Software Requirement specification

Title: Airline Reservation System

Prepared By: Gemtessa Deksisa Tolera

Software Engineering and Management(BA) Student at Haramaya University

Jun 2025.

Table of Contents

List of Figures

List of Tables

List of Abbreviations/Acronyms

1. **API**----------Application Programming Interface
2. **CRUD**---------- Create, Read, Update, Delete
3. **CSS**----------Cascading Style Sheets
4. **DBMS**----------Database Management System
5. **ERD**----------Entity Relationship Diagram
6. **GDS** ----------Global Distribution System
7. **HTML**----------Hypertext Markup Language
8. **IATA** ----------International Air Transport Association
9. **OTA** ----------Online Travel Agency
10. **OTP** ----------One-Time Password
11. **PNR** – ----------Passenger Name Record
12. **SQL** – ----------Structured Query Language
13. **SSL** – ----------Secure Sockets Layer
14. **TCP/IP** ----------Transmission Control Protocol / Internet Protocol
15. **UI**----------User Interface
16. **UX** ----------User Experience

Abstract

The rapid evolution of the aviation industry calls for innovative, user-centric solutions to streamline the booking process and enhance customer satisfaction. This project presents the design and implementation of an Airline Reservation System that aims to replace fragmented, manually driven processes with a robust, automated platform. The proposed system integrates functionalities such as user registration, real-time flight search, seat selection, booking management, and secure payment processing into a seamless, multi-tiered architecture.

Employing contemporary software development methodologies, the system is developed using agile techniques to ensure adaptability and continuous improvement. Comprehensive requirement analyses, including both functional and nonfunctional specifications, underpin the design. Detailed use case models, class diagrams, and system interaction workflows further illustrate the transformation of the existing booking challenges into an efficient, scalable solution. In addition, feasibility assessments across technical, economic, operational, and scheduling aspects validate the project's viability and highlight its potential for widespread application in the competitive airline industry.

By harnessing state-of-the-art technology and user interface design principles, the Airline Reservation System not only reduces operational overhead but also significantly enhances the end-user experience. This project demonstrates a commitment to innovation in airline service management, offering a blueprint for future advancements in transportation systems.

CHAPTER ONE

1.1 Introduction of the Project

In today's fast-paced world, the aviation industry faces constant pressure to deliver seamless travel experiences while managing complex flight operations. The Airline Reservation System project is conceived as a response to the challenges posed by outdated and fragmented booking methods. As traditional processes struggle with inefficiencies, error-prone manual interventions, and limited accessibility, this project aims to modernize airline reservations through an integrated, user-centric digital platform [1].

The project envisions a comprehensive system that not only facilitates real-time flight search and booking but also addresses core issues such as secure payment processing and dynamic seat allocation. By leveraging modern software engineering methodologies, the system is designed to enhance the overall travel experience for both passengers and airline operators. This initiative underscores the need for a scalable, robust solution that can adapt to evolving requirements in the competitive landscape of airline services.

Ultimately, the Airline Reservation System project represents a strategic effort to streamline flight management operations, reduce human error, and significantly elevate customer satisfaction. By embracing innovative technologies and adhering to rigorous development standards, the project establishes a foundation for the digital transformation of airline services, setting the stage for future enhancements and industry-wide best practices [2].

1.2 Motivation of the Project

In today’s rapidly evolving aviation landscape, customer expectations and technological advancements have redefined the way passengers interact with airlines. Today, travel is no longer just about getting from point A to point B—it is about convenience, reliability, and a personalized experience that aligns with the fast-paced demands of modern life. This project is motivated by the pressing need to replace outdated, manual airline booking procedures with an innovative, user-friendly digital system that enhances all aspects of the travel process.

A major driver behind this initiative is the inefficiency and complexity faced by both airline operators and customers in traditional reservation systems. With increasing flights and a surge in air travel, these legacy systems often falter under the strain of real-time processing and secure data management, leading to long queues, booking errors, and inadequate customer support. By adopting a modern IT solution, airlines can

streamline operations, optimize resource management, and provide a seamless interface for booking flights, managing reservations, and processing payments.

Furthermore, rapid technological growth, particularly in areas like cloud computing, mobile applications, and data analytics, presents an unprecedented opportunity to transform airline reservations. Leveraging these advancements not only allows for the creation of a responsive and scalable platform, but also empowers airlines to deliver personalized services, real-time updates, and enhanced security measures. This digital transformation can significantly boost operational efficiency and customer satisfaction, making air travel more accessible and enjoyable.

Ultimately, the motivation behind the Airline Reservation System project is to bridge the gap between traditional, cumbersome booking methods and the modern, tech-driven demands of today's travelers. This undertaking represents a strategic step towards innovation in the aviation industry—one that improves the travel experience, reduces operational costs, and sets the stage for future enhancements in airline service delivery [3].

1.3 Statement of the Problem

In today's fast-paced world, many travelers experience frustration when booking flights due to cumbersome, outdated systems that struggle to keep up with modern demands. Often, a customer finding the right flight is met with sluggish interfaces, confusing navigation, and long waiting times—all of which contribute to a less-than-ideal booking experience. These challenges are not only inconvenient for passengers but also create significant operational hurdles for airline staff who must manage these legacy systems [4].

Moreover, outdated reservation systems are prone to errors such as duplicate entries, conflicting information, and inefficient data processing. This often results in booking inaccuracies and schedule mismanagement, heightening the risk of overbooking and customer dissatisfaction. The lack of real-time updates and seamless integration between different components of the system further exacerbates these issues, placing both customers and service providers in a position of uncertainty.

Another critical concern is security. Legacy systems frequently have inadequate measures in place to protect sensitive customer data, making them vulnerable to cyber threats. With the increasing amount of personal and payment information processed daily, the risk of data breaches can severely impact customer trust and airline reputations.

In summary, there is a pronounced gap between the potential of modern digital solutions and the limitations

imposed by current airline reservation systems. This project aims to address these inefficiencies by developing a streamlined, secure, and user-friendly system that not only meets but exceeds the expectations of today's travelers and airline operators. By tackling these problems head-on, the proposed Airline Reservation System seeks to improve both the end-user experience and operational efficiency, setting the stage for a more reliable and enjoyable travel experience [3].

1.4 Objective of the Project

The primary goal of this project is to transform the traditional, fragmented airline booking process into a seamless, secure, and user-friendly experience for both passengers and airline operators. By leveraging modern software development practices, our system aims to address the inefficiencies of legacy systems while ensuring scalability, reliability, and enhanced usability [5].

1.4.1 General Objective

The primary objective of this project is to design, develop, and implement an integrated Airline Reservation System that transforms the traditional booking process into a seamless, efficient, and secure digital experience. This system aims to simplify flight selection, reservation, and payment procedures for travelers while providing airline operators with robust tools to manage schedules, seat allocations, and customer data in real time. By leveraging modern technologies and agile development methodologies, the project seeks to eliminate the inefficiencies of legacy systems, enhance user satisfaction, and support the evolving needs of the aviation industry.

1.4.2 Specific Objectives

1. **User Authentication and Management:** Develop a robust module for user registration, login, and profile management, ensuring secure access and personalized interactions.
2. **Real-Time Flight Search and Scheduling:** Implement a dynamic search engine that allows users to retrieve and compare up-to-date flight schedules, pricing, and availability in real time.
3. **Seamless Booking and Seat Selection:** Create an intuitive booking process that includes features such as seat selection, ticket confirmation, and the handling of modifications or cancellations, minimizing manual interventions and errors.
4. **Secure Payment Processing:** Integrate reliable and secure payment gateways to facilitate multiple payment options, ensuring transaction integrity and data protection.
5. **Administrative Dashboard for Operations:** Provide airline staff with an interactive dashboard to manage flight schedules, oversee bookings, update flight data, and generate operational reports efficiently.

6. **System Performance and Scalability:** Ensure that the system is highly scalable and performs reliably under varied load conditions, supporting growth in user traffic and transaction volumes.
7. **User-Centric Interface Design:** Deliver a responsive and user-friendly interface that is accessible across desktop and mobile devices, enhancing the overall booking experience.

1.5 Methodology of the Project

Developing an effective Airline Reservation System requires a structured approach that integrates comprehensive data collection, systematic development techniques, and the utilization of modern tools. This methodology ensures that the system addresses user needs, adheres to industry standards, and remains scalable for future enhancements.

1.5.1 Data Collection Methods

To capture accurate requirements and understand the challenges experienced by both passengers and airline staff, a combination of primary and secondary data collection methods will be employed:

- **Primary Data:**
  - **Surveys and Questionnaires:** These will be distributed among potential users (travelers and airline personnel) to gather quantitative and qualitative feedback about existing booking challenges and desired features in a digital system.
  - **Interviews and Focus Groups:** One-on-one interviews and group discussions with key stakeholders—such as customer service representatives and IT managers—will provide in-depth insights into the operational and technological pain points.
  - **Observational Studies:** Direct observation of current reservation processes will help identify inefficiencies and areas of friction, ensuring that the new system addresses real-life challenges.
- **Secondary Data:**
  - **Literature and Industry Reviews:** Analyzing academic research, industry reports, and case studies of similar systems will help benchmark best practices and inform the design choices.
  - **Online Resources and Existing Models:** Reviewing relevant documentation, technical articles, and prototypes will ensure that the system leverages state-of-the-art technologies and methodologies.

1.5.2 System Development Methods

The development process follows a structured, iterative methodology that emphasizes flexibility, user feedback, and robust design practices:

- **Agile Development Methodology:** The project will progress through iterative sprints where each cycle includes stages of planning, development, testing, and review. Regular stakeholder meetings and sprint reviews ensure that feedback is incorporated early, enabling rapid adjustments to meet evolving user requirements.
- **Object-Oriented Analysis and Design (OOAD):** Utilizing OOAD techniques, the project will create comprehensive models such as use case diagrams, class diagrams, and sequence diagrams. This approach helps in modularizing the system's functionality, ensuring scalability and ease of maintenance.
- **Prototyping:** Early prototypes, including wireframes and mock-ups, will be developed to visually represent the user interface and core interactions. Prototyping will help validate design decisions and provide stakeholders with tangible insights into the system's workflow, reducing the risk of major issues during later development stages.
- **Continuous Testing and Integration:** Throughout the development process, unit, integration, and system testing will be conducted in tandem with development. This continuous testing approach, supported by automated testing frameworks, ensures that each component functions as expected and integrates well into the overall system [6].

