

AIRLINE MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted By

TR DIVYA SREE 230701083

JAYA BHARATHI M 230701126

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024

BONAFIDE CERTIFICATE

Certified that this project report “**AIRLINE MANAGEMENT
SYSTEM**” is the bonafide work of “**TR DIVYA SREE (230701083),
JAYA BHARATHI M (230701126)**”

who carried out the project work under my supervision.

Submitted for the practical examination held on _____.

SIGNATURE

Mrs.Divya.M

Assistant Professor,

Computer Science and Engineering,

Rajalakshmi Engineering College,

Thandalam,Chennai 602105

INTERNAL EXAMINER

SIGNATURE

Mr.RAGHU.G

Assistant Professor

Computer Science and Engineering

Rajalakshmi Engineering College,

Thandalam,Chennai 602 105.

EXTERNAL EXAMINER

ABSTRACT:

This project on Airline Management System is the automation of registration process of airlines system. The system provides information like passenger's information, flight information, list of all passengers, it allows storing and retrieving data related to the airline industry and make transactions related to air travel etc. The system also allows us to add records when a passenger reserves a ticket. For data storage and retrieval we use MySQL Database. It enables us to add any number of records in our database. The project "Airline Management System" comprises of a large number of flights which belong to a particular airline. The system we have implemented manages different objects viz.

- Airline
- Airline Employees
- Customers/Traveller

Each of these accesses a database schema which has corresponding tables.

Language Used - Java Core

Concept Used - Swing

IDE Used - NetBeans

Database Used – MySQL

TABLE OF CONTENTS

Chapter 1

1 INTRODUCTION	-----6
1.1 INTRODUCTION	-----6
1.2 OBJECTIVES	-----7
1.3 MODULES	-----7

Chapter 2

2 SURVEY OF TECHNOLOGIES	-----9
2.1 SOFTWARE DESCRIPTION	-----9
2.2 LANGUAGES	-----9
2.2.1 JAVA(CORE JAVA WITH SWING)	-----10
2.2.2 MYSQL	-----10
2.2.3 JDBC	-----11
2.2.4 HTML	-----12

Chapter 3

3 REQUIREMENTS AND ANALYSIS	-----12
3.1 REQUIREMENT SPECIFICATION	-----12
3.1.1 FUNCTIONAL REQUIREMENTS	-----13
3.1.2 NON FUNCTIONAL REQUIREMENTS	-----13
3.2 HARDWARE AND SOFTWARE REQUIREMENTS	-----14
3.3 ARCHITECTURE DIAGRAM	-----15
3.4 ER DIAGRAM	-----16
3.5 NORMALIZATION	-----17

Chapter 4

4 PROGRAM CODE-----

4.1 PROGRAM CODE-----18

Chapter 5

5 RESULTS AND DISCUSSION-----

5.1 RESULTS AND DISCUSSION-----27

Chapter 6

6 CONCLUSION-----

6.1 CONCLUSION-----31

Chapter 7

7 REFERENCES-----

7.1 REFERENCES-----32

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

Airline Management System is the administration of airports and airlines. It includes the activities of setting the strategy of airports to gather and provide information on airline commercial and operational priorities. It covers a broad overview of the airline management. It is also studied as a branch of study that teaches management of airport and airlines. This provides a broad overview of the airline industry and creates awareness of the underlying marketing, financial, operational, and other factors influencing airline management. This study provides information on airline commercial and operational priorities, along with teaching the key characteristics of aircraft selection and the impact of airport decision making. It provides some amount of automation in airlines management and helps airline system in making their business more efficient. An added attraction for their potential customers. It will also show the attitude of the management that they are aware to the newly introduced technology and ready to adopt them.

1.1 Problem Definition

This project on Flight Management System is the automation of registration process of airline system. The system is able to provide much information like passenger's details, flight details and the booking details. The system allows us to add records when a passenger reserves a ticket. It also allows to delete and update the records based on passenger's requirements. For data storage and retrieval we use the MySQL database. It enables us to add any number of records in our database from the frontend which is Java core.

Any changes made in the frontend will be reflected at the backend.

1.2 Need

Electronically handling of flight's record to enhance the accuracy, flexibility, reliability and to remove the human's error. An airline provides air transport services for passengers, generally with a recognize operating. To provide accurate information about the addition, deletion and modified record. To provide, efficient, accurate, reliable, fast, and robust structure that can handle any number of records. The global airline industry continues to grow rapidly, but consistent and robust profitability is elusive. Measured by revenue, the industry has doubled over the past decade, from US\$369 billion in 2004 to a projected \$746 billion in 2014, according to the International Air Transport Association(IATA).Much of that growth has been driven by low-cost carriers(LCCs), which now control some 25 percent of the worldwide market and which have been expanding rapidly in emerging markets; growth also came from continued gains by carriers in developed markets, the IATA reported. Yet profit margins are still low, less than 3 percent overall. In the commercial aviation sector, just about every group in the aviation industry chain—airports, airplane manufacturers, jet engine makers, travel agents, and service companies, to name a few—turns a profit. It is seemingly ironic that the airline companies that actually move passengers from one place to another, the most crucial link in the chain, struggle to make a profit.

