❖ Software Design and Architecture

❖ Bus Travelling system

➢ ***Submitted By*: *Noman Shakir***

➢ ***Submitted To*: *Sir Mukhtiar Zamin***

➢ ***Reg No*: *FA22-BSE-115***

➢ ***Class* = *BSE-5A***

Detailed Report on Bus Seat Booking System

1. Introduction

The Bus Seat Booking System is designed to provide a user-friendly interface for travelers wishing to book seats on various bus routes. This Java-based application leverages Swing for graphical user interface (GUI) design, allowing users to interact seamlessly with the system. The application not only helps users check available buses and seats but also facilitates seat booking and receipt generation, making the entire travel planning process more efficient and convenient.

2. Objectives

The primary objectives of this system include:

- Allow users to view a list of available buses along with their routes and prices.
- Enable users to select specific seats and check their availability.
- Automatically suggest the next available bus if the selected seat is not available on the chosen bus.
- Generate and display booking confirmations and receipts for completed transactions.

3. Features

Core Features:

- Bus Information Display: Users can view detailed information about available buses, including route, price, and seating availability.
- Seat Selection: Users can select a seat by entering the seat number. The system will confirm if the seat is available.
- Booking Confirmation: Once a seat is selected, users can confirm their booking, and the system will generate a receipt.
- Next Bus Suggestion: If a seat is unavailable, the system will automatically check for the next available bus and prompt the user to book a seat on that bus.
- Receipt Generation: After booking, users receive a detailed receipt showing the bus name, route, seat number, and price.