1.5.3 System Development Tools

To support the chosen development methodologies, a suite of modern development tools will be employed to streamline coding, collaboration, and project management:

- **Programming Languages and Frameworks:**
    - **Backend Development:** Languages such as Java or Python will be used to build robust server-side applications.
    - **Frontend Development:** Frameworks like React or Angular will be employed to create an interactive and responsive user interface.
- **Database Management Systems (DBMS):** A relational database system such as MySQL or PostgreSQL will be used to securely store and efficiently manage user data, flight schedules, reservations, and transactional records.
- **Modeling and Diagramming Tools:** Tools like Lucidchart, Microsoft Visio, or draw.io will be utilized to create detailed UML diagrams (use case, class, and sequence diagrams) that visually document system architecture and workflows.
- **Integrated Development Environments (IDEs) and Code Editors:** IDEs such as Visual Studio Code, Eclipse, or IntelliJ IDEA will be used to facilitate code development, debugging, and testing, ensuring a smooth and productive coding process.

- **Version Control and Collaboration Platforms:** Git-based platforms like GitHub or GitLab will manage source code versions, support collaborative development, and help maintain a clear revision history throughout the project lifecycle.
- **Automated Testing Tools:** Frameworks like JUnit (for Java) or PyTest (for Python) will support continuous integration and provide automated testing capabilities, ensuring consistent quality throughout the development cycle [3].

1.6 Scope of the Study

This study focuses on the design, development, and preliminary testing of a comprehensive Airline Reservation System that transforms the traditional flight booking process into an efficient, user-friendly digital platform. The study encompasses all key phases of software development—from requirements gathering and system design to prototyping and iterative testing—ensuring that the final product addresses current inefficiencies in airline operations and meets modern customer expectations [7].

The project covers the following core areas:

- **User Interface and Experience:** The study involves the design and development of a responsive front-end interface that enables users to search for flights, select seats, execute bookings, and process payments with ease. User-centric design principles will be applied to create an intuitive, accessible experience across various devices.
- **System Architecture and Backend Processing:** A robust, multi-tiered architecture will be developed to manage real-time flight schedules, booking transactions, and user authentication. This includes the design of system workflows, use case modeling, and the creation of class diagrams that outline clear interactions between various system components.
- **Database and Data Management:** An integral part of the study is the development of a relational database system that securely stores and manages essential data such as user profiles, flight details, reservations, and transaction records. The study ensures that effective data normalization and management protocols are in place to facilitate quick and reliable data retrieval.
- **Security and Payment Gateway Integration:** The project will incorporate secure authentication mechanisms and integrate reliable payment processing modules, ensuring that sensitive user data and financial transactions are protected through state-of-the-art security measures.
- **Prototyping and System Testing:** Throughout the project, iterative prototyping and continuous testing (unit, integration, and system testing) will be carried out to validate functionality, performance, and usability. This iterative approach ensures that potential issues are identified and resolved early in the development cycle.

1.7 Significance of the Study

The development of a modern Airline Reservation System is not just about automating bookings; it is about transforming the entire travel experience for both passengers and airline operators. This project holds significant importance in several key areas:

- **Enhanced Customer Experience:** For travelers, time is precious. A user-friendly reservation system means that customers can search, compare, and book flights swiftly, without the frustration of dealing with outdated systems or long waiting times. By simplifying the interface and streamlining the booking process, the system directly contributes to a more positive and stress-free travel experience.
- **Improved Operational Efficiency:** For airlines, managing large volumes of booking data can be complex and error-prone when relying on legacy systems. The proposed system centralizes and automates key processes such as flight scheduling, seat allocation, and payment processing. This not only reduces manual errors but also helps operational staff manage flights more efficiently, leading to overall cost reductions.
- **Enhanced Data Security:** With increasing concerns about data breaches and cyber threats, implementing a secure reservation system is crucial. This project integrates robust security protocols to protect sensitive customer data, fostering trust among users and ensuring compliance with industry standards.
- **Scalability and Adaptability:** The aviation industry is dynamic, with fluctuating travel demands and rapid technological advancements. By designing a system that is both scalable and adaptable, this study lays the groundwork for future innovations. The system can evolve with emerging trends and user needs, ensuring long-term relevance in a competitive market.
- **Economic Impact:** A streamlined reservation process reduces administrative overheads and operational costs for airlines. These savings can be passed on to consumers in the form of lower fares and improved services. In addition, a more efficient system can boost airline revenues by minimizing booking errors and maximizing seat occupancy.

Overall, the significance of this study lies in its potential to set a new standard for airline reservation systems. By addressing the shortcomings of current methods and providing a comprehensive, modern solution, the project contributes to enhanced customer satisfaction, greater operational efficiency, improved data security, and a more resilient framework that can support future industry advancements. [4]

1.8 Feasibility Assessment

Assessing the viability of the Airline Reservation System is crucial to ensure that the proposed solution can be successfully developed, implemented, and maintained. This section examines the feasibility from four key perspectives:

1.8.1 Economic Feasibility

- **Cost Analysis:** Developing a modern reservation system does require a notable upfront investment; expenses include purchasing or leasing the necessary hardware, software development costs, licensing fees, and staff training expenses. However, these costs can be balanced against significant long-term savings. By automating manual tasks, reducing errors, and eliminating fragmented processes, the system will minimize labor costs and operational inefficiencies.
- **Revenue Potential and Return on Investment:** An improved reservation system can enhance customer satisfaction, leading to increased bookings and customer retention. The streamlined process can also reduce overbooking and other costly errors, boosting overall profitability. Therefore, while the initial investment is moderate, the anticipated cost savings and increased revenue streams indicate a strong potential for a positive return on investment.

1.8.2 Technical Feasibility

- **Availability of Modern Technologies:** The system will leverage widely adopted programming languages (such as Java or Python) and modern frameworks (like React or Angular) to build a robust front-end and back-end. With cloud computing and scalable database solutions (such as MySQL or PostgreSQL) readily available, the technical resources are within reach.
- **Compatibility and Integration:** The project design anticipates seamless integration with external data sources (like real-time flight information and payment gateways). Existing technologies and APIs make it possible to incorporate security protocols and data encryption methods that are crucial for protecting user information.
- **Skill Set and Expertise:** With the available development tools and a team experienced in agile methodologies and object-oriented design, the technical challenges can be effectively managed, confirming that the project is technically achievable with current industry standards.

1.8.3 Operational Feasibility

- **User Adoption and Interface Usability:** The system's design emphasizes a user-friendly interface that caters equally to tech-savvy users and first-time travelers. Its intuitive navigation, real-time

processing, and clear visual cues are designed to reduce the learning curve and encourage broad adoption among customers and airline staff.

- **Impact on Daily Operations:** By automating the reservations and administrative tasks, the system will streamline operations for airline personnel. This reduction in manual intervention minimizes errors, increases the accuracy of flight scheduling, and enhances overall operational efficiency. Training sessions and detailed user documentation will further facilitate a smooth transition to the new system.
- **Scalability and Maintenance:** The proposed architecture is scalable, ensuring that the system can handle increased user traffic and data loads as usage grows. Regular system updates and maintenance protocols will be instituted to ensure long-term operational reliability and ease of troubleshooting [4].

1.8.4 Schedule Feasibility

- **Realistic Timeline Planning:** The project is structured around agile development methodology, with clearly defined sprints and milestones. This iterative process allows for continuous evaluation and timely adjustments, ensuring that the project remains on track.
- **Resource Availability:** With a dedicated team and established workflows in place, the allocation of necessary human and technical resources is feasible within the proposed timeline. Early prototyping and user feedback cycles help in identifying potential delays, thereby allowing for proactive schedule adjustments.
- **Milestones and Delivery:** The project roadmap includes key stages such as requirements analysis, system design, initial prototyping, testing phases, and the final deployment. Each stage has measurable deliverables, making it easier to track progress and ensure that all critical components are completed within the scheduled time frame.

1.9 Organization of the Document

The project documentation is structured to guide readers through a clear, logical progression of the system's conception, design, and development. The main components of the document are organized as follows:

1. **Preliminary Pages:**
    - **Cover Page:** Includes the project title, author details, and any essential project metadata.
    - **Abstract:** A concise summary of the project, highlighting the motivation, objectives, methodology, and key outcomes.
    - **Lists:** Separate sections for the list of figures, tables, and abbreviations/acronyms are provided to help readers quickly reference visual aids and technical terms.
2. **Chapter One – Introduction of the Project:** This chapter lays the groundwork for the entire project. It includes:

- **Motivation of the Project:** Explains the reasons behind the initiative and the problems with current systems.
- **Statement of the Problem:** Identifies the challenges that the project intends to address.
- **Objective of the Project:** Outlines the overall goal (general objective) and breaks it down into clear, specific objectives.
- **Methodology:** Describes the data collection methods, system development approach, and the tools used to build the system.
- **Scope of the Study:** Defines the boundaries and focus areas of the project.
- **Significance of the Study:** Discusses the benefits and potential impacts for both the airline industry and its customers.
- **Feasibility Assessment:** Reviews the economic, technical, operational, and schedule feasibility of the project.

3. **Chapter Two – System Requirement and Specification:** This chapter details what the system must achieve and how it compares with existing solutions. It covers:
   - **Overview of the Existing and Proposed Systems:** A comparative glance discussing current limitations and the envisioned improvements.
   - **Supplementary Specifications:** Includes functional and nonfunctional requirements, business rules, and operational constraints.
   - **Modeling and Prototyping:** Presents the use case models, CRC (Class Responsibility Collaboration) cards, essential and system use case descriptions, and user interface prototypes.
4. **Chapter Three – System Design:** Focusing on the architecture and structural foundations, this chapter includes:
   - **Layered Architecture:** An explanation of how the system is organized across different layers (presentation, business logic, data access).
   - **Class Modeling:** Detailed class diagrams illustrating relationships (inheritance, association, aggregation, composition, dependency) along with attributes and methods.
   - **Sequence, Activity, and Object Diagrams:** Visual representations mapping the interactions, workflows, and object relationships within the system.
   - **Persistent Modeling/Database Design:** A thorough depiction of the normalized physical database model for secure and efficient data management.
   - **User Interface and Deployment:** Outlines the design of the user interface for optimal user experience and describes the deployment strategy through component and deployment diagrams.
5. **References:** This section lists all works cited in the document following the IEEE referencing style, ensuring the sources are properly acknowledged [6].

