AIRLINE MANAGEMENT SYSTEM

Project Report

Submitted by: Muhammad Yousif

AIRLINE MANAGEMENT SYSTEM

Project Report

Submitted by:

Muhammad Yousif

Table of Contents

---

1. Introduction

The Airline Management System is a comprehensive Java-based application designed to streamline airline operations and enhance passenger service. The system manages flight bookings, cancellations, passenger information, and administrator operations using an intuitive and modern graphical user interface.

This project is built using core Object-Oriented Programming (OOP) concepts such as encapsulation, inheritance, abstraction, and polymorphism. The application features a modern GUI with gradient backgrounds, customized UI components, and secure database connectivity for real-time data operations.

---

2. Overview of the System

The system provides a unified platform that supports passenger management, flight operations, ticket handling, and administrator controls. Its modular architecture ensures easy maintenance, scalability, and improved performance.

---

3. Key Features

1. User Authentication System
2. Passenger Management
3. Flight Booking System
4. Flight Management
5. Ticket Management
6. Administrative Utilities

---

4. Core Classes and Their Responsibilities

1. HomePage

Main dashboard and navigation controller.

2. Login

Handles user authentication and session verification.

3. ViewPassenger

Displays and manages passenger data.

4. BookFlight

Handles flight reservation and ticket generation.

5. UpdatePassenger

Allows modification of existing passenger details.

6. FlightZone

Manages all flight-related data.

7. CancelFlight

Handles ticket cancellation processes.

8. ViewBookedFlight

Displays booking histories and ticket statuses.

9. ViewCanceledTicket

Shows records of all cancelled tickets.

10. ConnectionClass

Manages secure database connectivity through JDBC.

---

5. Detailed System Modules

1. User Authentication System

The system verifies user credentials through the database using a secure username-password mechanism. Custom-styled UI components and gradient designs improve user experience and ensure professional interface quality.

---

2. Passenger Management

The module allows the addition, modification, viewing, and searching of passenger data. Passenger records are displayed in structured tables for quick access and readability.

---

3. Flight Booking System

This core module allows users to:

Check real-time flight availability

Select travel classes

View dynamic pricing

Generate unique ticket IDs

The booking is securely stored in the database and linked to passengers.

---

4. Flight Management

Manages:

Flight numbers

Routes

Schedules

Capacity

Status (On-time / Delayed)

Ensures accurate and efficient flight operations management.

---

5. Ticket Management

Handles:

Viewing booked tickets

Managing cancellations

Tracking cancellation reasons

Updating ticket statuses automatically

Enables easy tracking for both admins and passengers.

---

6. Administrative Features

Includes:

Flight zone management

Payment verification

Journey detail tracking

Reports generation

These features support decision-making and airline performance analysis.

---

6. Libraries and Packages Used

Java Swing & AWT (GUI Development)

Used to design modern, responsive UI components.

Packages:

`javax.swing.`

`java.awt.`

`java.awt.event.`

JDBC (Database Connectivity)

Used to connect Java with MySQL and perform CRUD operations.

Classes Used:

Connection

DriverManager

Statement

PreparedStatement

ResultSet

External Library:

`mysql-connector-j.jar`

Java Utility Packages

Used for collections, dates, and formatting.

Packages:

`java.util.`

`java.text.`

Custom Helper Classes

These enhance the GUI design and reusability:

`GradientPanel`

`ShadowLabel`

`PillButton`

`RoundButton`

---

7. Database Design

The system uses a relational database with interlinked tables:

passenger – stores passenger details

flight – contains schedule & route data

booking – records reservations

cancel – saves cancellation data

signup – stores user accounts

This structure ensures referential integrity and efficient data flow.

---

8. GUI Implementation

The GUI is built with:

Gradient backgrounds

Modern button styles

Shadow-based labels

Responsive layouts

Consistent color schemes inspired by airline themes

These elements create a clean, professional interface.

---

9. Technologies Used

Java (OOP)

Swing / AWT (GUI)

MySQL (Database)

JDBC (Connectivity)

NetBeans / IntelliJ IDEA

Git (Version Control)

---

10. Project Directory Structure

```
AMS/
■■■ HomePage.java
■■■ Login.java
■■■ ViewPassenger.java
■■■ BookFlight.java
■■■ UpdatePassenger.java
■■■ FlightZone.java
■■■ CancelFlight.java
■■■ ViewBookedFlight.java
■■■ ViewCanceledTicket.java
■■■ ConnectionClass.java
■
■■■ Helper Classes/
■■■ GradientPanel.java
■■■ ShadowLabel.java
■■■ PillButton.java
■■■ RoundButton.java
```

---

11. Learning Outcomes

Through this project we learned:

Practical implementation of OOP

GUI design with Swing

Real-time database interaction

Exception handling

Input validation

Software modularization

Team collaboration using Git

Event-driven programming

---

12. Challenges Faced

Managing multi-window GUI consistency

Establishing efficient JDBC connectivity

Validating large data inputs

Designing maintainable code architecture

Debugging SQL and UI synchronization

Handling multiple events simultaneously

---

13. Future Enhancements

Possible future improvements include:

Online payment integration

Email notification system

Mobile app development

Multi-language support

Cloud deployment

Advanced reporting & analytics

---

14. Conclusion

The Airline Management System demonstrates the effective use of Object-Oriented Programming concepts to build a complete, real-world airline automation platform. The system provides reliable flight booking, passenger handling, and administrative features through a modern, user-friendly GUI.

This project strengthened our skills in Java programming, GUI design, JDBC, teamwork, and problem-solving—forming a solid foundation for future professional development.