

Interim Report

Level 2

Appointment Management System

Encrypto

Registration Number	Name
204222R	G.V.G. Uththara
204044X	M.P.A. Disara
204244K	N.D. Jayasinghe
204137K	S.A.R. Nethmini
204190N	B.L.P. Sankalpana

Faculty of Information Technology

University of Moratuwa

2022

Interim Report

Level 2

Appointment Management System

Encrypto

Registration Number	Name
204222R	G.V.G. Uththara
204044X	M.P.A. Disara
204244K	N.D. Jayasinghe
204137K	S.A.R. Nethmini
204190N	B.L.P. Sankalpana

Supervised by:

.....

Mr. S.M.U Premasiri

(Department of Information Technology)

Faculty of Information Technology

University of Moratuwa

2022

Abstract

Booking an appointment online has grown in popularity over the past few years. Many different types of businesses use some type of online appointment management system to help make the appointments setting process more streamlined. An online appointment management system manages the process of scheduling appointments for a company's customers.

When customer wants to repair electronic devices such as laptops, TVs etc., most of the time they have to manually book an appointment. Even if they bring broken devices/items to the respective company/service center, the technician might say that the issue cannot be fixed which leads customers to the frustration. We decided to develop an online appointment management system addressing the above-mentioned issues.

AppoMo is an online appointment scheduling mobile application which allows customers to schedule appointments at their own convenience regarding the issues of their devices/items. And also, we decided to develop a web application for companies/service centers to handle appointments. Both Customers and company/service center admins need to login to the system. Customers have the facility to check whether the issue of their device/item can be fixed before scheduling an appointment which leads to save time and offer quality service. Customers are provided the calendars that indicates free time slots. The system is facilitated with automated appointment reminders and records successful payment confirmations.

To approach a better system, we analyze our customer requirements and design the UML diagrams, EER diagrams before implementation as they help to acquire an overall view of a system. We did a thorough research prior to choosing the technologies. Flutter was chosen as the front-end technology of the mobile app while ReactJS was chosen as the front end of the web application. For backend development, we chose NodeJS. MongoDB and AWS were chosen as the database technology and cloud provider of our system respectively.

This document gives details of the development process of our online appointment management system.

Table of Contents

Chapter 1	1
Appointment Management System.....	1
1.1 Introduction	1
1.2 Background and Motivation.....	1
1.3 Problem in Brief.....	2
1.4 Aim and Objectives.....	2
1.5 Summary	3
Chapter 2	4
Existing solutions for scheduling appointments.....	4
2.1 Introduction	4
2.2 Similar Products	4
2.2.1 Calendly	4
2.2.2 BookAFY	6
2.2.3 Appointlet.....	8
2.2.4 ScheduleOnce.....	10
2.3 Summary	12
Chapter 3	13
Technology Adapted.....	13
3.1 Introduction	13
3.2 Front-end Technologies	13
3.2.1 React Native	14
3.2.3 Bootstrap	15
3.3 Back-end Technologies.....	15
3.3.1 Nodejs.....	15

3.4 Database Technology	16
3.4.1 MongoDB	16
3.5 Cloud Service	16
3.5.1 Amazon Web Service	16
3.6 Summary	17
Chapter 4.....	18
Developing a Mobile App for scheduling appointments	18
4.1 Introduction.....	18
4.2 Software Process Model.....	18
4.3 Users, Input, Output of the system.....	18
4.4 Process	20
4.5 Project Management Plan	21
4.6 Summary	22
Chapter 5	23
Analysis and Design.....	23
5.1 Introduction.....	23
5.2 Analysis.....	23
5.3 Design	23
5.3.1 Use Case Diagram	23
5.3.2 Activity Diagram.....	25
5.3.3 Sequence Diagram.....	36
5.3.4 Class Diagram	47
5.3.5 ER Diagram.....	48
5.4 Summary	48
Chapter 6	49
Implementation	49

6.1 Introduction	49
6.2 Triggers and Implementation	49
6.2.1 Flow Charts	49
.....	49
6.2.2 Pseudo Codes	60
6.2.2.1 Customer Login and Registration	60
6.2.2.2 Customer Profile	61
6.2.2.3 Select a Service Center to submit issue	61
6.2.2.4 Select a company to submit issue	63
6.2.2.5 Issue Submission	63
6.2.2.6 Date and Time Selection	64
6.2.2.7 Admin Login and Registration	65
6.2.2.8 Company/Service Center Profile	66
6.2.2.9 Admin Dashboard	67
6.2.2.10 View Appointment Details	68
6.2.2.11 Repairment Completion	69
6.2.3 Code Segments	70
6.3 Summary	70
Chapter 7	71
Discussion	71
7.1 Introduction	71
7.2 Evaluation and Testing	71
7.3 How our solution differs from others	71
7.4 Further Development	71
7.5 Summary	71
Chapter 8	72
Reference	72
Chapter 9	74

Appendices	74
Appendix A. Individual Contribution	74
Appendix B. Action Plan	79
Appendix C. Mockups	80
Appendix D. Software Requirements Specification	99

List of Figures

Figure 2.2.1 Calendly	4
Figure 2.2.2 Calendly Mobile Interface	5
Figure 2.2.3 Calendly Web Interface for Booking	6
Figure 2.2.4 BookAFY	6
Figure 2.2.5 BookAFY Web Interface for Creating a Meeting	7
Figure 2.2.6 Appointlet	8
Figure 2.2.7 Appointlet Web Interface for Scheduling a Meeting	9
Figure 2.2.8 Appointlet Web Interface for View the Appointments	9
Figure 2.2.9 ScheduleOnce	10
Figure 2.2.10 ScheduleOnce Web Interface for Scheduling a Meeting	11
Figure 2.2.11 ScheduleOnce Web Interface for Scheduling a Meeting	12
Figure 3.2.1 React Native	14
Figure 3.2.2 ReactJS	14
Figure 3.2.3 Bootstrap.....	15
Figure 3.3.1 Nodejs	15
Figure 3.4.1 MongoDB	16
Figure 3.5.1 Amazon Web Service	16
Figure 4.5.1 Trello Board.....	21
Figure 5.3.1 Use Case Diagram	24
Figure 5.3.2 Activity Diagram - Customer Login & Registration and Payment Interface	26
Figure 5.3.3 Activity Diagram - Customer Profile and Date & Time Selection	27
Figure 5.3.4 Activity Diagram - Select a Company to Submit Issue.....	28
Figure 5.3.5 Activity Diagram - Select a Service Center to Submit Issue.....	29
Figure 5.3.6 Activity Diagram - Submit the Issue	30

Figure 5.3.7 Activity Diagram - Company/Service Center Admin login & registration.....	32
Figure 5.3.8 Activity Diagram - Add Company/Service Center Details	33
Figure 5.3.9 Activity Diagram - Company/Service Center Dashboard, View Appointment Details and Company/Service Center Profile,.....	34
Figure 5.3.10 Activity Diagram - Repairment Completion	35
Figure 5.3.11 Sequence Diagram - Customer Login and Registration	37
Figure 5.3.12 Sequence Diagram - Customer Profile and Date & Time Selection	38
Figure 5.3.13 Sequence Diagram - Select a Company to Submit Issue	39
Figure 5.3.14 Sequence Diagram - Select a Service Center to Submit Issue	40
Figure 5.3.15 Sequence Diagram - Submit the Issue	41
Figure 5.3.16 Sequence Diagram - Company/Service Center Admin Login & Registration.....	43
Figure 5.3.17 Sequence Diagram - Add Company/Service Center Details	44
Figure 5.3.18 Sequence Diagram - Company/Service Center Dashboard, View Appointment Details and Company/Service Center Profile	45
Figure 5.3.19 Sequence Diagram – Repairment Completion	46
Figure 5.3.20 ER Diagram	48

Figure 6.2.1 Flow Charts - Customer Login and Registration.....	49
Figure 6.2.2 Flow Charts - Customer Profile.....	50
Figure 6.2.3 Flow Charts - Issue Submission	51
Figure 6.2.4 Flow Charts - Date and Time Selection	52
Figure 6.2.5 Flow Charts - Repairment Completion.....	53
Figure 6.2.6 Flow Charts – View Service Center Details	54
Figure 6.2.7 Flow Charts - Admin Login and Registration	55
Figure 6.2.8 Flow Charts - Admin Dashboard.....	56
Figure 6.2.9 Flow Charts - View Appointment Details	57
Figure 6.2.10 Flow Charts - Company/Service Center Profile	58
Figure 6.2.11 Flow Charts - View Company Details.....	59
Figure 6.2.12 Implementation - Web Interface Frontend	70

Figure 9.1 Mockups - Customer Login.....	80
Figure 9.2 Mockups - Forgot Password.....	80
Figure 9.3 Mockups - Customer Registration.....	81
Figure 9.4 Mockups - Customer Dashboard	81
Figure 9.5 Mockups - Choose a Company Option	82
Figure 9.6 Mockups - View Company Details	82
Figure 9.7 Mockups - Choose a Service Center Option	83
Figure 9.8 Mockups - View Service Center Details	83
Figure 9.9 Mockups - Issue Submission	84
Figure 9.10 Mockups - Waiting for the Acceptance/Rejection of the Issue	84
Figure 9.11 Mockups - SMS of Issue Acceptance.....	Error! Bookmark not defined.
Figure 9.12 Mockups - SMS of Issue Submission.....	Error! Bookmark not defined.
Figure 9.13 Mockups – Date Selection	Error! Bookmark not defined.
Figure 9.14 Mockups - SMS of Issue Rejection	Error! Bookmark not defined.
Figure 9.15 Mockups – Payment of Advance.....	Error! Bookmark not defined.
Figure 9.16 Mockups – Time Slot Selection	Error! Bookmark not defined.
Figure 9.17 Mockups – Payment Successful Message	88
Figure 9.18 Mockups – SMS of Appointment Confirmation	88
Figure 9.19 Mockups – Customer User Profile	89
Figure 9.20 Mockups – Payment Unsuccessful Message	89
Figure 9.21 Mockups - Update Password	90
Figure 9.22 Mockups – Edit Profile.....	90
Figure 9.23 Mockups - Admin Login	91
Figure 9.24 Mockups - Forgot Password Web	91
Figure 9.25 Mockups - Company/Service Center Registration	92
Figure 9.26 Mockups - Registration Number Verification	92
Figure 9.27 Mockups - Waiting for Registration Number Verification	93
Figure 9.28 Mockups - Registration Number Verification Email	93
Figure 9.29 Mockups - Manage Branch Details	94
Figure 9.30 Mockups - Dashboard.....	94
Figure 9.31 Mockups - View Appointments.....	95

Figure 9.32 Mockups - Appointments	95
Figure 9.33 Mockups - Issue Accept/Reject	96
Figure 9.34 Mockups - Service Completion	96
Figure 9.35 Mockups - Company/Service Center Profile.....	97
Figure 9.36 Mockups - Reset Password.....	97
Figure 9.37 Mockups - Edit Branch Details	98
Figure 9.38 Mockups - Service Termination	98

List of Tables

Table 1.1 Users, Input, Output of the System.....	20
Table 1.2 Action Plan	79

Chapter 1

Appointment Management System

1.1 Introduction

An appointment management system is a software used by companies and service providers to streamline their service appointments. In this ever-evolving business world, the companies are at their top gear in moving at a higher pace competing with their competitors. It is challenging to find a convenient time for both the company and the customer to arrange meetups when both parties have their own fully occupied schedules [1]. By using this app, potential customers can know and choose their preferred appointment time according to the companies and service providers' available time slots easily.

1.2 Background and Motivation

In a traditional setup, booking appointments are made over phone calls and emails and can eat up a lot of time on both sides. For an instance, government offices are essentially hemorrhaging resources by allowing their staff to schedule appointments manually. Knowing that government agencies have hundreds of visitors daily, consider how much time staff spends trying to manually plan out the logistics of calls, text message, and email bookings. Online appointment systems [2] centralize the booking process into one simplified management solution eliminating the chances of double booking and reducing cancellations, thereby allowing staff to serve customers more efficiently. Online appointment scheduling is more than simply coordinating time availability with customers. It's also about managing follow-ups and planning in advance for changes in staff availability to ensure quality service provision.

Apart from that, the customer has to bring the devices to show it to the technician. When it comes to big items such as washing machines, PCs it is really inconvenient to transport them. Even after the customer somehow brings down the devices to the shop, the technician would say that the issue cannot be fixed. In such cases, the customer has to bring the device to another shop which makes him/her frustrated.

Hsenid Mobile, one of the leading software companies in Sri Lanka, is planning to develop an online appointment scheduling software as a type of digital solution that allows customers to book, schedule, or cancel their appointments through a mobile app. This mobile app is supposed to be

built on top of Hsenid Mobile Telco APIs [3].

HSenid Mobile designs and builds state-of-the-art, cutting-edge technology platforms which enable the digital transformation of businesses. They have gained global recognition for providing dynamic Telco grade platforms, combining Telco APIs, Analytics and Signaling for Telecommunication Companies, resulting in greater convenience with better operational efficiency.

1.3 Problem in Brief

As per the status of the current business world, the companies find it difficult to look into every phone call they get, and every email sent by customers requesting to schedule appointments. Apart from that, the customers also find it difficult to find a suitable time slot in scheduling a time preferred by both the company and them over the phone or via email. Especially with regard to the Sri Lanka's situation where queues have become a part of the everyday life, an appointment management system to book and schedule peoples' meetings can be considered essential. Moreover, people find it difficult to carry devices (i.e., washing machines, PCs) to the service centers without even knowing whether the issue can be fixed.

1.4 Aim and Objectives

Aim:

The aim of this project is to develop a system for scheduling appointments with the use of a mobile app and make the customers' life easier.

Objectives:

- Study SQL, React, .net, and other technologies which are used for making the System
- Design activity diagram and SRC document and Database
- Designing and developing a simple Mobile App to manage scheduling of appointments.
- Enabling SMS Based Communication with per message-based charging.
- Implementing efficient ways to data upload/update/display.
- Providing the facility to integrate with current calendar to ease and coordinate appointment bookings.

- Providing the freedom to book appointments at any time and from anywhere, allowing to easily check the available time slots.
- Providing the facility of automated reminders and follow-ups, once the booking date is near.
- Providing the feature to refill any canceled or rescheduled appointments.

1.5 Summary

In the Chapter 1 of this report, we provide a basic introduction to the Mobile Application and Web Application that are expected to be developed. Currently available existing software which address the same problem domain are discussed throughout the Chapter 2. Chapter 3 provides a basic understanding of the technologies adapted in developing the Mobile Application along with the Web Application. Chapter 4 addresses how you adopt the technology to develop the Mobile Application and Web Application. Chapter 5 contains the analysis and design of the proposed system. Implementation details are discussed under Chapter 6. Chapter 7, Chapter 8, Chapter 9 consists of discussion, references, and appendixes respectively.

Chapter 2

Existing solutions for scheduling appointments

2.1 Introduction

This chapter focuses on documenting the various approaches to addressing the same underlying problem for which we have chosen to offer a solution. People with busy schedules always try to use their time efficiently, and appointment management is one of the best ways to do so. Various industries require appointment management assistance, and they typically require some industry-specific features in addition to the basic features. As a result, many web and mobile applications are available to meet those requirements.

2.2 Similar Products

The software listed below can be identified as an important solution that directly addresses the above issue.

1. Calendly
2. BookAFY
3. Appointlet
4. ScheduleOnce

2.2.1 Calendly



Figure 2.2.1 Calendly

Calendly is one of the most user-friendly online schedulers. Simply create your calendar rules and share the link to your scheduling page with clients, and they can begin scheduling appointments. They will only see the times you are available, as well as the length and type of meeting you prefer. Calendly automates administrative tasks such as sending reminder emails and follow-ups, allowing you to focus on the work that grows your business and keeps customers coming back for more [4].

Features:

- Calendly is a great way to book appointments for individual professionals such as recruiters, teachers, salespeople, customer success professionals, and entrepreneurs.
- Useful for scheduling one-on-one meetings as well as determining collective availability. It also accounts for users' various time zones, ensuring that all meeting participants arrive on time.
- Integrates with all major calendar tools across devices, and its Zapier integrations with Zoom, Salesforce, and hundreds of others provide unparalleled business efficiency.
- Creating buffers between appointments to avoid overlap.
- Setting daily meeting limits [5]
- Emailing and texting confirmation and reminders

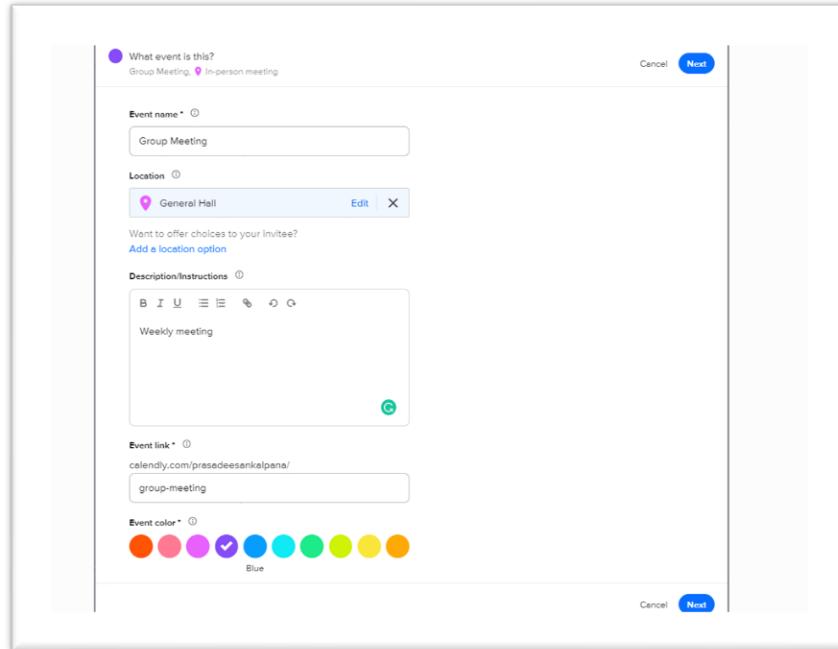


Figure 2.2.2 Calendly Mobile Interface

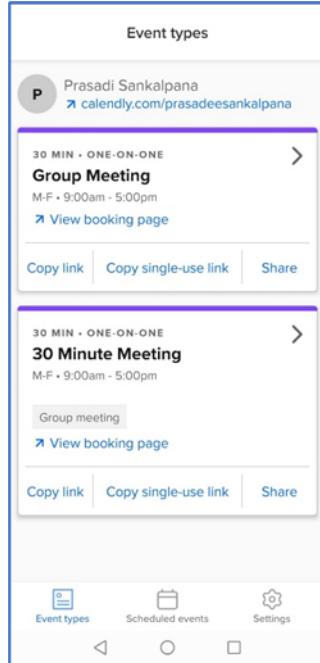


Figure 2.2.3 Calendly Web Interface for Booking

Calendly has both web and mobile applications, but the mobile app can only display available events; scheduling must always be done through a web browser. That means when we click on "View booking page" on the mobile app, it directs us to the web browser. This application allows users to schedule online and physical meetings, but it is primarily used to manage appointments within a company, such as meetings with co-workers. So that each individual is aware of the meeting times.

2.2.2 BookAFY



Figure 2.2.4 BookAFY

The way BookAFY handles calendars between people is one of the features that distinguishes it from other booking tools. When you want each party to be able to see the schedule of the other, you can enable two-way calendar syncing. Alternatively, use one-way syncing when someone

needs to see your calendar, but you don't need to see theirs. It enables you to schedule one-on-one meetings, multiple staff meetings with a single customer, group events, and even one-time and recurring appointments [4].

Features:

- BookAFY generates a unique link for a Zoom or GoToMeeting conference for each scheduled meeting.
- Custom fonts, colors, and messaging options can be applied to your scheduling page.
- You have complete control over all meeting details by configuring the following parameters, meetings should be scheduled with as much lead time as possible, time constraints on how far in advance someone can schedule time with you, interval between appointments, the percentage of your calendar that is visible to others.

The screenshot shows the BookAFY web interface for creating a meeting. At the top, there's a navigation bar with the BookAFY logo and a progress bar with four steps: 01 Partner Call, 02 Casey, 03 Dec 26th 09:30 PM, and 04 Details. Below the progress bar, there are two columns of input fields. The left column contains 'Full Name*', 'Email*', and 'Message'. The right column contains 'Appointment Types: Partner Call', 'Date: 26-12-2022', 'Time: 09:30 PM - 09:45 PM', and 'Time zone: New Delhi Time'. Below these columns is a section titled 'Add Guest' with a note: 'Please enter 1-4 emails below and we'll send them invites. Guest email addresses separated by commas.' A large teal button at the bottom right says 'Confirm appointment'.

Figure 2.2.5 BookAFY Web Interface for Creating a Meeting

This application also primarily intended for use in company meetings, but the user can view the schedules of all staff members and change the schedule as needed by changing the existing schedule [6].

There are web and mobile applications for BookAFy users, but we discovered some issues with the mobile app, such as the login application not working properly.

2.2.3 Appointlet



Figure 2.2.6 Appointlet

Appointlet is yet another scheduling app that allows users to book appointments in a variety of languages. This includes languages such as English, Spanish, German, Dutch, Italian, and others. Businesses can book an unlimited number of different types of meetings for free.

Features:

- Place a link to your scheduling page on your website, in emails, or on various landing pages. When customers schedule appointments with you, Appointlet automatically updates both their and your calendars.
- No worries if you're planning a long-distance meeting. Different time zones are recognized and accounted for by the system.
- Personalized scheduling pages [7]
- Customized availability displays: everyone at once or only selected team members
- Email confirmations and reminders can be customized and sent automatically.

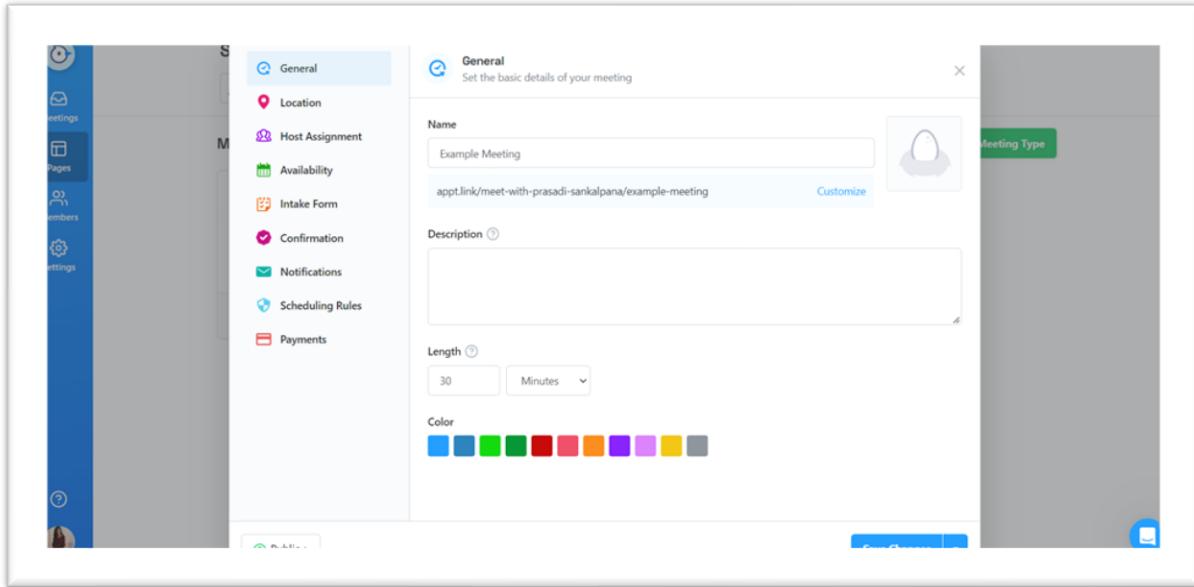


Figure 2.2.7 Appointlet Web Interface for Scheduling a Meeting

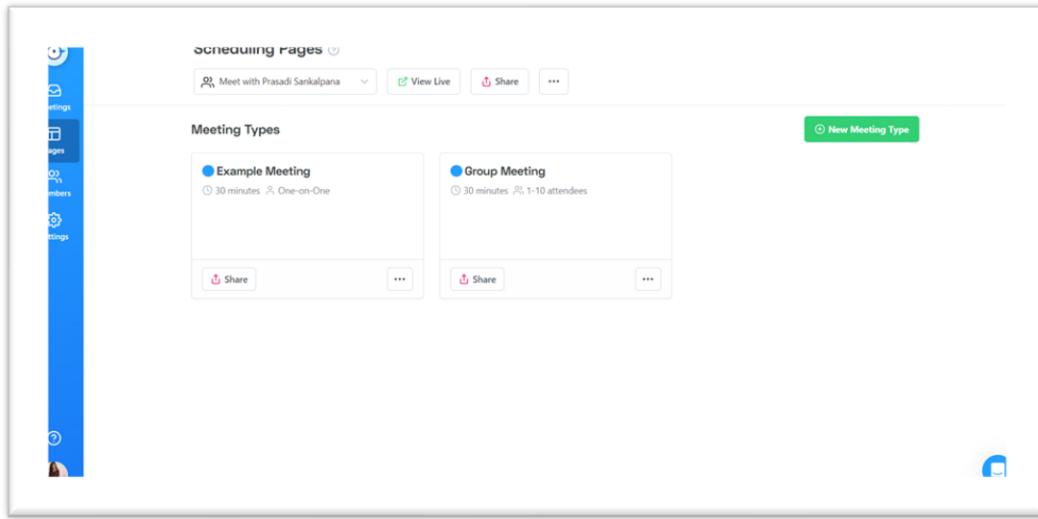


Figure 2.2.8 Appointlet Web Interface for View the Appointments

This can be used for any type of appointment management and is compatible with a variety of industries. Appointlet does not have a mobile app, but it can be accessed via a web browser on any device.

2.2.4 ScheduleOnce



Figure 2.2.9 ScheduleOnce

ScheduleOnce is a product of OnceHub, Inc. It differs in that it does not make use of the standard calendar feature. Many schedulers, especially free ones, only do this [8].

ScheduleOnce makes use of the power of other well-known tools such as Office 365 and Google Calendar. The most significant advantage of this approach is that it integrates all of the appointment scheduling components.

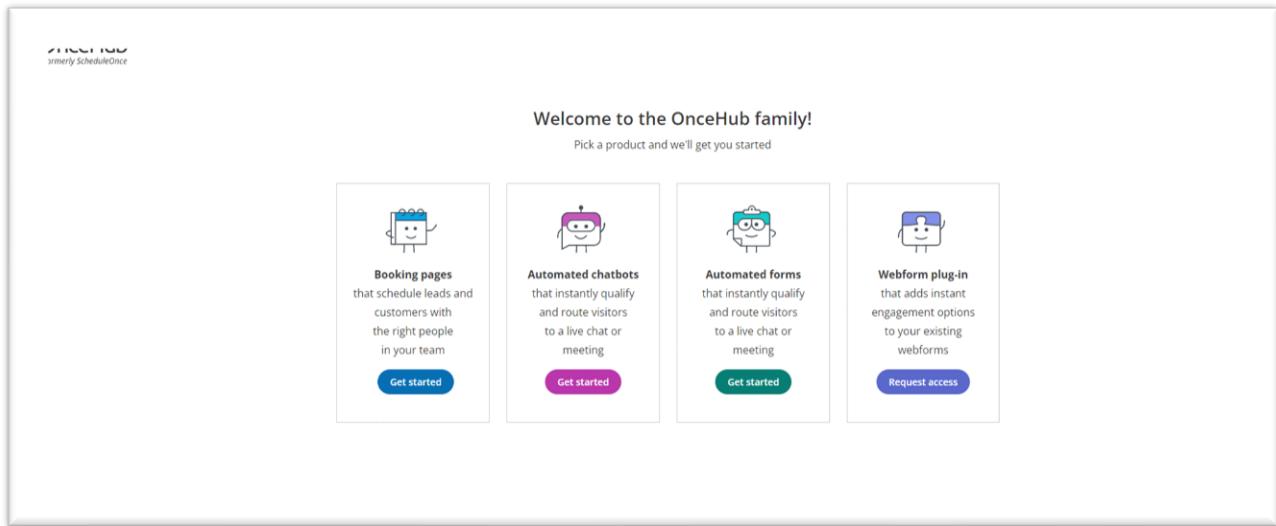


Figure 2.2.10 OnceHub Home Page

Features:

- Team managers can schedule meetings using the automatic calendar booking feature. Users can configure booking methods, reassign tasks, and choose the type of meeting. These appointments are then directly transferred to the employees' calendars.
- Schedule Once allows individual employees, teams, and entire departments to schedule independent or joint meetings.

- You decide whether to distribute client and prospect booking requests in a round-robin or priority-based manner. Your specific rescheduling and cancellation policies are also determined and incorporated into your scheduling page.
- Personalized scheduling pages and correspondence, Custom notifications, Resource pools based on department, skill, territory, or other user-specified criteria, Lead distribution and monitoring are both customizable.

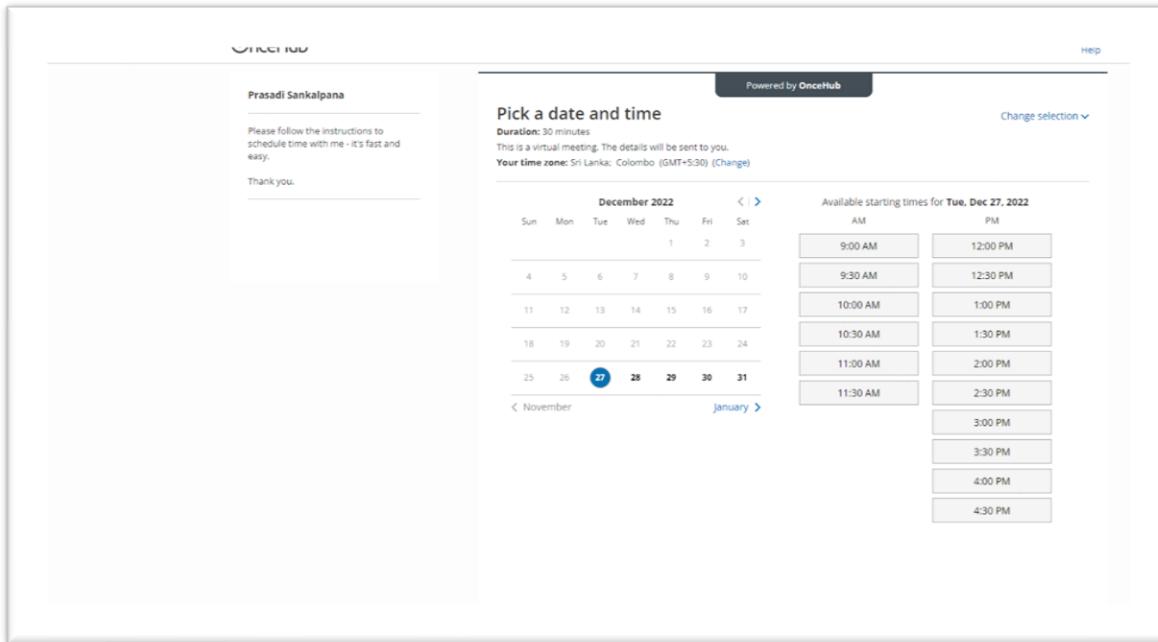


Figure 2.2.10 ScheduleOnce Web Interface for Scheduling a Meeting

Figure 2.2.11 ScheduleOnce Web Interface for Scheduling a Meeting

ScheduleOnce has some constraints. After all, its functionality is dependent on integration with other applications [8]. ScheduleOnce does not have a mobile app, and the application primarily focuses on online customer service.

2.3 Summary

Throughout this chapter, we discussed currently available software to schedule appointments. According to the research, the majority of the applications have been developed with the intention of being used within a company. The customer service application focuses primarily on online appointments. When we are considering our system, we can illustrate certain specific features

- Develop with the primary goal of assisting clients in repairing their electronic devices by scheduling physical meetings.
- The client can select a repairing center that uses the application for customer service.
- Customers can obtain updates on the progress of repairs.
- Any authorized company or repair shop can connect to the system and increase the number of customers.
- Before confirming the appointment, the company can review the device issue.

In the next chapter, technologies we have adapted for our system, are discussed.

Chapter 3

Technology Adapted

3.1 Introduction

In this chapter, we will discuss the technologies we have adapted to develop our system. Those technologies can be categorized under following sections.

1. Front-end Technologies

- Flutter
- ReactJS
- Bootstrap

2. Back-end Technologies

- Nodejs

3. Database Technology

- MongoDB

4. Cloud Provider

- Amazon Web Service

3.2 Front-end Technologies

The visual aspects of the system that can be seen and experienced by users are front-end. That means front end is the part of the system, end users can interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc. The following front-end technologies have been chosen for our system.

3.2.1 React Native

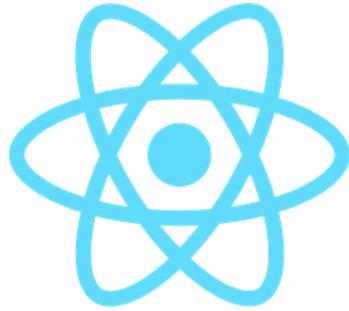


Figure 3.2.1 React Native

React Native is an open-source UI software framework for building native mobile apps based on JavaScript [9]. It provides a slick, smooth and responsive user interface, while significantly reducing load time. Since we develop a Mobile app, we need to develop it for both Android and iOS. Therefore, we chose React Native since it allows developers to develop apps using the same codebase across all operating systems. This is because they use JavaScript to define the entire front-end of their app, which allows for more efficient code sharing. Developers only need to write one set of code and it will run on both iOS and Android. Furthermore, React Native has higher community support because of the ease it provides by incorporating JavaScript.



Figure 3.2.2 ReactJS

ReactJS is a free and open-source front-end JavaScript library for building user interfaces based on UI components [10]. ReactJS can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks. We chose ReactJS because it provides state-of-the-art functionality and is an excellent choice for an easy-to-use and highly productive JavaScript framework. Using ReactJS, we can build complex UI interactions that communicate with the server in record time with JavaScript-driven pages.

3.2.3 Bootstrap



Figure 3.2.3 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development [11]. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Since we develop a web application, we chose Bootstrap because it enables designers and developers to build completely responsive websites quickly.

3.3 Back-end Technologies

The back end is the part of the website users cannot see and interact with. Everything that happens in the background can be attributed to the backend. It is really important to choose the right backend framework for the relevant requirements. The following back-end technologies have been chosen for our system.

