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

Co-Supervised By

Mr. MD. AZIZUL HAKIM
Sr. Lecturer
Department of CSE
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 *date*.

BOARD OF EXAMINERS

_____

(Name) **Chairman**
Designation
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

_____

(Name) **Internal Examiner**
Designation
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

_____

(Name) **External Examiner**
Designation
Department of -------
---------- University

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

Co-Supervised by:

_____

Mr. Md. Azizul Hakim
Sr. Lecturer
Department of CSE
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 & Management System*" 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 ecommerce platform and others are using the MultiVender 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 intendant 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.

CHAPTER 1
Introduction

1.0 Introduction

The Project illustrate that "**Android Based Electronic Product Servicing System**" is a virtual store on internet confide on aggregation model, where user can hire technician based on their Home appliance. Here the term "Home appliance" refers to electronic products, devices, or equipment used in various household purposes, such as TVs, refrigerators, ACs, or washing machines. 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 backend, it will use Nodejs and for the database, developer want to use MongoDB.

1.1 Project Overview

"**Android Based Electronic Product Servicing System**" is an optimal servicing solution for Home appliance commodity. From the site customer can hire technician and experts based on their product requirement in order to fix the item.

1.2 Requirement Analysis

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 necessaries. 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.

1.3 Ami of the Project

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. Which is generating a lot of controversy while many people are speaking strongly against this issue. 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.

1.4 Project Methodology

This project intendant to develop an android app and a web application for this service.

- For mobile application: We have use React native. It's a JavaScript framework. We know it's very difficult to make an android and a ios app. We will need two team to manage our app. It's 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 backend: 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 backend 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.

1.5 Proposed Solution

- 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. 6 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.

1.7 Conclusion

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.

1.8 Project Timeline

Task Name	Start	End	Status	Sep-Oct	Nov-Dec	Jan-Feb	Mar-Apr	May-Jum	July-Aug	Sep-Oct
Project Proposal	17-09-21	29-09-21	Complete							
Requirement collect & Analysis	02-10-21	17-12-21	Complete							
System Design	24-12-21	08-02-21	Complete							
Coding	09-01-21	24-05-22	Complete							
Testing	12-06-22	23-08-22	Complete							
Documentation & Report	09-08-22	05-09-22	Complete							

CHAPTER 2
REQUIREMENTS SPECIFICATION

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

Business Process Model for our system are given below

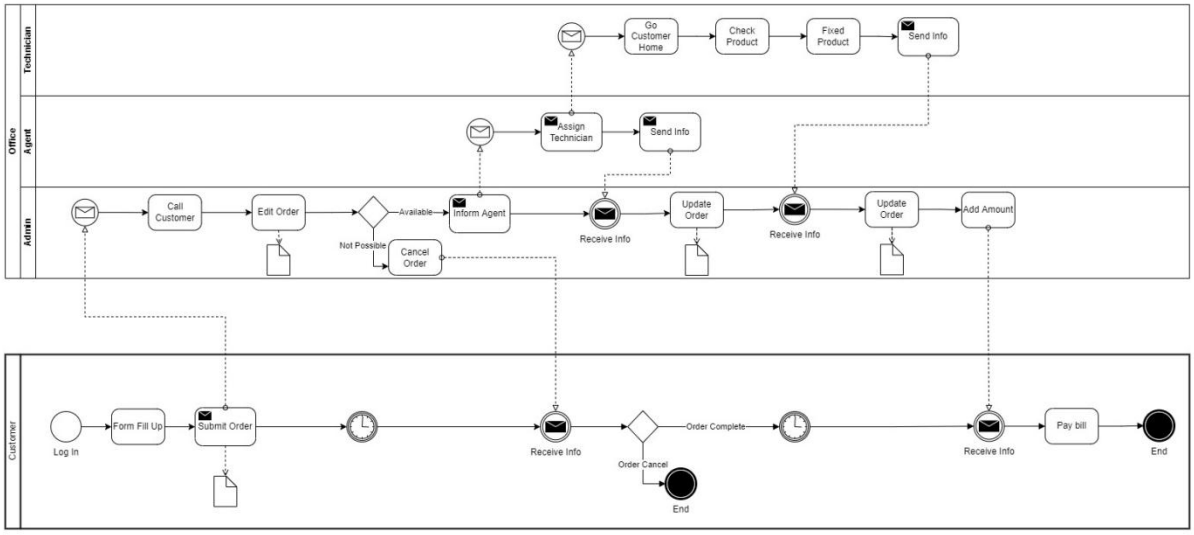

Figure 3.1.1: Business process model

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

Admi controls the database and keep 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 those who have contract with the system. He can add, and update any agent's information.

Manage Technician

Admin can view all the technicians those 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 then admin can view that order. He can accept that order, assign technician for that order and finally confirm the order by updating it.

Logout

Admin can logout form the system.

3.2.1.2 User requirements

User Login:

Description of the feature

Using this feature user can login the system. A user should login with his/her user name and password to the system after registration. Invalid user name or password is not allowed to enter the system. A user can also login into the system by a google account.

Functional requirement

- 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 implement to login 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

- System should able to verify and validate information.
- Password should be encrypted to provide security.

View and update own details:

Description of the feature

Customer can view/update his personal information. Customer can also set default address.

Choose a service:

Description of the feature

Customer can view all the service that are provided by the system at that time. Customer can also see the which brands are available and their model's number for a product to repair.

Repair request:

Description of the feature

Customer can select any services, then fill up a form including with the details of the product and select the cross-ponding 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

Customer can see all his previous service request and its details.

Notification:

Description of the feature

Customer will get notification on every step of the order.

Logout:

Description of the feature

Customer can logout from the system.

3.2.2 Non-Functional Requirements

3.2.2.1 Performance requirements

The system must accommodate 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 user friendly environment and easy to use.

3.2.2.3 Security

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

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

Administrator can only perform administrative task on pages they are privileged to access. Customer will not be allowed to access the administrator pages.

API should be access only by authorized users.

3.2.2.4 Error-handling

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 the order forms, service descriptions, price and website directly.

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

System should provide a reliable environment to customers and owner. All orders must be reached at admin without any error.

3.3 Use Case Diagram

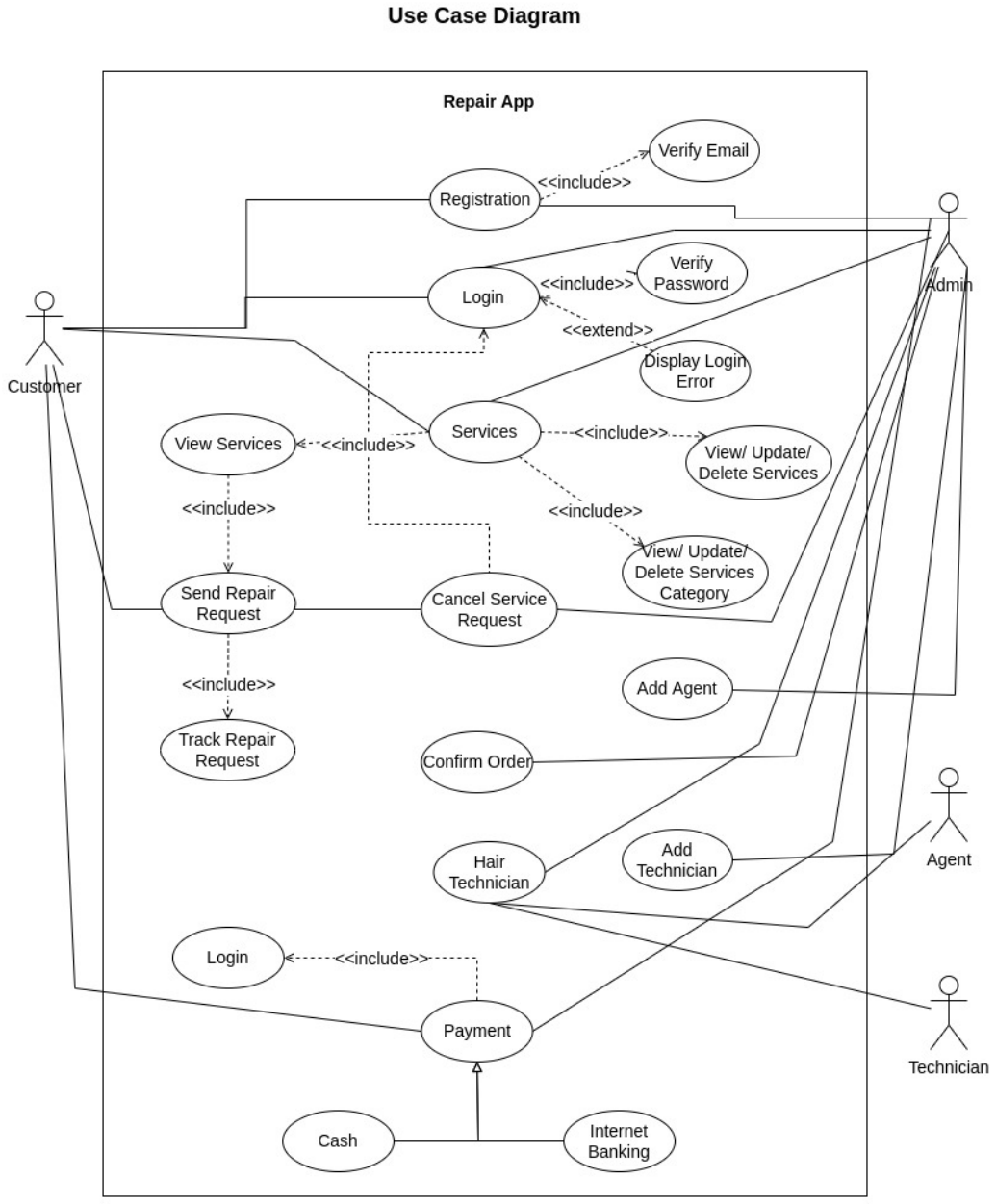

Figure 01: Use Case Diagram

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 email to verify their email address. They have to click the link provided in the email. After successfully verify their registration users can now login into the system.
Actor	o Admin o Customer
Pre-Condition	All the information Field must be filled up.
Post-Condition	Users can login into the system.
Flow	1. Fill up the registration form with necessary information. 2. Press Sign Up button. 3. System will verify the given information 4. System will send a verification email to the user's email address 5. User will click the link that is provided in the email 6. System will verify the user account 7. User can now login into the system
Include	Verified email address.

Table 02: 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 login into the system into two ways. By Email and password and by Google login. To login by email and password user have to verify their email address first. By providing the correct email and password or by 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	o Admin o Customer
Pre-Condition	All the information Field must be filled up.
Post Condition	Get access to the system.
Flow	1. Go to Login page a. Enter email address and password. b. Google login 2. Press Login button. 3. System will verify the account. 4. System will send JWT token for authorization and redirect to the home page 5. Otherwise display Login error.
Include	Verify password.

Table 03: Use Case Diagram (Login)

Use Case ID	UC3
Name	Service.
Description	In this use case admin can add, delete and update any service. They can also add, delete and update any product brand and product model. Customer 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 then select the product brand and a product model. After submitting the repair request admin will take farther action.
Actor	o Admin o Customer
Post Condition	Hire technician based on their product requirement.
Flow	1. Go to service section. 2. Select product brand and product model 3. Fill up the request form 4. Send service request 5. Track the repair request 6. Admin will receive this repair request 7. Admin can add, delete or update any service 8. Admin can add, delete or update product brand and product model
Include	o View Service o View, Update or Delete Services o View, Update or Delete Services brand and service model

Table 04: Use Case Diagram (Service)

Use Case ID	UC4

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

Table 05: Use Case Diagram (Cancel Request)

Use Case ID	UC5
Name	Confirm Order.
Description	Admin check all the necessaries and confirm the Order.
Actor	o Admin
Post Condition	Administrator agent allow Customer for the asking service.
Flow	o View request details from system database. o Verify request. o Confirm request.
Include	None

Table 06: 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. System will notify this state by a notification to the user app.
Actor	o Admin o Agent o Technician
Post Condition	Technician will arrive for the service.
Flow	o Check confirmation. o Select an agent o Select a technician close to the location o Notify the current stage by a notification to user
Include	None

Table 07: Use Case Diagram (Hair Technician)

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

Table 08: Use Case Diagram (Payment)

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

Table 09: Use Case Diagram (Add Agent)

Use Case ID	UC9
Name	Add Technician.
Description	This use case allows admin to add, view and update a technician. To add a technician, they have to fill up a form with necessary information and then click the add button to save the technician.
Actor	o Admin
Pre-Condition	Fill up the form with necessary information

Flow	o Click the add button o Fill up the form o 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 system database as well manage customers, orders, bookings and payments in the system.

