Bus Ticket Reservation Management System

Abstract

The Bus Ticket Reservation Management System (BTRMS) is designed to digitize and streamline the traditional bus ticketing process. It addresses key challenges of manual ticketing, such as overbooking, revenue leakages, and poor customer experience, by offering real-time seat availability, secure booking and payments, and instant e-ticket generation. The system is role-based, supporting both administrators and customers, with JWT-secured authentication ensuring secure access. Built with Spring Boot, MySQL, and React, the project demonstrates scalability, modularity, and adherence to industry standards. It provides a reliable solution that improves customer satisfaction while offering operators detailed reports for effective decision-making.

1. Introduction

Transportation plays a vital role in daily life, and ticket reservation systems have become indispensable for ensuring efficiency and customer convenience. Traditional bus ticketing methods rely heavily on manual record keeping, which leads to **errors, delays, and inefficiency**.

The **BTRMS** was developed to modernize bus ticketing through a centralized digital platform. Using cutting-edge technologies such as **Spring Boot (REST APIs)**, **MySQL (database)**, and **React (frontend)**, the system ensures **real-time data synchronization**, **secure user authentication**, and **seamless booking flows**.

This project not only benefits passengers by simplifying reservations but also empowers administrators with **reports and analytics** to make better operational decisions.

2. Problem Statement

Manual ticketing causes overbooking, revenue leakages, and poor customer experience.

3. Objectives

- Real-time seat locking
- Frictionless booking & payments
- Admin control
- Security and scalability

4. Scope

Roles

- **Admin:** Manage buses, routes, trips, pricing, reports, and monitor overall operations.
- **Customer:** Search trips, select seats, make payments, download e-tickets, and cancel bookings.

Security

- **JWT authentication** for login and authorization.
- **Role-based access control** to differentiate between Admin and Customer functionalities.

5. Technology Stack

Backend (Spring Boot + MySQL)

- **Spring Boot** – To build RESTful APIs for seamless data flow.
- **Spring Security + JWT** – To secure endpoints and manage user roles.
- **Hibernate/JPA** – For ORM-based interaction with the database.
- **MySQL** – Relational database for structured data management.
- **Swagger** – For interactive API documentation.

Frontend (React.js + Bootstrap)

- **React (Hooks + Router)** – For building a responsive and dynamic user interface.
- **Axios** – For API communication.
- **Bootstrap + Custom CSS** – For consistent UI design.
- **LocalStorage/SessionStorage** – To store JWT tokens for session management.

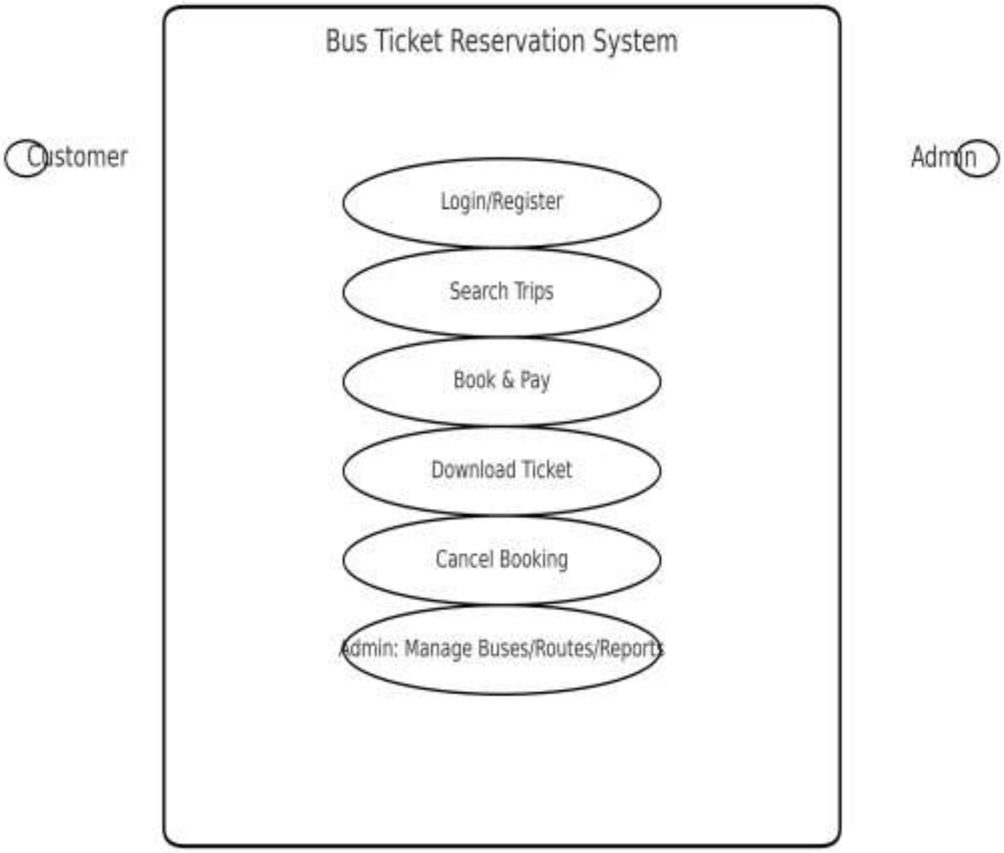
6. Core Modules

1. **Authentication & User Management** – Handles login, registration, JWT token management.

