AIRLINE RESERVATION SYSTEM

A PROJECT REPORT

Submitted by

HARIVARSHAN M (2303811710421058)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"AIRLINE RESERVATION SYSTEM"** is the bonafide work of **HARIVARSHAN M (2303811710421058)** who carriedout the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Mr. M. Saravanan, M.E.,

SUPERVISOR

ASSISTANTP ROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02.12.2024

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"AIRLINE RESERVATION SYSTEM"** is the result of original work done by me and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201- JAVA PROGRAMMING.**

Signature

HARIVARSHAN M

Place: Samayapuram

Date: 02.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide **Mr. M. SARAVANAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Airline Reservation System is a comprehensive software application designed to manage and streamline airline ticket booking processes. It enables users to search for available flights seats, book tickets, manage reservations, and process cancellations efficiently. The system incorporates key features such as real-time seat availability, fare management, and secure payment processing. It supports both customer and administrative functionalities, including flight scheduling, passenger record management, and reporting. By automating manual operations, the system reduces errors, enhances operational efficiency, and improves customer satisfaction. Its adaptability allows integration with global distribution systems (GDS) and other travel-related platforms, making it an essential tool for modern airline operations.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Airline Reservation System is a comprehensive software application designed to manage and streamline airline ticket booking processes. It enables users to search for available flights, book tickets, manage reservations, and process cancellations efficiently. The system incorporates key features such as real-time seat availability, fare management, and secure payment processing. It supports both customer and administrative functionalities, including flight scheduling, passenger record management, and reporting. By automating manual operations, the system reduces errors, enhances operational efficiency, and improves customer satisfaction. Its adaptability allows integration with global distribution systems (GDS) and other travel-related platforms, making it an essential tool for modern airline operations.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of our airline reservation system (ARS) is to facilitate the efficient management of flight bookings while enhancing the customer experience during the ticket purchasing process. At its core, the ARS aims to provide a streamlined and user-friendly platform for passengers to search for flights, check availability, and make reservations with ease. By centralizing critical data such as flight schedules, fare tariffs, and passenger name records (PNRs), the system enables airlines to manage their inventory effectively across various distribution channels, including direct sales and global distribution systems (GDS) used by travel agencies. Additionally, the automation of the reservation process minimizes human errors associated with manual systems, ensuring accuracy in data entry and retrieval. The ARS also focuses on improving customer service by offering timely information about flights and allowing passengers to manage their bookings independently, with features like seat selection and real-time updates.

1.2 Overview

The Airline Reservation System (ARS) project using Java is designed to streamline the process of booking tickets for both domestic and international flights. This system aims to provide a user-friendly interface where customers can easily search for flights, check seat availability, and make reservations online, thus eliminating the need for time-consuming manual processes typically associated with traditional booking methods.The project primarily focuses on two key modules: Domestic Flights and International Flights. Users can input their travel requirements, such as flight name, source, destination, and number of seats. The system then checks the backend database for available flights and provides users with detailed information about their options, including fare details and discounts. Upon confirming a reservation, the system generates a Passenger Name Record (PNR) and allows users to print their tickets.

1.3 Java Programming Concepts

The basic concepts of Object-Oriented Programming (OOP) are:

- ✓ **Class and Object:** A class is a blueprint, and an object is an instance of the class.
- ✓ **Encapsulation:** Bundles data and methods into a single unit (class) while restricting direct access to data.
- ✓ **Inheritance:** Enables a class (child) to inherit properties and methods from another class (parent), promoting code reuse.
- ✓ **Polymorphism:** Allows methods to perform differently based on the object context (e.g., method overloading and overriding).
- ✓ **Abstraction:** Hides implementation details and exposes only essential features, simplifying system design.

Project related concepts

1. Classes and Objects

- ✓ Classes: Airline Reservation System implemented in Java, key classes such as Flight, Passenger, Reservation, searching,and displaying encapsulate essential functionalities and data related to flight bookings and management
- ✓ Objects: Airline Reservation System (ARS), several key objects are essential for managing the various functionalities of the system.

2. Encapsulation

- Each class encapsulates its data and methods, providing a clear interface for interacting with other parts of the system while hiding internal implementation details.

