Project Report On Bus Management System

Abdullah Al Safayt Hossan Rafi (2019831026)
Sakibul Islam (2019831062)
Zawadul Karim Arnob (2019831064)
Mohammed Tusher Imran Mashrafi (2019831069)
Mahir Al Shahriar (2019831077)

Department Of Software Engineering
Shahjalal University Of Science and Technology, Sylhet

To

Dr.Ahsan Habib
Assistent Professor

Institute of Information and Communication Technology
Shahjalal University Of Science and Technology, Sylhet

November 16, 2023

Contents

1 Introduction

Our software project is called BUS-TICKET MANAGEMENT SYSTEM. Its an online web app designed to allow users to buy ticket online, and also allows BUS SERVICE Companies to manage various business processes and resources like buses, employees etc. Most companies like UNIQUE, ENA, and GREENLINE still conduct majority of their business processes manually, hence our software will create value in this market by automating most of these processes.

2 Requirement Gathering

We visited multiple Intercity Bus Service Company Offices in Sylhet, including: ENA, Unique, Greenline,

We analysed their As-is System by various Elicitiation technique:

1. Document Analysis: The documents gathered were :

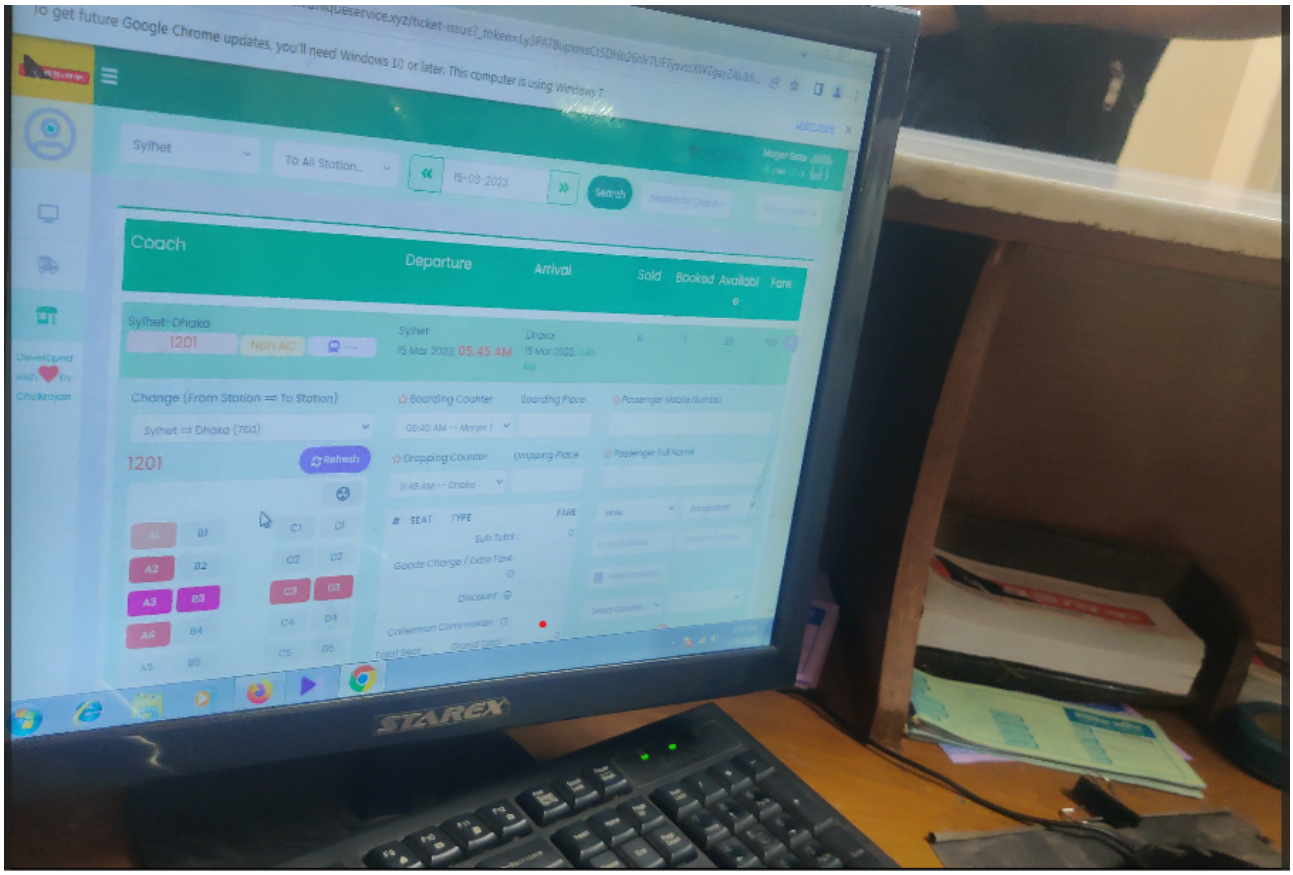

Figure 1: picture of existing ticket purchasing systems

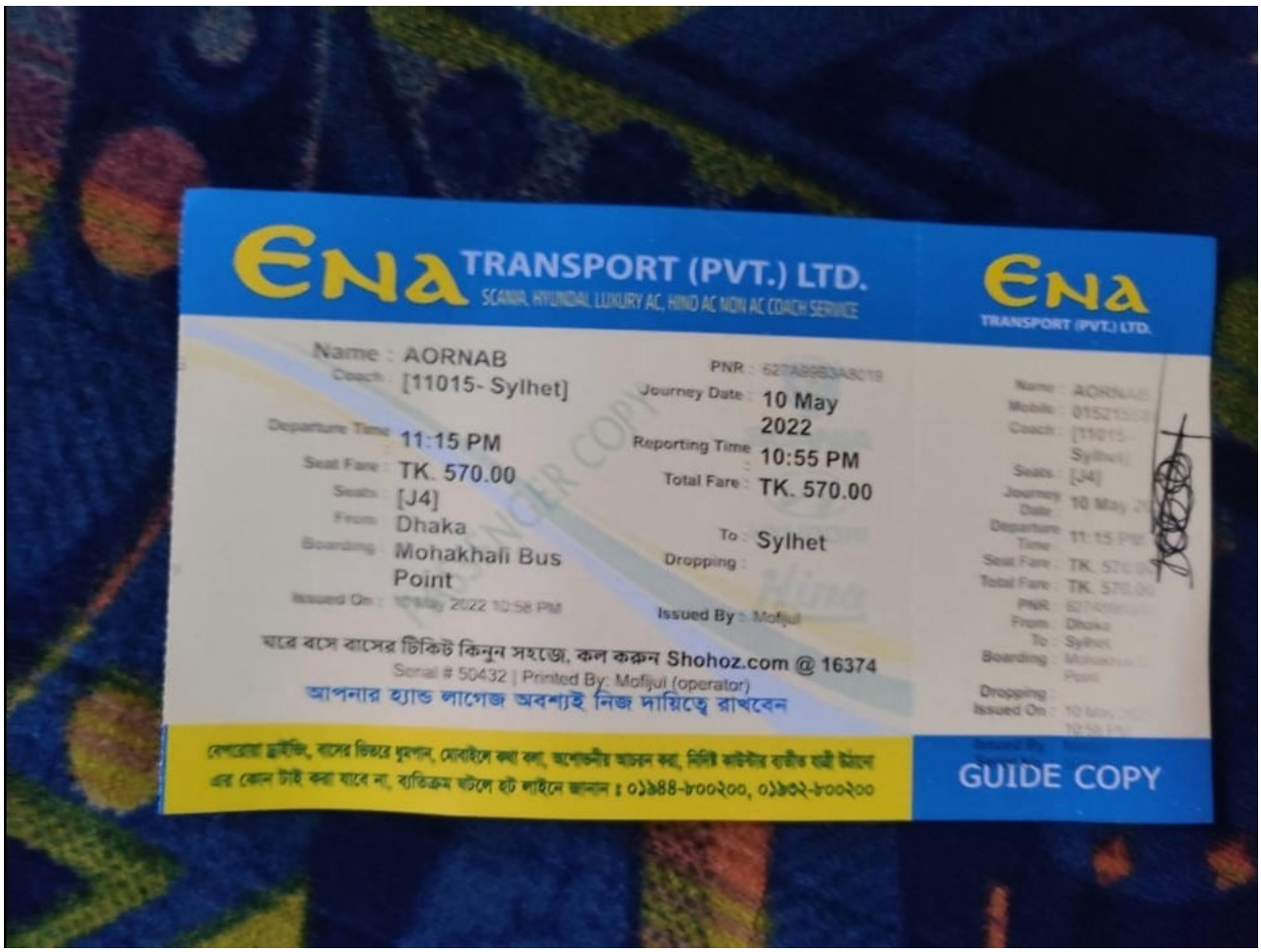

Figure 2: picture of Ticket

Figure 3: picture of Ticket

Figure 4: picture of ena bus route

2. Interview: click here to see Interview video

Figure 5: Interview with manager of unique counter at humayun chattar

And We interviewed a manager of Greenline Bus Service: click here to see Interview audio

We identified 2 main actors:

1. User representing the customer and passenger
2. Admin representing the people handling the business management operations.

