

**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. FOKHRAI 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

  
5/02/23

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

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

**Chairman**

  
29.01.23

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

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

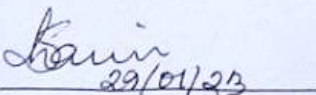
**Internal Examiner**

  
29.01.23

**Saiful Islam**  
**Assistant Professor**

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

**Internal Examiner**

  
29/01/23

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

Department of Computer Science and Engineering  
East West University

**External Examiner**

## DECLARATION

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

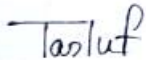
**Supervised by:**



---

**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%. [1]

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: INTRODUCTION</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	4
<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>5-8</b>
2.1 Introduction	5
2.2 Literature Review	5
2.3 Comparative Analysis	7
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>APPENDEX</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

Figures	Page No.
Figure 01 Business process model	09
Figure 02 Use Case Diagram	14
Figure 03 Admin Activity Diagram	21
Figure 04 Customer Activity Diagram	22
Figure 05 Admin Registration Sequence Diagram	23
Figure 06 Admin Login Sequence Diagram	24
Figure 07 Admin Login with Google Sequence Diagram	25
Figure 08 Admin Profile Sequence Diagram	25
Figure 09 Admin Forget Password Sequence Diagram	26
Figure 10 Admin View Order Sequence Diagram	27
Figure 11 Admin Add Agent Sequence Diagram	28
Figure 12 Admin Add Technician Sequence Diagram	28
Figure 13 Admin Update Order Sequence Diagram	29
Figure 14 Admin Update Agent Sequence Diagram	30
Figure 15 Admin Update Technician Sequence Diagram	31
Figure 16 Customer Registration Sequence Diagram	32
Figure 17 Customer Login Sequence Diagram	32
Figure 18 Customer Login with Google Sequence Diagram	33
Figure 19 Customer Profile Sequence Diagram	33
Figure 20 Customer Forget Password Sequence Diagram	34
Figure 21 Customer Add Order Sequence Diagram	35
Figure 22 Customer Add Address Sequence Diagram	35
Figure 23 Customer Update Address Sequence Diagram	36
Figure 24 Chart List Sequence Diagram	36

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

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

# CHAPTER 1

## INTRODUCTION

### 1.1 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. [2]

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 intend 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 to 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. [4]



**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. [4]

**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. [2] 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**

The importance of Electronic Product services in our society is increasing, which means that systematic approaches are more important to develop them. More and more services are now provided electronically through the internet. Companies strive to keep up with the ever-growing service sector by working diligently towards expert management of new service innovation endeavors. [14] Services make up a big sector of the global economy and trade in them is booming. Numerous electronic businesses are already in place that have migrated their traditional services and goods to the electronic world. Knowing the characteristics of each type of products is vital in this regard. [15]

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. [5] Some apps are doing B2B and some of them are doing B2C. We focused on B2B and B2C both with aggregation model. The key differentiator between electronic service and traditional service delivery is the means by which they are delivered, further they interact with their users with different methods. Services can only be obtained by customers visiting physical companies or stores, while they are able to get e-services via the electronic channels (Internet, PDA, Mobile phone) [6, 13]. E-service initiatives have been proven to deliver significant benefits, both for suppliers of electronic services (public authorities and organizations) and for the

public, to whom services are addressed [6]. Pavitt (2005) acknowledges that the continual improvement of services and the mapping of service artifacts to market needs are essential. However, given the lack of consideration for pervasive transparent feedback in electronic services, it has very fast cycle times. [14] The satisfaction of users can be seen as the sum of their feelings and attitudes towards several factors that affect their experience (Bailey et al., 1983).

Six composite dimensions of electronic product service quality have been identified. Including the provide of convenient and accurate electronic banking operations, ensure accessibility and reliability of service, implement effective queue management, personalize services, offer friendly and responsive customer service, and provide targeted customer service. [16] Quality drives long-term success and effective quality management is achieved through comprehensive measurement. The field of marketing academia is beginning to focus on quality measurement in the realm of electronic services. [17] Third-party administration of consumer ratings offers credibility to the information and increases trust from the consumer's perspective. This is because self-reported or advertised ratings by vendors are often viewed with suspicion, while ratings provided by an independent entity are seen as more credible. Furthermore, an independent entity managing vendor ratings can ensure standardization of data. [19]

An examination of existing literature on the topic shows that significant gaps in research exist both conceptually and methodologically. By drawing on previous research and the results of a qualitative study, we construct a hierarchical quality model for electronic services that is widely applicable. As a result, this app will be superior and distinct from others. By utilizing this application, we will partner with local businesses to preserve the local community. 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. [5] 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 e-valy, 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. [6]

### 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 crash. 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. [3] 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 layout that declares the functional and non-functional requirements, as well as includes a set of use case diagrams that describe user interactions with the software. [24]

### 3.2 Business Process Modeling

Business Process Model for our system are given below

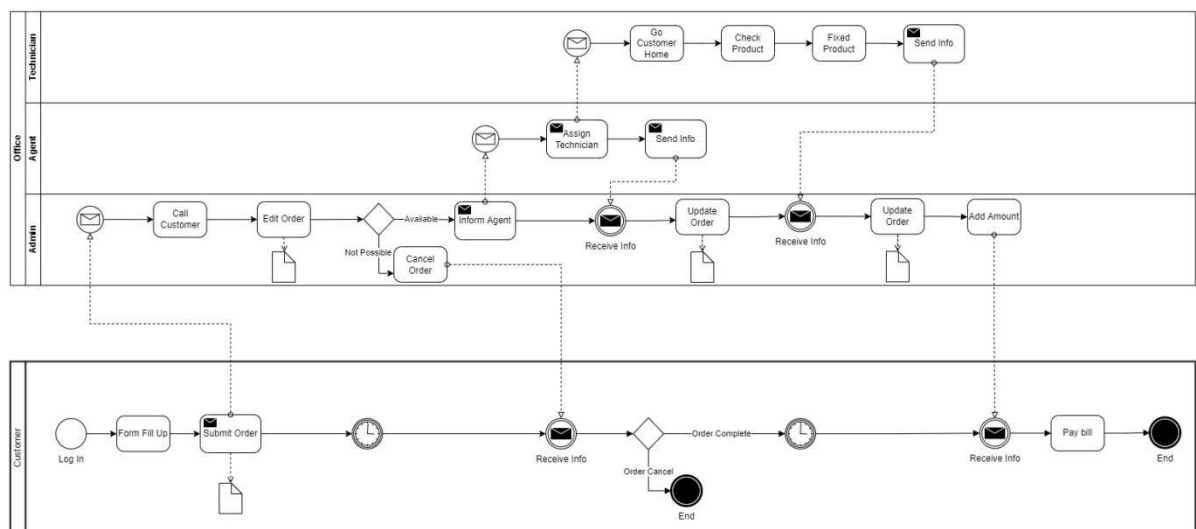


Figure 01: Business process model

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

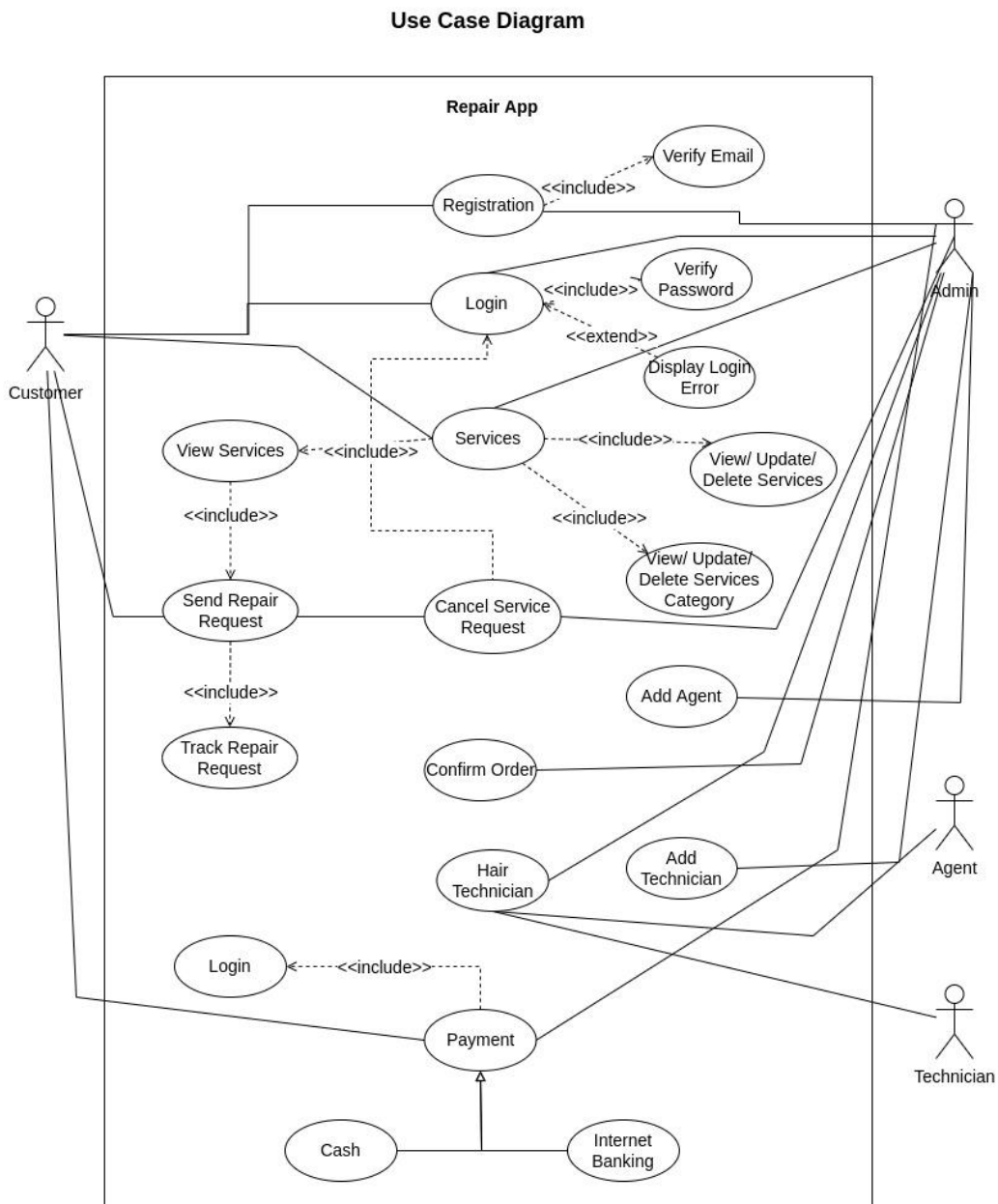


Figure 02: Use Case Diagram

### 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)

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

Table 02: Use Case Diagram (Login)

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

Table 03: Use Case Diagram (Service)

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

Table 04: Use Case Diagram (Cancel Request)

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

Table 05: Use Case Diagram (Confirm Order)

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

Table 06: Use Case Diagram (Hair Technician)

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

Table 07: Use Case Diagram (Payment)



<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.