3. Methods

- ✓ Methods AirlineReservationSystem() and bookSeat() will allow the user to enter a seat number, passenger name, and phone number, and methods like displayAvailableSeats() will Displays a message dialog with the available seats.and cancelReservation() Prompts the user to enter a booking ID to cancel.

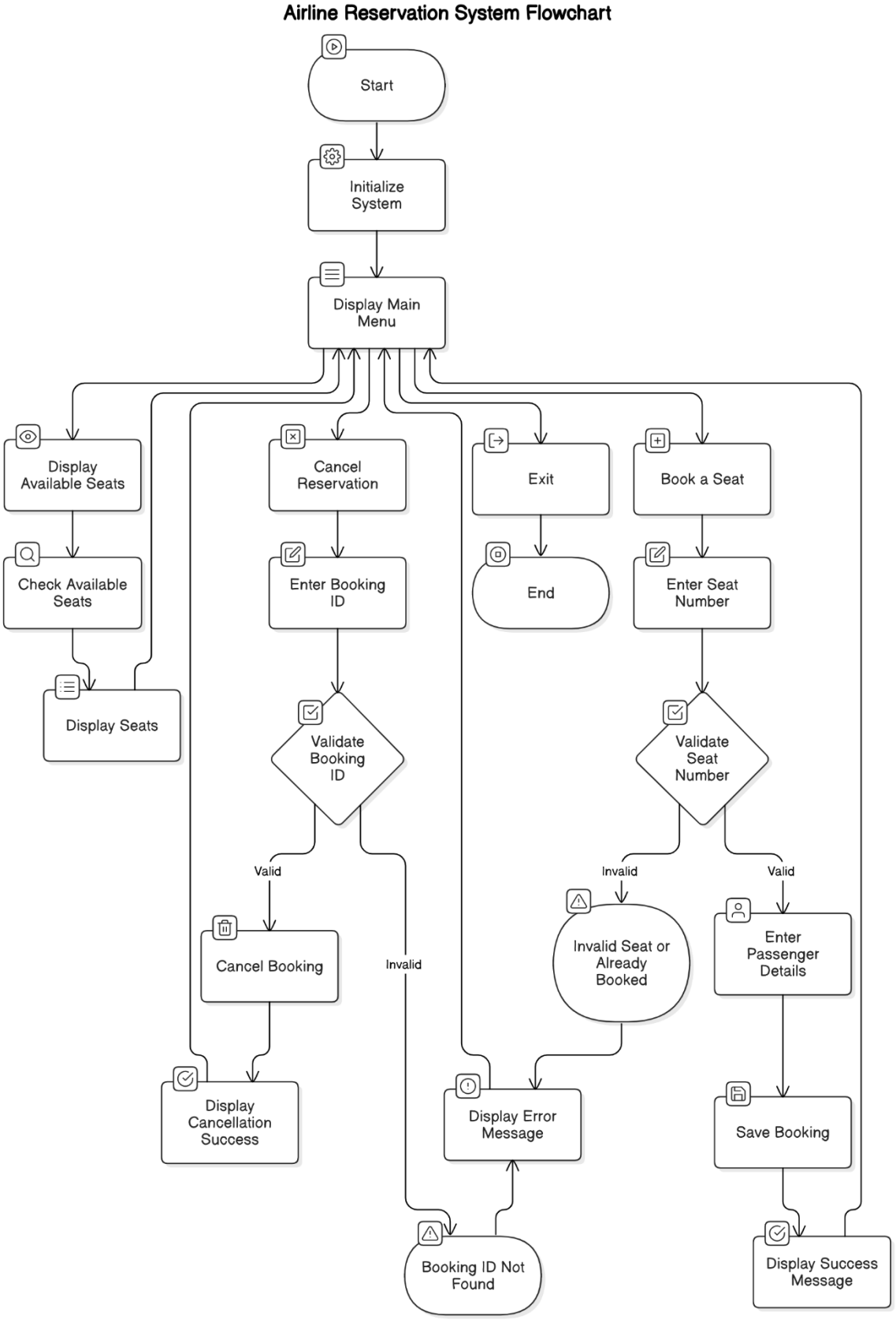
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed Airline Reservation System is a desktop application designed to simplify the process of booking airline seats through an intuitive graphical user interface (GUI) built using Java Swing. The system allows users to easily book seats by entering their desired seat number, passenger name, and phone number, while ensuring that the seat is valid and not already booked. Users can also view available seats at any time, helping them make informed decisions about their reservations. Additionally, the system provides functionality for canceling bookings by entering a unique booking ID, with confirmation messages displayed for both successful bookings and cancellations. The application maintains an in-memory database of bookings using a HashMap to store booking details, including a dynamically incrementing booking ID for each reservation. Key components of the system include the main class that sets up the GUI and handles user interactions, as well as a Booking class that encapsulates individual booking details. Overall, this Airline Reservation System offers a straightforward solution for managing flight bookings while providing opportunities for future enhancements, such as integrating persistent storage or adding features like user authentication and flight searches.

