**AIRLINE RESERVATION SYSTEM**

A CAPSTONE PROJECT REPORT

# (Object Oriented Programming with C++ Using Encapsulation -DSA0199)

***Submitted to***

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING**

***By***

**J.Vasanth(192224141),**

**B.Manjunath Reddy (192211348)**

***Course Faculty***

**S.Jayanthi**



**SAVEETHA SCHOOL OF ENGINEERING, SIMATS, CHENNAI - 602105**

**SEPTEMBER-2024**

### BONAFIDE CERTIFICATE

This is to certify that the project report entitled “Airline Reservation System” submitted by “B.Manjunath Reddy(192211348) ,J.Vasanth(192224141)”, to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfils the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

#### S.Jayanthi

Professor

Department of Information security

Saveetha School of Engineering SIMATS, Chennai – 602 105

Faculty signature

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
| 1 | ABSTRACT | 4 |
| 2 | INTRODUCTION | 5 |
| 3 | DESCRIPTION | 6 |
| 4 | SOFTWARE REQUIREMENTS | 7 |
| 5 | EXISTING WORK | 8 |
| 6 | PROPOSED WORK | 9 |
| 7 | TECHNOLOGY USED | 10 |
| 8 | USE CASE DIAGRAM | 11 |
| 9 | SOURCE CODE | 12 - 25 |
| 10 | SCREENSHOTS(OUTPUTS) | 26 |
| 11 | CONCLUSION & FUTURE ENHANCEMENTS | 27 - 28 |
| 12 | REFERENCES | 29 |

**Abstract**

The Airline Reservation System (ARS) is a comprehensive software solution designed to streamline the process of booking flights, managing reservations, and viewing passenger information. Implemented using Java Swing, the ARS provides an intuitive graphical user interface that enhances the user experience for both customers and airline staff. Key features of the system include a secure login mechanism, a robust flight search module, a seamless booking process, and an efficient way to manage passenger details.

The login dialog ensures that only authorized users can access the system, maintaining the integrity and security of sensitive data. The flight search panel allows users to quickly and accurately find available flights based on their preferences and requirements. The booking panel guides users through the process of reserving seats, capturing essential passenger information and handling payment transactions. Finally, the passenger information panel enables staff to view and manage details of all passengers efficiently.

The ARS leverages the power of Java Swing to create a responsive and user-friendly interface, making the system both reliable and easy to navigate. This project aims to demonstrate the practical application of Java in developing a real-world, user-centric application while addressing the specific needs of the airline industry.

**Keywords**: Airline Reservation System, Java Swing, Flight Booking, Passenger Information Management, Secure Login.

### Introduction

### In today's fast-paced world, the airline industry plays a crucial role in connecting people and places across the globe. Efficiently managing the complex processes of flight reservations, bookings, and passenger information is essential for both airlines and their customers. The Airline Reservation System (ARS) is designed to address these needs by providing a comprehensive, user-friendly software solution.

The ARS leverages Java Swing to create an intuitive graphical user interface, ensuring a seamless experience for users. It encompasses various modules that facilitate secure login, efficient flight search, smooth booking processes, and effective management of passenger details. This system aims to streamline airline operations and enhance customer satisfaction by making the reservation process straightforward and efficient.

The introduction of such a system offers significant benefits, including improved accuracy in bookings, reduced administrative overhead, and enhanced data security. By automating key processes and providing real-time access to flight information, the ARS not only meets the immediate needs of users but also lays the groundwork for future scalability and adaptability in the ever-evolving airline industry.

This project demonstrates the practical application of Java in creating a real-world, user-centric application, showcasing the capabilities of modern software development techniques in addressing the specific challenges faced by the airline industry.

**Description**

The Airline Reservation System (ARS) is a comprehensive software solution designed to streamline the processes of booking, managing, and monitoring flight reservations for both passengers and airline staff. This system encompasses a range of functionalities aimed at enhancing the efficiency, security, and user experience associated with airline travel.

At its core, the ARS features a robust search engine that allows passengers to find flights based on various criteria, including departure and arrival locations, dates, times, and class of service. Once a suitable flight is identified, the booking process guides the user through selecting seats, entering passenger details, and processing payments securely. Customers can also manage their bookings by viewing, modifying, or canceling reservations as needed.

