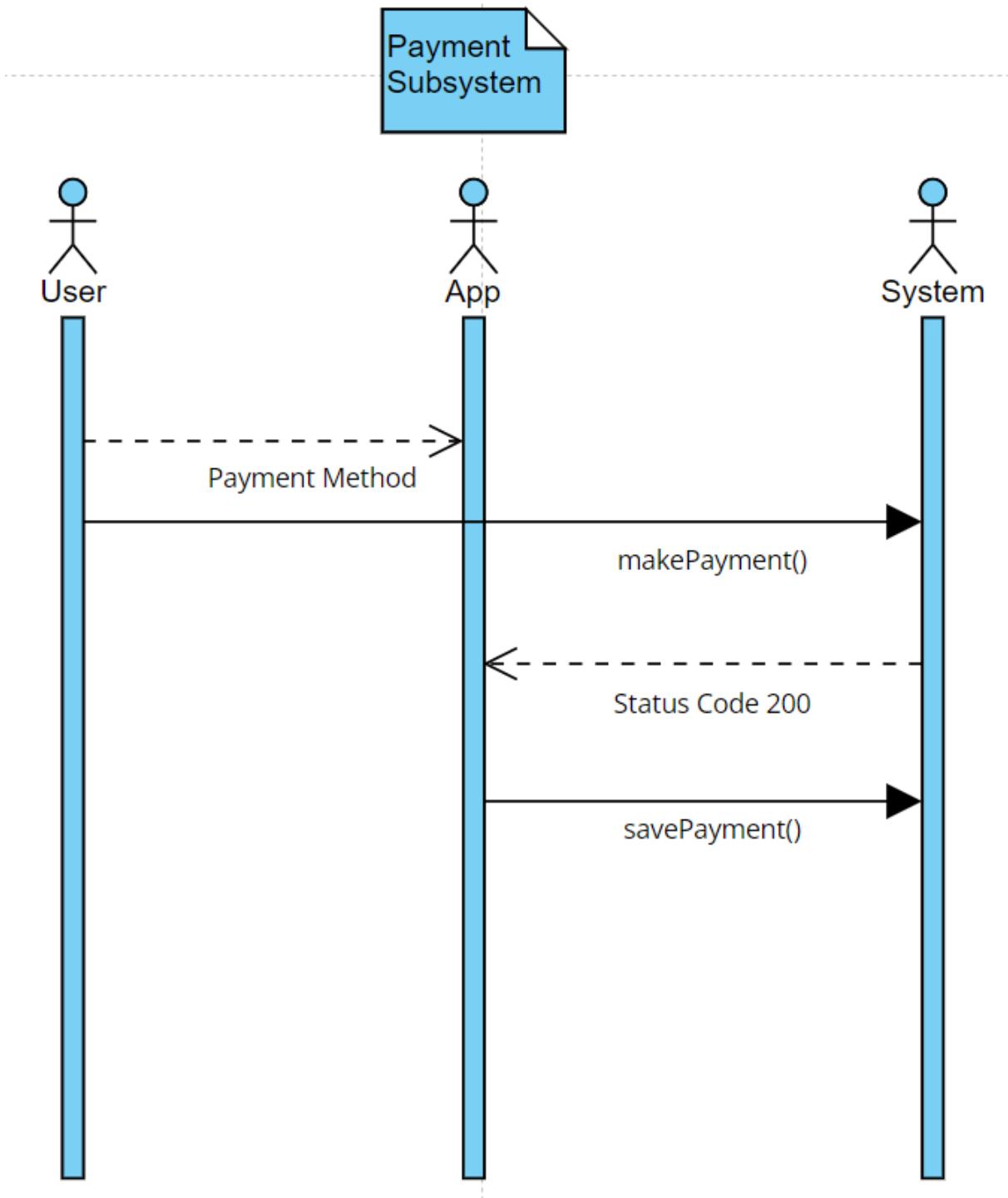


Figure 65. Payment Subsystem Sequence Diagram

3.6.5.11. Test Plan

Objective:

Verify the functionality and reliability of the Payment subsystem.

Scope:

This test plan covers payment initiation and processing.

Test Case 1	
Objective	Test the integration with the Khalti API for electronic wallet transactions
Action	<ul style="list-style-type: none">• Log in• Find Agent• Wait or end the service• In the menu click Khalti• Proceed with the payment
Expected Result	After logging in and finding the agent and the completion of the service, when users click on Khalti as the option Khalti's API should be called and payment should be carried out.
Actual Result	After logging in and finding the agent and ending the service during the payment option users were able to select Khalti and choose it as the payment option and pay for the service using the API.
RESULT	PASS

