

ANDROID BASED ELECTRONIC PRODUCT SERVICING SYSTEM

BY

**MD TASLUF MORSHED
ID: 191-15-12089**

AND

**MD ASSADUJJAMAN TILOK
ID: 191-15-12594**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

PROFESSOR DR. MD. FOKHRAY HOSSAIN
Dean, FSIT &
Director, International Affairs
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2022

APPROVAL

This Project titled "**Android Based Electronic Product Servicing System**", submitted by Md Tasluf Morshed and Md Assadujjaman Tilok to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 5th January 2023.

BOARD OF EXAMINERS

<hr/> (Name) Designation Department of CSE Faculty of Science & Information Technology Daffodil International University	Chairman
<hr/> (Name) Designation Department of CSE Faculty of Science & Information Technology Daffodil International University	Internal Examiner
<hr/> (Name) Designation Department of ----- University	External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of Professor **Dr. MD. FOKHRAY HOSSAIN**, Dean, FSIT & Director, International Affairs, Department of CSE Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Professor Dr. Md. Fokhray Hossain
Dean, FSIT &
Director, International Affairs
Daffodil International University

Submitted by:

Md Tasluf Morshed
ID: -191-15-12089
Department of CSE
Daffodil International University

Md Assadujjaman Tilok
ID: -191-15-12594
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to Professor **Dr. MD. FOKHRAY HOSSAIN**, Dean, FSIT & Director, International Affairs, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Mobile Application*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan**, Professor and Head, Department of CSE for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

The electronics industry in Bangladesh is one of the fastest-growing industries in the country with great potential. As of November 2020, the industry was estimated to be worth 26700 crore (US\$2.8 billion), with a yearly growth rate of 11%.

Currently most of the electronics and home appliance brands are opting for e-commerce platforms for the promotion and selling of their products. Some of them have launched their own E commerce platform and others are using the multi-vendor e-commerce platform for selling products. However no one have the concern for after sales service yet. Rural and suburban communities have incurred more horrific drawings because of this problem. There isn't always a support center nearby, and home service isn't always possible in these areas. As a result, customers must go to a suburb or area where customer care is accessible in order to obtain the services. Poor logistics and transportation facilities are another major challenge which makes it difficult to provide after sales service in this sector. Therefore, this project intended to build a service system, considered as "**Android Based Electronic Product Servicing System**" which could be the ultimate solution for the purpose. This project expect to reduce a lot of work load that people don't need to go out to find a servicing solution. User can simply register their problem to the system with a valid user profile then administrator will send a technician to solve the problem in a suitable time.

TABLE OF CONTENTS

CONTENTS	Page No.
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTORDUCTION	1-5
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Expected Outcomes	3
1.5 Project Management and Finance	4
1.6 Report Layout	5
CHAPTER 2: BACKGROUND	6-8
2.1 Introduction Preliminaries/Terminologies	6
2.2 Related Works	6
2.3 Comparative Analysis	6
2.4 Scope of the Problem	7
2.5 Challenges	8
CHAPTER 3: REQUIREMENT SPECIFICATION	9-40
3.1 Business Process Modeling	9
3.2 Requirement Collection and Analysis	10
3.3 Use Case Modeling and Description	14
3.4 Logical Data Model	21
3.5 Design Requirement	39

CHAPTER 4: DESIGN SPECIFICATION	40-62
4.1 Front-end Design	40
4.2 Back-end Design	59
4.3 Interaction Design and User Experience (UX)	62
4.4 Implementation Requirements	62
CHAPTER 5: IMPLEMENTATION AND TESTING	63-66
5.1 Implementation of Database	63
5.2 Implementation of Front-end Design	63
5.3 Testing Implementation	63
5.4 Test Results and Reports	64
Chapter 6: Impact on Society, Environment and Sustainability	67-69
6.1 Impact on Society	67
6.2 Impact on Environment	67
6.3 Ethical Aspects	68
6.4 Sustainability Plan	69
CHAPTER 7: CONCLUSION AND FUTURE SCOPE	70-70
7.1 Discussion and Conclusion	70
7.2 Scope for Further Developments	70
REFERENCES	71

List of Figures

Figures	Page No.
Figure 01 Business process model	09
Figure 02 Use Case Diagram	14
Figure 03 Admin Activity Diagram	21
Figure 04 Customer Activity Diagram	22
Figure 05 Admin Registration Sequence Diagram	23
Figure 06 Admin Login Sequence Diagram	24
Figure 07 Admin Login with Google Sequence Diagram	25
Figure 08 Admin Profile Sequence Diagram	25
Figure 09 Admin Forget Password Sequence Diagram	26
Figure 10 Admin View Order Sequence Diagram	27
Figure 11 Admin Add Agent Sequence Diagram	28
Figure 12 Admin Add Technician Sequence Diagram	28
Figure 13 Admin Update Order Sequence Diagram	29
Figure 14 Admin Update Agent Sequence Diagram	30
Figure 15 Admin Update Technician Sequence Diagram	31
Figure 16 Customer Registration Sequence Diagram	32
Figure 17 Customer Login Sequence Diagram	32
Figure 18 Customer Login with Google Sequence Diagram	33
Figure 19 Customer Profile Sequence Diagram	33
Figure 20 Customer Forget Password Sequence Diagram	34
Figure 21 Customer Add Order Sequence Diagram	35
Figure 22 Customer Add Address Sequence Diagram	35

Figure 23 Customer Update Address Sequence Diagram	36
Figure 24 Chart List Sequence Diagram	36
Figure 25 Track Order Sequence Diagram	37
Figure 26 Order History Sequence Diagram	37
Figure 27 Payment Sequence Diagram	38
Figure 28 Apps Welcome Page	40
Figure 29 Apps Login Page	41
Figure 30 Apps Sign Up Page	42
Figure 31 Apps Home Page	43
Figure 32 Apps Order Page	44
Figure 33 Apps My Chart Page	45
Figure 34 Apps Payment Page	46
Figure 35 Apps Profile Page	47
Figure 36 Apps History Page	48
Figure 37 Apps Edit Profile Page	49
Figure 38 Apps Address Page	50
Figure 39 Apps Notification Page	52
Figure 40 Website Login Page	53
Figure 41 Website Sign UP Page	53
Figure 42 Website Forget Password Page	54
Figure 43 Website Dashboard Page	54
Figure 44 Website Agent List Page	55
Figure 45 Website Add Agent Page	55
Figure 46 Website Update Agent Page	56
Figure 47 Website Technician List Page	56
Figure 48 Website Add Technician Page	57

List of Table

Table	Page No.
Table 01 Use Case Diagram for Registration	15
Table 02 Use Case Diagram for Login	16
Table 03 Use Case Diagram for Service	17
Table 04 Use Case Diagram for Cancel Request	18
Table 05 Use Case Diagram for Confirm Order	18
Table 06 Use Case Diagram for Hire Technician	19
Table 07 Use Case Diagram for Payment	19
Table 08 Use Case Diagram for add Agent	20
Table 09 Use Case Diagram for add Technician	20
Table 10: Automated Testing Result	64

CHAPTER 1

Introduction

1.0 Introduction

The Project Considered as “**Android Based Electronic Product Servicing System**” is a E commerce app or virtual store on internet confide on aggregation model, where user can hire technician in need of servicing there home appliance. E commerce, or electronic commerce, refers to transactions conducted via the internet. Every time individuals and companies are buying or selling products and services online, they’re engaging in e commerce and the term “Home appliance” refers to electronic products, devices, or equipment used in various household purposes, such as TVs, refrigerators, ACs, or washing machines. In this application business transactions occur either as business-to-business (B2B), business-to-consumer (B2C). In this android application are following business-to-business and business-to-consumer together, Where distributor assign specific task to a service provider and customer get the required service from distributor. This application will make business model easier and time saving. User can simply register themselves by using a valid email or phone number to the system in order to take the services. The system is a package, used by service provider to improve the efficiency to their B2C business.

The most widely used operating system in the smartphone is Android and ios. Therefore, as a developer of the project, we are working on an android app and web application for this service. To make an android app they want to use React native. It's a JavaScript framework that helps us to build an android and ios app. It's built on top of the React framework. For the web application, this will use React framework. For the back-end, it will use Nodejs and for the database, developer want to use MongoDB.

1.2 Motivation

In The 21th century while the world is vastly depending on electronic goods and technology in that very time people from Bangladesh are facing difficulties to find a optimal servicing solution for their household necessities. Sadly a number of factors can be accountable for the issue. Undoubtedly after sell service policy is the root of it. Many people in Bangladesh live in remote areas, Therefore,

- Whenever any electronic product run out of warranty user had to face difficulties. People have to go out door to door to find a servicing solution.
- Incompetent technicians can't solve problems like professionals which caused future issues in that product.
- Too much time consuming.
- Customers can't get any security from them how long it works.
- Service providers charge as much money they want for a simple solution.
- People don't have an idea what's going on that's why they are bound to provide the service charge which is unfair.

It is visible that, along with the rapid development of technology, the servicing system is not improving very speedily. It is considered to be a massive problem not only in Bangladesh but also South Asia. Even today, whenever a household product collapse, user have to take this to a service point and wait for a long time to fix it.

Now it is necessary to structure the service system which is based on time efficiency and skilled technicians. This advanced system may upgrade the UN ethical servicing trends by developing a user-friendly application for stakeholders. Therefore, the purpose of this project is to develop a "**Android-based Electronic Product Servicing System**" in order to reinforce the user's satisfaction.

1.3 Objective

The main objective of “**Android-based Electronic Product Servicing System**” is to make a faster and smarter way to get service of any electric products. This mobile application will serve as a B2B and B2C e-commerce platform. It is currently developed for android, windows, and ios featured devices as well web service for all platform.

- Developing Business Relations
- Cost-effective Management
- Enhances the Efficiency of Services
- Optimal Servicing Solution
- Sustaining Existing Customers
- Responsive E-Commerce App

1.4 Expected Outcomes

- People will not face any hassle when there electrical product are damaged or needs to repair.
- They don't need to find any service point or take the product to that service center to fix it.
- Expert can solved the problem more professionally as compared to local technicians and customer can get up to 1 year service warranty from the app.
- Sometime service center charge as much money for a simple solution and people don't have the idea what's going on. However in this apps service charge are fixed and updated in the website. That's why they are bound in any unfair charges.
- For the second hand product our team will ensure its quality and we will give additional 6 month service warranty for that product.

1.5 Project Management and Finance

This project aspired to develop a user-friendly servicing system on product servicing for the consumer. The system will contain an android application as well a web application in order to reach to customer satisfaction

For mobile application: We have use React native. It's a JavaScript framework. We know it's very difficult to develop to apps in a short time period, one for android device and other-one for ios. It's a verry time consuming and costly process. But in React Native we can make app that will run both on android and ios device. And we easily deploy in into the Play Store and App Store.

For the web application: We have use React framework. Again its a JavaScript framework. It is now one of the most trending framework to build a website. As our main app will build on JavaScript that's why it will be easy for us to learn one language and implement it on different area.

For the back-end: We have use Nodejs. It's a JavaScript run time environment to run JavaScript into any machine like Computer or mobile. It will help us to make good back-end for our application as well as for the web app.

For the database: We want to use MongoDB. Although we have not fixed it yet. But after analysis our customer data and all the date we will use in our app, then we will decide whether MongoDB is good for our app or we need to shift into MySQL.

As Stakeholders:

- Visitor: Visitor can view the available services on the site.
- Customer: Customer can choose any services and make payment from the site.
- Admin: An Admin have some additional privilege and access including all the privilege that visitor and customer had.

1.6 Report Layout

Chapter 1: Introduction In this chapter I will discuss about project Introduction, Motivation, Objective and also discuss as regards the Expected Outcome.

Chapter 2: Background I discuss about foundation of our project and literature review with competitor details. I also present in regard to the current features of restaurant app and what we make.

Chapter 3: Requirement Specification For this project I should have some requirements in our system and device. Also need the application buildup. All things related with those discuss in this section.

Chapter 4: Design Specification In this chapter I essentially converse the project back-end Design & front-end design, UX & interaction design and also talk about the requisite of implementation.

Chapter 5: Implementation and Testing In this section I will discuss about our hardware device and its design implementation. All the required device module will be discussed.

Chapter 6: Impact on Society, Environment and Sustainability In this section I will discuss about Impact on Society, Environment and Sustainability

CHAPTER 2

Background

2.1 Introduction Preliminaries/Terminologies

We get several E commerce apps in our daily life. But in this application user will get B2B and B2C service in one place with aggregation model. Herethis app work with Internet connection from a locally hosted database as MySql database. We develop this application by conducting React and React native, popularly known as javascript framework. We used Nodejs, Material UI, Jetpack, View Binding, Google cloud platform, Google Map, online payment gateway, as much the coding language and also use MySQL database.

2.2 Related works

As a B2B and B2C e-commerce app, there are many commercial competitors in our country. Likes Casio Electronics Service Center, Sheba XYZ, A.M. & Associates, RM Electronic Co. Some apps are doing B2B and some of them are doing B2C. We focused on B2B and B2C both with aggregation model. Therefor this app will be better and unique from others. With the help of this application we will collaborate with local business so that the local won't get disrupt. This app will authorize many vendor and service provider there workshop to create a strong brand repudiation. All request will be identified, verified and forwarded by system administrator, this will help to terminate irrelevant request. Then after service delivery process will be vend by admin. So that payment system will be secure.

2.3 Comparative Analysis

In our country, many website base servicing company are providing the same service. Only two or three app are use both android an web based service. Most of these android apps are confusing and default to use. For Example “Sheba ZYZ” they are providing the same services as we offers with a fixed rate chart. But whenever a product shows trouble how could the customer know what exactly the problem is? And which service he should take?

On the flip side, our application offers a form to give minimum details of a problem and analyzing that problem by professionals we suggest the customer best solution for his product. Sometime a platform attracts customer by offering unreasonable rate of discount. They manage to do so by degrading the service quality. Some apps like evaly, alisha mart, e-orange tried to unethically shatter local market. So, they fixed the price above the market rate. Product delivery system of these platform are so fragile. With the help of our application, which will collaborate local service provider to consumer in direct connection. so that customer can track his request to get direct update and the local market of business won't get disrupt. All the orders and request will be verified by admin, this will help to terminate irrelevant advertisement.

2.4 Scope of the problem

This is a prototype of “Android-based Electronic Product Servicing System” application. Customer can not buy electronic product from this site. To establish it as a direct B2C E commerce site we will add buying option in next update. The app is currently hosted locally. If users are more than 0.1 million then this app can be crushed. For that we will buy a paid server in future. We have tried to accomplish the necessity for user. Hence they use it very easily. Some access should be required.

- Internet connection needed.
- Must be registered in the app.
- User must need to install app.
- Permission to access the website.
- Must need login permission.

2.5 Challenges

We have to face several stages of challenges during this project. At first place, we know that Android device accounting for 70% market share while iOS contain around 28% as-well. To capture the market we have to target bot operating system therefor we have to make two different apps or come to a solution with an apps that can run on both operating system. After analyzing this issue we come with a solution that, we should develop the apps on ‘React Native’ which is popularly known as a java script framework that can run in both of thus operating system. We intended to provide the web service therefor we have to deal with ‘React’ which is one of the most trending software to build websites.

To design the application UI, we also learn XML We faced several bug and error during this project. This was the hardest part in this project that was solved later. To sing-in, customer had to add email or a number to verify his authenticity. We have to learn a completely new log in method as well Online Payment Gateways for payment method which is unknown to us. Then to reach a customer app there should be a map so that we learn about Google cloud platform, Google Map. And for database we learn MySQL. We faced several errors during connecting database and google map to connect to the application.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

Business Process Model for our system are given below

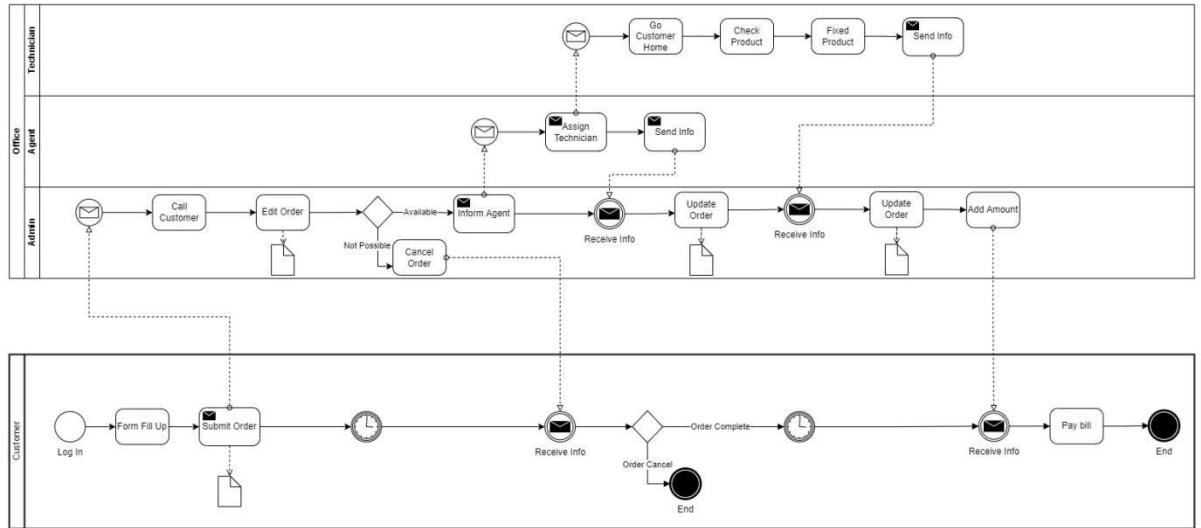


Figure 01: Business process model

Here a customer will login into the system then he/she will fill up a form for a repair request. Then this request will be reviewed by an admin. He will call the customer to get more details about the order. Then he will assign an agent for this task. An agent will select a technician for the task. The technician will go to the customer's home, review the product and try to fix that product. After completing the task admin will add cost to the customer account. And finally, the customer will pay their bills and close the process.

3.2 Requirement Collection and Analysis

3.2.1 Functional Requirements

3.2.1.1 Admin requirements

- Admin Login

This system to able to give permission to access the website to the admin and he can access the whole system.

- Database management

Admin controls the database and keeps track of all records of product and client details.

- Manage service

Admin can add, Update and delete any service category. He can also view, add, update and delete any service brand name and model number.

- View customer details

Admin views the personal details of the customer.

- Manage Agent

Admin can view all the agents who have contracts with the system. He can add, and update any agent's information.

- Manage Technician

Admin can view all the technicians who work for the system. He can add, and update any technician's information.

- Manage Order

Admin can view and update any order. When a repair request is submitted into the system by a customer the admin can view that order. He can accept that order, assign a technician for that order and finally confirm the order by updating it.

- Logout

Admin can log out from the system.

3.2.1.2 User requirements

User Login:

Description of the feature

Using this feature users can log in to the system. A user should log in with his/her username and password to the system after registration. An invalid username or password is not allowed to enter the system. A user can also login into the system with a google account.

Functional requirement

A username and password will be provided after user registration is confirmed.

Password must be hidden from others while typing it in the field.

A google login button should be implemented to log in by Google

New user registration:

Description of the feature

A new user will have to register in the system by providing essential details in order to purchase products in the system.

Functional Requirement

The system should be able to verify and validate the information.

Password should be encrypted to provide security.

View and update own details:

Description of the feature

Customers can view/update their personal information. Customers can also set a default address.

Choose a service:

Description of the feature

Customers can view all the services that are provided by the system at that time.

Customers can also see which brands are available and their model number for a product to repair.

Repair request:

Description of the feature

Customers can select any service, then fill up a form with the details of the product and select the cross-pounding brand and model number for the product to send a repair request. He can also remove any service request from the cart by clicking remove. He can track his request.

History:

Description of the feature

The customer can see all his previous service requests and their details.

Notification:

Description of the feature

The customer will get a notification on every step of the order.

Logout:

Description of the feature

The customer can log out of the system.

3.2.2 Non-Functional Requirements

3.2.2.1 Performance requirements

The system must accommodate a high number of items without any fault and view information could not take longer than 3 seconds to appear on the screen.

3.2.2.2 Usability requirements

The android app is designed for a user-friendly environment and is easy to use.

3.2.2.3 Security

Functions of the app must be accessed in the way they were intended to be accessed.

Included files shall not be accessed outside of their parent file.

Administrators can only perform the administrative task on pages they are privileged to access. The customer will not be allowed to access the administrator pages.

API should be accessed only by authorized users.

3.2.2.4 Error-handling

The app must handle its internal error and it should not terminate for an error. It should show the causes of the error to the users.

3.2.2.5 Efficiency and Maintainability

Page loads should be returned and formatted in a timely fashion depending on the request being made.

Administrators will have the ability to edit the aspect of order forms, service descriptions, prices, and websites directly.

The customer should send a repair request, track his order and get the notification of the order in an efficient manner.

3.2.2.6 Reliability Requirement

The system should provide a reliable environment to customers and owners. All orders must be reached at admin without any error.

3.3 Use Case Diagram

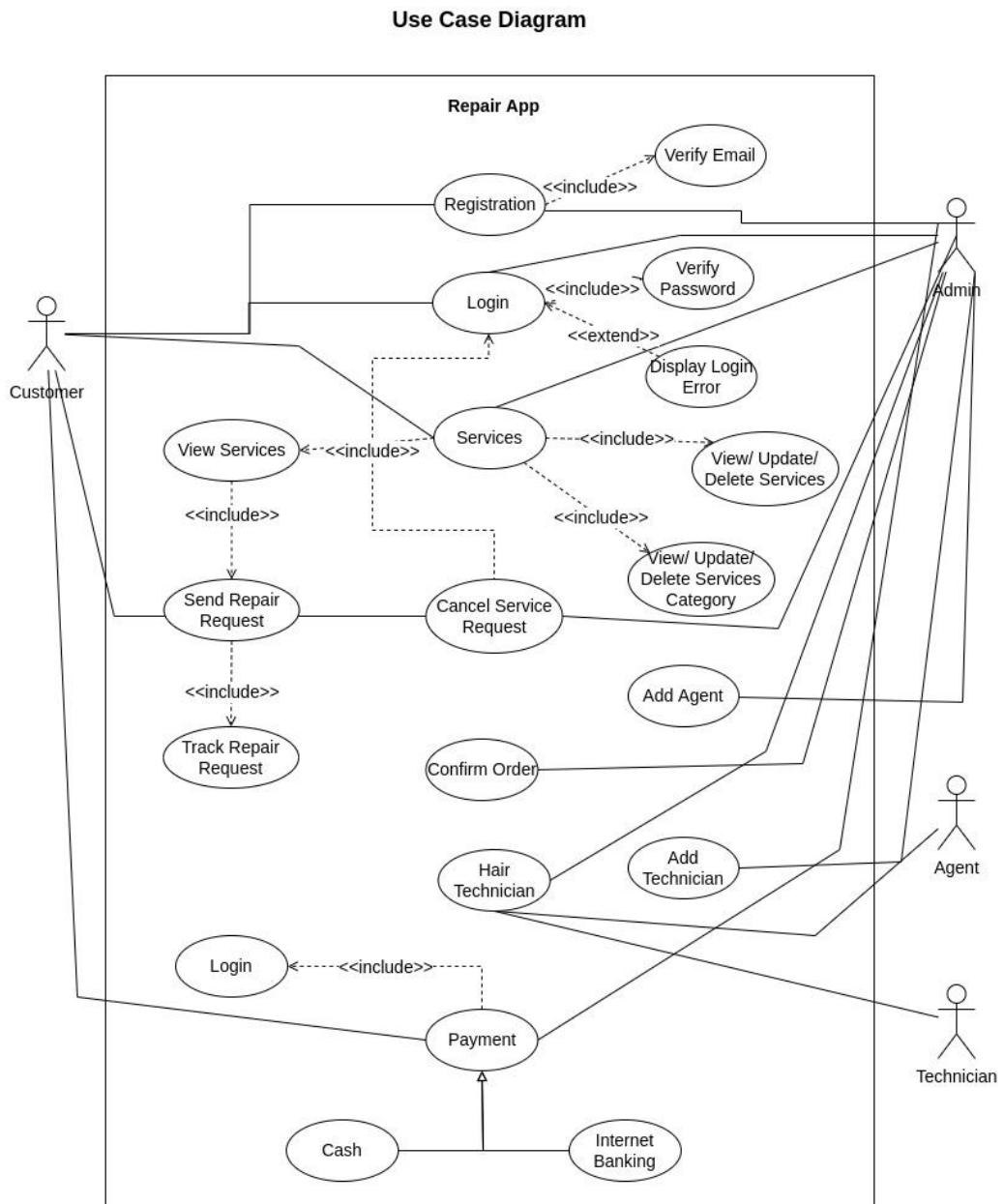


Figure 02: Use Case Diagram

3.3.1 Use Case Description

Use Case ID	UC1
Name	Registration.
Description	This use case allows users to register into the system to access the relevant functions according to the user's role. The various user roles are customer and admin. To register to the system, all users have to enter their name, email, and password. Then a verification email will be sent to the user's email to verify their email address. They have to click the link provided in the email. After successfully verifying their registered users can now login into the system.
Actor	Admin Customer
Pre-Condition	All the information Fields must be filled up.
Post-Condition	Users can login into the system.
Flow	Fill up the registration form with the necessary information. Press the Sign Up button. The system will verify the given information The system will send a verification email to the user's email address The user will click the link that is provided in the email The system will verify the user account Users can now login into the system
Include	Verified email address.

