SMIU

SINDH MADARESSATUL ISLAM UNIVERSITY, KARACHI

Lab Manual

Department: Software Engineering

Name: Nimra Babar

Student ID: BSE-23F-128

Subject: Software Requirement Engineering

Date of Submission:12th June,2025

Student Sign: Teacher Sign:

Software Requirements Specification

for

<Flight Management System>

Version 1.0 approved

Prepared by <Nimra Babar(BSE-23F-128)>

<Sindh Madreesatul Islam University>

<26-05-2025>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

LAB # 01

Objective:

To find the requirement specification (both functional and nonfunctional) of a given Problem.

Functional Requirements

User Registration

- **Requirement ID:** FR-001
- **Title:** User Registration
- **Description:** Users can create an account by providing their basic personal details.
- **Rationale:** Enables new users to access flight booking and related services.
- **Acceptance Criteria:**

    - A registration form collects necessary user details.
    - All mandatory fields must be filled to proceed.
    - On successful registration, users are redirected to the booking page.
    - If any field is missing, an error message appears.

User Login

- **Requirement ID:** FR-002
- **Title:** User Login
- **Description:** Registered users can log in using their email and password.
- **Rationale:** Secures access to personal bookings and preferences.
- **Acceptance Criteria:**

    - Email and password fields are required.
    - Users can opt to stay logged in on their device.
    - Incorrect login attempts are limited to three before temporary lockout.
    - A "Forgot Password" option is provided.

Flight Booking

- **Requirement ID:** FR-003
- **Title:** Search and Book a Flight
- **Description:** Users can search for flights by entering travel details and book available options.
- **Rationale:** Provides users with a way to plan and reserve flights.
- **Acceptance Criteria:**

    - Users can search by trip type, locations, dates, passengers, and class.
    - Search results include flight info like time, duration, class, and fare type.
    - Users can sort results and update their search criteria.
    - After selecting a flight, users can enter passenger details and proceed to booking.
    - After booking, users receive a confirmation and booking reference number.

Payment

- **Requirement ID:** FR-004
- **Title:** Payment
- **Description:** Users can complete their flight booking using different payment methods.

- **Rationale:** Allows users to confirm their reservation securely.
- **Acceptance Criteria:**

  - Payment options include card, bank transfer, or mobile wallet.
  - Missing or invalid fields show error messages.
  - On success, a booking confirmation and reference number are shown.
  - E-ticket can be generated after payment.
  - "Hold My Fare" option reserves booking without immediate payment.

Flight Status

- **Requirement ID:** FR-005
- **Title:** Flight Status
- **Description:** Users can check the status of flights using various methods.
- **Rationale:** Keeps users updated on flight timing and changes.
- **Acceptance Criteria:**

  - Users can search flight status by route, flight number, or upcoming bookings.
  - Displays include departure/arrival times, current status, terminal, and gate.

Baggage Details

- **Requirement ID:** FR-006
- **Title:** Baggage Details
- **Description:** Users can view baggage allowance based on their ticket class and route.
- **Rationale:** Helps users prepare and avoid extra baggage charges.
- **Acceptance Criteria:**

  - Shows standard and personal baggage limits.
  - Displays warnings about extra charges if limits are exceeded.

Weather and Emergency Alerts

- **Requirement ID:** FR-007
- **Title:** Weather and Emergency Alerts
- **Description:** Shows weather conditions and emergency alerts for departure/arrival airports.
- **Rationale:** Notifies users of possible delays or issues.
- **Acceptance Criteria:**

  - Displays current weather, delays, and alerts on relevant pages.
  - Alerts are shown for affected airports.

Staff Management

- **Requirement ID:** FR-008
- **Title:** Staff Management
- **Description:** Admins can manage staff records including pilots, crew, and ground staff.
- **Rationale:** Keeps internal staff information organized.
- **Acceptance Criteria:**

- Admin can add, update, or delete staff details.
- Staff can be searched by name or role.
- Only authorized admin users can access this section.

