

Airline Ticket Booking Requirements Specification

Version 1.0

May, 2025

Group Members:

Anxhela Kullafi

Frenki Zelo

Kledisa Collaku

Martin Shameti

Megi Selimi

Table of Contents

1. EXECUTIVE SUMMARY.....	1
1.1 Project Overview	2
1.2 Purpose and Scope of this Specification.....	2
2. PRODUCT – SERVICE DESCRIPTION	4
2.1 Product Context.....	2
2.2 User Characteristics	2
2.3 Assumptions	2
2.4 Constraints	2
2.5 Dependencies	2
3. REQUIREMENTS.....	4
3.1 Functional Requirements	5
3.2 Non - Functional Requirements	5
3.2.1 Product Requirements	6
3.2.1.1 User Interface Requirements.....	6
3.2.1.2 Usability.....	6
3.2.1.3 Efficiency	6
3.2.1.3.1 User Interface Requirements	6
3.2.1.3.1 Usability	6
3.2.1.4 Dependability	6
3.2.1.5 Security	6
3.2.2 Organizational Requirements	6
3.2.2.1 Environmental Requirements	6
3.2.2.2 Operational Requirements	6
3.2.2.3 Development Requirements	6
3.2.3 External Requirements	6
3.2.3.1 Regulatory Requirements	6
3.2.3.2 Ethical Requirements	6
3.2.3.3 Legislative Requirements	6

3.2.3.3.1 Accounting Requirements	6
3.2.3.3.1 Security Requirements	6
3.3 Domain Requirements	5
4. USER SCENARIOS – USE CASES.....	4
4.1 Major Functions	2
4.2 Use Cases	2
5. DIAGRAMS	4
5.1 Use Case Diagram	2
5.2 Activity Diagrams	2
5.3 ERD Diagram.....	2
5.4 Class Diagram	2
5.5 Component Diagram.....	2
5.5 Sequential Diagram	2

1. EXECUTIVE SUMMARY

The project involves the development of a dedicated airline ticket booking software designed specifically for a single airline. The system will allow passengers to search, filter, book flights, select seats, and make payments securely. It also provides administrative tools for managing flights, monitoring bookings, and a dedicated tech support user for system maintenance.

The software will focus on a user-friendly, secure, and efficient experience for both passengers and administrators, resembling modern airline booking systems.

1.1 Project Overview

This project focuses on creating an all-in-one airline ticket booking software aimed at simplifying the flight reservation process for passengers while offering powerful management tools for airline staff. The software will feature a range of functionalities such as ensuring seamless booking, secure payment handling etc.

Intended Audience:

- **Passengers:** Regular users booking flights.
- **Administrators:** Airline staff managing the flight and booking data.
- **Tech Support:** Personnel responsible for the software's technical health.

It will serve the following main purposes:

Passengers can:

- Register an account with their email.
- Log in and manage their account.
- Search and filter flights (by destination, date, number of persons, class).
- Select flights, choose seats, and complete payments securely.
- Receive booking confirmations via email.

Administrators (Managers) can:

- Add, modify, and delete flight details.
- Manage and oversee passenger bookings.
- Adjust ticket prices, flight schedules, and seat availability.
- View reports on flight occupancy, revenue, and performance.

Tech Support / Maintenance:

- A single responsible user tasked with ensuring the system remains operational, updating the software, and fixing any detected bugs.

The system will **not cover** additional airport services such as baggage handling, cargo logistics, or multi-airline management — focusing solely on **one airline's** booking and internal management processes.

1.2 Purpose and Scope of this Specification

The purpose of this document is to outline the requirements for an airline ticketing system. This system aims to provide seamless booking, payment, and management of airline tickets for customers while allowing airline administrators to oversee operations efficiently. Additionally, software maintenance technicians will be able to ensure the system runs smoothly by addressing any technical issues.

This specification is intended to:

- Serve as a guideline for the **development team**.
- Set **expectations** for the **end users** (passengers and admins).
- Act as a **baseline** for quality assurance and project validation.

The scope of this specification, the airline ticketing system will enable users to search for flights, book tickets, and manage their travel plans. Airline administrators will be responsible for managing flight schedules, ticket availability, and customer bookings. The software maintenance technician will handle system maintenance, updates, and troubleshooting.

- In Scope

This specification covers functionalities related to:

Passenger Services:

- Account registration, login, and profile management.
- Flight search with filters (destination, date, number of persons, seat class).
- Flight selection and seat reservation.
- Payment processing (via credit/debit card).
- Booking confirmation emails.

Administrator Services:

- Flight creation, modification, and deletion.
- Price and schedule management.
- User booking management and reporting tools.

Tech Support Services:

- System maintenance and updates.
- Bug tracking and resolution.
- Out of Scope

Management of **airport operations** (baggage, gates, lounges).

Multi-airline integration (system is for **one airline only**).

Third-party system integrations not explicitly described (e.g., external payment providers unless specified).

Future system expansions (such as hotel booking, rental car services) are **excluded** from the current scope and may be considered for future phases.

2. PRODUCT – SERVICE DESCRIPTION

Background Information

This airline ticket booking software is tailored to meet the dynamic demands of the airline industry and its passengers. With the travel sector rapidly embracing digital transformation, there is an increasing need for streamlined, intuitive, and secure booking systems. The software seeks to improve passengers' travel experiences while equipping airline teams and departments with powerful management tools.

Several key factors have shaped the development and requirements of this software:

- **User Experience:**
 - Passengers expect a simple and intuitive booking experience.
 - A clear interface for searching flights, selecting seats, and processing payments is critical.
 - Accessibility across different devices (PCs, tablets, smartphones) is important.
- **Security:**
 - Protecting user data, especially payment information, is a top priority.
 - Secure authentication and encryption mechanisms are required.
- **Real-Time Information:**
 - Passengers should see live updates about available flights, prices, and seating.
 - Flight changes or cancellations must be reflected instantly.
- **Operational Efficiency:**
 - The administrator/business manager needs efficient tools to manage flights, prices, and monitor booking trends.
 - Financial summaries and reporting features help with airline management decisions.
- **System Reliability and Maintenance:**
 - The tech support/maintenance team requires robust tools for system monitoring, updates, and troubleshooting.
 - Fast recovery in case of system issues ensures high service availability.
- **Customer Support and Notifications:**
 - Automated emails should notify passengers about booking confirmations, schedule changes, and other important updates.
 - Ensuring passengers stay informed reduces miscommunication and increases trust.

By addressing these factors, the platform ensures a **high-quality experience** for all users while maintaining strong security and operational efficiency.

2.1 Product Context

The airline ticket booking software can function as an independent, self-contained system with its primary purpose being to facilitate the booking of airline tickets for passengers.

The airline ticket booking system is an **independent, self-contained** platform focused solely on the **flight booking activities** for a **single airline**.

The core operations — including flight browsing, booking, payment, and notification — are managed entirely within the system.

However, to achieve a complete service offering, the system interfaces with the following external components:

- **Payment Gateway:**
 - Integration with a secure online payment processor for handling passenger transactions.
 - Supports debit cards, credit cards, and other secure digital payment options.
- **Email Notification System:**
 - Automated emailing system to send booking confirmations, payment receipts, and flight updates to passengers.
- **Internal Database:**
 - Secure storage of flight schedules, seat availability, passenger accounts, booking records, and payment history.

Summary of System Relationships:

- **Independent Capabilities:**
 - Complete booking cycle (flight selection, seat selection, payment, confirmation) managed internally.
- **Integrated Components:**
 - Secure Payment Processor (for financial transactions).
 - Email Notification System (for customer communications).

The system does **not** integrate with Global Distribution Systems (GDS) or third-party travel agencies, as it is intended to serve only **one airline's direct customers**.

2.2 User Characteristics

The system serves **three main types of users**, each with different expectations and technical abilities:

1. Passengers

- **Experience:**
 - Range from occasional travelers to frequent flyers.
 - Basic to moderate experience using digital booking systems.
- **Technical Expertise:**
 - Familiar with web browsing and mobile applications.
 - Expect clear navigation and fast performance.
- **General Characteristics:**
 - Prefer quick search, easy booking steps, and immediate confirmation.
 - Appreciate clear display of prices, baggage rules, and seat options.
 - Rely on timely emails for booking and schedule updates.

2. Administrator / Business Manager

- **Experience:**
 - Skilled in airline operations, booking management, and financial oversight.
- **Technical Expertise:**
 - Comfortable working with management dashboards and reporting tools.
 - Ability to create and modify flight schedules, manage pricing, view booking statistics.
- **General Characteristics:**
 - Require access to real-time booking data and financial summaries.
 - Need tools for adjusting flight schedules and prices quickly and securely.
 - Prioritize system reliability and operational transparency.

3. Tech Support / Maintenance

- **Experience:**
 - Specialized in IT systems, web server maintenance, and technical troubleshooting.
- **Technical Expertise:**
 - Advanced knowledge of backend systems, databases, software security, and server monitoring.
- **General Characteristics:**
 - Responsible for ensuring system uptime, applying updates, and fixing bugs.
 - Expected to monitor the system continuously, handle backups, and perform emergency maintenance.
 - Must ensure that data security and regulatory compliance are maintained.

2.3 Assumptions

Assumptions Affecting Requirements

1. Equipment Availability

- Assumption:
Users have access to modern computers, tablets, or smartphones.
- Impact:
The system must be responsive and compatible with various device types and screen sizes.
- Change Needed if Unavailable:
A simplified, lightweight version of the system would need to be developed for older devices.

2. Operating System

- Assumption:
Users are operating up-to-date versions of Windows, macOS, iOS, and Android.
- Impact:
The system should be tested against recent and several previous OS versions.
- Change Needed if Unavailable:
Expanded compatibility testing for older operating systems would be necessary.

3. Internet Access

- Assumption:
Users have stable and reliable internet access.
- Impact:
The system will be designed as a web-based platform requiring continuous internet connectivity.
- Change Needed if Unavailable:
Offline features or a downloadable mobile application might need to be considered.

4. Browser Compatibility

- Assumption:
Users utilize modern web browsers (e.g., Chrome, Firefox, Safari, Edge).
- Impact:
The platform will be optimized for these browsers.
- Change Needed if Unavailable:
Additional browser testing and fallback support for older browsers would be required.

5. Technical Expertise

- Assumption:
Users have a basic familiarity with online booking systems.
- Impact:
The user interface will focus on ease-of-use but assumes basic web navigation knowledge.
- Change Needed if Unavailable:
A more guided user experience (e.g., tutorials, onboarding help) would need to be integrated.

6. Payment Methods

- Assumption:
Users have access to digital payment methods such as debit/credit cards and digital wallets.
- Impact:
Integration with standard payment gateways is sufficient.
- Change Needed if Unavailable:
Manual payment options or support for alternative payment methods would be required.

7. Airline Reservation System Integration

- Assumption:
The system will primarily manage its own flight inventory without needing external GDS (Global Distribution Systems) integration.
- Impact:
Simplified internal flight management and booking processes.
- Change Needed if Unavailable:
External data source integration would need to be developed if broader flight options become necessary.

8. Security Standards

- Assumption:
Industry-standard security protocols (SSL/TLS encryption, PCI-DSS for payments) will be implemented.
- Impact:
Ensures user data protection and secure transactions.
- Change Needed if Unavailable:
Custom security protocols or enhanced compliance checks would be needed.

9. Regulatory Compliance

- Assumption:
The system complies with GDPR and local aviation industry regulations.

- Impact:
Maintains user trust and legal compliance.
- Change Needed if Unavailable:
Significant modifications to data handling and privacy mechanisms would be required.

Summary

These assumptions directly influence the software's design, features, and security. If any assumption proves invalid, the system requirements will need to be revised to maintain functionality, usability, and compliance.

2.4 Constraints

Design Constraints

1. Parallel Operation with Existing Systems

- Description:
The booking platform may need to operate alongside limited legacy systems during initial rollout.
- Impact:
 - Data Synchronization:
Accurate syncing between old and new booking records.
 - Compatibility:
New system must accept data formats used by legacy systems.
 - User Training:
Clear training must be provided for users interacting with both systems temporarily.

2. Audit Functions (Audit Trail, Log Files, etc.)

- Description:
Comprehensive tracking of all significant system events and user actions.
- Impact:
 - Data Logging:
Track creation, modification, cancellation of bookings with timestamps and user IDs.
 - Security:
Logs must be tamper-proof and accessible only to administrators.
 - Performance:
Logging should not interfere with booking speeds or overall system performance.
 - Compliance:
Meet regulations on auditability and data retention.

3. Access Management and Security

- Description:
Robust security controls must be implemented.
- Impact:
 - Authentication:
Multi-factor authentication for admin and critical functions.
 - Role-Based Access Control:
Different permissions for passengers, administrators, and support personnel.
 - Encryption:
Protect sensitive data both at rest and in transit.
 - Monitoring:
Detect unauthorized access attempts and log security events.
 - Compliance:
Meet GDPR and other relevant data protection laws.

4. Criticality of the Application

- Description:
The application is mission-critical for airline operations and customer trust.
- Impact:
 - High Availability:
Redundant servers and failover plans to minimize downtime.
 - Performance:
Ability to handle peak loads during promotions or peak seasons.
 - Backup and Recovery:
Regular automated backups and fast disaster recovery mechanisms.

5. System Resource Constraints

- Description:
Limited hardware or cloud infrastructure budget initially.
- Impact:
 - Optimization:
Efficient use of server resources, memory, and storage.
 - Scalability:
Easy scalability as passenger volumes increase.
 - Hardware Compatibility:
Must work with existing hosting or cloud platforms initially.

6. Other Design Constraints

- Programming Language / Framework:
System will be developed using modern frameworks like ASP.NET Core or Django.
- Standards and Guidelines:

- Follow web accessibility (WCAG) and usability best practices.
- Comply with secure coding guidelines and API standards.
- Integration Standards:
APIs and external systems must meet secure integration requirements.

Summary

These constraints influence technical design, security practices, and deployment strategies. Managing these constraints carefully is critical for delivering a secure, user-friendly, and high-performance ticket booking system.

2.5 Dependencies

Key Dependencies Affecting Requirements

1. Payment Gateway Integration

- Dependency:
Successful integration with secure online payment gateways (e.g., Stripe, PayPal).
- Impact:
 - Security and Compliance:
Must ensure PCI-DSS compliance.
 - Testing:
Thorough end-to-end testing of all payment flows is mandatory.
 - User Experience:
Seamless and trustworthy checkout process for users.

2. Email Notification System

- Dependency:
Integration with a reliable email service provider (e.g., SendGrid, AWS SES).
- Impact:
 - Notification Delivery:
Confirmations, flight changes, and other critical updates must be sent instantly.
 - Reliability:
High deliverability rates and email tracking are important for customer satisfaction.

3. Hosting and Infrastructure

- Dependency:
Availability of reliable web hosting or cloud services (e.g., AWS, Azure).
- Impact:
 - Scalability and Reliability:
Infrastructure must support high traffic without failures.

- Monitoring:
Proactive server monitoring to detect and address issues quickly.

4. Regulatory Compliance

- Dependency:
Adherence to GDPR, local aviation regulations, and cybersecurity standards.
- Impact:
 - Data Handling:
Secure, compliant storage and processing of user data.
 - Consent Management:
Proper mechanisms for obtaining user consent for data usage.
 - Audit Trails:
Maintain proper audit records to meet legal requirements.

Summary

The successful development and operation of the airline ticket booking system depends heavily on third-party services (payment, email, hosting) and strict regulatory compliance. Delays or failures in these areas could directly impact system launch and reliability.

3. REQUIREMENTS

3.1 Functional Requirements

ID	Requirement	Comments	Priority
1	Passengers must be able to register and authenticate via email and password.	Includes email registration, secure password rules, and prevention of duplicate accounts.	1
2	Passengers must be able to log in to the system securely.	Includes wrong input handling, and session redirection.	1
3	Tech Support must be able to log in securely with dedicated credentials.	Enables system monitoring and troubleshooting.	2
4	Admin must be able to log in securely with privileged access.	Access to dashboards, flight management, and report generation.	1
5	Passengers can search for flights by origin, destination, dates, class, etc.	Includes filters like airline, stops, and time range.	1
6	Passengers can book flights and enter passenger information.	Includes contact info, extras (meals, baggage), and booking summary.	1
7	Passengers can select seats during the booking process.	Interactive seating map or grid-based picker with class preference.	2
8	Passengers can make secure payments after booking flights.	Payment made via Credit or Debit card.	1
9	Passengers receive booking confirmations and e-tickets.	Sent via email; includes ticket number and invoice.	1
10	Passengers can reset their password securely.	Email-based reset with secure time-limited tokens.	1
11	Admin can add, update, or delete flights and manage availability.	Includes scheduling, pricing, and editing.	1
12	Admin can cancel flights and notify passengers.	Provides reasons for cancellation and options for refund.	1
13	Admin can modify flights and issue compensation.	Affects active bookings; compensation applied automatically or manually.	1

14	Admin can access system analytics dashboards.	Includes traffic data, usage trends, and filtering by date/feature.	2
15	Admin can generate statistical reports.	Aggregates booking trends, reviews, and flight frequency data.	2
16	Admin can access and download financial reports.	Data includes revenue, transactions, refunds; charts and tables available.	1
17	Passengers can enroll in a loyalty program.	Allows earning/redeeming points, upgrading tiers, and viewing balances.	3
18	Passengers can check in online during the valid window.	Generates digital boarding passes with QR codes.	2
19	Tech Support can monitor and maintain system performance.	Diagnoses issues, applies patches, and logs resolutions.	2

3.2 Non – Functional Requirements

3.2.1 Product Requirements

3.2.1.1 User Interface Requirements

- The system must offer a modern, responsive, and visually consistent interface that ensures a seamless experience across all major web browsers (Chrome, Firefox, Safari, Edge) and platforms (Windows, macOS, Linux, iOS, Android). The interface should adapt gracefully to various screen sizes and device resolutions.
- All interface components, including buttons, forms, dropdowns, and navigation panels, must follow universally accepted design principles to promote familiarity and user confidence.
- Search results for flights should be presented in a clean, tabulated format with clearly labeled attributes such as departure time, arrival time, ticket price, and flight class.
- Error messages should be informative, concise, and visually distinct. They must specify the nature of the issue and suggest steps the user can take to resolve it.
- Administrative dashboards should be structured logically with sortable tables, filter options, and exportable reports, enabling quick decision-making and efficient management of operational tasks.

3.2.1.2 Usability

- The user interface must be intuitive and user-friendly, allowing even first-time users to perform essential functions such as registration, booking, and payment without requiring technical support or training.
- Tooltips, progress indicators, and inline validation should be integrated throughout the interface to assist users in completing actions correctly and confidently.
- A comprehensive help section must be available, featuring frequently asked questions (FAQs), illustrated walkthroughs, video tutorials, and searchable guidance articles.
- The overall design should minimize user effort by reducing the number of steps required to complete core processes. For instance, the complete flight booking process from search to confirmation should not exceed five user actions.

3.2.1.3 Efficiency

3.2.1.3.1 Performance Requirements

- The platform must be capable of handling heavy concurrent traffic, supporting a minimum of 500 users simultaneously without noticeable delays or degradation in performance.
- All key operations such as user login, flight searches, and payment processing must complete within two seconds in 95% of cases under normal system loads.
- The response time for critical backend services such as booking confirmation and transaction processing should not exceed three seconds, even under peak load conditions.
- Administrative reporting tools must generate detailed analytics or summaries in under five seconds when processing datasets up to 10,000 entries.

3.2.1.3.2 Space Requirements

- The entire system should remain efficient in storage utilization, limiting server-side storage consumption to under 10 GB within the first year of operation.
- System-generated log files must be automatically rotated on a daily basis, compressed for efficient storage, and archived monthly. These logs should be securely deleted after a six-month retention period unless extended by legal or compliance policies.

- Uploaded user documents, transaction records, and downloadable content must be stored in an optimized format and subjected to regular clean-up routines to avoid bloat and redundancy.

3.2.1.4 Dependability

Availability

- The system must guarantee high availability with a minimum monthly uptime of 99.5%. This includes ensuring access to services such as account management, flight bookings, and support features.
- Any planned downtime or maintenance activities must be communicated to users at least 48 hours in advance and should not exceed a total of four hours per month.

Reliability

- The application should be capable of handling intermittent network issues and minor server disruptions without causing service interruptions or data loss. The system should be resilient and capable of self-recovery in such cases.
- Failures must be limited to no more than one occurrence per 1,000 transactions. Each failure must be recorded in detail and trigger appropriate alerts.

Monitoring

- Real-time monitoring tools must continuously track the system's performance, flagging abnormal conditions, and notifying technical support teams via automated alerts.
- Comprehensive logs should be maintained for all login attempts, payment activities, and any system errors. These logs must include timestamps, user IDs, and event specifics to assist in diagnostics and audits.

Maintenance

- The software must be designed in a modular fashion to allow quick and safe updates. New features or patches should be deployable independently, without requiring full system redeployment.

- Maintenance activities such as updates, bug fixes, and configuration changes should be executable with minimal user disruption.
- The platform must support rollback capabilities to previous stable versions in case of deployment issues.

