

**ANDROID BASED ELECTRONIC PRODUCT SERVICING SYSTEM**

**BY**

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

**AND**

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

**PROJECT ID: FL21D063**

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

Supervised By

**DR. MD. FOKHRAY HOSSAIN  
PROFESSOR**

Department of Computer Science and Engineering  
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY  
DHAKA, BANGLADESH  
JANUARY 2023**



## **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 29<sup>th</sup> January 2023.

## **BOARD OF EXAMINERS**

---

**Dr. Touhid Bhuiyan**  
**Professor and Head**

**Chairman**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Arif Mahmud**  
**Assistant Professor**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Saiful Islam**

**Assistant Professor**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

---

**Dr. Shamim H Ripon**  
**Professor**

**External Examiner**

Department of Computer Science and Engineering  
East West University

## **DECLARATION**

We hereby declare that, this project has been done by us under the supervision of **Professor Dr. MD. FOKHRAY HOSSAIN**, 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:**

---

**Dr. Md. Fokhray Hossain**  
**Professor**  
Department of Computer Science and Engineering  
Daffodil International University

### **Submitted by:**

---

**Md Tasluf Morshed**  
ID: -191-15-12089  
Department of Computer Science and Engineering  
Daffodil International University

---

**Md Assadujjaman Tilok**  
ID: -191-15-12594  
Department of Computer Science and Engineering  
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**, 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 in our country are using 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. Which is a major problem of this industry and 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

<b>CONTENTS</b>	<b>Page No.</b>
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
<b>CHAPTER</b>	
<b>CHAPTER 1: INTORDUCTION</b>	<b>1-5</b>
1.1 Background of the Project	1
1.2 Motivation of the Project	2
1.3 Problem Statement	2
1.4 Aim of the Project	3
1.5 Methodology	3
1.6 Proposed Solution	4
1.7 Conclusion	5
<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>6-8</b>
2.1 Introduction	6
2.2 Literature Review	6
2.3 Comparative Analysis	6
2.4 Scope of the Problem	7
2.5 Challenges	8
2.6 Conclusion	8
<b>CHAPTER 3: REQUIREMENT SPECIFICATION</b>	<b>9-40</b>
3.1 Introduction	9
3.2 Business Process Modeling	9
3.3 Requirement Collection and Analysis	10

3.4 Use Case Modeling and Description	14
3.5 Activity Diagram	21
3.6 Sequence Diagram	23
3.7 Design Requirement	39
3.8 Conclusion	39
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>40-62</b>
4.1 Introduction	40
4.2 Front-end Design	40
4.3 Back-end Design	59
4.4 Interaction Design and User Experience (UX)	62
4.5 Implementation Requirements	62
4.6 Conclusion	62
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>63-66</b>
5.1 Introduction	63
5.2 Implementation of Database	63
5.3 Implementation of Front-end Design	63
5.4 Testing Implementation	63
5.5 Test Results and Reports	64
5.6 Conclusion	66
<b>Chapter 6: Impact on Society, Environment and Sustainability</b>	<b>67-69</b>
6.1 Introduction	67
6.2 SWOT Analysis	67
6.3 Impact on Society	68
6.4 Impact on Environment	68
6.5 Ethical Aspects	69
6.6 Sustainability Plan	70

6.7 Conclusion	70
<b>CHAPTER 7: CONCLUSION</b>	<b>71-72</b>
7.1 Conclusion	71
7.2 Further Suggested Work	72
<b>REFERENCES</b>	<b>73</b>
<b>APPENDIX</b>	<b>A1-G1</b>
Appendix A: Report Layout	A1
Appendix B: Design Resource	B1
Appendix C: Main App Resource	C1-C2
Appendix D: Project GitHub Links	D1
Appendix E: REST API	E1-E2
Appendix F: Backend Integration Process	F1
Appendix G: Plagiarism	G1

## List of Figures

<b>Figures</b>	<b>Page No.</b>
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
Figure 49 Website Order Page	57

Figure 50 Website Order Details Page	58
Figure 51 Website Profile Page	58
Figure 52 All Schema for database	59
Figure 53 User Table for database	59
Figure 54 Agent Table for database	60
Figure 55 Technician Table for database	60
Figure 56 Address Table for database	61
Figure 57 Order Table for database	61
Figure 58 Product Table for database	62
Figure 59 Automated Test Figure 01	66
Figure 60 Automated Test Figure 02	66

## **List of Table**

<b>Table</b>	<b>Page No.</b>
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**

### **1.1 Background of the Project**

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, we use MongoDB.

## 1.2 Motivation of the Project

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 an 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. It is visible that, along with the rapid development of technology, the servicing system is not improving very speedily. Therefor we intendant to take the control of this situation.

Firstly, the main motivation comes from our honorable chairman sir of the Daffodil family, **Mr. Sabur khan**. Who said, “Don’t wait for a job, and Make your own job” that’s why we intended to work hard on a project. Secondly, we are connected with one of the service heads from Singer Bangladesh LTD. And we’ve observed closely what kinds of difficulty a customer, showroom manager, or service provider faced throughout the complete process. And we think this app can be the ultimate solution to the problem. Last but not least, recently I've faced some cooling issues on my AC which is out of warranty by March 2021. As a consequence, I've contacted local technicians who work three times on it, they take it from my house then days later back with it which is too much hassle for me but can't give me the solution.

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 Problem Statement**

Product Servicing 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. They have to face:

- 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.

### **1.4 Aim of the Project**

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

## 1.5 Methodology

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 apps in a short time period, one for android device and other-one for iOS. It's a very 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. After analysis our customer data and all the date we will use in our app, we have decided whether MongoDB would be a good support for our application.

## 1.6 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.

## **1.7 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.

## **CHAPTER 2**

## **LITERATURE REVIEW**

### **2.1 Introduction**

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. Here this app work with Internet connection from a locally hosted 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 for database and also use MongoDB.

### **2.2 Literature Review**

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 and 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 cannot 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 as locally. If users are more than 0.1 million then this app can be crush. 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 register 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 have to learn MongoDB. We faced several errors during connecting database and google map to connect to the application.

## **2.6 Calculation**

This Project “**Android Based Electronic Product Servicing System**” is an optimal servicing solution for Home appliance commodity. After researching and reviewing several work and analyzing current market status, we can developed the site as customer can hire technician and experts based on their product requirement in order to fix the item.

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1 Introduction

In a software, requirements specification refers to the description of necessary tools that are required to develop that system. SRS is a lays out that declare the functional and non-functional requirements, as well include a set of use cases diagram that describe user interactions that the software.

### 3.2 Business Process Modeling

Business Process Model for our system are given below

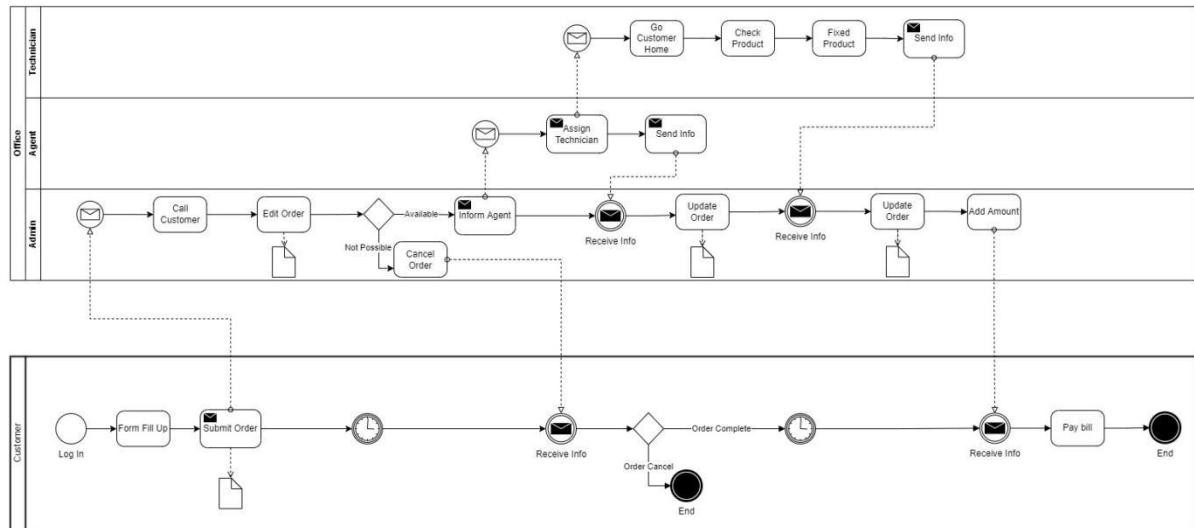


Figure 01: Business process model

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.

### **3.3 Requirement Collection and Analysis**

#### **3.3.1 Functional Requirements**

##### **3.3.1.1 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.

### **3.3.1.2 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 valid password will be provided after user registration is confirmed.
- Password must be hidden from others customer while typing it in the field.
- A google login button should be implemented to log in by Google account.

#### **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.

## **3.3.2 Non-Functional Requirements**

### **3.3.2.1 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.

### **3.3.2.2 Usability requirements**

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

### **3.3.2.3 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.

#### **3.3.2.4 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.

#### **3.3.2.5 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.

#### **3.3.2.6 Reliability Requirement**

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

### 3.4 Use Case Diagram

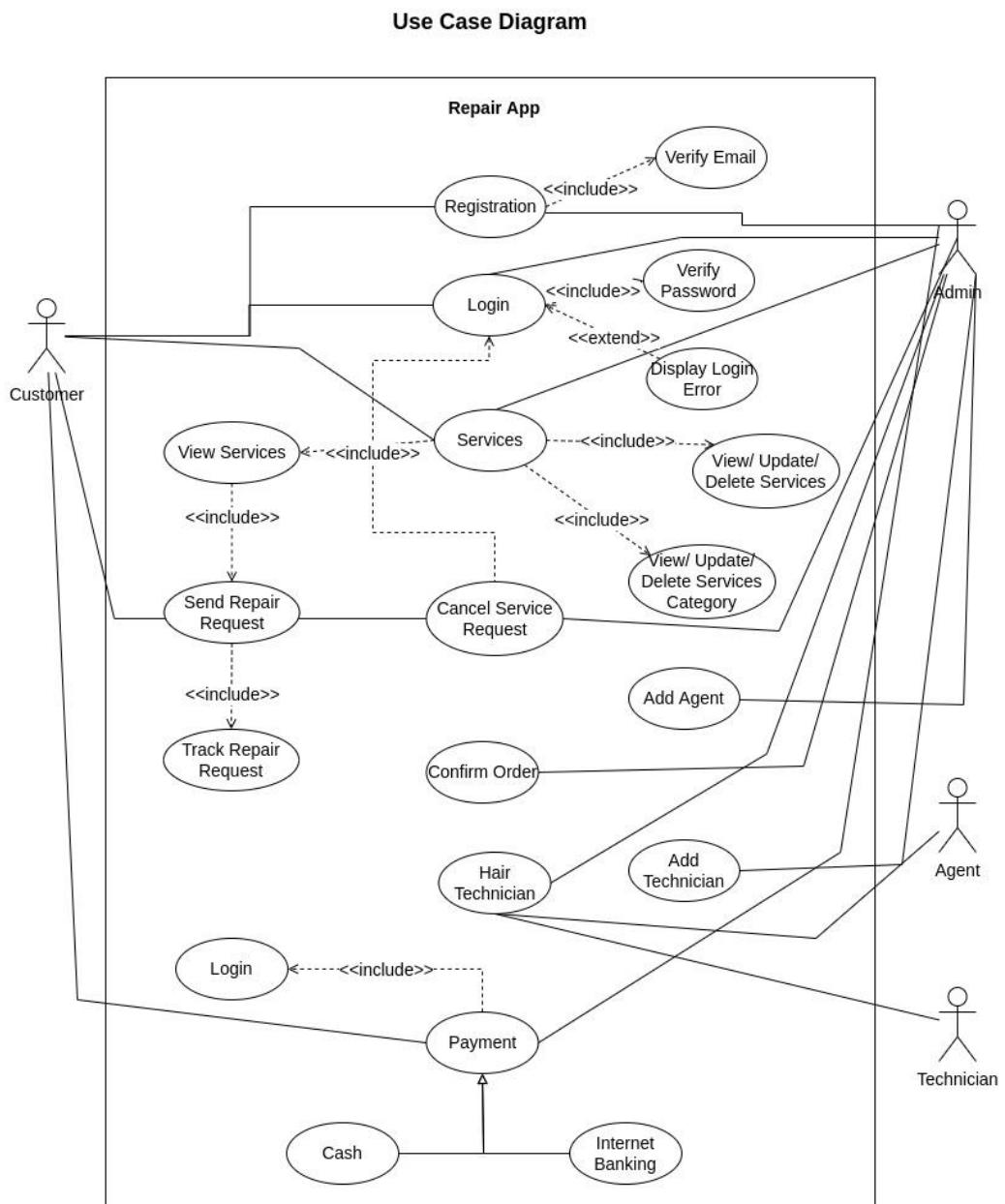


Figure 02: Use Case Diagram

### 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)

<b>Use Case ID</b>	<b>UC2</b>
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)

<b>Use Case ID</b>	<b>UC3</b>
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)

<b>Use Case ID</b>	<b>UC4</b>
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)

<b>Use Case ID</b>	<b>UC5</b>
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)

<b>Use Case ID</b>	<b>UC6</b>
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)

<b>Use Case ID</b>	<b>UC7</b>
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)

<b>Use Case ID</b>	<b>UC8</b>
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)

<b>Use Case ID</b>	<b>UC9</b>
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)