3.3.1 Nodejs



Figure 3.3.1 Nodejs

Node.js is a single-threaded, open-source, cross-platform runtime environment for building fast and scalable server-side and networking applications [12]. Node.js has an event-driven architecture and uses the “Single Threaded Event Loop” architecture to handle multiple clients at the same time. We chose Node.js because it can be used for building high-scale applications that need to support multiple concurrent requests. Single-threaded non-blocking I/O makes it an excellent choice for both real-time and data streaming applications.

3.4 Database Technology

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. Database technologies take information and store, organize, and process it in a way that enables users to easily and intuitively go back and find details they are searching for. A good database design should reduce redundant data, provide access with the information it requires to join the information in the tables together as needed and help support and ensure the accuracy and integrity of your information.

3.4.1 MongoDB



Figure 3.4.1 MongoDB

MongoDB is a source-available cross-platform document-oriented database program [13]. Classified as a NoSQL database program. We chose MongoDB because it provides the features like flexibility, flexible query model, native aggregation, and schema-less model. And also, MongoDB is faster and more scalable.

3.5 Cloud Service

Cloud services facilitate the flow of user data from front-end clients, through the internet, to the provider's systems, and back. Cloud services promote the building of cloud-native applications and the flexibility of working in the cloud.

3.5.1 Amazon Web Service



Figure 3.5.1 Amazon Web Service

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing that includes a mixture of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and packaged-software-as-a-service (SaaS) offerings [14]. We chose this to store our database and to host our system.

3.6 Summary

In this chapter, we discuss the technologies which we have adapted to develop our mobile application and web application along with the reasons for choosing those technologies. In the next chapter, we discuss the approach of our proposed solution.

Chapter 4

Developing a Mobile App for scheduling appointments

4.1 Introduction

AppoMo is a mobile/web application which allows customers to make appointments regarding repairs and problems of the items/devices they purchased. AppoMo provides the facility of explaining the issue and submit as a voice message and automated reminders and follow-ups, once the appointment date is near. Company/Service Center admin has the facility of checking the issue prior to confirming the appointment.

4.2 Software Process Model

A software process model is the mechanism of dividing software development work into distinct phases to improve design, product management and project management. We chose Waterfall model as the software process model for our system. Our client requirements are clear, fixed and fully understood. Almost all our requirements remain unchanged. We have a clear and precise structure for the development of our system. With all these reasons, we decided that Waterfall model would be the most suitable software process model for our system.

4.3 Users, Input, Output of the system

The users of the system, their activities. Inputs and outputs are listed in the following table.

User/Role	Activities	
Customer		<ul style="list-style-type: none">➤ Login to the system with username and password➤ Signup to the system if an unregistered customer➤ Select the relevant service center/company from the listed service centers/companies➤ Provide the invoice number of the bill of the item/device he/she purchased from the selected company

		<ul style="list-style-type: none"> ➤ Simply describe the issue regarding their item/device ➤ Make an appointment to hand over their item/device to the company ➤ Pay an advance through their mobile account ➤ Cancel an appointment
	Input	<ul style="list-style-type: none"> ➤ Username/email ➤ Password ➤ Name ➤ Address ➤ Contact number ➤ Invoice number ➤ Issue regarding their item/device
	Output	<ul style="list-style-type: none"> ➤ View free time slots ➤ Facility to make an appointment
Company/Service Center Admin	Activities	<ul style="list-style-type: none"> ➤ Accept/reject the issue ➤ Send a message to the customer accepting/rejecting the issue ➤ Send weekly updates to the customer regarding item/device that is being repaired ➤ Send a message to the customer once the repairing process is finished
	Input	<ul style="list-style-type: none"> ➤ Username/email ➤ Password ➤ Registration number ➤ Company Name ➤ CEO Name ➤ Company/Service Center address ➤ Company/Service Center contact number

		<ul style="list-style-type: none"> ➤ Branch Name ➤ Name of the branch manager ➤ No. of working hours ➤ No. of appointments preferred to accept per hour
	Output	<ul style="list-style-type: none"> ➤ View requests for appointments ➤ View confirmed appointment

Table 1.1 Users, Input, Output of the System

4.4 Process

There are 2 users in this system.

1. Customer
2. Company

For the customer, a mobile application will be developed whereas for the company, a web application will be developed.

Customer

Customer has the facility to create a user profile in order to make appointments regarding repairs and problems of the items they purchased. When creating a user profile, the user has to give their details such as name, address, contact number and email. Before making an appointment, the user has to select the relevant service center/company from the listed service centers/companies from which he is going to make an appointment. After that, the user will be asked to provide the invoice number of the bill of the item/device he/she purchased from the selected company. Once the invoice number is given, it will be checked whether the entered invoice number is valid or not. Only if the invoice number is valid, the details of the purchase and the free time slots will be displayed. First, the user has to simply describe the issue regarding their item/device and send it to the company. Then the company will send a message mentioning whether they accepted the issue. Once the company accepted the issue, the user will be allowed to make an appointment to hand over their item/device to the company to repair. The user has to pay an advance through their mobile account in order to confirm the appointment. Once the appointment is successfully made, the customer will receive a message confirming the appointment. On the appointed date, customer has to visit the company and handover the item that needs to be repaired. The customer

has the facility to cancel an appointment if there's an issue. The customer is facilitated with the feature of automated reminders and follow-ups, once the appointed date is near. And also, they have the facility to see the remaining warranty period of the purchased items. Furthermore, once the item is fully repaired, an SMS message will be sent informing the customer to come and collect the item. The payments will be done physically at the shop.

Company

Once the customer submits the issue, the company has to decide whether to accept the issue considering their capability. If they are capable of fixing the issue, they will send a message to the customer. Once the customer handed over the item, the repairing process will be begun. During the repairing process, the company will send weekly updates to the customer through messages. And once the repairing process is finished, the company will send a message to the relevant customer so he/she can come and collect the item.

4.5 Project Management Plan

We work on our project according to a project management plan to approach a successful final product. It is a formal, approved document that defines how our project is executed, monitored, and controlled. Our project management plan is attached under Appendix B.

We use Trello to assign tasks among the group members, set due dates, share documents.

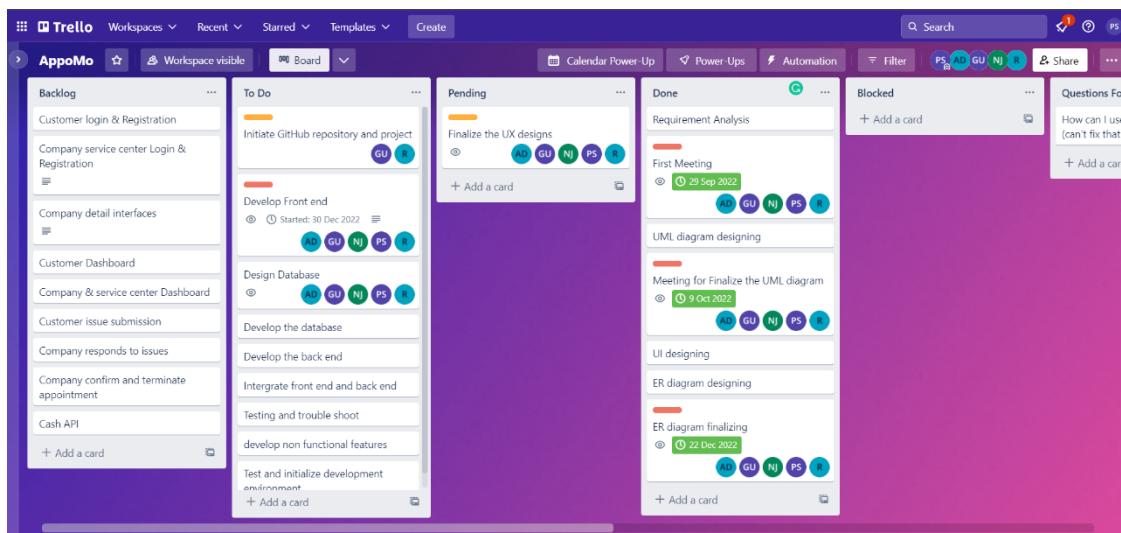


Figure 4.5.1 Trello Board

4.6 Summary

This chapter provides a clear understanding about how we adapt the chosen technologies to solve the identified problem with reference to users, inputs, outputs, process, technology that implements the solution. And also, this chapter provides a justification for why we chose Waterfall model as our software process model for developing proposed system. Details of analysis and design of our proposed solution are discussed throughout the Chapter 5.

Chapter 5

Analysis and Design

5.1 Introduction

User requirements were gathered once we identified the problem. After coming up with a proper solution for identified problem, we started to design some UML diagrams as it is important to have a well-defined and expressive notation to the process of software development.

5.2 Analysis

Object-Oriented analysis is a process that groups items that interact with one another, typically by class, data or behavior, to create a model that accurately represents the intended purpose of the system as a whole. Before designing the proposed system, we have identified the functional and non-functional requirements of the system according to our user requirements.

5.3 Design

The Unified Modeling Language (UML) is a graphical language for Object-Oriented analysis that gives a standard way to write a software system's blueprint. UML diagrams help to acquire an overall view of a system while describing what a system is supposed to do. We have used Lucidchart software tool to draw the diagrams. The following are the UML diagrams we drew.

1. Use Case Diagram
2. Activity Diagram
3. Sequence Diagram
4. Class Diagram
5. ER Diagram

After designing the above-mentioned diagrams, we were able to design the mockups for the system. We used Figma to design mock-ups. The mockups are attached in Appendix C.

5.3.1 Use Case Diagram

A use case diagram is a behavioral UML diagram that describes the sequence of actions a system performs yielding visible results. It shows the interaction of things outside the system with the system itself. Use case diagram helps to understand how a system should behave.

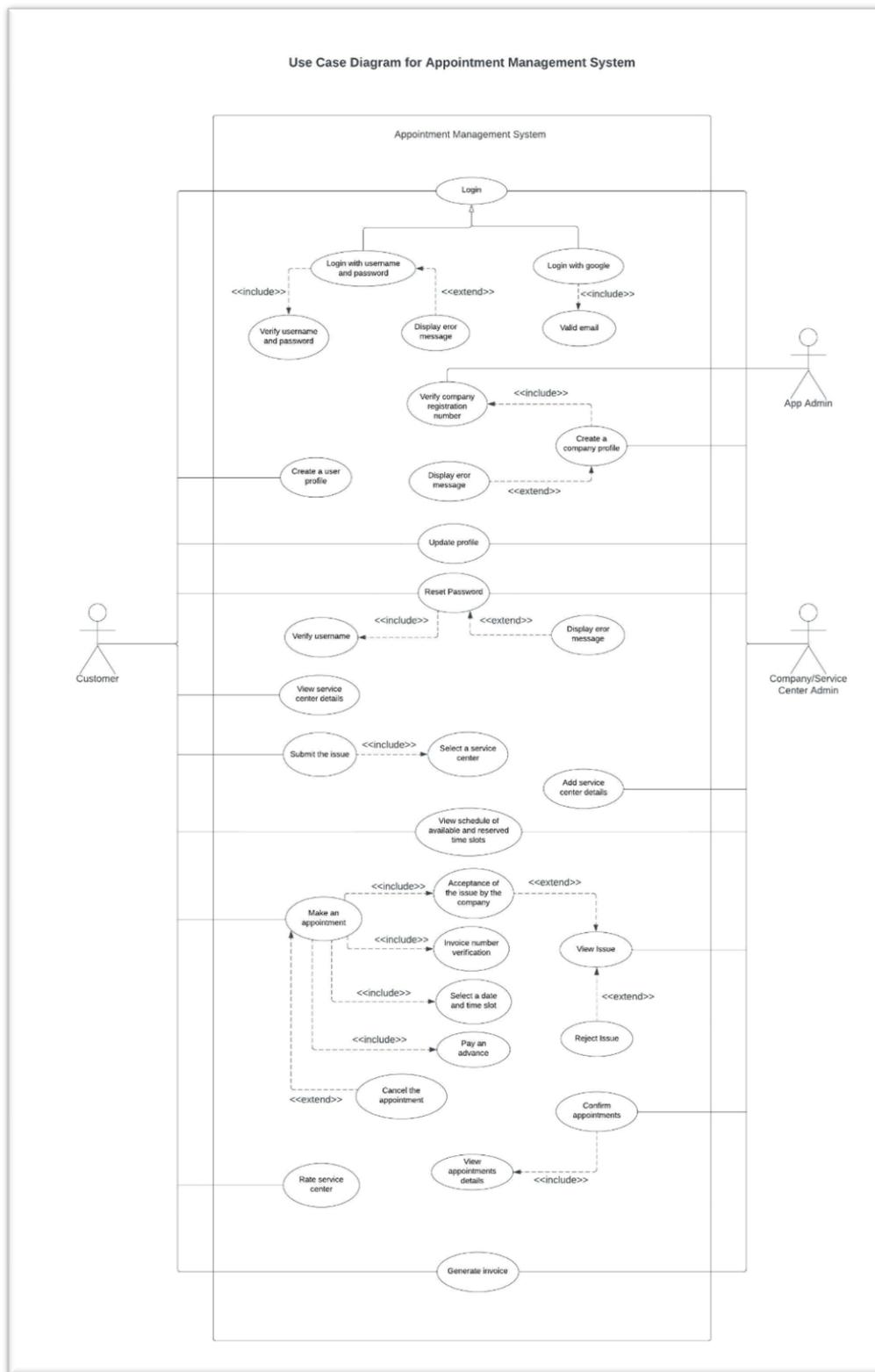


Figure 5.3.1 Use Case Diagram

5.3.2 Activity Diagram

The Activity diagram is a flowchart to represent the flow of control among the activities in a system and consists of activities, states and transitions between activities and states.

5.3.2.1 Activity Diagrams for Mobile Application

We designed several activity diagrams to demonstrate the major activities in our proposed mobile application.

- Customer login and registration
- Customer user profile
- Select a company to submit the issue and rate
- Select a service center to submit the issue and rate
- Submit the issue and appointment confirmation
- Date & time selection
- Payment Interface

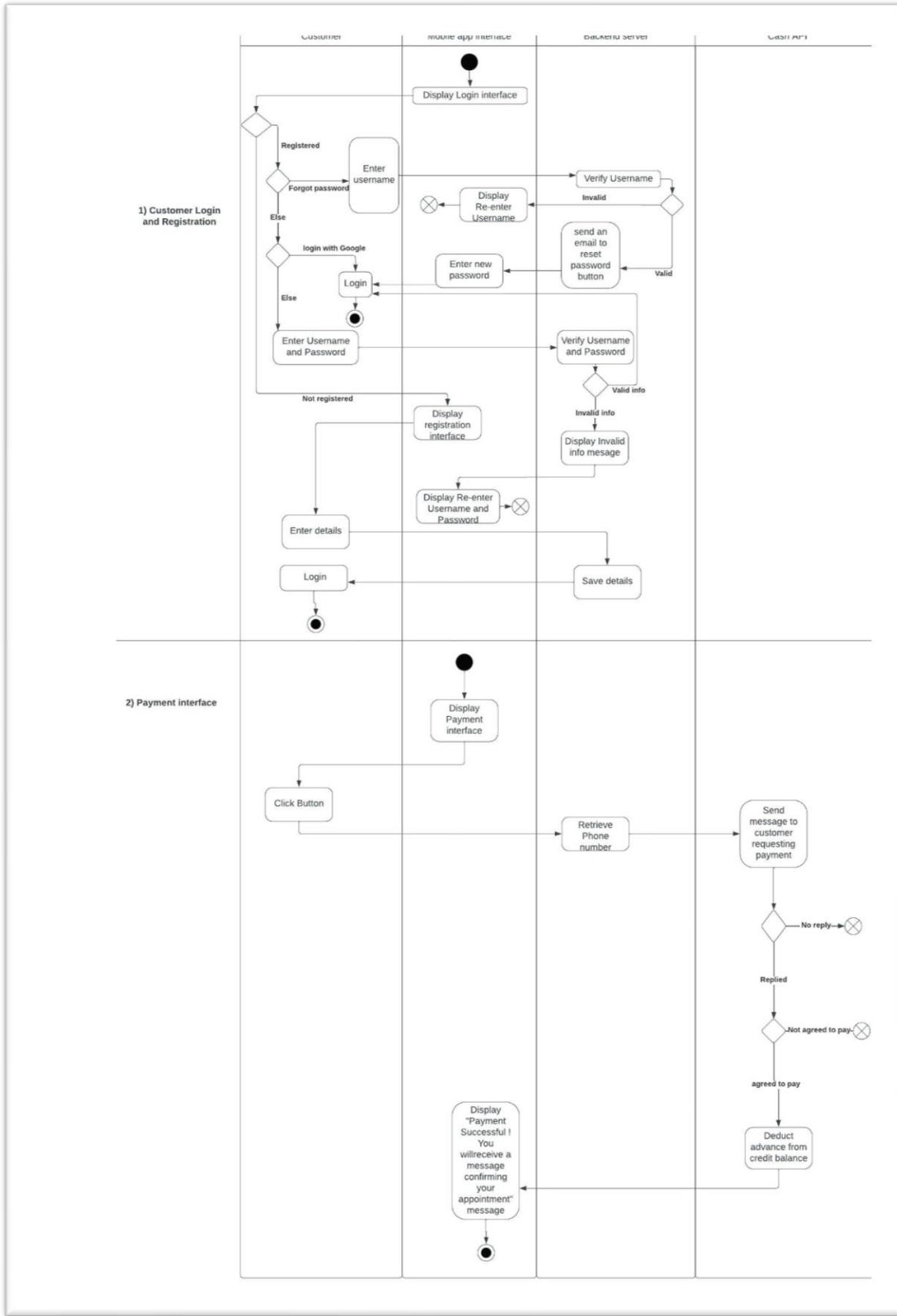


Figure 5.3.2 Activity Diagram - Customer Login & Registration and Payment Interface

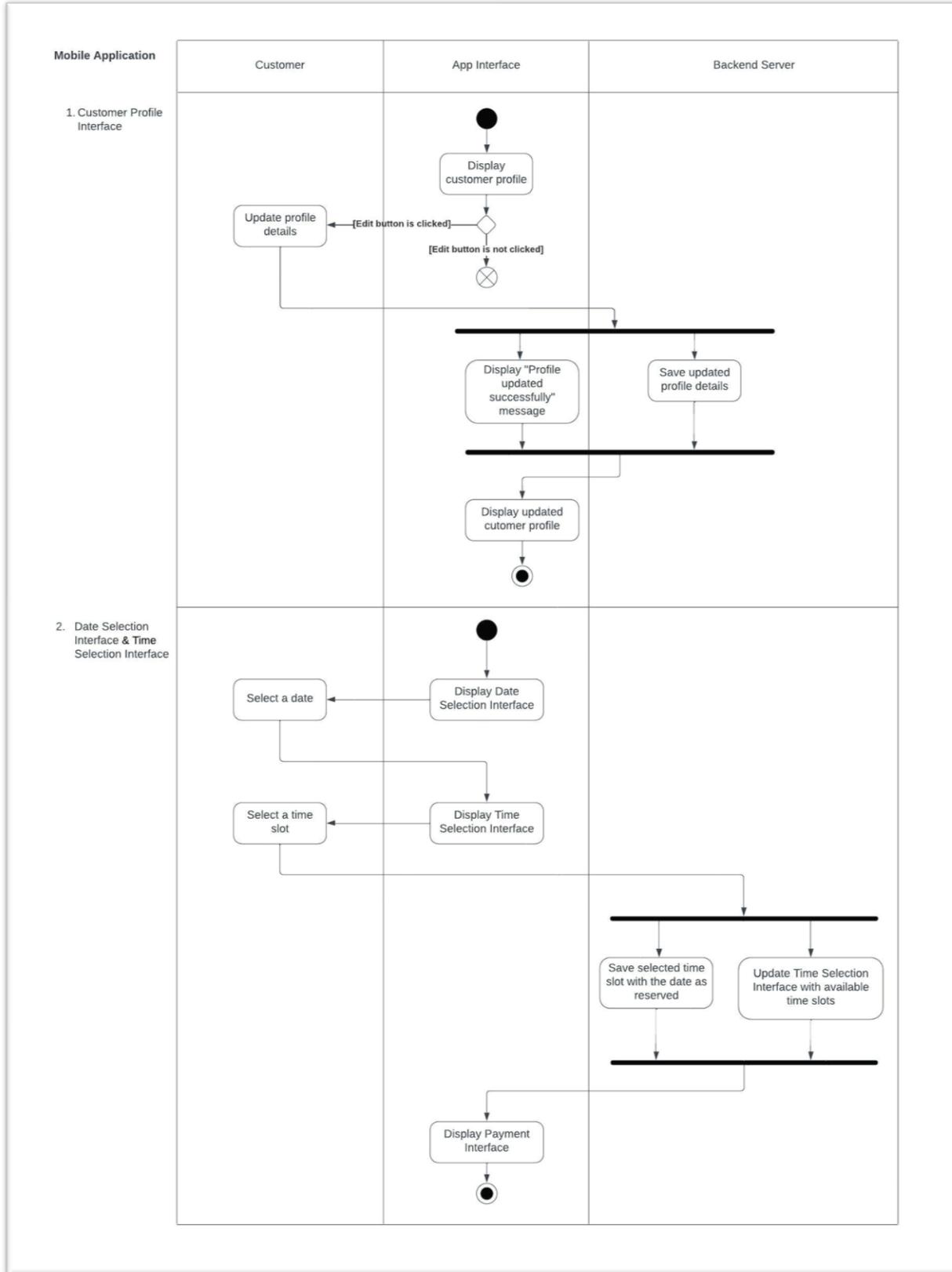


Figure 5.3.3 Activity Diagram - Customer Profile and Date & Time Selection

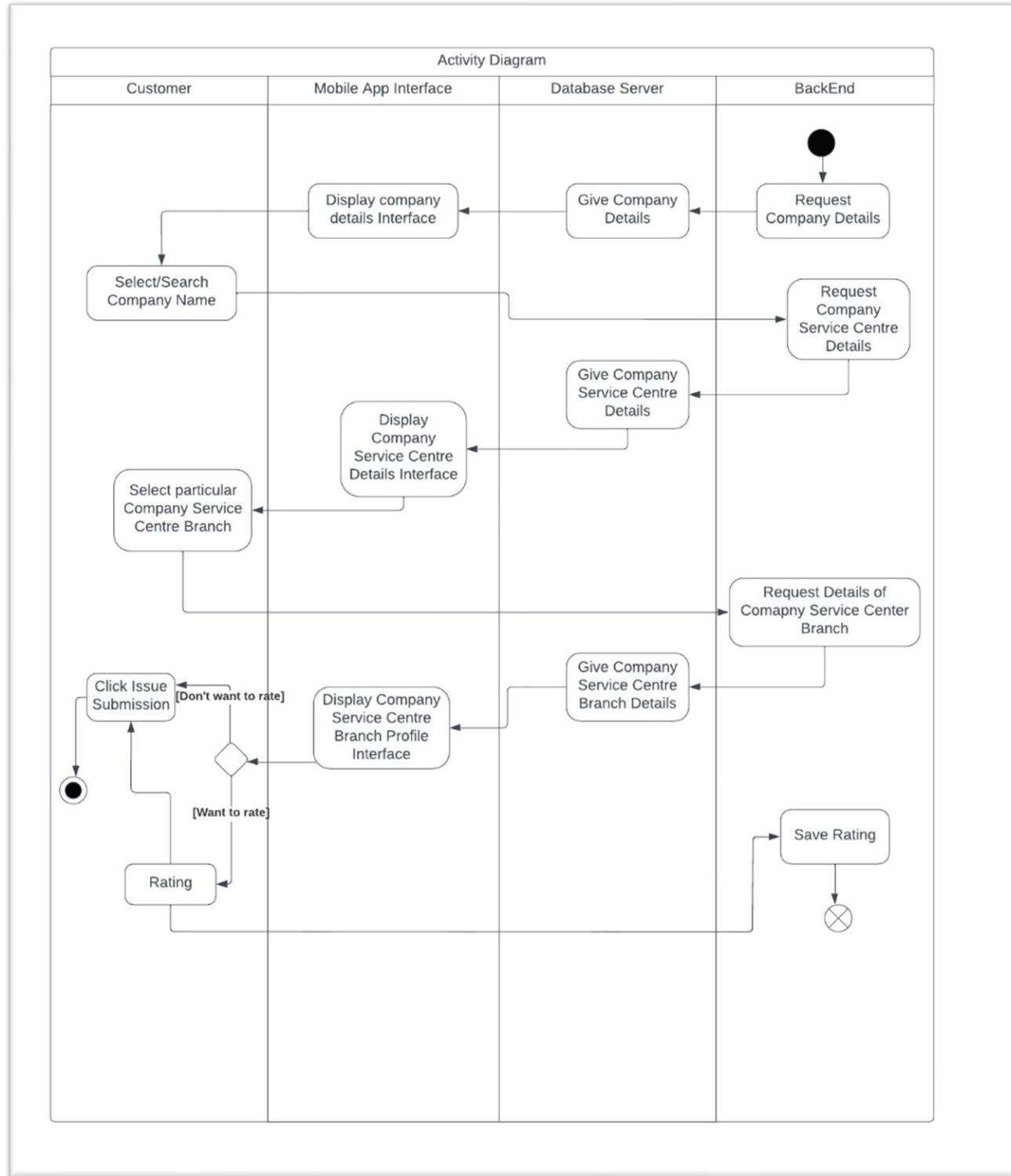


Figure 5.3.4 Activity Diagram - Select a Company to Submit Issue

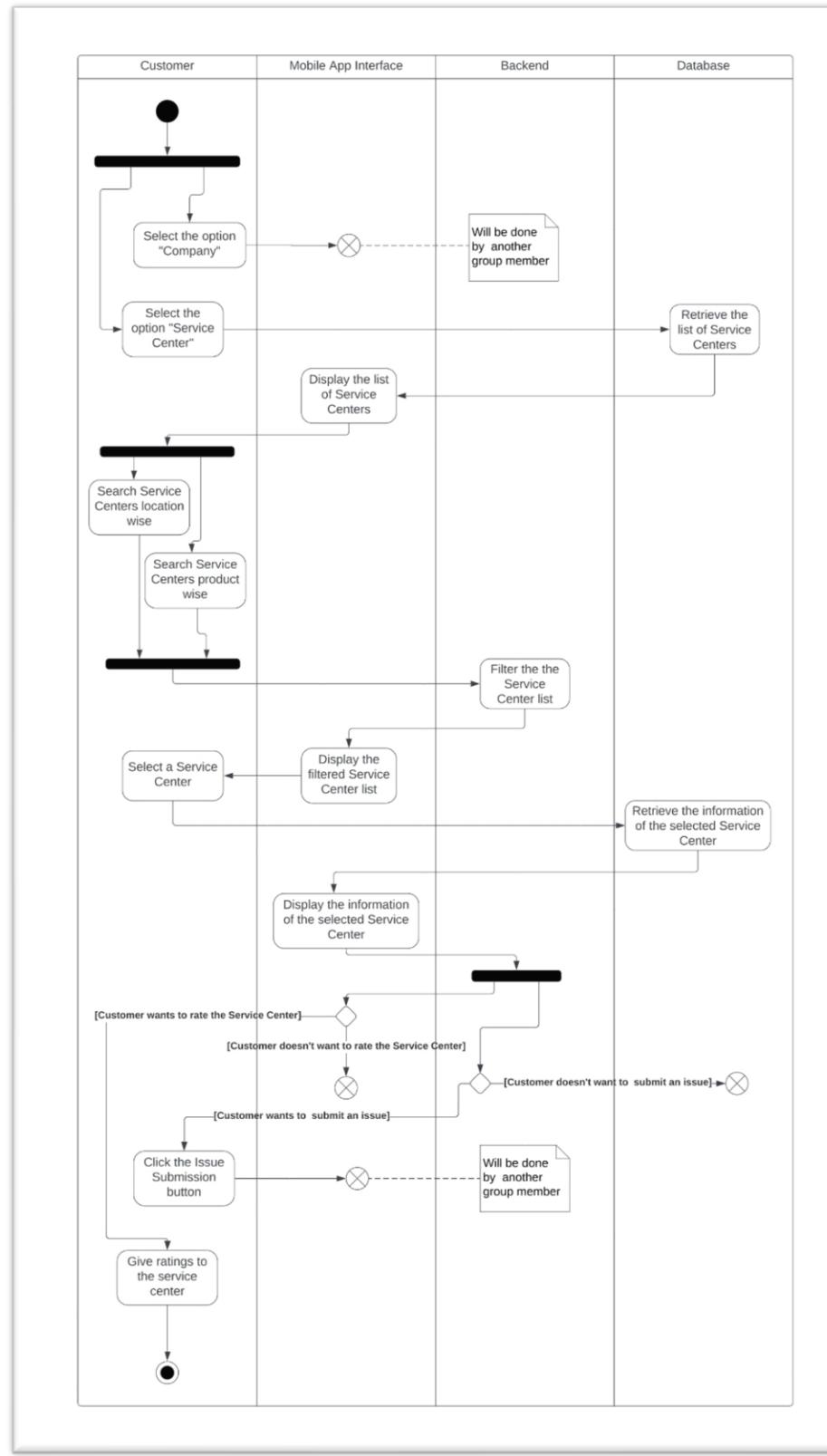


Figure 5.3.5 Activity Diagram - Select a Service Center to Submit Issue

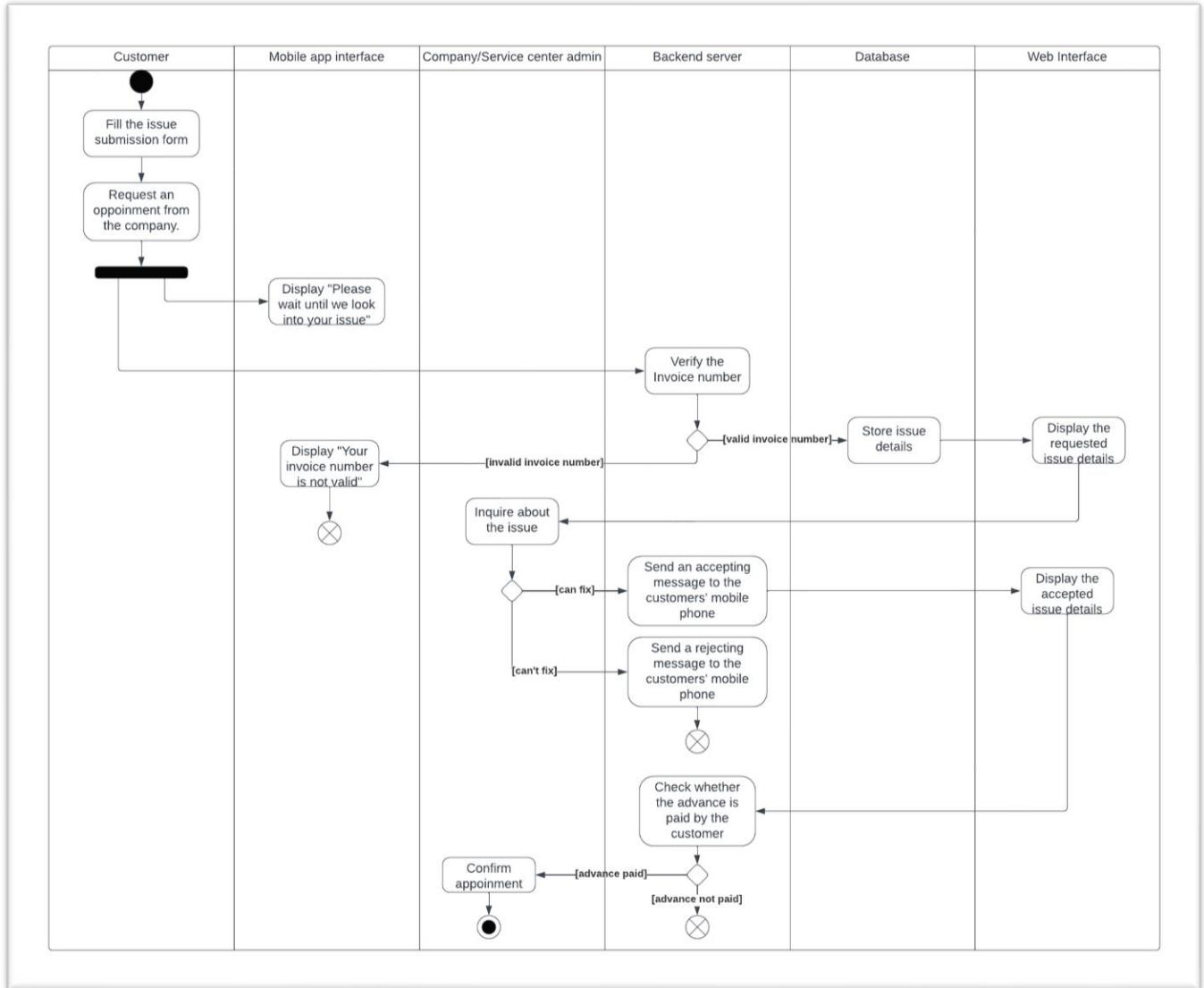


Figure 5.3.6 Activity Diagram - Submit the Issue

5.3.2.2 Activity Diagrams for Web Application

We designed several activity diagrams to demonstrate the major activities in our proposed web application.

