
Title: AIRLINE TRAFFIC MANAGEMENT SYSTEM

---

1. Problem Statement

The problem is well defined: managing flights, bookings, and passengers through a user-friendly interface, with a backend MySQL database to store and manage data. The system ensures seat availability and proper updates to the database when bookings are made or canceled.

---

2. Dataset Description

This project uses a relational database (e.g., MySQL, SQLite) to store flight and booking details. The table structure for storing this information includes the following fields:

1.ADD FLIGHTS:

Column Name	Data Type	Description
flight_number	VARCHAR(10)	Unique identifier for each flight.
origin	VARCHAR(50)	Starting location of the flight.
destination	VARCHAR(50)	Destination location of the flight.
available_seats	INT	Number of seats available for booking on the flight

2.BOOKINGS TABLE:

Column Name	Data Type	Description
booking_id	INT (AUTO_INCREMENT)	A unique identifier for each booking.
origin	VARCHAR(50)	The name of the passenger making the booking..
destination	VARCHAR(10)	The flight number associated with the booking.

Each field represents a specific attribute of a flight or booking, enabling efficient and organized data storage for retrieval, management, and updates in the Airline Traffic Management System.

---

3. Language and Concepts Used

Programming Language: Java

Database: SQL (structured as a relational database)

Key Concepts:

- **Classes and Objects:** For managing flights and bookings.
- **Encapsulation:** Protecting database credentials and methods.
- **Abstraction:** Hiding complex database operations behind simple methods.
- **Database Connectivity:** Using JDBC to interact with the MySQL database.
- **Exception Handling:** Managing errors like invalid input or database issues.
- **User Input Validation:** Ensuring valid and consistent input from users.