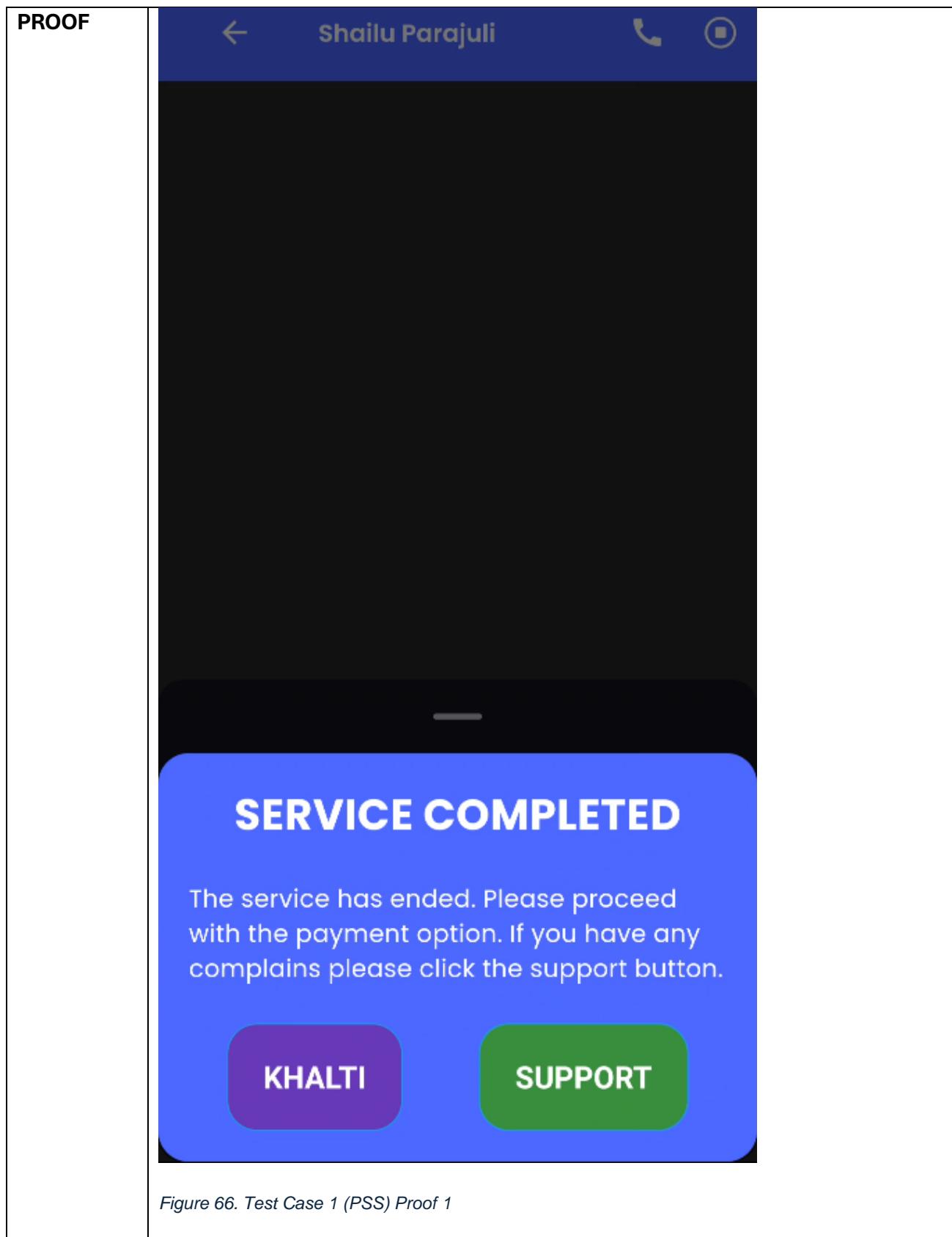


Figure 66. Test Case 1 (PSS) Proof 1

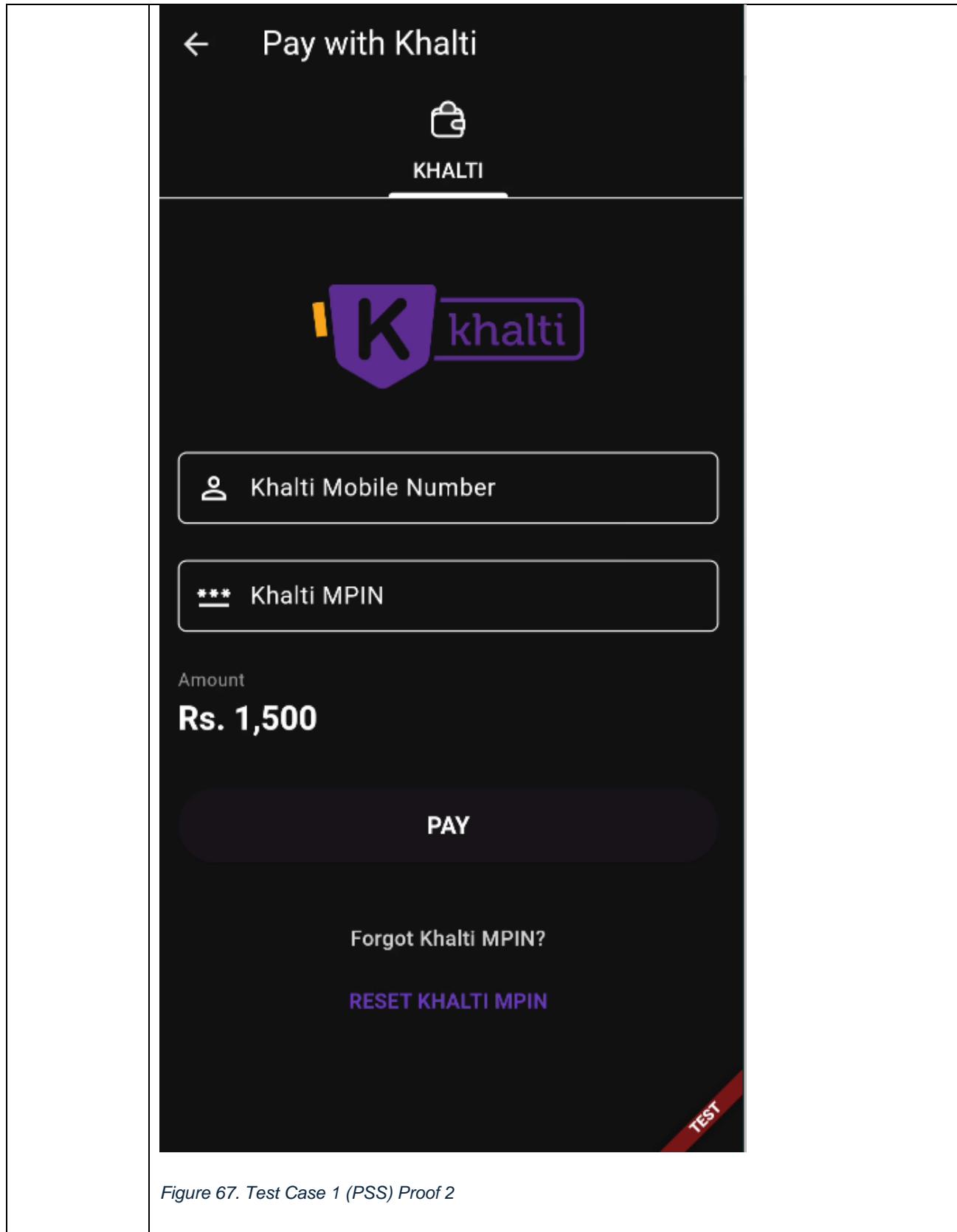


Table 22. Test Case 1 (PSS)

Test Case 2	
Objective	Test the functionality of support button
Action	<ul style="list-style-type: none"> • Log in • Find Agent • Wait or end the service • In the menu click support
Expected Result	After logging in and finding the agent and the completion of the service, when users click on support, QueueEase's support number should be shown in the dialpad.
Actual Result	After logging in and finding the agent and the completion of the service when users clicked on support the QueueEase's support number showed up in the dialpad.
RESULT	PASS