### 3.5 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.

#### 3.5.1 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.

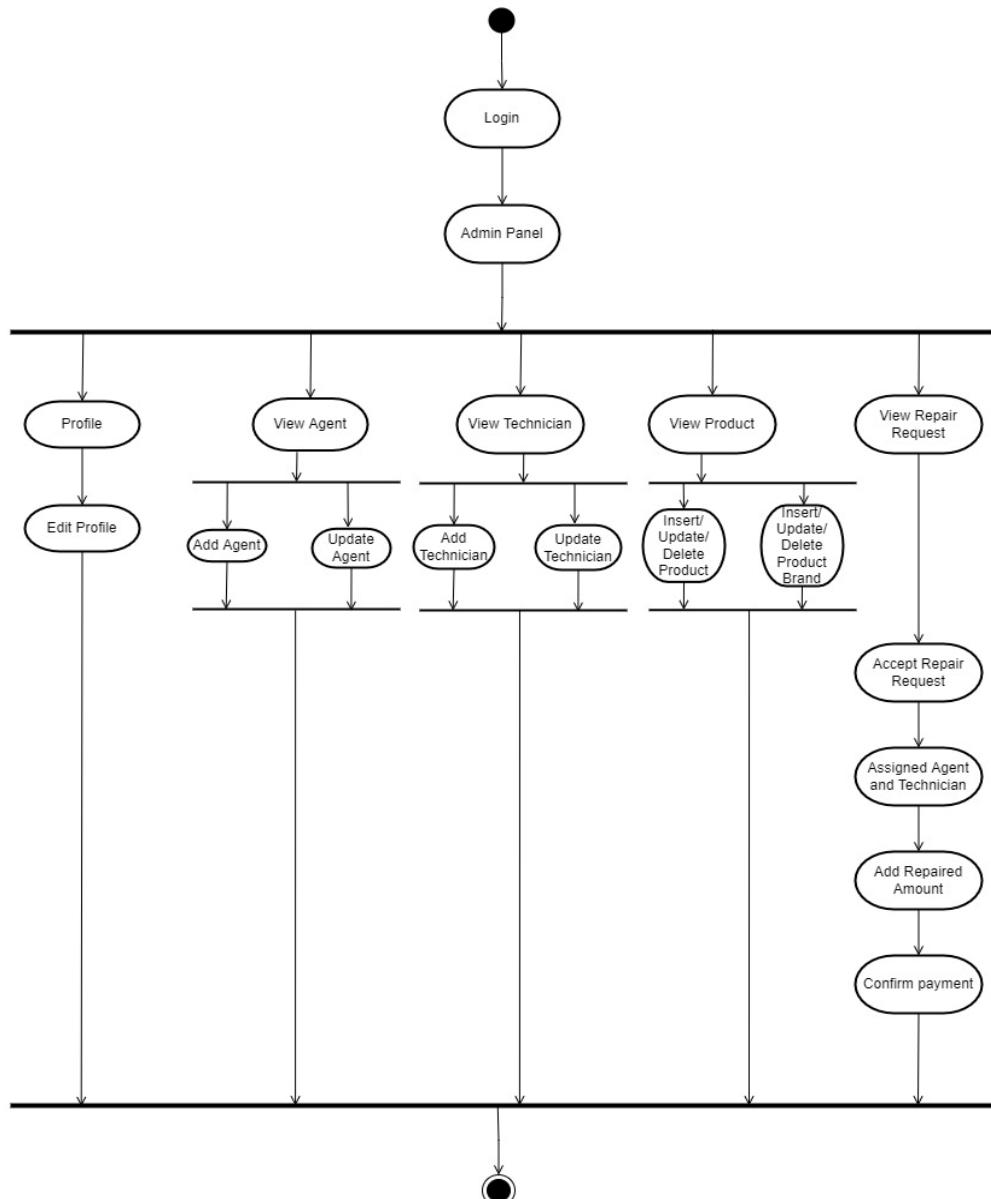


Figure 03: Admin Activity Diagram

### 3.5.2 Customer Activity Diagram

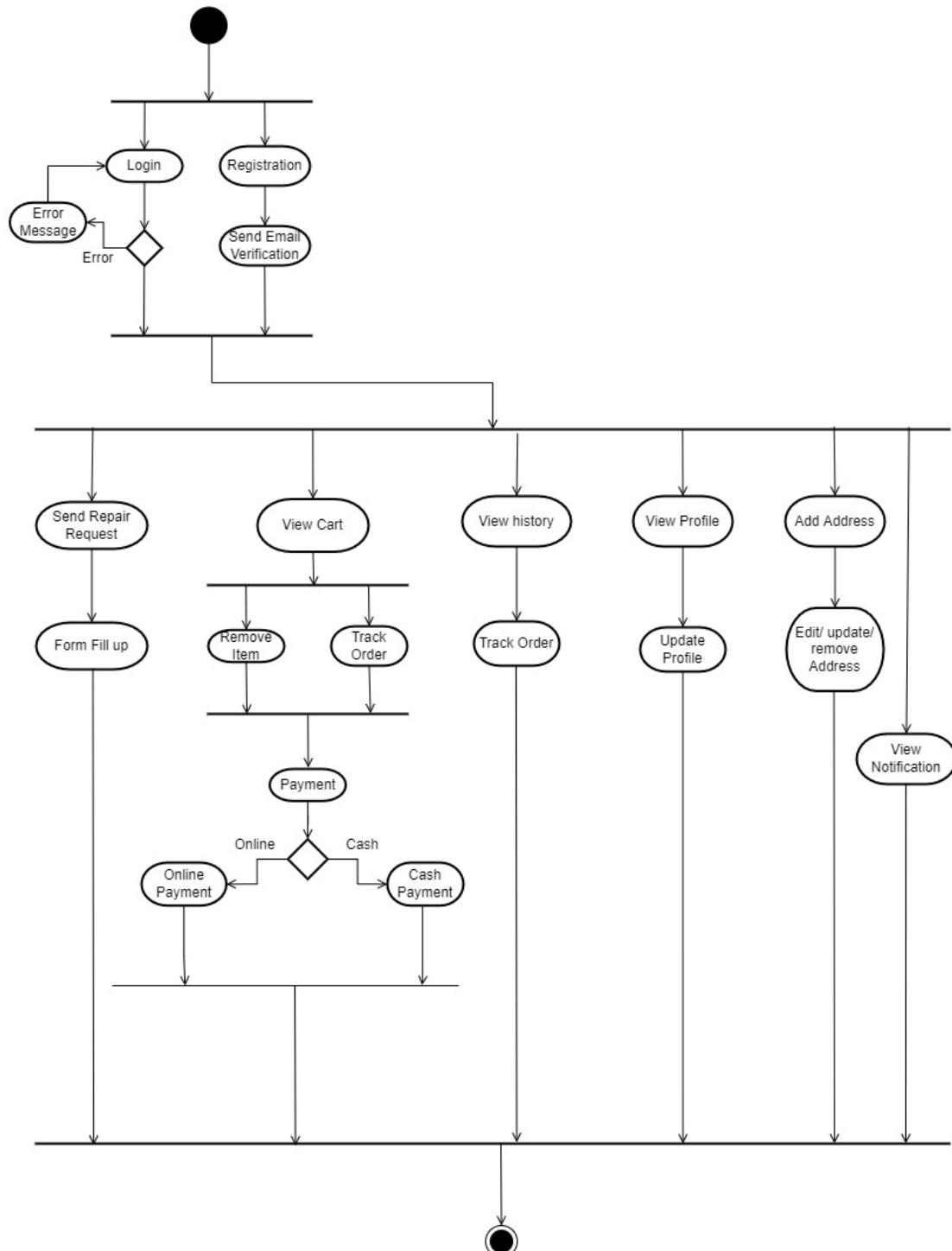


Figure 04: Customer Activity Diagram

### 3.6 Sequence Diagram

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

#### 3.6.1 Admin Registration

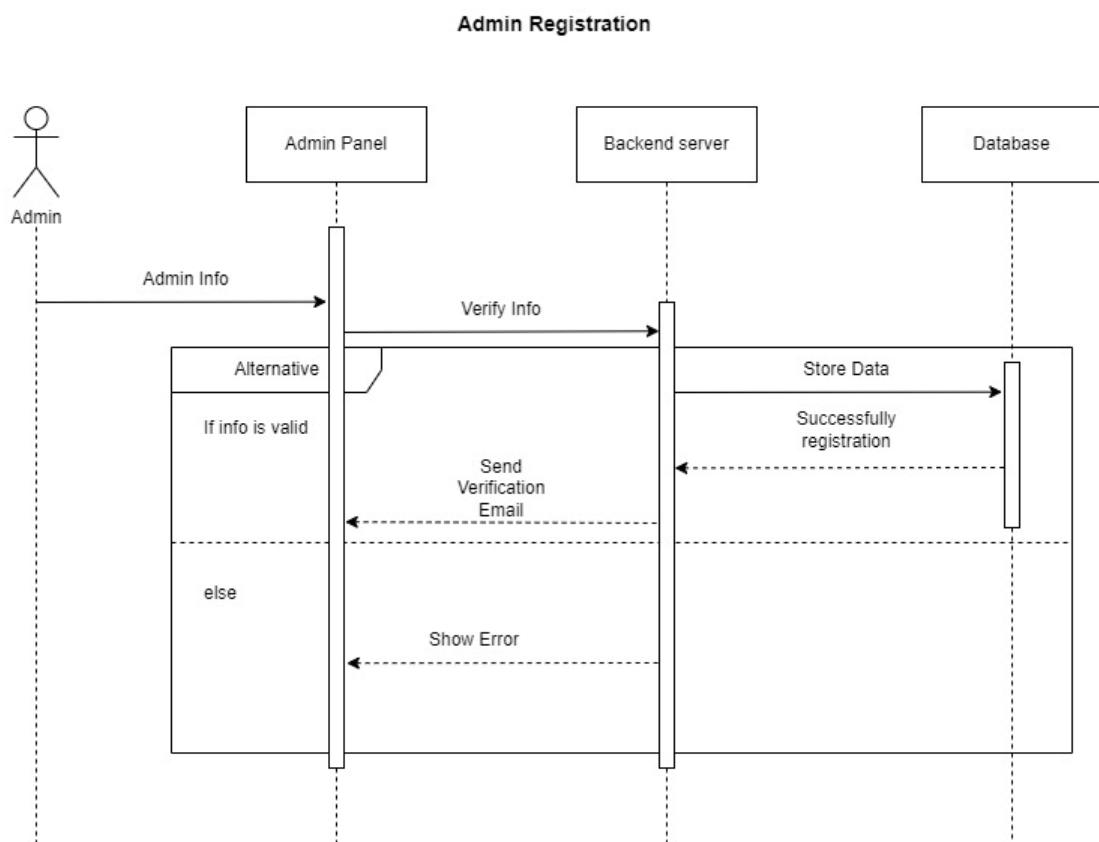


Figure 05: Admin Registration Sequence Diagram

### 3.6.2 Admin Login

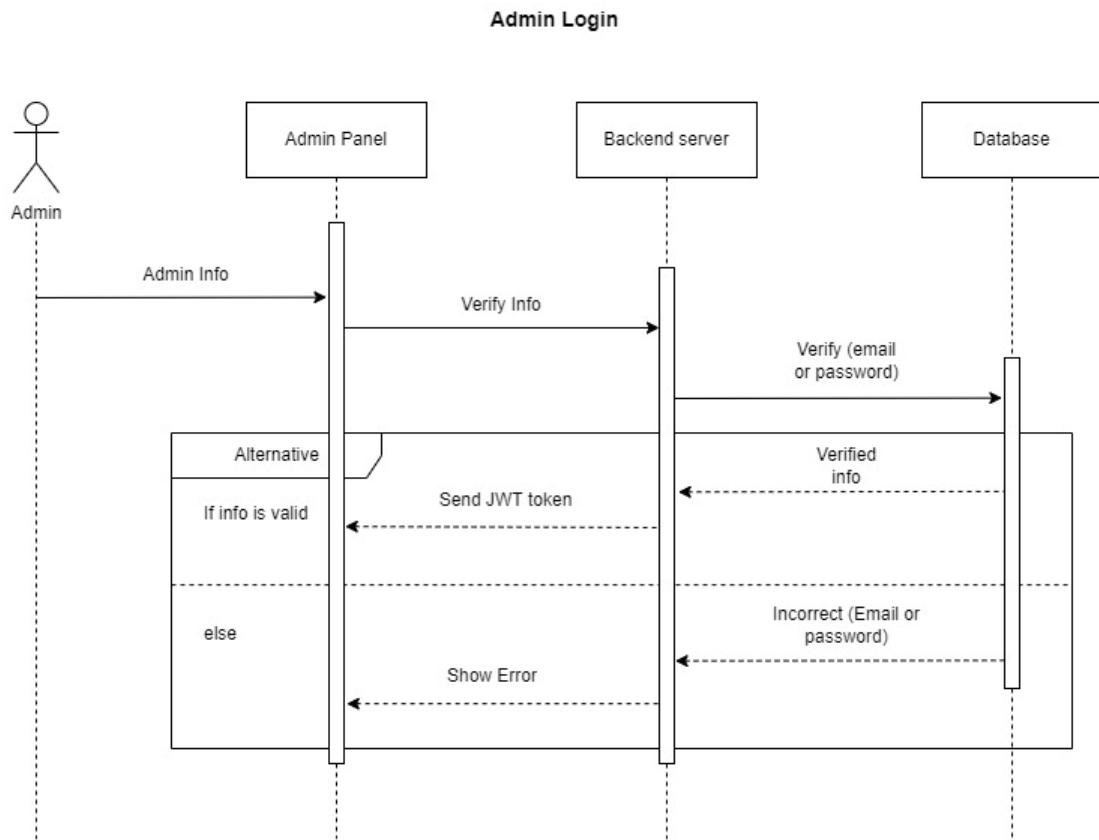


Figure 06: Admin Login Sequence Diagram

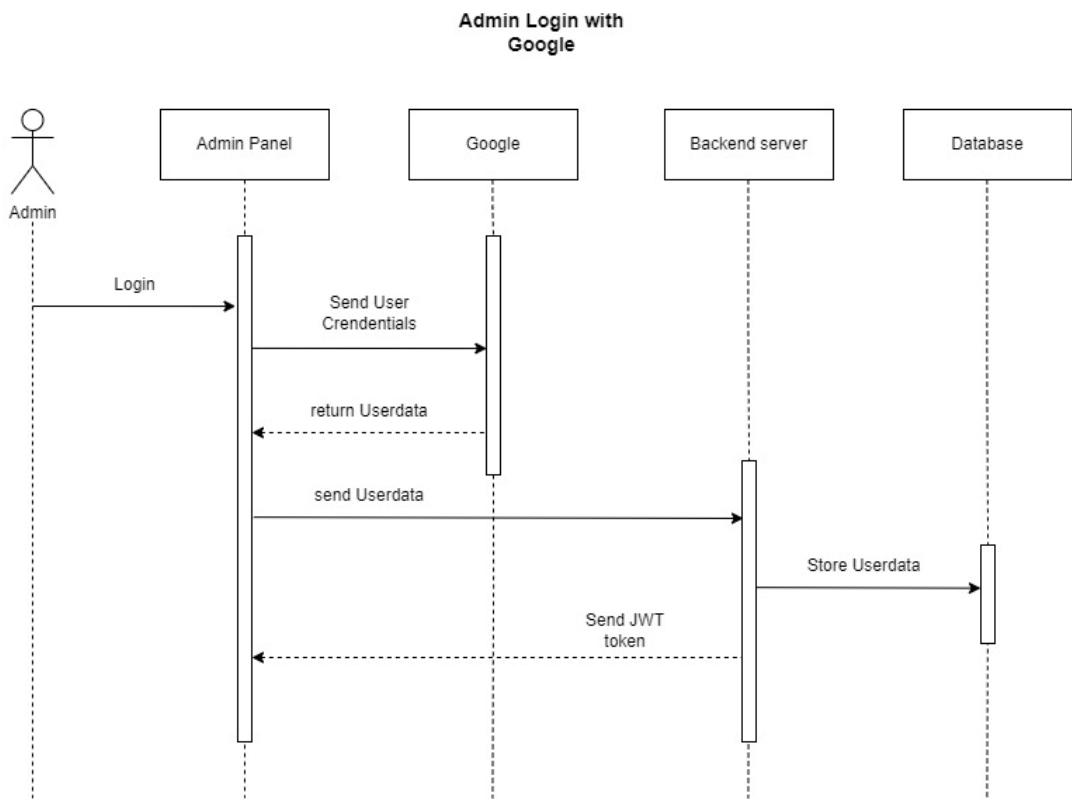


Figure 07: Admin Login with Google Sequence Diagram

### 3.6.3 Admin Profile

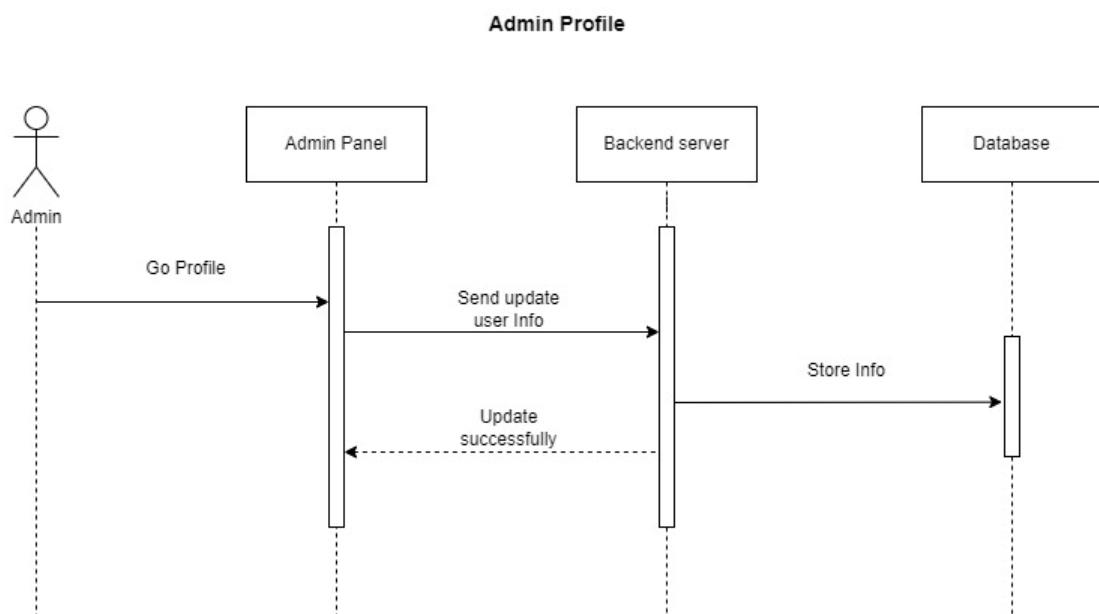


Figure 08: Admin Profile Sequence Diagram

### 3.6.4 Admin Forget Password

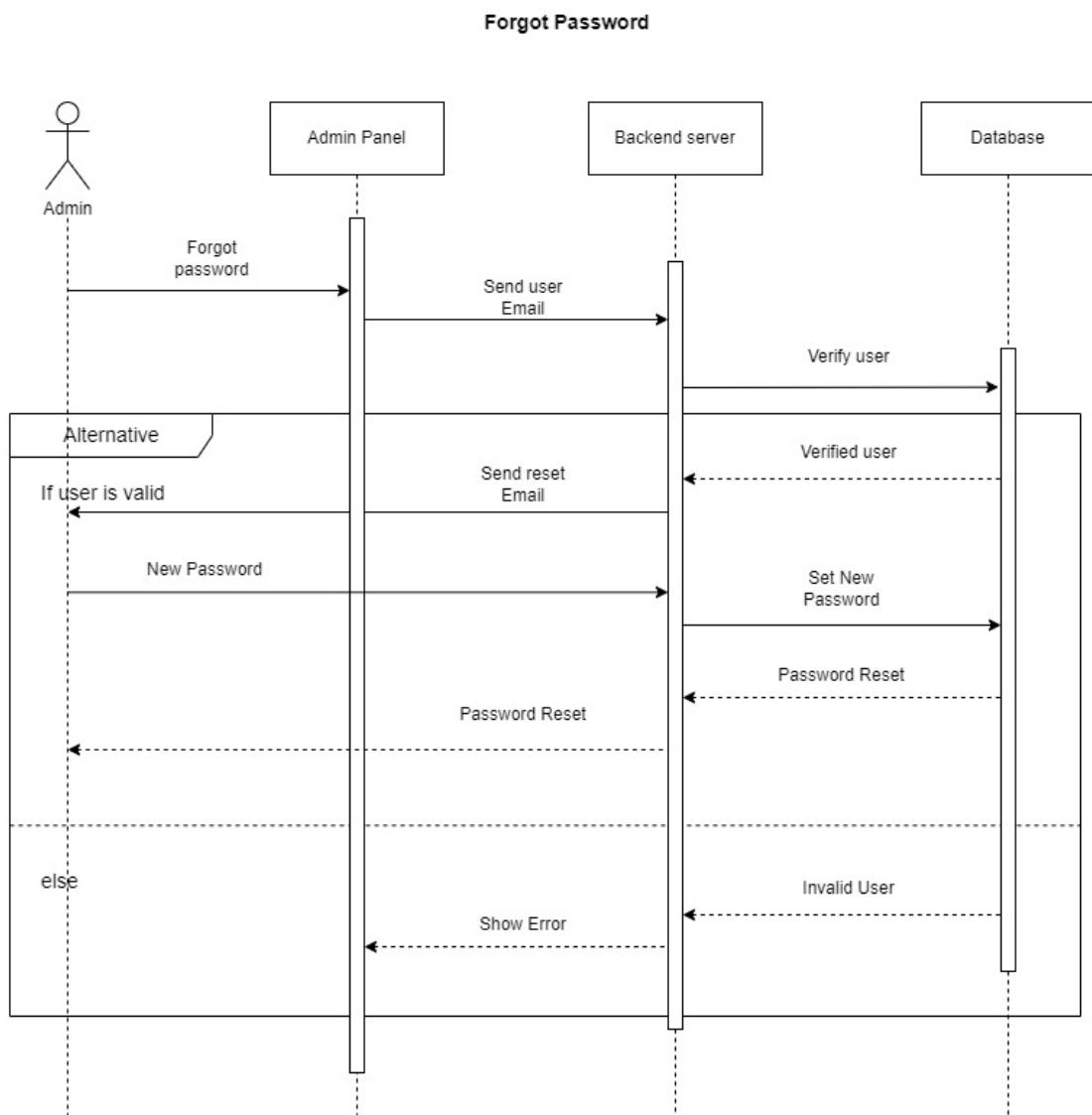


Figure 09: Admin Forget Password Sequence Diagram

### 3.6.5 Admin View Order

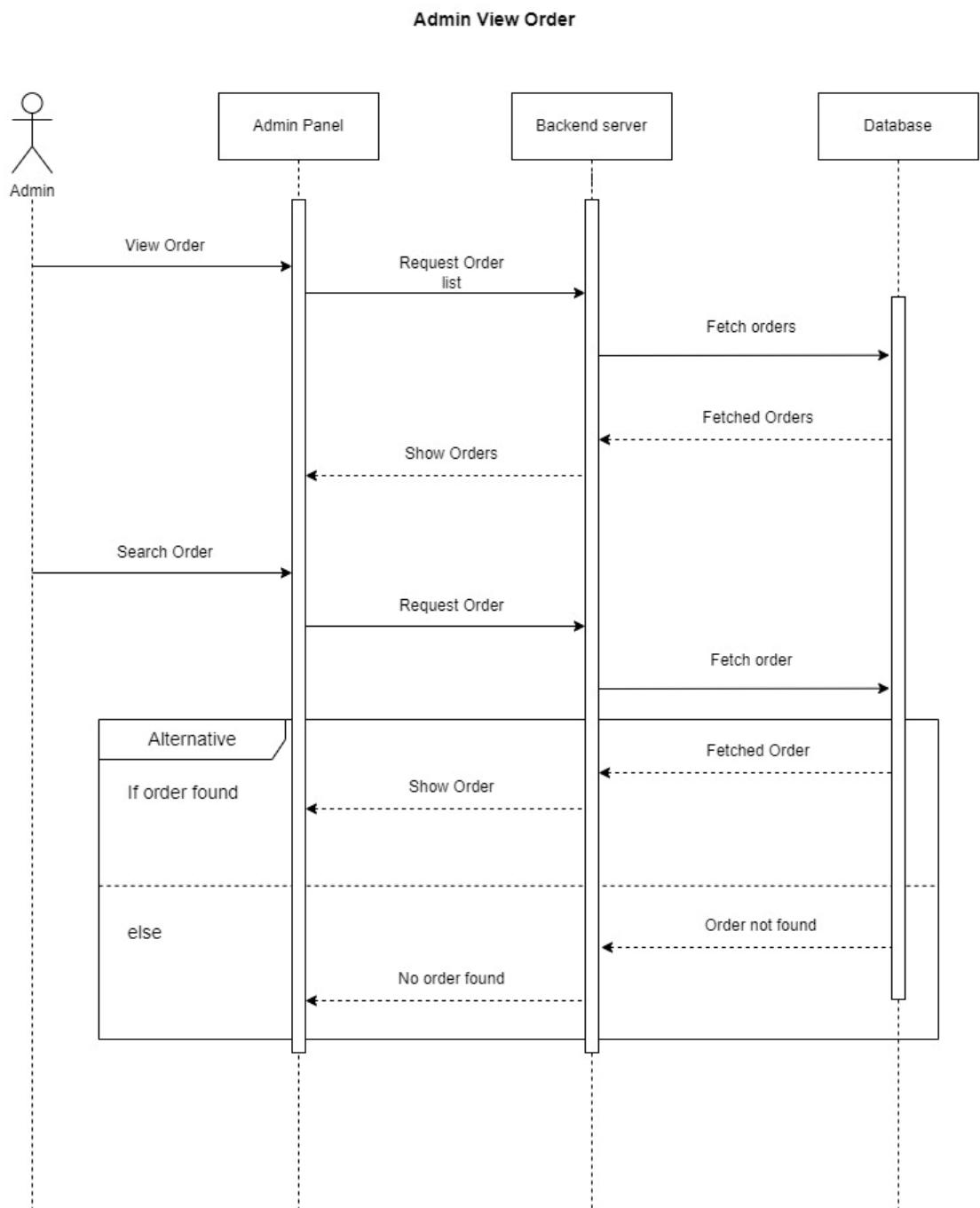


Figure 10: Admin View Order Sequence Diagram

### 3.6.6 Admin Add Agent

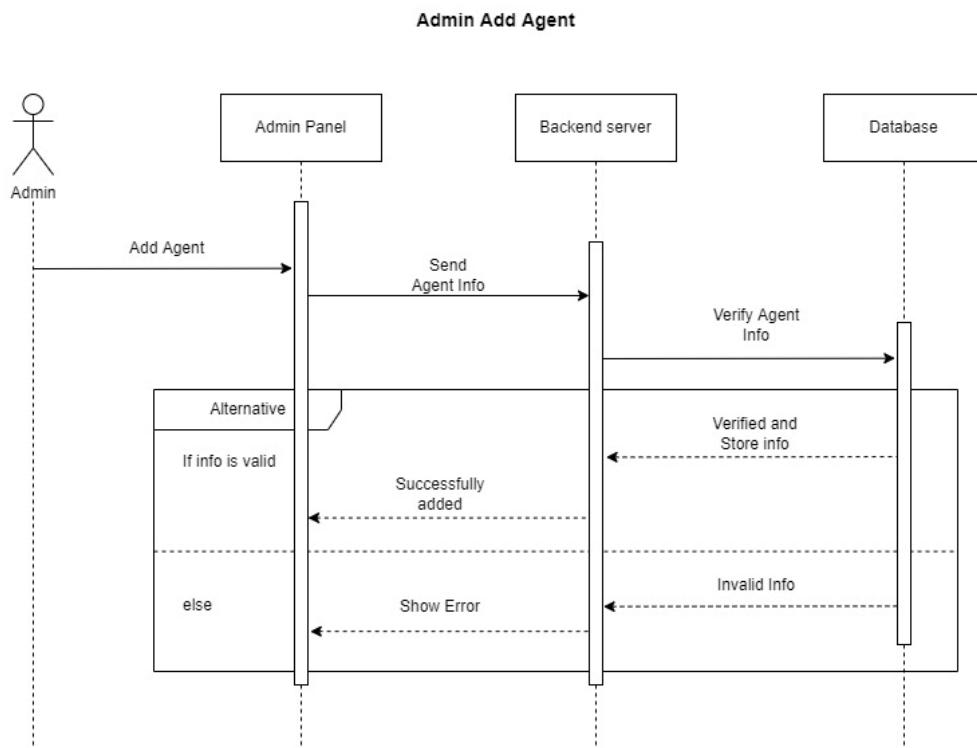


Figure 11: Admin Add Agent Sequence Diagram

### 3.6.7 Admin Add Technician

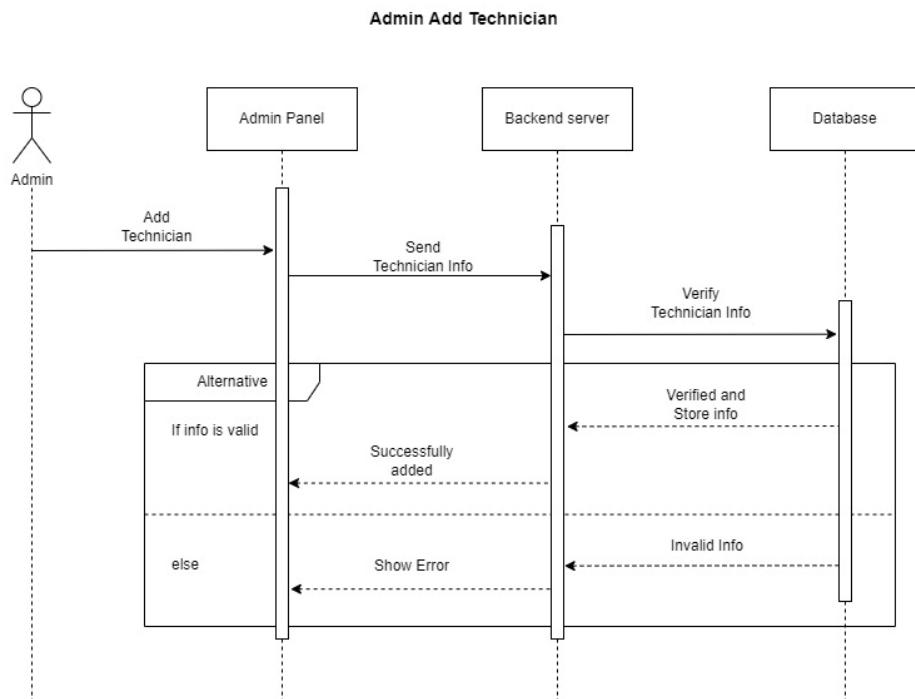