A few factors that directs us to develop a new system are given below -:

1) Faster System

2) Accuracy.

1.2 OBJECTIVES

1. Automate Airline Registration Process:

The system automates the traditionally manual registration process, such as booking tickets, storing passenger details, and managing flight information. This reduces time consumption and increases efficiency.

2. Data Management:

Using a relational database (MySQL), the system securely stores critical airline data, including passenger information, flight schedules, and booking records. This ensures data can be accessed, updated, and maintained systematically.

3. Efficient Record Handling:

The system provides functions to add, delete, and update records, allowing administrators to handle customer and booking data dynamically as requirements change.

4. Improved System Accuracy:

Automating tasks like passenger information entry and booking management reduces human error, ensuring accurate and reliable data processing.

5. Enhanced User Interaction:

A graphical user interface (GUI) designed using Java Swing enables easy interaction for both administrators and customers. Features include intuitive menus for flight booking, cancellation, and flight details.

6. Business Scalability:

The system is designed to handle a growing volume of data (passenger and flight records), making it scalable for expanding airline operations.

7. Modernization and Technological Adoption:

By adopting modern tools like Java Swing and MySQL, the system reflects the airline's readiness to embrace new technologies, appealing to tech-savvy users.

1.3 MODULES :

1. Login Module:

Purpose: Ensures secure access to the system by verifying user credentials.

Implementation:

Utilizes a login table in the database to store usernames and passwords.

Java Swing is used to design the login form. Incorrect credentials result in error messages.

2. Customer Management Module:

Purpose: Allows adding, updating, and managing customer details.

Key Features:

Capture customer details like name, address, nationality, gender, phone number, and passport details. Store these records in the passenger table within the database.

Benefits: Streamlines passenger registration, making it efficient and error-free.

3. Flight Management Module:

Purpose: Maintains flight-related information such as flight code, flight name, source, and destination.

Key Features:

Provides a database table called flight to store flight details. Displays flight information dynamically using GUI tables.

Benefits: Keeps administrators updated about available flights and their schedules.

4. Booking Module:

Purpose: Enables customers to book flights and generate tickets.

Workflow: Customers select the source, destination, and date of travel.

The system fetches flight options from the database. Generates a ticket with a unique Passenger Name Record (PNR) and ticket ID.

Key Database Tables: reservation, flight, passenger.

Benefits: Simplifies booking processes while storing relevant details for future reference.

5. Cancellation Module:

Purpose: Manages the cancellation of tickets.

Key Features: Updates the database to reflect canceled bookings. Handles refunds or changes based on cancellation requests.

Key Database Table: cancellation.

6. Database Connectivity Module:

Purpose: Facilitates seamless interaction between the GUI and the MySQL database.

Implementation:

Utilizes JDBC (Java Database Connectivity) to connect the frontend and backend. Ensures real-time updates in the database when changes are made in the GUI.

Benefits: Maintains consistency between the user interface and backend data.

7. Frontend GUI Module:

Purpose: Provides an easy-to-use interface for users to interact with the system.

Key Features:

Login Page: Secure entry for users.

Home Page: Central navigation for all functionalities like flight booking, cancellations, and customer details.

Other GUIs: Designed for viewing flight details, adding customer information, and booking tickets.

Technology Used: Java Swing, which supports a visually appealing and responsive interface.

8. Reports and Snapshots:

Purpose: Provides a visual representation of the database records, flight details, and user activity.

Implementation:

Use of graphical tables and scrollable panels to display data fetched from the database.

Benefits: Simplifies administrative tasks by displaying summarized information in an organized format.

Chapter 2

SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION

The Airline Management System is a comprehensive software solution designed to automate the key operations of an airline, including flight management, ticket booking, passenger management, and cancellations. This system leverages modern programming languages, databases, and technologies to ensure efficiency, accuracy, and scalability for airline operations. Below is a detailed description of the software components and their roles:

2.2 LANGUAGES

1. Java (Core Java with Swing)

Role:

Acts as the primary programming language for developing the application.

Powers the frontend interface and business logic of the system.

Usage:

GUI Development:

Used Swing to design user-friendly interfaces (login forms, booking forms, dropdowns, tables).