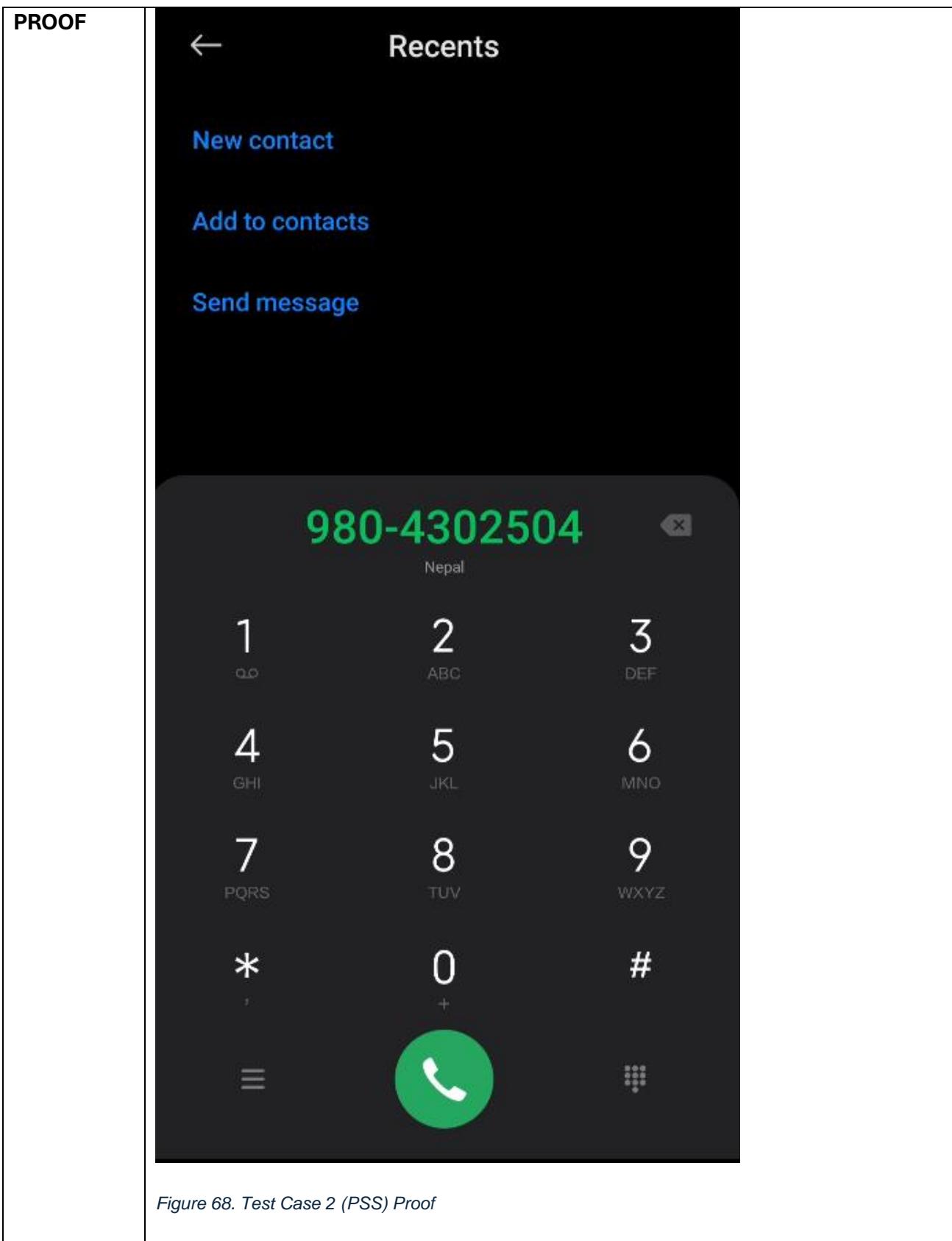


Figure 68. Test Case 2 (PSS) Proof

Table 23. Test Case 2 (PSS)

4. Conclusions

The main aims of the app is to help the citizens complete the government official tasks without needing to stress over it, side by side create employment opportunities for people and finally build an app to cover both the points and make it trustable and safe. After the completion of the app, it is safe to say that people who try it will definitely find it safe and easy to use. The mobile application also answers academic inquiries such as issues regarding user authentication, the app provides encryption techniques to handle user passwords during login and registration. The concerns regarding fraudulent activities and scam users is mitigated by the verification of the users beforehand from the document they provide and the reporting system by contacting support via the app. For booking and subsequent chat room QueueEase employs API calls to connect agents associated with the booking and with the use of real-time notification from Firebase and sockets for chatrooms, with these two technology agents receive real-time notifications and both entities can converse in real-time respectively. For payment QueueEase utilizes Khalti API to receive payments which is used to pay agents for their service. Finally, the use of well written code and proper management alongside user-centric design backed up by HCI principles, this app is designed to handle growing users and provide smooth and experience for users.

In summary the proposed mobile application emerges as a ground breaking solution. It revolutionizes official processes in Nepal by providing a platform for users to connect with verified agents which in turn help them to navigate the tedious bureaucratic system efficiently. Built with the help of Flutter for the frontend, Node.js for the backend, Firebase for the notifications and MongoDB for the database, alongside SCRUM methodology, QueueEase makes itself a contender for an ideal application. It not only simplifies the process but also provides user satisfaction and opens up job opportunities. The success of the app lies in its understanding of the user needs, the destination and the quick employment of verified agents to that location and what follows that is triumph.

5. Critical Evaluation

Every functionalities present in the code could not be summed up in the report but still the report encompasses majority of what the app QueueEase is all about. The main motive behind building the app, the pros and cons of related with it, the future scopes, the systems relevant to it which allowed for inspirations, the technologies which were used to build it all are mentioned in the report. When the project was in its initial planning phase many functionalities were mentioned but many were dropped due to complexity and time constraints. With better planning and management it could have been possible in hindsight, I could have utilized my Dashain-Tihar break more effectively by researching more on the technologies used and also during the actual development process focus more on functionalities rather than UI/UX. But just like no one is perfect, no app is perfect, hence even though the submission period is over this app will undergo updates with new features rolling every few months to ensure all the initial envisioned features are implemented and beyond. I plan to take advice from industry experts related to similar field like QueueEase and push it forward to Nepal's market. I think it has a potential in what it's trying to solve, the solution to navigating bureaucratic system. Informal surveys amongst friends and family and neighbors reveal that they'll definitely use it if given a chance therefore I have hope.

Reflecting back on my time with this project it was definitely a roller-coaster of emotions, I acknowledge both my strengths and areas for improvement. I commend myself for the lengths I went to complete this project. I went over every website, resource material, YouTube videos to gain insights and knowledge about the technology and techniques used, I want to believe I left no stone unturned. On the other hand, I found out and will take this time to reflect and improve on my time-management skills and proactively seek assistance when faced with challenges. I also would have to work on my habit of procrastinating, I will now work beforehand and procrastinate after. It was a fun experience, throughout the project I embraced opportunities for learning and growing. I gained knowledge in fields such as project management, research methodologies and software development. Now after conquering this task, I am motivated to apply these newfound skills and insights to my future endeavors, striving for knowledge, skills, improvement and professional development.