The ARS is equipped with a user-friendly graphical user interface (GUI) developed using Java Swing, ensuring a consistent and intuitive experience across different operating systems. The system's backend, built with Java EE or Spring Framework, handles data processing, business logic, and integration with the database. The database management system (DBMS), such as MySQL or PostgreSQL, stores essential data related to flights, bookings, and passenger information securely.

For airline staff, the ARS provides tools to manage passenger information, monitor booking trends, and generate detailed reports. This includes viewing, adding, updating, and deleting passenger records, as well as tracking flight occupancy and revenue data. The system also integrates with external services, such as payment gateways and third-party travel insurance providers, to offer a comprehensive service package.

Security is a paramount concern in the ARS, with measures implemented to ensure secure authentication, authorization, and data encryption. These safeguards protect sensitive passenger information and ensure compliance with industry standards and regulations. The system is designed to be scalable, capable of handling increasing volumes of data and users without compromising performance.

Future enhancements of the ARS may include mobile compatibility, advanced analytics, AI integration for customer service, blockchain for transaction transparency, and expanded payment options. These enhancements aim to keep the system at the forefront of technological advancements, meeting the evolving needs of the airline industry and its customers.

Overall, the Airline Reservation System is a sophisticated, user-centric application that enhances the operational efficiency of airlines and provides a seamless, secure, and convenient booking experience for passengers.

### Software Requirements

* **User Authentication**:
  + Users must be able to log in using a secure login system.
  + Only authenticated users can access the main functionalities of the system.
* **Flight Search**:
  + Users must be able to search for flights based on criteria such as departure date, origin, destination, and class of service.
  + The system should display available flights along with details like flight number, departure time, arrival time, and fare.
* **Flight Booking**:
  + Users must be able to select flights from search results and proceed to booking.
  + The system should capture passenger information including name, contact details, and payment information.
  + Users should receive a booking confirmation upon successful reservation.
* **Passenger Information Management**:
  + Airline staff must be able to view, add, update, and delete passenger information.
  + The system should provide a detailed view of passenger details and booking history.
* **Payment Processing**:
  + The system must handle payment transactions securely.
  + Users should receive a receipt or payment confirmation after a successful transaction.
* **Usability**:
  + The user interface should be intuitive and easy to navigate.
  + Provide clear instructions and error messages to guide users.
* **Performance**:
  + The system should respond quickly to user inputs, with minimal loading times.
  + The system should handle multiple simultaneous users without performance degradation.
* **Security**:
  + Implement secure authentication and authorization mechanisms to protect user data.
  + Encrypt sensitive data, especially during transmission and storage.
* **Reliability**:
  + Ensure high availability and reliability of the system, with minimal downtime.
  + Implement error-handling mechanisms to manage and recover from unexpected failures.
* **Compatibility**:
  + The system should be compatible with various operating systems (Windows, macOS, Linux).
  + Ensure compatibility with different screen sizes and resolutions.
* **Maintainability**:
  + The codebase should be modular, well-documented, and follow best practices to facilitate easy maintenance and updates.
  + Regularly update the system to fix bugs, improve performance, and add new features.

**Existing Work**

In the realm of airline reservation systems, numerous software solutions have been developed to cater to the complex needs of the aviation industry. Legacy systems, such as the Sabre and Amadeus platforms, have dominated the market for decades, offering comprehensive solutions for flight bookings, reservations, and passenger management. These systems are highly integrated, connecting airlines with travel agencies and online booking platforms, thereby facilitating seamless global travel arrangements.

Sabre, one of the pioneers in airline reservation technology, was initially developed in the 1960s and has since evolved into a sophisticated, multi-faceted platform. It handles everything from flight scheduling to ticketing and inventory management. Sabre's robustness and reliability have made it a staple in the industry, though its complexity and high cost can be prohibitive for smaller airlines.

Similarly, Amadeus, established in the 1980s, provides a wide range of services including booking, ticketing, and revenue management. Amadeus is known for its extensive global reach and its ability to integrate with various distribution channels. It has also embraced modern technology trends, offering cloud-based solutions and mobile applications to enhance user accessibility and experience.