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

3.1 Booking Module:

The booking module in the provided Airline Reservation System code is responsible for handling the booking of seats, including user input, validation, and storage of booking information.

Key Methods :

- bookSeat()-allows us to book flight tickets
- cancelReservation()-allows us to cancel flight tickets

3.2 User Interface Module

This module is responsible for creating and managing the graphical user interface (GUI) of the application using Java Swing components.

Components :

- JFrame-The main window of the application.
- JButton-Buttons for various actions (e.g., booking a seat, displaying available seats)

3.3 Data Management Module

This module manages the storage and retrieval of booking information. It uses a data structure to keep track of seat bookings.

Data Structures:

- Map<Integer, Booking>: A HashMap that maps seat numbers
- Booking objects. This allows efficient lookups and management of bookings.

3.4 Main Application Module

This module contains the main method that serves as the entry point of the application.

- Functionality:
  - Initializes and displays the main GUI window.
  - Uses SwingUtilities.invokeLater() to ensure that GUI creation is done on the Event Dispatch Thread (EDT) for thread safety.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

In conclusion, the Airline Reservation System implemented in Java using Swing provides a robust and user-friendly solution for managing flight bookings. The application effectively demonstrates key programming concepts such as object-oriented design, event-driven programming, and data management through its modular architecture.Key features of the system include the ability to book seats, display available seats, and cancel reservations, all facilitated by an intuitive graphical user interface. The use of a HashMap for storing booking information ensures efficient access and management of seat availability, while the dynamic assignment of booking IDs enhances tracking and organization.Overall, this project serves as an excellent foundation for understanding the principles of software development in Java, offering opportunities for further enhancements such as integrating database support for persistent storage, adding user authentication, or expanding functionalities to include flight searches and payment processing. The modular design not only makes the code easy to maintain but also allows for scalability as new features are added in future iterations. This system exemplifies how technology can streamline processes in the travel industry, improving both operational efficiency for airlines and convenience for passengers.

4.2 FUTURE SCOPE

The The Airline Reservation System implemented in Java using Swing has several promising avenues for future development and enhancement. One significant improvement would be the integration of a relational or NoSQL database to enable persistent storage of booking information, allowing users to retrieve their reservations even after the application is closed. Additionally, implementing user authentication and profiles would enhance security and provide passengers with the ability to manage their bookings more effectively, including updating personal information and accessing booking history. Enhancing the user interface by adopting a more modern design, possibly through JavaFX, could improve the overall user experience with features like tooltips and progress indicators during booking.Future developments could also include advanced flight search functionalities that allow users to filter flights based on various criteria, as well as detailed flight information such as departure times and baggage policies.

REFERENCES

Java Books:

1."Head First Java" by Kathy Sierra and Bert Bates

This book is a great resource for beginners learning Java, with a focus on object-oriented programming concepts and real-world application development.

2. "Effective Java" by Joshua Bloch

A deeper dive into best practices for writing clean, maintainable Java code. It covers advanced topics like Java collections, concurrency, and design patterns that could be applied to more complex payroll systems.

Websites:

1. javatpoint - Java Tutorials

- URL: https://www.javatpoint.com/
- A comprehensive collection of tutorials on Java, covering topics like classes, objects, inheritance, encapsulation, and more. Great for learning the core concepts of Java and applying them in projects like EPMS.