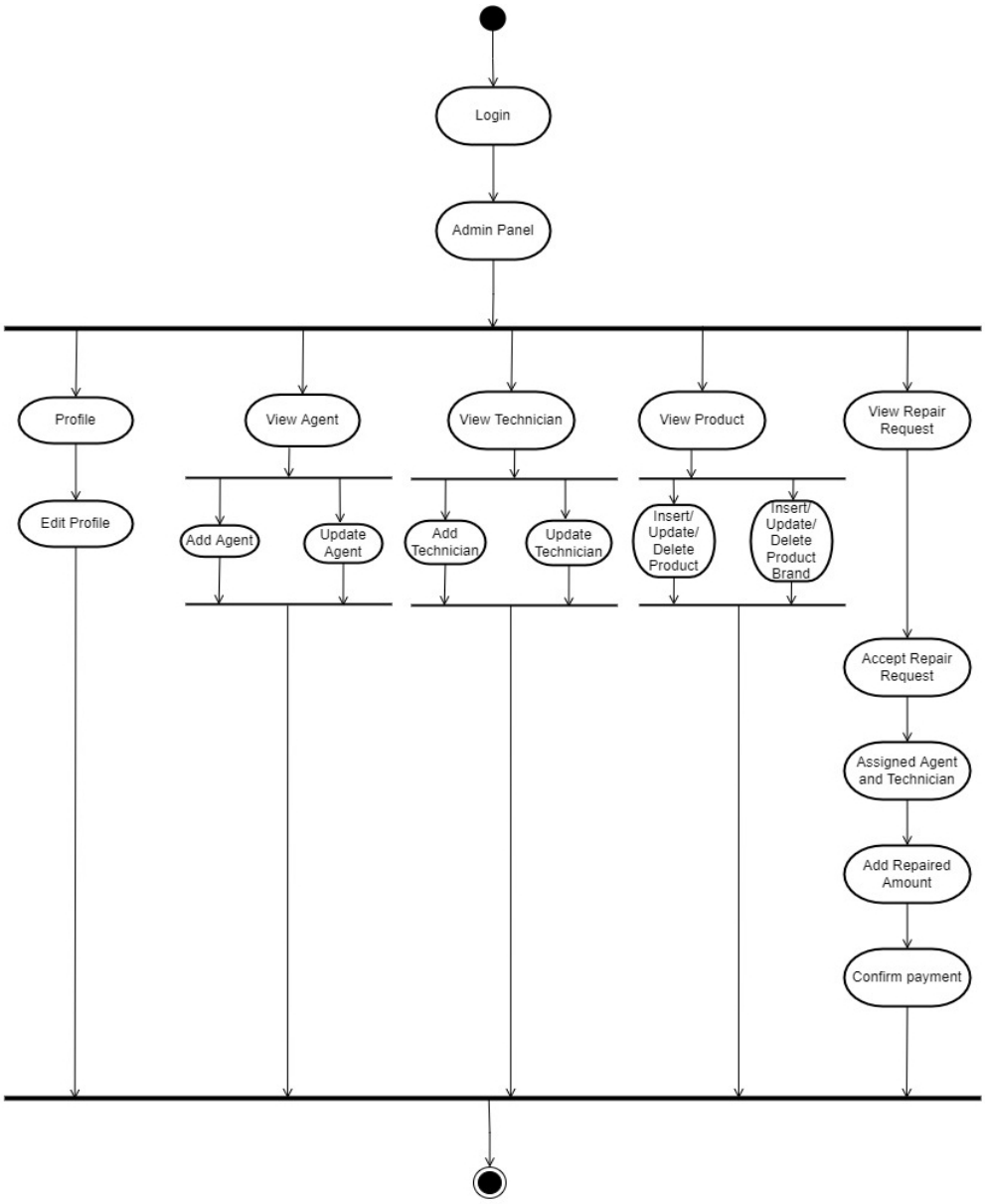

Figure 02: Admin Activity Diagram

3.4.2 Customer Activity Diagram

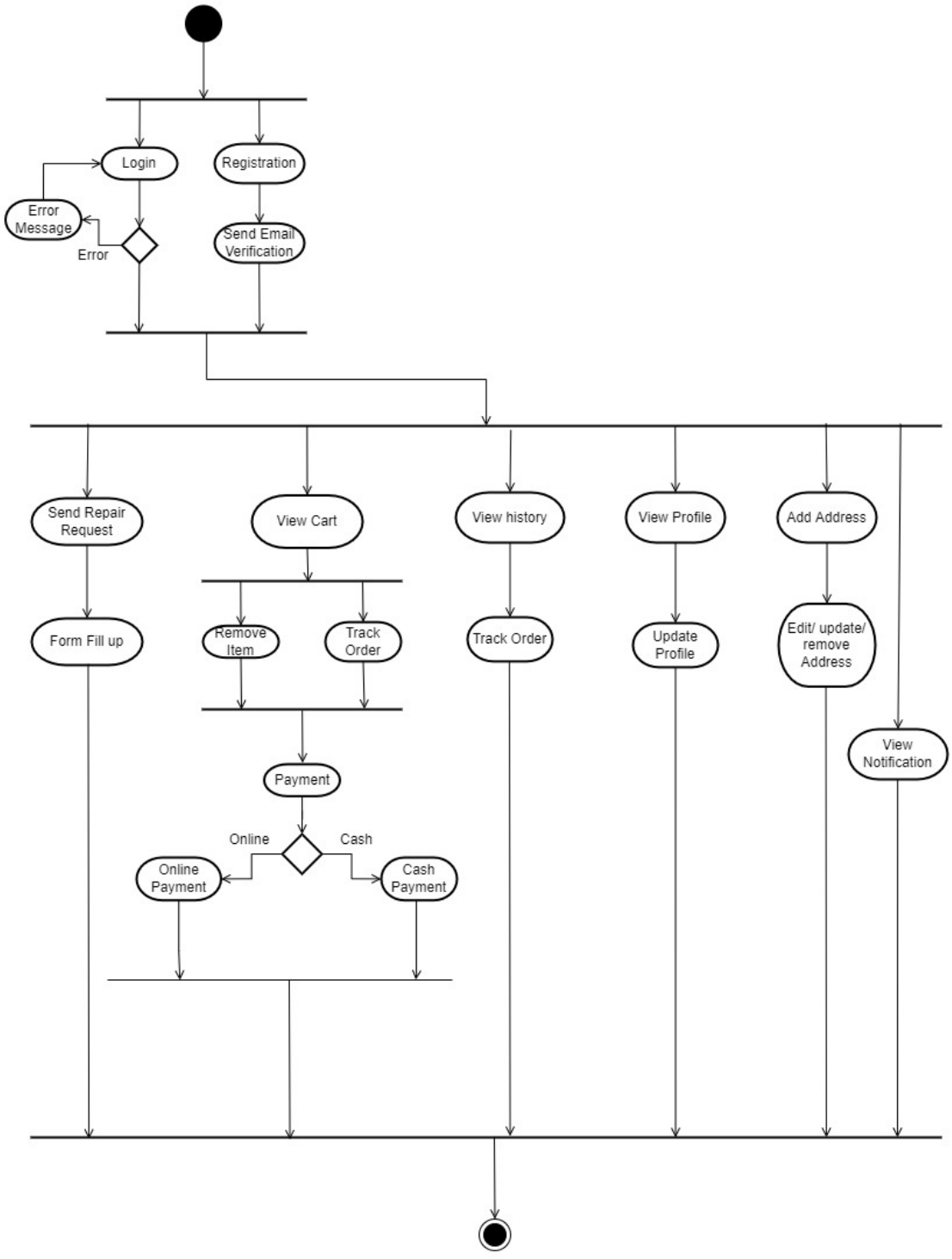

Figure 03: Customer Activity Diagram

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

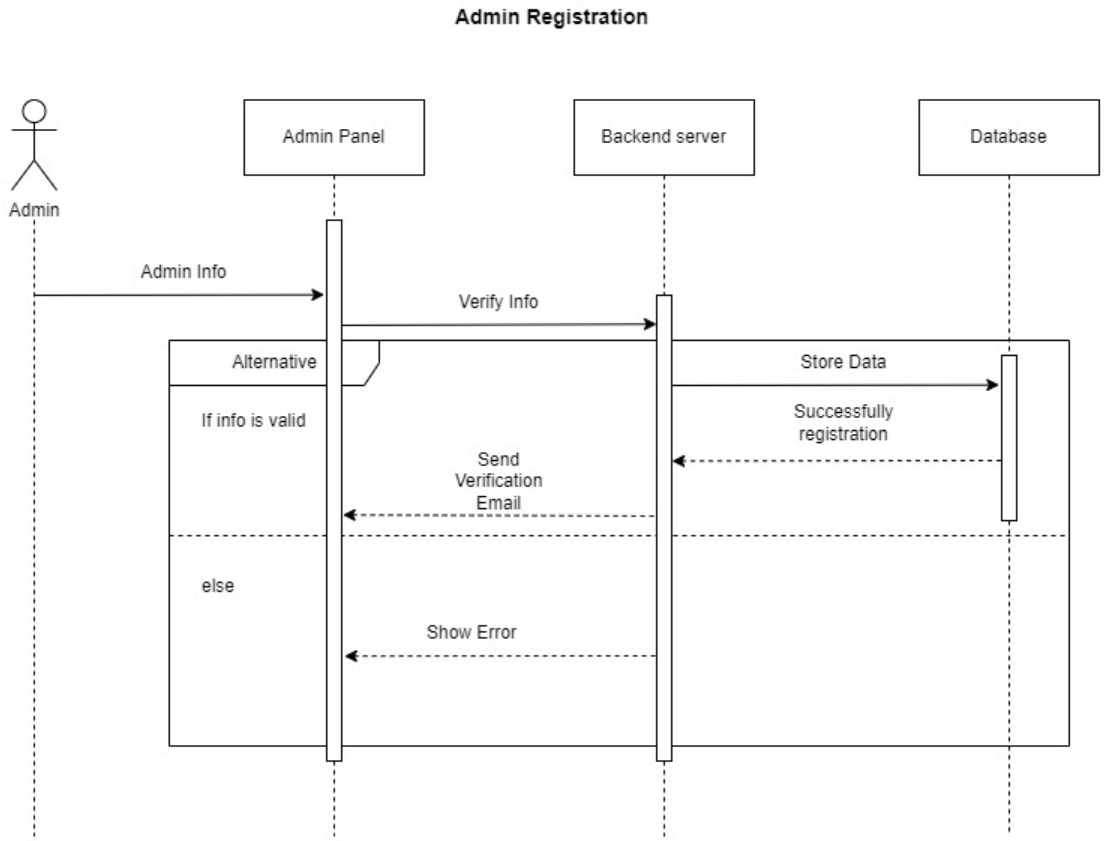

Figure 04: Admin Registration Sequence Diagram

3.5.2 Admin Login

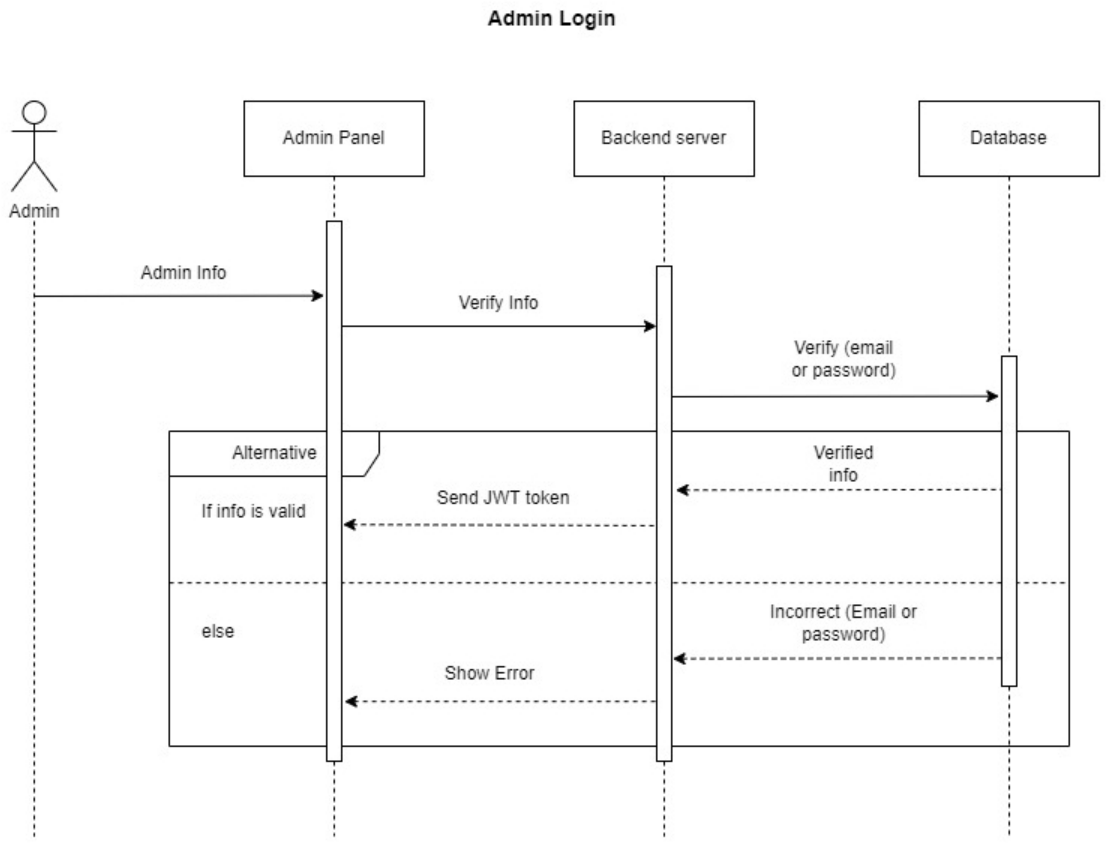

Figure 05: Admin Login Sequence Diagram

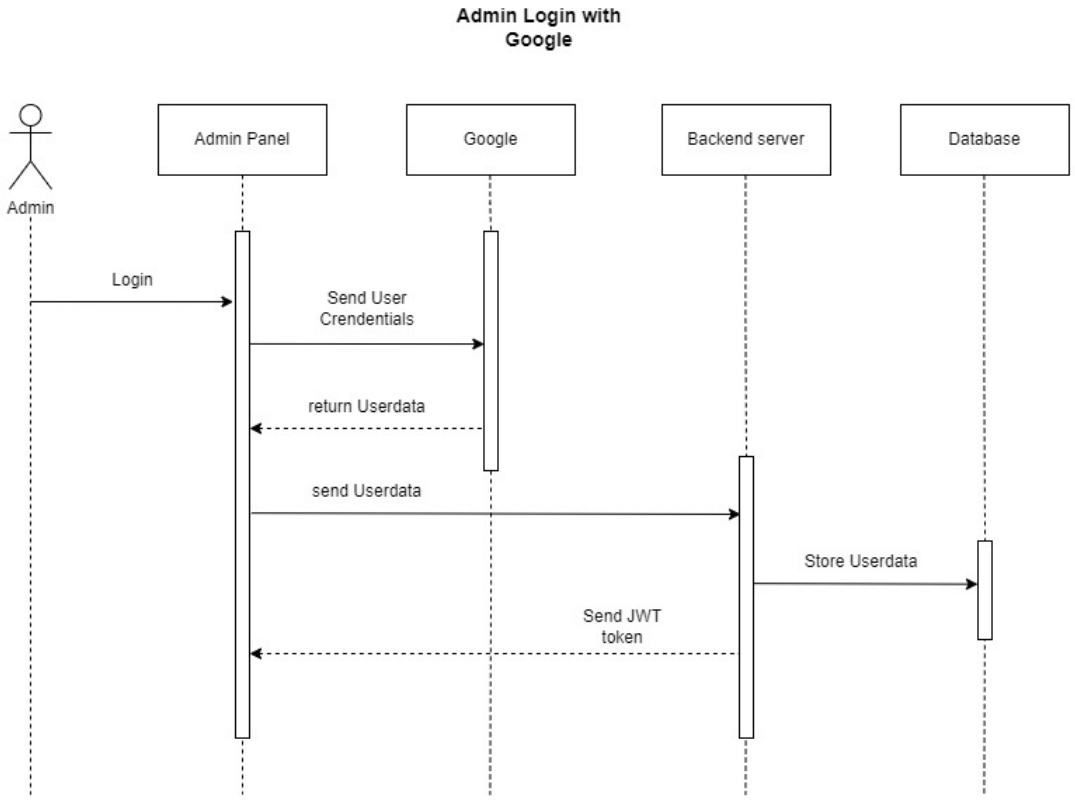

Figure 06: Admin Login with Google Sequence Diagram

3.5.3 Admin Profile

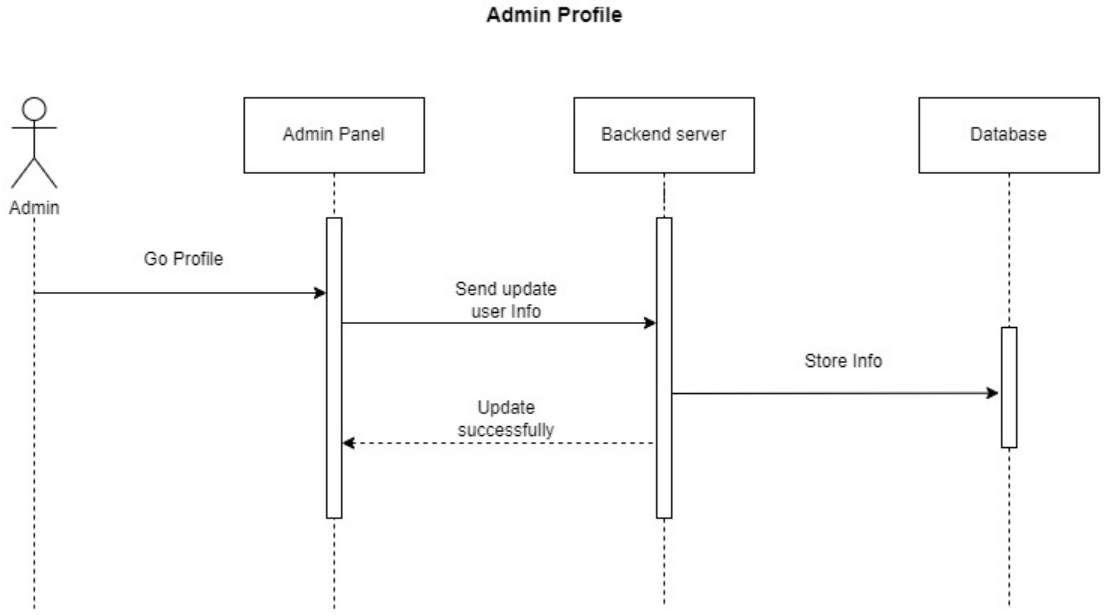

Figure 07: Admin Profile Sequence Diagram

3.5.4 Admin Forget Password

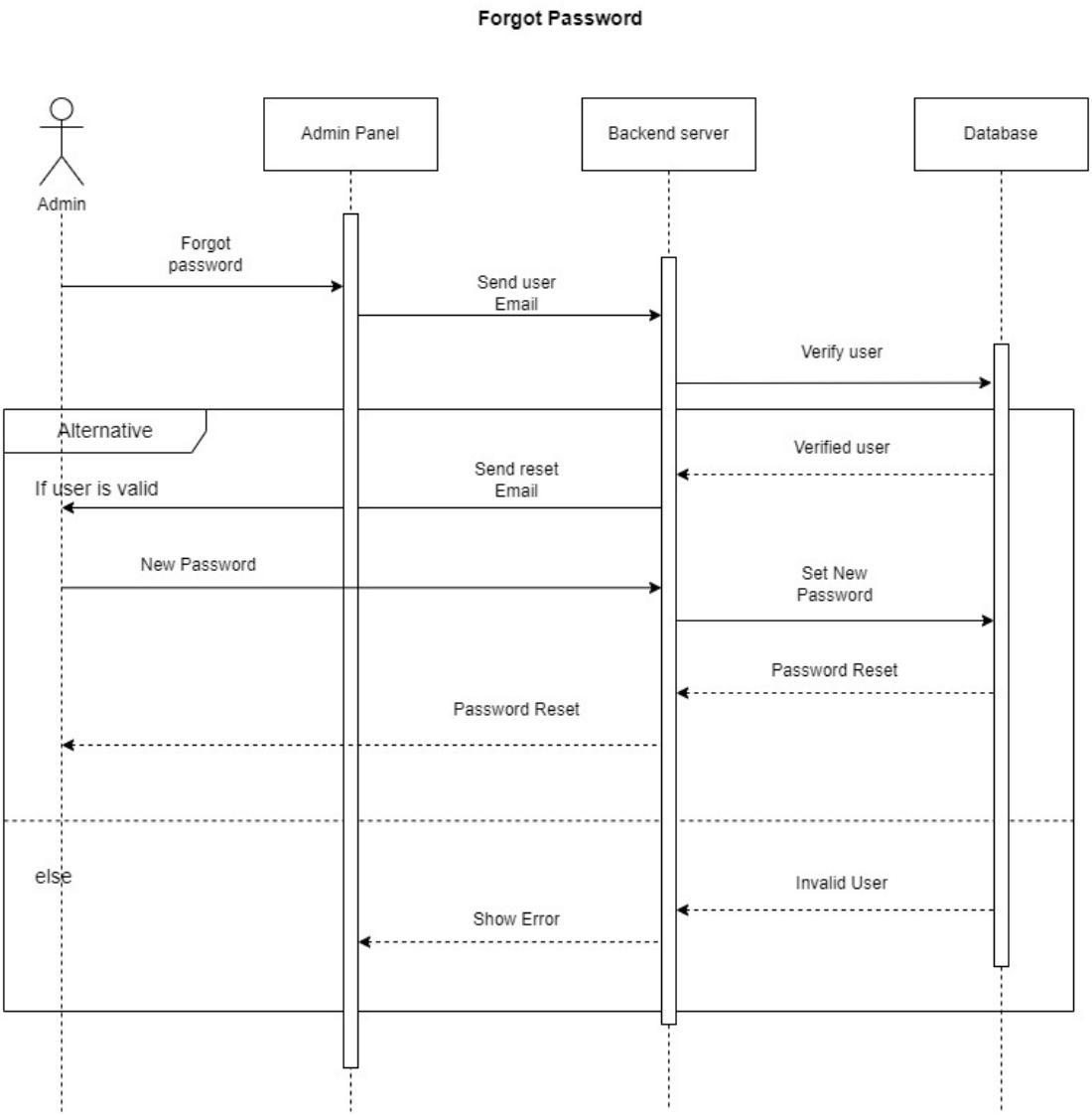

Figure 08: Admin Forget Password Sequence Diagram

3.5.5 Admin View Order

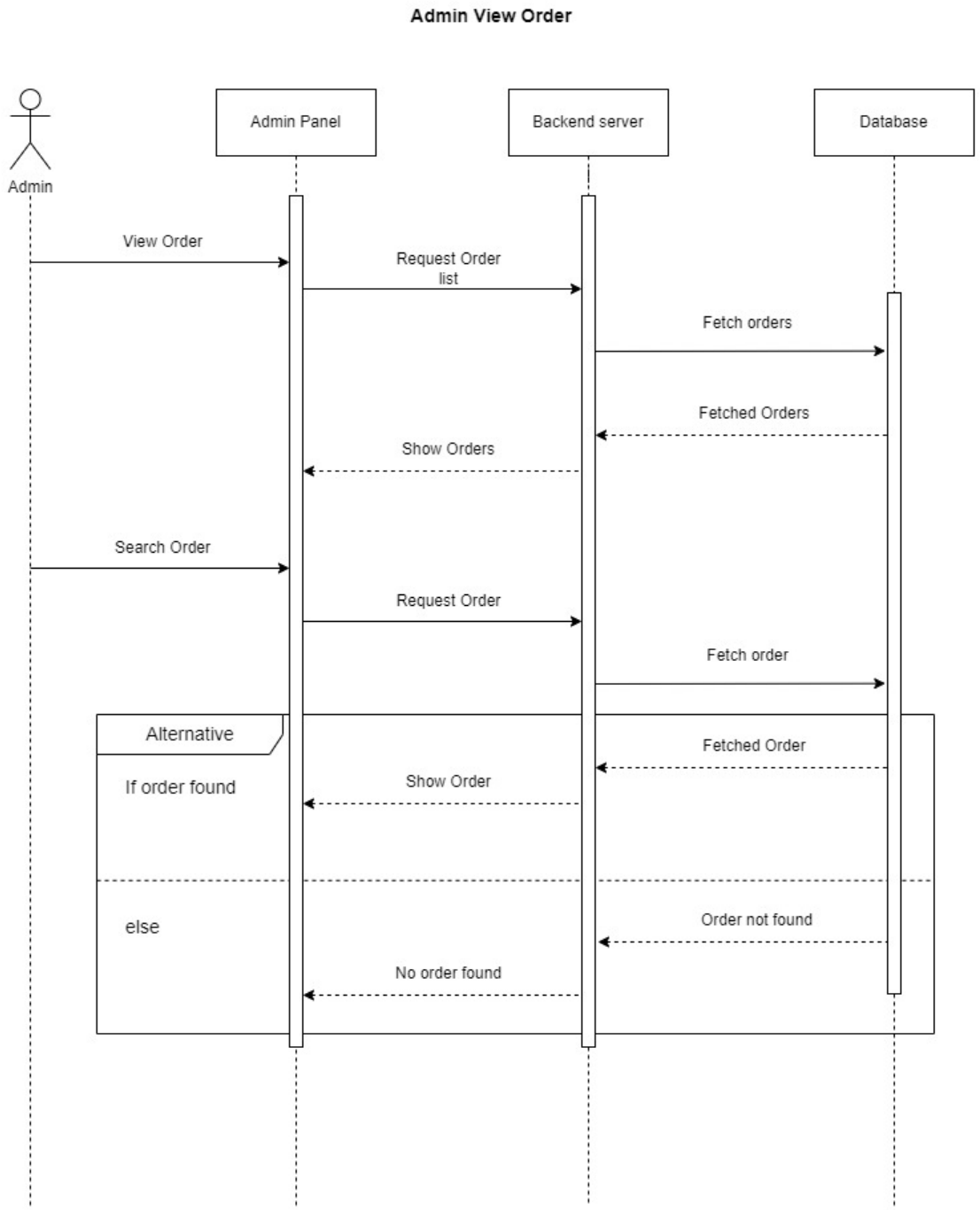

Figure 09: Admin View Order Sequence Diagram

3.5.6 Admin Add Agent

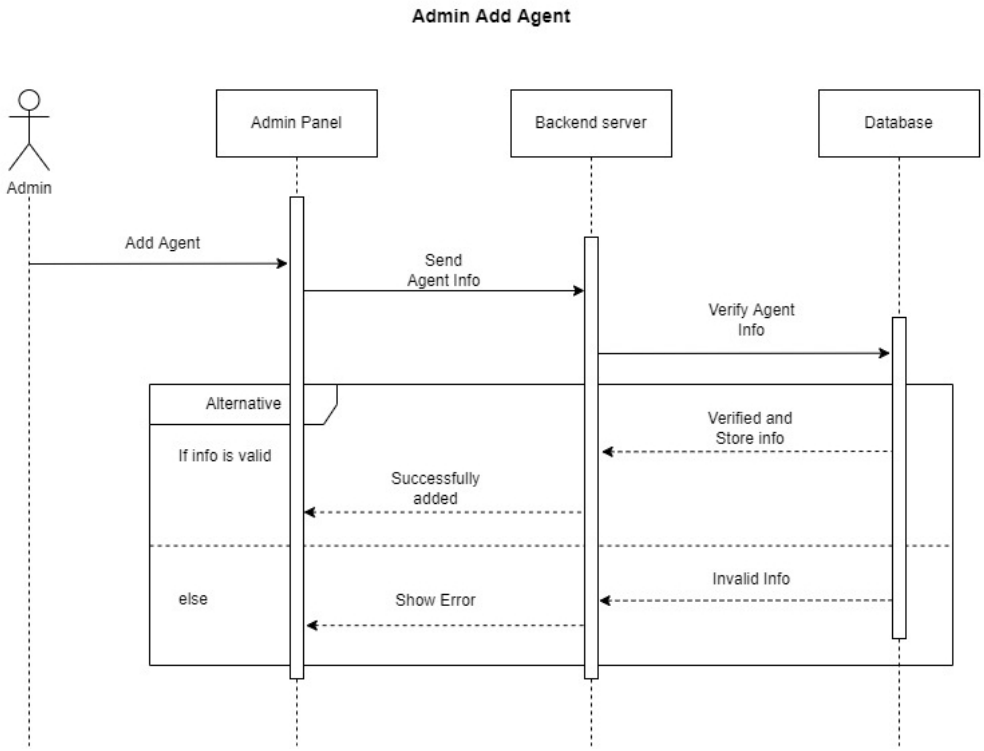

Figure 10: Admin Add Agent Sequence Diagram

3.5.7 Admin Add Technician

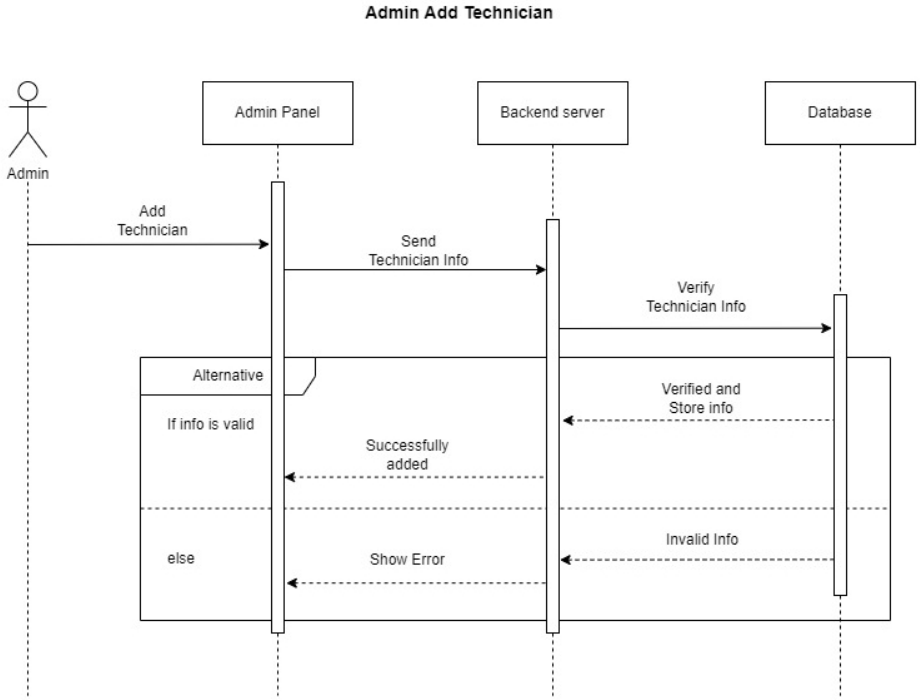

Figure 11: Admin Add Technician Sequence Diagram

3.5.8 Admin Update Order

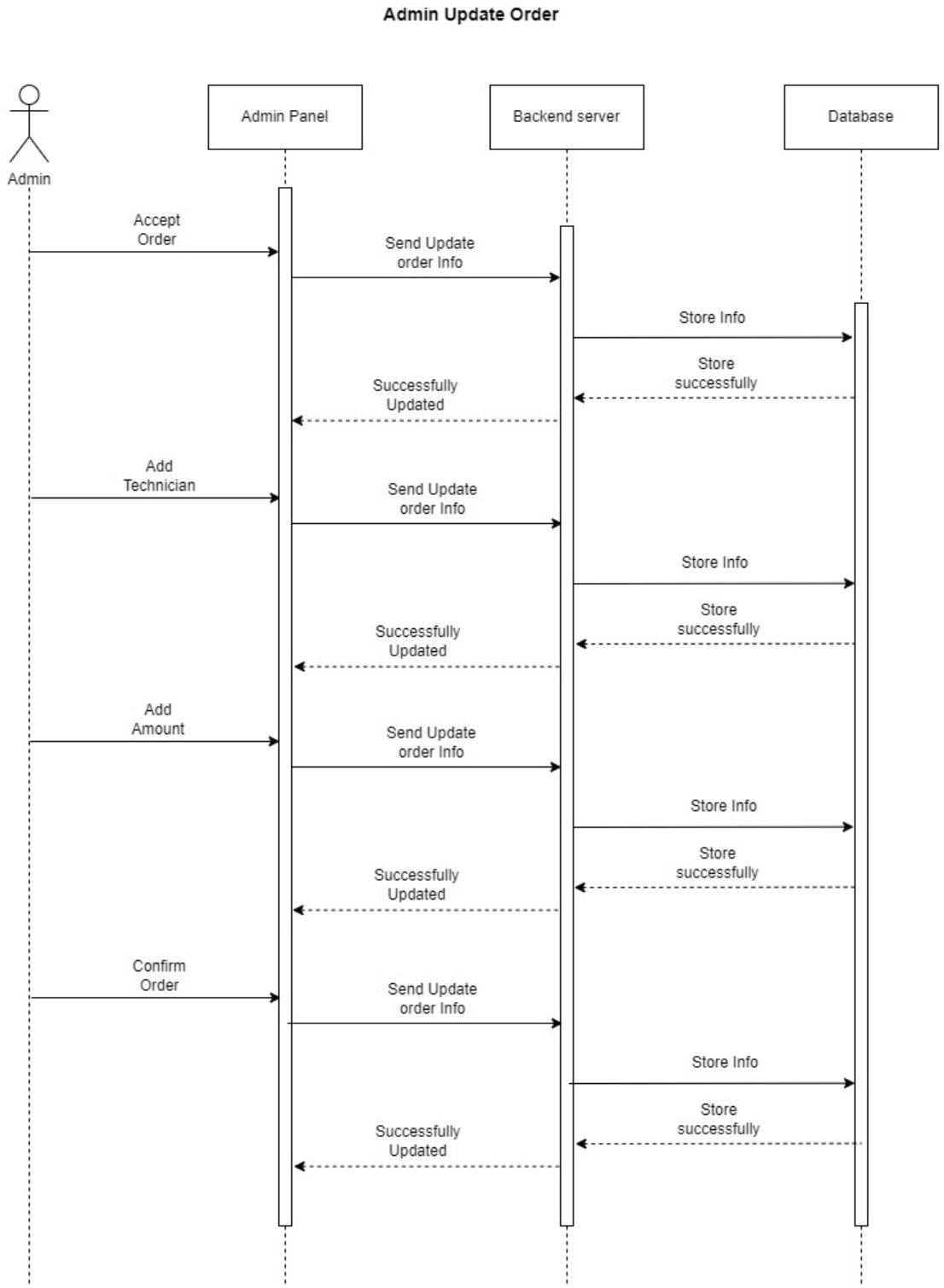

Figure 12: Admin Update Order Sequence Diagram