Figure 12: Admin Add Technician Sequence Diagram

### 3.6.8 Admin Update Order

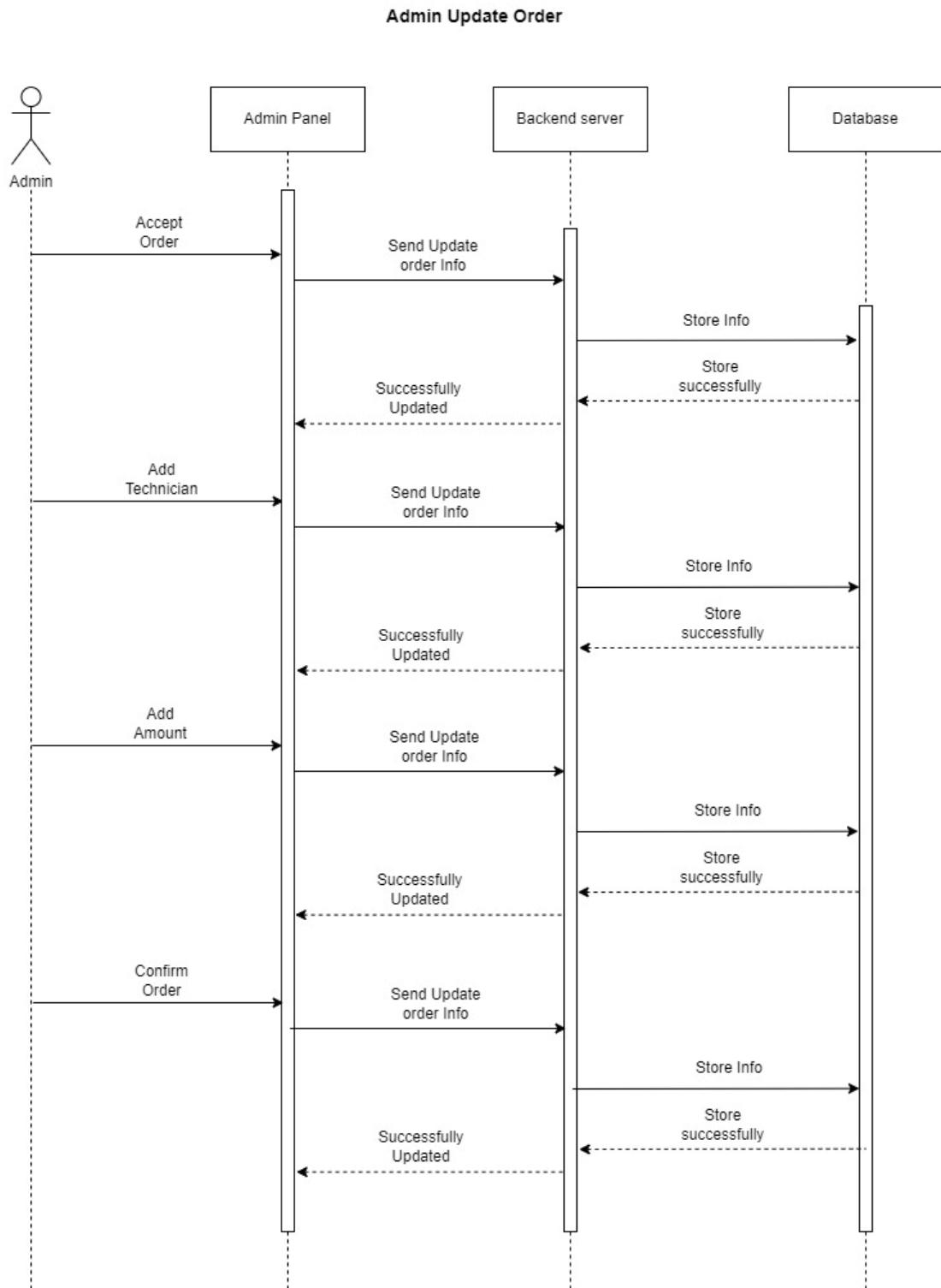


Figure 13: Admin Update Order Sequence Diagram

### 3.6.9 Admin Update Agent

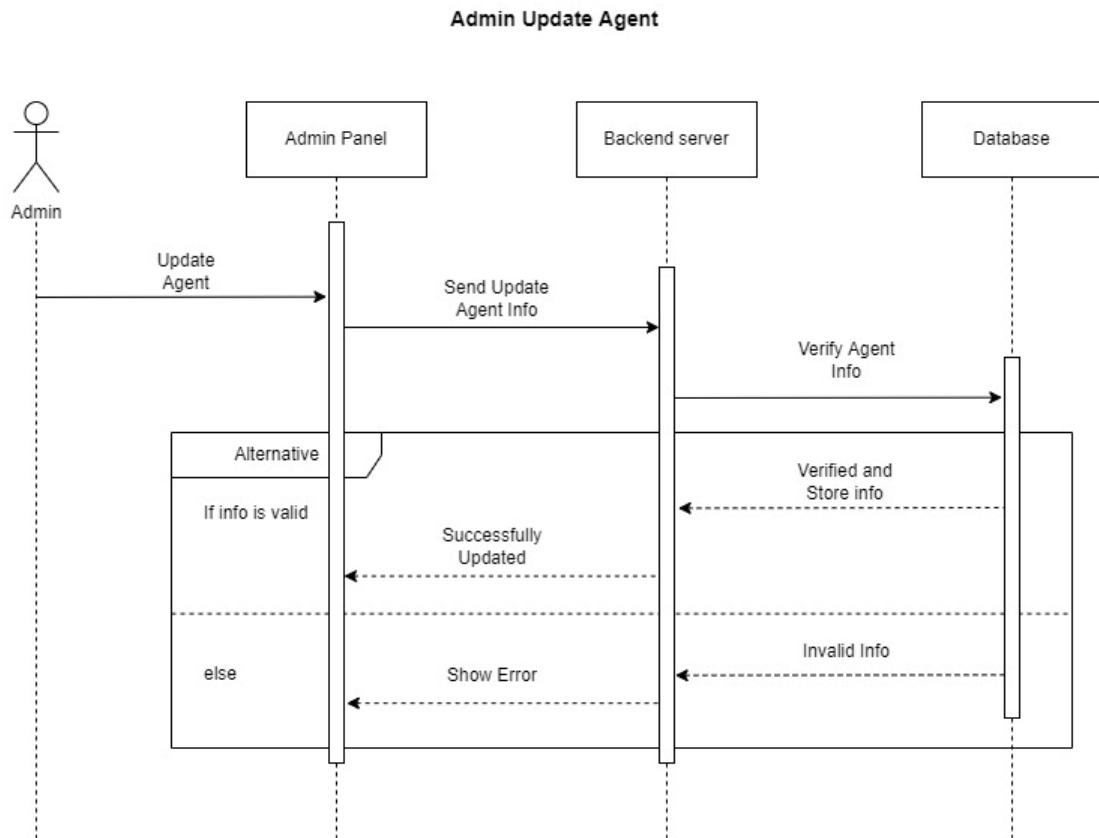


Figure 14: Admin Update Agent Sequence Diagram

### 3.6.10 Admin Update Technician

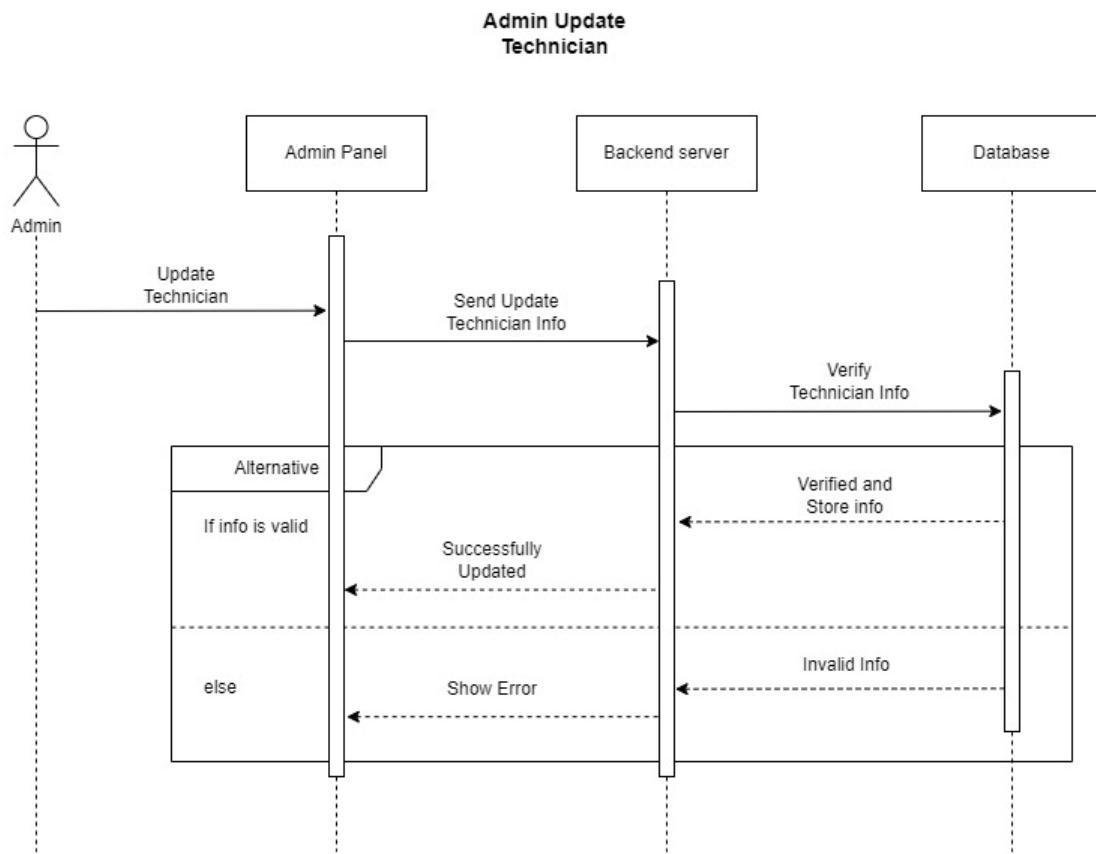


Figure 15: Admin Update Technician Sequence Diagram

### 3.6.11 Customer Registration

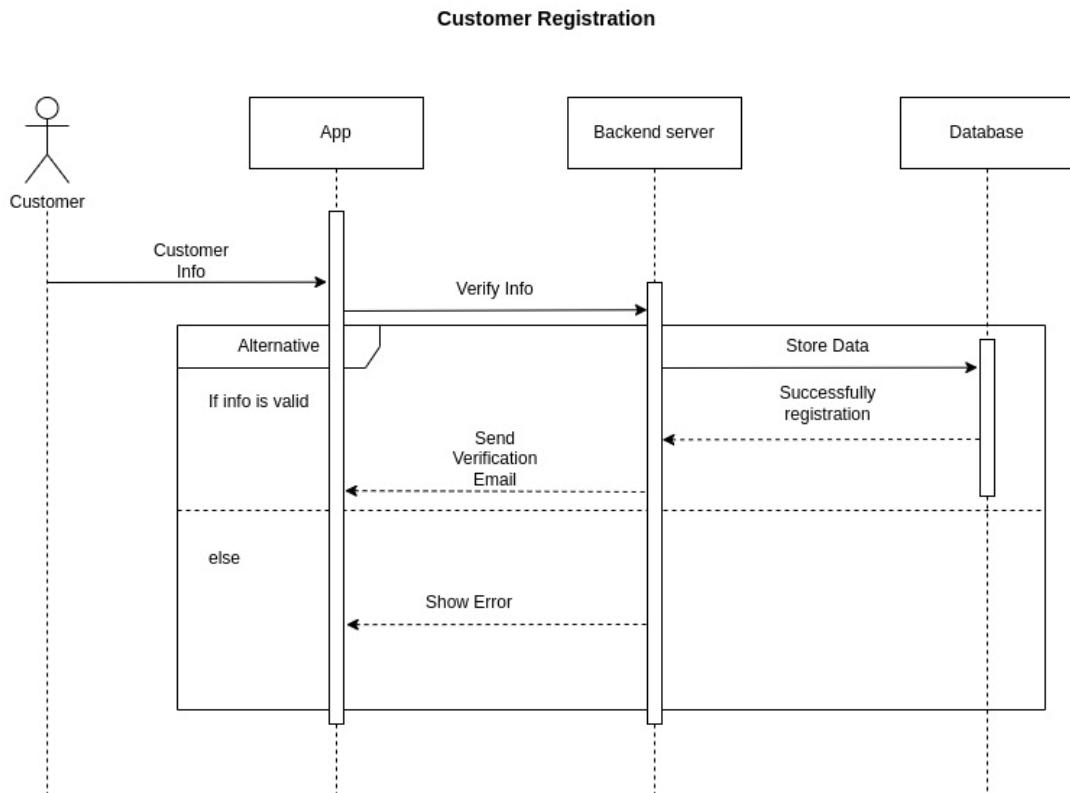


Figure 16: Customer Registration Sequence Diagram

### 3.6.12 Customer Login

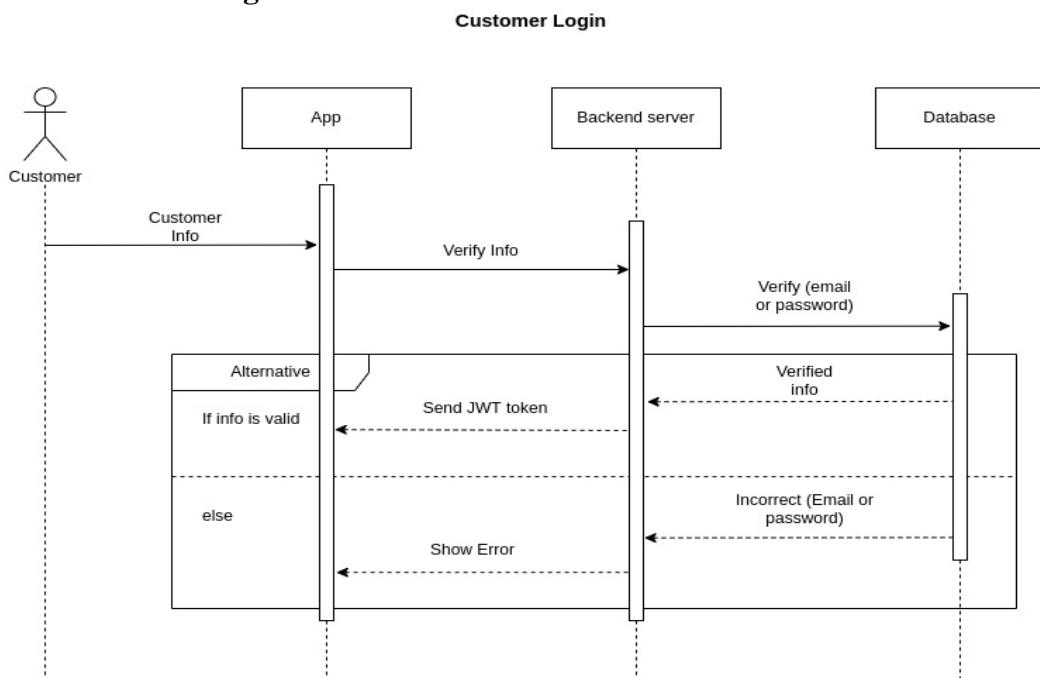


Figure 17: Customer Login Sequence Diagram

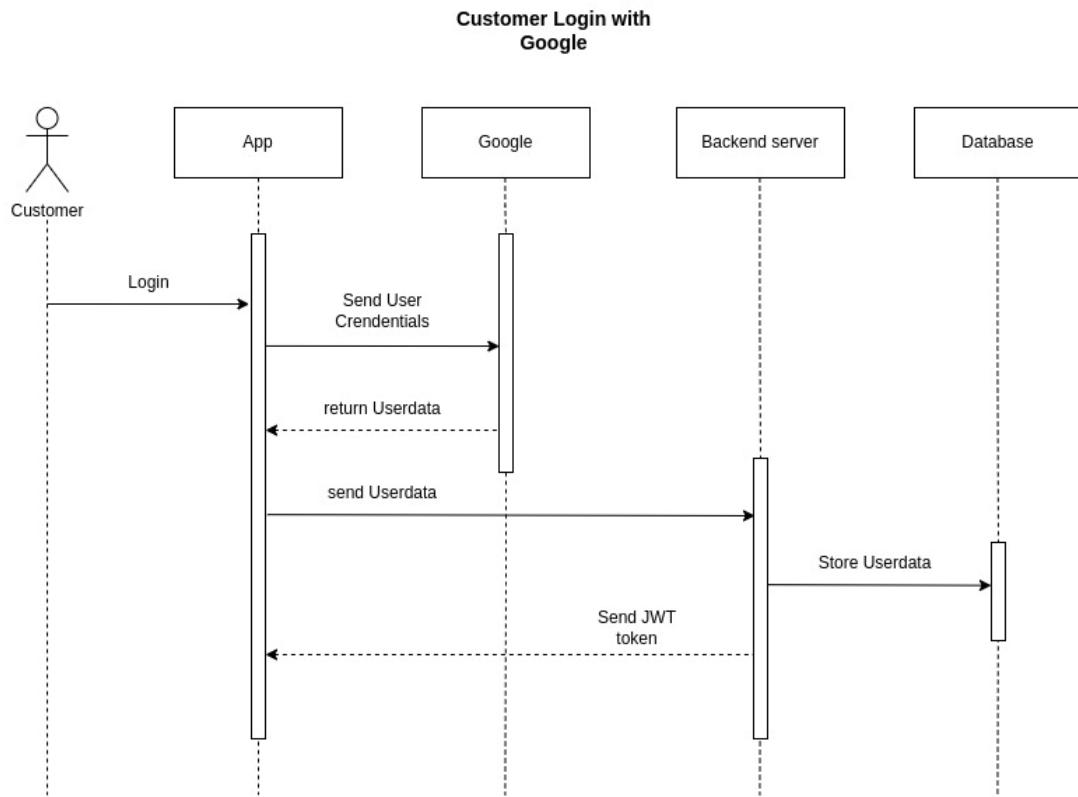


Figure 18: Customer Login with Google Sequence Diagram

### 3.6.13 Customer Profile

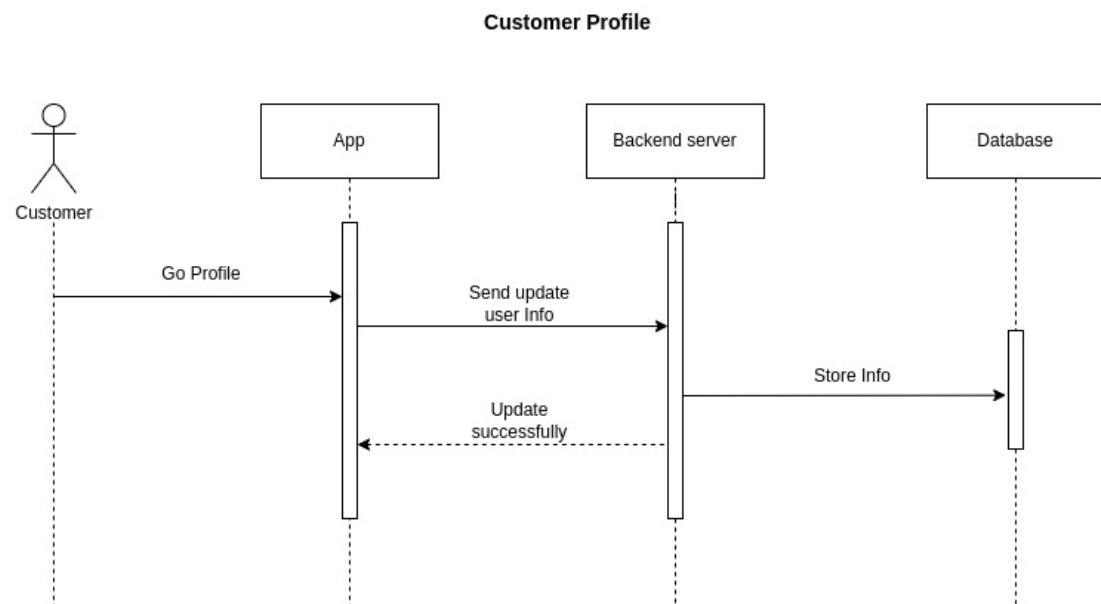


Figure 19: Customer Profile Sequence Diagram

### 3.6.14 Forget Password

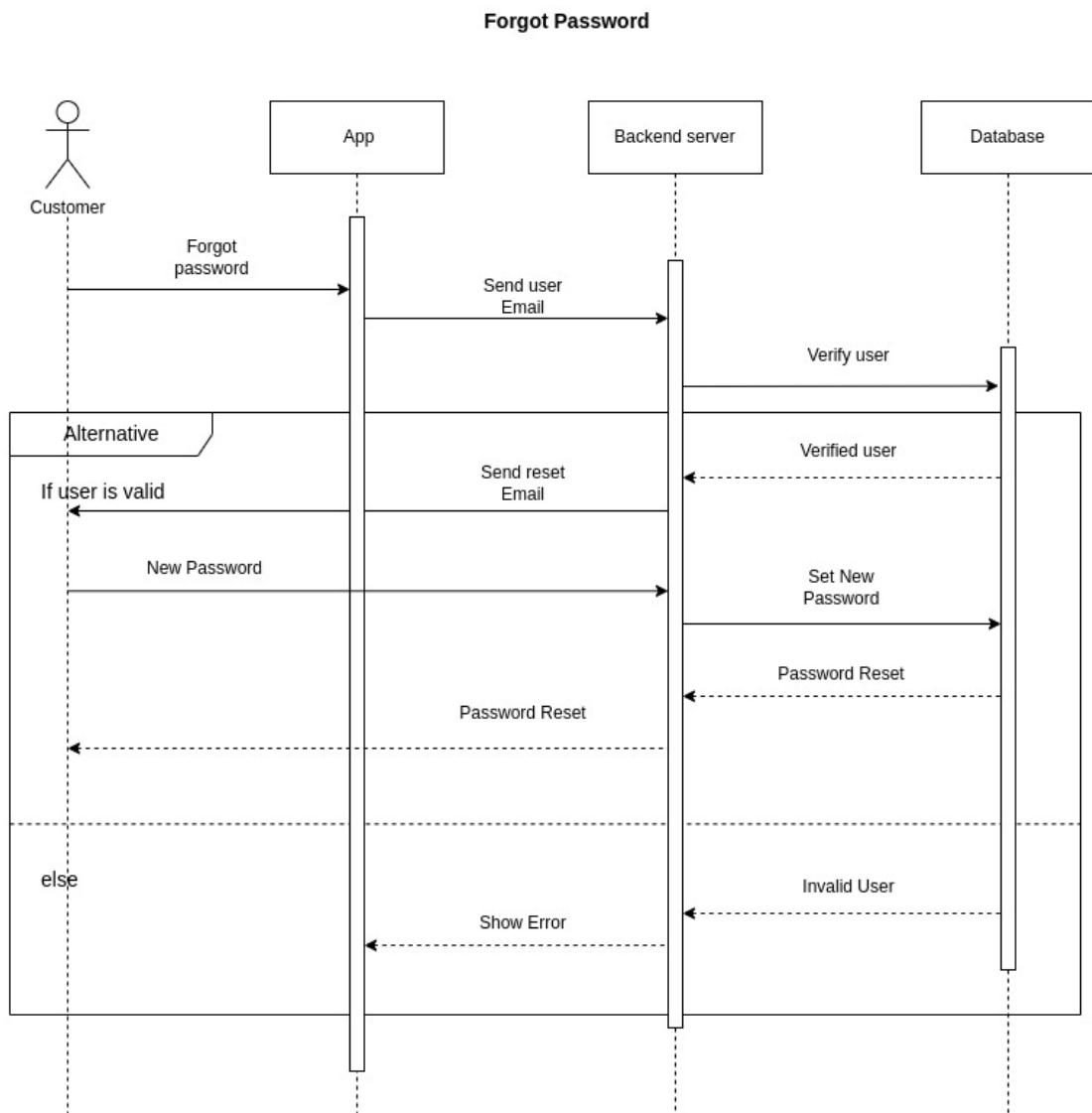


Figure 20: Customer Forget Password Sequence Diagram

### 3.6.15 Customer Add Order

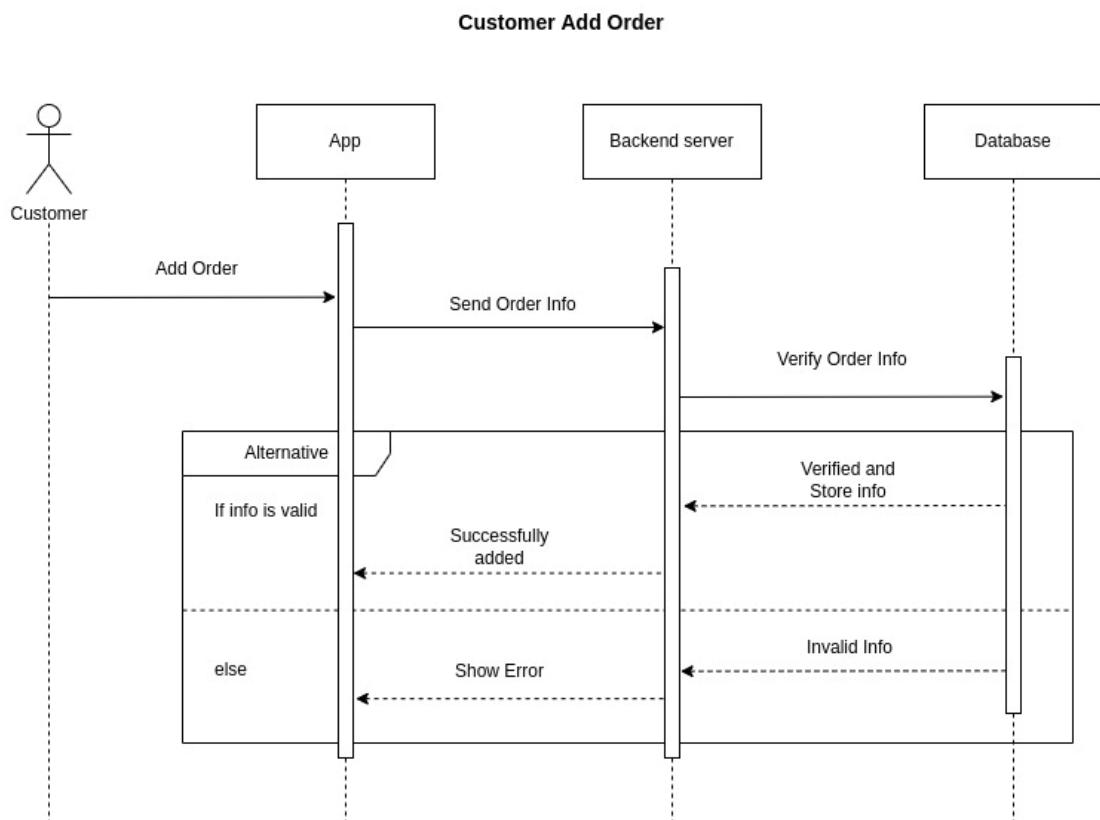


Figure 21: Customer Add Order Sequence Diagram

### 3.6.16 Customer Add Address

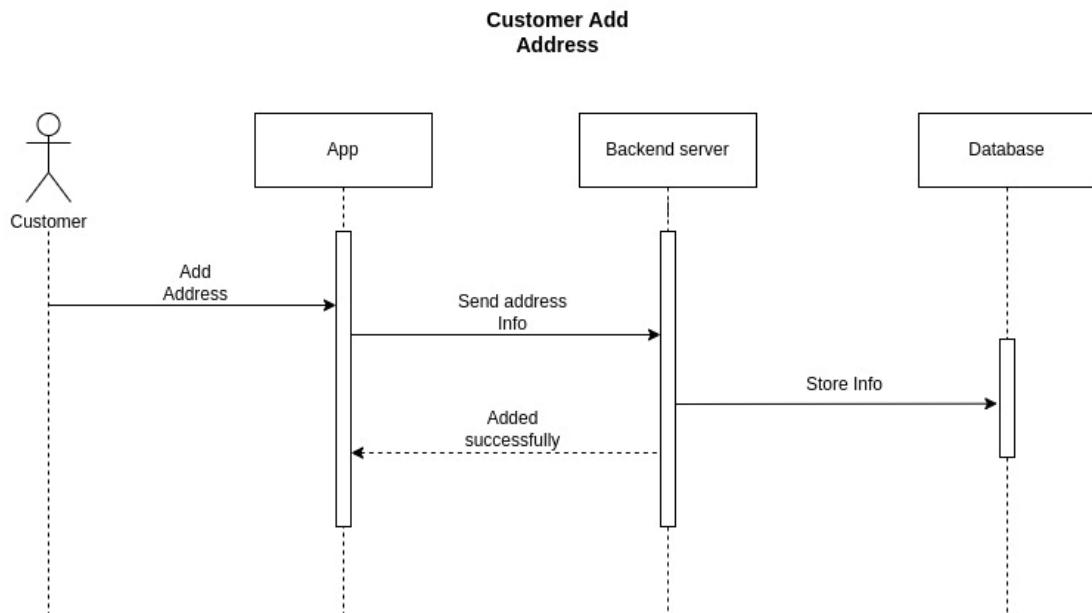


Figure 22: Customer Add Address Sequence Diagram

### 3.6.17 Customer Update Address

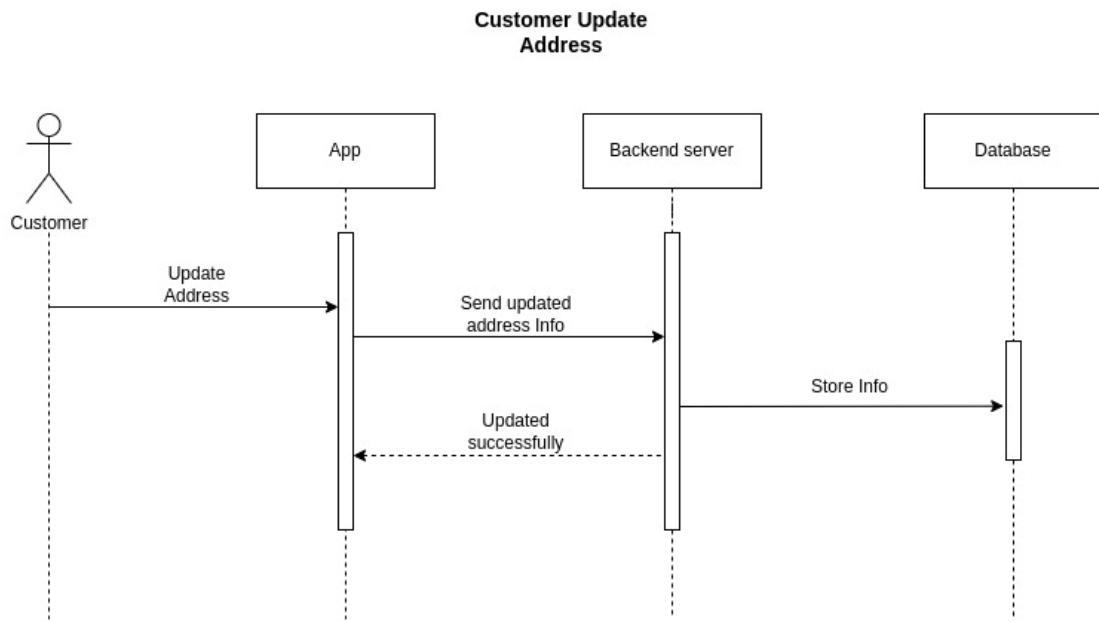


