DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi 110042

DEPARTMENT OF SOFTWARE ENGINEERING

Lab Code: 209
Software Engineering Methodologies
Lab File

Submitted To
Mr. Ankur

Submitted by
TANMAY KUMAR
23/SE/156
TANMAY KUMAR
23/SE/157
TEJAS TOMAR
23/SE/158
THANGKHANMUAN
23/SE/159

INDEX

S.NO	EXPERIMENT	DATE	SIGNATURE
1.	Problem Statement on Airline Management System		
2.	Use Case Diagram of Airline Management System		
3.	Use Case Description of Airline Management System		
4.	Data Flow Diagram for Airline Management System		
5.	Entity Relationship Diagram for Airline Management System		
6.	Software Requirement Specification for Airline Management System		
7.			
8.			
9.			
10.			
11.			
12.			
13.			

TOPIC

1. Problem Statement
    - 1.1 Objectives
    - 1.2 Introduction
    - 1.3 Theory
    - 1.4 Key Components

2. Case Diagram
    - 2.1 Objectives
    - 2.2 Introduction
    - 2.3 Theory
    - 2.4 Use Case Diagram
        - Flight Management.
        - Booking and Ticketing.
        - Passenger Management.
        - Fleet Management.
        - Customer Support.
        - Security Management.

3. Use Case Actors Description
    - 3.1 Objectives
    - 3.2 Introduction
    - 3.3 Theory
        - Flight Management.
        - Booking and Ticketing.
        - Passenger Management.
        - Fleet Management.
        - Customer Support.
        - Security Management.

4.Data Flow diagram DFD

- 4.1 Objectives
- 4.2 Introduction
- 4.3 Theory
- 4.4 Process Flow and Actors
    - **Level 0 DFD (Context Diagram)**
    - Level 1 DFD
    - Level 2 DFD

5. Entity relationship Diagram (ER)
    - 5.1 Objectives
    - 5.2 Introduction
    - 5.3 Theory
    - 5.4 Key Components
    - 5.5 ER Diagram

6. SRS
    - 6.1 Objectives
    - 6.2 Introduction
    - 6.3 Theory
    - 6.4 Key Component
    - 6.5 Overall Description
    - 6.6 Specific Requirement
        - Functional Requirement.
        - Non-Functional Requirement.
    - 6.7 System Module
    - 6.8 Appendices
    - 6.9

EXPERIMENT-1

Objectives

Write the Problem Statement for Airline Management System.

Introduction

We are going to identify the problem system for airline management system.

Theory

Background:

In the competitive aviation industry, airlines face numerous challenges in managing operations efficiently, ensuring customer satisfaction, and maintaining profitability. The traditional methods of managing flight schedules, bookings, and customer data are often plagued by inefficiencies, inaccuracies, and delays. As the industry grows, the need for a comprehensive, scalable, and user-friendly Airline Management System (AMS) has become increasingly critical.

Problem:

Airlines struggle with managing multiple facets of their operations, including flight scheduling, ticket bookings, passenger management, and maintenance tracking. The lack of an integrated system leads to issues such as double bookings, flight delays, underutilized capacities, and inadequate customer service. Furthermore, the absence of real-time data and analytics hampers decision-making, resulting in suboptimal performance and loss of revenue.

Objective:

The objective of the Airline Management System is to develop a robust, scalable, and efficient software platform that addresses key operational and customer-facing processes in the airline industry. The AMS should ensure that airline operations are efficient, safe, and customer-focused, while leveraging real-time data and analytics to improve decision-making and optimize resource usage. The system should integrate functionalities for flight management, booking and ticketing, passenger management, fleet management, real-time analytics, and customer support.

Key Components

1. Flight Management:
    a. Automation of Flight Scheduling and Tracking:

The system should automate the process of flight scheduling, tracking flight statuses in real-time, and ensuring that information about delays, cancellations, or rescheduling is promptly available to all stakeholders.

    b. Real-Time Flight Status Updates:

Provide real-time updates on flight delays, cancellations, and rescheduling through online platforms, SMS, or mobile apps, ensuring passengers are always informed and airlines can quickly respond to operational changes.

2. Booking and Ticketing:
    a. Seamless Customer Experience:

The booking system should allow customers to search for flights, select seats, make payments, and receive booking confirmations quickly and easily. The platform should integrate with payment gateways and support multiple currencies and payment options.

    b. Managing Reservations and Modifications:

Passengers should be able to modify their reservations, cancel tickets, and request refunds through a user-friendly interface, ensuring a smooth customer experience. The system must handle these processes without errors or overbooking issues.

    c. Prevention of Overbooking:

The system should prevent overbooking by automatically tracking seat availability, considering multiple sales channels (online, call centers, and travel agents), and updating the central database in real-time to avoid errors.

3. Passenger Management:
    a. Accurate Passenger Records:

Maintain an up-to-date database of passenger information, including personal details, booking history, preferences (e.g., seat preference, meal options), and special requirements (e.g., assistance for disabled passengers).

    b. Personalized Services:

Use passenger information to provide personalized services, such as targeted marketing, loyalty program management, and tailored in-flight services. The system should also comply with industry regulations regarding data protection and passenger privacy.

    c. Compliance with Regulations:

Ensure compliance with global and regional airline regulations, including passenger manifest reporting, customs documentation, and adherence to safety protocols like passenger identification and screening procedures.

4. Fleet Management:

    a. Aircraft Maintenance Scheduling:

The system should track the maintenance schedules of all aircraft in the fleet, ensuring timely inspections, repairs, and part replacements to maintain safety and regulatory compliance.

b. Crew Assignment and Management:

Efficiently manage crew schedules, ensuring that staff assignments are aligned with crew availability, flight duration, and safety regulations. Crew availability and certifications should be tracked to ensure the right personnel are assigned to each flight.

c. Fuel Consumption and Efficiency Monitoring:

Provide detailed tracking of fuel consumption to ensure operational efficiency and reduce costs. Fuel usage data can help airlines optimize flight routes and minimize fuel wastage.

5. Real-Time Analytics:
   a. Data-Driven Decision Making:

Offer real-time analytics on flight operations, passenger behavior, booking trends, and operational efficiency. Airlines can use this data for route optimization, demand forecasting, and pricing strategy adjustments.

b. Route Optimization and Demand Forecasting:

Use historical and real-time data to forecast demand for certain routes, adjust ticket prices, and optimize flight schedules to maximize profitability. Predictive models can help airlines better allocate resources and manage peak travel periods.

c. Customer Satisfaction Analysis:

Analyze feedback from passengers and measure key performance indicators such as on-time performance, service quality, and in-flight experience to improve customer satisfaction and loyalty.

6. Customer Support:
   a. Automated Support Systems:

Provide automated support for common inquiries like flight status, booking confirmation, and baggage policies through chatbots, mobile apps, or self-service portals, reducing the workload on human agents and improving response times.

b. Self-Service Options:

Allow passengers to check-in online, choose seats, and print boarding passes through the web or mobile apps, offering convenience and reducing airport congestion.

c. Real-Time Communication Channels:

Enable real-time communication through email, SMS, or chat platforms to address urgent passenger issues, such as last-minute flight cancellations, rebooking, or lost baggage tracking.