- Profile Management System: One of the functional requirements of our system is the users are allowed to login, register, and manage/edit their profile information. Our system also allows the admin to login, giving access to resource management and system parameters
- Business Resource Management System: The System allows the Admin to manage Resources. Buses, Employees, Trips, and Bus Counters. These follow the basic CRUD system.
- Ticket Purchasing System: Our System Allows Users to purchase Tickets online, by first invoicing their payment, and letting the Admin confirm the Payment Transaction before mailing the ticket to their email.

Expanding on the functional requirements, we get the following use cases:

2.1 Use Case Diagram

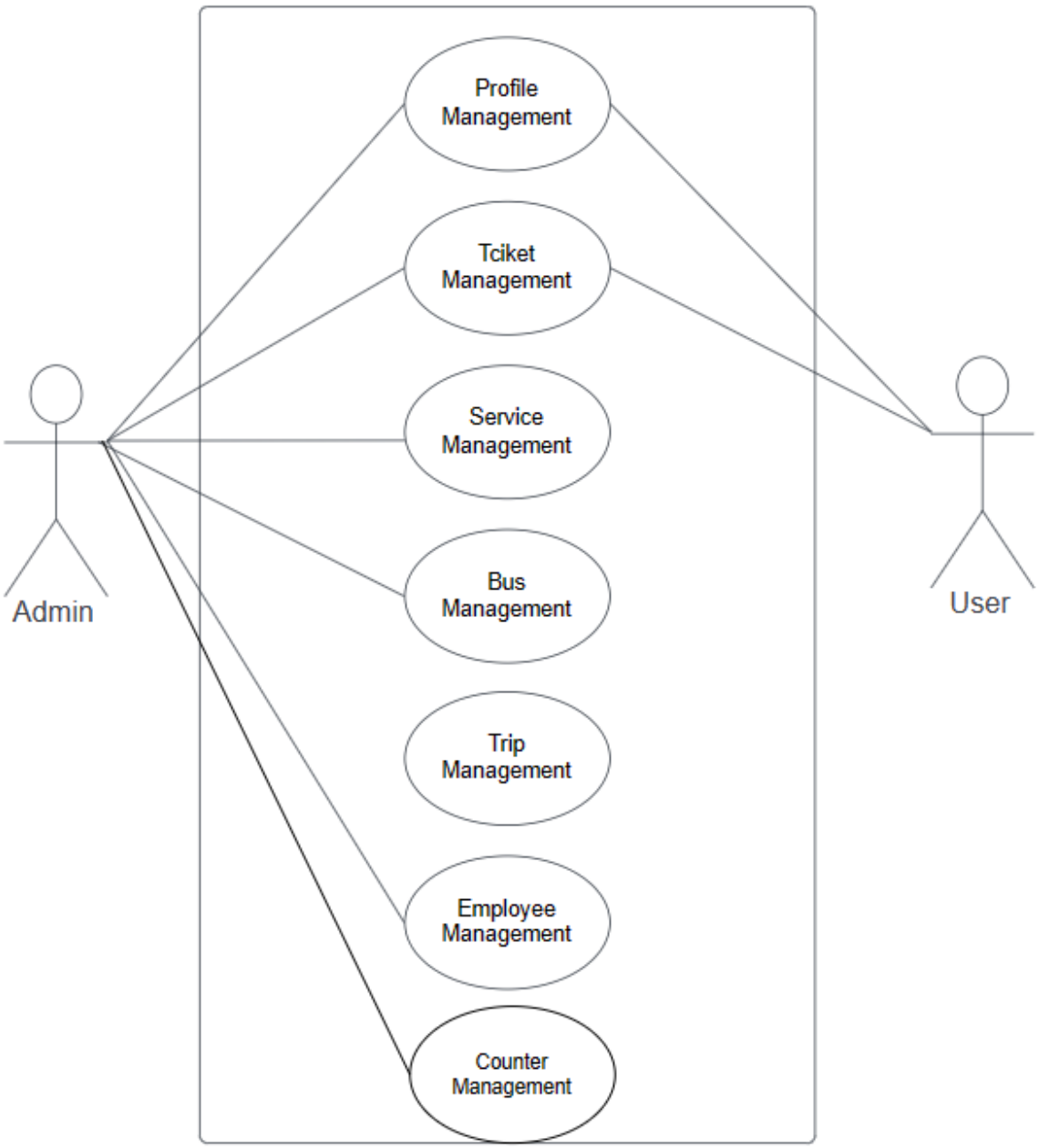

Figure 6: Usecase Diagram showing all the major processes

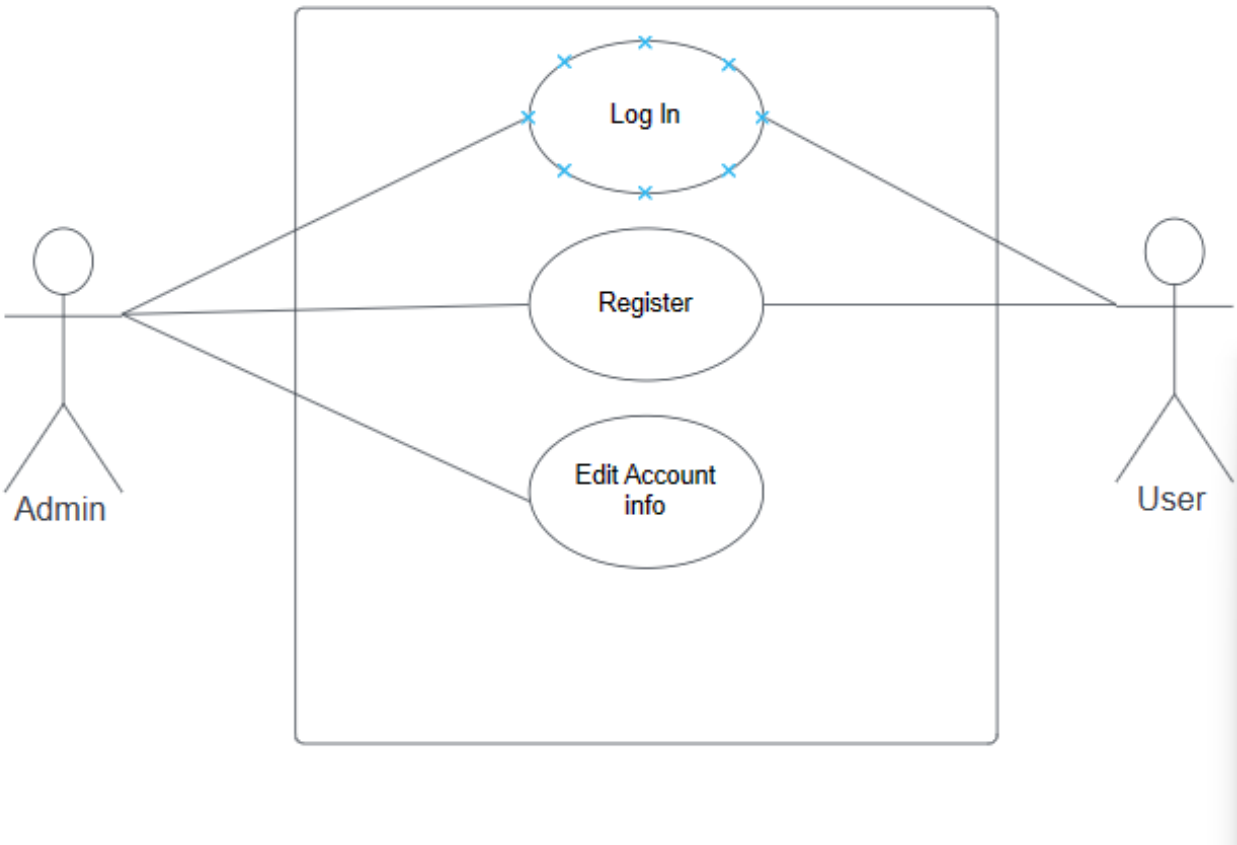

Figure 7: Usecase diagram for Profile Management

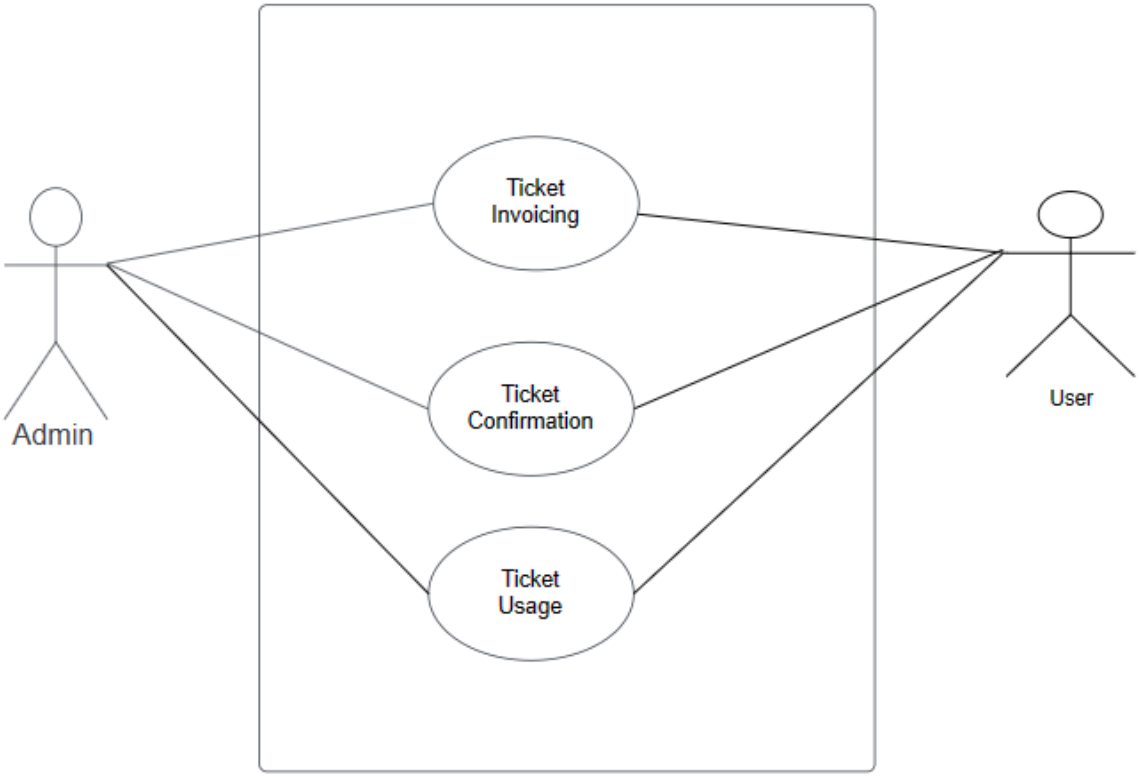

Figure 8: Use Case Diagram for Ticket Management

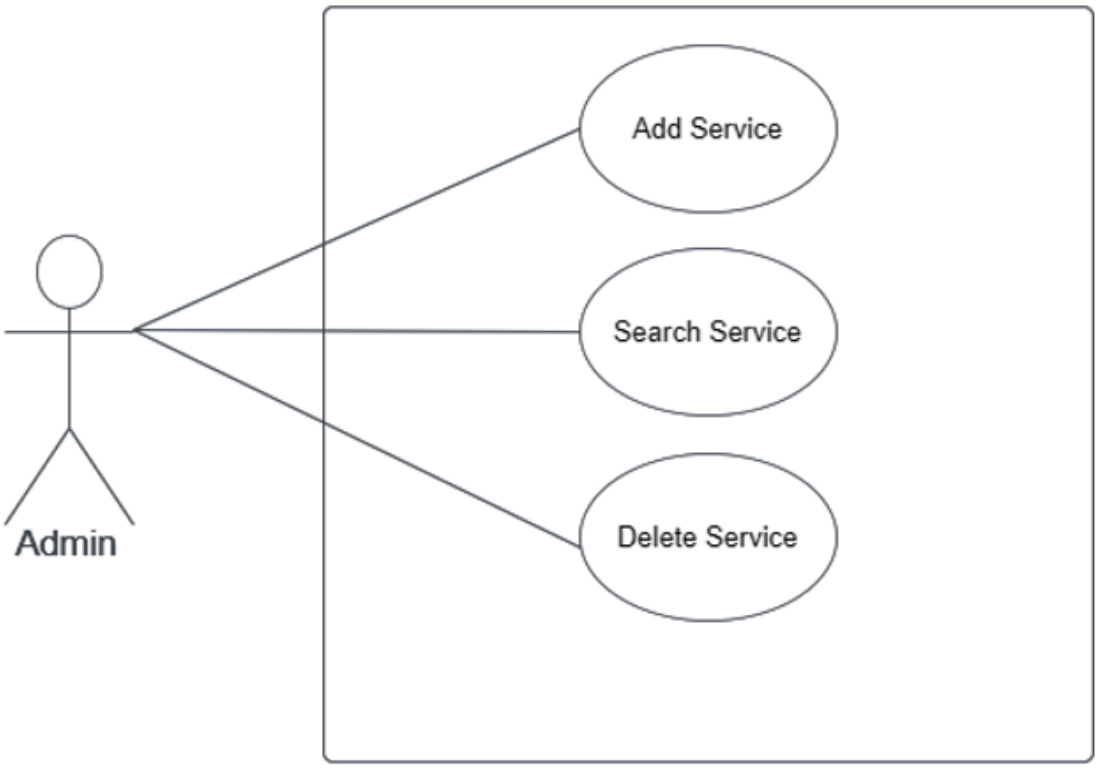

Figure 9: Use Case Diagram for Service Management