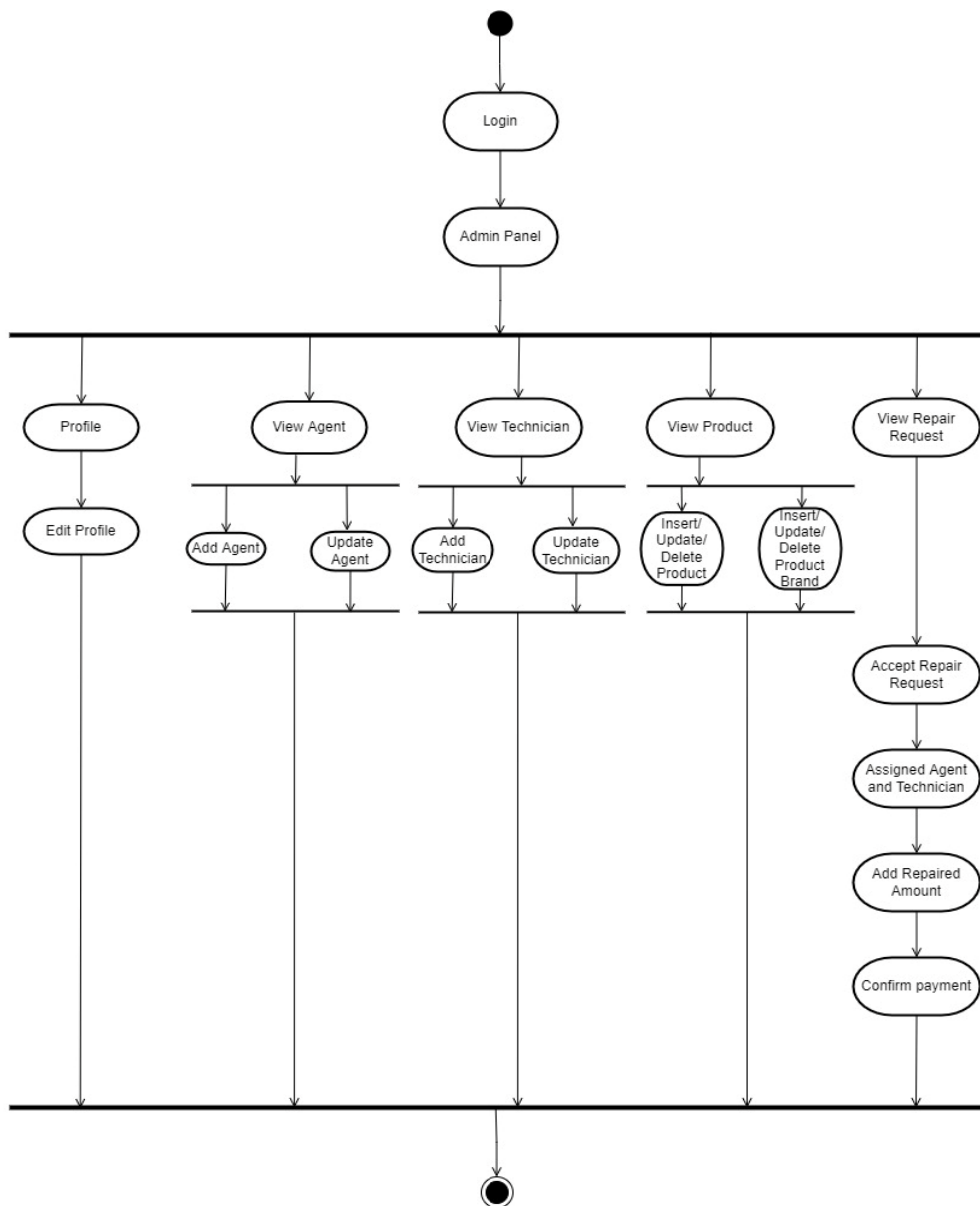


Figure 03: Admin Activity Diagram

### 3.5.2 Customer Activity Diagram

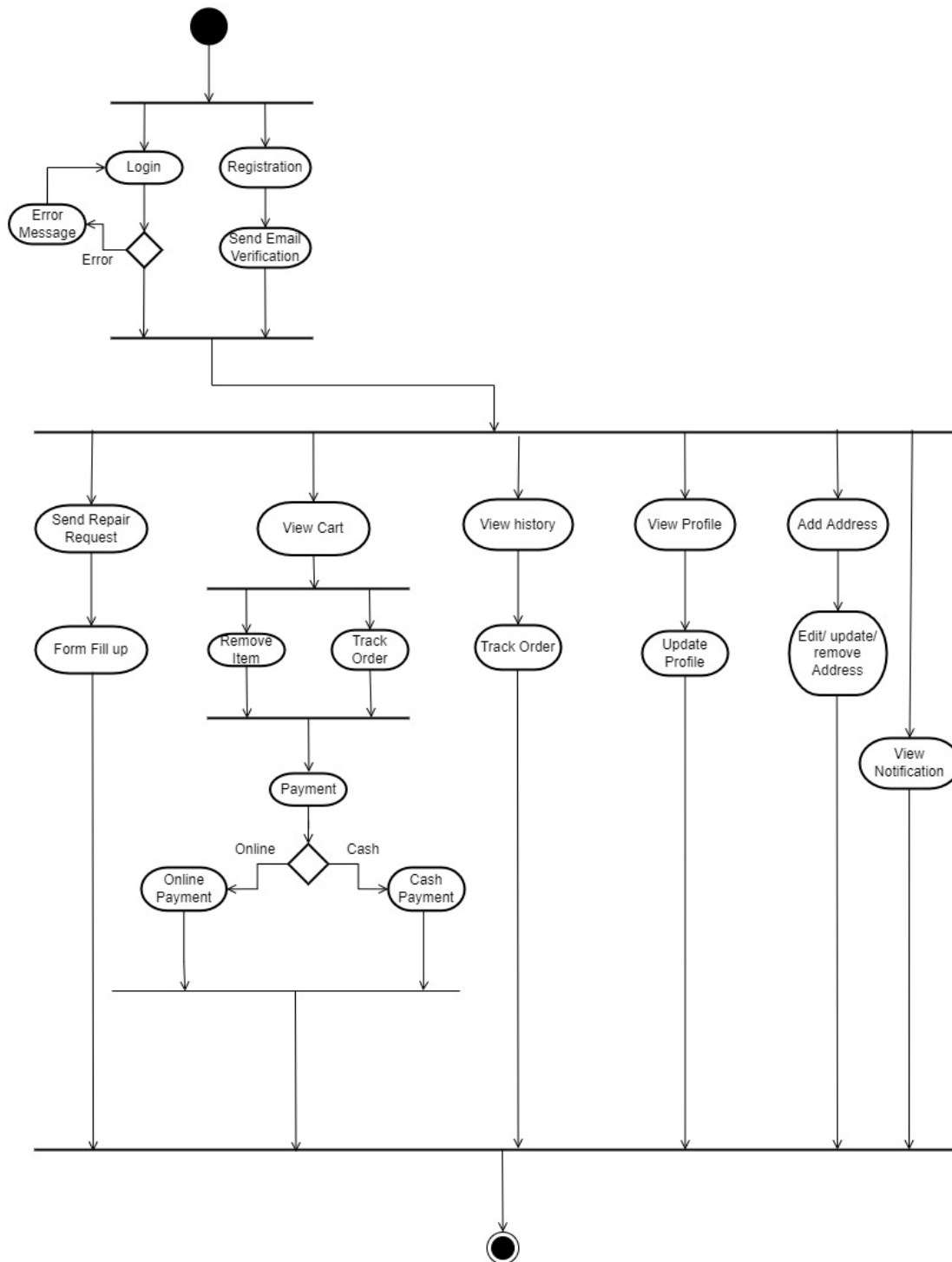


Figure 04: Customer Activity Diagram

### 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

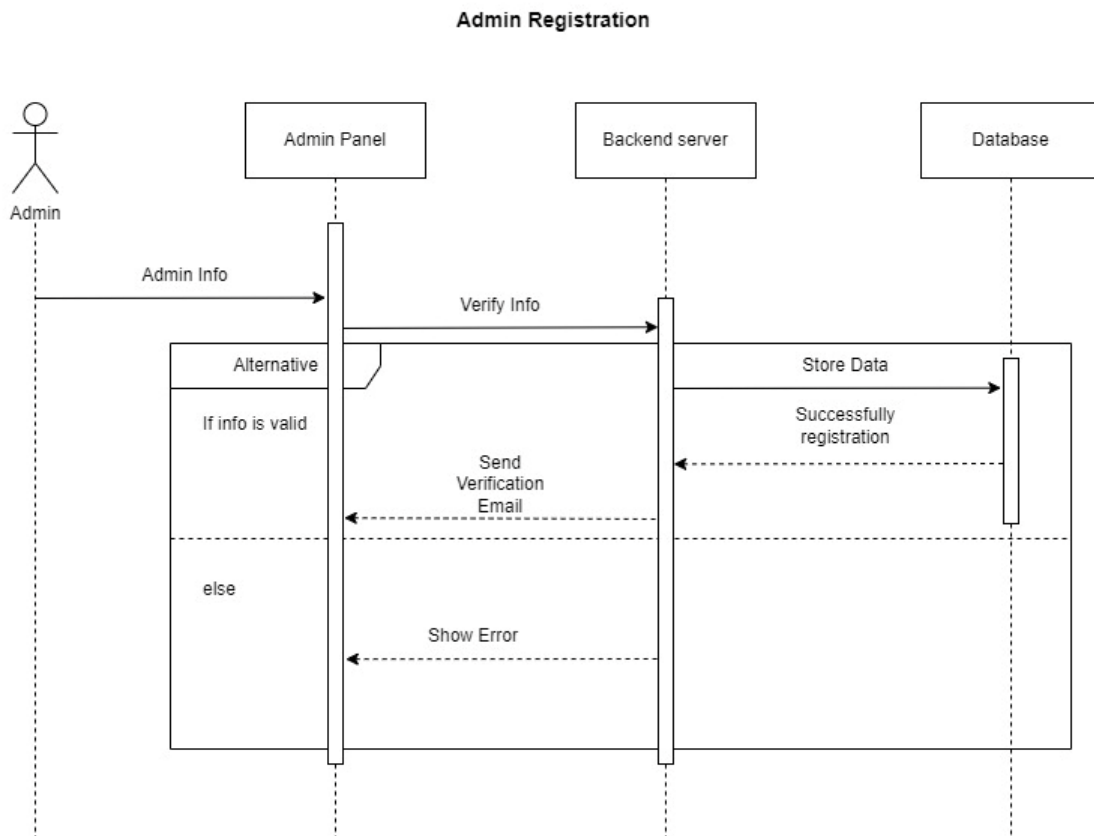


Figure 05: Admin Registration Sequence Diagram

### 3.6.2 Admin Login

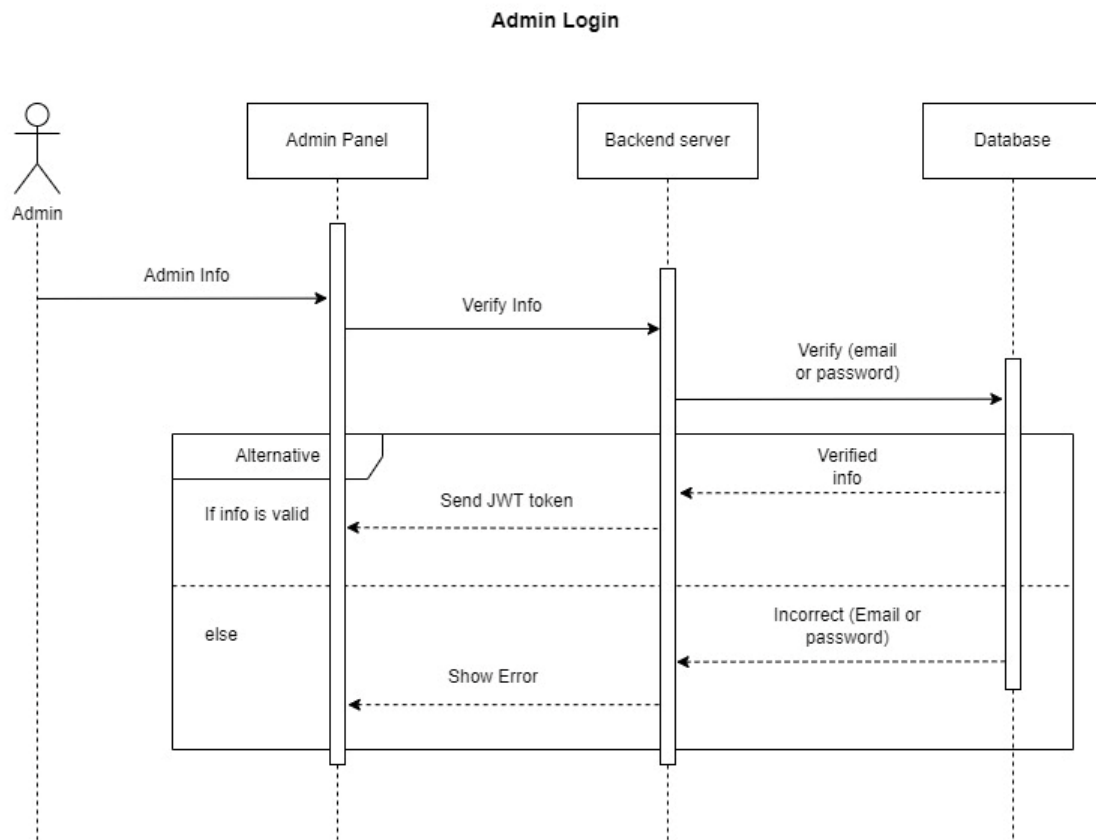


Figure 06: Admin Login Sequence Diagram

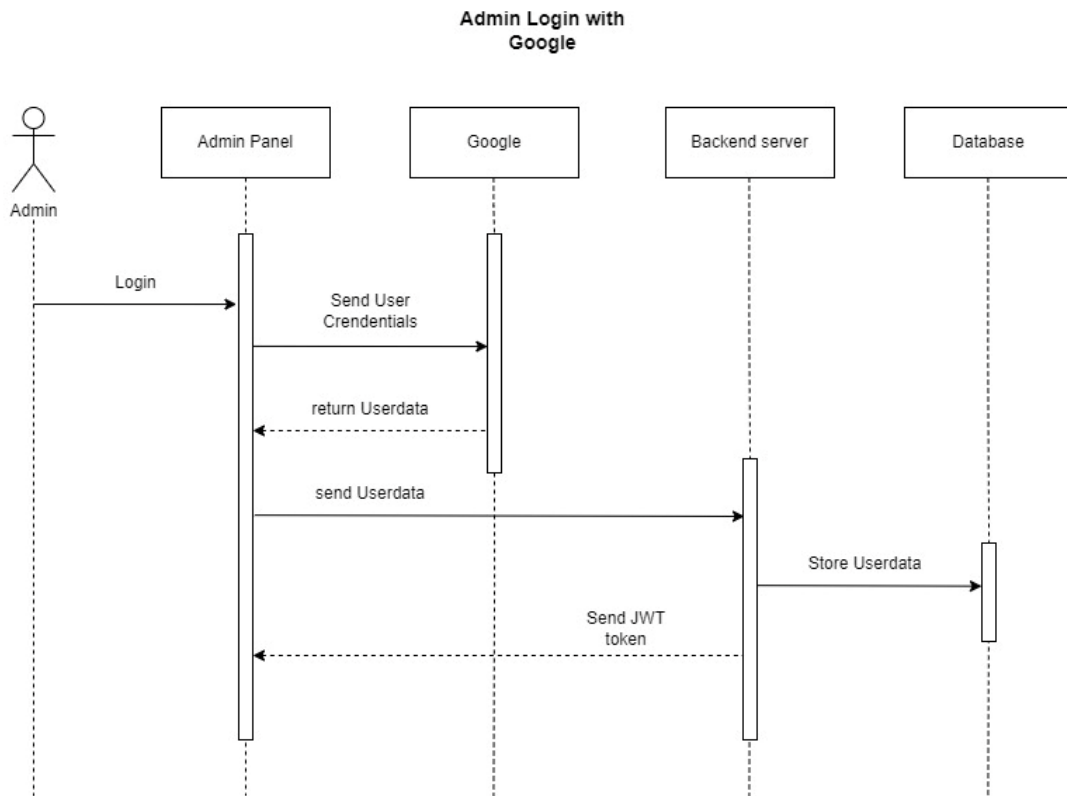


Figure 07: Admin Login with Google Sequence Diagram

### 3.6.3 Admin Profile

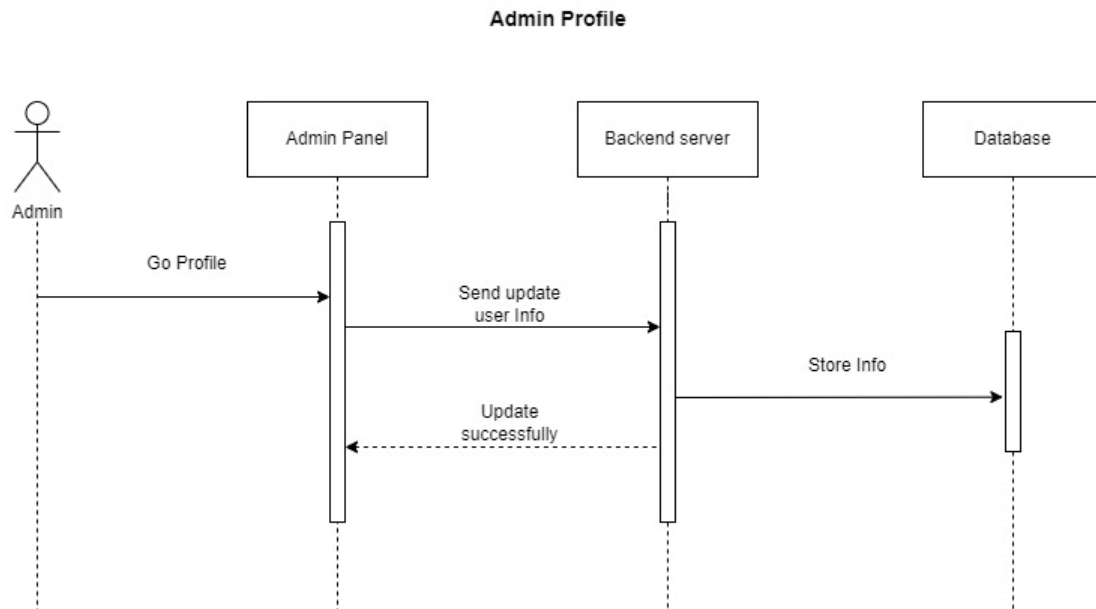


Figure 08: Admin Profile Sequence Diagram

### 3.6.4 Admin Forget Password

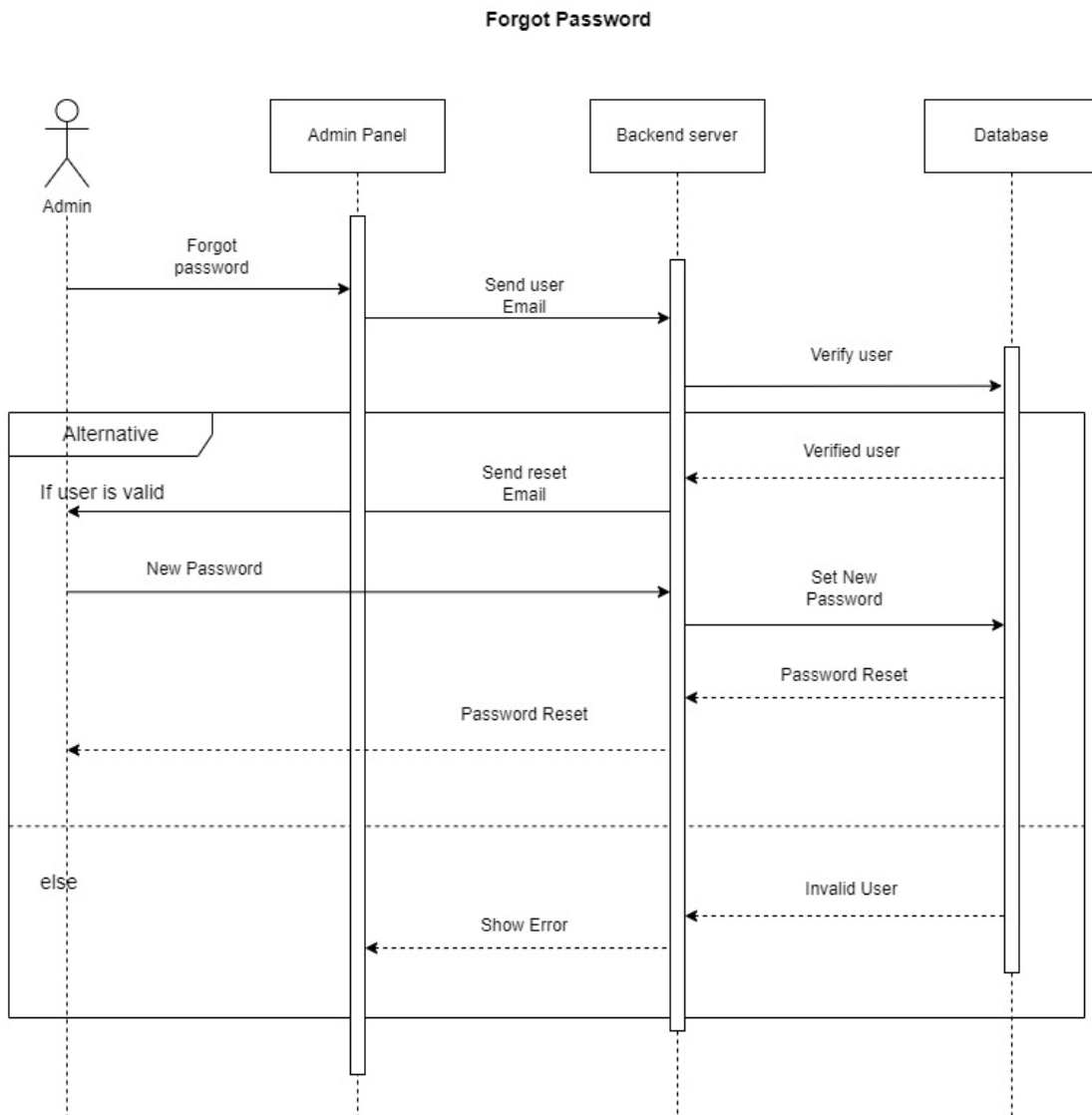


Figure 09: Admin Forget Password Sequence Diagram

### 3.6.5 Admin View Order

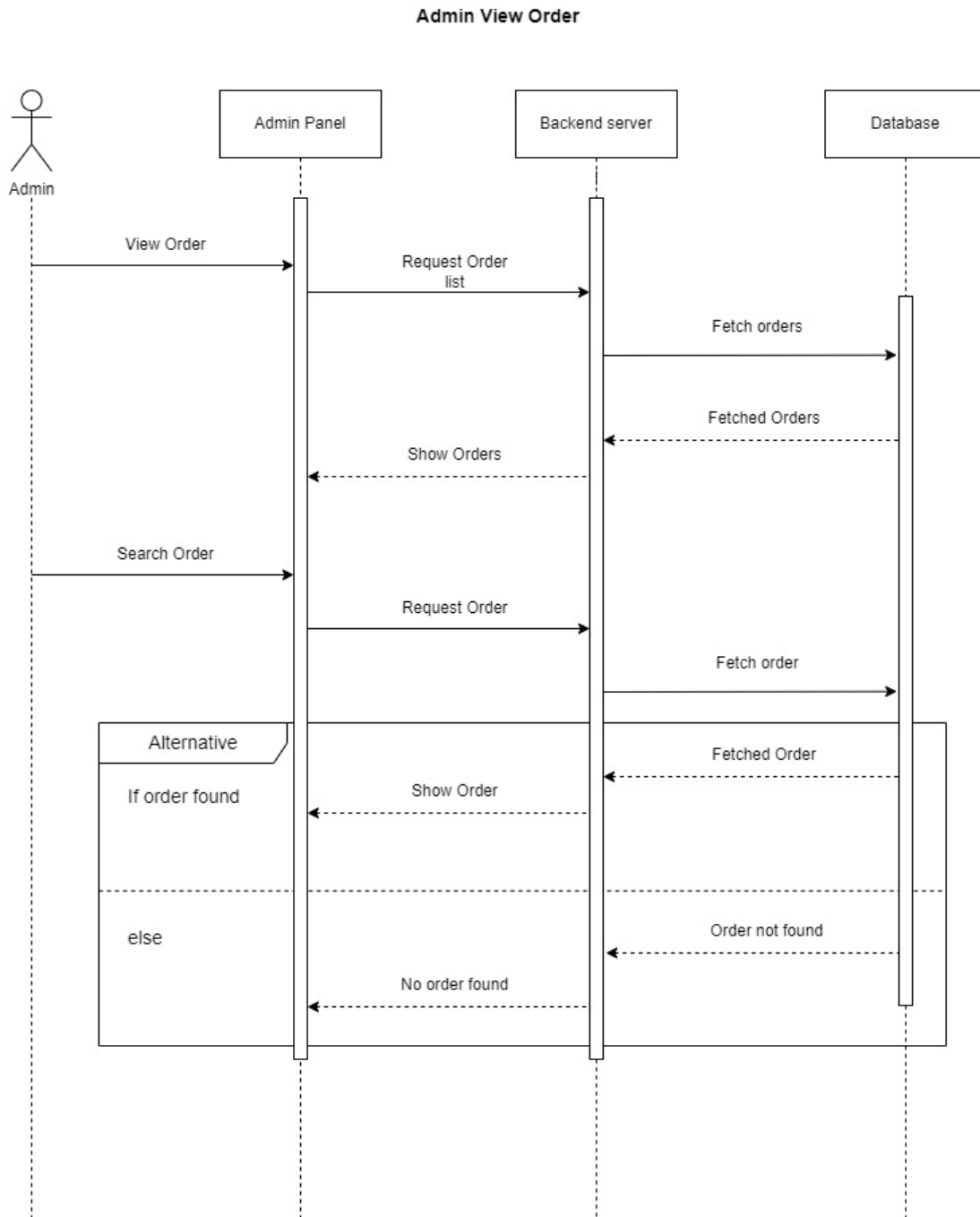


Figure 10: Admin View Order Sequence Diagram



### 3.6.6 Admin Add Agent

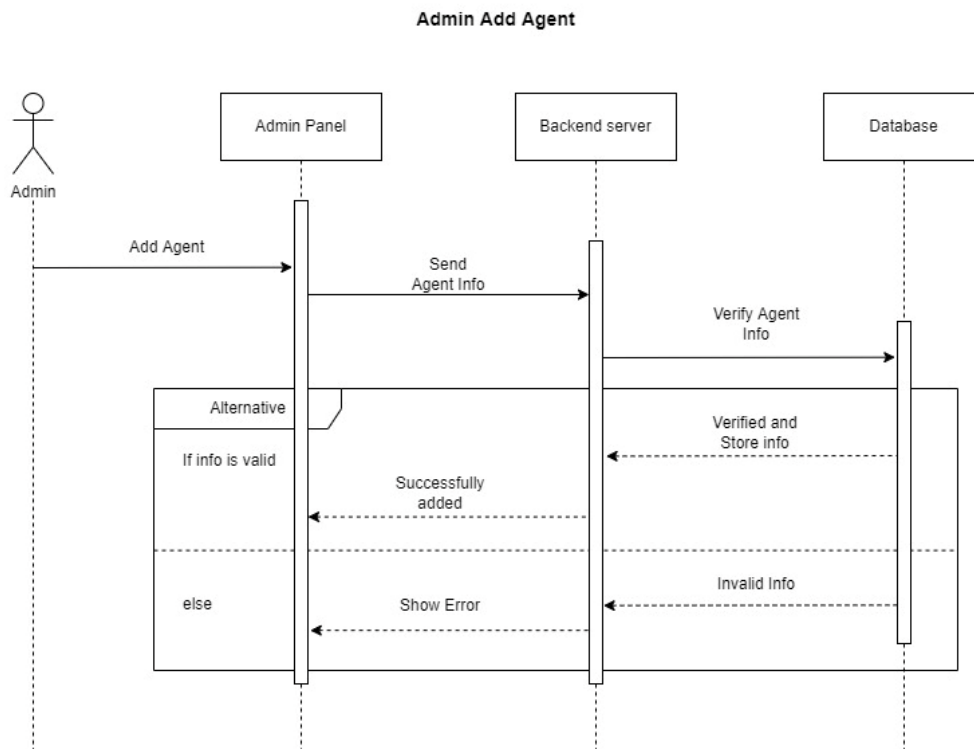


Figure 11: Admin Add Agent Sequence Diagram

### 3.6.7 Admin Add Technician

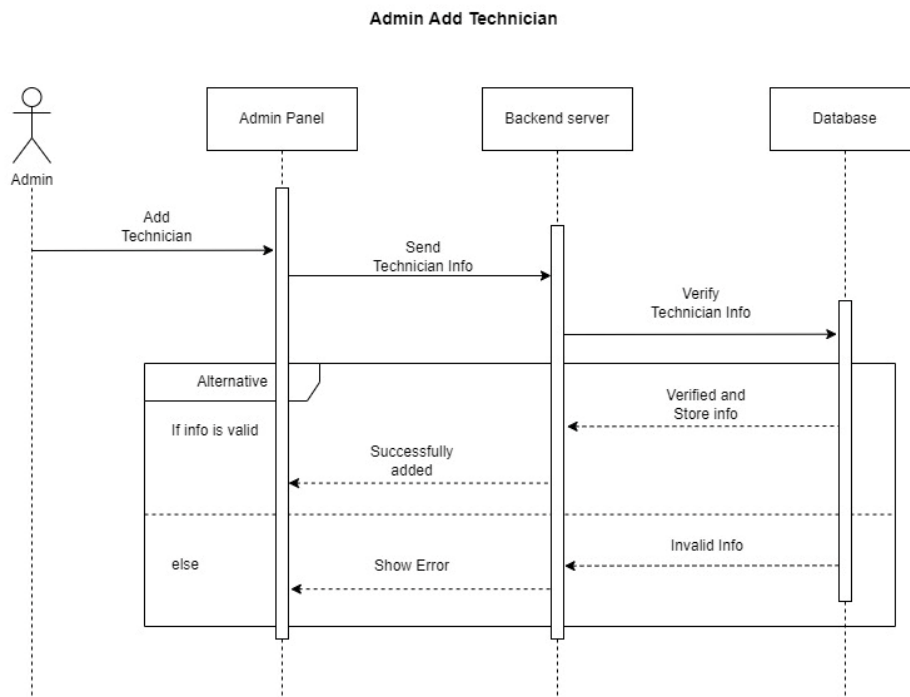


Figure 12: Admin Add Technician Sequence Diagram

### 3.6.8 Admin Update Order

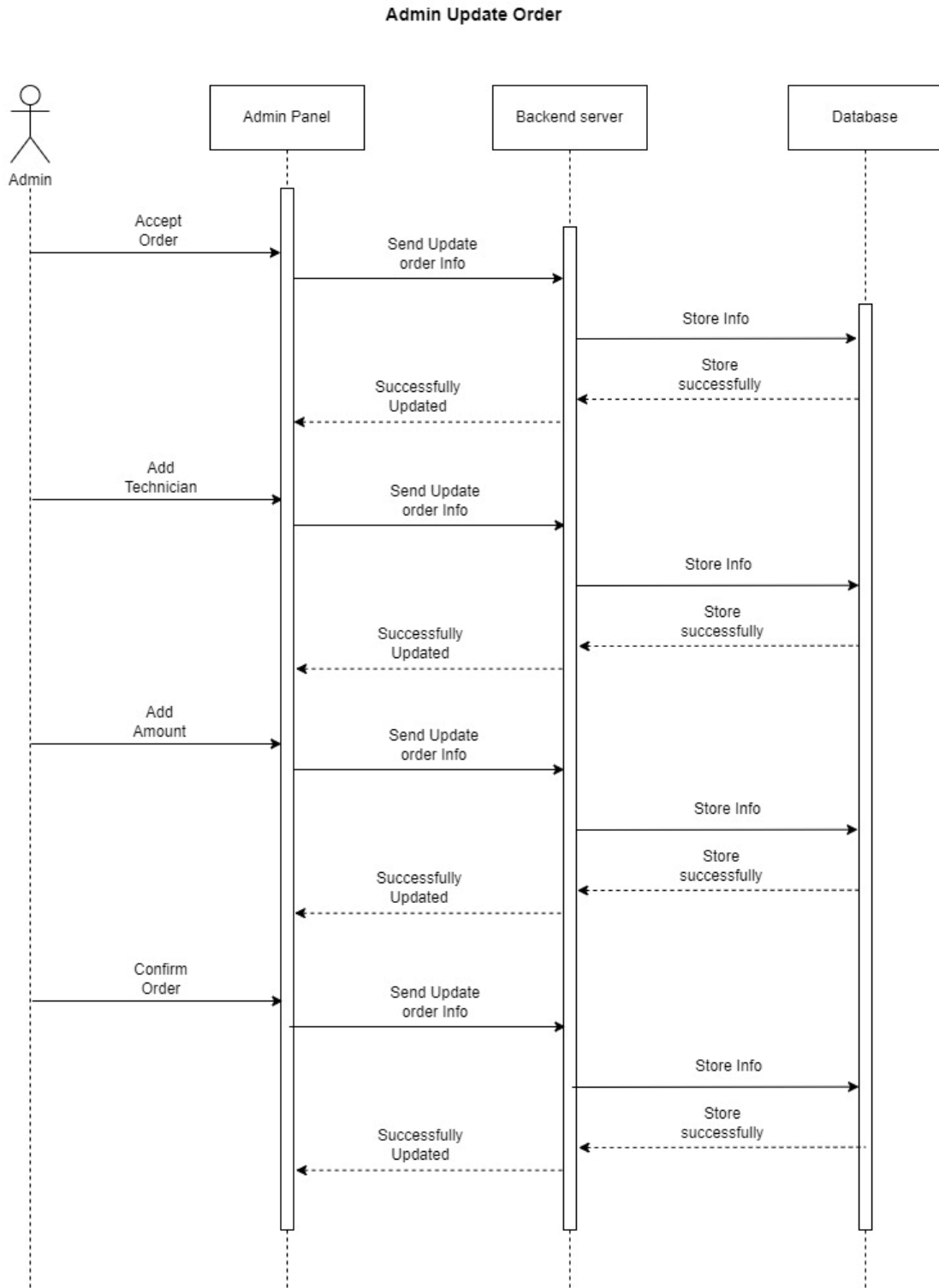


Figure 13: Admin Update Order Sequence Diagram

### 3.6.9 Admin Update Agent

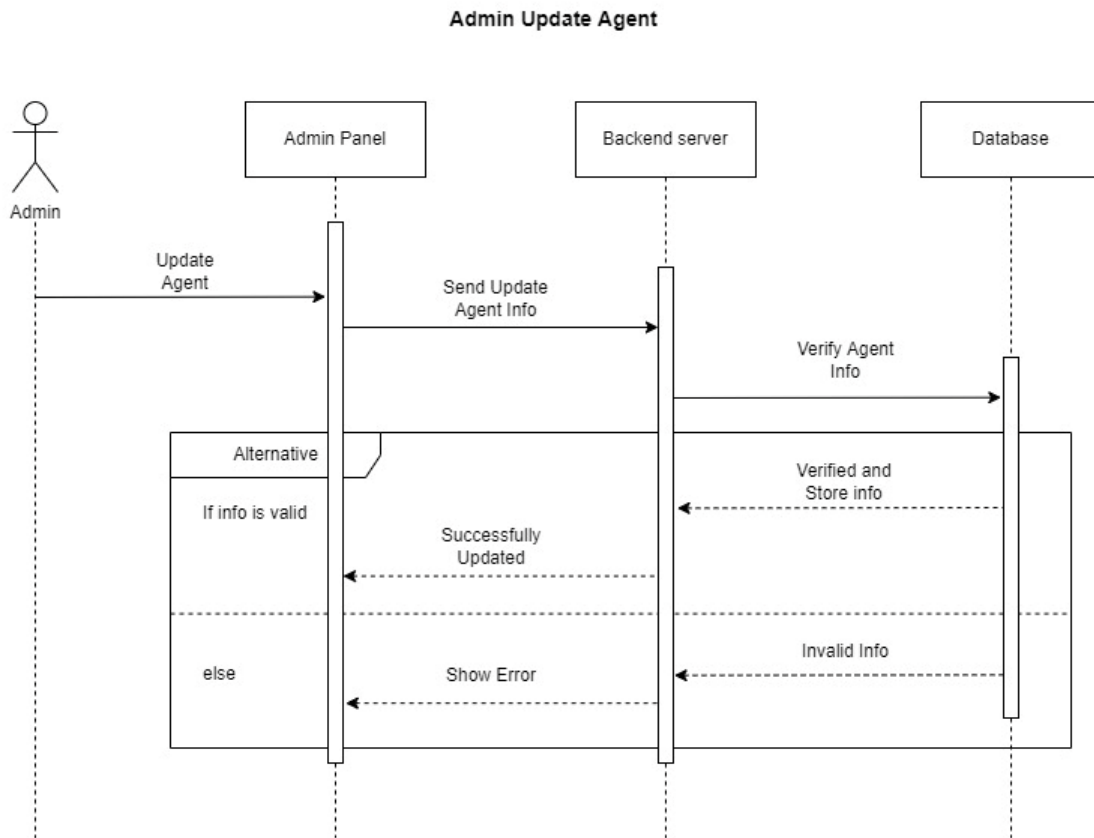


Figure 14: Admin Update Agent Sequence Diagram

### 3.6.10 Admin Update Technician

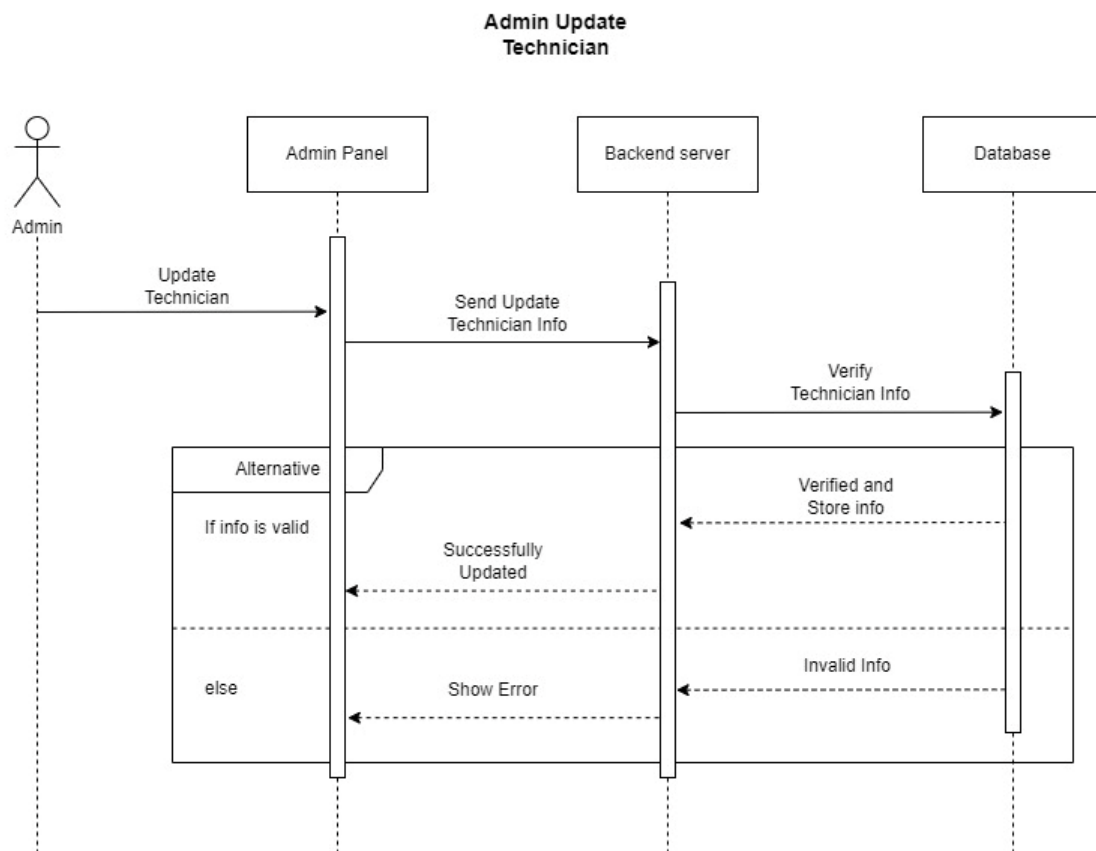


Figure 15: Admin Update Technician Sequence Diagram

### 3.6.11 Customer Registration

Customer Registration

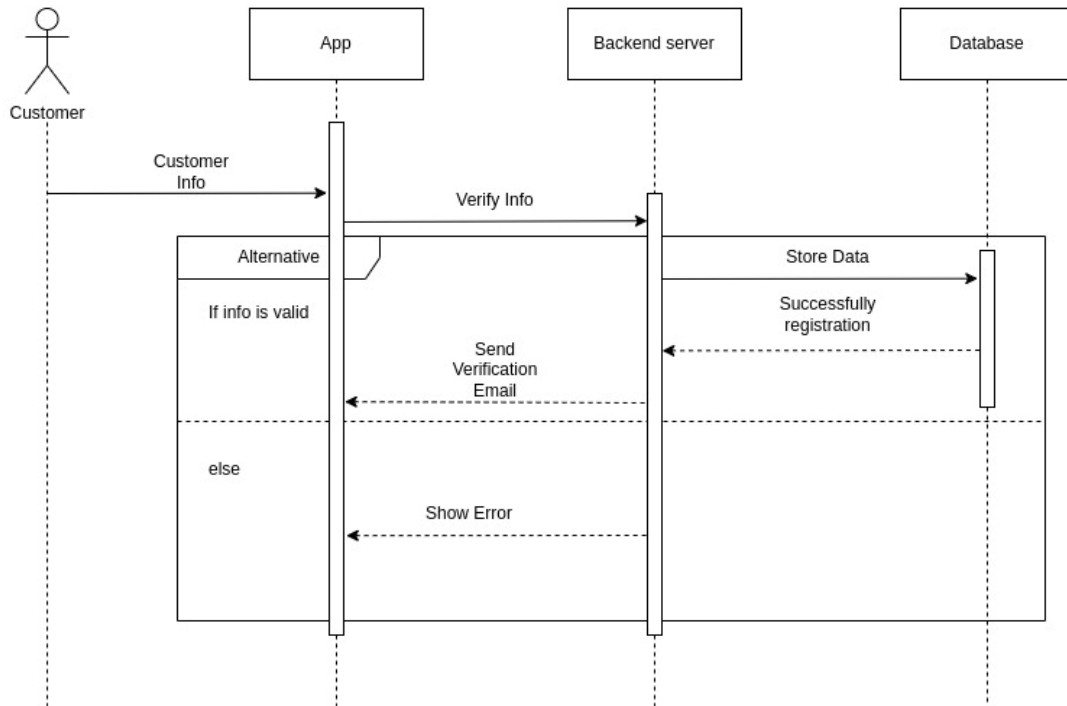


Figure 16: Customer Registration Sequence Diagram

### 3.6.12 Customer Login

Customer Login

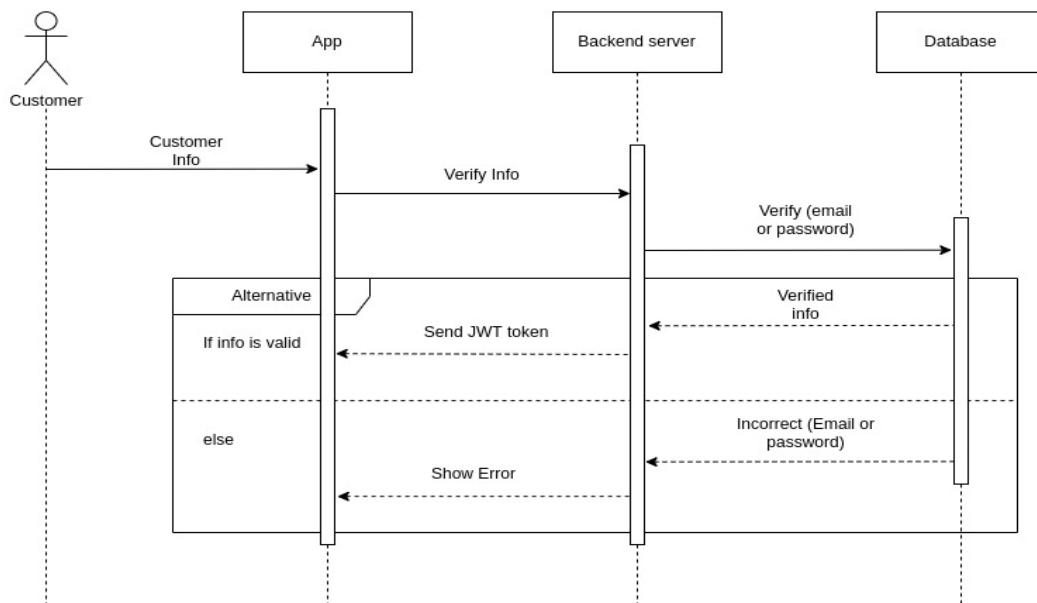


Figure 17: Customer Login Sequence Diagram

