

---

# **Software Requirements and Design Document**

**for**

## **AIRWAY SYNC**

**Prepared by**

**Muhammad Zayyam Hassan - 22i-1247**

**Muhammad Rayyan - 22i-1022**

**Umer Farooq - 22i-1007**

**NEURONEX**

**26-November-2024**

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Product Scope .....	1
1.3 Title.....	1
1.4 Objectives .....	1
1.5 Problem Statement .....	1
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions .....	2
2.3 List of Use Cases .....	2
2.4 Extended Use Cases .....	3
2.5 Use Case Diagram.....	16
<b>3. Other Nonfunctional Requirements .....</b>	<b>17</b>
3.1 Performance Requirements .....	17
3.2 Safety Requirements .....	17
3.3 Security Requirements .....	17
3.4 Software Quality Attributes .....	17
3.5 Business Rules .....	17
3.6 Operating Environment.....	17
3.7 User Interfaces .....	17
<b>4. Domain Model .....</b>	<b>18</b>
<b>5. System Sequence Diagram .....</b>	<b>19</b>
<b>6. Sequence Diagram .....</b>	<b>23</b>
<b>7. Class Diagram .....</b>	<b>26</b>
<b>8. Component Diagram .....</b>	<b>27</b>
<b>9. Package Diagram .....</b>	<b>27</b>
<b>10. Deployment Diagram.....</b>	<b>28</b>

# 1. Introduction

## 1.1 Purpose

This document outlines the requirements and design of the Integrated Airport Management System for Operational Efficiency. Developed by NeuroNex, the system aims to enhance airport operations by providing a unified platform that integrates flight scheduling, passenger services, baggage tracking, and employee management. The solution is specifically designed for handling the operations of one airline at a time to maximize resource utilization and passenger satisfaction.

## 1.2 Product Scope

The Integrated Airport Management System will be implemented in mid- to large-sized airports to optimize various processes such as gate allocation, passenger check-ins, queue management, and emergency handling. It will replace traditional, disconnected tools with a centralized system leveraging IoT, cloud computing, and advanced automation technologies to provide real-time updates and improve decision-making.

## 1.3 Title

**Integrated Airport Management System for Operational Efficiency**  
A scalable and comprehensive solution for managing airport operations efficiently.

## 1.4 Objectives

The system aims to achieve the following objectives:

- **Optimize operational processes:** Streamline flight scheduling, gate allocation, and queue management.
- **Enhance passenger experience:** Reduce wait times and ensure smooth check-ins and baggage handling.
- **Improve coordination:** Centralize data and processes for efficient decision-making.
- **Scalability and reliability:** Leverage modern technologies to support increasing operational demands.

## 1.5 Problem Statement

Airports face challenges due to independent systems managing various operations such as flight scheduling, passenger services, and baggage tracking. This lack of integration results in delays, communication gaps, and inefficiencies that affect both airport operations and passenger satisfaction.

Our system focuses on single airline operations, consolidating all critical functions into an integrated platform. By automating processes, reducing manual errors, and providing real-time data, the solution ensures efficient airport operations and a seamless passenger experience.

## 2. Overall Description

### 2.1 Product Perspective

The Integrated Airport Management System introduces a centralized platform replacing fragmented systems traditionally used in airports. Designed to operate with existing infrastructure, the system will provide interfaces for passengers, airport staff, and management. Advanced features like IoT-enabled tracking, cloud-based scalability, and automation will ensure a seamless integration of services.

### 2.2 Product Functions

The system provides the following core functionalities:

- **Flight Management:** Includes scheduling, gate allocation, and cancellation.
- **Passenger Services:** Streamlined processes for check-ins, boarding passes, and baggage tracking.
- **Emergency Handling:** Automated notifications and protocols for managing critical situations.
- **User Authentication:** Secure login and sign-up mechanisms for authorized access.

### 2.3 List of Use Cases

- Schedule Flight
- Allocate Gate
- Book Flight
- Check-In
- Issue Boarding Pass
- Manage Queue
- Add Emergency Situation
- Notify Authority
- Track Baggage
- Cancel flight
- Search Flight
- Get Queue Wait times
- Manage Notifications

### 2.4 Extended Use Cases

#### USE CASE 1 : SCHEDULE FLIGHT

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Airline Staff
- **Stakeholders and Interests:**
  - **Airlines:** Want efficient flight scheduling to maximize capacity and resource use.
  - **Airport Authorities:** Require timely flight schedules for effective traffic management.
  - **Passengers:** Desire predictable flight schedules for planning travel.
- **Preconditions:**
  - The airline staff is authenticated in the system.
  - The system has information on available aircraft and time slots.
- **Postconditions:**
  - The flight is scheduled in the system.
  - All relevant stakeholders are notified of the new schedule.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Airline Staff logs into the system.	2. System authenticates the staff.
3. Staff selects "Schedule Flight".	4. System displays available time slots and resources (aircraft, crew).
5. Staff enters flight details (route, aircraft, date, time).	6. System checks for scheduling conflicts and validates input.
7. Staff confirms the schedule.	8. System saves the flight details and updates relevant systems (gates, crew schedules, etc.).

**Extensions:**

Condition	Action
1a. Login fails	System displays an error message prompting re-entry of credentials.
4a. No available time slots or resources	System alerts the staff and suggests alternative time slots or resources.
6a. Scheduling conflicts arise	System prompts the staff to resolve conflicts before proceeding.
8a. Input details are incorrect	System logs the errors and allows the staff to make corrections.

**USE CASE 2 : ALLOCATE GATE**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Flight Operations Officer

- **Stakeholders and Interests:**
  - **Airlines:** Need gates for arriving and departing flights to ensure timely operations.
  - **Passengers:** Require accurate and timely gate assignments for smooth boarding.
  - **Ground Crew:** Need gate information to prepare for arrivals and departures.
- **Preconditions:** The flight is scheduled, and gate availability is confirmed.
- **Postconditions:** The flight is assigned a gate, and relevant parties are notified.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Flight Operations Officer logs into the system.	2. System authenticates the officer.
3. Officer selects the flight for gate assignment.	4. System displays available gates.
5. Officer assigns a gate to the flight.	6. System checks for conflicts with other flights.
7. Officer confirms the gate assignment.	8. System updates the flight's gate assignment and notifies relevant parties (passengers, ground crew, and airlines).

**Extensions:**

Condition	Action
1a. Login fails	System displays an error message prompting re-entry of credentials.
4a. No available gates	System alerts the officer and suggests alternative gates or adjustments to the flight schedule.
6a. Gate conflict detected	System notifies the officer of the conflict and prompts for resolution.
8a. Assignment successful	System logs the gate assignment for future reference and updates relevant displays.

**USE CASE 3 : BOOK FLIGHT**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Passenger
- **Stakeholders and Interests:**
  - **Passenger:** Wants to book a flight quickly and efficiently.
  - **Airlines:** Aim to maximize capacity and ensure smooth operations.
  - **Airport Operations:** Need accurate booking information for planning and coordination.
- **Preconditions:** The flight is available for booking, and the passenger has access to the booking system.
- **Postconditions:** The flight is booked, and the passenger receives a confirmation.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Passenger accesses the booking system.	2. System displays available flights.
3. Passenger selects a flight.	4. System shows available seats and fare options.
5. Passenger enters personal details and payment information.	6. System processes the payment.
7. Passenger confirms the booking.	8. System generates a booking confirmation and sends it to the passenger.

**Extensions:**

Condition	Action
1a. System is unavailable	System displays an error message indicating that the booking system is currently down.
4a. No available seats	System notifies the passenger that the selected flight is fully booked.
6a. Payment fails	System prompts the passenger to re-enter payment information or choose an alternative method.
7a. Booking information is incomplete	System prompts the passenger to provide missing details before confirming the booking.

**USE CASE 4 : CHECK-IN**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Passenger
- **Stakeholders and Interests:**
  - **Passenger:** Wants to check in quickly and receive a boarding pass.
  - **Airlines:** Need to finalize the passenger manifest and ensure smooth boarding operations.
  - **Airport Operations:** Require accurate records of checked-in passengers for security and logistics.
- **Preconditions:** The passenger has a valid booking, and check-in is open for the flight.
- **Postconditions:** The passenger is checked in, and a boarding pass is issued.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Passenger arrives at the check-in kiosk or accesses the online check-in.	2. System prompts for booking information (e.g., booking reference, last name).

Actor Action (or Intention)	System Responsibility
3. Passenger enters the required booking information.	4. System verifies the booking and displays the passenger's flight details.
5. Passenger selects a seat (if available).	6. System confirms seat selection and calculates any applicable baggage fees.
7. Passenger confirms check-in.	8. System issues a boarding pass and updates the passenger manifest.