Figure 23: Customer Update Address Sequence Diagram

### 3.6.18 Cart List

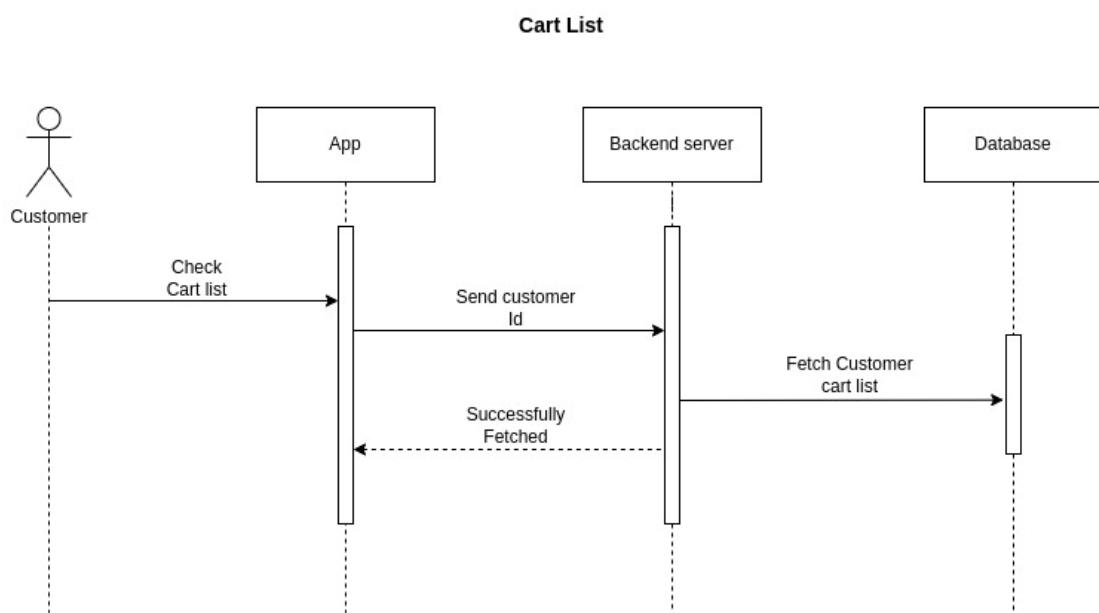


Figure 24: Cart List Sequence Diagram

### 3.6.19 Track Order

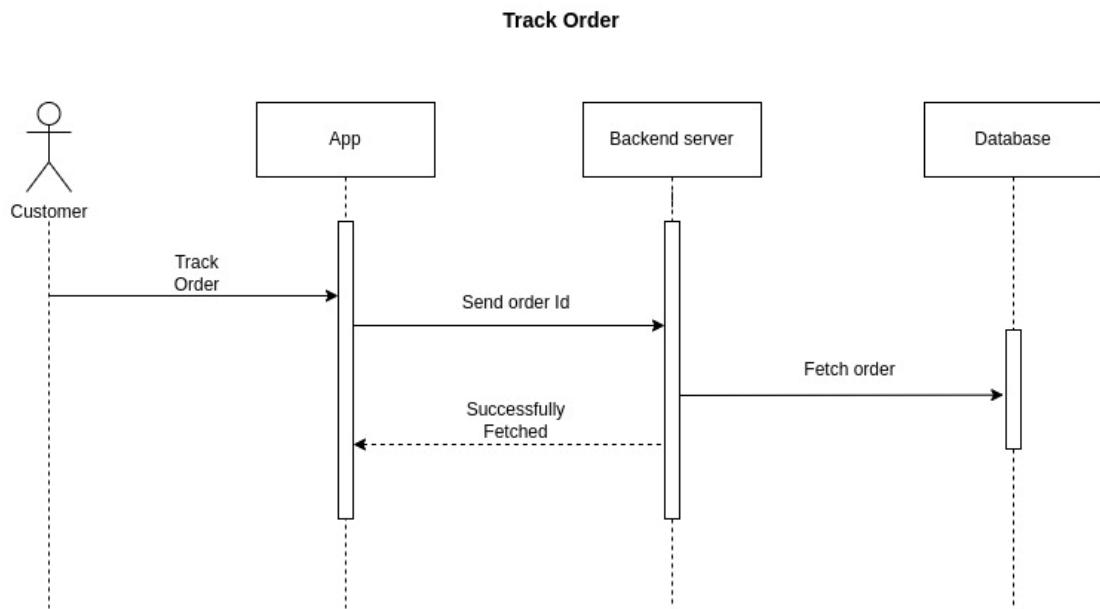


Figure 25: Track Order Sequence Diagram

### 3.6.20 Order History

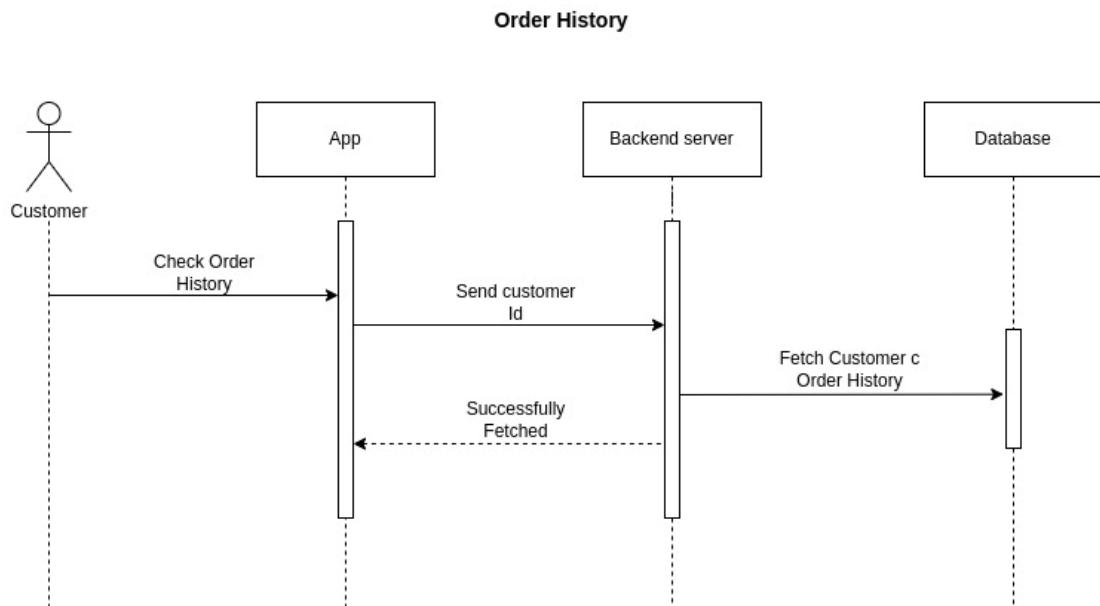


Figure 26: Order History Sequence Diagram

### 3.6.21 Payment

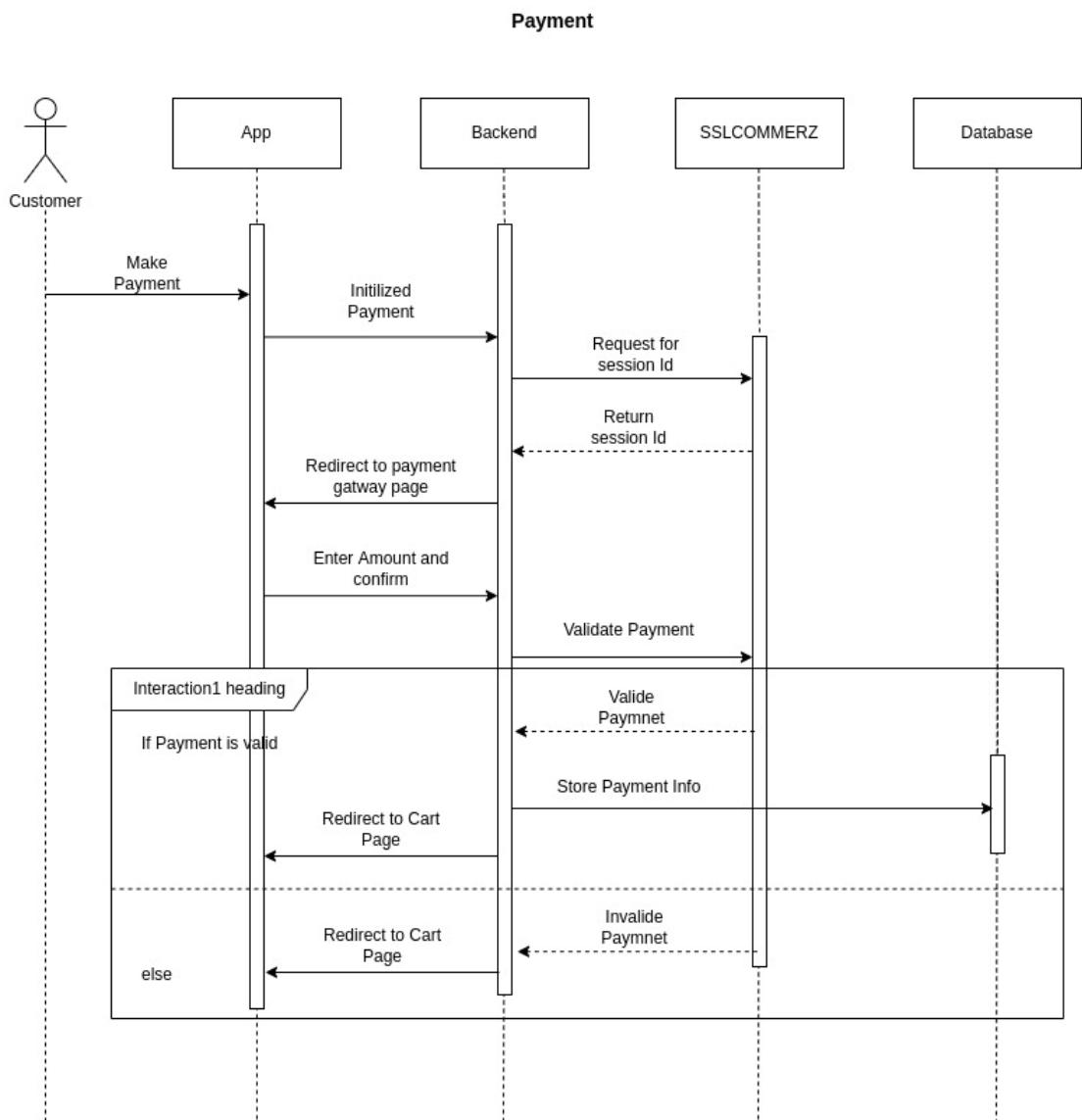


Figure 27: Payment Sequence Diagram

### **3.7 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

### **3.8 Conclusion**

To develop a project, developer needs to have clear and thorough understanding about necessary requirement and functionalities. SRS provide a realistic basic estimating costs, risks, and schedules of a project. From the above declaration, we can see a clear picture of our project.

## CHAPTER 4

# DESIGN SPECIFICATION

### 4.1 Introduction

Software design specification is the visual illustration of written document, that can explains a software functionality and user interaction from both (developer and user) end. Design specification elaborate the details of a project that, how the software will accomplish and how it will function including project overview, main goals, and target audience etc.