3.5.9 Admin Update Agent

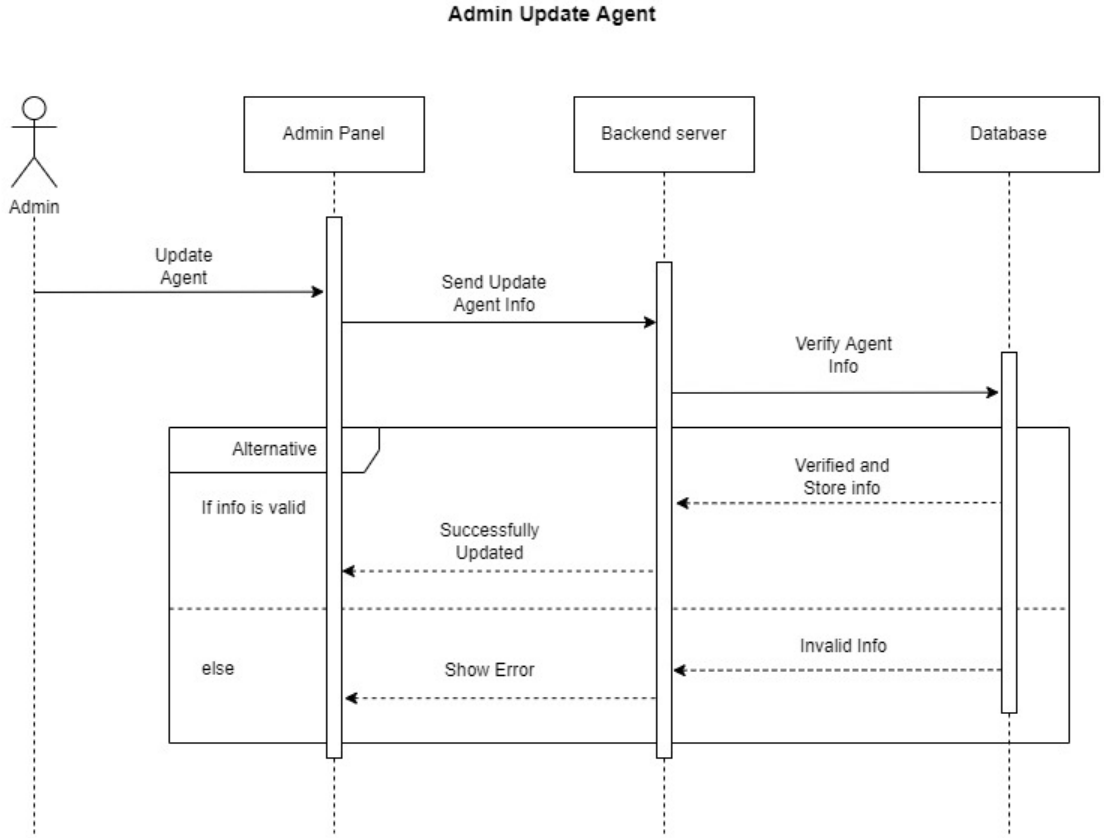

Figure 13: Admin Update Agent Sequence Diagram

3.5.10 Admin Update Technician

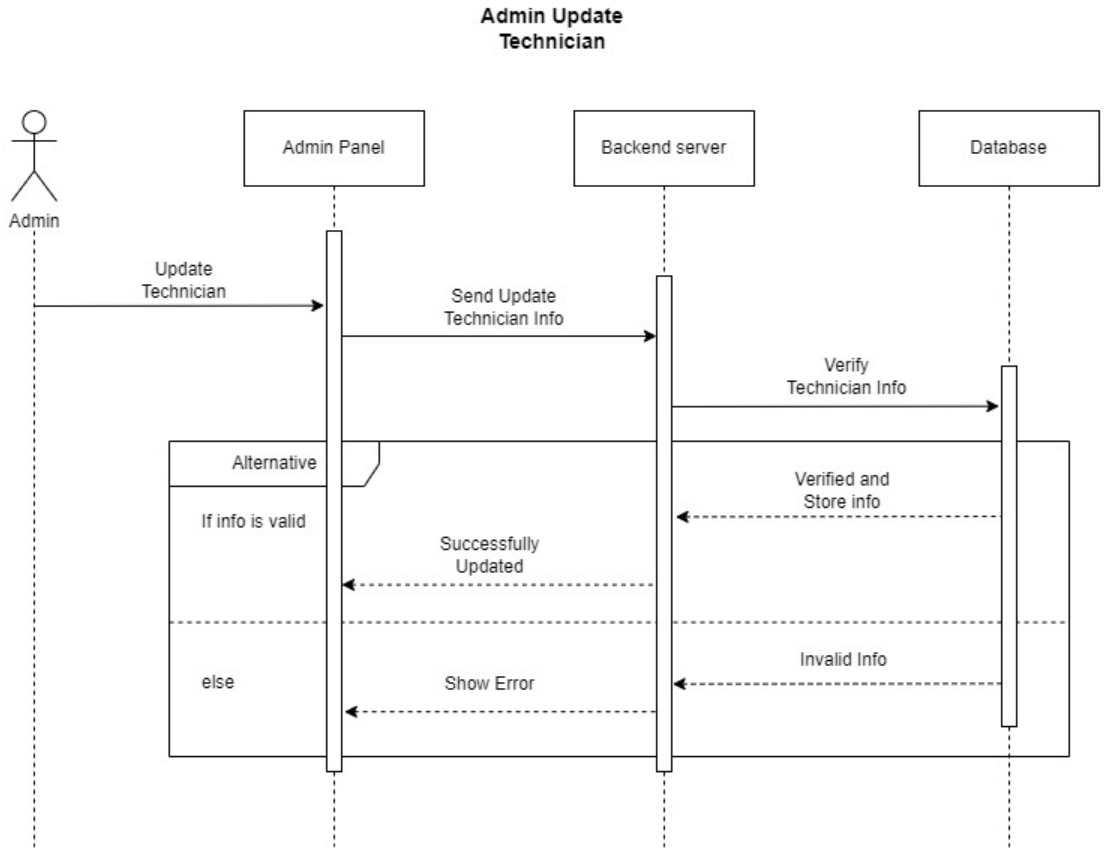

Figure 14: Admin Update Technician Sequence Diagram

3.5.11 Customer Registration

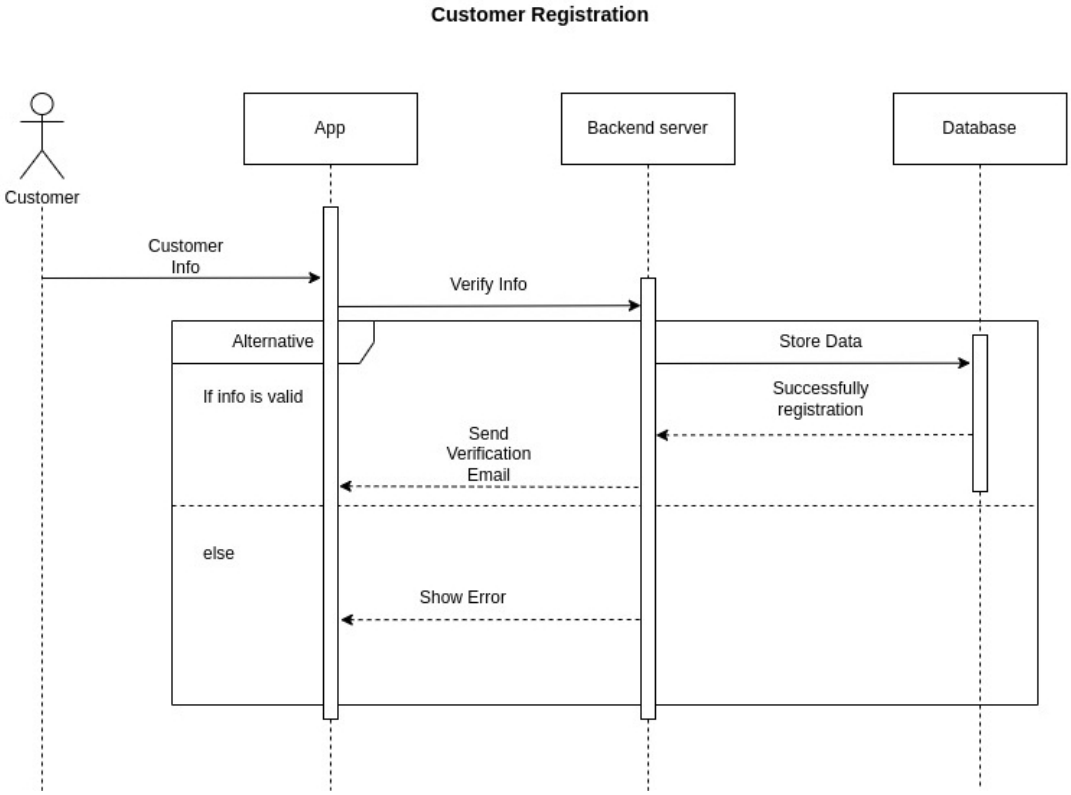

Figure 15: Customer Registration Sequence Diagram

3.5.12 Customer Login

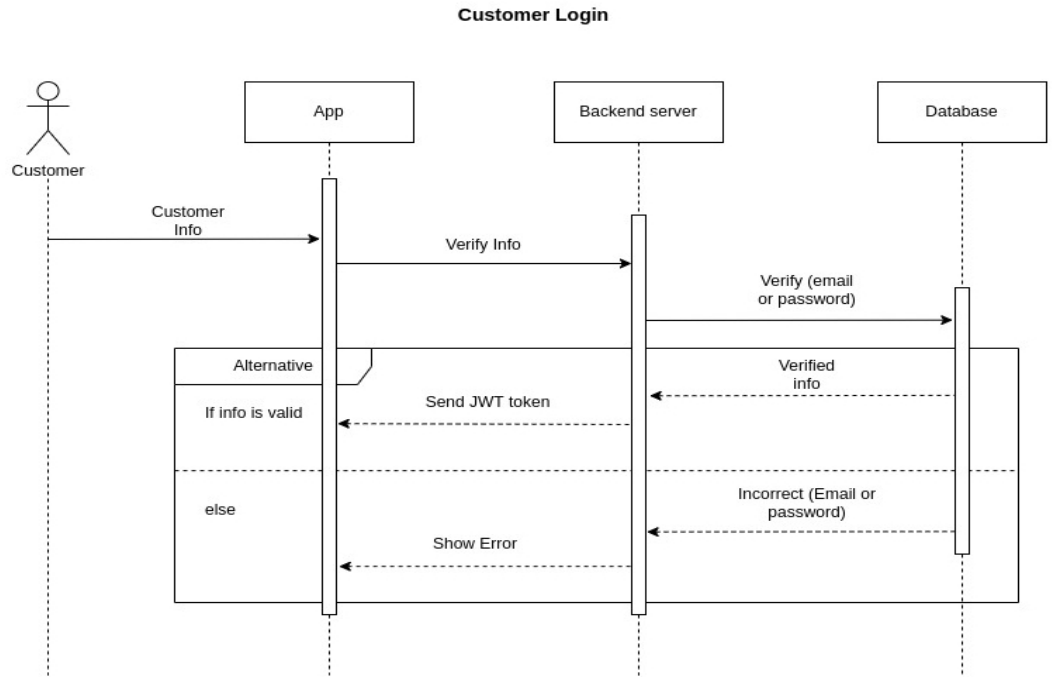

Figure 16: Customer Login Sequence Diagram

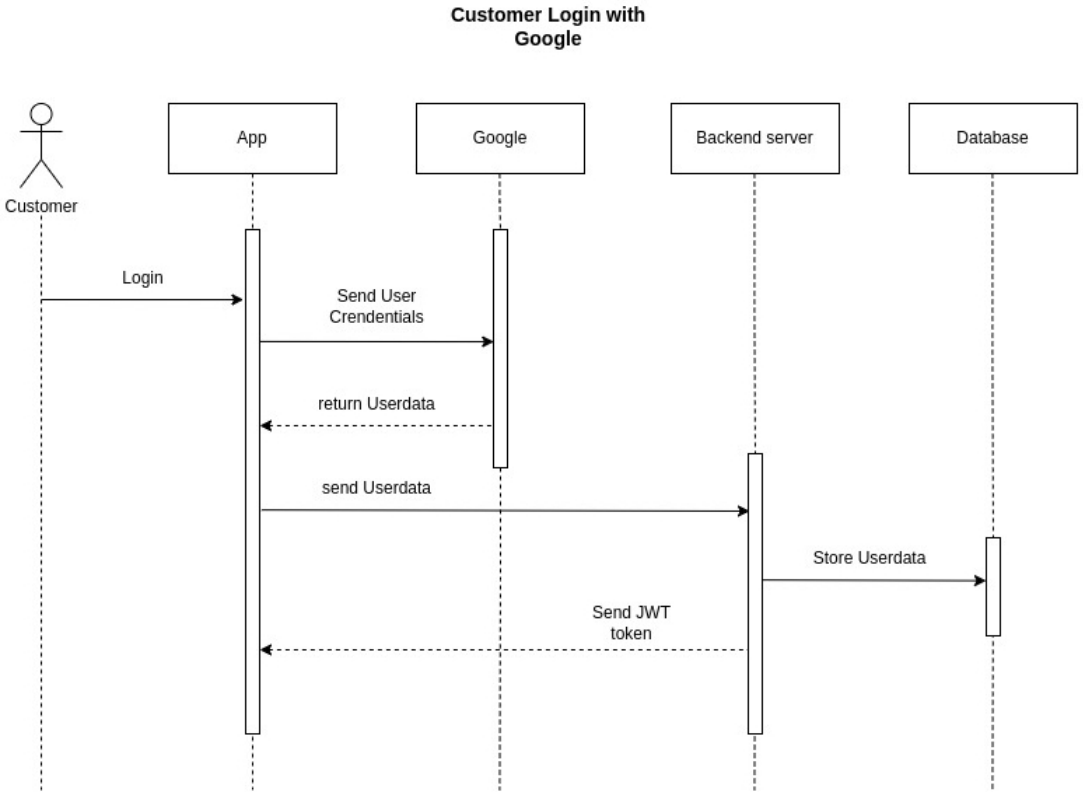

Figure 17: Customer Login with Google Sequence Diagram

3.5.13 Customer Profile

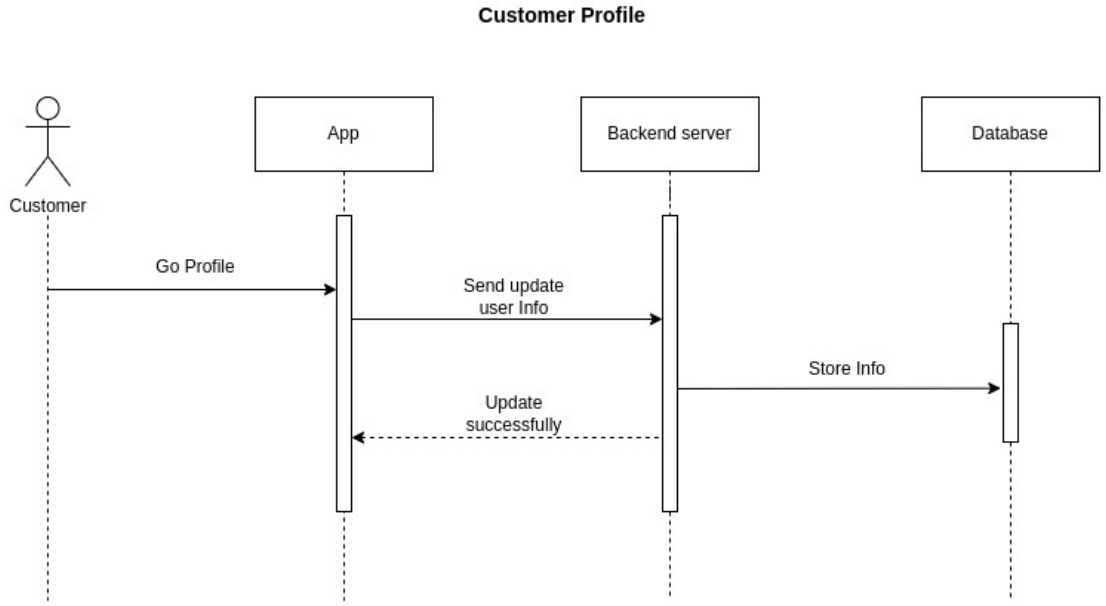

Figure 18: Customer Profile Sequence Diagram

3.5.14 Forget Password

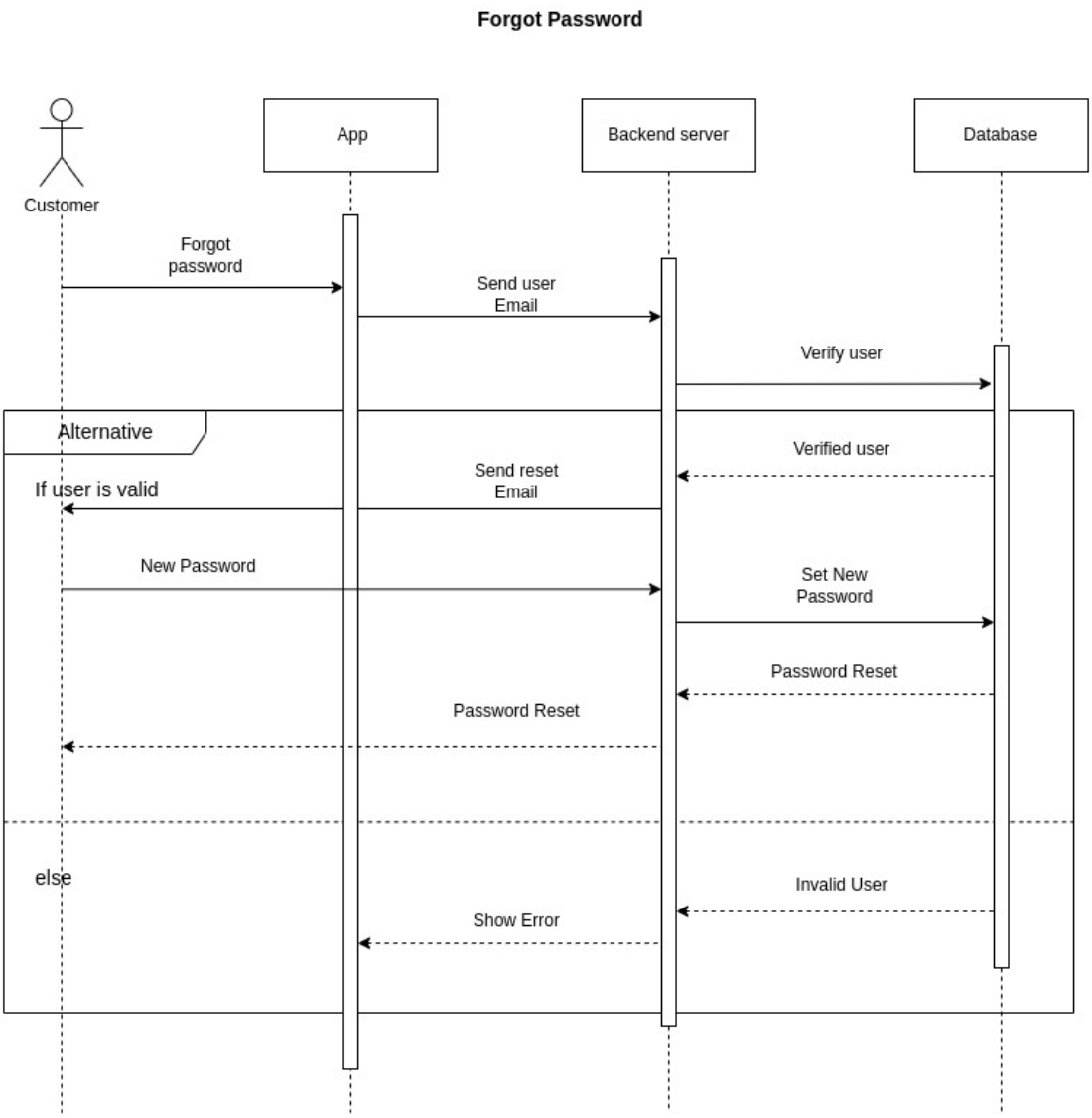

Figure 19: Customer Forget Password Sequence Diagram

3.5.15 Customer Add Order

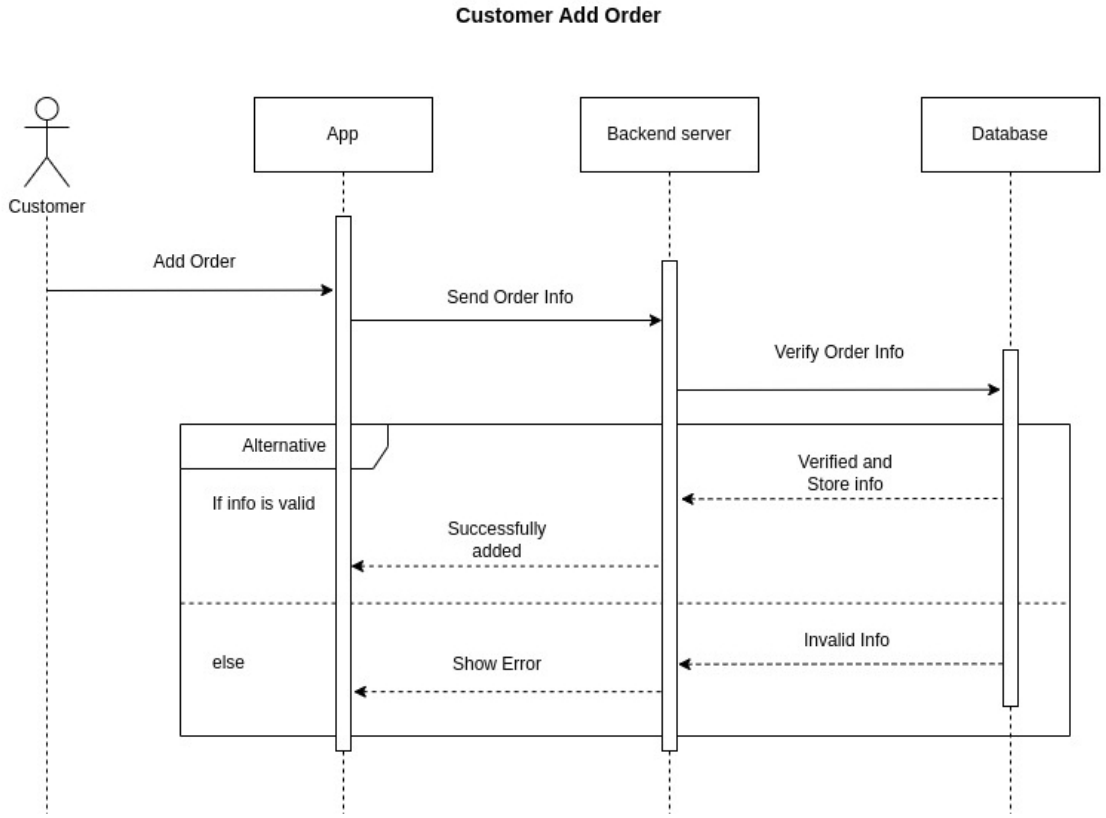

Figure 20: Customer Add Order Sequence Diagram

3.5.16 Customer Add Address

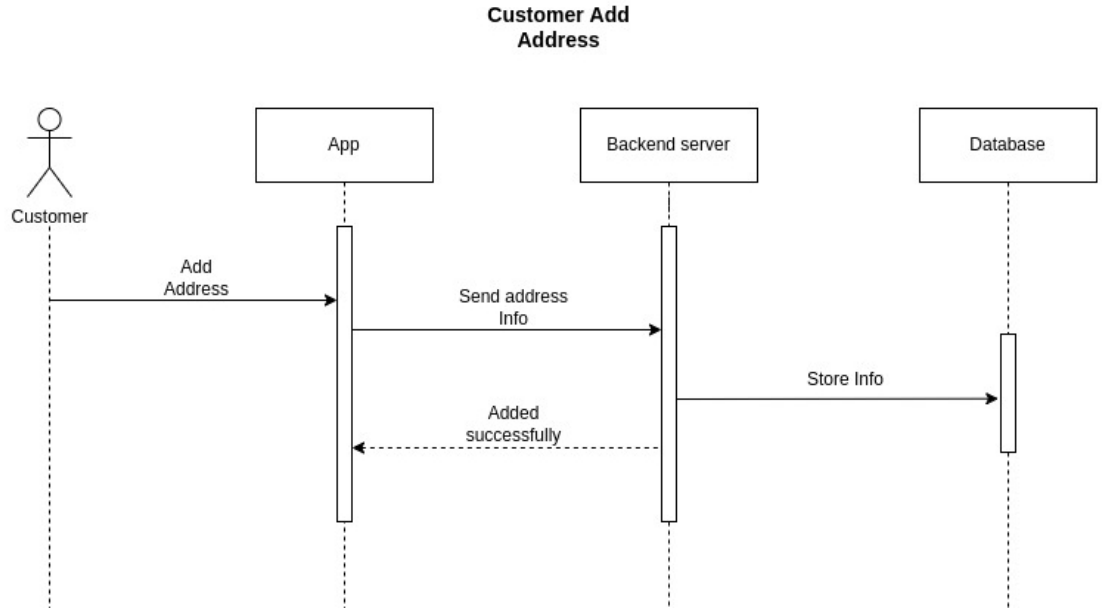

Figure 21: Customer Add Order Sequence Diagram

3.5.17 Customer Update Address

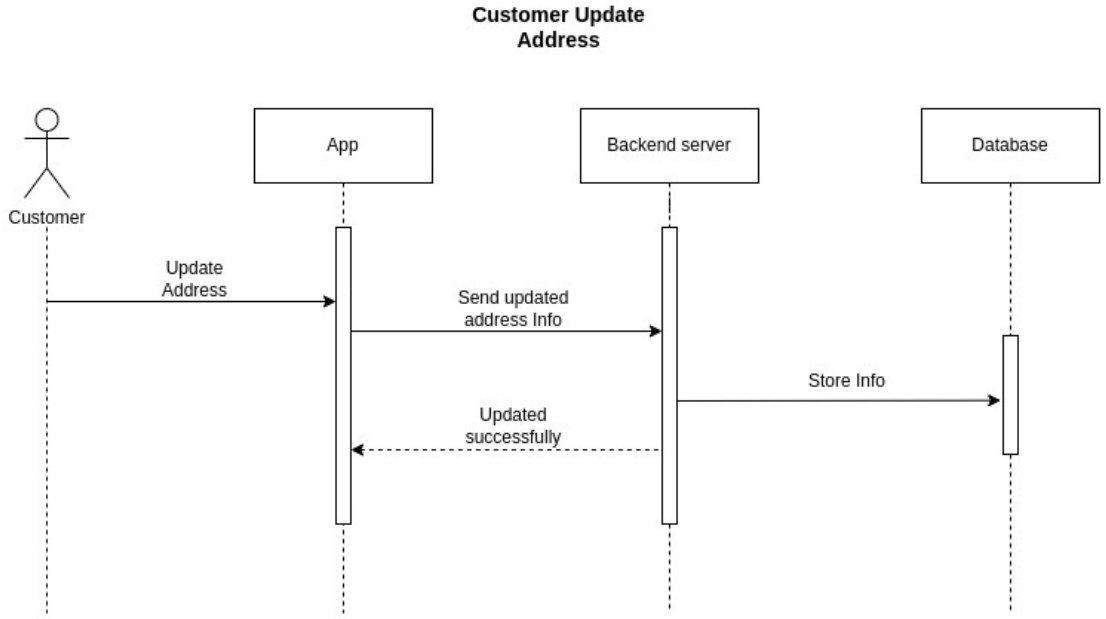

Figure 22: Customer Update Address Sequence Diagram

3.5.18 Cart List