6. Evidence of Project Management

6.1. Logs:

Faculty of Science and Engineering School of Mathematics and Computer Science		 UNIVERSITY OF WOLVERHAMPTON
PROJECT MANAGEMENT LOG		
First Name: Nayan	Surname: Raj Khanal	
Student Number: 2227486	Supervisor: Ms. Erin Shakya	
Project Title: On-Demand Queue Service App	Month: March	
What have you done since the last meeting		
<ol style="list-style-type: none">1. Finished professionalism report2. Created http CRUD operator functions3. Created REGEX validator functions4. Created Change Password Screen5. Started learning mongoDB (created sample data in mongosh)		
What do you aim to complete before the next meeting		
<ol style="list-style-type: none">1. Look into MongoDB Compass2. Look into postman3. Work on implementing the HTTP functions in the app4. Create drawer and settings page		
Supervisor comments		
<p>1. Finish frontend and work on backend asap</p>		

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 17/03/2024

Supervisor Signature: Erin Shakya

Date: 17/03/2024

PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: February
What have you done since the last meeting	
<ol style="list-style-type: none">1. Watched and noted flutter courses2. Added App Configuration: Folder structure, Light & Dark Theme, Custom Widgets Theme, HTTP Functions, Validators, Constants & Utility Classes	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Rewrite codes for splash screen and onboarding screen2. Watch and note node courses	
Supervisor comments	
<ol style="list-style-type: none">1. Watch videos and continue the work	

We confirm that the information given in this form is true, complete and accurate.

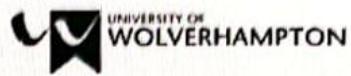
Student Signature: Nayan

Date: 25/02/2024

Supervisor Signature: Erin Shakya

Date: 25/02/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: March
What have you done since the last meeting	
<ol style="list-style-type: none">1. Rewrote codes for splash screen and onboarding screen2. Refactored codes for both splash screen and onboarding screen	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Rewrite and complete the code for User Account Management Subsystem	
Supervisor comments	
<ol style="list-style-type: none">1. Look into AI systems if applicable and try to implement2. Start looking into professionalism report	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 03/03/2024

Supervisor Signature: Erin Shakya

Date: 03/03/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG

First Name: Nayan **Surname:** Raj Khanal

Student Number: 2227486 **Supervisor:** Ms. Erin Shakya

Project Title: On-Demand Queue Service App **Month:** January

What have you done since the last meeting

1. Created initial design in Figma and prototypes (Splash Screen, Welcome Screen, Login & Signup page)
2. Initialized Flutter
3. Set up GitHub repository and created folder structure and started coding
4. Started working on Functional Decomposition Diagram

What do you aim to complete before the next meeting

1. Finish Splash Screen coding
2. Complete design for authentication part
3. Work on FDD for one component (Authentication)

Supervisor comments

1. Secure login, signup page (Regex)
2. Work on FDD first before ERD

We confirm that the information given in this form is true, complete and accurate.

Student Signature: _____ *Nayan*

Date: 21/01/2024

Supervisor Signature: _____ *Erin Shakya*

Date: 21/01/2024

PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: February
What have you done since the last meeting	
<ol style="list-style-type: none">1. Worked on Frontend, complete UI design overhaul.2. Written code for login.3. Completed FDD two components user account management and verification4. Worked on SRS for the above two components	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Complete SRS for user account management component	
Supervisor comments	
<ol style="list-style-type: none">1. Finish SRS	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan Date: 4/2/2024
Supervisor Signature: E.S. Date: 4/2/2024

PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: January
What have you done since the last meeting	
<ol style="list-style-type: none">1. Finished Splash Screen coding2. Completed design for authentication part3. Worked on FDD for user account management and verification and security subsystem	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Translate the UI into code for the login and signup page2. Work on FDD for another component3. Look into SRS	
Supervisor comments	
<ol style="list-style-type: none">1. Work on frontend2. Start working on SRS	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan Date: 28/01/2024

Supervisor Signature: Erin Shakya Date: 28/01/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: February
What have you done since the last meeting	
<ol style="list-style-type: none">Completed SRS and test plan for the whole systemCreated wireframes according to the functional requirementsSet up environmental variables and databaseCreated a workflow to start coding	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">Continue coding for the next subsystem and research accordingly	
Supervisor comments	
<ol style="list-style-type: none">For admin dashboard show database with admin	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 18/2/2024

Supervisor Signature: Erin Shakya

Date: 18/2/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: April
What have you done since the last meeting	
<ol style="list-style-type: none">1. Refined google map and added high accuracy pinpointing2. Looked into sockets and other technology for implementing chat functionality3. Continued with final report	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Add chat feature.2. Look into implementing Khalti as a payment method3. Work on backend of completed subsystems	
Supervisor comments	
<ol style="list-style-type: none">1. Prepare for 75% project defence2. Work on backend	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 14/4/2024

Supervisor Signature: Erin Shakya

Date: 14/4/2024

PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: March
What have you done since the last meeting	
<ol style="list-style-type: none">1. Added welcome, login, signup, profile page2. Started professionalism report	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Implement backend for login and signup2. Finish professionalism report	
Supervisor comments	
<ol style="list-style-type: none">1. Implement security features, like encryption2. Work on professionalism report	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 10/3/2024