**Extensions:**

Condition	Action
1a. System is unavailable	System displays an error message indicating that the check-in system is currently down.
4a. Booking not found	System notifies the passenger that the booking could not be found and prompts for re-entry.
6a. Baggage fees are applicable	System informs the passenger of any baggage fees and prompts for payment.
7a. Check-in is not allowed	System notifies the passenger that check-in is closed for the flight and provides further instructions.

**USE CASE 5 : BOARDING PASS**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Check-In Agent / Passenger
- **Stakeholders and Interests:**
  - **Passenger:** Wants to receive a boarding pass for their flight.
  - **Airlines:** Need to ensure that passengers have valid boarding passes for boarding.
  - **Airport Operations:** Require accurate records of passengers who are boarding.
- **Preconditions:** The passenger has completed the check-in process and is eligible to receive a boarding pass.
- **Postconditions:** The boarding pass is issued, and the passenger's status is updated in the system.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Passenger or check-in agent requests the boarding pass.	2. System retrieves the passenger's check-in details.
3. System displays the boarding pass information (flight details, seat number, etc.).	4. Passenger or agent confirms issuance of the boarding pass.



Actor Action (or Intention)	System Responsibility
5. System generates the boarding pass.	6. System prints the boarding pass or sends it electronically to the passenger.

**Extensions:**

Condition	Action
1a. System is unavailable	System displays an error message indicating that the boarding pass issuance system is down.
3a. Check-in details not found	System notifies the agent or passenger that the check-in details could not be found.
4a. Flight is delayed or cancelled	System informs the passenger or agent and provides options for rebooking or updating.
5a. Boarding pass printing fails	System alerts the agent and allows for reattempting the printing process or manual entry.

**USE CASE 6 : MANAGE QUEUE**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Airport Officer
- **Stakeholders and Interests:**
  - **Passengers:** Want to minimize wait times and have a smooth experience during boarding or check-in.
  - **Airport Operations:** Need efficient management of passenger flow to ensure timely operations.
  - **Airlines:** Require timely boarding processes to adhere to schedules and improve passenger satisfaction.
- **Preconditions:** The queue management system is operational, and passengers are present in the queue.
- **Postconditions:** The queue is managed efficiently, and passengers are processed in a timely manner.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Airport Officer logs into the queue management system.	2. System authenticates the officer.
3. Officer reviews current queue status and wait times.	4. System displays real-time data on queue lengths and passenger flow.
5. Officer reallocates staff or resources to optimize the queue.	6. System updates staff assignments and notifies relevant personnel.

Actor Action (or Intention)	System Responsibility
7. Officer monitors queue performance metrics.	8. System sends notifications to passengers about estimated wait times and boarding instructions.
	9. System generates reports on queue efficiency and passenger satisfaction.

**Extensions:**

Condition	Action
1a. Login fails	System displays an error message prompting re-entry of credentials.
4a. Queue exceeds defined length	System alerts the officer and suggests additional resources or staff to manage the queue.
5a. Staff is unavailable	System notifies the officer of the staffing issue and prompts for reallocation or hiring of additional staff.
7a. Notifications fail to send	System alerts the officer and allows for manual communication with passengers.
8a. Report generation fails	System alerts the officer and allows for manual entry of data for reporting.

**USE CASE 7 : ADD EMERGENCY SITUATION**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Airport Officer / Security Officer
- **Stakeholders and Interests:**
  - **Passengers:** Want timely and effective communication during emergencies for safety and security.
  - **Airport Staff:** Need clear instructions and coordination during emergency situations.
  - **Emergency Services:** Require accurate information to respond effectively to incidents.
- **Preconditions:** An emergency has been identified (e.g., security threat, medical emergency, fire).
- **Postconditions:** The emergency is managed effectively, and all relevant parties are informed and safe.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Airport Officer identifies an emergency.	2. System detects emergency and initiates emergency protocols.
3. Officer assesses the situation and determines required actions.	4. System compiles relevant data (e.g., location, type of emergency).

Actor Action (or Intention)	System Responsibility
5. Officer communicates with emergency services and relevant staff.	6. System sends alerts to all relevant personnel and displays emergency instructions.
7. Officer coordinates the response efforts.	8. System tracks the status of the emergency response.
9. Officer reviews and documents the emergency response.	10. System generates a report on the incident for future analysis.

**Extensions:**

Condition	Action
1a. Emergency escalates	System alerts the officer of the escalation and prompts immediate action.
4a. Emergency data is incomplete	System notifies the officer of missing data and prompts for a manual entry or correction.
6a. Alerts fail to send	System retries sending alerts or notifies the officer of the failure.
7a. Coordination issues arise	System provides communication tools for the officer to facilitate coordination.
10a. Report generation fails	System alerts the officer and allows for manual entry of data for reporting.

**USE CASE 8 : NOTIFY AUTHORITY**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Airport Officer
- **Stakeholders and Interests:**
  - **Authorities:** Need timely information to respond effectively to incidents or emergencies at the airport.
  - **Passengers:** Benefit from effective communication regarding safety and security measures.
  - **Airport Operations:** Require coordination with authorities for efficient management of airport operations.
- **Preconditions:** An incident requiring notification has occurred (e.g., emergency, security threat).
- **Postconditions:** Authorities are informed, and necessary actions are initiated.
- **Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Airport Officer identifies an incident requiring notification.	2. System prompts the officer to confirm the need to notify authorities.

Actor Action (or Intention)	System Responsibility
3. Officer provides details about the incident (e.g., type, location).	4. System retrieves contact information for relevant authorities.
5. Officer confirms the details of the notification.	6. System compiles the notification message based on the incident details.
7. System sends the notification to the relevant authorities.	8. System logs the notification attempt and updates the status (delivered, pending, failed).

- Extensions:**

Condition	Action
1a. Incident details are incomplete	System alerts the officer to provide all necessary details before proceeding.
4a. Authority contact information is outdated	System notifies the officer to verify and update contact details before sending.
6a. Notification message fails to send	System retries sending the notification or alerts the officer of the failure.
8a. Logging fails	System alerts the officer and allows for manual entry of notification details for record-keeping.

### USE CASE 9 : TRACK BAGGAGE

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actors:**
  - **Baggage Handler**
  - **Passenger**
- **Stakeholders and Interests:**
  - **Passengers:** Want to know the location and status of their baggage throughout the journey.
  - **Airlines:** Need to ensure that baggage is tracked accurately to avoid loss and ensure customer satisfaction.
  - **Airport Operations:** Require real-time baggage tracking to manage resources and improve operational efficiency.
- **Preconditions:** Baggage has been checked in and is associated with a specific flight.
- **Postconditions:** Baggage status is updated, and passengers and staff receive relevant information.

#### Main Success Scenario:

Actor Action (or Intention)	System Responsibility
<b>Passenger</b> scans their baggage tag using a mobile app.	<b>System</b> retrieves baggage details from the database.

Actor Action (or Intention)	System Responsibility
<b>Passenger</b> inputs their current location or queries the baggage status.	<b>System</b> updates the baggage status in real-time.
<b>Baggage Handler</b> scans the baggage tag upon arrival at the baggage claim area.	<b>System</b> tracks the location and updates the baggage status.
<b>System</b> notifies passengers of their baggage location via SMS or app notification.	
<b>Baggage Handler</b> assists passengers with any inquiries regarding their baggage.	
<b>System</b> logs the baggage tracking updates for record-keeping.	

**Extensions:**

Condition	Action
<b>1a.</b> Baggage tag scan fails	System alerts the handler or passenger to re-scan or manually enter the baggage information.
<b>2a.</b> Baggage details are unavailable	System notifies the handler that baggage information cannot be retrieved.
<b>3a.</b> Notifications fail to send	System retries sending notifications or alerts the handler or passenger of the failure.
<b>4a.</b> Baggage is reported lost	System prompts the handler to initiate a lost baggage report and communicate with the passenger.
<b>5a.</b> Logging fails	System alerts the handler and allows for manual entry of tracking details for record-keeping.

**USE CASE 10 : CANCEL FLIGHT**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Flight Operations Officer
- **Stakeholders and Interests:**
  - **Passengers:** Want timely updates on flight statuses.
  - **Airport Staff:** Need real-time flight information for coordination.
  - **Air Traffic Control:** Requires updated flight statuses for scheduling.
- **Preconditions:** The flight is scheduled, and the system is operational.
- **Postconditions:** The flight status is updated, and relevant parties are informed.
- **Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Flight Operations Officer logs into the system.	2. System authenticates the officer.