Logic Implementation:

Handled business logic for operations like booking tickets, updating customer records, and handling cancellations.

Event Handling:

Managed user interactions (e.g., button clicks, menu selections).

Integration with MySQL:

JDBC is used to connect Java with the database for executing SQL queries.

Advantages:**Platform Independence:**

Runs on any operating system with a Java Virtual Machine (JVM).

Rich GUI Features:

Swing provides advanced components like menus, buttons, and tables for an interactive user experience.

Object-Oriented:

Promotes modular development and code reusability.

Robust and Secure:

Handles errors gracefully and ensures data security during operations.

2. MySQL

Role:

Serves as the database management system for storing, retrieving, and managing airline data.

Usage:**Data Storage:**

Maintains tables for passengers, flights, reservations, cancellations, and more.

Query Execution:

Processes SQL queries to add, delete, and modify records.

Data Relationships:

Uses relational database principles to establish connections between different data entities (e.g., passenger and reservation).

Data Integrity:

Ensures the consistency and reliability of data through constraints.

Advantages:

Scalability:

Handles large datasets efficiently, making it suitable for growing airline operations.

Open Source:

Freely available, reducing development costs.

Performance:

Optimized for speed and performance in data retrieval and manipulation.

Compatibility:

Integrates seamlessly with Java via JDBC.

Security:

Provides robust mechanisms for data protection and user access control.

3. Java Database Connectivity (JDBC)

Role:

Acts as a middleware to connect Java applications with the MySQL database.

Usage:

Database Connectivity:

Establishes a connection between the frontend and backend.

Query Execution:

Enables Java to execute SQL queries dynamically, such as fetching flight details or updating booking records.

Result Handling:

Processes and displays the results of database queries within the application.

Advantages:

Flexibility:

Works with any relational database, not just MySQL.

Ease of Use:

Simplifies database operations in Java applications.

Real-Time Updates:

Reflects changes made in the database instantly in the application.

Lightweight:

Adds minimal overhead to the application.

4. HTML (Optional for Additional Interfaces)

Role:

Could be used for designing web-based interfaces for the system if needed.

Usage:

Static Pages:

Create web-based views for flight details, booking confirmations, etc.

Integration:

Can be combined with Java applets for browser-based interaction.

Advantages:

Simplicity:

Easy to learn and implement for creating structured interfaces.

Compatibility:

Works across all web browsers.

Low Resource Requirements:

Lightweight and does not require advanced hardware.

Chapter 3 REQUIREMENTS AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION

3.1.1 Functional Requirements

Functional requirements specify the features and operations the system must perform. These requirements directly address what the system should do.

Core Functionalities:

1. User Authentication:

Provide a secure login system to verify user credentials.

Allow administrators and staff to access authorized functionalities.

2. Customer Management:

Add, update, and delete customer information such as name, nationality, address, and contact details.

Store customer data in a relational database for easy retrieval.

3. Flight Management:

Allow administrators to manage flight details, including:

Flight code.

Flight name.

Source and destination locations.

Departure and arrival times.

4. Ticket Booking:

Enable customers or administrators to:

Select a flight based on source, destination, and date.

Reserve a ticket and generate a unique PNR (Passenger Name Record).

Store booking details in the database.

5. Ticket Cancellation:

Allow users to cancel tickets by providing PNR or ticket ID.

Update the database to reflect the cancellation.

6. Database Management:

Store and organize data in tables for flights, customers, bookings, and cancellations.

Maintain relationships between tables for consistency.

7. Search and Retrieval:

Provide options to search:

Flights based on source, destination, and date.

Customer details based on contact information or passport number.

8. Payment Management:

Record payment details like card number, transaction ID, and paid amount.

Ensure payments are securely processed and stored.

9. Report Generation:

Generate reports or views:

Passenger details for a flight.

Flight schedule.

Booking and cancellation history.

3.1.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system. These ensure the system meets performance, usability, reliability, and other criteria.

Performance Requirements:

1. Scalability:

The system should handle a large number of records as the airline grows.

Support multiple users simultaneously accessing the system.

2. Efficiency:

Responses to user actions (e.g., login, booking search) should occur within a few seconds. Database queries should be optimized for speed.

Usability Requirements:

3. Ease of Use:

Provide a user-friendly interface with clear navigation and intuitive design.

Use consistent GUI elements for a seamless experience.

4. Accessibility:

Ensure the system is accessible to non-technical users through a graphical interface.

Security Requirements:

5. Data Security:

Use authentication mechanisms to prevent unauthorized access. Encrypt sensitive data like passwords and payment information.

6. Data Integrity:

Ensure that data updates in the database are consistent and error-free.