Table 01: Use Case Diagram (Registration)

Use Case ID	UC2
Name	Login.
Description	This use case allows users to login into the system to access the relevant functions according to the user's role. The various user roles are customer and admin. Users can log in to the system in two ways. By Email and password and by Google login. To log in by email and password, users have to verify their email address first. By providing the correct email and password or by the google login function user can successfully login into the system. They will receive a JWT token for the authorization and redirect to the home page.
Actor	Admin Customer
Pre-Condition	All the information Fields must be filled up.
Post Condition	Get access to the system.
Flow	Go to the Login page Enter your email address and password. Google login Press the Login button. The system will verify the account. The system will send a JWT token for authorization and redirect to the home page Otherwise, display a Login error.
Include	Verify password.

Table 02: Use Case Diagram (Login)

Use Case ID	UC3
Name	Service.
Description	In this use case, the admin can add, delete and update any service. They can also add, delete and update any product brand and product model. Customers will browse all the available services. They can send select any services and send a repair request to the server. To send a repair request they have to fill up a form and then select the product brand and a product model. After submitting the repair request admin will take further action.
Actor	Admin Customer
Post Condition	Hire technicians based on their product requirements.
Flow	Go to the service section. Select product brand and product model Fill up the request form Send service request Track the repair request Admin will receive this repair request Admin can add, delete or update any service Admin can add, delete or update product brand and product model
Include	View Service View, Update or Delete Services View, Update or Delete Services brand and service model

Table 03: Use Case Diagram (Service)

Use Case ID	UC4
Name	Cancel Request.
Description	The customer is not interested anymore in the service or the required service isn't available on the system.
Actor	Admin Customer
Post Condition	Cancel Service Requests in order to reinforce the user's requirement.
Flow	View the order list. Select the specific booking. Give a reason and Press the Cancel Request button.
Include	Login.

Table 04: Use Case Diagram (Cancel Request)

Use Case ID	UC5
Name	Confirm Order.
Description	Admin checks all the necessities and confirms the Order.
Actor	Admin
Post Condition	The administrator agent allows the Customer for the asking service.
Flow	View request details from the system database. Verify request. Confirm request.
Include	None

Table 05: Use Case Diagram (Confirm Order)

Use Case ID	UC6
Name	Hair Technician.
Description	Admin will select an agent that is close to the order request address. Then admin will assign a technician for the job. The system will notify this state by a notification to the user app.
Actor	Admin Agent Technician
Post Condition	Technician will arrive for the service.
Flow	Check confirmation. Select an agent Select a technician close to the location Notify the current stage by a notification to user
Include	None

Table 06: Use Case Diagram (Hair Technician)

Use Case ID	UC7
Name	Payment.
Description	After successfully repairing the product, the customer will select the payment option. They can make payments either cash on delivery or by internet banking. After a successful payment, this process will end.
Actor	Admin Customer
Pre-Condition	Customer Get the expected servicing solution.
Flow	View the service tracker to get the amount total. Choose a payment method. For instant payment select cash. For digital payment select internet banking. Add this service to the History section.
Include	Login

Table 07: Use Case Diagram (Payment)

Use Case ID	UC8
Name	Add Agent.
Description	This use case allows the admin to add, view, and update an agent. To add an agent, they have to fill up a form with the necessary information and then click the add button to save the agent.
Actor	Admin
Pre-Condition	Fill up the form with the necessary information
Flow	Click the add button Fill up the form Submit the information.
Include	None

Table 08: Use Case Diagram (Add Agent)

Use Case ID	UC9
Name	Add Technician.
Description	This use case allows the admin to add, view, and update a technician. To add a technician, they have to fill up a form with the necessary information and then click the add button to save the technician.
Actor	Admin
Pre-Condition	Fill up the form with the necessary information
Flow	Click the add button Fill up the form Submit the information.
Include	None

Table 10: Use Case Diagram (Add Technician)

3.4 Activity Diagram

An activity diagram is used to understand the flow of work that an object or component performs. It can also be used to visualize the interaction between different use cases.

3.4.1 System Admin Activity Diagram

Admin manage system content by creating, updating, or deleting content from the system database as well manage customers, orders, bookings, and payments in the system.

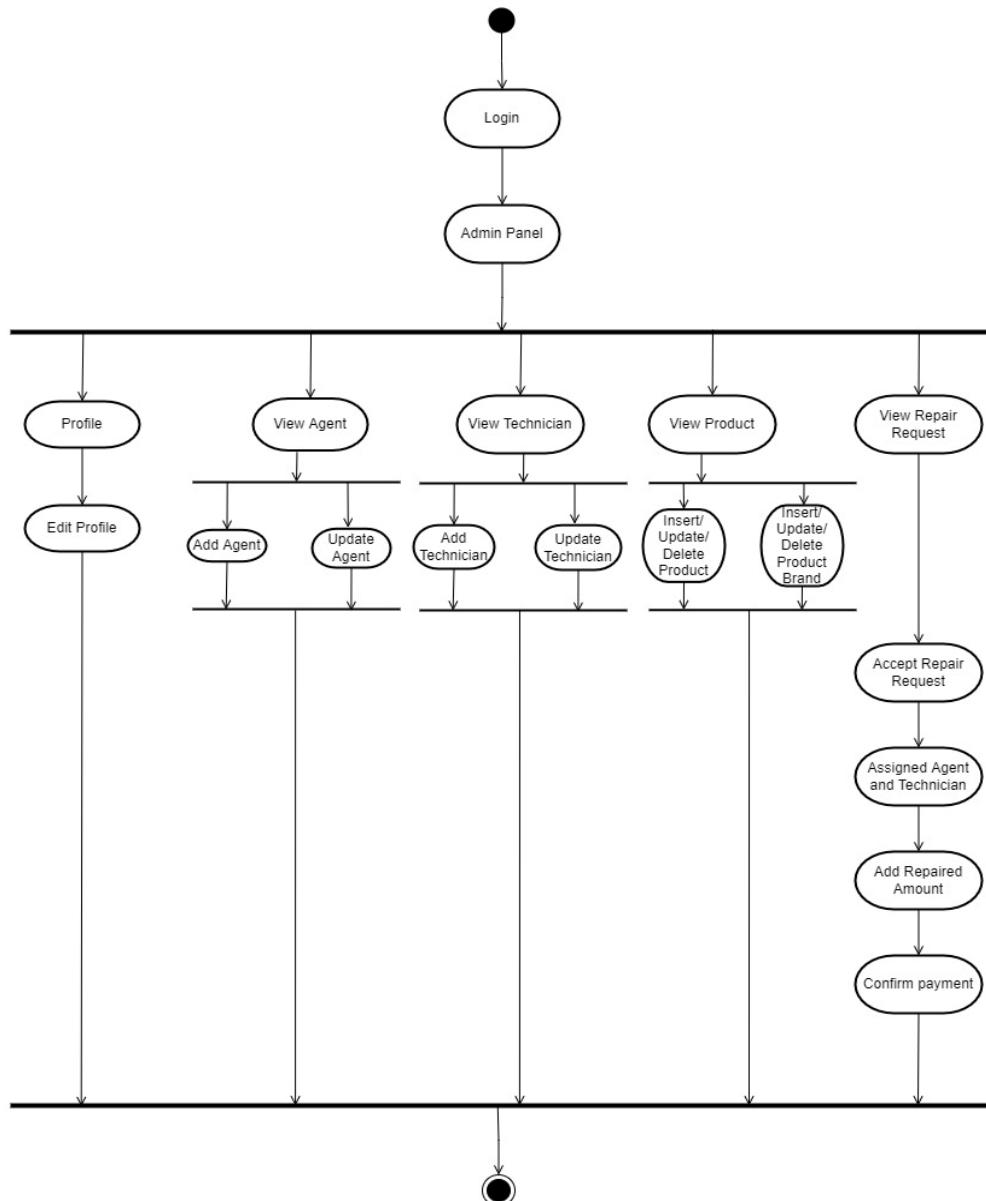


Figure 03: Admin Activity Diagram

3.4.2 Customer Activity Diagram

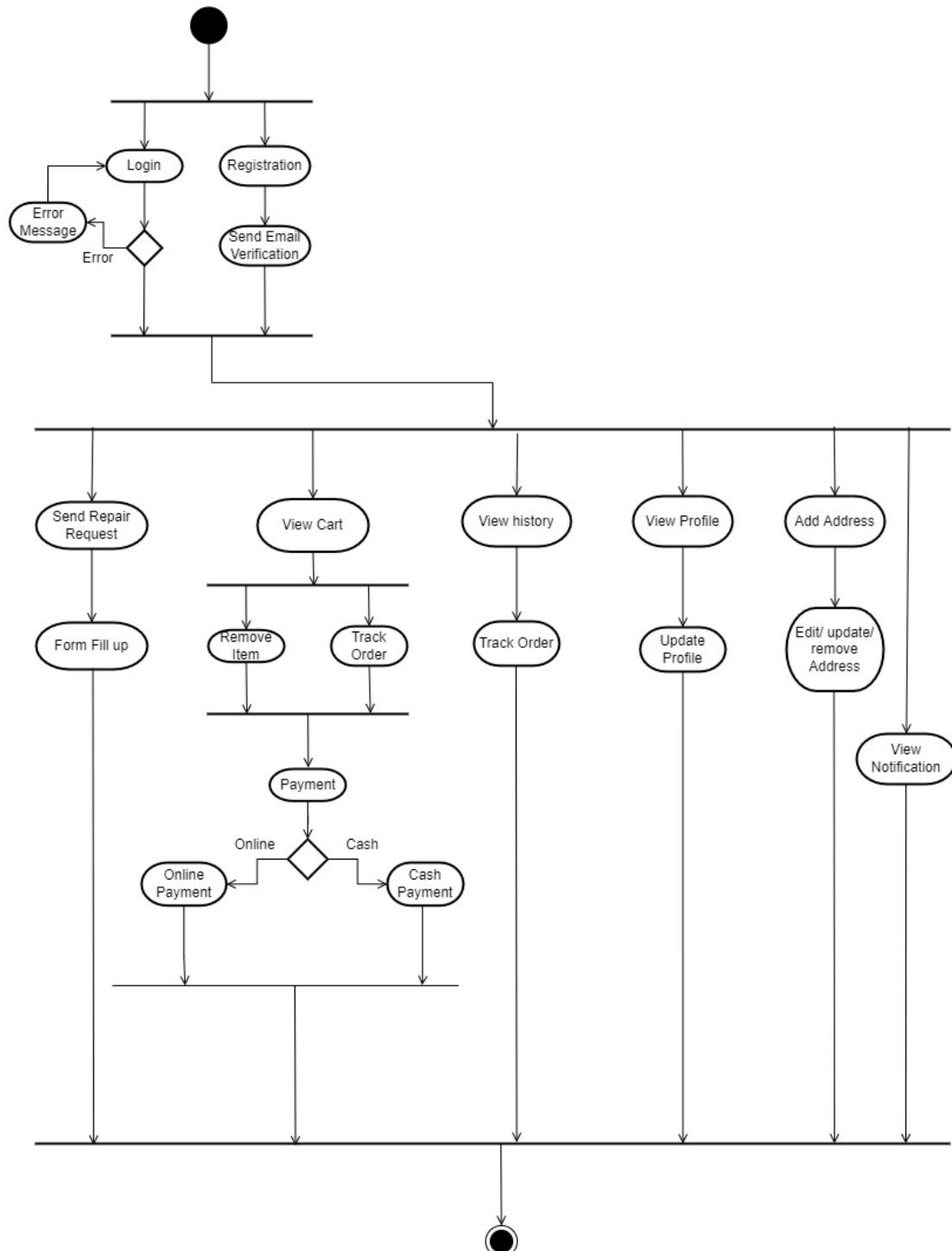


Figure 04: Customer Activity Diagram

3.5 Sequence Diagram

A Sequence diagram shows the sequence of messages exchanged by the set of objects performing a certain task.