3.2.1.5 Security

- All passwords must be stored using secure hashing algorithms such as bcrypt or SHA-256, combined with salting techniques to prevent brute-force attacks.
- The entire system must use HTTPS for secure data transmission, and internal APIs must be protected with authentication tokens and TLS encryption.
- Role-based access control (RBAC) must be implemented to define user permissions and restrict access to sensitive data and administrative functions.
- Admin and Tech Support users must be required to use two-factor authentication (2FA) for added protection.
- Payment information must never be stored on the system's servers. All financial transactions should be handled through PCI-DSS-compliant gateways, and only non-sensitive data should be retained for reference and reporting.

3.2.2 Organizational Requirements

3.2.2.1 Environmental Requirements

- The platform must be hosted in a containerized infrastructure such as Docker, with compatibility for popular cloud providers including AWS, Azure, and Google Cloud Platform.
- Redundancy must be supported across geographic regions to ensure continuous service in case of local outages or natural disasters.

3.2.2.2 Operational Requirements

- Admins must have access to a secure, role-restricted dashboard from which they can view system logs, usage reports, and alert messages.
- The system must allow dynamic configuration of features such as toggling promotions, adjusting booking thresholds, or controlling logging levels without redeployment.

- Operations staff must be able to export log files in human-readable formats such as CSV, JSON, or XML for third-party audits or compliance checks.

3.2.2.3 Development Requirements

- The development team must follow Agile methodologies with sprint-based planning and bi-weekly review cycles.
- All project code must be tracked using Git, with branching strategies and pull request reviews enforced as part of continuous integration practices.
- Automated testing tools must be integrated into the development pipeline to ensure code quality, security compliance, and maintainability.

3.2.3 External Requirements

3.2.3.1 Regulatory Requirements

- The platform must comply fully with the General Data Protection Regulation (GDPR) regarding data retention, user consent, right to erasure, and access controls.
- Privacy policies and terms of service must be made readily accessible and updated transparently when changes occur.

3.2.3.2 Ethical Requirements

- The application must not exploit user data for commercial gain without explicit consent. This includes ensuring that no personal or behavioral data is sold or shared with third parties without prior approval.
- Automated features such as refund eligibility, seat assignment, or bonus calculation must operate transparently, and users must be able to review how decisions are made.

3.2.3.3 Legislative Requirements

3.2.3.3.1 Accounting Requirements

- All transactions must be itemized, time-stamped, and associated with a specific user and session. This applies to both successful and failed attempts.
- Refunds and compensations must include justifications and processing logs to ensure legal accountability.
- Monthly reports generated for accounting purposes must comply with industry-standard formats and be exportable to spreadsheets or financial software.

3.2.3.3.2 Security Requirements

- All database changes made by admin users must be recorded with before-and-after data snapshots, along with timestamps and operator IDs.
- The system must implement logging for all sensitive activities including user role updates, system configuration changes, and payment API calls.
- Access logs and audit trails must be securely stored for a minimum of six months and only accessible to users with appropriate clearance.

3.3 Domain Requirements

- The system must incorporate airline-specific capabilities such as frequent flyer enrollment, point redemption, real-time seat availability tracking, and class-based ticket differentiation.
- Industry-standard practices such as IATA flight codes, baggage rules, and cancellation policies must be integrated into the system's operational logic.
- Integration with third-party tools such as airport data services or airline API feeds must be supported for up-to-date flight scheduling and disruption alerts.
- The loyalty program must reflect industry-standard models, allowing for dynamic tier upgrades, retroactive point accrual, and partner airline collaboration.

These non-functional requirements ensure the airline booking platform is not only feature-rich but also secure, scalable, and reliable—ensuring a smooth and satisfying experience for all stakeholders involved.

4. USER SCENARIOS – USE CASES

4.1 Major Functions:

1. Account Management and Authorization

- **Registration and Login:**
Users can create accounts, securely log in, and recover their password if needed.
- **Profile Management:**
Users can manage their personal information, update contact details, and change or add payment methods.

2. Passenger Services and Feedback

- **Flight Search and Booking:**
Passengers can search for flights by various criteria (e.g., destination, date, class), view detailed flight information, and proceed with booking and payment.
- **Booking Management:**
Passengers can cancel, modify, or rebook flights if needed. They can also view the status of their booking (confirmed, pending, canceled).
- **Feedback and Ratings:**
Passengers can provide feedback about their flight experience and rate their service through the system. This feedback helps improve the service quality.

3. Financial Transaction Management

- **Revenue Performance Reports:**
Finance teams can access comprehensive reports on the system's revenue performance, including charts and graphs showing data by route, booking time, and other relevant metrics.

4. Flight Management by Air Control

- **Flight Planning and Management:**
Authorized personnel can create and modify flight plans. This includes adding or deleting flights, modifying departure/arrival times, aircraft details, and other key flight information.

5. Operator and Report Management by Managers

- **User Account Oversight:**
Managers can oversee user accounts (except admins), approve changes, and ensure proper security and compliance. They use specific authorization codes for these actions.
- **Performance Monitoring and Reporting:**
Managers can track system performance, review operator performance, access operational data, and generate reports based on customer feedback, flight frequency, and booking trends.

6. Customer Support by Operators

- **Customer Support Information:**

The system provides passengers with easy access to customer support contact information (e.g., phone, email, live chat).

- **Notifications:**

Automated notifications are sent to passengers about booking confirmations, cancellations, or any changes to their flight. These are sent via email or SMS as per the user's preferences.

4.2 Use Cases

0. Admin Authentication:

The system shall enable user (Admin) to securely authenticate himself/herself and gain access to the account.

UC Name	Admin Login UC-001
Summary	This use case allows the registered system administrator to securely log in to the platform in order to access administrative functionalities.
Dependency	None
Actors	Primary Actor: Admin Secondary Actor: System
Preconditions	<ul style="list-style-type: none">✓ The administrator account has already been created and assigned appropriate credentials.✓ The administrator possesses valid login information (email and password).✓ The system is operational and accessible.
Description of the Main Sequence	Step 1: Administrator accesses the login page. Step 2: Administrator enters their credentials. Step 3: System verifies the provided credentials. Step 4: If valid, the system grants access to the administrator dashboard.

Description of the Alternative Sequence	Invalid Credentials Step 1: If the provided credentials are incorrect, the system denies access. Step 2: System displays an error message prompting the administrator to retry.
Non functional requirements	Security: Login credentials must be encrypted. Performance: Authentication should complete in under 2 seconds. Reliability: Ensure secure and consistent login functionality.
Postconditions	The administrator is successfully logged in and redirected to the admin dashboard. Unsuccessful attempts result in error messages and logging of attempts.

The system shall enable user (Tech support) to securely authenticate himself/herself and gain access to the account.

UC Name	Tech Support Login UC-002
Summary	This use case allows the registered tech support personnel to log into the platform to monitor, diagnose, and manage technical issues within the system.
Dependency	None
Actors	Primary Actor: Tech Support Secondary Actor: System
Preconditions	<ul style="list-style-type: none">✓ The tech support account is pre-configured with system access rights.✓ The tech support staff possesses valid credentials.✓ The system is operational and accessible.

Description of the Main Sequence	Step 1: Tech support opens the login page. Step 2: Staff member inputs valid login credentials. Step 3: System authenticates the credentials. Step 4: Upon successful verification, the tech support dashboard is displayed.
Description of the Alternative Sequence	Invalid Credentials: Step 1: Credentials do not match system records. Step 2: System returns an error message.
Non-functional requirements	Security: Strong authentication and encryption must be enforced. Performance: Login process should execute within 2 seconds. Auditability: All access attempts must be logged.
Postconditions	Tech support is logged in and can begin monitoring system performance. Failed login attempts are logged and monitored for potential security issues.

1. Passenger Registration & Authentication:

The system shall enable users (Passengers) to initiate the creation of new accounts.

UC Name	Passenger Registration UC-101
Summary	Enabling passengers to register & create new accounts within the system.
Dependency	None
Actors	Primary Actor: User (Passenger) Secondary Actor: System

Preconditions	<p>1.The user attempts to sign up using personal information.</p> <p>2.The user’s unique account identifiers (email) do not match with another user’s in the system.</p> <p>3.The user agrees to the terms and conditions of service before proceeding with the sign-up process.</p> <p>4. The user provides all required information fields (such as name, email, password) during the sign-up attempt.</p>
Description of the Main Sequence	<p>Step 1: The user navigates to the sign-up page on the system.</p> <p>Step 2: The user fills in the required personal information such as name, email address, and password.</p> <p>Step 3: The system validates the entered information to ensure all required fields are filled correctly.</p> <p>Step 4: The system checks if the provided email address is unique and not already associated with an existing account.</p> <p>Step 5: The user receives a confirmation message indicating successful account creation.</p> <p>Step 6: The user can now log in to the system using their email address and password.</p>
Description of the Alternative Sequence	<p>Alternative sequence 1-Email already in use: If the email is already registered, the system notifies the user and prevents registration.</p> <p>Alternative sequence 2-Missing required fields: If any required field is left blank, the system prompts the user to fill it.</p>

Non functional requirements	<p>Security: The security requirement for the sign-up process ensures the protection of user accounts and sensitive information from unauthorized access and potential breaches.</p> <p>Performance: The authentication process is expected to conclude within a specified duration of seconds.</p> <p>Scalability: The authentication database system should be capable of handling a large number of accounts.</p>
Postconditions	<p>A new user account is created and stored in the system.</p> <p>The user is authenticated and redirected to the dashboard (or login page).</p> <p>User's email is marked as verified in the system.</p>

Login-The system shall allow users and passengers to securely authenticate themselves and gain access to their accounts.

UC Name	Passenger Account Log-In UC-102
Summary	Enabling passengers to securely access their accounts within the system.
Dependency	User Registration (UC-101)
Actors	Primary Actor: User/Passenger Secondary Actor: System

Preconditions	<p>1.The user(passenger) has already created an account with the system or has been given one by the Administrator of the Airline.</p> <p>2.The user possesses valid credentials: registered email address &password.</p> <p>3.The user has agreed to the terms and conditions before attempting log in.</p> <p>4.The system is operational and functional in order to allow login.</p>
Description of the Main Sequence	<p>Step 1: The user enters their credentials to log in to the system.</p> <p>Step 2: The system verifies the provided credentials against the stored user data.</p> <p>Step 3: If the credentials match an existing user account, the system grants access to the user interface.</p>
Description of the Alternative Sequence	<p>Alternative Sequence 1: If the credentials do not match or are invalid, the system denies access and prompts the user to try again or reset their password.</p>
Non functional requirements	<p>Security: The security requirement for the login process ensures the protection of user accounts and sensitive information from unauthorized access and potential breaches.</p> <p>Performance: The authentication process is expected to conclude within a specified duration of seconds.</p> <p>Scalability: The authentication database system should be capable of handling a large number of accounts.</p>
Postconditions	<p>1. If the user's credentials are valid, they gain access to their account.</p> <p>2. Upon successful login, the system may redirect the user to their account dashboard.</p> <p>3. Invalid credentials result in access denial with error messages.</p>

	4. After multiple failed login attempts (maybe 3 or 4) , the system may lock the account for security.
--	--

System shall allow passengers/users to reset their account password

UC Name	User Password Reset UC-103
Summary	Enabling passengers to securely reset their accounts within the system.
Dependency	User Account Creation (UC-101)
Actors	Primary Actor: User Secondary Actor: System
Preconditions	The user has a registered account in the system. The user initiates reset process from the login or help screen. The user provides a valid, registered email address. The system must be able to send emails to the user's address.
Description of the Main Sequence	Step 1: The user clicks the "Forgot Password" button. Step 2: User enter his email. Step 3: The system verifies that the user email exists in the user database. Step 4: The system prompts the user to enter and confirm a new password.

	<p>Step 5: The system validates the new password (e.g., complexity rules, matching confirmation).</p> <p>Step 6: The system updates the user's password in the database.</p> <p>Step 6: The user receives confirmation that their password has been successfully reset.</p> <p>Step 7: The user can now log in using the new password.</p>
Description of the Alternative Sequence	<p>Alternative Sequence: Invalid Credentials</p> <p>Step 1: The user enters again the credentials.</p> <p>Step 2: The system checks the email/password against the defined policy.</p> <p>Step 3: If not correct, the system rejects the password and displays message.</p> <p>Step 4: The user is prompted to re-enter a stronger password.</p> <p>Step 5: If the new password still does not meet requirements, the system repeats Steps 2-5.</p> <p>Step 6: Once the user enters a password that meets the criteria, the system proceeds to the next step in the Main Sequence (Step 4).</p>
Non functional requirements	<p>Security: The security requirement for the reset process ensures the protection of user accounts and information from unauthorized access and potential breaches.</p> <p>Usability: Maintain an intuitive user interface.</p>
Postconditions	<p>The user has a registered account in the system.</p> <p>The user initiates the password reset process from the login or help screen.</p> <p>The user provides a valid, registered email address.</p> <p>The system must be able to send emails to the user's address.</p>

2. Flight Search & Filtering:

The system shall enable users (Passengers) to initiate the creation of new accounts.

UC Name	Filter Flights UC-201
Summary	The system must enable users to search for different flights efficiently. This feature allows users to specify their travel preferences and find relevant flight options based on criteria such as origin, destination, departure date, price, and class. By providing robust flight search functionality, the system enhances user experience and facilitates seamless flight booking.
Dependency	User Registration (UC 101)
Actors	Primary Actor: Passengers
Preconditions	Before utilizing the flight search functionality, users must be authenticated and logged into their account within the airline software, ensuring that only authorized users can access and utilize the search feature.
Description of the Main Sequence	<p>Step 1: User Input</p> <p>Users input their flight preferences, such as departure city, destination, travel dates, price, and class, into the search interface.</p> <p>Step 2: Query Submission</p> <p>Upon inputting preferences, users submit their search query by clicking a search button or similar action.</p> <p>Step 3: Search Processing</p>

	<p>The system processes the search query, analyzing the user's input criteria to retrieve relevant flight options from the database.</p> <p>Step 4: Flight Retrieval</p> <p>Based on the search criteria, the system retrieves available flight options that match the user's preferences, including flight schedules, fares, and availability.</p> <p>Step 5: Display Results</p> <p>The system displays the retrieved flight options in a clear and organized manner, presenting essential details such as departure times, arrival times, airlines, and ticket prices.</p> <p>Step 6: Selection</p> <p>Users review the displayed flight options and select the one that best fits their preferences and requirements.</p>
Description of the Alternative Sequence	<p>Step 1: User Input</p> <p>Users input their flight preferences, such as departure city, destination, travel dates, and class, into the search interface.</p> <p>Step 2: Query Submission</p> <p>Upon inputting preferences, users submit their search query by clicking a search button or similar action.</p> <p>Step 3: Search Processing</p> <p>The system processes the search query, analyzing the user's input criteria to retrieve relevant flight options from the database.</p> <p>Step 4: Flight Retrieval</p> <p>The system checks the database for available flight options that match the user's preferences.</p> <p>Step 5: No Matching Flights Found</p> <p>If there are no flights that match the user's input criteria, the system informs the user that no matching flights were found.</p>

	<p>Step 6: Return to Search</p> <p>Users have the option to return to the flight search interface to modify their search criteria and perform a new search.</p>
Nonfunctional requirements	<p>Performance: Ensure fast response times.</p> <p>Usability: Maintain an intuitive user interface.</p> <p>Accessibility: Comply with accessibility standards.</p> <p>Reliability: Provide accurate and reliable search results.</p> <p>Scalability: Handle increasing user load without performance degradation.</p> <p>Security: Protect user data during search queries.</p>
Postconditions	<p>After selecting a flight, users are seamlessly guided through the booking process, where they can confirm their flight selection, provide necessary passenger information, and complete the reservation. Upon successful booking, users receive a confirmation of their flight reservation along with relevant booking details.</p>

3. Flight Booking and Payment:

The system shall allow the user to book a selected flight

UC Name	Book Flight UC-301
Summary	<p>The system must enable users to book a selected flight seamlessly. This feature allows users to confirm their flight selection, provide passenger details, and complete the reservation process efficiently within the airline software. By providing robust flight booking functionality, the system enhances user experience and facilitates bookings</p>

Dependency	User Registration & Login (UC 101&102) Filter Flights (UC 201)
Actors	Primary Actor: Passenger
Preconditions	Passenger has searched the flight and selected to see the details for that flight.
Description of the Main Sequence	<p>Step 1: Flight Selection</p> <p>After searching for and selecting a desired flight, users proceed to book the chosen flight.</p> <p>Step 2: Flight Details Review</p> <p>Users review the details of the selected flight, including departure and arrival times, fares, and other relevant information.</p> <p>Step 3: Passenger Information</p> <p>Users provide necessary passenger details such as names, contact information, and any special requirements.</p> <p>Step 4: Seat Selection</p> <p>If seat selection is available, users may choose their preferred seats or seating options for the flight.</p> <p>Step 6: Booking</p> <p>Users review flight summary and finalize the flight booking.</p>
Description of the Alternative Sequence	<p>Flight Unavailability</p> <p>Step 1: If the selected flight becomes unavailable due to real-time updates, the system notifies the user.</p>

	Step 2: User is taken back to the Flight Filtering tab where he/she searches for similar alternative flights.
Nonfunctional requirements	<p>Performance: Ensure fast response times during the booking process.</p> <p>Usability: Maintain an intuitive and user-friendly interface for seamless booking.</p> <p>Accessibility: Comply with accessibility standards to accommodate users with disabilities.</p> <p>Reliability: Provide reliable booking functionality with minimal downtime.</p> <p>Scalability: Handle concurrent booking requests from multiple users without performance degradation.</p> <p>Error Handling: Effectively handle errors and edge cases during booking to ensure a smooth user experience.</p> <p>Availability: Ensure the booking system is available 24/7 to accommodate users.</p>
Postconditions	After successfully completing the booking process, users receive a summary of their booking and proceed to the payment step

The system shall allow the user to pay for a booked flight

UC Name	Flight Payment Processing UC-302
Summary	The system must enable users to book a selected flight seamlessly. This feature allows users to confirm their flight selection, provide passenger details, and complete the reservation process efficiently within the airline software. By providing robust flight booking functionality, the system

	enhances user experience and facilitates hassle-free flight reservations.
Dependency	Secure Payment Transactions
Actors	Primary Actor: Passenger
Preconditions	Passenger has booked the particular flight. (Optionally- has redeemed loyalty points)
Description of the Main Sequence	Step 1: User selects the preferred payment method (credit/debit card). Step 2: User enters payment details (e.g., card number, expiry date, CVV). Step 3: System validates the payment details. Step 4: If payment is successful, the system generates a booking confirmation. Step 5: User is redirected to the confirmation page with booking details and generated ticket from those details.
Description of the Alternative Sequence	Alternative Sequence: Payment Failure, incorrect credentials Step 1: System attempts to process the payment but detects incorrect payment details (for example wrong card number format). Step 2: System displays an error message informing the user of the failed transaction. Step 3: User is prompted to try again. Step 4: If the user successfully completes the payment with correct details, they return to the main sequence at Step 5 (confirmation). Step 5: If credentials are invalid, alternative sequence repeated.

Nonfunctional requirements	<p>Performance: Ensure fast response times during the payment process.</p> <p>Usability: Maintain an intuitive and user-friendly interface.</p> <p>Accessibility: Comply with accessibility standards to accommodate users with disabilities.</p> <p>Reliability: Provide reliable payment functionality.</p> <p>Scalability: Handle concurrent requests from multiple users without performance degradation.</p> <p>Security: Safeguard user payment and personal information.</p> <p>Error Handling: Effectively handle errors and edge cases.</p> <p>Availability: Ensure the booking system is available 24/7 to accommodate users from different time zones.</p>
Postconditions	<p>After successfully completing the booking process, users receive a confirmation of their flight reservation along with relevant booking details. The system generates electronic tickets or booking references, which users can use for check-in and boarding. Optionally, users may receive an email confirmation of their booking for their records.</p>

4. Loyalty Program:

User enrolls in loyalty program

UC Name	Loyalty Program UC 401
Summary	System allows user to redeem loyalty points in order to receive a discount.
Dependency	User Registration & Login (UC 101&102) Optionally (Loyalty Program UC 401)
Actors	Primary actor: User/Passenger Secondary actors: System
Preconditions	1. The passenger must have a valid account.

Description of the Main Sequence	<p>Step 1. Points are earned by client/passenger for eligible purchases or flights.</p> <p>Step 2: When the client/passenger wants to apply loyalty discount based on his/her points he clicks the “Apply loyalty discount button”</p> <p>Step 3. User redeems points and receives discounts.</p>
Description of the Alternative Sequence	<p>Passenger has no loyalty points:</p> <p>If passenger has no points, price remains the same.</p>
Non-functional Requirements	<p>Performance: Ensure fast response of the program</p> <p>Usability: Maintain an intuitive and user-friendly interface.</p> <p>Accessibility: Comply with accessibility standards to accommodate users with disabilities.</p> <p>Reliability: Provide reliable payment functionality.</p> <p>Scalability: Handle concurrent requests from multiple users without performance degradation.</p> <p>Security: Safeguard user personal information.</p> <p>Availability: Ensure the booking system is available 24/7 to accommodate users from different time zones.</p>
Postconditions	The passenger/client can earn/redeem points successfully.

5. Airline Admin Panel:

Adds new flights.

