**AIRLINE RESERVATION SYSTEM**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

S. Lokesh AP23110011424

S. Lakshmikar AP23110011410

S. Pavan Vamsi Krishna AP23110011412

V. Prasanna Sai AP23110011430

**A picture containing text

Description automatically generated**

Under the Guidance of

**(Supervisor Name)**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**November, 2024**

# Certificate

Date: 20-Nov-24

This is to certify that the work present in this Project entitled “**Airline reservation system**” has been carried out by **Lokesh, Lakshimkar, Pavan, Prasanna** under our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

# Acknowledgements

I express my profound gratitude to my mentor and instructor for their expertise, patience, and unwavering support, which have been instrumental throughout my C++ project journey. Their invaluable insights and constructive criticism have significantly enhanced the quality of my work, while their encouragement has been a constant source of motivation.

I am equally thankful to my peers for their collaborative spirit and constructive feedback, which have been crucial in refining my project. Their diverse perspectives and innovative ideas have enriched my understanding and have been essential in overcoming numerous challenges.

Lastly, I acknowledge the steadfast support of my family and friends. Their understanding and encouragement have been a pillar of strength, enabling me to dedicate myself fully to this endeavor. Their belief in my abilities has been a source of inspiration, and their patience during the demanding phases of this project has been greatly appreciated.

# Table of Contents

[Certificate i](#_gjdgxs)

[Acknowledgements .ii](#_30j0zll)

[Table of Contents iii](#_1fob9te)

[Abstract i](#_3znysh7)v

[Statement of Contributions v](#_tyjcwt)

[Abbreviations v](#_3dy6vkm)i

[List of Tables vii](#_1t3h5sf)

[List of Figures viii](#_4d34og8)

[1.](#_17dp8vu) Introduction 1

[2.](#_lnxbz9) Methodology 2

[2.1](#_35nkun2) Class Design………………………………………………………………………….2

2.2 Search Functionality ………………………………………………………………...2

2.3 User Interface………………………………………………………………………...2

2.4 Data Management…………………………………………………………………...2

2.5 Error Handling………………………………………………………………………2

2.6 Testing and Validation……………………………………………………………...3

2.7 Documentation……………………………………………………………………...3

[3.](#_44sinio) Discussion 4

[4.](#_2jxsxqh) Concluding Remarks 5

[5.](#_z337ya) Future Work 6

[References 7](#_3j2qqm3)

# Abstract

My C++ project is centered around creating a robust system that facilitates the search for destinations and available flights, manages airline reservations, processes cancellations, and handles payments. The primary goal of this project is to simplify the booking process, improve user experience, and ensure secure transactions. By incorporating advanced search algorithms and a user-friendly interface, the project caters to the needs of both travelers and airlines, offering a seamless and efficient reservation experience.

The system is designed to be comprehensive, covering all aspects of the airline reservation process. It allows users to easily search for their desired destinations and find available flights that suit their schedules. Once a flight is selected, the system enables users to make reservations quickly and efficiently. In the event that a reservation needs to be canceled, the system provides a straightforward process for doing so, ensuring that users can manage their bookings with ease.

Payment processing is another critical component of the project. The system is equipped to handle various payment methods securely, ensuring that transactions are safe and reliable. This feature is essential for building trust with users and encouraging them to use the system for their travel needs.

Overall, the project aims to address the challenges faced by travelers and airlines in the reservation process. By offering a streamlined and user-friendly solution, it enhances the overall travel experience and supports the efficient operation of airlines. The integration of advanced technology and a focus on user satisfaction make this C++ project a valuable tool for the travel industry.

# [Statement of Contributions](#_tyjcwt)

**1.Pavan**

**Testing and Data Simulation**

**Testing:**

Test core functions (reservation, cancellation, payment).

Handle edge cases like invalid inputs and exceeding member limits.

**Data Simulation:**

Generate sample flight, reservation, and passenger data for testing.

Simulate scenarios like overbooking and incomplete payments.

**2.Prasanna sai**

**Debugging and Error Handling**

**Debugging:**

Identify and fix logical issues in functions like cancelMember.

Ensure correct shifting of members and removal of empty reservations.

**Error Handling:**

Add validation for user inputs (flight numbers, member details).

Handle invalid operations like duplicate cancellations or nonexistent flights.