- Company/Service Center Admin login & registration
- Company/Service Center profile
- Company/Service Center dashboard
- Add company/service center details
- View appointment details
- Repairment Completion

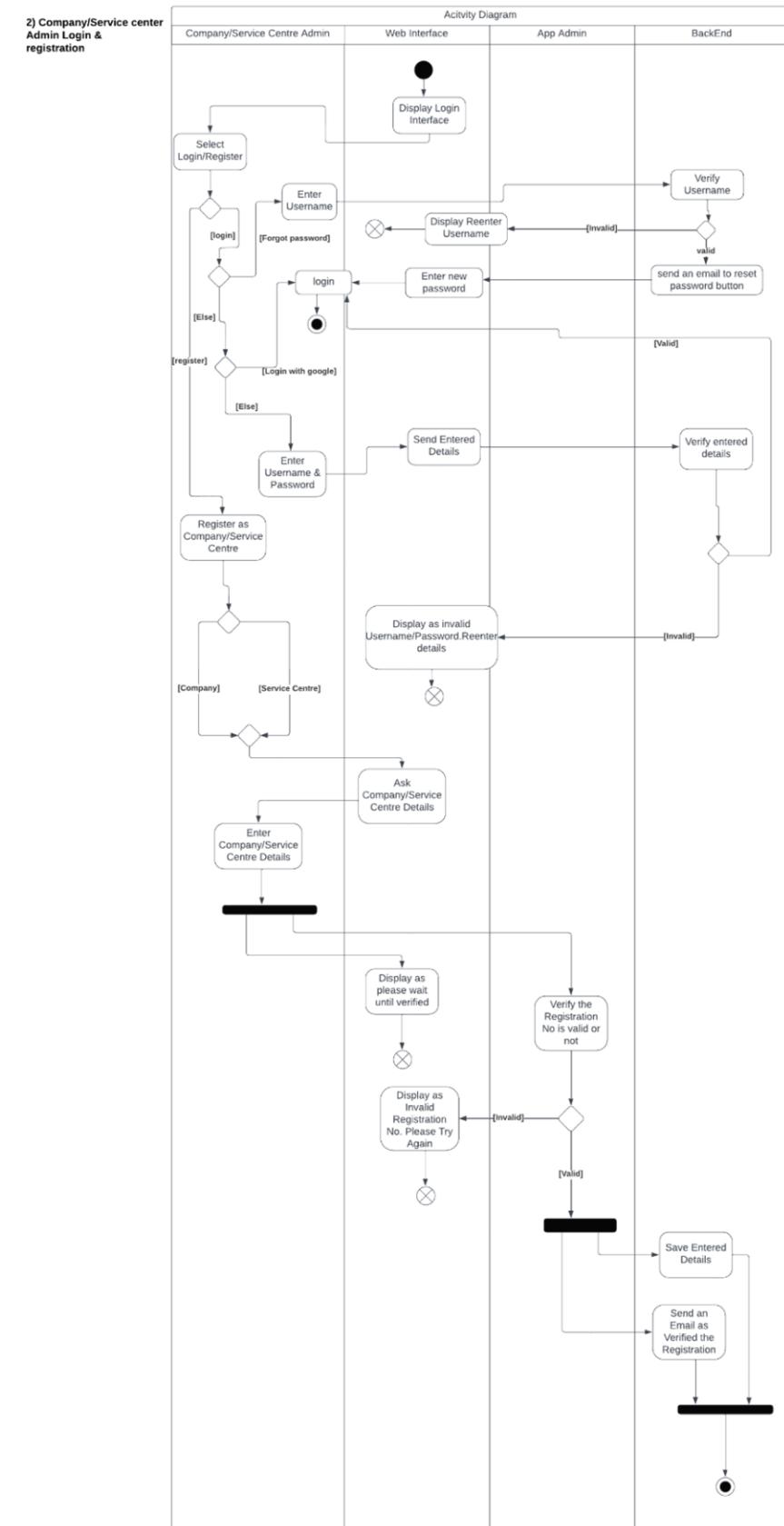


Figure 5.3.7 Activity Diagram - Company/Service Center Admin login & registration

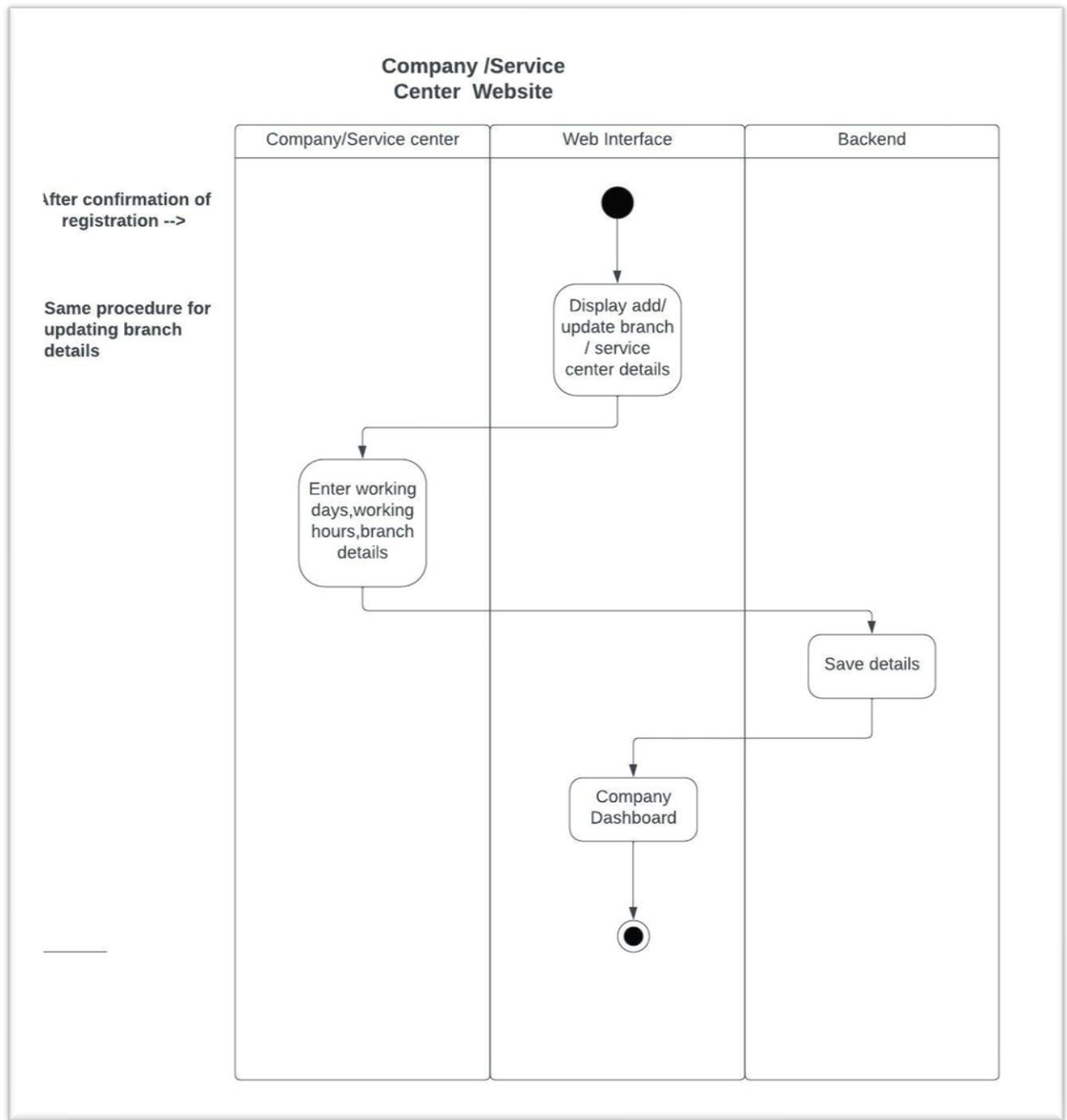


Figure 5.3.8 Activity Diagram - Add Company/Service Center Details

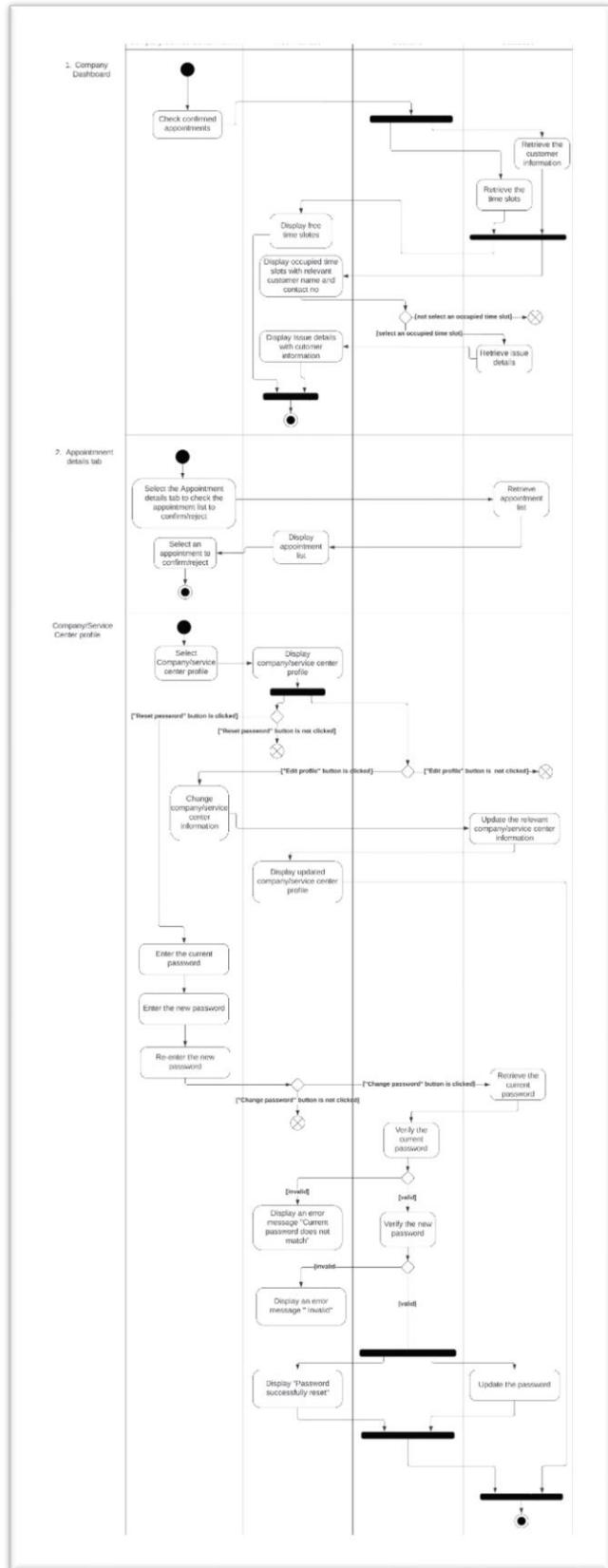


Figure 5.3.9 Activity Diagram - Company/Service Center Dashboard, View Appointment Details and Company/Service Center Profile,

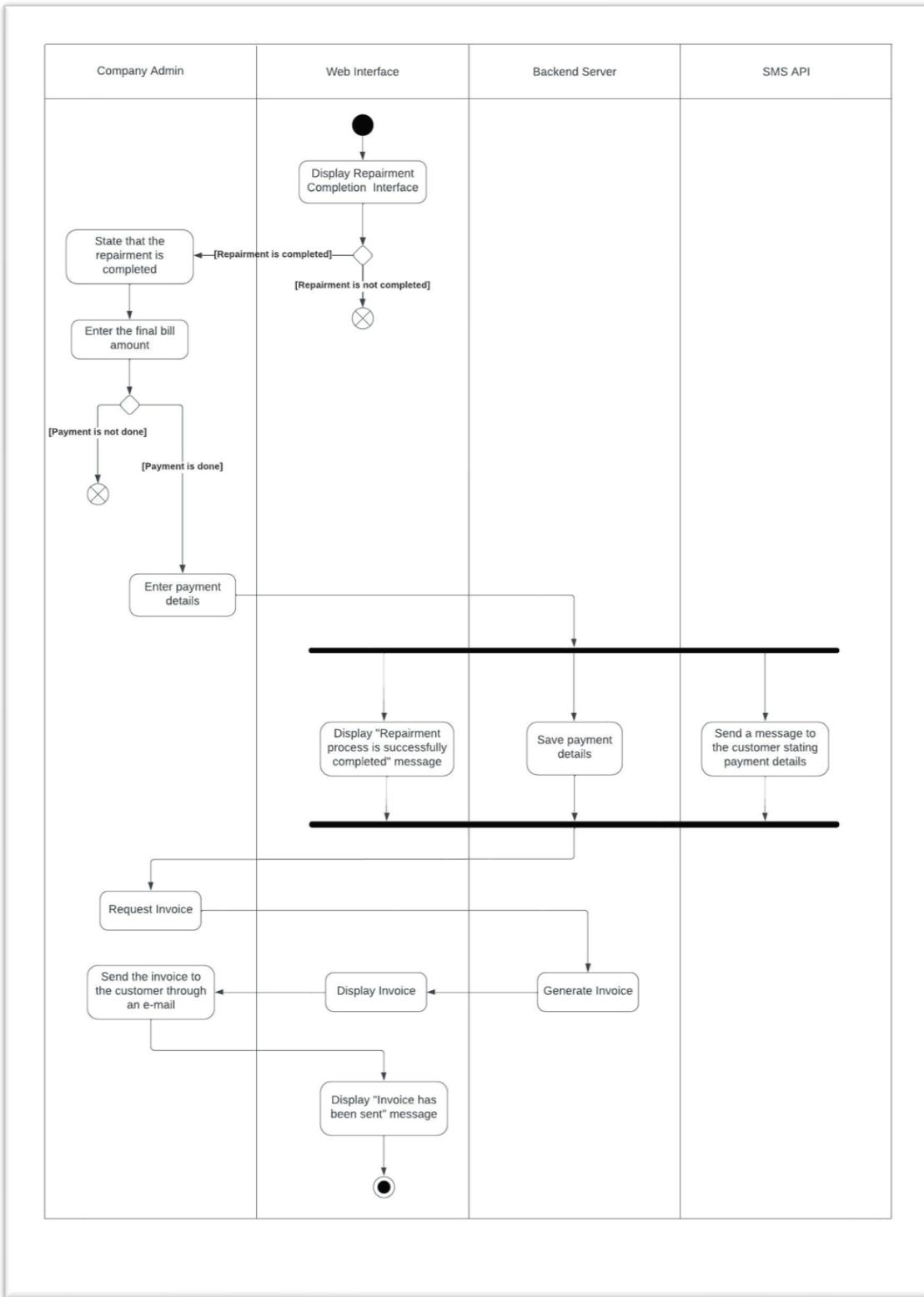


Figure 5.3.10 Activity Diagram - Repairment Completion

5.3.3 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order, the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function. Sequence diagrams capture high-level interactions between user of the system and the system, between the system and other systems, or between subsystems.

5.3.3.1 Sequence Diagrams for Mobile Application

We designed sequence diagrams for the following functions in our proposed mobile application.

- Customer login and registration
- Customer user profile
- Select a company to submit the issue and rate
- Select a service center to submit the issue and rate
- Submit the issue and appointment confirmation
- Date & time selection
- Payment Interface

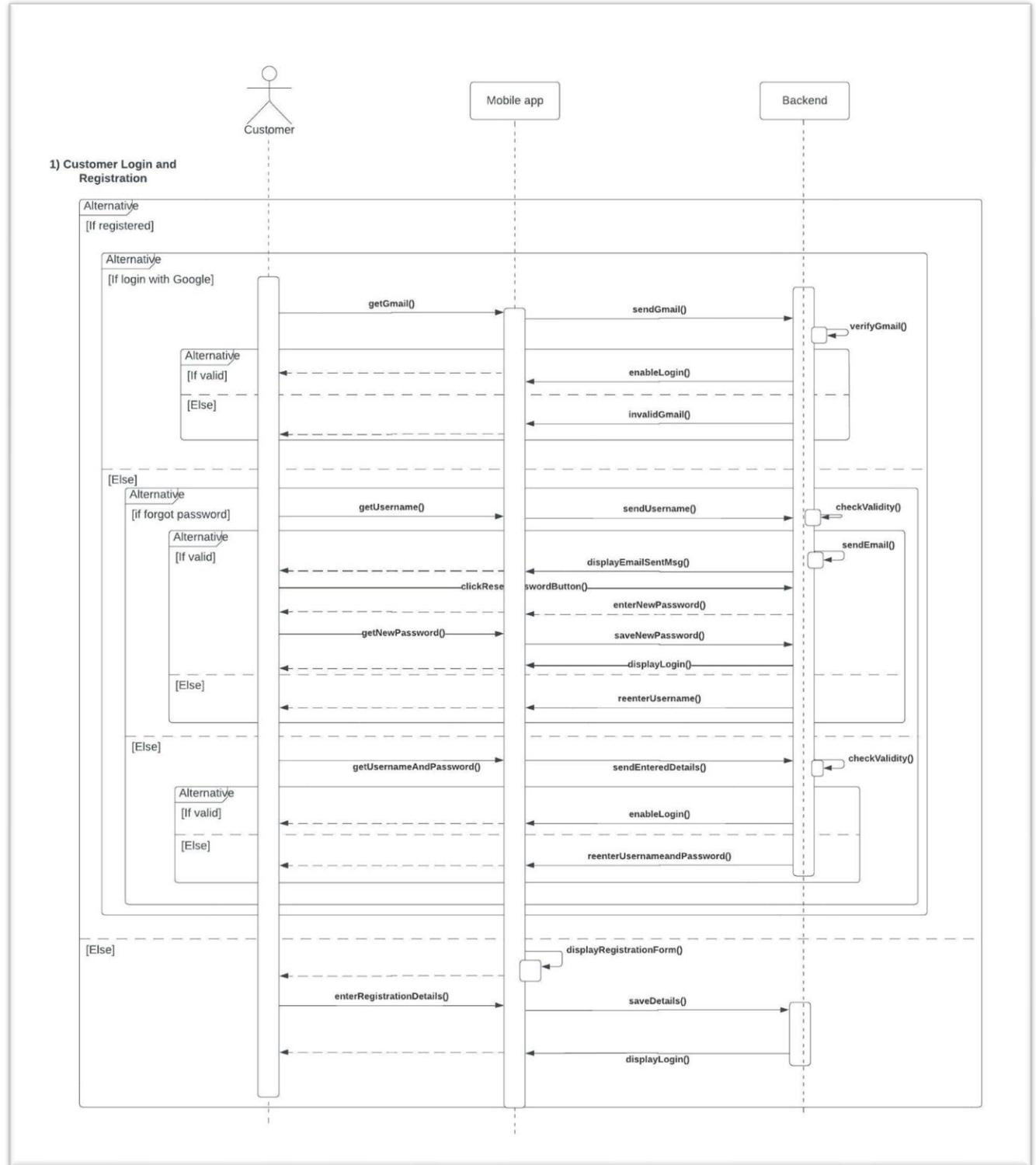
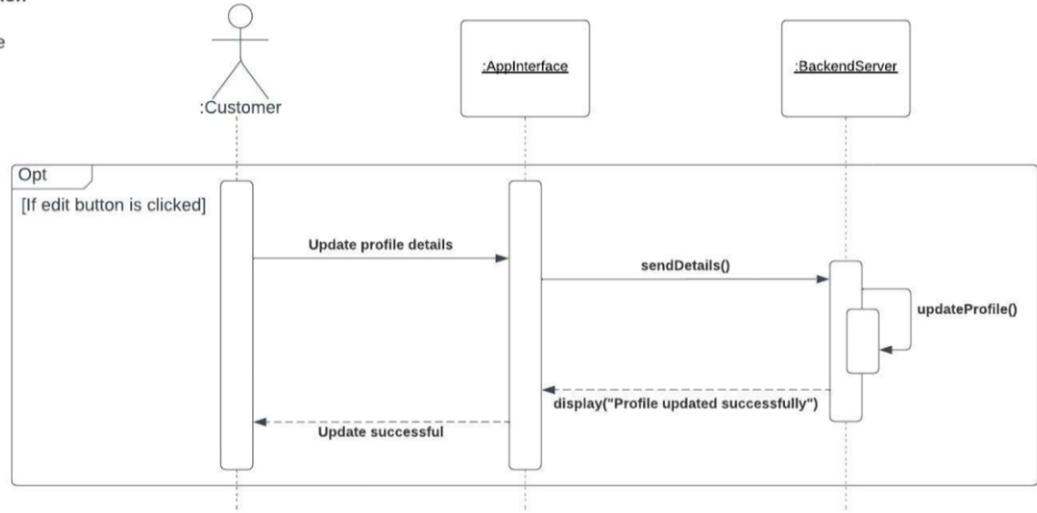


Figure 5.3.11 Sequence Diagram - Customer Login and Registration

Mobile Application

1. Customer Profile Interface



2. Date Selection Interface & Time Selection Interface

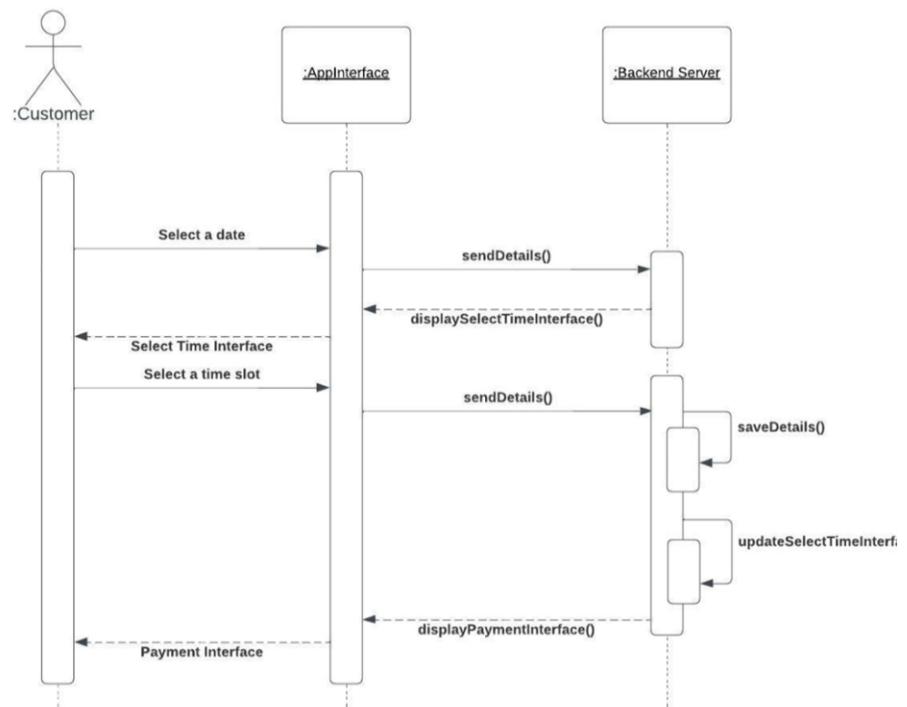


Figure 5.3.12 Sequence Diagram - Customer Profile and Date & Time Selection

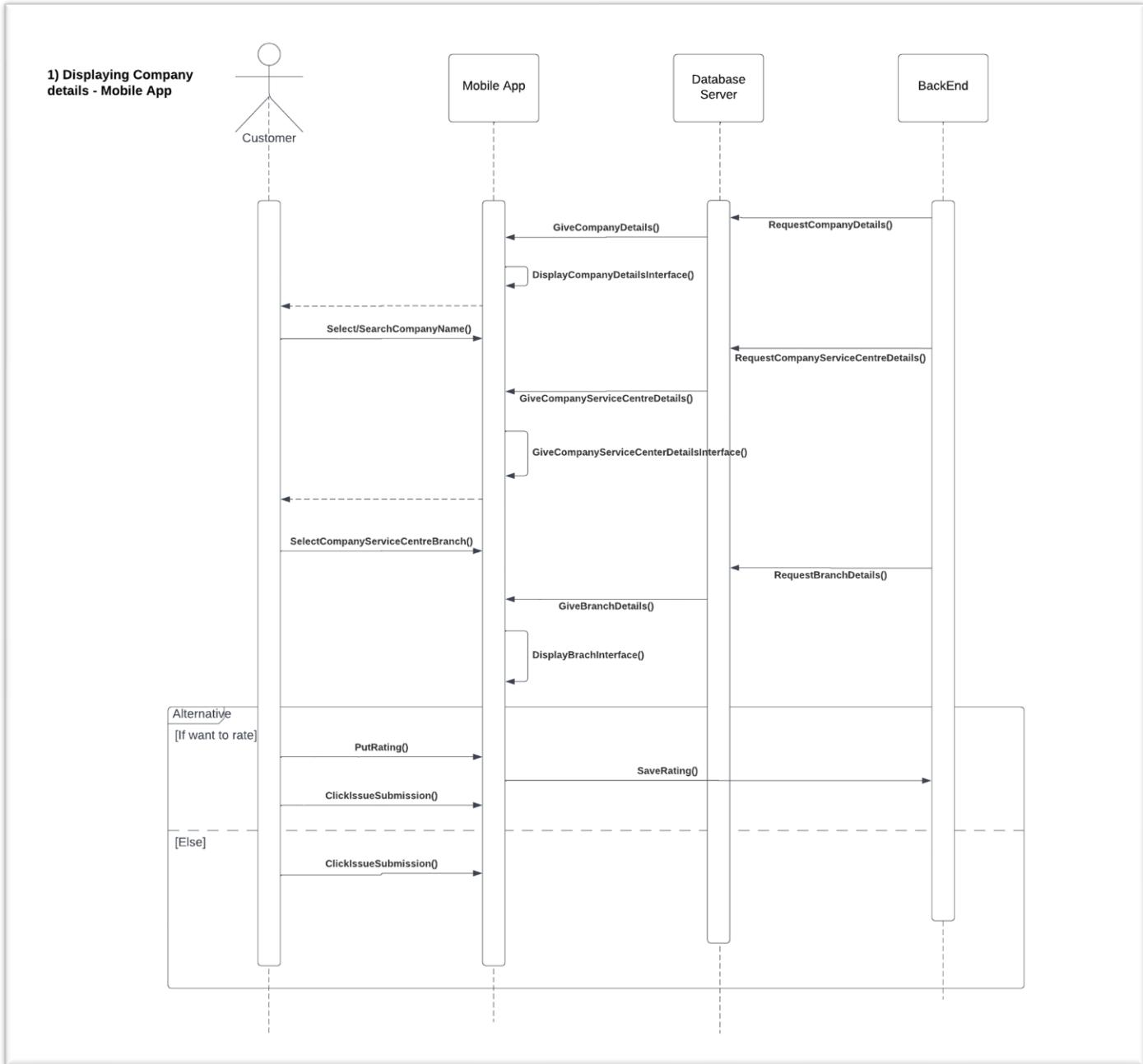


Figure 5.3.13 Sequence Diagram - Select a Company to Submit Issue

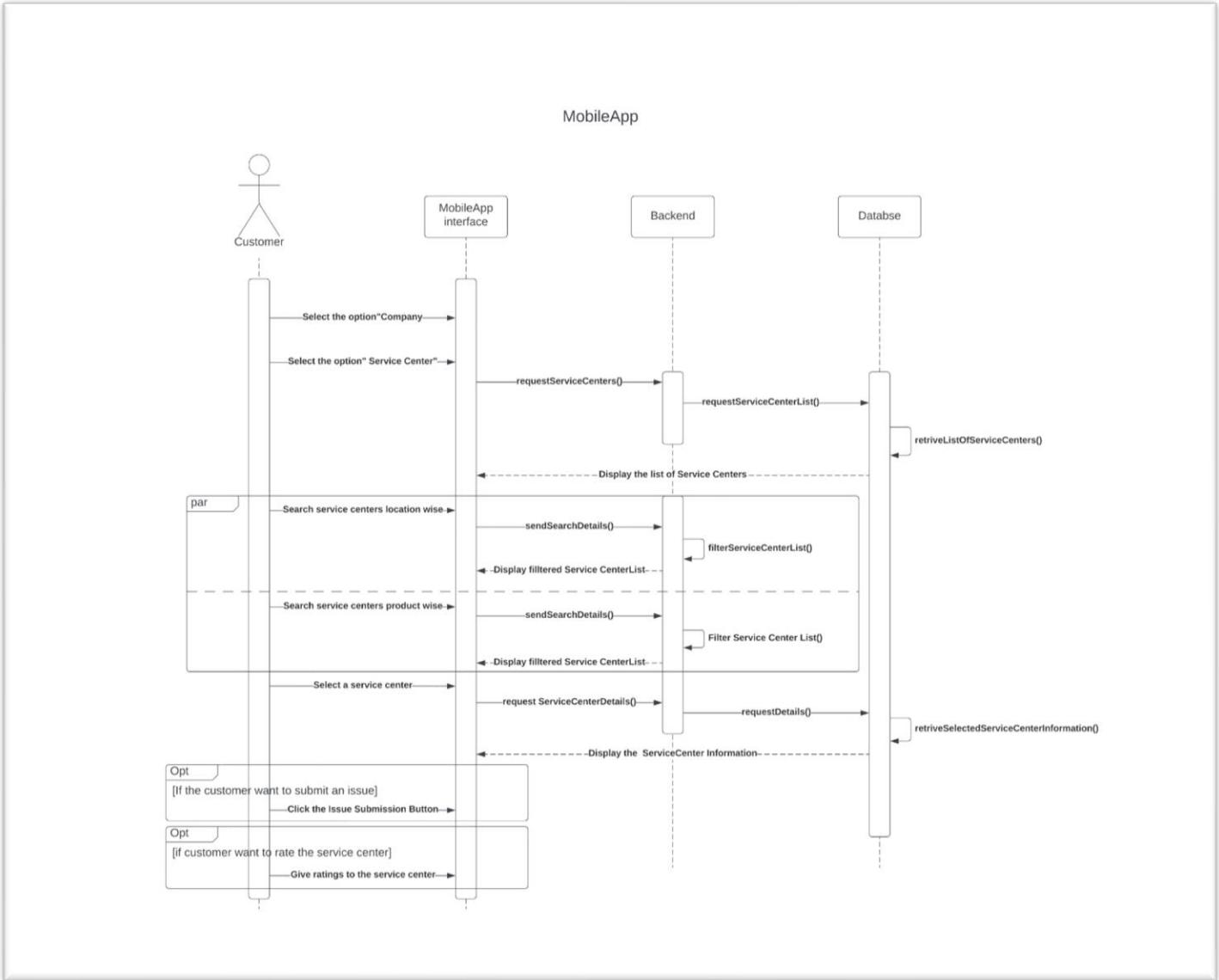
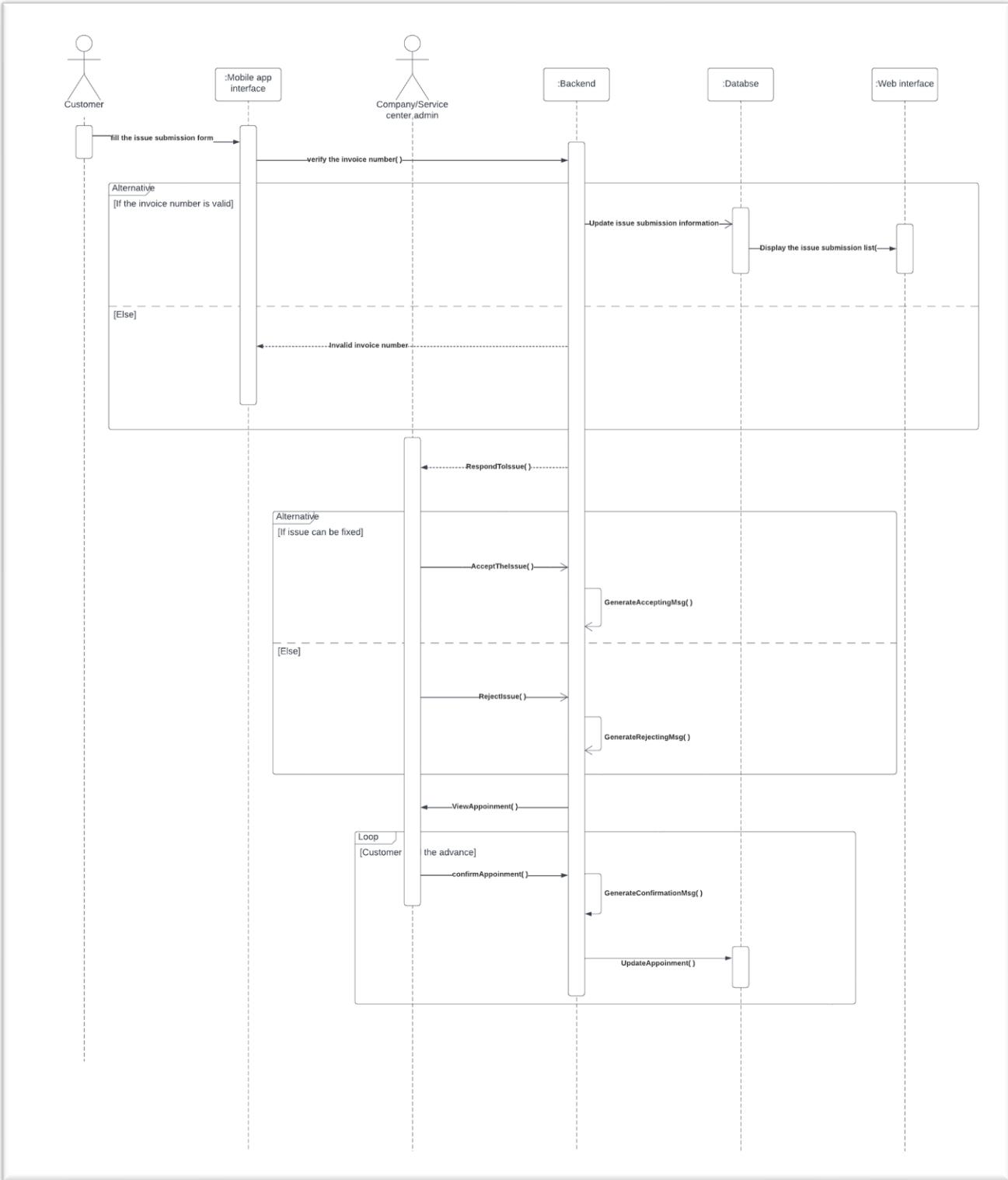


Figure 5.3.14 Sequence Diagram - Select a Service Center to Submit Issue



0

Figure 5.3.15 Sequence Diagram - Submit the Issue

5.3.3.1 Sequence Diagrams for Web Application

The following are the sequence diagrams we designed for our web application.