CHAPTER TWO

System Requirement and Specification

The following sections detail the system requirements and specifications, providing a clear blueprint to transition from problem identification to an effective, scalable solution. This chapter covers the assessment of the existing system, outlines the envisioned improvements, and breaks down the detailed functional and nonfunctional aspects of the proposed Airline Reservation System [7].

2.1 Overview of the Existing System

Current airline reservation systems are often built on legacy infrastructure that suffers from fragmentation and outdated technologies. These systems typically involve multiple, disconnected modules that handle tasks such as flight scheduling, ticket booking, and customer data management separately. This disjointed approach leads to several challenges:

- **Inefficiency in Processing:** Many existing systems rely heavily on manual data entry and routine paperwork, which results in slow booking processes and increased likelihood of human error. The lack of automation often translates into delayed responses during peak booking times.
- **Limited Integration and Data Redundancy:** Since various system components operate in isolation, data synchronization between flight scheduling, booking, and payment modules is poor. This can result in duplicated information and discrepancies that complicate management tasks.
- **User Interface Limitations:** The user interfaces of these older systems are typically not designed with modern usability principles. Customers and airline staff alike encounter clunky navigation, non-responsive design elements, and a steep learning curve when interacting with the system.
- **Security Vulnerabilities:** Older platforms generally do not incorporate the advanced security protocols necessary in today's digital environment. This exposes sensitive customer information and financial transactions to greater risk of data breaches.
- **Scalability Concerns:** As passenger volumes increase, these systems struggle to cope with the additional load. Their rigid architectures limit the ability to scale quickly or efficiently, potentially leading to performance bottlenecks during high-demand periods.

2. 2 Overview of the Proposed System

The proposed Airline Reservation System is designed to overcome the limitations of current platforms by delivering a fully integrated, modern, and secure digital solution. This new system will streamline the entire flight booking process while enhancing both the customer and operational experience. Key features of the proposed system include:

- **Real-Time Flight Management:** Leveraging dynamic data capabilities, the system will provide real-time updates on flight schedules and seat availability. Users can perform instant flight searches and receive immediate feedback, ensuring a seamless booking experience.
- **Integrated Booking Process:** The new system unifies all aspects of flight reservations—from initial search and ticket booking to seat selection and payment processing—into a single platform. This integration minimizes redundancy and reduces the risk of errors, supporting a smooth lifecycle for each booking.
- **User-Friendly Interface:** The system will feature a modern, responsive design that caters to both desktop and mobile users. By adopting contemporary UI/UX principles, the platform ensures intuitive navigation, clear visual cues, and a reduced learning curve for users, making it accessible to a wide range of passengers.
- **Advanced Security Measures:** Recognizing the importance of protecting sensitive personal and financial data, the proposed system implements state-of-the-art encryption and secure payment gateways. This will mitigate the risks of data breaches and build stronger trust with users.
- **Scalability and Operational Efficiency:** Built on a modular, multi-tiered architecture, the system is designed to scale with increasing demand. Its flexibility not only supports current operational requirements but also allows for future enhancements and easier maintenance, thereby reducing long-term administrative overhead [5].

2.3 Supplementary Specification

To ensure the Airline Reservation System meets both operational needs and quality standards, this section outlines the supplementary specifications that govern its functionality and overall performance.

2.3.1 Functional Requirements

These are the specific capabilities the system must provide:

- **User Authentication & Profile Management:**
    - **Registration and Login:** Allow users to create accounts and securely authenticate using email/password or multi-factor authentication.
    - **Profile Management:** Enable users to view and update personal details (e.g., contact information, travel preferences).
- **Flight Search & Scheduling:**
    - **Dynamic Flight Search:** Provide an interface for users to search for flights by date, destination, and price range.
    - **Real-Time Flight Data:** Ensure that flight schedules and seat availability are updated in real time.

- **Booking and Reservation Management:**
    - **Seat Selection:** Offer interactive seat maps that let passengers choose specific seats.
    - **Booking Confirmation:** Automatically generate booking confirmations along with relevant details (ticket number, itinerary).
    - **Change/Cancellation:** Provide options for users to modify or cancel bookings as per airline policies.
- **Payment Processing:**
    - **Secure Transactions:** Integrate with trusted payment gateways to process credit/debit card payments and digital wallets securely.
    - **Receipt Management:** Automatically issue transactional receipts and maintain a log of payment history.
- **Administrative Functions:**
    - **Flight Management:** Allow airline staff to add, update, or remove flight details and schedules.
    - **Reporting and Analytics:** Generate reports on booking trends, revenue, and user demographics.
    - **User Management:** Facilitate role-based access control for staff to manage reservations and customer queries.

2.3.2 Nonfunctional Requirements

These criteria define the overall quality attributes of the system:

- **Performance:**
    - **Responsiveness:** The system must ensure quick response times during searches and booking transactions, even under peak loads.
    - **Scalability:** Designed to support increasing numbers of users and data volume without performance degradation.
- **Reliability and Availability:**
    - **Uptime:** The system should maintain high availability with minimal downtime for maintenance.
    - **Fault Tolerance:** Implement backup and recovery procedures to ensure data integrity and system continuity in case of failures.
- **Security:**
    - **Data Protection:** Incorporate end-to-end encryption for user data and secure transaction processing.

    - **Access Control:** Enforce strict access policies to protect sensitive data from unauthorized access.
    - **Compliance:** Adhere to industry standards and regulations (e.g., GDPR, PCI-DSS).
- **Usability:**
    - **Intuitive Interface:** Ensure the user interface is user-friendly and consistent across desktop and mobile platforms.
    - **Accessibility:** Design interfaces to be accessible to users with disabilities.
- **Maintainability and Extensibility:**
    - **Modular Architecture:** Develop the system in modular components to simplify future updates and feature enhancements.
    - **Documentation:** Provide comprehensive documentation for both users and developers.

2.3.3 Business Rules

These rules govern how the Airline Reservation System will operate in a real-world context:

- **Dynamic Pricing:**
    - Flight fares may vary based on demand, booking timing, and seat class. Pricing algorithms will adjust rates dynamically.
- **Cancellation and Refund Policies:**
    - Clearly define conditions under which bookings can be canceled and how refunds are processed, following industry practices.
    - Implement penalty rules for last-minute cancellations as per airline guidelines.
- **Loyalty Programs:**
    - Integrate mechanisms for tracking frequent flyer points and enabling reward redemptions within the booking process.
- **Compliance with Airline Regulations:**
    - Ensure that all functional operations adhere to the operational policies and regulatory requirements specific to the aviation industry.

2.3.4 Constraints

These limitations affect the design and implementation of the system:

- **Technical Constraints:**
    - **Integration Limitations:** The system's capabilities depend on the availability and compatibility of third-party APIs (e.g., payment gateways, real-time flight data providers).

- **Infrastructure Dependence:** The performance of the system may be influenced by the underlying hardware and hosting environment.

- **Regulatory Constraints:**
  - **Data Protection Regulations:** The system must comply with data protection laws such as GDPR, affecting how user data is stored and processed.
  - **Industry Standards:** Adherence to standards (e.g., PCI-DSS for payments) may necessitate additional security measures.
- **Resource Constraints:**
  - **Budget:** Limited financial resources may restrict the scope of initial features and affect long-term scalability.
  - **Time:** Development and deployment must align with project deadlines, influencing the extent and complexity of features included in the first iteration.

2.4 CRC (Class Responsibility Collaboration)

Class Responsibility Collaboration (CRC) is an **object-oriented design** method used to structure system components efficiently. It identifies **key classes, their responsibilities, and the collaborations** between them to ensure modular and maintainable system design.

Key Classes and Their Roles

Passenger <<actor class>>	
Manage user authentication and profile. Search for flights based on criteria. Make bookings and payments.	Booking Class (to reserve tickets). Payment Class (to process transactions). Notification Class (to receive booking confirmations).

Flight <<in business class>>	
Store flight details (destination, schedule, price, seat availability). Provide real-time flight status updates. Ensure seat allocation upon booking.	Booking Class (to link flights with reservations). Admin Class (to modify schedules). Flight Database Class (to store flight information).

Flight <<in business class>>	
Handle flight reservations. Validate seat availability before confirmation. Generate booking details.	Passenger Class (to receive booking requests). Flight Class (to verify seat availability). Payment Class (to process payments)

Payment <<in business class>>	
Process transactions securely through external payment gateways. Validate payment details before approval. Handle refunds and cancellations.	Booking Class (to finalize reservations). Payment Gateway Class (to process transactions).

Admin<<actor Class>	
Manage flight schedules and seat availability. Generate reports on bookings and payments. Assist passengers with inquiries or changes.	Flight Class (to update flight details). Booking Class (to manage reservations).

Notification<< business Class>>	
Manage flight schedules and seat availability. Generate reports on bookings and payments. Assist passengers with inquiries or changes.	Flight Class (to update flight details). Booking Class (to manage reservations).

Figure 1: **CRC** diagram.

2.5 **Use Case Modeling**

Use Case Modeling represents the **interaction between users and the system** by defining the **essential functionalities**, **system operations**, and expected user behavior

2.5.1 Essential Use Case Modeling

Essential Use Case Modeling focuses on **what the user wants to achieve** (user goals) without describing how the system does it. It's **technology-independent** and describes **interactions at a high level**.

Each essential use case typically includes:

I. **Actor**: Who is using the system
II. **Goal**: What they want to accomplish
III. **Steps**: High-level interaction (e.g., "User selects destination and date

Essential Use case modeling graph.