HARDWARE AND SOFTWARE REQUIREMENTS:

1. Hardware Requirements

The hardware requirements specify the physical components needed to develop and run the system effectively.

Minimum Hardware Specifications:

1. Processor:

Intel Core i3 or equivalent (2.3 GHz or higher).

Recommended: Core i5 or higher for faster performance.

2. Memory (RAM):

Minimum: 2 GB.

Recommended: 4 GB or more for smooth multitasking.

3. Storage:

Minimum: 400 GB hard disk.

Recommended: SSD storage for better read/write speeds.

4. Monitor:

Minimum resolution: 1366x768.

Recommended: 1920x1080 for better GUI design experience.

5. Input Devices:

Keyboard and mouse.

6. Network:

Internet connection for downloading required tools and libraries.

2. Software Requirements

The software requirements define the necessary tools, operating systems, and platforms for the system.

Development Environment:

1. Operating System:

Minimum: Windows 10 (64-bit) or higher.

Optional: Linux or macOS (as Java and MySQL are cross-platform).

2. Programming Tools:

Java Development Kit (JDK):

Version: JDK 8 or higher.

IDE:

NetBeans 8.2 (recommended for Swing-based projects).

3. Database:

MySQL:

Version: 8.0 or higher.

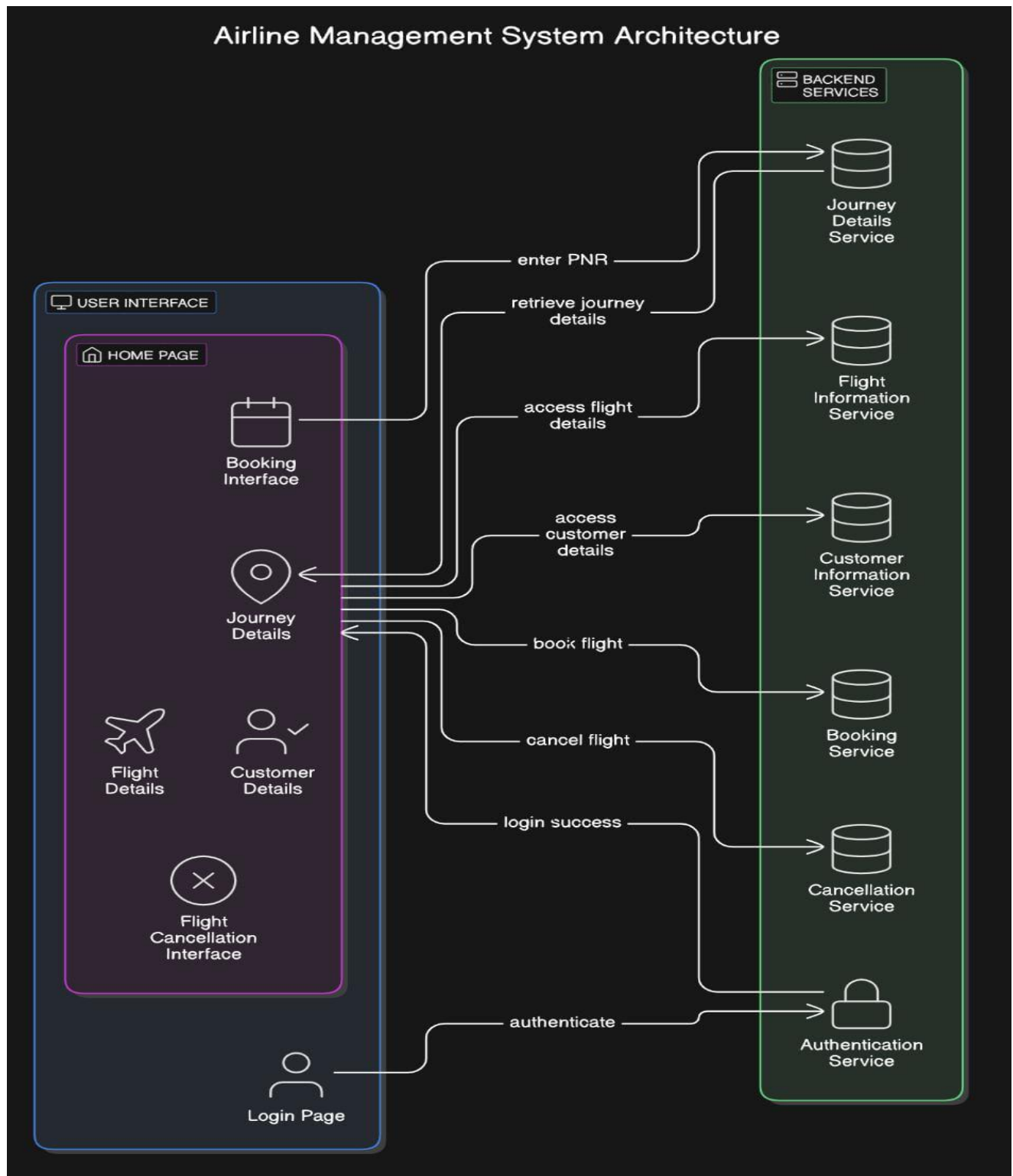
Used to store airline data such as passenger, flight, booking, and cancellation details.