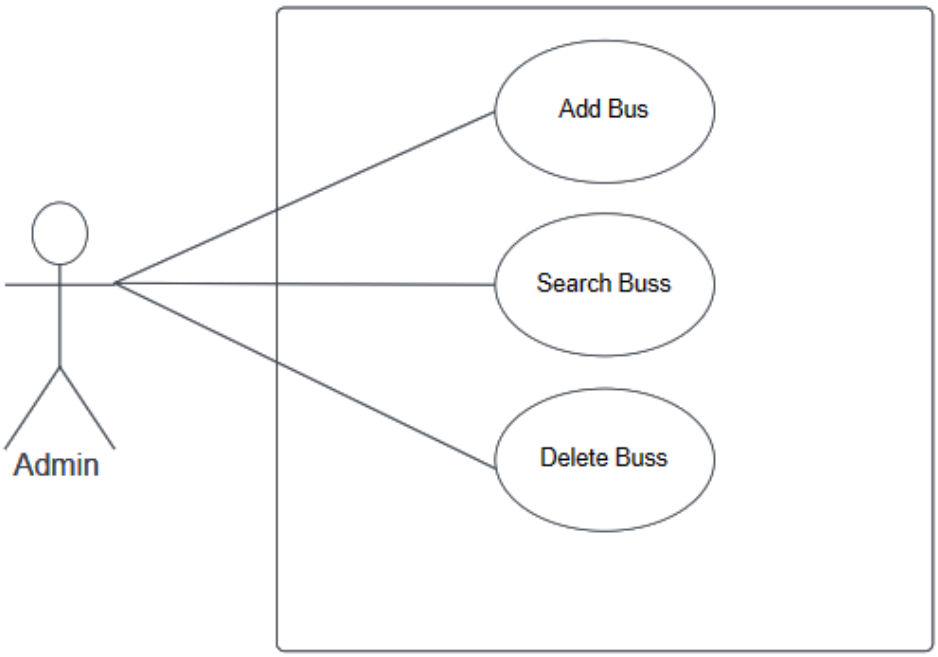

Figure 10: Use Case Diagram for Bus Management

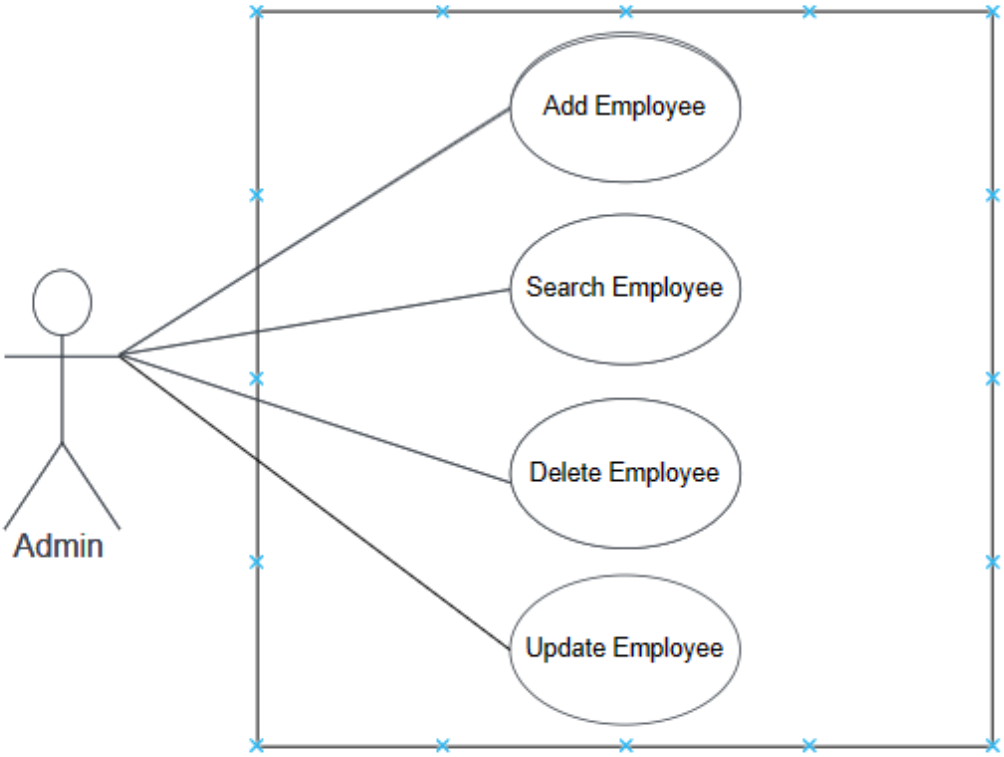

Figure 11: Use Case Diagram for Employee Management

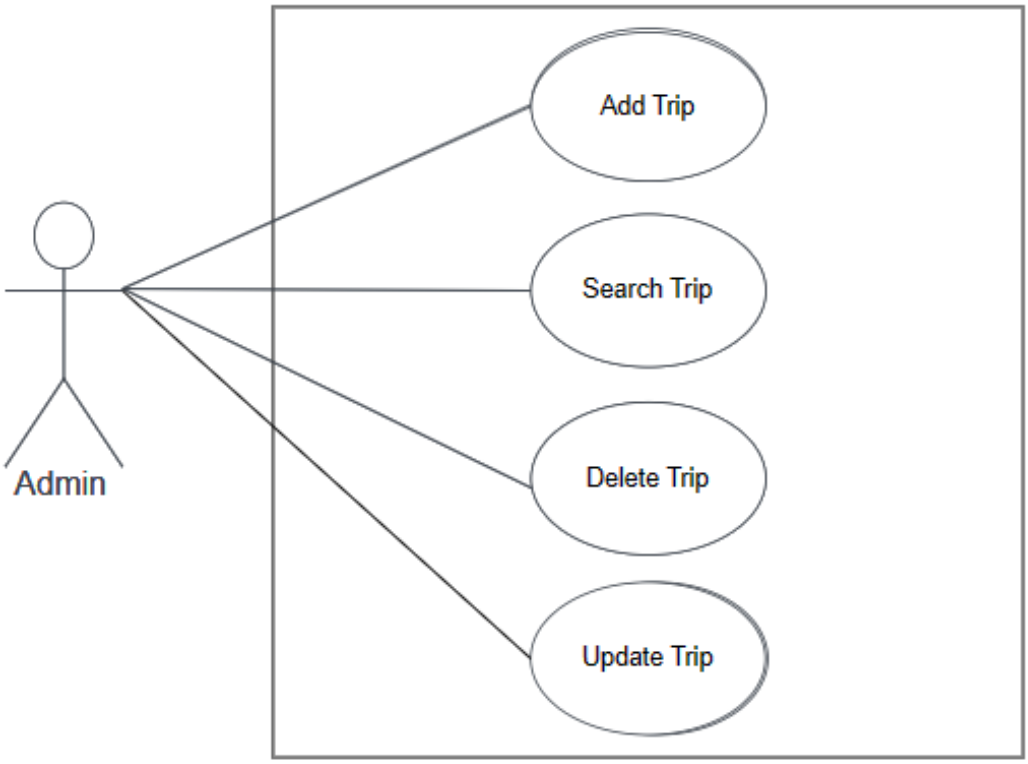

Figure 12: Use Case Diagram for Trip Managemen

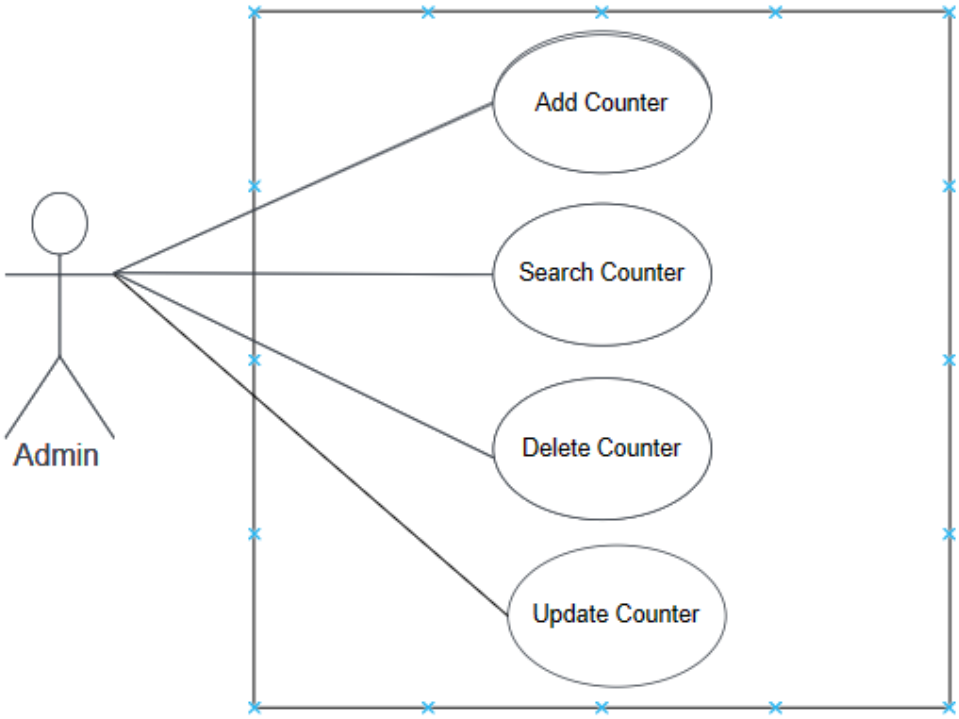

Figure 13: Use Case Diagram for counter Managemen

2.2 Process Models:

We used data flow diagramming technique, to model the processes and the flow of data through the system. The notation used are Gane And Sarson Symbols.

