

CHANDIGARH
UNIVERSITY
Discover. Learn. Empower.

Project Report

on

Minor Project

Bus Ticket Booking Systems

Subject Code: 24CAP-652 (Advanced Internet Programming Lab)

Submitted By:
Name: Suraj Kumar
UID: 24MCA20013
Class: 24MCA
Section: 4_A

Submitted To:
Ms. Rohini
Assistant Professor
E14603

University Institute of Computing
Chandigarh University, Gharuan,
Mohali

Table of Contents

- Introduction
- Project Objectives
- Tools & Technologies Used
- Features Overview
- Responsive Design
- Java Functionalities
- Application Analytics
- Future Enhancements
- Conclusions

1. Introduction

The **Bus Ticket Booking System** is a Java-based desktop application designed to simplify and digitize the process of booking bus tickets. This system enables users to search for bus routes, view available seats, and book tickets quickly and efficiently. By replacing traditional paper-based booking methods, this software aims to make ticket reservations more reliable, convenient, and user-friendly.



2. Project Objectives

- To develop a user-friendly system for booking and managing bus tickets.
- To streamline ticket reservation and seat allocation.
- To maintain accurate records of users, routes, and bookings.
- To reduce human errors in manual booking.
- To provide an efficient backend database for data storage and retrieval.

3. Tools & Technologies Used

Tool / Technology Purpose

Java (Swing/AWT) Front-end GUI development

MySQL Database management

JDBC Java-MySQL database connection

NetBeans IDE Project development environment

RS2XML / DbUtils Displaying ResultSets in JTable

4. Features Overview

1. User Registration & Login

- **Registration** allows new users to create an account by providing basic details like name, email, username, and password.
- **Login** uses the registered username and password to authenticate the user.
- Ensures secure access and maintains separate sessions for admin and regular users.

● **Why it matters:** Prevents unauthorized access and enables personalized booking history.

2. Bus Route and Schedule Management (Admin Panel)

- Admin can **add, edit, or remove** bus routes and schedules.
- Fields include source, destination, departure time, arrival time, date, and fare.
- Real-time update to the database ensures users always see the latest routes.

● **Why it matters:** Keeps the system up to date with accurate travel options and schedules.

3. Seat Availability Check

- When a user selects a route and date, the system shows available buses and their seat status.
- Booked seats are marked as unavailable (usually shown in red or grayed out).
- Remaining seats are dynamically updated after each booking.

● **Why it matters:** Prevents double bookings and improves user trust in the system.

4. Bus Ticket Booking

- Allows users to:
 - Select a bus
 - Choose seats
 - Confirm booking
- Upon booking, ticket data is stored in the database with details like seat number, user ID, and payment status.

● **Why it matters:** Core functionality — simplifies the traditional ticket booking process.

5. Booking History

- Users can view their **previous bookings**, including:
 - Bus details
 - Dates
 - Seat numbers
 - Booking status
- Stored in the backend and fetched using their account ID.

● **Why it matters:** Helps users track their journeys and serves as proof of booking.

6. Ticket Cancellation

- Users can cancel their bookings before the journey date.
- Seats become available again for others to book.
- Cancellation status is updated in both booking history and admin panel.

● **Why it matters:** Adds flexibility and control for users, mimicking real-life booking systems.

7. Real-Time Fare Calculation

- Fare is automatically calculated based on:
 - Selected route
 - Number of seats
 - Fare per seat
- Total cost is displayed before confirmation.

● **Why it matters:** Increases transparency and avoids manual fare calculations.

8. Print or Save Ticket

- After successful booking, the user can **print** or **save** the ticket.
- Ticket contains:
 - Booking ID
 - Passenger name
 - Bus and route details
 - Date and time
 - Total fare
- Can be implemented using Java's `PrinterJob` or external libraries like `iText` for PDF generation.

● **Why it matters:** Gives users a physical or digital copy for future reference or travel verification.