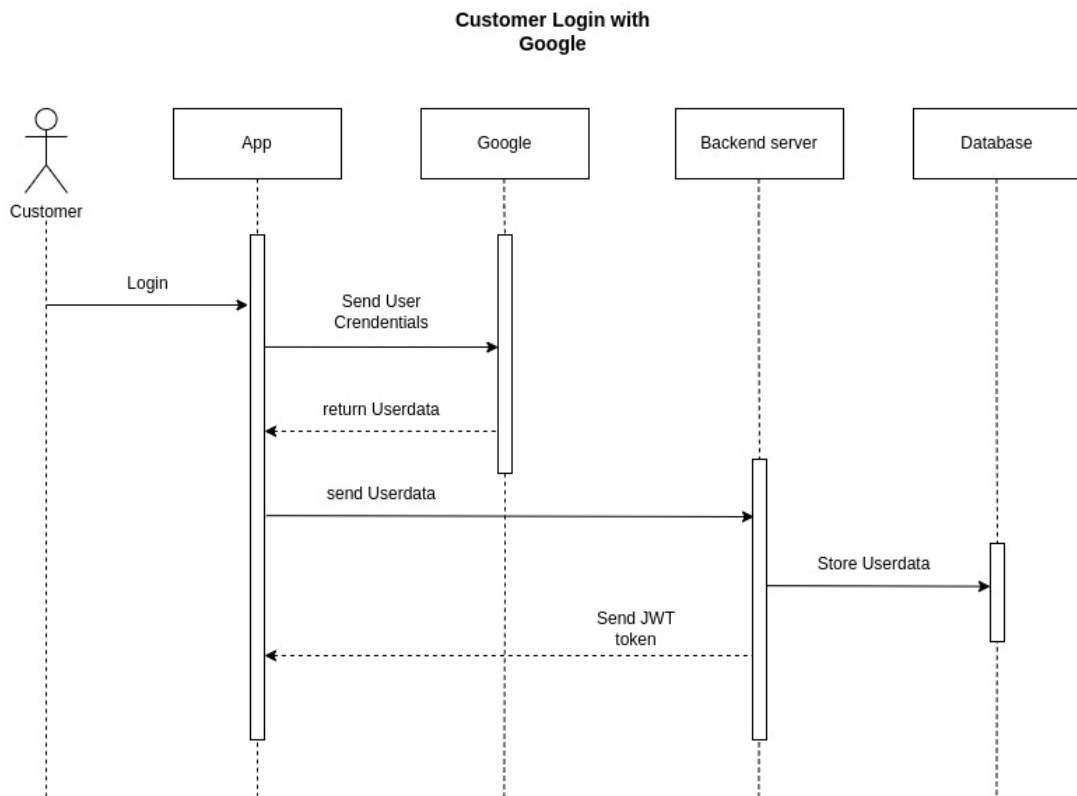


Figure 18: Customer Login with Google Sequence Diagram

### 3.6.13 Customer Profile

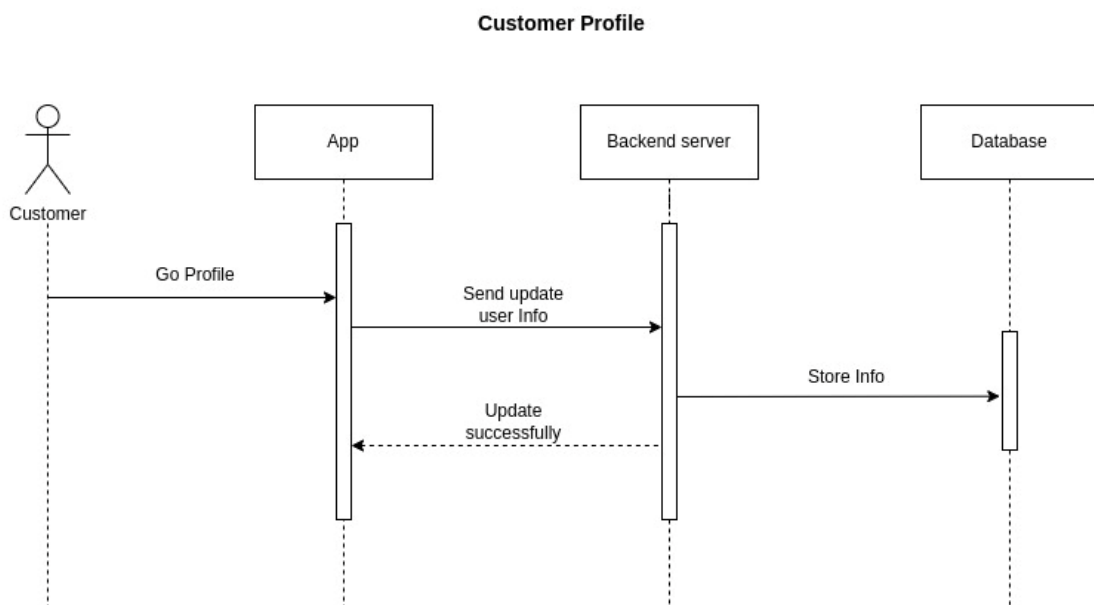


Figure 19: Customer Profile Sequence Diagram

### 3.6.14 Forget Password

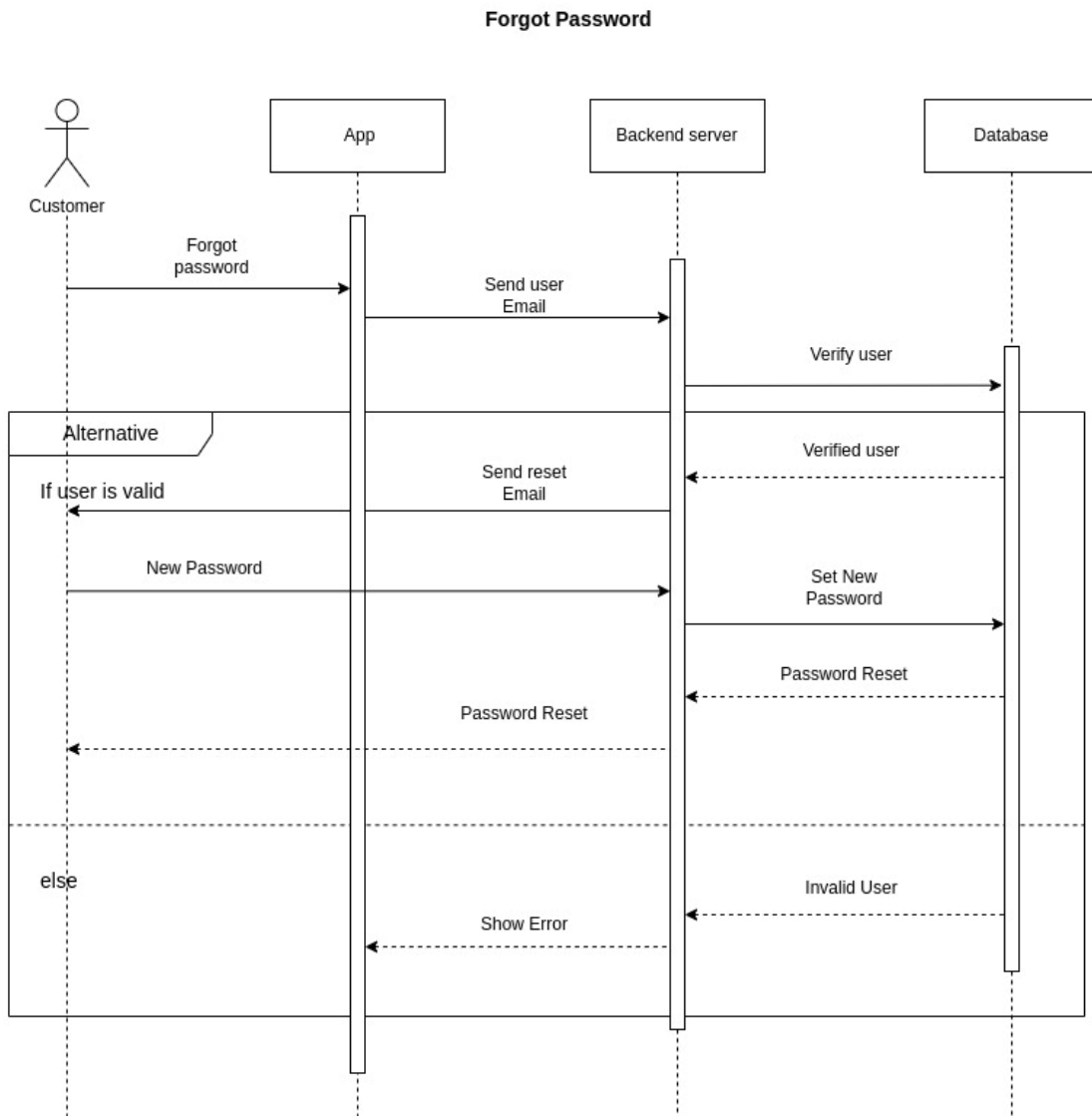


Figure 20: Customer Forget Password Sequence Diagram

### 3.6.15 Customer Add Order

Customer Add Order

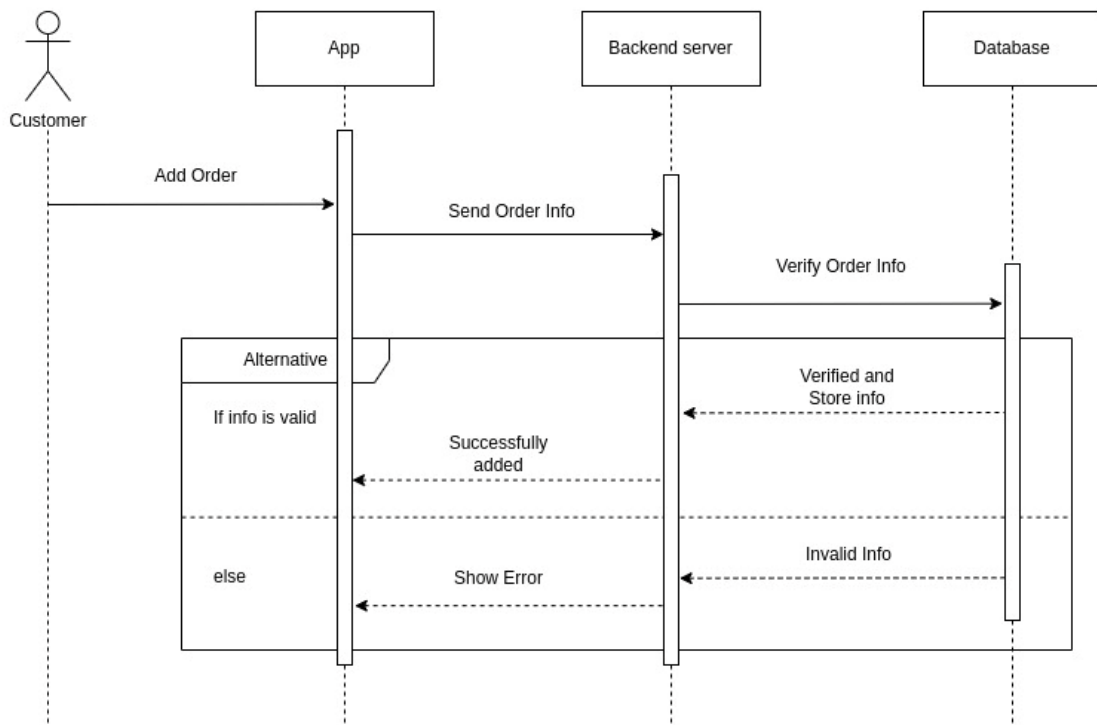


Figure 21: Customer Add Order Sequence Diagram

### 3.6.16 Customer Add Address

Customer Add Address

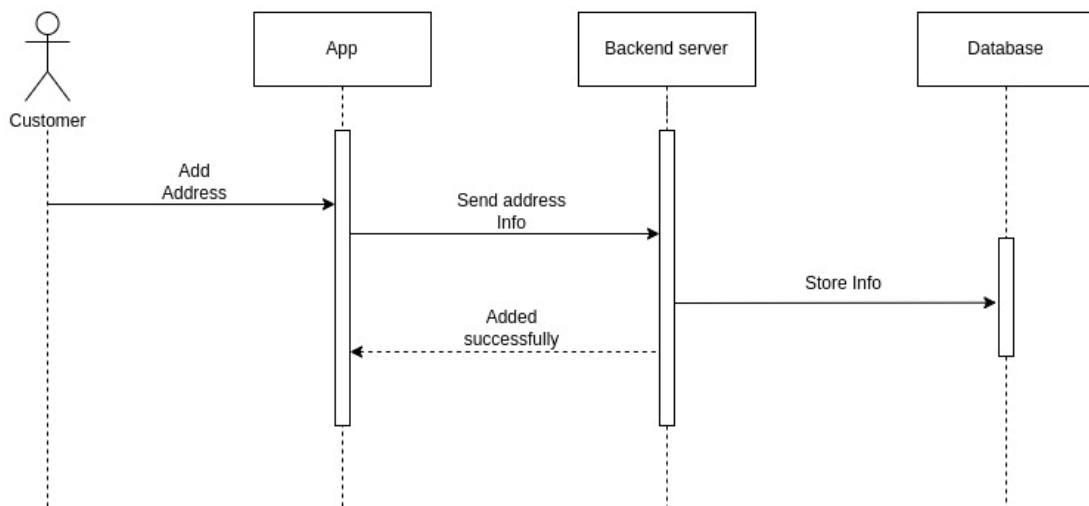


Figure 22: Customer Add Address Sequence Diagram



### 3.6.17 Customer Update Address

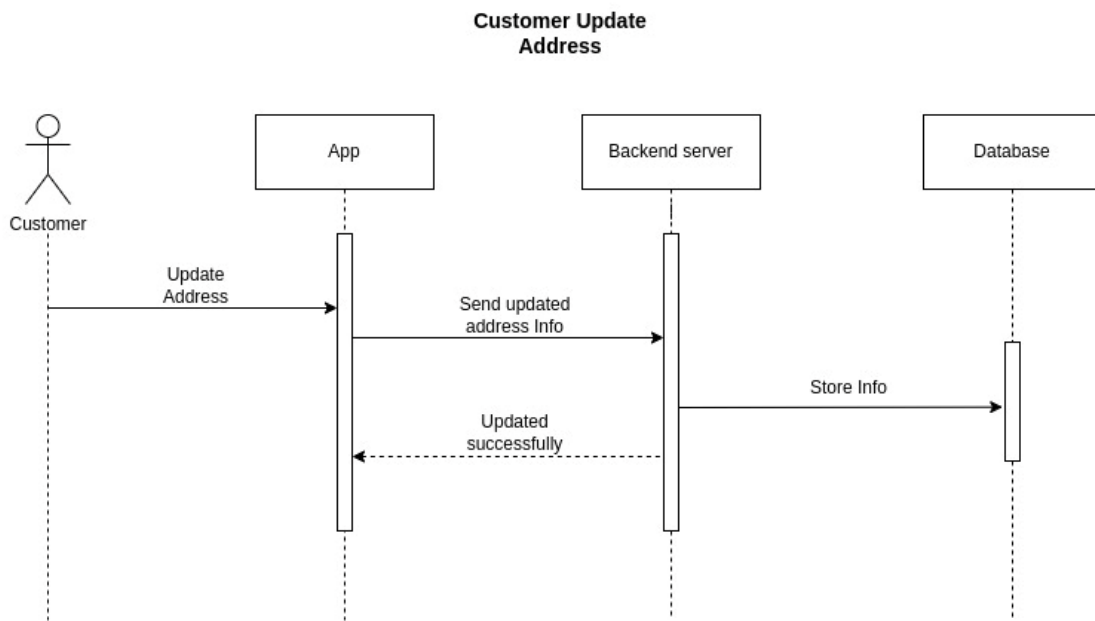


Figure 23: Customer Update Address Sequence Diagram

### 3.6.18 Cart List

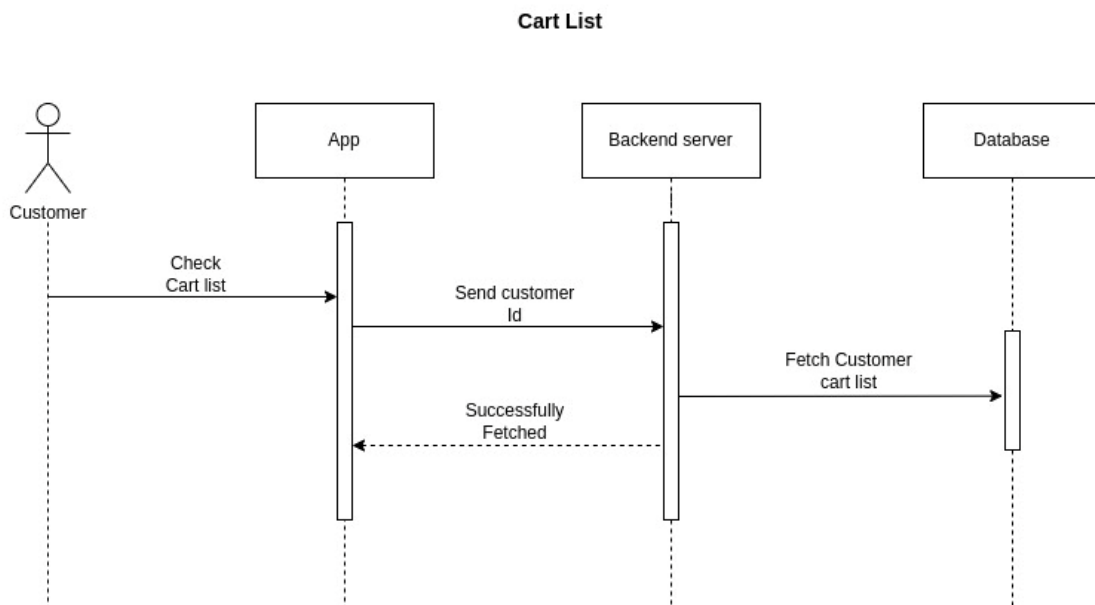


Figure 24: Cart List Sequence Diagram

### 3.6.19 Track Order

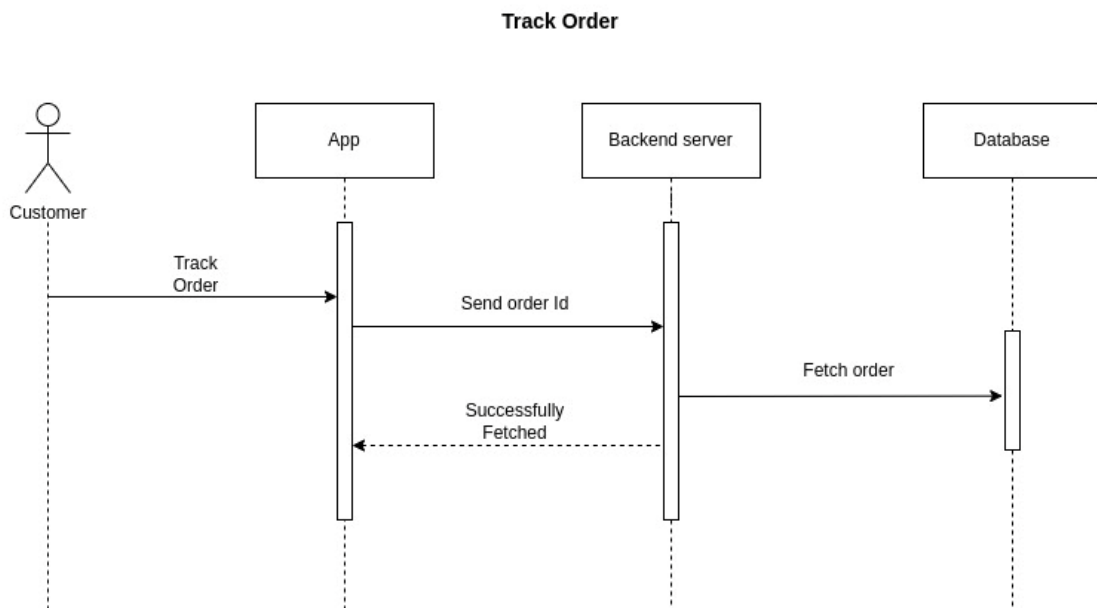


Figure 25: Track Order Sequence Diagram

### 3.6.20 Order History

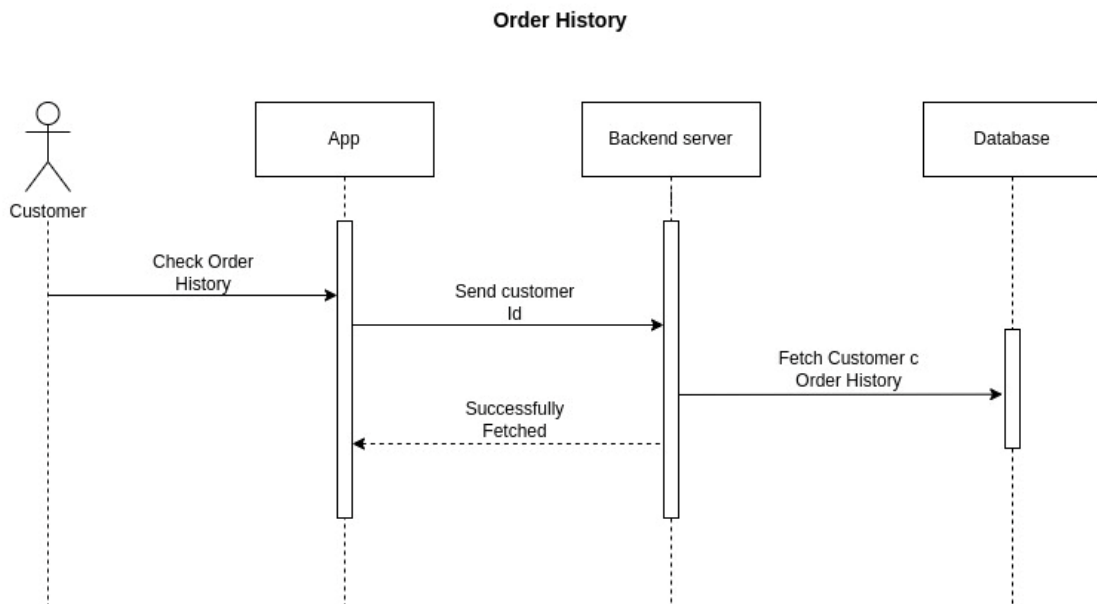


Figure 26: Order History Sequence Diagram

### 3.6.21 Payment

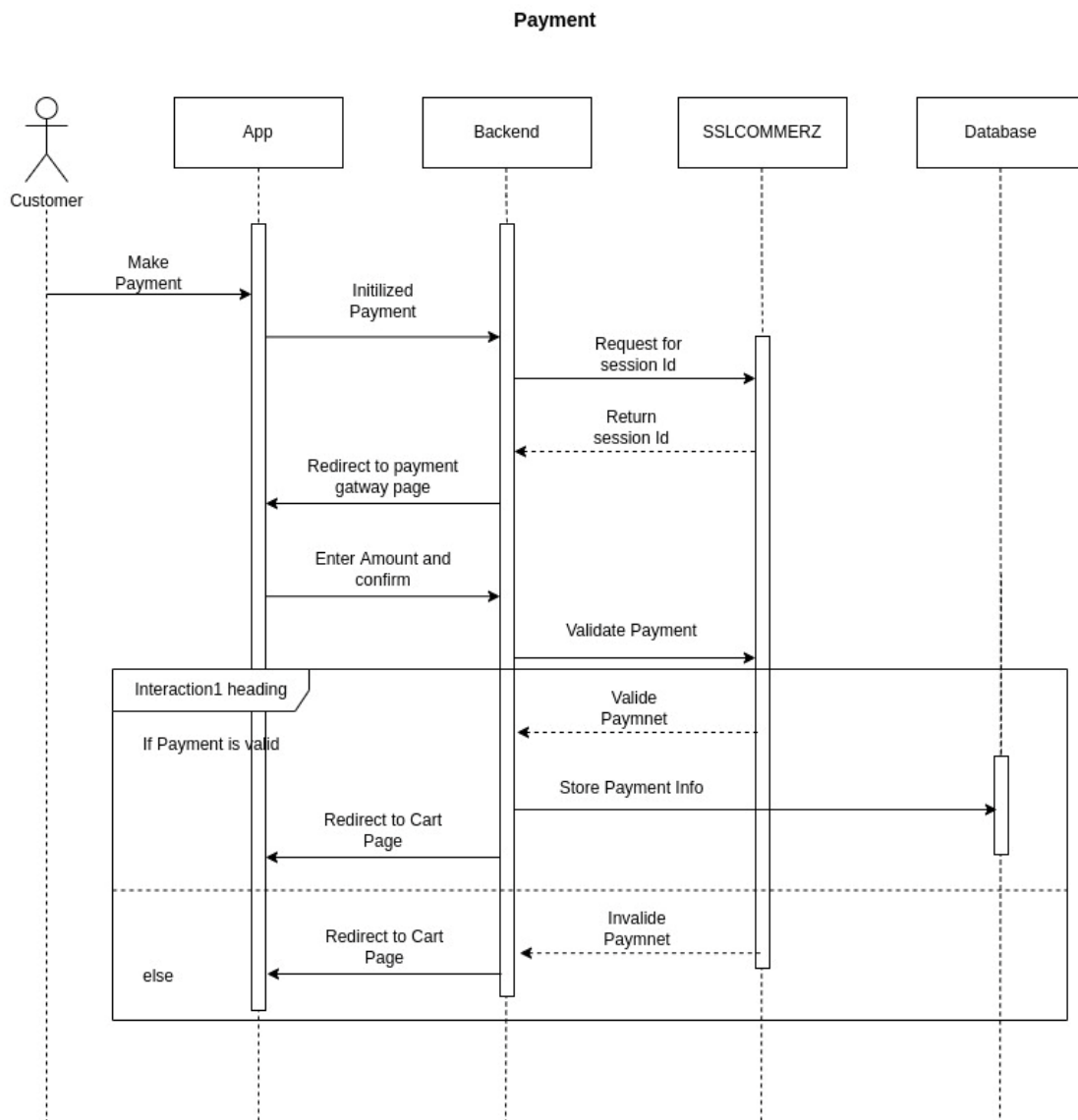


Figure 27: Payment Sequence Diagram

### **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 explain a software functionality and user interaction from both (developer and user) end. [23] Design specification elaborates 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

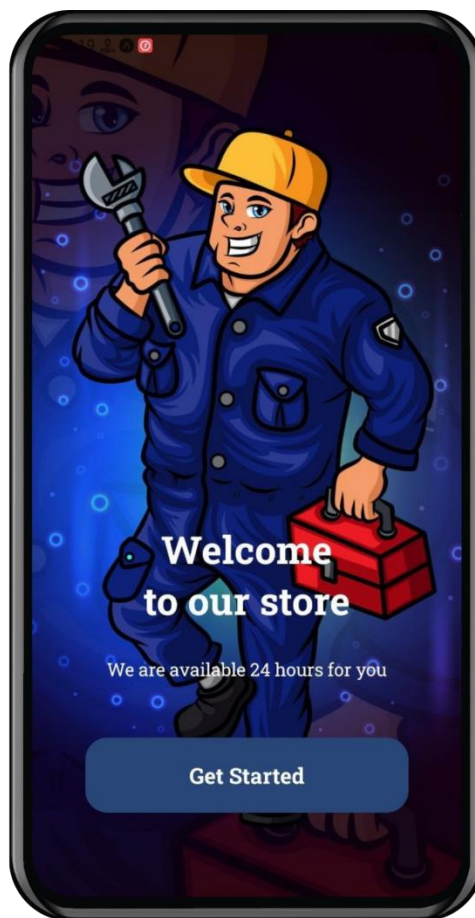


Figure 28: Welcome Page

## 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.

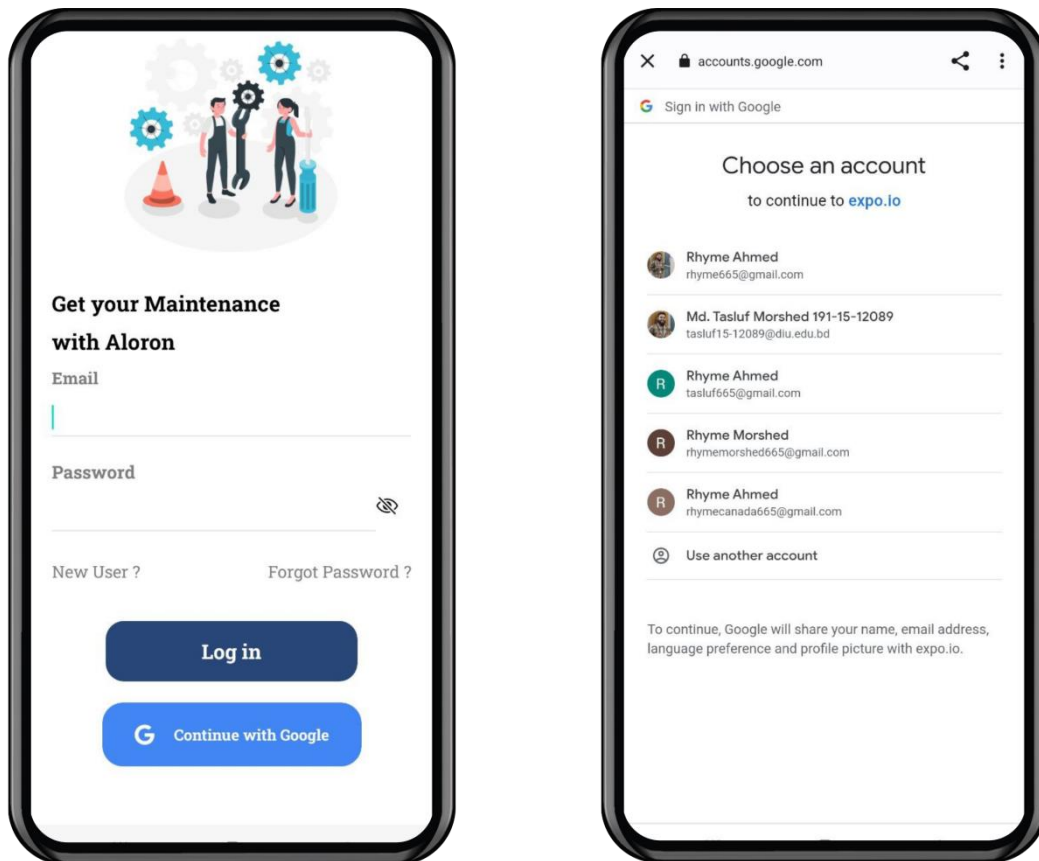
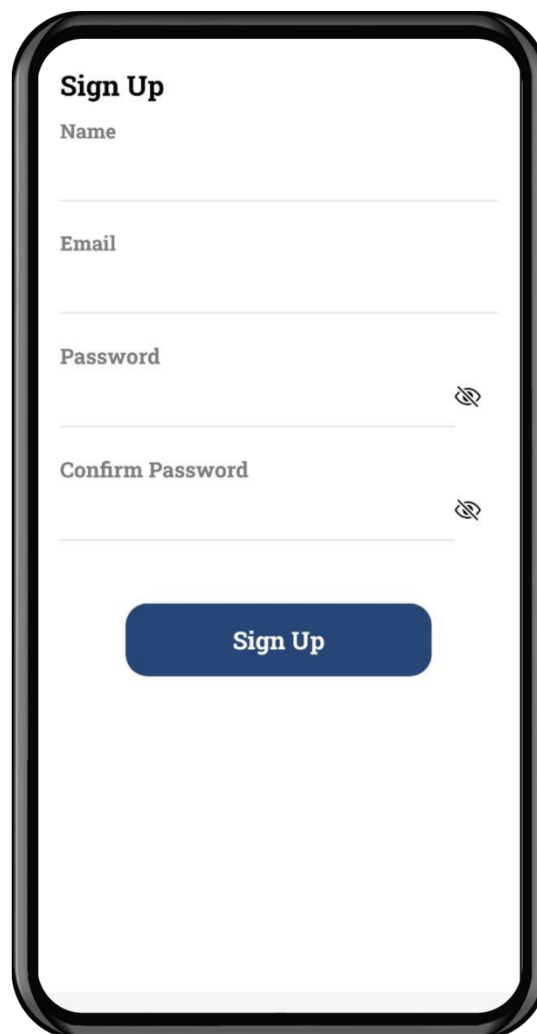


Figure 29: Login Page