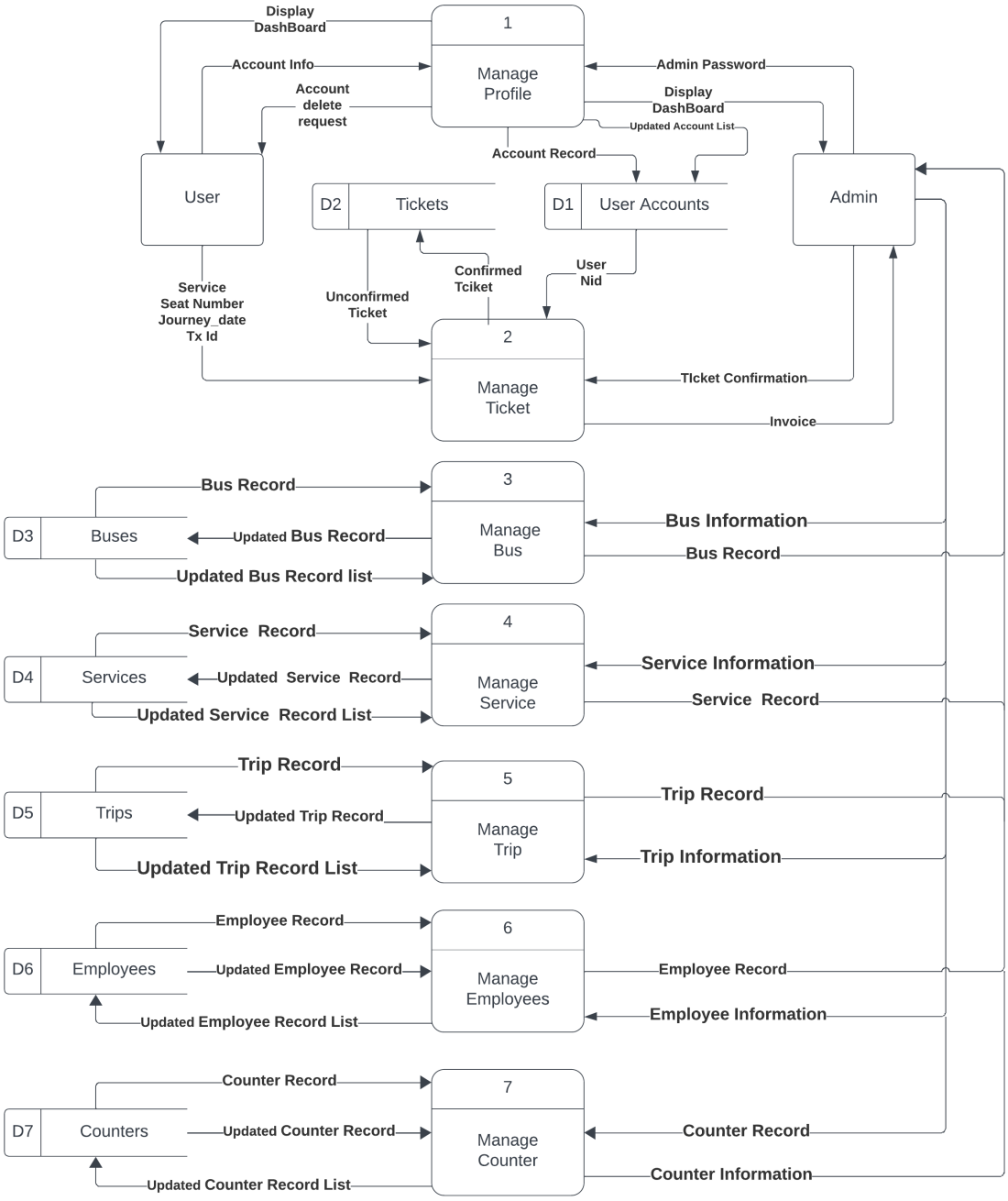

Figure 14: Level 1 DFD

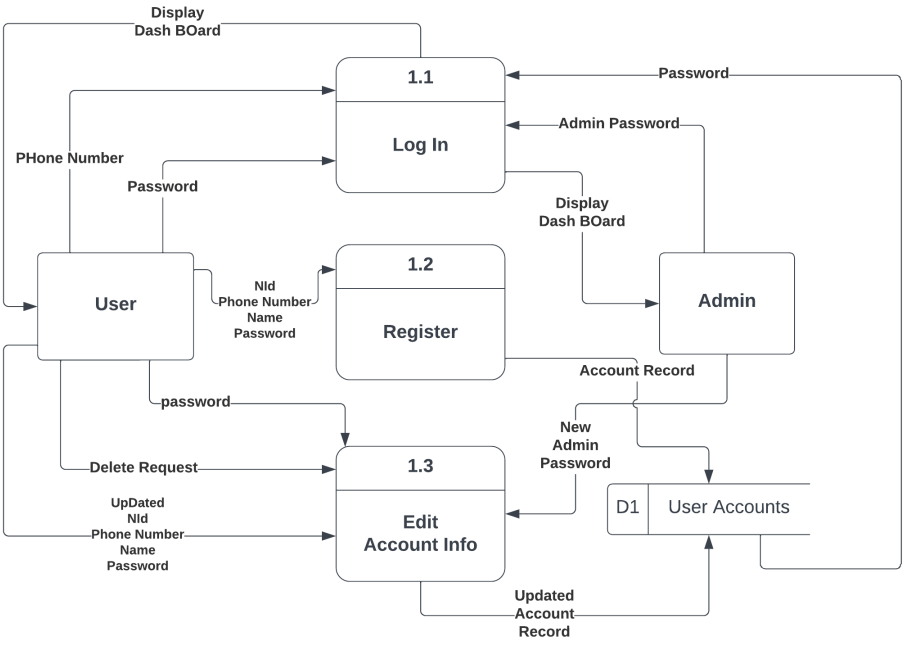

Figure 15: Level 2 DFD for profile management process

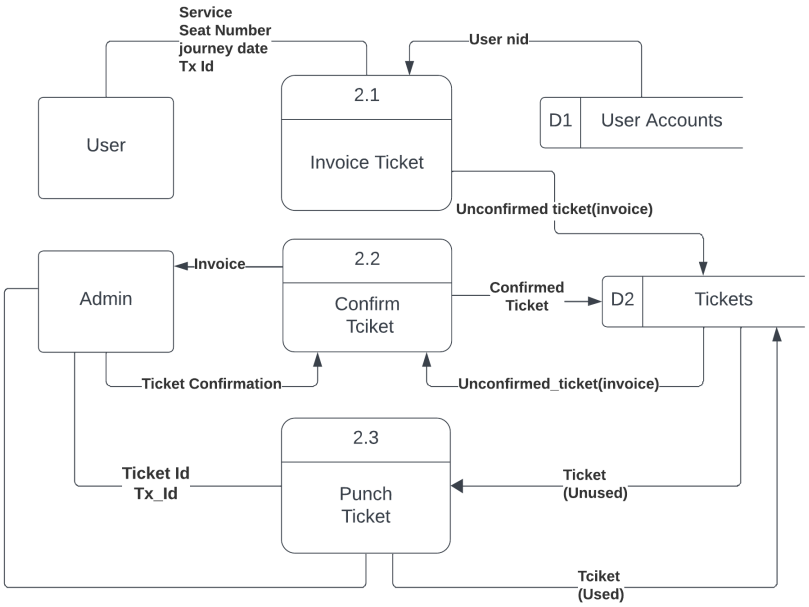

Figure 16: Level 2 DFD forTicket management process

2.3 Data Modeling

We used Entity relationship diagramming to model the major data used in our system. The strong Entities could be identified by inspecting the data stores in the dfd, and the data elements that flow into and out of them.