**3.Lokesh**

**Code Optimization and System Integration:**

**Code Optimization:**

Refactor code for better performance (e.g., efficient member shifting).

Remove redundant code in reservation functions.

**System Integration:**

Ensure smooth interaction between flight, reservation, and member data.

Test the entire reservation flow from booking to payment.

**4.Lakshmikar**

**Documentation and Advanced Reporting**

**Documentation:**

Write comments and user documentation explaining the system’s functions and flow.

**Advanced Reporting:**

Add features to generate summary reports (e.g., total reservations, payment status).

Implement filters to view reservations by flight or payment status.

# Abbreviations

**1**. **Publications for Search for Destination:**

- **ETW**: "Exploring the World: A Guide to Top Travel Destinations"

- **DD**: "Destination Discovery: Unveiling Hidden Gems"

**- UTG**: "The Ultimate Travel Guide: Finding Your Perfect Getaway"

**2. Abbreviations for Available Flight:**

**- AA**: American Airlines

**- DL**: Delta Airlines - UA: United Airlines

**- SW**: Southwest Airlines

**3. Airline Reservation**:

**- HBF**: "How to Book Your Flight: A Step-by-Step Guide"

**- NAR**: "Navigating Airline Reservations: Tips and Tricks"

**- CGF**: "The Complete Guide to Booking Flights Online"

**4. Cancellation of Reservation:**

**- UACP**: "Understanding Airline Cancellation Policies"

**- HCFR**: "How to Cancel Your Flight Reservation: A Comprehensive Guide"

**- FC**: "Flight Cancellations: What You Need to Know"

**5. Payment:**

**- SPM**: "Secure Payment Methods for Booking Flights"

**- UAPO**: "Understanding Airline Payment Options"

**- GP**: "A Guide to Paying for Your Flight: Tips and Advice"

# List of Tables

|  |  |  |
| --- | --- | --- |
| **Departure location** | **Destination** **Location** | **Flight I’d** |
| Delhi | Pune | 6E6212 |
| Hyderabad | Delhi | 6E6659 |
| Mumbai | Nagpur | 6E 835 |
| Mumbai | Kolkata | 6E2148 |
| Goa | Chennai | 6E2239 |
| Chennai | Mumbai | 6E2103 |
| Delhi | Patna | 6E2342 |

# Introduction

The airline reservation system is a comprehensive platform designed to enhance the travel booking experience by integrating a robust search feature that allows users to efficiently locate destinations and available flights. This system is meticulously crafted to streamline the process of airline reservations, ensuring that users can effortlessly book their desired flights with ease and convenience.

In addition to the core functionality of flight search and booking, the system is equipped with advanced features that cater to the diverse needs of travelers. One of the key functionalities is the ability to manage reservations, which includes the option to cancel bookings if necessary. This feature is particularly beneficial for users who may need to alter their travel plans due to unforeseen circumstances, providing them with the flexibility to adjust their itineraries without hassle.

Furthermore, the system incorporates a secure and efficient payment processing module, which facilitates seamless transactions for users. This ensures that all financial dealings are conducted with the utmost security and reliability, instilling confidence in users as they complete their bookings.

The entire system is developed using object-oriented programming (OOP) concepts, which offer a structured and modular approach to software development. This methodology not only enhances the maintainability and scalability of the system but also allows for the integration of additional features and improvements in the future. By leveraging OOP principles, the airline reservation system is designed to deliver a user-friendly and efficient experience, making it an indispensable tool for modern travelers.

# Methodology

To develop a comprehensive airline reservation system using the object-oriented programming (OOP) concept, we need to consider several key components and methodologies. The system should be designed to efficiently handle various tasks such as searching for destinations, checking available flights, managing airline reservations, processing cancellations, and handling payments. Here is an expanded methodology for the project:

1**. Class Design:**

- **Destination**: This class will encapsulate details about various destinations, including attributes like destination name, airport code, and country. Methods will include retrieving destination information and listing popular destinations.

- **Flight**: This class will manage flight details, including flight number, departure and arrival times, airline, and seat availability. Methods will include checking seat availability, updating flight schedules, and retrieving flight information.

- **Reservation:** This class will handle reservation details, including reservation ID, passenger information, flight details, and reservation status. Methods will include creating a new reservation, modifying existing reservations, and retrieving reservation details.