## 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.



The image shows a mobile application sign-up page. At the top, the title "Sign Up" is displayed in bold. Below the title are four input fields: "Name", "Email", "Password", and "Confirm Password". Each field has a corresponding label and a horizontal line for text entry. To the right of the "Password" and "Confirm Password" fields, there are small icons representing an eye with a slash, indicating a toggle for password visibility. At the bottom of the form is a prominent blue button with the text "Sign Up" in white.

Figure 30: Sign Up Page

## Home Page

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

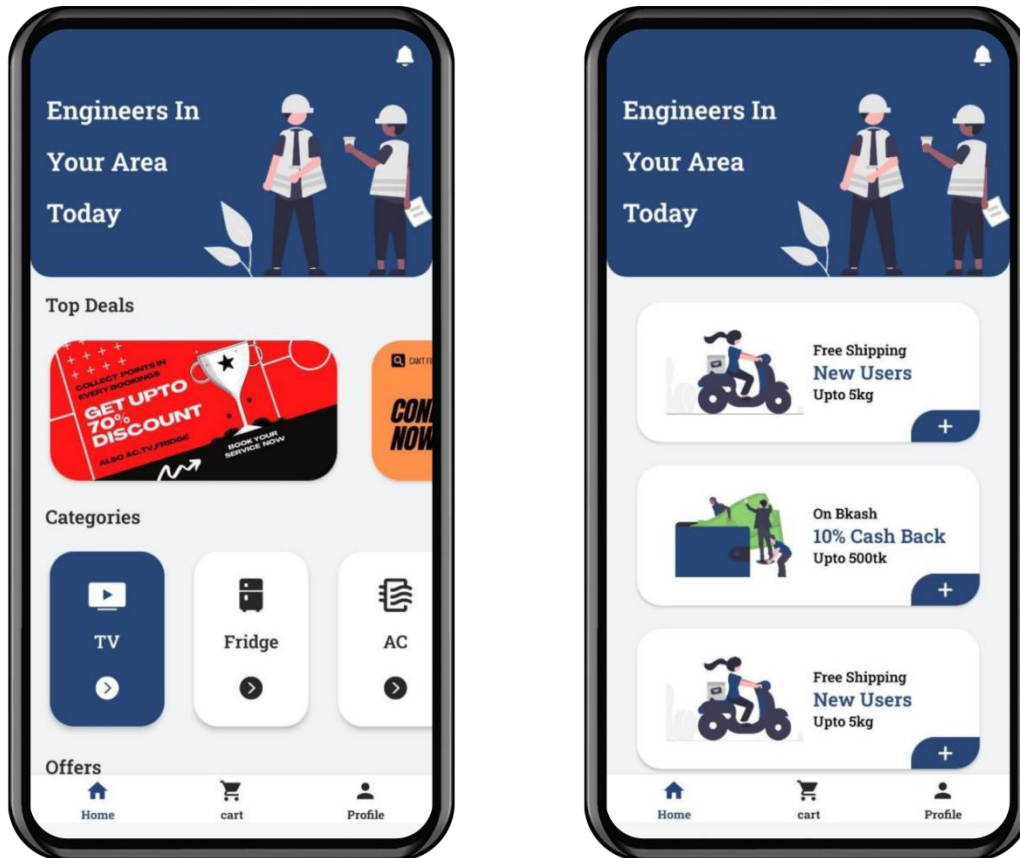


Figure 31: Home Page



## 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.

The image displays two mobile app screens for the 'Order' page. Both screens feature a navigation bar at the top with a back arrow and the title 'Order'. The left screen shows a video player at the top, followed by the heading 'Service Booking' and a placeholder text 'Lorem ipsum dolor sit amet, consectetur sds adip iscing elit. Imperdiet habitant'. Below this are form fields for 'Name' (containing 'Rhyme'), 'Date', 'Time', 'Address' (containing 'oad 2/3, Ambikapur Board Office, Faridpur, Dhaka'), 'Brand' (with a 'Select' dropdown), 'Model' (with a 'Select' dropdown), and 'Problem Detatils'. The right screen shows the same form fields, but with the 'Date' field containing a calendar icon and the 'Time' field containing a clock icon. The 'Address' field contains the same text. The 'Brand' and 'Model' dropdowns are also present. The 'Problem Detatils' field is empty. Below the form fields are two more fields: 'Spacial Note' and 'Contact Number' (containing '01951303939'). A blue 'BOOK NOW' button is positioned at the bottom of the form. Both screens have a bottom navigation bar with icons for 'Home', 'cart', and 'Profile'.