- Company/Service Center Admin login & Registration
- Company/Service Center profile
- Company/Service Center dashboard
- Add company/service center details
- View appointment details
- Repairment Completion

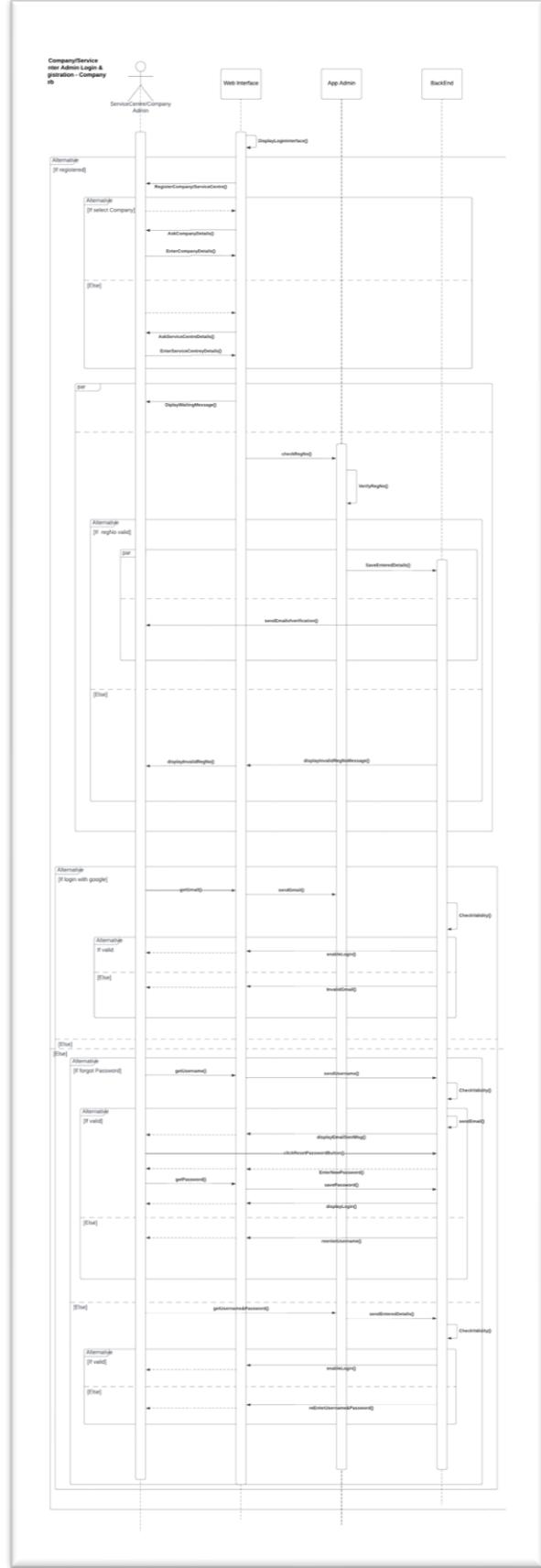


Figure 5.3.16 Sequence Diagram - Company/Service Center Admin Login & Registration

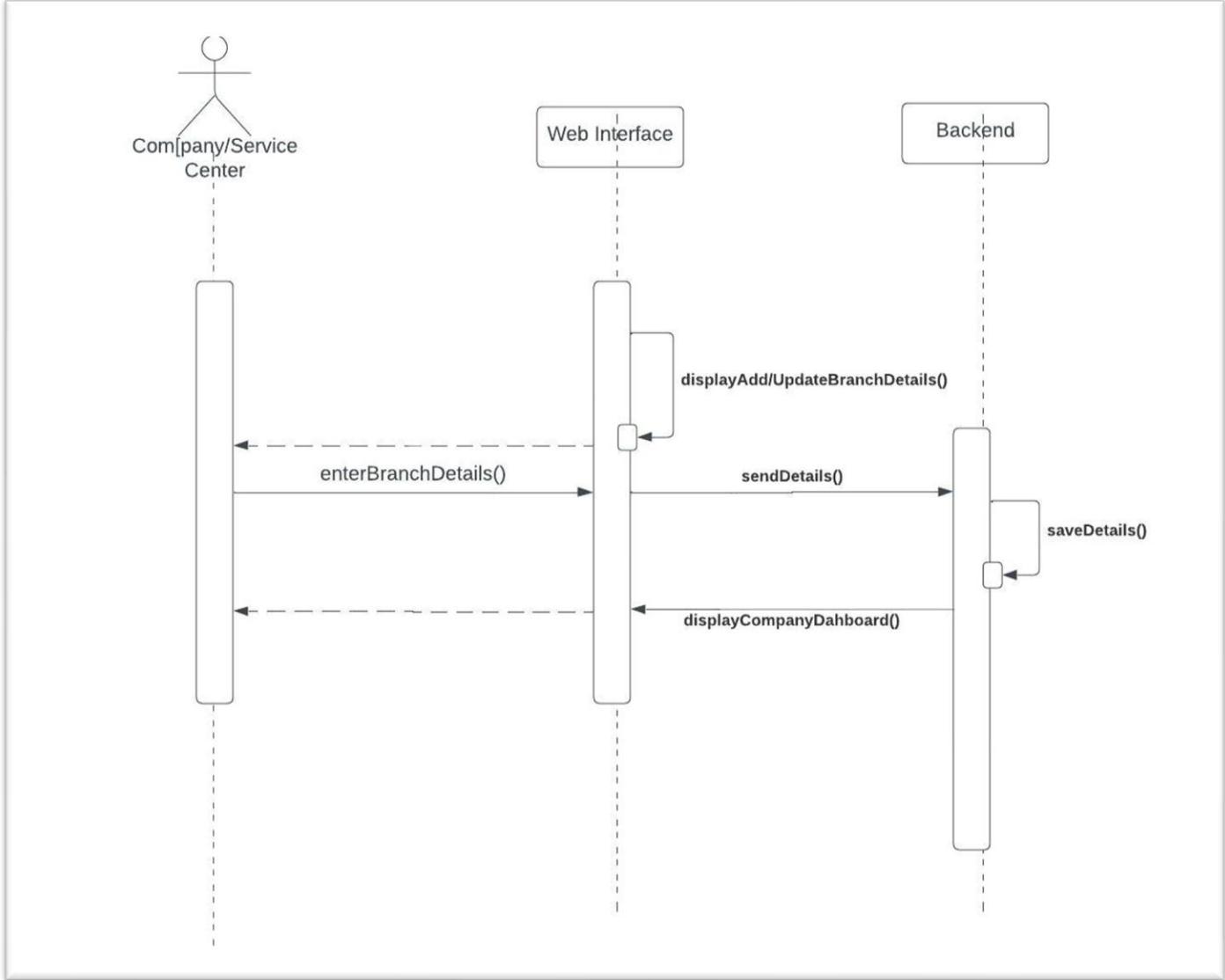


Figure 5.3.17 Sequence Diagram - Add Company/Service Center Details

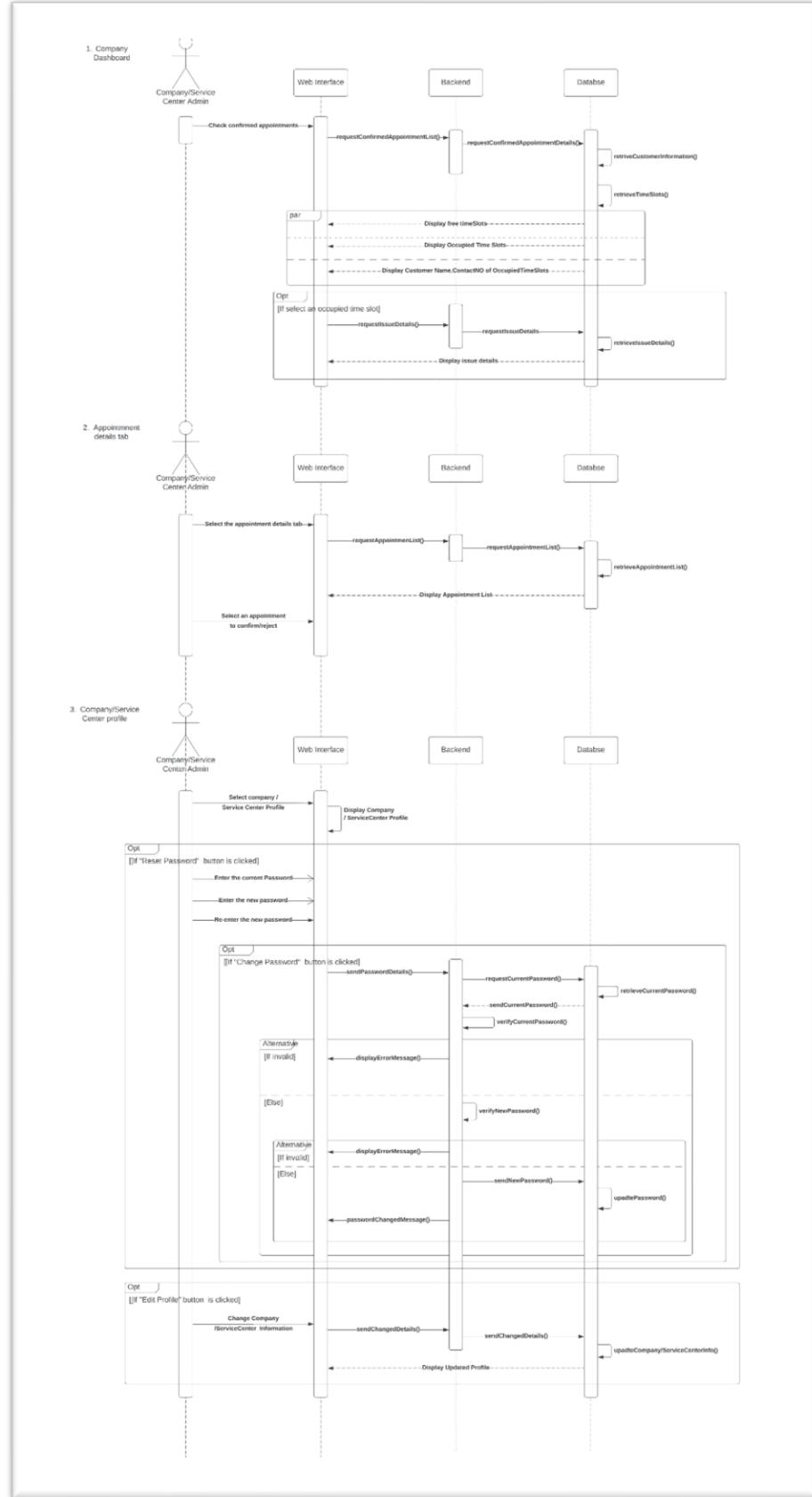


Figure 5.3.18 Sequence Diagram - Company/Service Center Dashboard, View Appointment Details and Company/Service Center Profile

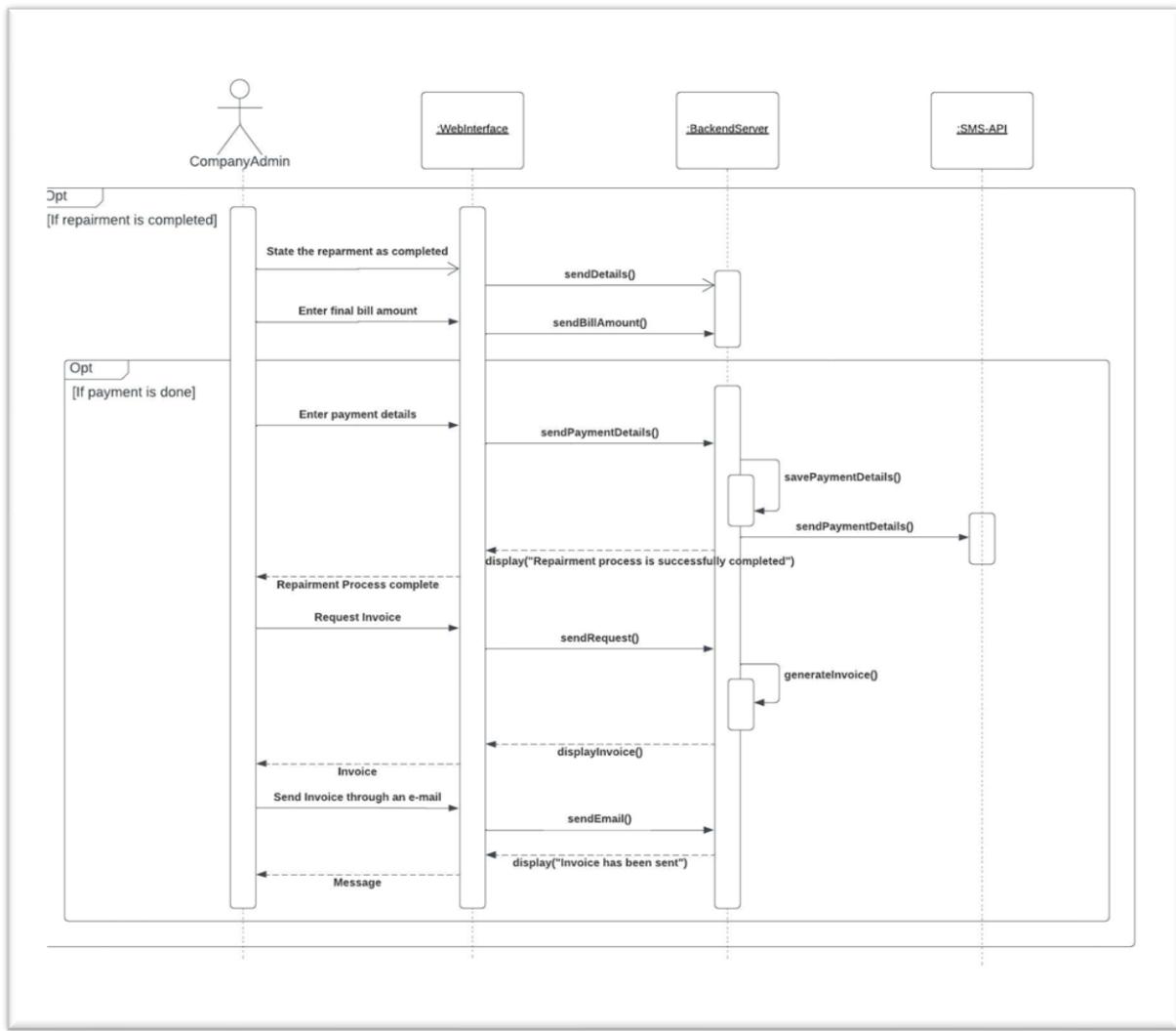


Figure 5.3.19 Sequence Diagram – Repayment Completion

5.3.4 Class Diagram

Class diagram describes the structure of a system by showing the system's

- Classes
- Attributes of the classes
- Operations (or methods) of the classes
- Relationships among objects

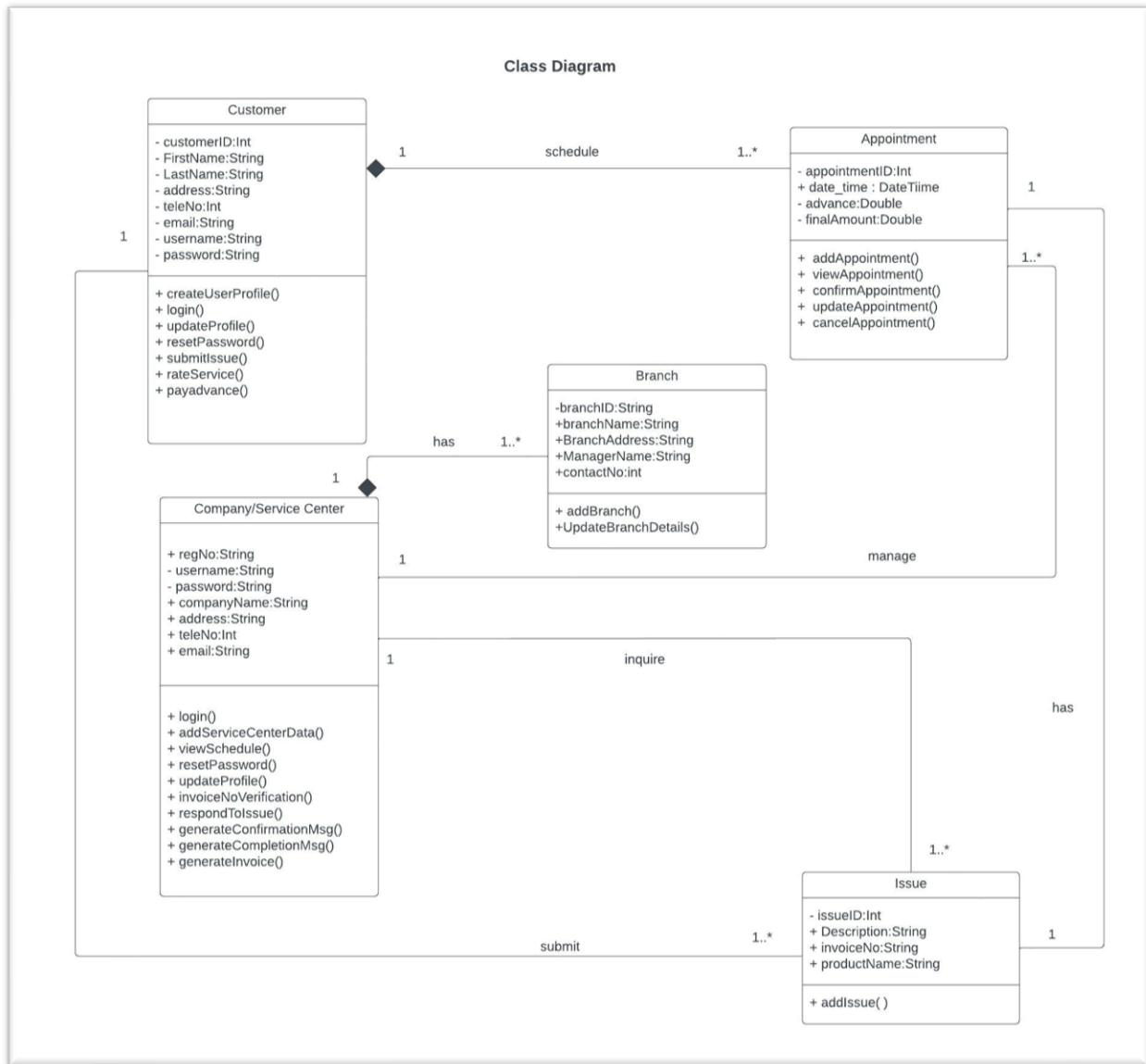


Figure 5.3.1 Class Diagram

5.3.5 ER Diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

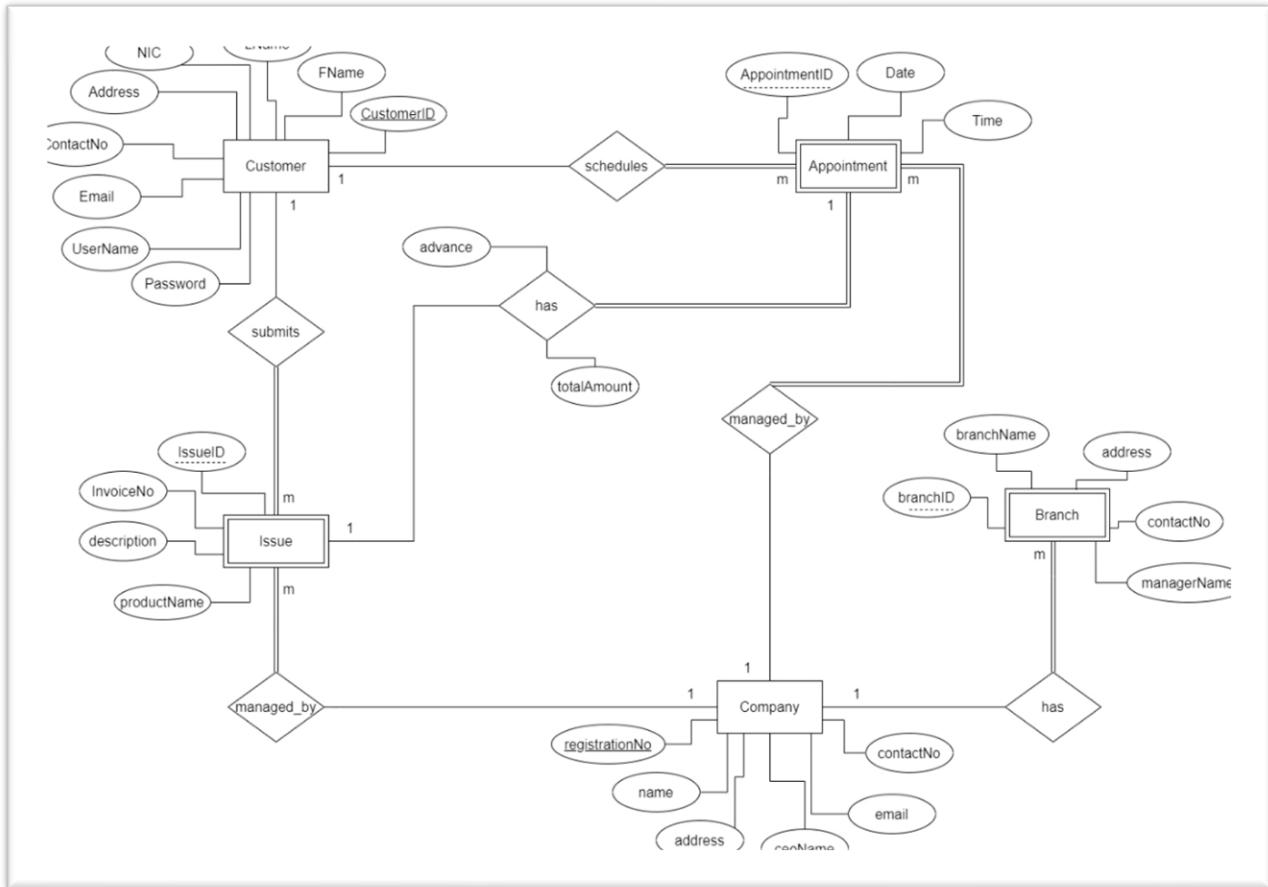


Figure 5.3.20 ER Diagram

5.4 Summary

In this chapter, we presented the use case diagram, activity diagrams, sequence diagrams and the class diagram of our system. Implementation details are discussed in Chapter 6.

Chapter 6

Implementation

6.1 Introduction

This chapter describes the implementation of our system after identifying and analyzing all user requirements. waterfall model was chosen as our software process model. Then we drew UML diagrams such as use case, activity, sequence, class and EER to get a clear picture of our development process. Our system consists of a mobile application and a web application. Currently, we are developing the front-end of the system.