4. Database Connectivity:

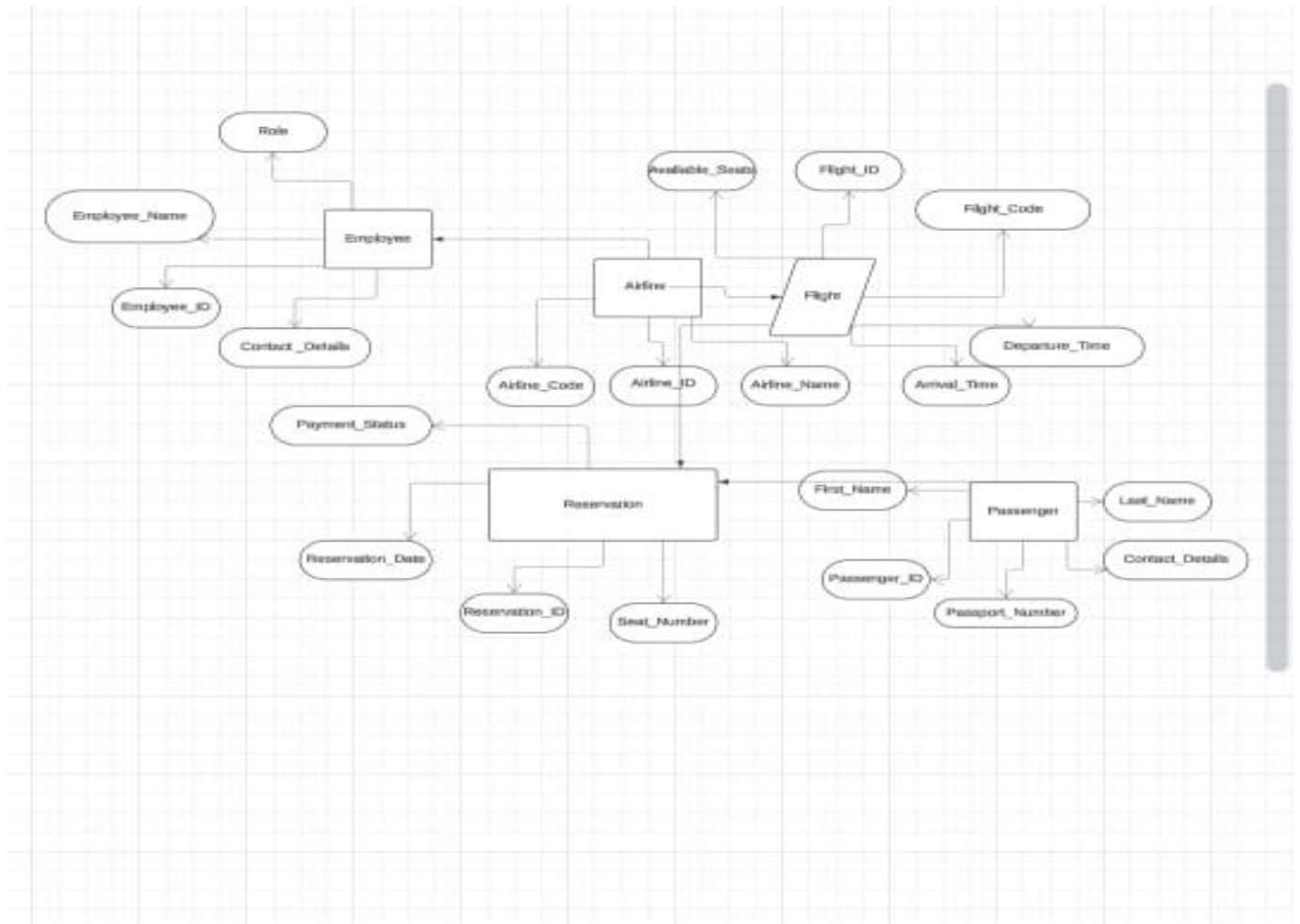
JDBC Driver:

Required for connecting Java with the MySQL database.

3.3 ARCHITECTURE DIAGRAM



ENTITY RELATIONSHIP DIAGRAM



3.5 NORMALISATION

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing a database into two or more tables and defining relationships between the tables. The steps to normalize a database table for Blood Bank Management System are as follows.