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.

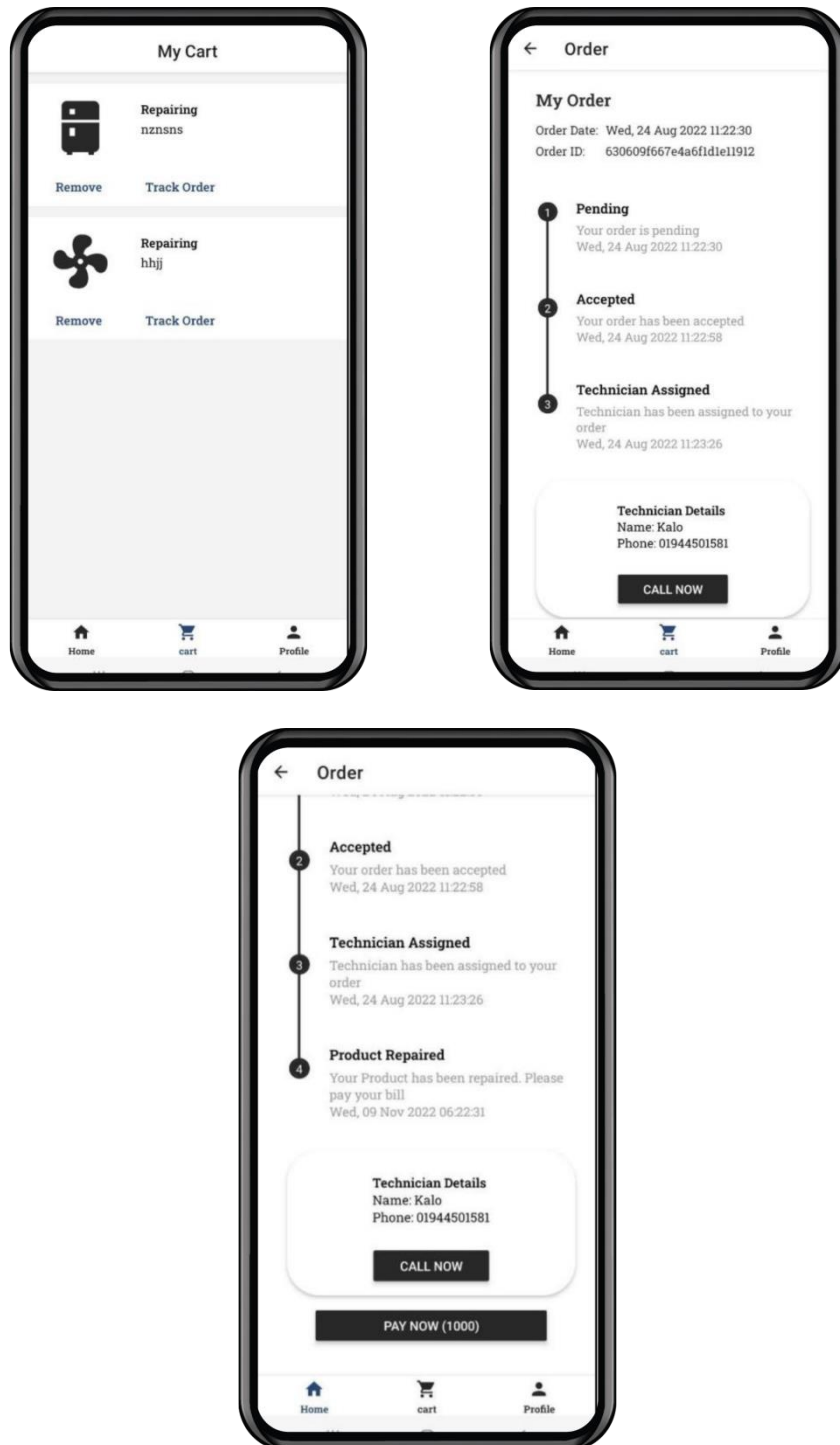


Figure 33: My Cart Page

## Payment Page

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

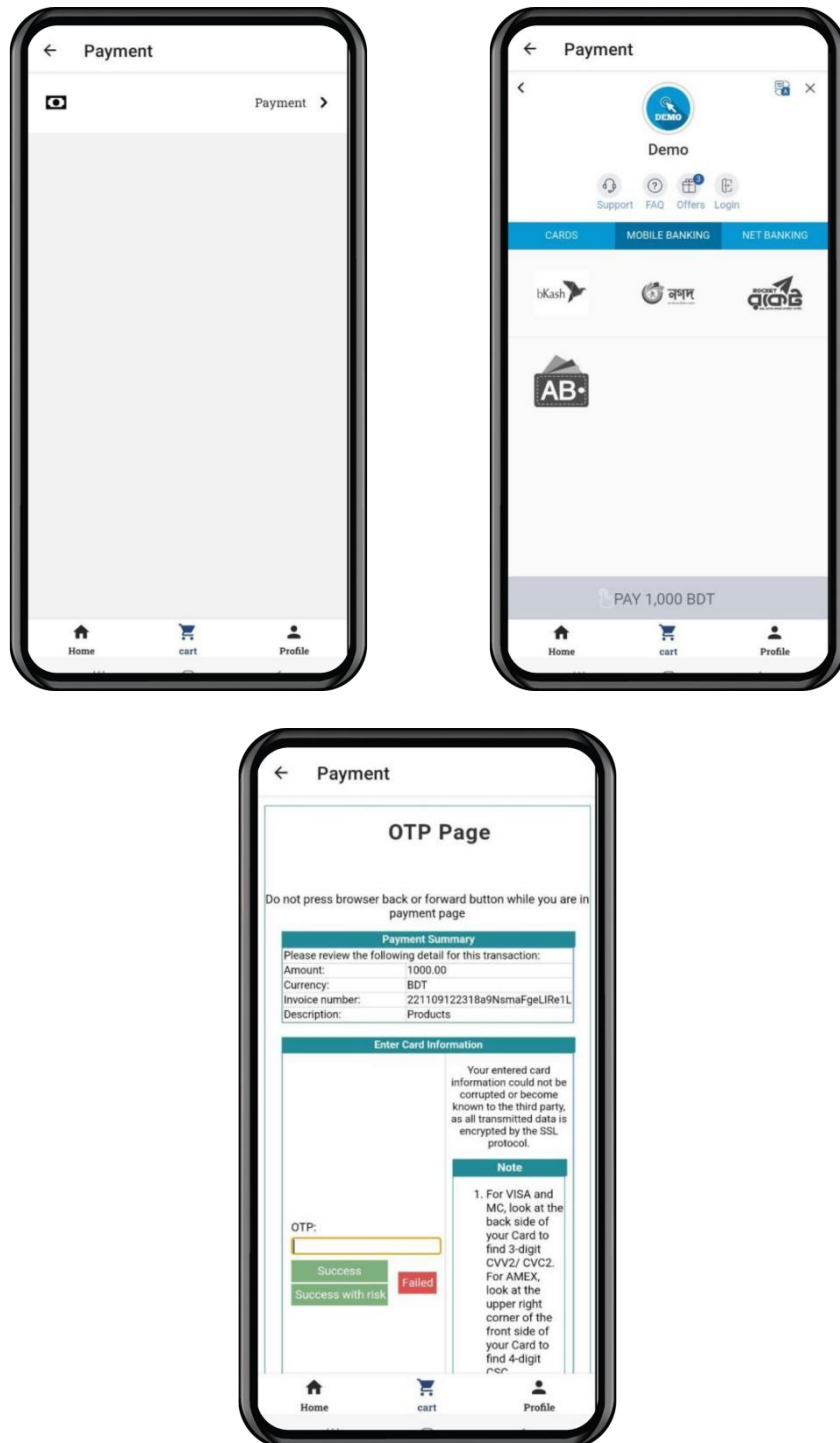


Figure 34: Payment Page

## 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.

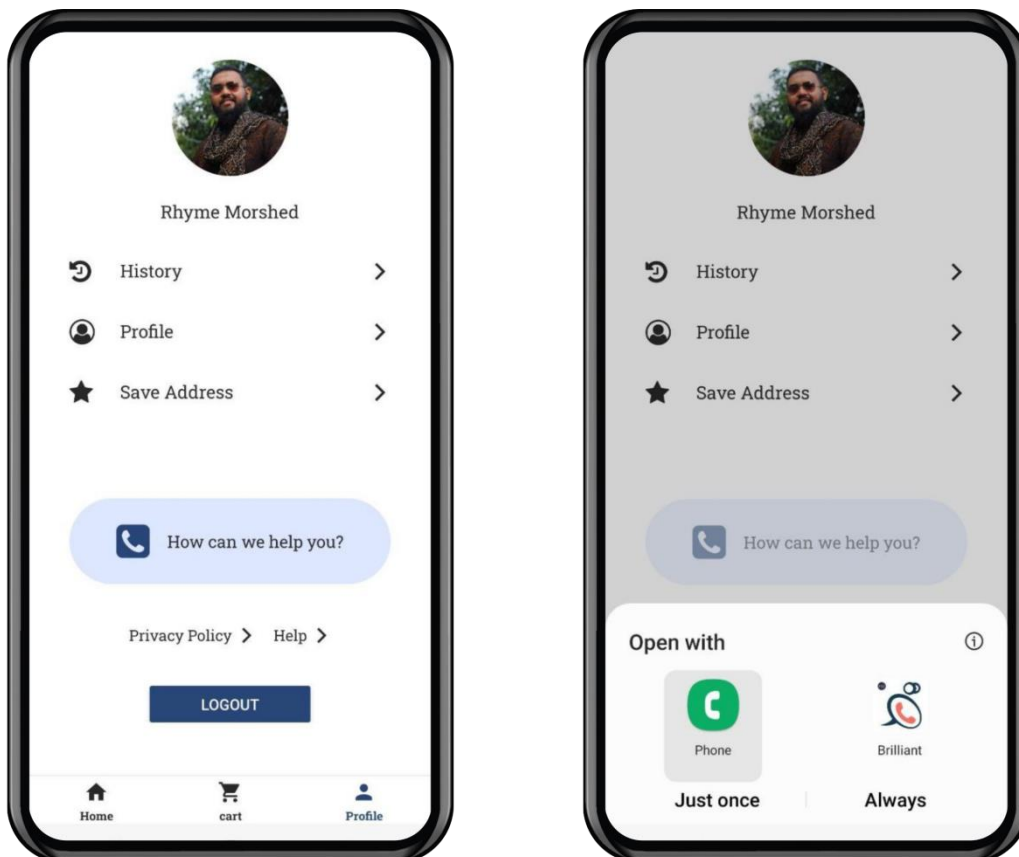


Figure 35: Profile Page

## 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.

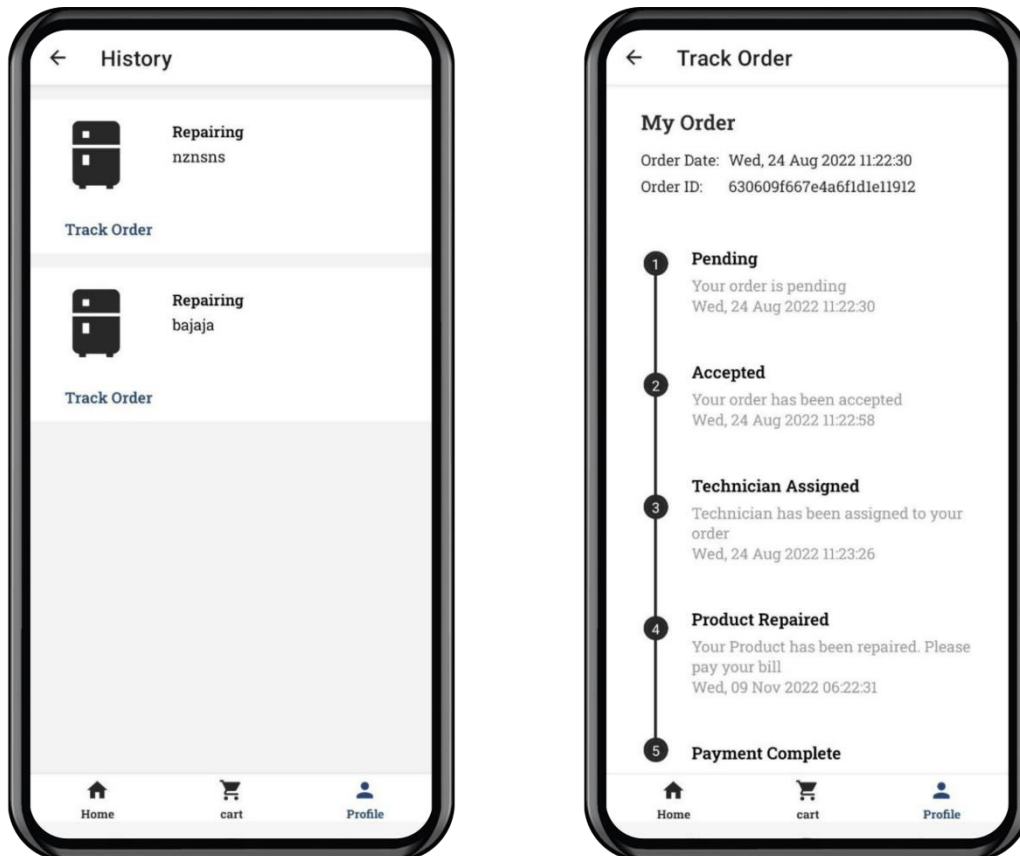


Figure 36: History Page

## Profile Edit Page

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

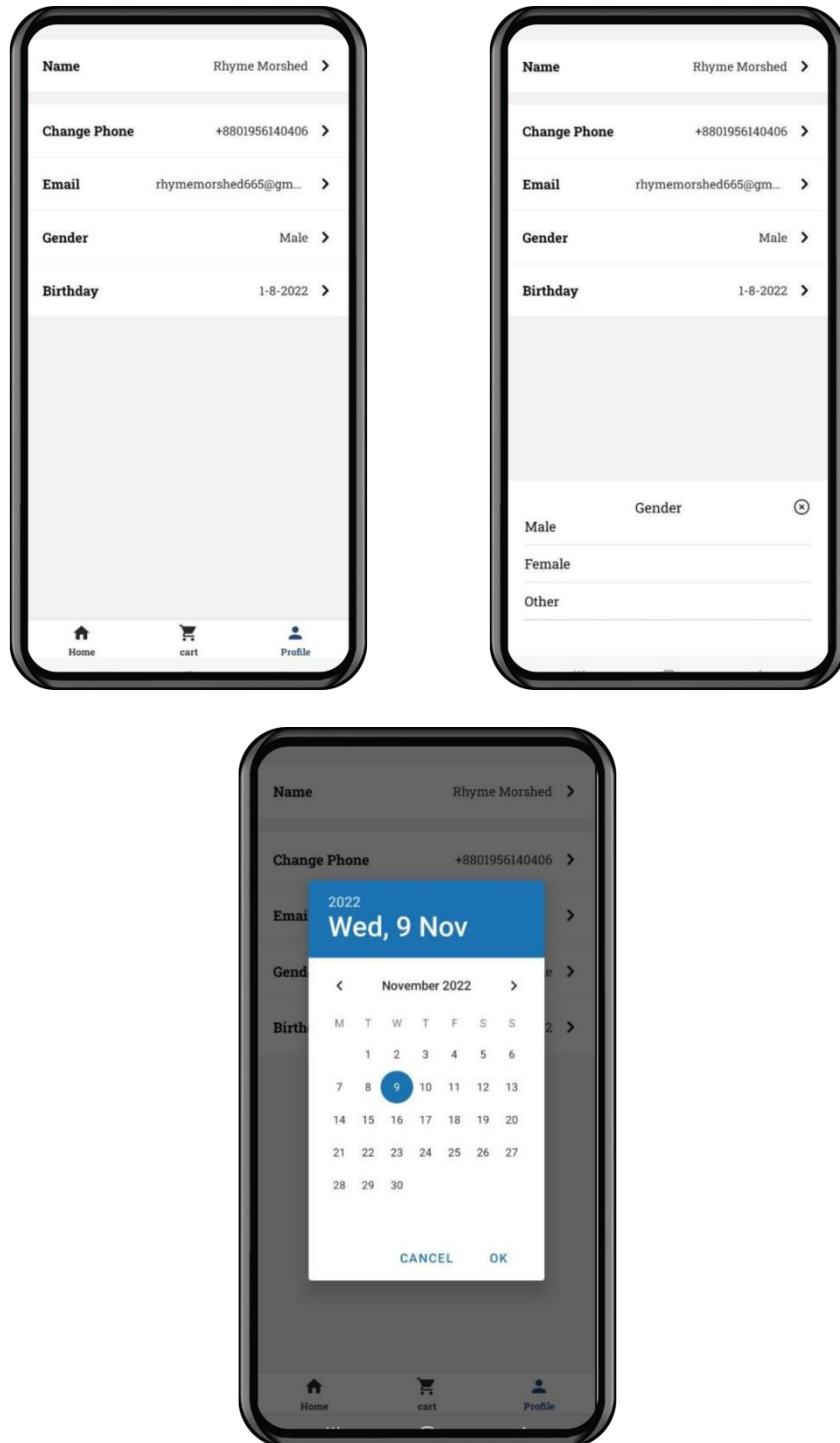


Figure 37: Profile Edit Page

## Address Page

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

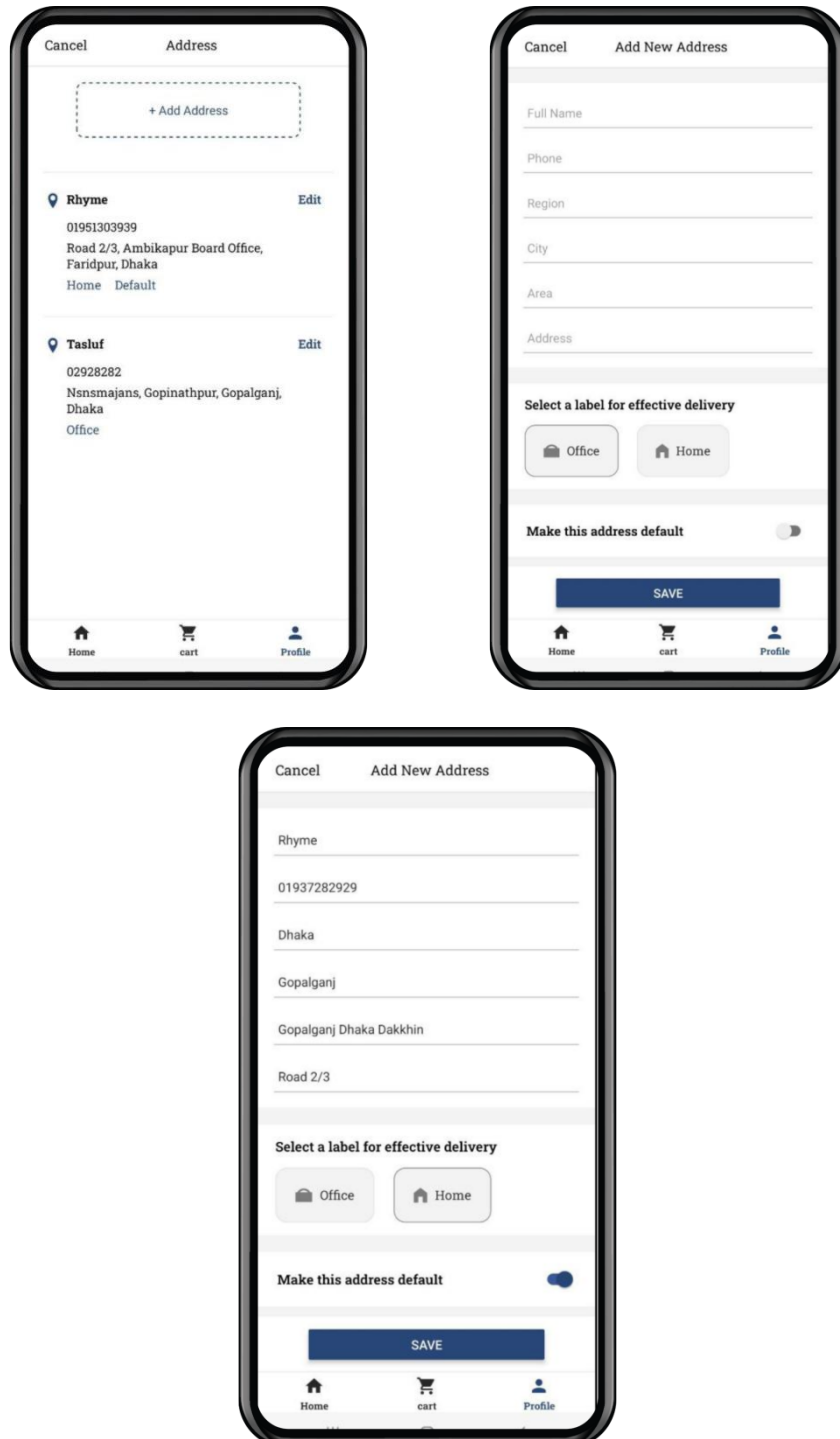


Figure 38: Address Page

## Select Address Page

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

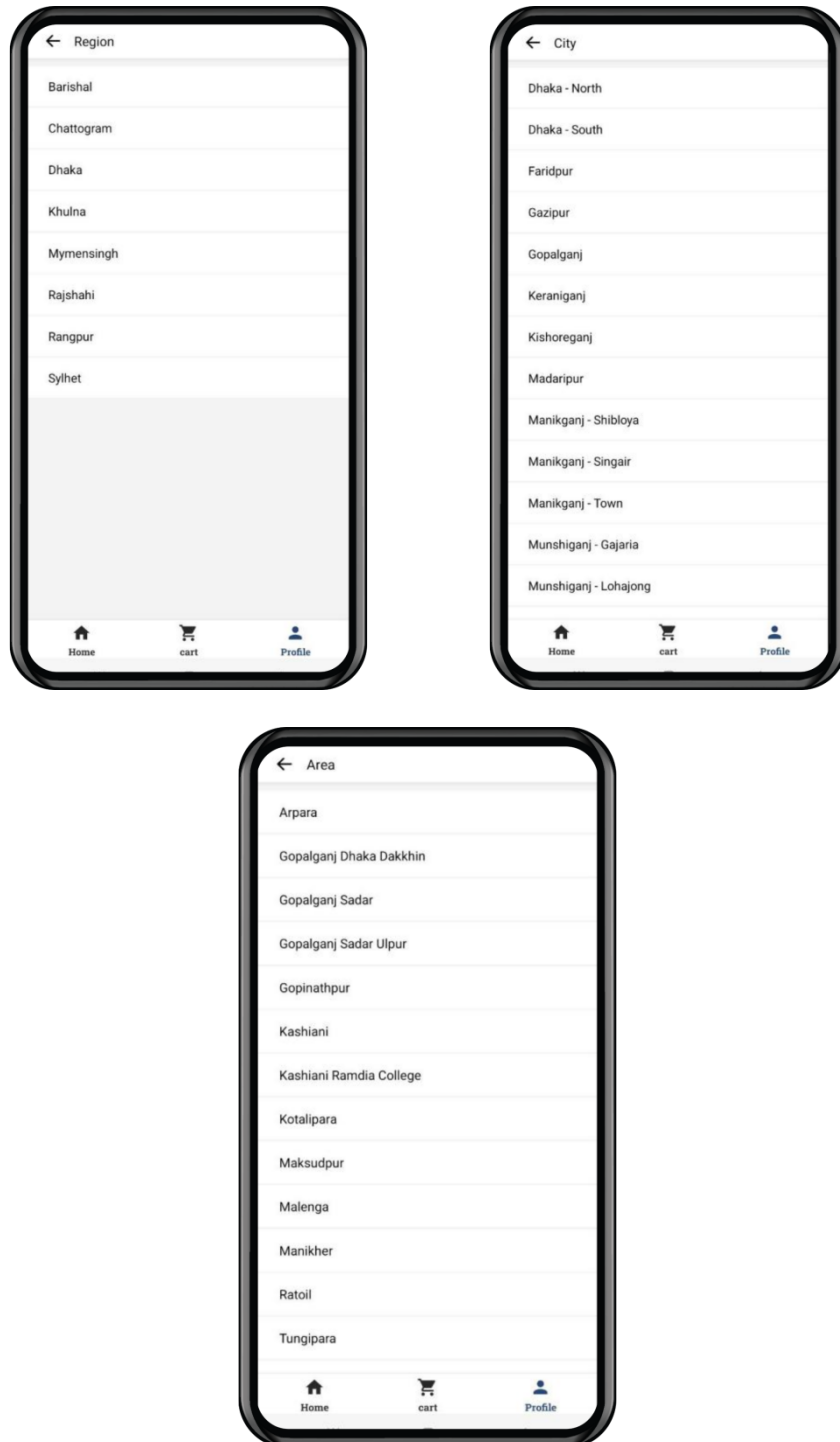


Figure 38: Select Address Page



## Notification Page

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

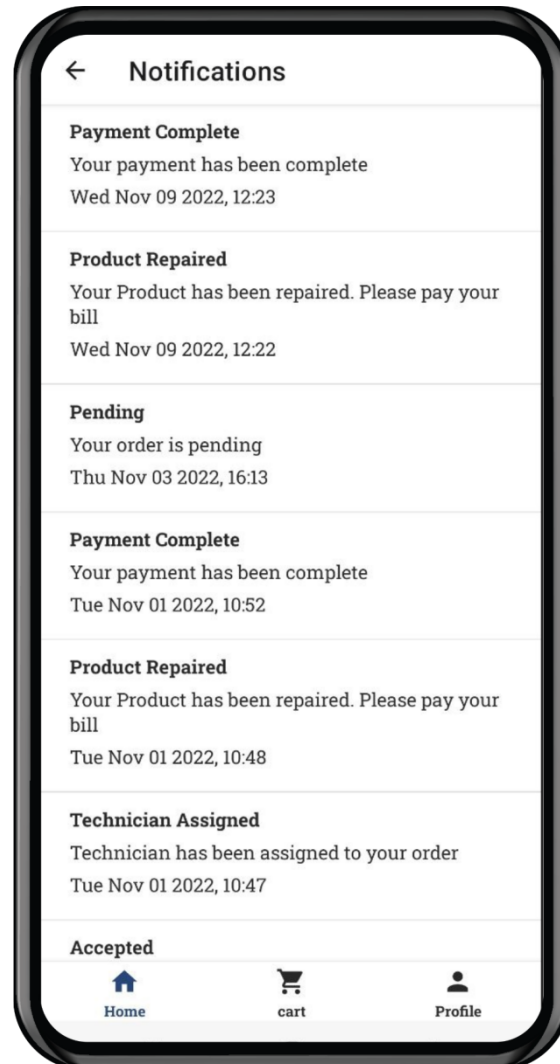
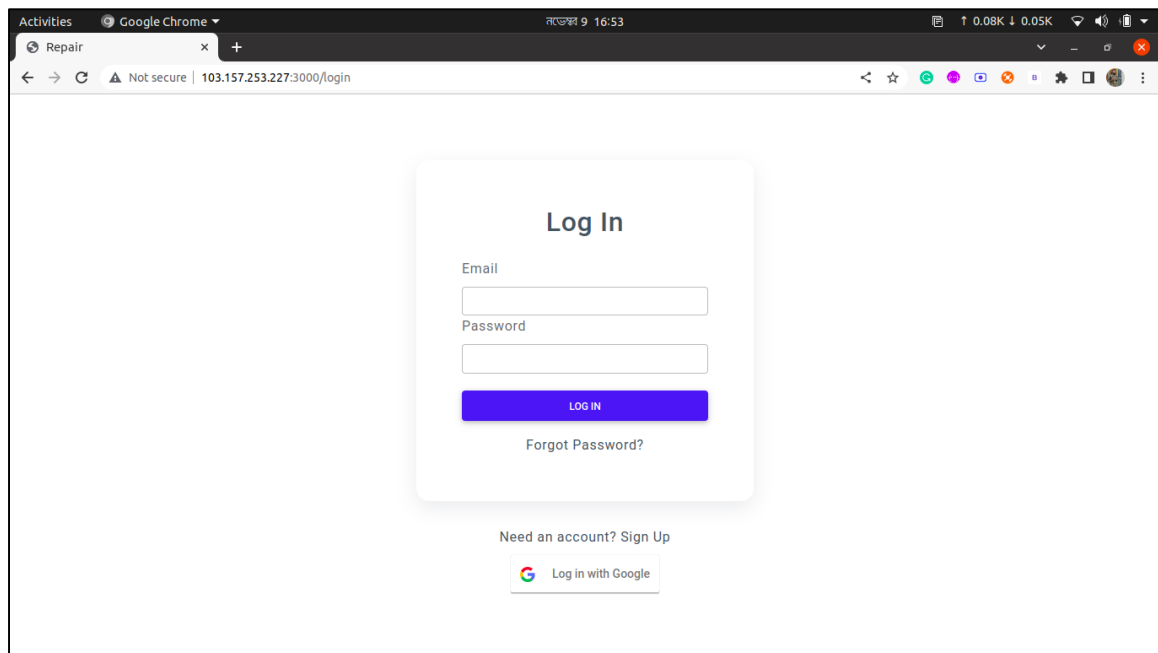


Figure 39: Notification Page

## 4.2.2 Admin Panel

### Login Page



The screenshot shows a web browser window with the address bar displaying "103.157.253.227:3000/login". The page features a central "Log In" form with the following elements:

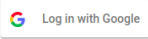
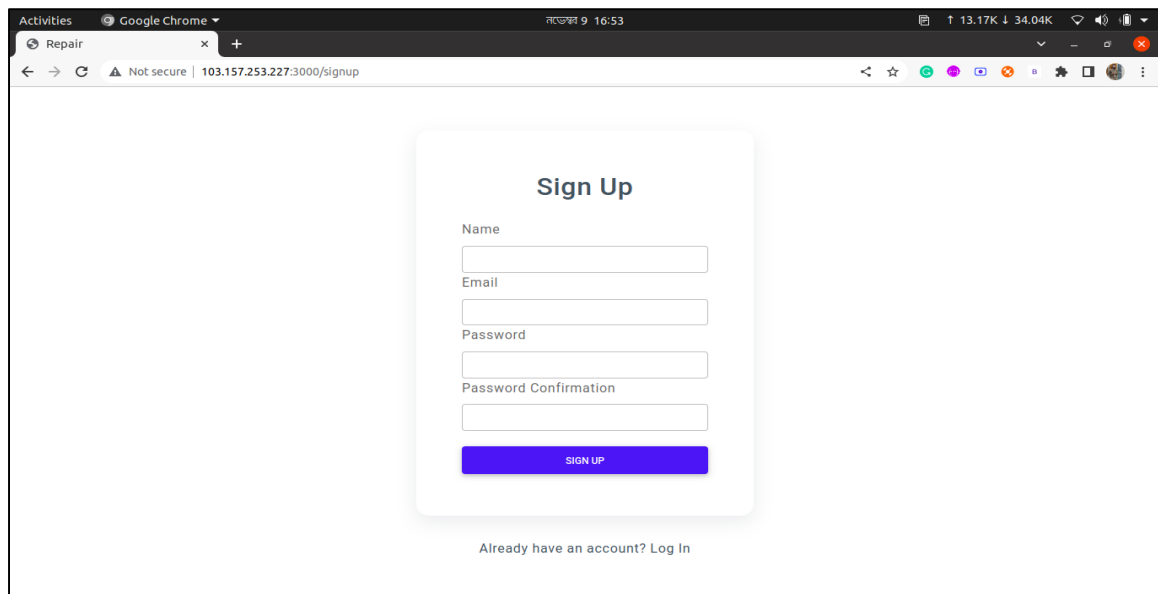
- Log In** (Title)
- Email** (Label) with an input field
- Password** (Label) with an input field
- LOG IN** (Blue button)
- [Forgot Password?](#) (Text link)
- [Need an account? Sign Up](#) (Text link)
-  (Button with Google logo)

Figure 40: Login Page

### Sign Up Page



The screenshot shows a web browser window with the address bar displaying "103.157.253.227:3000/signup". The page features a central "Sign Up" form with the following elements:

- Sign Up** (Title)
- Name** (Label) with an input field
- Email** (Label) with an input field
- Password** (Label) with an input field
- Password Confirmation** (Label) with an input field