### 4.2 Front End Design

#### 4.2.1 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.

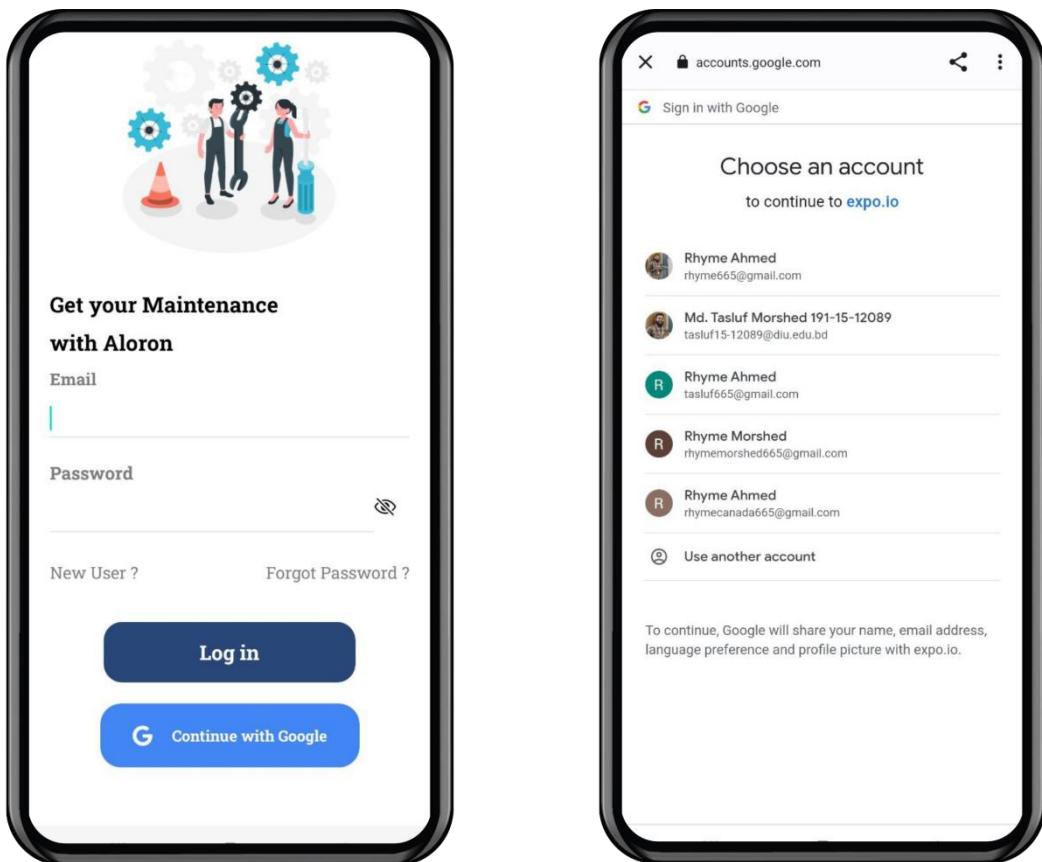


Figure 29: Login Page

## 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.

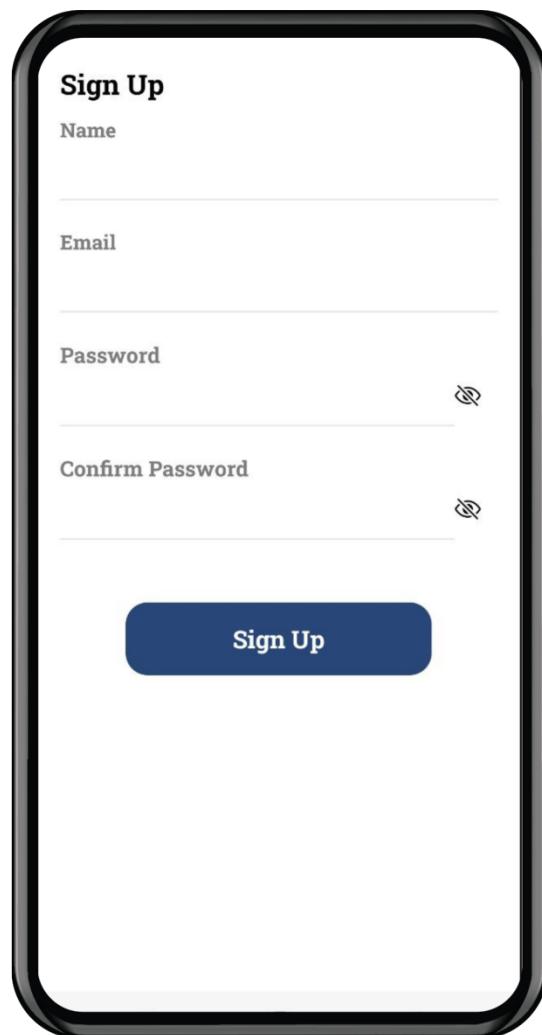


Figure 30: Sign Up Page

## Home Page

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

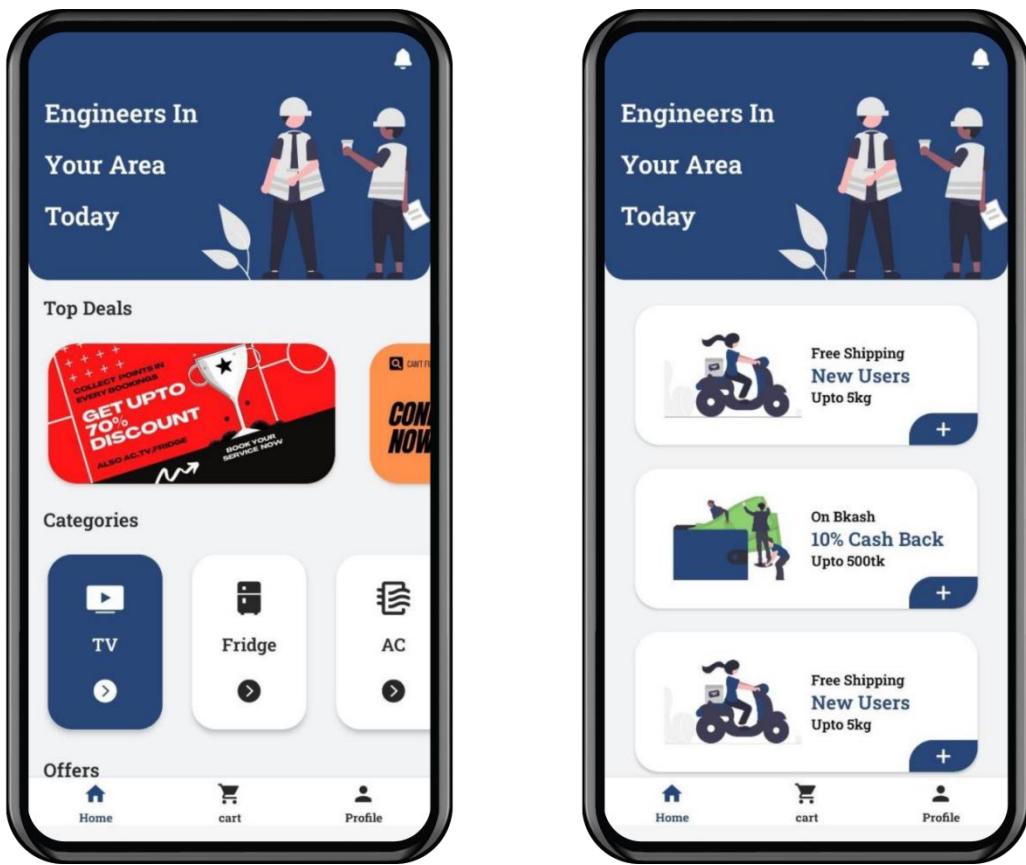


Figure 31: Home Page

## 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.

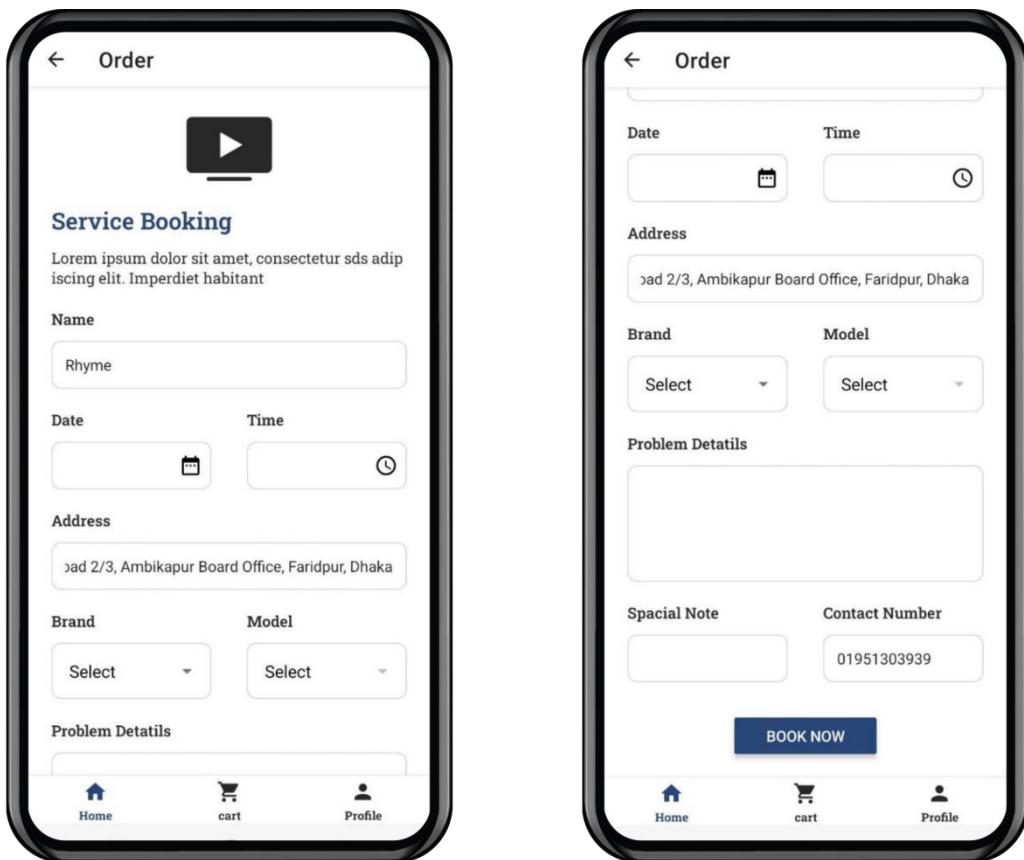


Figure 32: Order Page

## 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.

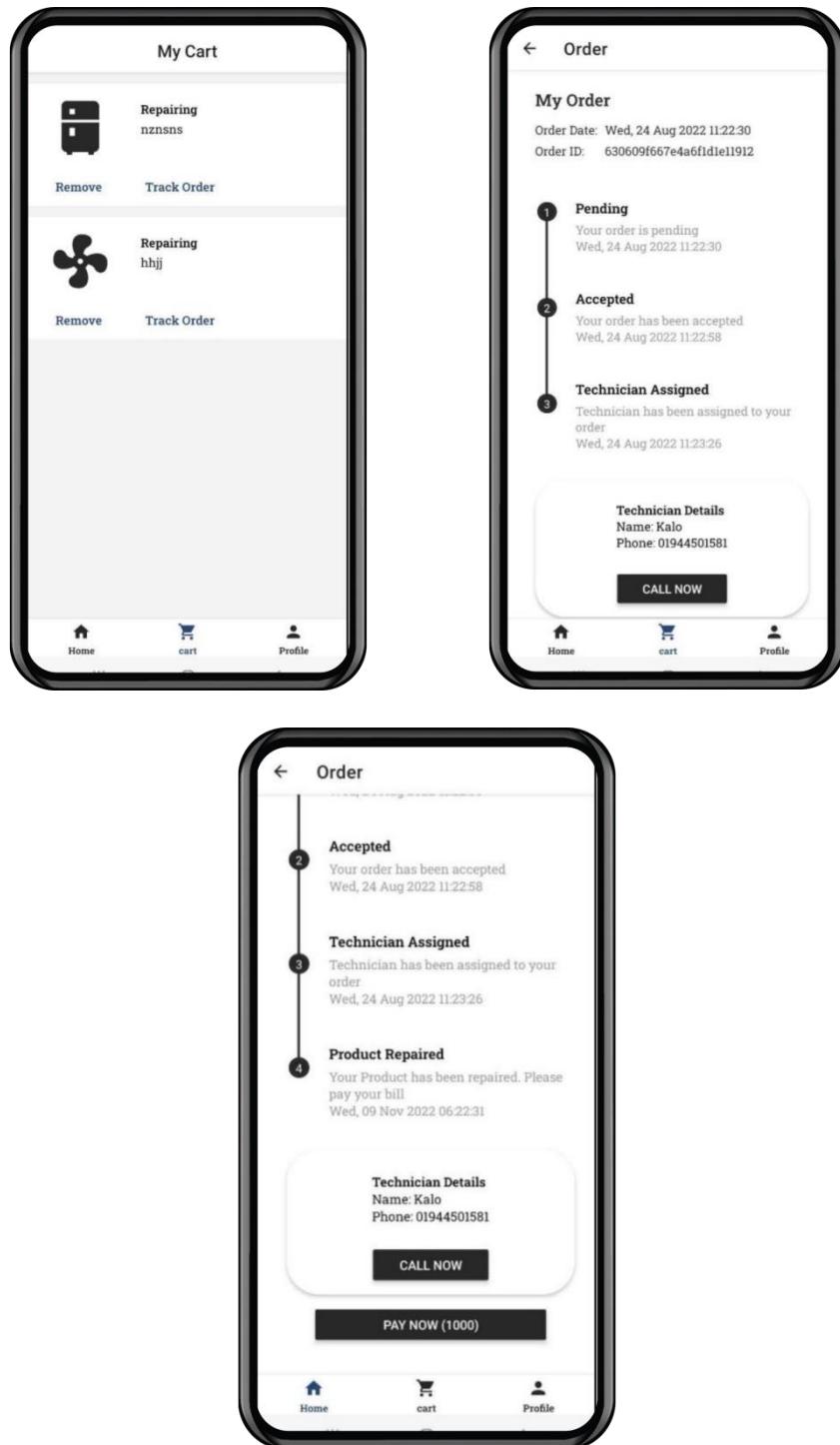


Figure 33: My Cart Page

## 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.

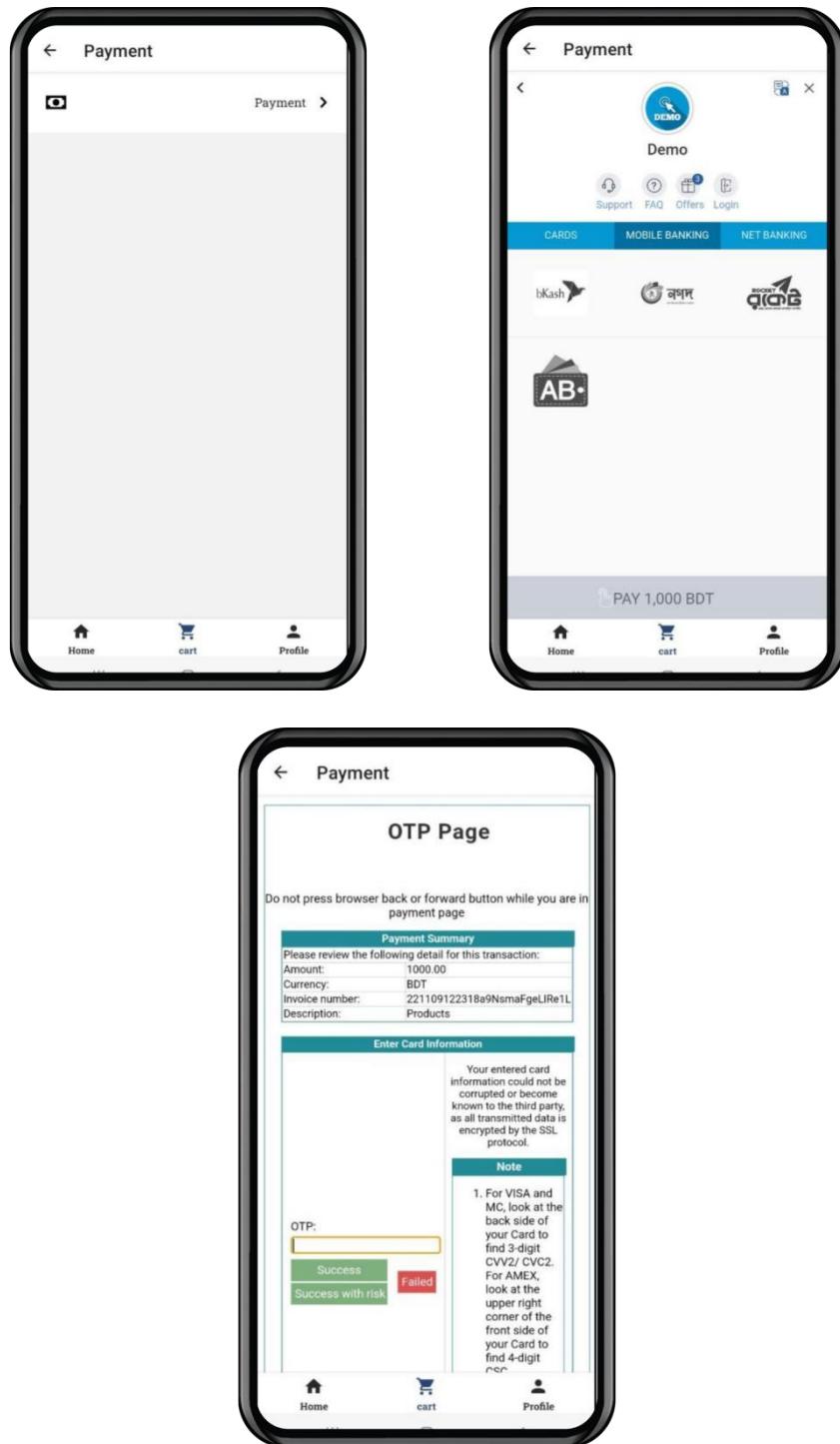


Figure 34: Payment Page

## 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.

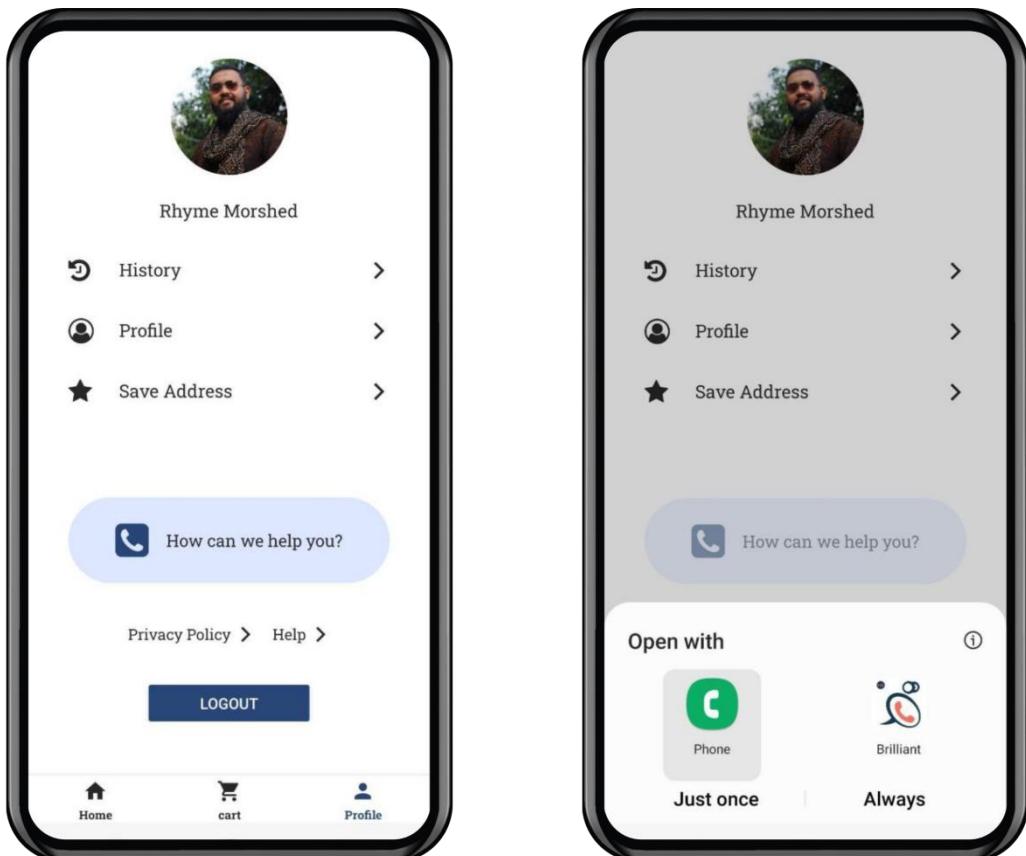


Figure 35: Profile Page

## 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.

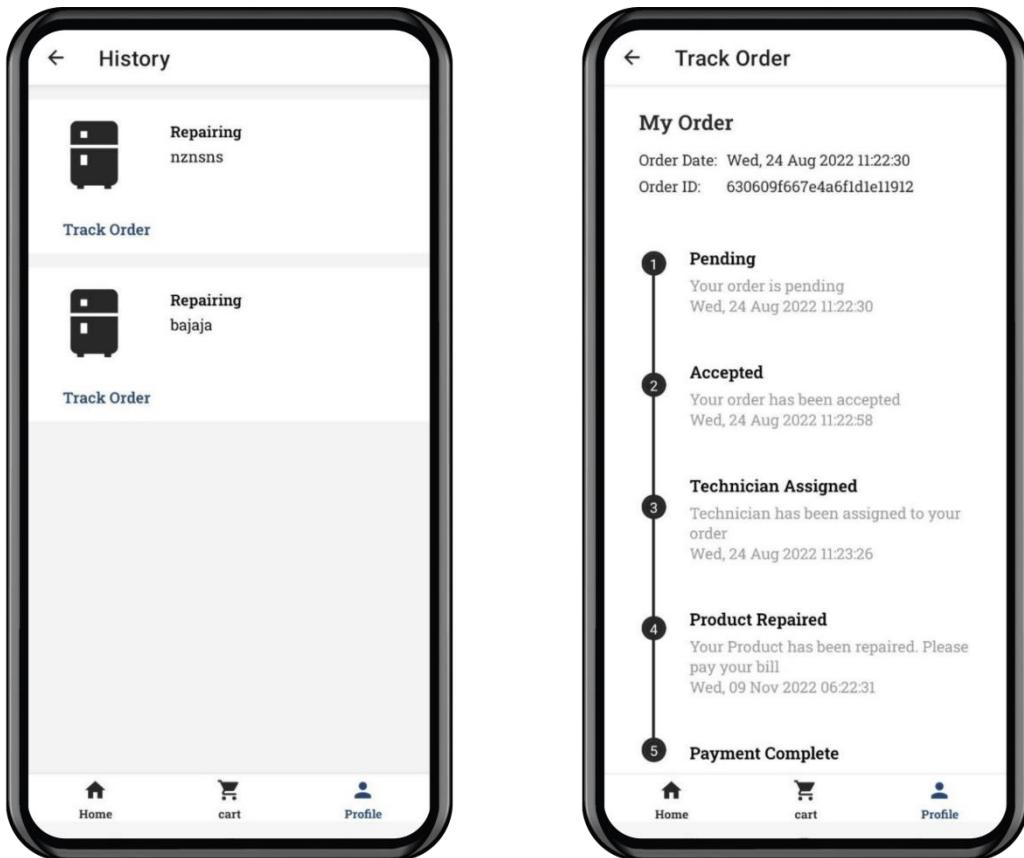


Figure 36: History Page

## Profile Edit Page

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

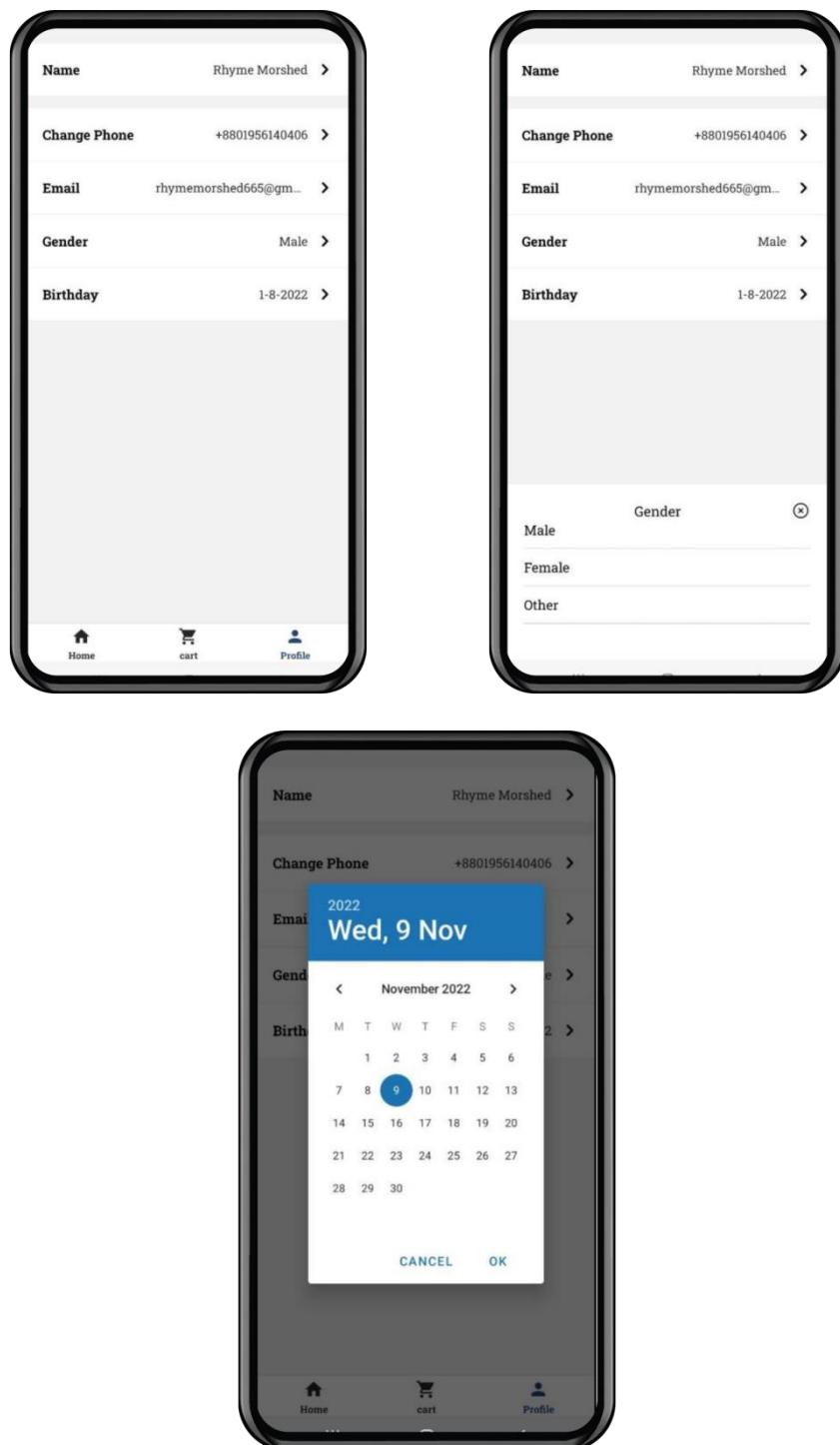


Figure 37: Profile Edit Page

## Address Page

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

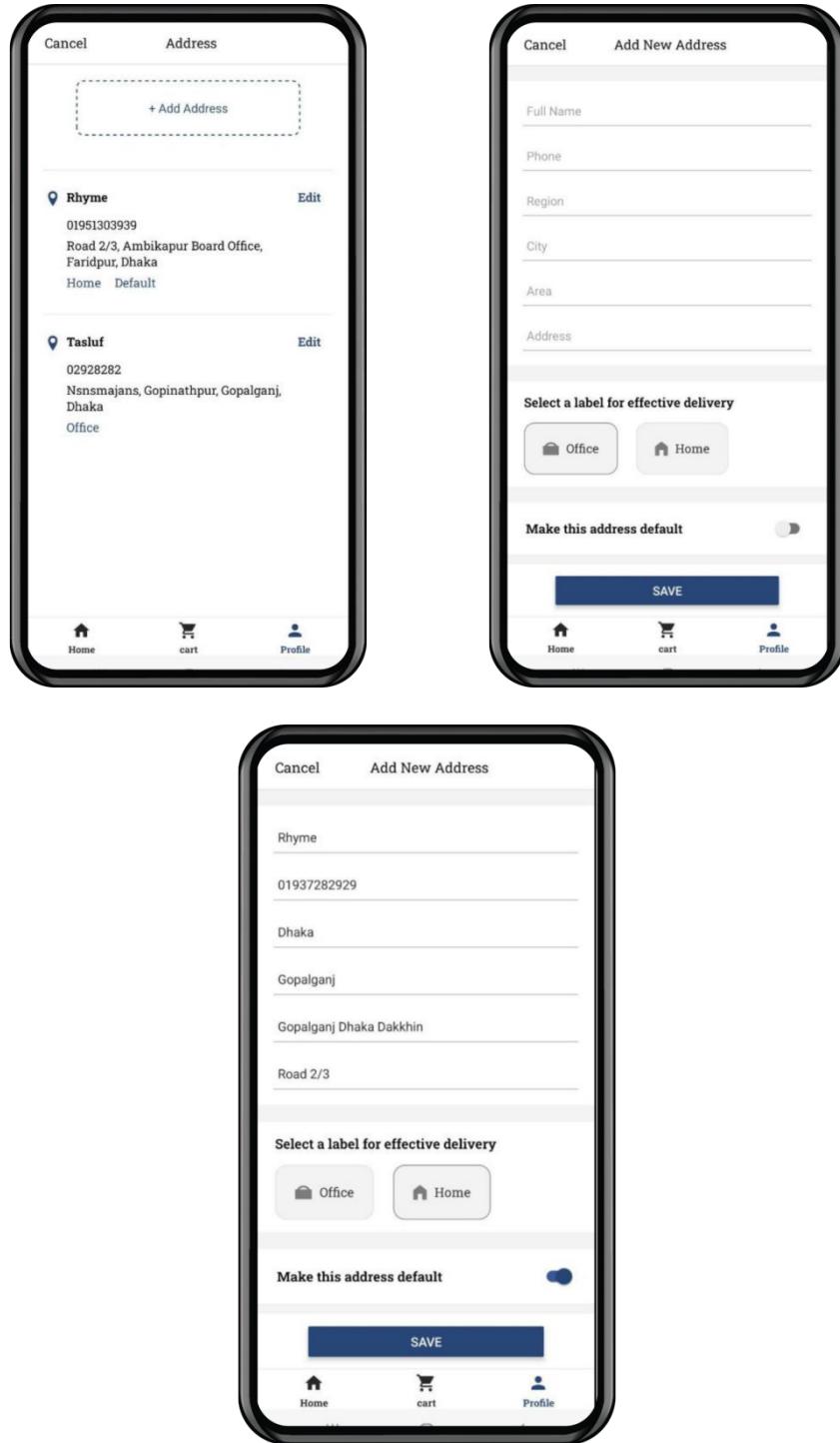


Figure 38: Address Page

## Select Address Page

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

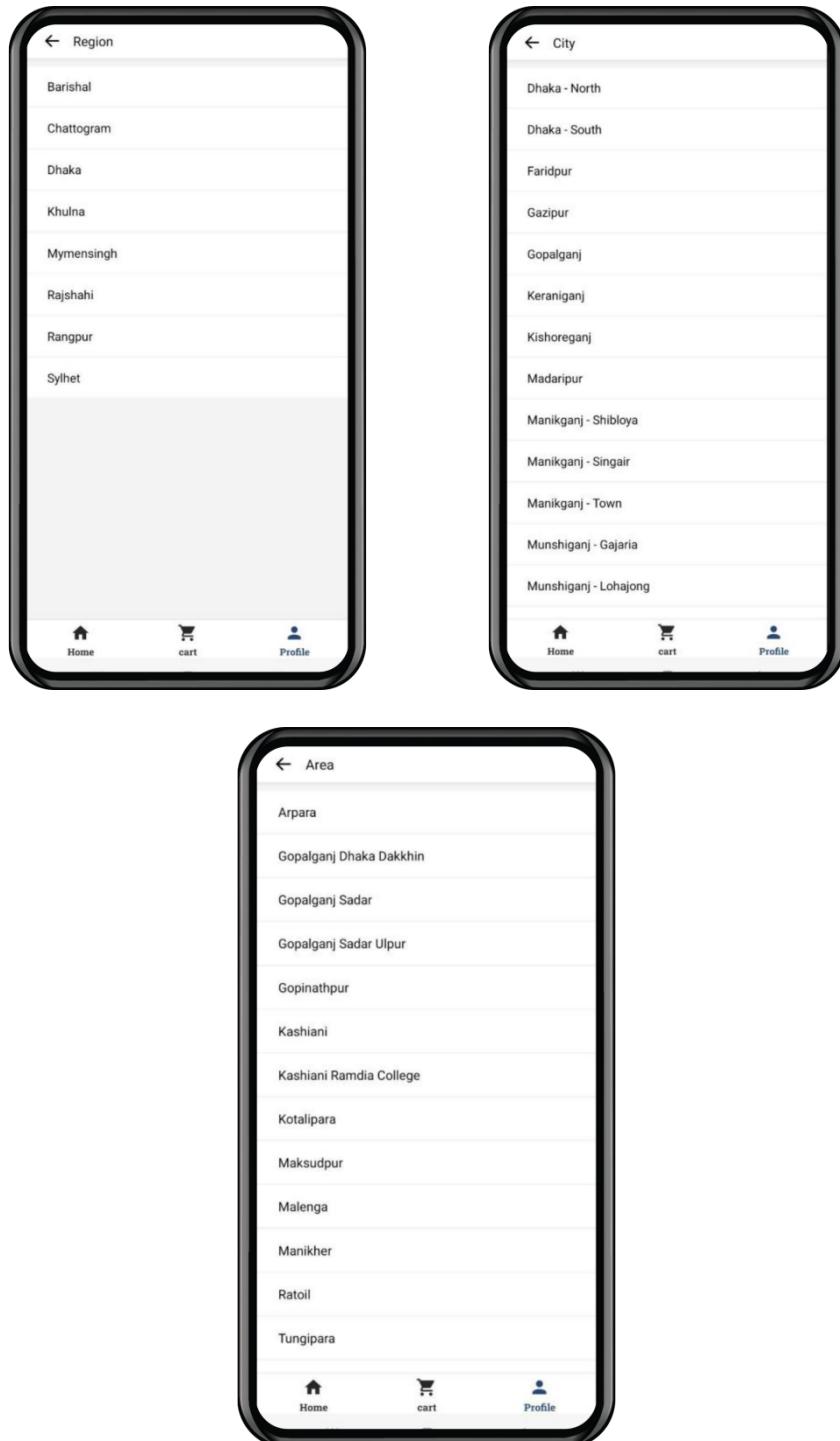


Figure 38: Select Address Page

## Notification Page

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

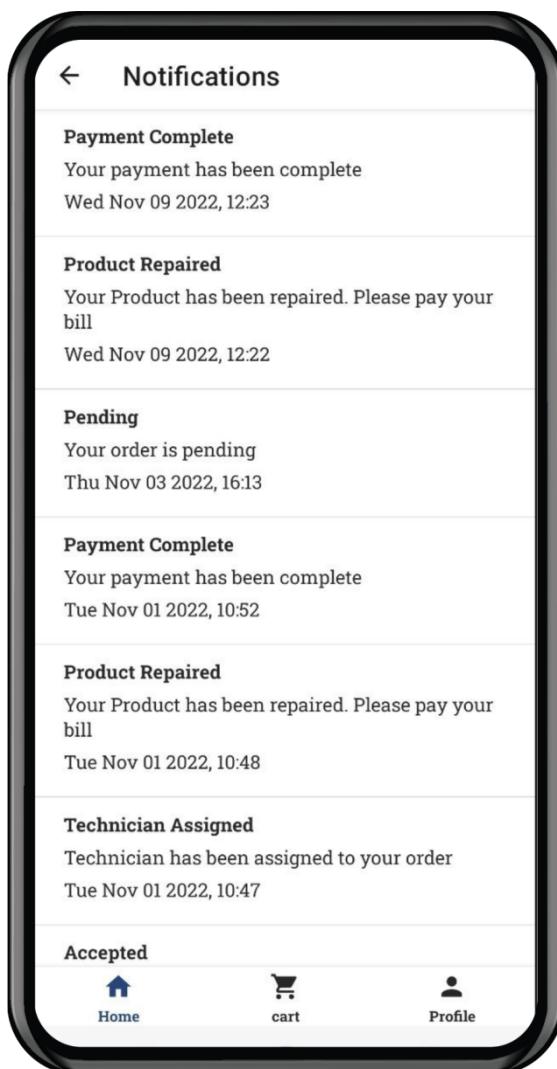
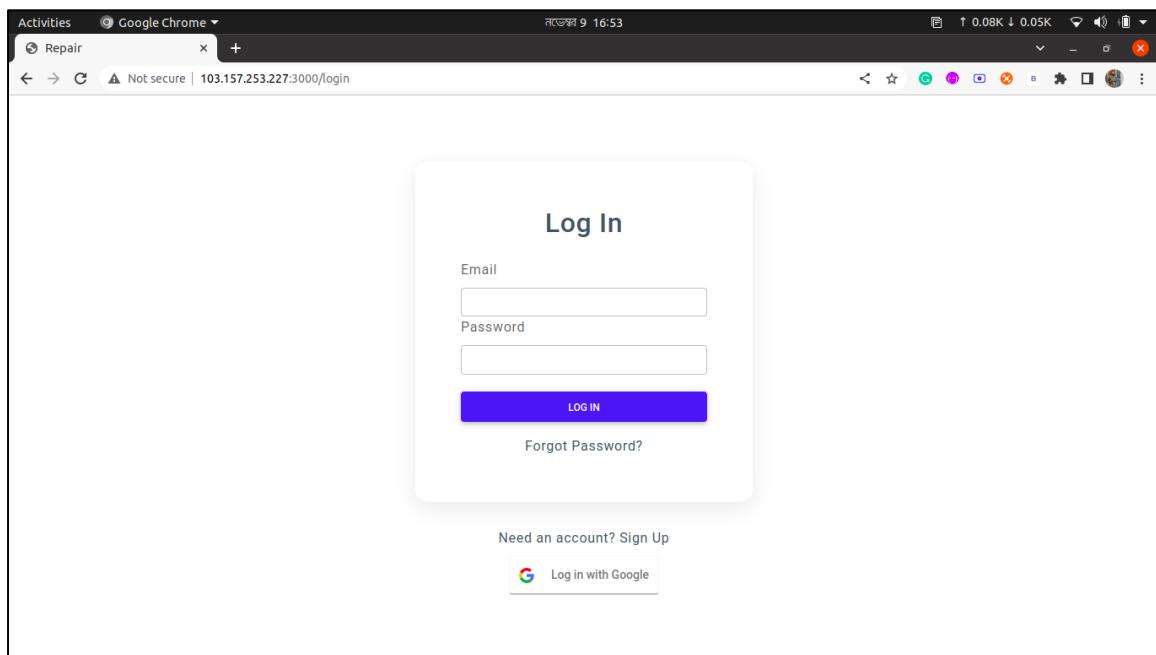