Raw Database

Attribute	Datatype	Example
Booking_ID	VARCHAR(10)	B001
Passenger_Name	VARCHAR(50)	JOHNDOE
Flight_Code	VARCHAR(10)	F101
Source	VARCHAR(50)	NEWYORK
Destination	VARCHAR(50)	LONDON
Passenger_Phone	VARCHAR(15)	8907654321
Pass_Address	VARCHAR(100)	123 Main st
Ticket_ID	VARCHAR(10)	123345
Flight_Date	DATE	25-11-2025
Flight_Time	TIME	10.30 AM

FIRST NORMAL FORM(1NF):

In the above raw data:

The Flight_Name column has a composite value (e.g., "New York to London"), which we should split into Source and Destination.

The Flight_Date and Flight_Time should be atomic, so no further splitting is needed there.

1NF Table (with atomic values):

Now, the table satisfies 1NF since all values are atomic and there are no repeating groups.

SECOND NORMAL FORM (2NF)

Definition: Achieve 1NF and eliminate partial dependencies, where a non-prime attribute is dependent on only part of a composite primary key.

In the above 1NF table:

The Booking_ID can be considered as the primary key for the booking. However, Passenger_Name, Passenger_Phone, and Passenger_Address depend only on Booking_ID, not on Flight_Code.

The Flight_Code, Flight_Date, Flight_Time, Source, and Destination depend on Flight 2NF _Code, not Booking_ID.

To achieve 2NF, we need to split the data into separate tables to remove partial dependencies:

Decomposing into 2NF Tables:

Table 1: Bookings (Booking Details)

Table 2: Flights (Flight Details)

Chapter 4 **PROGRAM CODE**

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home extends JFrame implements ActionListener{

    public Home() {
        setLayout(null);

        ImageIcon i1 = new
        ImageIcon(ClassLoader.getResource("airlinemanagements
        ystem/icons/file.jpg"));

        JLabel image = new JLabel(i1);
        image.setBounds(0, 0, 1600, 800);
        add(image);

        JLabel heading = new JLabel(" EMIRATES WELCOMES YOU ");
        heading.setBounds(500, 40, 1500, 40);
        heading.setForeground(Color.BLACK);
```

```
heading.setFont(new Font("Tahoma", Font.PLAIN, 36));  
image.add(heading);
```

```
JMenuBar menubar = new JMenuBar();  
setJMenuBar(menubar);
```

```
JMenu details = new JMenu("Details");  
menubar.add(details);
```

```
JMenuItem flightDetails = new JMenuItem("Flight Details");  
flightDetails.addActionListener(this);  
details.add(flightDetails);
```

```
JMenuItem customerDetails = new JMenuItem("Add Customer  
Details");  
customerDetails.addActionListener(this);  
details.add(customerDetails);
```

```
JMenuItem bookFlight = new JMenuItem("Book Flight");  
bookFlight.addActionListener(this);  
details.add(bookFlight);
```

```
JMenuItem journeyDetails = new JMenuItem("Journey Details");  
journeyDetails.addActionListener(this);
```

```
details.add(journeyDetails);
```

```
JMenuItem ticketCancellation = new JMenuItem("Cancel  
Ticket");
```

```
ticketCancellation.addActionListener(this);
```

```
details.add(ticketCancellation);
```

```
JMenu ticket = new JMenu("Ticket");
```

```
menubar.add(ticket);
```

```
JMenuItem boardingPass = new JMenuItem("Boarding Pass");
```

```
ticket.add(boardingPass);
```

```
setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    String text = ae.getActionCommand();
```

```
    if (text.equals("Add Customer Details")) {
```

```
        new AddCustomer();
```

```
    } else if (text.equals("Flight Details")) {
```

```
        new FlightInfo();
    } else if (text.equals("Book Flight")) {
        new BookFlight();
    } else if (text.equals("Journey Details")) {
        new JourneyDetails();
    } else if (text.equals("Cancel Ticket")) {
        new Cancel();
    }
}