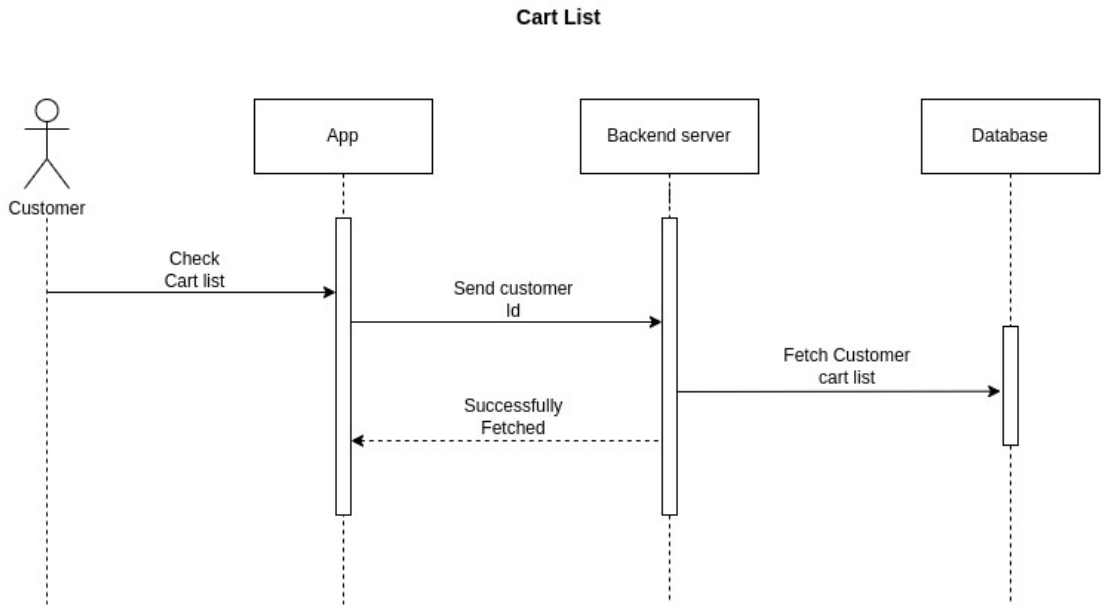

Figure 23: Cart List Sequence Diagram

3.5.19 Track Order

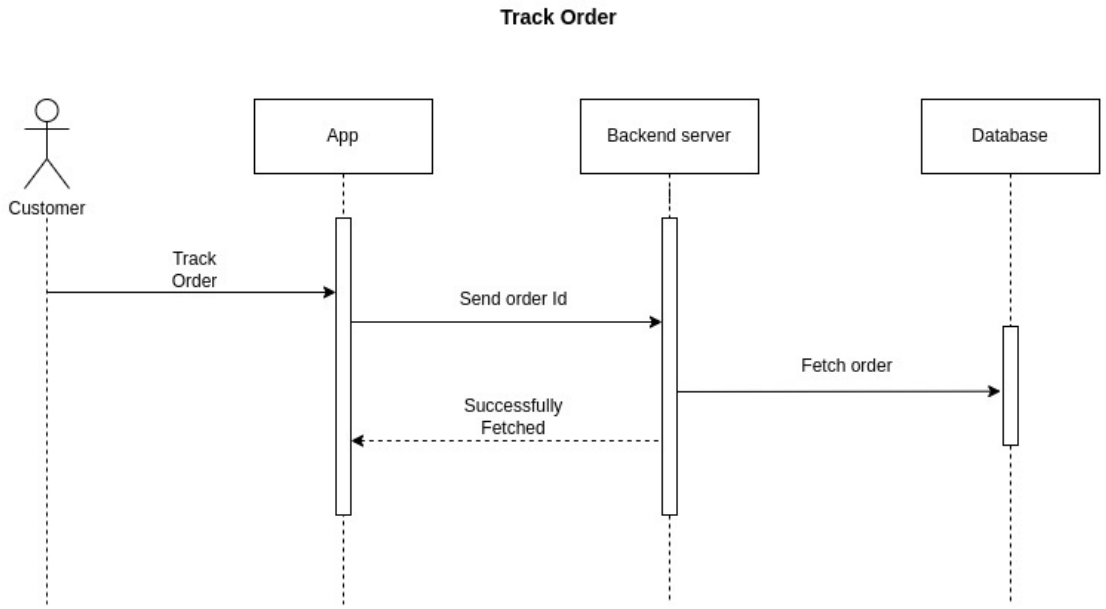

Figure 24: Track Order Sequence Diagram

3.5.20 Order History

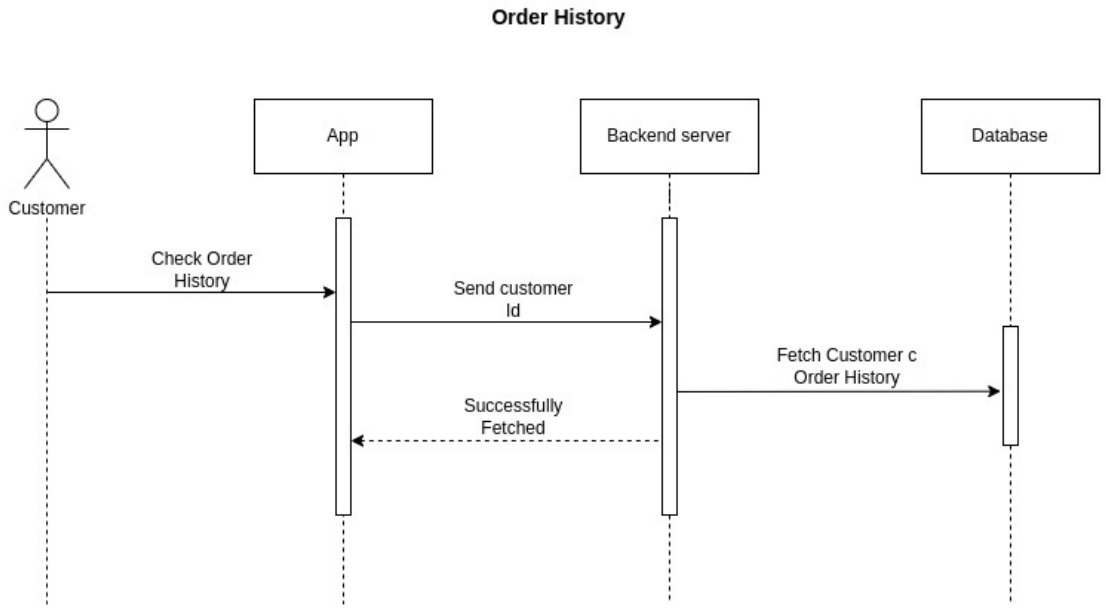

Figure 25: Order History Sequence Diagram

3.5.21 Payment

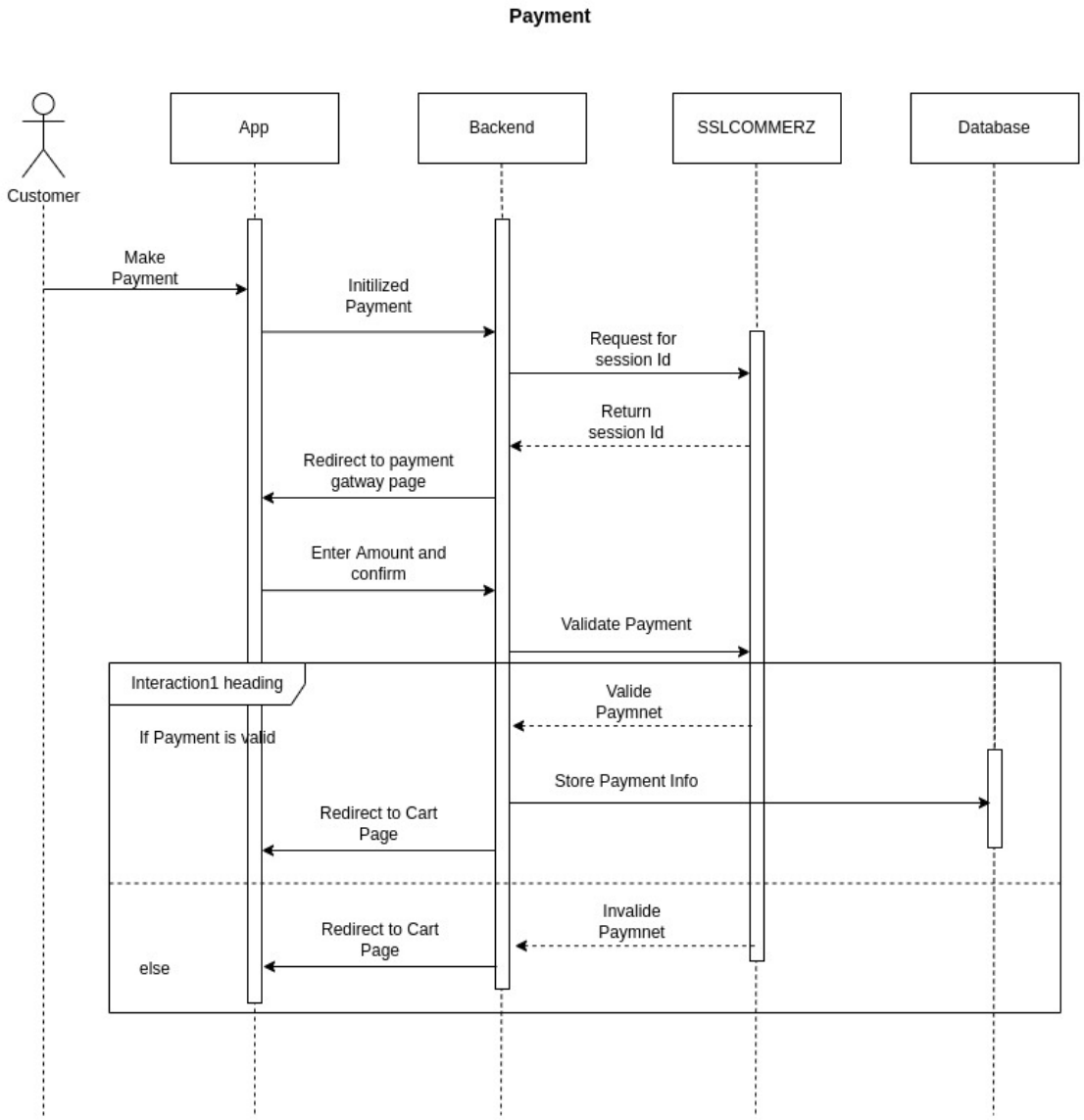

Figure 26: Payment Sequence Diagram

3.6 Design Requirement

We research and analyze so many designs for making an attractive design for our project. We have three parts in 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, 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

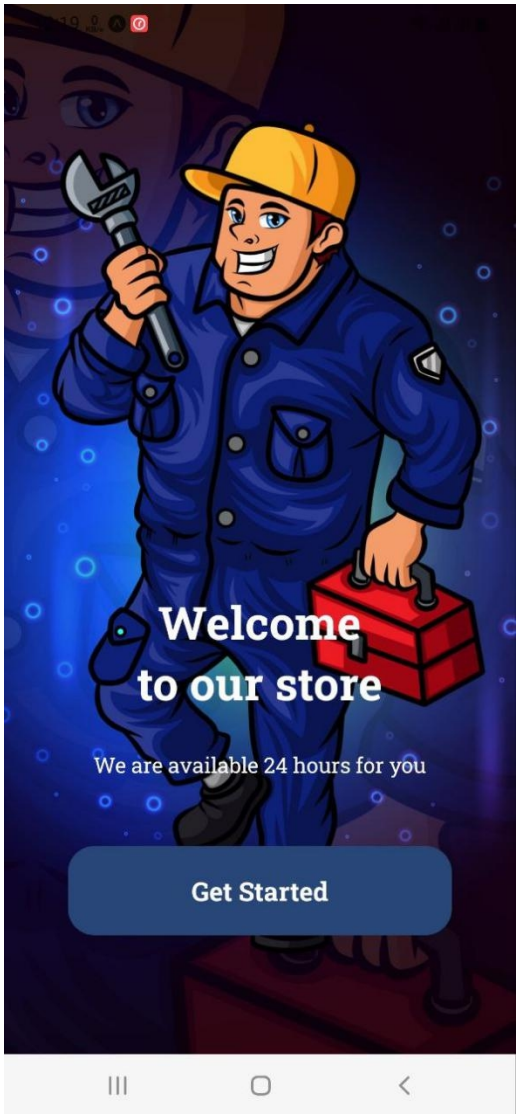

Figure 27: Welcome Page

Login Page

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

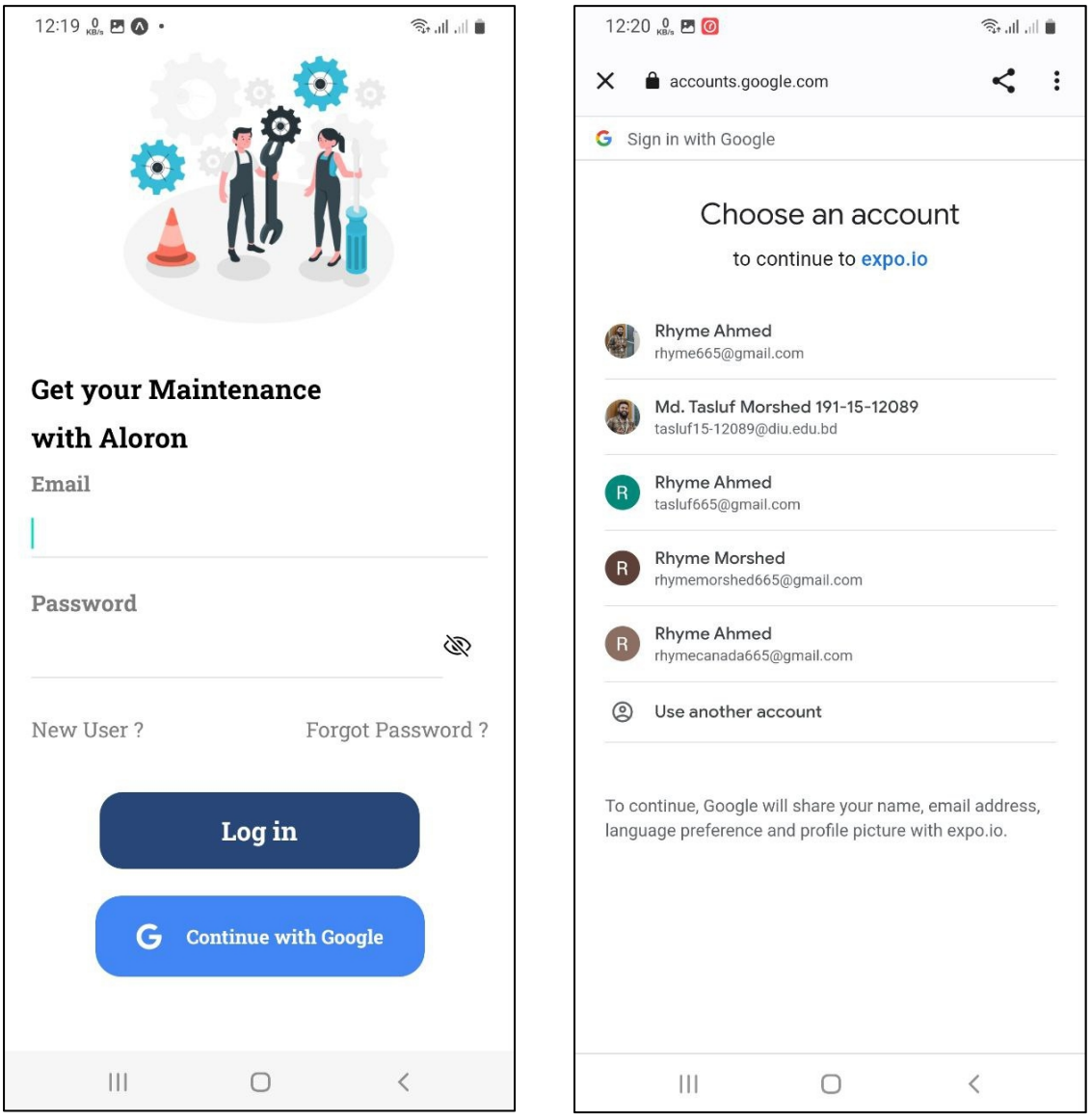

Figure 28: Login Page

Sign Up Page

Backend authenticate signup need to enter in this application. Required information (1) Name, (2) Email, (3) Password (4) Confirm Password then user should Click on "Sign Up" button. Then user get a verification email with a verification link. If user click that link backend will automatically verify user account and allow that user to login.