7. Challenges Addressed by the System
   a. Operational Efficiency:

AMS automates flight scheduling, crew management, and fleet maintenance, reducing operational delays and ensuring that all flights operate safely and on time.

b. Passenger Satisfaction:

The system provides seamless booking experience, personalized services, and real-time updates, increasing customer satisfaction and loyalty.

c. Cost Optimization:

By optimizing routes, improving fuel efficiency, and automating various processes, the system helps airlines reduce operational costs and improve profitability.

d. Data-Driven Insights:

Real-time analytics enable better decision-making, allowing airlines to optimize resource allocation, adjust pricing strategies, and respond to market trends faster.

Expected Outcome:

The implementation of this Airline Management System will result in improved operational efficiency, increased customer satisfaction, and optimized resource management, leading to enhanced competitiveness and profitability for the airline.

EXPERIMENT-2

Objectives

To design a Use Case Diagram for the Airline Management System based on the problem statement

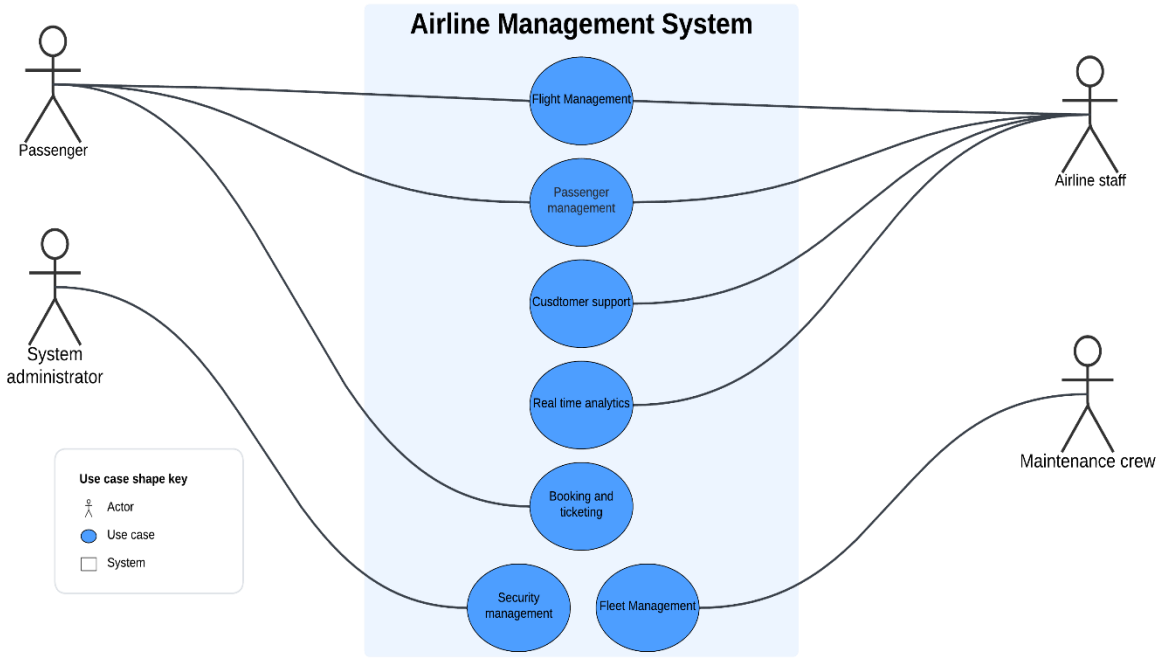
Introduction

A Use Case Diagram is a behavioral UML diagram that captures the functional aspects of a system. It represents the interactions between various actors (users or external systems) and the system, showcasing the key functionalities that the system must support.

Theory

The purpose of the Use Case Diagram is to provide a high-level visual representation of the interactions between users and the Airline Management System, ensuring that all necessary functionalities are addressed in the design and development phase.

Use Case Diagram:

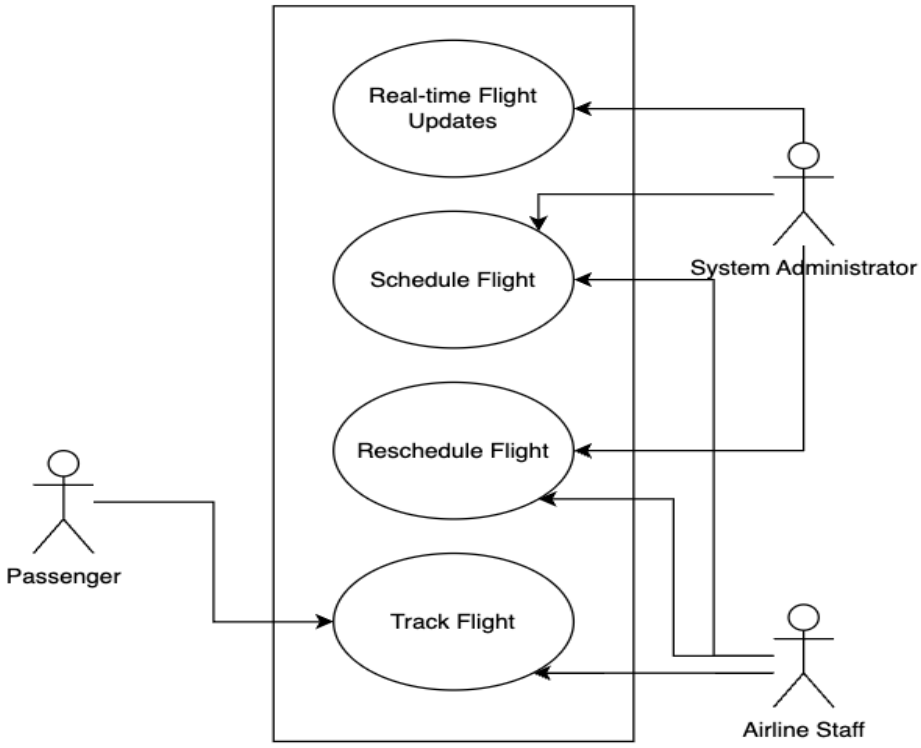


Actors:

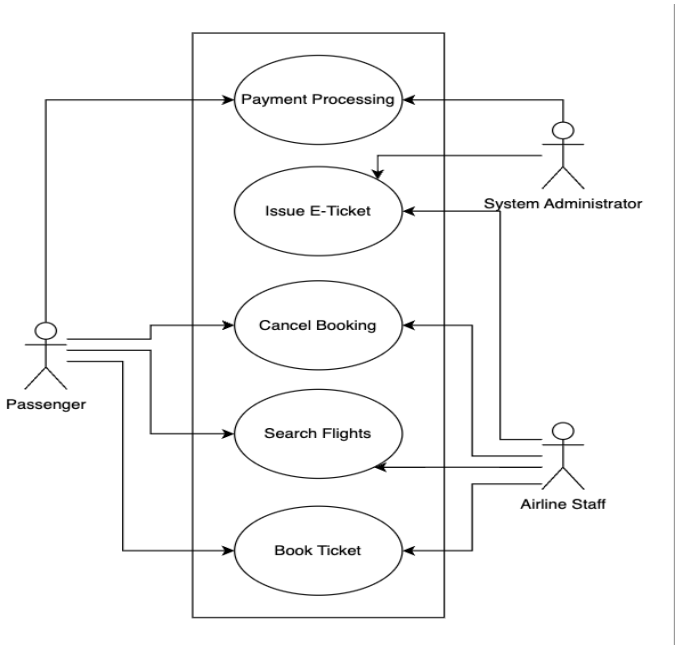
1. Passenger: Books, cancels, and modifies flight reservations; receives flight updates and support.
2. Airline Staff: Manages flights, schedules, and provides customer support.
3. System Administrator: Manages system configurations, flight data, and handles security.
4. Maintenance Crew: Manages fleet status and maintenance schedules.