3.5.1 Admin Registration

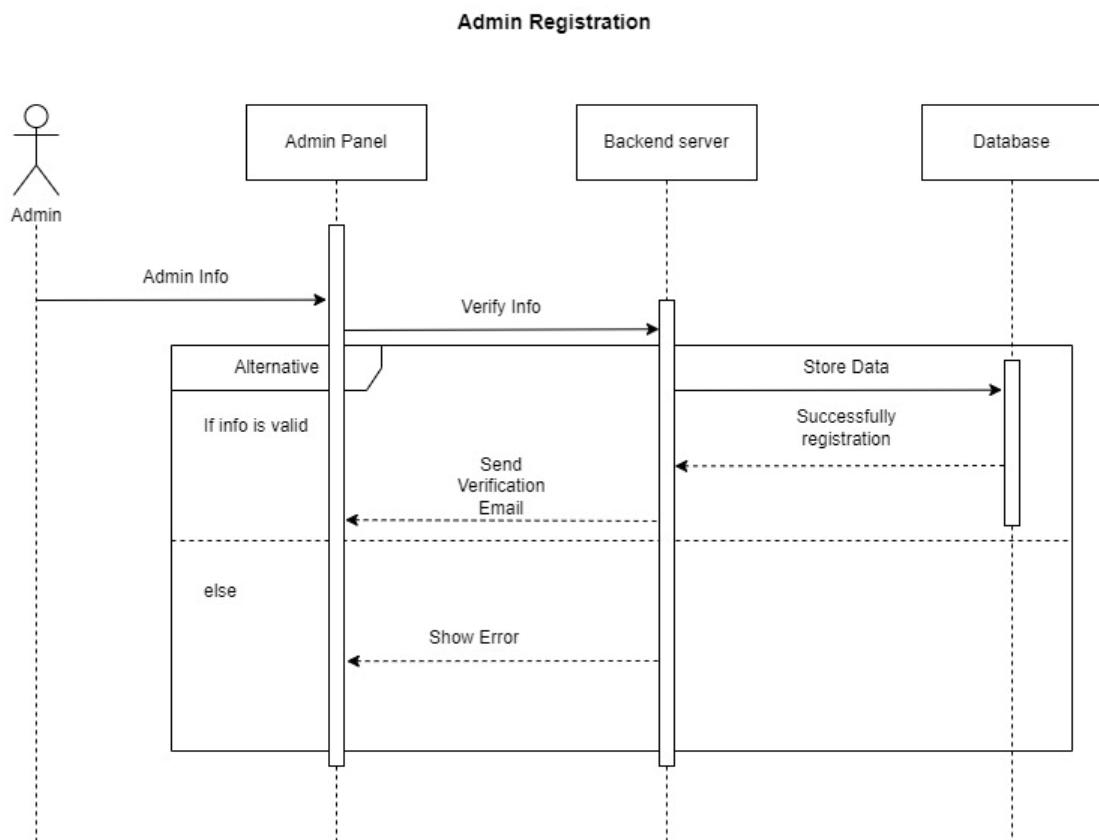


Figure 05: Admin Registration Sequence Diagram

3.5.2 Admin Login

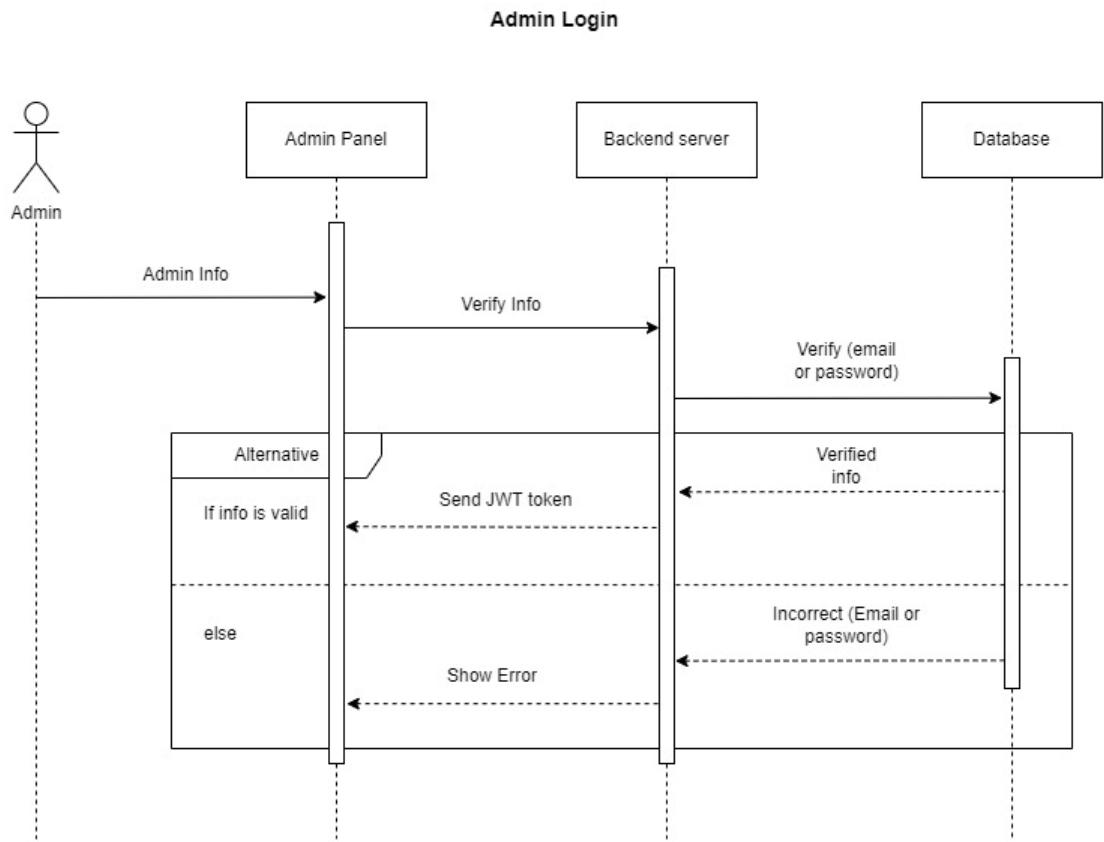


Figure 06: Admin Login Sequence Diagram

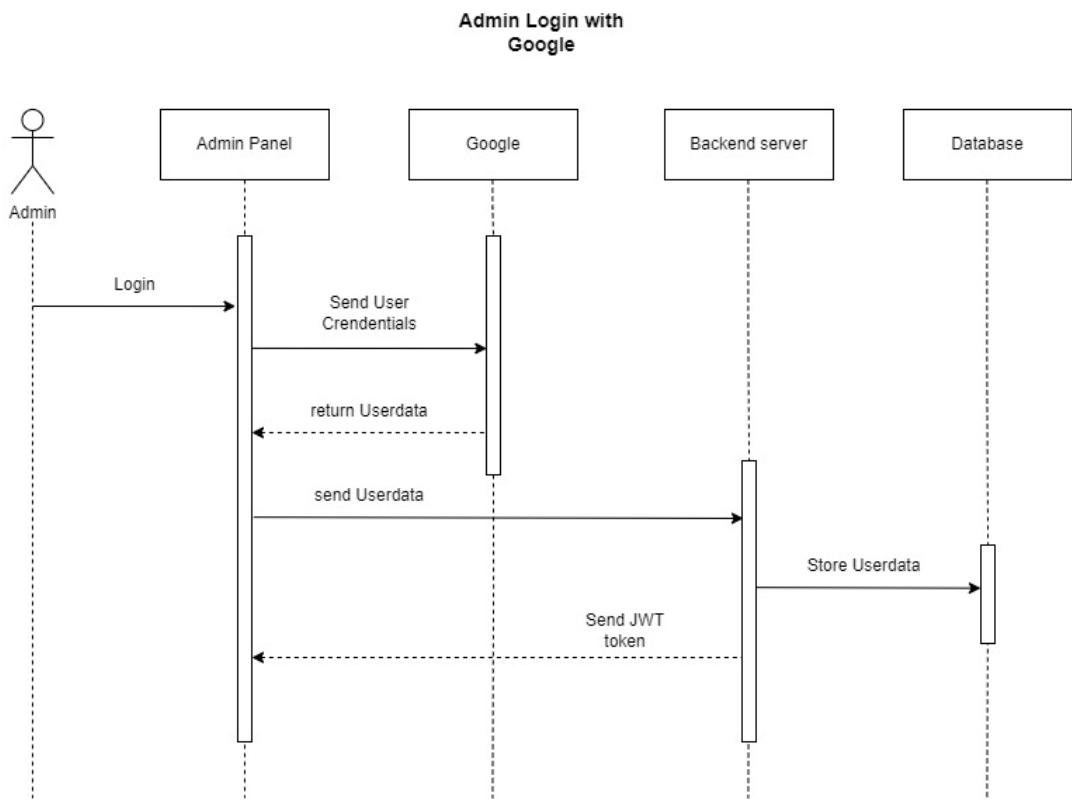


Figure 07: Admin Login with Google Sequence Diagram

3.5.3 Admin Profile

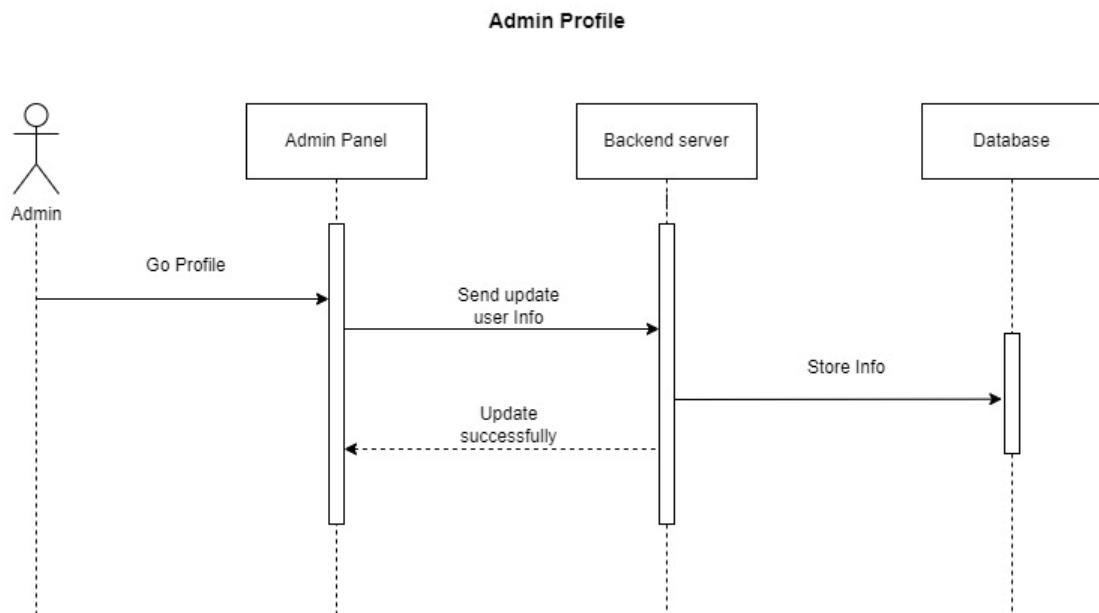


Figure 08: Admin Profile Sequence Diagram

3.5.4 Admin Forget Password

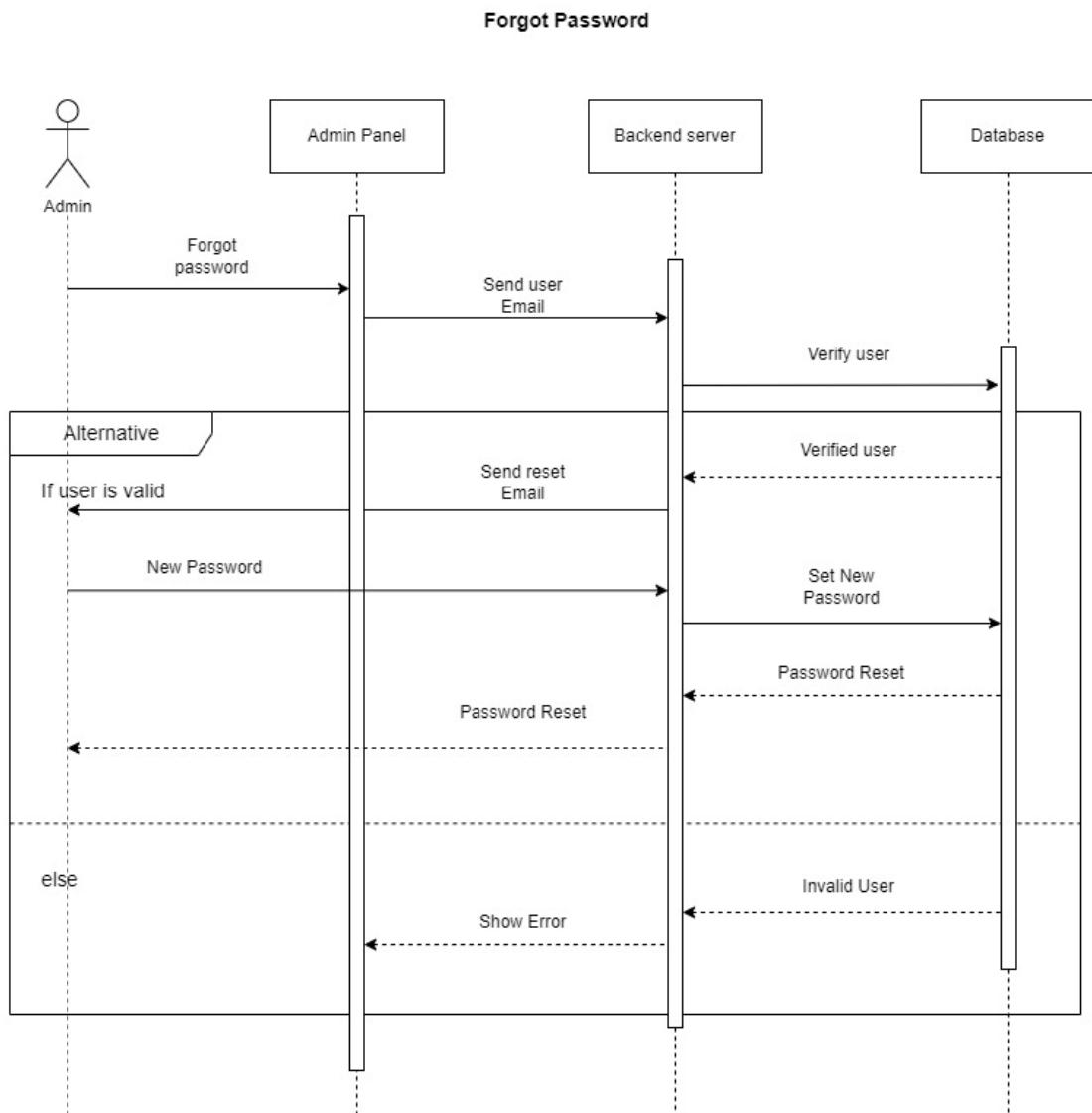


Figure 09: Admin Forget Password Sequence Diagram

3.5.5 Admin View Order

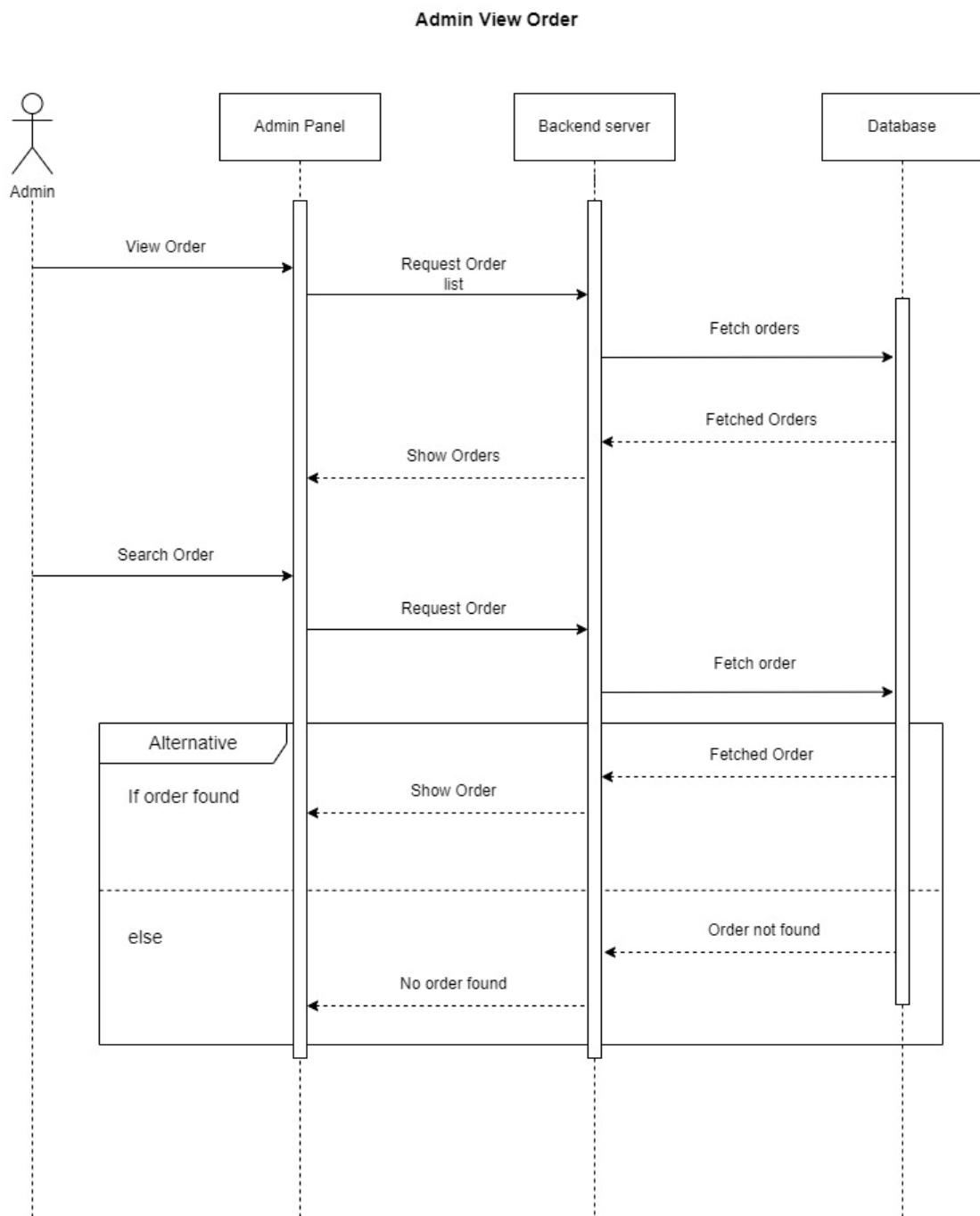


Figure 10: Admin View Order Sequence Diagram

3.5.6 Admin Add Agent

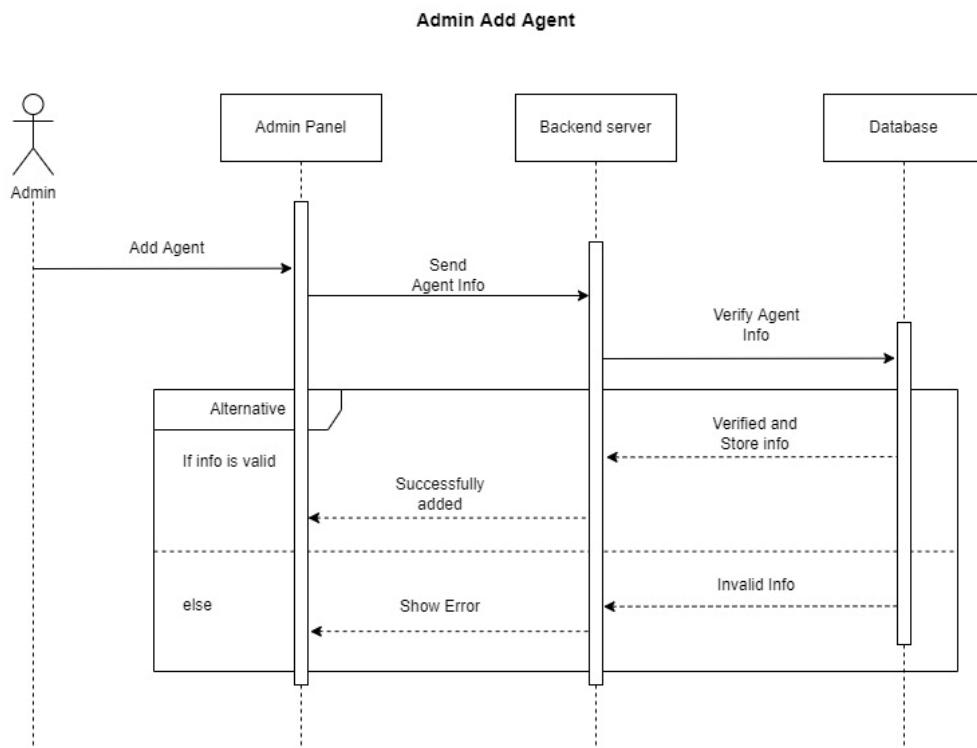


Figure 11: Admin Add Agent Sequence Diagram

3.5.7 Admin Add Technician

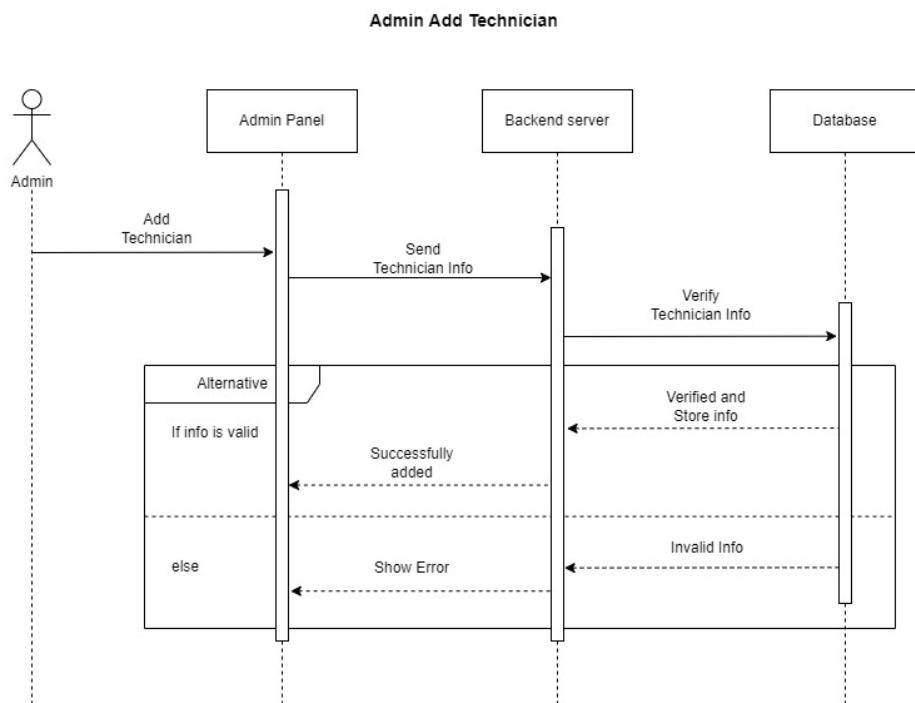


Figure 12: Admin Add Technician Sequence Diagram

3.5.8 Admin Update Order

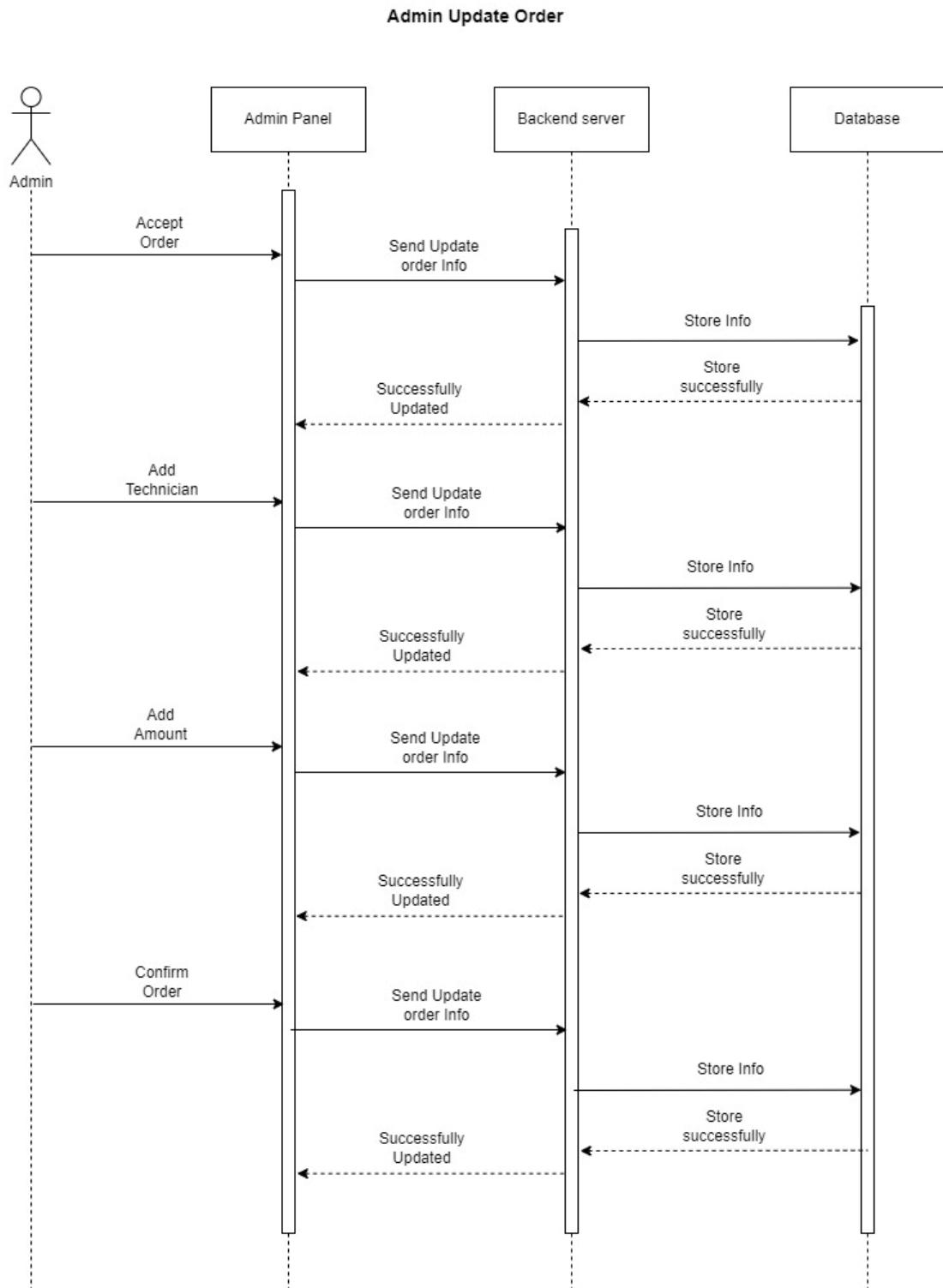


Figure 13: Admin Update Order Sequence Diagram

3.5.9 Admin Update Agent

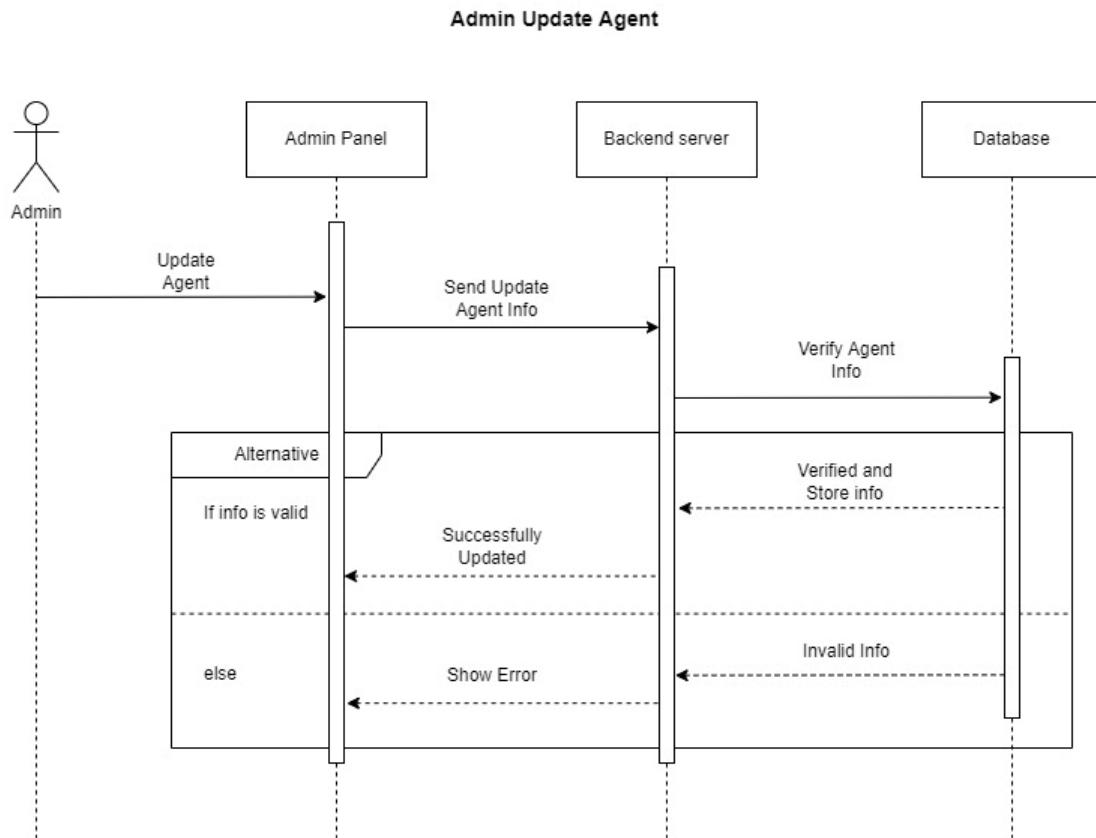


Figure 14: Admin Update Agent Sequence Diagram

3.5.10 Admin Update Technician

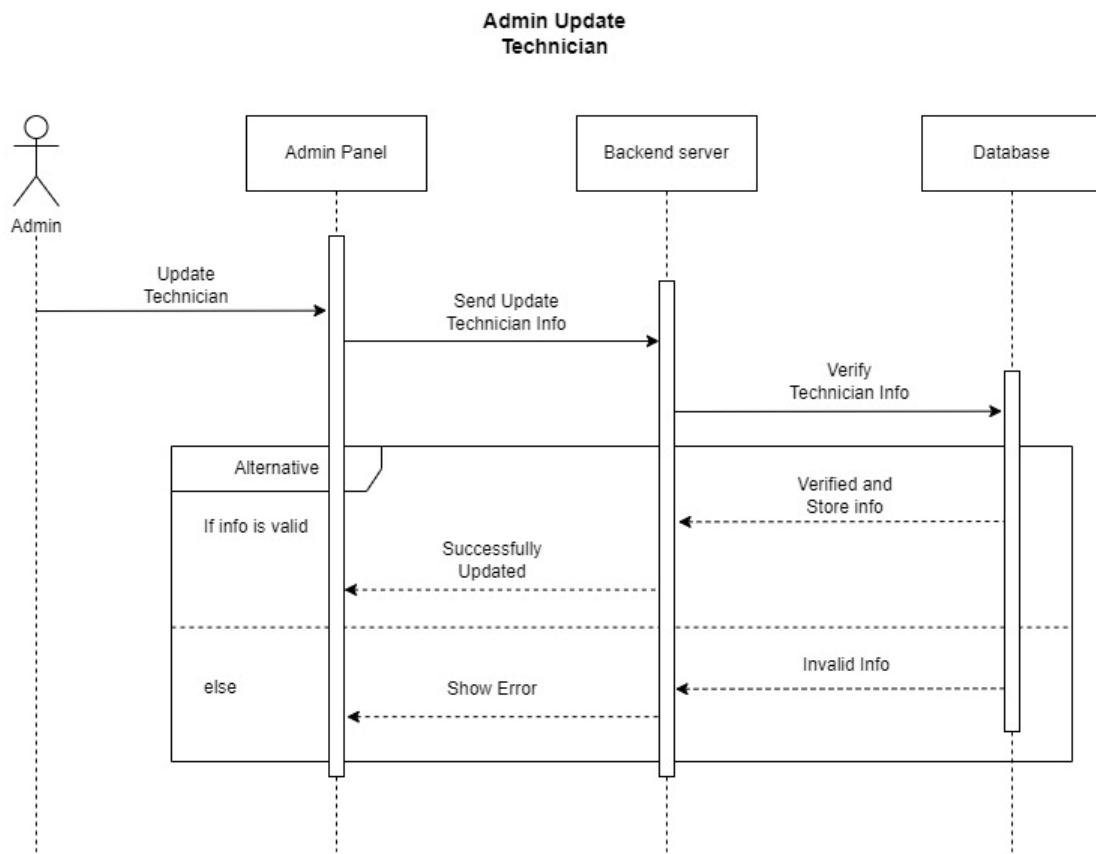


Figure 15: Admin Update Technician Sequence Diagram

3.5.11 Customer Registration

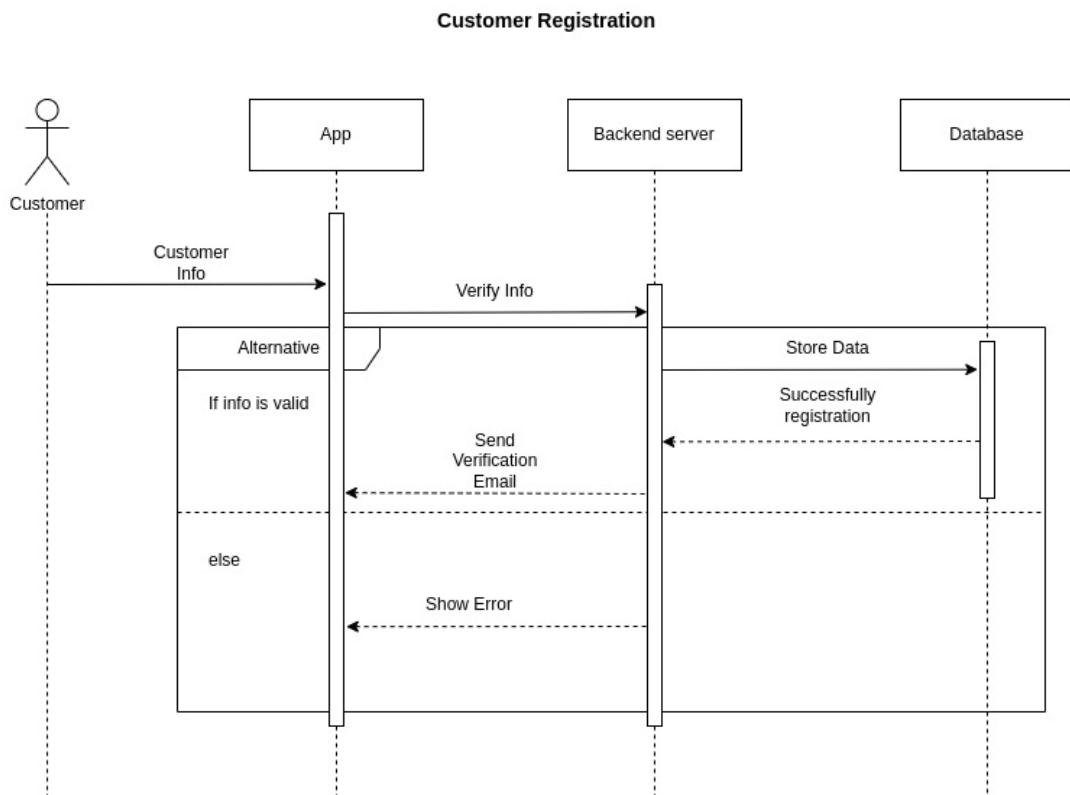


Figure 16: Customer Registration Sequence Diagram

3.5.12 Customer Login

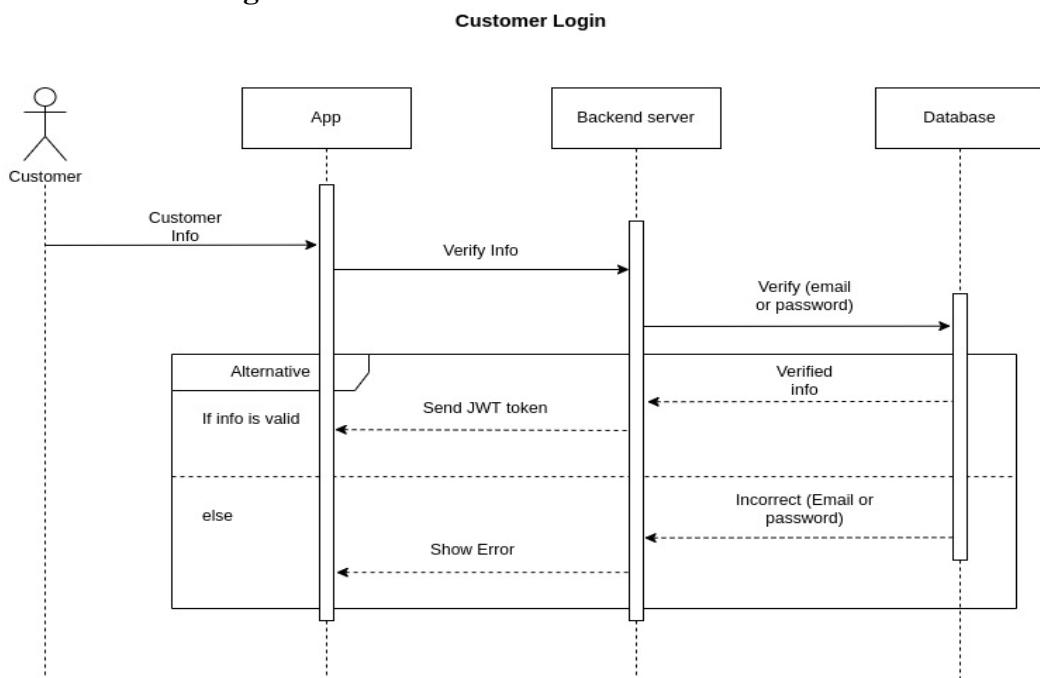


Figure 17: Customer Login Sequence Diagram

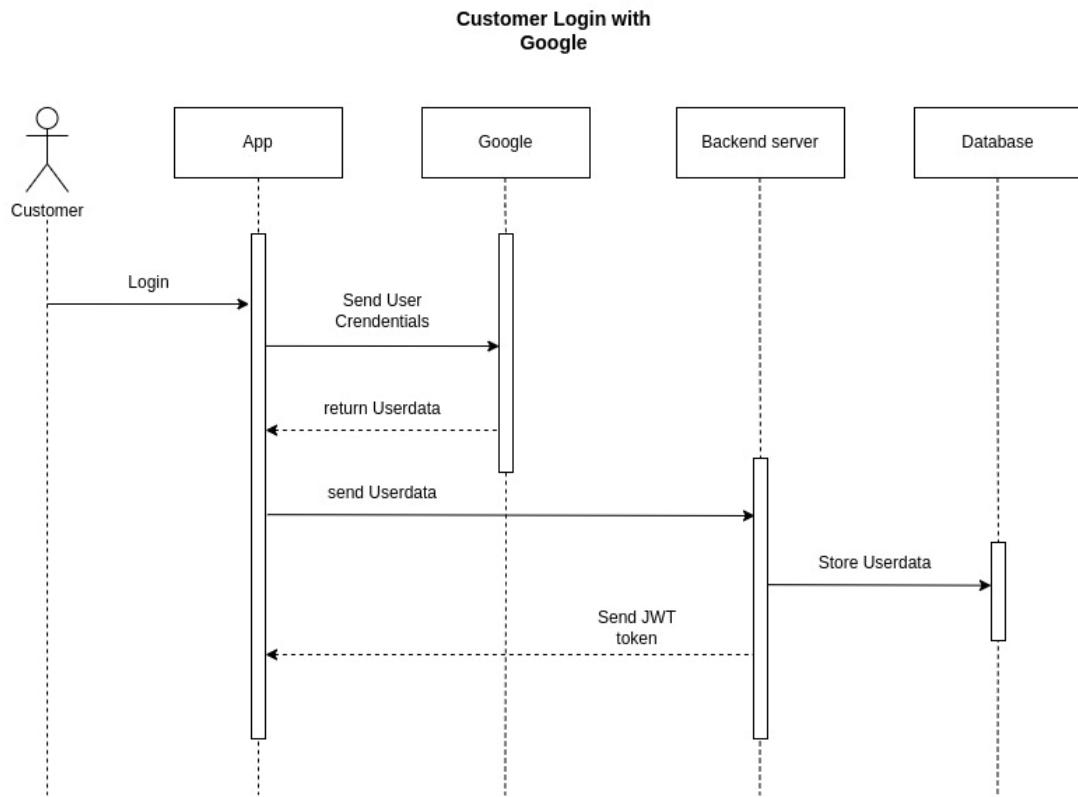


Figure 18: Customer Login with Google Sequence Diagram

3.5.13 Customer Profile

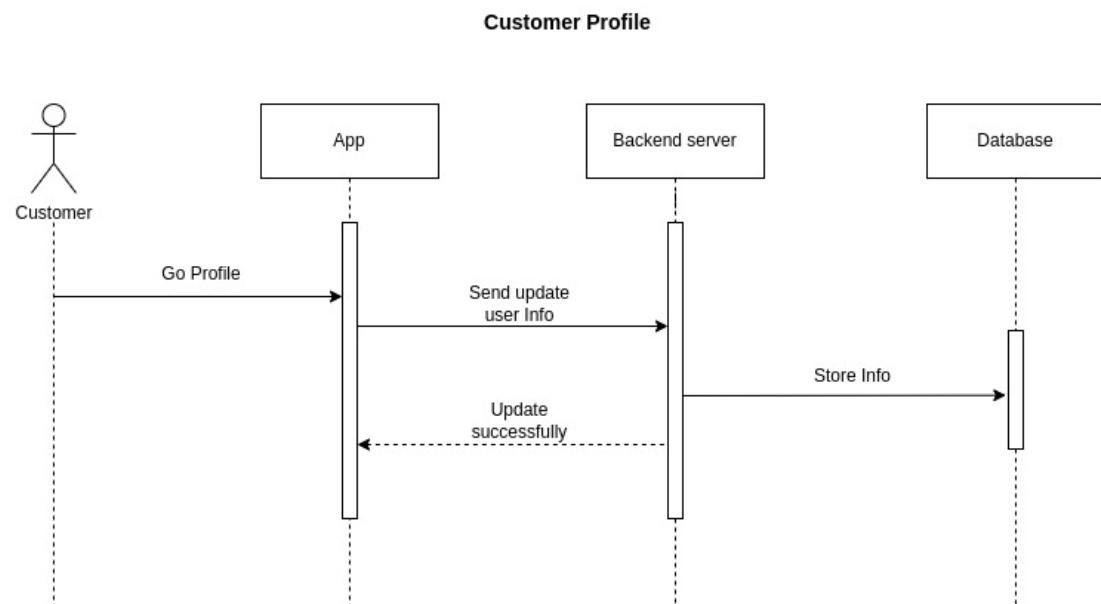