Figure 39: Notification Page

#### 4.2.2 Admin Panel

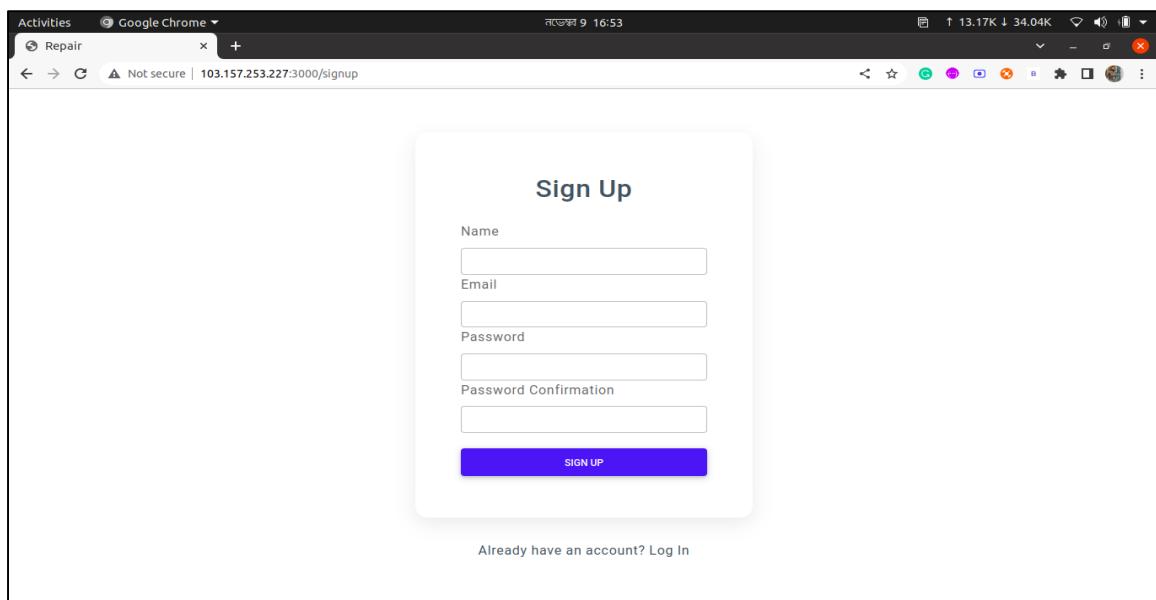
##### Login Page



A screenshot of a Google Chrome browser window showing the login page. The URL in the address bar is `103.157.253.227:3000/login`. The page features a central "Log In" form with fields for "Email" and "Password", a "LOG IN" button, and a "Forgot Password?" link. Below the form, there's a "Need an account? Sign Up" link and a "Log in with Google" button.

Figure 40: Login Page

##### Sign Up Page



A screenshot of a Google Chrome browser window showing the sign up page. The URL in the address bar is `103.157.253.227:3000/signup`. The page features a central "Sign Up" form with fields for "Name", "Email", "Password", and "Password Confirmation", a "SIGN UP" button, and a "Log In" link at the bottom.

Figure 41: Sign Up Page

## Forgot Password Page

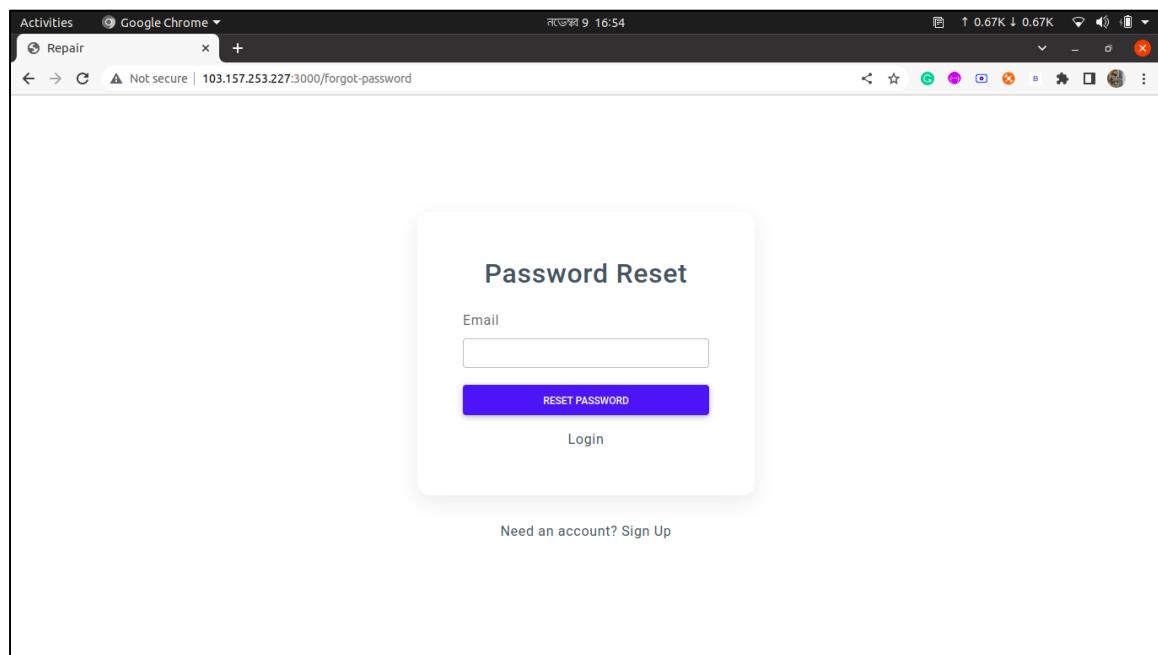


Figure 42: Forgot Password Page

## Dashboard Page

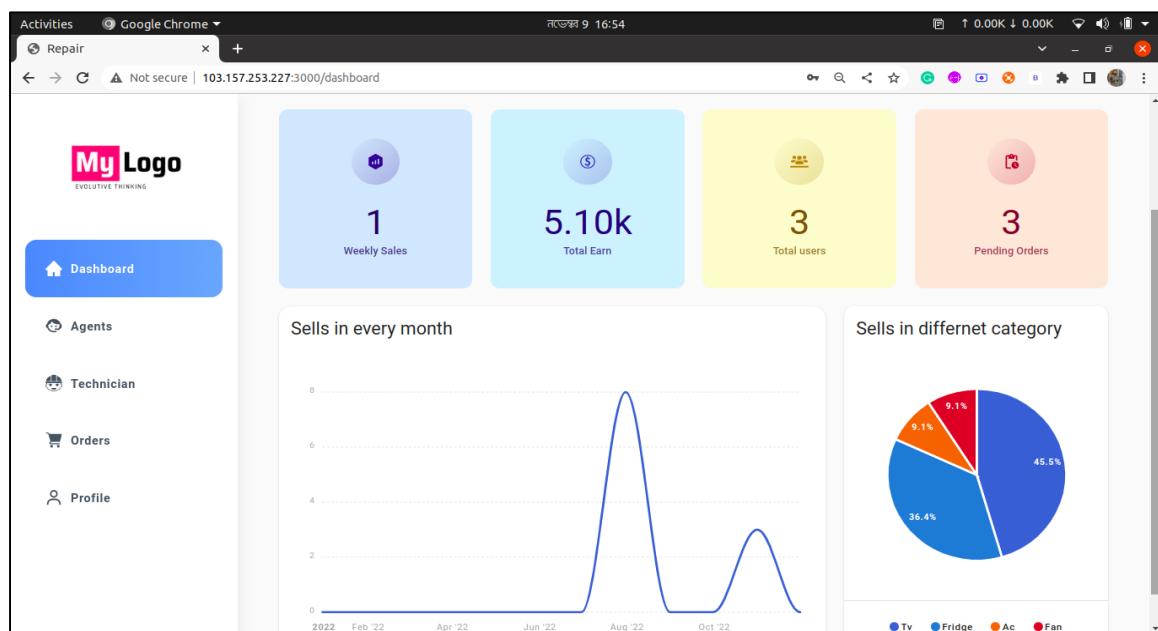


Figure 43: Dashboard Page

## Agent Page

The screenshot shows a web application interface titled "Agents". On the left, there is a sidebar with icons for Dashboard, Agents (which is selected and highlighted in blue), Technician, Orders, and Profile. The main content area has a header "Agents" and a search bar "Search with Phone Number". Below is a table listing three agents:

	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 content 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 content area has a header "Add Agent" and 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

The screenshot shows a web application interface for adding a technician. At the top, there's a navigation bar with 'Activities' and 'Repair' tabs, and a URL 'Not secure | 103.157.253.227:3000/addtechnician'. On the left, a sidebar menu includes 'Dashboard', 'Agents', 'Technician' (which is selected), 'Orders', and 'Profile'. The main content area is titled 'Add Technician' and contains the following fields:

Name	Email
Select Name	Select Email
Phone	Whatsapp Number
Select Phone	Select Whatsapp Number
Location	Region
Select Location	Barishal
City	Area
Agent	Raida, Rupganj Kanchan

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

Figure 48: Add Technician Page

## Order Page

The screenshot shows a web application interface for managing orders. At the top, there's a navigation bar with 'Activities' and 'Repair' tabs, and a URL 'Not secure | 103.157.253.227:3000/orders'. On the left, a sidebar menu includes 'Dashboard', 'Agents', 'Technician', 'Orders' (which is selected and highlighted in blue), and 'Profile'. The main content area displays a table of order details:

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

A blue '1' icon with a circle is at the bottom right of the table, indicating page 1 of 2.

Figure 49: Order Page

## Order Details Page

The screenshot shows a web browser window titled "Repair" in Google Chrome. The URL is 103.157.253.227:3000/updateorder. The page has a sidebar on the left with icons for Dashboard, Agents, Technician, Orders, and Profile. The main content area is titled "Update Order". It contains several input fields for updating an order:

Name	Phone
nsnsns	029382822
Address	jajaja
Booking Time	Arrival Time
8/11/2022	8/12/2022, 4:25:40 PM
Category	Category Type
Repairing	youtube-tv
Brand	Model
Sony Plus	SONY PLUS 24" HD LED TV
Problem	bsnsnns
Note	Status
nssnsns	Payment Complete
Agent	

A blue "EDIT" button is located in the top right corner of the form area.

Figure 50: Order Details Page

## Profile Page

The screenshot shows a web browser window titled "Repair" in Google Chrome. The URL is 103.157.253.227:3000/profile. The page has a sidebar on the left with icons for Dashboard, Agents, Technician, Orders, and Profile. The main content area shows a user profile with edit fields:

Name	Phone
Rhyme	01956140407
Gender	Birthday
Male	12/16/1999

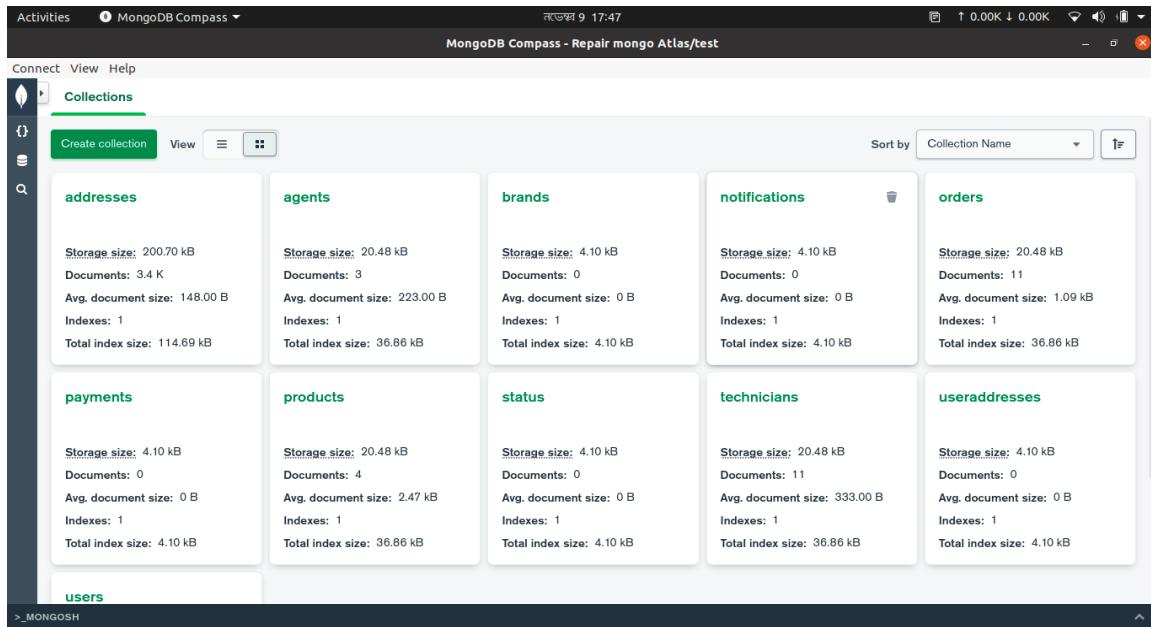
A blue "SUBMIT" button is located at the bottom right of the form area. A small user icon is in the top right corner of the main content area.

Figure 51: Profile Page

## 4.3 Back End Design

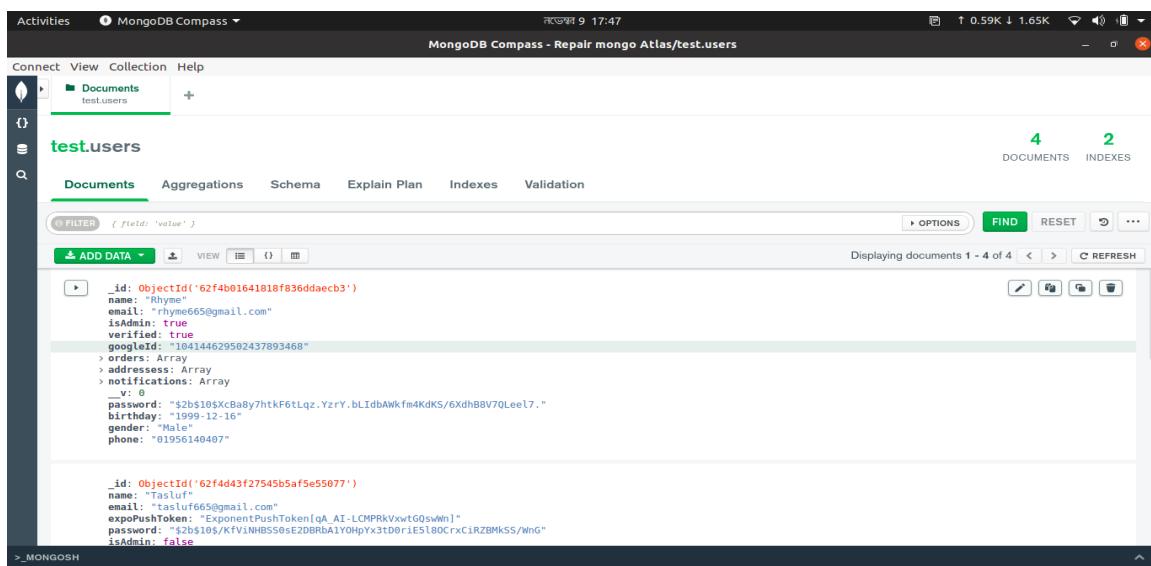
### Database

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 includes a "Create collection" button, a "View" button, and a "Sort by Collection Name" dropdown. Below the cards, there's a "users" collection card and a footer bar with "MONGOSH" and a terminal icon.

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" and "MongoDB Compass" buttons. The main area is titled "test.users" and shows a list of documents. At the top, it says "4 DOCUMENTS 2 INDEXES". Below that, there are tabs for "Documents", "Aggregations", "Schema", "Explain Plan", "Indexes", and "Validation". A search bar at the top has a filter "( field: 'value' )". Below the search bar, there are buttons for "ADD DATA", "VIEW", and "OPTIONS". The results table shows two documents highlighted in green. The first document has fields: \_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". The second document has fields: \_id: ObjectId('62f4d43f27545b5af5e55077'), name: "Taslu", email: "taslu665@gmail.com", expoPushToken: "ExponentPushToken[qA\_AI-LCMPrkVxvtG0gwWh]", password: "\$2b\$10\$/KfViNHbSS0sE2D8RA21Y0Hprx3tD0riE5l80CrxCirZBMK5S/WnG", isAdmin: false. Below the table, it says "Displaying documents 1 - 4 of 4".

Figure 53: User Table

MongoDB Compass - Repair mongo Atlas/test.agents

test.agents

3 DOCUMENTS 1 INDEXES

**Documents**

**test.agents**

**FIND** **RESET**

**ADD DATA**

**VIEW**

**OPTION**

**REFRESH**

**EDIT** **DELETE** **CREATE**

**Documents** **Aggregations** **Schema** **Explain Plan** **Indexes** **Validation**

**Filter** { field: 'value' }

**Displaying documents 1 - 3 of 3**

```

{
  "_id": ObjectId("62eb2bd4af51be3516a0bc9b"),
  "name": "Rhyme",
  "email": "rhyme665@gmail.com",
  "phone": "01956140407",
  "whatsappNumber": "01956140407",
  "region": "Dhaka",
  "city": "Munshiganj - Town",
  "area": "Munshiganj Sadar Mirkadim",
  "location": "Road 2/20",
  "__v": 0
}

{
  "_id": ObjectId("62eb2c094f51be3516a0bc5"),
  "name": "Tilok",
  "email": "tilok@gmail.com",
  "phone": "01956140407",
  "whatsappNumber": "01956140407",
  "region": "Dhaka",
  "city": "Gazipur",
  "area": "Board Bazar",
  "location": "Road 2/34",
  "__v": 0
}

```

>\_MONGOSH

Figure 54: Agent Table

MongoDB Compass - Repair mongo Atlas/test.technicians

test.technicians

11 DOCUMENTS 1 INDEXES

**Documents**

**test.technicians**

**FIND** **RESET**

**ADD DATA**

**VIEW**

**OPTION**

**REFRESH**

**EDIT** **DELETE** **CREATE**

**Documents** **Aggregations** **Schema** **Explain Plan** **Indexes** **Validation**

**Filter** { field: 'value' }

**Displaying documents 1 - 11 of 11**

```

{
  "_id": ObjectId("62eb2c284f51be3516a0bc0"),
  "name": "Tina",
  "email": "tasrif665@gmail.com",
  "phone": "+886100735",
  "whatsappNumber": "+886100735",
  "region": "Dhaka",
  "city": "Munshiganj - Town",
  "area": "Dighirpar",
  "location": "Road 2/4",
  "agent": Object,
  "__v": 0
}

{
  "_id": ObjectId("62eb2c414f51be3516a0bcbe"),
  "name": "Kalo",
  "email": "Kalog@gmail.com",
  "phone": "01944501581",
  "whatsappNumber": "01944501581",
  "region": "Mymensingh",
  "city": "Mymensingh - Muktagacha",
  "area": "Muktagacha Sadar",
  "location": "Road 2/34",
  "__v": 0
}

```

>\_MONGOSH

Figure 55: Technician Table

The screenshot shows the MongoDB Compass interface for the 'test.addresses' collection. At the top, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The 'Documents' tab is selected. Below the tabs, there is a search bar with a 'FILTER' button containing the query '{ field: 'value' }'. To the right of the search bar are buttons for 'OPTIONS', 'FIND', 'RESET', and three dots. The main area displays 3357 documents, with 20 shown per page. The first few documents listed are:

```

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

{
  "_id": ObjectId("62eb2bbb2d2c3918b03d598e"),
  "id": "R88308476",
  "name": "Chattogram",
  "nameLocal": "Chattogram",
  "parentId": "R184649",
  "displayName": "Chattogram"
}

{
  "_id": ObjectId("62eb2bbb2d2c3918b03d598f"),
  "id": "R88308476",
  "name": "Barishal - Sahebganj",
  "nameLocal": "Barishal - Sahebganj",
  "parentId": "R3921298"
}

```

At the bottom left, there is a 'MONGOSH' prompt.

Figure 56: Address Table

The screenshot shows the MongoDB Compass interface for the 'test.orders' collection. At the top, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The 'Documents' tab is selected. Below the tabs, there is a search bar with a 'FILTER' button containing the query '{ field: 'value' }'. To the right of the search bar are buttons for 'OPTIONS', 'FIND', 'RESET', and three dots. The main area displays 11 documents, with 11 shown per page. The first document listed is:

```

{
  "_id": ObjectId("62f4e7476504589c9758f578"),
  "name": "nssnsns",
  "phone": "029382822",
  "address": "jajaja",
  "arrivalDate": "2022-08-12T11:25:38.713+00:00",
  "arrivalTime": "2022-08-11T10:25:40.602+00:00",
  "category": "Repairing",
  "categoryType": "youtube-tv",
  "brand": "62f32a30657316510269a963",
  "model": "62f3483bbf20bcfc7faef977",
  "problem": "bsnsnsn",
  "note": "nssnsns",
  "bookingTime": "2022-08-11T11:25:59.386+00:00",
  "status": Array,
  "payment": ObjectID("62f4d43f27545b5af5e55077"),
  "userId": ObjectID("62f49af56271c8181354a869"),
  "v": 4,
  "technicianId": ObjectID("62f5062c25ecadd351c1eac4"),
  "amount": 500
}

```

At the bottom left, there is a 'MONGOSH' prompt.