Flight Management

- **Requirement ID:** FR-009
- **Title:** Flight Management
- **Description:** Admins manage flight information such as schedules, crew, and aircraft.
- **Rationale:** Ensures flight data is accurate and updated.
- **Acceptance Criteria:**

  - Admin can add, update, or cancel flights.
  - Updates reflect in user-facing modules like booking and flight status.

Online Check-In

- **Requirement ID:** FR-010
- **Title:** Online Check-In
- **Description:** Users can check in online, select seats, and get their boarding pass.
- **Rationale:** Saves time and avoids airport delays.
- **Acceptance Criteria:**

  - Check-in available before departure within a specific time window.
  - Seat selection and e-boarding pass options are provided.
  - Check-in unavailable after the cutoff time.

Non Functional Requirements

Security

- **Requirement ID:** NFR-001
- **Title:** Security
- **Description:** The system must protect user data and payment information using encryption and secure authentication.
- **Rationale:** To prevent unauthorized access and ensure user privacy.
- **Acceptance Criteria:**

  - All sensitive data must be encrypted in storage and transmission.
  - User passwords must be stored securely using hashing.
  - Login attempts are limited to prevent brute-force attacks.
  - Payment information complies with industry security standards (e.g., PCI-DSS).

Performance

- **Requirement ID:** NFR-002
- **Title:** Performance
- **Description:** The system should respond quickly and perform efficiently under normal and peak usage.
- **Rationale:** To ensure a smooth and responsive experience for users.

- **Acceptance Criteria:**

  - Search and booking operations must complete within 3 seconds.
  - The system must handle at least 1000 concurrent users without performance degradation.
  - System response times for page navigation must not exceed 2 seconds.

Availability

- **Requirement ID:** NFR-003
- **Title:** Availability
- **Description:** The system should be accessible and functional most of the time without unexpected downtime.
- **Rationale:** To provide continuous service to users when needed.
- **Acceptance Criteria:**

  - The system must maintain at least 99.5% uptime per month.
  - Scheduled maintenance should be announced 24 hours in advance.
  - In case of system failure, recovery should occur within 15 minutes.

Usability

- **Requirement ID:** NFR-004
- **Title:** Usability
- **Description:** The system must provide an intuitive and user-friendly interface for all user types.
- **Rationale:** To reduce the learning curve and improve user satisfaction.
- **Acceptance Criteria:**

  - The interface should support both desktop and mobile views.
  - Common tasks (e.g., booking a flight) should be completable within 3 clicks.
  - The system should follow standard UI/UX best practices.
  - Help or tooltips must be available for important fields or features.

Scalability

- **Requirement ID:** NFR-005
- **Title:** Scalability
- **Description:** The system should be capable of expanding to support an increasing number of users and data.
- **Rationale:** To accommodate future growth and peak usage scenarios.
- **Acceptance Criteria:**

  - The system should support a minimum of 5000 concurrent users.
  - Additional servers or resources can be added without major architectural changes.
  - Database should scale horizontally or vertically as needed.

Introduction

Purpose

This Software Requirements Specification (SRS) describes the requirements for the **Flight Management System (FMS)**, which is designed to help users book flights, check flight status, manage bookings, and handle all related services. The system also includes admin functionalities to manage flights and staff.

This document outlines both functional (what the system should do) and non-functional (how the system should behave) requirements for the system. It includes details on modules like User Registration**,** User Login, Flight Booking, Payment, Flight Status, Baggage Details, Weather and Emergency Alerts, Staff Management, Flight Management, and Online Check-In.

This SRS is written for **Version 1.0** of the FMS and focuses on the core modules needed for a complete and functional online flight booking and management experience.

Document Conventions

Requirement IDs follow the format:

- **FR-XXX** for Functional Requirements
- **NFR-XXX** for Non-Functional Requirements

Bold titles indicate section headers or important features throughout the document.

