Airline Reservation System - Java Console Project with MySQL Integration

Submitted By :

1] Pawar Abhinav Sujit.

2] Virkar Mayuri Balu.

1. Introduction

The airline industry has witnessed rapid technological advancements over the years, particularly in the field of reservation systems. An Airline Reservation System (ARS) is a computerized system used to store and retrieve information and conduct transactions related to air travel. This project, titled **"Airline Reservation System"**, aims to design and develop a console-based Java application that interacts with a MySQL database to provide essential functionalities such as user login, flight management, and ticket booking.

The objective of this system is to automate the process of booking airline tickets and managing flight-related information, thereby reducing manual effort and the chances of human error. By leveraging the power of Java for application logic and MySQL for data management, this system ensures secure, efficient, and reliable operation. It supports operations like adding new flights, viewing existing flights, booking tickets, and maintaining passenger records.

In the current digital era, such systems are crucial for improving customer satisfaction, ensuring quick service, and maintaining organized data records. This project offers a hands-on understanding of integrating front-end Java programming with back-end database operations and provides a foundational understanding of how real-world airline reservation systems operate.

The project follows a modular approach to ensure scalability and maintainability. Each function is implemented as a dedicated module which interacts with the database using Java Database Connectivity (JDBC). The system also handles basic error management and validations to ensure smooth user experience.

2. Objective

The primary objective of this **Airline Reservation System** project is to develop a simple yet functional console-based application that allows users to interact with a centralized database to perform core airline-related operations such as managing flights, booking tickets, and viewing reservation details. This system is intended to provide a seamless and efficient solution to the traditional, manual method of handling airline bookings and record-keeping.

The specific objectives of this project include:

1. **Automation of Reservation Process**: To reduce the manual effort involved in booking airline tickets by implementing a user-friendly interface that can perform all necessary operations through simple menu-driven interactions.
2. **Centralized Data Management**: To integrate a MySQL database with the Java application to store, retrieve, and manage flight and booking information in an organized and secure manner.
3. **User Authentication**: To include a login module that allows only authorized users to access the system, thus ensuring the security and confidentiality of the data.
4. **Flight Information Management**: To provide administrators the ability to add, update, and view flight details including flight ID, source, destination, date, time, and available seats.
5. **Ticket Booking Functionality**: To allow users to search for flights and book tickets by entering relevant passenger information and storing the same into the database.
6. **Real-Time Updates**: To ensure that the system reflects real-time updates in seat availability once a booking is made.
7. **Error Handling and Validation**: To handle user input errors and database exceptions gracefully, providing a stable and reliable user experience.
8. **Learning Integration**: To provide an educational perspective on combining Java programming with relational database concepts, applying JDBC (Java Database Connectivity) effectively in a real-world use case.

Overall, this project aims to serve as a learning tool that demonstrates the core principles of object-oriented programming, structured query language, and user-centric system design in a practical scenario.

3. Software Requirements

The successful development and execution of the **Airline Reservation System** require a well-defined set of software tools and platforms. The software requirements for both the development and deployment environments are listed below:

1. Operating System

- **Windows 10/11** or any other OS compatible with Java and MySQL.
- Linux and macOS are also supported with proper configuration.

2. Programming Language

- **Java (JDK 8 or above)**
  Used for developing the core logic and handling user interaction via the console.

3. Integrated Development Environment (IDE)

- **Visual Studio Code**, **Eclipse**, or **IntelliJ IDEA**
  Any modern IDE that supports Java development and extensions for managing database connections.

4. Database Management System (DBMS)

- **MySQL Server (version 5.7 or above)**
  Used to store flight data, user credentials, and booking details in a structured format.

5. JDBC Driver

- **MySQL Connector/J**
  A JDBC driver to enable communication between the Java application and the MySQL database.

6. Document Tools

- **Microsoft Word** or **Google Docs**
  For documentation and report preparation.

7. Optional Tools

- **XAMPP** (if using phpMyAdmin for visual DB management)
- **DB Browser for MySQL** for easier database management (GUI-based)

8. Terminal/Command Line Interface

- Windows PowerShell or Command Prompt for compiling and running Java classes using `javac` and `java` commands.

4. System Architecture

The Airline Reservation System follows a **three-tier architecture** consisting of:

1. **Presentation Layer –** This is the console-based user interface that interacts with the user. It takes input and displays output such as flight details, booking status, etc.
2. **Business Logic Layer –** This layer handles all the core processing, such as login validation, booking, flight management, and ticket operations. It processes data and acts as a bridge between the UI and the database.
3. **Data Access Layer –** This layer interacts with the MySQL database using JDBC. It performs all database operations like inserting, updating, deleting, and fetching data.

This modular structure ensures easy development, testing, and future scalability of the system.

5. Database Design

The Airline Reservation System uses a **MySQL** database to manage users, flights, bookings, and passengers. The design follows a **relational model** to ensure data consistency and integrity.

Main Tables:

- **Users**: Stores login details (`user_id`, `username`, `password`).
- **Flights**: Holds flight information (`flight_id`, `flight_name`, `source`, `destination`, `price`).
- **Bookings**: Tracks booking records (`booking_id`, `user_id`, `flight_id`, `booking_date`).
- **Passengers**: Contains passenger details per booking (`passenger_id`, `booking_id`, `name`, `age`).

Relationships are maintained using **primary** and **foreign keys**. The structure is normalized to avoid redundancy and maintain efficient data retrieval.

6. Key Features
- Login system to verify user credentials
- View all available flights
- Add new flights (admin only)
- Book tickets and store booking details
- Handle invalid inputs and display relevant messages

7. Implementation Details
The application is written entirely in Java and uses JDBC to connect to the MySQL database. The application prompts users through the console for login, flight viewing, and booking. Based on the user's role (admin or user), different operations can be performed.

8. Code Overview
1. `ConnectionManager.java`: Handles database connections.
2. `MainApp.java`: Contains the main method and user interface logic.
3. `Flight.java`: Handles flight-related operations (add, view).
4. `Booking.java`: Handles ticket bookings.
5. `User.java`: Manages login and credential verification.

9. Testing and Validation

The system has been tested using sample data and multiple test cases such as:
- Valid and invalid logins
- Adding flights
- Booking available and unavailable seats
- Viewing flights and bookings

All functionalities worked as expected during testing.

10. Limitations

- Console-based interface is less intuitive compared to GUI applications.
- No encryption for stored passwords.
- No support for ticket cancellations or modifications.

11. Future Enhancements

- Add GUI using JavaFX or Swing.
- Implement role-based access for admin and passengers.
- Add support for email confirmations and reports.
- Add payment gateway integration.

12. Conclusion

The Airline Reservation System project successfully demonstrates the integration of Java with a MySQL database to create a reliable and efficient console-based application. Through its modular design and database connectivity, the system simplifies key operations such as managing flights, booking tickets, and storing passenger details.

This project not only showcases essential concepts of **Object-Oriented Programming (OOP)** and **JDBC (Java Database Connectivity)** but also emphasizes real-world software development practices. The system is scalable and can be enhanced in the future with features like a graphical user interface (GUI), payment gateways, or online check-in.

Overall, this project strengthens understanding of Java programming, database design, and the logic behind managing real-time applications in the travel domain.