- SIGN UP** (Blue button)
- [Already have an account? Log In](#) (Text link)

Figure 41: Sign Up Page

## Forgot Password Page

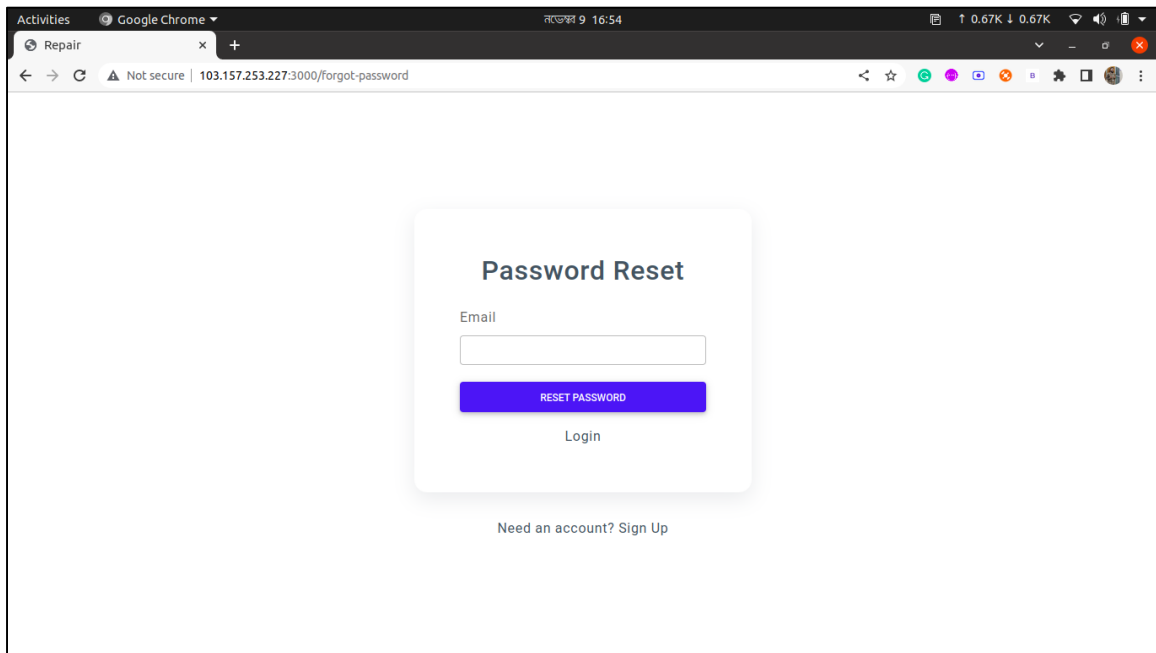


Figure 42: Forgot Password Page

## Dashboard Page

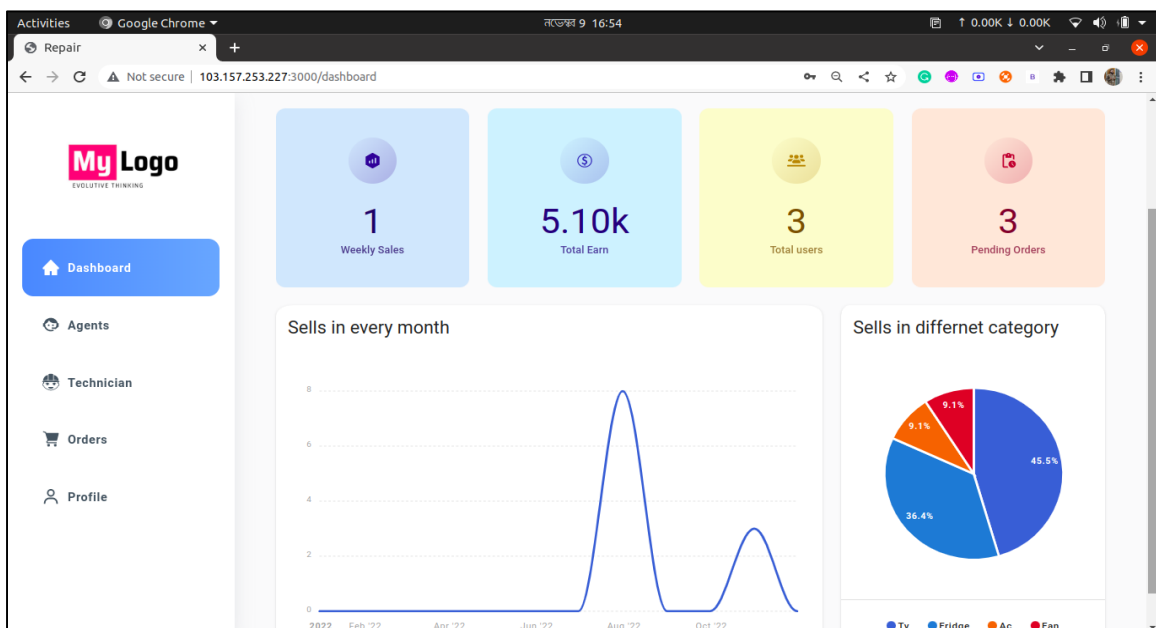


Figure 43: Dashboard Page

## Agent Page

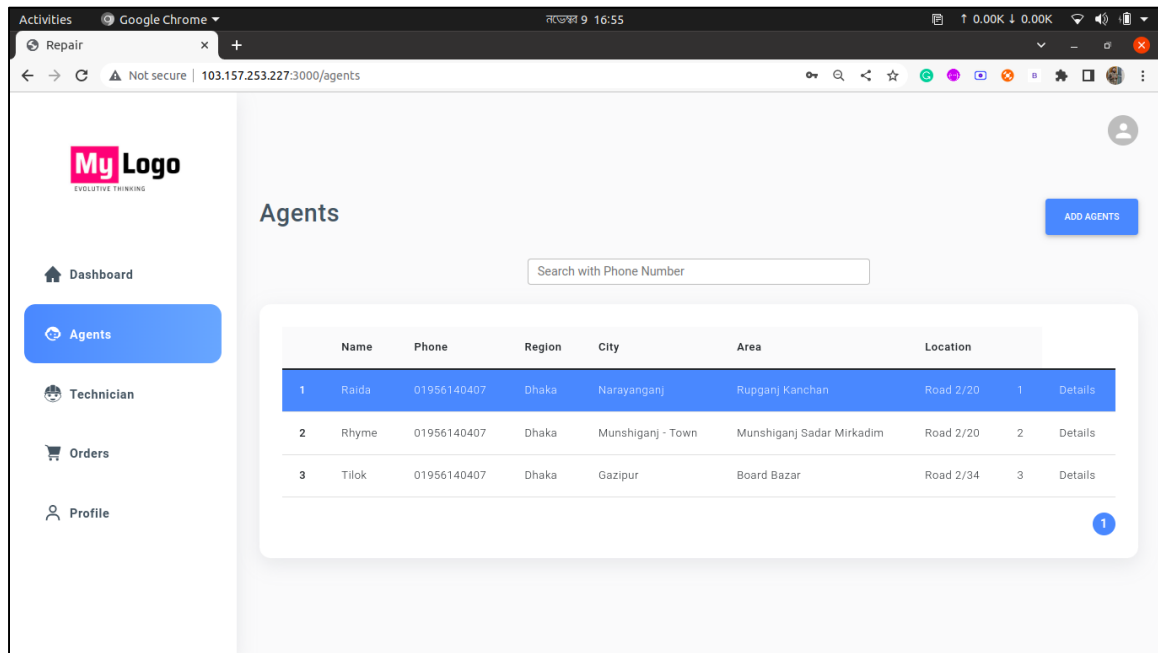


Figure 44: Agent Page

## Add Agent Page

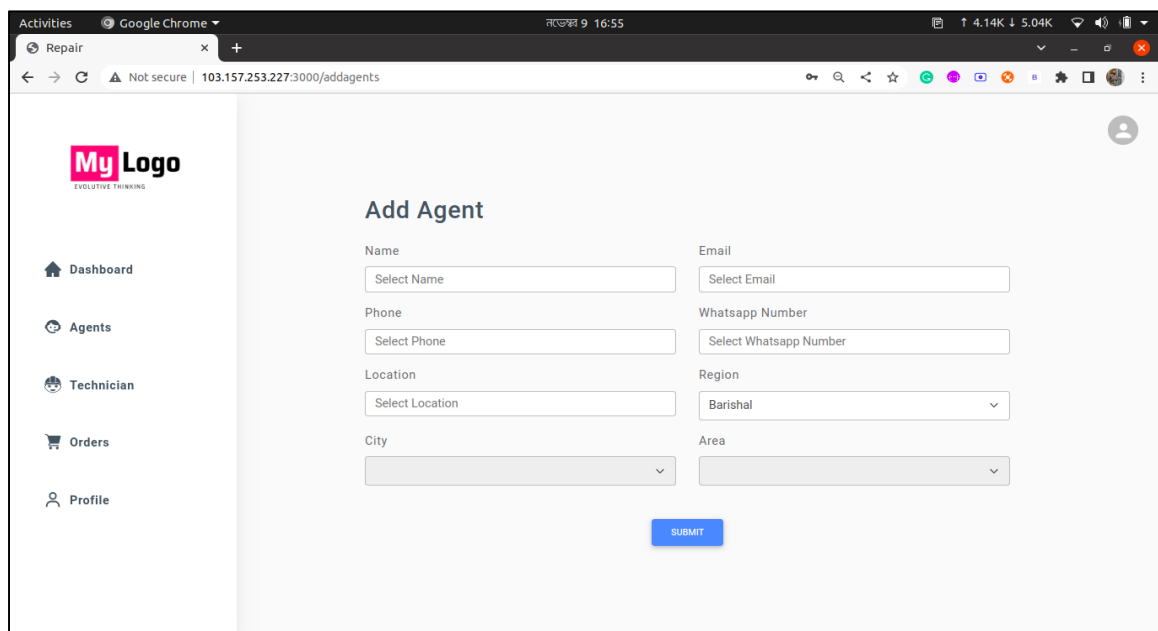


Figure 45: Add Agent Page

## Update Agent Page

**My Logo**  
EVOLUTIVE THINKING

### Update Agent

[EDIT](#)

Name: Raida  
 Email: raida665@gmail.com  
 Phone: 01956140407  
 Whatsapp Number: 01956140407  
 Location: Road 2/20  
 Region: Dhaka  
 City: Narayanganj  
 Area: Rugganj 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	Muktagachha Sadar	3	Tilok Details
4	Raju	5866100735	Chattogram	Chhagalnaia	Chhagalnaia Daraga Hat	4	Rhyme Details
5	Ridoy	01956140407	Dhaka	Narayanganj	Rugganj Murapara	5	Rhyme Details
6	Rifat	5866100735	Dhaka	Narayanganj	Rugganj 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 Nittyandapur	10	Rhyme Details

Figure 47: Technician Page

## Add Technician Page

**My Logo**  
EVOLUTIVE THINKING

### Add Technician

Name:

Email:

Phone:

Whatsapp Number:

Location:

Region:

City:

Area:

Agent:

**SUBMIT**

Figure 48: Add Technician Page

## Order Page

**My Logo**  
EVOLUTIVE THINKING

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

1 2

Figure 49: Order Page

## Order Details Page

My Logo  
EVOLUTIVE THINKING

Dashboard  
Agents  
Technician  
Orders  
Profile

### Update Order

EDIT

Name: nsnsns Phone: 029382822

Address: jajaja

Booking Time: 8/11/2022 Arrival Time: 8/12/2022, 4:25:40 PM

Category: Repairing Category Type: youtube-tv

Brand: Sony Plus Model: SONY PLUS 24" HD LED TV

Problem: bsnsns

Note: nssnsns Status: Payment Complete

Agent

Figure 50: Order Details Page

## Profile Page

My Logo  
EVOLUTIVE THINKING

Dashboard  
Agents  
Technician  
Orders  
Profile

Name: Rhyme Phone: 01956140407

Gender: Male Birthday: 12/16/1999

SUBMIT

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].