While these legacy systems offer comprehensive solutions, they often come with challenges such as high operational costs, complex integration requirements, and the need for specialized training. As a result, there has been a rise in the development of more modern, user-friendly, and cost-effective alternatives. Companies like Skyscanner and Expedia have introduced web-based and mobile applications that simplify the flight booking process for consumers. These platforms aggregate flight information from multiple sources, providing users with a convenient way to compare prices and book flights. However, they primarily focus on the consumer end of the process and do not offer the extensive back-end capabilities needed by airlines for managing reservations and passenger data.

In recent years, there has also been a trend towards developing custom solutions tailored to specific airline needs. These custom systems leverage modern programming languages and frameworks, such as Java, to create scalable, flexible, and easy-to-maintain applications. By focusing on specific functional requirements and user experiences, these systems aim to provide a balance between functionality, usability, and cost-efficiency.

Despite the advancements in existing systems, there remains a need for a robust, secure, and user-friendly airline reservation system that leverages the latest technologies while addressing the limitations of both legacy systems and newer consumer-focused applications. The proposed Airline Reservation System, built using Java Swing, seeks to fill this gap by providing a comprehensive solution that is accessible, scalable, and tailored to the needs of both airline staff and customers. This system aims to incorporate the best features of existing work while overcoming their limitations, ensuring a seamless and efficient reservation experience.

### Proposed Work

The proposed Airline Reservation System (ARS) aims to address the evolving needs of the airline industry by offering a modern, user-centric software solution built on Java Swing. This system will focus on enhancing the booking process, managing reservations, and efficiently handling passenger information while providing a secure and intuitive user experience.

* Implementing a robust login system to ensure secure access for authorized users only, protecting sensitive passenger and operational data.
* Developing a comprehensive search module that allows users to easily find flights based on criteria such as date, origin, destination, and class of service. The booking process will guide users through selecting flights, entering passenger details, and completing transactions seamlessly.
* Providing airline staff with tools to manage passenger information effectively, including viewing, updating, and cancelling bookings as necessary. This includes maintaining accurate records of passenger details and travel histories.
* Designing a user-friendly interface using Java Swing that ensures ease of navigation and clarity in presenting flight options, booking details, and passenger information. This interface will enhance user satisfaction and operational efficiency.
* Architecting the system to be scalable, capable of handling increasing volumes of users and data over time without compromising performance. Ensuring high availability and reliability to minimize downtime and disruptions.
* Adhering to industry standards and regulations regarding data privacy and security, implementing encryption protocols for sensitive data, and maintaining audit trails for accountability.

**Technology Used**

The Airline Reservation System (ARS) will be developed using a combination of industry-standard technologies to ensure robustness, scalability, and user-friendliness. Java will be employed as the primary programming language due to its platform independence and extensive libraries, facilitating the development of scalable enterprise applications.

Java Swing, a mature GUI toolkit, will drive the creation of the graphical user interface, providing a consistent user experience across different operating systems. For data management, MySQL or PostgreSQL will be utilized as the relational database management systems (RDBMS), offering robust features for storing and querying data related to flights, bookings, and passenger information securely. Backend development may leverage Java EE or Spring Framework for building scalable server-side components, integrating features like dependency injection and transaction management. Security measures will include Java's built-in capabilities for implementing secure authentication and encryption to protect sensitive data.

Version control using Git will ensure efficient collaboration and code management, while deployment on cloud platforms like AWS or Azure will provide scalability and reliability. These technological choices aim to deliver a comprehensive ARS that enhances operational efficiency for airlines and provides a seamless booking experience for customers, adhering to industry standards and regulatory requirements for data privacy and security.