Use Cases:

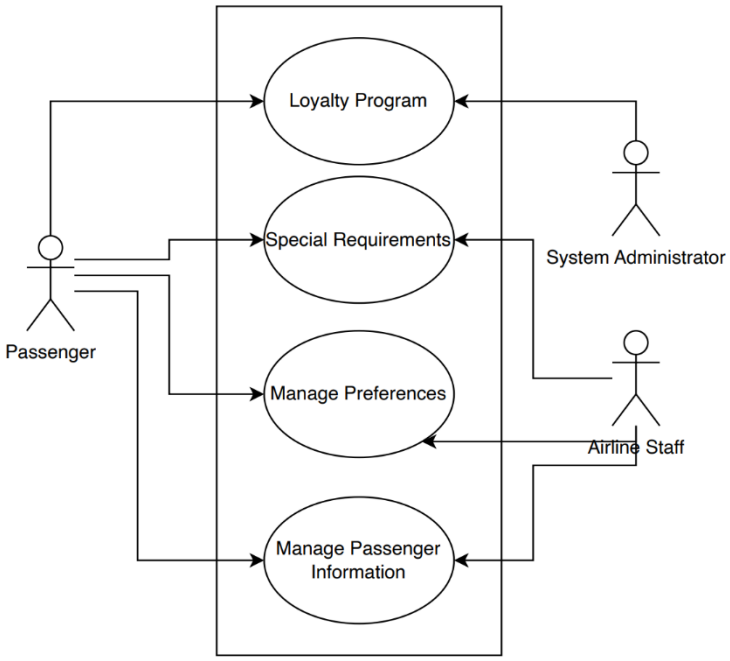
- Flight Management: Schedule, reschedule, and track flights, along with real-time updates.



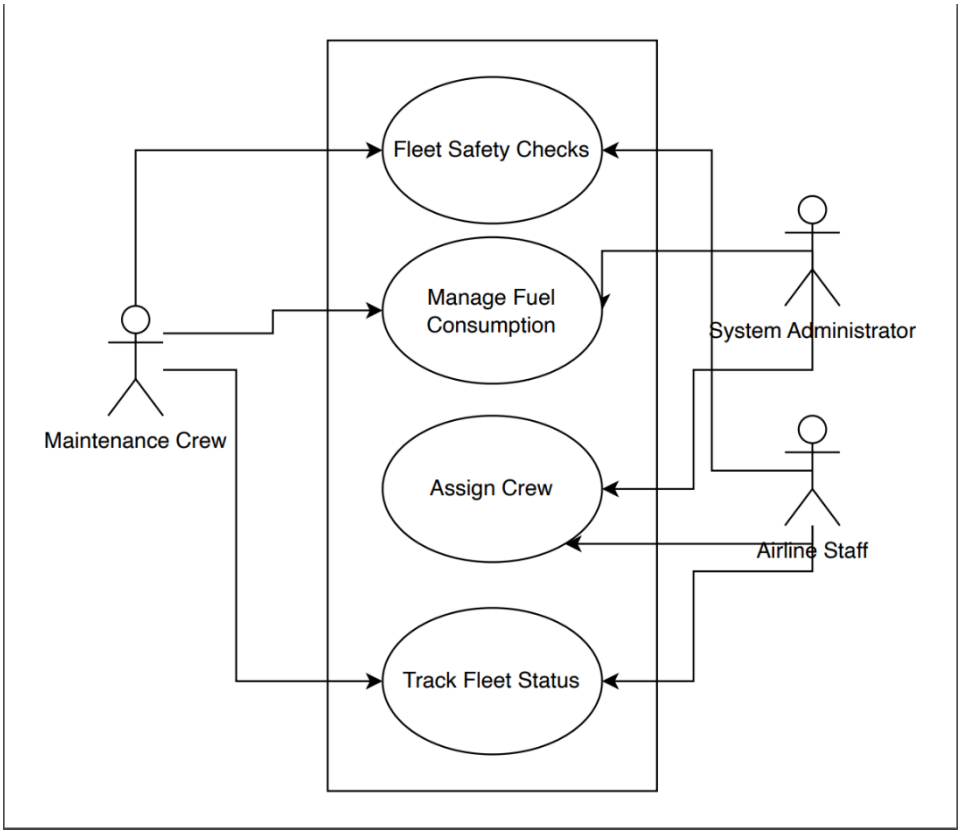
- Booking and Ticketing: Enable passengers to search for flights, book tickets, manage reservations, and cancel bookings.



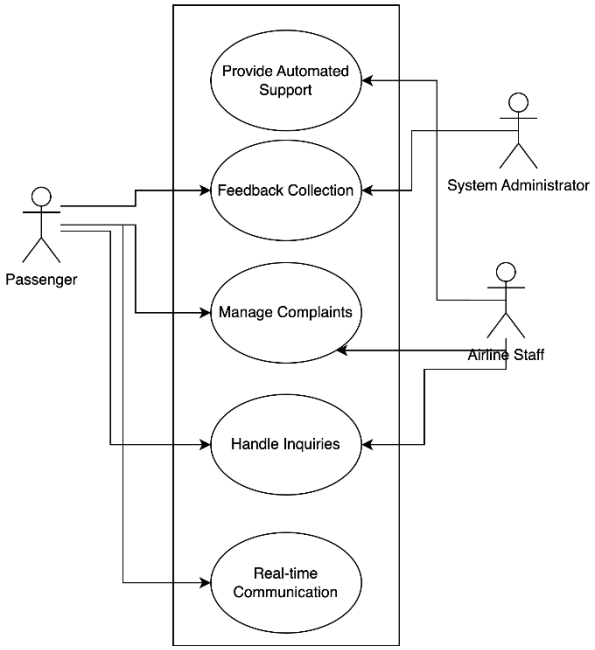
- Passenger Management: Maintain passenger records, manage preferences, and provide personalized services.