6.2 Triggers and Implementation

6.2.1 Flow

Charts

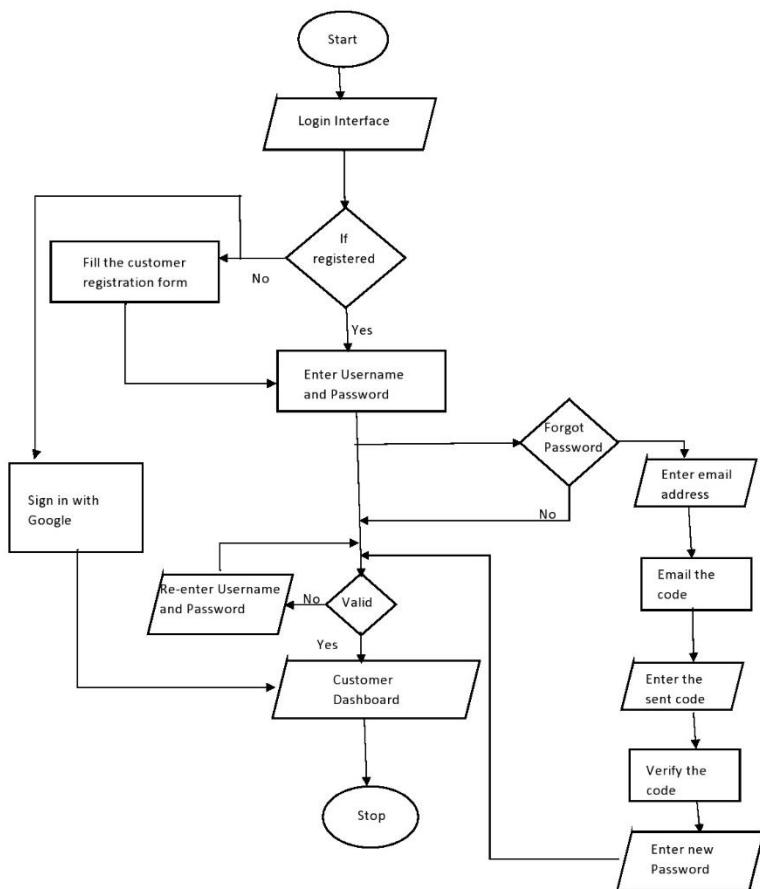


Figure 6.2.1 Flow Charts - Customer Login and Registration

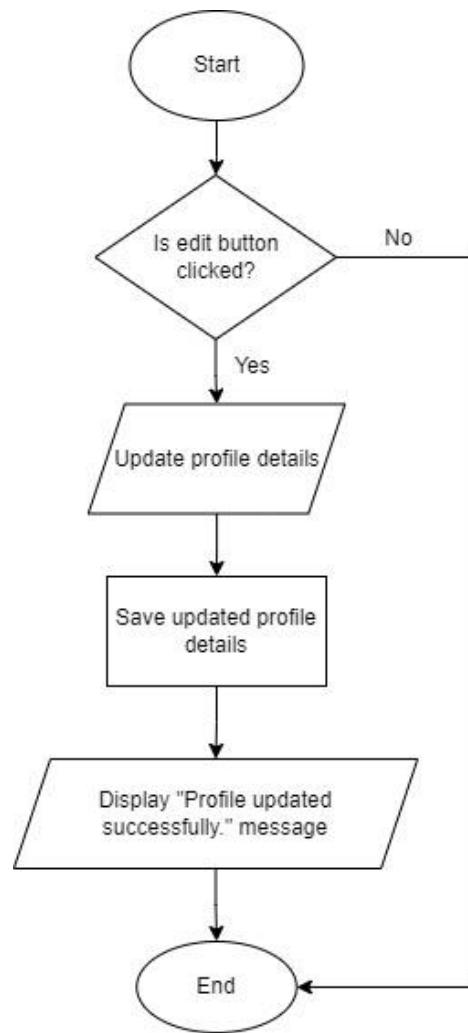


Figure 6.2.2 Flow Charts - Customer Profile

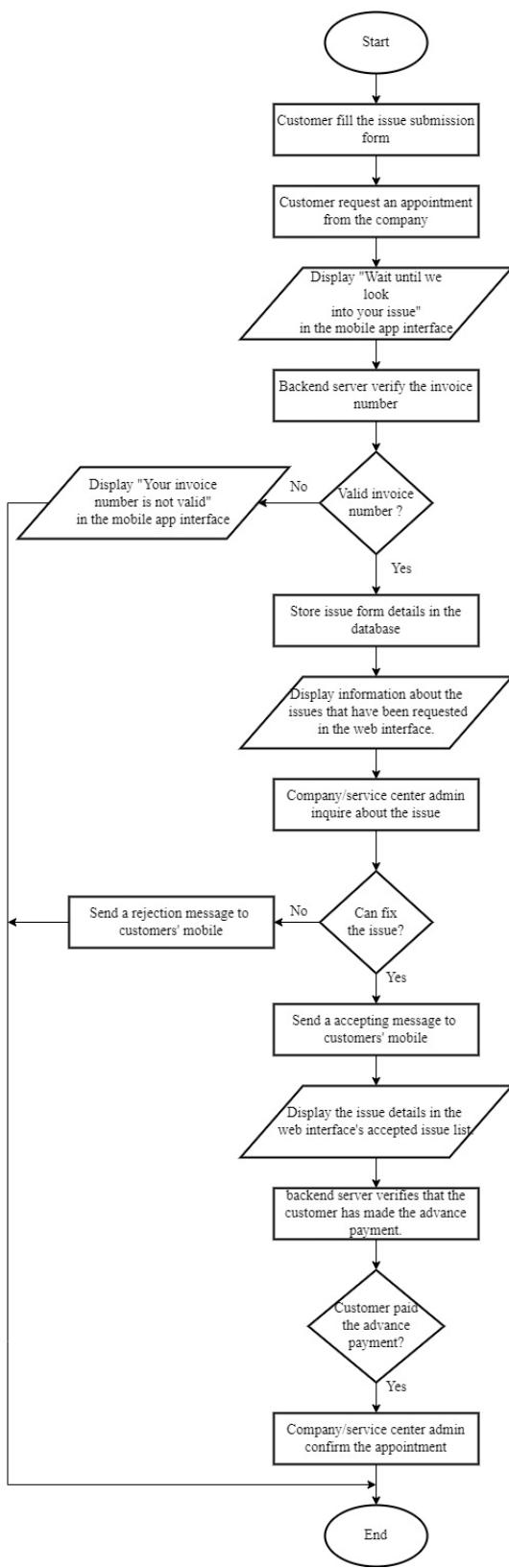


Figure 6.2.3 Flow Charts - Issue Submission

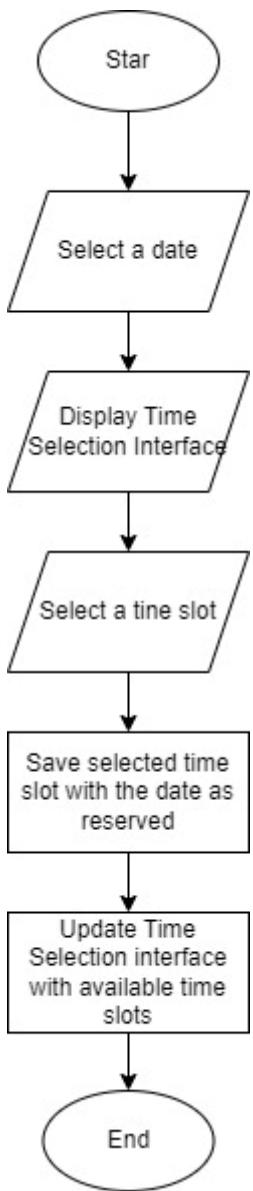


Figure 6.2.4 Flow Charts - Date and Time Selection

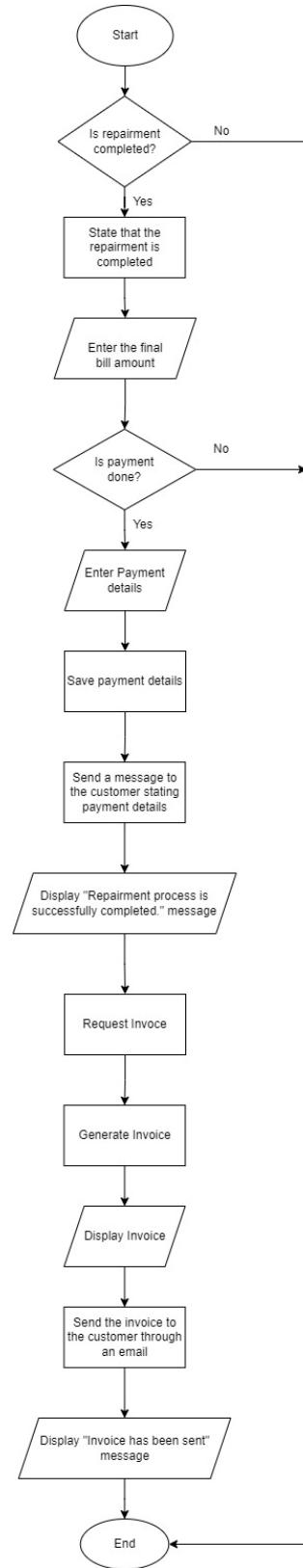


Figure 6.2.5 Flow Charts - Repairment Completion

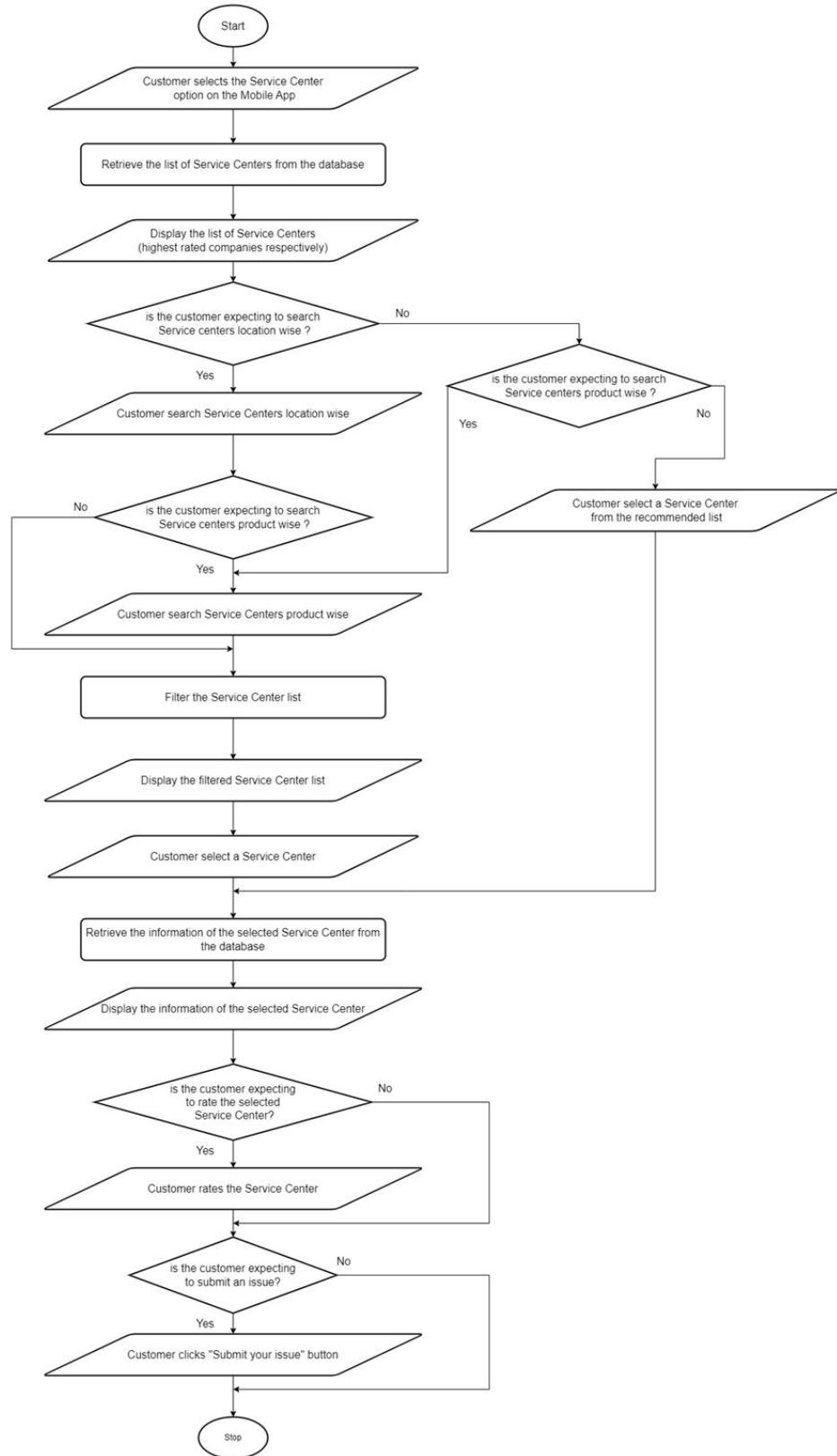


Figure 6.2.6 Flow Charts – View Service Center Details

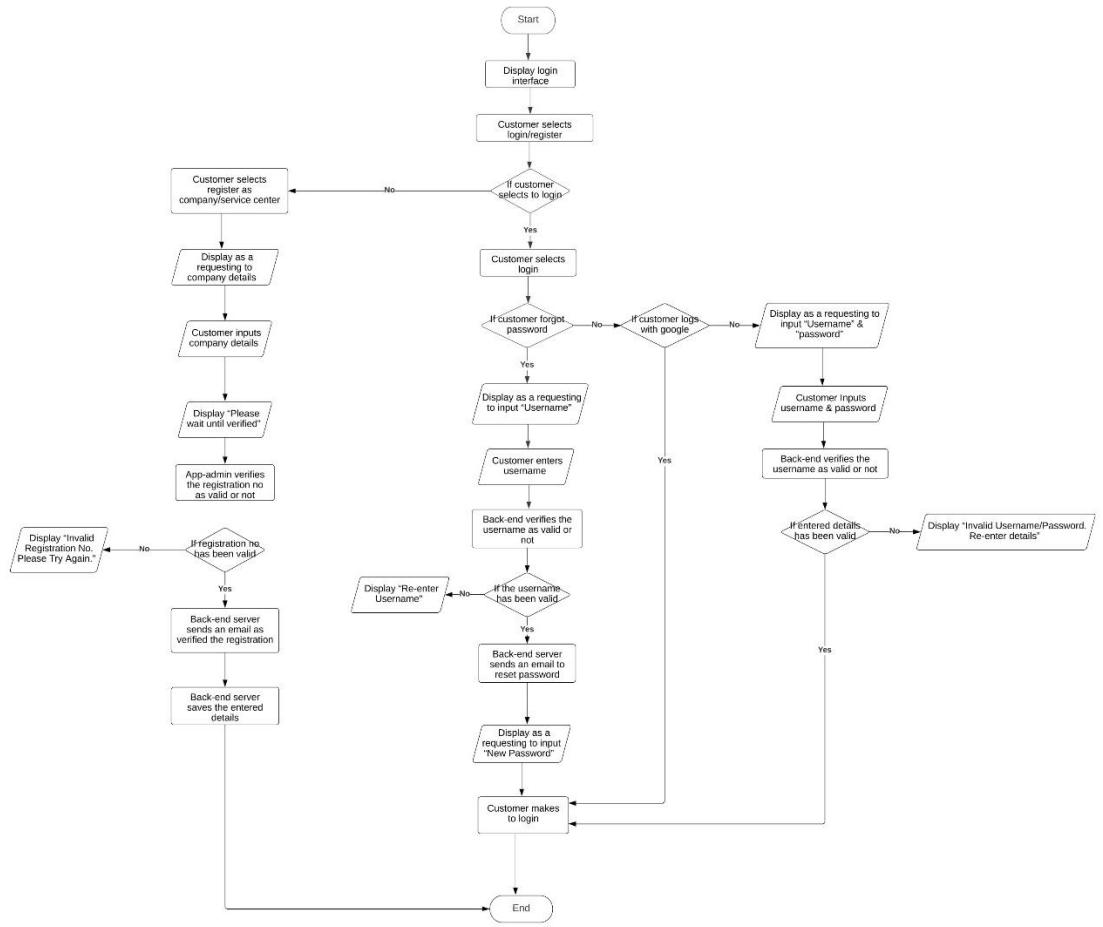


Figure 6.2.7 Flow Charts - Admin Login and Registration

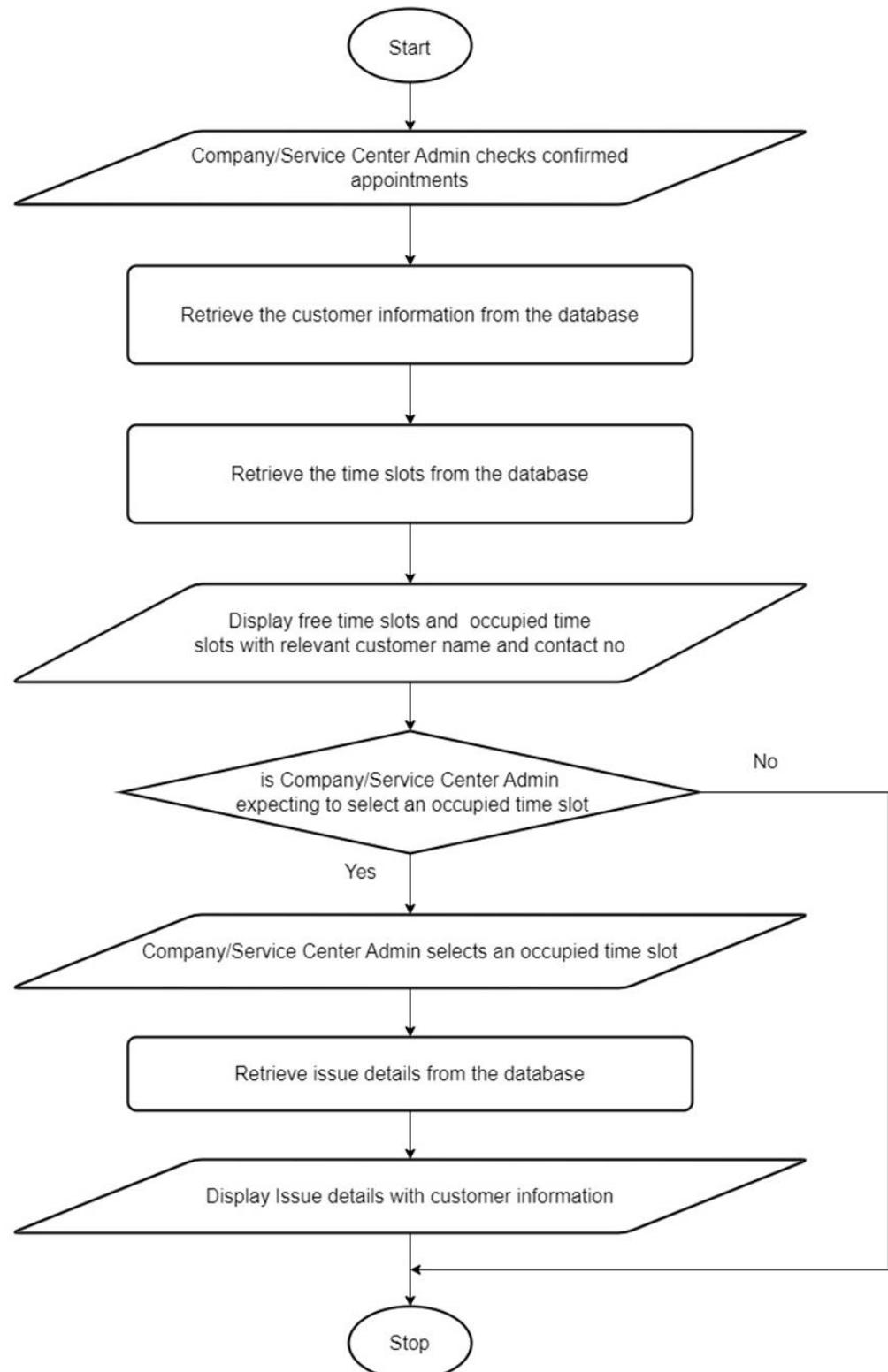


Figure 6.2.8 Flow Charts - Admin Dashboard

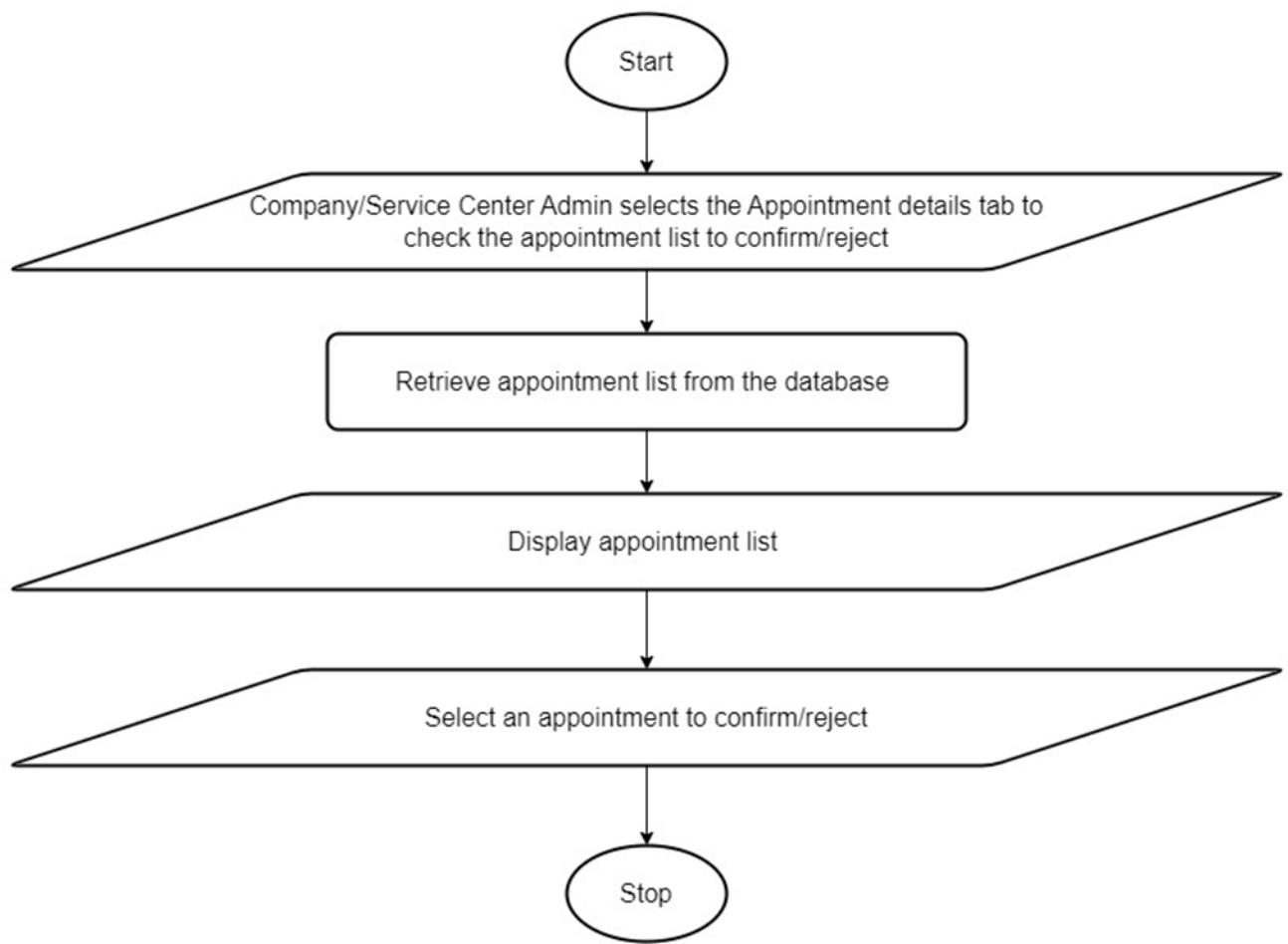


Figure 6.2.9 Flow Charts - View Appointment Details

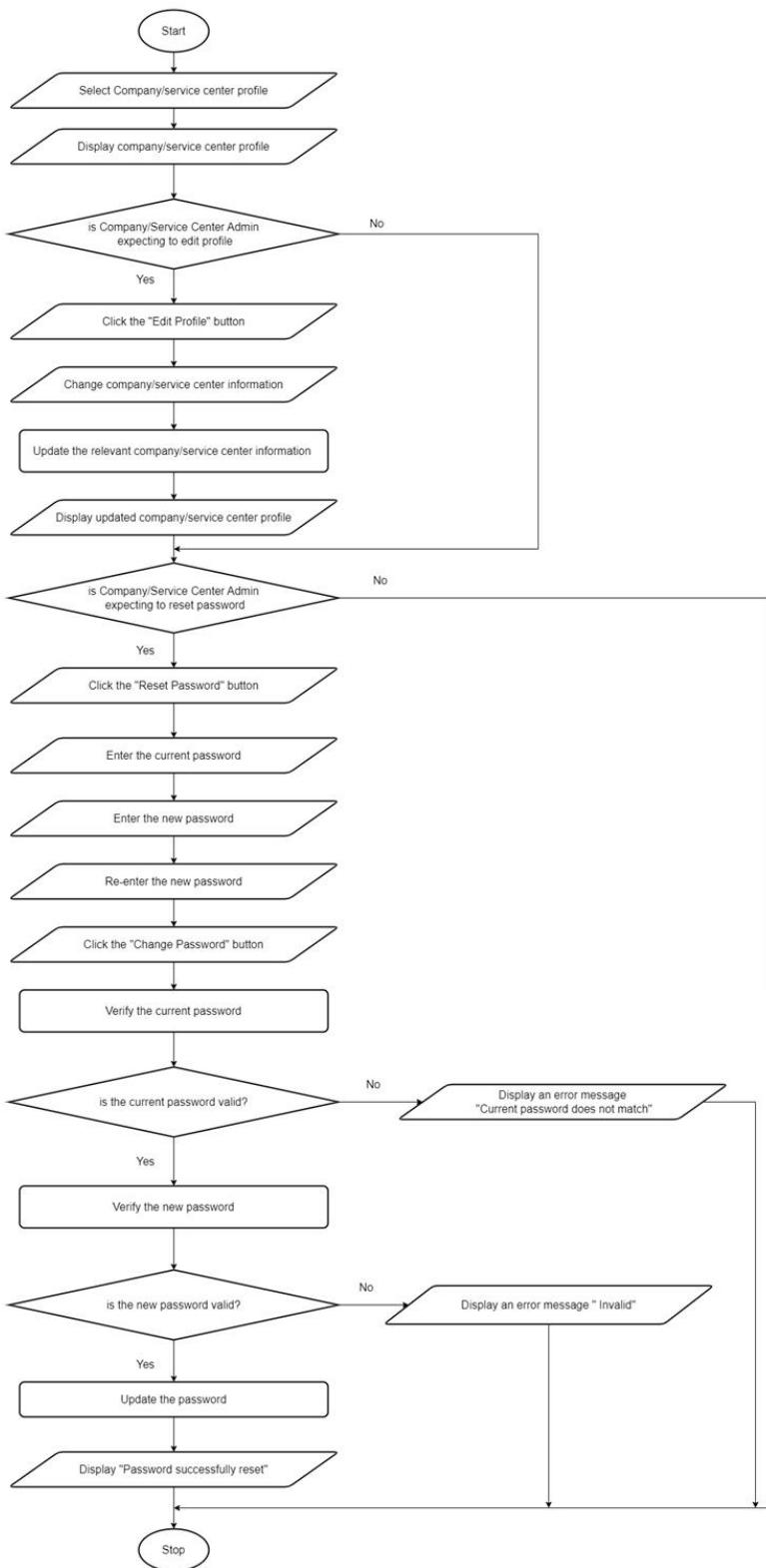


Figure 6.2.10 Flow Charts - Company/Service Center Profile

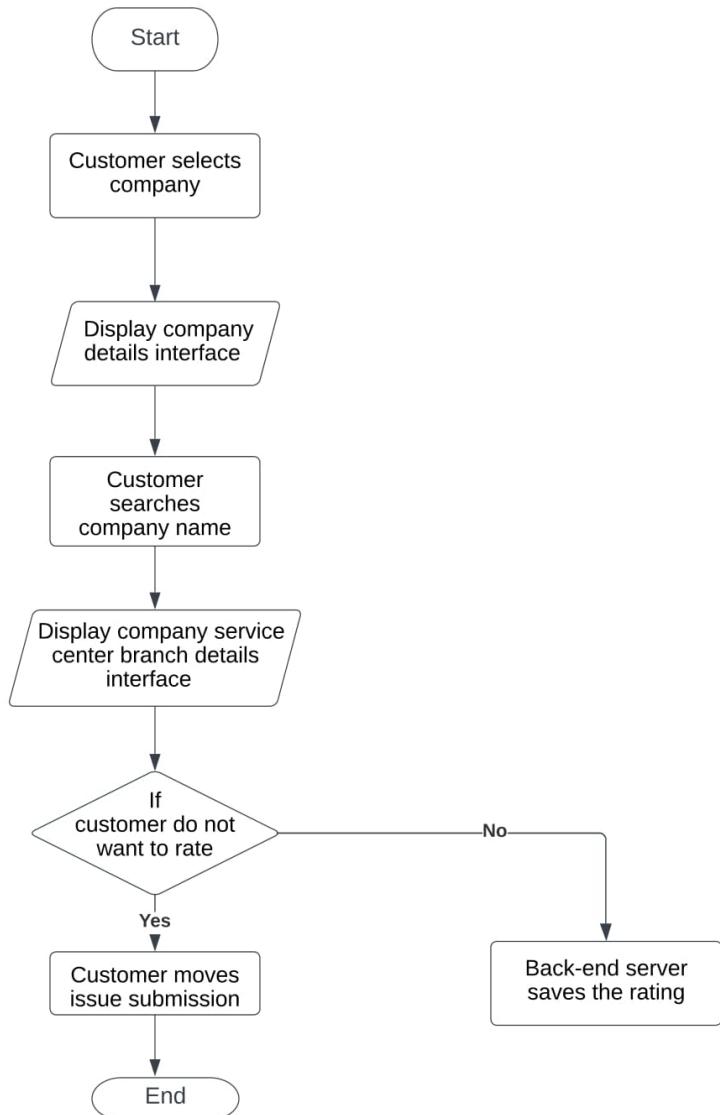


Figure 6.2.11 Flow Charts - View Company Details

6.2.2 Pseudo Codes

6.2.2.1 Customer Login and Registration

BEGIN

OUTPUT login interface

IF the customer is not registered THEN

 Fill the customer registration form

ELSE

 Sign in with Google

 Edit customer profile

END IF

IF the customer is registered THEN

 IF password is forgotten THEN

 Enter email address

 Send the code to the email

 Enter the code

 Enter New Password

 ELSE

 Enter Username and Password

 IF Valid THEN

 OUTPUT Customer Dashboard

 Edit customer profile

 ELSE

 Re-enter Username and Password

 END IF

 END IF

END IF

END

6.2.2.2 Customer Profile

BEGIN

IF Edit button is clicked THEN

 Update profile details

 Save updated profile details

 OUTPUT Profile updated successfully

END IF

END

6.2.2.3 Select a Service Center to submit issue

BEGIN

Select the Service Center option on the mobile app

Retrieve the list of Service Centers from the database

OUTPUT the list of Service Centers (highest rated companies respectively)

IF the customer is expecting to search Service centers location wise THEN

 Customer search Service Centers location wise

 IF the customer is expecting to search Service centers product wise THEN

 Customer search Service Centers product wise

 ENDIF

 Filter the Service Center list

 OUTPUT the filtered Service Center list

 Customer selects a Service Center

ELSE

 IF the customer is expecting to search Service centers product wise THEN

Customer search Service Centers product wise

 Filter the Service Center list

 Display the filtered Service Center list

 Customer selects a Service Centre

 ELSE

 Customer selects a Service Center from the recommended list

 ENDIF

ENDIF

Retrieve the information of the selected Service Center from the database

Display the information of the selected Service Center

IF the customer is expecting to rate the selected Service Center THEN

 Customer rates the Service Center

ENDIF

IF the customer is expecting to submit an issue THEN

 Customer clicks "Submit your issue" button

ENDIF

END

6.2.2.4 Select a company to submit issue

BEGIN

Customer selects company

OUTPUT company details interface

Customer searches company name

OUTPUT company service center branch details interface

IF customer do not want to rate THEN

 Customer moves issue submission

ELSE

 Back-end server saves the rating

END IF

END

6.2.2.5 Issue Submission

BEGIN

Customer fills the Issue submission form

Customer requests an appointment from the company

OUTPUT “Wait until we look into your issue”

Backend server verifies the invoice number

IF the invoice number is valid THEN

 Store issue form details in the database

 OUTPUT information about the issues that have been requested in the web interface

 Company/service center admin inquire about the issue

 IF the issue can be fixed THEN

Send an accepting message to customers' mobile

OUTPUT the issue details in the web interface's accepted issue list

Backend server verifies that the customer has made the advance payment

IF the customer has paid the advance payment THEN

 Company/service center admin confirm the appointment

END IF

ELSE

 Send a rejection message to customers' mobile

END IF

ELSE

 Display "Your invoice number is not valid" in the mobile app interface

END IF

END

6.2.2.6 Date and Time Selection

BEGIN

Select a date

OUTPUT Time Selection Interface

Select a time slot

Save selected time slot with the date as reserved

Update Time Selection Interface with available time slots

END

6.2.2.7 Admin Login and Registration

BEGIN

Display login interface

Admin selects login/register

IF Admin selects to login THEN

 IF customer forgot password THEN

 Admin enters username and password

 IF the username and password are valid THEN

 Admin login successful

 ELSE

 Display “Invalid Login”

 ENDIF

 ELSE

 IF customer logs with google THEN

 Admin enters email

 Login successful

 ELSE

 Admin enters username & password

 IF entered are valid THEN

 Admin login successful

 ELSE

 Display “Invalid Username/Password. Re-enter details”

 ENDIF

 ENDIF

 ENDIF

ELSE Customer selects register as company/service center

Customer inputs company details

Display “Please wait until verified”

IF registration no has been valid THEN
 Back-end server sends an email as verified the registration
 Back-end server saves the entered details
ELSE
 Display “Invalid Registration No. Please Try Again.”
END IF
END IF
END

6.2.2.8 Company/Service Center Profile

BEGIN
Select Company/service center profile
OUTPUT company/service center profile
IF Company/Service Center Admin is expecting to edit profile THEN
 Click the "Edit Profile" button
 Update the relevant company/service center information
 OUTPUT updated company/service center profile
ENDIF
IF Company/Service Center Admin is expecting to reset password THEN
 Click the "Reset Password" button
 Enter the current password
 Enter the new password
 Re-enter the new password
 Click the "Change Password" button
 Verify the current password
 IF the current password is valid THEN
 Verify the new password
 IF the new password is valid THEN
 Update the password

```
        OUTPUT "Password successfully reset"  
    ELSE  
        OUTPUT an error message " Invalid"  
    ENDIF  
ELSE  
    OUTPUT an error message "Current password does not match"  
ENDIF  
ENDIF  
END
```

6.2.2.9 Admin Dashboard

BEGIN

Company/Service Center Admin checks confirmed appointments

Retrieve the customer information from the database

Retrieve the time slots from the database

Display free time slots and occupied time slots with relevant customer name and contact no

IF Company/Service Center Admin is expecting to select an occupied time slot THEN

 Company/Service Center Admin selects an occupied time slot

 Retrieve issue details from the database

 OUTPUT Issue details with customer information

ENDIF

END

6.2.2.10 View Appointment Details

BEGIN

Company/Service Center Admin selects the Appointment details tab and confirm or reject appointments.

Retrieve appointment list from the database

OUTPUT appointment list

Select an appointment to confirm/reject

END

6.2.2.11 Repairment Completion

BEGIN

IF Repairment is completed THEN

 State that the repairment is completed

 Enter the final bill amount

 IF Payment is done THEN

 Enter payment details

 Save payment details

 Send a message to the customer stating payment details

 OUTPUT "Repairment process is successfully completed."

 Request Invoice

 Generate Invoice

 OUTPUT Invoice

 Send the invoice to the customer through an email

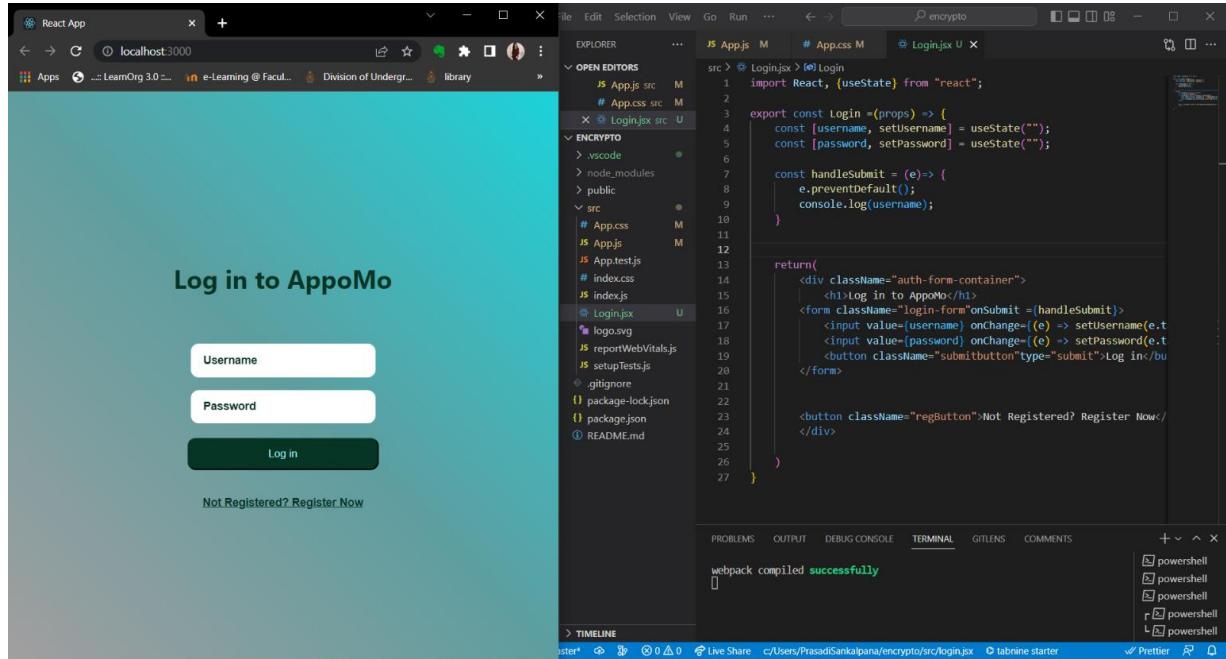
 OUTPUT "Invoice has been sent" message

END IF

END IF

END

6.2.3 Code Segments



```
import React, {useState} from "react";
const [username, setUsername] = useState("");
const [password, setPassword] = useState("");
const handleSubmit = (e) => {
  e.preventDefault();
  console.log(username);
}
return(
  <div className="auth-form-container">
    <h1>Log in to AppoMo</h1>
    <form className="login-form" onSubmit={handleSubmit}>
      <input value={username} onChange={(e) => setUsername(e.target.value)} />
      <input value={password} onChange={(e) => setPassword(e.target.value)} />
      <button className="submitbutton" type="submit">Log in</button>
    </form>
    <button className="regButton">Not Registered? Register Now</button>
  </div>
)
}
```

Figure 6.2 12 Implementation - Web Interface Frontend

6.3 Summary

Throughout this chapter, we discussed the implementation of our system up to now.

Chapter 7

Discussion

7.1 Introduction

Through this chapter, we discuss the evaluation and testing of our solution and how our solution differs from other solutions in the market. Further developments and implementation are also discussed in this chapter.

7.2 Evaluation and Testing

Currently, we have not evaluated or tested our solution as we are currently in the initial stage of development. so through this chapter we like to discuss how our solution differs from other products in the market.

7.3 How our solution differs from others

According to our literature survey, there is not an app or system where customer can know if the issue of the item/device can be fixed in advance. But through our system, we provide the facility to submit the issue and check whether the issue can be fixed or not. This feature helps to save the time while resulting a higher customer satisfaction.

7.4 Further Development

As of now, the verification of the company registration is done by manually. But in the future, we hope to automate the process of company registration verification.

7.5 Summary

Through the above paragraph, we have discussed the functions that we perform through our application and how those functions differ from existing applications and the future developments that we expect to perform.

Chapter 8

Reference

- [1] "Appointment management system - DevProjects", DevProjects, 2022. [Online]. Available: <https://www.codementor.io/projects/web/appointment-management-system-compu19a0t>. [Accessed: 21- Sep- 2022]
- [2] N. Tennyson, "The Advantages of Using Online Appointment Scheduling Software for Government Offices", QLess, 2022. [Online]. Available: <https://qless.com/the-advantages-of-using-online-appointment-scheduling-software-for-government-offices/>. [Accessed: 21- Sep- 2022]
- [3] "Telco API Management Platforms | CPaaS | Enterprise CPaaS | hSenid Mobile Solutions", hSenid Mobile, 2022. [Online]. Available: <https://www.hsenidmobile.com/>. [Accessed: 21- Sep- 2022]
- [4] The 12 best appointment booking and scheduling apps: Calendly (no date) Calendly.com. Available at: <https://calendly.com/blog/best-appointment-scheduling-apps> (Accessed: December 30, 2022).
- [5] Free online appointment scheduling software: Calendly (no date) Calendly.com. Available at: <https://calendly.com/> (Accessed: December 30, 2022).
- [6] Best free online appointment scheduling software: Bookafy • Bookafy (2022) Bookafy. Available at: <https://www.bookafy.com/> (Accessed: December 30, 2022).
- [7] Home (2022) Appointlet. Available at: https://www.appointlet.com/?tap_a=3391-c9deb2&tap_s=3341549-4f0e92 (Accessed: December 30, 2022).
- [8] Booking pages: Lead capture & scheduling app (no date) OnceHub. Available at: <https://oncehub.com/scheduleonce-booking-pages> (Accessed: December 30, 2022).
- [9] Grajcar, J. (2022) Why use react native for your mobile app?, stxnext. STX Next. Available at: <https://www.stxnext.com/blog/why-use-react-native-your-mobile-app/#:~:text=React%20Native%20is%20great%20for,compromise%20on%20quality%20and%20functionality>. (Accessed: January 1, 2023).
- [10] Pisuwala, U. (no date) The benefits of reactjs and reasons to choose it for your project,

Peerbits. Available at: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html#:~:text=One%20of%20the%20main%20benefits,other%20parts%20of%20the%20application>. (Accessed: January 1, 2023).

[11] Why bootstrap is an ideal design framework | clutch.co (no date). Available at: <https://clutch.co/web-designers/resources/why-bootstrap-is-ideal-design-framework> (Accessed: January 1, 2023).

[12] What is node.js and why you should use it (2022) Kinsta®. Available at: <https://kinsta.com/knowledgebase/what-is-node-js/> (Accessed: December 30, 2022).

[13] Simplilearn (2021) Why use mongodb: What it is and what are the benefits, Simplilearn.com. Simplilearn. Available at: <https://www.simplilearn.com/tutorials/mongodb-tutorial/what-is-mongodb> (Accessed: December 30, 2022).

[14] Barney, N. and Gillis, A.S. (2022) What is AWS (Amazon Web Services) and how does it work?, SearchAWS. TechTarget. Available at: <https://www.techtarget.com/searchaws/definition/Amazon-Web-Services> (Accessed: January 1, 2023).

Chapter 9

Appendices

Appendix A. Individual Contribution

204222R – G.V.G. Uththara

In our online appointment management system, I am responsible for development of both front end and back end of the customer profile interface, date and time selection interfaces and repairment completion interface.

After analyzing the customer requirements we gathered, we started to design UML diagrams. I gave my contribution to my group members to draw Use Case diagram, Class diagram and ER diagram. I individually drew activity diagrams and sequence diagrams for the respective functionalities I am responsible of. It helped me to know exactly what we are getting, in advance since software system is professionally designed and documented before it is coded.

We designed the prototypes of the interfaces of both mobile application and web application using Figma. It helped me to get an idea about how I can make interfaces more user friendly and how I can improve my UI designs in order to provide a higher customer satisfaction.

Currently I am learning React Native and ReactJS from YouTube tutorials and online resources as we use those technologies are used as front-end technologies of our system. And also, I am learning Nodejs and MongoDB since we use it as our back-end technology and database technology respectively. Once the front-end development of the interfaces is done coding of the back end will be started.

I hope to give my best to approach a successful final product along with my group members.

204044X – M.P.A.Disara

In this project I'm responsible for the development of both frontend and backend parts of login interface of the mobile app, the customer registration form, the development of the advance payment function and the branch update function. The customer should be registered to login to the mobile app in order to book an appointment. He/she should register by filling the customer registration form. Once it is done, the customer can login either by entering the username and password or he/she can sign in with Google. If he/she forgets the password, the "Forgot Password" option can be used to recreate a password. Once clicked on it, it will request for the email address and send a code to the email. The sent code should be entered in the app and if correct, the customer will be asked to enter the new password.

In order to confirm an appointment, the customer has to pay a small amount as an advance. It will be deducted by his/her mobile account once he/she confirms payment. We use the support of a Cash API given by hSenid mobile, to handle this process. The Cash API requests for the contact number of the customer, which will be given by retrieving data submitted via the registration form. The deduction of the amount and checking credit balance of the customer's phone will all be done by the Cash API. I will be reading more and educating myself and my group members regarding the integration of this API to our system along with the admirable support and guidance given by our mentor from the company.

We have already drawn the Use case diagram, class diagram and ER diagram as a group whereas the activity diagrams and sequence diagrams were drawn individually according to the tasks distributed among us. I believe that we got a very good exposure on how each and every diagram relate to one another and how prominent each diagram is for the development of a new software. Moving on, we designed all our mobile app and web interfaces using Figma which probably gave us a clear picture of the product we are ultimately giving birth to. I designed the interfaces I was responsible to design and helped with some others too.

Now that we are in the stage of development, I started following YouTube tutorials on React JS and React Native, which are the two technologies that we hope to use for the frontend development of the website and the mobile app respectively. Once the front-end development is done, I'll be moving on to the backend and the database areas which are going to be the most crucial but most exciting parts of this project.

It is my hope to give the best possible outcome for the job I was given, with much dedication.

204244K – N.D. Jayasinghe

I am responsible for the parts that carries the company/service center for their registration and allows the customers to know the details about the company service centers. Customer can use the mobile application for the getting the details such as the branch name and address where the branch is located, telephone no etc. Also, customer can put the rating for the company service center branch if they liked. If not, customer can move to the issue submission form.

Company/service center can use the web application for registering process. Company service center admin can register/login into the system. If the admin wants to register, he wants to enter the details with the registration number. There is an app admin to check the whether the registration number is valid or not. If not, customer should reenter the registration number. The entered details of the company admin are saved in the database. Also, there is an option to login though the company admin forgot the password. Also, there is an option to login with google.