- Bullet points (•) list specific items or features.
- Arrows (➢) are used for clarity in acceptance criteria.

Each requirement includes:

- A unique ID
- A clear title and description
- A rationale (why the feature is needed)
- Acceptance criteria (when the requirement is considered fulfilled)

Font Specifications:

- **Headings:** Times New Roman, 14pt (or 18pt if preferred for major sections like "1. Introduction")
- **Body Text:** Arial, 11pt

Intended Audience and Reading Suggestions

This document is intended for a wide range of readers involved in the development and use of the FMS, including:

- **Software Developers**: for implementation of modules and features
- **Project Managers**: to track the progress and ensure all modules meet requirements
- **Testers/QA Team**: to verify if the system meets all criteria before release
- **Clients and Stakeholders**: to understand what the system offers
- **End Users and Support Teams**: to learn the system’s capabilities and usage flow

Reading suggestions:

- Begin with Section 1 (Introduction) to understand the overall purpose and audience.
- Proceed to Section 2 (Overall Description) for system overview and assumptions.
- Review Section 3 (System Features) for detailed functional requirements.
- Section 4 (Non-Functional Requirements) outlines how the system should behave under various conditions.
- Use the References section if you need supporting documents or standards.

LAB # 02

Product Scope

The **Flight Management System (FMS)** is a web-based application that enables users to:

- Register and log into their accounts securely
- Search and book flights based on preferences
- Make payments using multiple methods
- Check real-time flight status and weather alerts
- Manage baggage allowances and avoid extra charges
- Perform online check-in and get boarding passes

The system also includes **administrative tools** to manage flights, schedules, crew members, and ground staff.

Goals of the FMS:

- Simplify the flight booking experience
- Provide up-to-date travel and flight information
- Enable secure and reliable services
- Ensure scalability and usability for future growth

It is aligned with modern digital trends in the airline industry and aims to provide a seamless travel planning experience for passengers while keeping operational tasks organized for administrators.

References

- [1] Thai Airways International, "Book Flights to Bangkok, Thailand." Accessed: June 7, 2025. [Online]. Available: https://www.thaiairways.com/en/plan_my_trip/Special_fare/Offers_Booking/thailand/bangkok.page
- [2] China Southern Airlines, "Flight Booking & Ticket Inquiry | Book Your Flight Easily." Accessed: June 7, 2025. [Online]. Available: https://www.csair.com/
- [3] Krazytech, "Software Requirements Specification document with example." Accessed: June 7, 2025. [Online]. Available: https://krazytech.com/software-requirements-specification-document-with-example/

LAB # 03

Overall Description

Product Perspective

The Flight Management System (FMS) is a **new, standalone software solution** designed to manage the end-to-end process of booking and managing flights. It does not depend on any previous system and is being developed from scratch. It will serve both passengers and administrative staff in an integrated environment.

Although the FMS is an independent system, it will connect with **external systems** like:

- Payment gateways for processing online payments
- Weather data providers for real-time alerts
- Third-party authentication services for secure user login

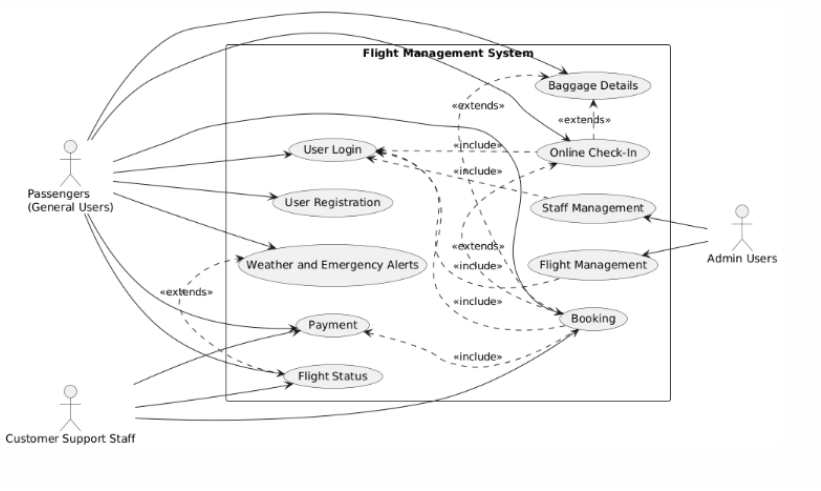
This system aims to replace traditional, manual, or disconnected systems used by airlines for booking, flight management, and customer service.