Figure 19: Customer Profile Sequence Diagram

3.5.14 Forget Password

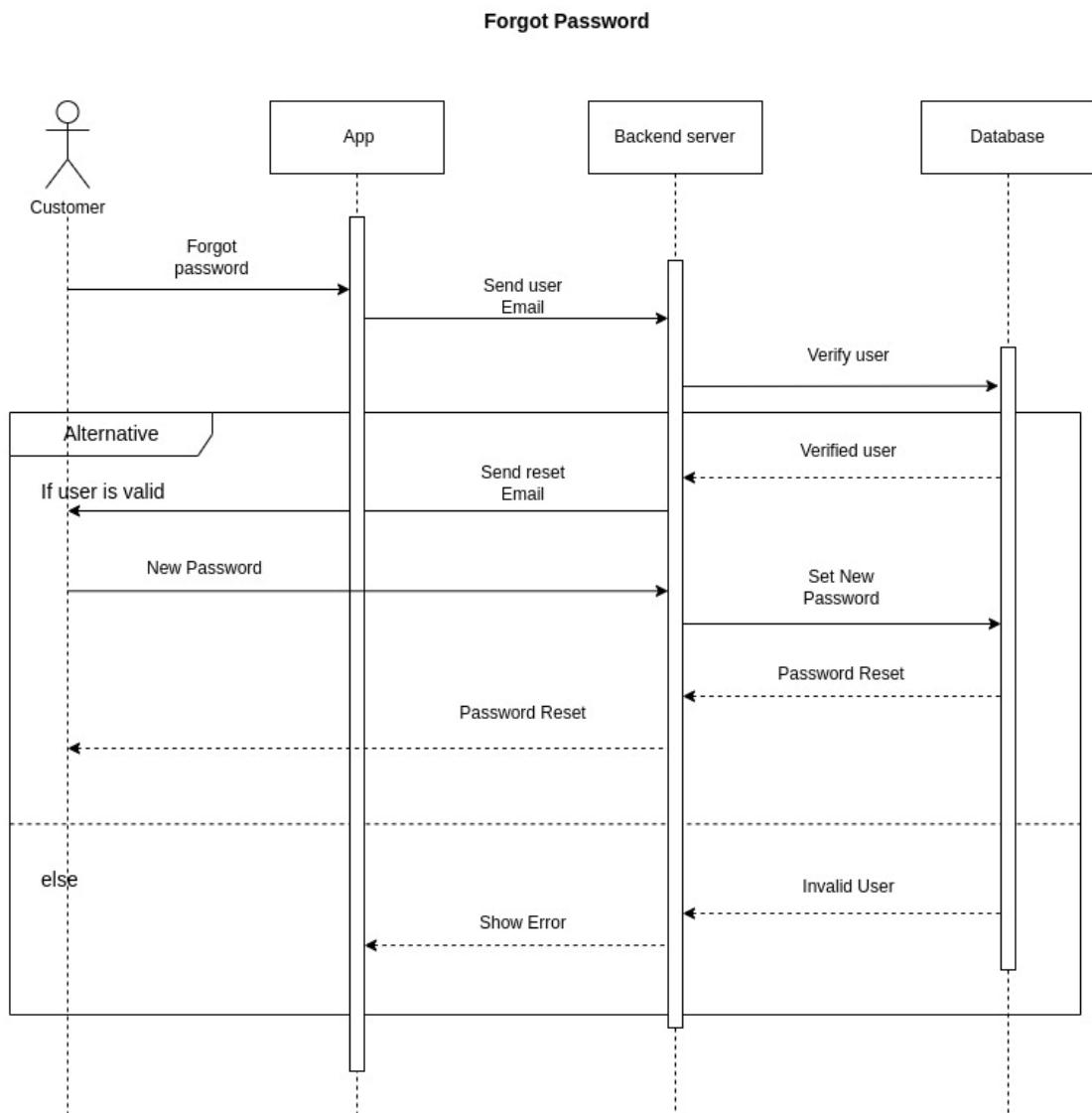


Figure 20: Customer Forget Password Sequence Diagram

3.5.15 Customer Add Order

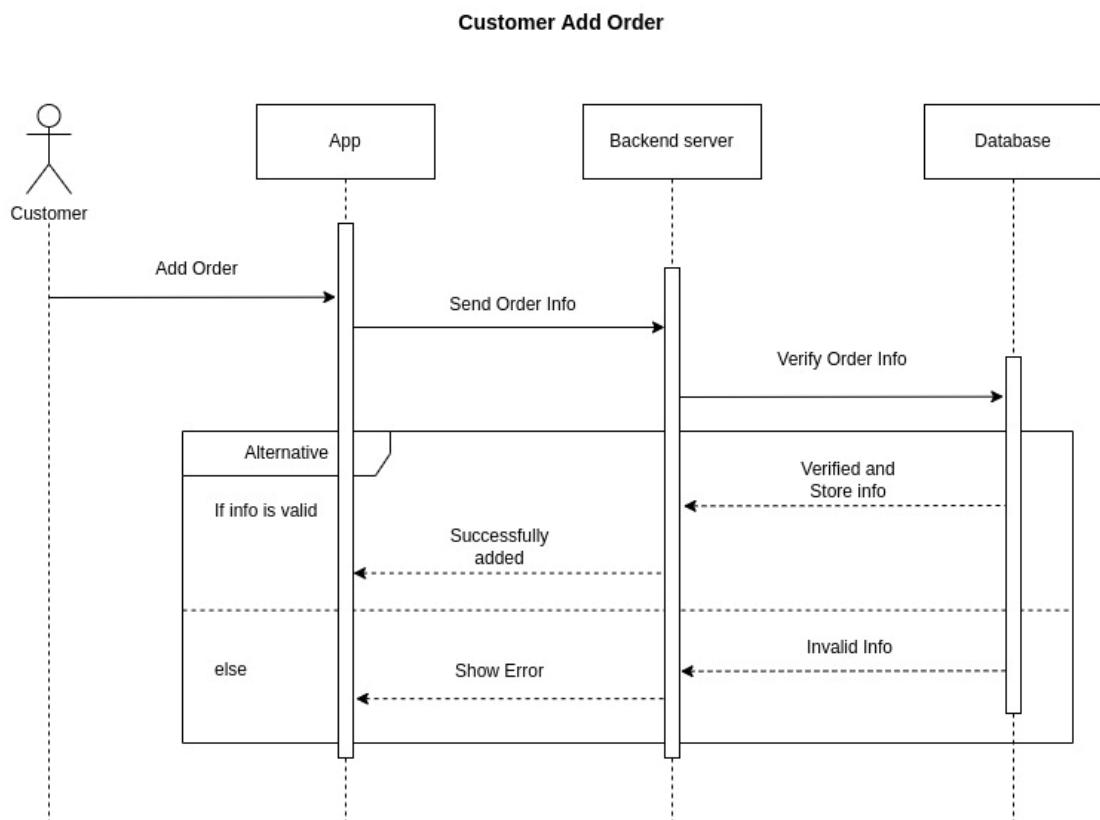


Figure 21: Customer Add Order Sequence Diagram

3.5.16 Customer Add Address

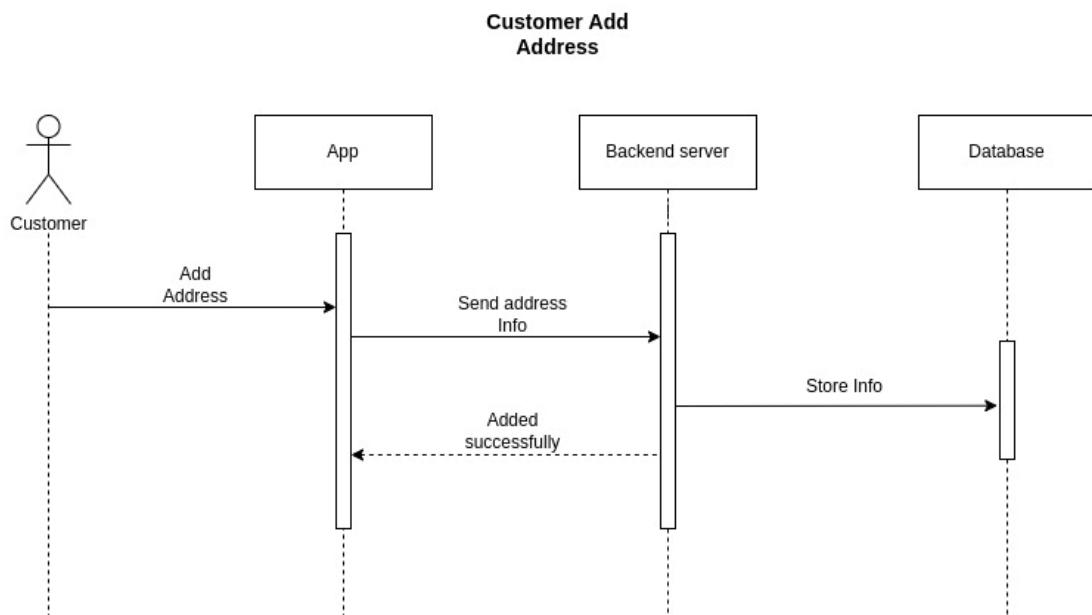


Figure 22: Customer Add Address Sequence Diagram

3.5.17 Customer Update Address

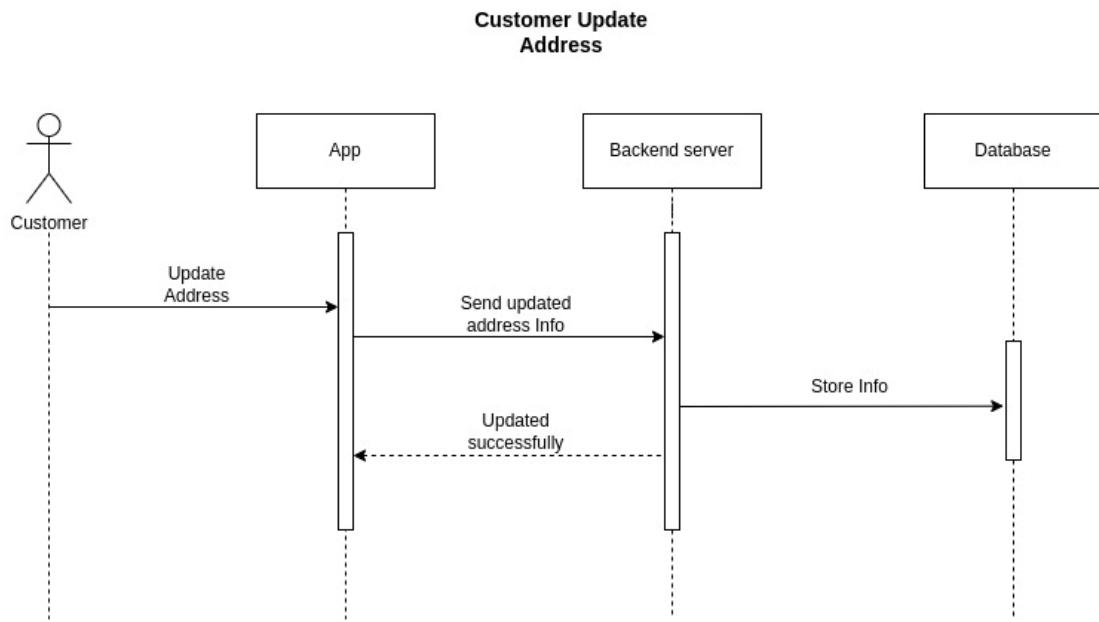


Figure 23: Customer Update Address Sequence Diagram

3.5.18 Cart List

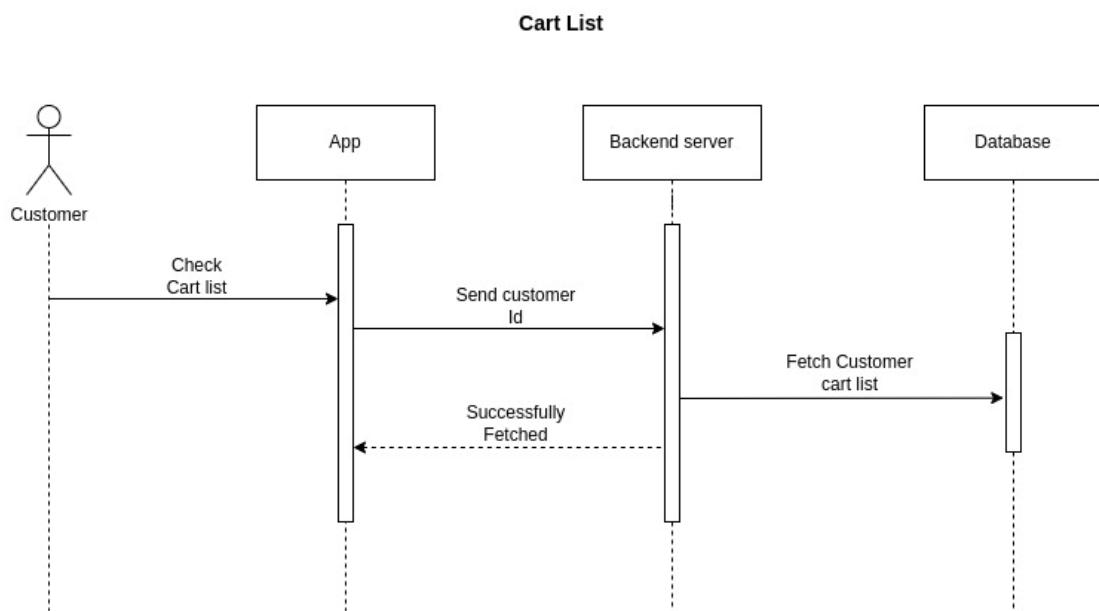


Figure 24: Cart List Sequence Diagram

3.5.19 Track Order

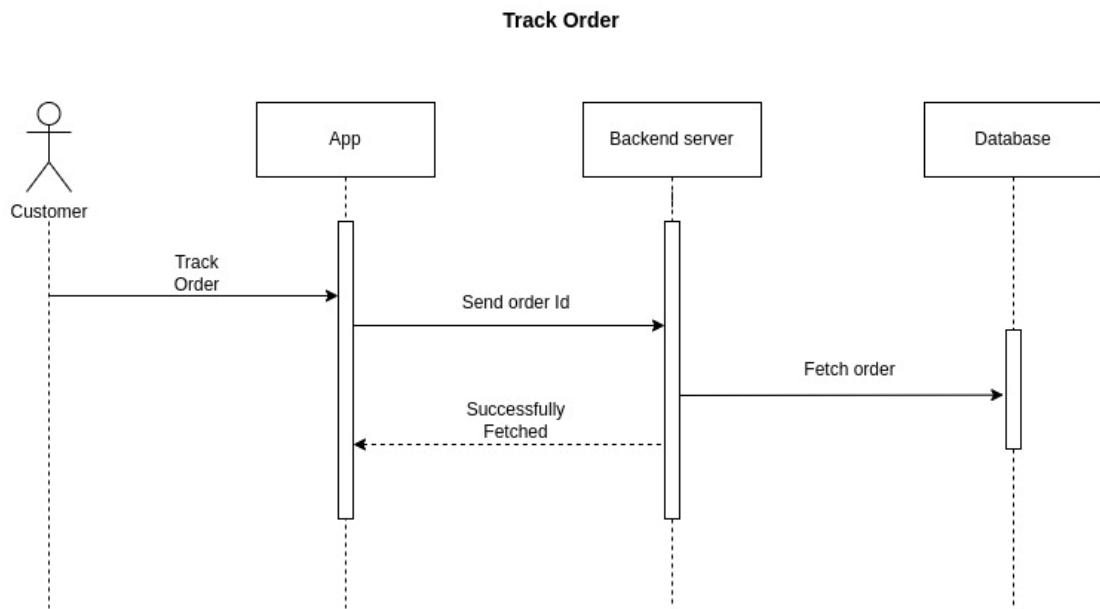


Figure 25: Track Order Sequence Diagram

3.5.20 Order History

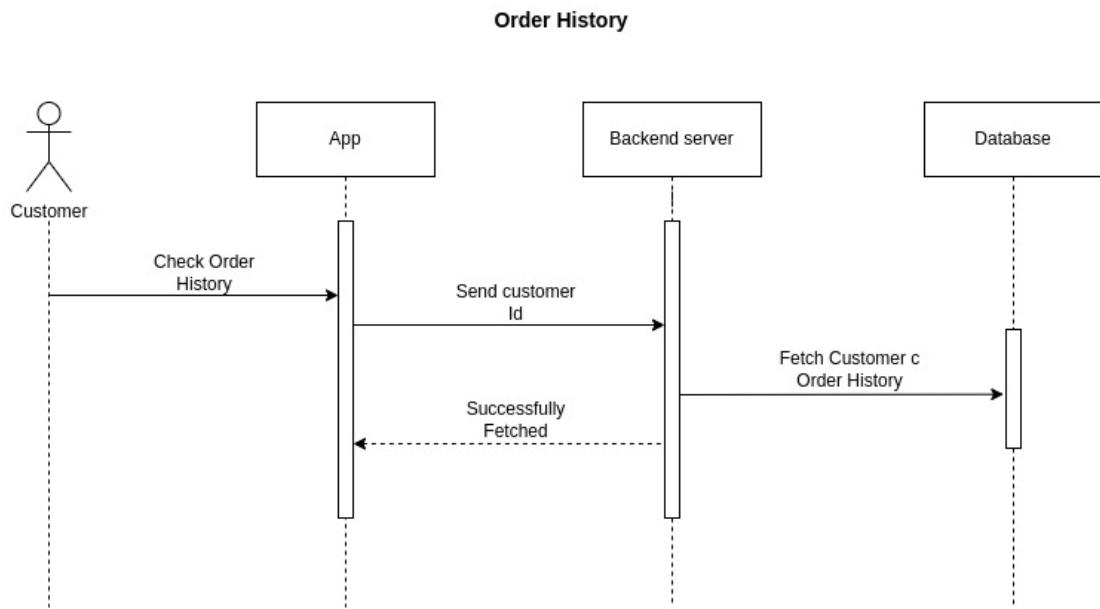


Figure 26: Order History Sequence Diagram

3.5.21 Payment

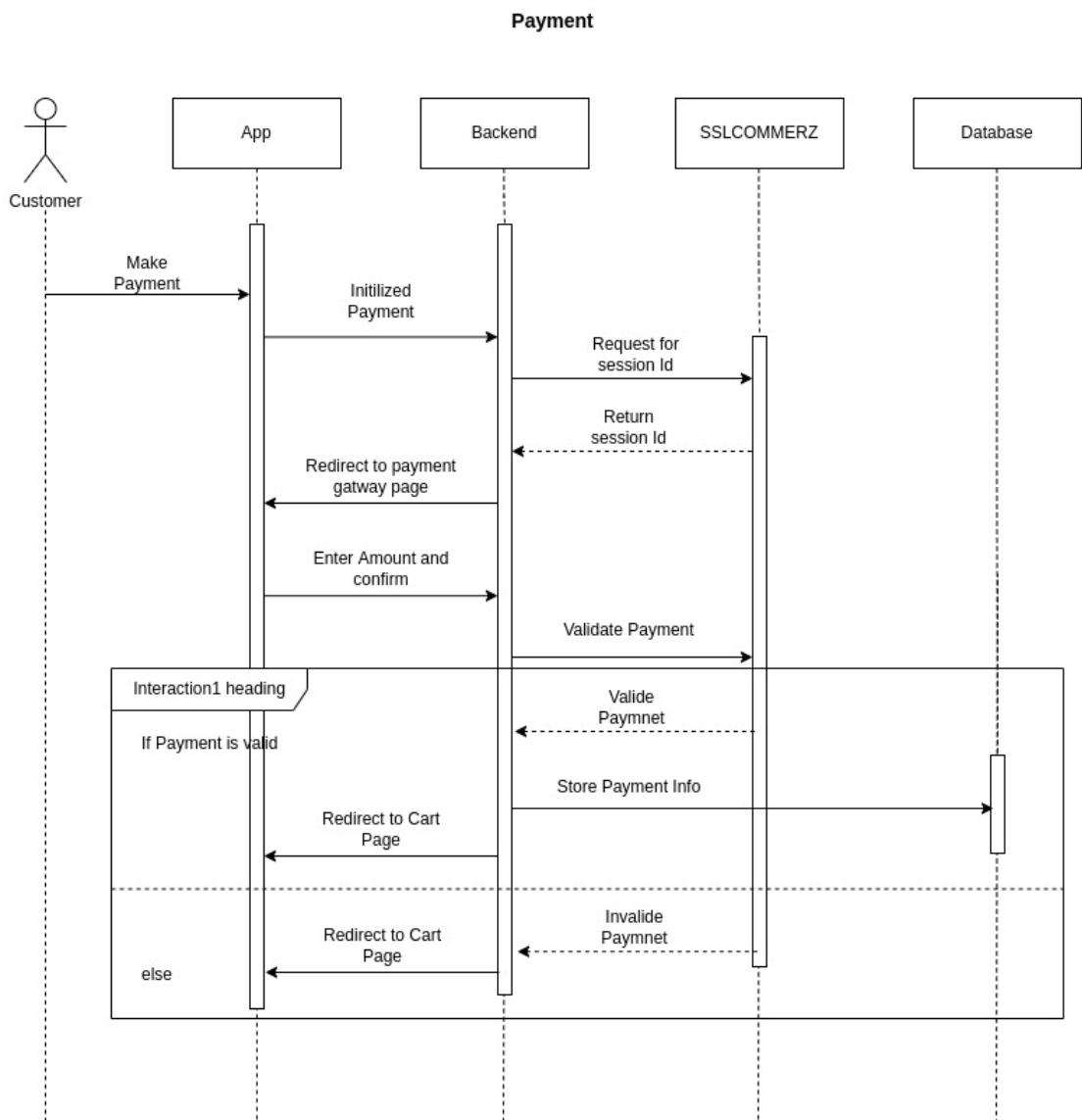


Figure 27: Payment Sequence Diagram

3.6 Design Requirement

We research and analyze so many designs for making an attractive design for our project. We have three parts to our project. First the android app, second the admin panel for controlling the system, and third the back-end application. For the android app, we will use React Native to make a cross-platform android app. For the admin panel, we will use HTML5, CSS3, Bootstrap, and React JS and for the back-end application, we will use NodeJS and MongoDB database.

A design must need to be,

Simple

Responsive

Easy to access

Different user can access

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front End Design

4.1.1 Android App

Welcome Screen

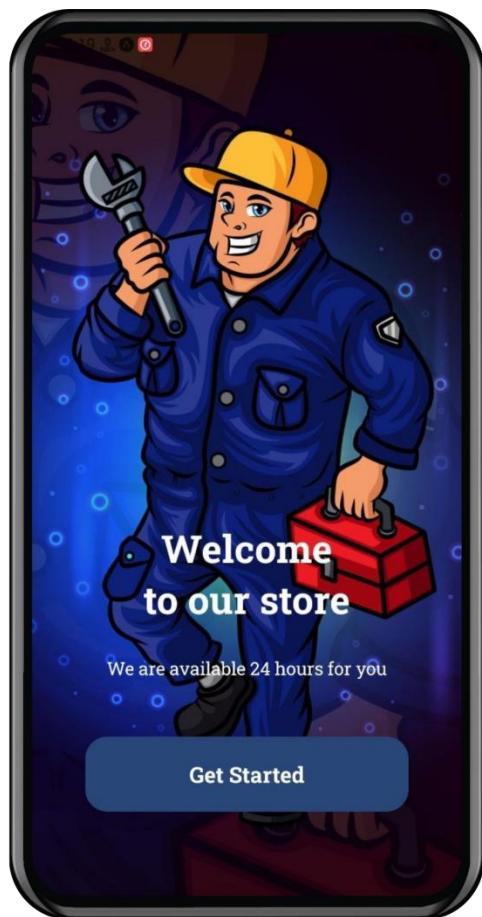