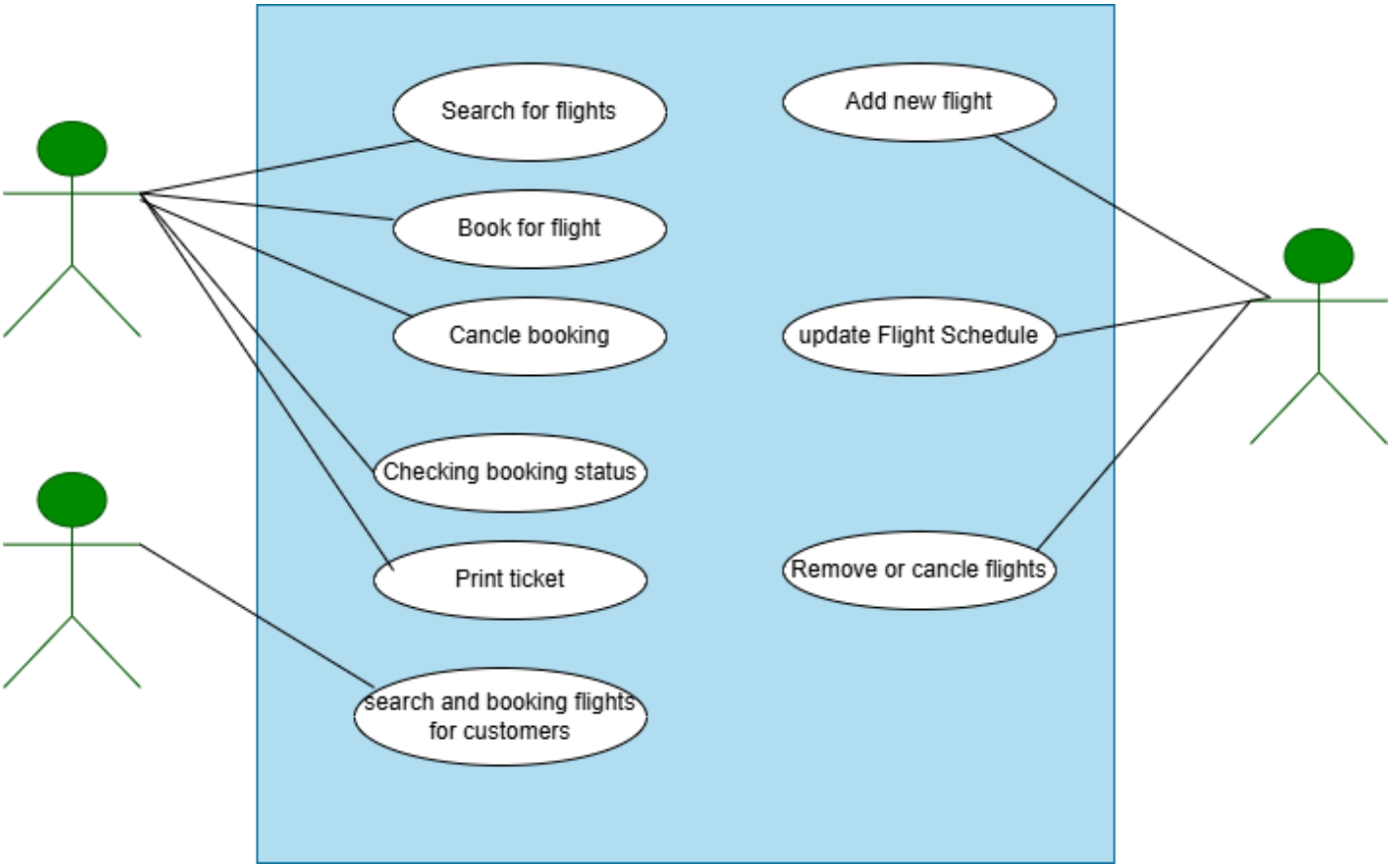

figure 2.1: Essential use case modelling

2.5.2 Essential Use Case Description

An **Essential Use Case** is a **simplified and abstract** version of a use case. It avoids technical system details and focuses on **what the user wants to do** and the **system's responsibilities**. It answers the **what**, not the **how**

Use Case	Actor	Precondition	Main Success Scenario	Postcondition	Alternative Flows
1. Search for Flights	Passenger	Passenger has access to the system and wants to search	- Enters flight details - System retrieves matching flights - Passenger selects a flight	Passenger proceeds to booking	- System suggests other dates if none found - Prompts for correct input if invalid

2. Book a Flight	Passenger	Passenger has selected a flight	- Enters passenger info - Selects seat and payment - System confirms booking	Booking is confirmed	- Retry payment on failure - Choose different seats if unavailable
3. Cancel Booking	Passenger	Passenger has an existing booking	- Opens booking - Requests cancellation - System processes it - Receives confirmation	Flight is canceled and seat is released	- Informs if booking is non-refundable
4. Check Booking Status	Passenger	Passenger has an active booking	- Logs in - Navigates to booking history - System shows booking and status	Passenger sees current flight status	N/A
5. Print Ticket	Passenger	Passenger has a confirmed booking	- Opens booking - Selects “Print Ticket” - System generates printable version	Passenger gets a soft/hard copy of ticket	N/A
6. Search & Book for Customers	Travel Agent	Authorized to book for customers	- Searches flights - Inputs customer details - Confirms and pays	Flight is booked for customer	- Retry or offer alternatives if payment fails
7. Add New Flight	Admin	Authorized to manage flight schedules	- Accesses panel - Inputs flight info (route, capacity, price) - Saves flight	New flight is listed for booking	N/A
8. Update Flight Schedule	Admin	Flight exists in system	- Opens flight panel - Updates schedule - Changes saved and passengers notified	Schedule updated in database	N/A
9. Remove or Cancel Flights	Admin	Flight exists in system	- Selects flight - Checks for reservations - Notifies passengers - Removes flight from system	Flight is removed or canceled	- Offers refunds/rebooking if there are booked passengers

Table 1: description of Essential use case

2.5.3 System use case modeling

System Use Case Modeling describes **how the system actually behaves** to fulfill user goals. It includes **system responses, technical interactions**, validations, and external systems.

Each system use case includes:

I. **Actor**: Human or external system
II. **System behavior**: What the system does step-by-step
III. Preconditions, post conditions, alternative flows, etc.

System use case diagram for Airline reservation system.

Figure 2.2 : System use case diagram

2.5.4 System use case Description

A **System Use Case** describes the interaction between a **user (actor)** and the **system** to achieve a specific goal. It includes detailed steps like inputs, system responses, and alternative flows. It focuses on **how** the system works and reacts.

Use Case	Actor	Precondition	Main Success Scenario	Postcondition	Alternative Flows
1. Login/Register	Passenger	Access to the system	- Enter credentials or register - System validates - Access dashboard	Passenger is authenticated	- Retry on incorrect login - Offer registration if unregistered
2. Search Available Flights	Passenger	Logged into the system	- Enter search criteria - System queries DB - View and select flight	Passenger proceeds to booking	- Suggest alternate dates/routes if no flights found
3. View Flight Details	Passenger	Has searched for flights	- Selects a flight - System displays schedule, price, availability	Passenger confirms flight	- Alert admin if flight data is missing
4. Select Flight & Enter Details	Passenger	Selected a flight	- Choose seat - Enter personal details - System saves	Booking is ready for payment	- Prompt for missing info
5. Make Payment	Passenger, Payment Gateway	Booking details are entered	- Choose payment method - Process via gateway - Confirmation sent	Payment successful, booking confirmed	- Retry if payment fails
6. Receive Booking Confirmation	Passenger, Email Server	Flight booked successfully	- Generate confirmation - Email/SMS sent	Confirmation received	- Offer on-screen download if email fails
7. Cancel Booking	Passenger	Has an existing booking	- Request cancellation - Verify refund - Confirm cancellation	Seat released	- Notify if non-refundable
8. View/Print Ticket	Passenger	Has a confirmed booking	- Access booking - Choose print/download	Ticket obtained	- Offer alternate download method if errors occur
9. Bulk Booking for Customers	Travel Agent	Authenticated in the system	- Search flights - Select multiple flights - Enter multiple	Multiple customers booked	- Retry payment if it fails

			details - Confirm booking and payment		
10. Login to Admin Panel	Admin	Has admin credentials	- Enter login info - System verifies and grants access	Admin gains management access	- Offer password reset on login failure
11. Add/Edit/Delete Flights	Admin	Logged in	- Input/edit/delete flight data - System updates database	Flight DB updated	- Prompt correction if data is invalid
12. View Booking Reports	Admin	Logged in	- Access reporting tools - Generate reports on bookings and revenue	Admin reviews analytics	- Suggest data refresh if incomplete
13. Check Seat Availability	Passenger	Selecting a flight	- System checks seat map - Passenger picks preferred seat	Seat reserved	- Suggest alternatives if seat is unavailable
14. Generate Ticket & Booking ID	System	Flight booked successfully	- Assign booking ID - Generate and send e-ticket	Booking and ticket finalized	- Retry ticket generation on failure

Table 2: system use case table

2.6 User Interface Prototyping

User Interface Prototyping is an essential process in system design. It helps to visualize the layout, interactions, and workflows before the actual implementation [7].

2.6.1 Traditional User-Interface Prototyping

Traditional UI prototyping involves creating **static visual representations** of system interfaces. These representations define **page layouts, user interactions, and basic designs** without functionality