Supervisor Signature: Erin Shakya

Date: 10/3/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG

First Name: Nayan **Surname:** Raj Khanal

Student Number: 2227486 **Supervisor:** Ms. Erin Shakya

Project Title: On-Demand Queue Service App **Month:** January

What have you done since the last meeting

1. Created initial design in Figma and prototypes (Splash Screen, Welcome Screen, Login & Signup page)
2. Initialized Flutter
3. Set up GitHub repository and created folder structure and started coding
4. Started working on Functional Decomposition Diagram

What do you aim to complete before the next meeting

1. Finish Splash Screen coding
2. Complete design for authentication part
3. Work on FDD for one component (Authentication)

Supervisor comments

1. Secure login, signup page (Regex)
2. Work on FDD first before ERD

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 21/01/2024

Supervisor Signature: Erin Shakya

Date: 21/01/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: March
What have you done since the last meeting	
<ol style="list-style-type: none">1. Created drawer and settings page2. Researched about postman, thunderclient and MongoDB Compass3. Set up project, network and access in MongoDB4. Set up connection between project and MongoDB5. Created add, delete user and login user controllers	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Create agent-side frontend pages2. Start working on draft report	
Supervisor comments	
<ol style="list-style-type: none">1. Finish frontend work on backend2. Work on draft report	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan Date: 31/3/2024
Supervisor Signature: Erin Shakya Date: 31/3/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: April
What have you done since the last meeting	
<ol style="list-style-type: none">1. Added payment method via Khalti2. Login signup backend implemented	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Add edit profile and agent details backend2. Continue with the report	
Supervisor comments	
<ol style="list-style-type: none">1. Change the pitch black UI colour, use maximum three colours2. Try to show agents location on the map	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan Date: 28/4/2024

Supervisor Signature: Erin Date: 28/4/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**

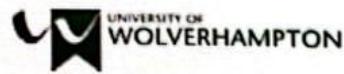


PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: February
What have you done since the last meeting	
<ol style="list-style-type: none">1. Wrote the SRS for user component2. Completed diagrams for user component	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Complete SRS for other components2. Change the SRS into tabular format	
Supervisor comments	
<ol style="list-style-type: none">1. Research on MongoDB and how data is stored2. Create test plans, what I aim to test after I finish coding3. Change the SRS into tabular format	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan Date: 11/2/2024
Supervisor Signature: Erin Shakya Date: 11/2/2024

**Faculty of Science and Engineering
School of Mathematics and Computer Science**



PROJECT MANAGEMENT LOG	
First Name: Nayan	Surname: Raj Khanal
Student Number: 2227486	Supervisor: Ms. Erin Shakya
Project Title: On-Demand Queue Service App	Month: April
What have you done since the last meeting	
<ol style="list-style-type: none">1. Created agent-side fronted pages2. Created booking-side fronted pages3. Completing draft report4. Created homepage with google map API	
What do you aim to complete before the next meeting	
<ol style="list-style-type: none">1. Continue with frontend of chat subsystem2. Refine google map display3. Side-by-side work with backend of completed screens	
Supervisor comments	
<ol style="list-style-type: none">1. In chat subsystem add ability to call2. Continue with draft report	