UC Name	Add Flights UC 501
Summary	This use case allows administrators to add new flight details, including route, schedule, and capacity.
Dependency	UC 001
Actors	<p>Primary actor: Administrator</p> <p>Secondary actors: System (flight database platform)</p>
Preconditions	Necessary permissions must be assigned to admin.
Description of the Main Sequence	<p>Step 1: Admin accesses the flight management interface by logging in using their credentials.</p> <p>Step 2: Administrator navigates to the "Add Flight" section and enters new flight details such as flight number, route, schedule, capacity, and pricing.</p> <p>Step 3: System validates the entered flight details, checking for missing fields or invalid formats, and ensures the flight number is unique.</p> <p>Step 4: System saves the new flight details in the database once validation is successful.</p>

	Step 5: Administrator receives a confirmation message, and the new flight is made available on user-facing platforms for booking.
Description of the Alternative Sequence	Step 1: Administrator provides incorrect or incomplete flight details, prompting the system to display an error message. Step 2: Administrator corrects the information and resubmits the details. Step 3: If the system fails during the saving process (e.g., due to a network issue), the system notifies the administrator to retry or seek support.
Non-functional Requirements	Performance-System must handle flight additions within 5 seconds. Usability & Accesability-Interface must be intuitive and accesible. Security-Data must be stored securely to prevent unauthorized access.
Postconditions	New flight is added and made available for users to view and book.

Admin authorizes Flight Cancelation

UC Name	Cancel Flights UC 502
Summary	This use case allows administrators to cancel flights affected by natural disasters and notify passengers.
Dependency	(UC 501)
Actors	Primary actor: Administrator Secondary actors: System
Preconditions	1. Natural disaster impacts flight safety or viability. 2. Administrator has proper access rights.
Description of the Main Sequence	Step 1: Admin search for flight to cancel. Step 2: Administrator press "Cancel Flight" button. Step 3: System prompts for confirmation with flight details. Step 4: Upon confirmation, the system cancels the flight(s) and updates user-facing platforms. Step 5: System provides compensation via loyalty points.
Description of the Alternative Sequence	Cancellation failed Step 1: In case of failure, system prompts Admin to retry.
Non-functional Requirements	Performance-System must handle flight deletions quickly (within 5 seconds).

Airline Ticket Booking Software Specification

	Usability & Accesability-Interface must be intuitive and accesible. Security-Data must be stored and then deleted to securely to prevent unauthorized access.
Postconditions	Flight is successfully canceled, passengers are notified, and the system reflects updated flight schedules.

Airline Admin modifies existing bookings.

UC Name	Modify Flight UC 503
Summary	This use case allows administrators to modify existing flight details and compensate affected clients with active bookings.
Dependency	(UC 101,102, 201, 301, 501)
Actors	Primary actor: Administrator Secondary actors: System
Preconditions	The flight must already exist in the system. The administrator must have modification rights.
Description of the Main Sequence	Step 1: Administrator accesses the flight management interface. Step 2: Administrator selects the flight to modify by searching the flight database. Step 3: Administrator edits the required flight details (e.g., schedule, route etc.). Step 4: System validates the changes and checks for conflicts Step 5: System applies the modifications and updates related systems
Description of the Alternative Sequence	Modification failed Step 1: In case of failure, system prompts Admin to retry.
Non-functional Requirements	Performance-System must handle flight deletions quickly (within 5 seconds). Usability & Accesability-Interface must be intuitive and accesible. Security-Data must be stored and then deleted to securely to prevent unauthorized access.
Postconditions	Flight details are successfully updated, affected clients are compensated, and the system reflects the modifications.

The system shall grant admins access to comprehensive analytics, aiding informed decision-making.

UC Name	Analytics Access UC 504
Summary	This use case allows administrators to access detailed analytics within the system, facilitating informed decision-making processes.
Dependency	
Actors	Primary Actor: Airline Administrator Secondary Actor: System
Preconditions	The system must be operational and accessible. The administrator must be authenticated and logged into the system. Detailed analytics must be generated and available within the system. The admin must have the necessary permissions to access analytics.
Description of the Main Sequence	Step 1: The admin navigates to the analytics section within the system. Step 2: The system presents analytics report.
Description of the Alternative Sequence	Step 1: In case of empty report, system shows a message to the user for empty report.
Non functional requirements	Performance: Ensure prompt response and scalability of the performance management system. Security: Restrict access to authorized personnel, encrypt data, and maintain audit trails for accountability. Reliability: Maintain high availability and perform regular backups for data recovery. Usability: Provide an intuitive interface and allow customization options to enhance user experience.

	Compliance: Adhere to data privacy regulations and retention policies to ensure legal compliance.
Postconditions	The admin has successfully accessed and utilized the desired analytics, enabling them to make informed decisions based on the presented information.

The system shall allow administrators to access financial reports.

UC Name	Financial Reports UC 505
Summary	This use case involves administrators accessing financial reports
Dependency	
Actors	Primary Actor: Administrator Secondary Actor: System
Preconditions	The system must be operational and accessible. The administrator must be authenticated and logged into the system.
Description of the Main Sequence	Step 1: Administrators are directed to the financial reports interface. Step 1: The admin navigates to the financial section within the system. Step 2: The system presents financial report.

Description of the Alternative Sequence	Step 1: If the system encounters an error and is unable to retrieve the required data, it displays an error message and administrators notify of the issue.
Non functional requirements	Security: Access to financial reports should be restricted to authorized shareholders to maintain data confidentiality. Usability: The system interface for accessing financial reports should be intuitive and easy to navigate for shareholders.
Postconditions	Administrators have successfully accessed financial reports

6. Tech Support:

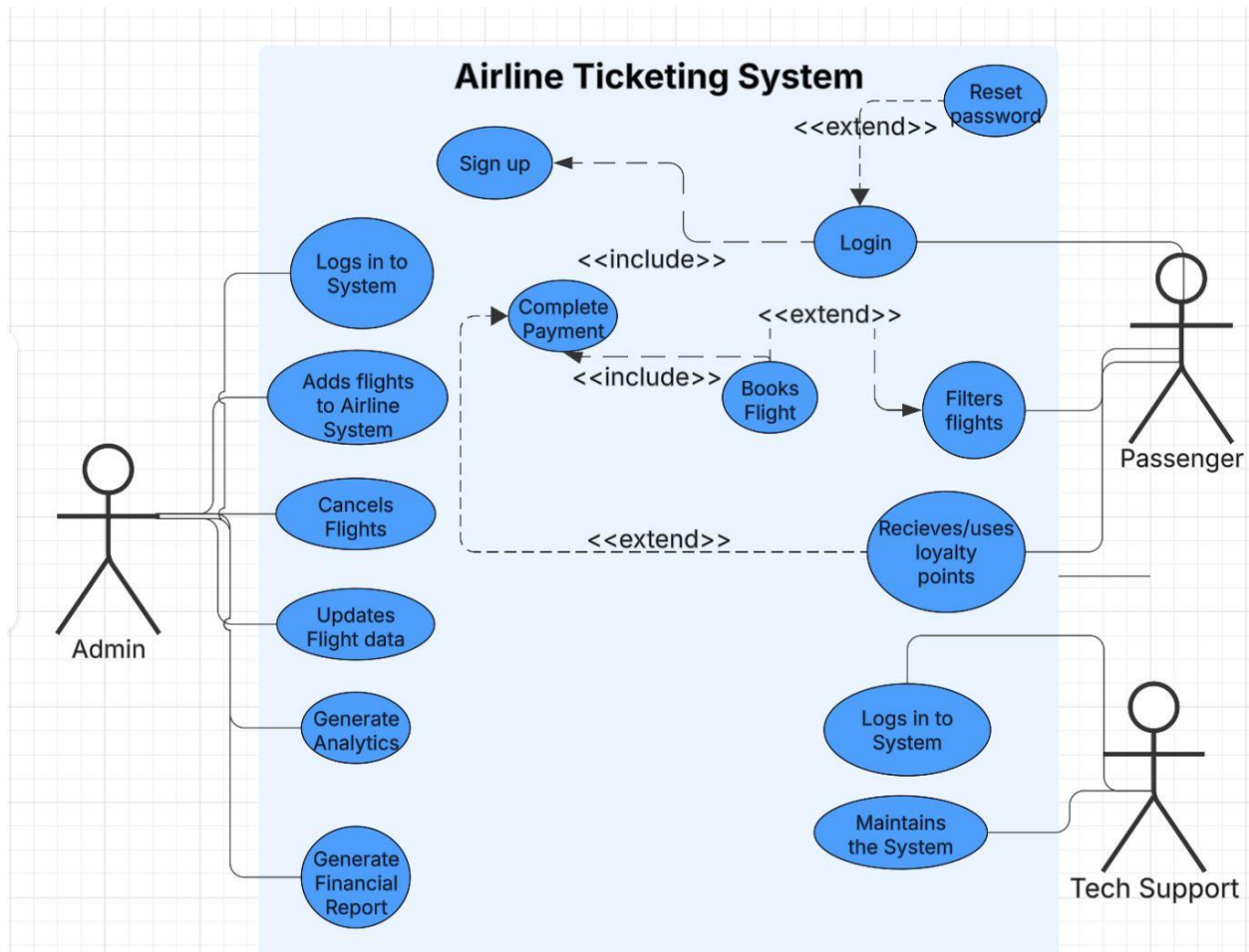
Tech Support Ensuring Software Functionality

UC Name	Software Functionality (UC 601)
Summary	This use case enables the tech support team to identify, troubleshoot, and resolve software issues to ensure smooth operation.
Dependency	
Actors	Primary actor: Tech Support Secondary actors: System
Preconditions	1. Software must be deployed and operational. 2. Tech support team must have appropriate access credentials and tools.
Description of the Main Sequence	Step 1: Tech support identifies a potential problem. Step 2: Tech support analyzes and diagnose the issue via Health Checks Button. Step 3: If system is not performing well, tech support refreshes the system via button Refresh. Step 4: If needed, tech support generated the pdf with Health Checks and Refreshes listed.
Description of the Alternative Sequence	Step 1: If the system is performing well from the analysis of Health Check Button, tech support just refreshes the page.
Non-functional Requirements	Issues must be identified and acknowledged within 5 minutes of detection or reporting. Resolutions must be implemented with

	minimal downtime for end users. All actions must be logged for auditing and future reference.
--	---