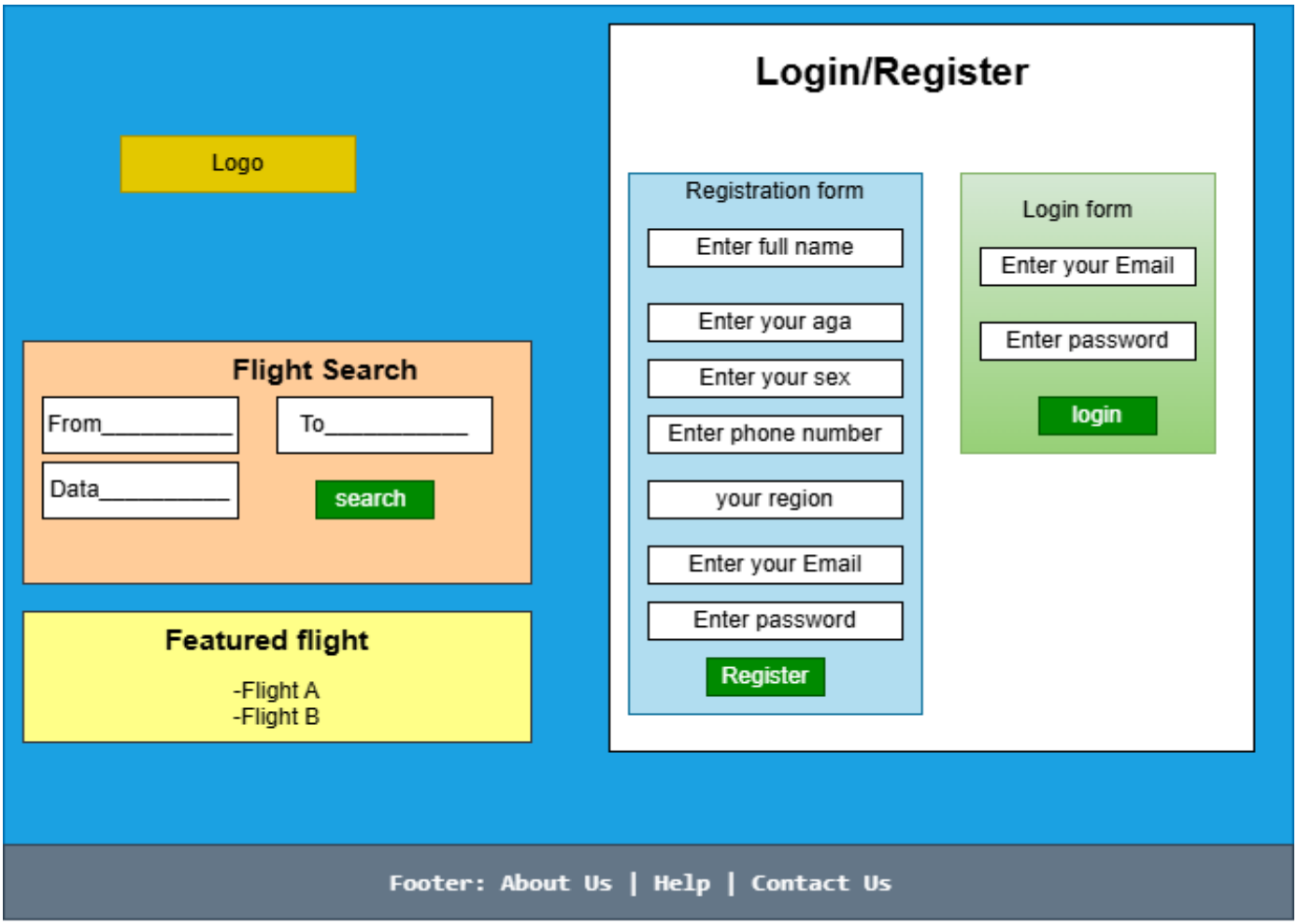

Figure 2.3 Traditional user interface prototype

2.7 User-Interface Flow Diagramming

User-Interface Flow Diagrams illustrate **how users navigate through different screens** in the system. This ensures logical **interaction and usability**.

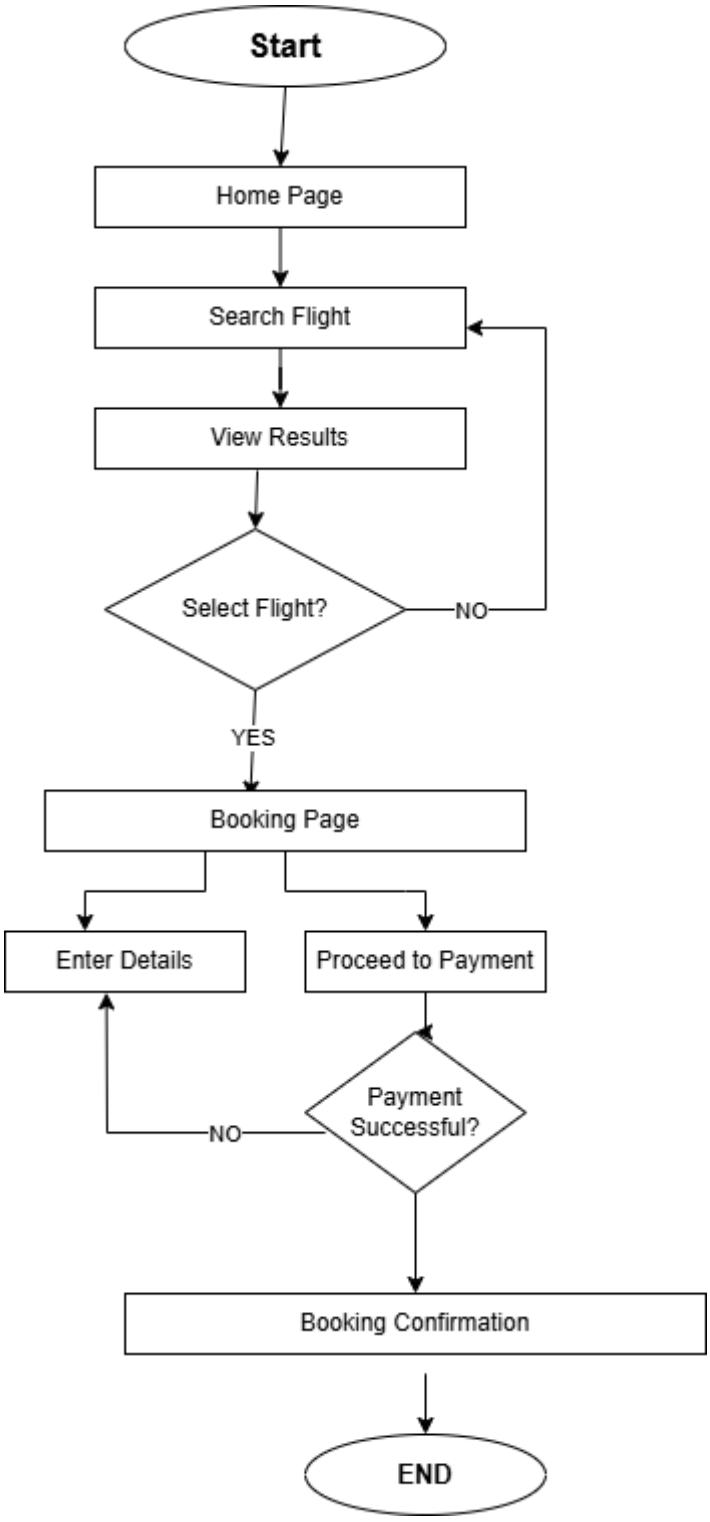

Figure 2.4 User inerface flow diagram

CHAPTER THREE

System Design

System design is the process of defining the architecture, components, interfaces, and data flow of a system to meet specific requirements. It involves structuring functional and non-functional aspects, ensuring efficiency, scalability, and reliability in system operation.

3.1 Layered Architecture

The Layered Architecture ensures modular design and separation of concerns in the Airline Reservation System. It divides the system into distinct layers, each handling specific responsibilities.

The **Presentation Layer (User Interface)** handles interaction with passengers, travel agents, and admins. It includes login/register, flight search, booking, payment, and confirmation pages. Technologies used in this layer include HTML, CSS, JavaScript, React, and Vue.js.

The **Business Logic Layer (Application Processing)** processes flight reservations, payment transactions, and cancellations. It includes booking validation, pricing calculation, and refund processing. Technologies used in this layer include Java, Python, .NET, and Node.js.

The **Data Access Layer (Database Communication)** connects the system with flight, booking, and customer databases. It manages queries, updates, deletions, and data retrieval. Technologies used in this layer include SQL, NoSQL (such as MongoDB and Firebase), and ORMs like Hibernate and Entity Framework.

The **Database Layer (Storage)** stores flight schedules, seat availability, booking history, and payment transactions. It ensures data integrity, security, and backup. Technologies used in this layer include Microsoft SQL Server, PostgreSQL, MySQL, and Oracle DB.

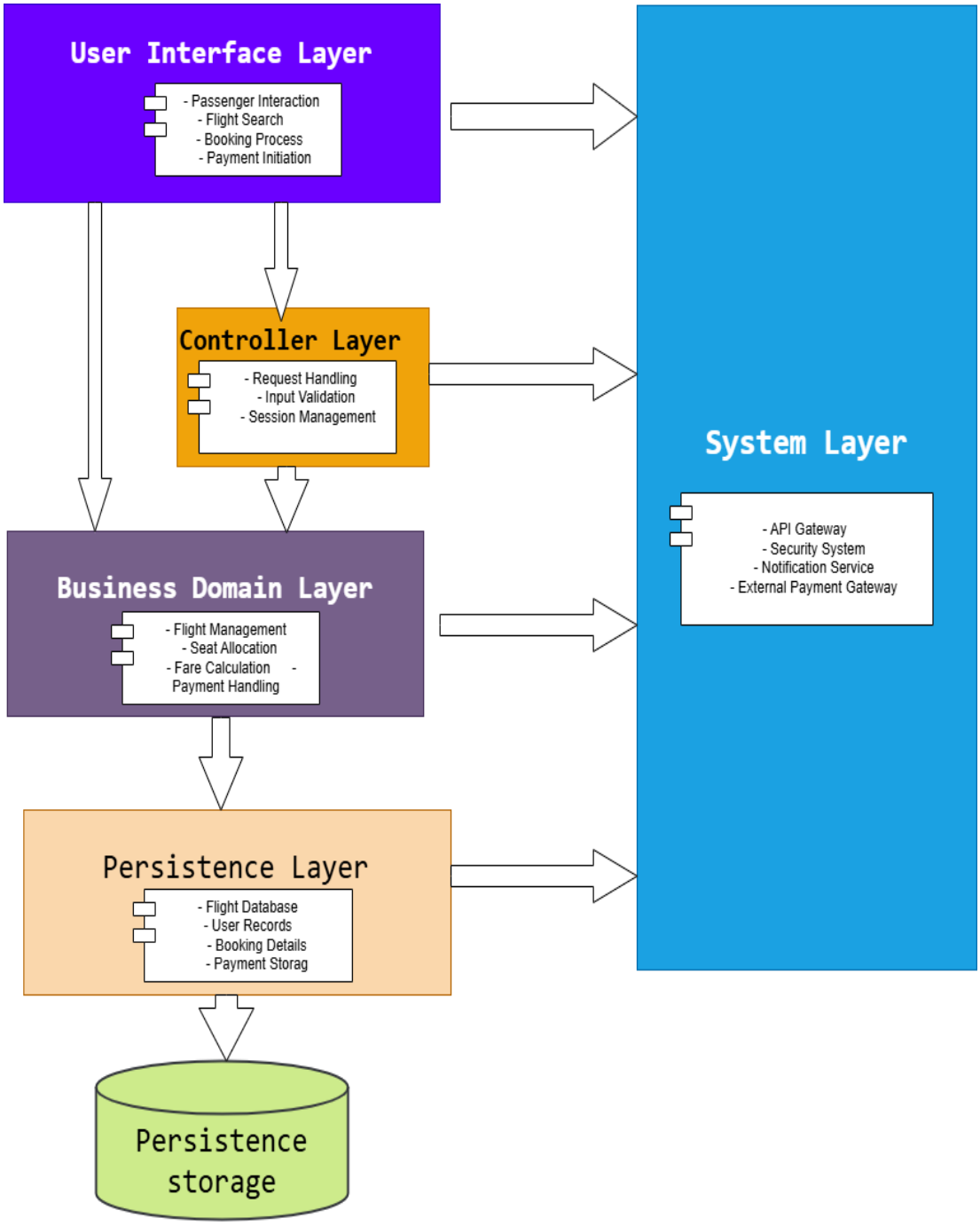

Figure 6: Layered Architecture diagram

3.2 Class modeling

Class modeling is an essential part of system design, defining **objects, their relationships, attributes, and behaviors** to ensure an efficient software architecture.