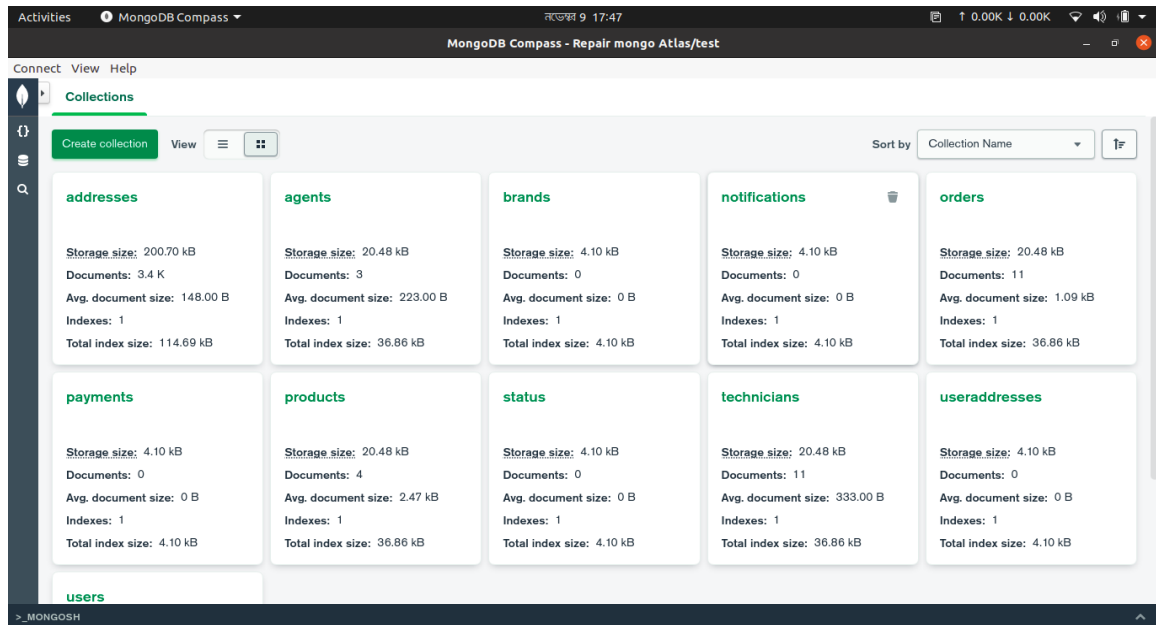


Figure 52: All Schema for database

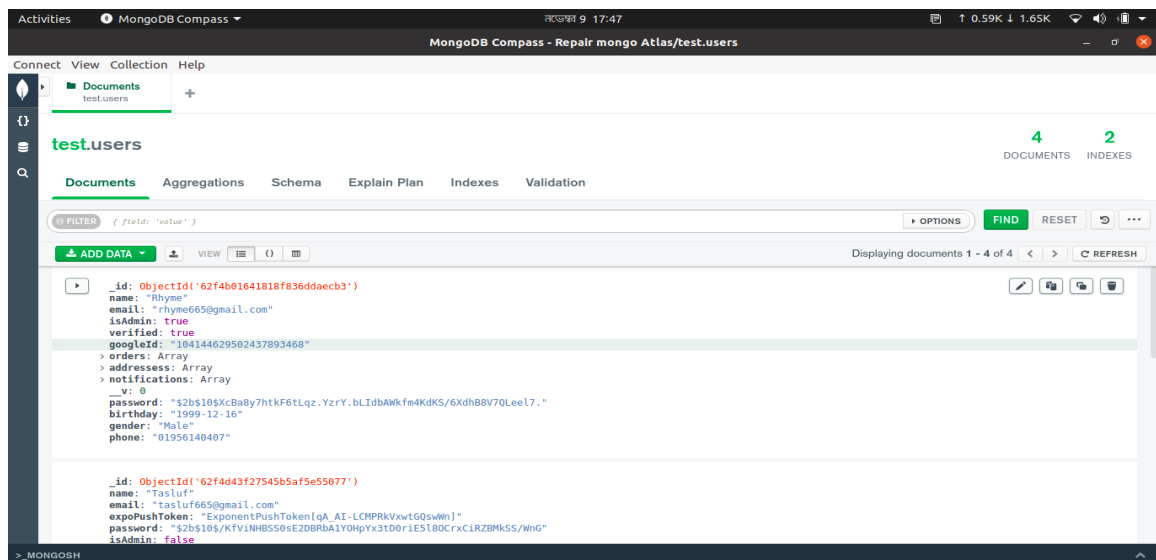


Figure 53: User Table



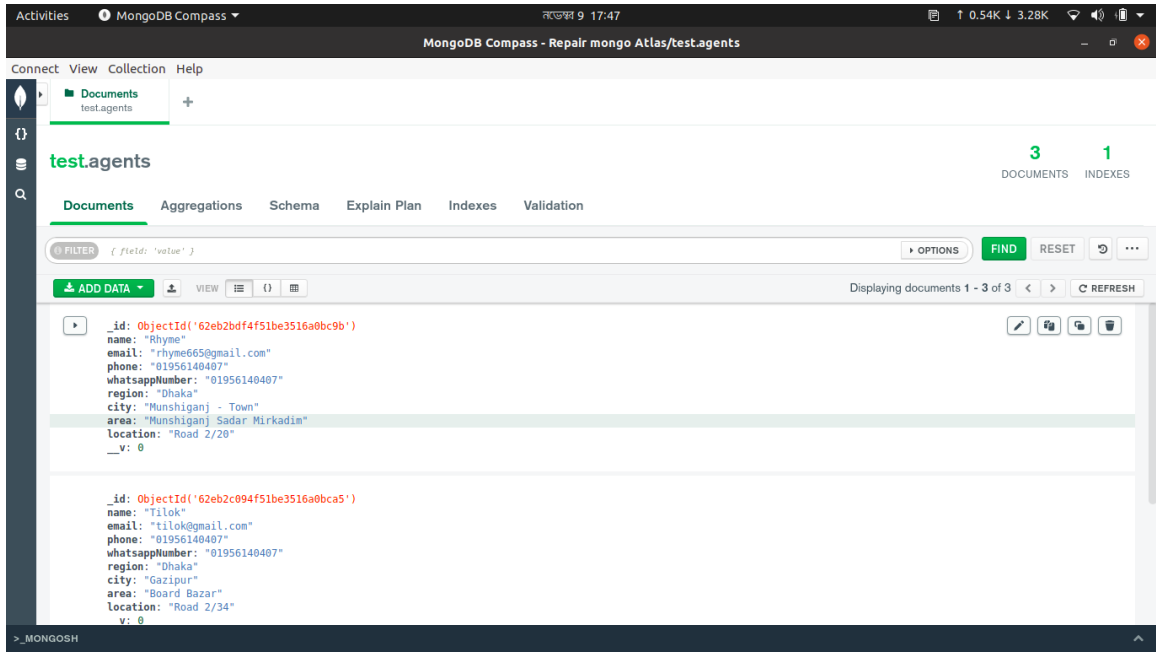


Figure 54: Agent Table

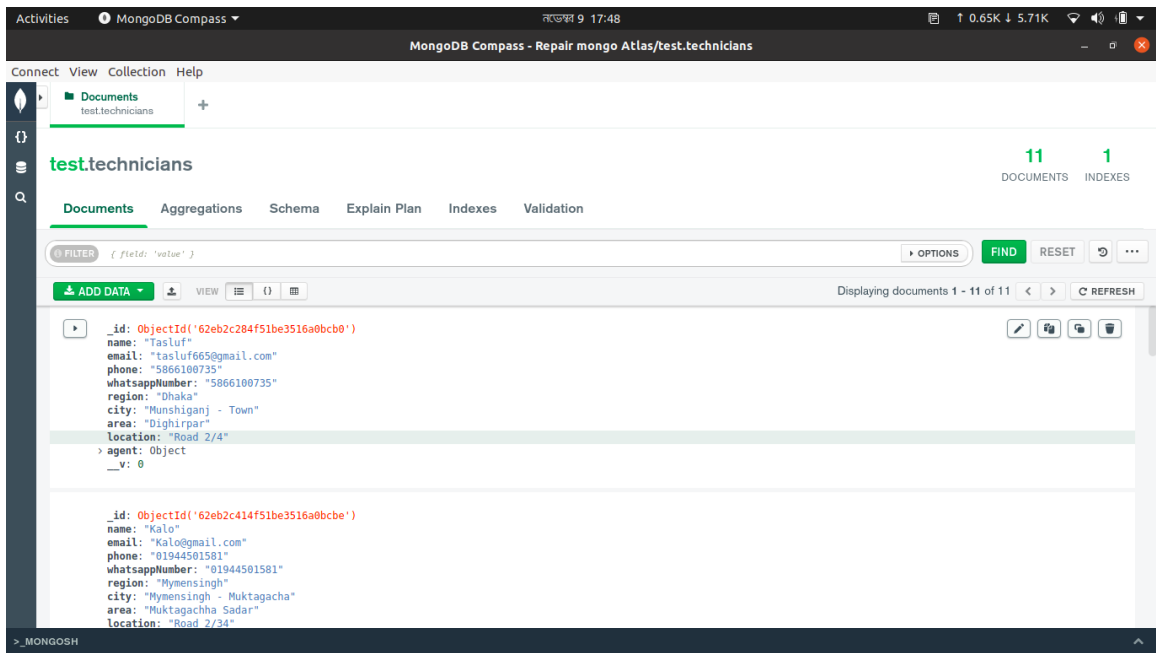


Figure 55: Technician Table

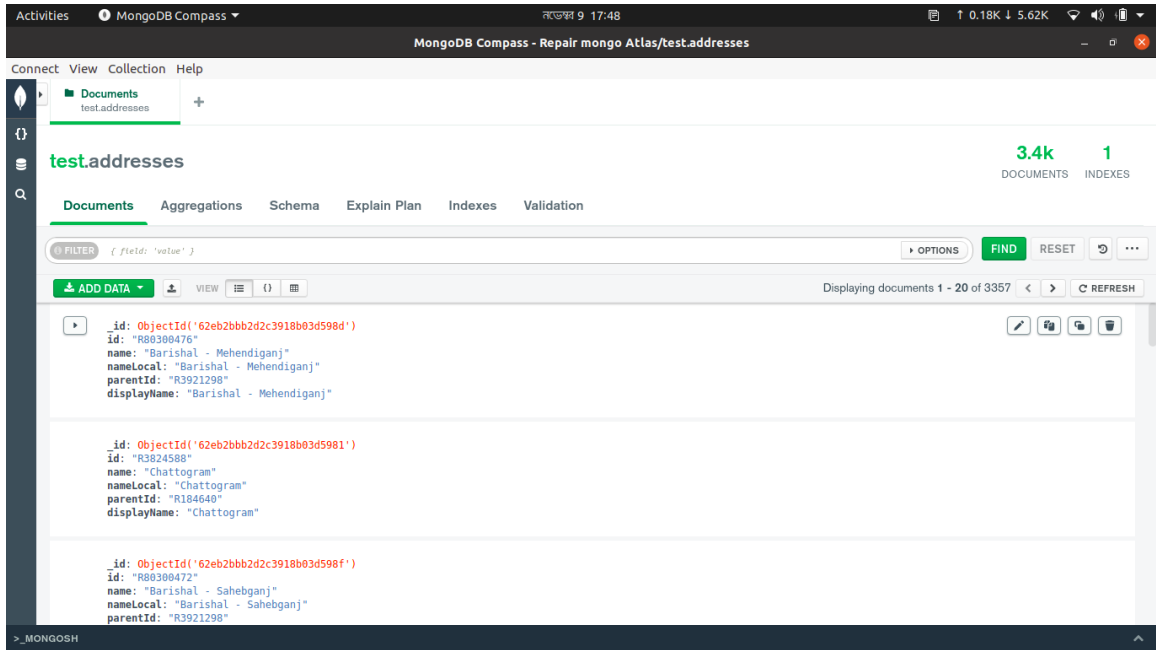


Figure 56: Address Table

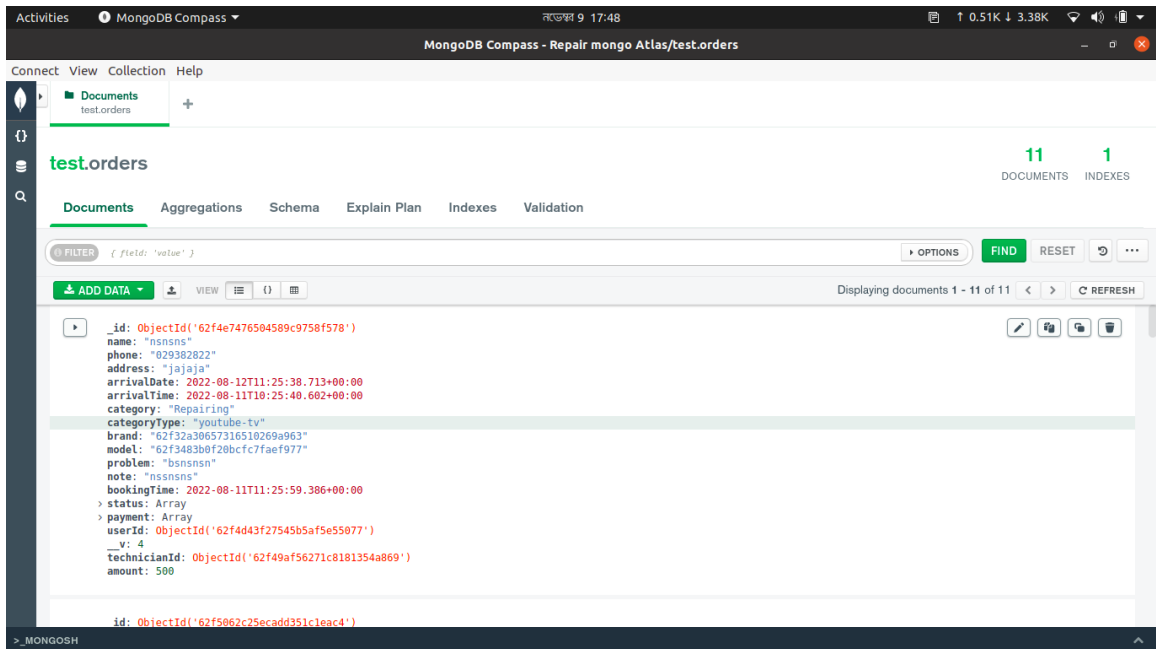


Figure 57: Orders Table

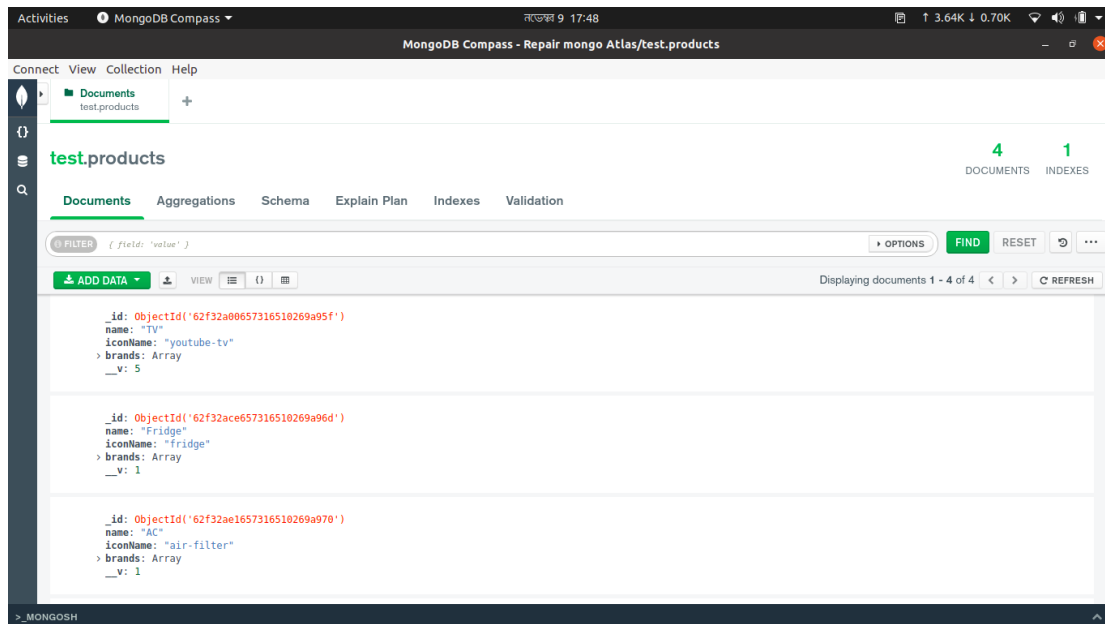


Figure 58: Products Table

#### 4.4 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 factors accurately reflect the strengths of our app and make it cost-effective. The user experience is continually improving as we prioritize enhancing user satisfaction and providing an enjoyable experience while using the app in commerce.

#### 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 examine 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:**

```

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

console.log
  Listening on port 3001...
    at Server.log (index.js:60:32)

console.log
  Connected with mongodb
    at log (index.js:40:23)

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

```

Figure 59: Automated Test figure 1

```

✓ 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?
rhyme@rhyme-HP: /media/rhyme/E Drive/Courses/Programming/Node/Backend_Repair$

```

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. [13] 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. [20]



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. [21] In are 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, therefor it has a significant impact on our society. As people spend more hour on our portable device, we need to ensure that the apps we build are secure and reliable for them. Therefor the developer of this app will create a unique innovative and user friendly app. We offer a wide range of services, including servicing, booking, solidity, guaranty, customer care, cost saving facilities and much more. [6]

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 indris-trial revolution day by day, in that very time some factor are threating 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. [7] Researcher expected it can doubled its size within next ten years. [22] 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. E-waste is often disposed of in landfills, where toxic materials can seep into groundwater and harm both terrestrial and marine wildlife, as well as people who consume contaminated food. [8]

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. [9]

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. [10] 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 21st 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%. [12] 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

## REFERENCES

- [1] “Electronic Goods Repairing & Servicing in Bangladesh,” *www.bdtradeinfo.com*.  
<https://www.bdtradeinfo.com/yp-data/electronic-goods-repairing-servicing>.
- [2] T. H. Phạm, “B2B Brand Management,” *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*.  
<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>
- [14] C. Riedl, J. M. Leimeister, and H. Krcmar, “New Service Development for Electronic Services – A Literature Review,” *www.researchgate.net*, Jun. 09, 2009.  
[https://www.researchgate.net/publication/220890600\\_New\\_Service\\_Development\\_for\\_Electronic\\_Services\\_-\\_A\\_Literature\\_Review](https://www.researchgate.net/publication/220890600_New_Service_Development_for_Electronic_Services_-_A_Literature_Review) (accessed Oct. 31, 2022).
- [15] H. Taherdoost, S. Sahibuddin, and N. Jalaliyoon, “Features’ Evaluation of Goods, Services and E-Services, Electronic Service Characteristics Exploration,” *www.researchgate.net*, Dec. 2014.  
[https://www.researchgate.net/publication/260009999\\_Features'\\_Evaluation\\_of\\_Goods\\_Services\\_and\\_E-Services\\_Electronic\\_Service\\_Characteristics\\_Exploration](https://www.researchgate.net/publication/260009999_Features'_Evaluation_of_Goods_Services_and_E-Services_Electronic_Service_Characteristics_Exploration) (accessed Jul. 13, 2022).
- [16] E. Ankrah, “Customer Satisfaction of Electronic Products and Services in Ghanaian Banks.” [Online]. Available: <https://core.ac.uk/download/pdf/234671279.pdf>

- [17] M. Fassnacht and I. Koese, "Quality of Electronic Services," *Journal of Service Research*, vol. 9, no. 1, pp. 19–37, Aug. 2006, doi: 10.1177/1094670506289531.
- [18] "International Journal of Electronic Trade (IJETrade) Inderscience Publishers - linking academia, business and industry through research," *www.inderscience.com*.  
<https://www.inderscience.com/jhome.php?jcode=ijetrade> (accessed Jan. 31, 2023).
- [19] D. A. S, "A Study on Customer Preference Towards Electronic Goods," *Bonfring International Journal of Industrial Engineering and Management Science*, vol. 2, no. 1, pp. 44–51, 2012, Accessed: Jan. 31, 2023. [Online]. Available:  
[https://www.academia.edu/7795904/A\\_Study\\_on\\_Customer\\_Preference\\_Towards\\_Electronic\\_Goods](https://www.academia.edu/7795904/A_Study_on_Customer_Preference_Towards_Electronic_Goods)
- [20] admin, "The Fastest-Growing Service-Based Industries through 2021," *Project & Service Management*, Jun. 10, 2022.  
<https://www.timelinxsoftware.com/the-fastest-growing-service-based-industries-through-2021/>
- [21] "How SWOT (Strength, Weakness, Opportunity, and Threat) Analysis Works," *Investopedia*.  
<https://www.investopedia.com/terms/s/swot.asp#:~:text=Threats%20refer%20to%20factors%20that>
- [22] "Global E-Waste Surging: Up 21% in 5 Years - United Nations University," *unu.edu*.  
<https://unu.edu/media-relations/releases/global-e-waste-surging-up-21-in-5-years.html>
- [23] "How To Write The Design Specification? [Quick Guide]," *Northell*.  
<https://northell.design/blog/how-to-write-the-design-specifications-quick-guide>
- [24] "What Is a Software Requirement Specification? (Definition) | Built In," *builtin.com*.  
<https://builtin.com/software-engineering-perspectives/software-requirement-specification> (accessed Jan. 31, 2023).

# 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

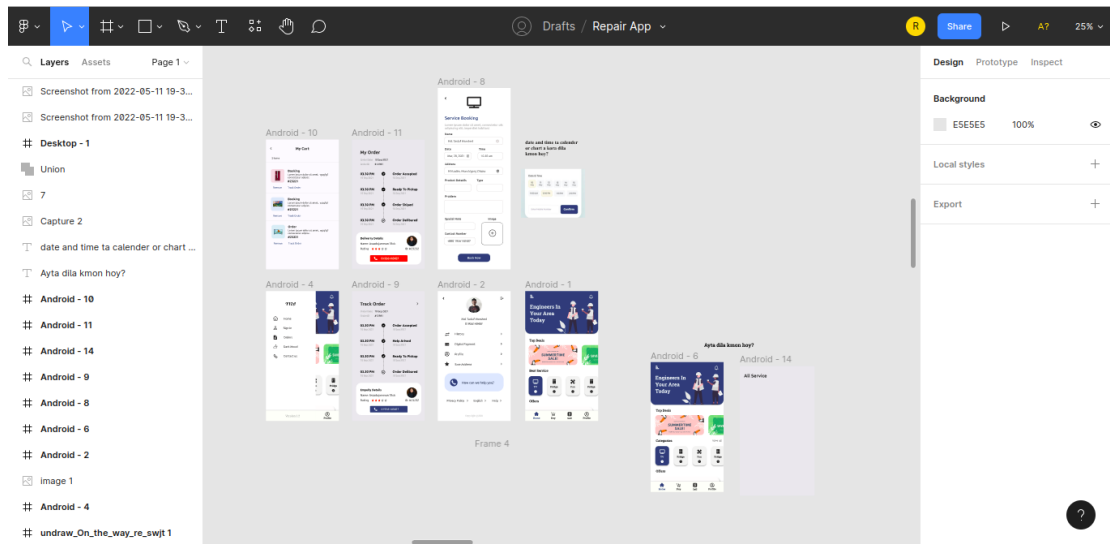
Chapter 7: Conclusion and further suggested work in this section. In conclusion we discuss based on three aspect what, why & how.



## Appendix B

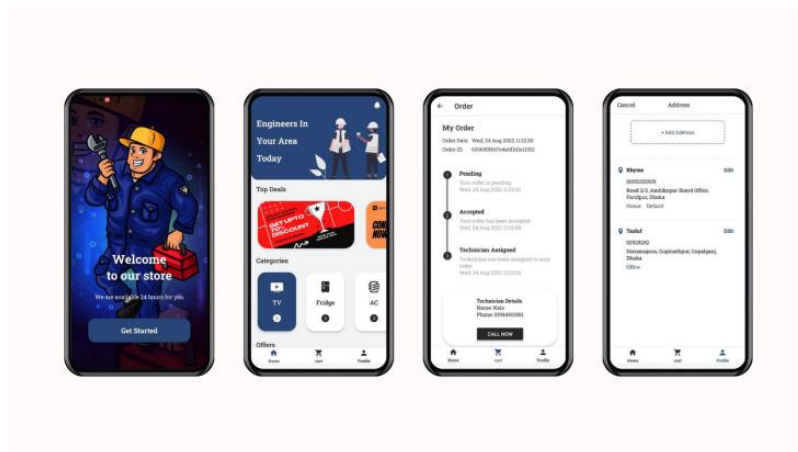
Design Resource: We design our app in Figma.