9. Database Integration (MySQL)

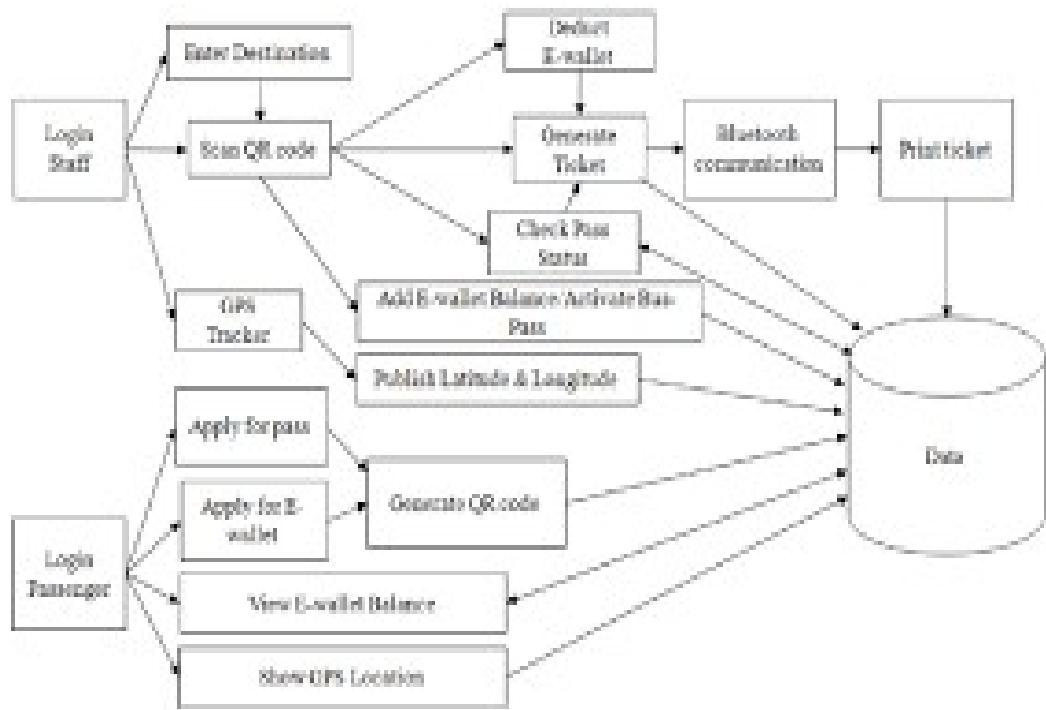
- All data (users, routes, bookings) is stored in a **MySQL** database.
- **JDBC** is used for data communication between Java and MySQL.
- Ensures persistent storage and real-time updates.

● **Why it matters:** Robust backend ensures data integrity and smooth performance.

10. Admin Reports and Monitoring

- Admin can view:
 - Number of bookings per day
 - Popular routes
 - Cancelled tickets
 - Total income collected
- Can be shown using `JTable` or exported as reports.

● **Why it matters:** Helps in decision-making and route optimization.