public static void main(String[] args) {
    new Home();
}
}
```

Chapter 5

RESULTS AND DISCUSSION



RESERVATION HOME PAGE

f_code	f_name	source	destination
9696	AI-1252	Bangalore	Mumbai
1616	AI-1455	Chennai	Goa
3131	AI-1872	Mumbai	Kashmir
2626	AI-3222	Hyderabad	Amritsar
1111	AI-1212	kerala	Delhi
2727	AI-1243	Trichy	Mumbai
8080	AI-1434	Mysore	Goa
1212	AI-1342	Mumbai	Chennai
5050	AI-3283	Uttar Pradesh	Amritsar
3030	AI-3434	Madurai	Ayodhya
1212	AI-1342	kashmir	Chennai

FLIGHT DETAILS

ADD CUSTOMER DETAILS

Name

Nationality

Aadhar Number

Address


Gender

☐ Male

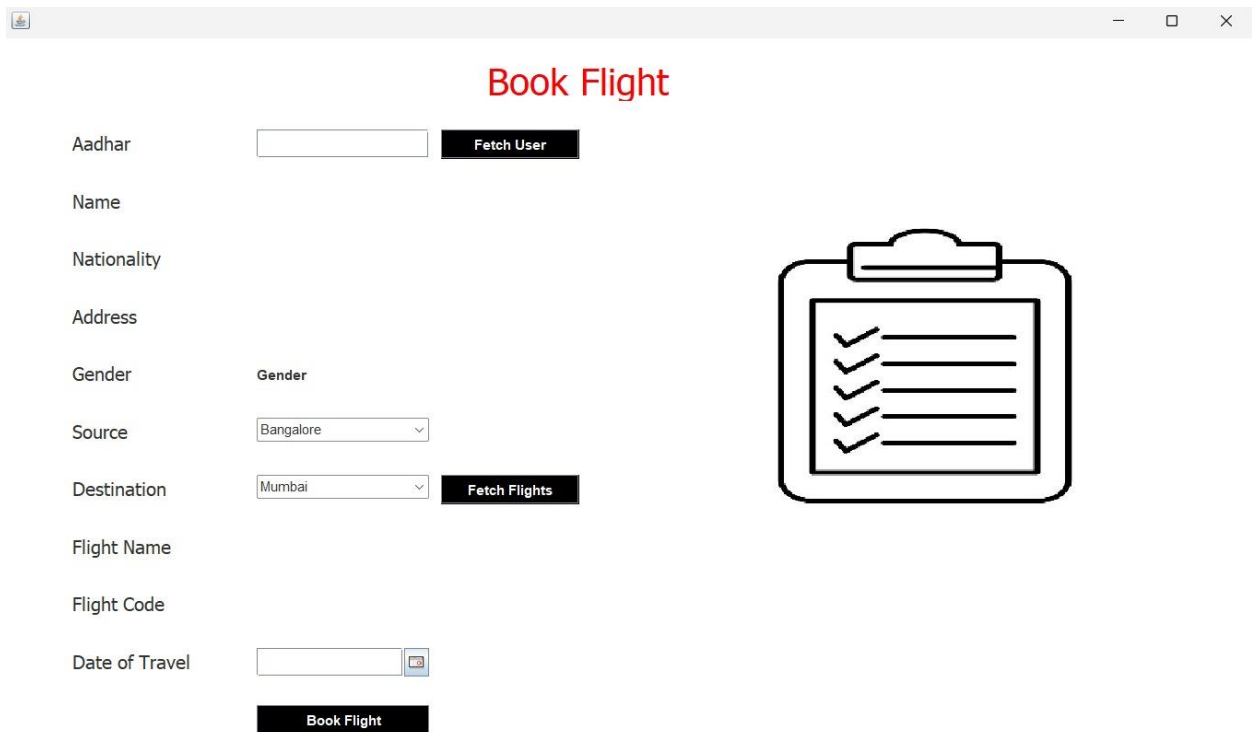
☐ Female

Phone

SAVE



ADD CUSTOMER DETAILS

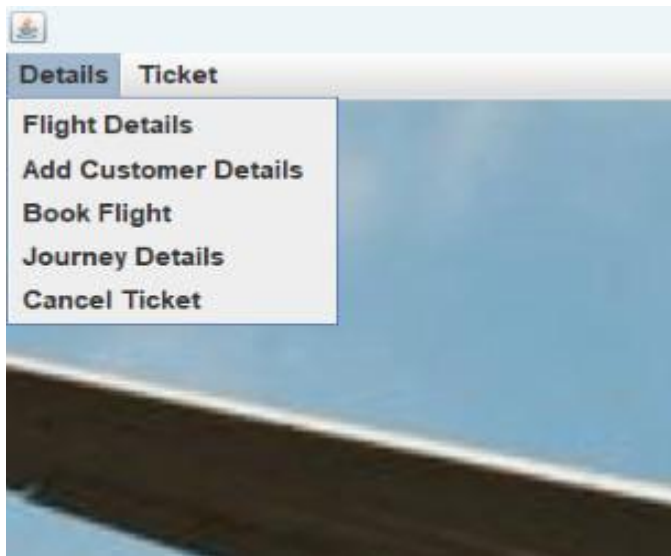


The screenshot shows a web application window titled "Book Flight". The form contains the following fields and controls:

- Aadhar:** A text input field followed by a "Fetch User" button.
- Name:** A text input field.
- Nationality:** A text input field.
- Address:** A text input field.
- Gender:** A text input field, with a "Gender" label and a dropdown menu below it.
- Source:** A dropdown menu with "Bangalore" selected.
- Destination:** A dropdown menu with "Mumbai" selected, followed by a "Fetch Flights" button.
- Flight Name:** A text input field.
- Flight Code:** A text input field.
- Date of Travel:** A date input field with a calendar icon, followed by a "Book Flight" button.

To the right of the form is a large icon of a clipboard with a checklist, indicating a list of items to be checked or completed.

FLIGHT BOOKING



DROP_DOWN MENU BAR

— □ ×

CANCELLATION

PNR Number

Show Details

Name

Cancellation No

349797

Flight Code

Date

Cancel

CANCELLATION

BACKEND(DATABASE)

Result Grid						
		Filter Rows:		Export:	Wrap Cell Content:	
	name	nationality	phone	address	aadhar	gender
▶	Regina	Indian	7299002599	Anna nagar ,chennai	123456789876	Female
	kirthiga	indian	8610963654	nehru nagar ,chennai	123456786456	Female
	divyasree	indian	8122713694	Thirumanagalam ,Chennai	345678865467	Female
	thirumurugan	indian	9677143694	Anna nagar ,chennai	123456784568	Male
	yaswanth	indian	73672863838	nehru nagar ,chennai	234567894673	Male
	dinisha	indian	4578964356	red hills ,chennai	5467389476803	Female
	harsha	indian	637835478	mayiladuthurai	23456783756	Female
	himeshwar	indian	6846859937	tambaram	45367836374	Male
	ganesh	indian	456786466	mylapore	42637946579	Male
	jyoshna	indian	4568753457	arakkonam	123454353456	Female
	pooja	indian	3456765435	nehru nagar ,chennai	46283648675845	Female

	f_code	f_name	source	destination
▶	9696	AI-1252	Bangalore	Mumbai
	1616	AI-1455	Chennai	Goa
	3131	AI-1872	Mumbai	Kashmir
	2626	AI-3222	Hyderabad	Amritsar