I designed the Use case diagram, Class diagram and The ER diagram along with the group members. I have designed the Activity diagram, Sequence diagram for myself for the features that I am responsible with.

We will start developing the functions of the front-end. For that I still studying ReactJs. Then we will start to develop backend functions, developing databases and developing other remaining functions.

204137K-Nethmini S.A.R.

I am responsible for the development of both the frontend and backend parts of displaying the service centers on the mobile app. When a customer selects the service center option on the mobile application, I have to display the list of service centers (location-wise/product wise search is also possible) and the details, rating option of the selected service center.

I am also responsible for the development of both frontend and backend parts of the dashboard, appointment interfaces and company/service center profile on the website.

After the customer submits his/her issue (from the mobile application), it will be shown in the appointment tab. Then the company admin can confirm/reject the appointment. If company admin confirms the appointment, the name and contact number of the customer will be displayed on the dashboard next to the time slot. I must also provide the company or service center the ability to edit their profile and change their password.

We started off our project by conducting a literature review. We drew the UML diagrams, which included the use case diagram, class diagram, activity diagram, and sequence diagram. I have created the activity diagram and sequence diagram for the features that fall under my scope. I contributed to the drawing of the ER diagram as well. Figma was used to design the interfaces. I contributed to create the UX design as well.

Now I'm referring to certain YouTube videos about "ReactJs" and "React Native". Also, I am following some online materials for both and working on some interfaces. We hope to work on the backend component and databases after the frontend development is complete.

204190N- B.L.P. Sankalpana

I am responsible for the features that allow customers to report issues to the company and for those that allow the company to respond to those issues. The customer can use the mobile app to submit their problem. The correct invoice number must be provided by the customer when submitting an issue to a company. Once submitted, the company will check the invoice number first, and if it is verified, the company admin will inquire about the problem. When technician makes a decision about whether they can fix the problem or not, company/service center admin will send an SMS to the customer.

After receiving approval for our project, we conducted a literature review and discovered that there aren't many existing systems that are similar to what we're going to create.

Since we drew the Use case diagram, Class diagram, and ER diagram as a team, I have contributed to their creation. For the features for which I am responsible, I have drawn the activity diagram and sequence diagram. It provided us with a detailed understanding of how the features operate and relate to one another.

We gathered to discuss the UX/UI designs structure after drawing the diagrams. We devote a great deal of time planning the interfaces' flow together. Then, using Figma, we design the interfaces for which we are responsible.

The development of the front end is our next step, so I'm currently learning ReactJs and React Native, which we'll be using for those development tasks. In order to get started on the development, I referred some online guides and YouTube videos for ReactJs and I'm currently working on some interfaces.

After finishing the frontend development, we hope to work on the backend part and databases to finish the project in the highest possible quality.

Appendix B. Action Plan

Task	January	February	March	April	May
Develop the front end					
Design database					
Develop the back end					
Integrate front end with back end					
Complete non functional requirements					
Testing & Trouble shooting					
Trial Run					
Deploy final product &Final report preparation					

Table 1.2 Action Plan

Appendix C. Mockups

Interfaces of Mobile Application

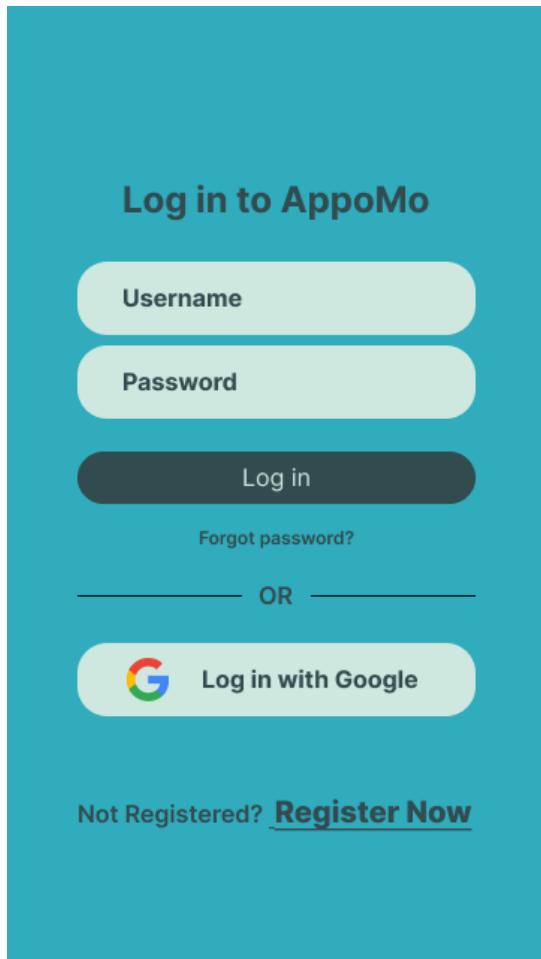


Figure 9.1 Mockups - Customer Login

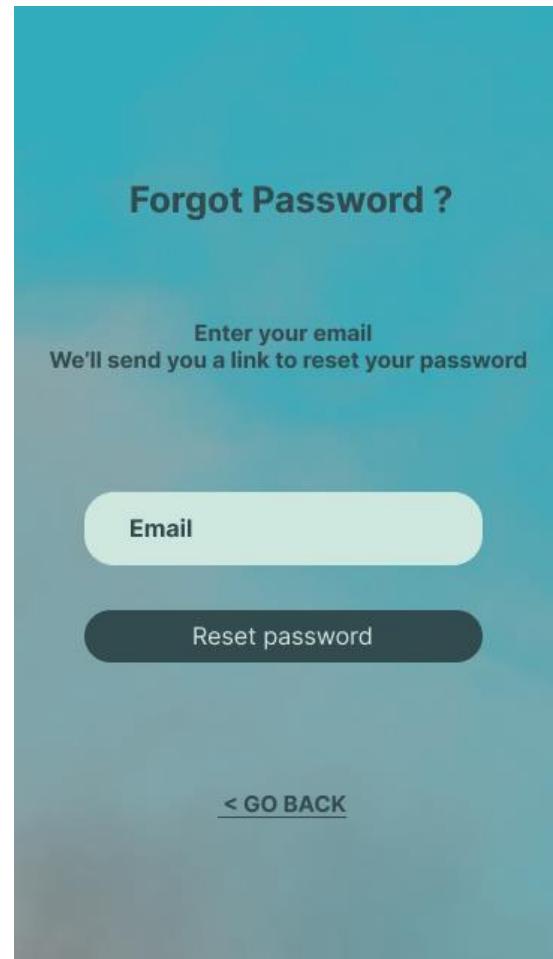
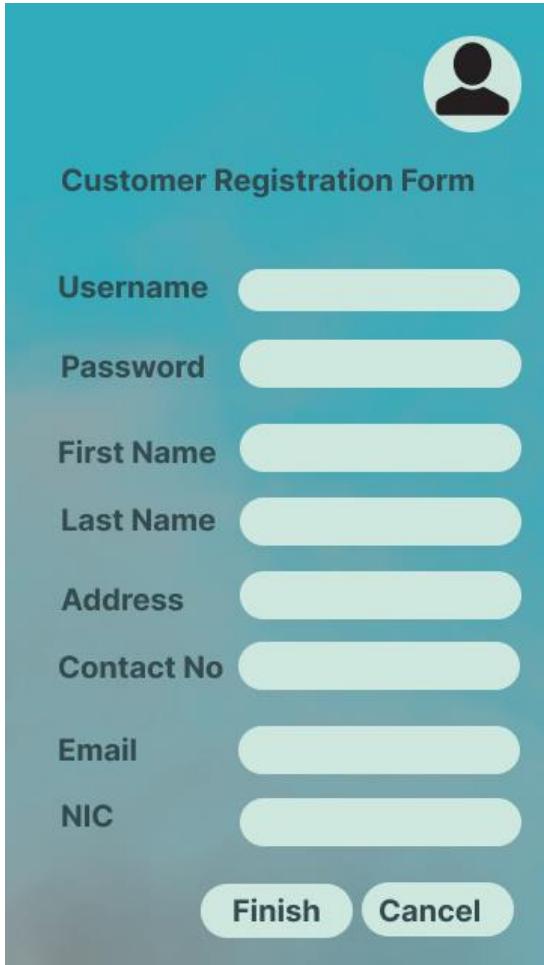


Figure 9.2 Mockups - Forgot Password



Customer Registration Form

Username

Password

First Name

Last Name

Address

Contact No

Email

NIC

Finish **Cancel**

This mockup shows a customer registration form with a teal header and footer. It includes fields for Username, Password, First Name, Last Name, Address, Contact No, Email, and NIC, each with a light blue placeholder bar. At the bottom are 'Finish' and 'Cancel' buttons.

Figure 9.4 Mockups - Customer Registration

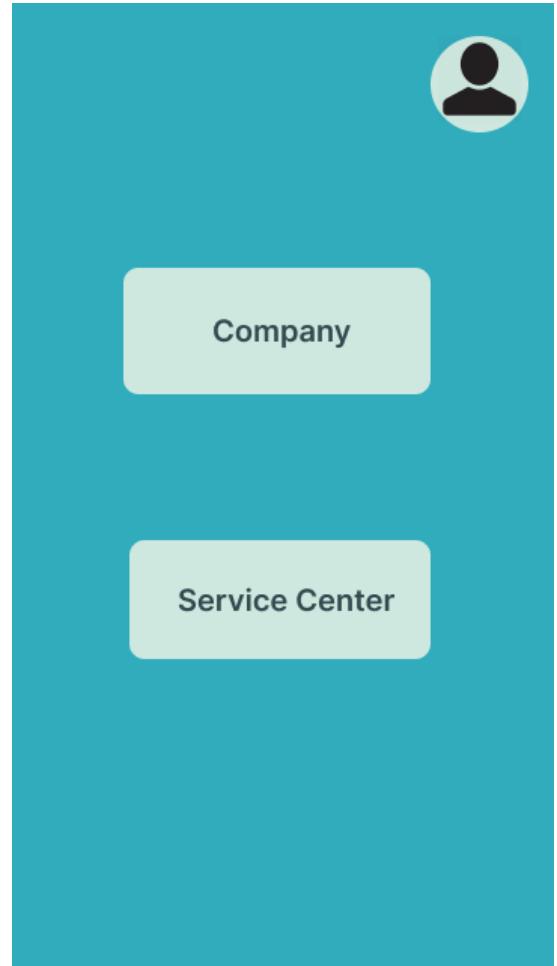


Figure 9.3 Mockups - Customer Dashboard

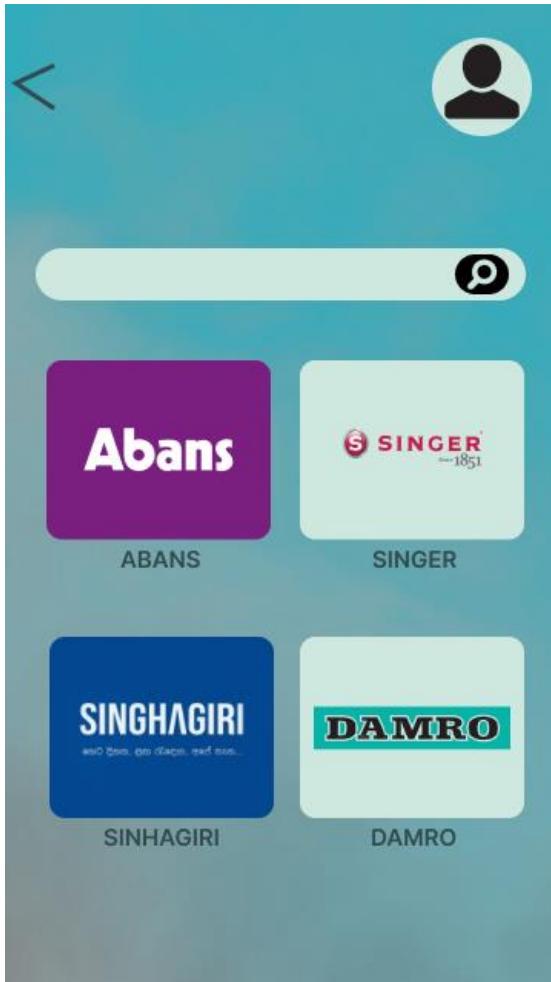


Figure 9.6 Mockups - Choose a Company Option

This mockup displays detailed information for the selected company, Abans. At the top right is a user profile icon. Below it is a search bar with a magnifying glass icon. The title "Abans" is prominently displayed in a purple header. The main content area includes:

- Service Center Name**: Anuradhapura, Badulla, Batticaloa, Dambulla.
- Address**: 521/87, 5th Lane, New Bus Stand, Anuradhapura; No. 46, Hunukotuwa Rd, Badulla; No 48, Station Road, Batticaloa; No 33D, Kurunegala Road, Dambulla.
- Tel.no**: 0382245789, 0349907650, 0382189033, 0342267123

Below this is a "Rating" section with five stars (three yellow, two white) and a "Submit your issue" button at the bottom.