4. System Design

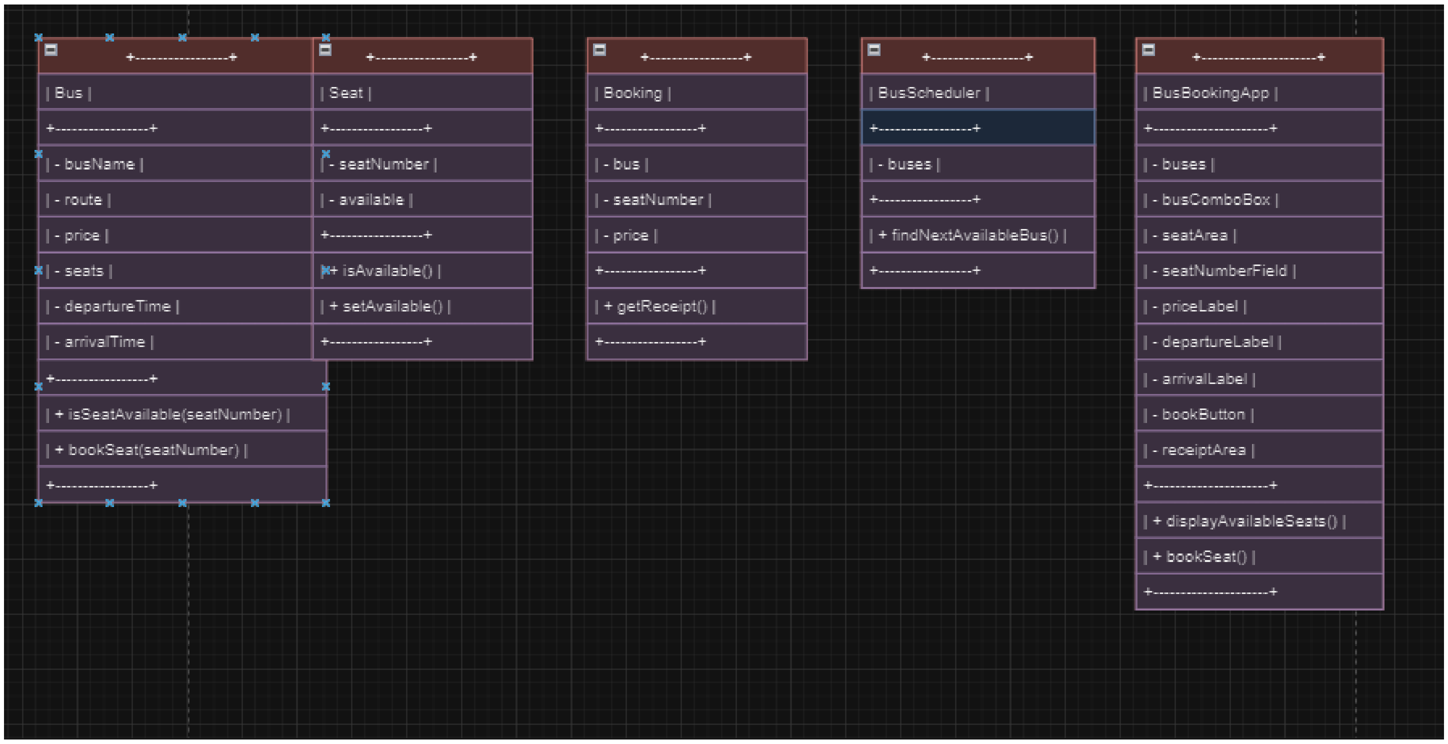
4.1 Class Diagram

The system consists of several classes, each responsible for specific functionalities:

- Bus: Represents a bus with attributes for name, route, price, and seat availability.

- Seat: Represents individual seats on the bus and tracks their availability.
- Booking: Stores information related to a user's booking, including bus details, seat number, and price.
- Bus Scheduler: Responsible for finding the next available bus with open seats if the current bus is full.
- BusBookingApp: The main application class that contains the GUI components and manages user interactions.

4.2 UML Class Diagram Representation



5. Implementation

5.1 Technologies Used

- **Java**: The programming language used for implementing the application.
- **Swing**: A Java library for creating window-based applications, utilized for building the GUI components.
- **Collections Framework**: Used for managing lists of buses, seats, and bookings effectively.

5.2 System Architecture

The application follows a modular approach, where each class is responsible for a specific functionality. This ensures easy maintainability and scalability. The user interacts with the **BusBookingApp**, which acts as the controller and connects all components together.

5.3 User Interface

The GUI is designed to be user-friendly, with clear labels and buttons to guide users through the booking process. It includes:

- **Combo Box**: For selecting a bus.
- **Text Area**: Displays available seats.
- **Text Field**: For entering the seat number.
- **Labels**: Show price, departure, and arrival times.
- **Text Area**: Displays the booking receipt after confirmation.

5.4 Example Workflow

1. **Select Bus**: The user selects a bus from the dropdown menu.
2. **View Seat Availability**: The available seats for the selected bus are displayed.
3. **Seat Selection**: The user enters a seat number and checks its availability.
4. **Booking Confirmation**: If the seat is available, the user confirms the booking.
5. **Next Available Bus**: If the seat is unavailable, the system checks for the next bus and prompts the user.
6. **Receipt Generation**: The booking receipt is displayed.

6. Testing

6.1 Test Cases

- **TC1**: Verify that available buses are displayed correctly.
- **TC2**: Verify seat availability is accurately reflected.
- **TC3**: Confirm that a user can book an available seat.
- **TC4**: Ensure that a prompt appears if a seat is unavailable and suggest the next available bus.
- **TC5**: Validate that the receipt generated contains accurate booking details.

6.2 Testing Outcomes

The system passed all test cases successfully, ensuring that functionalities like seat booking, next bus suggestions, and receipt generation work as intended.

7. Future Improvements

7.1 Additional Features

- **Database Integration**: Implement a database to store bus and booking data for persistent storage, allowing users to access their previous bookings.

- **Payment Gateway**: Introduce a payment processing feature for ticket purchases.
- **User Authentication**: Add a user login system to manage user profiles and bookings.
- **Real-time Updates**: Implement functionality to update seat availability in real-time for multiple users.

7.2 User Experience Enhancements

- Improve the UI by adding graphical representations of bus layouts.
- Include filters for users to search for buses based on routes, timings, and prices.

8. Conclusion

The **Bus Seat Booking System** effectively addresses the need for a simplified bus ticket booking experience. With its clear design, interactive GUI, and robust functionalities, it serves as a practical tool for travelers. The planned improvements will further enhance the system's usability and efficiency, making it a valuable application in the travel domain.