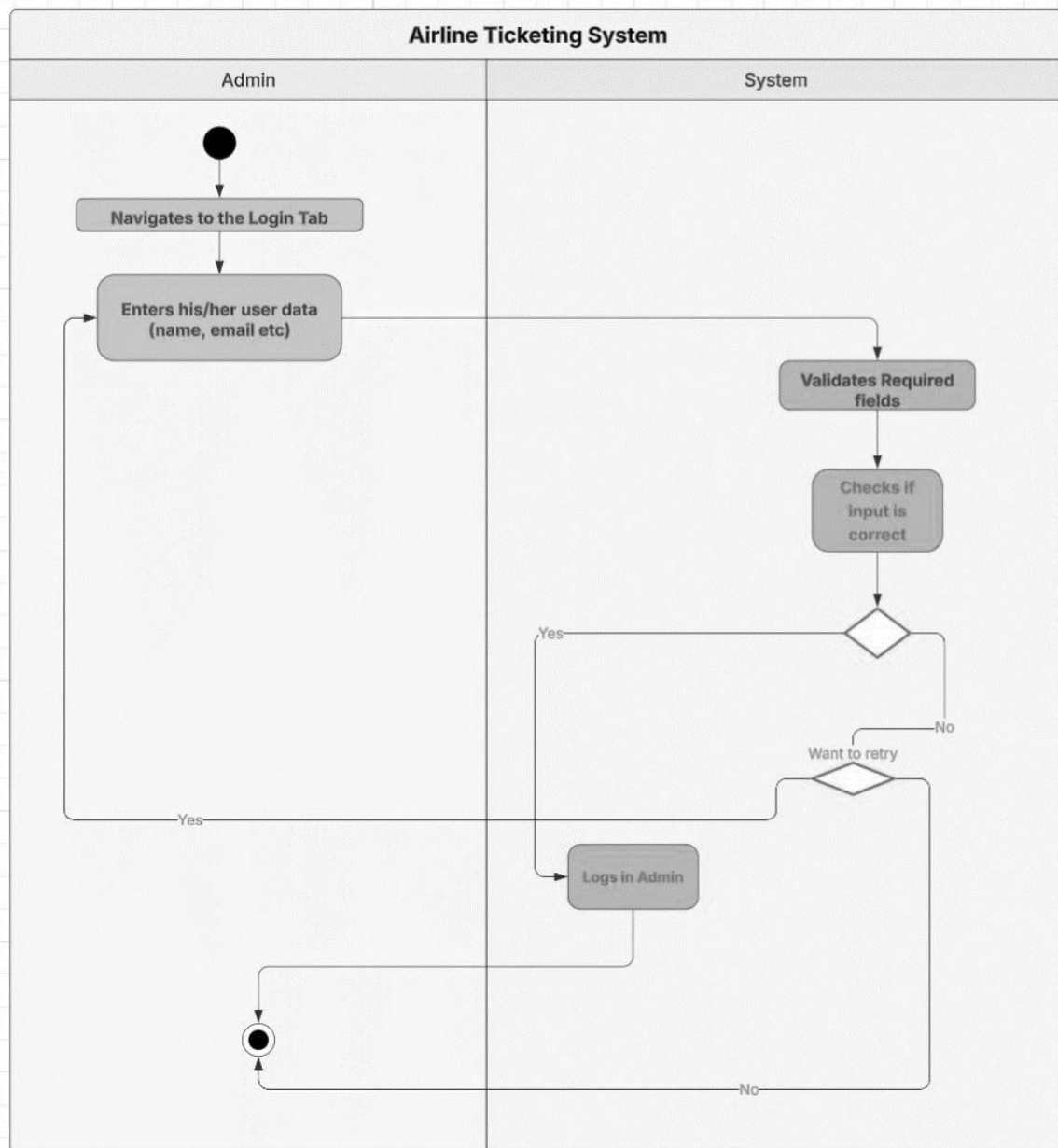
5. DIAGRAMS

5.1 Use Case Diagram

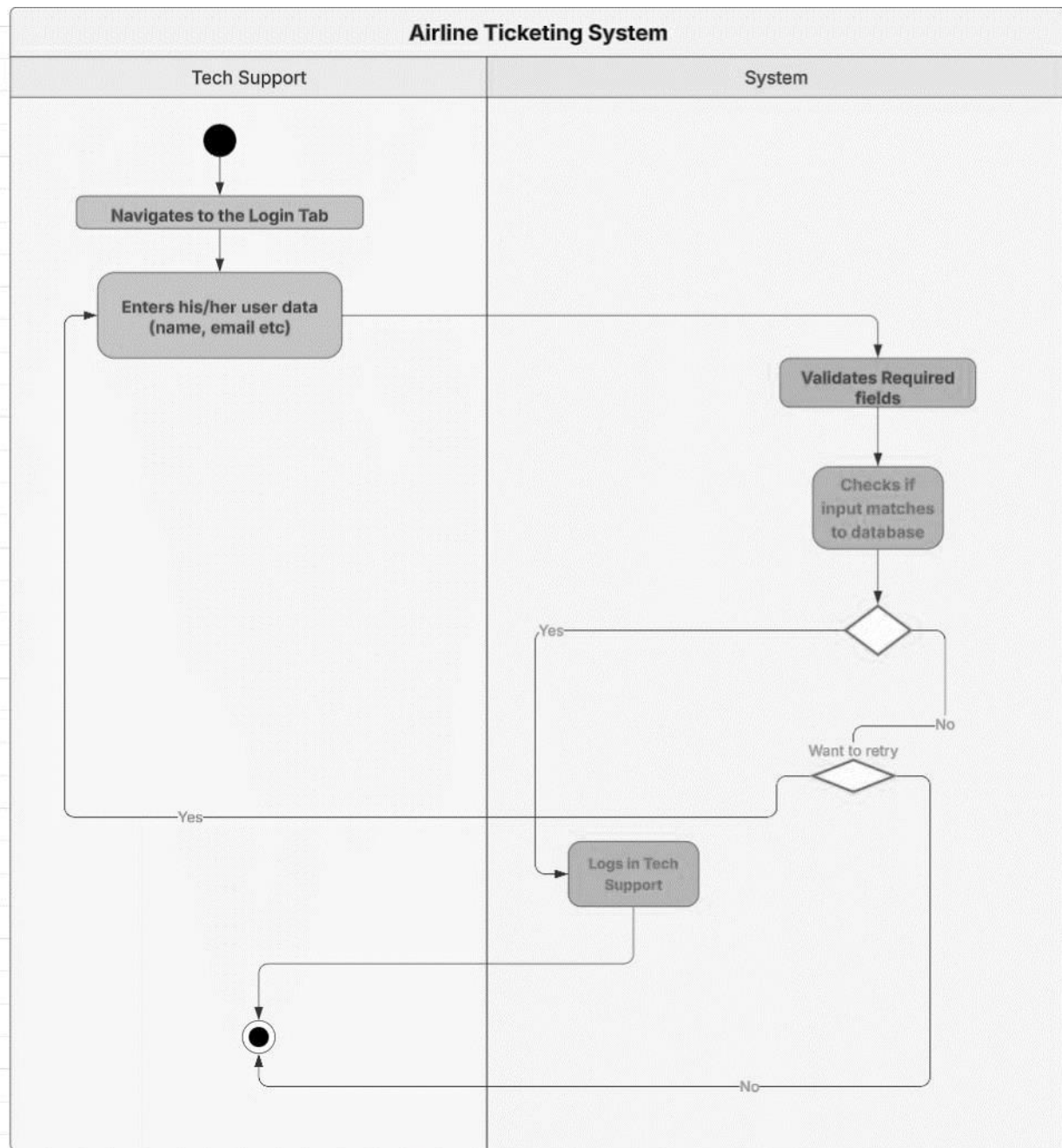


5.2 Activity Diagram

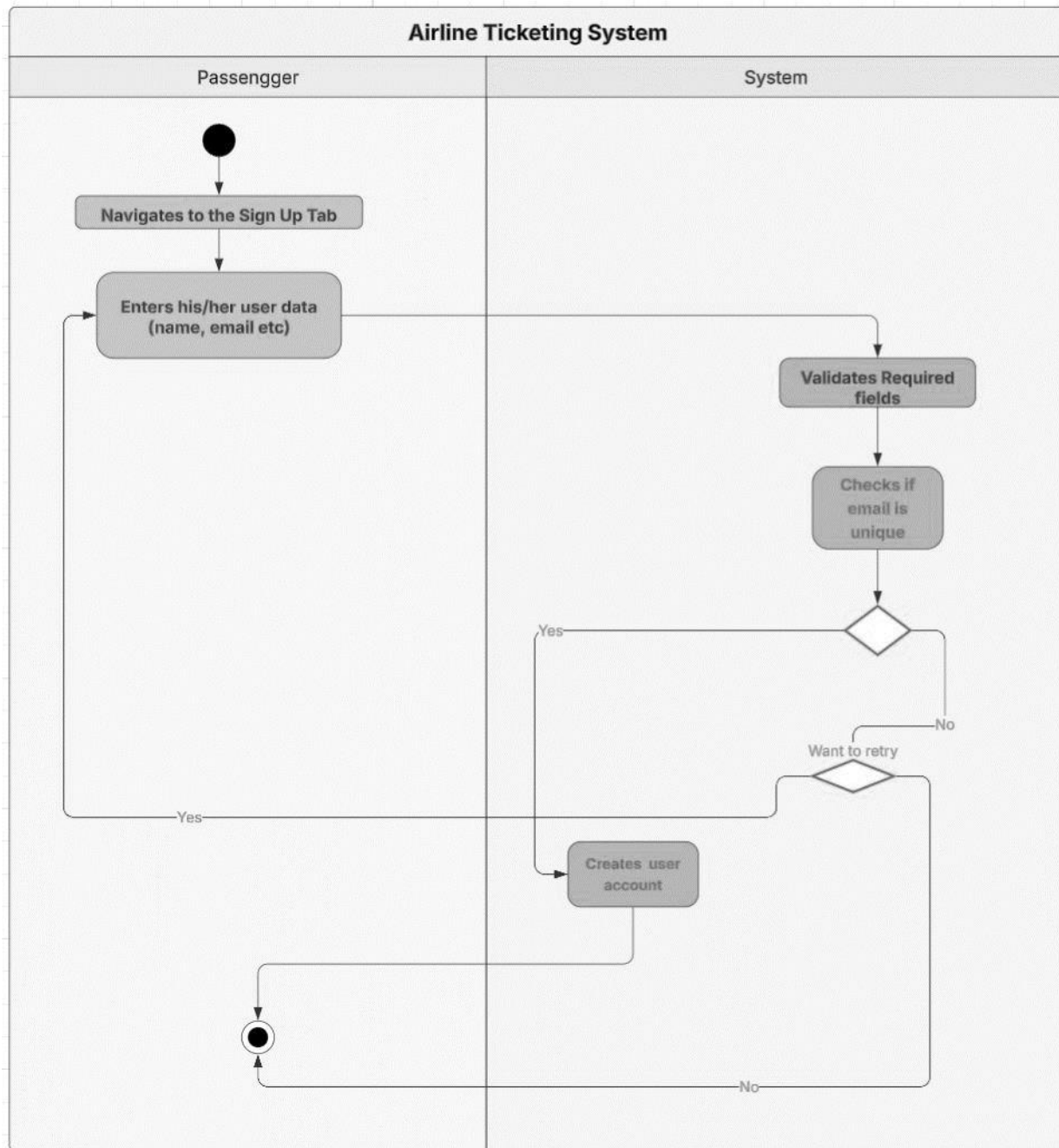
UC – 001: Admin Log In



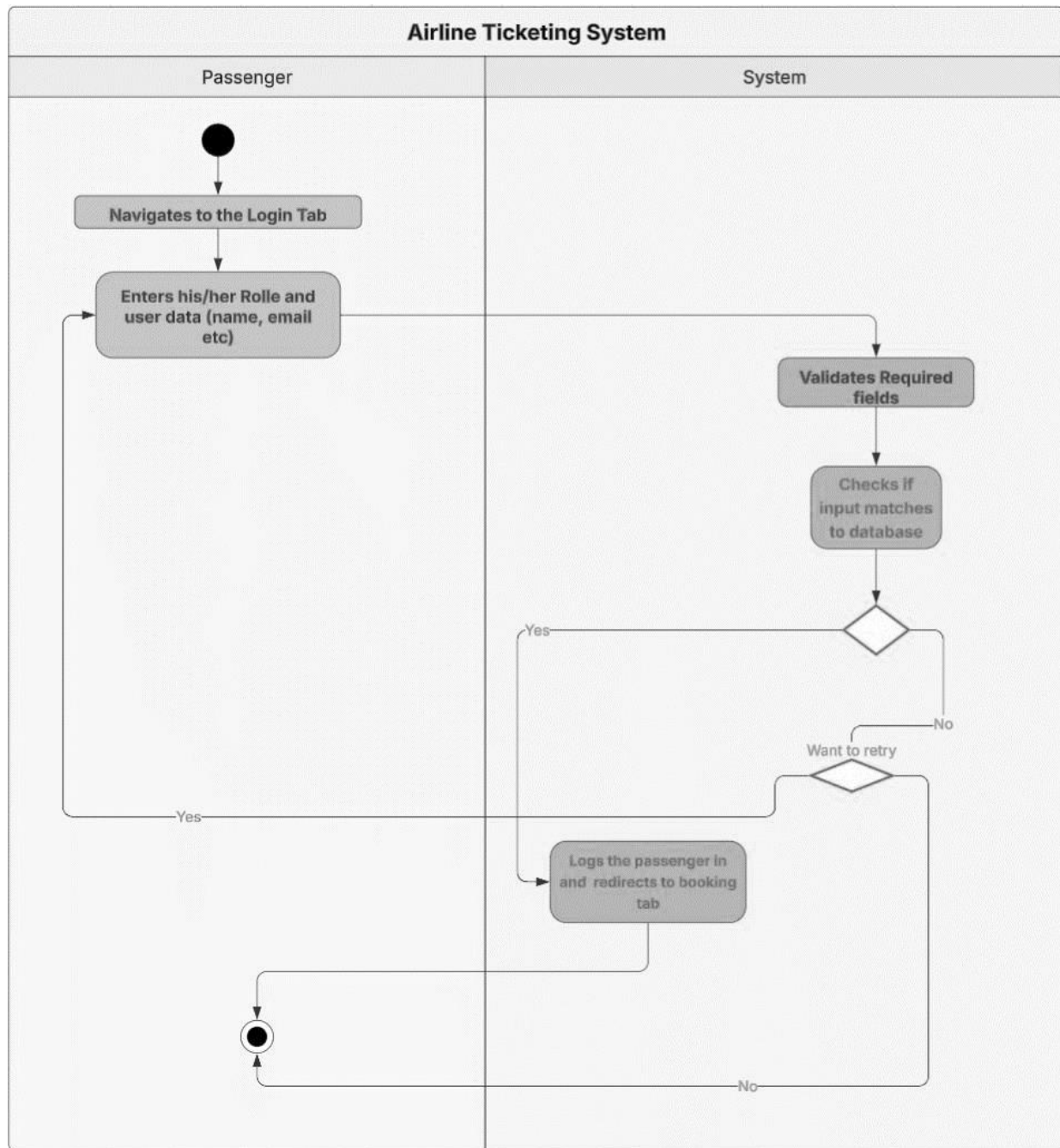
UC – 002: User – Tech Support Log In



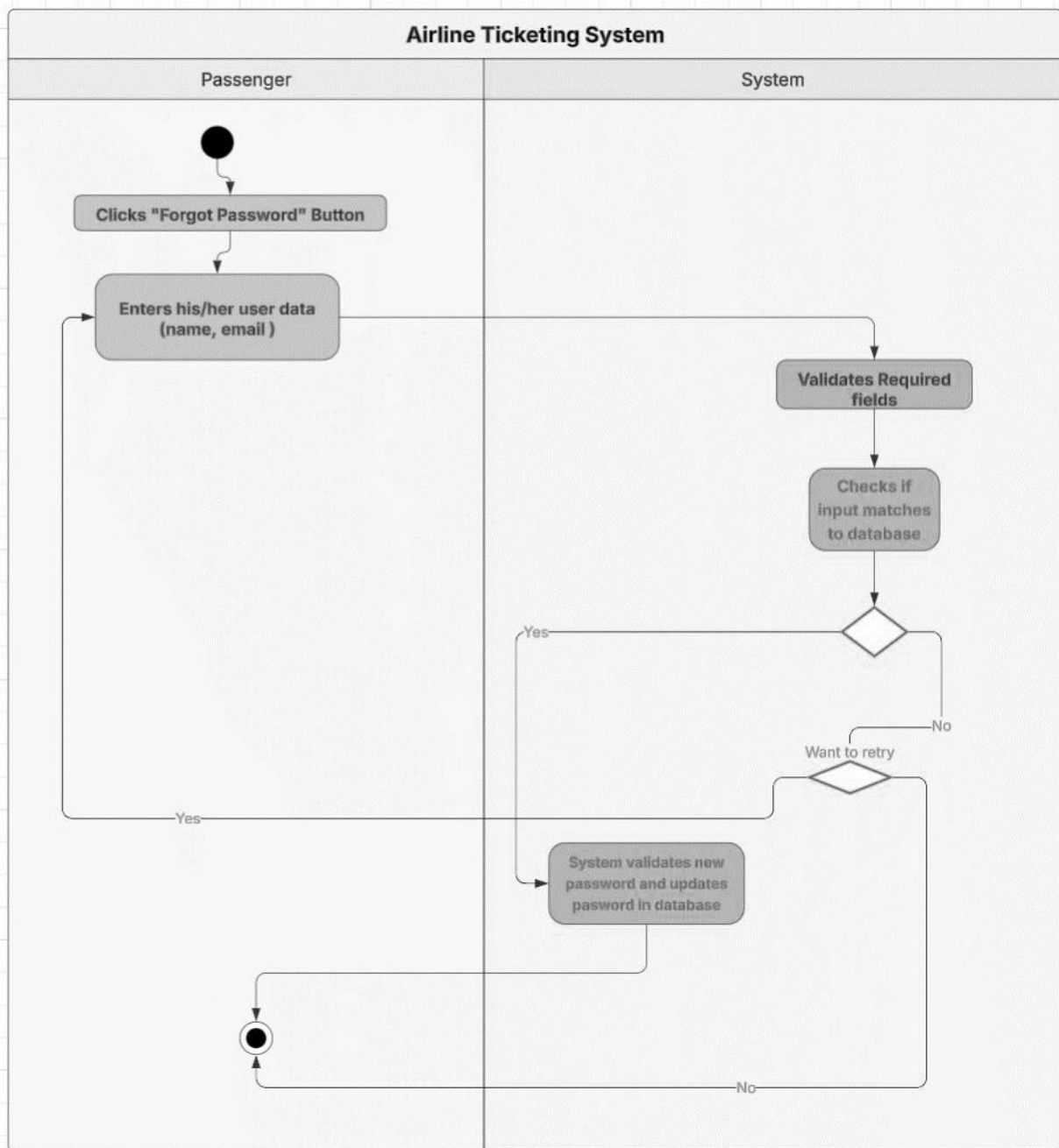
UC – 101: User – Passenger Registration



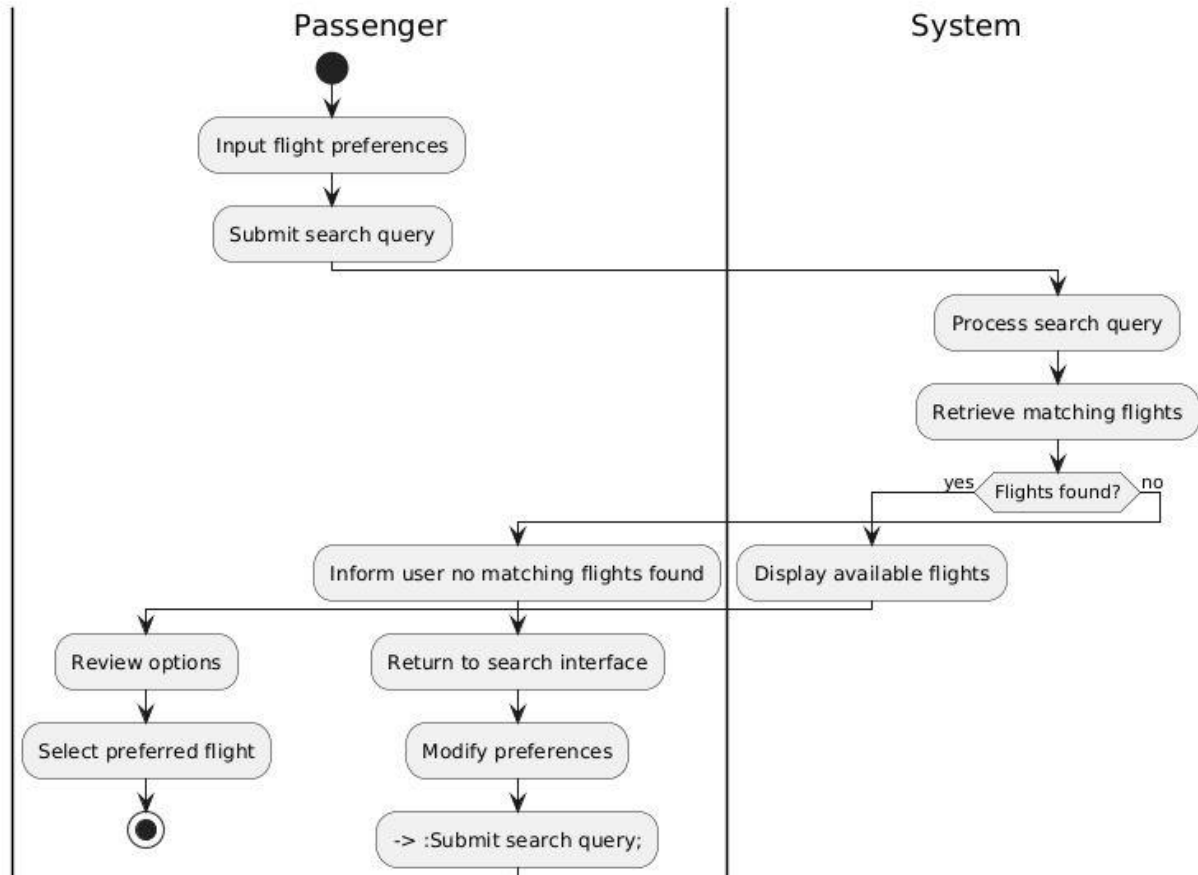
UC – 102: User Account Log – in



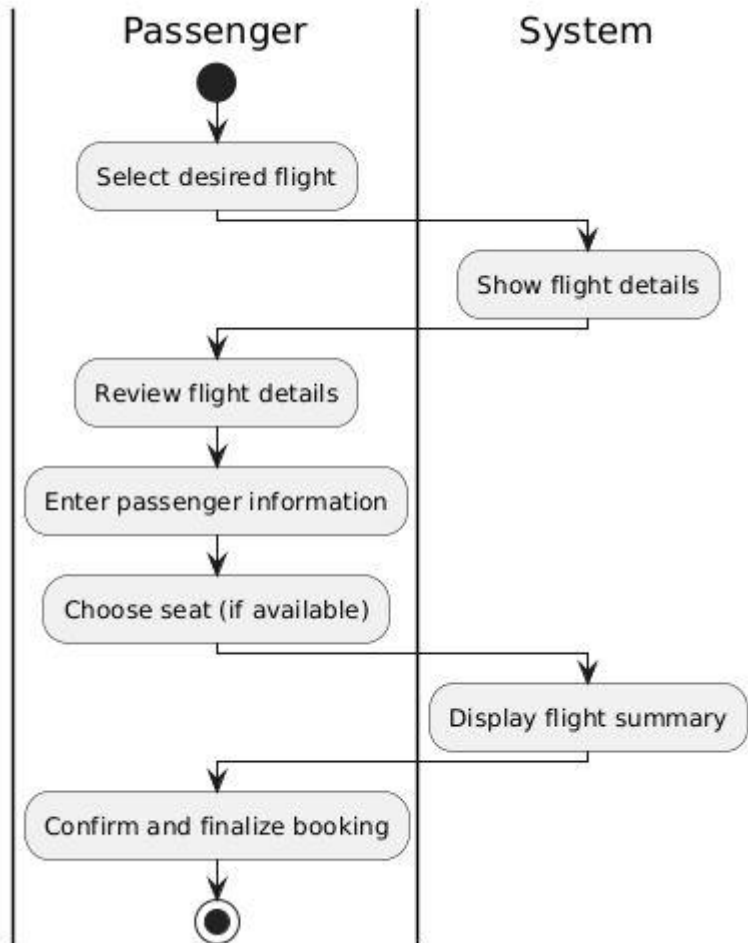
UC – 103: User Password Reset