2. W3Schools - Java Tutorial

- URL: https://www.w3schools.com/java/
- A beginner-friendly resource that offers tutorials on Java programming, including object-oriented principles and core Java concepts.

YouTube Links:

1. Java for Beginners – Error makes clever

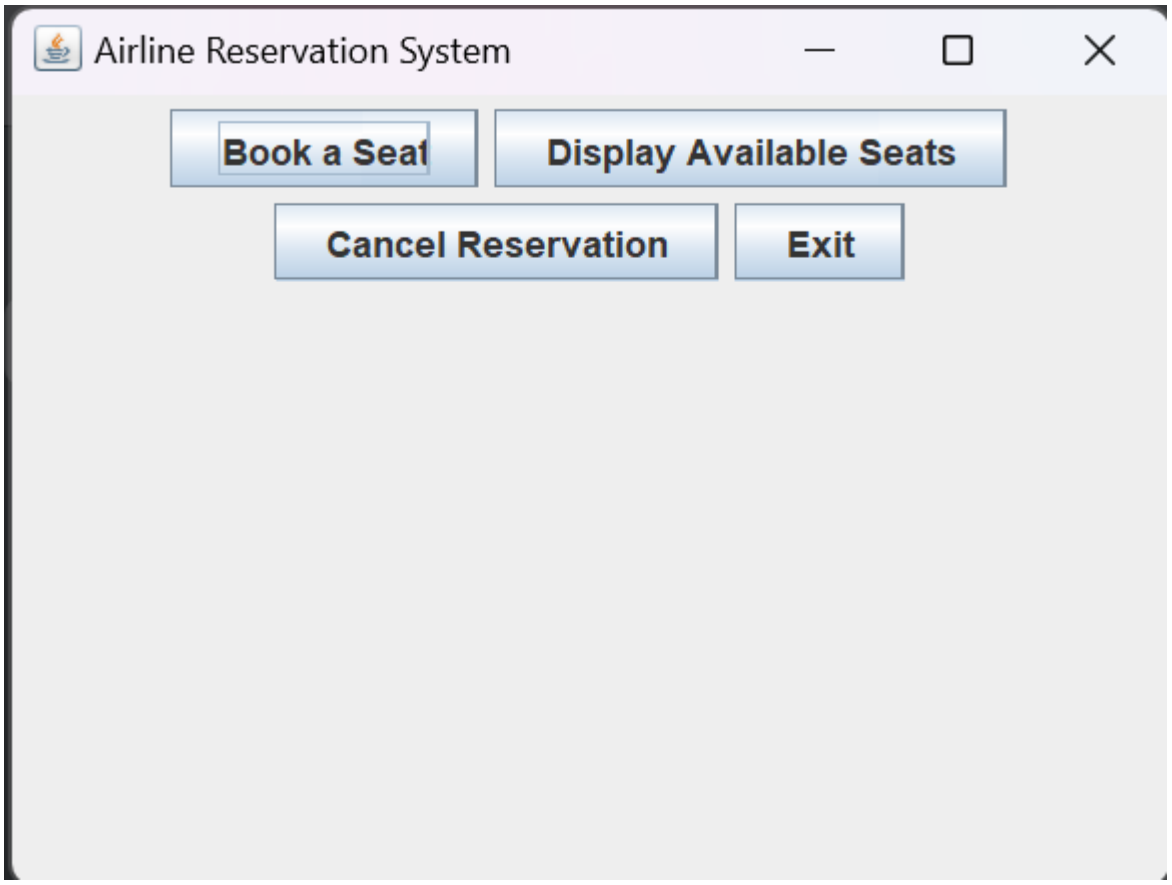
- URL: https://youtu.be/IT2durkDCXM?feature=shared
- Offers Java tutorials from the basics to advanced concepts. The channel provides detailed guides on Java programming, including working with objects and classes, which are crucial for building an EPMS.