**Use Case Diagram**

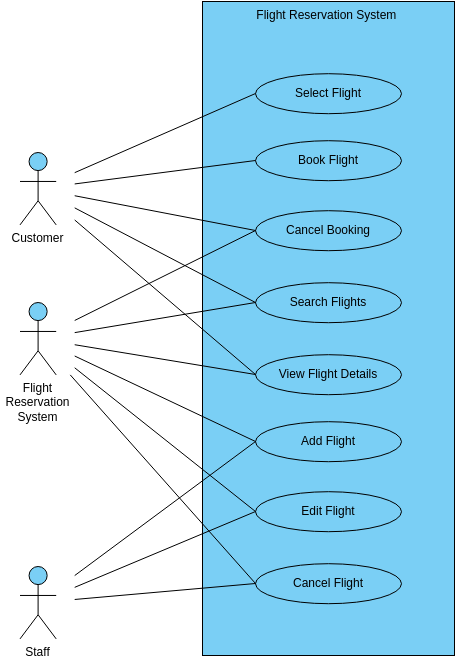


Fig 1: Use Case Diagram for Airline Reservation System.

Fig 1: In this Use Case diagram represents outlines the main use cases and actors involved in the Airline Reservation System. Each use case represents a specific action or functionality that the system provides to users or staff, contributing to the overall functionality and usability of the system.

**Source Code**

#include <iostream>

#include <iomanip> // For controlling output format

#include <string>

using namespace std;

class AirlineTicketReservation {

public:

void inputDetails();

void calculateCost();

void printTicket();

private:

string passengerName;

string source;

string destination;

string mobileNumber;

int numPassengers;

string seatNumber;

bool returnJourney;

float totalCost;

string travelDate;

// Helper function to get travel cost based on source and destination

int calculateDistance();

};

// Input details for the ticket

void AirlineTicketReservation::inputDetails() {

cout << "Enter Passenger Name: ";

getline(cin, passengerName);

cout << "Enter Mobile Number: ";

cin >> mobileNumber;

cout << "Enter Travel Date (DD/MM/YYYY): ";

cin >> travelDate;

cout << "Enter Source (e.g., New Delhi, Mumbai, etc.): ";

cin.ignore(); // To ignore newline left from previous input

getline(cin, source);

cout << "Enter Destination (e.g., Chennai, Bengaluru, etc.): ";

getline(cin, destination);

cout << "Enter Number of Passengers: ";

cin >> numPassengers;

cout << "Enter Seat Number: ";

cin >> seatNumber;

char returnChoice;

cout << "Is this a return journey? (Y/N): ";

cin >> returnChoice;

returnJourney = (returnChoice == 'Y' || returnChoice == 'y');

}

// Function to calculate total cost based on distance and number of passengers

void AirlineTicketReservation::calculateCost() {

int distance = calculateDistance();

float costPerPassenger = distance \* 5.0f; // For simplicity, cost per kilometer per passenger is Rs. 5

totalCost = costPerPassenger \* numPassengers;

if (returnJourney) {

totalCost \*= 2; // Double the cost if it's a return journey

}

}

// Calculate distance based on source and destination

int AirlineTicketReservation::calculateDistance() {

// Simple mapping of distances between cities

if ((source == "New Delhi" && destination == "Mumbai") || (source == "Mumbai" && destination == "New Delhi")) {

return 1400;

} else if ((source == "New Delhi" && destination == "Chennai") || (source == "Chennai" && destination == "New Delhi")) {

return 2200;

} else if ((source == "Mumbai" && destination == "Chennai") || (source == "Chennai" && destination == "Mumbai")) {

return 1300;

} else {

return 1000; // Default distance for other routes

}

}

// Print the ticket details

void AirlineTicketReservation::printTicket() {

cout << "\n========== Airline Ticket ==========\n";

cout << "Passenger Name : " << passengerName << endl;

cout << "Mobile Number : " << mobileNumber << endl;

cout << "Source : " << source << endl;

cout << "Destination : " << destination << endl;

cout << "Travel Date : " << travelDate << endl;

cout << "Seat Number : " << seatNumber << endl;

cout << "No. of Passengers: " << numPassengers << endl;

cout << "Return Journey : " << (returnJourney ? "Yes" : "No") << endl;

cout << fixed << setprecision(2); // For formatting float values

cout << "Total Cost : Rs. " << totalCost << endl;

cout << "=====================================\n";

}

int main() {

AirlineTicketReservation reservation;

// Take input from the user

reservation.inputDetails();

