Student Project Task: Flight Reservation System

Student: Nathan Armand Maurice Bert (Nathan.Bert@fesb.hr)

Course: Programming in Java

Project Type: GUI-Based Application

Estimated Time: 4-8 weeks (end of May 2025)

Project Overview

Air travel is one of the most common means of transportation worldwide, and managing flight reservations efficiently is crucial for airlines and passengers. In this project, students will develop a **Flight Reservation System** using **Java**, mimicking real-world airline ticket booking platforms. The system will allow users to **search for available flights**, **book tickets**, **select or assign seats**, and **cancel reservations** while maintaining an up-to-date database of available flights and booked seats.

This project will serve as a **real-world simulation** of an airline booking system, reinforcing key programming concepts such as **Object-Oriented Programming (OOP), database interaction (JDBC with MySQL/SQLite), exception handling, multi-threading for concurrent seat bookings, and optional GUI development using Java Swing or JavaFX**. Additionally, students will practice file handling (if a database is not used), unit testing, and logging.

The **Flight Reservation System** will be designed to accommodate two types of users:

1. **Passengers** – who can search flights, book tickets, manage their reservations, and cancel bookings if needed.
2. **Admins** – who can manage flight schedules, update flight details, and oversee the reservation database.

This project encourages students to think critically about **data structures** for managing flights and reservations, **handling simultaneous user requests** (multi-threading), and implementing a **user-friendly interface**. By the end of the project, students will have built a functional airline reservation system that could be expanded with additional features such as **payment integration, email confirmations, and a mobile-friendly GUI**.

Project Objectives

By completing this project, students will:

- Learn how to **apply OOP principles** in a real-world scenario.
- Understand how to **integrate Java with databases** using JDBC.
- Work with **data structures** (lists, maps, etc.) to manage flight and booking data.
- Implement **multi-threading** to handle concurrent seat bookings.
- Build a **basic or advanced GUI** using Java Swing or JavaFX .
- Apply **exception handling** and **file handling** techniques.
- Gain experience with **unit testing** and **logging** in Java.

Application Features

The Flight Reservation System should include the following features:

1. User Management

- Users should be able to register and log in.
- Two types of users: Passengers and Admins.
- Password encryption for security.

2. Flight Search & Booking

- Users can search for flights by entering:
    - Departure location
    - Destination
    - Date of travel
- The system displays available flights, ticket prices, and available seats.
- Passengers can book a seat on a selected flight.
- A unique Booking ID should be generated for each reservation.

3. Seat Assignment

- Users can view available seats for a flight.
- Users should be able to select a specific seat during booking.
- Ensure that concurrent seat bookings are handled properly.

4. Booking Management

- Users can view their current and past bookings.
- Allow users to cancel their booking before departure.

- Seats from cancelled bookings should become available for other users.

5. Admin Panel

- Admins should be able to:
    - Add new flights with details such as flight number, route, date, time, available seats, and ticket price.
    - Update flight details (e.g., price, seat availability).
    - Delete flights if they are no longer active.
    - View all bookings.

6. Payment Processing

- Simulate a payment gateway where users enter card details.
- Validate payment before confirming the booking (simulation).

7. Logging & Exception Handling

- Implement error handling for invalid inputs and system errors.
- Use logging to keep track of system activities.

8. File Handling (If No Database)

- Store flights, users, and bookings in text files or JSON/XML.
- Implement file reading/writing operations.

9. GUI Implementation (Optional)

- Build a graphical user interface (GUI) using Java Swing or JavaFX.
- Display available seats in a seat selection UI.

Technical Requirements

1. Programming Language

- Java

2. Development Environment

- IntelliJ IDEA, Eclipse, or NetBeans
- JDK (Java Development Kit)

3. Database (Choose One)

- **MySQL** (Recommended) or **SQLite**
- JDBC for database connectivity
- SQL queries for **CRUD operations** (Create, Read, Update, Delete)

4. Libraries

- **JDBC** – Database connectivity
- **Java Swing / JavaFX** – GUI
- **JUnit** – Testing (Optional)
- **Log4j** – Logging (Optional)

5. Multi-threading

- Implement a mechanism to **handle simultaneous seat bookings** to avoid conflicts.

6. Exception Handling

- Use try-catch blocks to handle invalid user input.
- Prevent **SQL injection** and invalid queries.

Project Deliverables

Students are expected to submit the following:

1. **Source Code**
   - Well-commented Java code.
   - Proper use of OOP concepts (Encapsulation, Inheritance, Polymorphism).
2. **Database Scripts** (If applicable)
   - SQL scripts for **creating tables** and **sample flight data**.
3. **User Guide / Documentation**
   - Explanation of **features, setup instructions, and usage**.
4. **Presentation**
   - A short PowerPoint or PDF presentation summarizing the project.