Actor Action (or Intention)	System Responsibility
3. Officer selects the flight to update.	4. System displays the current status of the selected flight.
5. Officer updates the flight status (e.g., delayed, on-time, canceled).	6. System verifies the new status and checks for any existing conflicts.
7. Officer confirms the update.	8. System updates the flight status in the database.
	9. System sends notifications to passengers, airport staff, and updates displays.

- Extensions:**

Condition	Action
1a. Login fails	System displays an error message prompting re-entry of credentials.
4a. Flight is not found	System alerts the officer.
5a. New status conflicts	System prompts for clarification.
6a. Update is successful	System logs the update for record-keeping.

### USE CASE 1 1: SEARCH FLIGHT

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Passenger
- **Stakeholders and Interests:**
  - **Passenger:** Wants to search for available flights based on various parameters (destination, date, etc.).
  - **Airlines:** Need to provide flight availability to passengers efficiently and ensure booking is streamlined.
  - **Airport Staff:** Need access to flight information to manage passenger flows and operations.
- **Preconditions:** The system is operational, and flight information is available in the system.
- **Postconditions:** The system displays available flights to the passenger for selection.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. Passenger opens the flight search interface.	2. System displays a search form for flight parameters.
3. Passenger enters search criteria (e.g., destination, date, class).	4. System validates the entered data and prepares search query.
5. Passenger submits the search query.	6. System searches the database for flights matching the criteria.

Actor Action (or Intention)	System Responsibility
7. System displays a list of available flights.	8. Passenger reviews the search results.

**Extensions:**

Condition	Action
3a. If passenger enters invalid search criteria (e.g., past date)	System notifies the passenger of invalid criteria and prompts for re-entry.
6a. If no flights match the search criteria	System notifies the passenger that no flights are available and suggests alternative dates or airports.
7a. If there is a system failure during the search	System alerts the passenger and displays an error message with retry options.

**USE CASE 1 2: SEND NOTIFICATION**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** SMS Provider
- **Stakeholders and Interests:**
  - **Passengers:** Want to receive timely notifications regarding baggage status, emergencies, and flight updates.
  - **Airport Staff:** Require updates to manage operations effectively and respond to emergencies.
  - **Airlines:** Need accurate communication to maintain operational efficiency and improve customer satisfaction.
- **Preconditions:** Events triggering notifications (e.g., baggage status updates, emergency alerts, flight status changes) occur in the system.
- **Postconditions:** Relevant notifications are sent to passengers and staff, ensuring all parties are informed.

**Main Success Scenario:**

Actor Action (or Intention)	System Responsibility
1. SMS Provider monitors for notification triggers from the system.	2. System detects events that require notifications (e.g., baggage tracking updates, emergencies, flight status changes).
3. SMS Provider prepares messages for each event type (baggage status, emergencies, flight updates).	4. System identifies affected passengers and relevant staff or officers for each notification type.
5. SMS Provider sends notifications to passengers about baggage status and flight updates.	6. System logs each notification attempt and updates the status (delivered, pending, failed).

Actor Action (or Intention)	System Responsibility
7. SMS Provider sends alerts to staff regarding emergencies or critical flight updates.	8. System confirms the delivery of notifications to the intended recipients.

**Extensions:**

Condition	Action
1a. Notification trigger fails	System alerts the SMS Provider that the notification cannot be processed.
4a. Recipient information is missing	System prompts the SMS Provider to verify and update the recipient details before sending notifications.
5a. SMS sending fails	System retries sending the SMS notification or alerts the SMS Provider of the failure.
6a. Notification content is invalid	System alerts the SMS Provider to correct the message content before sending.
7a. Emergency alerts require immediate attention	System prioritizes the alert and notifies all relevant personnel without delay.

**USE CASE 13: GET QUEUE WAIT TIMES**

- **Scope:** Airport Management System
- **Level:** User Goal
- **Primary Actor:** Passenger
- **Stakeholders and Interests:**
  - **Passenger:** Wants to know estimated wait times in queues to plan their airport activities better.
  - **Airport Staff:** Requires accurate queue data to manage passenger flows efficiently.
  - **Airlines:** Interested in minimizing delays caused by long queues.
- **Preconditions:** The system is operational, and queue data is updated in real-time.
- **Postconditions:** The estimated wait time for the selected queue is displayed to the passenger.

**Main Success Scenario:**

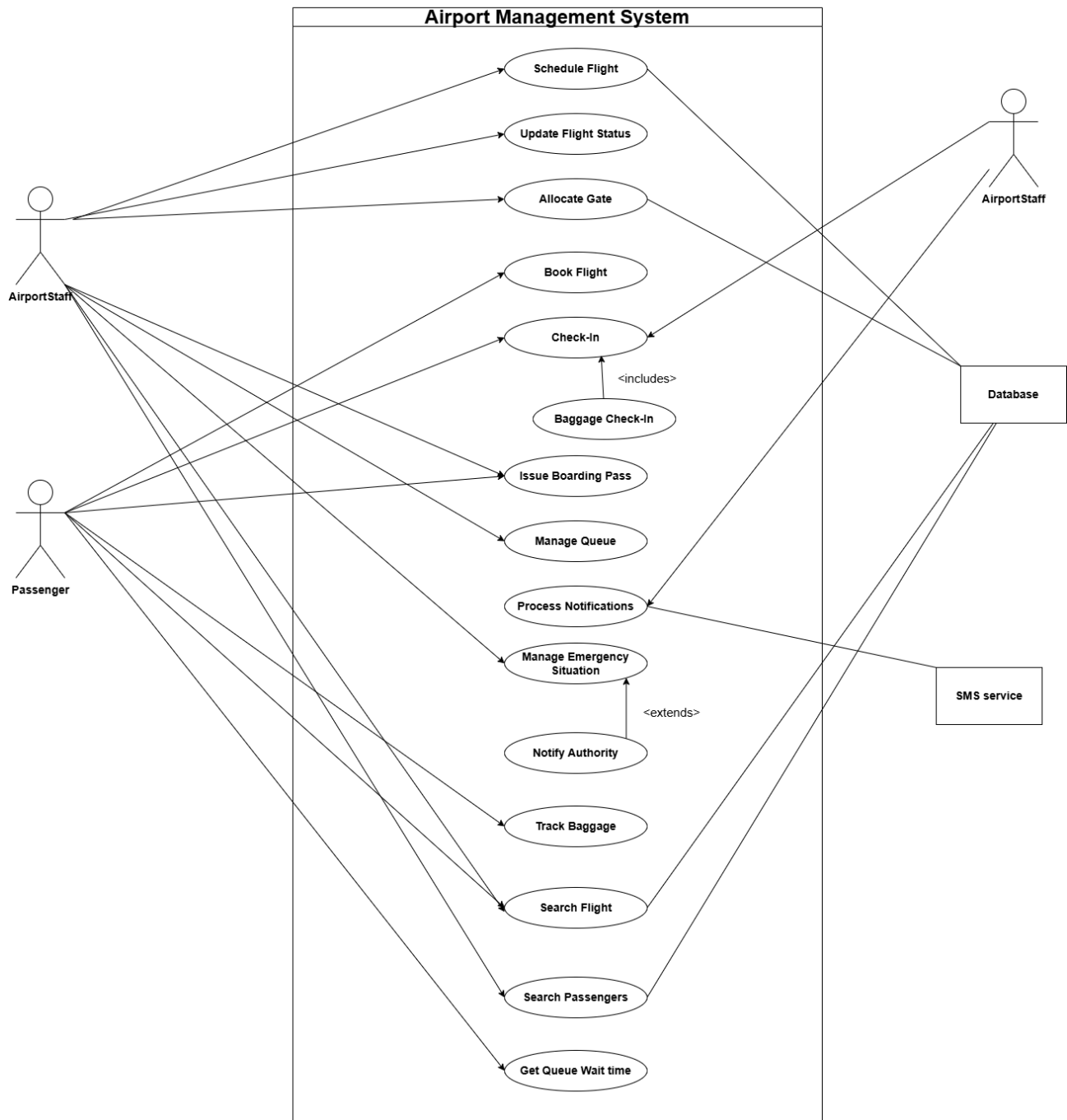
Actor Action (or Intention)	System Responsibility
1. Passenger accesses the "Get Queue Wait Time" feature.	2. System displays a list of queues (e.g., security, check-in).
3. Passenger selects the relevant queue.	4. System retrieves the current queue data and calculates the estimated wait time.
5. System displays the estimated wait time.	6. Passenger reviews the displayed wait time.