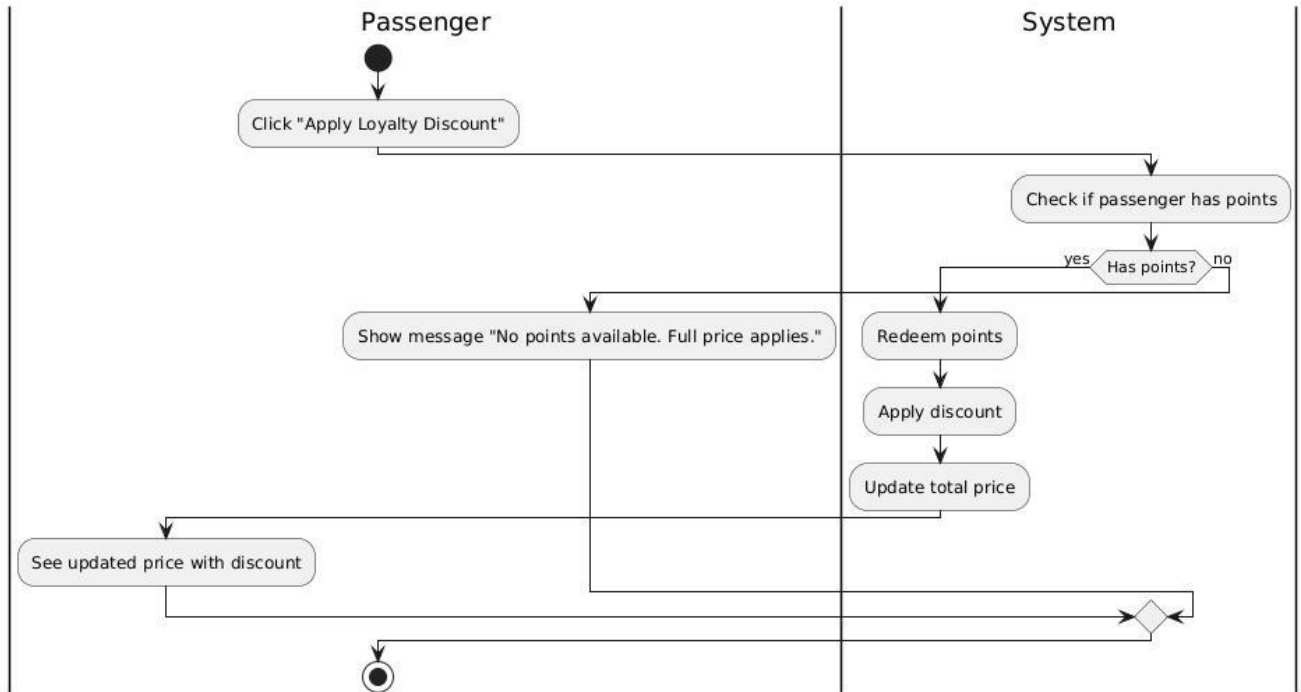
UC – 201: Filter Flights



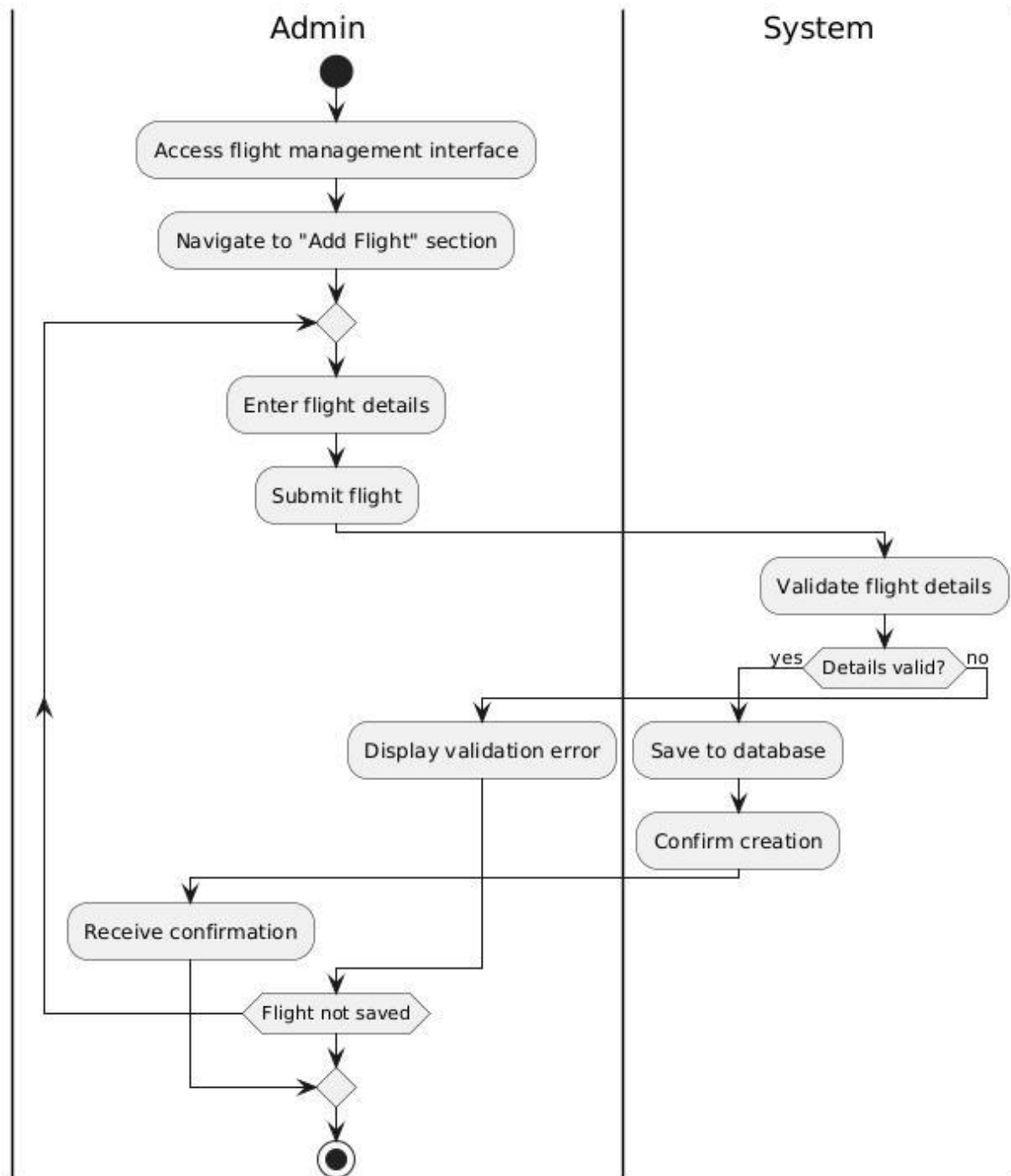
UC – 301: Book Flight



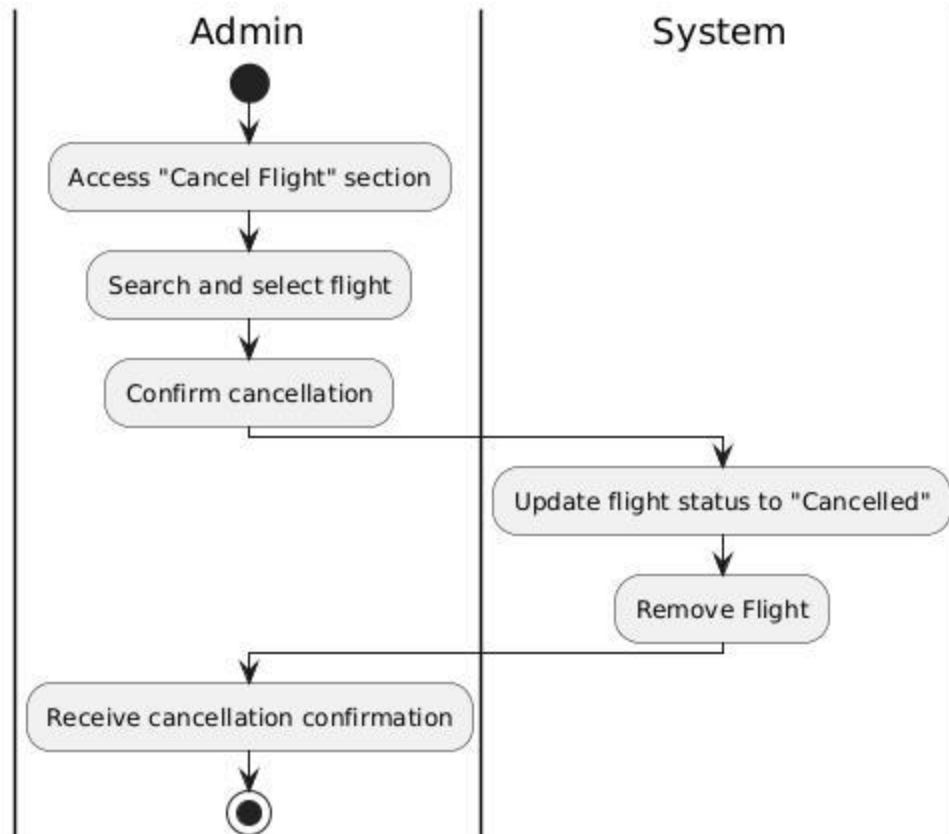
UC – 401: Loyalty Program



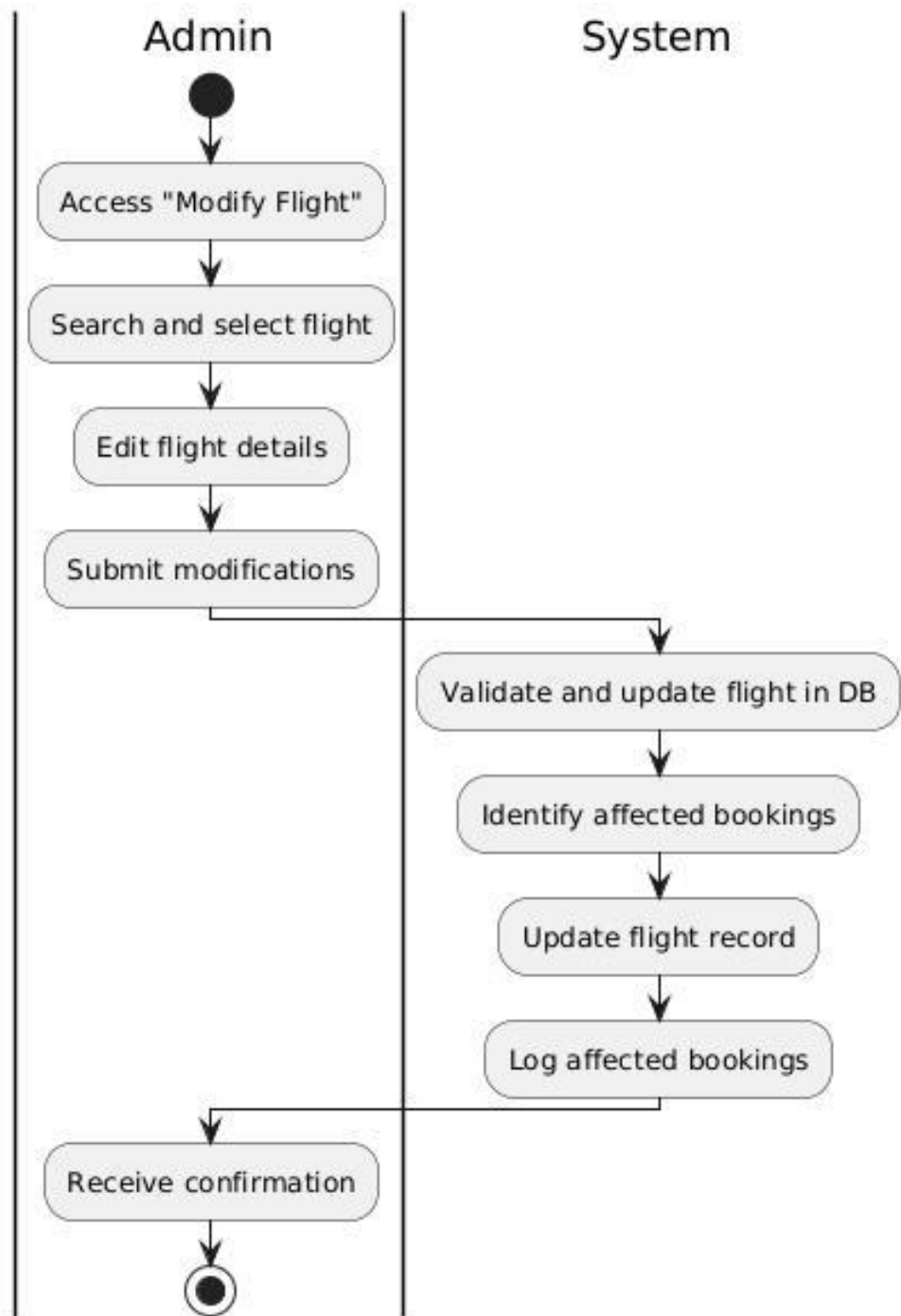
UC – 501: Add Flight



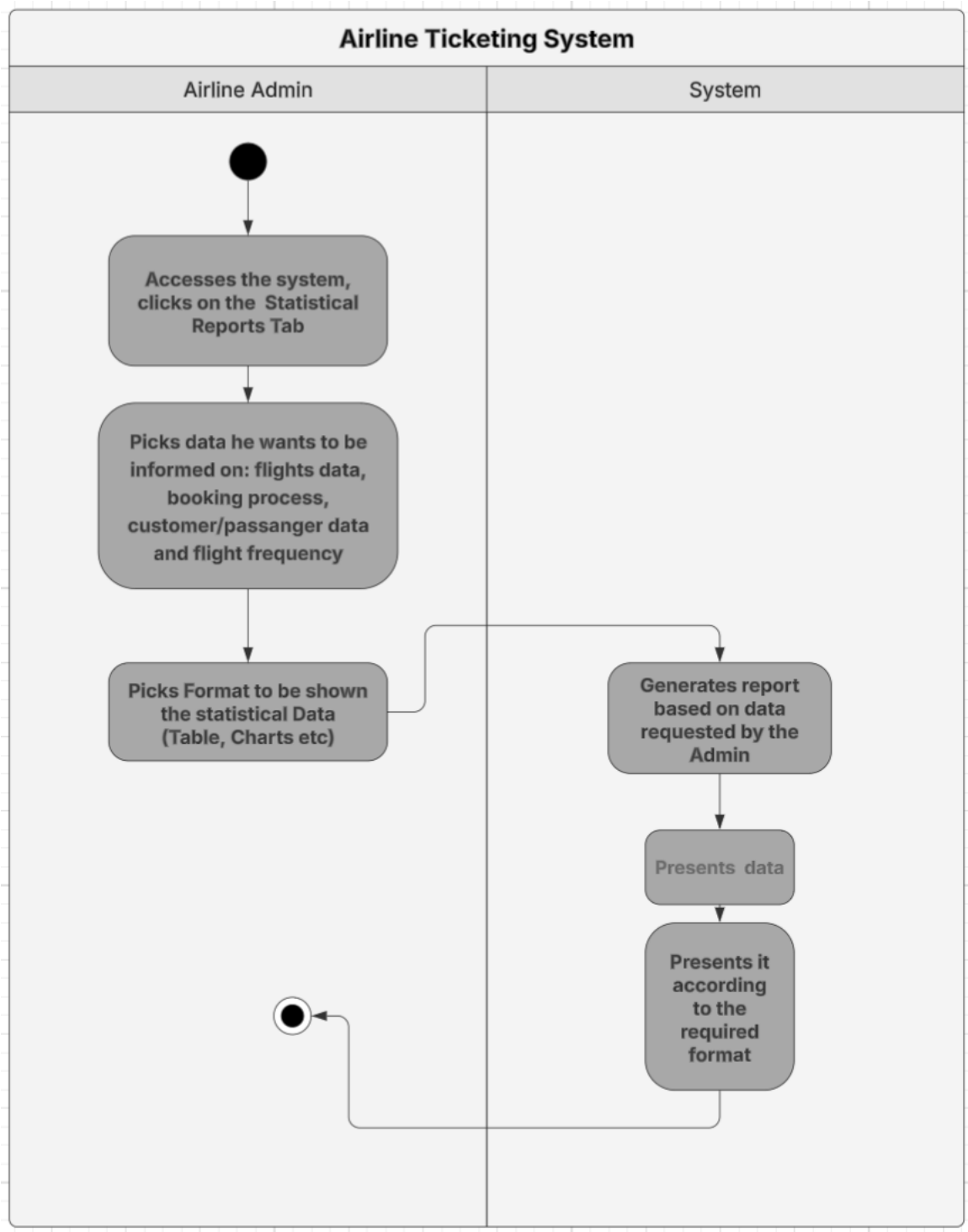
UC – 502: Cancel Flight



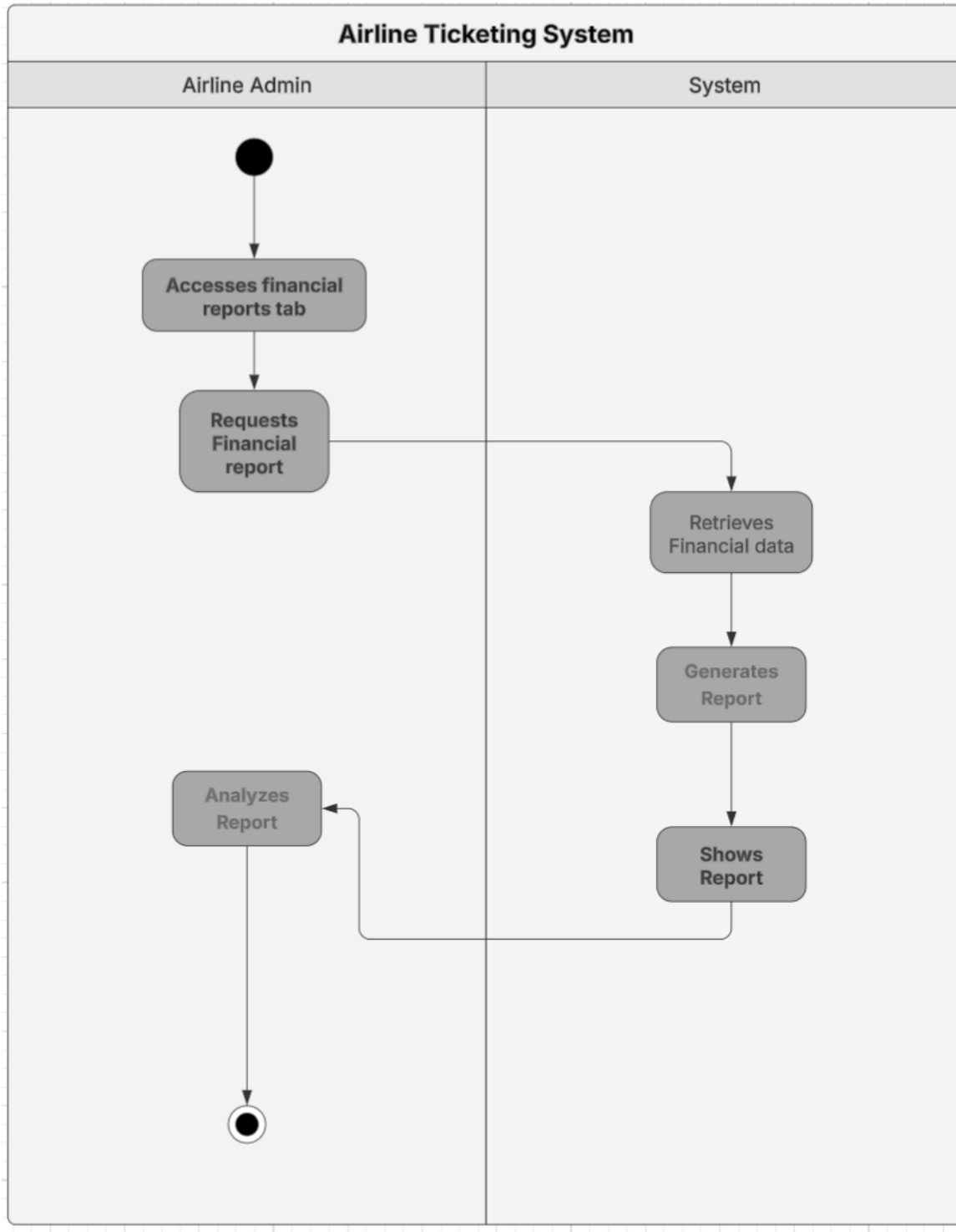
UC – 503: Modify Flight



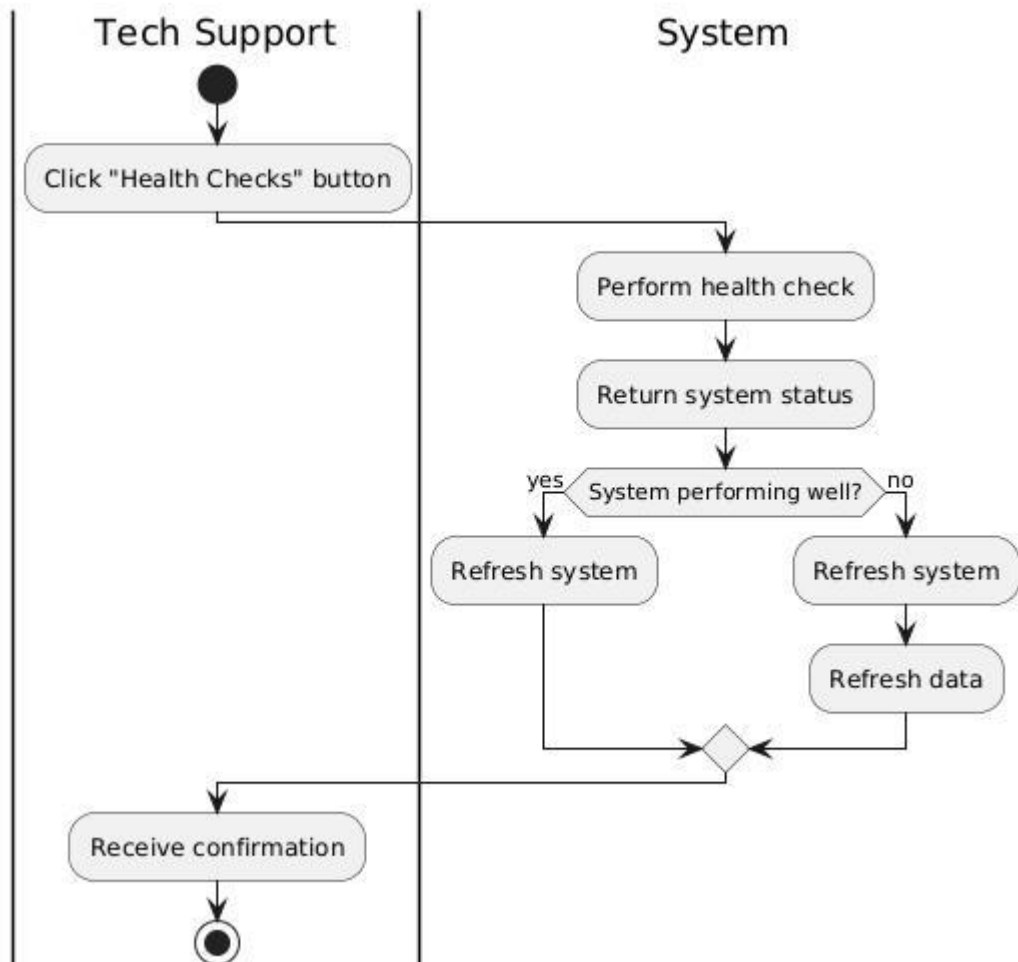
UC – 504: Analytics Analysis



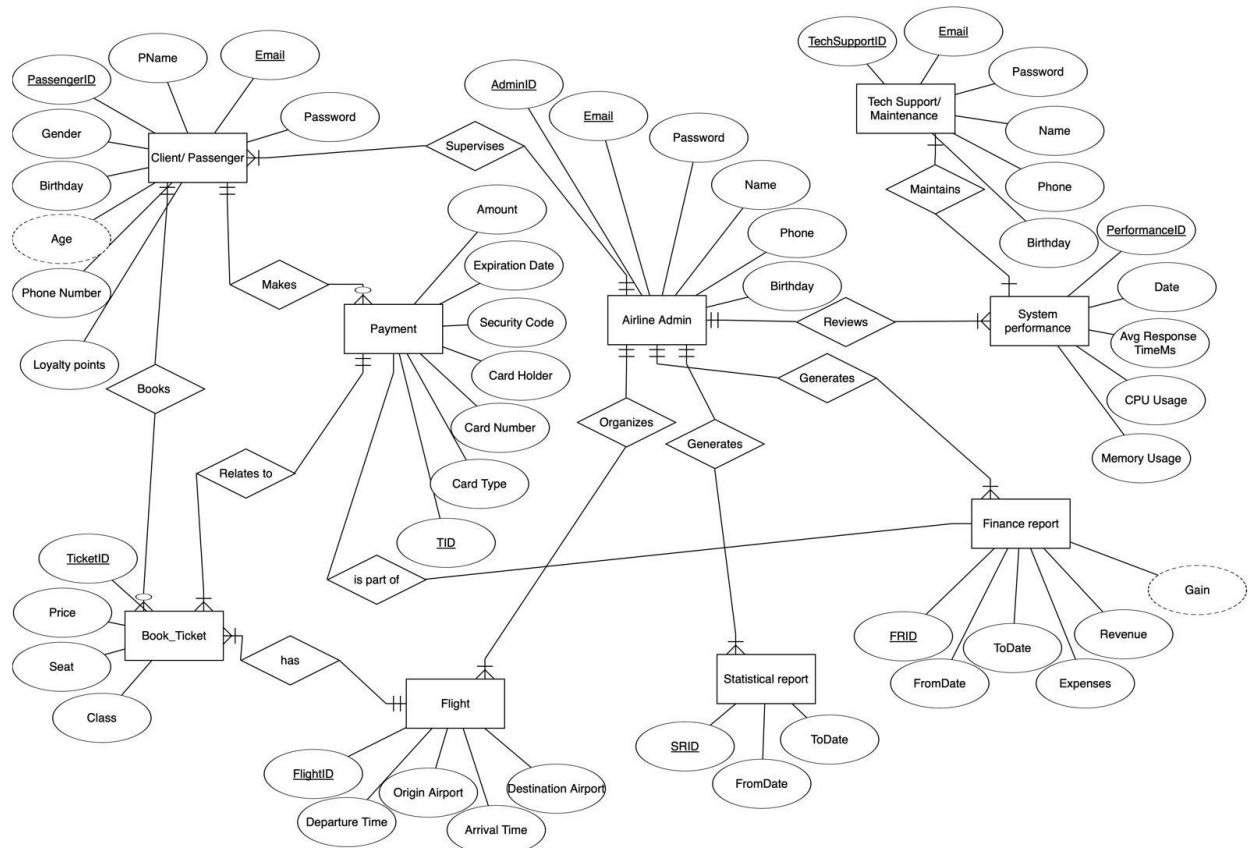