	1111	AI-1212	kerala	Delhi
	2727	AI-1243	Trichy	Mumbai
	8080	AI-1434	Mysore	Goa
	1212	AI-1342	Mumbai	Chennai
	5050	AI-3283	Uttar Pradesh	Amritsar
	3030	AI-3434	Madurai	Ayodhya
	1212	AI-1342	kashmir	Chennai

	PNR	TICKET	aadhar	name	nationality	flightname	flightcode	src	des	ddate
▶	PNR-181006	TIC-6671	123456789876	Regina	Indian	AI-1872	3131	Mumbai	Kashmir	13 Nov 2024
	PNR-243743	TIC-5367	46283648675845	pooja	indian	AI-1212	1111	kerala	Delhi	20 Nov 2024

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	username	password
▶	regina	regina

CONCLUSION

The Airline Management System is an innovative and efficient solution designed to streamline airline operations and enhance user experience. By automating core functionalities like ticket booking, cancellations, and data management, the system ensures accuracy, reliability, and scalability. With a user-friendly interface built using Java Swing, robust data management through MySQL, and seamless integration via JDBC, the system eliminates redundancy, reduces human error, and simplifies complex airline workflows.

This project not only highlights the importance of modern technology in enhancing operational efficiency but also demonstrates the potential for further innovation, such as integrating online payment gateways, real-time flight tracking, and customer loyalty programs. It serves as a foundational system that can be scaled and extended to meet the dynamic needs of the airline industry.

The implementation of this system not only benefits airline administrators by providing centralized data management but also improves customer satisfaction by offering faster and more accurate services. This project exemplifies the power of combining robust programming techniques with effective database management to create a seamless, reliable, and forward-thinking application.

REFERENCES

1. Programming Language:

Oracle. Java Documentation.

<https://docs.oracle.com/javase/>

2. Database Management:

MySQL. MySQL Reference Manual.

<https://dev.mysql.com/doc/>

3. Development Environment:

Apache NetBeans. NetBeans IDE Documentation.

<https://netbeans.apache.org/>

4. Normalization Concepts:

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks.

Communications of the ACM.

5. GUI Development:

Oracle. Java Swing API Documentation.

<https://docs.oracle.com/javase/8/docs/technotes/guides/swing/>

6. Database Connectivity:

Oracle. Java Database Connectivity (JDBC) Guide.

<https://docs.oracle.com/javase/tutorial/jdbc/>

7. Project Inspiration:

Airline Industry Trends and Challenges:

International Air Transport Association (IATA). Reports and Insights.

<https://www.iata.org/>