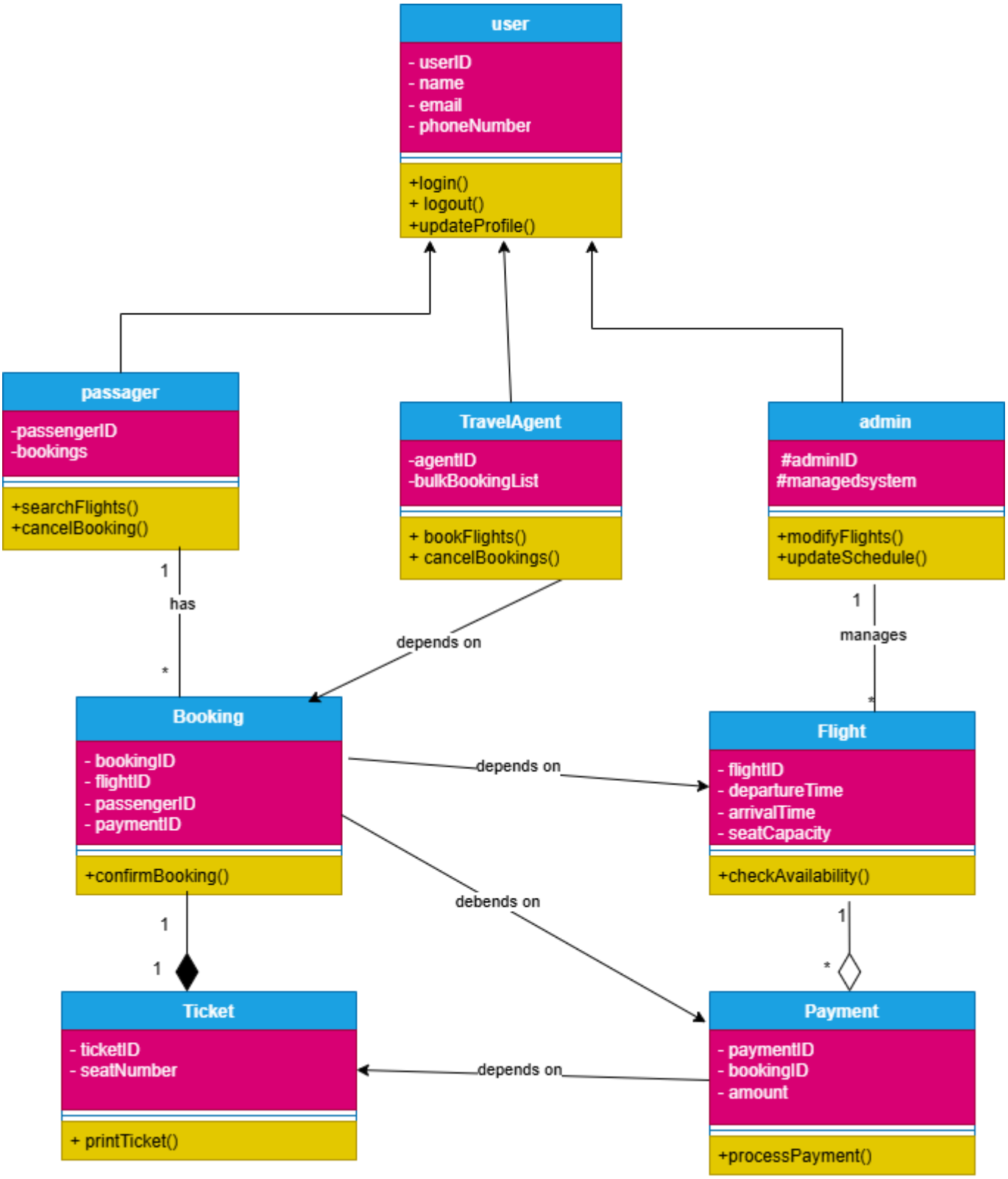

Figure 3.2: class modelling Diagram

3. 3 Sequence Diagram for Airline Reservation System

A **Sequence Diagram** is a UML representation that illustrates the **step-by-step interaction** between system components and users in **chronological order**. It highlights the **messages exchanged** between objects during a specific process, such as **flight booking** or **payment processing**.

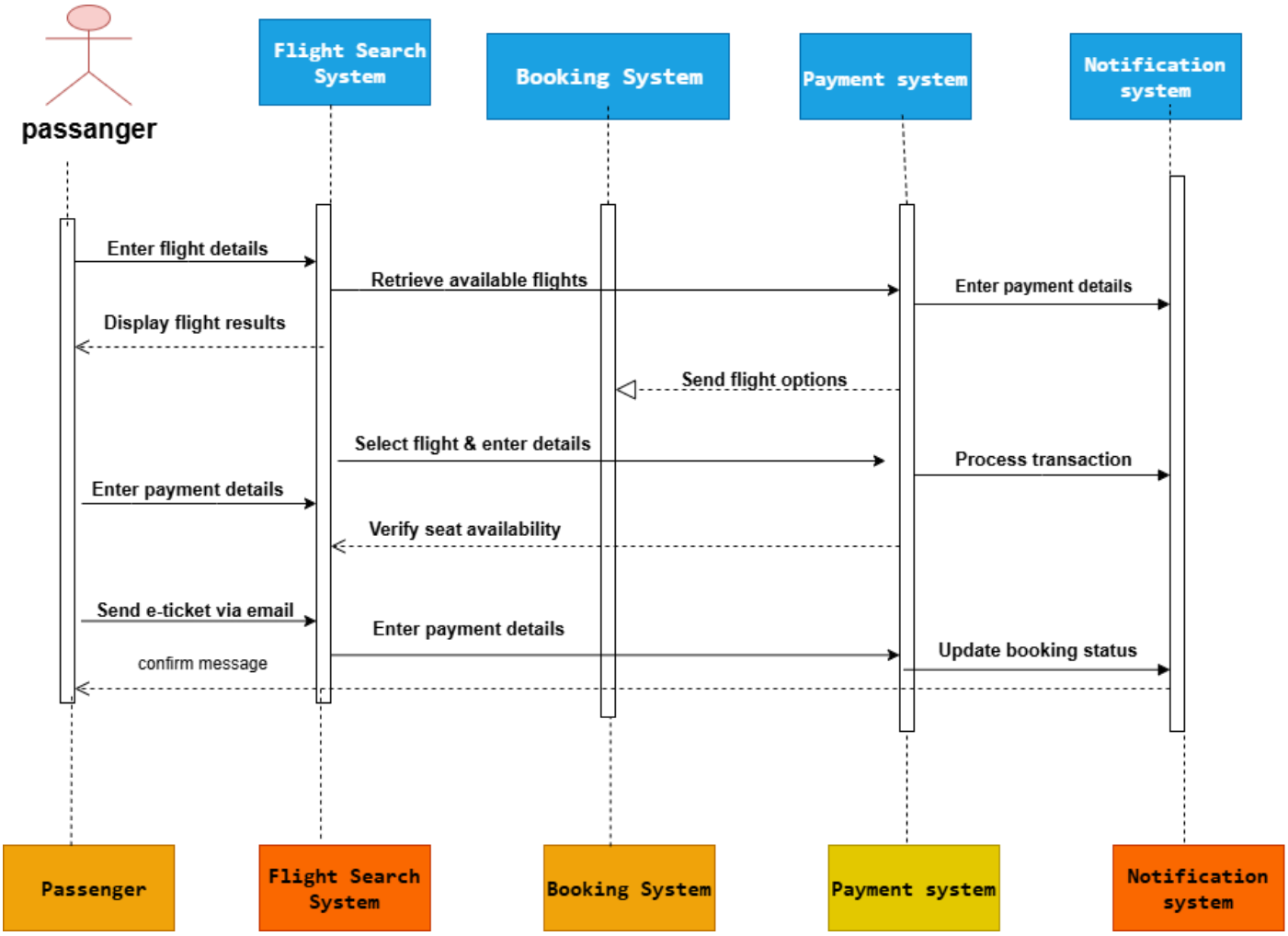

Figure 3.3: sequence Diagram

3.4 Activity Diagram

Activity diagram is a UML diagram used to model **workflow logic** by showing **user actions, system processes, and decision points**. It visualizes how a task **progresses step by step**, including conditional branching and parallel operation.