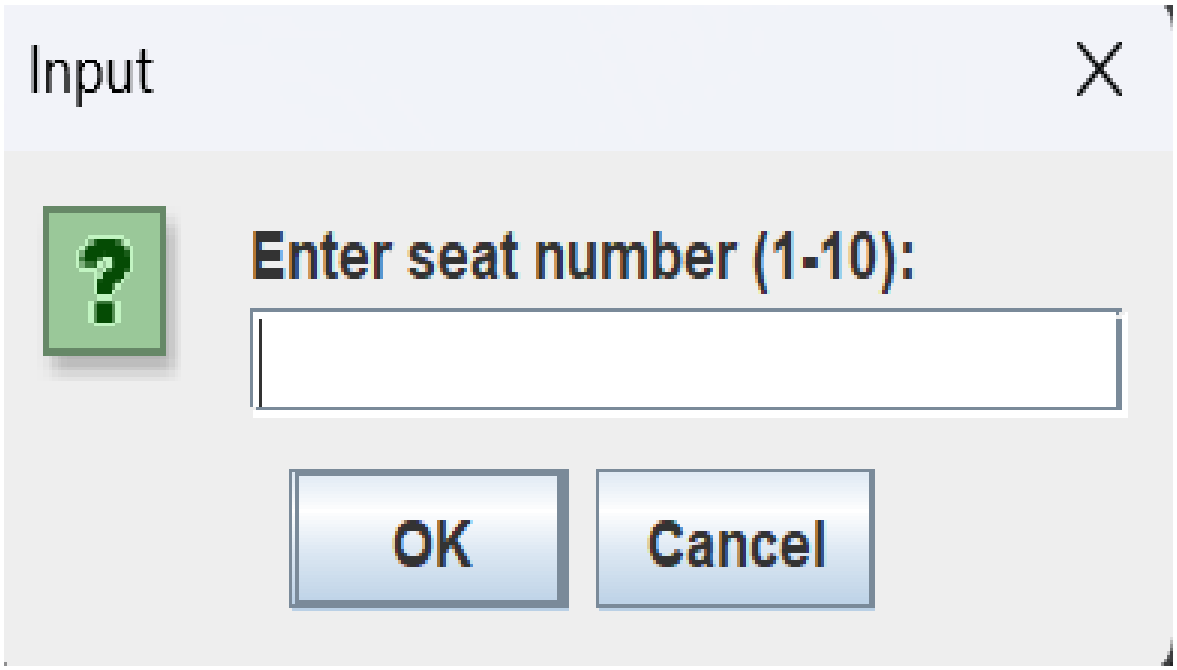
APPENDIX A (SOURCE CODE)