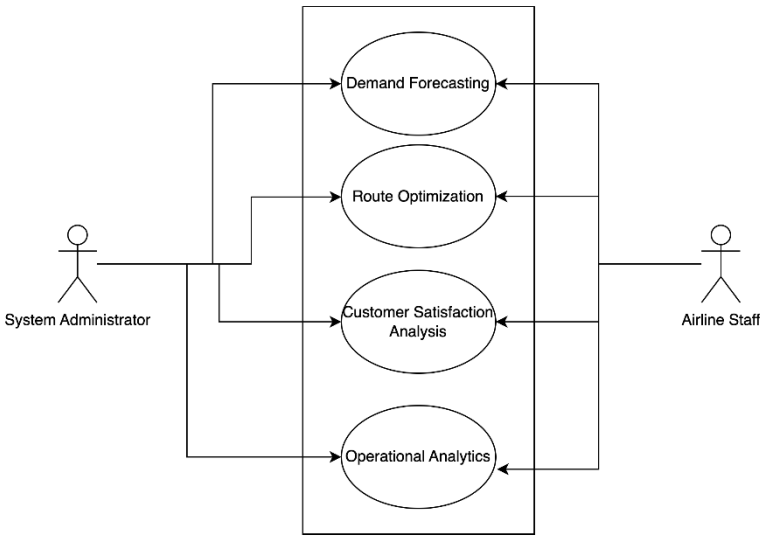
- Fleet Management: Track aircraft status, schedule maintenance, and assign crew.



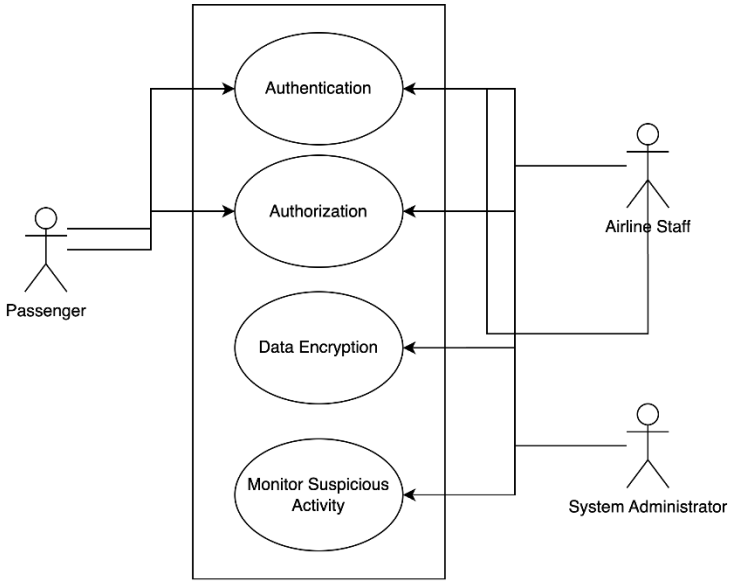
- Customer Support: Provide automated or manual support, manage customer inquiries and issues.



- Real-Time Analytics: Offer insights on demand forecasting, route optimization, and customer satisfaction analysis.



- Security Management: Manage authentication, data integrity, and ensure system compliance.



Expected Outcome:

The Use Case Diagram will provide a detailed and organized visual representation of the interactions between the actors and the Airline Management System. This will guide the development of a system that meets both functional and non-functional requirements, ensuring smooth operations for the airline.

EXPERIMENT-3

Objectives

To write the use case descriptions of the actors involved in the airline Management system based on the use case diagram.

Introduction:

A Use Case Description is a textual explanation of how an actor interacts with the system to achieve a specific goal. It provides a detailed understanding of each use case by outlining the steps involved in the interaction, the flow of events, and any alternate or exceptional scenarios that may occur.

For the Airline Management System, we have several actors, including Passenger, Airline Staff, System Administrator, and Maintenance Crew, each performing different roles in the system. Each actor will have corresponding use cases related to flight management, booking, customer support, fleet maintenance, and more.

Theory:

Background:

The Airline Management System consists of multiple actors with specific roles and responsibilities. These actors represent the external entities interacting with the system, and they perform different functions such as managing flights, booking tickets, handling customer support, and ensuring system security. Understanding these roles is crucial for defining the system's requirements and ensuring that all functionalities are properly addressed.

1. Use Case Name: Flight Management
    a. Actors: Airline Staff, System Administrator
    b. Preconditions: Flight data must be available in the system.
    c. Postconditions: Flights are successfully scheduled or rescheduled, and real-time tracking data is available.
    d. Basic Flow:
        i. Airline staff logs into the system.
        ii. They access the flight scheduling module.
        iii. The staff schedules new flights or reschedules existing ones.
        iv. The system updates flight information and notifies relevant departments.
        v. The system tracks flight status in real-time and sends updates to passengers.
    e. Alternate Flow: If a flight is delayed, staff reschedules the flight, and the system automatically notifies passengers.
    f. Exceptions: If a flight is canceled due to weather, the system triggers an emergency response to reschedule passengers.

2. Use Case Name: Booking and Ticketing
    a. Actors: Passenger, Airline Staff
    b. Preconditions: Flights must be available for booking.
    c. Postconditions: The booking is confirmed, and the ticket is generated.
    d. Basic Flow:
        i. Passenger searches for available flights.
        ii. The system displays options based on criteria like date, destination, and price.
        iii. Passenger selects a flight and proceeds with booking.
        iv. The system processes the payment and generates a ticket.
        v. Passenger receives confirmation via email or SMS.
    e. Alternate Flow: If a passenger cancels a booking, the system processes the cancellation and initiates a refund if applicable.
    f. Exceptions: If payment fails, the system notifies the passenger and requests alternative payment.

3. Use Case Name: Passenger Management
    a. Actors: Passenger, Airline Staff
    b. Preconditions: The passenger must have an account in the system.
    c. Postconditions: Passenger records are updated with the latest information and preferences.
    d. Basic Flow:
        i. Passenger logs into their account.
        ii. They update personal details, preferences (e.g., seat, meal), and special requirements (e.g., wheelchair assistance).
        iii. The system saves the updates and reflects them in future bookings.
    e. Alternate Flow: If a passenger has special needs, the system generates an alert for staff to provide personalized services.
    f. Exceptions: If the system fails to save updates, an error message is shown, and the process is retried.

4. Use Case Name: Fleet Management
    a. Actors: Maintenance Crew, System Administrator
    b. Preconditions: Aircraft must be registered in the system.
    c. Postconditions: Aircraft maintenance schedules are updated, and crew assignments are managed.
    d. Basic Flow:
        i. Maintenance crew logs into the system.
        ii. They access fleet maintenance records.
        iii. The crew updates the maintenance schedule or logs completed maintenance.
        iv. The system assigns crew members to flights based on availability and regulatory requirements.
        v. The system tracks fuel consumption and updates operational efficiency metrics.
    e. Alternate Flow: The crew requests additional resources if an aircraft requires immediate repairs.
    f. Exceptions: If an aircraft is grounded, the system triggers alerts to adjust flight schedules accordingly.

5. Use Case Name: Customer Support
    a. Actors: Passenger, Airline Staff
    b. Preconditions: Customer inquiries or issues must exist.
    c. Postconditions: The inquiry is resolved, or support is provided.
    d. Basic Flow:
        i. Passenger initiates a support request via chat, phone, or email.
        ii. The system routes the request to the appropriate support agent or provides automated responses through chatbots.
        iii. The agent resolves the issue (e.g., flight rebooking, baggage inquiry).
        iv. The system updates the ticket and closes the request.
    e. Alternate Flow: If the inquiry is simple, the system provides automated responses.
    f. Exceptions: If the issue is escalated, the system logs the escalation and forwards it to a higher-level support agent.