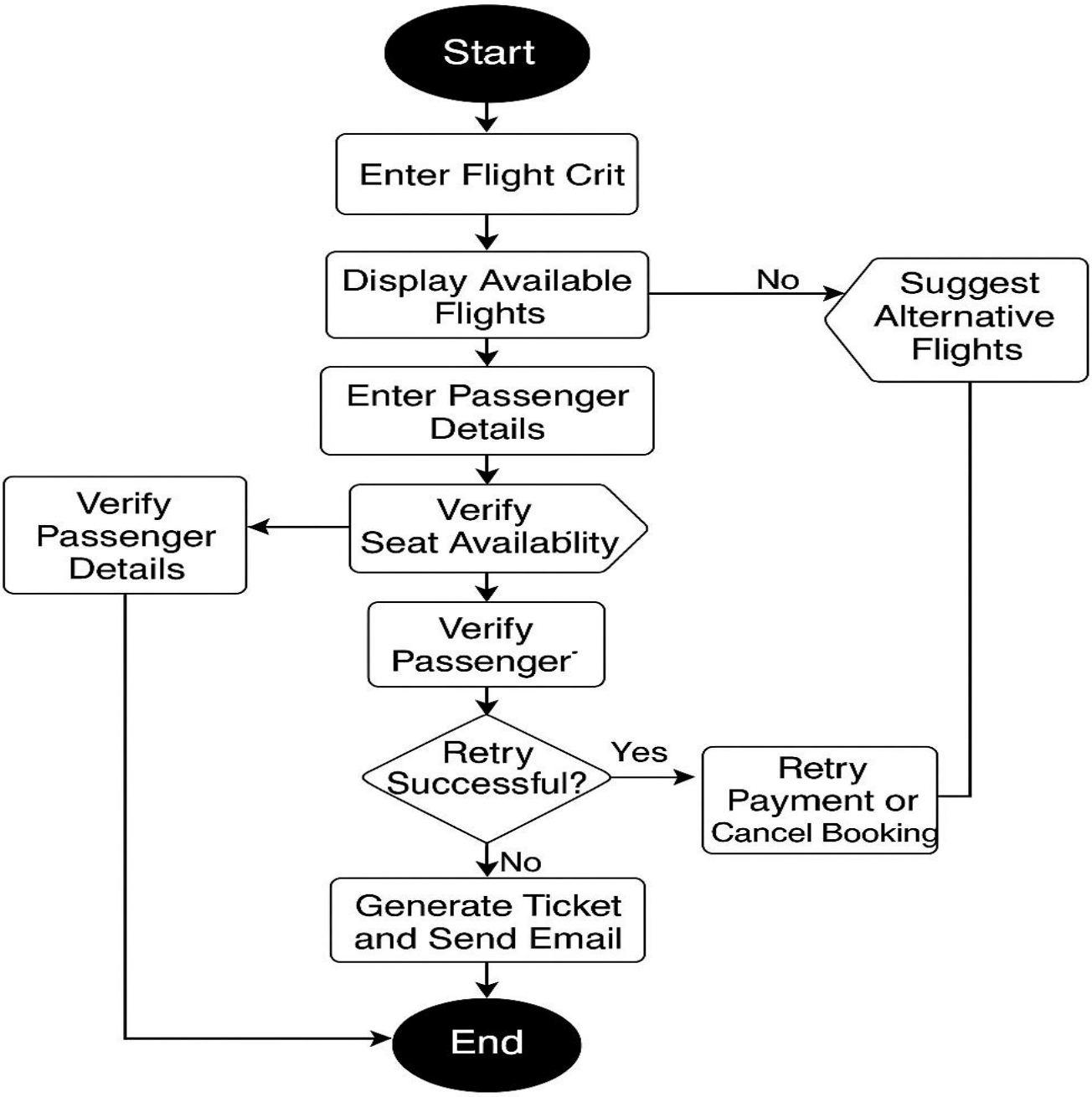

Figure 3.4: Activity Diagram

3.5 Object Diagram for Reservation System

An **Object Diagram** represents **real-time instances** of system objects, showing how objects relate and interact **at a specific moment**. It provides insights into **object attributes, states, and relationships**, helping validate **class structures and dependencies** in the system

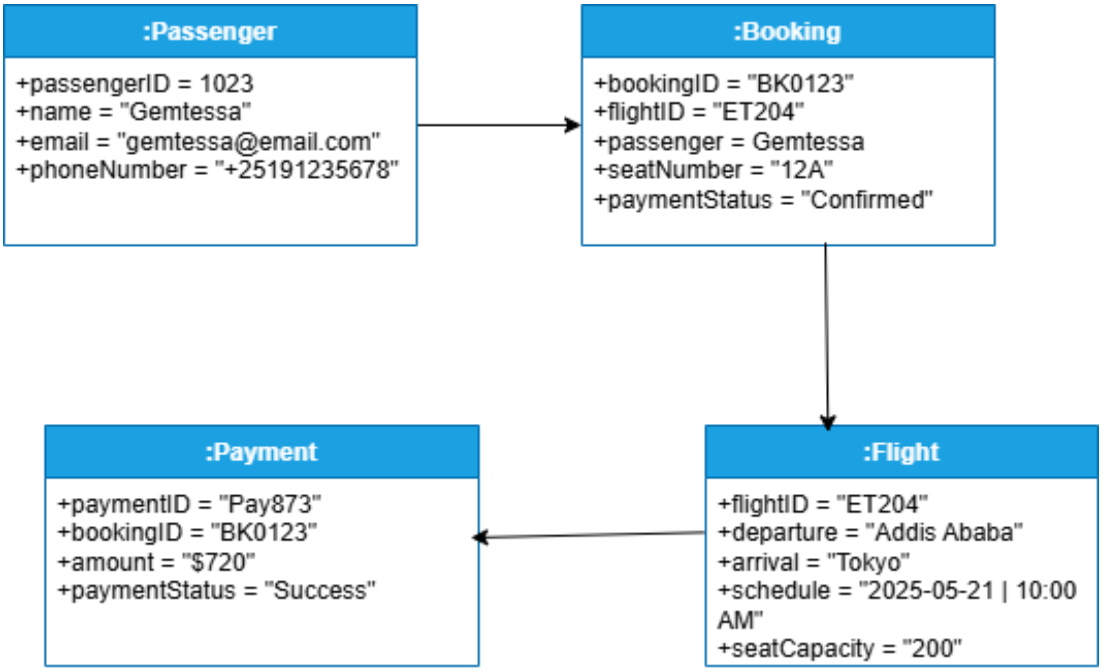

Figure 3.5 object diagram

3.6 Persistent Modeling / Database Design

Persistent modeling ensures **efficient data storage** while maintaining **accuracy, security, and scalability**. It structures data for **long-term consistency** in systems like **airline reservations**, ensuring smooth retrieval and modifications.

- **Normalized Physical Database Model:** A **Normalized Physical Database Model** eliminates **data redundancy** by organizing information into structured tables using **normalization techniques (1NF, 2NF, 3NF)**. It improves **data integrity, efficiency, and query performance** in large-scale applications.

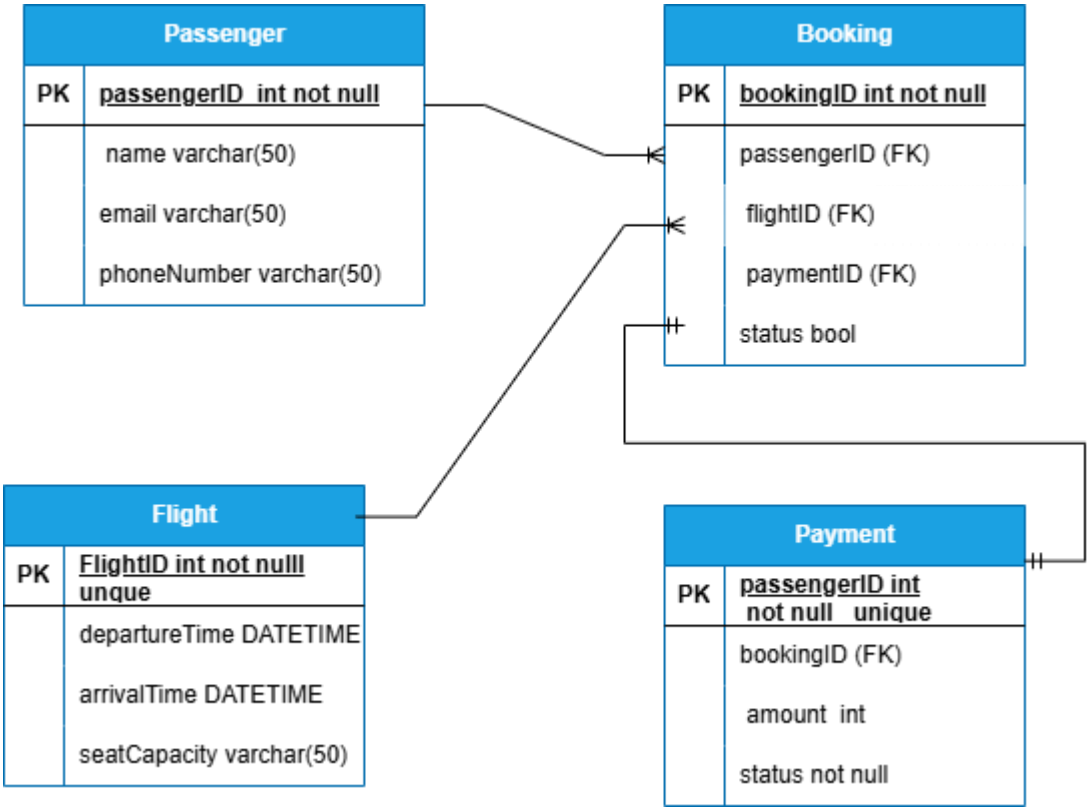

Figure 3.6 Database Diagram

3.7 User Interface Design

User Interface (UI) Design focuses on the visual layout and interaction that the user has with the system. It ensures a smooth, intuitive, and user-friendly experience.

Purpose in Airline Reservation System:

- Allows users to **search flights**, **book tickets**, and **make payments**.
- Offers administrators the ability to **manage flights** and **view booking reports**.

Key Elements:

- **Header**: Company logo, login/register buttons.
- **Flight Search Form**: "From", "To", "Date", and Search button.
- **Available Flights**: Table listing flight ID, destination, price, and a select option.
- **Booking Section**: User input for name, passport number, and payment method.
- **Admin Dashboard** (optional): For flight management by staff.
- **Footer**: Links to About Us, Help, and Contact Us.