Figure 57: Orders Table

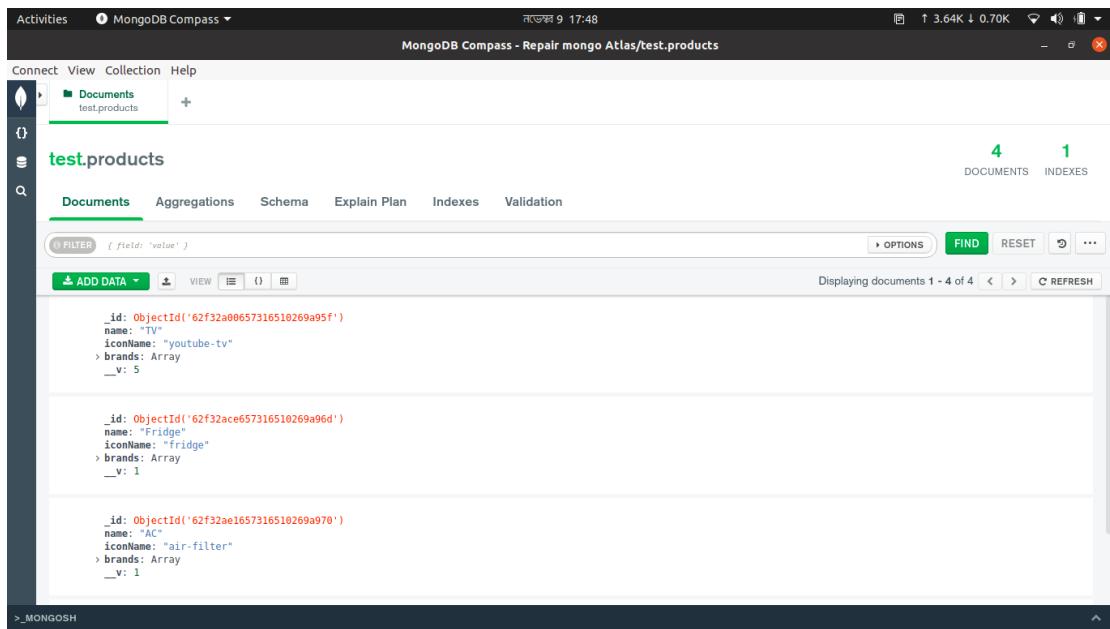


Figure 58: Products Table

#### 4.4 Interaction Design and User Experience (UX)

We 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 and 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.

#### 4.5 Implementation Requirement

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

#### 4.6 Conclusion

In our project, all this design specifications gave a clear understanding of what the product about, how it should look like, how users should interact with it.

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### **5.2 Introduction**

Once the software implementation was done, the next main activity was to evaluate the implemented software. By using software testing method we will examination our code as well as execute that code in various environments and conditions.

#### **5.2 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.

#### **5.3 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.

#### **5.4 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.

## 5.5 Test Results and Reports

### 5.5.1 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
      ✓ Get all agents with admin token (89 ms)
      ✓ Try to get all agents with user token (12 ms)
```

The bottom status bar shows "Ln 326, Col 8 Spaces:2 UTF-8 LF () Babel JavaScript ⚡ Go Live ✅ Prettier ⌂".

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:

```
✓ 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)
    ✓ Get 401 error without authentication token (11 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?
```

The bottom status bar shows "Ln 326, Col 8 Spaces:2 UTF-8 LF () Babel JavaScript ⚡ Go Live ✅ Prettier ⌂".

Figure 60: Automated Test figure 2

## 5.6 Conclusion

Once the software implementation was done, the next main activity was to evaluate the implemented software. In other words it should be evaluated whether the software meets all the requirements identified at the initial stage of finalizing SRS and the provided functionality works properly.

# **CHAPTER 6**

## **IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY**

### **6.1 Introduction**

The impact of modern technology on the environment is no longer negligible. In the modern time people become more dependent on smartphone or other secular devices as a result an unstable system growing. This chapter will look after the social, environmental and sustainability issues of our applications. By performing a SWAT analysis of our app we can reduce the issues of our app.

### **6.2 SWOT Analysis**

SWOT analysis is a strategic planning technique that use to evaluate a company's strength, weakness, opportunity and threats in order to determine current and future potential. It is necessary for an organization to learn about internal and external factor that can impacts on decision making or other entity. A SWAT analysis can guide businesses towards competitive strategies that can solve complex problem in a manageable way. In our application, after SWOT analysis we have find some clear picture of potential questions. Which are:

#### **Strengths**

Strengths determines the unique strategies of a company in order to gain competitive advancement from its competitor. In our apps, we can consider the aggregation business model as our main strength where we connect vendor and customer together in order to create a healthy business environment.

#### **Weaknesses**

Weaknesses stops an organization from performing potential task, reached towards its goal. In our apps, there are areas where the business needs to improve. Such as Updated UI, error handling, incident handling, pay load, internet connection and other issues that can cause customer satisfaction lead to potential business loss.

#### **Opportunities**

Opportunities refer to favorable factors that could give competitive advantage to an org. Service industry is one of the fastest growing industry with 11% growth rate.

Whereas customer are not satisfied yet from this industry. If we gain this customer trust we can go farther with other business using the same brand value.

### **Threats**

Threats are external factors that can potentially harm an organization. In business there already many companies or large organization that already providing the same service as we offers. On top of them, Current E commerce trend of our country is not on our favor. That can be a massive threat.

### **6.3 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, therefore 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. Therefore 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.

### **6.4 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. This chemical 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.

## **6.5 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 install an apps or visit 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 analyze 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 informed 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 professional 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.

## **6.6 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.

## **6.7 Conclusion**

The functionality of this application has been designed based on above features so that the apps can impose a positive impact on society and environment and sustainable for next 10 years.

## CHAPTER 7

# CONCLUSION

### 7.1 Conclusion

In the 21th century while worlds is finding modern solution for their problems and the E commerce business is growing faster to allay with this, our country also flowing with the trend. Current E Commerce market of our country is projected to reach \$9.603 Billion dollar in 2023 with an annual growth rate of 16.26%. Therefor we have seen a lots of opportunities here. With the intention of solving some real time problem we develop this project. To recapitulate, we can define this as:

This Project “**Android based electronic product servicing system**” is an E commerce application that run on aggregation model. This site provides an easier servicing solution for users as well create a unique business model for its vendor.

This smart system aggregate customer who are in need of product servicing and service provider who can gave service to spend their business. By this apps Customer can book a service request based on their suitable time. They simply go to the site and with a valid profile gave a minimum details of his product and the admin will assign a technician for the request.

This apps 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 their requirement also save time as well. It will develop the business supply chain and economy.

## **7.2 Further Suggested Work**

We plan to make something in future with this application.

- Add new features and business
- Car, Bike and Electric vehicle servicing.
- Second hand product buying and selling
- House, furniture renovation.
- Expend service Centre
- Expend in different Cities
- Establish company own service Centre.
- Improve and update quality
- Improve database and payment gateway.
- Update previous bug and incident handling method
- Update UI design based on user requirement

## REFERANCE

- [1] "Electronic Goods Repairing & Servicing in Bangladesh," [www.bdtradeinfo.com](http://www.bdtradeinfo.com).  
<https://www.bdtradeinfo.com/yp-data/electronic-goods-repairing-servicing>.
- [2] T. H. Pham, "B2B Brand Management," [www.academia.edu](http://www.academia.edu), [Online]. Available:  
[https://www.academia.edu/14558996/B2B\\_Brand\\_Management](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](http://www.youtube.com).  
<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.
- [13] W. Kenton, "SWOT Analysis: How to with Table and Example," *Investopedia*, 2022.  
<https://www.investopedia.com/terms/s/swot.asp>

# **APPENDIX**

## **Appendix A**

### **Report Layout**

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

**Chapter 2: Background** we 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 we 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 we 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 we will discuss about our hardware device and its design implementation. All the required device module will be discussed.

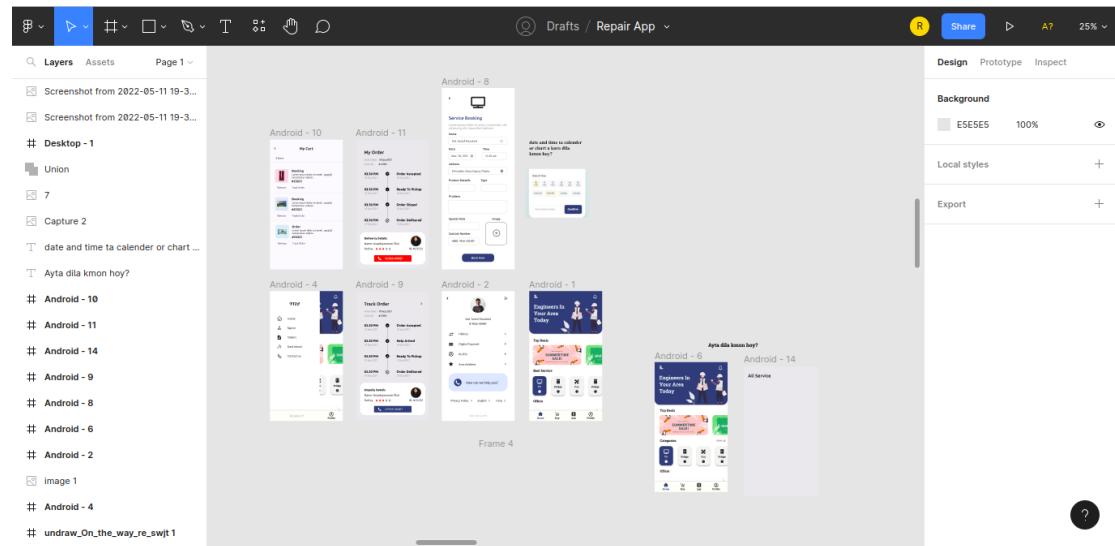
**Chapter 6: Impact on Society, Environment and Sustainability** In this section we will discuss about Impact on Society, Environment and Sustainability

**Chapter 7: Conclusion and further suggested work** in this section. In conclusion we discuss based on three aspect what, why & how.

## Appendix B

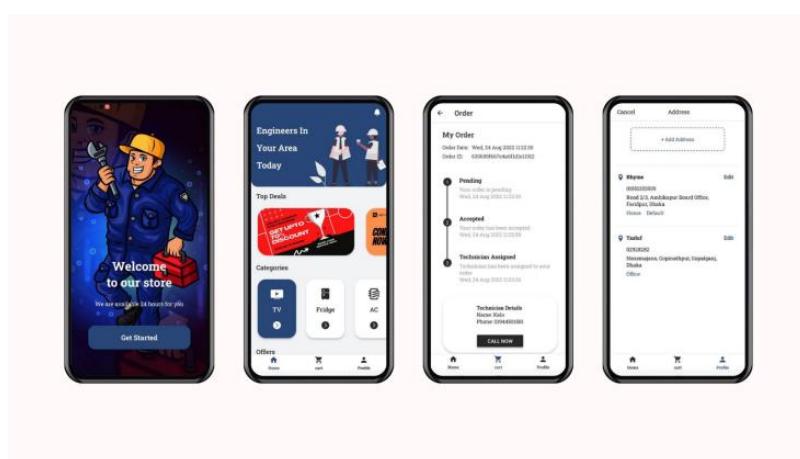
Design Resource: We design our app in Figma.

<https://www.figma.com/>



For logo design and flayer design we use Canva.

<https://www.canva.com/>



For App Promo Video, we use After Effect video editor.

<https://www.youtube.com/watch?v=wWMqArSEQqg>

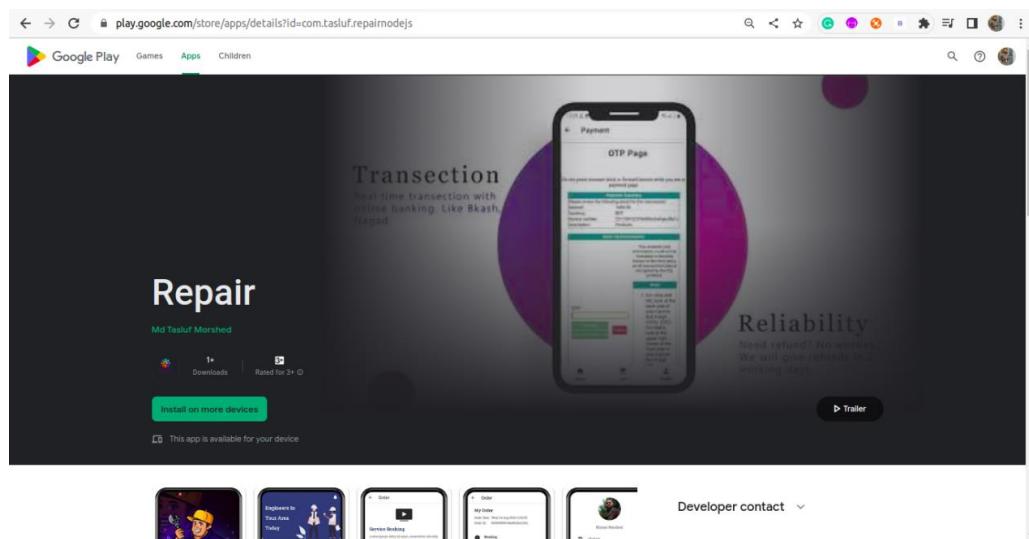
## Appendix C

Main App Resource:

First we deploy our app in expo server.

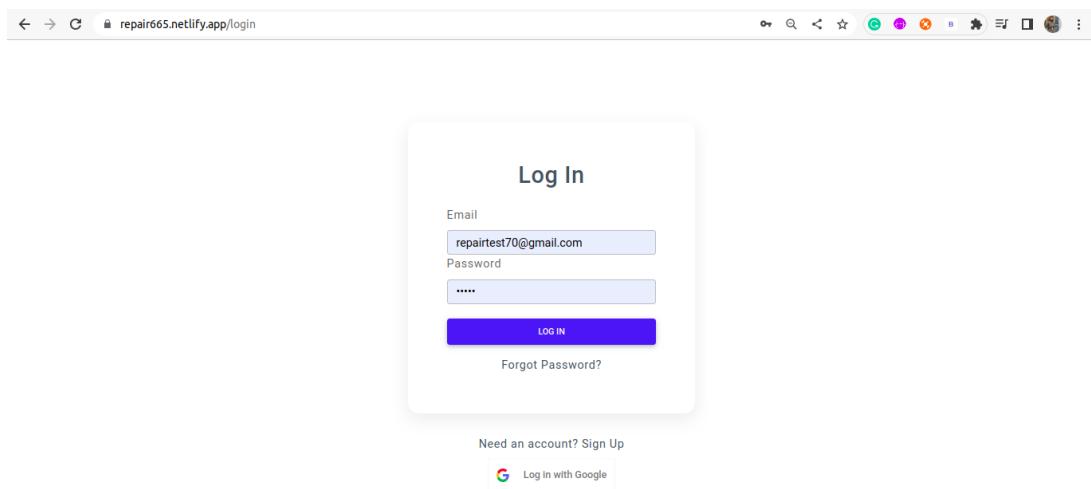
<https://expo.dev/@tasluf/repair-nodejs?serviceType=classic&distribution=expo-go>

Play store link: <https://play.google.com/store/apps/details?id=com.tasluf.repairnodejs>



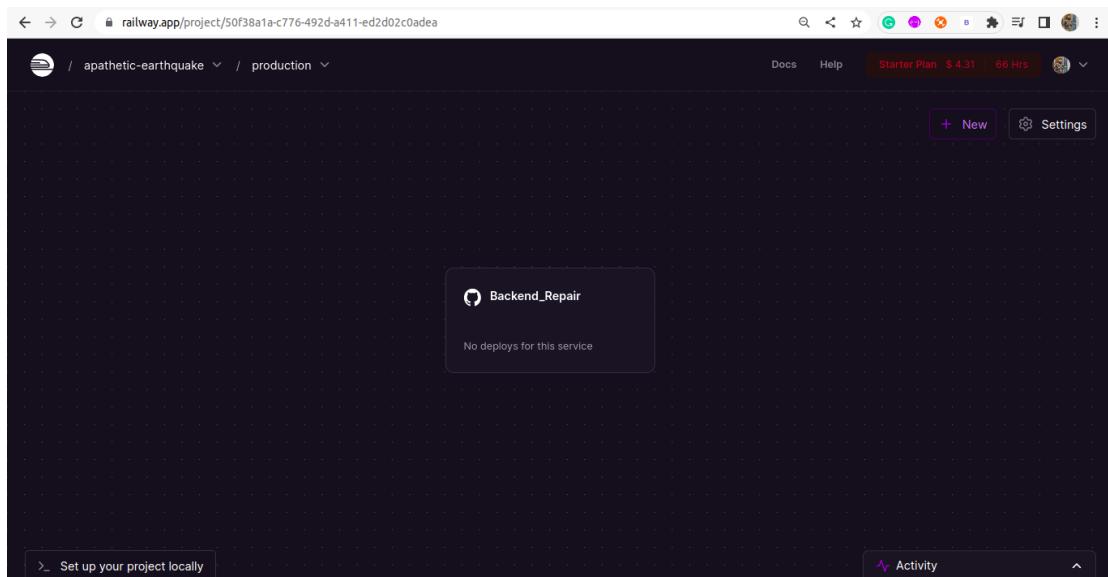
We deploy our Admin panel in Netlify.

Admin panel Link: <https://repair665.netlify.app/login>



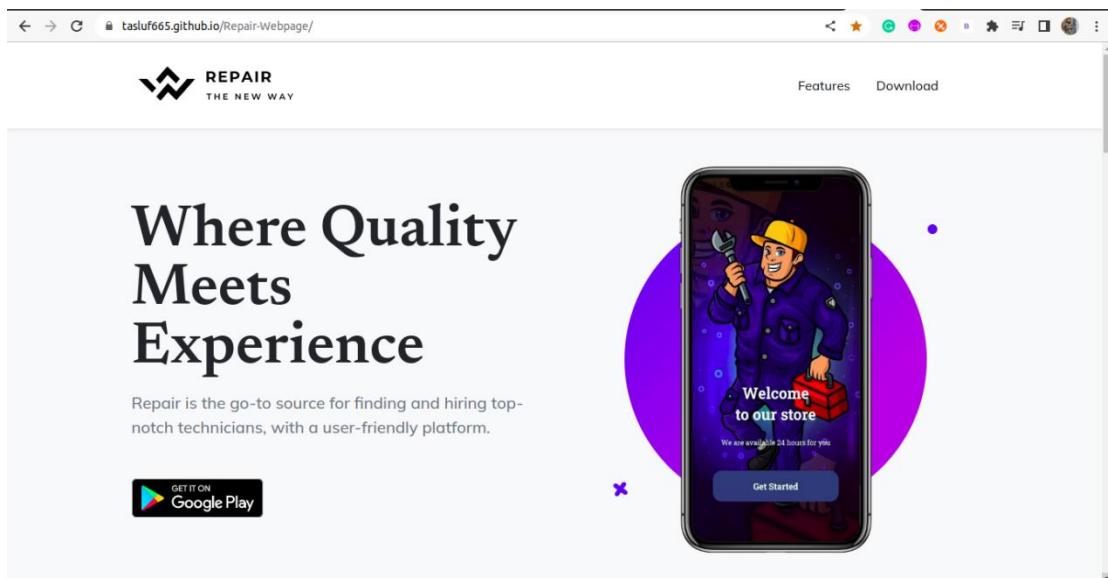
We deploy our Backend part (Nodejs) in railway.

Backend REST API link: <https://backendrepair-production.up.railway.app/>



App Official Website Link:

<https://tasluf665.github.io/Repair-Webpage/>



Full App overview video

<https://www.youtube.com/watch?v=lok9iQl6jb4>

## **Appendix D**

Project github Links:

1. Android App: [https://github.com/Tasluf665/Repair\\_Android\\_With\\_Nodejs](https://github.com/Tasluf665/Repair_Android_With_Nodejs)

2. Admin Panel: [https://github.com/Tasluf665/Repaire\\_Dashboard\\_Nodejs](https://github.com/Tasluf665/Repaire_Dashboard_Nodejs)

3. Backend (NodeJS):

[https://github.com/Tasluf665/Backend\\_Repair](https://github.com/Tasluf665/Backend_Repair)

4. App Official website:

<https://github.com/Tasluf665/Repair-Webpage>

## Appendix E

## Backend Part REST API

## API Routes:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "BACKEND\_REPAIR". The "auth" folder is selected, containing "auth.js" and other files like "routes", "address.js", "agents.js", etc.
- Code Editor:** The "auth.js" file is open, displaying code for password reset logic using Express.js and a utility function for sending emails.
- Status Bar:** Shows "Ln 16, Col 22", "Spaces: 2", "UTF-8", "LF", "Babel JavaScript", "Go Live", and "Prettier" status indicators.

```
routes: 12 sendResetPasswordEmail,
13 } = require("../utils/SendEmail");
14
15 router.post(
16   "/forgot-password",
17   asyncMiddleware(async (req, res) => {
18     const { error } = validateForgotPassword(req.body);
19     if (error) return res.status(400).send({ error: error.details[0].message });
20
21     let user = await User.findOne({ email: req.body.email });
22     if (!user)
23       return res
24         .status(400)
25         .send({ error: "User not found with this given email" });
26
27     sendResetPasswordEmail(user.email, user._id);
28     res.send({ success: "Check your email to reset your password" });
29   })
30 );
31
32 router.get(
33   "/reset-password/:token",
34   asyncMiddleware(async (req, res) => {
35     const { token } = req.params;
36
37     if (!token)
38       return res
39         .status(401)
40         .send({ error: "Access denied. No token provided" });
41     try {
42       jwt.verify(token, config.get("jwtPrivateKey"));
43       res.render("ResetPasswordForm", { token: token });
44     } catch (err) {
45       res.status(400).send({ error: err.message });
46     }
47   })
48 );
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** orders.js - Backend\_Repair - Visual Studio Code
- Explorer:** Shows the project structure under 'BACKEND\_REPAIR'. The 'orders.js' file is selected in the 'routes' folder.
- Editor:** Displays the code for 'orders.js'. The code handles sending notifications for accepted orders via an API endpoint.
- Bottom Status Bar:** Lines 313, Col 1, Spaces: 2, UTF-8, LF, Babel JavaScript, Go Live, Prettier.

```
const notification = new Notification({
  statusDetails: req.body.statusDetails,
  statusState: req.body.statusState,
  orderId: order._id,
});
user.notifications.push(notification);

order.problem = req.body.problem ? req.body.problem : order.problem;
order.note = req.body.note ? req.body.note : order.note;

await order.save();
await user.save();

await fetch("https://exp.host/--/api/v2/push/send", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    to: user.expoPushToken,
    title: status.statusState,
    body: status.statusDetails,
  }),
});

res.send({
  success: "Order is Accepted",
  data: order,
});
```

address.js - Backend\_Repair - Visual Studio Code

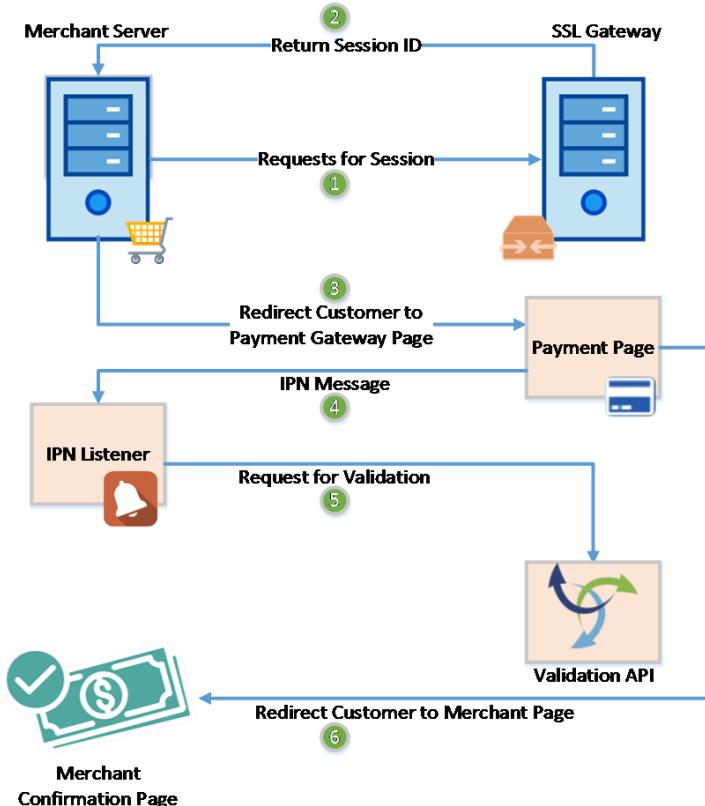
```
File Edit Selection View Go Run Terminal Help address.js
```