6. Use Case Name: Real-Time Analytics
    a. Actors: Airline Staff, System Administrator
    b. Preconditions: Data must be collected from various airline operations.
    c. Postconditions: Insights are generated for decision-making.
    d. Basic Flow:
        i. The system collects data on flight operations, bookings, and customer feedback.
        ii. Airline staff access the analytics dashboard.
        iii. The system generates reports on demand forecasting, route optimization, and customer satisfaction.
        iv. Staff make decisions based on the insights, adjusting flight schedules or marketing strategies.
    e. Alternate Flow: If data shows a decline in customer satisfaction, staff investigates and implements service improvements.
    f. Exceptions: If data is incomplete or unavailable, the system notifies administrators to investigate data collection errors.

7. Use Case Name: Security Management
    a. Actors: System Administrator
    b. Preconditions: Users must have accounts and permissions to access the system.
    c. Postconditions: System security is ensured, and data is protected.
    d. Basic Flow:
        i. The system administrator configures user roles and permissions.
        ii. They monitor login attempts and system access logs for potential security breaches.
        iii. The system enforces data encryption, secure authentication (e.g., multi-factor authentication), and regular security audits.
        iv. If unauthorized access is detected, the system triggers alerts and restricts access.
    e. Alternate Flow: If a user forgets their password, the system initiates a password recovery process.
    f. Exceptions: If the system detects a major security breach, emergency protocols are triggered to lock down sensitive data.

Expected Outcome:

After completing this experiment, a clear and detailed set of use case descriptions will be available, ensuring that all system functionalities are well-defined and that the interactions between actors and the system are properly documented. This will serve as a blueprint for system developers and analysts during the implementation phase, ensuring that all functional and security requirements are addressed effectively.

EXPERIMENT-4

Objectives

To create Data Flow Diagrams (DFDs) at Level 0, Level 1, and Level 2 for the Airline Management System.

Introduction:

A Data Flow Diagram (DFD) is a graphical representation that depicts the flow of data through a system, illustrating how input data is transformed into output data through various processes. In this experiment, we will develop DFDs at different levels (0, 1, and 2) for the Airline Management System, detailing the flow of information between external entities (actors) and the system, as well as how the system processes this information internally.

- Level 0 DFD (also called the Context Diagram) provides a high-level view of the system, showing the system as a single process interacting with external entities.
- Level 1 DFD breaks down the main system process into its sub-processes, providing more detail about the flow of data.
- Level 2 DFD goes even deeper, further decomposing individual processes from Level 1 into more detailed sub-processes.

Theory:

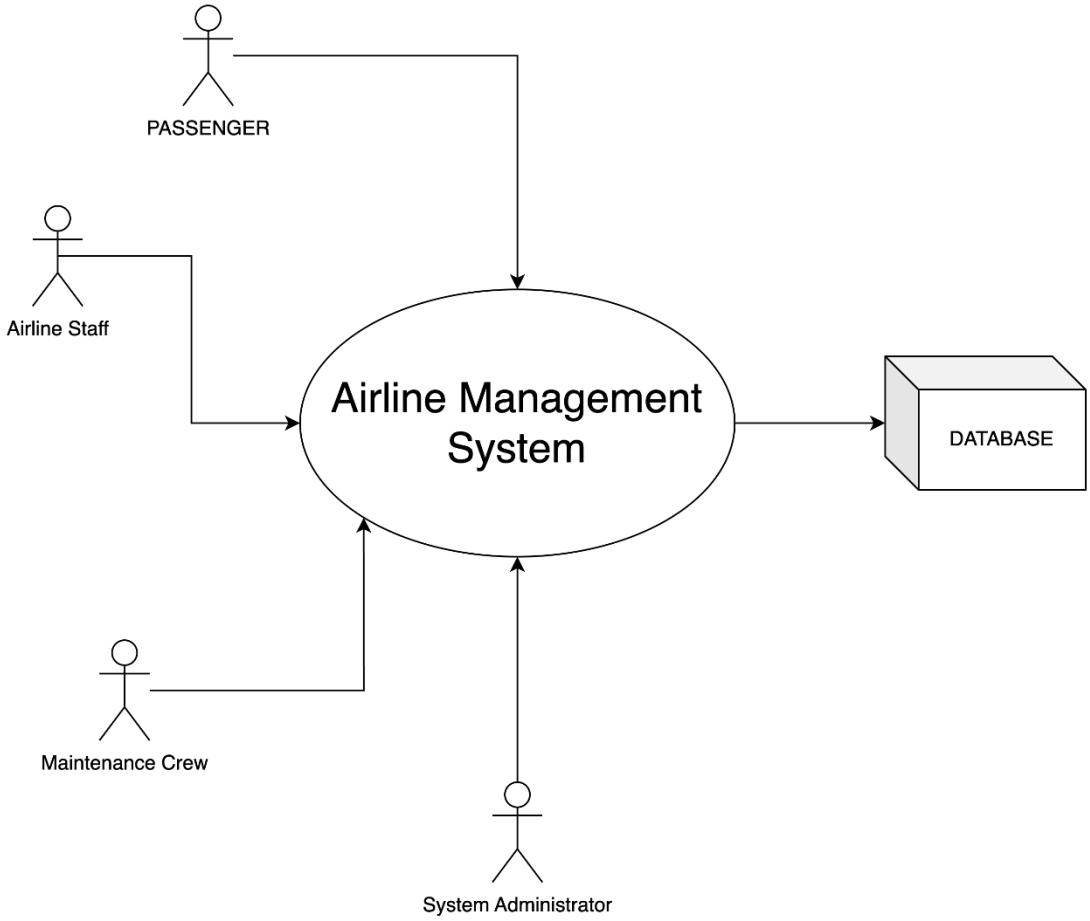
- **Entities**: External agents interacting with the system (e.g., passengers, airline staff).
- **Processes**: Functions or operations performed by the system (e.g., booking flights, managing passengers).
- **Data Stores**: Places where data is stored within the system (e.g., passenger database, flight schedule database).
- **Data Flows**: Represent the flow of information between entities, processes, and data stores.

Process Flow and Actors:

1. Level 0 DFD (Context Diagram):

The Level 0 DFD provides a high-level overview of the Airline Management System, showing how the system interacts with external entities like Passengers, Airline Staff, Maintenance Crew, and System Administrator. At this level, the system is represented as a single process.