**Extensions:**

Condition	Action
<b>1a.</b> Passenger is unable to access the feature	System notifies the passenger of a temporary unavailability of the service.
<b>3a.</b> Passenger selects an invalid or unavailable queue	System alerts the passenger and prompts them to select a valid queue.
<b>4a.</b> Queue data is outdated or unavailable	System informs the passenger that the wait time cannot be calculated and offers retry options.
<b>5a.</b> Calculated wait time exceeds acceptable thresholds	System alerts relevant airport staff for intervention to reduce congestion.

## 2.5 Use Case Diagram



### **3. Other Nonfunctional Requirements**

#### **3.1 Performance Requirements**

- Must support 100 concurrent users with less than 3 seconds response time for critical functions.
- Real-time updates on flight status, gate allocation, and baggage tracking must be reflected within 2 seconds.

#### **3.2 Safety Requirements**

- The system must comply with international aviation safety standards.
- Critical features such as emergency management must operate with a failure rate below 0.01%.

#### **3.3 Security Requirements**

- User authentication must include multi-factor verification for airport staff and management.
- Sensitive passenger data must be encrypted in transit and at rest.

#### **3.4 Software Quality Attributes**

- **Reliability:** Ensure 99.9% uptime.
- **Maintainability:** System updates must require less than 2 hours of downtime.
- **Usability:** Interfaces must be intuitive, requiring minimal training for staff and passengers.

#### **3.5 Business Rules**

- Only authorized personnel may schedule flights or manage emergencies.
- Passengers must confirm their identity before accessing services such as check-ins or baggage tracking.

#### **3.6 Operating Environment**

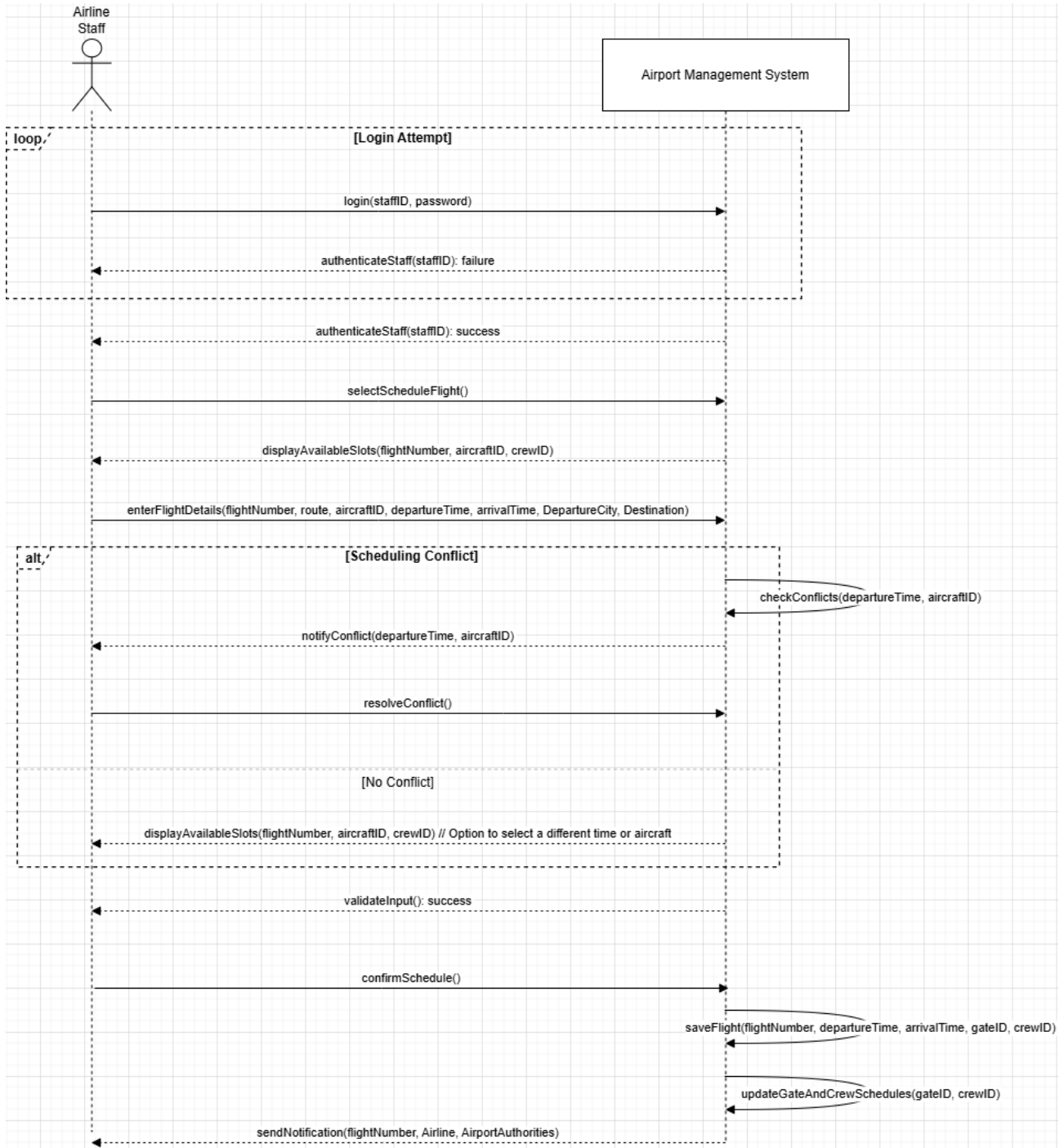
- The system must be compatible with major operating systems, including Windows, Linux, and Android/iOS for mobile applications.
- Cloud-based infrastructure for scalability and disaster recovery.

