Airline management System

Name: Yash Labade

Roll No.: 92

# Title

Airline management System

This document outlines the implementation of an Airline Management System using Hibernate ORM in Java. It includes the setup, configuration, and execution of core airline functionalities like flight registration, passenger management, booking flights, canceling bookings, and viewing transaction history.

**Objective**

To develop an airline management system using Hibernate to demonstrate the use of object-relational mapping for database operations.

**Tools and Technologies Used**

1. Java
2. Hibernate ORM
3. MySQL Database
4. Maven
5. Eclipse/IntelliJ IDEA

**System Requirements**

1. JDK 8 or above
2. MySQL Server
3. Hibernate Core Library
4. Maven for dependency management

**System Design**

The system includes the following major components:

* Flight Entity
* Passenger Entity
* BookingTransaction Entity
* Hibernate DAO Layer
* Service Layer
* Main Application Layer

**Hibernate Configuration**

xml

CopyEdit

<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/airlinedb</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">password</property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>

<property name="show\_sql">true</property>

<property name="hbm2ddl.auto">update</property>

<mapping class="model.Flight"/>

<mapping class="model.Passenger"/>

<mapping class="model.BookingTransaction"/>

</session-factory>

</hibernate-configuration>

**Entity Classes**

**Flight.java**

java

CopyEdit

@Entity

public class Flight {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String flightNumber;

private String destination;

private int capacity;

}

**Passenger.java**

java

CopyEdit

@Entity

public class Passenger {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String email;

}

**BookingTransaction.java**

java

CopyEdit

@Entity

public class BookingTransaction {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String status; // Booked / Canceled

private Date date;

@ManyToOne

private Flight flight;

@ManyToOne

private Passenger passenger;

}

**DAO and Service Layer**

Same structure as the BankService and HibernateUtil classes, adjusted for Flight, Passenger, and BookingTransaction handling.

**Main Application Class**

**Main.java**

java

CopyEdit

public class Main {

public static void main(String[] args) {

AirlineService service = new AirlineService();

// Add a passenger

Passenger passenger = new Passenger();

passenger.setName("John Doe");

passenger.setEmail("john@example.com");

service.registerPassenger(passenger);

// Add a flight

Flight flight = new Flight();

flight.setFlightNumber("AI101");

flight.setDestination("New York");

flight.setCapacity(200);

service.addFlight(flight);

// Book and cancel flight

service.bookFlight(flight.getId(), passenger.getId());

service.cancelBooking(flight.getId(), passenger.getId());

}

}