Online Bus Reservation System

Project Overview

The Online Bus Reservation System is a comprehensive application designed to facilitate the process of booking bus tickets and managing related services. It provides a user-friendly platform for users to reserve bus seats, provide feedback, and manage their travel schedules. Additionally, the system offers robust administrative tools for managing routes, buses, and reservations.

Project Goals

- To create a seamless and efficient bus reservation experience for users.
- To provide an easy-to-use administrative interface for managing bus services.
- To ensure data integrity and security throughout the reservation process.
- To offer features for users to provide feedback and rate their travel experience.

Key Features

1. **User Management:**
   - User registration and authentication.
   - Profile management including contact details and email.
2. **Reservation Management:**
   - Booking, updating, and canceling reservations.
   - Viewing all reservations made by a user.
   - Providing reservation details like status, type, date, time, source, and destination.
3. **Bus and Route Management:**
   - Adding, updating, and deleting buses.
   - Viewing bus details such as name, driver, type, routes, and seat availability.
   - Managing routes including origin, destination, and distance.
   - Viewing all routes and buses associated with them.
4. **Feedback Management:**
   - Allowing users to submit feedback on their travel experience.
   - Capturing ratings for drivers, service, and overall experience.
   - Viewing and managing feedback for administrative purposes.
5. **Administrative Tools:**
   - Admin login and secure management of the system.
   - Management of users, buses, routes, and reservations.
   - Ensuring all operations are tracked and logged for security and auditing.

Technologies Used

- **Backend:**
  - Java with Spring Boot for building RESTful web services.
  - Hibernate/jpa for ORM (Object-Relational Mapping) and database interaction.
  - MySQL as the relational database management system.

- **Frontend:**
  - HTML, CSS, and JavaScript for building the user interface.
  - React.js for creating dynamic and responsive user experiences.
  - Bootstrap for styling and responsive design.
- **Additional Tools:**
  - Maven for project management and dependency handling.

Modules

1. **Admin Module:**
   - Manage users, buses, routes, reservations, and feedback.
   - Secure login and session management.
2. **User Module:**
   - Registration, login, and profile management.
   - Making and managing reservations.
   - Viewing travel history and providing feedback.
3. **Reservation Module:**
   - Detailed reservation process including selection of bus, date, and time.
   - Real-time seat availability check.
   - Email notifications for reservation confirmation and updates.
4. **Bus Module:**
   - Comprehensive bus management including adding, updating, and viewing bus details.
   - Integration with routes and reservations.
5. **Route Module:**
   - Management of bus routes including adding, updating, and viewing route details.
   - Association of routes with buses.
6. **Feedback Module:**
   - Collection and management of user feedback.
   - Display of feedback to admins for quality improvement.

Example POJO Classes

Admin Class

```
public class Admin {
    private int adminId;
    private String adminUsername;
    private String adminPassword;

    // Getters and Setters
}
```

User Class

```
public class User {
    private int userLoginId;
    private String username;
    private String password;
    private String firstName;
    private String lastName;
    private String contact;
    private String email;
    private Reservation reservation; // has-a relation

    // Getters and Setters
}
```

Reservation Class

```
import java.util.Date;

public class Reservation {
    private int reservationId;
    private String reservationStatus;
    private String reservationType;
    private Date reservationDate;
    private String reservationTime;
    private String source;
    private String destination;
    private Bus bus; // has-a relation

    // Getters and Setters
}
```

Feedback Class

```
import java.util.Date;

public class Feedback {
    private int feedbackId;
    private int driverRating;
    private int serviceRating;
    private int overallRating;
    private String comments;
    private Date feedbackDate;
    private User user; // has-a relation
    private Bus bus; // has-a relation

    // Getters and Setters
}
```

Bus Class

```
public class Bus {
    private int busId;
    private String busName;
    private String driverName;
    private String busType;
    private String routeFrom;
    private String routeTo;
    private String arrivalTime;
    private String departureTime;
    private int seats;
    private int availableSeats;

    // Getters and Setters
}
```

Route Class

```
import java.util.List;

public class Route {
    private int routeId;
    private String routeFrom;
    private String routeTo;
    private double distance;
    private List<Bus> buses; // List of buses on this route

    // Getters and Setters
}
```