Product Functions

The FMS offers the following core functionalities for users and administrators:

- **User Registration and Login**
  Enables users to create accounts and securely log in.
- **Flight Search and Booking**
  Users can search for flights based on various criteria and complete reservations.
- **Payment Processing**
  Supports secure payments through multiple methods like card, mobile wallet, or bank transfer.
- **Flight Status Tracking**
  Real-time updates on flight status, terminals, and delays.
- **Online Check-In**
  Allows users to check in online, select seats, and download boarding passes.
- **Baggage Allowance Details**
  View baggage policies based on ticket class and route.
- **Weather and Emergency Alerts**
  Users are notified of delays or emergencies due to weather or operational issues.
- **Admin Panel for Flight and Staff Management**
  Admins can manage flight schedules, assign staff, and maintain crew.

User Classes and Characteristics



The system is intended for the following user types:

1. Passengers (General Users)

- **Characteristics:**
    - Primary users of the system for flight-related activities.
    - Require a way to register and log in to the system.
    - Can search for available flights and proceed with booking.
    - Perform payments for booked flights.
    - Check real-time flight status.
    - Access information regarding baggage details.
    - Receive weather and emergency alerts related to flights.
    - Can perform online check-in for their flights.

2. Customer Support Staff

- **Characteristics:**
    - Assist passengers with their flight-related queries and issues.
    - Can perform flight booking on behalf of passengers.
    - Process payments for bookings.
    - Check flight status to provide accurate information to customers.
    - Likely have access to passenger information and booking details for assistance.

3. Admin Users

- **Characteristics:**
    - Have administrative privileges over the Flight Management System.
    - Responsible for managing staff accounts and roles within the system.
    - Handle the overall management of flights, including adding, modifying, or canceling flights.
    - Possess elevated access rights compared to passengers and customer support staff.

LAB # 04

Operating Environment

The Flight Management System will operate in the following environments:

- Web-based Application accessible via standard browsers (Chrome, Firefox, Edge, Safari)
- Compatible with both desktop and mobile devices
- Backend system runs on a Linux/Windows-based server
- Developed using technologies like Java/PHP, MySQL/PostgreSQL, and secure web protocols (HTTPS, SSL)

Design and Implementation Constraints

Several factors will guide or limit the development process:

- **Compliance with PCI-DSS** standards for handling payments
- **Security requirements**: data encryption, user authentication, limited login attempts
- Must support **scalability** and allow cloud-based deployment
- **Responsive design** for mobile and desktop compatibility
- Interface with **external APIs** (for weather, payments)
- Code should follow **standard naming conventions** and **clean architecture** for maintainability

User Documentation

The following documentation will be provided with the system:

- **User Manual**: Step-by-step instructions for using all user features
- **Admin Guide**: Instructions for managing flights, staff, and users
- **Quick Start Guide**: Brief overview for first-time users
- **Online Help/Tutorials**: Integrated tooltips and FAQs in the system interface

These documents will be available in PDF format and accessible from within the application interface.