Figure 28: Welcome Page

Login Page

Users can log in to the system in two ways. They can use email and password for their login or they can use a Google account for login. If anyone forgot their password then there is a forgot password button that will send a reset email to the email address to reset the password.

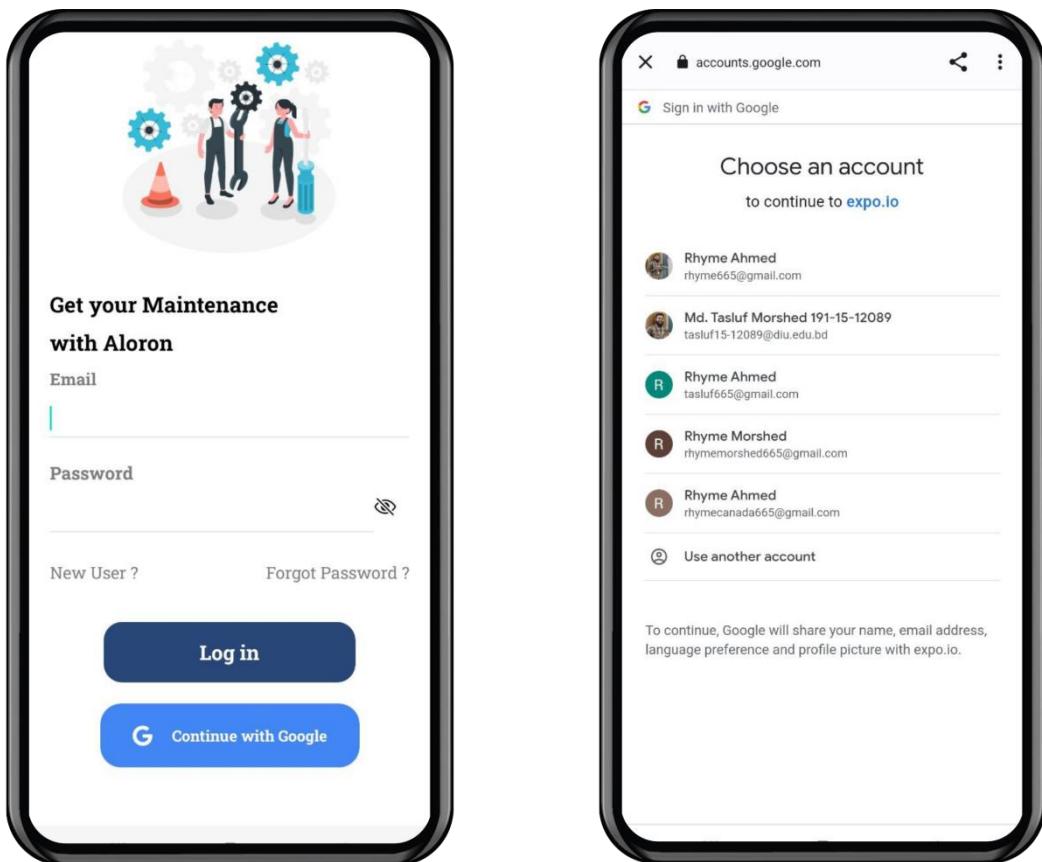


Figure 29: Login Page

Sign Up Page

Back-end authenticates sign-up need to enter in this application. Required information (1) Name, (2) Email, (3) Password (4) Confirm Password then the user should Click on the “Sign Up” button. Then the user gets a verification email with a verification link. If the user clicks that link back-end will automatically verify the user account and allow that user to log in.

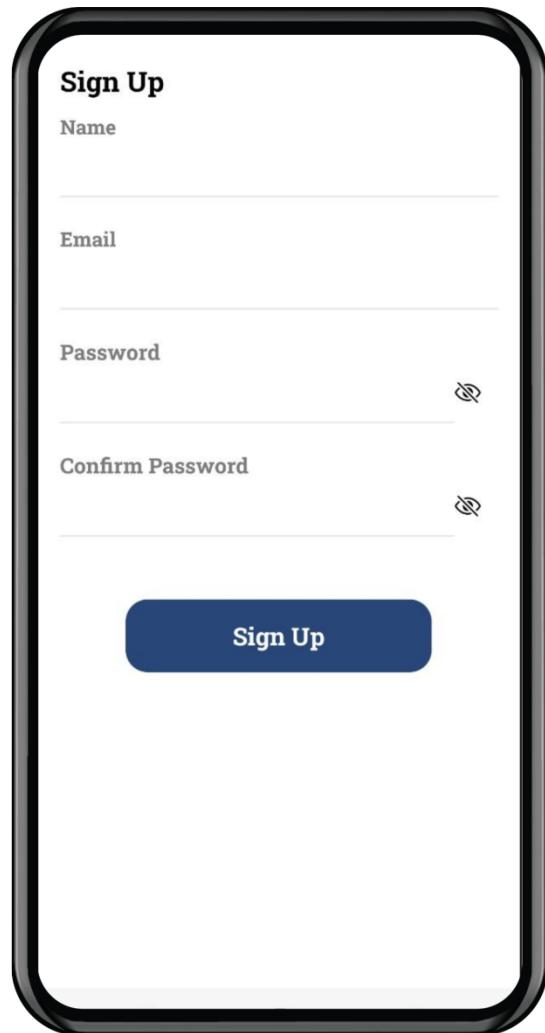


Figure 30: Sign Up Page

Home Page

After successful login in or sign-up, the user will redirect to the home page.

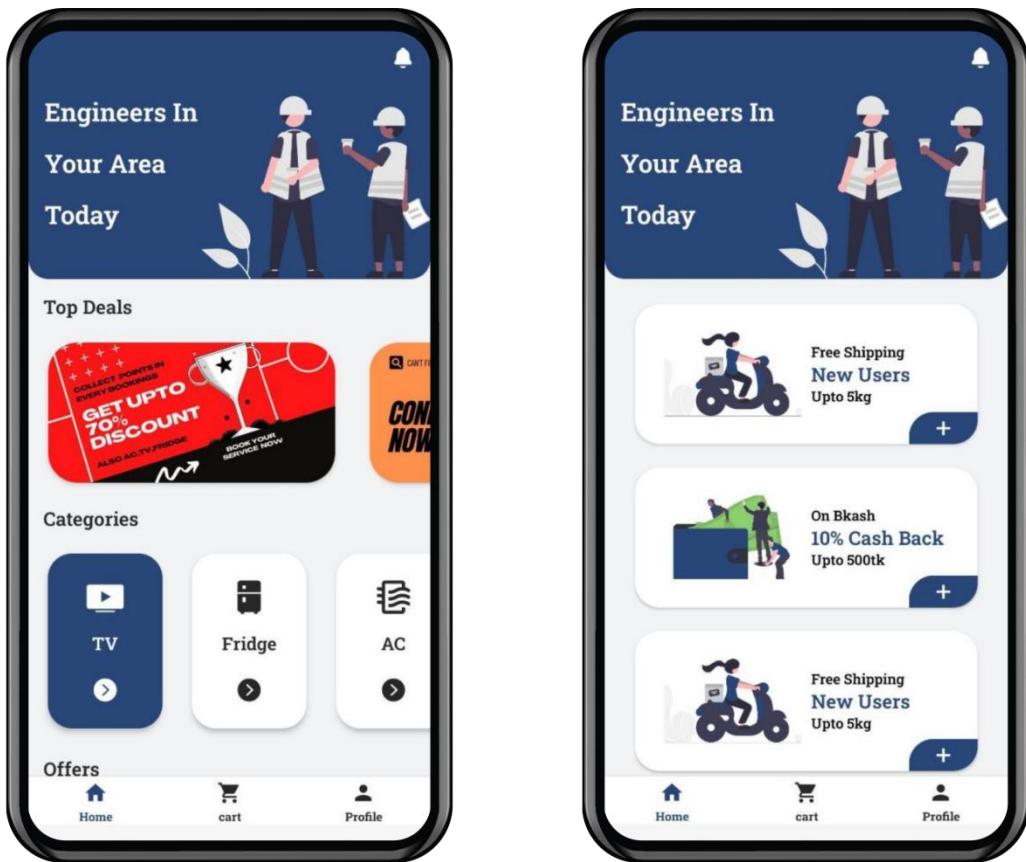


Figure 31: Home Page

Order Page

By submitting a form users can hire technicians to repair their products. They have to fill up some information and book an order.

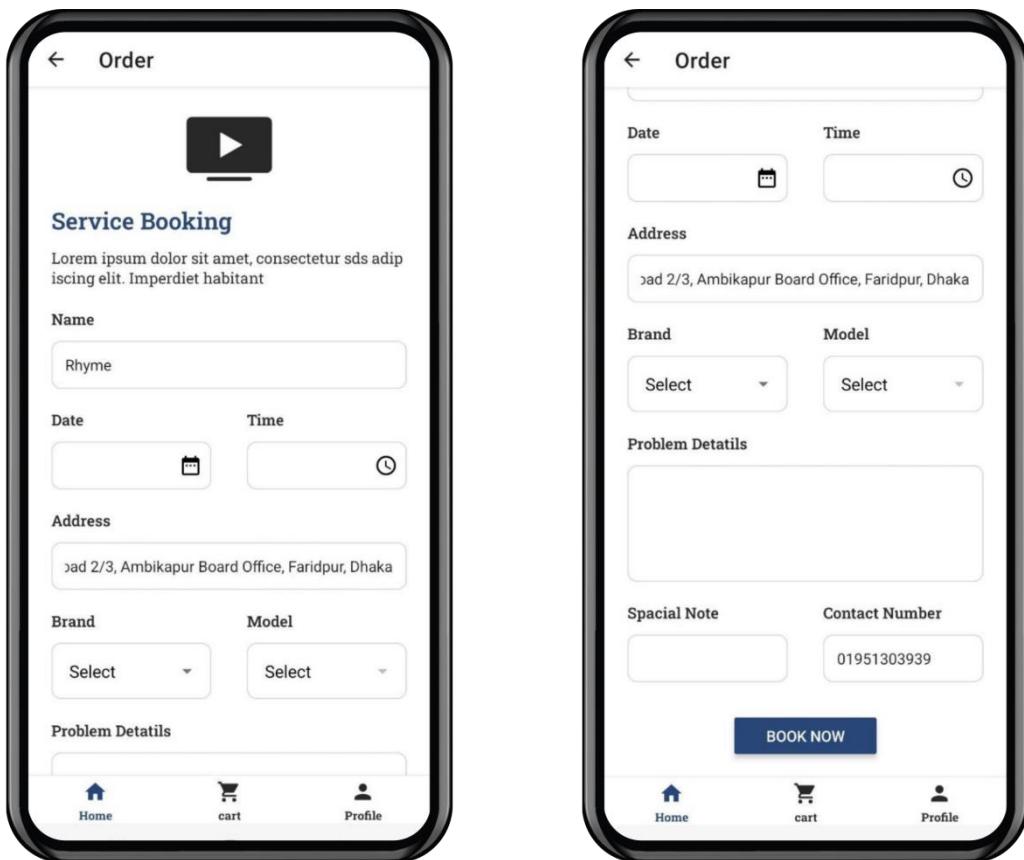


Figure 32: Order Page

My Cart Page

After submitting an order users can track their order on the My cart page. And after repairing the product user will be able to pay their bills.

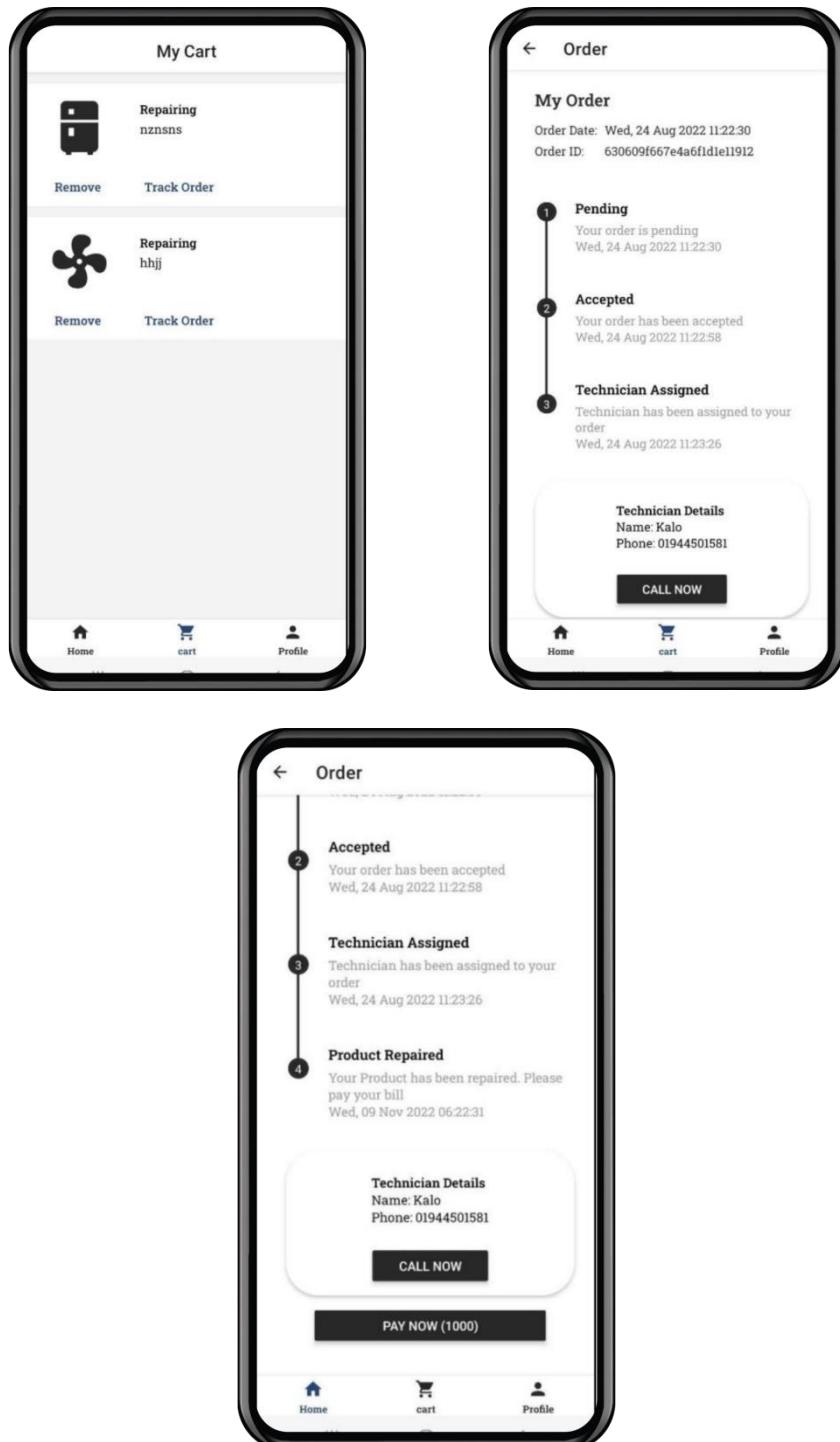


Figure 33: My Cart Page

Payment Page

Users are able to pay their bills by sslcommerz payment gateway. With this getaway, they can use any kind of internet banking system like Bkash, Nagad, Rocket, etc. They can also use Visa and Master Card for payment.