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Nayan

Date: 07/04/2024

Supervisor Signature: Erin Shakya

Date: 07/04/2024

6.2. Full-Detail Gantt Chart

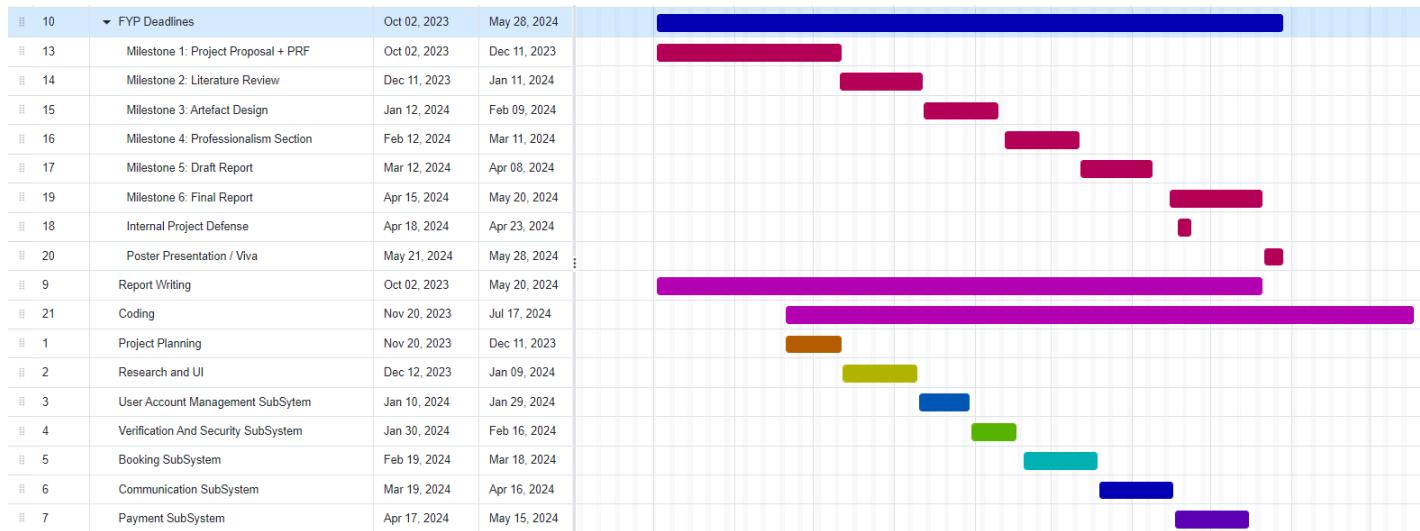


Figure 69. Full-Detail Gantt Chart

7. References and Bibliography

altexsoft, 2022. *The Good and the Bad of Flutter App Development*. [Online]

Available at: <https://www.altexsoft.com/blog/pros-and-cons-of-flutter-app-development/> [Accessed 18 April 2024].

Boog, J., 2022. *What Is An Android Emulator & How Is It Used For Testing?*. [Online]

Available at: <https://theqalead.com/test-management/what-is-an-android-emulator/> [Accessed 18 April 2024].

Chandana, 2024. *Scrum Project Management: Advantages and Disadvantages*. [Online]

Available at: <https://simplilearn.com/scrum-project-management-article> [Accessed 18 April 2024].

Chaudhry, R. et al., 2024. *Fraud Detection and Prevention for a Secure Financial Future Using Artificial Intelligence*. Punjab, IEEE.

Demir, D., 2023. *What is Postman (A Tutorial for Beginners)*. [Online]

Available at: <https://apidog.com/blog/what-is-postman/> [Accessed 18 April 2024].

DevSamurai, L., 2023. *Understanding the MoSCoW prioritization | How to implement it into your project*. [Online]

Available at: <https://community.atlassian.com/t5/App-Central/Understanding-the-MoSCoW-prioritization-How-to-implement-it-into/ba-p/2463999>

[Accessed 18 April 2024].

Genymobile, 2020. *scrcpy*. [Online]

Available at: <https://github.com/Genymobile/scrcpy>

[Accessed 18 April 2024].

GigoDev, 2023. *What is Vscode*. [Online]

Available at: https://dev.to/gigo_dev/what-is-vscode-49fl

[Accessed 18 April 2024].

Jiao, Z., 2022. *Research and Application of Human-Computer Interface Based on User Experience*. Wuhan, IEEE.

Juviler, J., 2024. *What Is GitHub? (And What Is It Used For?)*. [Online]

Available at: <https://blog.hubspot.com/website/what-is-github-used-for>

[Accessed 18 April 2024].

Kulkarni, N., Tanwani, S. & Chaudhari, N., 2021. *Modelling Trust to Preserve End-User Privacy in Location Based Services*. Indore, IEEE.

Mishra, B., 2019. Risks and opportunities on adoption of ride-sharing applications-an exploratory study of Kathmandu. *LBEF Research Journal of Science, Technology and Management*, 3(4), pp. 2705-4683.

Mohamed, K. et al., 2019. An Improved User Authentication Model for Mobile Application Systems: An Expert Review Verification. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1.4), pp. 2278-3091.

Özşahan, H., 2023. *What is Firebase? Learn The Basics From A to Z.* [Online]

Available at: <https://www.resmo.com/blog/what-is-firebase>

[Accessed 18 April 2024].

Park, J., Amendah, E., Lee, Y. & Hyun, H., 2019. M-payment service: Interplay of perceived risk, benefit, and trust in service adoption. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 29(1), pp. 31-43.

Soman, S. et al., 2020. *Mobile-Augmented Smart Queue Management System for Hospitals*. Noida, IEEE.

Sufiyan, T., 2023. *What is Node.js: A Comprehensive Guide.* [Online]

Available at: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>

[Accessed 18 April 2024].

Taylor, D., 2024. *What is MongoDB? Introduction, Architecture, Features & Example.* [Online]

Available at: <https://www.guru99.com/what-is-mongodb.html>

[Accessed 18 April 2024].

Vadhineni, R., 2023. *Thunder Client CLI — A new way to test APIs inside VS Code.* [Online]

Available at: <https://medium.com/thunderclient/thunder-client-cli-a-new-way-to-test-apis-inside-vscode-d91eb5c71d8e>

[Accessed 18 April 2024].

Wu, Y., Zheng, J. & Guo, M., 2022. *A Novel Mobile Instant Messaging System*. Beijing, IEEE.