Figure 9.5 Mockups - View Company Details

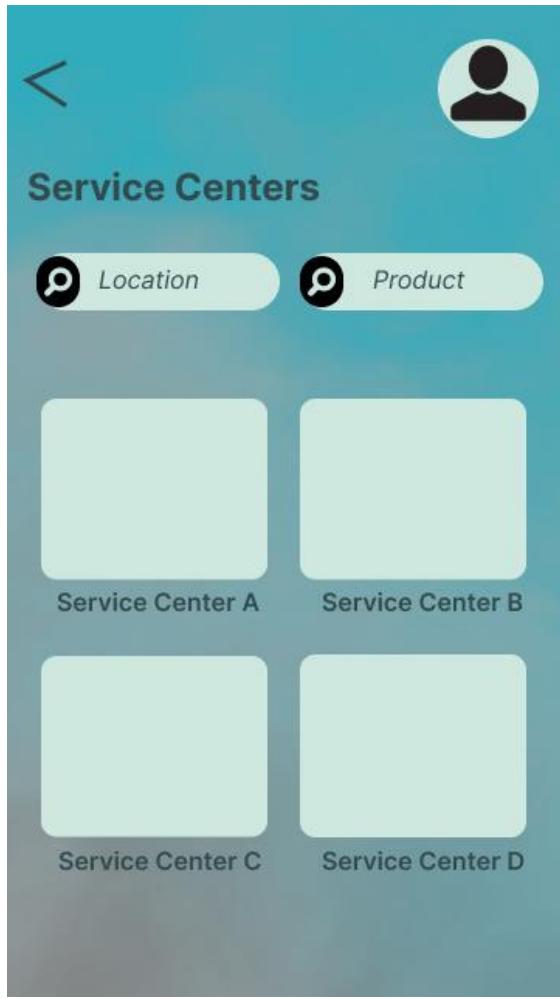


Figure 9.7 Mockups - Choose a Service Center Option

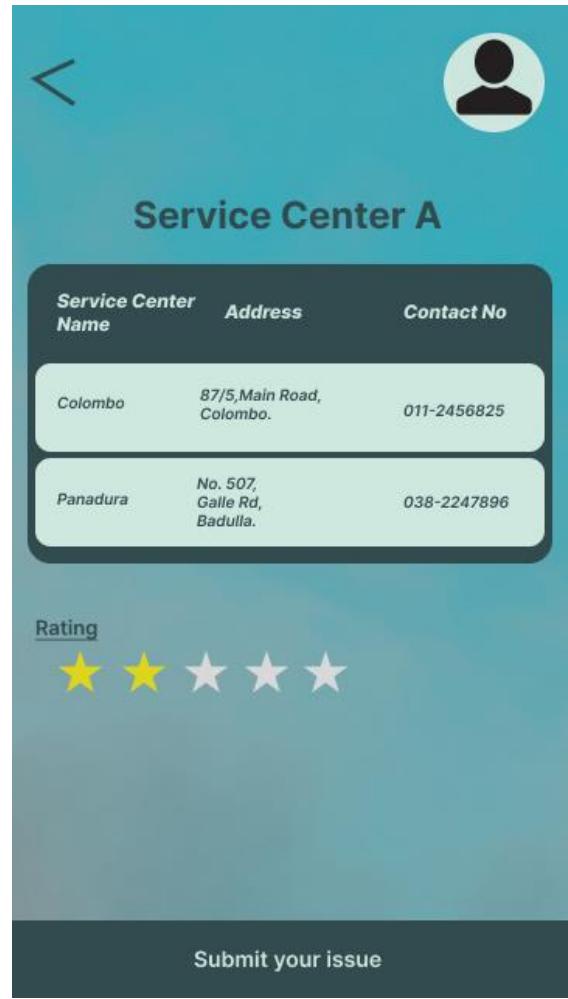


Figure 9.8 Mockups - View Service Center Details

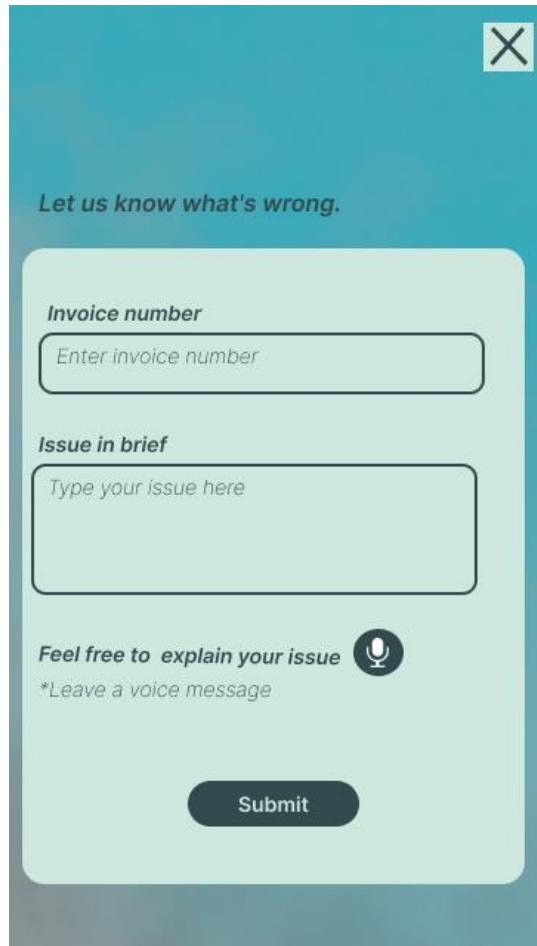


Figure 9.10 Mockups - Issue Submission

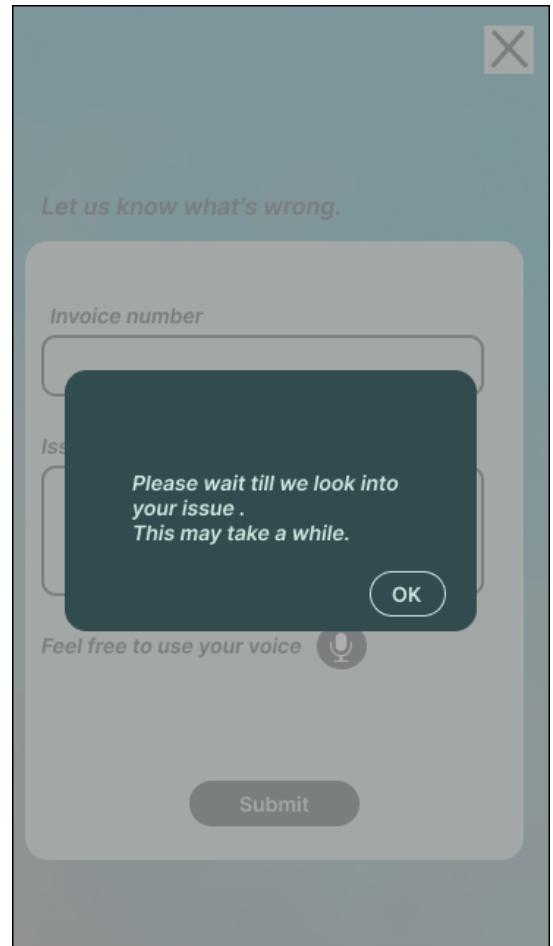


Figure 9.9 Mockups - Waiting for the Acceptance/Rejection of the Issue

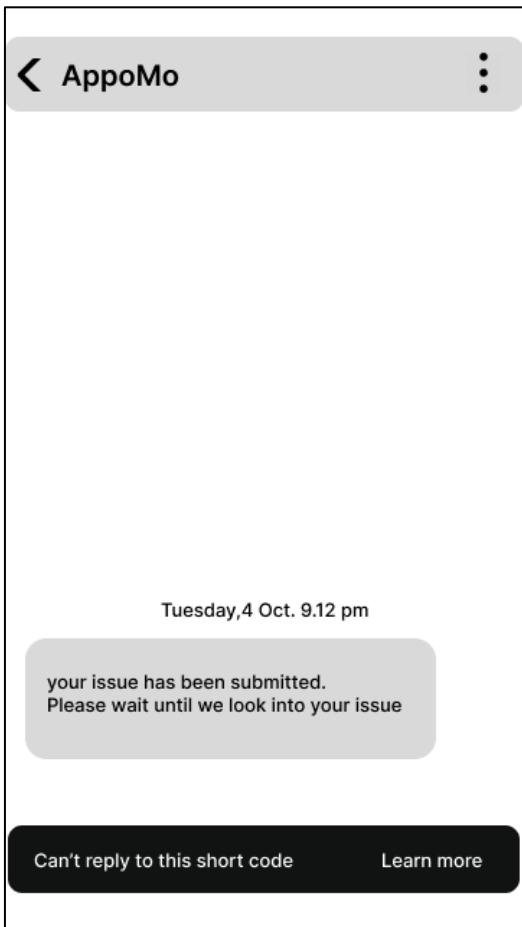


Figure 6.2.11 Mockups – SMS of Issue Submission

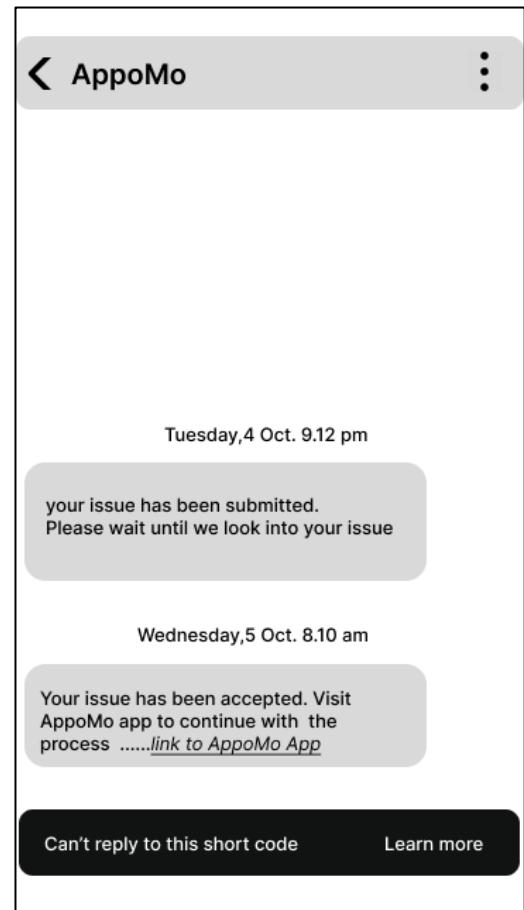


Figure 6.2.12 Mockups – SMS of Issue Acceptance

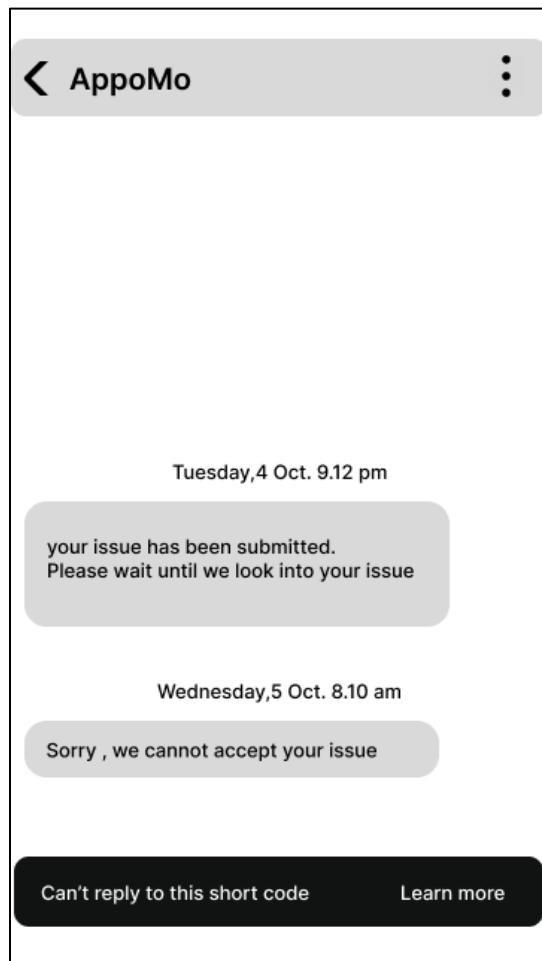


Figure 6.2.13 Mockups – SMS of Issue Rejection

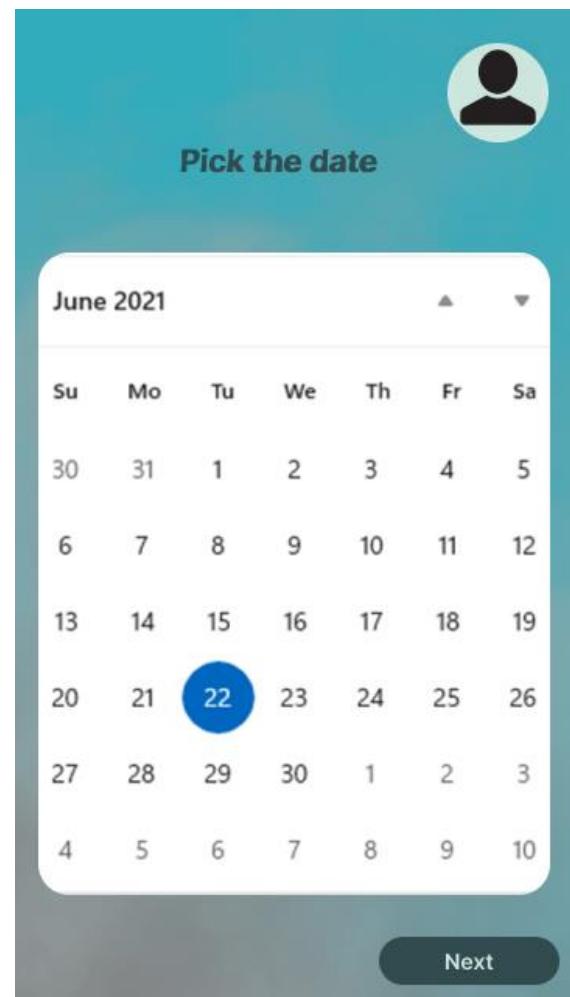


Figure 6.2.14 Mockups – Date Selection

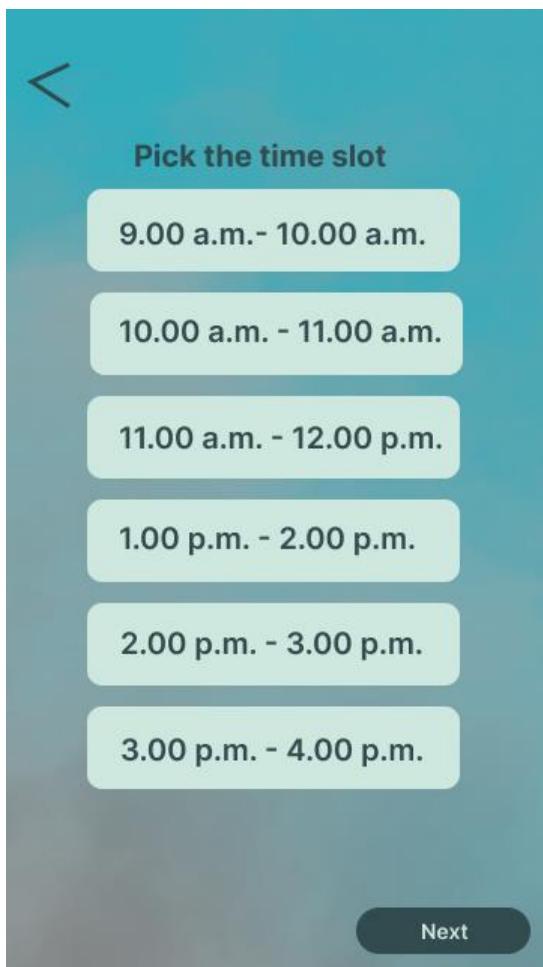


Figure 6.2.16 Time Slot Selection

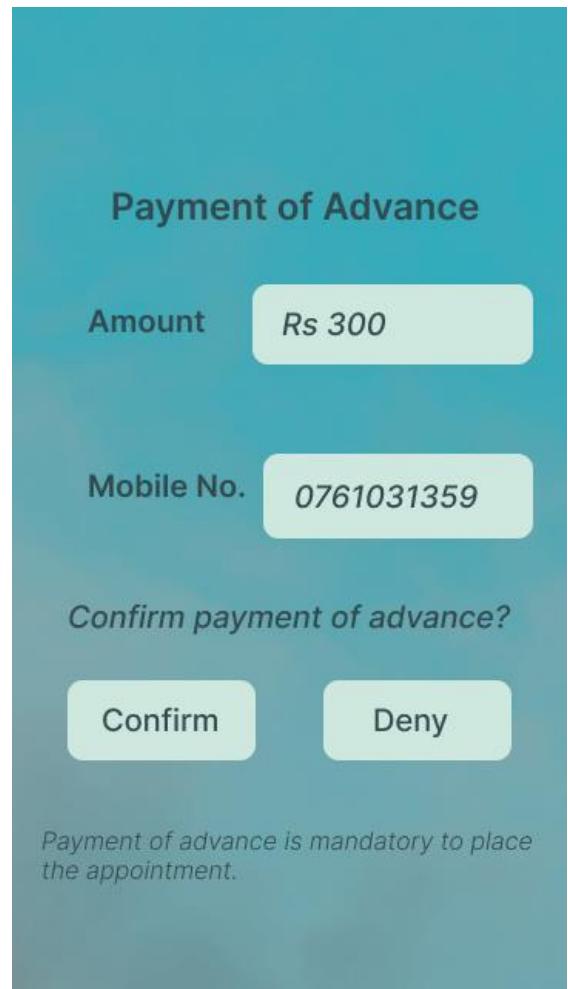


Figure 6.2.15 Mockups – Payment of Advance

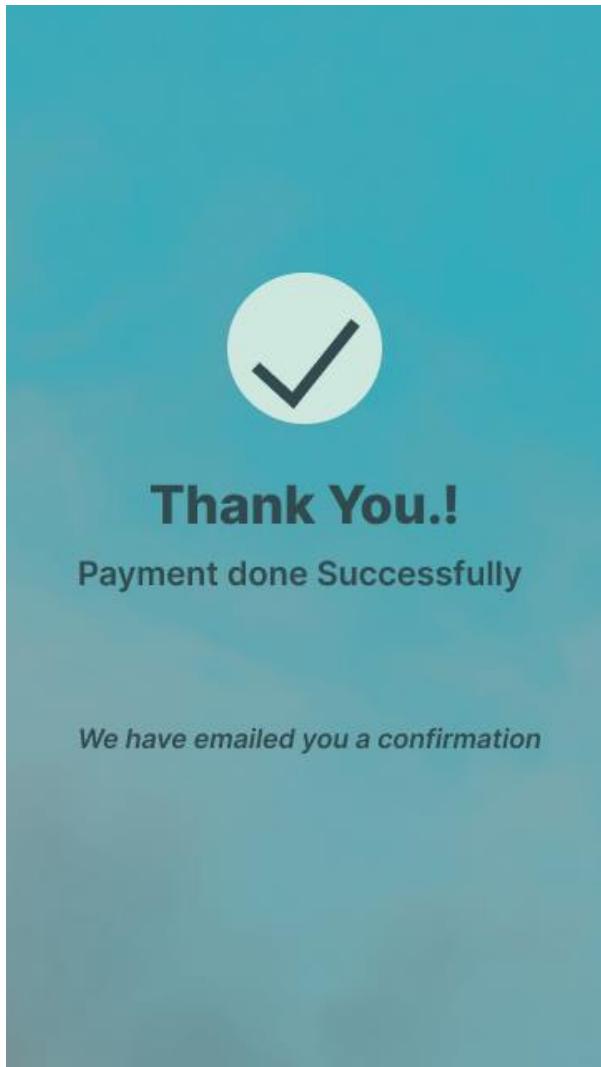


Figure 9.12 Mockups – Payment Successful Message

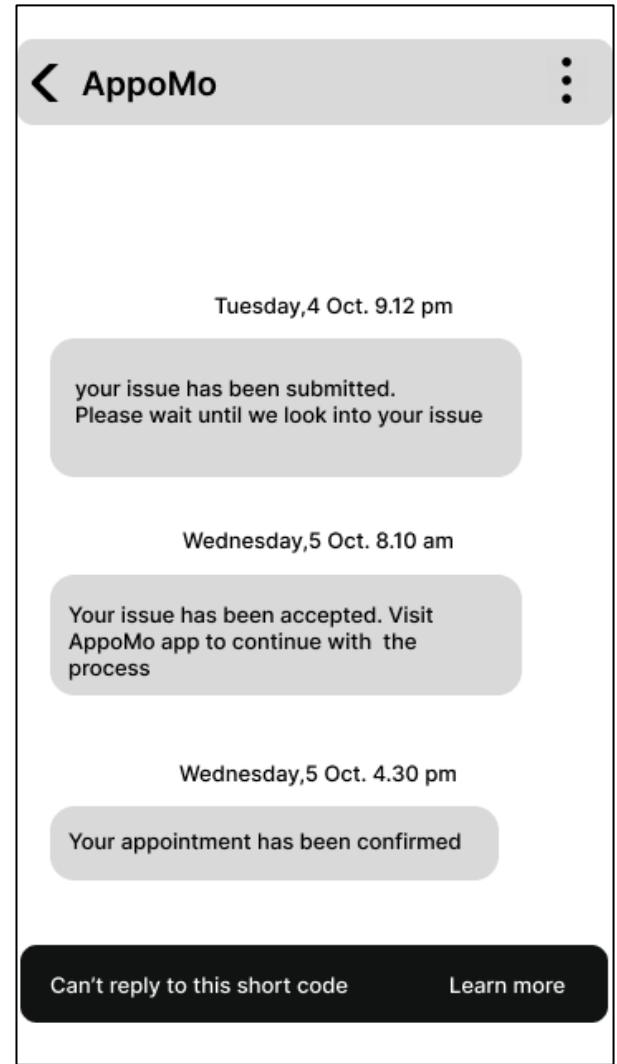


Figure 9.11 Mockups – SMS of Appointment Confirmation

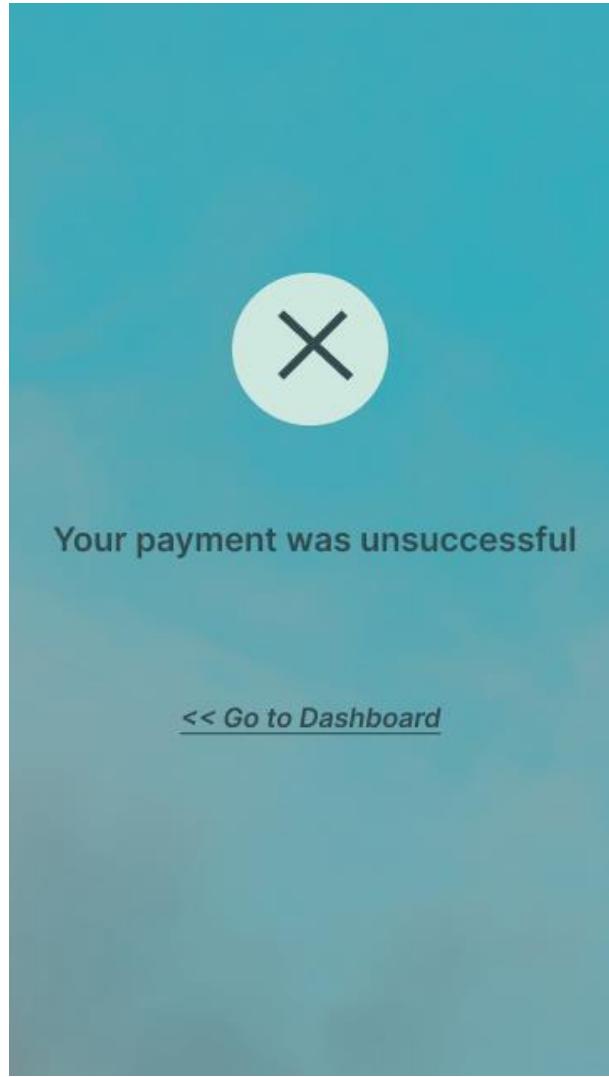


Figure 9.14 Mockups – Payment Unsuccessful Message

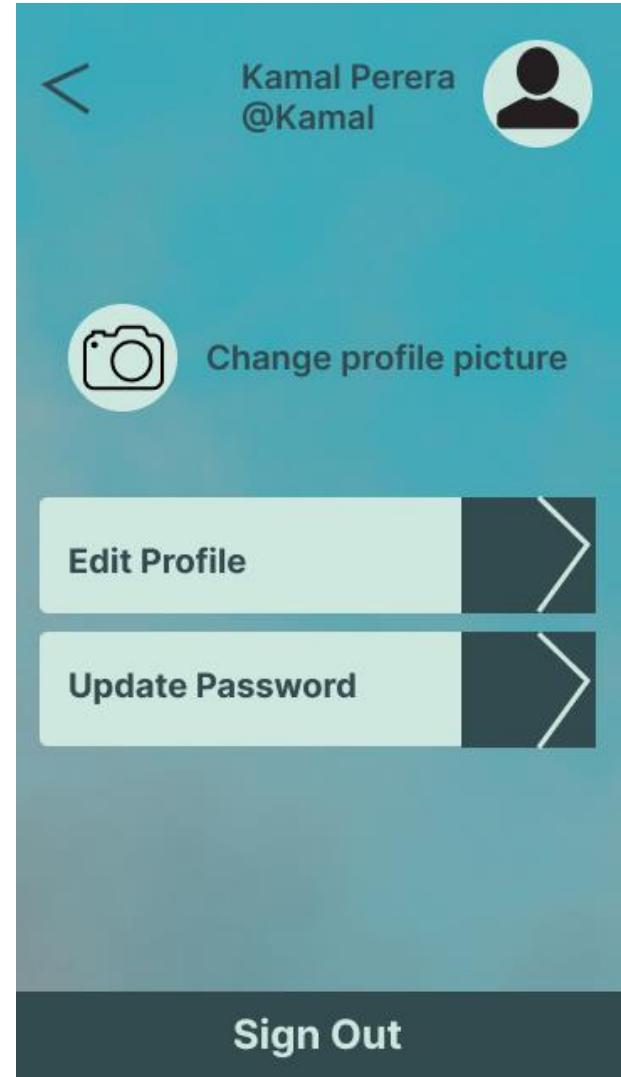
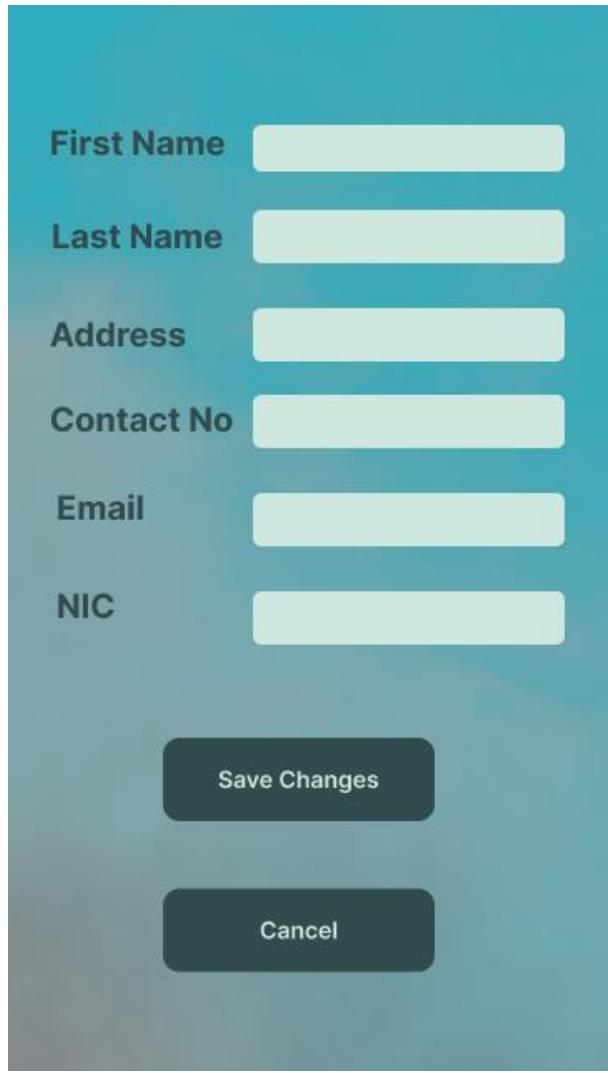


Figure 9.13 Mockups – Customer User Profile

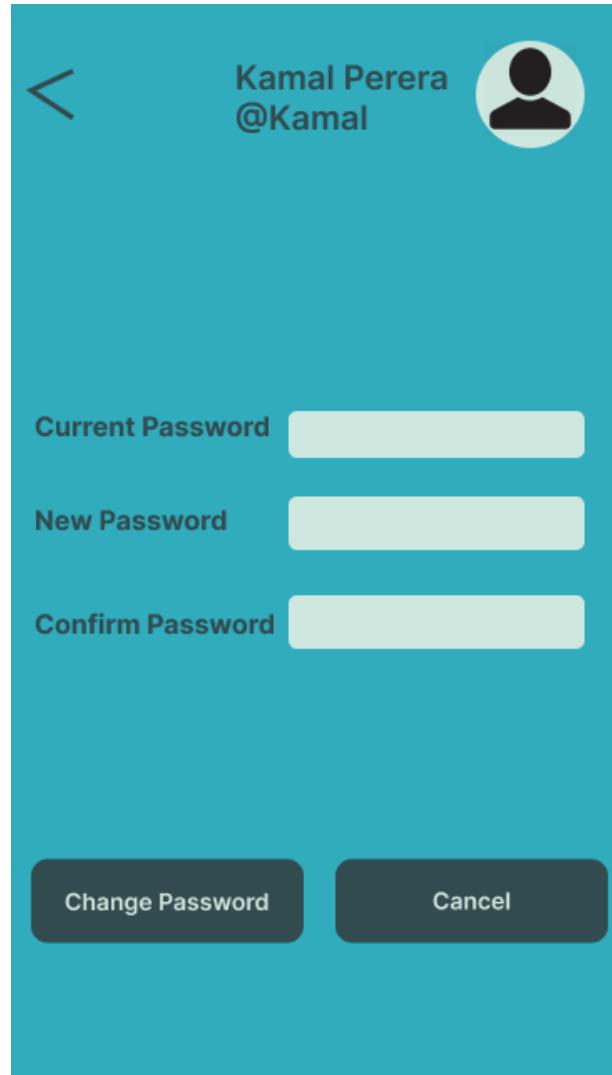


A mobile application screen titled "Edit Profile". It features a list of input fields for personal information:

- First Name
- Last Name
- Address
- Contact No
- Email
- NIC

Below the input fields are two buttons: "Save Changes" and "Cancel".

Figure 9.15 Mockups – Edit Profile



A mobile application screen titled "Kamal Perera @Kamal" with a profile icon. It includes a back arrow and a list of password-related fields:

- Current Password
- New Password
- Confirm Password

At the bottom are two buttons: "Change Password" and "Cancel".

Figure 9.16 Mockups - Update Password

Interfaces of Web Application

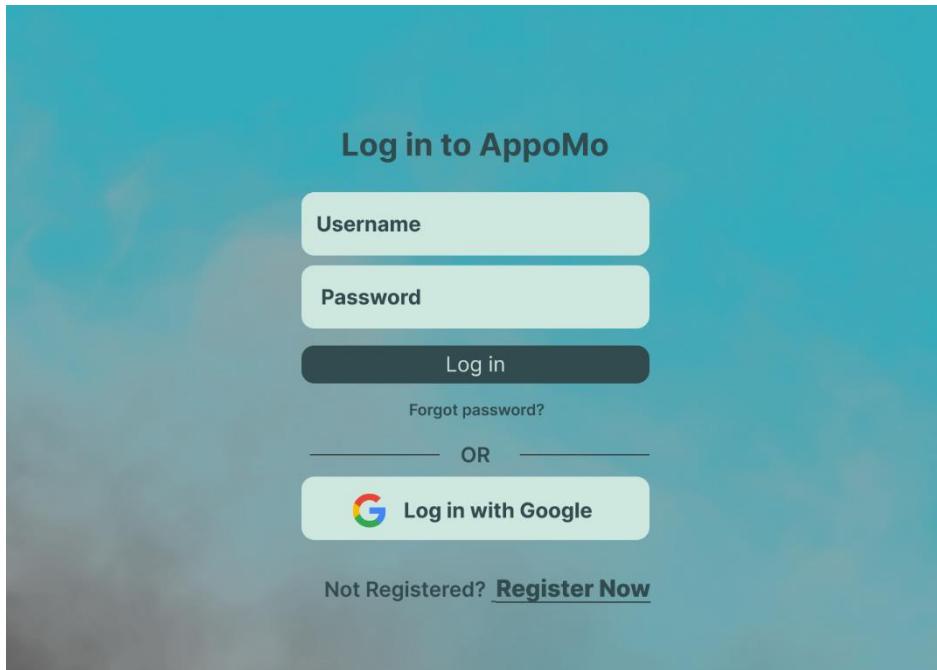


Figure 9.17 Mockups - Admin Login

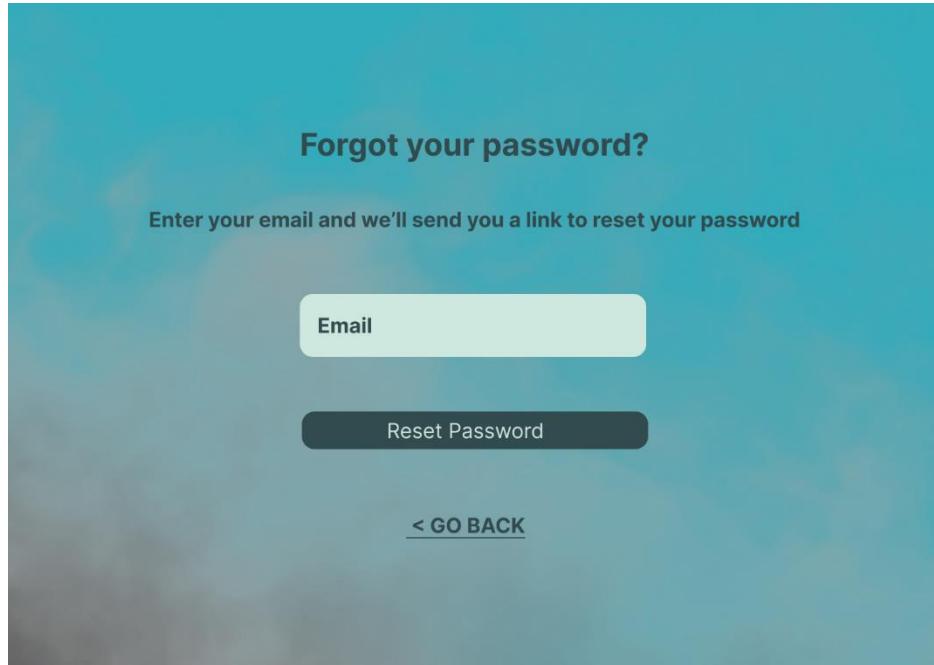
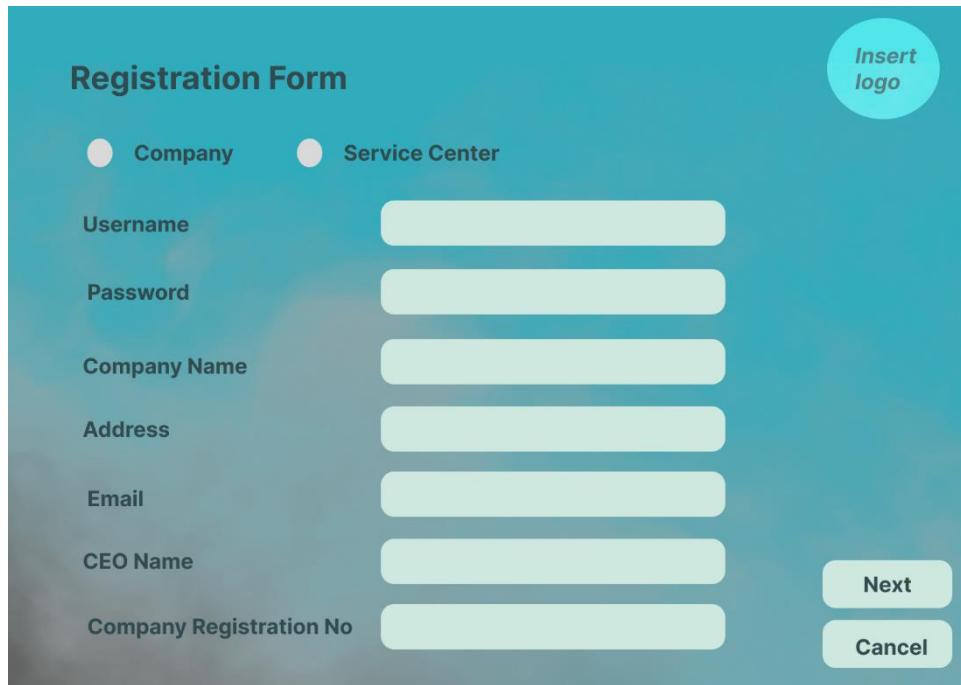


Figure 9.18 Mockups - Forgot Password Web



Registration Form

Company Service Center

Username

Password

Company Name

Address

Email

CEO Name

Company Registration No

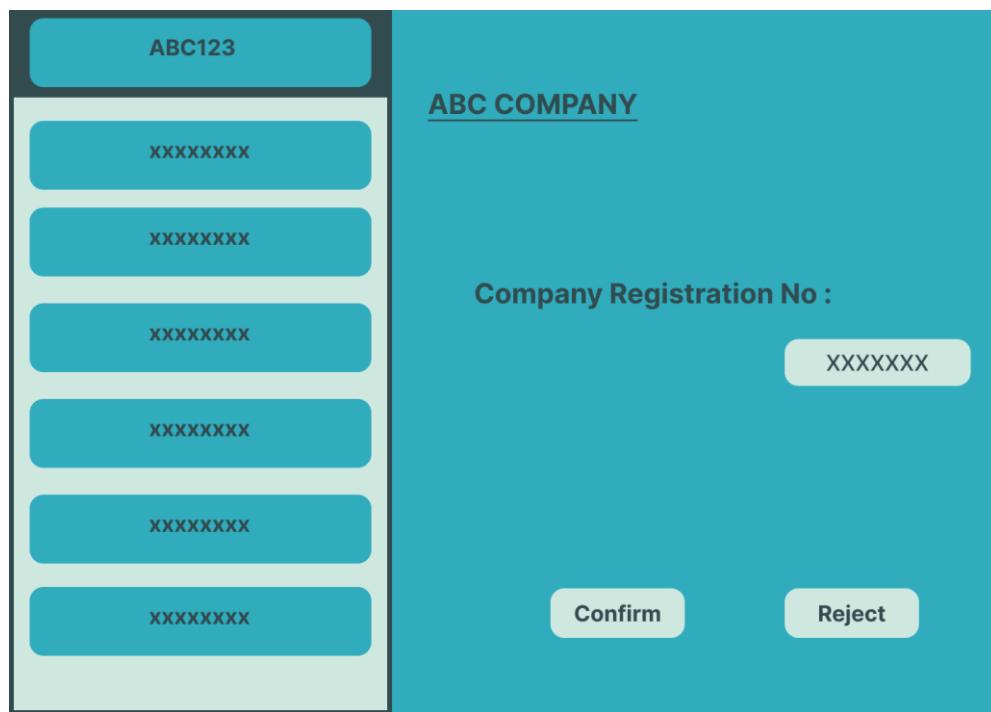
Insert logo

Next

Cancel

This mockup shows a registration form with a teal background. At the top right is a circular placeholder for a logo with the text "Insert logo". Below the logo are two radio buttons for selecting "Company" or "Service Center". The form fields include "Username", "Password", "Company Name", "Address", "Email", "CEO Name", and "Company Registration No", each with a corresponding input field. At the bottom right are "Next" and "Cancel" buttons.

Figure 9.19 Mockups - Company/Service Center Registration



ABC123

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

ABC COMPANY

Company Registration No :

XXXXXXX

Confirm

Reject

This mockup shows a verification screen. On the left is a vertical list of registration numbers: ABC123, XXXXXXX, XXXXXXX, XXXXXXX, XXXXXXX, XXXXXXX, and XXXXXXX. To the right is the company name "ABC COMPANY" underlined. Below it is the label "Company Registration No :" followed by a placeholder input field containing "XXXXXXX". At the bottom are "Confirm" and "Reject" buttons.

Figure 9.20 Mockups - Registration Number Verification

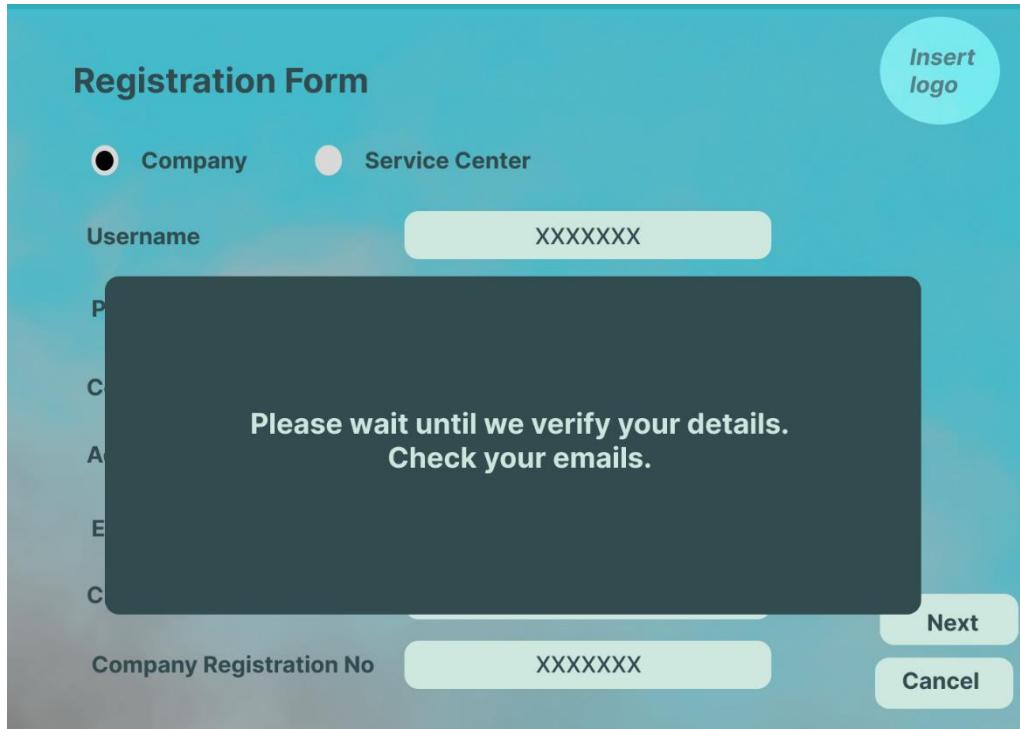


Figure 9.21 Mockups - Waiting for Registration Number Verification

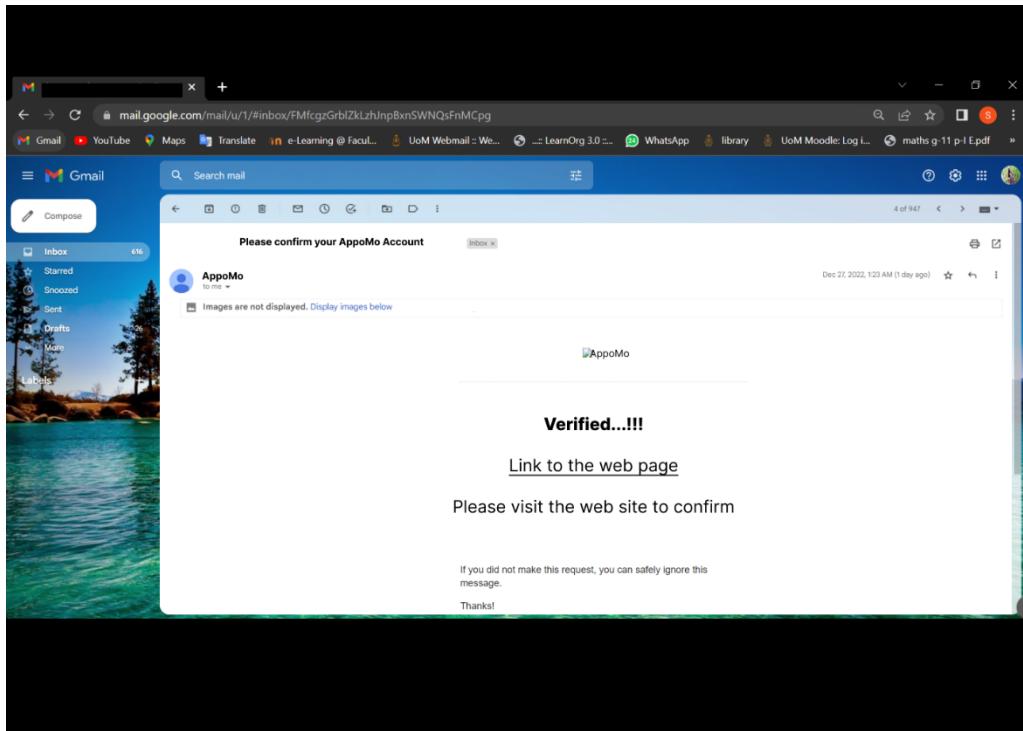


Figure 9.22 Mockups - Registration Number Verification Email

Branch Details

Weekdays - Open	From	To
Weekends- Open	From	To
No of appointments per hour		

Fill if the company has branches other than the main branch

Branch Name	Name of Manager	Contact No	Address

Add New **Update** **Finish**

Figure 9.23 Mockups - Manage Branch Details

The dashboard displays appointment details for three time slots:

- 9 a.m. - 10 a.m.: Rashini Nethmini, 0769970184
- 9 a.m. - 10 a.m.: No appointments
- 10 a.m. - 11 a.m.: Prasadi Sankalpana, 0779684785

Below the main section, there is a date selector showing "2023.02.12" and another appointment entry for "9 a.m. - 10 a.m." which also shows "No appointments".