Assumptions and Dependencies

This SRS is based on the following assumptions and dependencies:

- The system will have reliable internet connectivity at all times
- Payment gateway APIs (e.g., card payments, mobile wallets) will be available and properly integrated
- External APIs for weather and flight status will respond reliably
- All users will have access to a modern browser
- Admins and staff will be trained before using the system
- Future updates or feature additions will follow the initial modular structure of the software

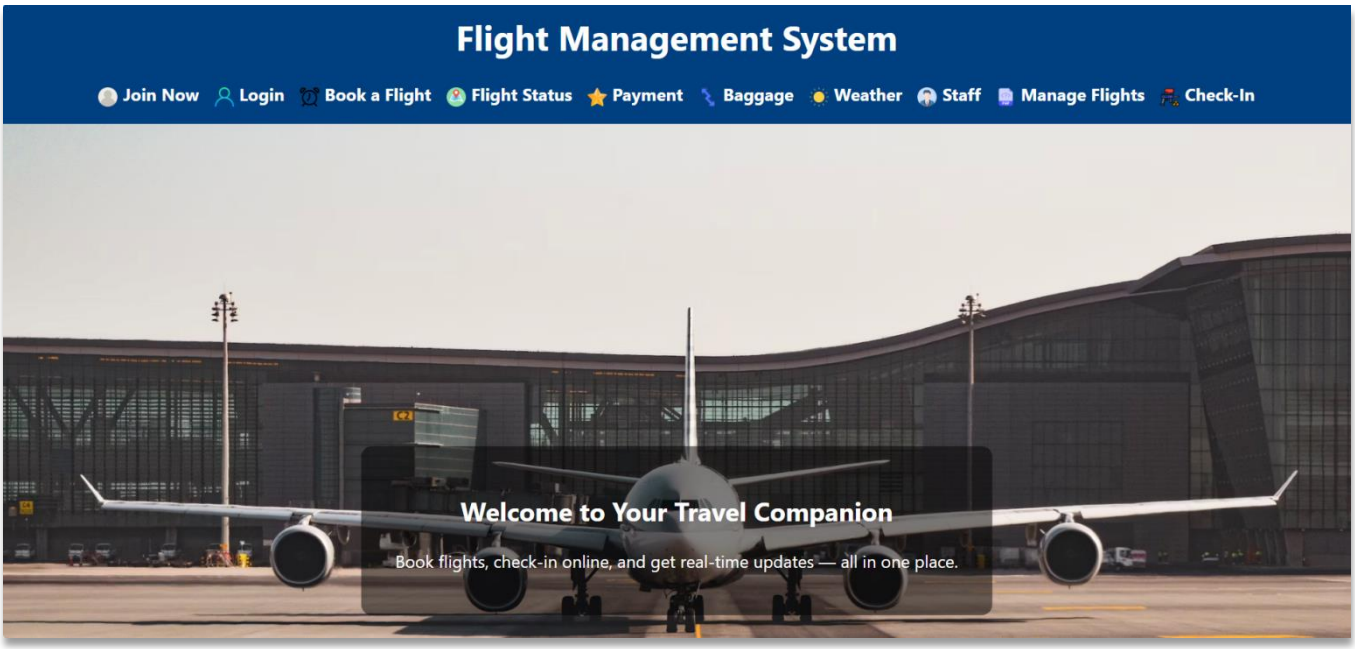
If any of these assumptions change, system functionality or delivery timelines could be impacted.