UC – 506: Financial Report



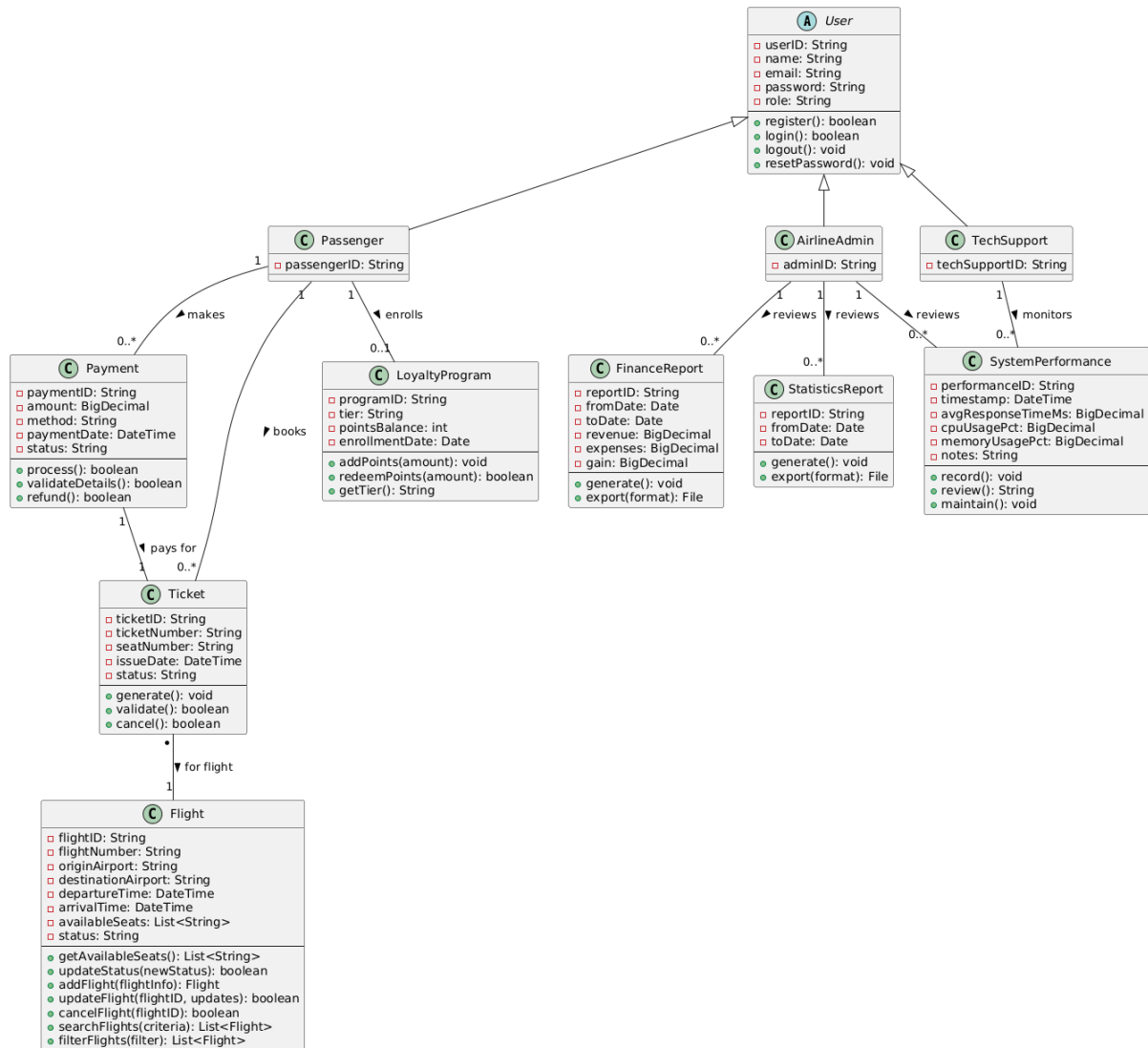
UC – 601: Software Functionality



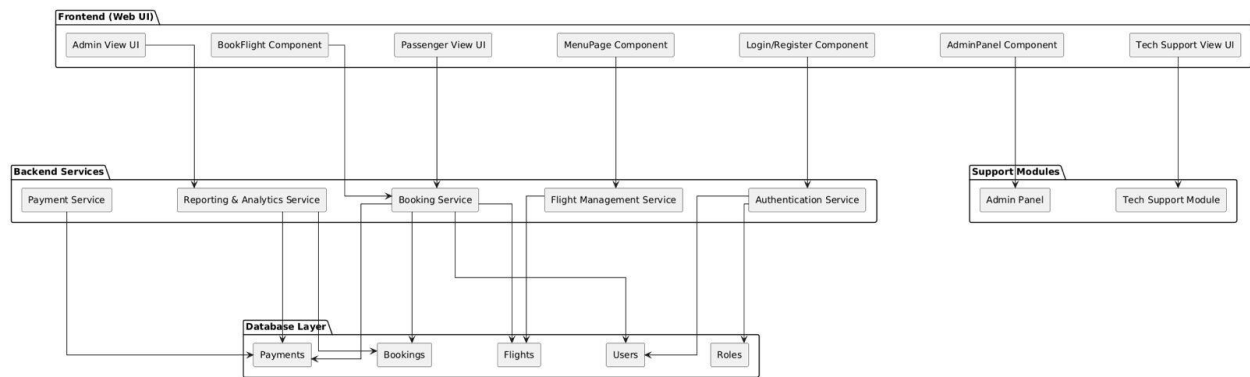
5.3 ERD Diagram



5.4 Class Diagram

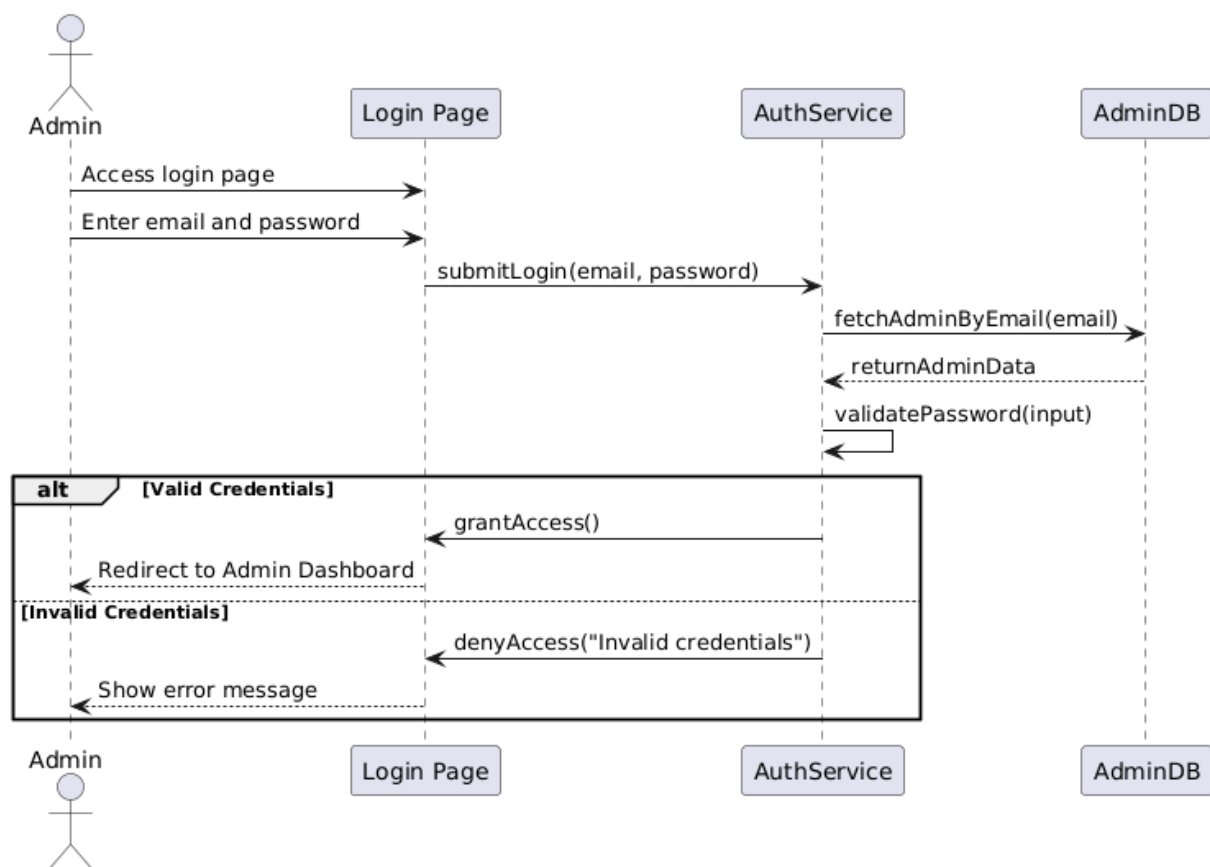


5.5 Component Diagram

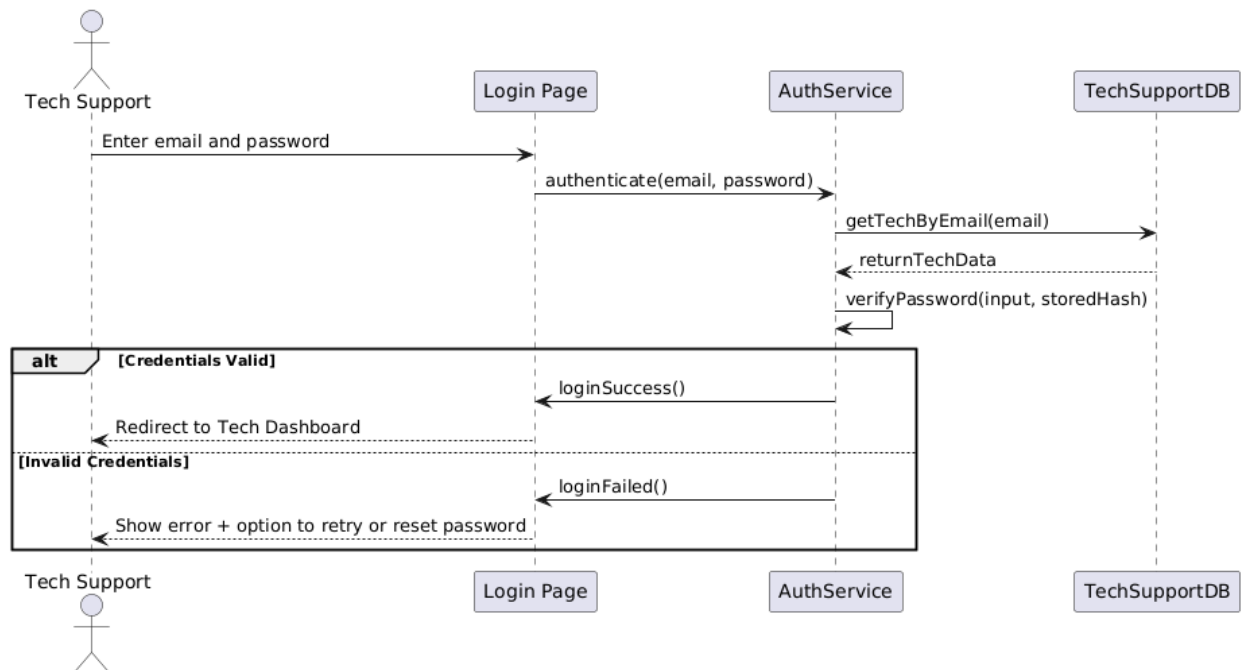


5.6 Sequence Diagrams

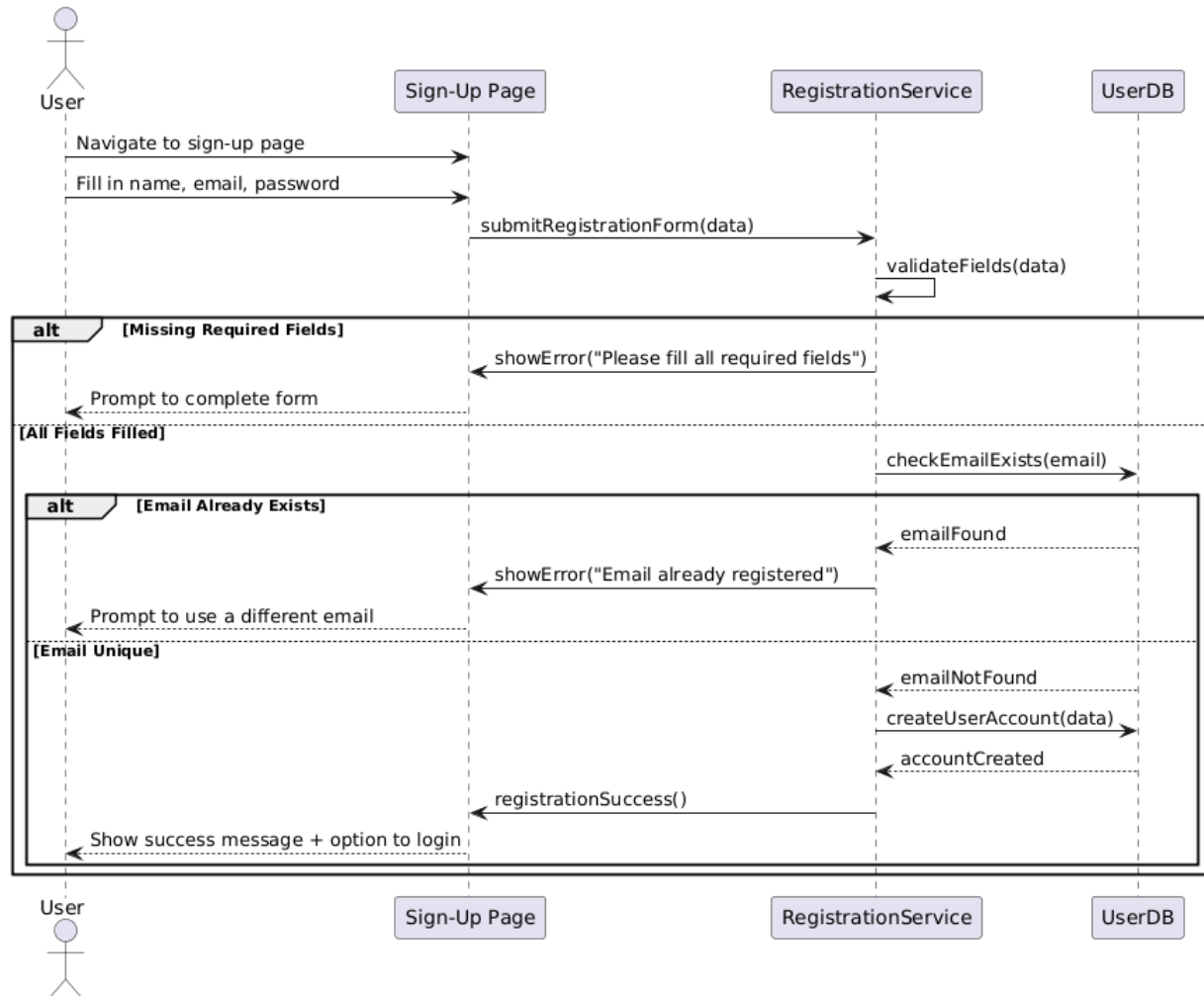
UC-001: Admin Log In

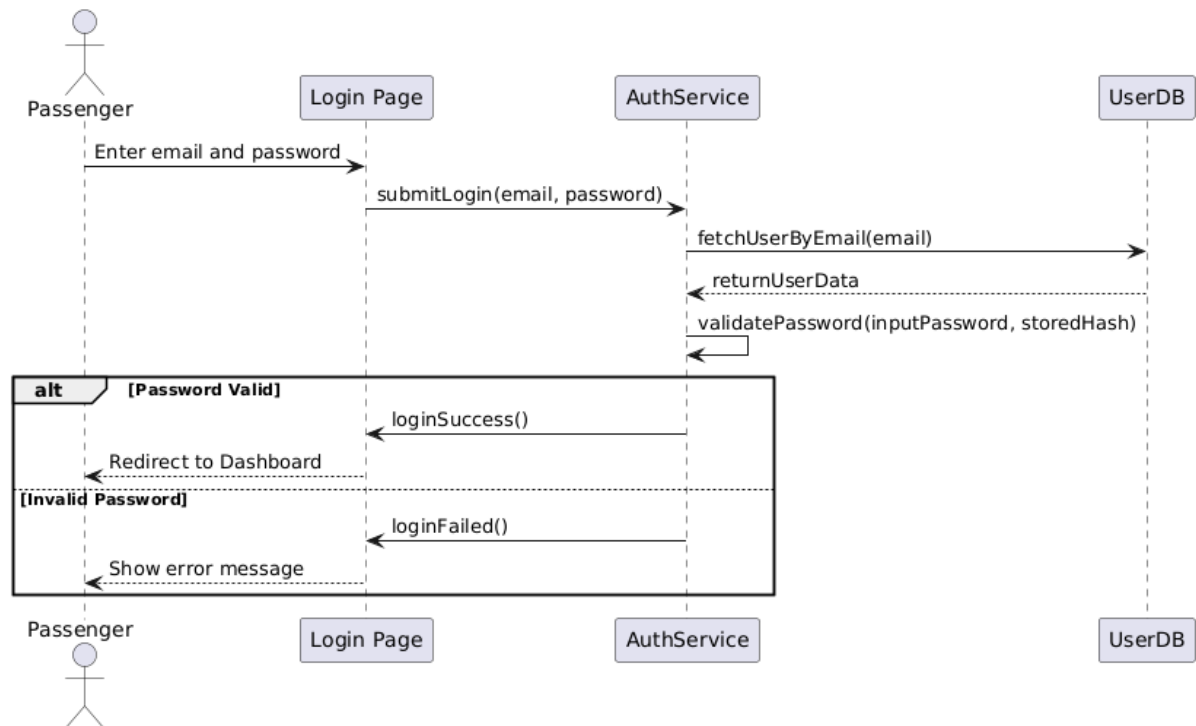


UC-002: Tech Support Log In



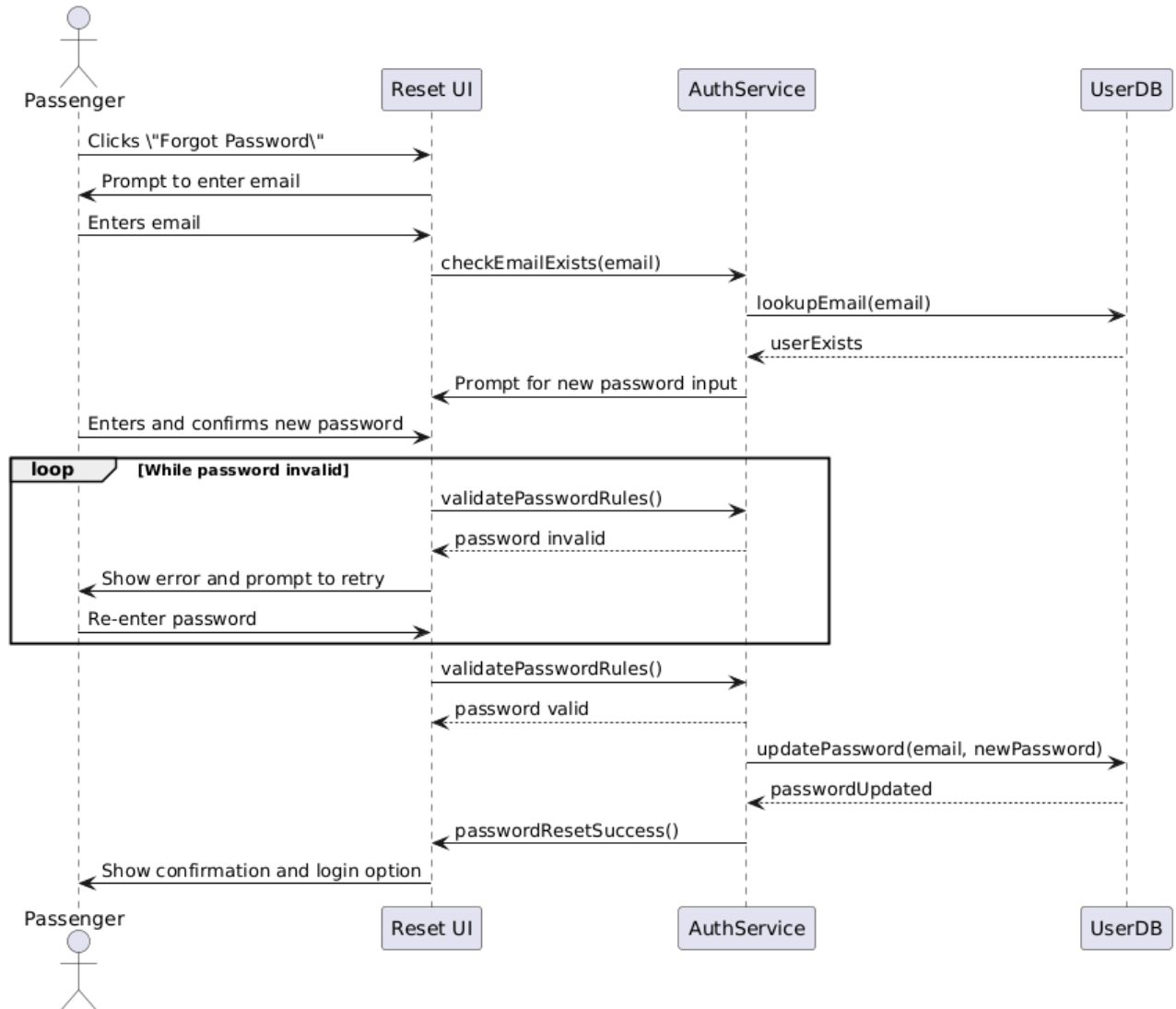
UC-101: Passenger Sign Up