EXPLORER routes > address.js > address.js

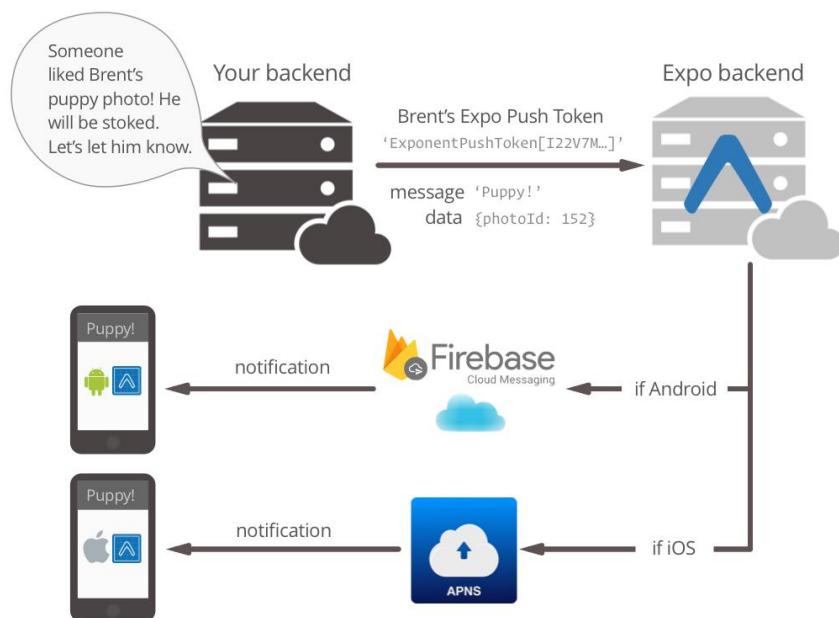
```
1 const express = require("express");
2 const router = express.Router();
3 const { Address, validateAddress } = require("../models/address");
4 const asyncMiddleware = require("../middleware/async");
5 const auth = require("../middleware/auth");
6 const admin = require("../middleware/admin");
7
8 router.get(
9   "/",
10  [auth],
11  asyncMiddleware(async (req, res) => {
12    if (!req.query.id) req.query.id = "R184640";
13    const address = await Address.find({ parentId: req.query.id })
14      .sort("displayName")
15      .select("-_v");
16    if (!address)
17      return res
18        .status(404)
19        .send({ error: "The address with the given ID was not found" });
20
21    res.send({
22      success: "Address is fetched successfully",
23      data: address,
24    });
25  });
26
27 router.post(
28   "/",
29   [auth, admin],
30   asyncMiddleware(async (req, res) => {
31     const { error } = validateAddress(req.body);
32
33     if (error) return res.status(400).send({ error: error.details[0].message });
34
35     const agent = new Agent({
36       name: req.body.name,
37       email: req.body.email,
38       phone: req.body.phone,
39       whatsappNumber: req.body.whatsappNumber,
40       region: req.body.region,
41       city: req.body.city,
42       area: req.body.area,
43       location: req.body.location,
44     });
45
46     await agent.save();
47     res.send({
48       success: "Agent is added successfully",
49       data: agent,
50     });
51   });
52
53 });
54
55 );
56
57 );
58
59 );
60
61 );
62
63 );
64
65 router.post(
66   "/",
67   [auth, admin],
68   asyncMiddleware(async (req, res) => {
69     const { error } = validateAddress(req.body);
70
71     if (error) return res.status(400).send({ error: error.details[0].message });
72
73     const agent = new Agent({
74       name: req.body.name,
75       email: req.body.email,
76       phone: req.body.phone,
77       whatsappNumber: req.body.whatsappNumber,
78       region: req.body.region,
79       city: req.body.city,
80       area: req.body.area,
81       location: req.body.location,
82     });
83
84     await agent.save();
85     res.send({
86       success: "Agent is added successfully",
87       data: agent,
88     });
89   });
90
91 );
92
93 );
94
95 );
96
97 );
98
99 );
100
101 );
102
103 );
104
105 );
106
107 );
108
109 );
110
111 );
112
113 );
114
115 );
116
117 );
118
119 );
120
121 );
122
123 );
124
125 );
126
127 );
128
129 );
130
131 );
132
133 );
134
135 );
136
137 );
138
139 );
140
141 );
142
143 );
144
145 );
146
147 );
148
149 );
150
151 );
152
153 );
154
155 );
156
157 );
158
159 );
160
161 );
162
163 );
164
165 );
166
167 );
168
169 );
170
171 );
172
173 );
174
175 );
176
177 );
178
179 );
180
181 );
182
183 );
184
185 );
186
187 );
188
189 );
190
191 );
192
193 );
194
195 );
196
197 );
198
199 );
199
200 );
200
201 );
201
202 );
202
203 );
203
204 );
204
205 );
205
206 );
206
207 );
207
208 );
208
209 );
209
210 );
210
211 );
211
212 );
212
213 );
213
214 );
214
215 );
215
216 );
216
217 );
217
218 );
218
219 );
219
220 );
220
221 );
221
222 );
222
223 );
223
224 );
224
225 );
225
226 );
226
227 );
227
228 );
228
229 );
229
230 );
230
231 );
231
232 );
232
233 );
233
234 );
234
235 );
235
236 );
236
237 );
237
238 );
238
239 );
239
240 );
240
241 );
241
242 );
242
243 );
243
244 );
244
245 );
245
246 );
246
247 );
247
248 );
248
249 );
249
250 );
250
251 );
251
252 );
252
253 );
253
254 );
254
255 );
255
256 );
256
257 );
257
258 );
258
259 );
259
260 );
260
261 );
261
262 );
262
263 );
263
264 );
264
265 );
265
266 );
266
267 );
267
268 );
268
269 );
269
270 );
270
271 );
271
272 );
272
273 );
273
274 );
274
275 );
275
276 );
276
277 );
277
278 );
278
279 );
279
280 );
280
281 );
281
282 );
282
283 );
283
284 );
284
285 );
285
286 );
286
287 );
287
288 );
288
289 );
289
290 );
290
291 );
291
292 );
292
293 );
293
294 );
294
295 );
295
296 );
296
297 );
297
298 );
298
299 );
299
300 );
300
301 );
301
302 );
302
303 );
303
304 );
304
305 );
305
306 );
306
307 );
307
308 );
308
309 );
309
310 );
310
311 );
311
312 );
312
313 );
313
314 );
314
315 );
315
316 );
316
317 );
317
318 );
318
319 );
319
320 );
320
321 );
321
322 );
322
323 );
323
324 );
324
325 );
325
326 );
326
327 );
327
328 );
328
329 );
329
330 );
330
331 );
331
332 );
332
333 );
333
334 );
334
335 );
335
336 );
336
337 );
337
338 );
338
339 );
339
340 );
340
341 );
341
342 );
342
343 );
343
344 );
344
345 );
345
346 );
346
347 );
347
348 );
348
349 );
349
350 );
350
351 );
351
352 );
352
353 );
353
354 );
354
355 );
355
356 );
356
357 );
357
358 );
358
359 );
359
360 );
360
361 );
361
362 );
362
363 );
363
364 );
364
365 );
365
366 );
366
367 );
367
368 );
368
369 );
369
370 );
370
371 );
371
372 );
372
373 );
373
374 );
374
375 );
375
376 );
376
377 );
377
378 );
378
379 );
379
380 );
380
381 );
381
382 );
382
383 );
383
384 );
384
385 );
385
386 );
386
387 );
387
388 );
388
389 );
389
390 );
390
391 );
391
392 );
392
393 );
393
394 );
394
395 );
395
396 );
396
397 );
397
398 );
398
399 );
399
400 );
400
401 );
401
402 );
402
403 );
403
404 );
404
405 );
405
406 );
406
407 );
407
408 );
408
409 );
409
410 );
410
411 );
411
412 );
412
413 );
413
414 );
414
415 );
415
416 );
416
417 );
417
418 );
418
419 );
419
420 );
420
421 );
421
422 );
422
423 );
423
424 );
424
425 );
425
426 );
426
427 );
427
428 );
428
429 );
429
430 );
430
431 );
431
432 );
432
433 );
433
434 );
434
435 );
435
436 );
436
437 );
437
438 );
438
439 );
439
440 );
440
441 );
441
442 );
442
443 );
443
444 );
444
445 );
445
446 );
446
447 );
447
448 );
448
449 );
449
450 );
450
451 );
451
452 );
452
453 );
453
454 );
454
455 );
455
456 );
456
457 );
457
458 );
458
459 );
459
460 );
460
461 );
461
462 );
462
463 );
463
464 );
464
465 );
465
466 );
466
467 );
467
468 );
468
469 );
469
470 );
470
471 );
471
472 );
472
473 );
473
474 );
474
475 );
475
476 );
476
477 );
477
478 );
478
479 );
479
480 );
480
481 );
481
482 );
482
483 );
483
484 );
484
485 );
485
486 );
486
487 );
487
488 );
488
489 );
489
490 );
490
491 );
491
492 );
492
493 );
493
494 );
494
495 );
495
496 );
496
497 );
497
498 );
498
499 );
499
500 );
500
501 );
501
502 );
502
503 );
503
504 );
504
505 );
505
506 );
506
507 );
507
508 );
508
509 );
509
510 );
510
511 );
511
512 );
512
513 );
513
514 );
514
515 );
515
516 );
516
517 );
517
518 );
518
519 );
519
520 );
520
521 );
521
522 );
522
523 );
523
524 );
524
525 );
525
526 );
526
527 );
527
528 );
528
529 );
529
530 );
530
531 );
531
532 );
532
533 );
533
534 );
534
535 );
535
536 );
536
537 );
537
538 );
538
539 );
539
540 );
540
541 );
541
542 );
542
543 );
543
544 );
544
545 );
545
546 );
546
547 );
547
548 );
548
549 );
549
550 );
550
551 );
551
552 );
552
553 );
553
554 );
554
555 );
555
556 );
556
557 );
557
558 );
558
559 );
559
560 );
560
561 );
561
562 );
562
563 );
563
564 );
564
565 );
565
566 );
566
567 );
567
568 );
568
569 );
569
570 );
570
571 );
571
572 );
572
573 );
573
574 );
574
575 );
575
576 );
576
577 );
577
578 );
578
579 );
579
580 );
580
581 );
581
582 );
582
583 );
583
584 );
584
585 );
585
586 );
586
587 );
587
588 );
588
589 );
589
590 );
590
591 );
591
592 );
592
593 );
593
594 );
594
595 );
595
596 );
596
597 );
597
598 );
598
599 );
599
600 );
600
601 );
601
602 );
602
603 );
603
604 );
604
605 );
605
606 );
606
607 );
607
608 );
608
609 );
609
610 );
610
611 );
611
612 );
612
613 );
613
614 );
614
615 );
615
616 );
616
617 );
617
618 );
618
619 );
619
620 );
620
621 );
621
622 );
622
623 );
623
624 );
624
625 );
625
626 );
626
627 );
627
628 );
628
629 );
629
630 );
630
631 );
631
632 );
632
633 );
633
634 );
634
635 );
635
636 );
636
637 );
637
638 );
638
639 );
639
640 );
640
641 );
641
642 );
642
643 );
643
644 );
644
645 );
645
646 );
646
647 );
647
648 );
648
649 );
649
650 );
650
651 );
651
652 );
652
653 );
653
654 );
654
655 );
655
656 );
656
657 );
657
658 );
658
659 );
659
660 );
660
661 );
661
662 );
662
663 );
663
664 );
664
665 );
665
666 );
666
667 );
667
668 );
668
669 );
669
670 );
670
671 );
671
672 );
672
673 );
673
674 );
674
675 );
675
676 );
676
677 );
677
678 );
678
679 );
679
680 );
680
681 );
681
682 );
682
683 );
683
684 );
684
685 );
685
686 );
686
687 );
687
688 );
688
689 );
689
690 );
690
691 );
691
692 );
692
693 );
693
694 );
694
695 );
695
696 );
696
697 );
697
698 );
698
699 );
699
700 );
700
701 );
701
702 );
702
703 );
703
704 );
704
705 );
705
706 );
706
707 );
707
708 );
708
709 );
709
710 );
710
711 );
711
712 );
712
713 );
713
714 );
714
715 );
715
716 );
716
717 );
717
718 );
718
719 );
719
720 );
720
721 );
721
722 );
722
723 );
723
724 );
724
725 );
725
726 );
726
727 );
727
728 );
728
729 );
729
730 );
730
731 );
731
732 );
732
733 );
733
734 );
734
735 );
735
736 );
736
737 );
737
738 );
738
739 );
739
740 );
740
741 );
741
742 );
742
743 );
743
744 );
744
745 );
745
746 );
746
747 );
747
748 );
748
749 );
749
750 );
750
751 );
751
752 );
752
753 );
753
754 );
754
755 );
755
756 );
756
757 );
757
758 );
758
759 );
759
760 );
760
761 );
761
762 );
762
763 );
763
764 );
764
765 );
765
766 );
766
767 );
767
768 );
768
769 );
769
770 );
770
771 );
771
772 );
772
773 );
773
774 );
774
775 );
775
776 );
776
777 );
777
778 );
778
779 );
779
780 );
780
781 );
781
782 );
782
783 );
783
784 );
784
785 );
785
786 );
786
787 );
787
788 );
788
789 );
789
790 );
790
791 );
791
792 );
792
793 );
793
794 );
794
795 );
795
796 );
796
797 );
797
798 );
798
799 );
799
800 );
800
801 );
801
802 );
802
803 );
803
804 );
804
805 );
805
806 );
806
807 );
807
808 );
808
809 );
809
810 );
810
811 );
811
812 );
812
813 );
813
814 );
814
815 );
815
816 );
816
817 );
817
818 );
818
819 );
819
820 );
820
821 );
821
822 );
822
823 );
823
824 );
824
825 );
825
826 );
826
827 );
827
828 );
828
829 );
829
830 );
830
831 );
831
832 );
832
833 );
833
834 );
834
835 );
835
836 );
836
837 );
837
838 );
838
839 );
839
840 );
840
841 );
841
842 );
842
843 );
843
844 );
844
845 );
845
846 );
846
847 );
847
848 );
848
849 );
849
850 );
850
851 );
851
852 );
852
853 );
853
854 );
854
855 );
855
856 );
856
857 );
857
858 );
858
859 );
859
860 );
860
861 );
861
862 );
862
863 );
863
864 );
864
865 );
865
866 );
866
867 );
867
868 );
868
869 );
869
870 );
870
871 );
871
872 );
872
873 );
873
874 );
874
875 );
875
876 );
876
877 );
877
878 );
878
879 );
879
880 );
880
881 );
881
882 );
882
883 );
883
884 );
884
885 );
885
886 );
886
887 );
887
888 );
888
889 );
889
890 );
890
891 );
891
892 );
892
893 );
893
894 );
894
895 );
895
896 );
896
897 );
897
898 );
898
899 );
899
900 );
900
901 );
901
902 );
902
903 );
903
904 );
904
905 );
905
906 );
906
907 );
907
908 );
908
909 );
909
910 );
910
911 );
911
912 );
912
913 );
913
914 );
914
915 );
915
916 );
916
917 );
917
918 );
918
919 );
919
920 );
920
921 );
921
922 );
922
923 );
923
924 );
924
925 );
925
926 );
926
927 );
927
928 );
928
929 );
929
930 );
930
931 );
931
932 );
932
933 );
933
934 );
934
935 );
935
936 );
936
937 );
937
938 );
938
939 );
939
940 );
940
941 );
941
942 );
942
943 );
943
944 );
944
945 );
945
946 );
946
947 );
947
948 );
948
949 );
949
950 );
950
951 );
951
952 );
952
953 );
953
954 );
954
955 );
955
956 );
956
957 );
957
958 );
958
959 );
959
960 );
960
961 );
961
962 );
962
963 );
963
964 );
964
965 );
965
966 );
966
967 );
967
968 );
968
969 );
969
970 );
970
971 );
971
972 );
972
973 );
973
974 );
974
975 );
975
976 );
976
977 );
977
978 );
978
979 );
979
980 );
980
981 );
981
982 );
982
983 );
983
984 );
984
985 );
985
986 );
986
987 );
987
988 );
988
989 );
989
990 );
990
991 );
991
992 );
992
993 );
993
994 );
994
995 );
995
996 );
996
997 );
997
998 );
998
999 );
999
1000 );
1000
1001 );
1001
1002 );
1002
1003 );
1003
1004 );
1004
1005 );
1005
1006 );
1006
1007 );
1007
1008 );
1008
1009 );
1009
1010 );
1010
1011 );
1011
1012 );
1012
1013 );
1013
1014 );
1014
1015 );
1015
1016 );
1016
1017 );
1017
1018 );
1018
1019 );
1019
1020 );
1020
1021 );
1021
1022 );
1022
1023 );
1023
1024 );
1024
1025 );
1025
1026 );
1026
1027 );
1027
1028 );
1028
1029 );
1029
1030 );
1030
1031 );
1031
1032 );
1032
1033 );
1033
1034 );
1034
1035 );
1035
1036 );
1036
1037 );
1037
1038 );
1038
1039 );
1039
1040 );
1040
1041 );
1041
1042 );
1042
1043 );
1043
1044 );
1044
1045 );
1045
1046 );
1046
1047 );
1047
1048 );
1048
1049 );
1049
1050 );
1050
1051 );
1051
1052 );
1052
1053 );
1053
1054 );
1054
1055 );
1055
1056 );
1056
1057 );
1057
1058 );
1058
1059 );
1059
1060 );
1060
1061 );
1061
1062 );
1062
1063 );
1063
1064 );
1064
1065 );
1065
1066 );
1066
1067 );
1067
1068 );
1068
1069 );
1069
1070 );
1070
1071 );
1071
1072 );
1072
1073 );
1073
1074 );
1074
1075 );
1075
1076 );
1076
1077 );
1077
1078 );
1078
1079 );
1079
1080 );
1080
1081 );
1081
1082 );
1082
1083 );
1083
1084 );
1084
1085 );
1085
1086 );
1086
1087 );
1087
1088 );
1088
1089 );
1089
1090 );
1090
1091 );
1091
1092 );
1092
1093 );
1093
1094 );
1094
1095 );
1095
1096 );
1096
1097 );
1097
1098 );
1098
1099 );
1099
1100 );
1100
1101 );
1101
1102 );
1102
1103 );
1103
1104 );
1104
1105 );
1105
1106 );
1106
1107 );
1107
1108 );
1108
1109 );
1109
1110 );
1110
1111 );
1111
1112 );
1112
1113 );
1113
1114 );
1114
1115 );
1115
1116 );
1116
1117 );
1117
1118 );
1118
1119 );
1119
1120 );
1120
1121 );
1121
1122 );
1122
1123 );
1123
1124 );
1124
1125 );
1125
1126 );
1126
1127 );
1127
1128 );
1128
1129 );
1129
1130 );
1130
1131 );
1131
1132 );
1132
1133 );
1133
1134 );
1134
1135 );
1135
1136 );
1136
1137 );
1137
1138 );
1138
1139 );
1139
1140 );
1140
1141 );
1141
1142 );
1142
1143 );
1143
1144 );
1144
1145 );
1145
1146 );
1146
1147 );
1147
1148 );
1148
1149 );
1149
1150 );
1150
1151 );
1151
1152 );
1152
1153 );
1153
1154 );
1154
1155 );
1155
1156 );
1156
1157 );
1157
1158 );
1158
1159 );
1159
1160 );
1160
1161 );
1161
1162 );
1162
1163 );
1163
1164 );
1164
1165 );
1165
1166 );
1166
1167 );
1167
1168 );
1168
1169 );
1169
1170 );
1170
1171 );
1171
1172 );
1172
1173 );
1173
1174 );
1174
1175 );
1175
1176 );
1176
1177 );
1177
1178 );
1178
1179 );
1179
1180 );
1180
1181 );
1181
1182 );
1182
1183 );
1183
1184 );
1184
1185 );
1185
1186 );
1186
1187 );
1187
1188 );
1188
1189 );
1189
1190 );
1190
1191 );
1191
1192 );
1192
1193 );
1193
1194 );
1194
1195 );
1195
1196 );
1196
1197 );
1197
1198 );
1198
1199 );
1199
1200 );
1200
1201 );
1201
1202 );
1202
1203 );
1203
1204 );
1204
1205 );
1205
1206 );
1206
1207 );
1207
1208 );
1208
1209 );
1209
1210 );
1210
1211 );
1211
1212 );
1212
1213 );
1213
1214 );
1214
1215 );
1215
1216 );
1216
1217 );
1217
1218 );
1218
1219 );
1219
1220 );
1220
1221 );
1221
1222 );
1222
1223 );
1223
1224 );
1224
1225 );
1225
1226 );
1226
1227 );
1227
1228 );
1228
1229 );
1229
1230 );
1230
1231 );
1231
1232 );
1232
1233 );
1233
1234 );
1234
1235 );
1235
1236 );
1236
1237 );
1237
1238 );
1238
1239 );
1239
1240 );
1240
1241 );
1241
1242 );
1242
1243 );
1243
1244 );
1244
1245 );
1245
1246 );
1246
1247 );
1247
1248 );
1248
1249 );
1249
1250 );
1250
1251 );
1251
1252 );
1252
1253 );
1253
1254 );
1254
1255 );
1255
1256 );
1256
1257 );
1257
1258 );
1258
1259 );
1259
1260 );
1260
1261 );
1261
1262 );
1262
1263 );
1263
1264 );
1264
1265 );
1265
1266 );
1266
1267 );
1267
1268 );
1268
1269 );
1269
1270 );
1270
1271 );
1271
1272 );
1272
1273 );
1273
1274 );
1274
1275 );
1275
1276 );
1276
1277 );
1277
1278 );
1278
1279 );
1279
1280 );
1280
1281 );
1281
1282 );
1282
1283 );
1283
1284 );
1284
1285 );
1285
1286 );
1286
1287 );
1287
1288 );
1288
1289 );
1289
1290 );
1290
1291 );
1291
1292 );
1292
1293 );
1293
1294 );
1294
1295 );
1295
1296 );
1296
1297 );
1297
1298 );
1298
1299 );
1299
1300 );
1300
1301 );
1301
1302 );
1302
1303 );
1303
1304 );
1304
1305 );
1305
1306 );
1306
1307 );
1307
1308 );
1308
1309 );
1309
1310 );
1310
1311 );
1311
1312 );
1312
1313 );
1313
1314 );
1314
1315 );
1315
1316 );
1316
1317 );
1317
1318 );
1318
1319 );
1319
1320 );
1320
1321 );
1321
1322 );
1322
1323 );
1323
1324 );
1324
1325 );
1325
1326 );
1326
1327 );
1327
1328 );
1328
1329 );
1329
1330 );
1330
1331 );
1331
1332 );
1332
1333 );
1333
1334 );
1334
1335 );
1335
1336 );
1336
1337 );
1337
1338 );
1338
1339 );
1339
1340 );
1340
1341 );
1341
1342 );
1342
1343 );
1343
1344 );
1344
1345 );
1345
1346 );
1346
1347 );
1347
1348 );
1348
1349 );
1349
1350 );
1350
1351 );
1351
1352 );
1352
1353 );
1353
1354 );
1354
1355 );
1355
1356 );
1356
1357 );
1357
```

## Appendix F

**SSLCOMMERZ** Technical or Backend Integration Process. This diagram explain how our payment system is working.



Notifications with Expo's Push API. This diagram show how backend server will send notification to the user app by Expo server.



## Appendix G

1/4/23, 11:59 AM

Turnitin - Originality Report - 191-15-12594

### Turnitin Originality Report

Processed on: 04-Jan-2023 11:58 +06

ID: 1988429762

Word Count: 8475

Submitted: 1

191-15-12594 By Md. Assadujjaman  
Tilok

Similarity Index  
**22%**

Similarity by Source  
Internet Sources: 17%  
Publications: 1%  
Student Papers: 18%

2% match (Internet from 15-Apr-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7771/182-15-11440%20%2823%25%29.pdf?isAllowed=y&sequence=1>

2% match (student papers from 02-Jan-2023)  
[Submitted to Daffodil International University on 2023-01-02](#)

2% match (Internet from 04-Sep-2022)  
<https://businessinspection.com.bd/home-appliance-industry-in Bd/>

1% match (Internet from 21-Nov-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6798/172-15-9824%20%20%2821%25%29.pdf?isAllowed=y&sequence=1>

1% match (Internet from 26-Oct-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7395/173-15-10295%20%2822%25%29.pdf?isAllowed=y&sequence=1>

1% match (Internet from 27-Jan-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6738/142-15-3747%20%2823%25%29%20clearance.pdf?isAllowed=y&sequence=1>

1% match (student papers from 28-Sep-2011)  
[Submitted to Higher Education Commission Pakistan on 2011-09-28](#)

1% match (student papers from 26-Sep-2011)  
[Submitted to Higher Education Commission Pakistan on 2011-09-26](#)

1% match (student papers from 07-Jan-2022)  
[Submitted to National Open University of Nigeria on 2022-01-07](#)

1% match (student papers from 28-Nov-2014)  
[Submitted to New University of Astana on 2014-11-28](#)

1% match (student papers from 31-May-2019)  
[Submitted to Kelvin Grove State College on 2019-05-31](#)

1% match (student papers from 25-Sep-2020)  
[Submitted to University of Portsmouth on 2020-09-25](#)

< 1% match (Internet from 13-Feb-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6796/172-15-9833%20%2820%25%29%20clearance.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 21-Nov-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6799/172-15-9821%20%2812%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 20-Nov-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5726/163-15-8529%20%2820%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 20-Nov-2022)