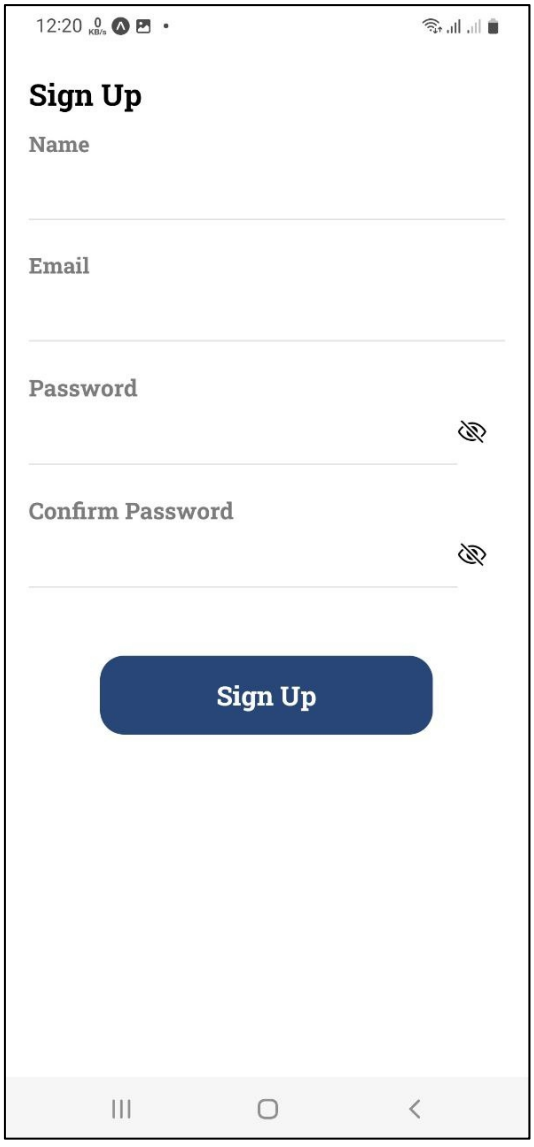

Figure 29: Sign Up Page

Home Page

After successfully login or signup user will redirect to the home page.

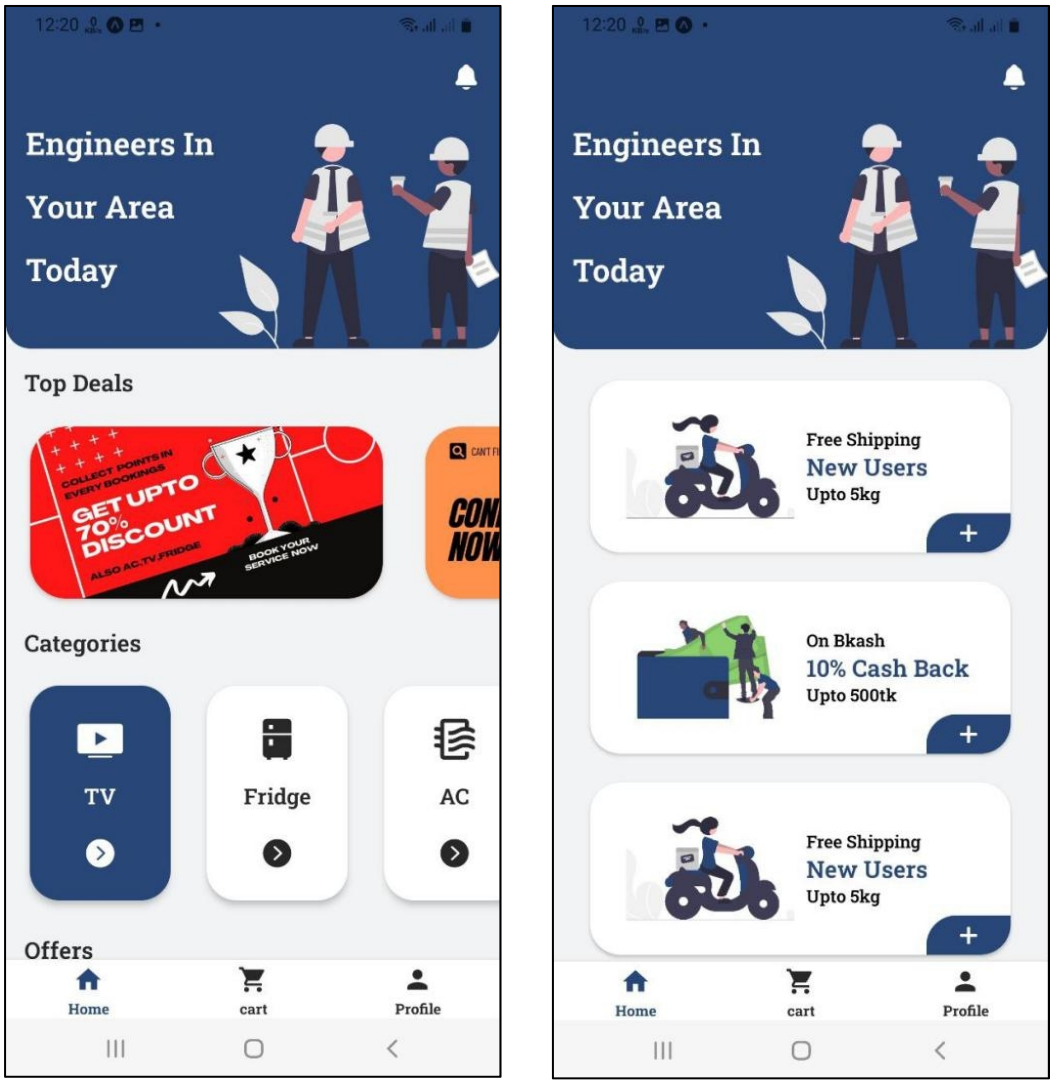

Figure 30: Home Page

Order Page

By submitting a form user can hire techenicien to repair their product. They have fill up some information and book for an order.

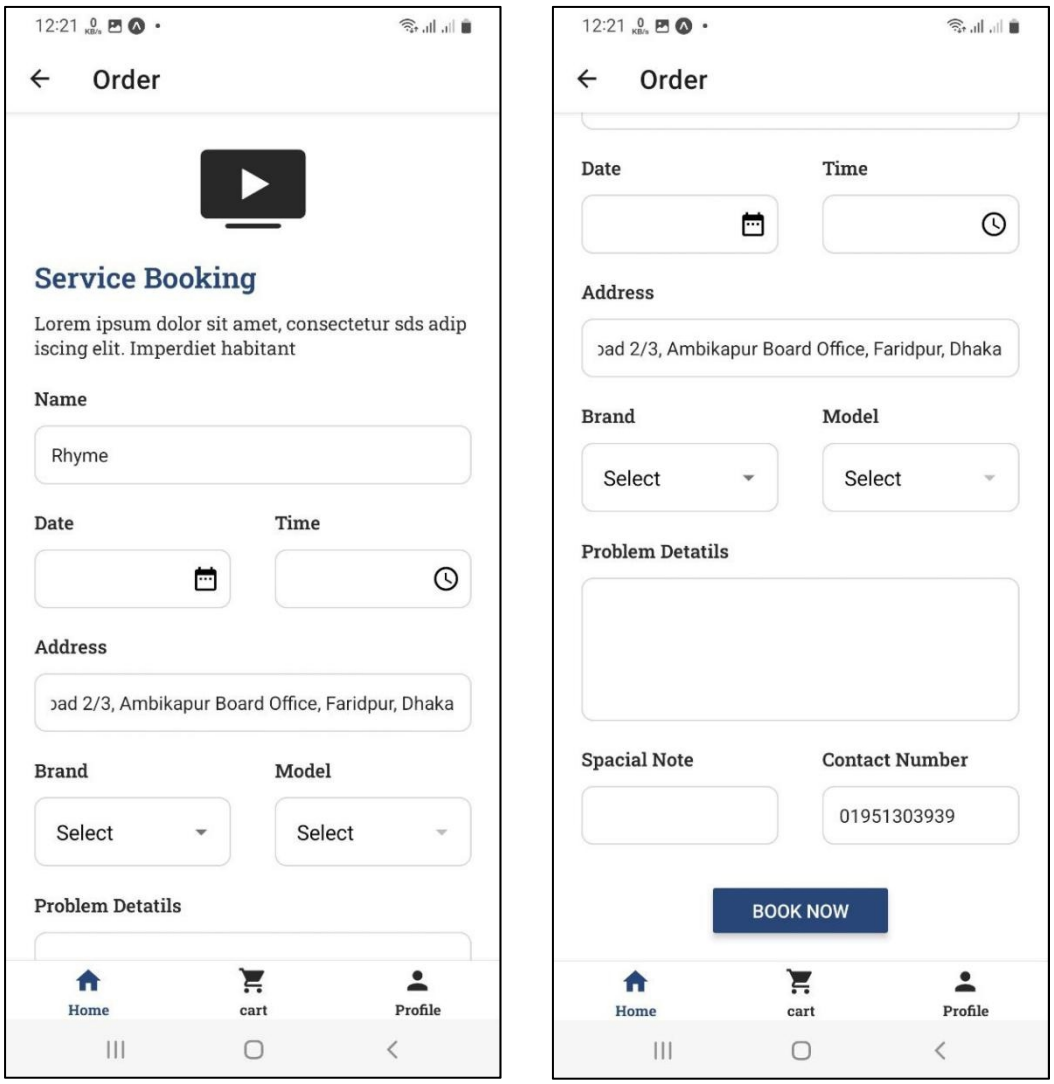

Figure 31: Order Page

My Cart Page

After submitting an order user can track their order in the My cart page. And after repering the product user will able to pay their bills.

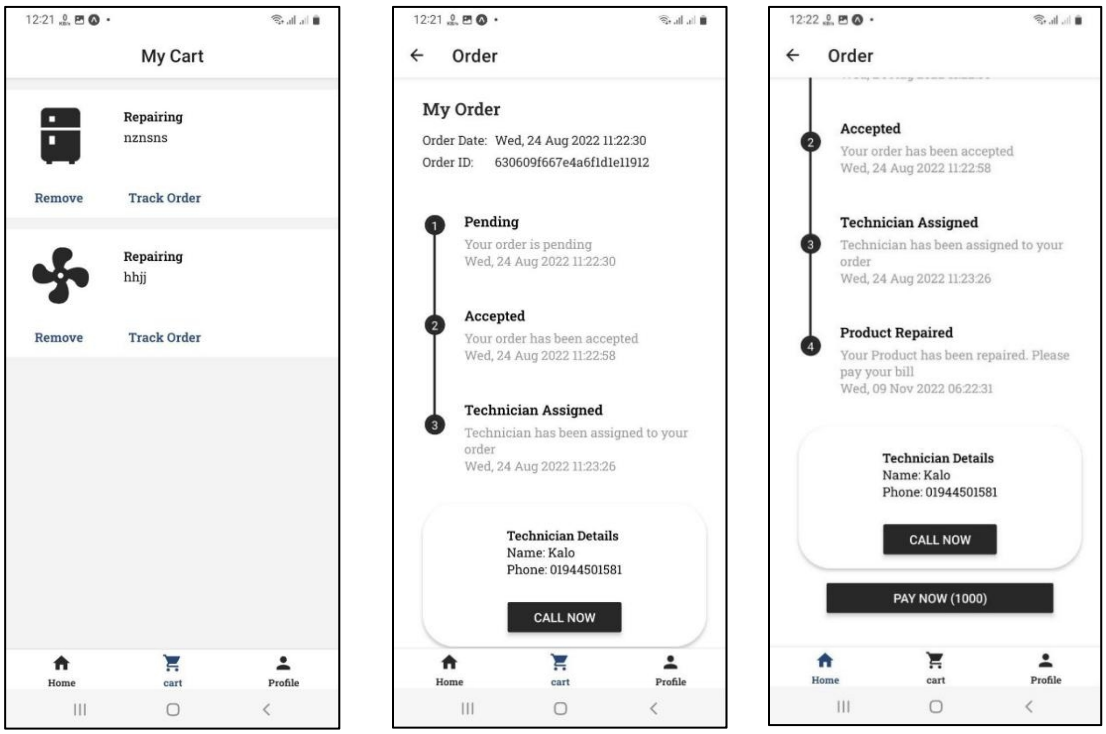

Figure 32: My Cart Page

Payment Page

User are able to pay their bill by sslcommerz payment getway. By this getway they can use any kind of internet banking system like Bkash, Nagad, Rocket etc. They can also use Visa and master Card for payment.