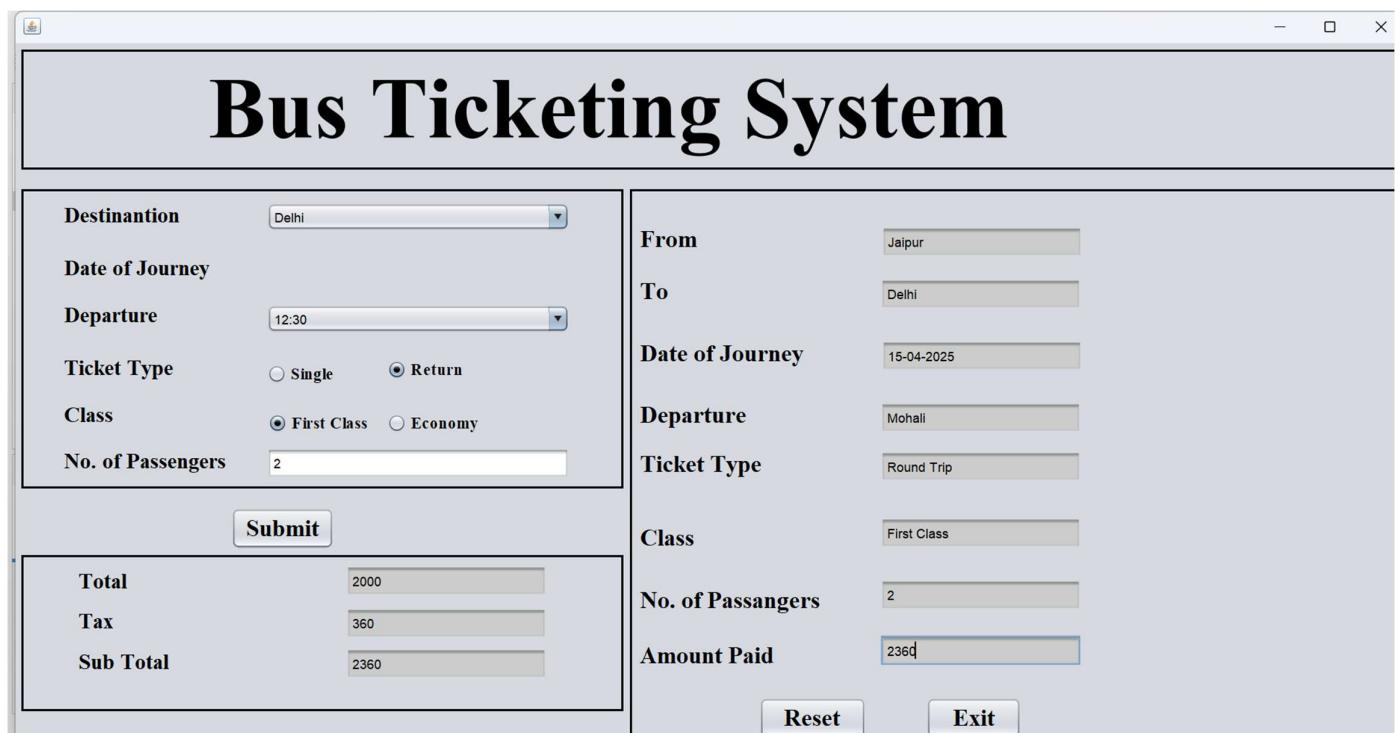


5. Responsive Design

Although this is a **desktop application**, we have implemented a **form of responsive layout** using Java Swing's layout managers to ensure a user-friendly experience across different screen resolutions and sizes. Components such as buttons, labels, tables, and text fields are placed dynamically or with proportional bounds, making the interface accessible on both small and large screens.

If this were a web-based application, responsiveness would extend to various devices like mobile phones and tablets, using CSS and media queries.

- Key UI elements like seat selection, schedule viewing, and booking forms are laid out to avoid clutter and maintain usability.



6. Java Functionalities

The Bus Ticket Booking System utilizes the **full power of Java** to handle user interaction, database communication, and UI design. Below are the key Java features used:

Core Java Functionalities Used:

- Java Swing:** For building the graphical user interface (buttons, tables, labels, panels).
- Event Handling (ActionListener):** Used to respond to user actions like button clicks and form submissions.
- JTable:** For displaying data such as available routes, bookings, and user history in tabular format.
- JDateChooser:** Allows users to select journey dates from a calendar pop-up.
- JPasswordField:** Used to securely enter passwords during login or registration.
- Java JDBC (Java Database Connectivity):** Connects Java with the MySQL database to store and retrieve data dynamically.

- **PreparedStatement:** Ensures safe database queries and protects against SQL injection.
- **Exception Handling:** try-catch blocks are used throughout the code to handle unexpected errors smoothly.
- **Modular Class Design:** Code is separated into different classes like DBconnect, LoginForm, RegisterForm, and Dashboard to ensure maintainability.

● These Java technologies together make the application reliable, modular, and scalable.

7. Application Analytics

Since the current system is a **Java desktop application**, tools like **Google Analytics** are not directly applicable.

However, if the system were redeveloped as a **web-based app**, Google Analytics could track:

- User visits
- Booking trends
- Device types
- Session duration
- Abandonment rates

In a desktop version, we can optionally implement **local analytics** using:

- Log files
- In-app dashboards for the admin
- Database activity tracking (e.g., total bookings, most used routes)

● Future integration with analytics tools can help administrators understand user behavior and optimize services.

8. Future Enhancements

To expand the system's capabilities and make it suitable for real-world use, the following enhancements can be planned:

Suggested Future Enhancements:

1. **Online Payment Integration**
 - Connect with payment gateways (Paytm, Razorpay, UPI, etc.)
 - Secure online transactions for booking
2. **Mobile App Version**
 - Develop an Android app using Java or Kotlin for on-the-go booking
3. **Live Bus Tracking (GPS Integration)**
 - Show bus location in real time for travelers
4. **SMS and Email Notifications**
 - Send confirmation messages and reminders to users
5. **Ticket PDF Generation**
 - Generate downloadable or printable ticket PDFs using libraries like iText or Apache PDFBox

6. **Multi-language Support**
 - Localize the app for different regions and languages
7. **Admin Analytics Dashboard**
 - Show charts for route popularity, booking trends, revenue, etc.
8. **Coupon & Promo Code System**
 - Allow users to apply discount codes while booking

● *These enhancements would make the system more robust, user-friendly, and commercially viable.*

9. Conclusion

The **Bus Ticket Booking System** successfully achieves its goal of digitizing the traditional bus reservation process. It provides a convenient, error-free, and efficient way for users to book tickets while also giving administrators the tools to manage routes and monitor bookings.

With the use of **Java, MySQL, and Swing**, the system is both powerful and maintainable. While currently a desktop application, it has the potential to evolve into a full-fledged online platform or mobile app with real-time features, payment integration, and user analytics.

● *This project not only strengthens core development skills but also provides a foundation for building real-world transport solutions.*