The entities corresponding to the data stores D1 to D6 are :

1. USER
2. TICKET
3. BUS
4. SERVICE
5. TRIP
6. EMPLOYEE
7. COUNTER

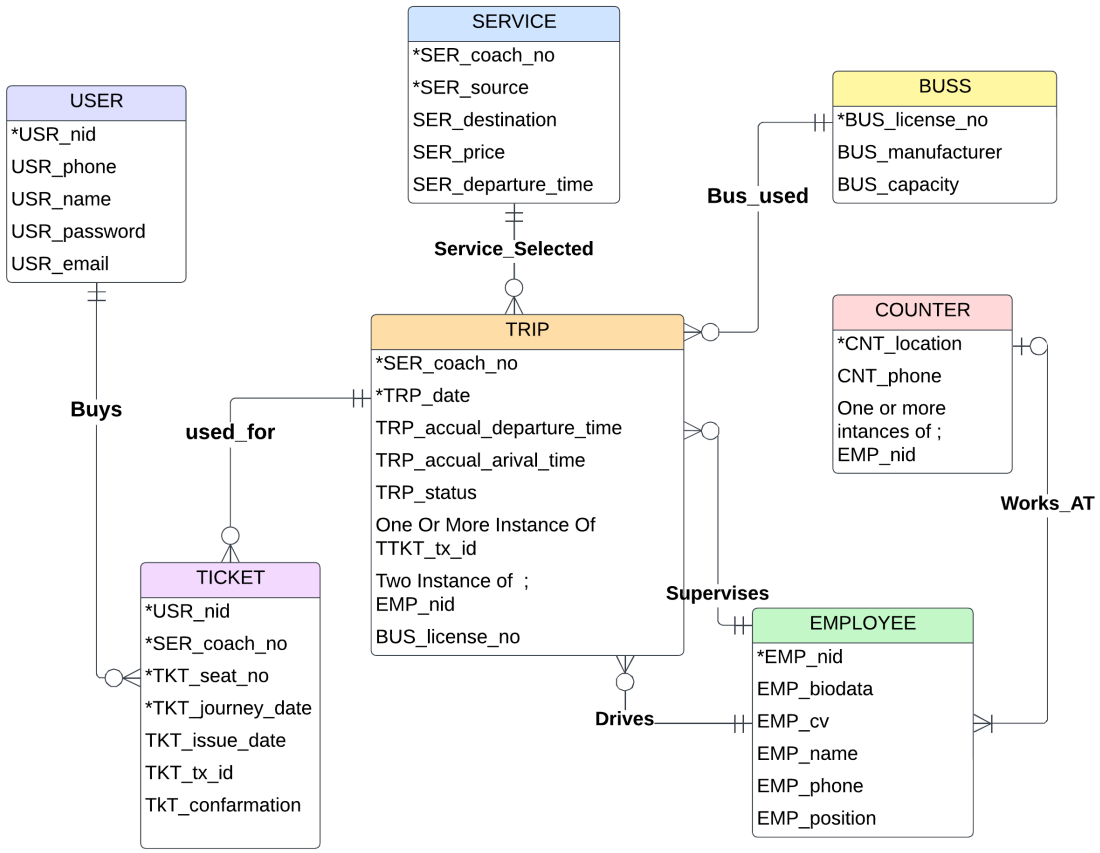

Figure 17: Logical Entity Relationship Diagram

3 SYSTEM DESIGN:

3.1 Data Design

We validated the Logical ERD, by normalising the entities, and balancing it with the DFD. Then it was converted to a physical Data model, which is depicted by the Database diagram produced by XAMP working on mysql code. Notable changes are:

1. Multiple instances of employees working at a counter were normalized by creating an intersection entity: **COUNTER_EMPLOYEE**.
2. Multiple instances of on-board passengers with tickets on a TRIP were resolved by adding a **used** field in the **TICKET** entity, to mark if the ticket instance was punched or not.