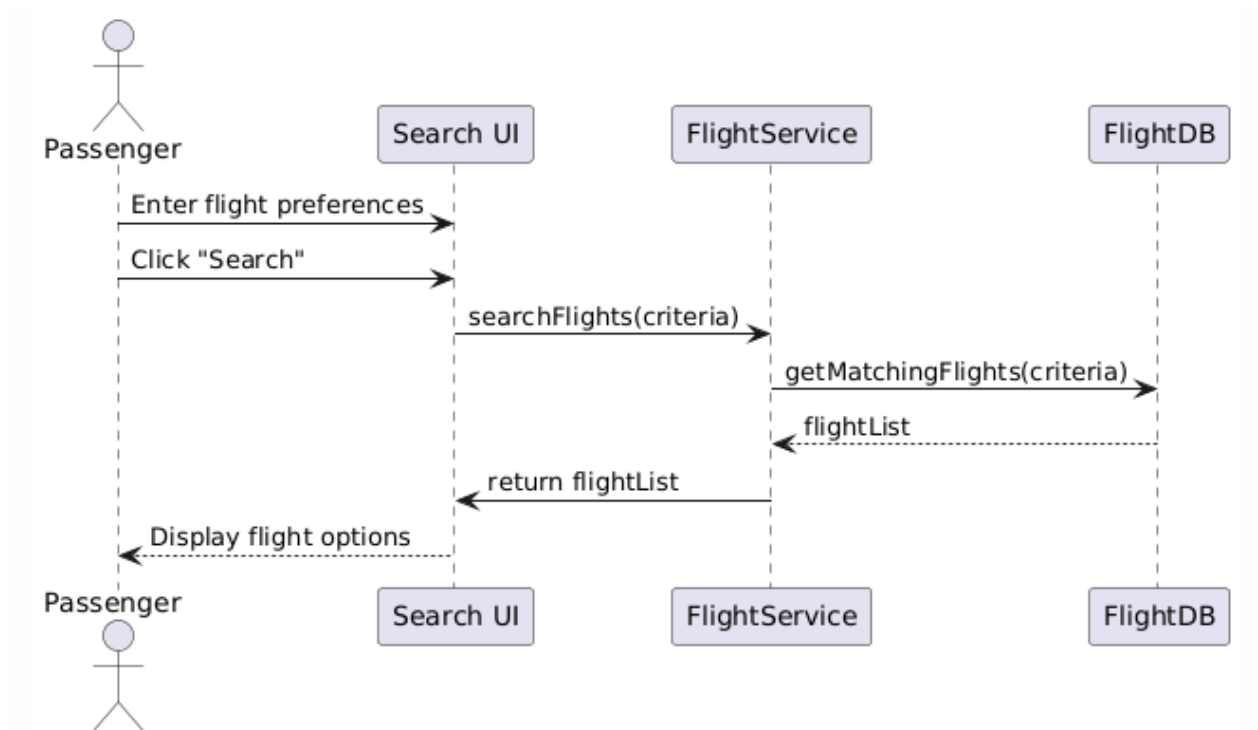


UC-102: Passenger Log In

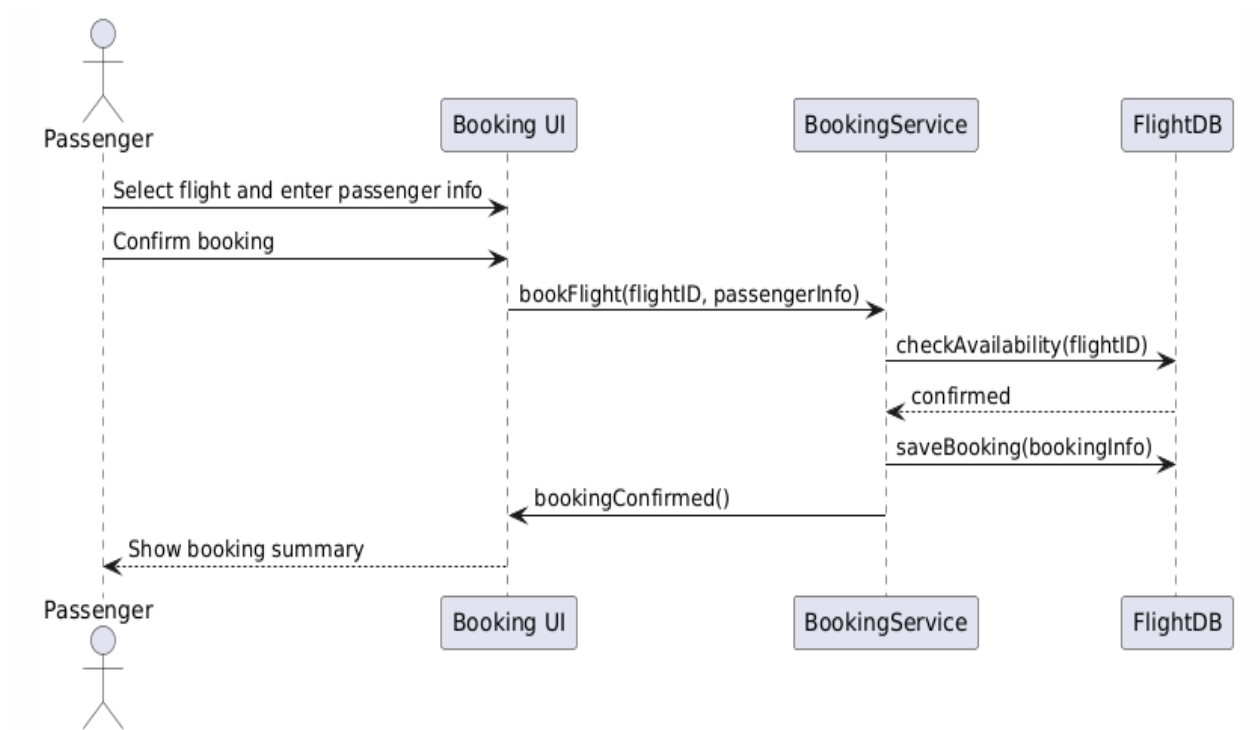
C-103: Password Reset – Sequence Diagram



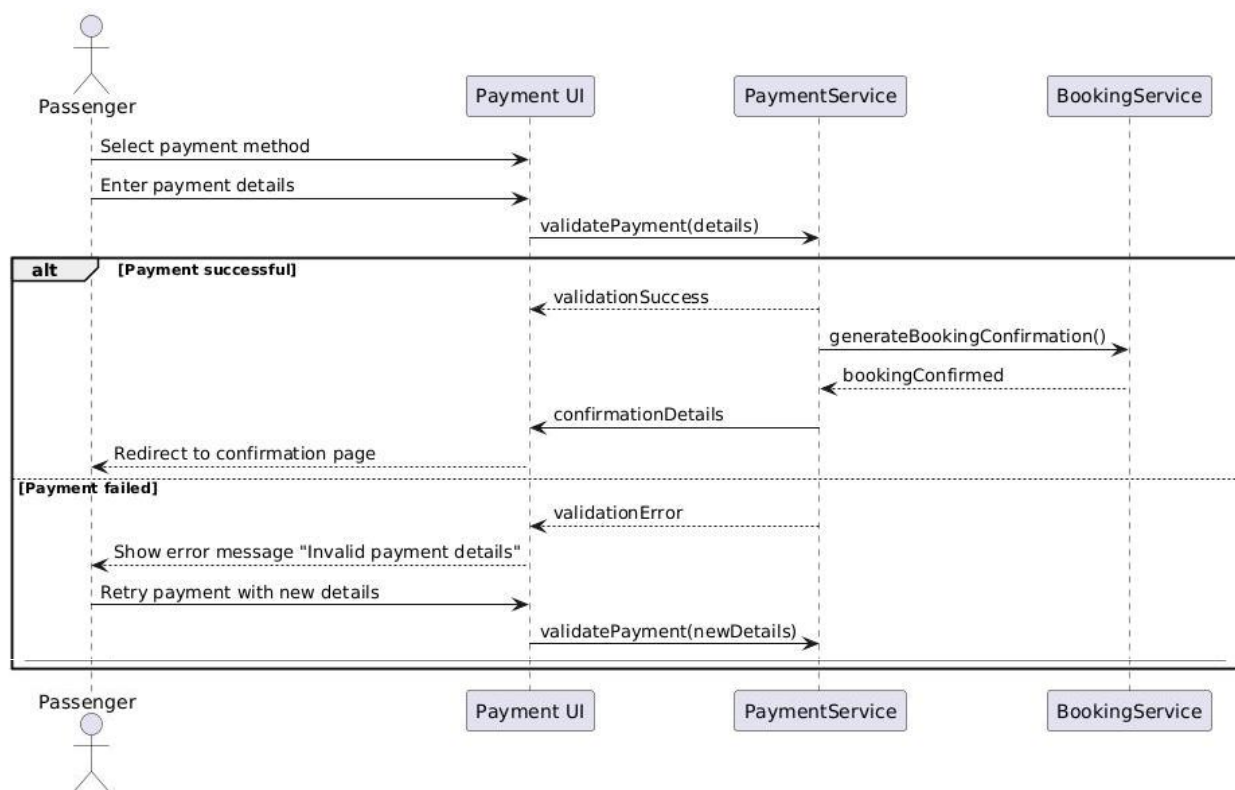
UC-201: Filter Flights – Sequence Diagram



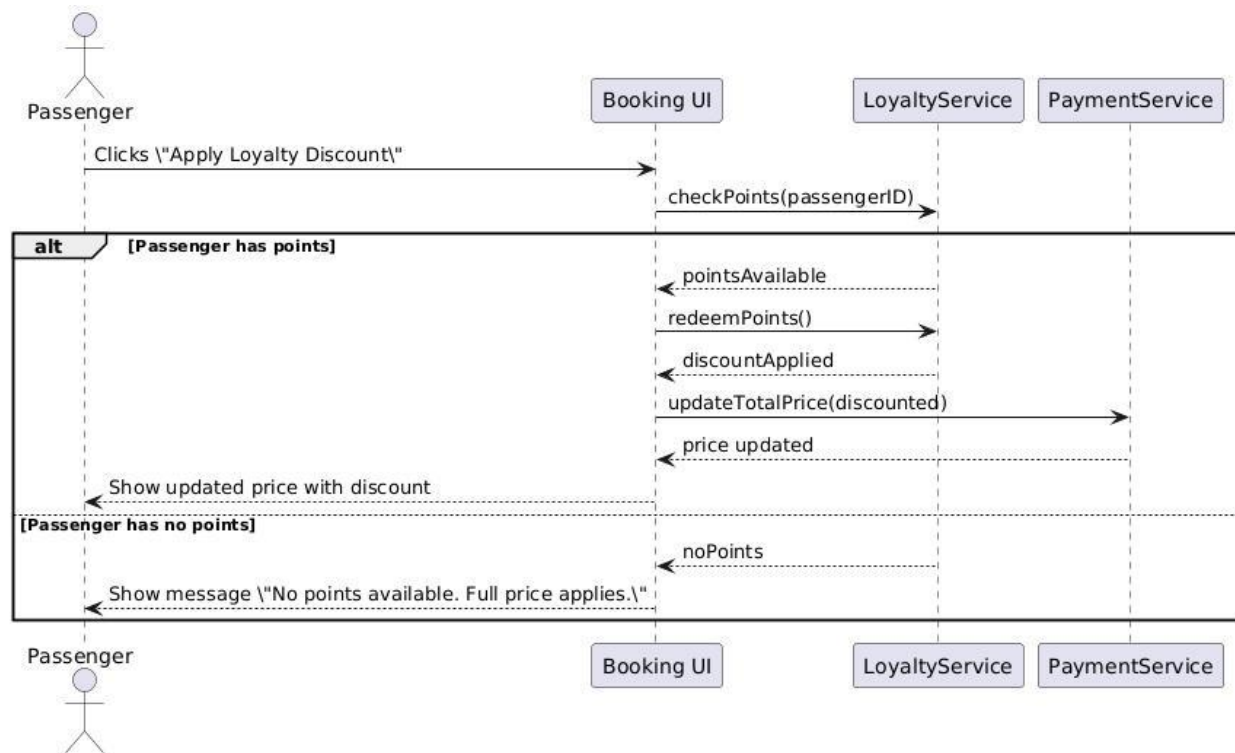
UC-301: Book Flight – Sequence Diagram



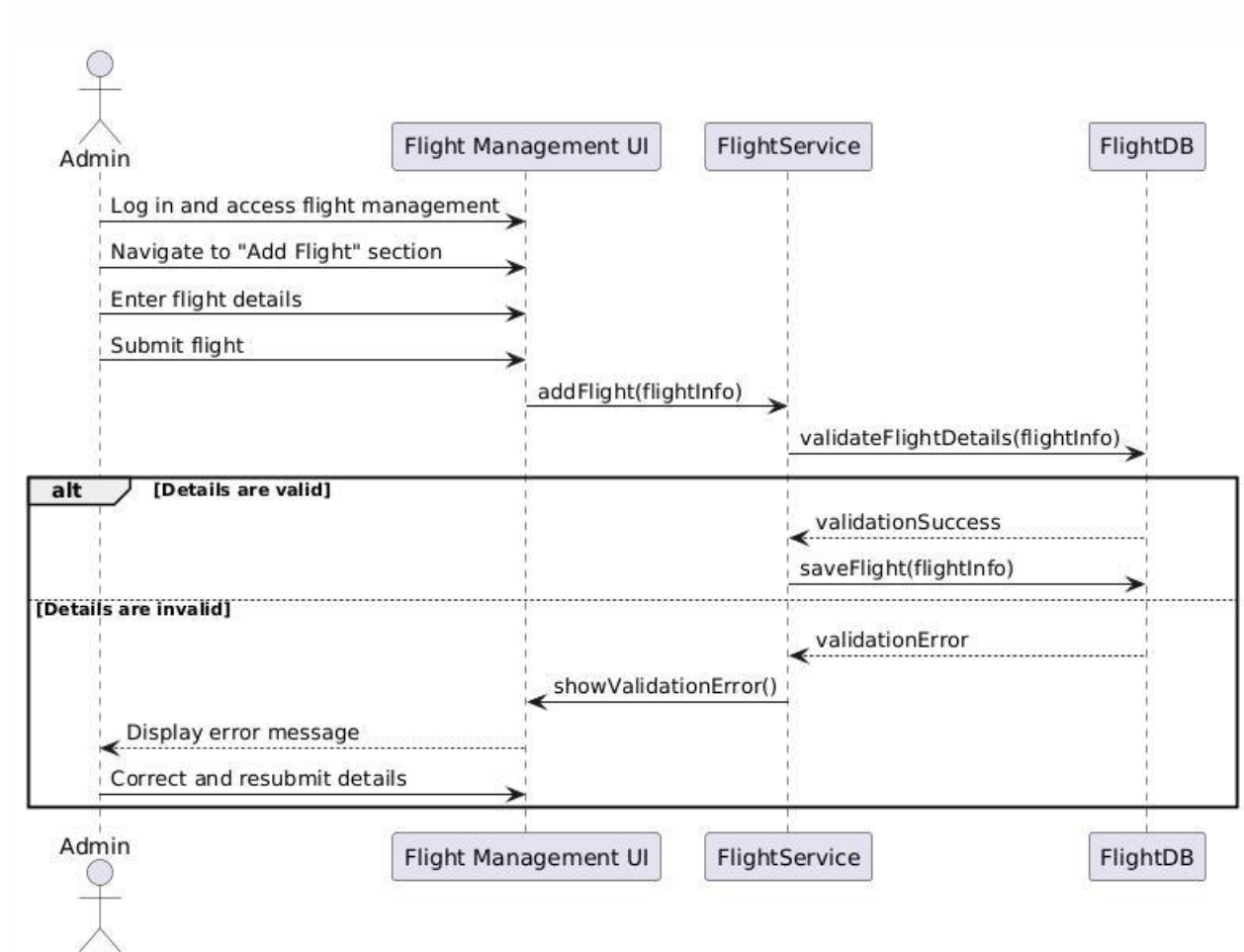
UC-302: Payment Processing – Sequence Diagram



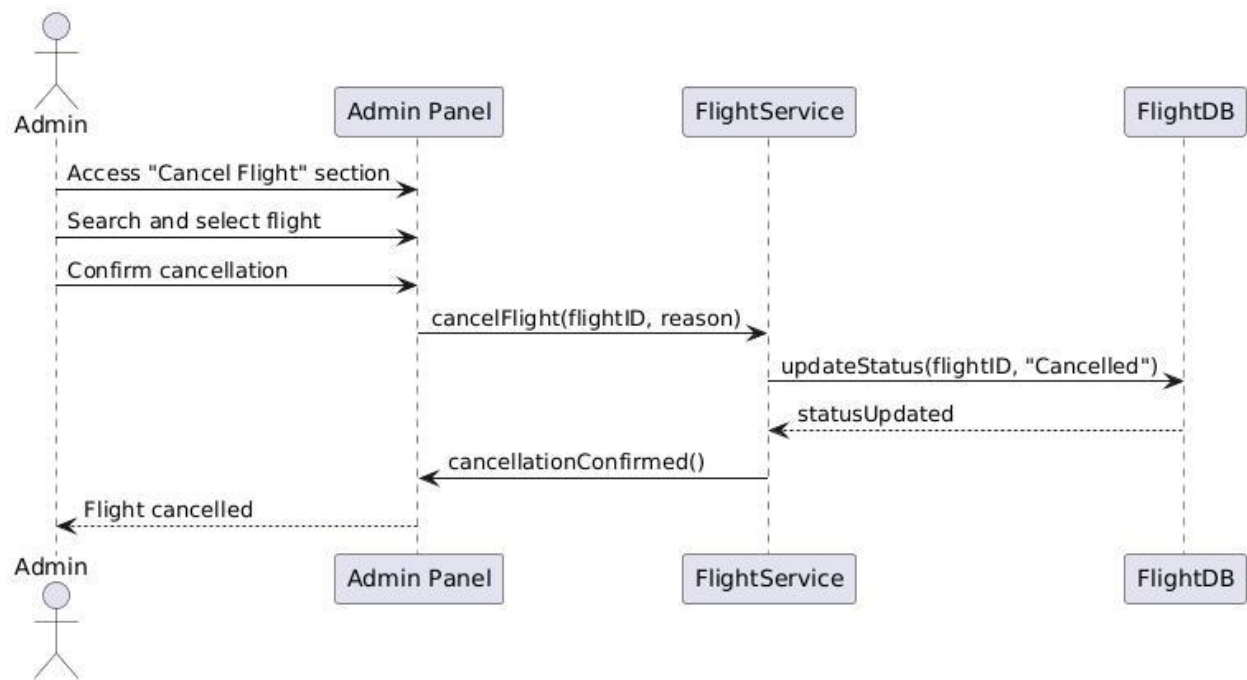
UC-401: Loyalty Program Enrollment – Sequence Diagram