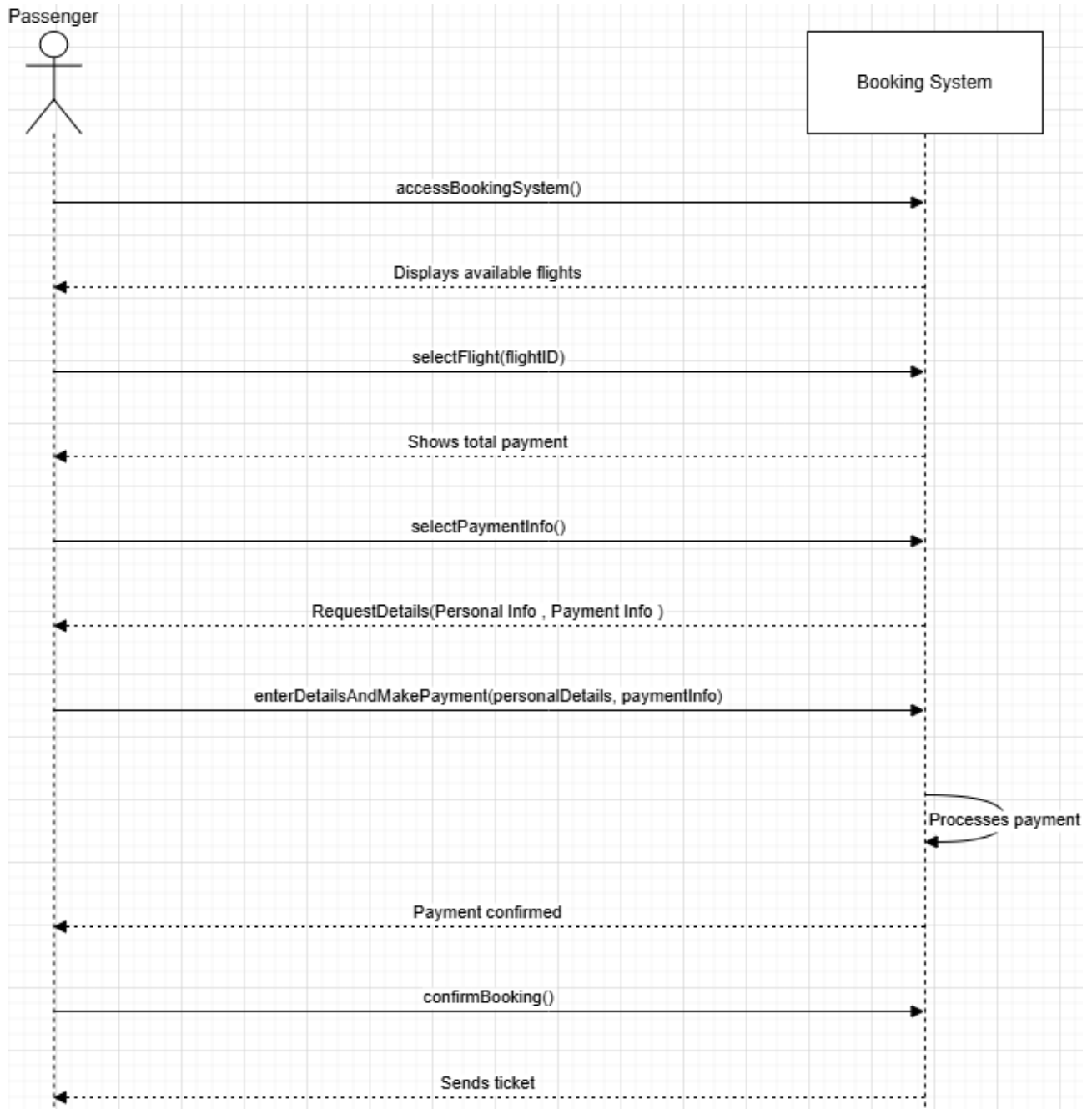
#### **3.7 User Interfaces**

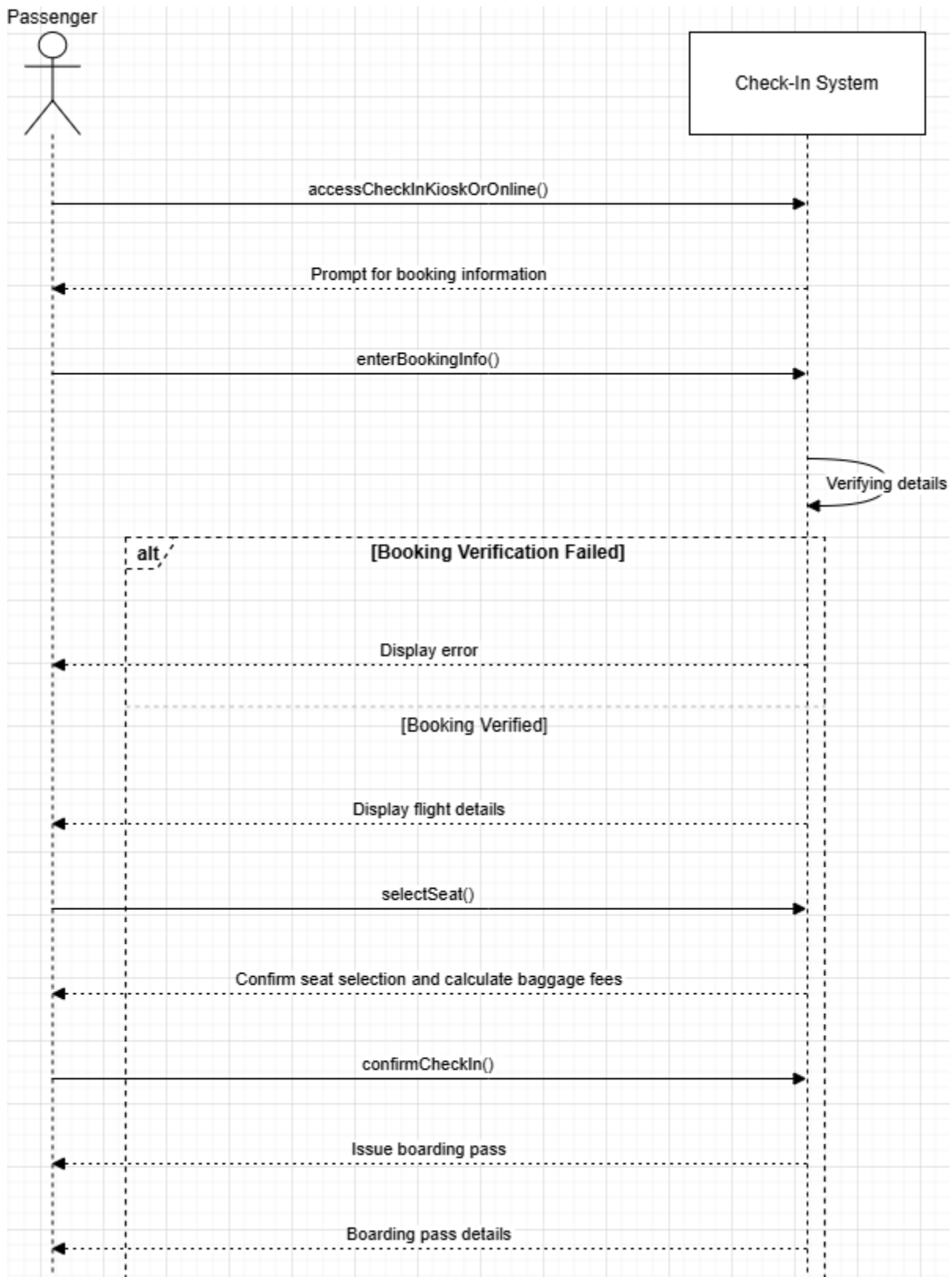
- User-friendly web and mobile interfaces for passengers.
- Admin dashboards for airport staff and management.