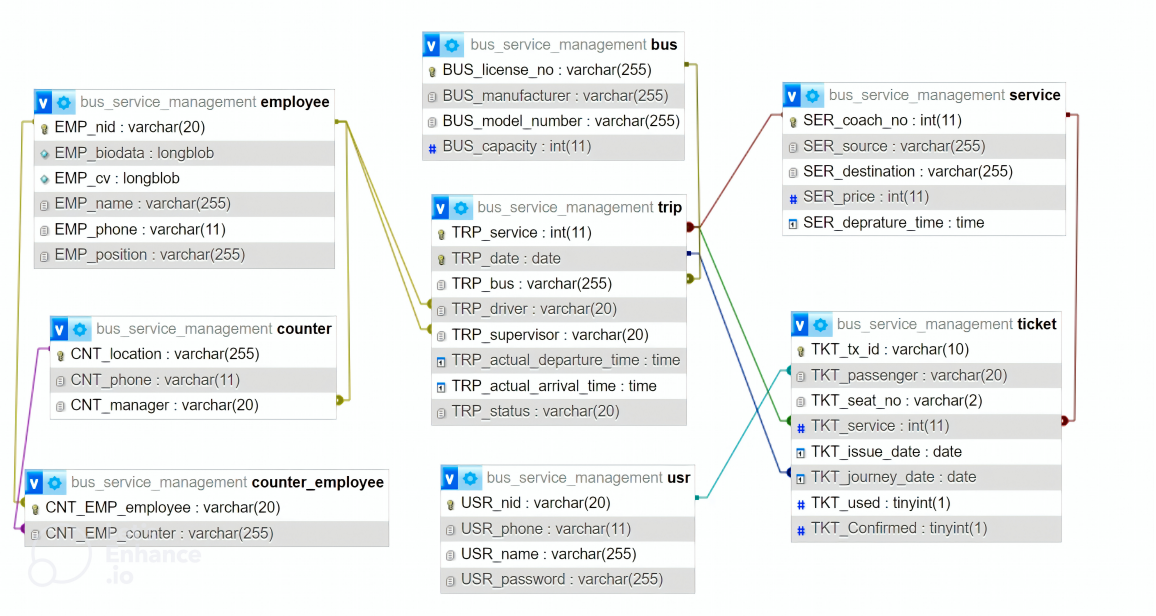

Figure 18: Database Diagram

3.2 UI Design

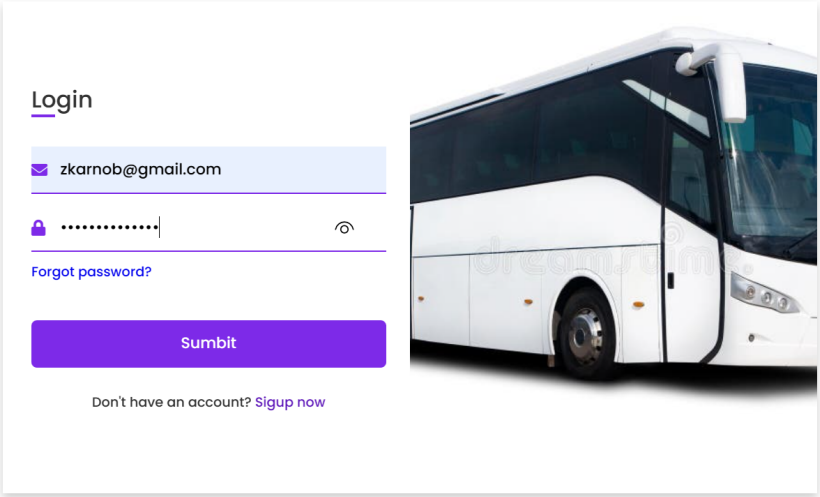

Figure 19: User Login Page

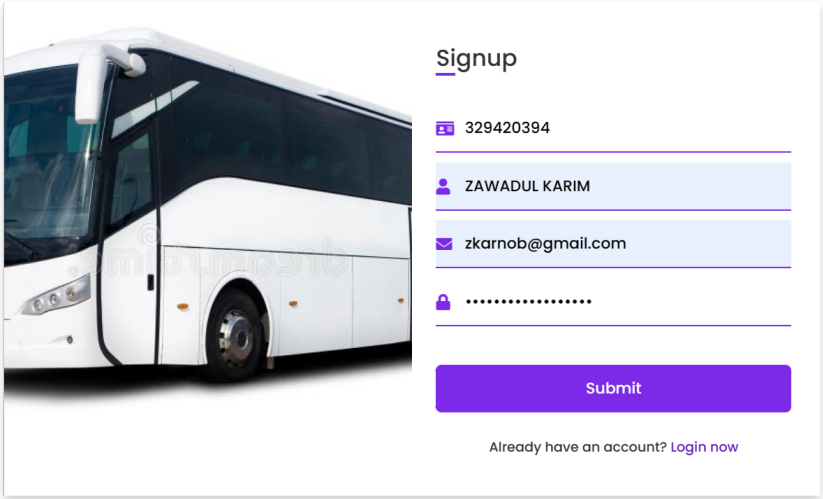

Figure 20: User Signup Page

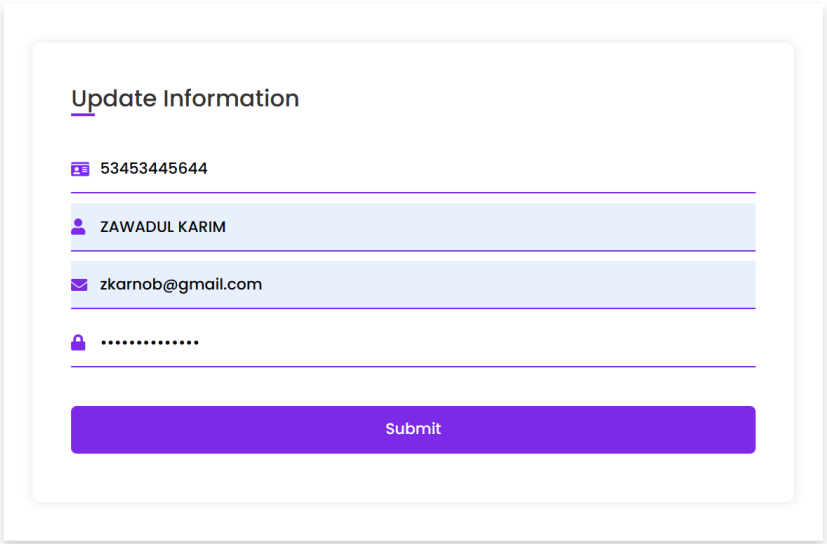

Figure 21: User Update Profile

Figure 22: Service Selection for Ticket Purchase

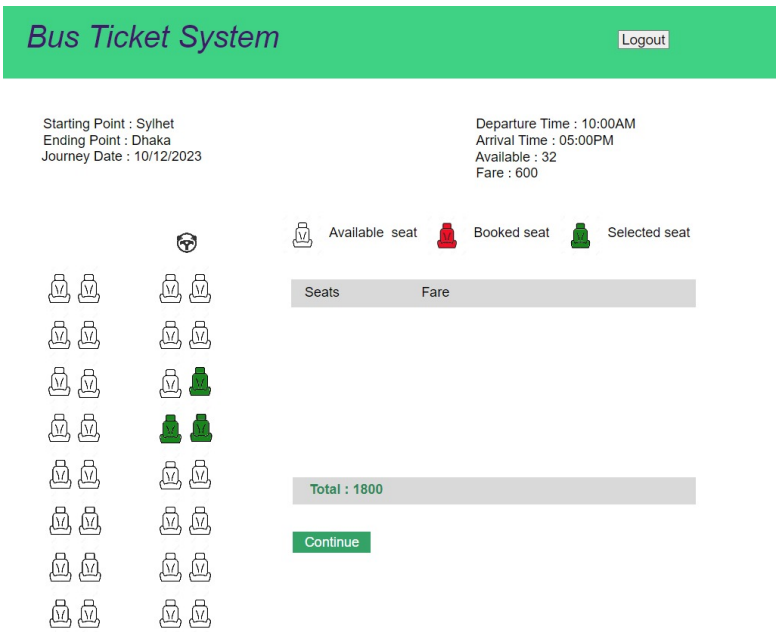

Figure 23: Seat Selection

Figure 24: Admin Dashboard

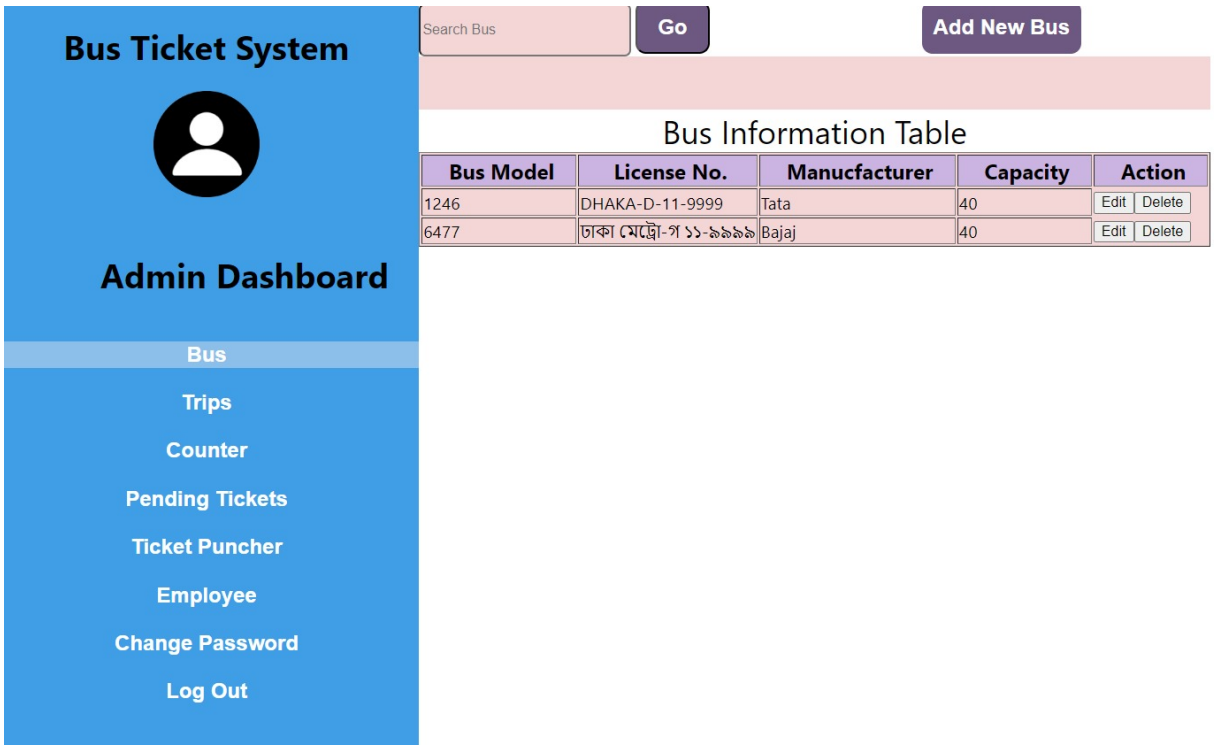

Figure 25: Bus Management Page

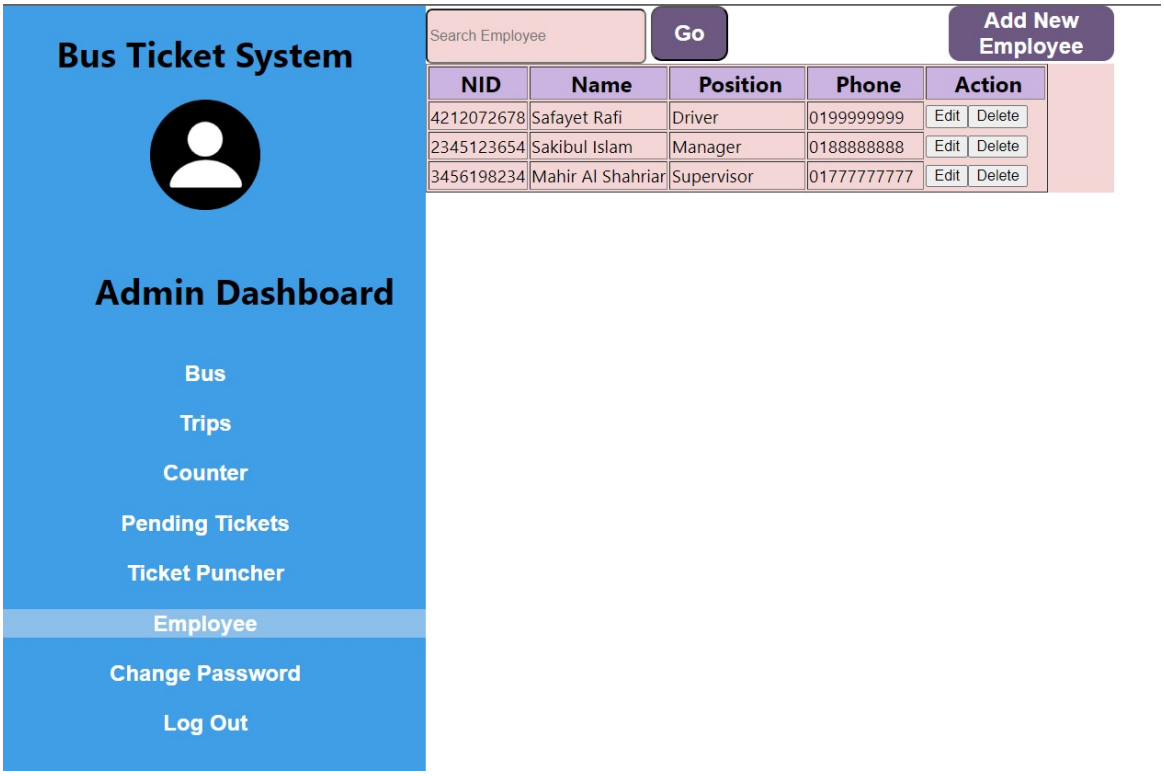

Figure 26: Employee Management Page

Figure 27: Trip Management Page

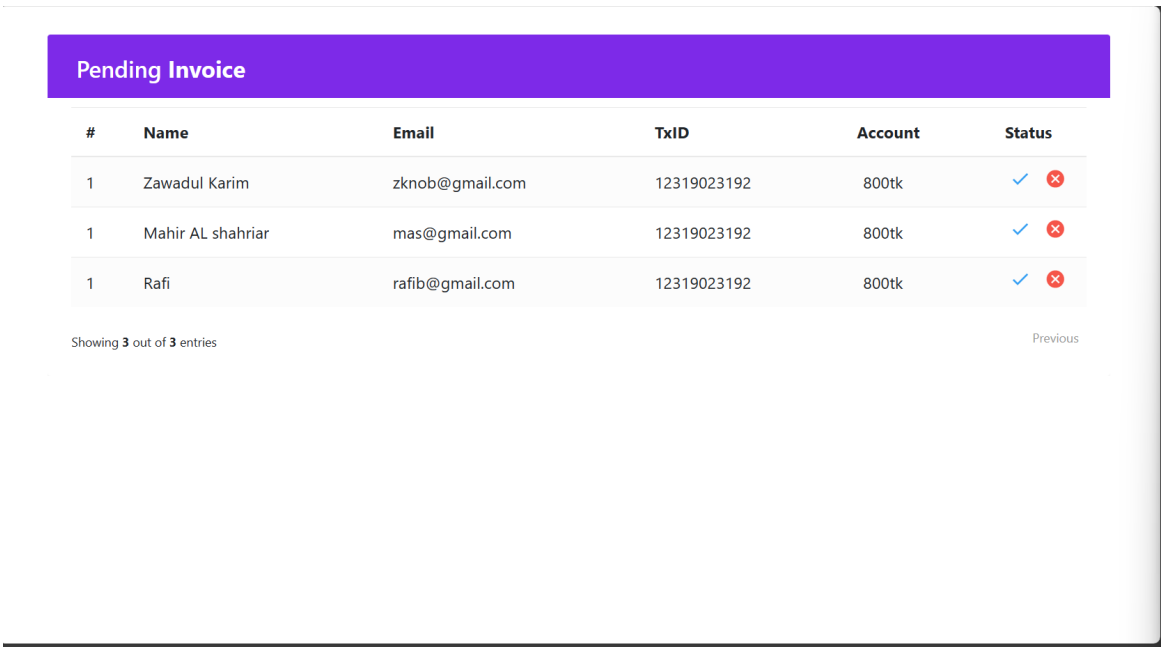

Figure 28: Page for Confirming Ticket Invoices

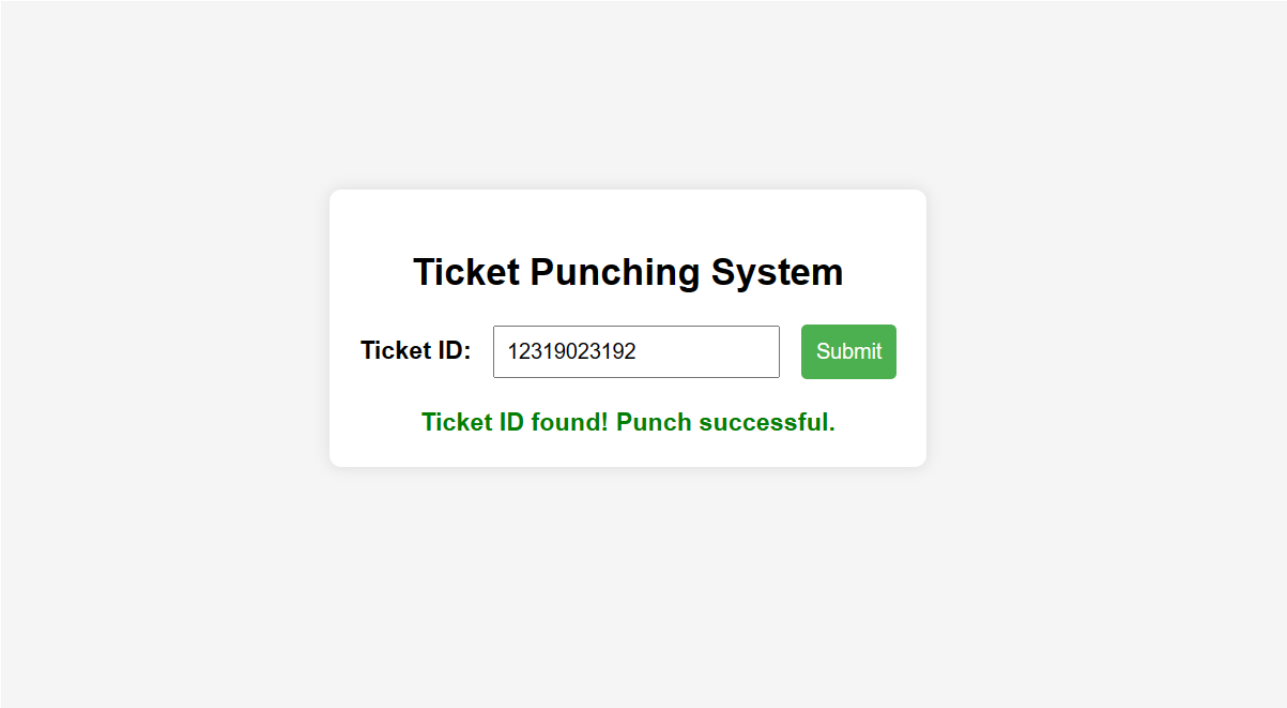

Figure 29: Ticket Punching Page

4 Implementation

Our Project was implemented using some technologies and languages. The Languages are the following

1. JavaScript
2. HTML
3. CSS
4. MySQL

5. Technologies used while creating the project:
   (a) NodeJs
   (b) Express
   (c) ReactJs
   (d) JSON Web Token

The actual code implementation of our project is on Github to track our work progress.

5 Testing

We did unit test on the modules that were implemented. The tests included if our backend routes were sending requests and receiving them correctly while it also helped us to sync with our database correctly. The tool used for testing our routes and request response correctly is "Postman". Though we are yet to do testing with live users on the project that would be done when the whole system comes together and it is ready for the system testing phase

6 Training and Maintenance

This portion contains our nonfunctional requirements and security assurance steps

6.1 Non-Functional Requirements

A nonfunctional requirement is an attribute that dictates how a system operates. It makes applications or software run more efficiently and illustrates the system's quality. We tried our best to ensure some of the most important non-functional requirements like security, performance, reliability, data integrity and easier navigation for our users. Some of the non-functional requirements we want to highlight are listed below:

1. The User Interface was made easier to navigate from one page to another, and the steps to complete tasks are simple for a smooth user experience.
2. The load time for the website was kept at a minimum.
3. User account safety was ensured using strong password systems and hash functions while storing.
4. User information change can be done only when the user is authenticated through the use of JSON Web Token.
5. The contest creation and participation were also made secure with the use of passwords.

6. The system is easily accessible throughout the time our system is up and running without delay.
7. The data was handled carefully to ensure data integrity, and the data accuracy remains at its peak.