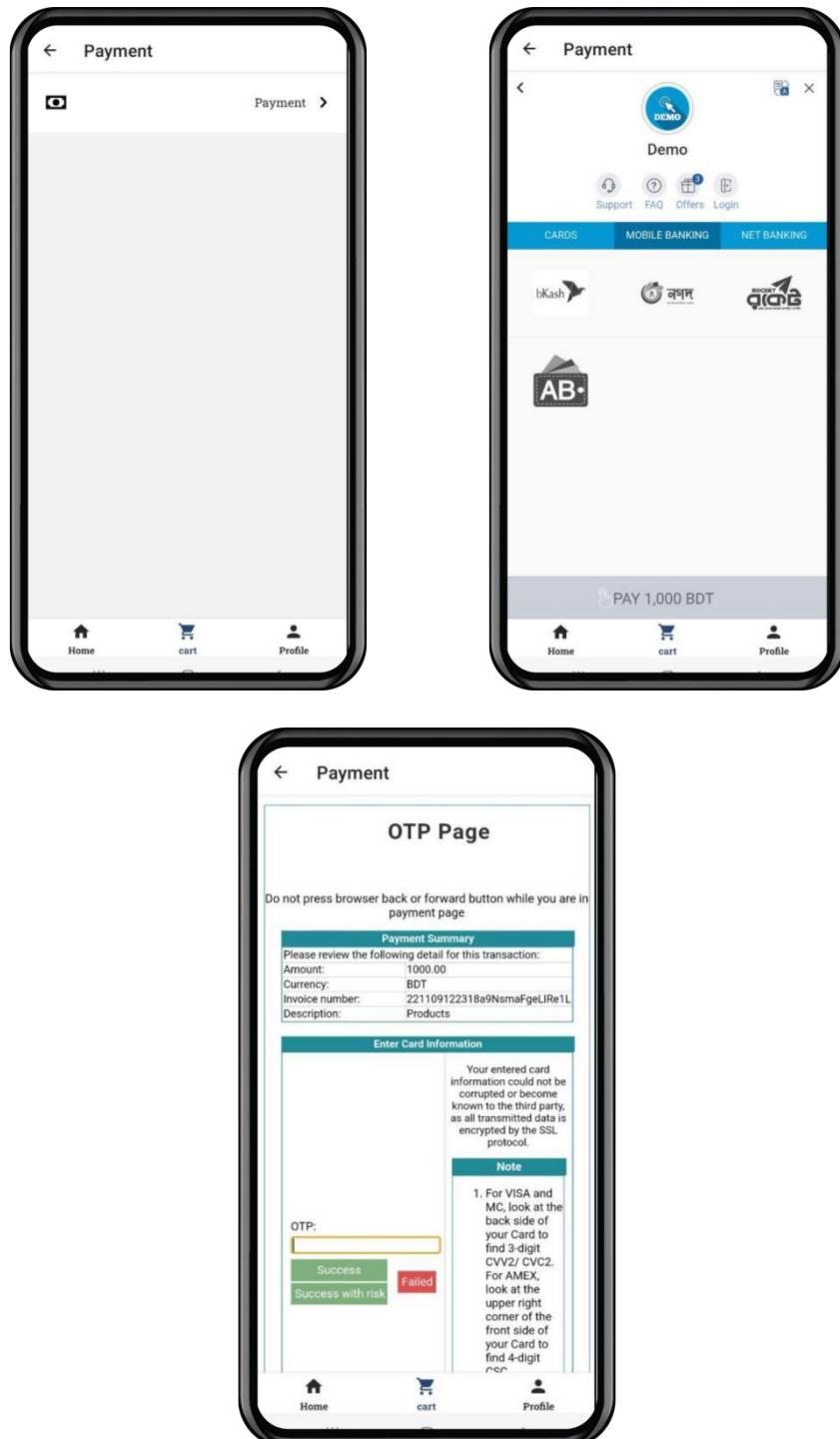


Figure 34: Payment Page

Profile Page

User will see their profile and other information on this page. They can change their profile picture. If the user press How can we help you? then it will make a phone call to customer care.

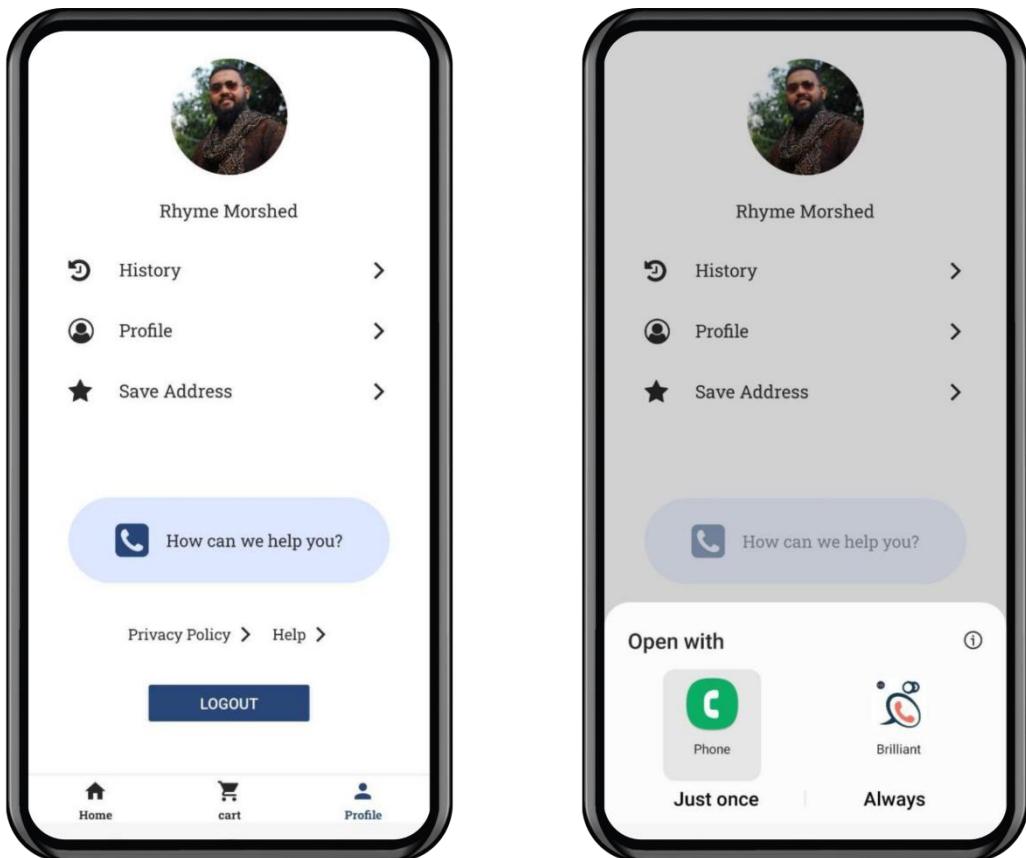


Figure 35: Profile Page

History Page

After repair, the product users order will be listed on the history page. Here user can see all his/her previous order details.

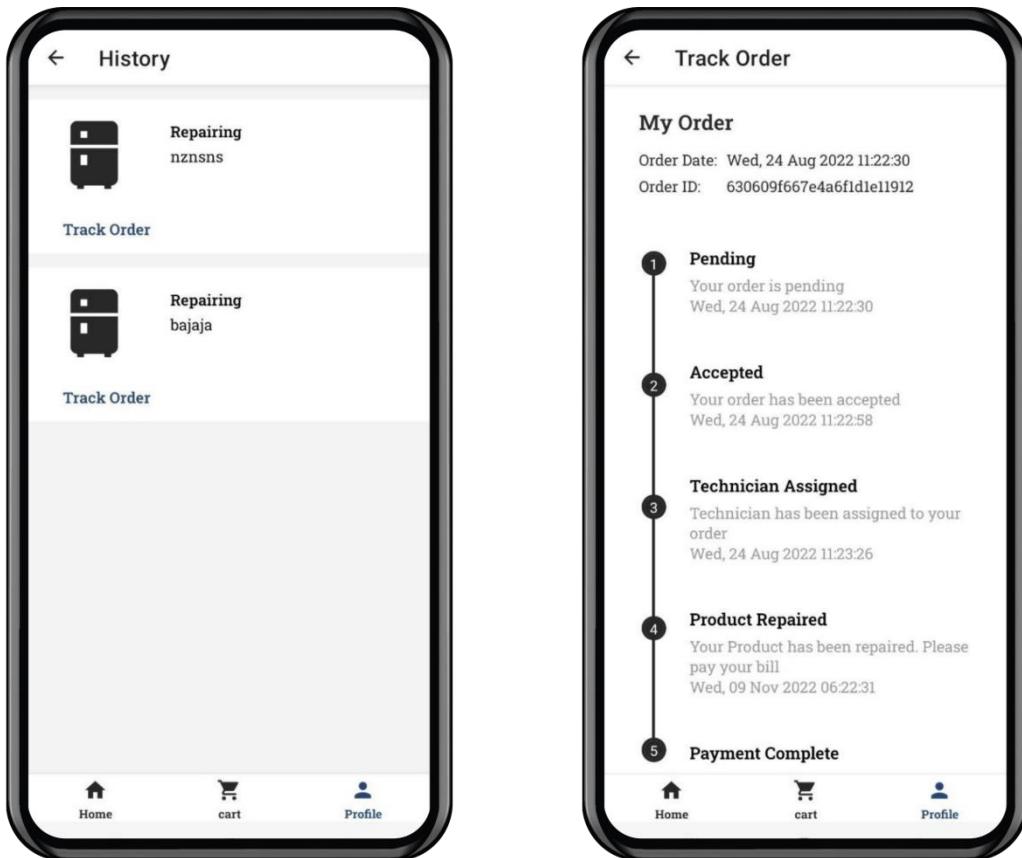


Figure 36: History Page

Profile Edit Page

Here user can update their profile. They can edit their name. Add phone number, select gender and select birthday.

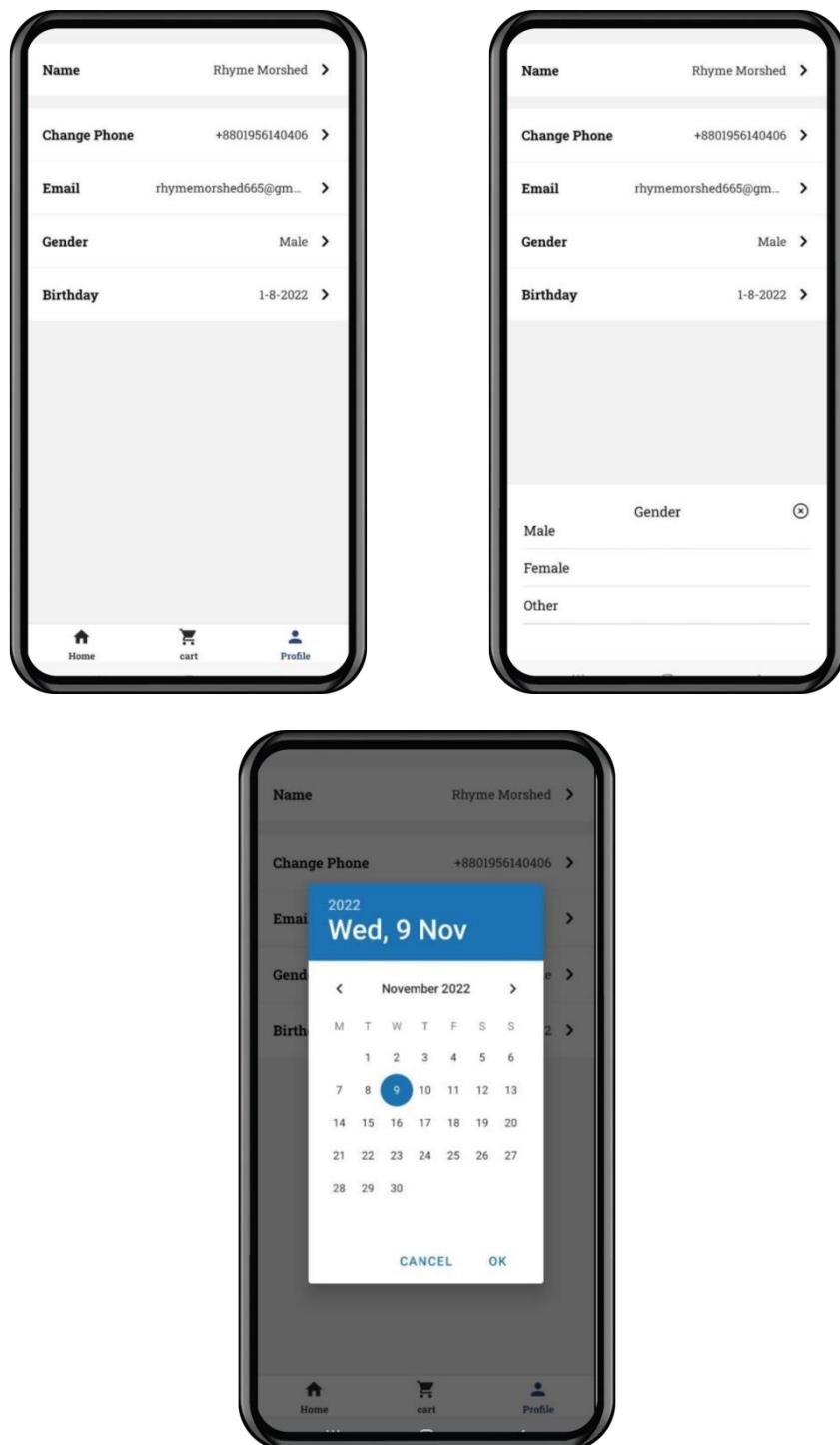


Figure 37: Profile Edit Page

Address Page

User can add their home and office address. Update and delete the current address.

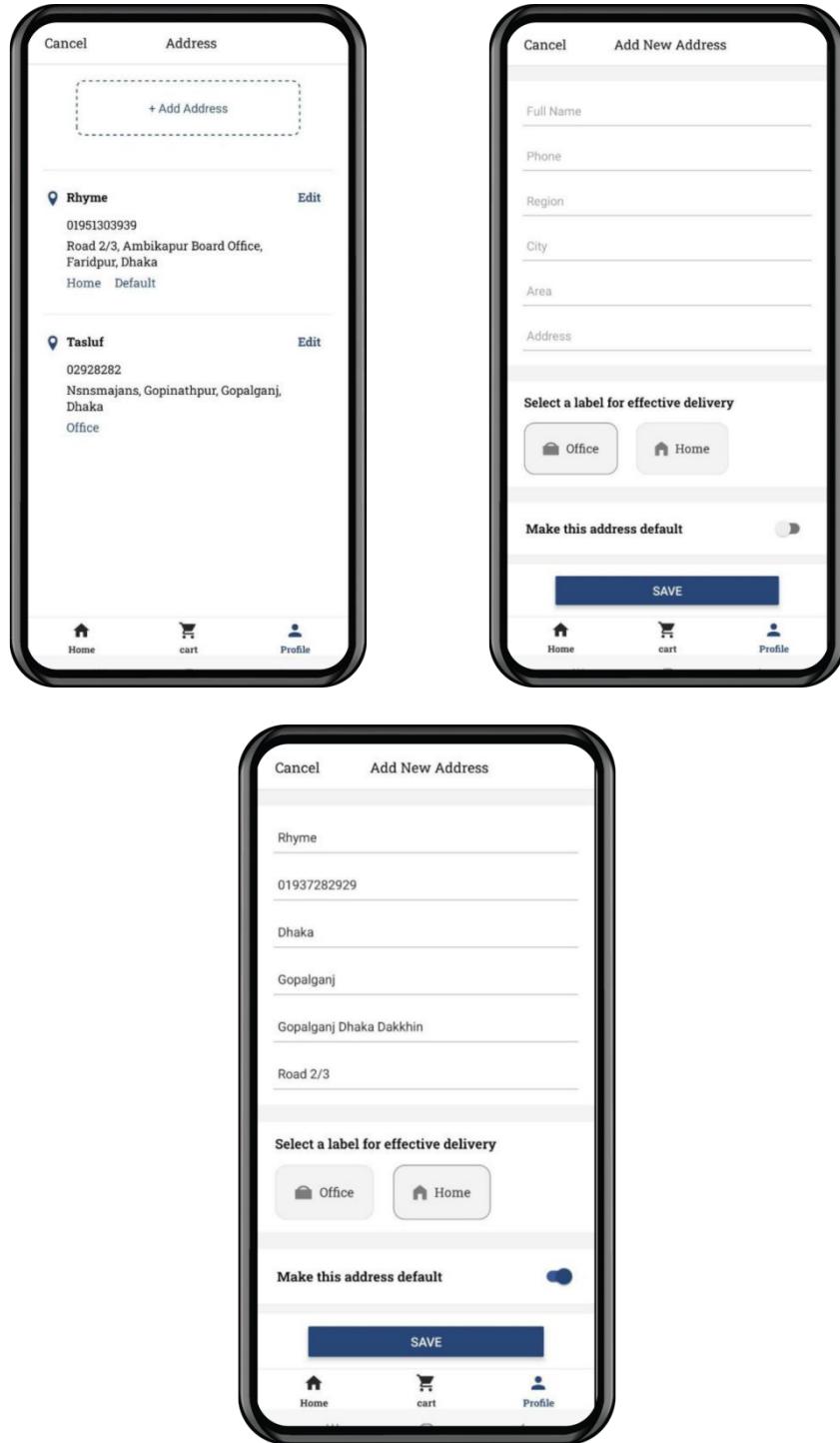


Figure 38: Address Page

Select Address Page

Users can select the Region, City, and area of their address

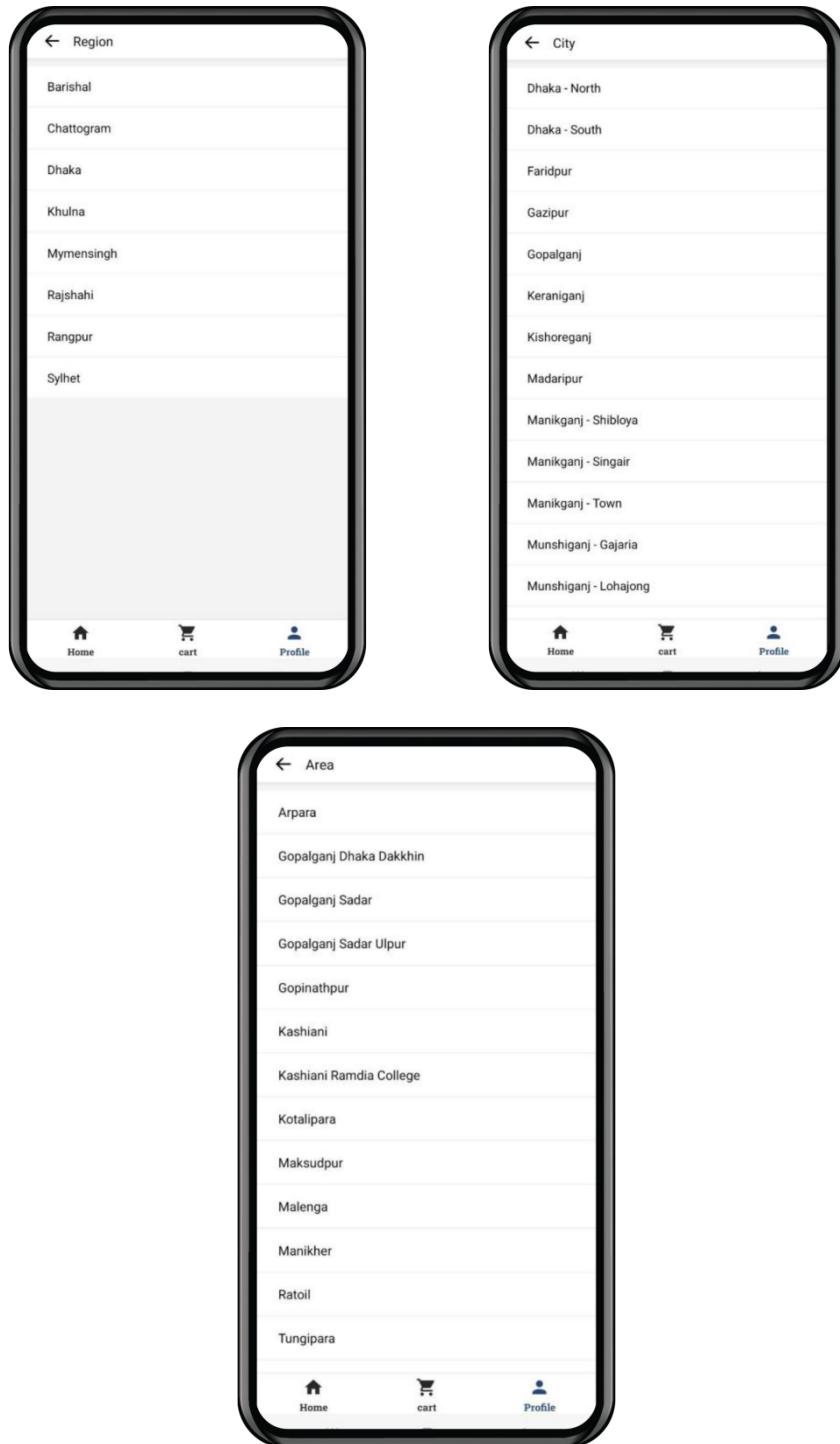


Figure 38: Select Address Page

Notification Page

All the notifications of the app will be shown on this page.