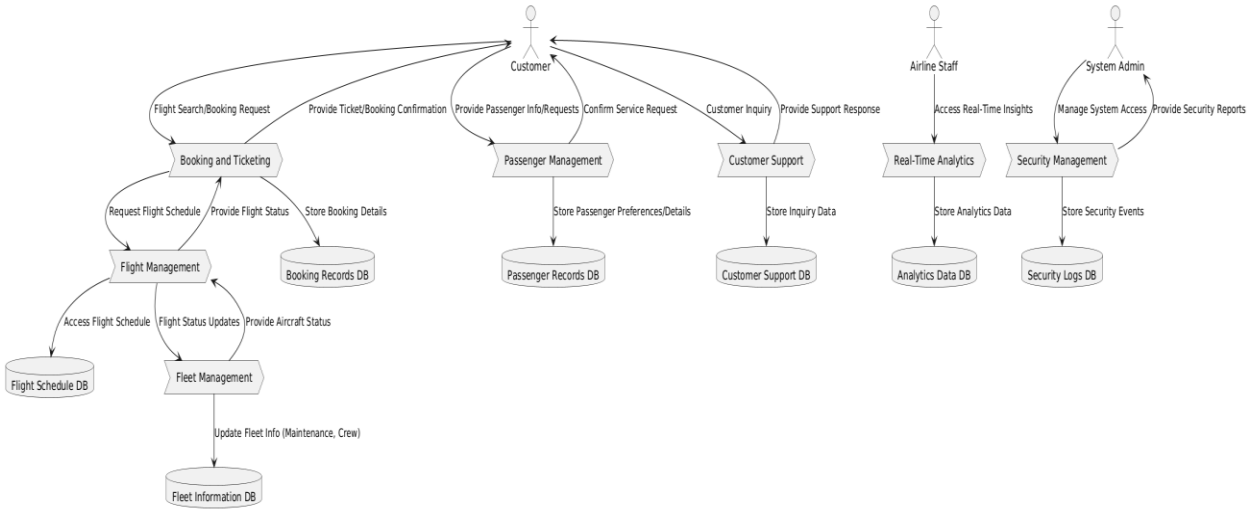
- Passenger: Interacts with the system to search for flights, book tickets, check flight status, manage reservations, and receive customer support.
- Airline Staff: Manages flight schedules, bookings, and provides customer support.
- Maintenance Crew: Updates and tracks maintenance schedules and fleet status.
- System Administrator: Oversees system configurations, security, and overall system performance.



2. Level 1 DFD:

The Level 1 DFD breaks down the single process from Level 0 into major sub-processes like Booking and Ticketing, Flight Management, Passenger Management, Fleet Management, Customer Support, and Security Management. Data flows between these processes and external entities, as well as between processes and data stores.

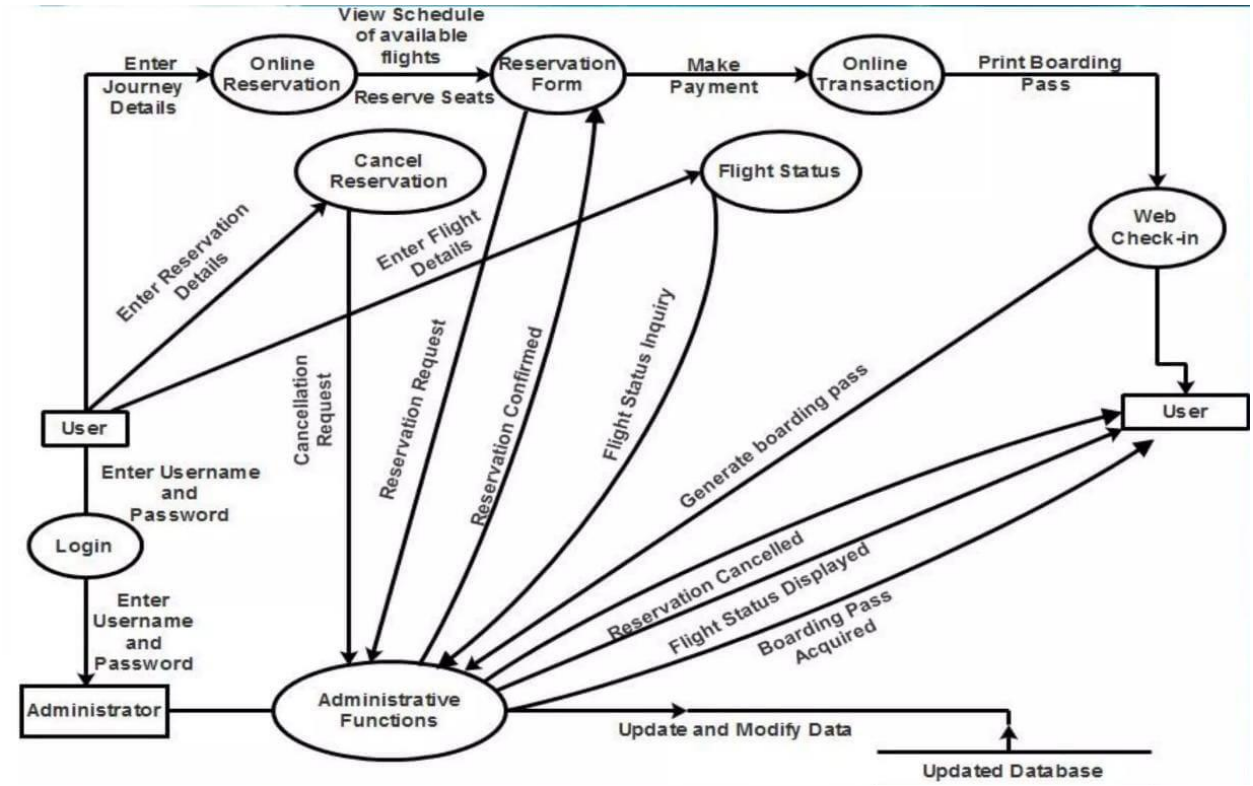
- Flight Management: Manages flight schedules, rescheduling, and tracking of flight status.
- Booking and Ticketing: Manages flight searches, bookings, cancellations, and payment processing.
- Passenger Management: Maintains passenger records, preferences, and special service requests.
- Fleet Management: Manages aircraft maintenance schedules, fuel consumption, and crew assignments.
- Customer Support: Handles customer inquiries and provides automated and manual support.
- Real-Time Analytics: Provides insights for decision-making and operational improvements.
- Security Management: Ensures secure access to the system and manages data integrity.



3. Level 2 DFD:

The Level 2 DFD provides a detailed breakdown of the major sub-processes shown in Level 1. Each key sub-process is broken down into its functional components, showing how data moves within that process.

1. Flight Management:
    - Schedule Flights: Handles scheduling of new flights.
    - Reschedule Flights: Manages the rescheduling of flights in case of changes.
    - Track Flights: Provides real-time flight status updates.
2. Booking and Ticketing:
    - Search Flights: Allows passengers to search for available flights.
    - Book Flights: Manages the booking process.
    - Cancel Flights: Allows passengers to cancel bookings and receive refunds.
    - Payment Processing: Handles secure payments for bookings.
3. Passenger Management:
    - Manage Passenger Information: Allows passengers to update their profiles and personal details.
    - Manage Preferences: Passengers set preferences for seating and meals.
    - Handle Special Requests: Manages requests for special services such as wheelchair assistance.
4. Fleet Management:
    - Update Maintenance Schedules: Tracks and updates aircraft maintenance logs.
    - Assign Crew: Manages crew assignments for flights.
    - Monitor Fleet Status: Tracks the operational status of aircraft.
5. Customer Support:
    - Automated Support: Provides FAQ and chat-based assistance.
    - Manual Support: Allows airline staff to handle customer inquiries and issues.
    - Complaint Management: Manages passenger complaints and feedback.
6. Real-Time Analytics:
    - Demand Forecasting: Analyzes demand trends for flight routes.
    - Route Optimization: Optimizes flight routes for efficiency.
    - Customer Satisfaction: Tracks customer feedback to assess service quality.
7. Security Management:
    - Authentication: Ensures secure login and identity verification.
    - Authorization: Manages access control for various user roles.
    - Data Encryption: Ensures that sensitive passenger and flight data is securely stored.