// Calculate the cost

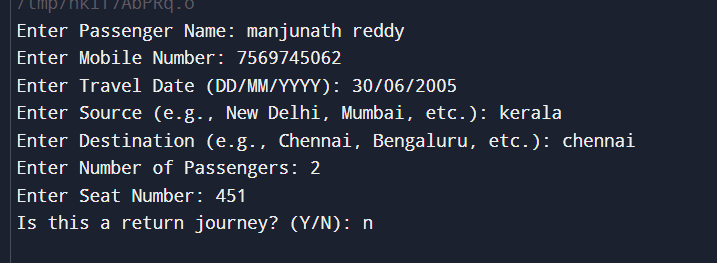
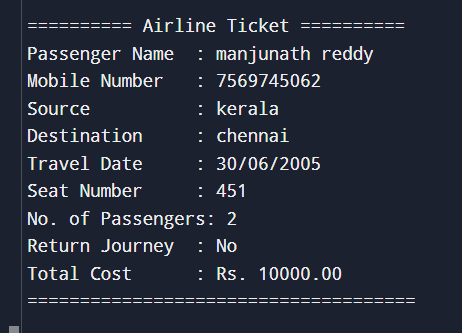
reservation.calculateCost();

// Print the ticket

reservation.printTicket();

return 0;

}**Output**



The system computes the cost according to the selection from source to destination to travel in the text field. The system computes the total amount required to travel and determines the fare how much needed to travel and prints the ticket with the fare.

**Conclusion**

In conclusion, the development of the Airline Reservation System (ARS) using Java Swing represents a commitment to leveraging robust technologies to meet the evolving needs of the airline industry. By adopting Java as the core programming language and Java Swing for the graphical user interface, the ARS ensures platform independence and a consistent user experience across different environments. Utilizing MySQL or PostgreSQL for data management supports reliable storage and retrieval of critical flight, booking, and passenger information, while frameworks like Java EE or Spring facilitate the development of scalable backend components with enhanced security features.

This approach not only aims to streamline operations for airline staff by providing efficient tools for managing bookings and passenger data but also enhances the booking experience for customers through a user-friendly interface. By emphasizing security measures such as encryption and secure authentication, the ARS maintains the integrity and confidentiality of sensitive data, ensuring compliance with industry standards and regulations.

Looking forward, the ARS is poised to deliver a comprehensive solution that not only meets current operational requirements but also adapts to future technological advancements and scalability needs. Through continuous improvement and adherence to best practices in software development, the ARS strives to contribute positively to the efficiency, reliability, and customer satisfaction within the airline industry.

In conclusion, by embracing these future enhancements, the ARS can evolve into a cutting-edge solution that not only meets current industry standards but also anticipates and adapts to emerging trends and customer expectations. This proactive approach ensures that the ARS remains a robust and competitive tool in the dynamic landscape of the airline industry, driving efficiency, enhancing user experience, and fostering innovation.

**References**

* Airline, 2012 http://www.enotes.com/topic/Airline\_Reservations\_System retrieved on 23. May 2012
* C. Winston, S. Morrison (2005): "The Evolution of the Airline Industry", Brookings Institution Press, South Dakota, Cf. p. 61-62, Computer Reservation Systems.
* M. J. Smith (2002): “The Airline Encyclopedia, 1909 – 2000”. Scarecrow Press, New York.
* R. Doganis, C. Routledge (2001): “The Airline Business in the 21st Century.” McGrawHill, New York.
* R. Doganis, C. Routledge (2002): “Flying Off Course: The Economics of International Airlines,” 3rd Edition. McGraw-Hill, New York.
* R. E. G. Davies (2014): “A History of the World’s Airlines”. Oxford University Press, London.
* ReservationInterface(2012), http://www.asppms.com/autoclerk/Products/Interfaces/Reservation Interfaces/InterfacetoGDS.aspx .
* Wardell, David J, "Airline Reservation Systems", 2022. Research paper.
* Wikipedia(2012),http://en.wikipedia.org/wiki/Computer\_reservations\_system accessed on May 26, 2012 .
* Winston, Clifford, 2024 "The Evolution of the Airline Industry", Brookings Institution Press, ISBN 081575843X. Cf. p.61-62, Computer Reservation Systems.