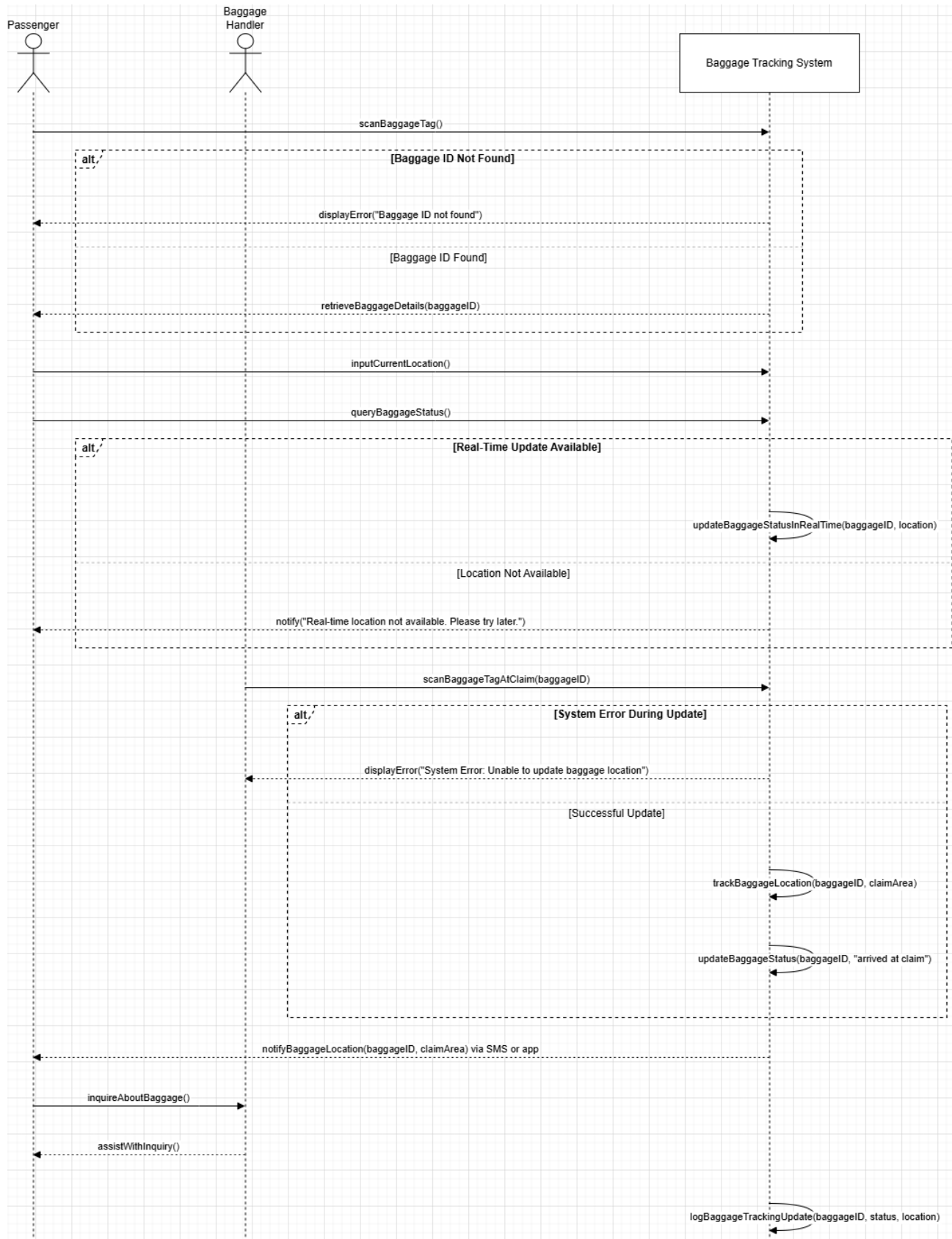


## 5. System Sequence Diagram



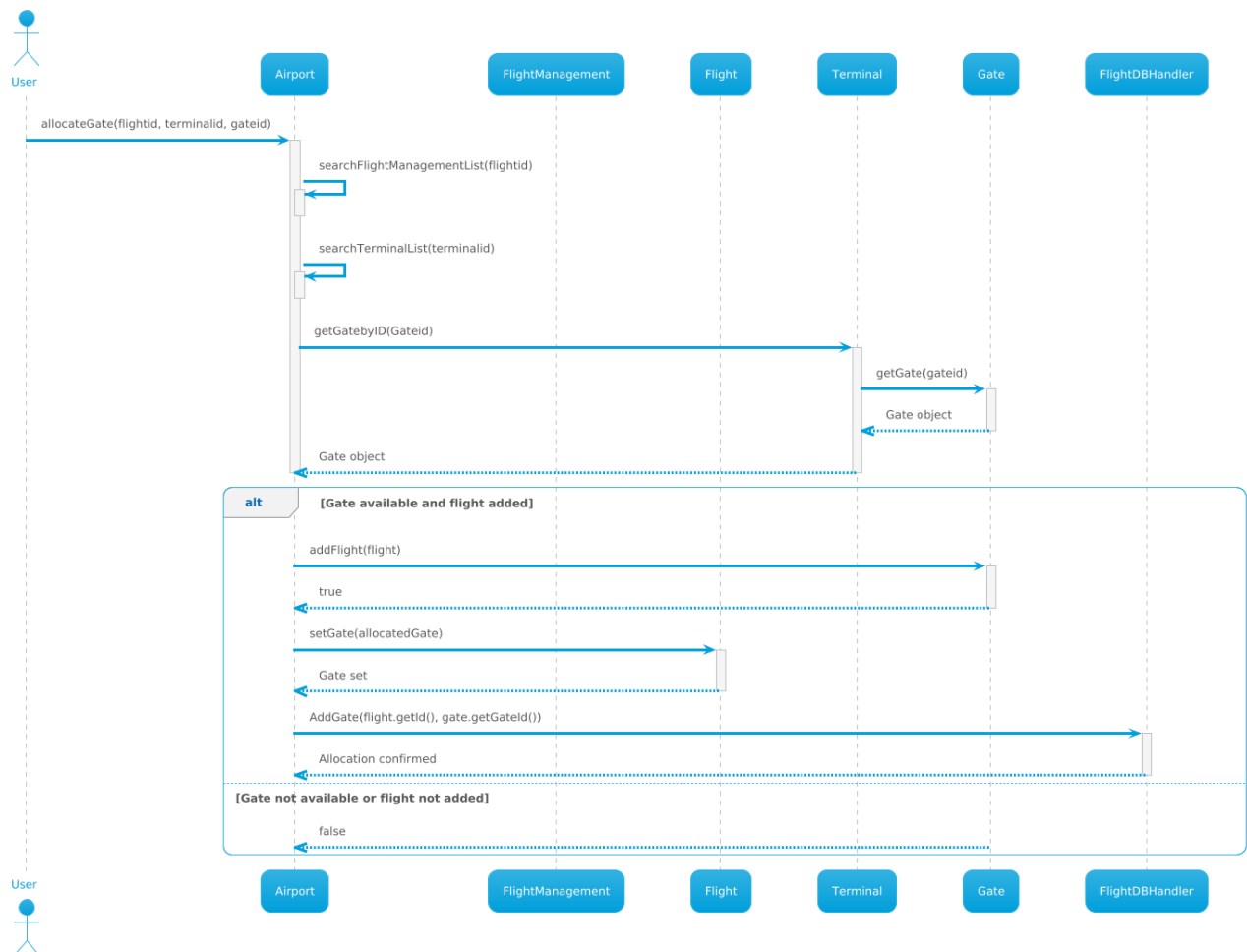
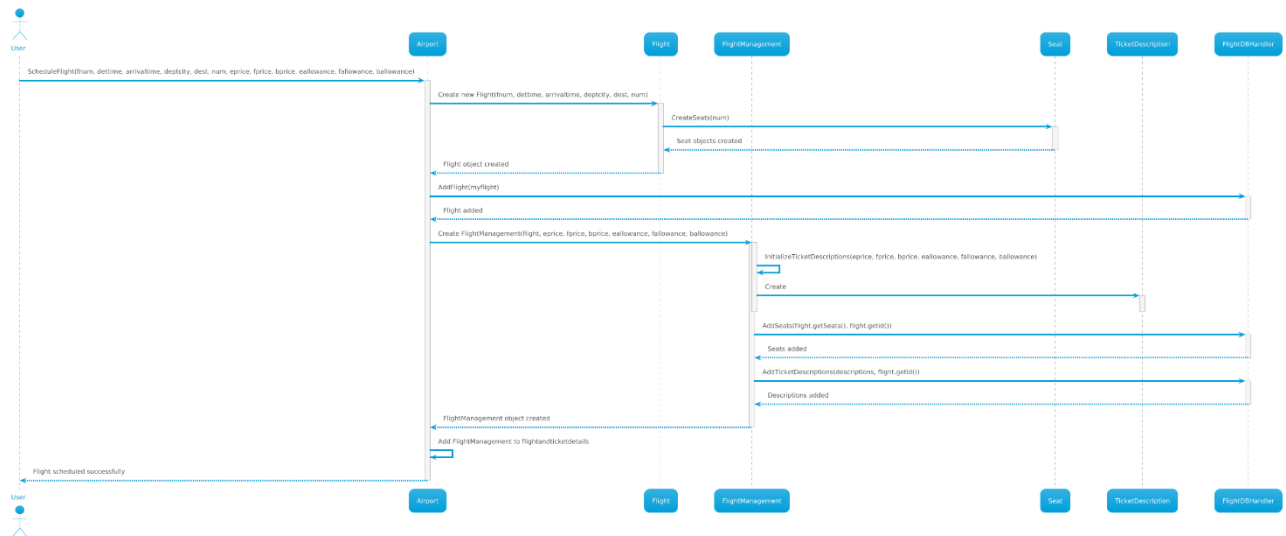