Expected Outcome:

By constructing Data Flow Diagrams (DFD) at different levels, we will gain a clearer understanding of how data moves through the Airline Management System. The DFDs will help visualize the relationships between external actors and the system, as well as how the system processes and manages data internally. This will serve as a guide for system design and implementation.

EXPERIMENT-5

Objectives

Draw ER Diagram of the given CASE STUDY.

Introduction

An ER Dia gram is a structural diagram that visually represents the relationships between entities in a system. It helps in understanding how data is organized, stored, and rela ted in a database, ensuring that the database design a ligns with the system's functional requirements. For the given case study, the ER diagram will capture the key entities, their attributes, and relationships, providing a blueprint for databa se design.

Theory

Background:

An ER Dia gram helps in the logical design of a database by identifying the entities (objects or things) involved in the system and their relationships. It serves as the foundation for structuring data in a manner that ensures consistency, integrity, and efficiency in the database management system. Each entity is represented as a rectangle, attributes a re denoted by ova ls, and relationships between entities a re depicted using diamonds.

Objective:

The objective is to translate the case study into an ER Diagram, capturing all the entities, their relationships, and any constraints or key attributes necessary for database design. This diagram will aid in the development of the database schema for the system.

Key Components:

1. **Entities:**

The entities are the core objects involved in the system. In the context of the given case study, these entities represent the rea l-world objects that the system will manage.

Example entities from the case study:

- Pa ssenger: Represents individuals booking flights.
- Flight: Represents scheduled flights in the system.
- Ticket: Represents tickets booked by passengers.
- Aircra ft: Represents the airline's fleet of airplanes.

- Sta ff: Represents the airline staff managing flights and passengers.
- Airport: Represents the departure and arrival locations.

2. Attributes:

Each entity has specific characteristics or properties called attributes, which define the data stored for each instance of an entity.

Example attributes for the entities:

- Pa ssenger: Pa ssengerID (Prima ry Key), Name, Contact Information, Preferences.
- Flight: FlightID (Prima ry Key), Destination, DepartureTime, Sta tus.
- Ticket: TicketID (Primary Key), BookingDate, SeatNumber, Price.
- Aircra ft: Aircra ftID (Primary Key), Model, Capacity, MaintenanceSchedule.
- Sta ff: Sta ffID (Prima ry Key), Role, Schedule, Certifications.
- Airport: AirportCode (Primary Key), Location, Facilities.

3. Relationships:

The relationships between entities indicate how the entities interact with one another. Example

relationships:

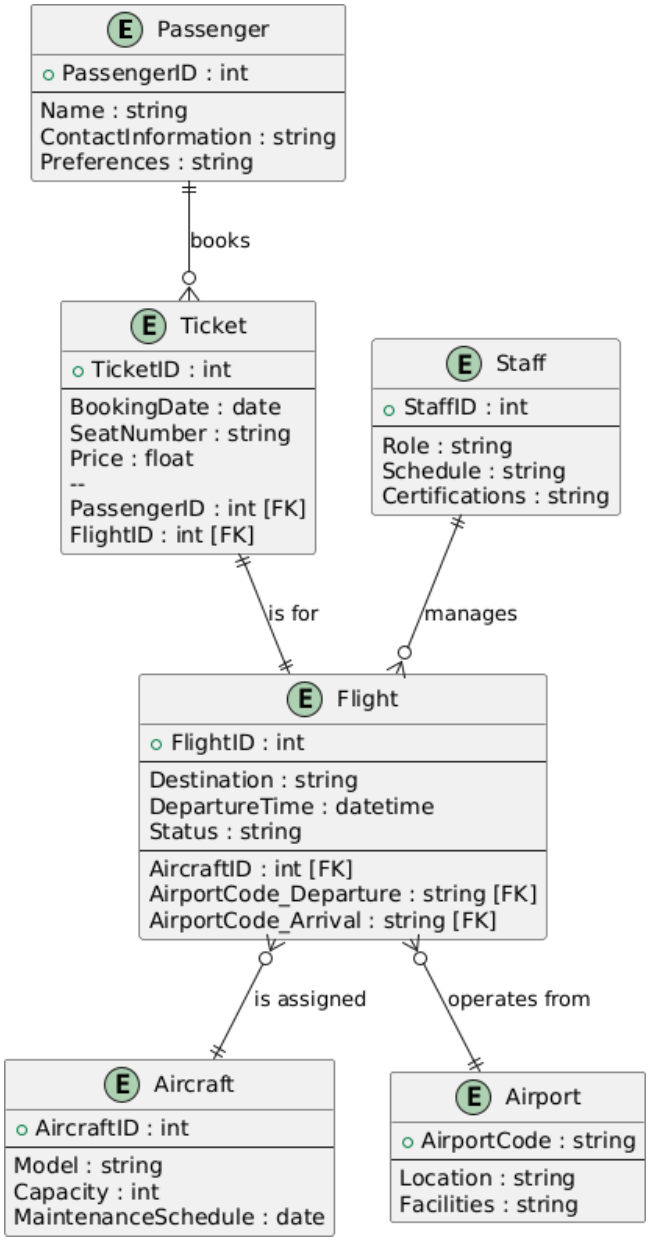
- Pa ssenger books Ticket.
- Flight operates from Airport.
- Aircra ft is assigned to Flight.
- Sta ff manages Flight.

Cardinality (one-to-one, one-to-many, many-to-many) is used to describe the nature of these relationships.

4. Constraints:

Constraints specify rules that govern the relationships and attributes. For example, a passenger can book multiple tickets (one-to-many), but a ticket is associated with only one flight (one-to-one). Keys and foreign keys are also used to enforce integrity constraints in the ER model.

ER Diagram:



Expected Outcome:

The ER Diagram will provide a detailed and organized visual representation of the entities, their attributes, and rela tionships in the system. It will serve as a guideline for database crea tion and ensure that all necessa ry data elements and interactions a re accounted for in the design of the database. This will facilitate efficient data stora ge, retrieva l, and manipulation in the Airline Management Syst

Experiment – 6

Objectives

To design a Software Requirement Specification (SRS) document according to the IEEE standard for the given case study.

Introduction

A Software Requirement Specification (SRS) document is a comprehensive description of the intended software system, outlining its functionality, performance, and constra ints. According to the IEEE 830 sta ndard, the SRS provides a detailed account of the software's purpose, user requirements, system features, and design constraints.