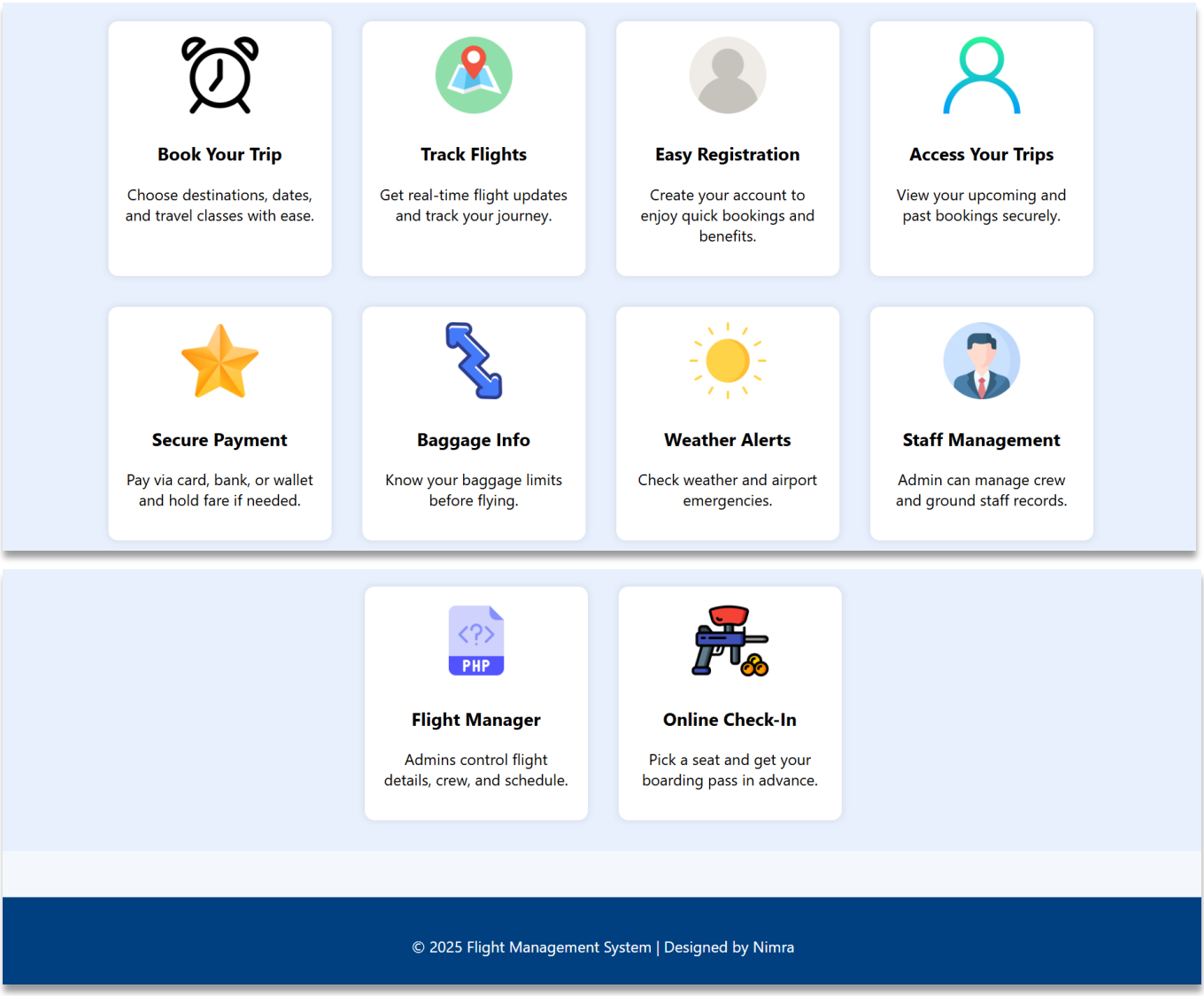
LAB # 05

External Interface Requirements

User Interfaces

This section elaborates on the interaction points for different user roles within the Flight Management System, maintaining a clean and user-friendly interface across web and mobile.