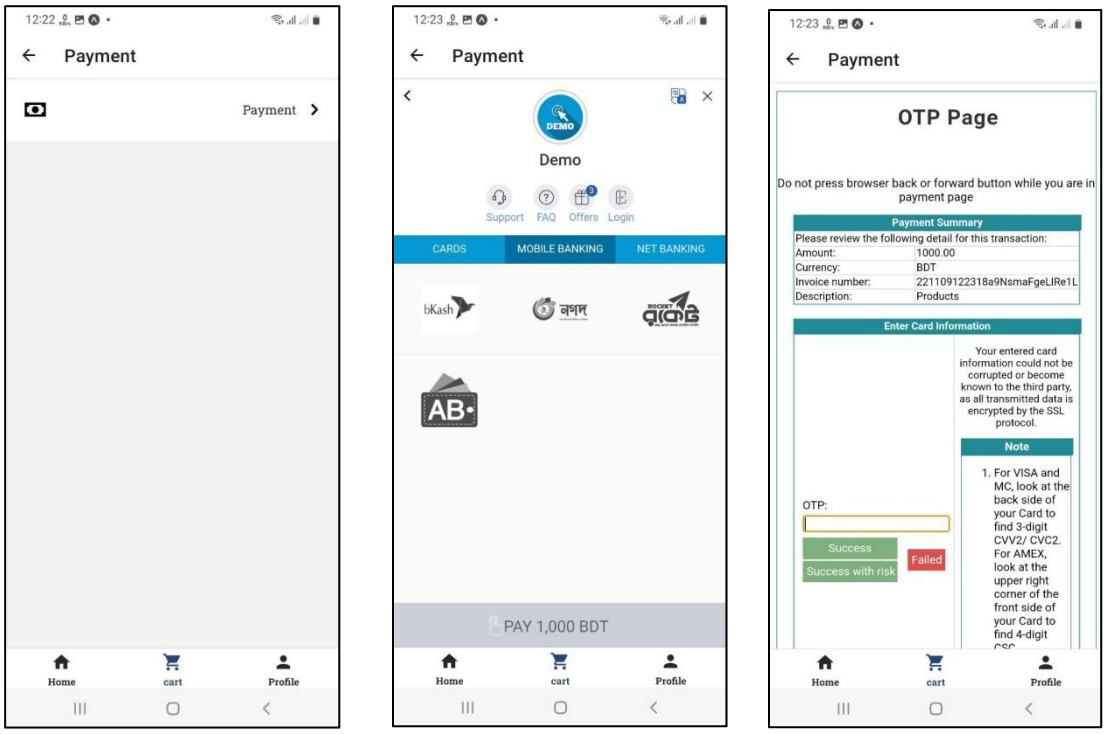

Figure 33: Payment Page

Profile Page

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

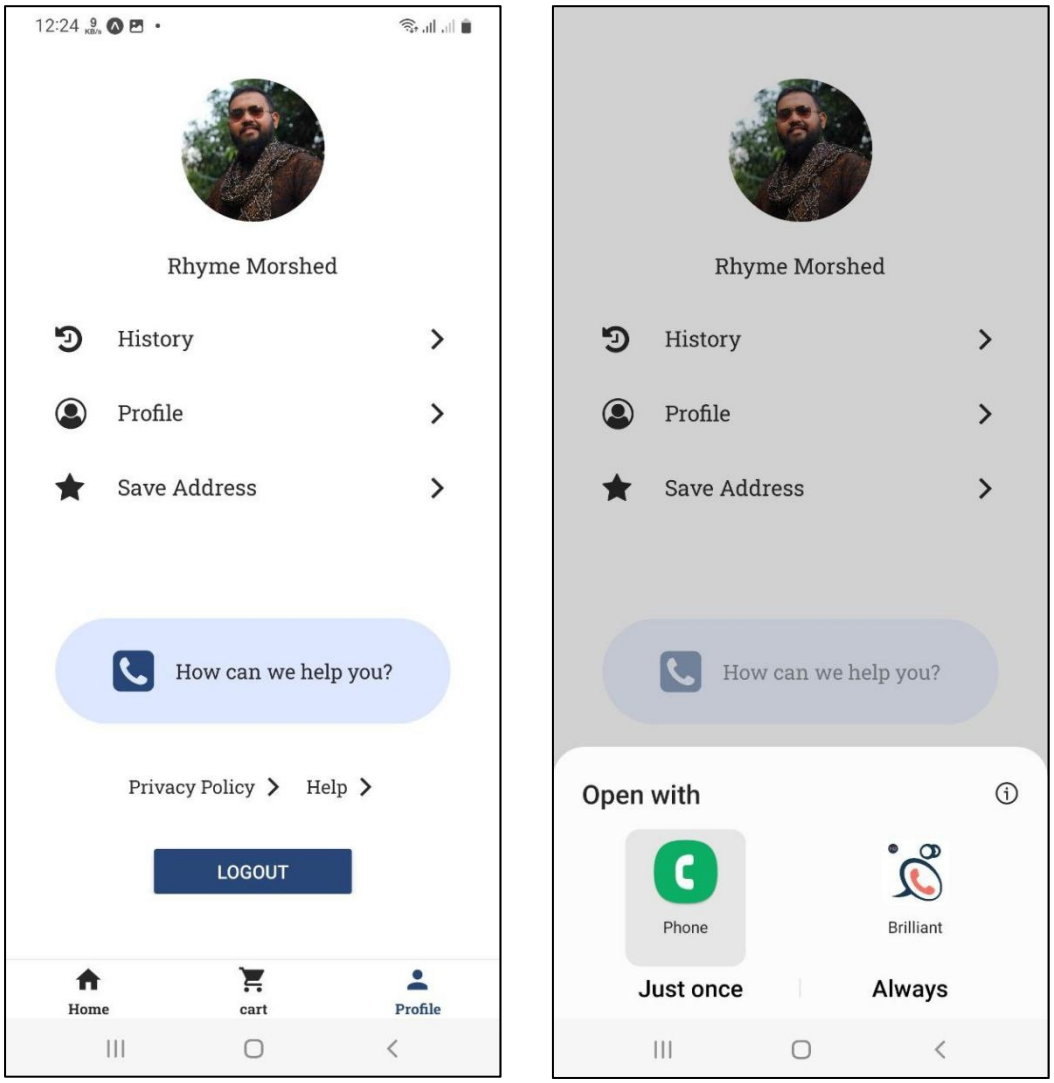

Figure 34: Profile Page

History Page

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

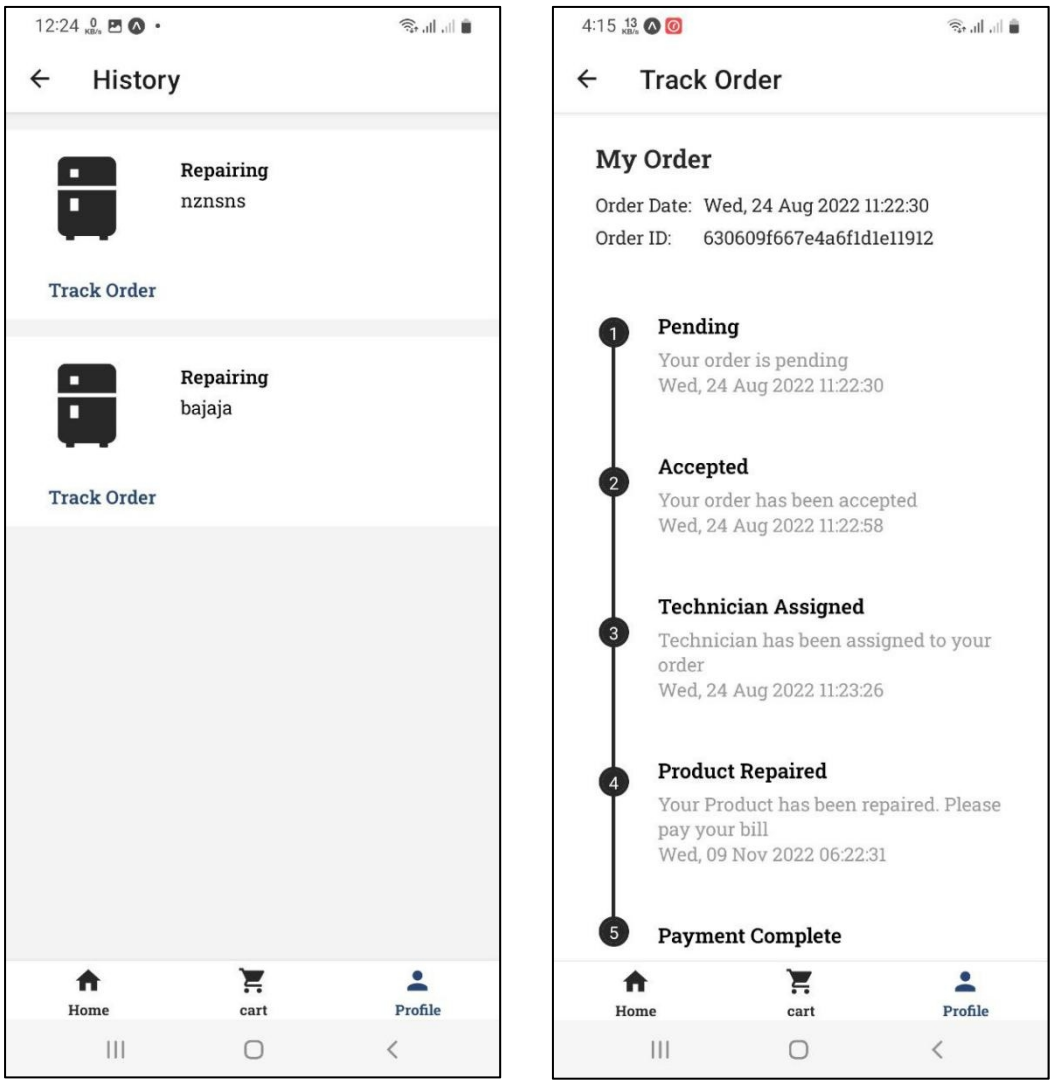

Figure 35: History Page

Profile Edit Page

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

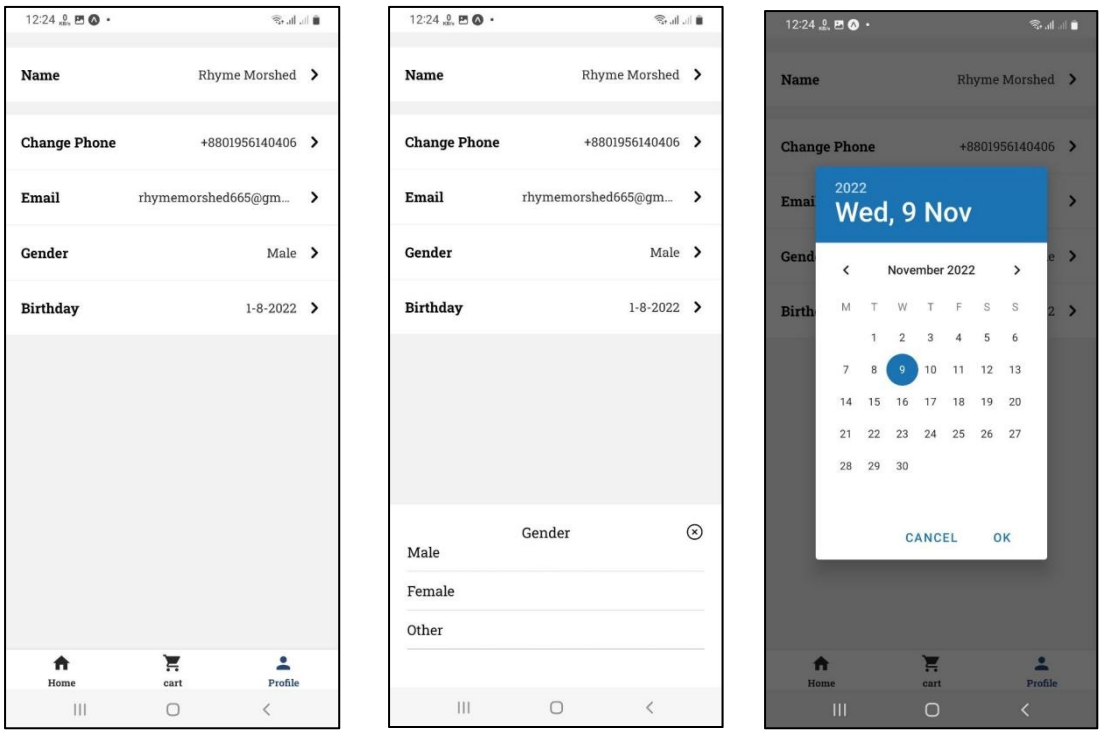

Figure 36: Profile Edit Page

Address Page

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

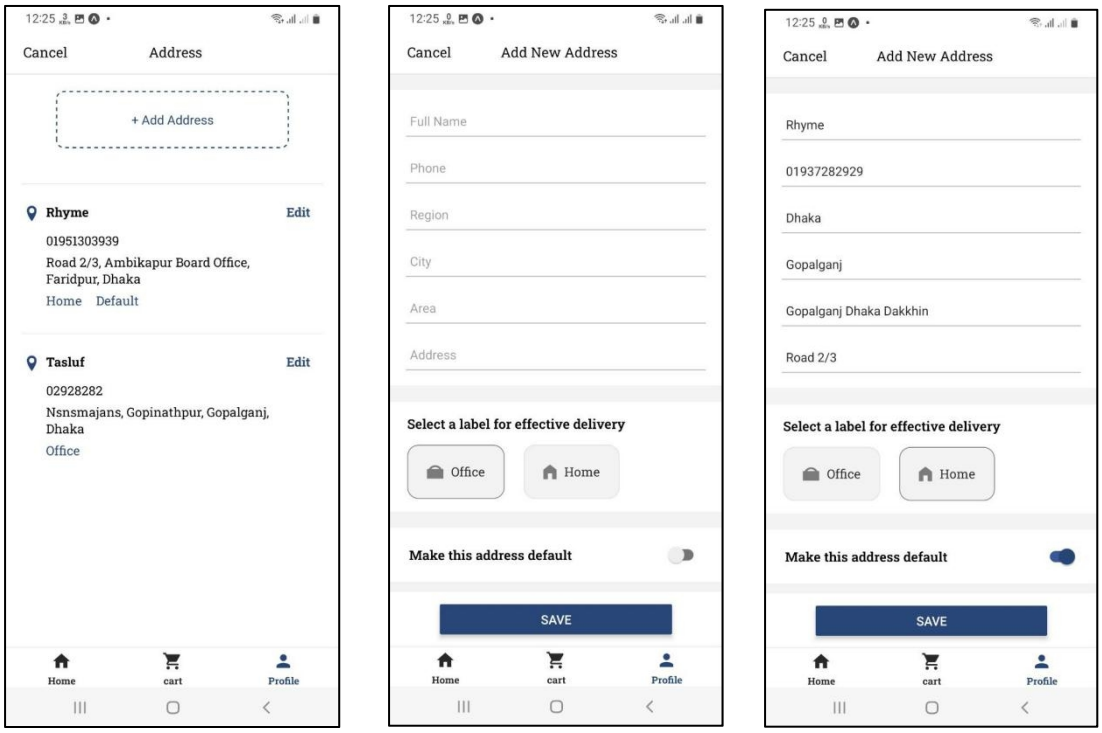

Figure 37: Address Page

Select Address Page

User can select their Region, City and area of their address.

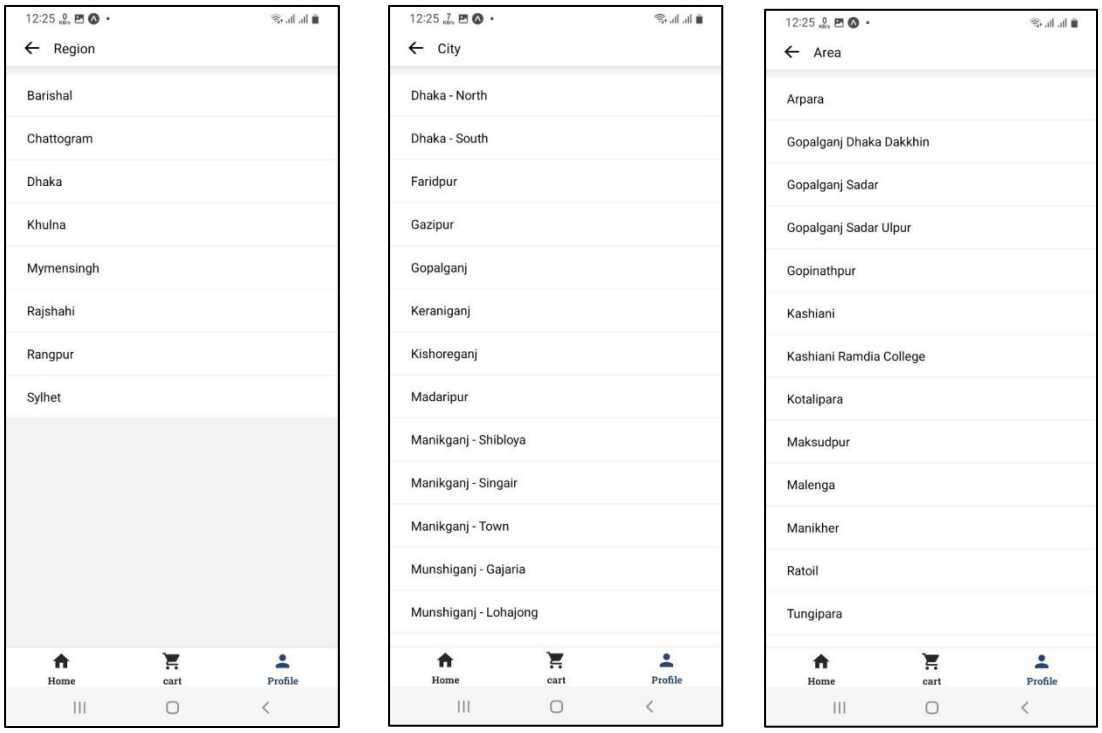

Figure 38: Select Address Page

Notification Page

All the notification of the app will be show in this page.