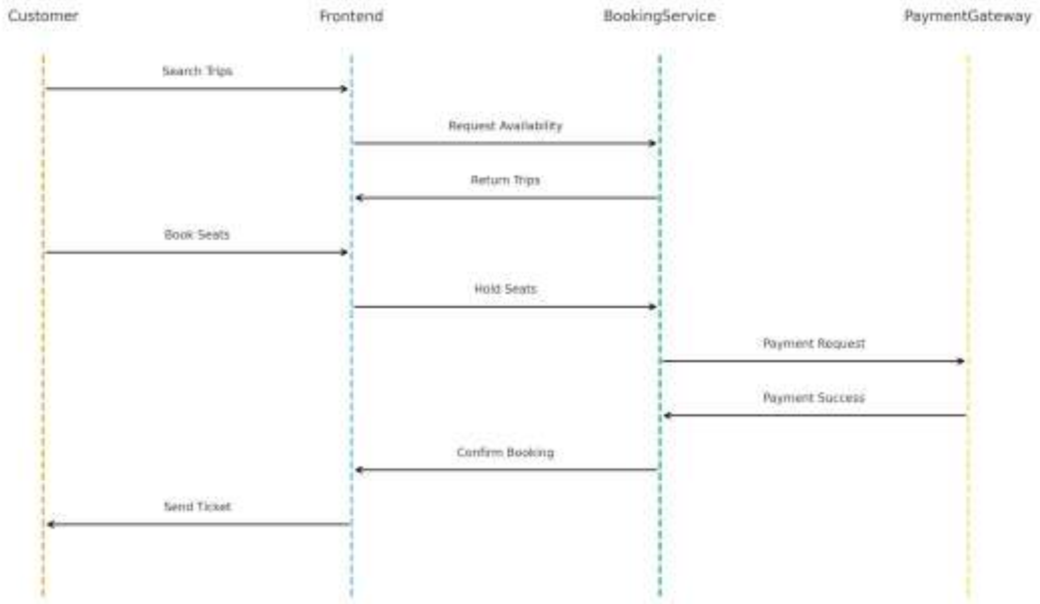
2. **Bus & Route Management** – Define buses, seat layouts, routes, and stops.
3. **Trip Scheduling & Seat Inventory** – Create trips, assign buses, set fares, and manage seat availability.
4. **Booking & Payment Processing** – Secure seat hold, booking confirmation, and payment processing.
5. **Ticketing, Cancellations & Notifications** – Generate e-tickets (PDF/QR), handle cancellations, and notify via email/SMS.
6. **Reports & Dashboards** – Admin insights on sales, occupancy, and route performance.

7. UML Diagrams

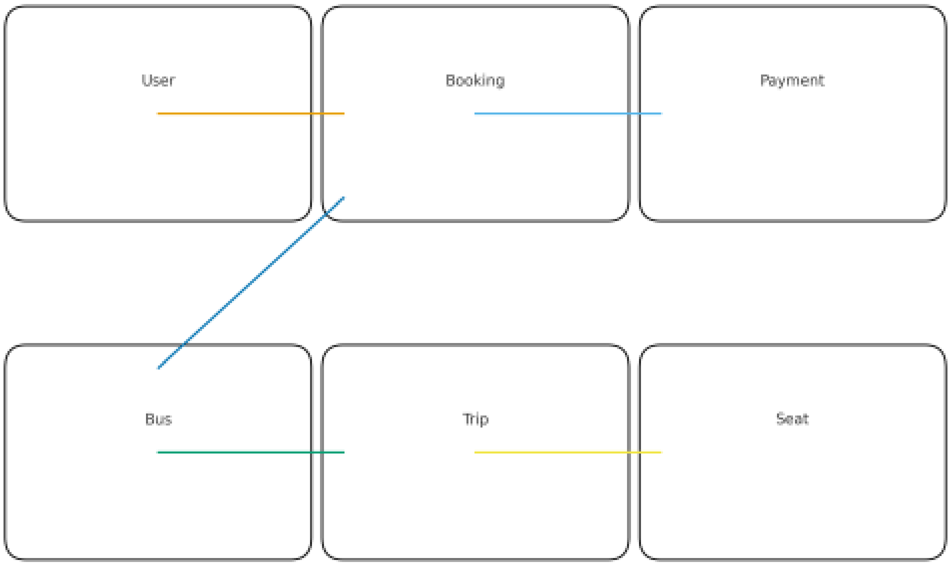
Use Case Diagram



Sequence Diagram



Class Diagram



8. Database Design

Entities: User, Bus, Route, Trip, Seat, Booking, Payment.

9. API Endpoints

Module	Endpoint	Method	Access	Description
Auth	/api/v1/auth/login	POST	Public	Login
Trips	/api/v1/trips/search	GET	All	Search trips
Booking	/api/v1/bookings/hold	POST	Customer	Hold seats
Payment	/api/v1/payments/checkout	POST	Customer	Process payment

10. Non-Functional Requirements

- **Security:** BCrypt password hashing, HTTPS, input validation, rate limiting.
- **Performance:** Seat locking within <150 ms.
- **Reliability:** Idempotent payment callbacks.
- **Auditability:** Persistent transaction and booking logs.
- **Scalability:** Modular architecture, future-ready for microservices.

11. Conclusion

The **Bus Ticket Reservation Management System** effectively modernizes the ticketing process by combining security, scalability, and convenience. Customers benefit from real-time seat availability, instant e-tickets, and easy cancellations, while administrators gain insights through detailed reports and dashboards. This project highlights the application of **Spring Boot, MySQL, React, and JWT** in developing a real-world, production-ready solution.