Figure 9.24 Mockups - Dashboard

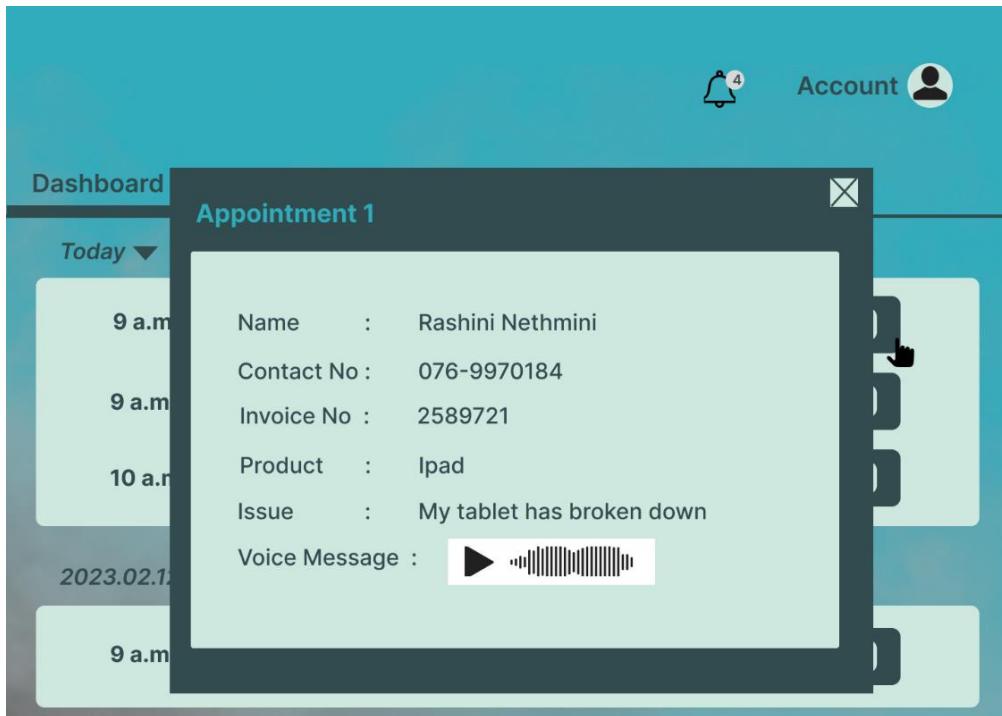


Figure 9.25 Mockups - View Appointments

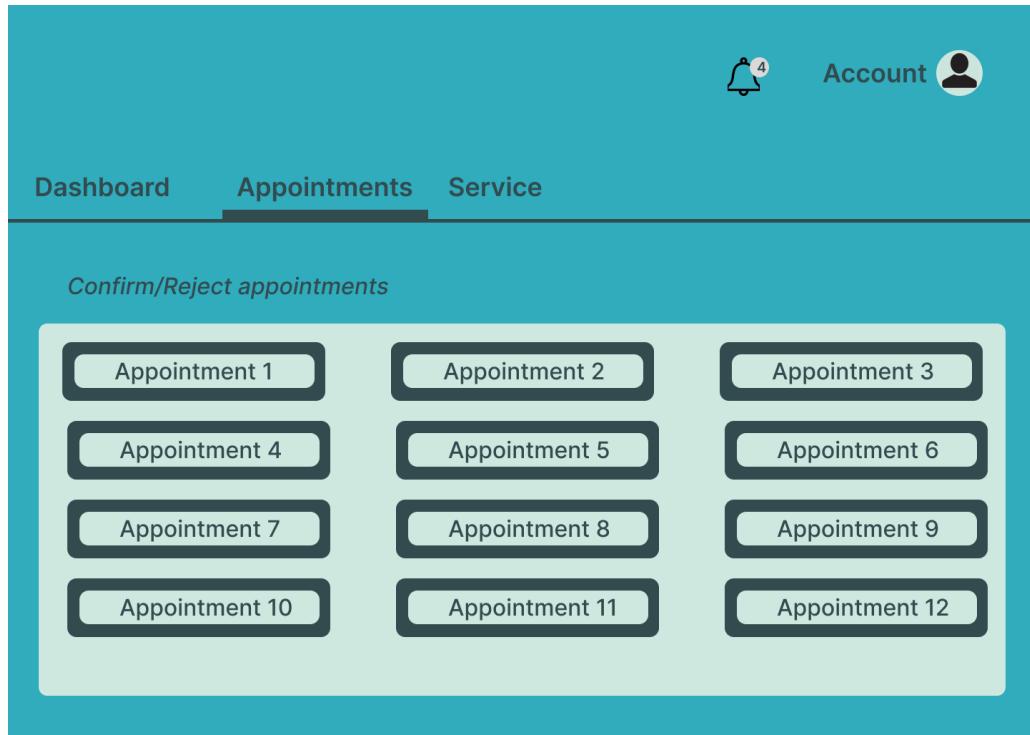


Figure 9.26 Mockups - Appointments

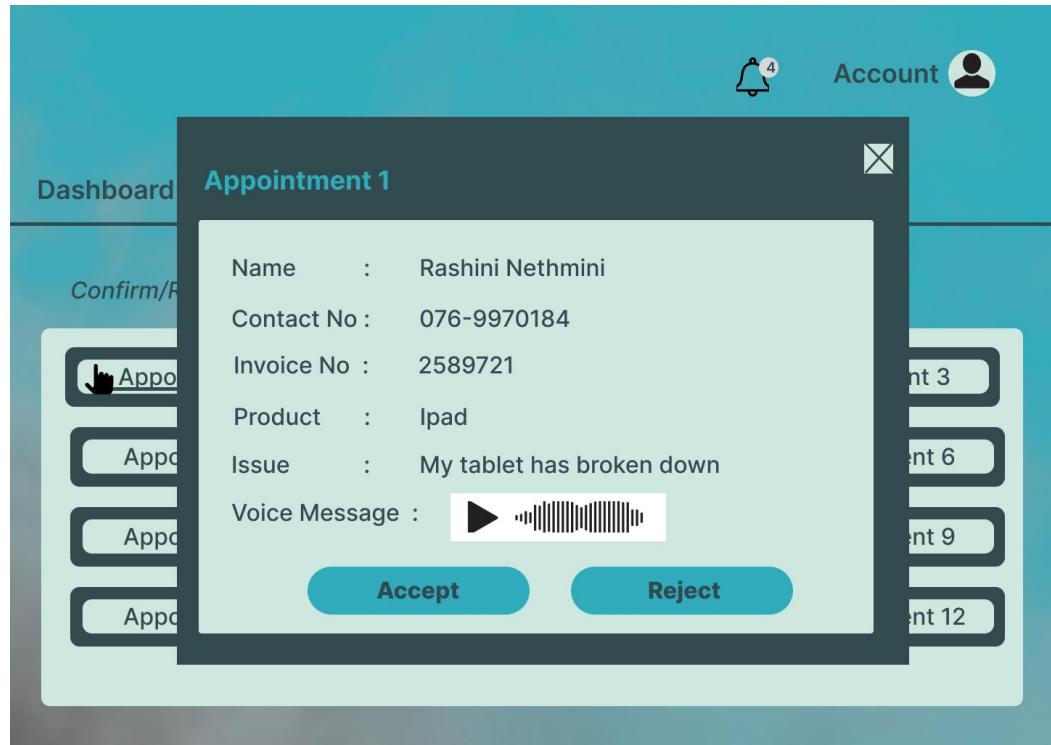


Figure 9.27 Mockups - Issue Accept/Reject

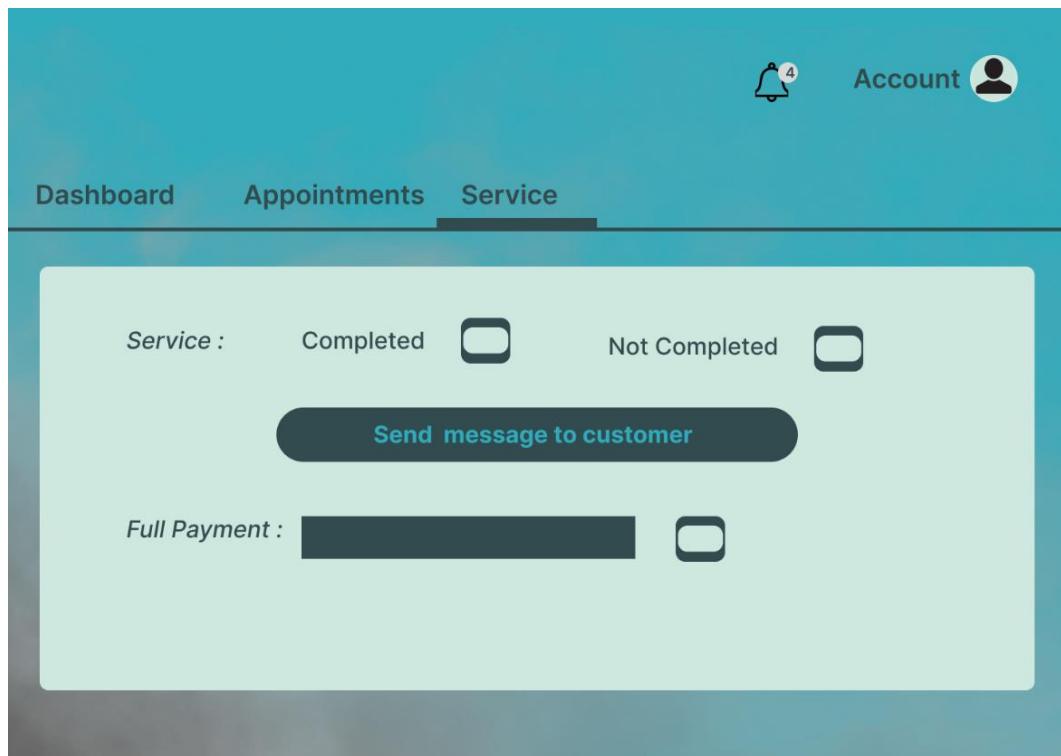


Figure 9.28 Mockups - Service Completion

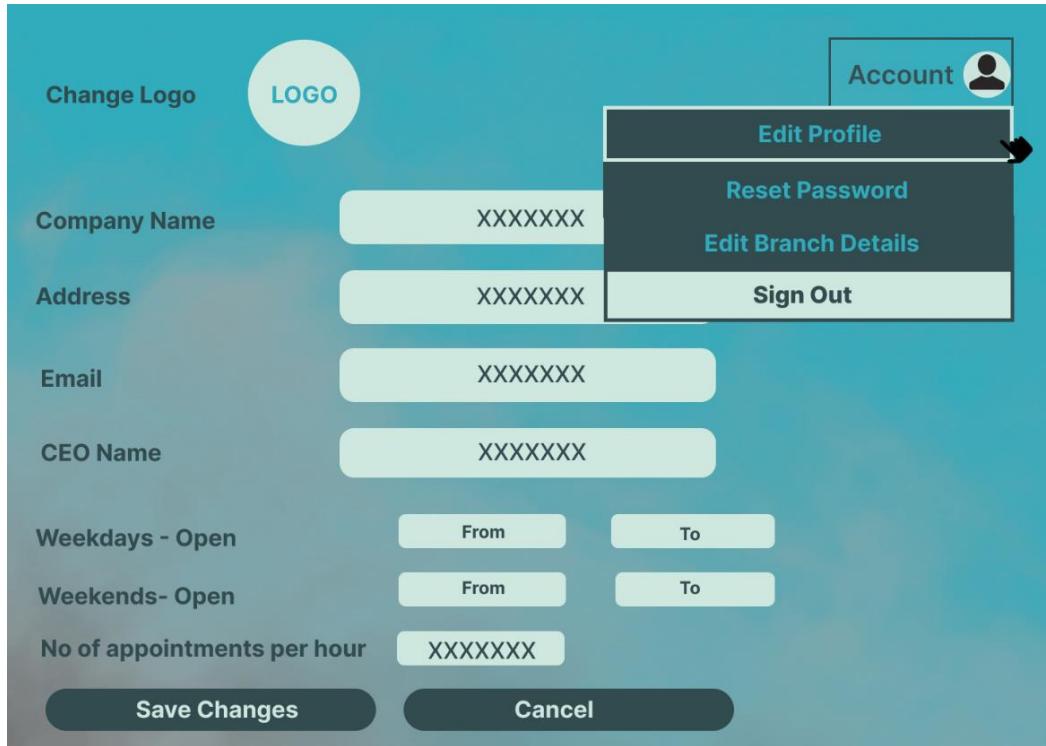


Figure 9.29 Mockups - Company/Service Center Profile

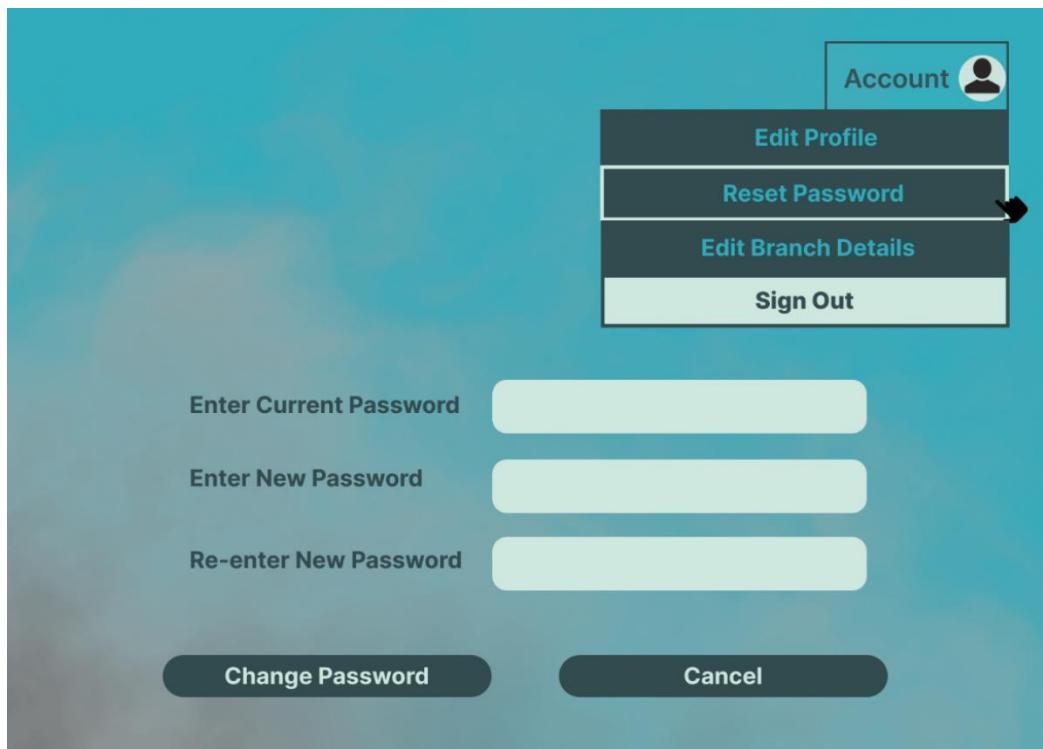


Figure 9.30 Mockups - Reset Password

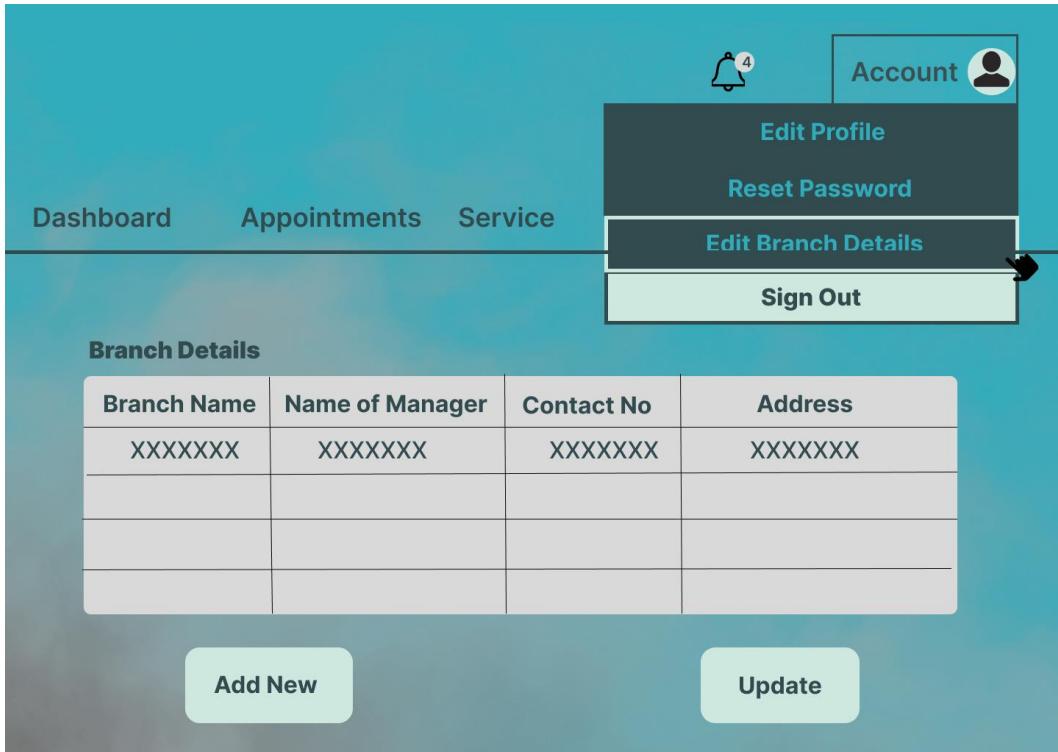


Figure 9.31 Mockups - Edit Branch Details

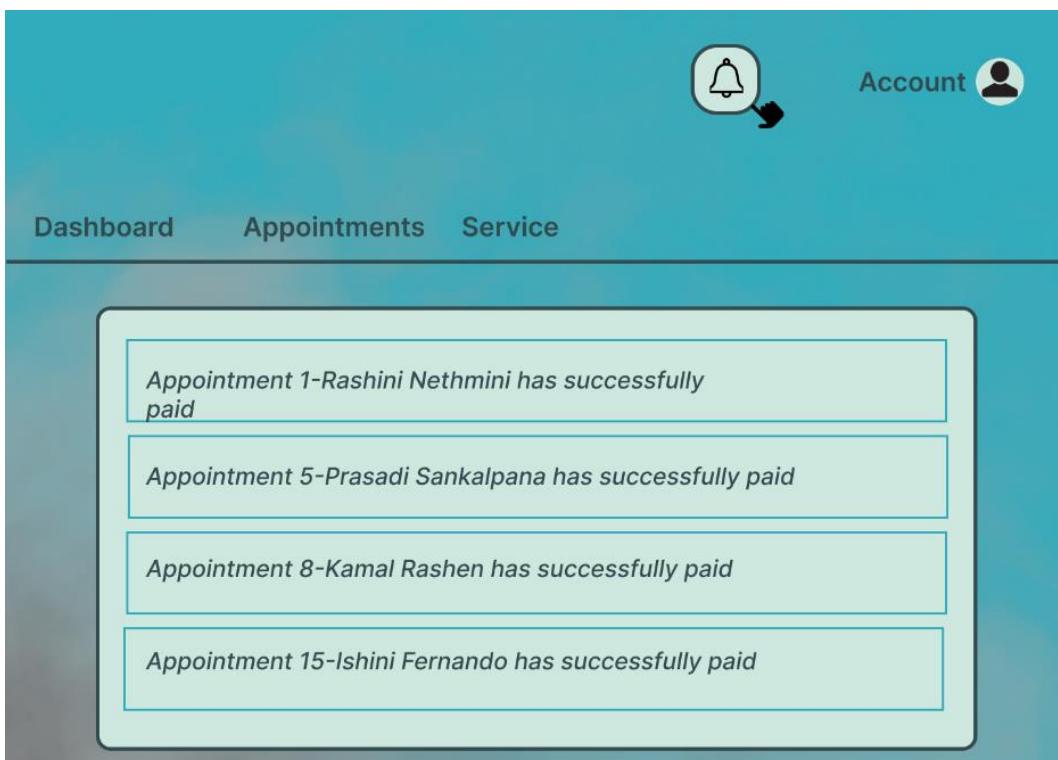


Figure 9.32 Mockups - Service Termination

Appendix D. Software Requirements Specification

The software requirements specification report has been attached to this document as an appendix. Note that captions, page numbers, figures numbers, and table numbers in the SRS document are not linked with the project interim report.

Software Requirements Specification

For

Appointment management System

Version 1.2

Prepared by Encrypto

Faculty of Information Technology

University of Moratuwa

Contents

List of Figures.....	3
List of Tables	3
Revision History	4
1.Introduction.....	5
1.1 Purpose	5
1.2 Document Conventions.....	5
1.3 Intended Audience and Reading Suggestions.....	5
1.4 Product Scope.....	5
1.5 References.....	6
2. Overall Description.....	7
2.1 Product Perspective	7
2.2 Product Functions	7
2.3 User Characteristics	8
2.4 Operating Environment	8
2.5 Design Development and Implementation Constraints	9
2.6 User Documentation	9
2.7 Assumptions and Dependencies.....	9
3. External Interface Requirements	9
3.1. User Interfaces	9
3.2. Hardware Interfaces.....	10
3.3. Software Interfaces	10
3.4. Communications Interfaces	10
4. System Features	11
4.1 Customer Operations.....	11
4.1.1. Description and Priority.....	11
4.1.2. Functional Requirements	11
4.2 Company Admin Operations	12
4.2.1. Description and Priority.....	12
4.2.2. Functional Requirements	12
4.3 App Admin Operations	12
4.3.1. Description and Priority.....	12
4.3.2. Functional Requirements	12
5. Other Non-functional Requirements.....	12
5.1. Performance Requirements	12
5.2. Safety Requirements.....	13

5.3. Security Requirements	13
5.4. Software Quality Attributes.....	13
5.5. Business Rules	14
5.6. Deployment prerequisite	14
5.6.1. On premise deployment prerequisites.....	14
5.6.2. Cloud deployment environment	14
6. Appendix.....	15
6.1. Network diagram	15
6.2. Data flow diagram.....	17

List of Figures

Figure 1- Network Diagram,	16
Figure 2-Context Diagram	17
Figure 3- Level 0 Data Flow Diagram...	18

List of Tables

Table 1 Network diagram overview.....	15
--	----

Revision History

Date	Version	Description
Sep 20,2022	1.0	Initial Version
Oct 30,2022	1.1	Added Network Diagram and Data Flow Diagram
Nov 20,2022	1.1.1	Modified dataflow diagram
Dec 30,2022	1.2	Added revision History to SRS.

1. Introduction

1.1 Purpose

The major purpose of this document is to define and demonstrate the functionality of the appointment management system. This consists of a detailed description about the functional and non-functional requirements and external interface requirements of this system. In addition to that, this document provides a detailed idea on how this application would aid the users in managing their appointments in their day to day lives.

1.2 Document Conventions

This document was prepared according to the IEEE 830-1998 standard for writing SRS documents.

1.3 Intended Audience and Reading Suggestions

This document is intended to be referred by the users, developers and testers of the system to get a clear picture of the functionalities of this system.

All users are recommended to read through the full description of this document whereas developers and testers are recommended to do a keen study on the functional and non-functional requirements and the external interface requirements mentioned in this document.

1.4 Product Scope

The appointment management software is an application where we can select suitable service providers, make appointments, confirm issue mitigation, track repairment status and get notified when the job is done.

The main purpose of this application is to facilitate the customer to choose the best service provider to fix their broken item out of a wide range of service providers island wide and make an appointment to communicate and handover the item to be fixed, before physically visiting the store. The company can decide if the issue can be fixed or not prior to setting the appointment.

1.5 References

- [1] "Appointment management system - DevProjects", DevProjects, 2022. [Online]. Available: <https://www.codementor.io/projects/web/appointment-management-system-compu19a0t>. [Accessed: 21- Sep- 2022]
- [2] IEEE 830-1998 standard for writing SRS documents.
- [3] "Telco API Management Platforms | CPaaS | Enterprise CPaaS | hSenid Mobile Solutions", hSenid Mobile, 2022. [Online]. Available: <https://www.hsenidmobile.com/>. [Accessed: 21- Sep- 2022]

2. Overall Description

2.1 Product Perspective

The aim of this project is to design a mobile and web application that facilitates making appointments, which will ease the lives of both the customers and service providers. Using the mobile app, the customer is capable of selecting the service provider that suits his/her item to be repaired and its circumstance and make an appointment to that particular service provider, by setting a preferred time slot to meetup. This aids the customer to select the most convenient and suitable service provider, confirm whether the particular service provider can definitely fix the issue and furthermore, save time without wasting his/her valuable time in queues at service centres.

Apart from that, the service provider can reduce their time wasted on meetups with customers that bring items that they can't fix. Moreover, they will be able to deliver a more efficient and satisfactory service by investing their time on a specific person at a time. The service provider can confirm the appointment using the web application and once the item is handed over by the customer, the status of the repairment will be notified to the customer via SMS. Once the item is fully repaired, the customer will be notified to come and pick up the item by making the full payment.

2.2 Product Functions

- 1.Customer logs into the mobile application and search for the suitable service provider.
- 2.The customer can sort service providers according to location and product type. (It can be either a branded company or just a service centre)
- 3.Once a preferred service provider is selected, the customer can select the closest or most convenient branch to him/her (if there are many branches of the same service provider).
- 4.After selecting the exact company/service centre, the customer can type the issue with the product or else leave a voice message explaining the issue and submit it.
- 5.If the customer has selected the same company from which he/she bought the item, then he/she has to enter the invoice number with the issue description. Otherwise, the invoice number is not required.
- 6.Once the issue is submitted, the customer receives a SMS requesting him/her to wait until the company confirms that the issue can be practically addressed by them.

7. If the company confirms that the issue can be fixed, an email is sent to the customer asking him/her to visit the app and resume with placing the appointment.
8. The customer then can choose a preferred date and time slot and pay a small advance confirming the appointment.
10. Once the advance payment is made, the company admin regards the appointment as confirmed and reserves the selected time slot.
11. An SMS is sent to the customer with the appointment details at the point of appointment confirmation.
12. When the date of appointment is close, an SMS will be sent to the customer's phone with the appointment details again, as a reminder.
13. Once the item is handed over, an SMS will be sent to the customer informing the status of repairment.
14. Once the job is done, the customer will receive an SMS asking him/her to come, make the full payment and collect the item.
15. When the transaction is completed the customer will receive an email with the payment invoice attached.
16. The company admin can view the appointments that are already confirmed and those which are yet to be confirmed.
17. Both the company admin and users can view and edit their user profiles.
18. The company can add or update its branch details.

2.3 User Charachteristics

Users of this application are customers, service provider admins and app admin.

2.4 Operating Environment

The website can be accessed by any machine that has a windows operating system with a version greater than windows 7 and a stable internet connection.

2.5 Design Development and Implementation Constraints

- Front end of the tool is to be designed using React for web and React Native for mobile.
- We will use Node JS to design the backend of the web and mobile application.
- MongoDB is used as the database technology of this application.
- AWS is used as the cloud system.
- Cash API which is a Telco API provided by hSenid Mobile is used to implement payment of advance functionality.
- SMS API which is also a Telco API provided by hSenid Mobile is used to implement the message sending functionality.

2.6 User Documentation

A user manual is provided with this software to the company for future reference. Apart from that, include online help in the application.

2.7 Assumptions and Dependencies

All the users of the system should be connected to the internet for proper functioning of the system.

3. External Interface Requirements

3.1. User Interfaces

The user is can view his/her user profile, pick company/service centre from the initial interfaces. He/she can edit profile, sign out and reset password as well from the drop down menu. The user can submit issues using the text box option as well as the voice message option.

The company can view appointments both requested and confirmed from the initial interfaces. The company admin also can edit profile, sign out and reset password as well from the drop-down menu. This admin can accept or deny issues by referring to the text and voice message sent by the customer.

3.2. Hardware Interfaces

- In this system we expect to implement as a web-based application and a mobile application system. So, any device that can work with any kind of web browser, can also access without any interruption.
- Hardware specification for normal browsing, will be the minimum hardware requirement for running this application.

3.3. Software Interfaces

- We are using React for web and React Native for mobile to design the front-end because it is easy to learn and is easy to integrate.

Backend of the web application will be built using Node JS. We chose this because of its impressive speed and performance, is easy to share knowledge within a team and it provides better efficiency and overall developer productivity.

- We choose MongoDB as database technology. We omit the MS SQL technology for this project because, MongoDB is faster and more scalable than MS SQL. Moreover, MongoDB accommodates large amounts of data unlike MS SQL.

MongoDB also provides the features like flexibility, flexible query model, native aggregation, and schema-less model.

- A Cash API and SMS API are used for the advance payment functionality and SMS sending functionality.

3.4. Communications Interfaces

- We have planned to host the project in a cloud system.
- We intend to implement the external API(Cash and SMS) for communication in this system.
- When sending and saving passwords we use each related encrypting methods.
- Application and backend communication will happen on HTTPS web requests.

4. System Features

This illustrates organizing the functional requirements for the product by system features, the major services provided by the product. We can divide the functional requirements into following categories.

4.1 Customer Operations

4.1.1. Description and Priority

Every customer that logs into the mobile app is a user and has a separate user profile. They are the ones who place the appointments with the preferred service provider.

4.1.2. Functional Requirements

REQ-1: The mobile app should load the UI

REQ-2: The customer should register him/herself

REQ-3: Has the ability to reset password.

REQ-4: Has the ability to select a company or service centre

REQ-5: Has the ability to search a company by name

REQ-6: Has the ability to search a service centre by location or by product.

REQ-7: Has the ability to type the issue in words.

REQ-8: Has the ability to explain the issue and submit as a voice message.

REQ- 9: Should receive an SMS once the issue is submitted asking to stay until the issue is accepted by the company.

REQ- 10: Should receive an SMS once the company has either accepted or rejected the issue.

Only if the issue is accepted by the company,

REQ- 11: Has the ability to pick a date and time slot to place the appointment.

REQ- 12: Has the ability to pay an amount of advance to confirm the appointment.

REQ- 13: Should receive an SMS once the advance amount is paid, with the appointment details.

REQ- 14: Should receive an SMS once the date of appointment is near.

REQ- 15: Should receive an SMS once the job is done, asking the customer to come and collect the item.

4.2 Company Admin Operations

4.2.1. Description and Priority

The company admin uses the website to manage the appointments and should basically be able to perform the following functions.

4.2.2. Functional Requirements

REQ-16: Should be able to register the company

REQ-17: Add/update branch details

REQ-18: Confirm or reject the issue submitted by the customer

REQ-19: Review the confirmed appointment details

REQ-20: View the confirmed appointment details

4.3 App Admin Operations

4.3.1. Description and Priority

App admin is the intermediate admin who is responsible for validating the business registration number of the company during the company registration process.

4.3.2. Functional Requirements

REQ- 21: Has the ability to view the data submitted by the company admin and check the validity of the entered company registration number.

5. Other Non-functional Requirements

5.1. Performance Requirements

This mobile app and website are designed predominantly to facilitate the customer and the company with placing appointments in a convenient time and ease out the way of life of both parties. The customer uses the mobile app to place the appointment and the website is used by the company admin to confirm, view and manage the appointments. The app should be able to notify the customer via SMS using the SMS API and to provide the facility of advance payment using the Cash API , both which are provided by hSenid Mobile. This system enables the company to pick only the situations where they can definitely fix the item, which will reduce their hustle within the premises. Only the customers with solvable issues will be accepted. The customer should be able to pick a convenient date and time from the given time slots.

5.2. Safety Requirements

Data entered into the system is backed up regularly. All backups are encrypted and stored at multiple offsite locations to ensure that they are available in the unlikely event that a restore is necessary.

All production servers are running a long-term Support distribution of their operating system to ensure timely updates are available.

Apart from that, system maintenance schedule will frequently be evaluated to ensure that we keep user impact as low as possible.

5.3. Security Requirements

In certain cases when our system has legal purpose to do so, it may keep user personal data. The system does not automatically delete the content created by an individual user in the system, upon deletion of an individual user account.

Users may have installed third party applications or custom applications. System requires a username and a password of minimum of 8 characters. Repeated failed login attempts trigger a 30 second lock before a user can retry.

System will send a link to the email associated with the account that will enable the user to create a new password on account creation and password reset. Passwords are stored in a hashed form and will never be sent via email.

5.4. Software Quality Attributes

Using this mobile app, customers can easily search for the company or the service provider of his/her preference from anywhere, anytime. The reliability of all service providers and companies available on service in this app are ensured. Both the company and the customer have the comfort of choosing the convenient time for themselves, to hold a meetup.

What is unique in this system is that the company can look into the issue with the item prior to confirming the appointment and check if they are actually capable of fixing it. Since it would be a waste of time for both the customer and the service provider to physically meet just to know that the service provider is incapable of fixing the issue, this is surely going to be a huge advantage.

The customer has the ease of paying the amount of advance just by giving his/her contact number. The Cash API will handle the rest of the course. They do not have to have the fear about giving bank details and card details to an app.

Apart from above qualities it has additional features such as availability, accuracy, interoperability, reliability, reusability, robustness, testability, and usability.

5.5. Business Rules

Customer submits the issue and pays the advance if the company accepts the issue and he/she agrees to place the appointment. The company views the submitted issue and accepts it if they can fix the problem, otherwise they reject it. App admin checks the company registration number and verifies it.

5.6. Deployment prerequisite

5.6.1. On premise deployment prerequisites

- CPU 2x Physical Xeon ◊ RAM 16GB ECC RAM
- DISK 5x (or more) 72GB or (better) 15K SCSI
- RAID Caching RAID controller with 256MB RAM, 2 Channel. RAID 5 or 6 recommended.
- POWER Redundant PSU
- SOFTWARE Windows Server x64 2012R2, Microsoft SQL Server x64 2014 (Standard or Enterprise Edition)

5.6.2. Cloud deployment environment

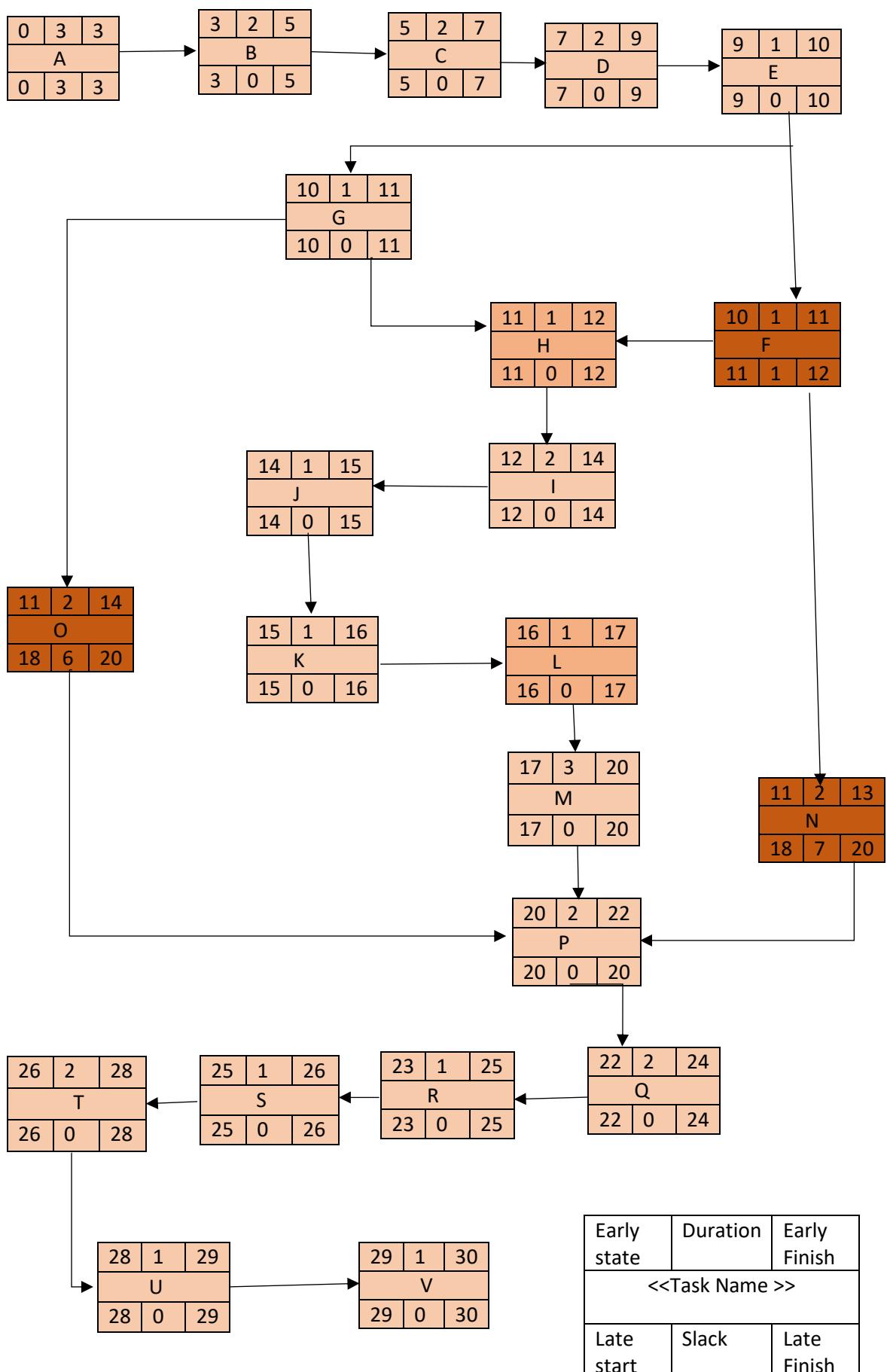
For this system deployment we use Amazon DynamoDB web hosting service, the reason being it is a fast and flexible NoSQL database with seamless scalability.

6. Appendix

6.1. Network diagram

Table 1- Network diagram overview

Task ID	Description	Dependency	Duration (Week)
A	Identify the requirements	--	3
B	Submit proposal and SRS	A	2
C	Allocate task	B	2
D	Design wireframe	C	2
E	Evaluate wireframe	D	1
F	Design web interfaces	E	1
G	Design mobile user interfaces	E	1
H	Identify the entities	F,G	1
I	Design the ERD	H	2
J	Finalized the databases	I	1
K	Create databases	J	1
L	Implement security functions	K	1
M	Implement business logic	L	3
N	Create web interfaces	F	2
O	Create mobile interfaces	G	2
P	Connect frontend with backend	M,N,O	2
Q	Test in the dev environment	P	2
R	Initialize deployment environment	Q	2
S	Deploy the beta release	R	1
T	Test beta release	S	2
U	Debug errors	T	1
V	Deploy final product	U	1



6.2. Data flow diagram

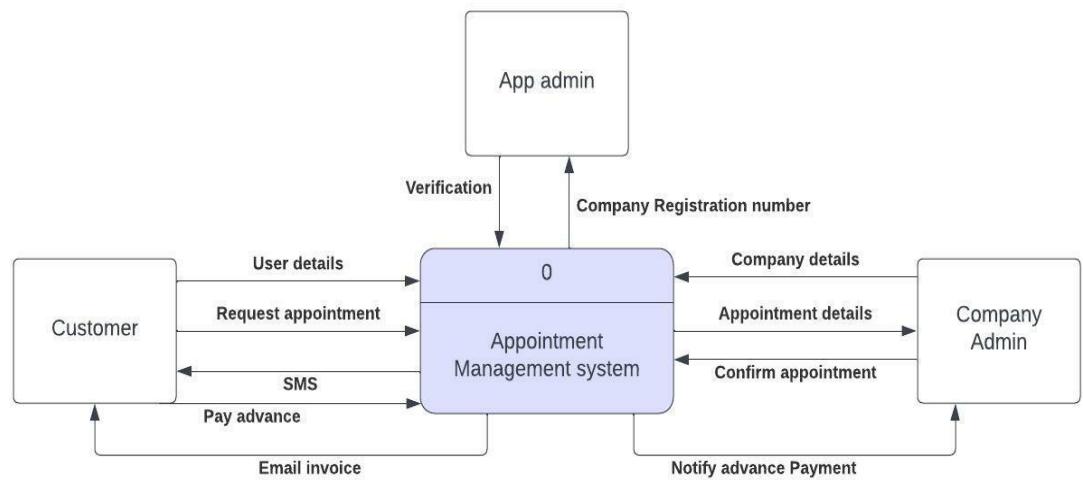


Figure 2-Context Diagram

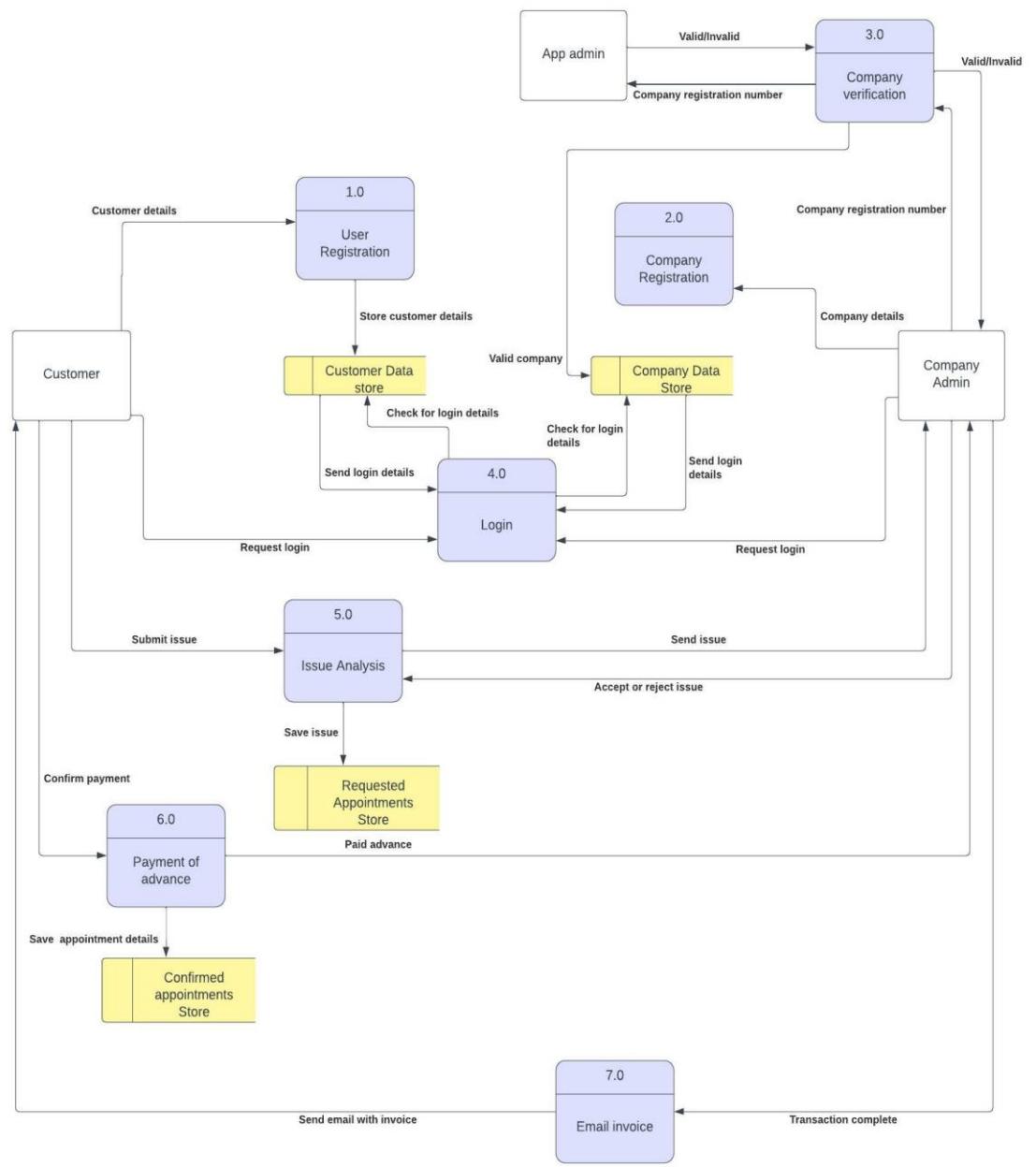


Figure 3-Level 0 Data Flow Diagram