---

4. Code

```
import java.sql.*;
import java.util.Scanner;

public class AirlineTrafficManagementSystem {
   // Database credentials and URL
   private static final String URL = "jdbc:mysql://localhost:3306/AirlineDB5";
   private static final String USER = "root";       // Replace with your MySQL username
   private static final String PASSWORD = "kovarthini@2005"; // Replace with your MySQL password

   // Establish database connection
   private Connection connect() throws SQLException {
      return DriverManager.getConnection(URL, USER, PASSWORD);
   }

   // Add a new flight
   public void addFlight(String flightNumber, String origin, String destination, int availableSeats) {
      String query = "INSERT INTO flights4 (flight_number, origin, destination, available_seats) VALUES (?, ?, ?, ?)";
      try (Connection conn = connect(); PreparedStatement stmt = conn.prepareStatement(query)) {
```

```
            stmt.setString(1, flightNumber);
            stmt.setString(2, origin);
            stmt.setString(3, destination);
            stmt.setInt(4, availableSeats);
            stmt.executeUpdate();
            System.out.println("Flight added successfully.");
        } catch (SQLException e) {
            System.err.println("Error adding flight: " + e.getMessage());
        }
    }

    // View all flights
    public void viewFlights() {
        String query = "SELECT * FROM flights4";
        try (Connection conn = connect(); PreparedStatement stmt =
conn.prepareStatement(query)) {
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                System.out.println("Flight Number: " + rs.getString("flight_number"));
                System.out.println("Origin: " + rs.getString("origin"));
                System.out.println("Destination: " + rs.getString("destination"));
                System.out.println("Available Seats: " + rs.getInt("available_seats"));
                System.out.println("----------------------------");
            }
        } catch (SQLException e) {
            System.err.println("Error viewing flights: " + e.getMessage());
        }
    }

    // Book a ticket for a passenger
    public void bookTicket(String passengerName, String flightNumber) {
        String selectQuery = "SELECT available_seats FROM flights4 WHERE flight_number = ?";
        String updateQuery = "UPDATE flights4 SET available_seats = available_seats - 1 WHERE
flight_number = ?";
        String insertQuery = "INSERT INTO bookings4 (passenger_name, flight_number) VALUES (?,
?)";

        try (Connection conn = connect();
             PreparedStatement selectStmt = conn.prepareStatement(selectQuery);
             PreparedStatement updateStmt = conn.prepareStatement(updateQuery);
             PreparedStatement insertStmt = conn.prepareStatement(insertQuery)) {

            // Check available seats
            selectStmt.setString(1, flightNumber);
            ResultSet rs = selectStmt.executeQuery();
            if (rs.next()) {
                int availableSeats = rs.getInt("available_seats");
                if (availableSeats > 0) {
                    // Update seat count
                    updateStmt.setString(1, flightNumber);
                    updateStmt.executeUpdate();

                    // Insert booking
                    insertStmt.setString(1, passengerName);
```

```
                insertStmt.setString(2, flightNumber);
                insertStmt.executeUpdate();

                System.out.println("Booking successful!");
            } else {
                System.out.println("No available seats on this flight.");
            }
        } else {
            System.out.println("Flight not found.");
        }
    } catch (SQLException e) {
        System.err.println("Error booking ticket: " + e.getMessage());
    }
}

// View all bookings
public void viewBookings() {
    String query = "SELECT * FROM bookings4";
    try (Connection conn = connect(); PreparedStatement stmt =
conn.prepareStatement(query)) {
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            System.out.println("Booking ID: " + rs.getInt("booking_id"));
            System.out.println("Passenger Name: " + rs.getString("passenger_name"));
            System.out.println("Flight Number: " + rs.getString("flight_number"));
            System.out.println("----------------------------");
        }
    } catch (SQLException e) {
        System.err.println("Error viewing bookings: " + e.getMessage());
    }
}

// Main method
public static void main(String[] args) {
    AirlineTrafficManagementSystem system = new AirlineTrafficManagementSystem();
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("\n--- Airline Traffic Management System ---");
        System.out.println("1. Add Flight");
        System.out.println("2. View Flights");
        System.out.println("3. Book Ticket");
        System.out.println("4. View Bookings");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");

        int choice;
        while (true) {
            if (scanner.hasNextInt()) {
                choice = scanner.nextInt();
                scanner.nextLine(); // Consume newline
                break;
            } else {
                System.out.println("Invalid input. Please enter a valid number.");
```

```
                scanner.nextLine(); // Clear invalid input
            }
        }

        switch (choice) {
            case 1:
                System.out.print("Enter flight number: ");
                String flightNumber = scanner.nextLine();
                System.out.print("Enter origin: ");
                String origin = scanner.nextLine();
                System.out.print("Enter destination: ");
                String destination = scanner.nextLine();
                System.out.print("Enter available seats: ");
                int seats = scanner.nextInt();
                system.addFlight(flightNumber, origin, destination, seats);
                break;

            case 2:
                system.viewFlights();
                break;

            case 3:
                System.out.print("Enter passenger name: ");
                String passengerName = scanner.nextLine();
                System.out.print("Enter flight number: ");
                String bookFlightNumber = scanner.nextLine();
                system.bookTicket(passengerName, bookFlightNumber);
                break;

            case 4:
                system.viewBookings();
                break;

            case 5:
                System.out.println("Exiting... Goodbye!");
                scanner.close();
                return;

            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
  }
}
```

---

5. Result

The **Airline Traffic Management System** successfully performs the following:

- **Add Flight**: Allows users to add new flight details to the database, including flight number, origin, destination, and available seats.

- **View Flights**: Displays a list of all available flights with flight numbers, origin, destination, and available seats.
- **Book Ticket**: Enables passengers to book tickets on flights, updates the available seats, and records the booking in the system.
- **View Bookings**: Displays all bookings made, including booking IDs, passenger names, and the respective flight numbers.

Example Output:

Airline Traffic Management Menu:
1. Add Flight
2. View Flights
3. Book Ticket
4. View Bookings
5. Exit Choose an option:

---

6. Conclusion

This project showcases how to create a simple yet effective airline traffic management system using Java and MySQL. By incorporating key programming concepts such as object-oriented design, JDBC for database communication, and input validation, the system successfully manages flights and bookings. It also serves as a foundation for further development into a more sophisticated airline management system, adding features such as cancellation, scheduling, and improved error handling.