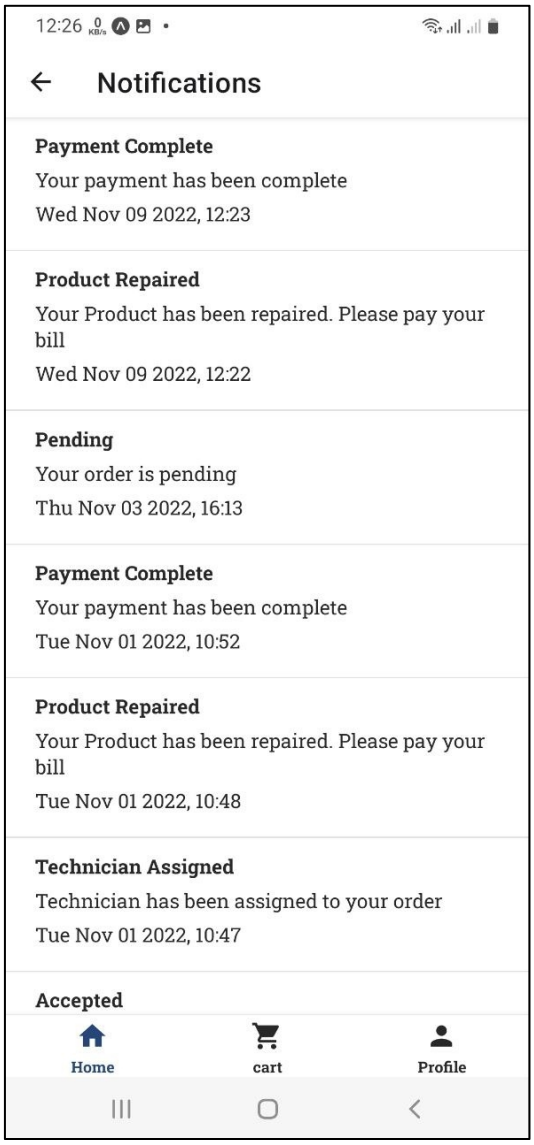

Figure 39: Notification Page

4.1.2 Admin Panel

Login Page

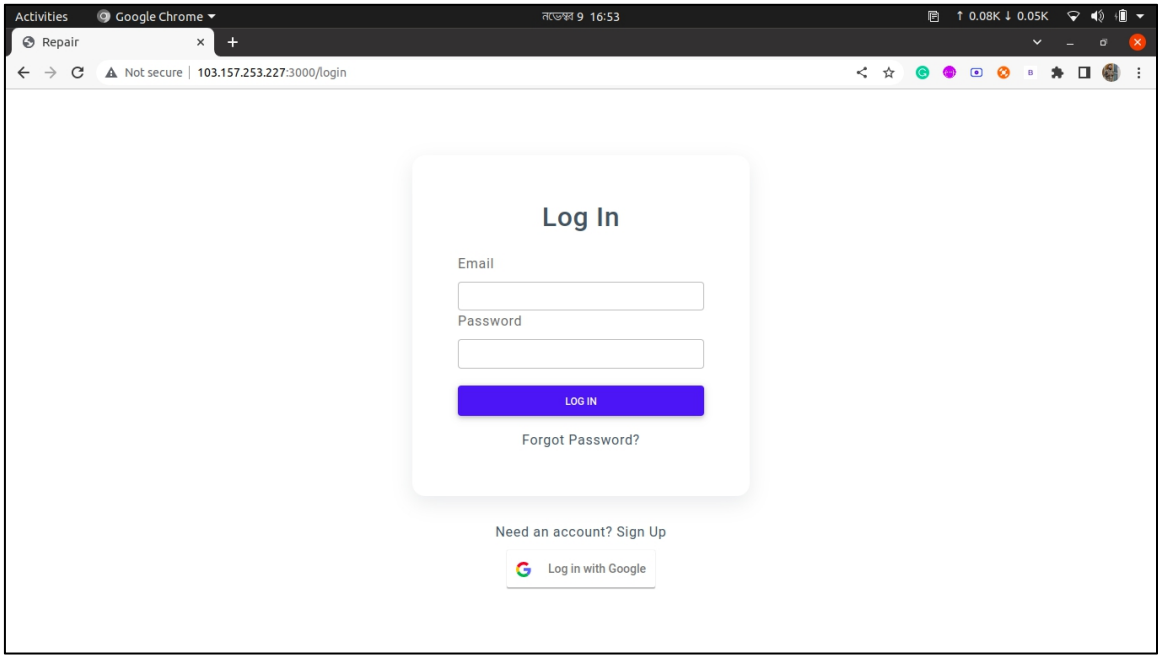

Figure 40: Login Page

Sign Up Page

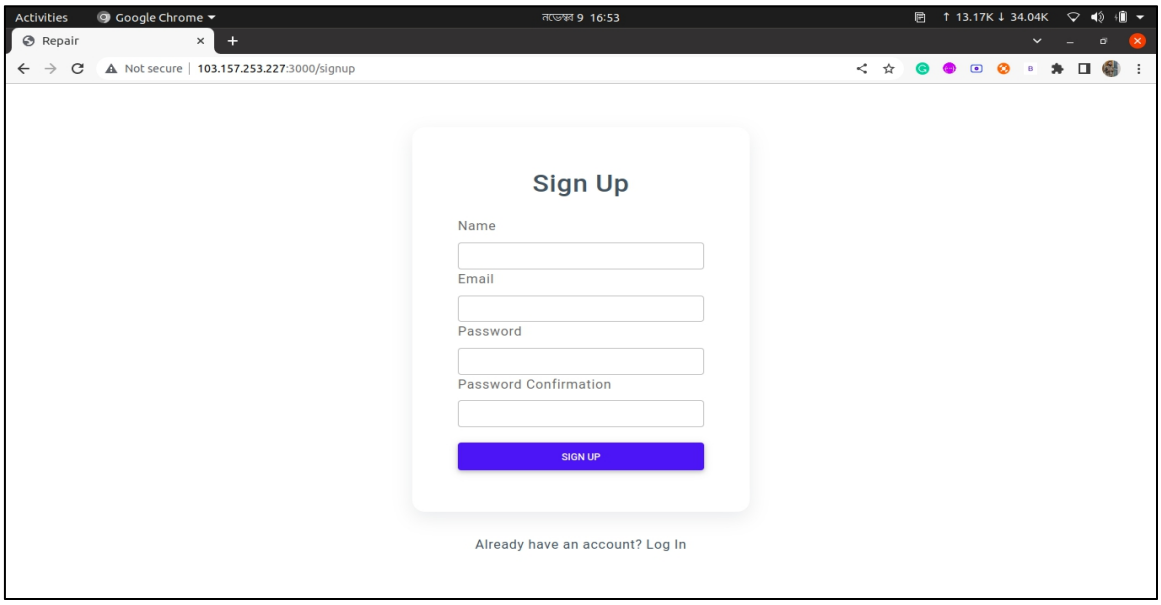

Figure 41: Sign Up Page

Forgot Password Page

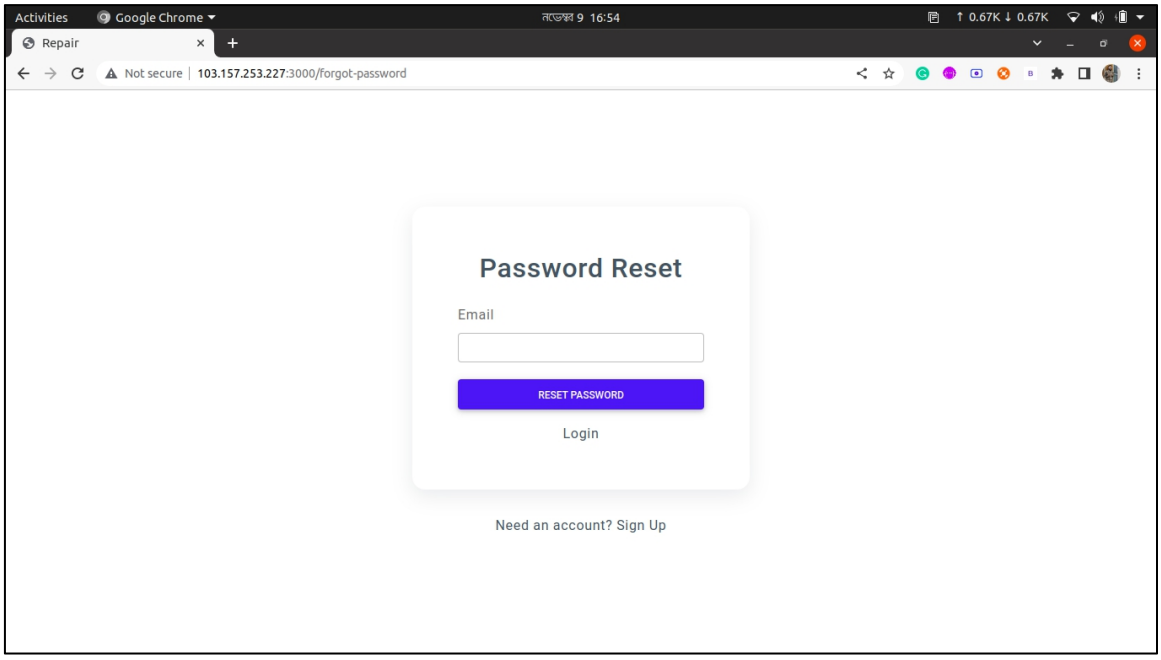

Figure 42: Forgot Password Page

Dashboard Page

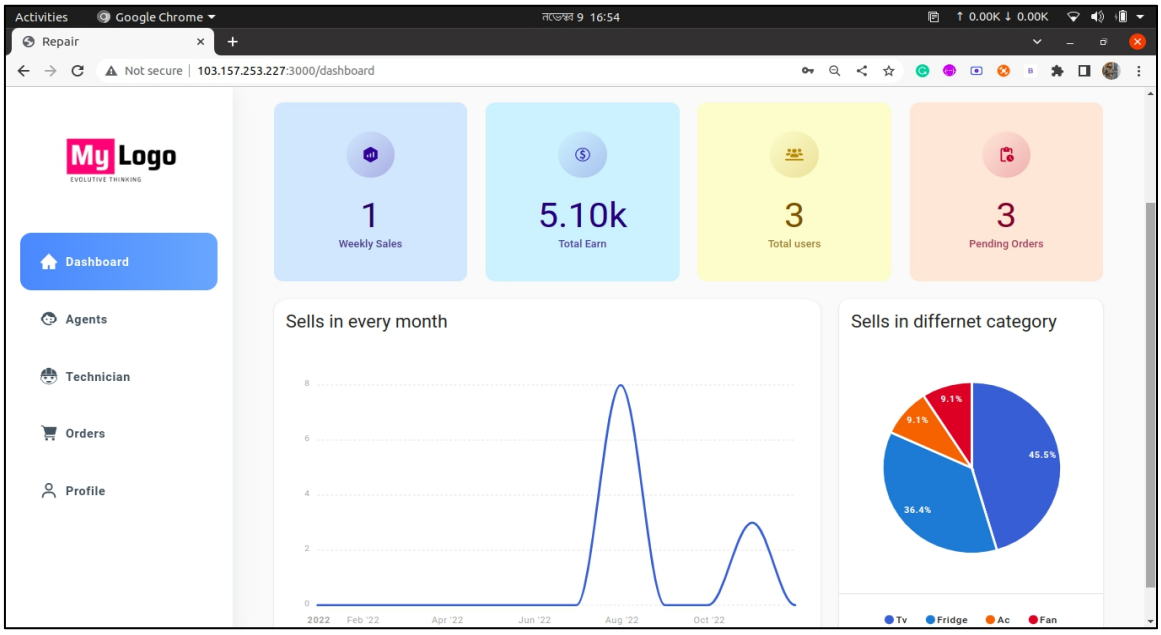

Figure 43: Dashboard Page

Agent Page

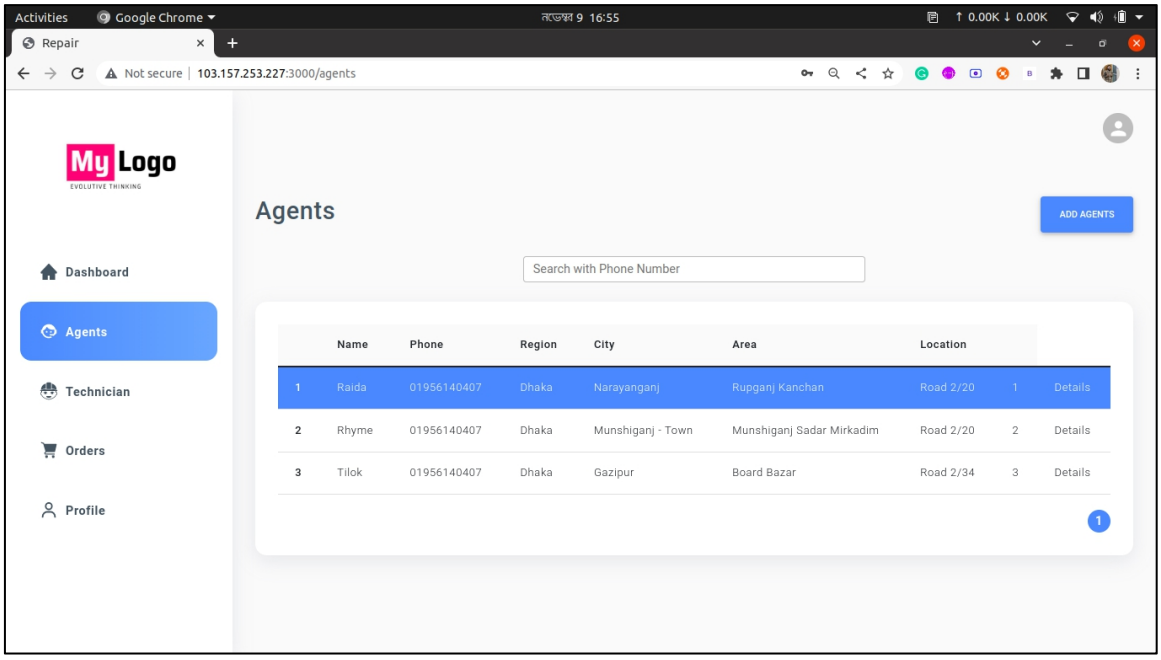

Figure 44: Agent Page

Add Agent Page

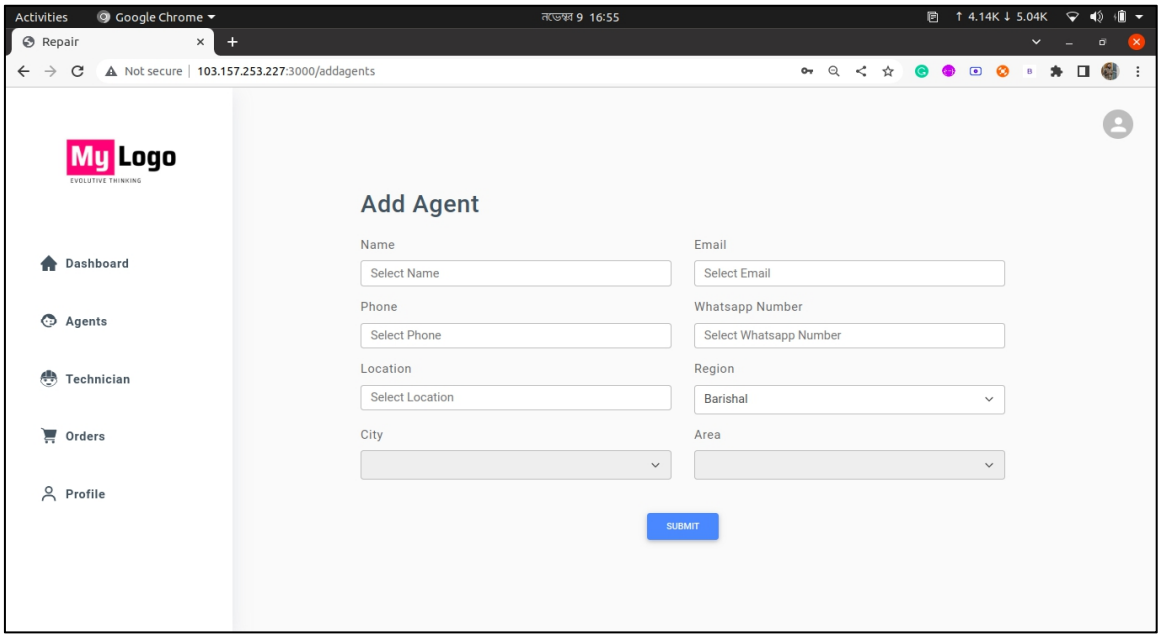

Figure 45: Add Agent Page

Update Agent Page

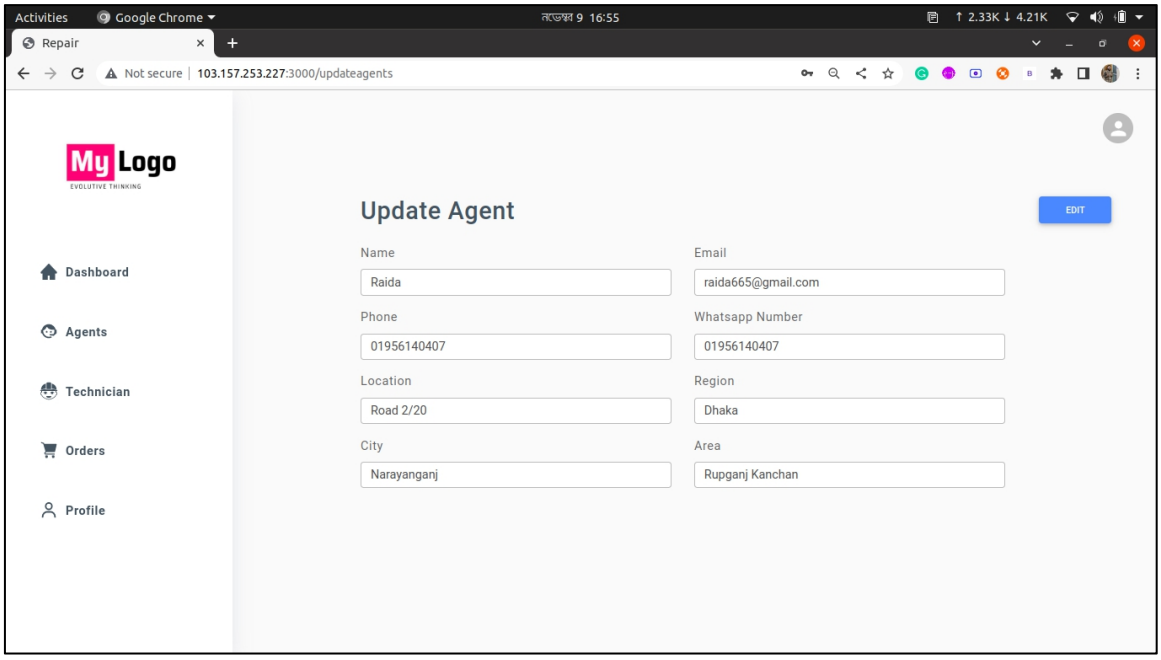

Figure 46: Update Agent Page

Technician Page

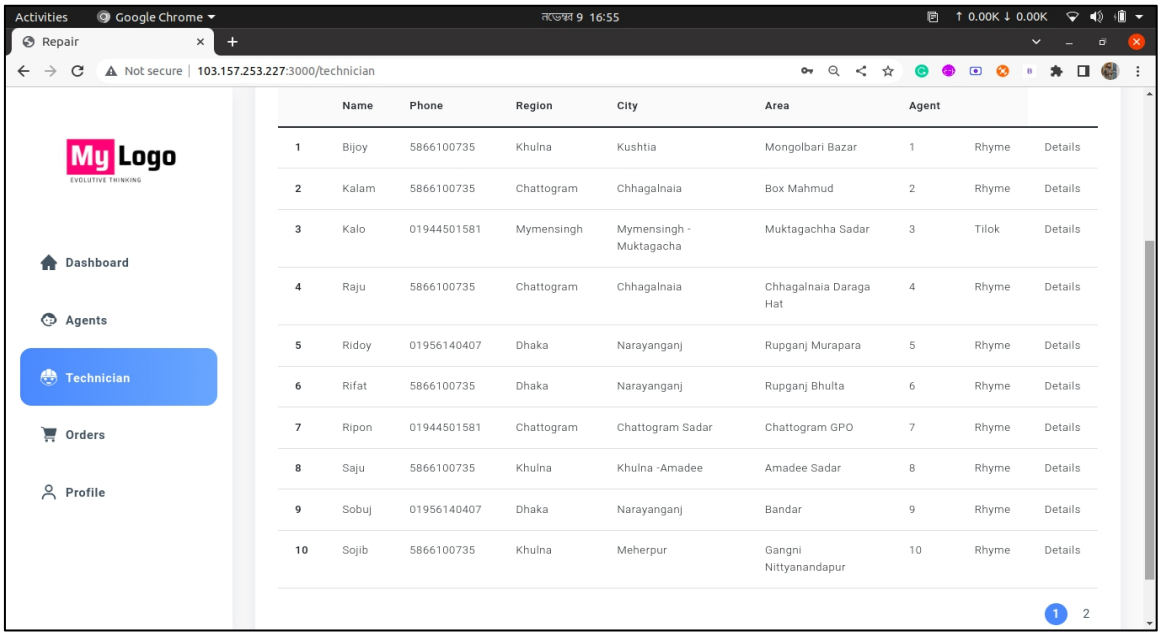

Figure 47: Technician Page

Add Technician Page

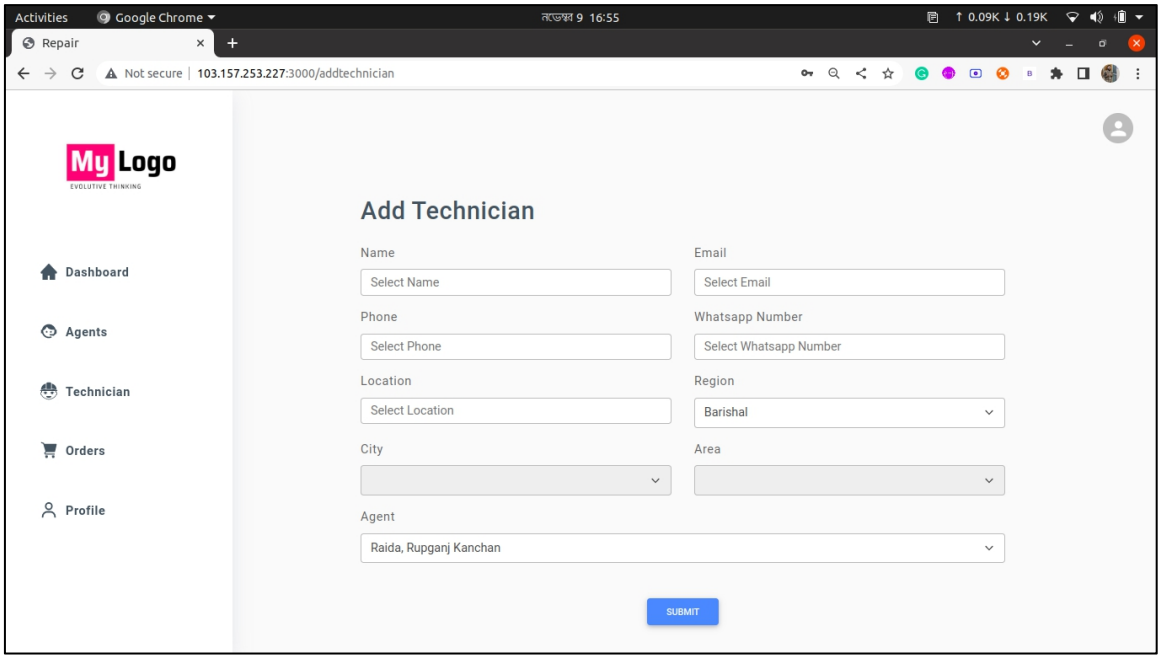

Figure 48: Add Technician Page

Order Page

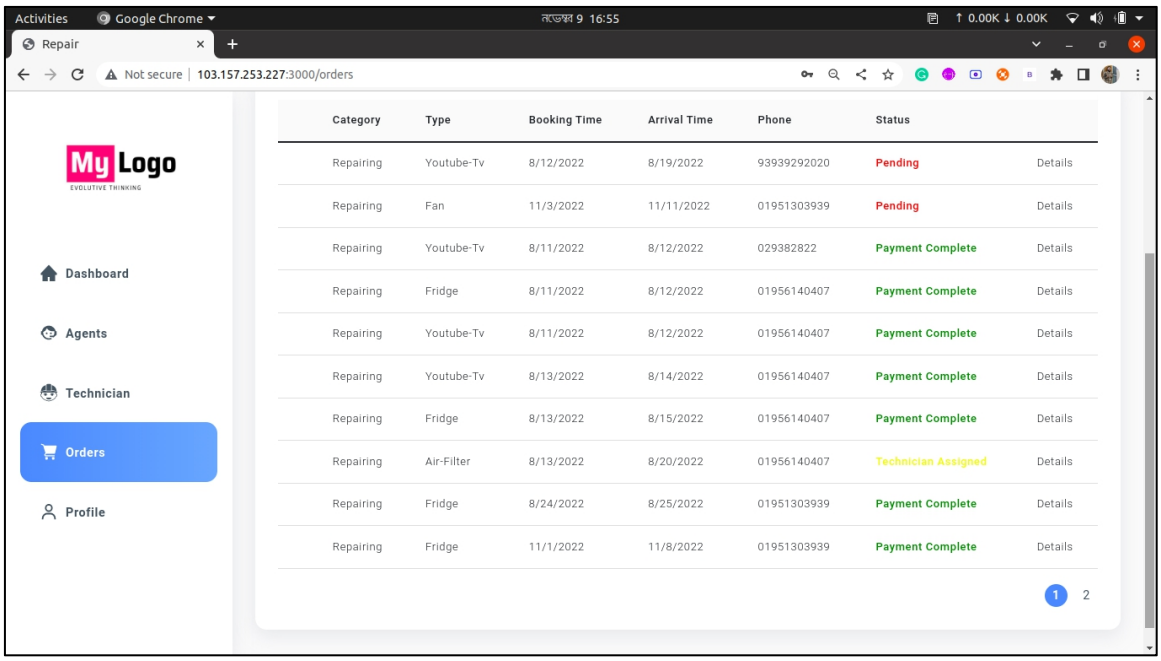

Figure 49: Order Page

Order Details Page

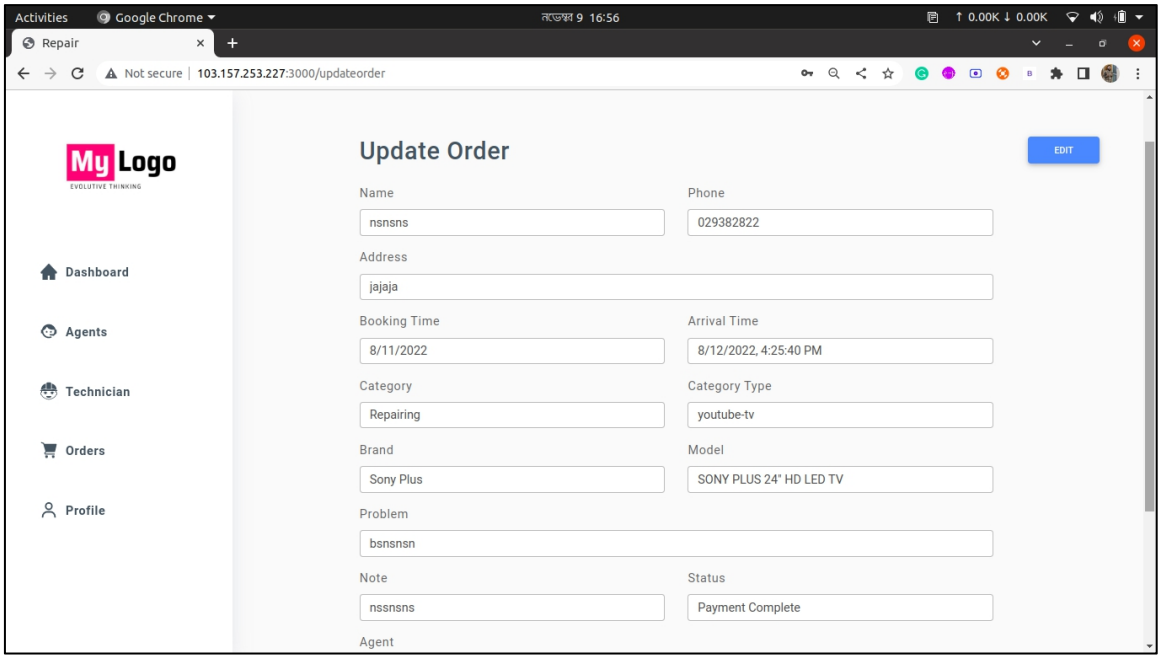

Figure 50: Order Details Page

Profile Page

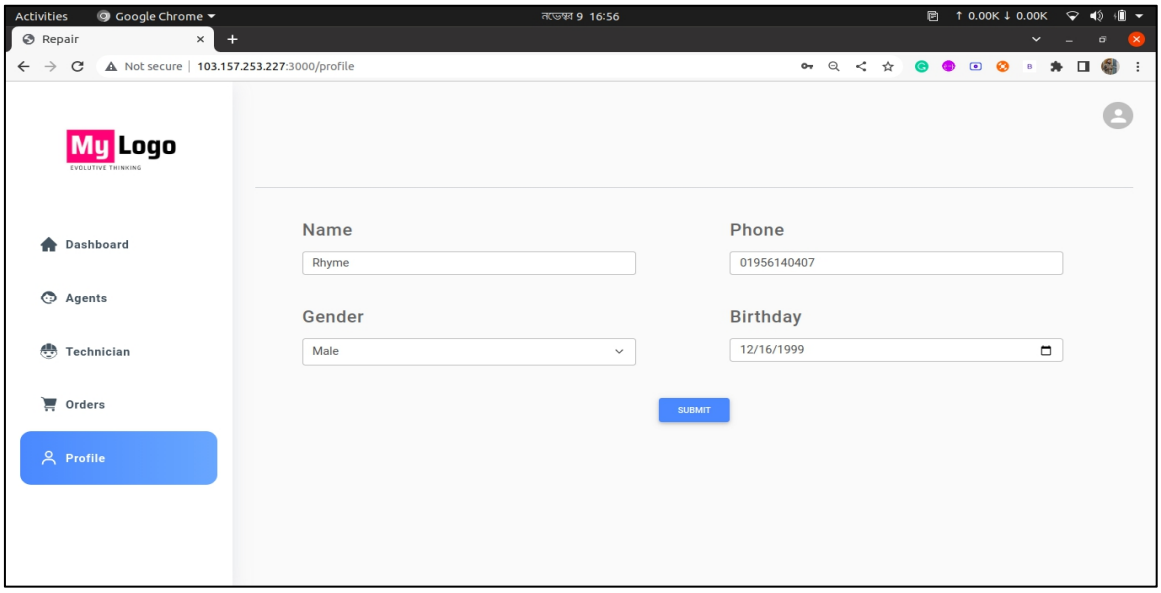

Figure 51: Profile Page

4.2 Back End Design

Database

We used nodejs for user authentication for this project. For storage we use MongoDB atlas NoSQL database [cloude database storage].