The document ensures that all stakeholders share a common understanding of the system's objectives and the expected functionality.

Theory

Background:

An SRS serves as a formal agreement between the client and the development team, specifying wha t the softwa re system must achieve. It provides a detailed outline of both functional and non -functional requirements, laying the foundation for subsequent stages of softwa re design, development, and testing. Following the IEEE 830 sta ndard ensures that the SRS is clear, complete, and consistent.

Objective:

The objective is to create a Software Requirement Specification document that adheres to the IEEE 830 standard. This document will describe the scope, features, performance requirements, and constraints for the system outlined in the case study.

Key Components (As per IEEE Standard 830)

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the Airline Management System (AMS). It is designed to ensure the efficient, secure, and seamless operation of airline services, including flight management, booking and ticketing, passenger management, fleet ma intena nce, customer support, and rea l-time analytics.

1.2 Scope

The AMS will be a scalable and robust system that streamlines flight operations, enhances customer satisfaction, and improves resource management. The key functionalities of the system will include:

- Flight management (scheduling, real-time updates, cancellations)
- Booking and ticketing (online reservations, modifications, refunds)
- Pa ssenger management (personal data, preferences, special services)
- Fleet management (aircraft maintenance, crew assignment)
- Real-time analytics (operational insights, demand forecasting)
- Customer support (automated help, live chat)

The system will ensure real-time data updates, compliance with aviation regulations, and personalized passenger experiences.

1.3 Definitions, Acronyms, and Abbreviations

- AMS: Airline Management System
- SRS: Softwa re Requirements Specifica tion
- API: Applica tion Programming Interface
- SMS: Short Message Service
- UML: Unified Modeling Language
- GUI: Gra phical User Interface

1.4 References

1. IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
2. Airline Management System Project Charter.
3. Airline Industry Regulations and Compliance Guidelines.

2. Overall Description

2.1 Product Perspective

The AMS will replace traditional manual systems with an integrated software platform, consolidating multiple a irline operations into a single interface. The system will integrate with external APIs (for payment gateways, customer notifications), and support multiple platforms (desktop, web, mobile).

2.2 Product Functions

- Flight Management: Automates flight scheduling, cancellations, and real-time tracking.
- Booking & Ticketing: Manages flight reservations, modifications, and cancellations.
- Pa ssenger Management: Stores and manages passenger records, preferences, and services.
- Fleet Management: Tracks aircraft maintenance schedules and assigns crew members.
- Real-Time Analytics: Provides operational insights for decision-making.
- Customer Support: Automated and live support for flight queries and rebooking.

2.3 User Characteristics

- Pa ssengers: Book and manage flight reservations, receive notifications.
- Airline Staff: Manage flight operations, passenger services, and customer support.
- System Administrators: Oversee system security, configurations, and ensure compliance.
- Maintenance Crew: Track and update aircraft maintenance and safety schedules.

2.4 Constraints

- The system must comply with international aviation regulations and data protection laws.
- The system must handle high-traffic periods (e.g., holiday seasons) without performance degradation.
- Real-time data updates must occur within 5 seconds.

3. Specific Requirements

3.1 Functional Requirements

3.1.1Flight Management

- Automa tion of Flight Scheduling:
- The system shall allow airline staff to create and modify flight schedules.
- Flight status (on-time, delayed, canceled) shall be updated in real-time.
- Notifica tions of schedule changes shall be sent to passengers via email/SMS.
- Real-Time Flight Status Updates:
- The system shall provide real-time tracking of all flights.
- Pa ssengers shall receive flight delay and rescheduling notifications within 5 minutes of any changes.

3.1.2Booking and Ticketing

Online Reservations:

The system shall enable passengers to search for available flights, choose seats, and complete the booking process.

Integration with payment gateways shall support multiple currencies and payment methods.

Reservation Modifications:

The system shall allow passengers to cancel or modify bookings, with automated refunds where applicable. The system shall prevent overbooking by checking seat availability in real-time.

3.1.3Passenger Management

Pa ssenger Record Management:

- The system shall store passenger information, including preferences, meal selections, and any special assistance requests.
- Compliance with GDPR or equivalent privacy regulations shall be enforced for all passenger data.

Persona lized Services:

- The system shall use passenger data to provide personalized notifications and service offerings (e.g., loya lty program rewards).
-

3.1.4Fleet Management

Aircra ft Maintenance Scheduling:

- The system shall track and schedule routine maintenance, inspections, and repairs for all aircraft.
- Maintenance logs shall be updated in real-time by the maintenance crew.

Crew Assignment:

- The system shall manage and assign crew members to flights based on availability, certifications, and regula tory compliance.

- Crew members shall be automatically assigned based on flight duration and staff availability.

3.1.5Real-Time Analytics

Opera tional Insights:

- The system shall provide data -driven insights into flight operations, passenger behavior, and booking trends.
- Predictive models for demand forecasting shall help optimize flight schedules and resource allocation.

Route Optimization:

- Historical and real-time data shall be used to recommend optimal flight routes, minimizing fuel consumption and operational costs.

3.1.6Customer Support

Automa ted Assista nce:

The system sha ll offer automated responses for common pa ssenger queries such as flight sta tus and baggage policies via chatbots or mobile apps.

Real-Time Communication:

The system shall allow passengers to communicate with airline staff through SMS, chat, or email for immedia te concerns like flight dela ys or cancella tions.

3.2 Non-Functional Requirements

3.2.1Performance Requirements

- The system shall support up to 100,000 concurrent users without performance degradation.
- System response times for critical functions (e.g., flight booking, schedule updates) shall not exceed 2 seconds.

3.2.2Security Requirements

- The system shall implement multi-factor authentication for all admin and airline staff logins.
- All passenger data shall be encrypted during transmission and at rest.
- The system shall log and monitor all access attempts for security audits

3.2.3Usability Requirements

- The system shall have a user-friendly GUI for passengers, with accessible design (WCAG 2.0 compliance).

- Airline staff shall be trained in using the system with minimal onboarding (less than 2 weeks).

3.2.4Reliability Requirements

- The system shall maintain 99.9% uptime, with minimal service interruptions.
- The system shall recover from any crashes or downtime within 10 minutes.

3.2.5Maintainability Requirements

- The system shall be modular and support easy updates and maintenance.
- Maintenance logs shall be kept for all updates or changes to the system.

4. System Models

4.1 Use Case Diagrams

4.1.1 Flight Management

- Actors: Airline Staff, System Administra tor
- Use Cases: Scheduling flights, real-time updates, notifying passengers.

4.1.2 Booking and Ticketing

- Actors: Passengers, Airline Staff
- Use Cases: Searching flights, booking tickets, modifying or canceling reservations.

4.1.3 Passenger Management

- Actors: Passengers, Airline Staff
- Use Cases: Managing personal data, preferences, and special requests.

4.1.4 Fleet Management

- Actors: Maintenance Crew, System Administra tor
- Use Cases: Scheduling maintenance, updating aircraft records, assigning crew.

5. Appendices

A. Glossary of Terms

- AMS: Airline Management System

- GUI: Graphical User Interface
- GDPR: General Data Protection Regulation