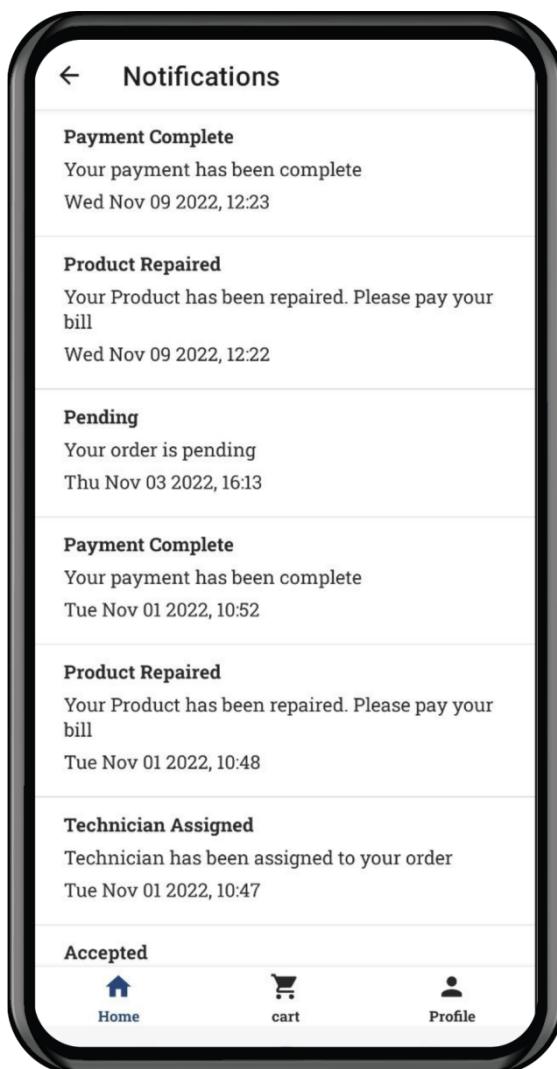
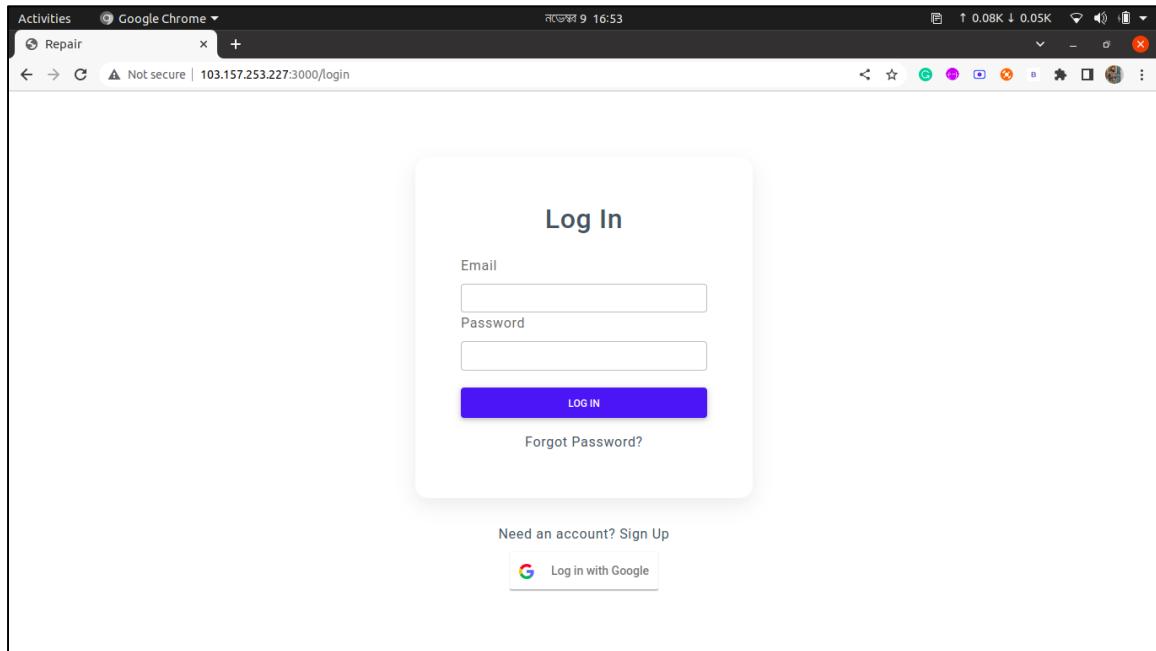


Figure 39: Notification Page

4.1.2 Admin Panel

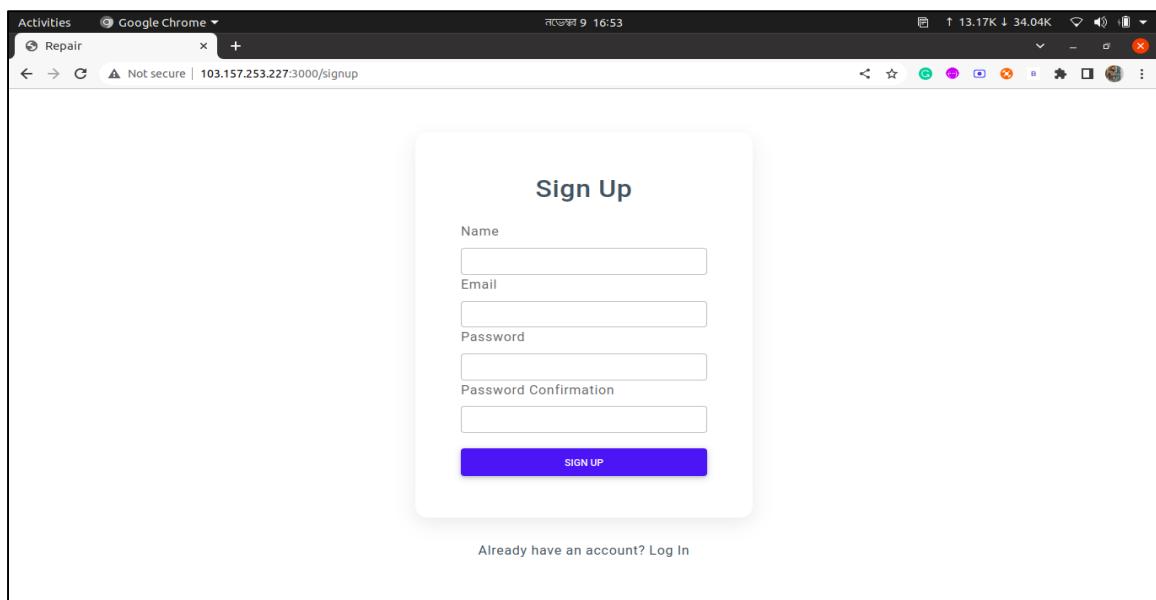
Login Page



A screenshot of a web browser window showing the login page. The title bar says "Activities Google Chrome". The address bar shows "Not secure | 103.157.253.227:3000/login". The main content is a "Log In" form with fields for Email and Password, a "LOG IN" button, a "Forgot Password?" link, and links for "Need an account? Sign Up" and "Log in with Google".

Figure 40: Login Page

Sign Up Page



A screenshot of a web browser window showing the sign up page. The title bar says "Activities Google Chrome". The address bar shows "Not secure | 103.157.253.227:3000/signup". The main content is a "Sign Up" form with fields for Name, Email, Password, and Password Confirmation, a "SIGN UP" button, and a link for "Already have an account? Log In".

Figure 41: Sign Up Page

Forgot Password Page

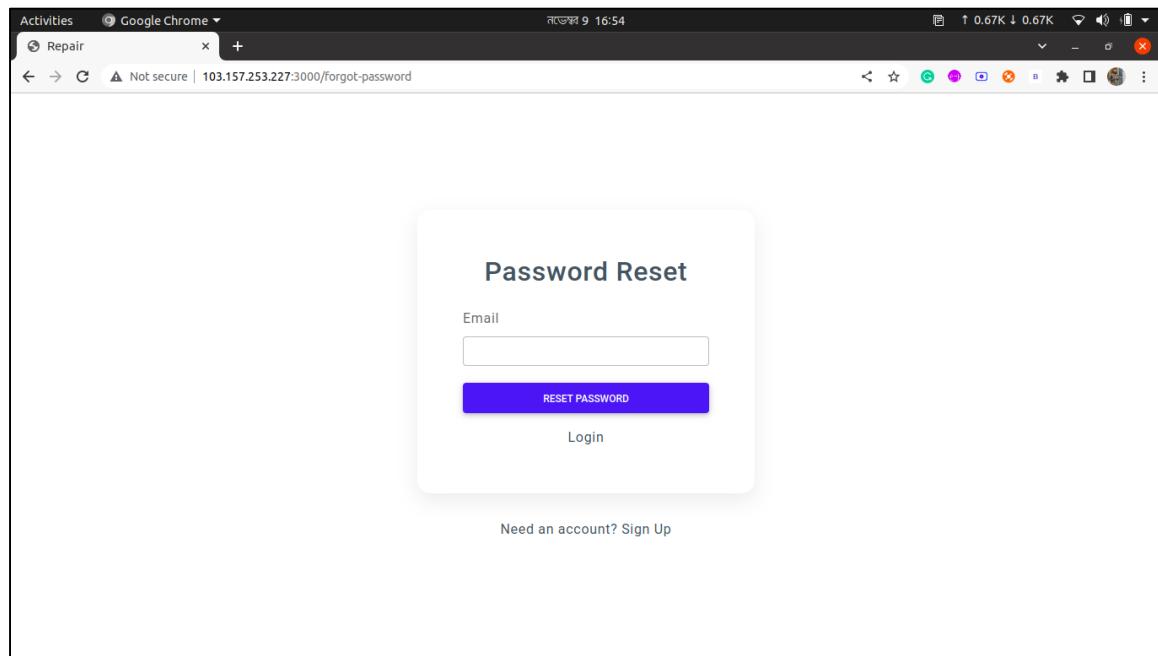


Figure 42: Forgot Password Page

Dashboard Page

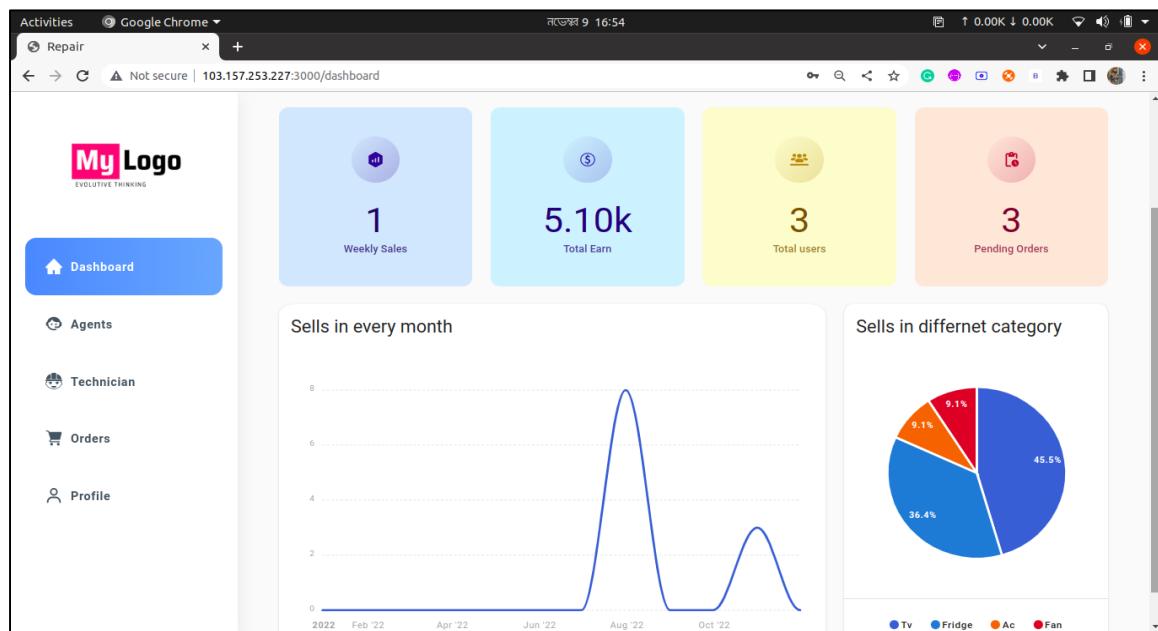


Figure 43: Dashboard Page

Agent Page

The screenshot shows a web application interface titled "Agents". On the left, there is a sidebar with icons for Dashboard, Agents (selected), Technician, Orders, and Profile. The main area displays a table with the following data:

	Name	Phone	Region	City	Area	Location	
1	Raida	01956140407	Dhaka	Narayanganj	Rupganj Kanchan	Road 2/20	1
2	Rhyme	01956140407	Dhaka	Munshiganj - Town	Munshiganj Sadar Mirkadim	Road 2/20	2
3	Tilok	01956140407	Dhaka	Gazipur	Board Bazar	Road 2/34	3

A blue "ADD AGENTS" button is located in the top right corner of the main area.

Figure 44: Agent Page

Add Agent Page

The screenshot shows a web application interface titled "Add Agent". On the left, there is a sidebar with icons for Dashboard, Agents, Technician, Orders, and Profile. The main area contains a form with the following fields:

Name	Email
Select Name	Select Email
Phone	Whatsapp Number
Select Phone	Select Whatsapp Number
Location	Region
Select Location	Barishal
City	Area

A blue "SUBMIT" button is located at the bottom right of the form.

Figure 45: Add Agent Page

Update Agent Page

My Logo
EVOLUTIVE THINKING

Update Agent

EDIT

Name	Email
Raida	raida665@gmail.com
Phone	Whatsapp Number
01956140407	01956140407
Location	Region
Road 2/20	Dhaka
City	Area
Narayanganj	Rupganj Kanchan

Figure 46: Update Agent Page

Technician Page

My Logo
EVOLUTIVE THINKING

	Name	Phone	Region	City	Area	Agent		
1	Bijoy	5866100735	Khulna	Kushtia	Mongolbari Bazar	1	Rhyme	Details
2	Kalam	5866100735	Chattogram	Chhagalnaia	Box Mahmud	2	Rhyme	Details
3	Kalo	01944501581	Mymensingh	Mymensingh - Muktagacha	Muktagacha Sadar	3	Tilok	Details
4	Raju	5866100735	Chattogram	Chhagalnaia	Chhagalnaia Daraga Hat	4	Rhyme	Details
5	Ridoy	01956140407	Dhaka	Narayanganj	Rupganj Murapara	5	Rhyme	Details
6	Rifat	5866100735	Dhaka	Narayanganj	Rupganj Bhulta	6	Rhyme	Details
7	Ripon	01944501581	Chattogram	Chattogram Sadar	Chattogram GPO	7	Rhyme	Details
8	Saju	5866100735	Khulna	Khulna -Amadee	Amadee Sadar	8	Rhyme	Details
9	Sobuj	01956140407	Dhaka	Narayanganj	Bandar	9	Rhyme	Details
10	Sojib	5866100735	Khulna	Meherpur	Gangni Nittyanandapur	10	Rhyme	Details

Figure 47: Technician Page

Add Technician Page

My Logo
EVOLUTIVE THINKING

Add Technician

Name

Email

Phone

Whatsapp Number

Location

Region

City

Area

Agent

SUBMIT

Figure 48: Add Technician Page

Order Page

Activities Google Chrome ▾

Repair x +

Not secure | 103.157.253.227:3000/orders

My Logo
EVOLUTIVE THINKING

Dashboard

Agents

Technician

Orders

Profile

Category	Type	Booking Time	Arrival Time	Phone	Status	Details
Repairing	Youtube-Tv	8/12/2022	8/19/2022	93939292020	Pending	Details
Repairing	Fan	11/3/2022	11/11/2022	01951303939	Pending	Details
Repairing	Youtube-Tv	8/11/2022	8/12/2022	029382822	Payment Complete	Details
Repairing	Fridge	8/11/2022	8/12/2022	01956140407	Payment Complete	Details
Repairing	Youtube-Tv	8/11/2022	8/12/2022	01956140407	Payment Complete	Details
Repairing	Youtube-Tv	8/13/2022	8/14/2022	01956140407	Payment Complete	Details
Repairing	Fridge	8/13/2022	8/15/2022	01956140407	Payment Complete	Details
Repairing	Air-Filter	8/13/2022	8/20/2022	01956140407	Technician Assigned	Details
Repairing	Fridge	8/24/2022	8/25/2022	01951303939	Payment Complete	Details
Repairing	Fridge	11/1/2022	11/8/2022	01951303939	Payment Complete	Details

1 2

Figure 49: Order Page

Order Details Page

The screenshot shows a web browser window titled "Update Order" with the URL "103.157.253.227:3000/updateorder". On the left, there is a sidebar with icons for Dashboard, Agents, Technician, Orders, and Profile. The Profile icon is highlighted with a blue background. The main content area contains fields for Name (nsnsns), Phone (029382822), Address (jajaja), Booking Time (8/11/2022), Arrival Time (8/12/2022, 4:25:40 PM), Category (Repairing), Category Type (youtube-tv), Brand (Sony Plus), Model (SONY PLUS 24" HD LED TV), Problem (bsnsnsn), Note (nssnsns), Status (Payment Complete), and Agent (Agent). A blue "EDIT" button is located in the top right corner.

Figure 50: Order Details Page

Profile Page

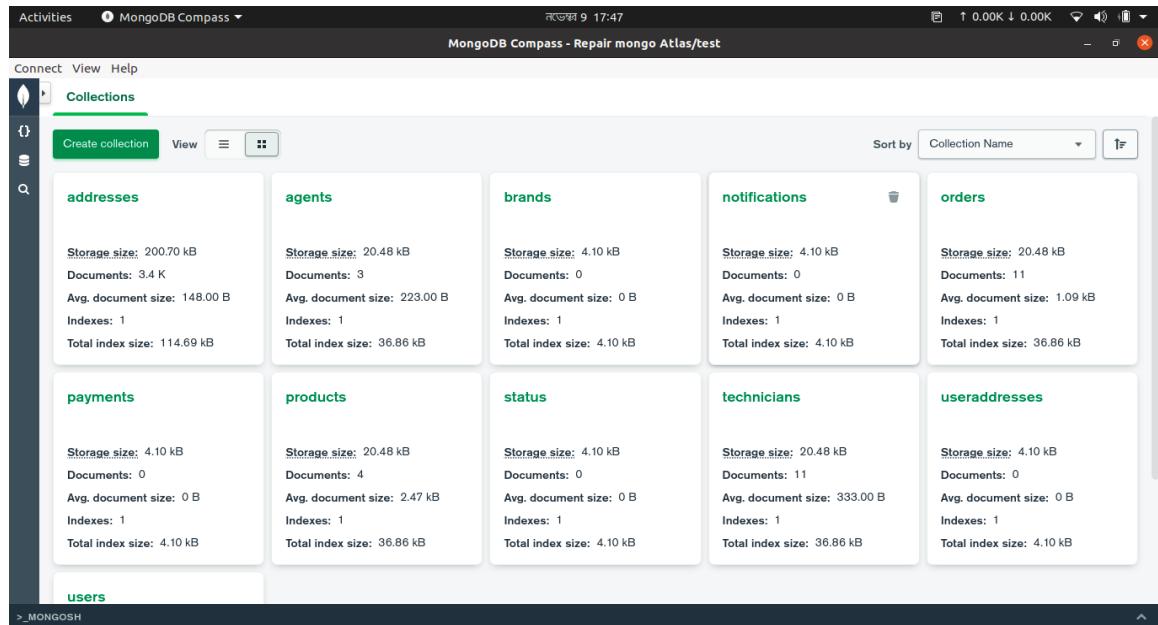
The screenshot shows a web browser window titled "Profile" with the URL "103.157.253.227:3000/profile". On the left, there is a sidebar with icons for Dashboard, Agents, Technician, Orders, and Profile. The Profile icon is highlighted with a blue background. The main content area contains fields for Name (Rhyme), Phone (01956140407), Gender (Male), and Birthday (12/16/1999). A blue "SUBMIT" button is located at the bottom right. A small user icon is visible in the top right corner of the main content area.

Figure 51: Profile Page

4.2 Back End Design

Database

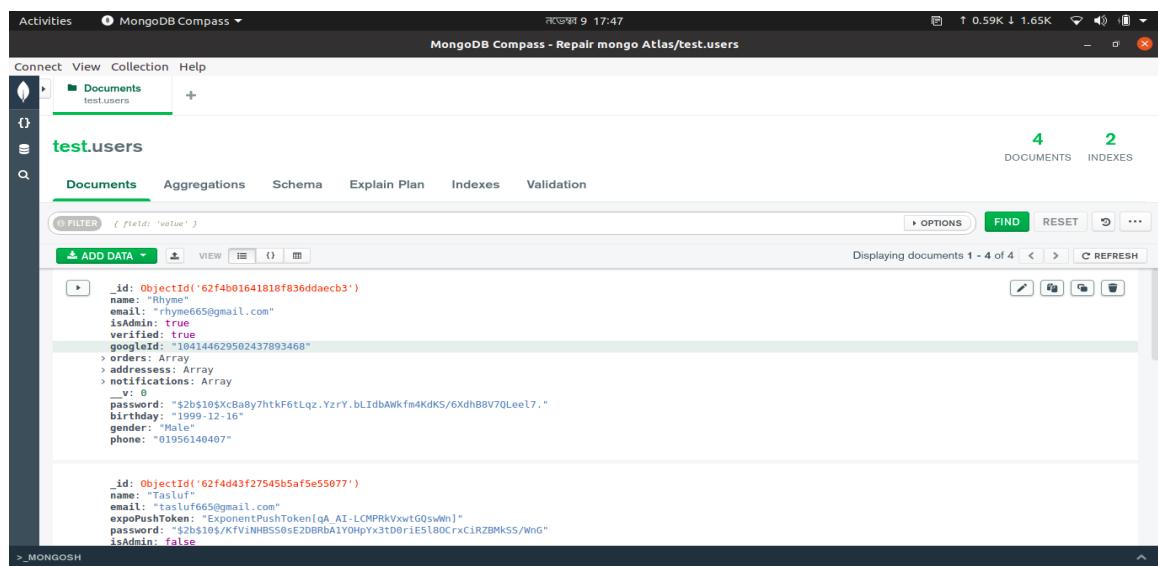
We used NodeJS for user authentication for this project. For storage, we use MongoDB atlas NoSQL database [cloud database storage].



The screenshot shows the MongoDB Compass interface with the title "MongoDB Compass - Repair mongo Atlas/test". The left sidebar has "Activities" and "MongoDB Compass" buttons. The main area is titled "Collections" and lists ten collections: addresses, agents, brands, notifications, orders, payments, products, status, technicians, and useraddresses. Each collection card displays its storage size, document count, average document size, index count, and total index size. Below the collections is a "users" section. At the bottom, there's a "MONGOSH" prompt.

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
addresses	200.70 kB	3.4 K	148.00 B	1	114.69 kB
agents	20.48 kB	3	223.00 B	1	36.86 kB
brands	4.10 kB	0	0 B	1	4.10 kB
notifications	4.10 kB	0	0 B	1	4.10 kB
orders	20.48 kB	11	1.09 kB	1	36.86 kB
payments	4.10 kB	0	0 B	1	4.10 kB
products	20.48 kB	4	2.47 kB	1	36.86 kB
status	4.10 kB	0	0 B	1	4.10 kB
technicians	20.48 kB	11	333.00 B	1	36.86 kB
useraddresses	4.10 kB	0	0 B	1	4.10 kB
users					

Figure 52: All Schema for database



The screenshot shows the MongoDB Compass interface with the title "MongoDB Compass - Repair mongo Atlas/test.users". The left sidebar has "Activities", "View", "Collection", and "Help" buttons. The main area is titled "test.users" and shows a table with two documents. The top document is for "Rhyme" and the bottom for "Taslu". Each document is expanded to show its fields: _id, name, email, isAdmin, verified, googleId, orders, addresses, notifications, password, birthday, gender, and phone. The bottom document also shows its exPushToken field.