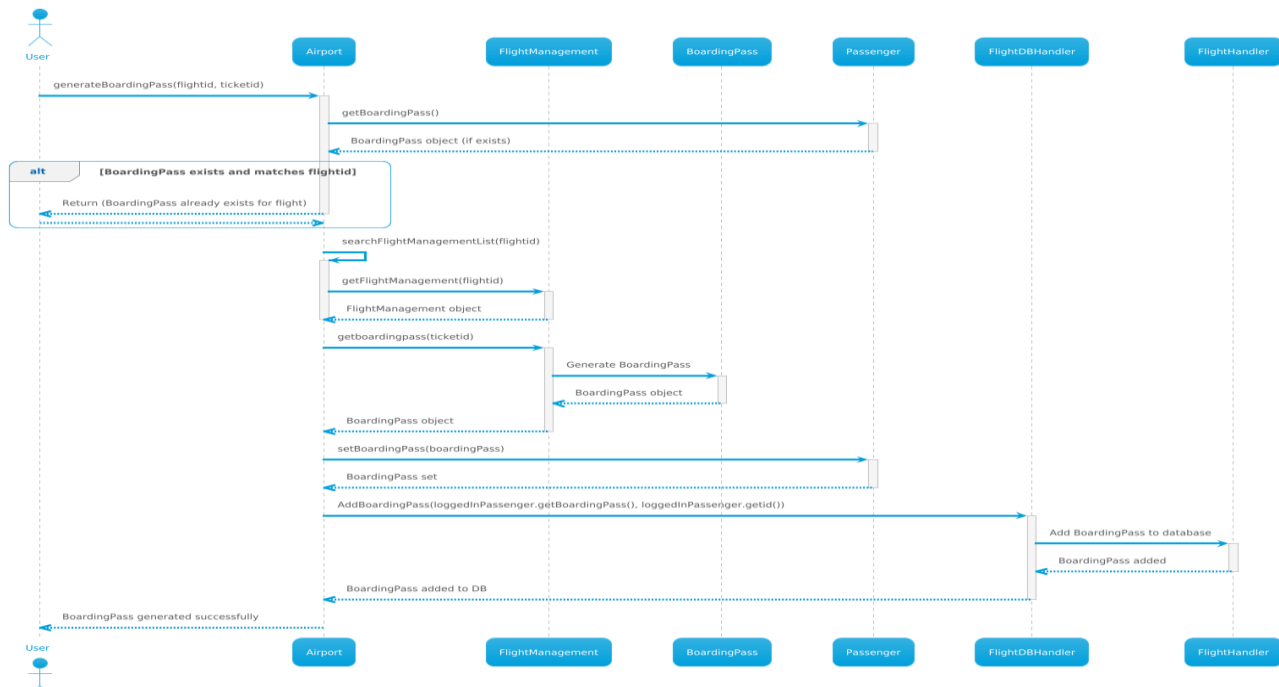
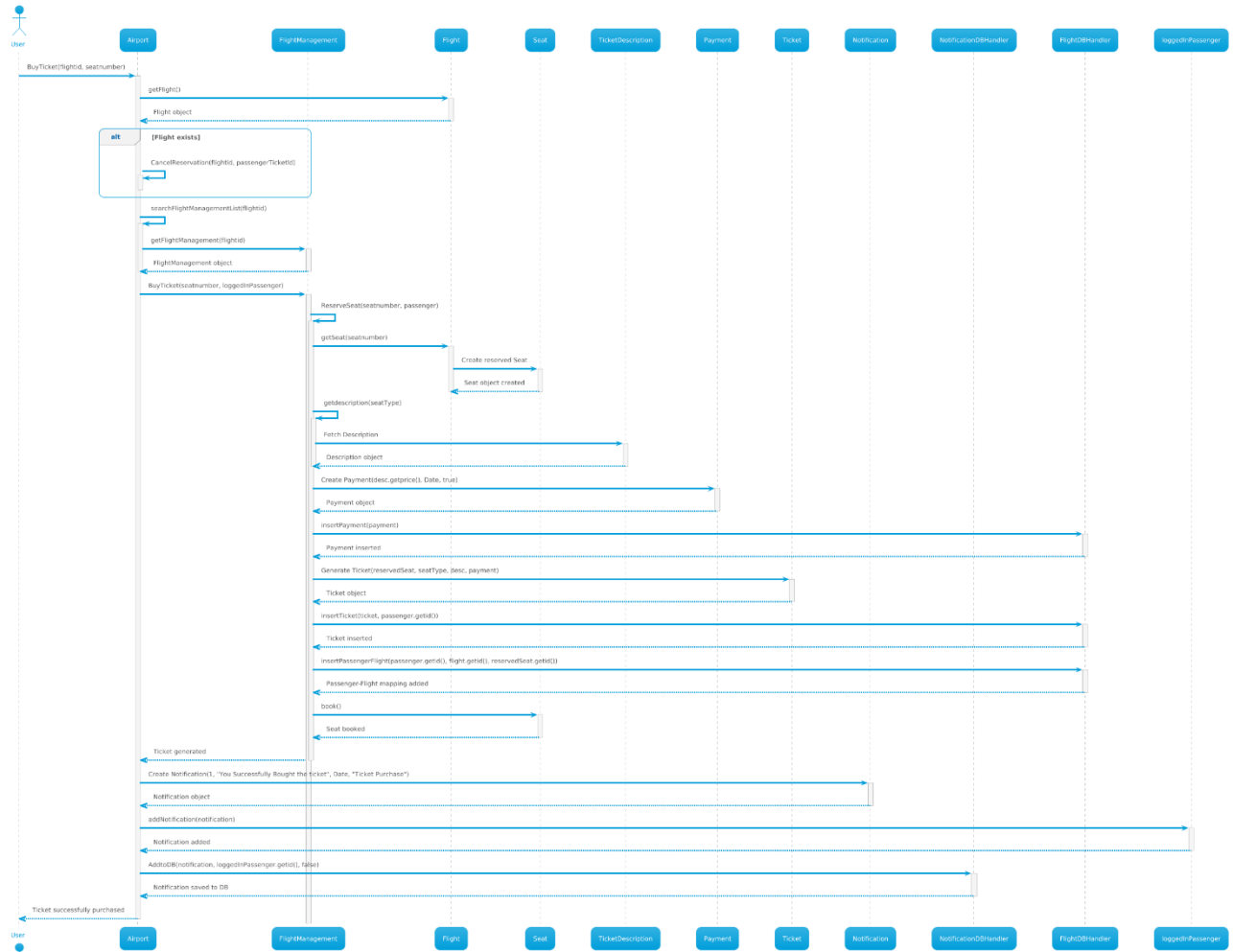