8. Appendices

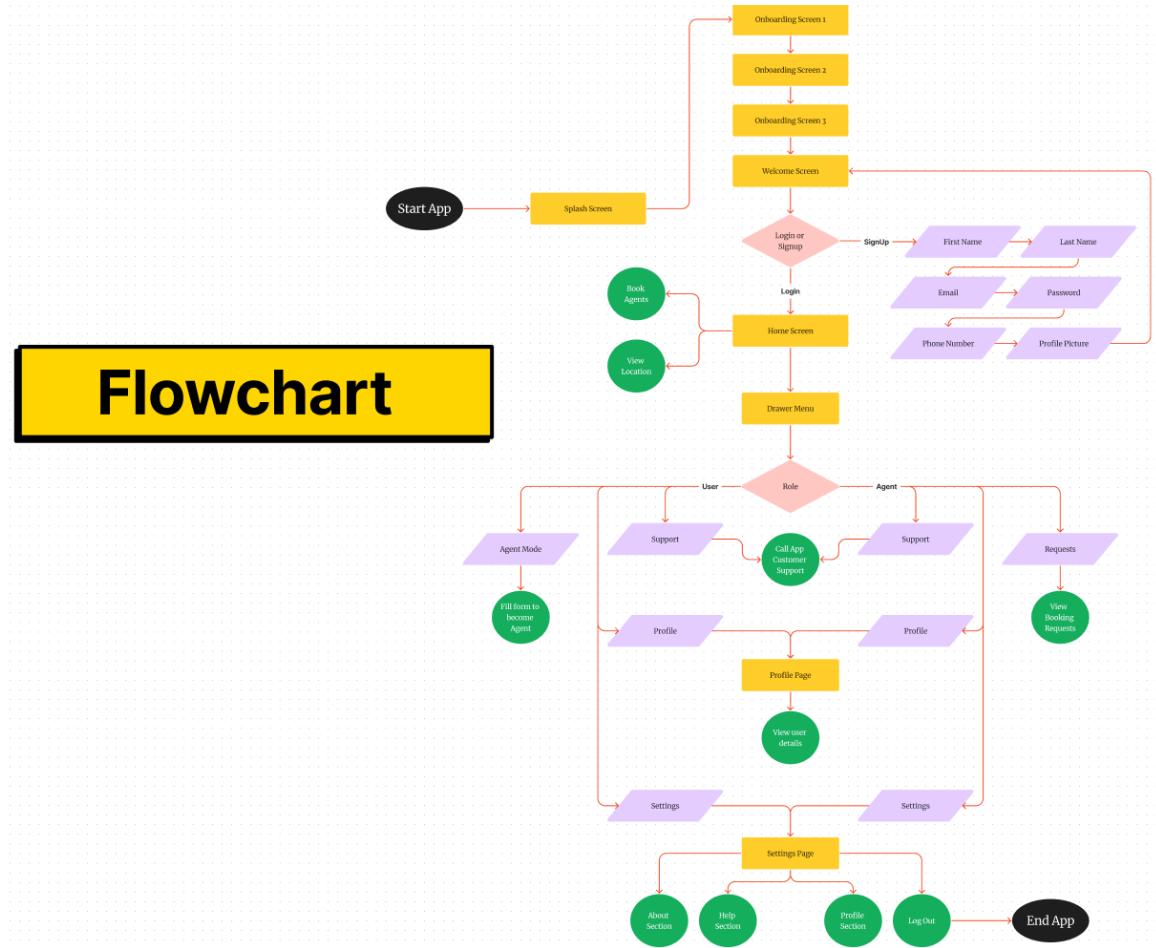


Figure 70. App Flowchart

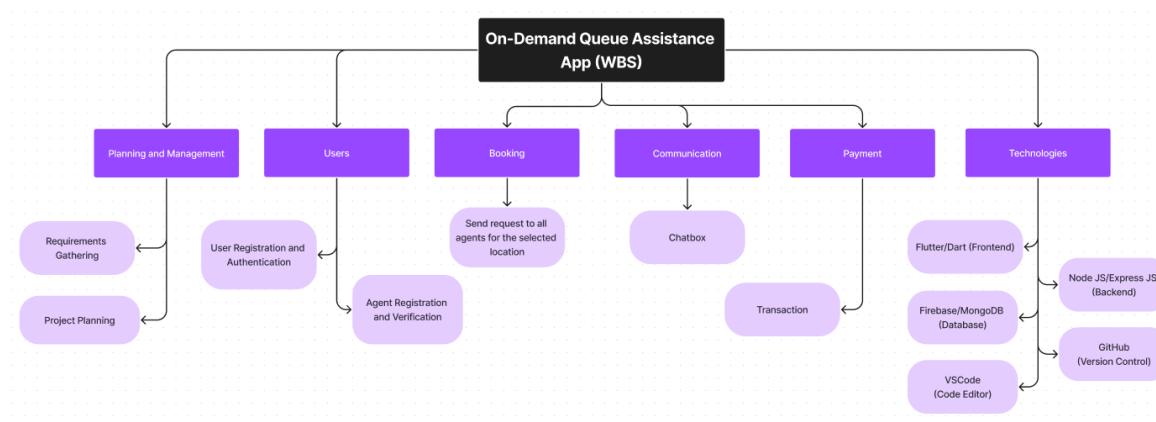


Figure 71. WBS