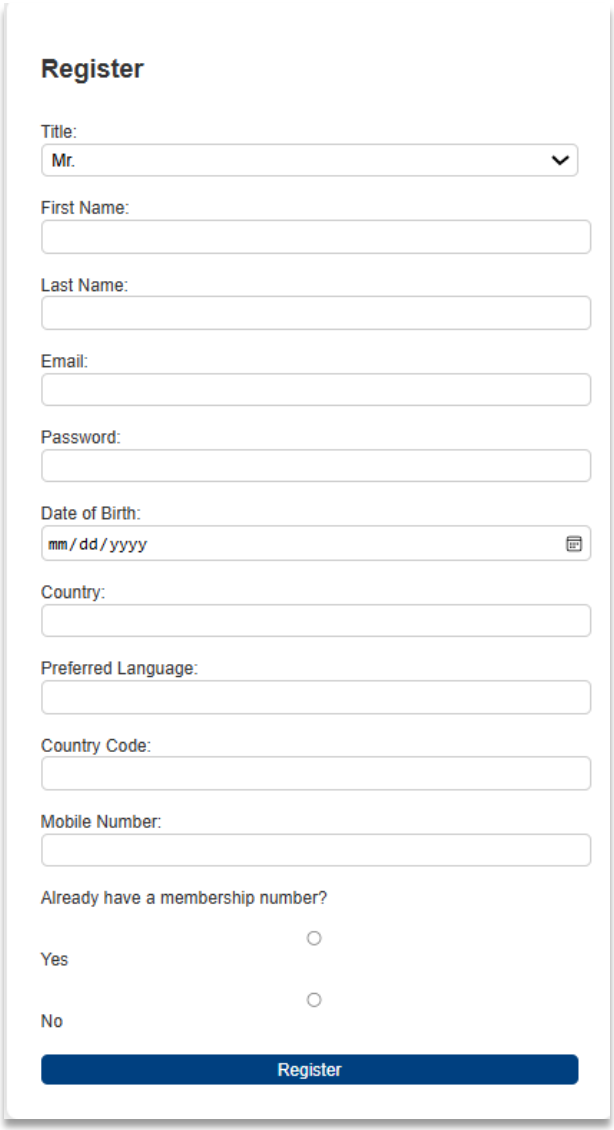




User Registration

- **Layout:** Centered form with "Register" title. Simple, vertical layout.
- **Elements:** Title (Dropdown: Mr., Ms., Mrs.), First Name (Text), Last Name (Text), Email (Text), Country (Text), Preferred Language (Text), Country Code (Text), Password (Hidden), Date of Birth (Date picker), Mobile Number (Telephone), Membership? (Radio: Yes/No).
- **Buttons:** Register (Submits form).
- **Navigation:**

➢ Success: Redirects to booking/login.
➢ Errors: Shows messages to fix missing/wrong details.

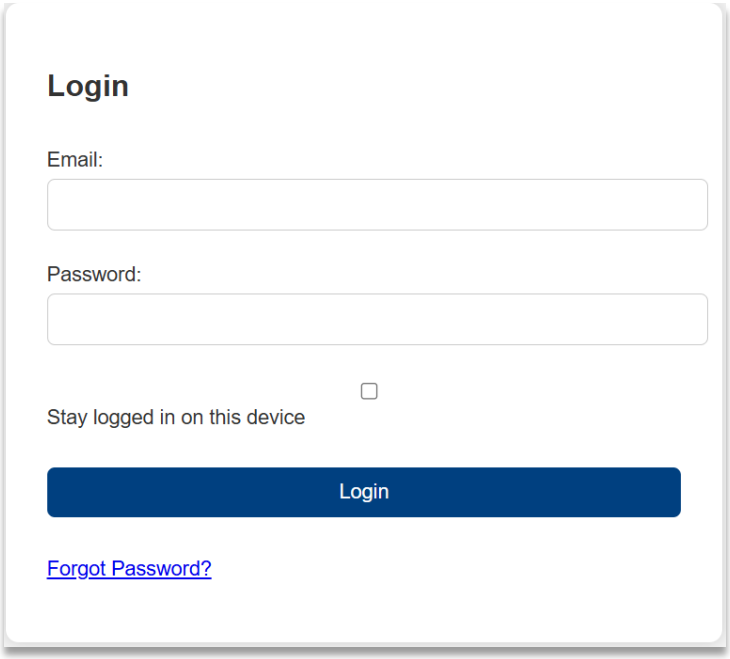


Login

- **Layout:** Clean, centered form with "Login" title. Input fields stacked vertically.
- **Elements:**

➢ Email Field: Label "Email", Placeholder "Enter your registered email".
➢ Password Field: Label "Password", Placeholder "Enter your password", Hidden input.
➢ Checkbox: "Stay logged in on this device" (optional).
➢ Link: "Forgot Password?" (redirects to recovery).