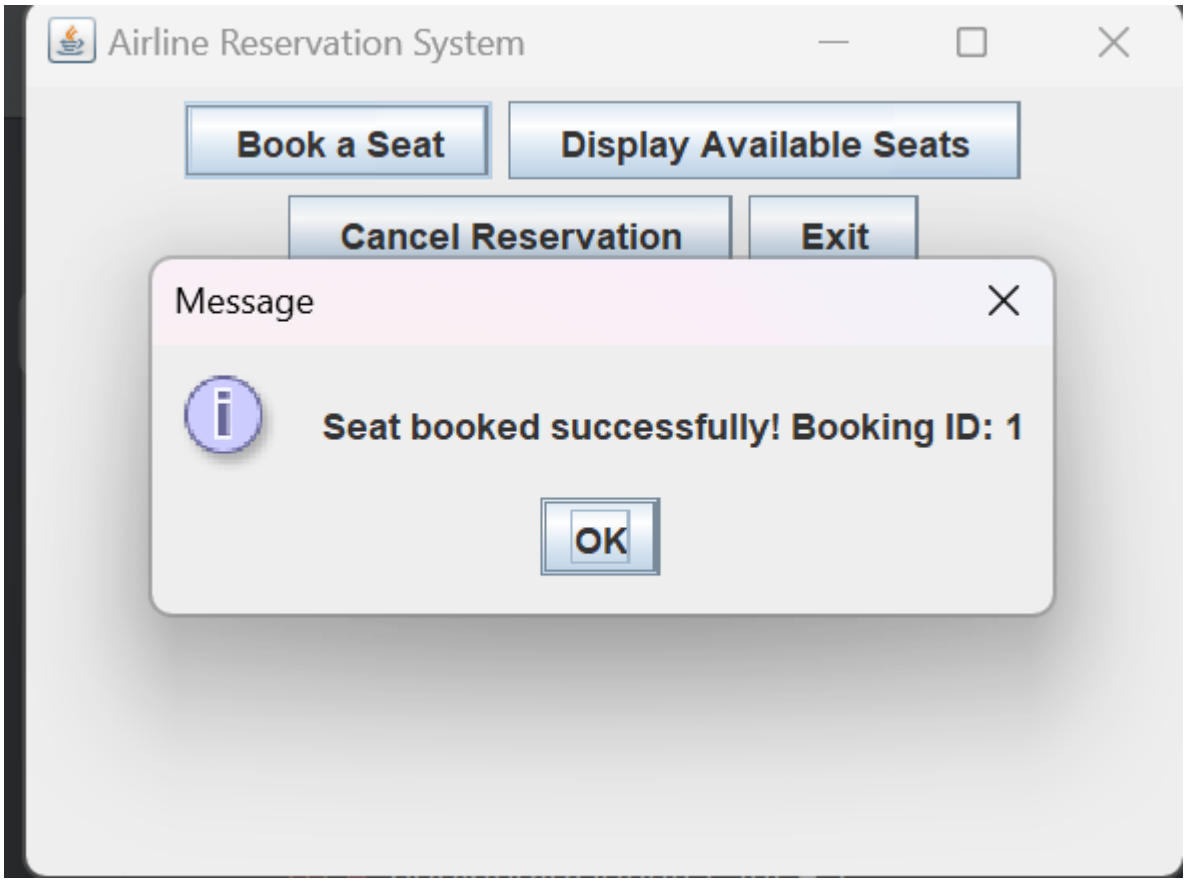
```
import javax.swing.*;
import java.awt.*;
import java.util.HashMap;
import java.util.Map;

public class AirlineReservationSystem extends JFrame {
    private Map<Integer, Booking> bookings = new HashMap<>();
    private int bookingIdCounter = 1;

    public AirlineReservationSystem() {
        setTitle("Airline Reservation System");
        setSize(400, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

        JButton bookSeatButton = new JButton("Book a Seat");
        JButton displaySeatsButton = new JButton("Display Available Seats");
        JButton cancelReservationButton = new JButton("Cancel Reservation");
        JButton exitButton = new JButton("Exit");

        bookSeatButton.addActionListener(e -> bookSeat());
        displaySeatsButton.addActionListener(e -> displayAvailableSeats());
        cancelReservationButton.addActionListener(e -> cancelReservation());
        exitButton.addActionListener(e -> System.exit(0));

        add(bookSeatButton);
        add(displaySeatsButton);
        add(cancelReservationButton);
        add(exitButton);
    }

    private void bookSeat() {
        String seatInput = JOptionPane.showInputDialog("Enter seat number (1-10):");
        int seatNumber = Integer.parseInt(seatInput);
        String name = JOptionPane.showInputDialog("Enter passenger name:");
        String phone = JOptionPane.showInputDialog("Enter phone number:");

        if (seatNumber < 1 || seatNumber > 10 || bookings.containsKey(seatNumber)) {
            JOptionPane.showMessageDialog(this, "Invalid seat number or seat already booked.");
            return;
        }

        bookings.put(seatNumber, new Booking(bookingIdCounter++, name, phone));
        JOptionPane.showMessageDialog(this, "Seat booked successfully! Booking ID: " + (bookingIdCounter
- 1));
    }

private void displayAvailableSeats() {
    StringBuilder availableSeats = new StringBuilder("Available Seats:\n");
    for (int i = 1; i <= 10; i++) {
```