- **Cancellation**: This class will manage the cancellation process, including attributes like cancellation ID, reservation ID, and refund status. Methods will include processing cancellations, updating reservation status, and calculating refunds.

**- Payment:** This class will handle payment processing, including attributes like payment ID, amount, payment method, and transaction status. Methods will include processing payments, issuing refunds, and generating payment receipts.

**2. Search Functionality**:

- Implement a search algorithm to allow users to search for destinations and available flights. The search should be efficient and support various filters such as date, time, airline, and price range.

**3. User Interface:** Design a user-friendly interface that allows users to easily navigate through the system. The interface should provide options for searching flights, making reservations, and managing bookings.

4. **Data Management:**

- Use a database to store and manage data related to destinations, flights, reservations, cancellations, and payments. Ensure data integrity and implement security measures to protect sensitive information.

5. **Error Handling:**

- Implement robust error handling to manage exceptions and ensure the system remains stable and reliable. This includes handling invalid inputs, network issues, and payment processing errors.

6. **Testing and Validation:**

- Conduct thorough testing to validate the functionality of the system. This includes unit testing for individual classes and methods, integration testing for the overall system, and user acceptance testing to ensure the system meets user requirements.

7. **Documentation:**

- Provide comprehensive documentation for the system, including user manuals, technical documentation, and API references. This will aid in future maintenance and updates.

By following this methodology, the airline reservation system will be well-structured, efficient, and user-friendly, leveraging the principles of object-oriented programming to ensure scalability and maintainability.

# Discussion

In the context of an airline reservation system, the object-oriented programming (OOP) concept can be effectively utilized to manage various functionalities such as searching for destinations, checking available flights, making reservations, handling cancellations, and processing payments.

The system can be designed with several classes, each representing a different aspect of the airline reservation process. For instance, a `Destination` class can encapsulate details about various destinations, including attributes like destination name, airport code, and country. This class can have methods to search and filter destinations based on user preferences.

The `Flight` class can represent individual flights, containing attributes such as flight number, departure and arrival times, and available seats. Methods within this class can allow users to search for flights based on criteria like date, time, and destination.

The Reservation class can handle the booking process, storing information about the passenger, selected flight, and seat assignment. It can include methods to create, modify, and cancel reservations, ensuring that seat availability is updated in real-time.

For handling cancellations, a `Cancellation` class can be introduced, which manages the process of canceling a reservation and updating the system accordingly. This class can also handle any penalties or refunds associated with the cancellation.

Finally, a `Payment` class can manage the financial transactions involved in booking a flight. This class can include methods for processing payments, issuing refunds, and generating receipts. It can also integrate with external payment gateways to ensure secure and efficient transactions.

By utilizing OOP principles, the airline reservation system can be modular, scalable, and easy to maintain. Each class can be developed and tested independently, allowing for flexibility in adding new features or modifying existing ones. This approach not only enhances the system's functionality but also improves the overall user experience by providing a seamless and efficient booking process.

# Concluding Remarks

The airline reservation system, designed with object-oriented programming (OOP) concepts, serves as a robust platform for managing various aspects of airline operations. It efficiently handles the search for destinations and available flights, ensuring that users can easily find and book their desired travel options. The system's architecture allows for seamless integration of different functionalities, providing a user-friendly interface for both customers and airline staff.

In addition to facilitating reservations, the system also manages the cancellation of reservations and processes payments. This comprehensive approach ensures that all transactions are handled smoothly and securely, enhancing the overall user experience. By leveraging OOP principles, the system is modular and scalable, allowing for future enhancements and adaptations to meet evolving business needs.

# Future Work

Future work for the airline reservation system involves enhancing the search functionality for destinations and available flights. This will include implementing advanced search algorithms to provide users with more accurate and efficient results. Additionally, the system will be designed to handle airline reservations and cancellations seamlessly, ensuring a smooth user experience. The integration of a robust payment system will also be a priority, allowing for secure and convenient transactions. These improvements will be developed using the object-oriented programming (OOP) concept, which will facilitate better organization and scalability of the code.

Incorporating OOP principles, the project will focus on creating modular and reusable components. Classes and objects will be utilized to represent various entities such as flights, reservations, and payments, promoting code reusability and maintainability. By leveraging inheritance and polymorphism, the system will be able to adapt to future changes and enhancements with minimal effort. This approach will not only streamline the development process but also ensure that the system remains flexible and efficient in meeting the evolving needs of users.