Document	Fields
Rhyme	<pre>_id: ObjectId('62f4b01641818f836ddaecb3') name: "Rhyme" email: "rhyme665@gmail.com" isAdmin: true verified: true googleId: "104144629502437893468" orders: Array addresses: Array notifications: Array password: "\$2b\$10\$xcBa8y7htkF6tLqz.YzrY.bLIdbAwKfm4KdKS/6XdhB8V7QLeel7." birthday: "1999-12-16" gender: "Male" phone: "01956140407"</pre>
Taslu	<pre>_id: ObjectId('62f4d43f27545b5af5e55077') name: "Taslu" email: "taslu665@gmail.com" exPushToken: "ExponentPushToken(qA_AI-LCMPrkVxvtG0gwWh)" password: "\$2b\$10\$/KfViNHBS5S0E2D8RA21Y0Hprx3tD0riE5l80CrxCirZBMK5S/WnG" isAdmin: false</pre>

Figure 53: User Table

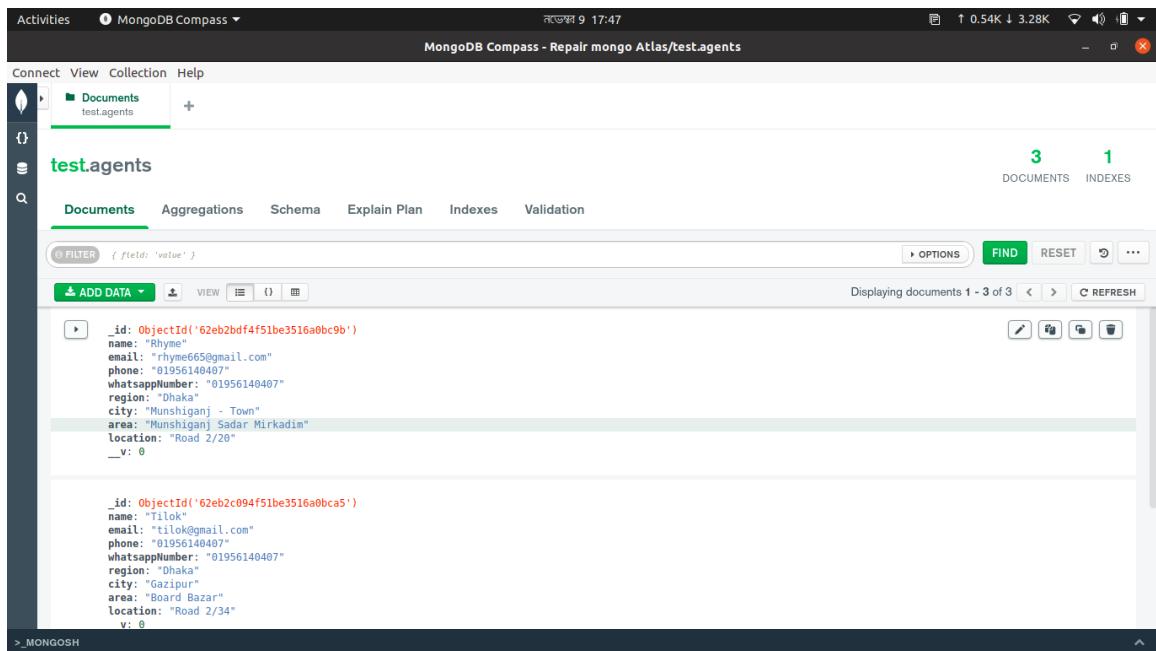


Figure 54: Agent Table

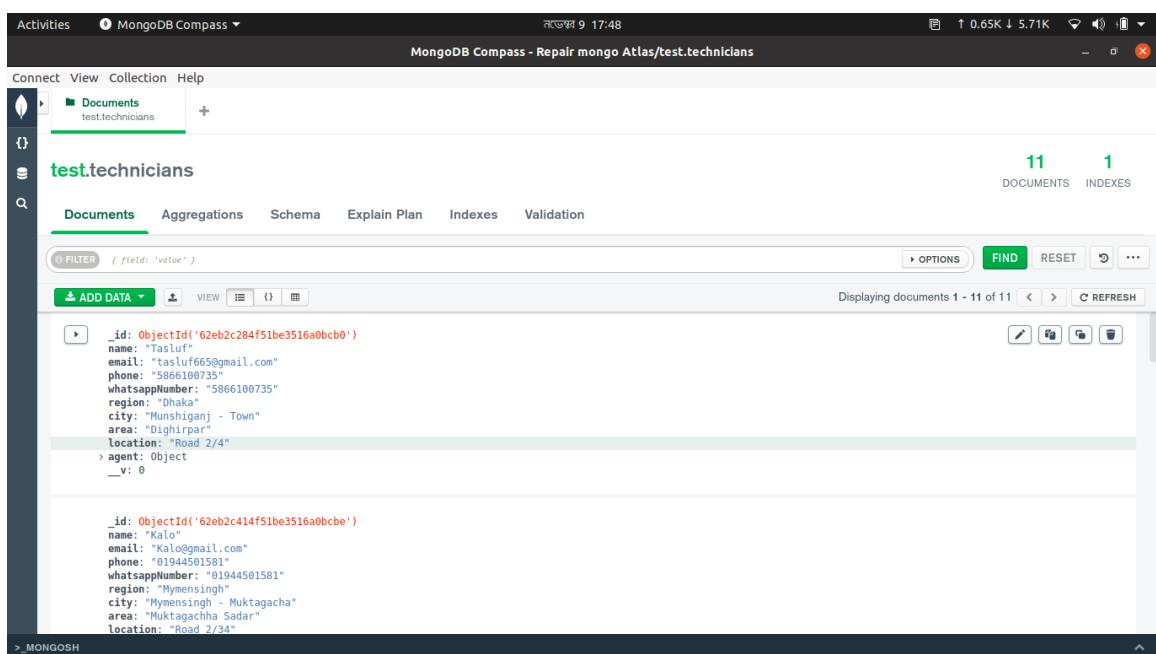


Figure 55: Technician Table

The screenshot shows the MongoDB Compass interface for the 'test.addresses' collection. The top bar displays 'MongoDB Compass - Repair mongo Atlas/test.addresses' and the date/time '২০২২/১০/১৫ ১৭:৪৮'. The left sidebar has 'Documents' selected under 'test.addresses'. The main area shows three documents with the following data:

```

{
  "_id": ObjectId("62eb2bbb2d2c3918b03d598d"),
  "id": "R88308476",
  "name": "Barishal - Mehendiganj",
  "nameLocal": "Barishal - Mehendiganj",
  "parentId": "R3921298",
  "displayName": "Barishal - Mehendiganj"
}

{
  "_id": ObjectId("62eb2bbb2d2c3918b03d598e"),
  "id": "R88308488",
  "name": "Chattogram",
  "nameLocal": "Chattogram",
  "parentId": "R184649",
  "displayName": "Chattogram"
}

{
  "_id": ObjectId("62eb2bbb2d2c3918b03d598f"),
  "id": "R88308472",
  "name": "Barishal - Sahebganj",
  "nameLocal": "Barishal - Sahebganj",
  "parentId": "R3921298"
}

```

Below the documents, the status bar shows 'Displaying documents 1 - 20 of 3357'.

Figure 56: Address Table

The screenshot shows the MongoDB Compass interface for the 'test.orders' collection. The top bar displays 'MongoDB Compass - Repair mongo Atlas/test.orders' and the date/time '২০২২/১০/১৫ ১৭:৪৮'. The left sidebar has 'Documents' selected under 'test.orders'. The main area shows eleven documents with the following data:

```

{
  "_id": ObjectId("62f4e7476504589c9758f578"),
  "name": "nssnsn",
  "phone": "029382822",
  "address": "[a][a]a",
  "arrivalDate": "2022-08-12T11:25:38.713+00:00",
  "arrivalTime": "2022-08-11T10:25:40.662+00:00",
  "category": "Repairing",
  "categoryType": "youtube-tv",
  "brand": "62f32a306572165102694963",
  "model": "62f3483bf20bcfc7faef977",
  "problem": "bsnsnsn",
  "note": "dssnsns",
  "bookingTime": "2022-08-11T11:25:59.386+00:00",
  "status": Array,
  "payment": Array,
  "userId": ObjectId("62f4d43f27545b5af5e55077"),
  "technicianId": ObjectId("62f49af56271c8181354a869"),
  "amount": 500
}

{
  "_id": ObjectId("62f5062c25ecadd351c1ea4")
}

```

Below the documents, the status bar shows 'Displaying documents 1 - 11 of 11'.

Figure 57: Orders Table

Figure 58: Products Table

4.3 Interaction Design and User Experience (UX)

I have used Figma [an online UX design application] for the initial design of our application. Then we implemented this design in our android app and admin panel. We use different React and React native libraries. And we also use Bootstrap for the design implementation. These elements represent our app well and make it affordable. By working with the user through this application, it is possible to get the satisfaction of the user in any business case. The user experience is getting better due to the process of enhancing stoner satisfaction with the app and pleasure handed in the commerce with the application.

4.4 Implementation Requirement

1. Figma for initial design
2. HTML, CSS and JavaScript
3. Bootstrap and Material UI for design implementation
4. React to make the admin panel.
5. React Native to implement the Android native application.
6. NodeJS for back-end implementation.
7. Express for Rest API
8. MongoDB for database

CHAPTER 5

Implementation and Testing

5.1 Implementation of Database

We have used MongoDB atlas for the database. It's a cloud NoSQL database. It's a secure and reliable database. First, our app will send requests to the back-end server. Then our server will authenticate the user and fetch data from the MongoDB database. Our NodeJS back-end app will make a connection with the cloud MongoDB database and whenever an authenticated user will request any data, it will send a response with that data. To make the connection we use Express and mongoose libraries for our back-end application.

5.2 Implementation of Front-end Design

For the Android Front-end part, we have used React-Native components like Text view, button, Card, View, etc. for the design part. We also use some popular libraries of React-Native. And for the Admin panel, we use React to make the component. We also use CSS, Material UI, Bootstrap, and other React libraries. We have divided the web page into smaller components and reused it every time whenever needed. Which makes our code reusable and easy to debug.

5.3 Testing Implementation

We implemented manual testing and automated testing for our application. By manual testing, users use our app and test different features of our app. We also define automate testing for several different test cases. This automated testing will automatically run every time before the application start. We have defined this automated testing for our Back-end part to secure our REST API endpoints. For the automated testing, we have used the JEST library in NodeJS.

5.4 Test Results and Reports

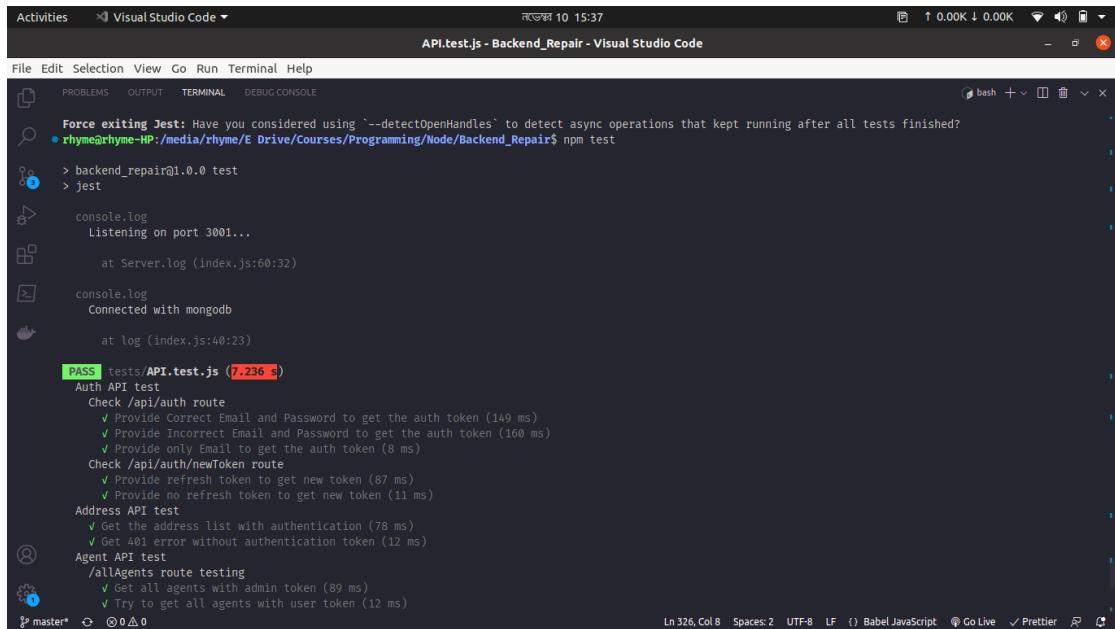
5.4.1 Automated Testing

Test Case	Test Input	Expected Outcome	Obtained Outcome	Pass/Fail	Tested On
Login	Valid Email and Password	Successfully Login	Successfully Login	Pass	10-11-22
Login	Valid Email and Invalid Password	Get 400 Error	Get 400 Error	Pass	10-11-22
Login	Valid Email and no Password	Get 400 Error	Get 400 Error	Pass	10-11-22
Refresh-Token	Provide refresh-token	Get new token	Get new token	Pass	10-11-22
Refresh-Token	Provide no refresh-token	Get 401 Error	Get 401 Error	Pass	10-11-22
Address List	Provide Token	Get address List	Get address List	Pass	10-11-22
Address List	Provide No Token	Get 401 Error	Get 401 Error	Pass	10-11-22
Agents List	Provide Admin Token	Get Agents List	Get Agents List	Pass	10-11-22
Agents List	Provide User Token	Get 403 Error	Get 403 Error	Pass	10-11-22
Agents List	Provide No Token	Get 401 Error	Get 401 Error	Pass	10-11-22

Technician List	Provide admin Token	Get Technician List	Get Technician List	Pass	10-11-22
Technician List	Provide User Token	Get 403 Error	Get 403 Error	Pass	10-11-22
Technician List	Provide No Token	Get 401 Error	Get 401 Error	Pass	10-11-22
Order List	Provide admin Token	Get Order List	Get Order List	Pass	10-11-22
Order List	Provide User Token	Get 403 Error	Get 403 Error	Pass	10-11-22
Order List	Provide No Token	Get 401 Error	Get 401 Error	Pass	10-11-22

Table 10: Automated Testing Result

Coding example of Automated Test by JEST framework:



The screenshot shows the Visual Studio Code interface with the title bar "API.test.js - Backend_Repair - Visual Studio Code". The terminal tab is active, displaying Jest test results. The output is as follows:

```
Force exiting Jest: Have you considered using '--detectOpenHandles' to detect async operations that kept running after all tests finished?
● rhyme@rhyme-HP:/media/rhyme/E Drive/Courses/Programming/Node/Backend_Repair$ npm test
  > backend_repair@1.0.0 test
  > jest

  console.log
    Listening on port 3001...

  at Server.log (index.js:60:32)

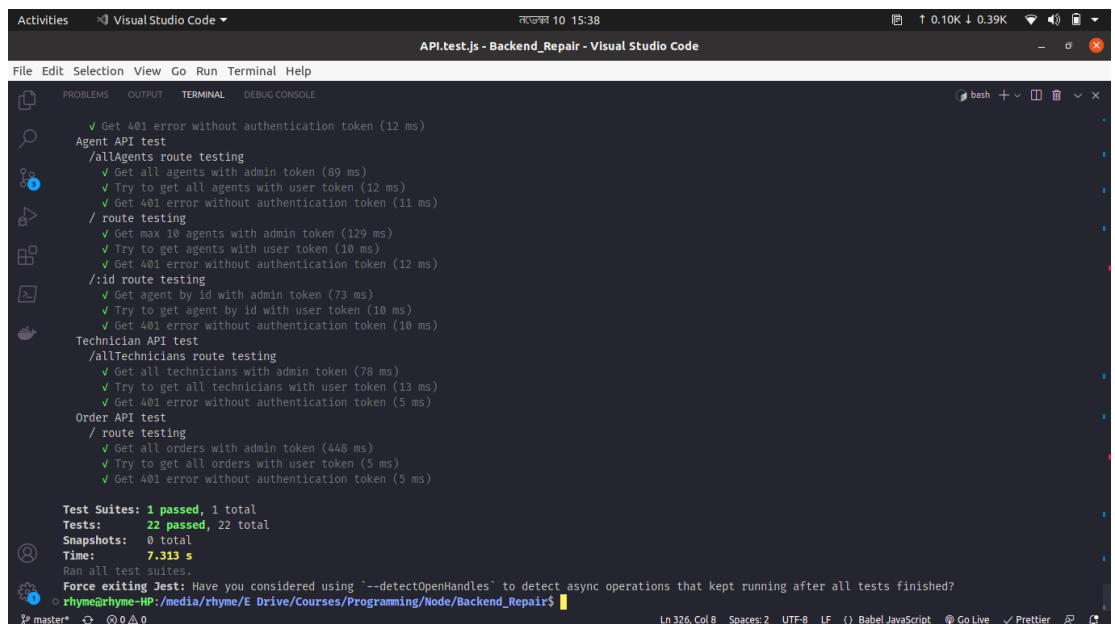
  console.log
    Connected with mongodb

  at log (index.js:40:23)

  PASS  tests/API.test.js (7.236 s)
    Auth API test
      Check /api/auth route
        ✓ Provide Correct Email and Password to get the auth token (149 ms)
        ✓ Provide Incorrect Email and Password to get the auth token (160 ms)
        ✓ Provide only Email to get the auth token (8 ms)
    Check /api/auth/newToken route
      ✓ Provide refresh token to get new token (87 ms)
      ✓ Provide no refresh token to get new token (11 ms)
    Address API test
      ✓ Get the address list with authentication (78 ms)
      ✓ Get 401 error without authentication token (12 ms)
    Agent API test
      /allAgents route testing
        ✓ Get all agents with admin token (89 ms)
        ✓ Try to get all agents with user token (12 ms)

  master* ✘ 0 0 △ 0
```

Figure 59: Automated Test figure 1



The screenshot shows the Visual Studio Code interface with the title bar "API.test.js - Backend_Repair - Visual Studio Code". The terminal tab is active, displaying Jest test results. The output is as follows:

```
Agent API test
  /allAgents route testing
    ✓ Get all agents with admin token (89 ms)
    ✓ Try to get all agents with user token (12 ms)
  / route testing
    ✓ Get max 10 agents with admin token (129 ms)
    ✓ Try to get agents with user token (10 ms)
    ✓ Get 401 error without authentication token (12 ms)
  /:id route testing
    ✓ Get agent by id with admin token (73 ms)
    ✓ Try to get agent by id with user token (10 ms)
    ✓ Get 401 error without authentication token (10 ms)

Technician API test
  /allTechnicians route testing
    ✓ Get all technicians with admin token (78 ms)
    ✓ Try to get all technicians with user token (13 ms)
    ✓ Get 401 error without authentication token (5 ms)

Order API test
  / route testing
    ✓ Get all orders with admin token (448 ms)
    ✓ Try to get all orders with user token (5 ms)
    ✓ Get 401 error without authentication token (5 ms)

Test Suites: 1 passed, 1 total
Tests:       22 passed, 22 total
Snapshots:  0 total
Time:        7.313 s
Ran all test suites.
Force exiting Jest: Have you considered using '--detectOpenHandles' to detect async operations that kept running after all tests finished?
master* ✘ 0 0 △ 0
```

Figure 60: Automated Test figure 2

CHAPTER 6

Impact on Society, Environment and Sustainability

6.1 Impact on Society

Mobile app development is one of the fastest-growing areas of modern time. Now a days, it's become an essential part of our daily lives, therefor it has a significant impact on our society. As people spend more hour on our portable device, we need to ensure that the apps we build are secure and reliable for them. Therefor the developer of this app will create a unique innovative and user friendly app. We offer a wide range of services, including servicing, booking, solidity, guaranty, customer care, cost saving facilities and much more.

We can say that "**Android Based Electronic Product Servicing System**" is a Mobile application as well web-based service system that is essential for any society. It will create an easier path for people to get an instant servicing solution. Professional expertise, strong logistics, transportation facilities could be the ultimate solution for a society. This project expect to reduce a lot of work load, by providing all this facilities for people that can create a positive impact on society.

6.2 Impact on Environment

In the 21th century while the world is developing based on industrial revolution day by day, in that very time some factor are threatening this positive growth. E-waste is considered as one of the major factor among them. E-waste refers to electronic products, that are unwanted, not working, or at the end of their life span, such as computers, televisions, VCRs, fax machines, motherboard, processor, RAM and other types of electronic products. The current global production of E-waste is estimated to be 20–25 million tons per year and the number increase quite sharply. Researcher expected it can doubled its size within next ten years. E-waste contains valuable metals as well potentially harmful materials such as lead, beryllium, cadmium, or brominated flame retardants. In our local market, whenever an AC, TV, refrigerator, washing machine or oven are serviced by local technician, they didn't follow the proper safety rules. Therefore, the harmful chemical that contain by this product, is

released into the environment which cause damage to human blood, kidneys, as well as peripheral nervous systems. Burning E-waste may generate dioxins, furans, PAHs, PHAHs, and hydrogen chloride. These chemicals are released into the air damaging the atmosphere. Most E-waste is thrown away in landfills their toxic materials seep into groundwater, affecting both land and sea animals. This can also affect the health of the people in consumption of this as food.

In this application whenever we register a vendor or work shop as a service provider, we will take all the details of their work policy in present. First we will make sure they followed all the safety rules based on human, environment and ecological balance. We will aware, teach and monitor them to prevent this threat which reduce E waste and contribute on a healthy environment.

6.3 Ethical Aspects

Ethical aspects refer to the moral philosophy that involves systematizing, defending, and recommending concept of right and wrong behavior of an application. Whenever a user installs an app or visits a website on his device it needs some excusable permission. Based on this permission, a site can get access to user personal data. By analyzing this data with the help of AI, it can determine characteristics of people most likely to engage in certain behavior. Then target the person to attract as new businesses consumers. It can compromise the confidentiality, integrity, availability, authenticity of user personal information. Some websites have security flaws which are unknown to its developer team. Attacker analyzes this vulnerability and compromise system control in order to get unauthorized access to that system as a result potential data loss.

To make sure our customer security, before collecting any data we will inform about the data collection purpose and use policies of a business or organization, in details opt-out email form. Then they have the authority to decide whether or not to interact with that business or organization. Without snooping on any other information, we will ensure stronger protection of this sensitive data and keep this data only as long as needed. We will go through “3R” of ethics as rules, responsibility and respect. We will provide a specific guidelines for professionals to perform their

roles and how to conduct themselves when situation with professional practice. To secure from external threat we will go through a proper training and testing process. We will evaluate potential risk level and risk frequency in a scale of high, medium and low as well update incident handling process on a regular bases to prevent security threat and fulfill user requirements. Sometime service center charge as much money for a simple solution and people don't have the idea what's going on. However in this apps service charge are fixed and updated in the website. That's why they are bound in any unfair charges.

6.4 Sustainability Plan

The sustainability in this app we update my brand ethos. We will implement our very own service center to support the customer as well brand value. We will add sustainable electronic product selling because now it's a prototype. We will reduce plastic packaging and use ecofriendly materials to support the eco system. Also, there will be a strict recycling policies. This app will reduce E-waste and energy waste. Most importantly we will create an offset service charge at checkout that will increase customer satisfaction and hold then with the app. Consider an ecommerce marketplace to sell we will be use merchandise business policy that will increase business opportunity and support sustainability of our application.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion

“Android based electronic product servicing system” is an application or site that provides an easier and unique business system. This smart and improving system helps to get any types of servicing effortlessly and make a stander business cycle, creating more business opportunity and local market won’t get disrupt. Moreover, it can be the best solution for vendors, service provider and customers everyone’s products by providing a necessary service based on there requirement also save time as well. It will develop the business supply chain and economy.

7.2 Scope for Further Developments

We plan to make something in future with this application.

- Add new features and business
- Electronic product buying
- Expend service Centre
- Expend in different cities
- Establish company own service Centre.
- Improve and update quality
- Fix previous bug and incident handling method
- Update UI design based on user requirement
- Improve database and payment gateway.

Reference

- [1] "Electronic Goods Repairing & Servicing in Bangladesh," [www.bdtradeinfo.com](http://www.bdtradeinfo.com/yp-data/electronic-goods-repairing-servicing).
<https://www.bdtradeinfo.com/yp-data/electronic-goods-repairing-servicing>.
- [2] T. H. Phạm, "B2B Brand Management," [www.academia.edu](http://www.academia.edu/14558996/B2B_Brand_Management), [Online]. Available:
https://www.academia.edu/14558996/B2B_Brand_Management
- [3] StatCounter, "Mobile Operating System Market Share Worldwide," *StatCounter Global Stats*, 2022.
<https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [4] "React Native Tutorial for Beginners - Build a React Native App," [www.youtube.com](https://www.youtube.com/watch?v=0-S5a0eXPoc).
<https://www.youtube.com/watch?v=0-S5a0eXPoc>.
- [5] "Consumer Electronics: Home Appliance Industry in Bangladesh," *Business Inspection BD*, Mar. 04, 2022. <https://businessinspection.com.bd/home-appliance-industry-in-bd/>
- [6] "Massive electrification sparked electrical goods industry," *The Business Standard*, Dec. 10, 2021.
<https://www.tbsnews.net/economy/industry/massive-electrification-sparked-electrical-goods-industry-341755>
- [7] R. E. Anderson and S. S. Srinivasan, "E-satisfaction and e-loyalty: A contingency framework," *Psychology and Marketing*, vol. 20, no. 2, pp. 123–138, Jan. 2003, doi: 10.1002/mar.10063.
- [8] "E-waste Management for Environmental Sustainability: an Exploratory Study," *Procedia CIRP*, vol. 98, pp. 193–198, Jan. 2021, doi: 10.1016/j.procir.2021.01.029.
- [9] R. E. & S. S. S. Anderson, " E-satisfaction and e-loyalty: A contingency framework," *Psychology & marketing*, 20(2), 123-138., 2003.
- [10] M. A. Cusumano, "The Changing Software Business: Moving from Products to Services," *Computer*, vol. 41, no. 1, pp. 20–27, Jan. 2008, doi: 10.1109/mc.2008.29.
- [11] Y. M. K. C. & F. S. Lu, Current standards landscape for smart manufacturing systems, National Institute of Standards and Technology, NISTIR, 8107, 3, 2016.
- [12] M. Mohiuddin, Overview the e-commerce in Bangladesh, IOSR Journal of Business and Management, 16(7), 01-06., 2014.