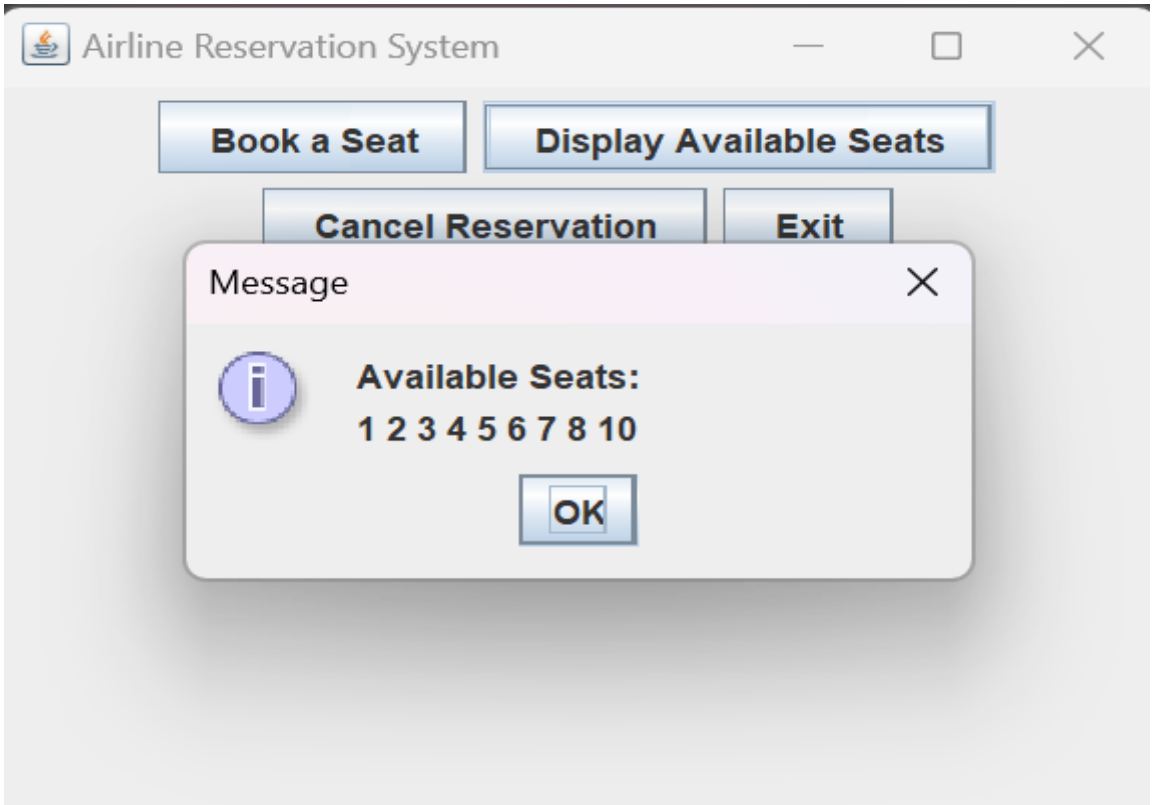
```
        if (!bookings.containsKey(i)) {
            availableSeats.append(i).append(" ");
        }
    }

JOptionPane.showMessageDialog(this, availableSeats.toString());
    }

    private void cancelReservation() {
        String bookingIdInput = JOptionPane.showInputDialog("Enter booking ID to
cancel:");
        int bookingId = Integer.parseInt(bookingIdInput);
        Booking bookingToCancel = null;

        for (Map.Entry<Integer, Booking> entry : bookings.entrySet()) {
            if (entry.getValue().getBookingId() == bookingId) {
                bookingToCancel = entry.getValue();
                bookings.remove(entry.getKey());
                break;
            }
        }

        if (bookingToCancel != null) {
            JOptionPane.showMessageDialog(this, "Reservation cancelled. Passenger Name: "
+ bookingToCancel.getName());
        } else {
            JOptionPane.showMessageDialog(this, "Booking ID not found.");
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            AirlineReservationSystem system = new AirlineReservationSystem();
            system.setVisible(true);
        });
    }

    class Booking {
        private int bookingId;
        private String name;
        private String phone;

        public Booking(int bookingId, String name, String phone) {
            this.bookingId = bookingId;
            this.name = name;
            this.phone = phone;
        }

        public int getBookingId() {
            return bookingId;
        }

        public String getName() {
            return name;
        }
    }
}
```

APPENDIX B(SCREENSHOTS)






Airline Reservation System
Book a Seat
Display Available Seats
Cancel Reservation
Exit
Message
Seat booked successfully! Booking ID: 1
OK


Airline Reservation System
Book a Seat
Display Available Seats
Cancel Reservation
Exit
Message
Available Seats:
1 2 3 4 5 6 7 8 10
OK