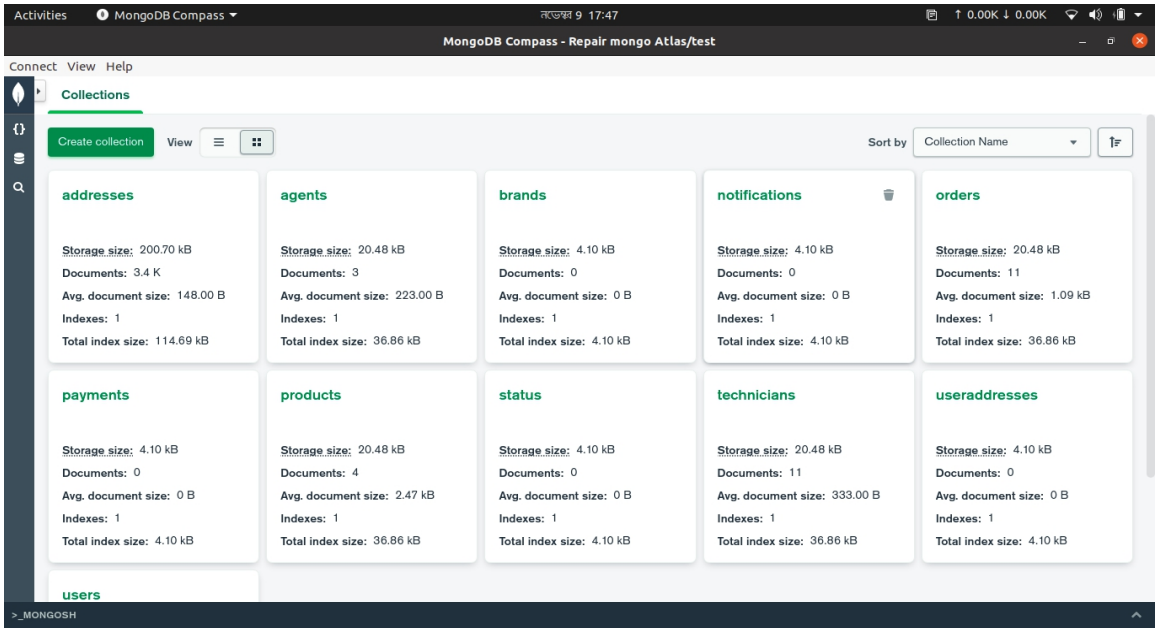

Figure 52: All Schema for database

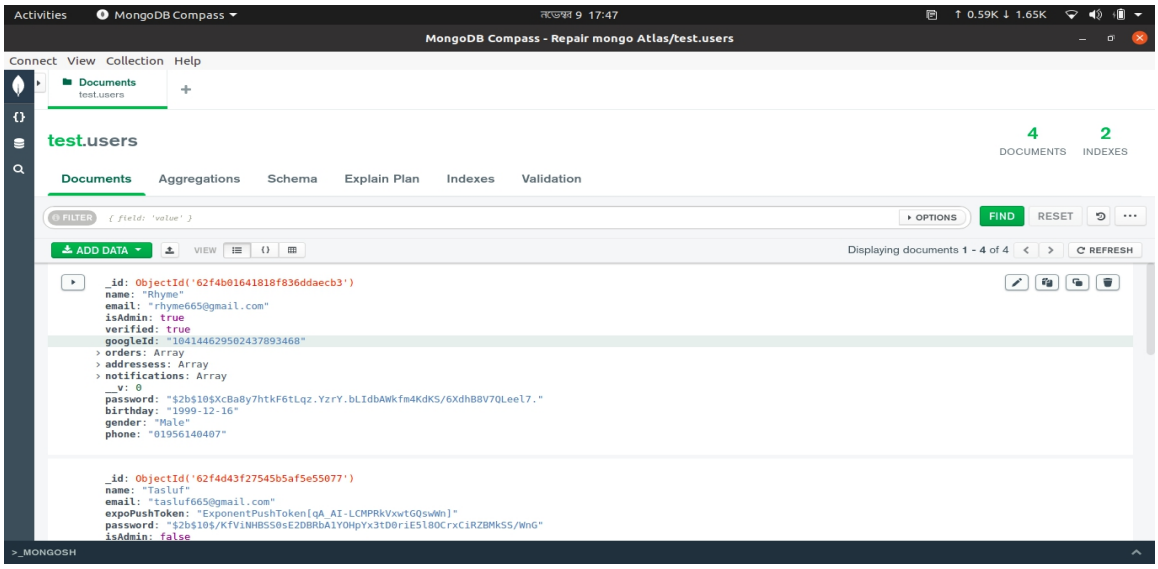

Figure 53: User Table

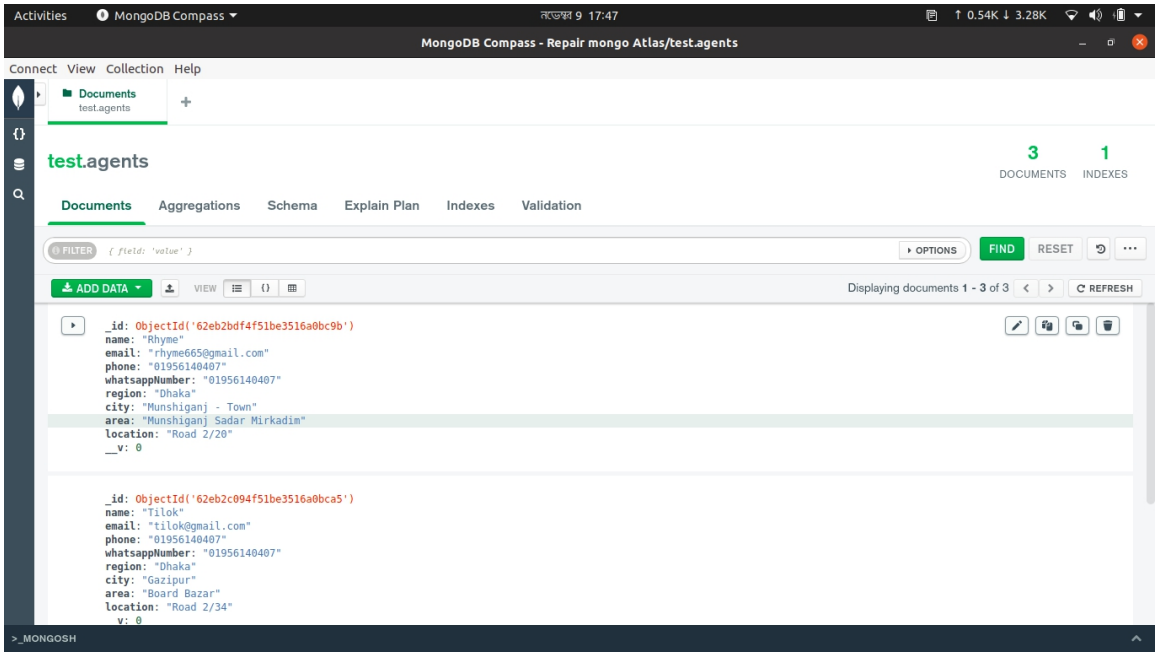

Figure 54: Agent Table

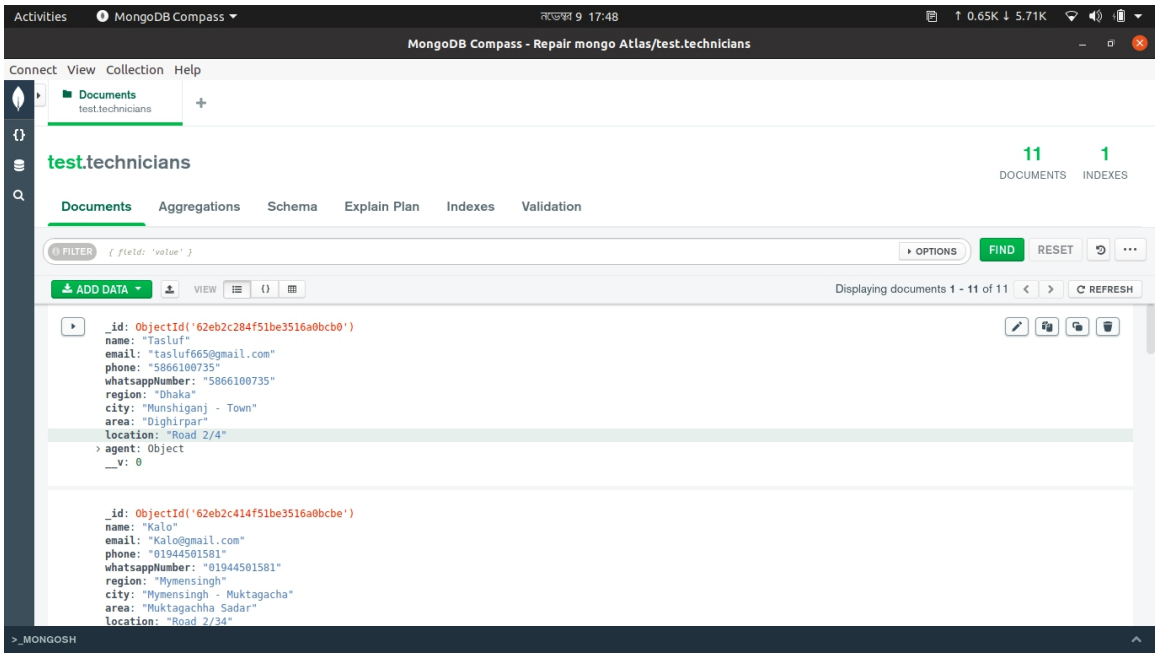

Figure 55: Technician Table

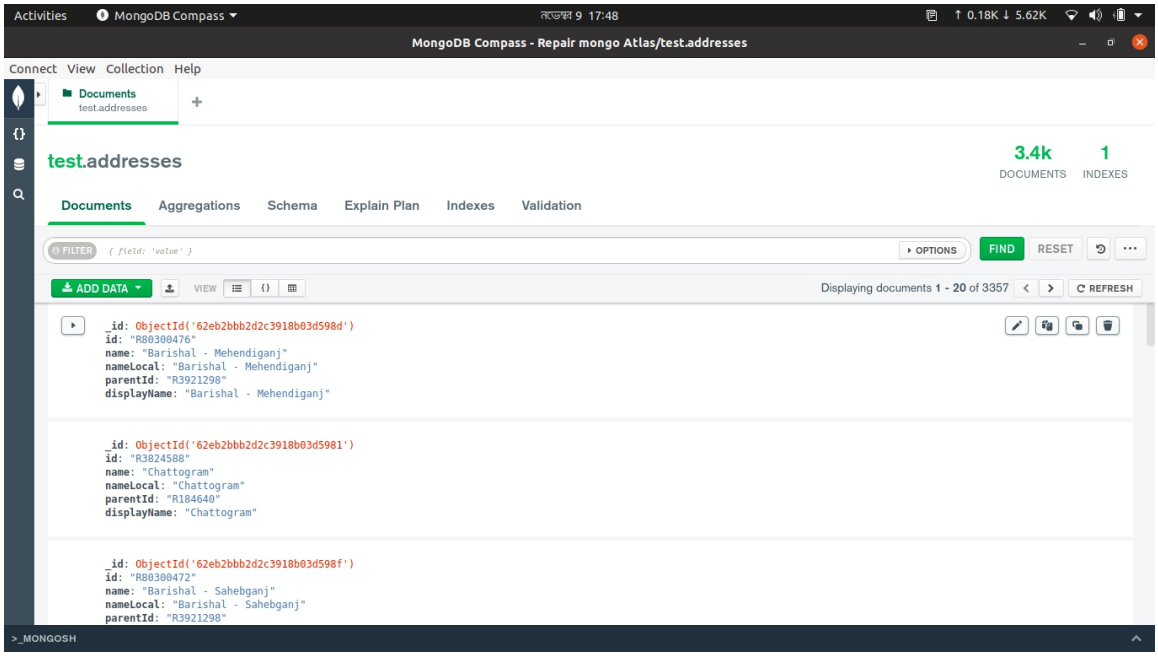

Figure 56: Address Table

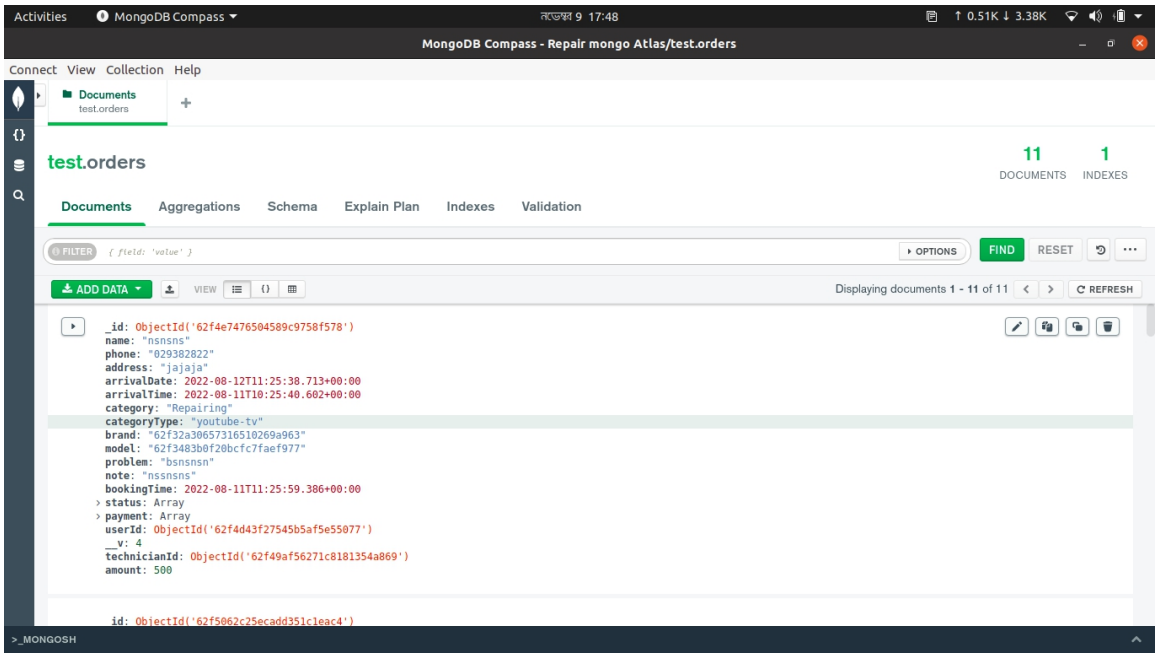

Figure 57: Orders Table

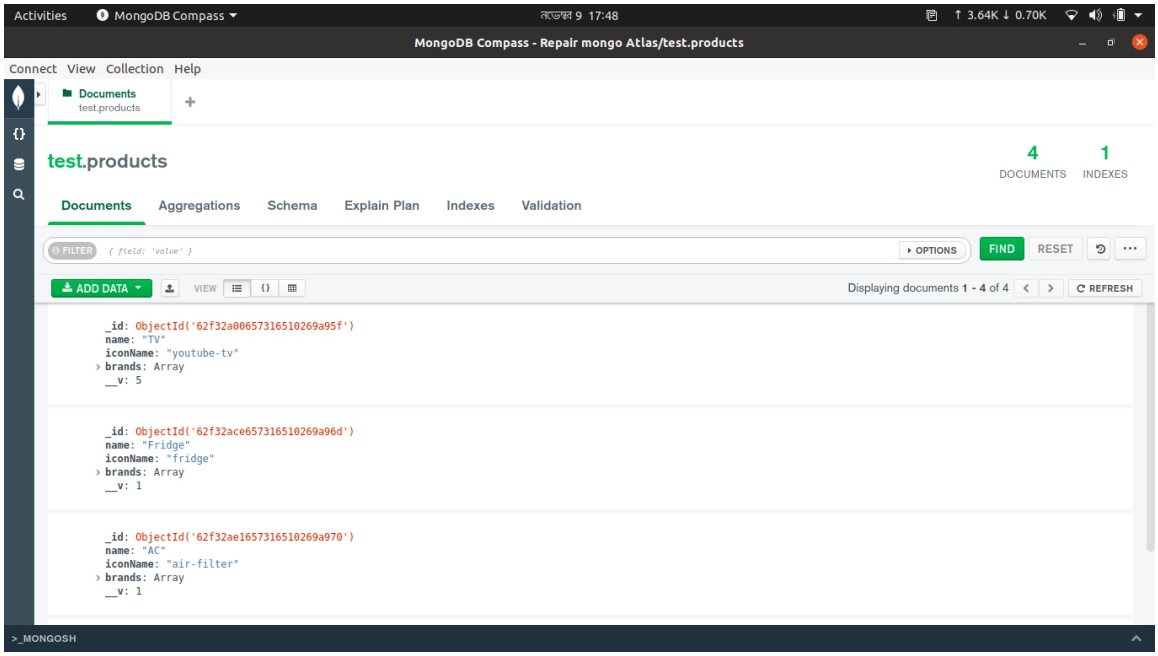

Figure 58: Products Table

4.3 Interaction Design and User Experience (UX)

I have used Figma [online UX design application] for the initial design of our application. Then we have implement this design in our android app and admin panel. We use different React and React native libraies. 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 backend 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 backend server. Then our server will authenticate the user and fetch data from the MongoDB database. Our NodeJS backend 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 backend 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 had implement 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 Backend 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	Valide Email and Password	Successfully Login	Successfully Login	Pass	10-11-22
Login	Valide Email and Invalide Password	Get 400 Error	Get 400 Error	Pass	10-11-22
Login	Valide 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

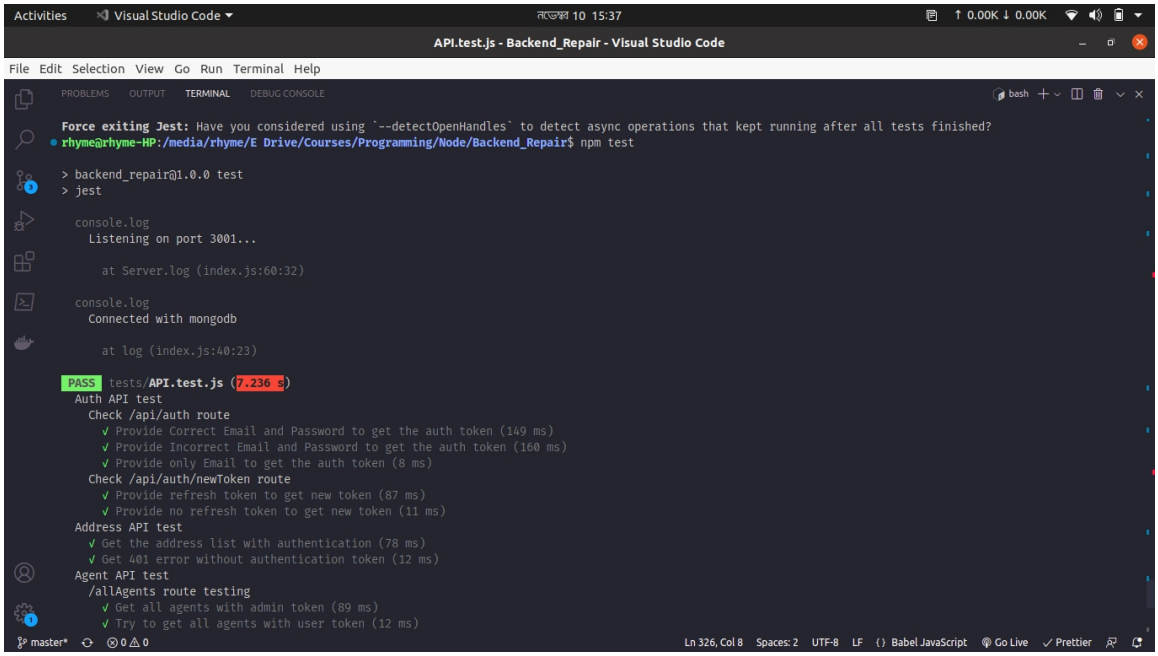

Figure 59: Automated Test figure 1

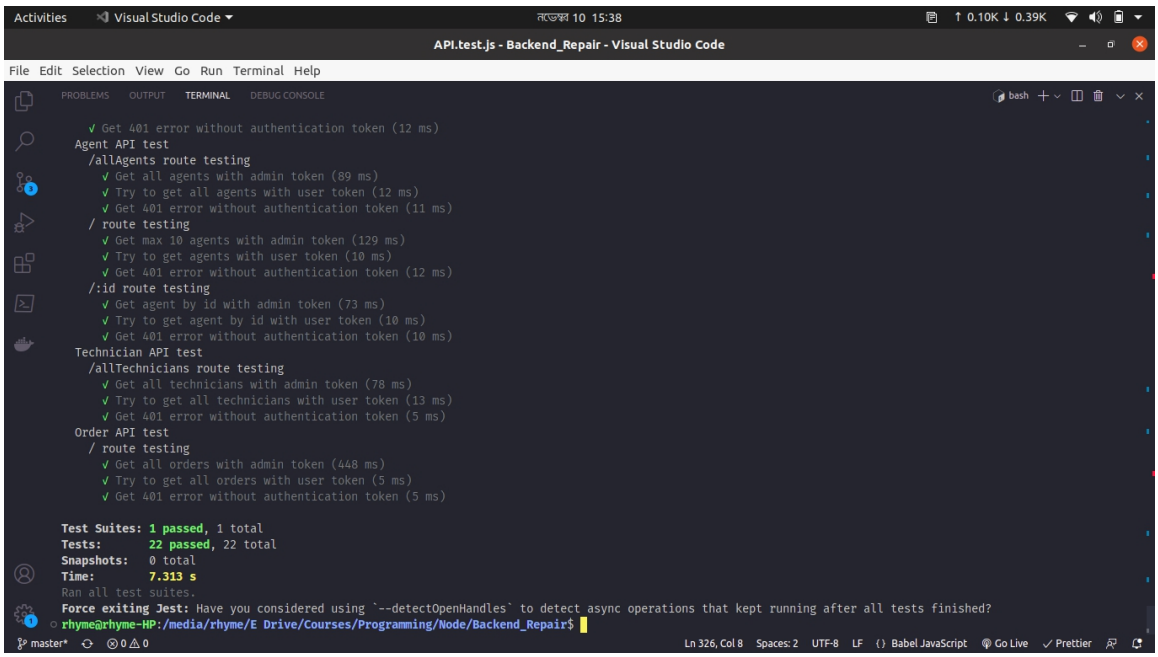

Figure 60: Automated Test figure 2

5.4.1 Manual Testing

Reference: Font-10

All references to books, papers, and other publications must be fully and correctly quoted. There are several methods of quoting references. One is to state the name of the author and a serial number in the main text with the full details of the reference in the Reference section of the report, for example:

In the text:
....The analysis of the algorithms has been extensively reviewed by Yorozu et al. [1] and will ....

In the References section:

[1] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987.

Conference/Journal Papers:

[1] Author1, Author2, and Author3, "Paper Title", Conference/Journal, Volume, page number, Month and year.

Example:

[1] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987.

Books:

[2] Author, Book Title, Edition/Volume, Publisher, Year, Page number

Example:

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3rd Edition, The MIT Press, 2009, pp. 120-122.

Websites:

[3] Name/Title of the Website, available at << https://URL>>, last accessed on Date at Time.

Example:

[3] Learn about Wikipedia, available at << http://www.wikipedia.org/>>, last accessed on 06-06-2019 at 12:00 PM.