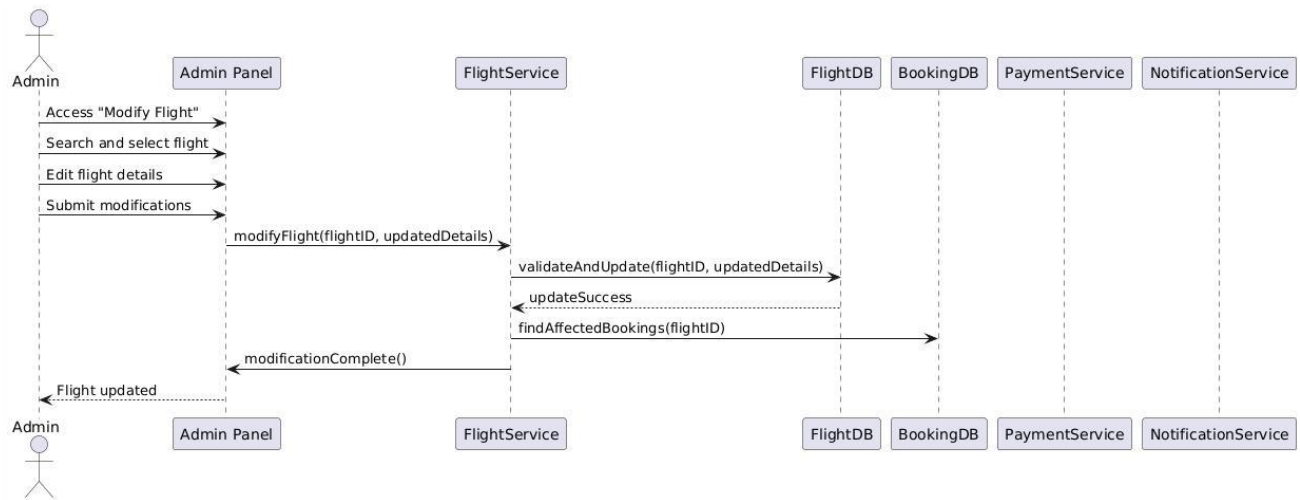
UC-501: Add Flight – Sequence Diagram



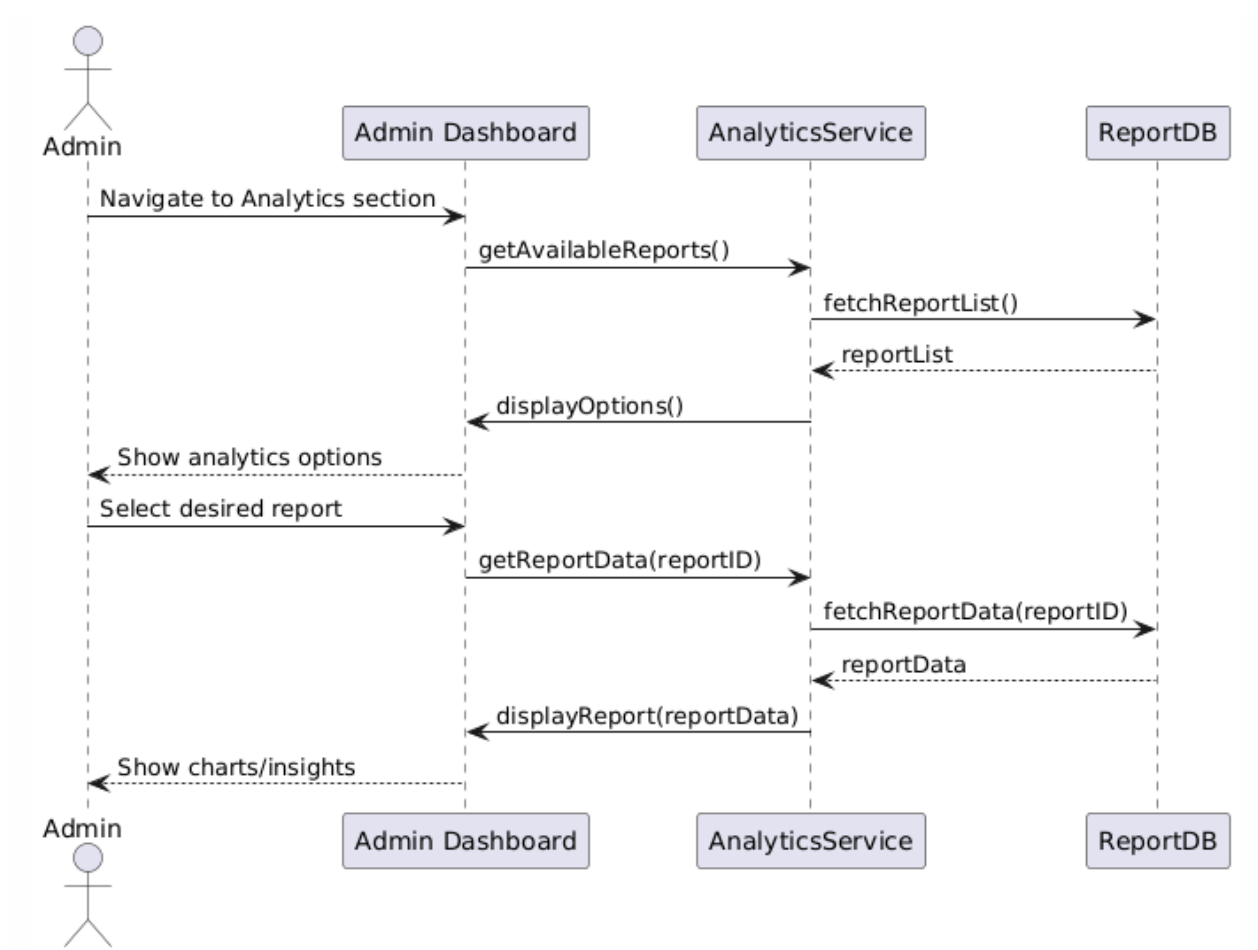
UC-502: Cancel Flight – Sequence Diagram

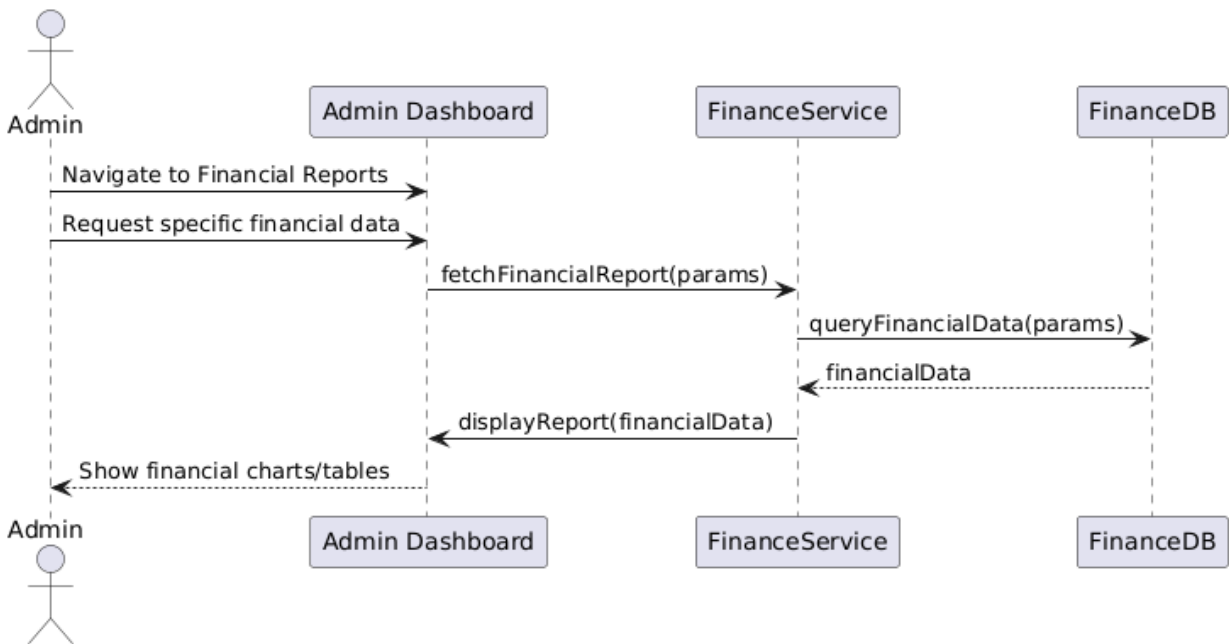


UC-503: Modify Flight & Issue Compensation – Sequence Diagram



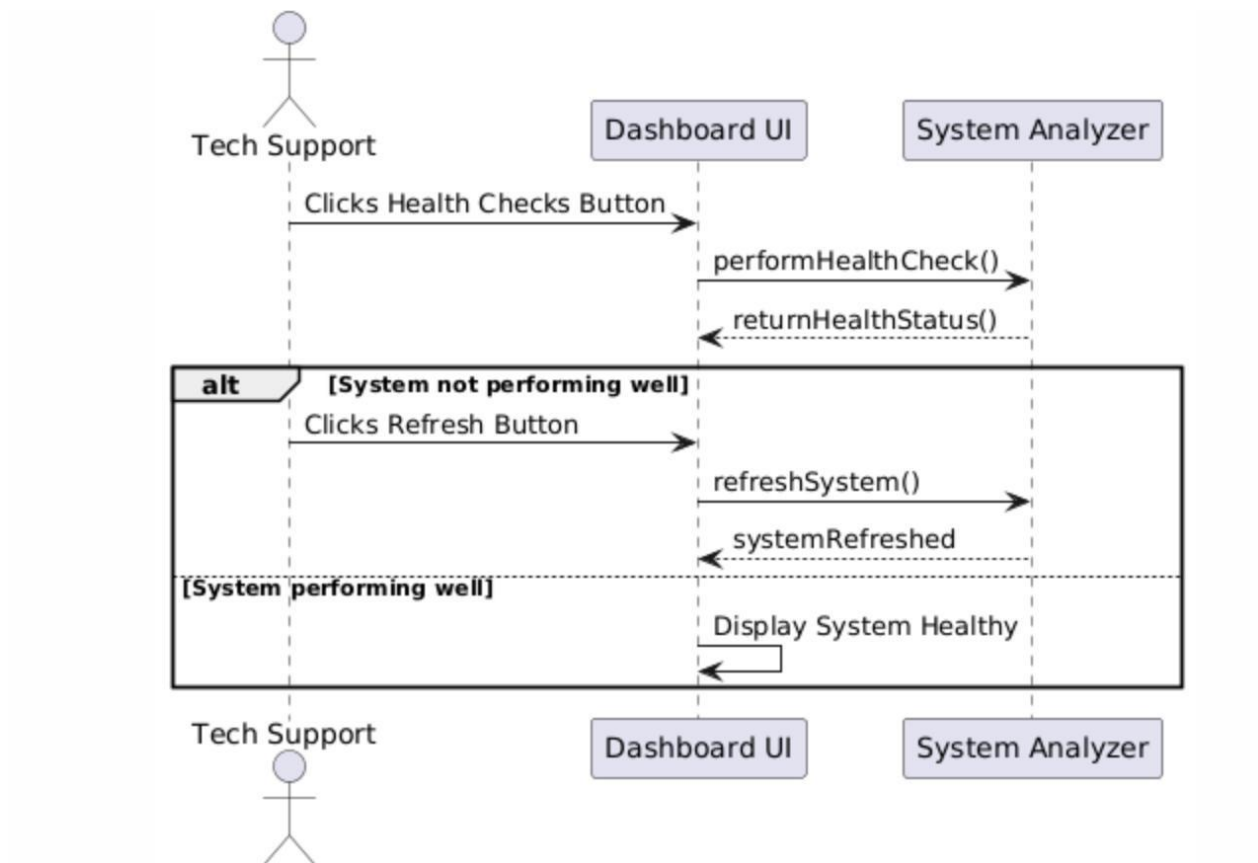
UC-504: Analytics Access – Sequence Diagram





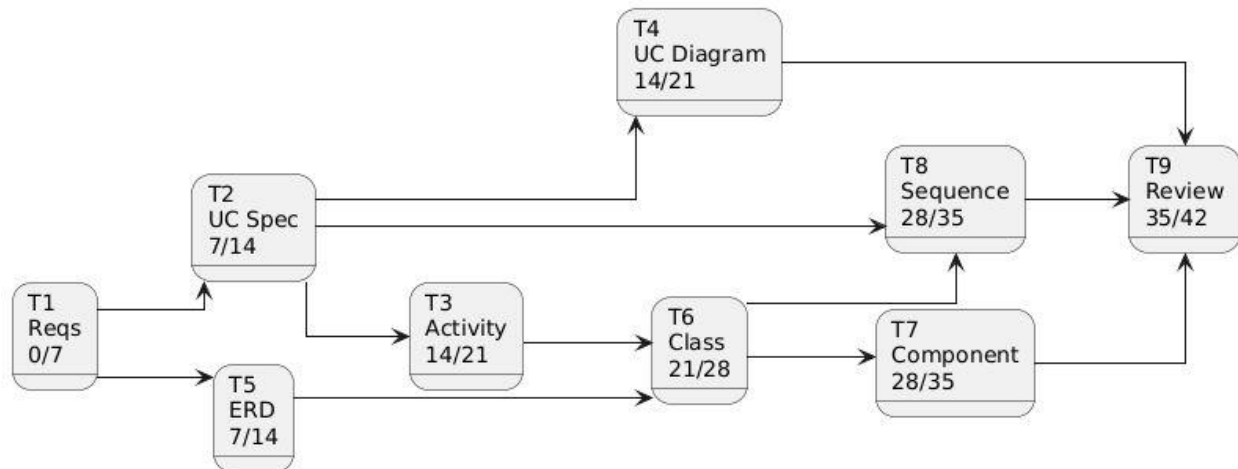
UC-505: Access Financial Reports – Sequence Diagram

UC-601: Software Functionality – Tech Support Troubleshooting – Sequence Diagram

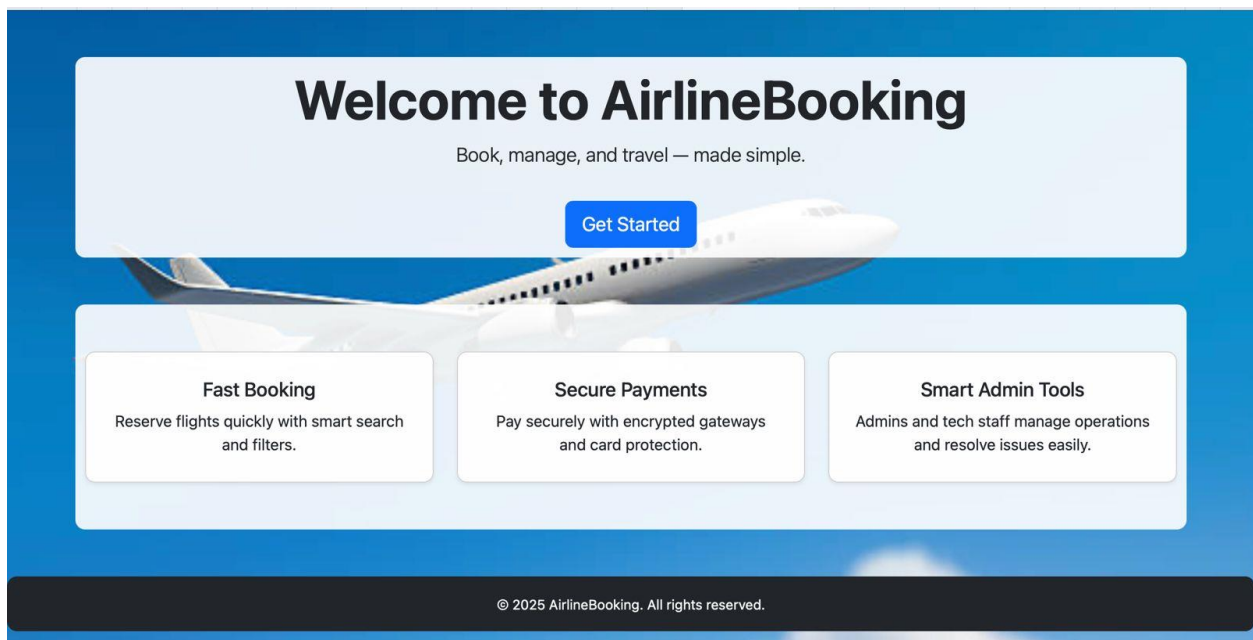


PROJECT PLANNING:

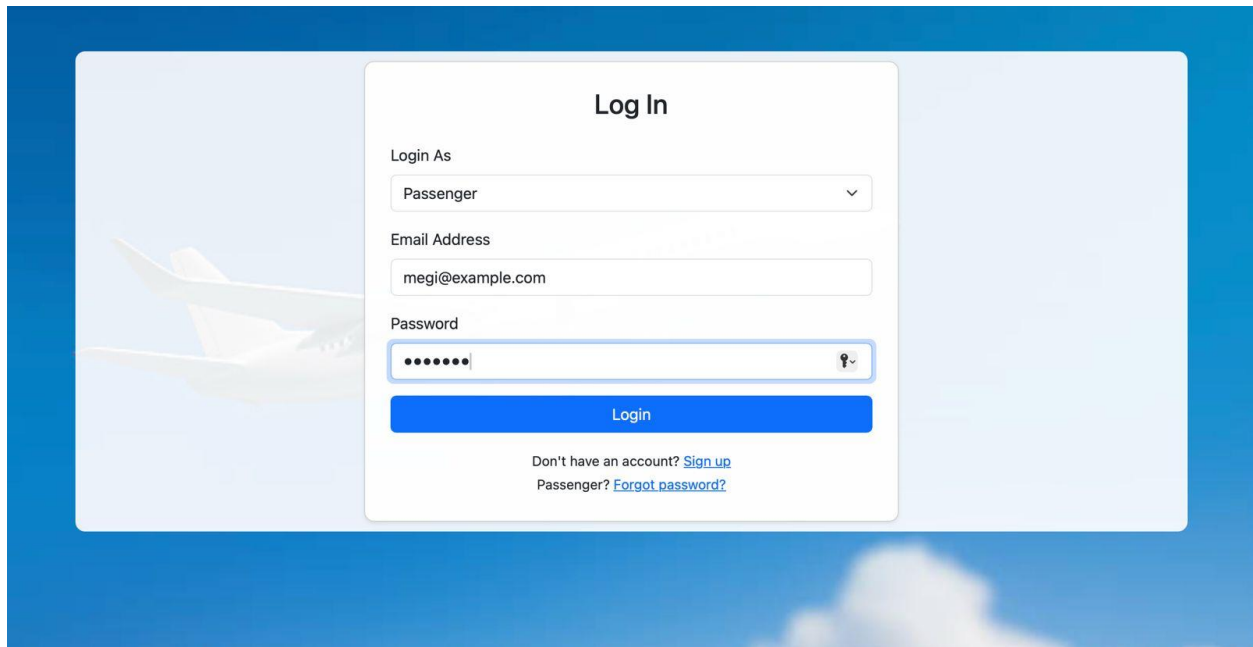
Task ID	Task Name	Description	Duration	Dependencies
T1	Requirements Gathering	Functional + non-functional requirements	7 days	—
T2	Use Case Specification	Write all use cases with actors & flows	7 days	T1
T3	Activity Diagrams	One per use case, showing flow of actions	7 days	T2
T4	Use Case Diagram	Visual overview of all use cases and actors	7 days	T2
T5	ERD Diagram	Define entities, attributes, and relationships	7 days	T1
T6	Class Diagram	Design system classes & relationships	7 days	T3, T5
T7	Component Diagram	High-level software modules/components	7 days	T6
T8	Sequence Diagrams	Detail object interaction for key scenarios	7 days	T2, T6
T9	Final Review & Validation	Cross-check all diagrams and specifications	7 days	T4, T7, T8



Code Implementation Output:



Airline Ticket Booking Software Specification



Log In

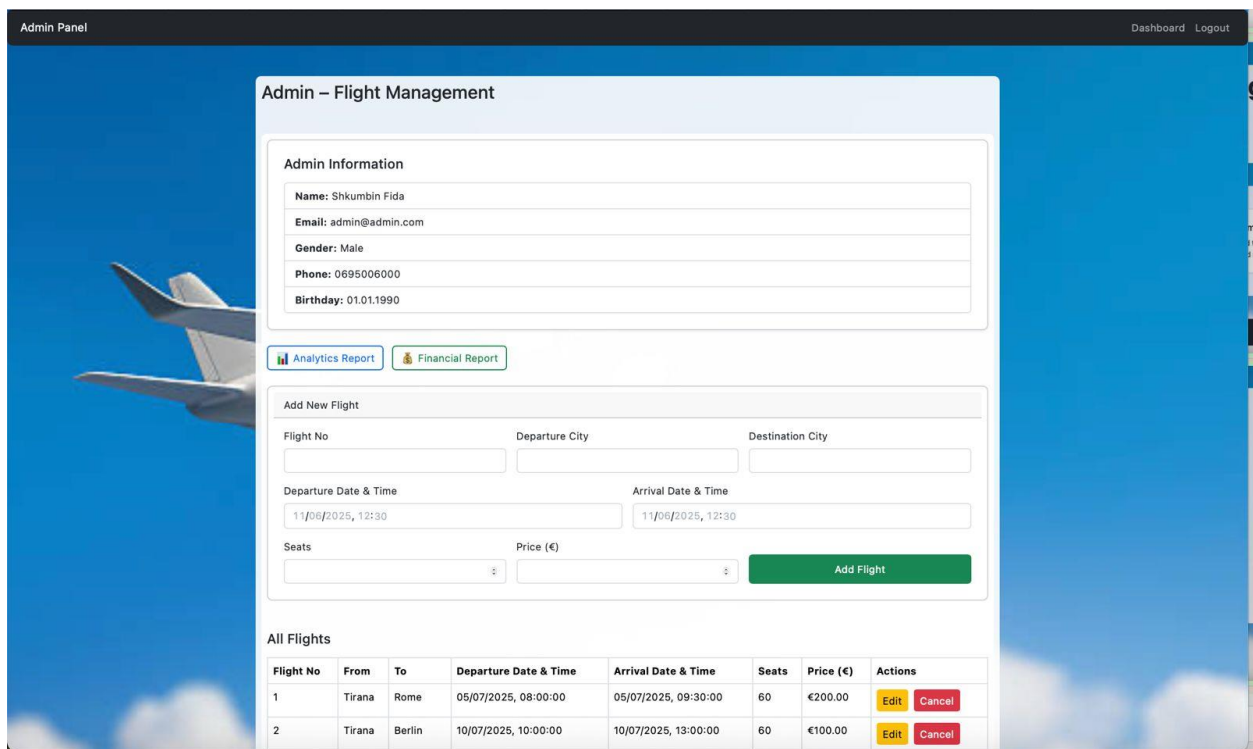
Login As
Passenger

Email Address
megi@example.com

Password
••••••

Login

Don't have an account? [Sign up](#)
Passenger? [Forgot password?](#)



Admin Panel Dashboard Logout

Admin – Flight Management

Admin Information

Name: Shkumbin Fida
Email: admin@admin.com
Gender: Male
Phone: 0695006000
Birthday: 01.01.1990

[Analytics Report](#) [Financial Report](#)

Add New Flight

Flight No: Departure City: Destination City:

Departure Date & Time: Arrival Date & Time:

Seats: Price (€): [Add Flight](#)

All Flights

Flight No	From	To	Departure Date & Time	Arrival Date & Time	Seats	Price (€)	Actions
1	Tirana	Rome	05/07/2025, 08:00:00	05/07/2025, 09:30:00	60	€200.00	Edit Cancel
2	Tirana	Berlin	10/07/2025, 10:00:00	10/07/2025, 13:00:00	60	€100.00	Edit Cancel

Tech Dashboard

MaintenanceLogout

Tech Support Information

Name: Edlira Cani

Email: tech@tech.com

Gender: Female

Phone: 0692000002

Birthday: 01.01.1990

Tech Support – System Maintenance

System Health Check

Run Health Check

Refresh System State

Refresh

Download System Report

Show System Report