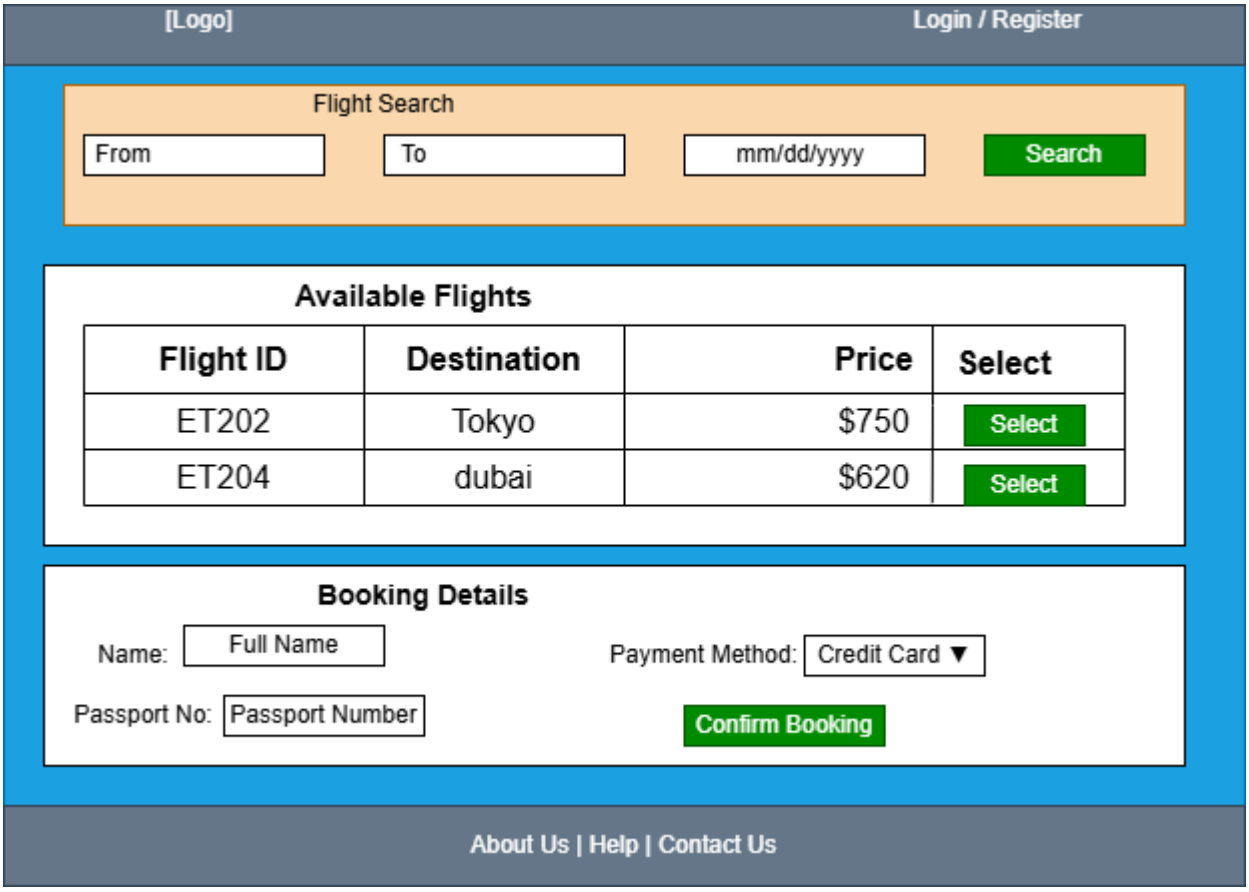

Figure 3.7 user interface design

3.8 Component Diagram

component diagram shows the **software structure** of the system in terms of **logical components/modules components/modules** and how they **interact** with each other.

Main Components:

1. **User Interface (UI)**
   - Handles user input and display
   - Example: Login form, flight search form
2. **Business Logic Layer**
   - Manages rules for searching flights, booking tickets, and payment processing
   - Example: FlightValidator, BookingProcessor
3. **Data Access Layer**
   - Communicates with the database
   - Example: FlightDatabase, PaymentGateway

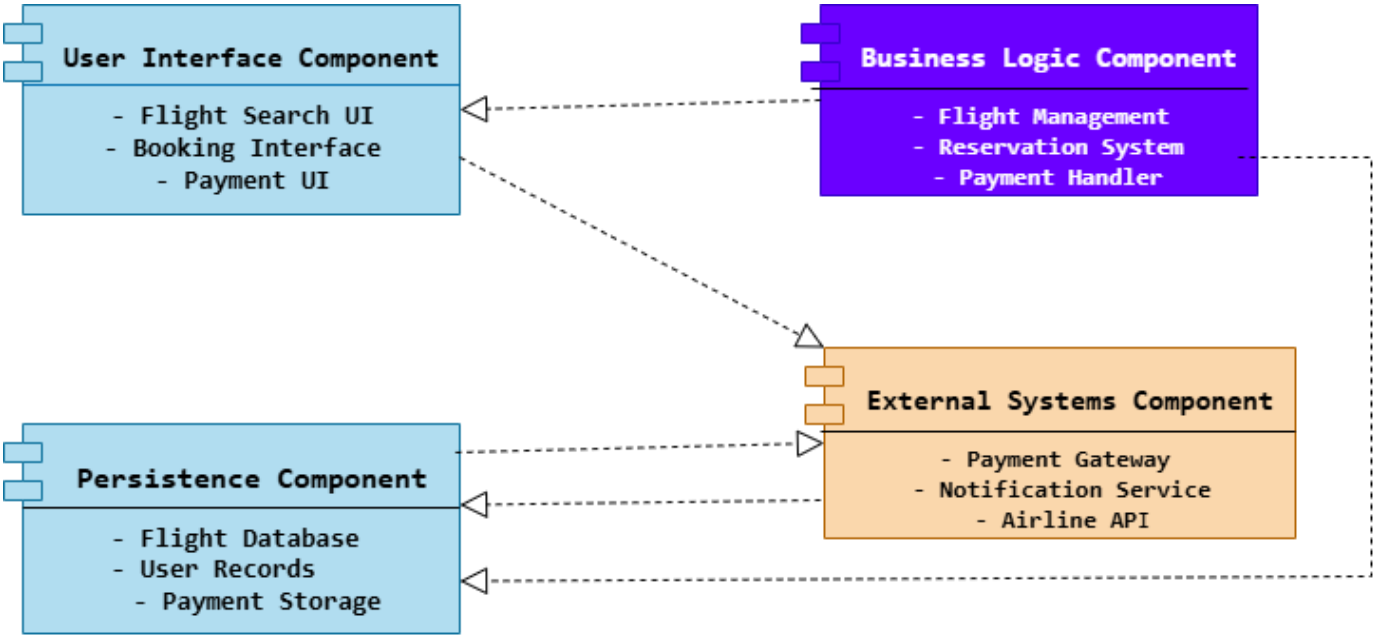

Figure 3.8 component diagram

3.9 Deployment Diagram

A deployment diagram shows the **hardware infrastructure** of the system and how **software components are deployed** on these hardware nodes.

Its Purpose

To represent how the system is **physically deployed** especially useful for understanding **network setup** and **server distribution**.

- **Deployment Example:**

1. **Passenger Web App**
   - Installed on the user's browser/device
2. **Application Server**
   - Hosts business logic and APIs (e.g., booking, payment)
3. **Database Server**
   - Stores flight schedules, passenger data, booking records

deployment diagram

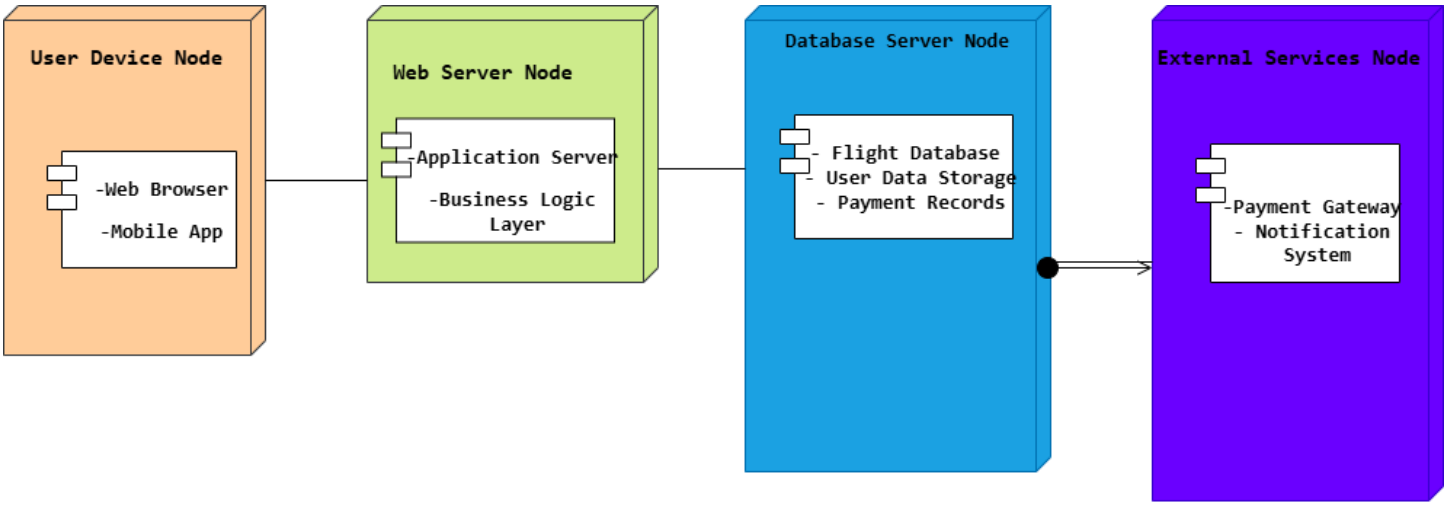

Figure 3.9: deployment Diagram

3.10 References

[1] Wikipedia Contributors, "Airline Reservations System," *Wikipedia.*

[2] Kaavya Kuppa, "References Airline Reservation System," *Kansas State University.*

[3] Damian Ojimba, "Airline Reservation System Project Documentation," 2017.

[4] nternational Air Transport Association (IATA), Computer Reservation Systems and Airline Ticketing, IATA Publications, 2005.

[5] Amadeus IT Group, "Airline Reservation Systems: Evolution and Future Trends," Amadeus Research Papers, 2018.

[6] Sabre Corporation, "Global Distribution Systems and Airline Reservations," : Sabre Research Papers, 2018.

[7] Dr. Daniel Andresen, Dr. Torben Amtoft, Dr. Mitchell L. Neilsen, Airline Reservation System: A Software Engineering Approach, Kansas State University, 2012.