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



For logo design and flyer 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>

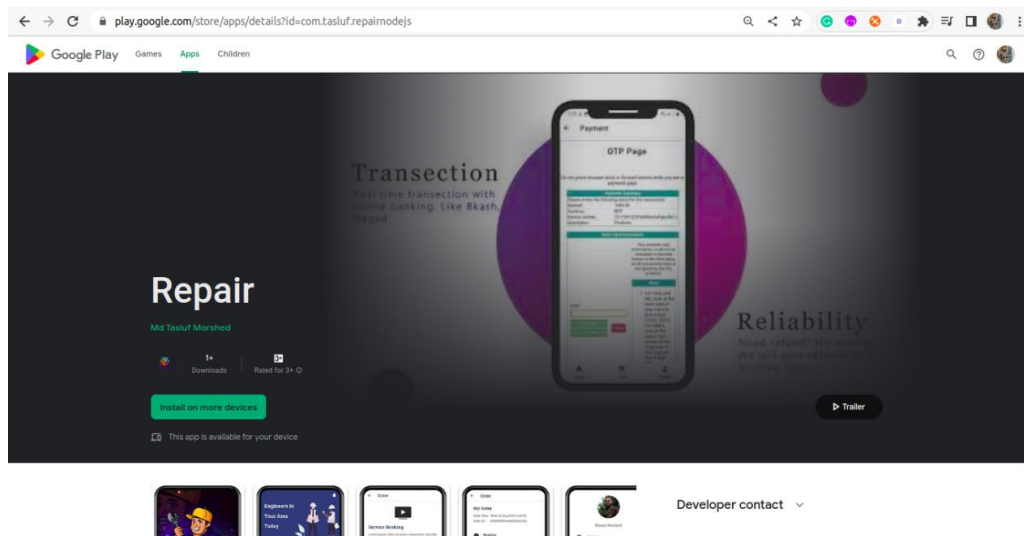
## Appendix C

Main App Resource:

First we deploy our app in expo server.

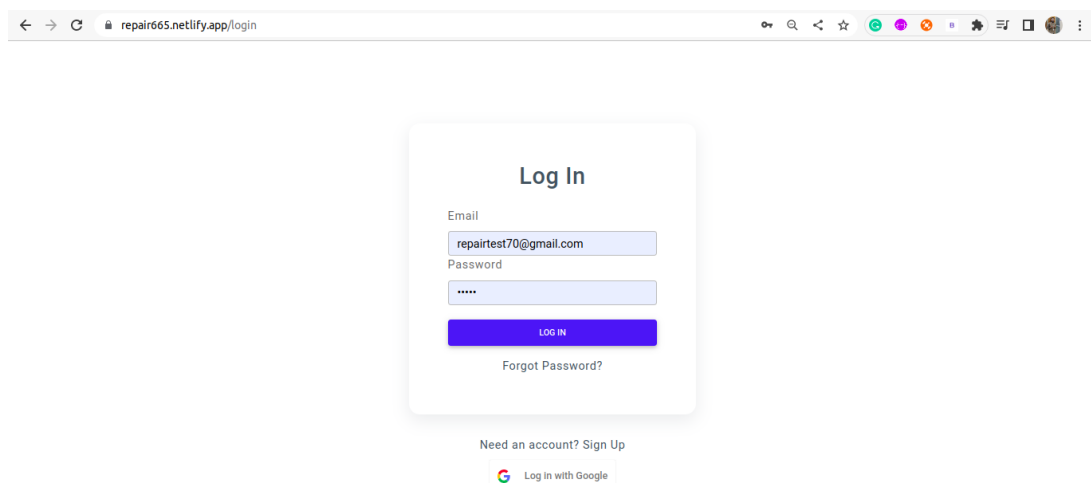
<https://expo.dev/@tasluf/repair-nodejs?serviceType=classic&distribution=expo-go>

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



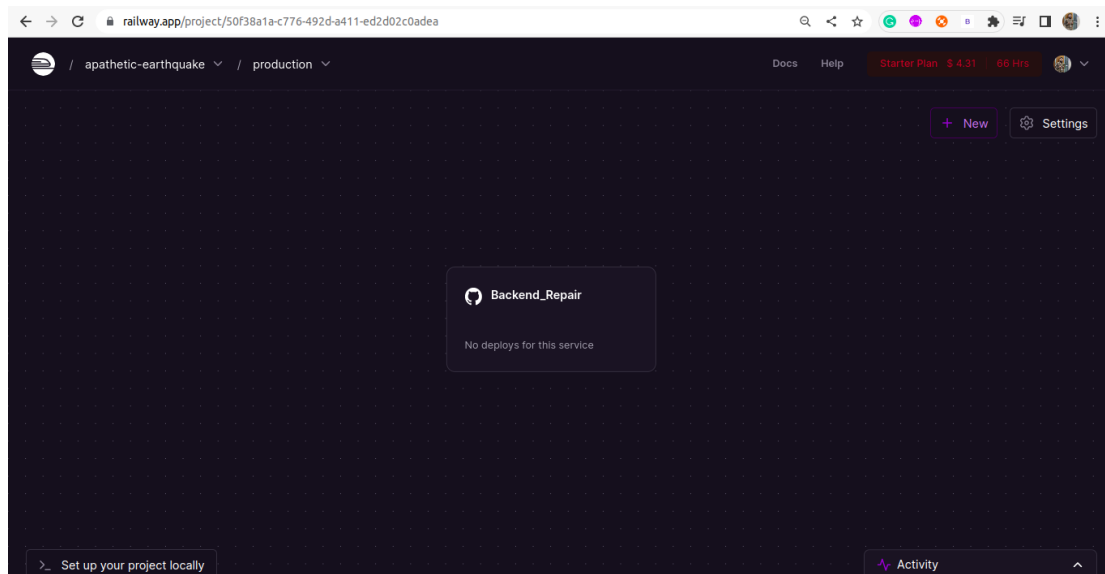
We deploy our Admin panel in Netlify.

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



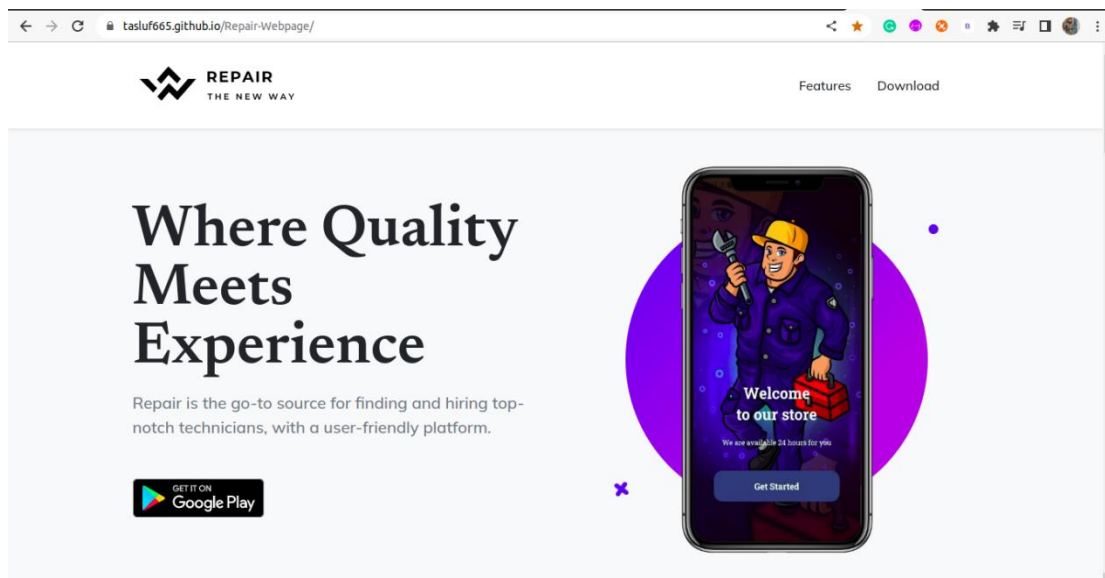
We deploy our Backend part (Nodejs) in railway.

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

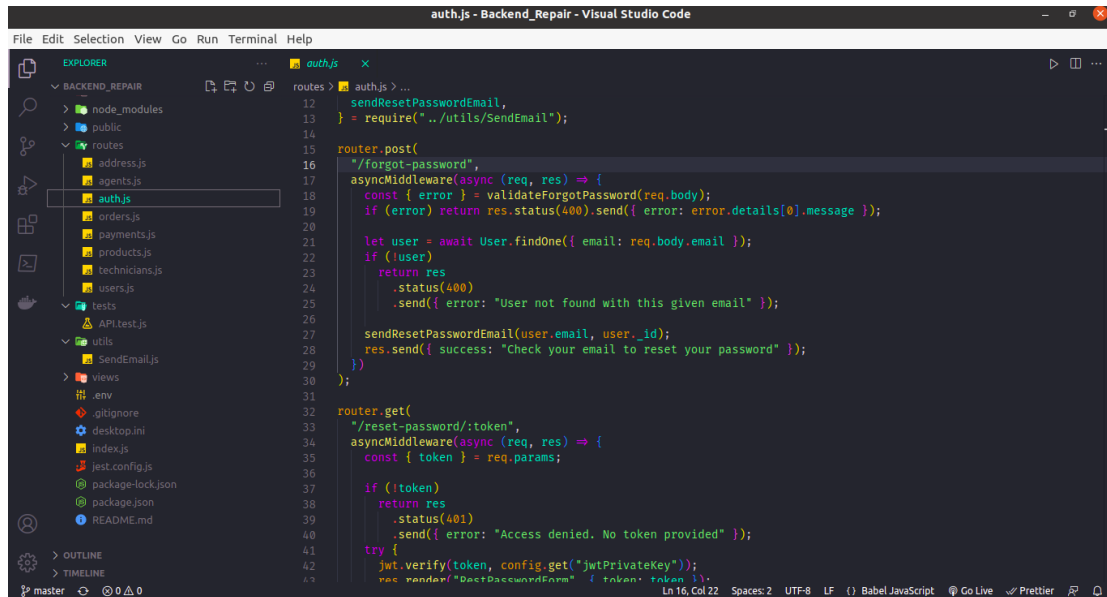
Project github Links:

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

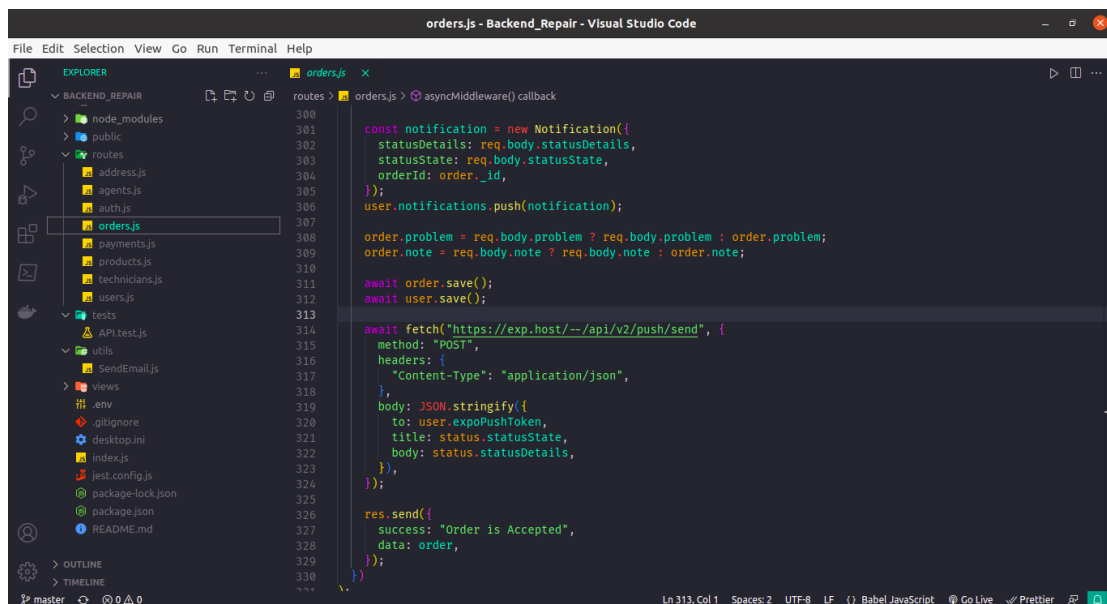
## Backend Part REST API

### API Routes:



```
auth.js - Backend_Repair - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
BACKEND_REPAIR
  node_modules
  public
  routes
    address.js
    agents.js
    auth.js
    orders.js
    payments.js
    products.js
    technicians.js
    users.js
  tests
  API.test.js
  utils
    SendEmail.js
  views
  .env
  .gitignore
  desktop.ini
  index.js
  jest.config.js
  package-lock.json
  package.json
  README.md
OUTLINE
TIMELINE
master

routes > auth.js > ...
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
33 router.get(
34   "/reset-password/:token",
35   asyncMiddleware(async (req, res) => {
36     const { token } = req.params;
37
38     if (!token)
39       return res
40         .status(401)
41         .send({ error: "Access denied. No token provided" });
42
43     try {
44       jwt.verify(token, config.get("jwtPrivateKey"));
45       res.render("ResetPasswordForm", { token: token });
46     } catch (error) {
47       return res.status(401).send({ error: "Access denied. No token provided" });
48     }
49   });
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
Ln 16, Col 22 Spaces: 2 UTF-8 LF ( ) Babel JavaScript Go Live Prettier
```



```
orders.js - Backend_Repair - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
BACKEND_REPAIR
  node_modules
  public
  routes
    address.js
    agents.js
    auth.js
    orders.js
    payments.js
    products.js
    technicians.js
    users.js
  tests
  API.test.js
  utils
    SendEmail.js
  views
  .env
  .gitignore
  desktop.ini
  index.js
  jest.config.js
  package-lock.json
  package.json
  README.md
OUTLINE
TIMELINE
master

routes > orders.js > asyncMiddleware() callback
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
Ln 313, Col 1 Spaces: 2 UTF-8 LF ( ) Babel JavaScript Go Live Prettier
```

```
address.js - Backend_Repair - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
BACKEND_REPAIR
node_modules
public
routes
address.js
agents.js
auth.js
orders.js
payments.js
products.js
technicians.js
users.js
tests
API.test.js
utils
SendEmail.js
views
.env
.gitignore
desktop.ini
index.js
jest.config.js
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
master 0 0 0

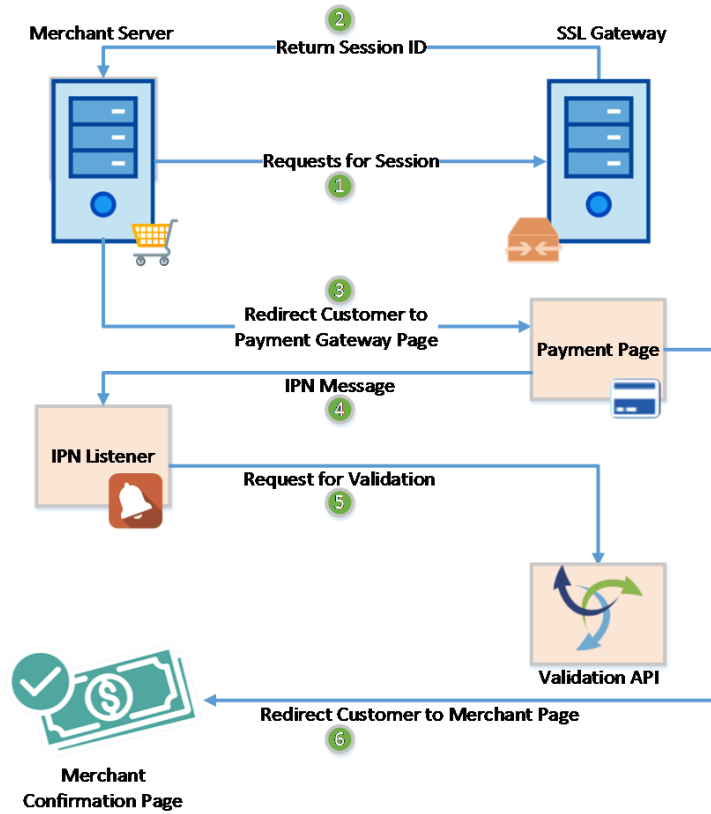
routes > address.js > asyncMiddleware() callback
1 const express = require("express");
2 const router = express.Router();
3 const { Address, validateAddress } = require("../models/address");
4 const asyncMiddleware = require("../middleware/async");
5 const auth = require("../middleware/auth");
6 const 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("-_id");
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
28 router.post(
29   "/",
30   [auth, admin],
31   asyncMiddleware(async (req, res) => {
32     const { error } = validateAddress(req.body);
```

```
agents.js - Backend_Repair - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
BACKEND_REPAIR
node_modules
public
routes
address.js
agents.js
auth.js
orders.js
payments.js
products.js
technicians.js
users.js
tests
API.test.js
utils
SendEmail.js
views
.env
.gitignore
desktop.ini
index.js
jest.config.js
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
master 0 0 0

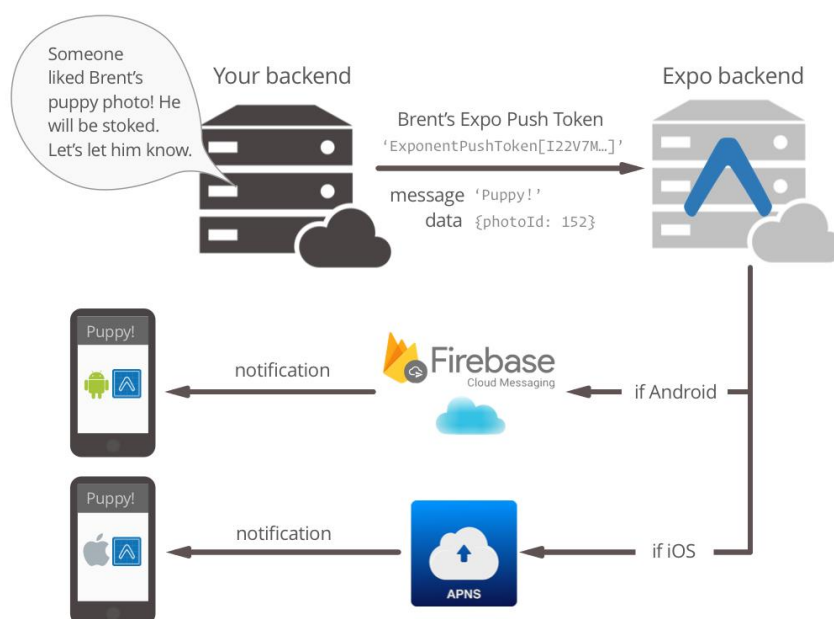
routes > agents.js > asyncMiddleware() callback
60     data: agent,
61   });
62 });
63 );
64
65 router.post(
66   "/",
67   [auth, admin],
68   asyncMiddleware(async (req, res) => {
69     const { error } = validateAgent(req.body);
70     if (error) return res.status(400).send({ error: error.details[0].message });
71
72     const agent = new Agent({
73       name: req.body.name,
74       email: req.body.email,
75       phone: req.body.phone,
76       whatsappNumber: req.body.whatsappNumber,
77       region: req.body.region,
78       city: req.body.city,
79       area: req.body.area,
80       location: req.body.location,
81     });
82
83     await agent.save();
84     res.send({
85       success: "Agent is added successfully",
86       data: agent,
87     });
88   });
89 );
90
```

## Appendix F

SSLCOMMERZ Technical or Backend Integration Process. This diagram explain how our payment system is working. E2



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

<h2>Turnitin Originality Report</h2> <p>Processed on: 04-Jan-2023 11:58 +06 ID: 1988429762 Word Count: 8475 Submitted: 1</p> <p><b>191-15-12594 By Md. Assadujjaman Tilok</b></p>		<table border="1"><thead><tr><th>Similarity Index</th><th>Similarity by Source</th></tr></thead><tbody><tr><td><b>22%</b></td><td>Internet Sources: 17% Publications: 1% Student Papers: 18%</td></tr></tbody></table>	Similarity Index	Similarity by Source	<b>22%</b>	Internet Sources: 17% Publications: 1% Student Papers: 18%
Similarity Index	Similarity by Source					
<b>22%</b>	Internet Sources: 17% Publications: 1% Student Papers: 18%					

2% match (Internet from 15-Apr-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7771/182-15-11440%20%2823%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7771/182-15-11440%20%2823%25%29.pdf?isAllowed=y&amp;sequence=1</a>
2% match (student papers from 02-Jan-2023) <a href="#">Submitted to Daffodil International University on 2023-01-02</a>
2% match (Internet from 04-Sep-2022) <a href="https://businessinspection.com.bd/home-appliance-industry-in-bd/">https://businessinspection.com.bd/home-appliance-industry-in-bd/</a>
1% match (Internet from 21-Nov-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6798/172-15-9824%20%20%2821%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6798/172-15-9824%20%20%2821%25%29.pdf?isAllowed=y&amp;sequence=1</a>
1% match (Internet from 26-Oct-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7395/173-15-10295%20%2822%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7395/173-15-10295%20%2822%25%29.pdf?isAllowed=y&amp;sequence=1</a>
1% match (Internet from 27-Jan-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6738/142-15-3747%20%2823%25%29%20clearance.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6738/142-15-3747%20%2823%25%29%20clearance.pdf?isAllowed=y&amp;sequence=1</a>
1% match (student papers from 28-Sep-2011) <a href="#">Submitted to Higher Education Commission Pakistan on 2011-09-28</a>
1% match (student papers from 26-Sep-2011) <a href="#">Submitted to Higher Education Commission Pakistan on 2011-09-26</a>
1% match (student papers from 07-Jan-2022) <a href="#">Submitted to National Open University of Nigeria on 2022-01-07</a>
1% match (student papers from 28-Nov-2014) <a href="#">Submitted to New University of Astana on 2014-11-28</a>
1% match (student papers from 31-May-2019) <a href="#">Submitted to Kelvin Grove State College on 2019-05-31</a>
1% match (student papers from 25-Sep-2020) <a href="#">Submitted to University of Portsmouth on 2020-09-25</a>
< 1% match (Internet from 13-Feb-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6796/172-15-9833%20%2820%25%29%20clearance.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6796/172-15-9833%20%2820%25%29%20clearance.pdf?isAllowed=y&amp;sequence=1</a>
< 1% match (Internet from 21-Nov-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6799/172-15-9821%20%2812%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6799/172-15-9821%20%2812%25%29.pdf?isAllowed=y&amp;sequence=1</a>
< 1% match (Internet from 20-Nov-2022) <a href="http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5726/163-15-8529%20%2820%25%29.pdf?isAllowed=y&amp;sequence=1">http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5726/163-15-8529%20%2820%25%29.pdf?isAllowed=y&amp;sequence=1</a>
< 1% match (Internet from 20-Nov-2022)

https://www.turnitin.com/newreport\_printview.asp?eq=1&eb=1&esm=10&oid=1988429762&sid=0&n=0&m=2&svr=41&r=43.03213559745418&iang=e... 1/11