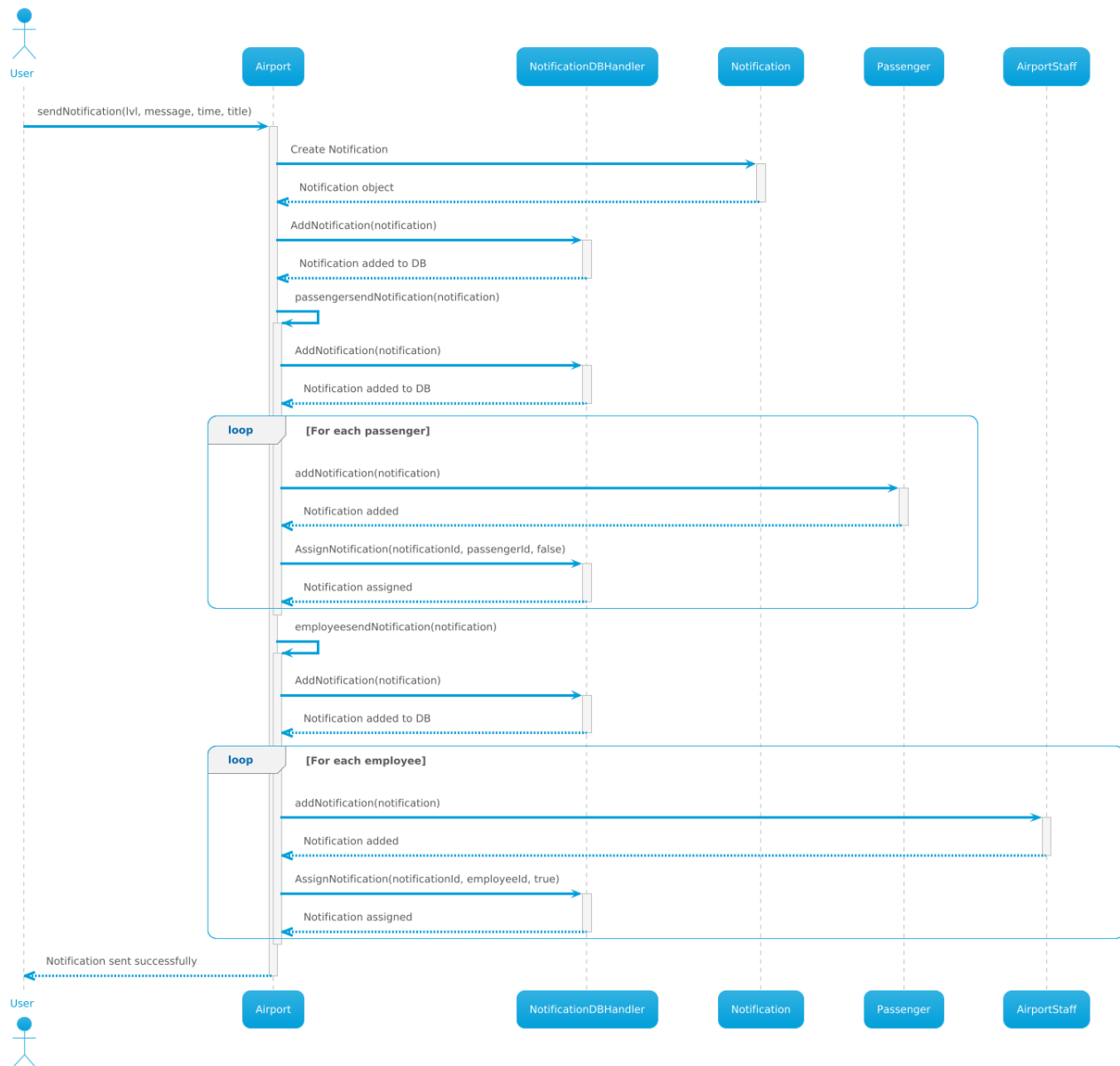




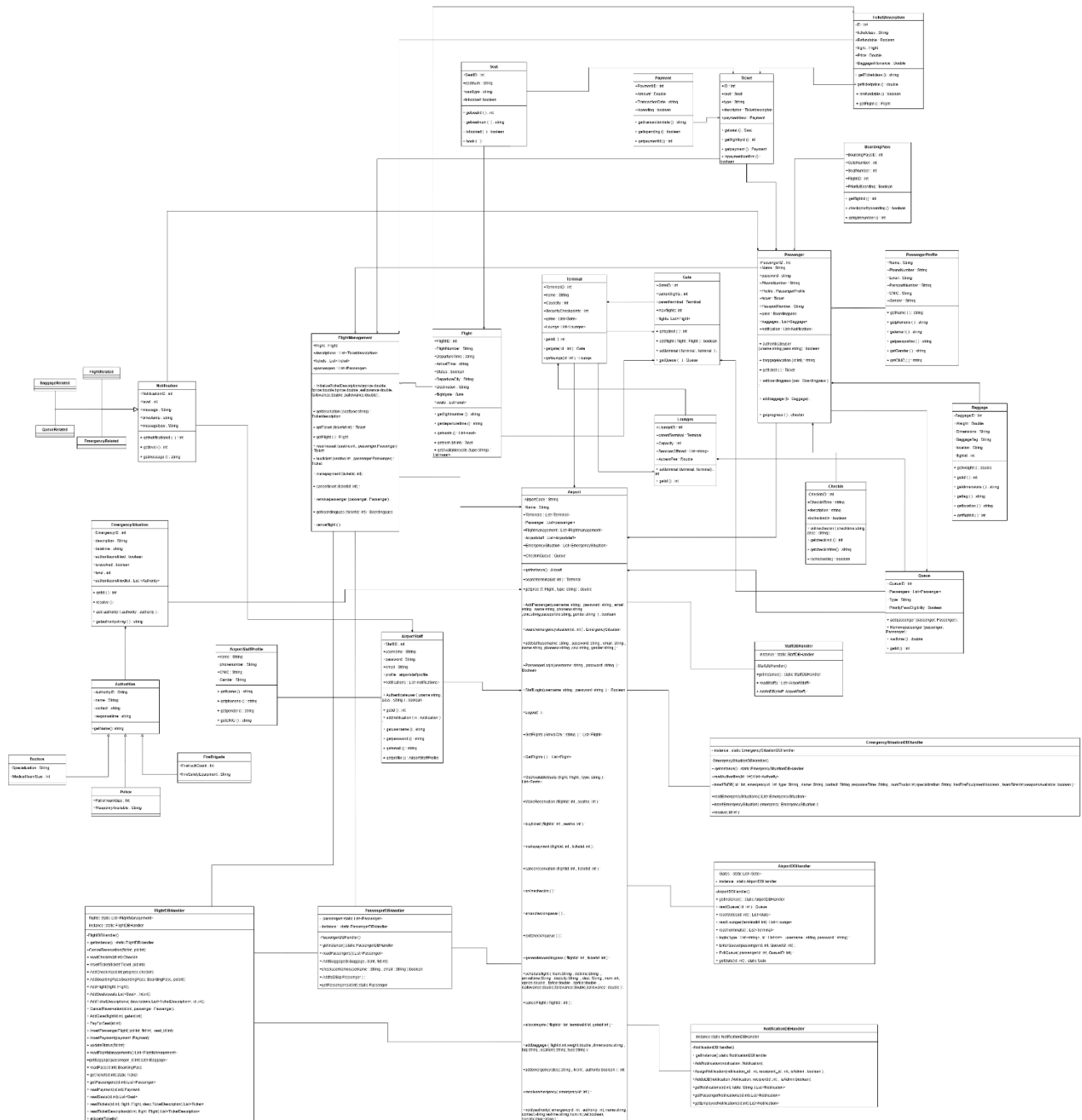
## 6. Sequence Diagram



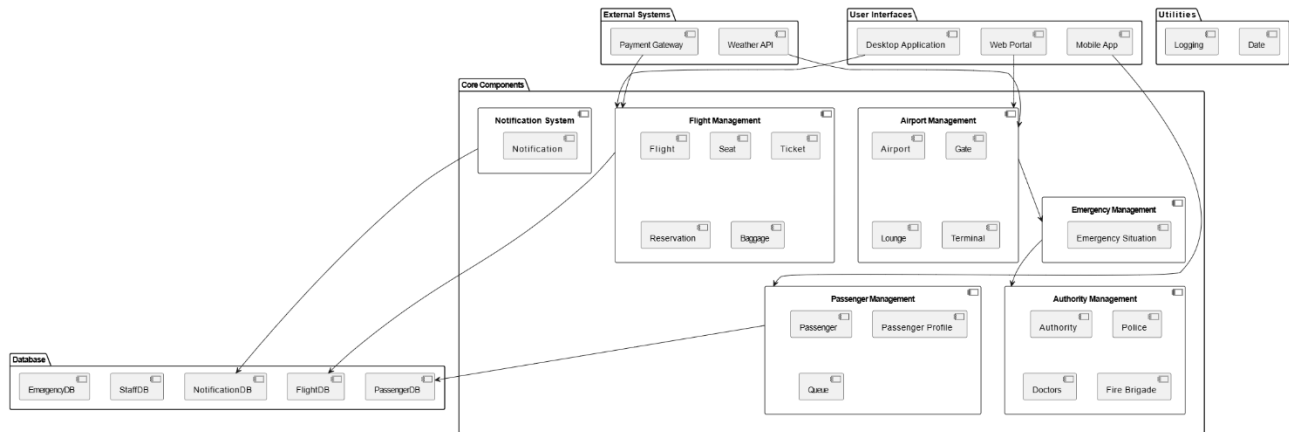




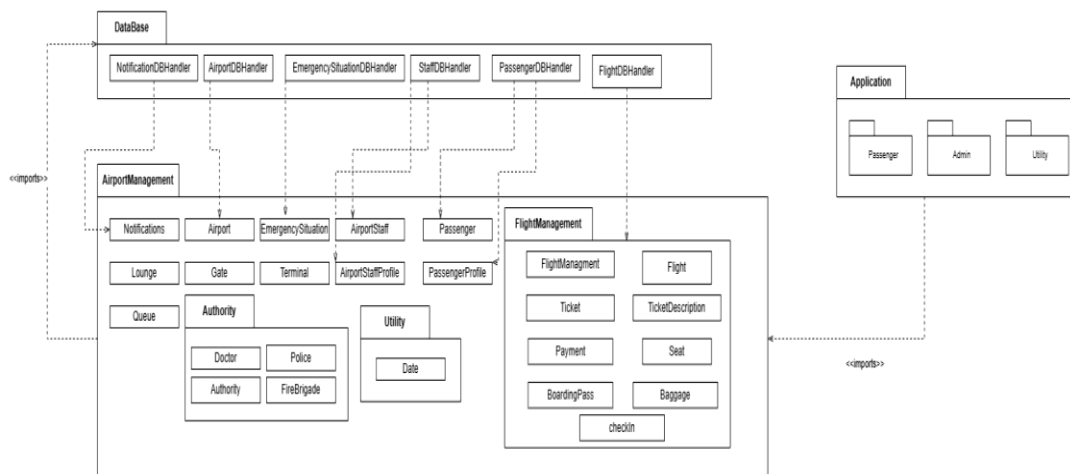
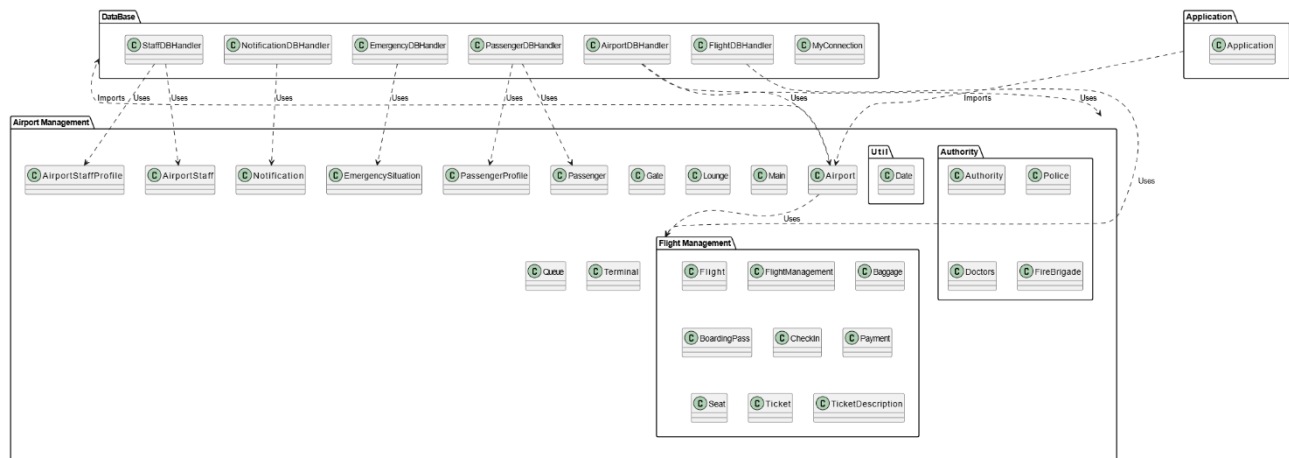
# 7. Class Diagram



## 8. Component Diagram



## 9. Package Diagram



## 10. Deployment Diagram