- **Buttons:** Login (Submits credentials).
- **Navigation:** Redirects to Booking Page.



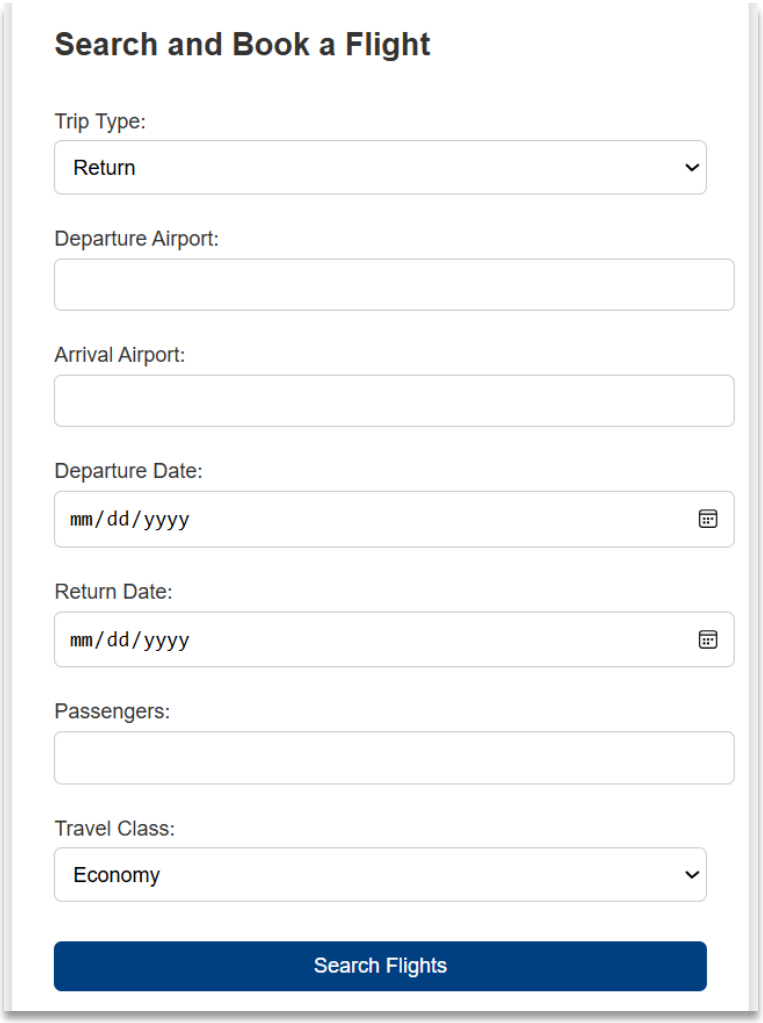
Flight Booking

- **Layout:** Clear form titled “Search and Book a Flight.” Elements arranged vertically.
- **Elements:**

➢ Dropdown: Trip Type (Return, One Way, Multi-City).
➢ Text Input: Departure Airport (Required), Arrival Airport (Required).
➢ Date Selector: Departure Date (Required), Return Date (Optional).
➢ Number Input: Passengers (Required).
➢ Dropdown: Travel Class (Economy, Business).

- **Buttons:** Search Flights (Initiates search).
- **Navigation:**

➢ Enter details, submit to view flights.
➢ Select flights, proceed to passenger details and payment.



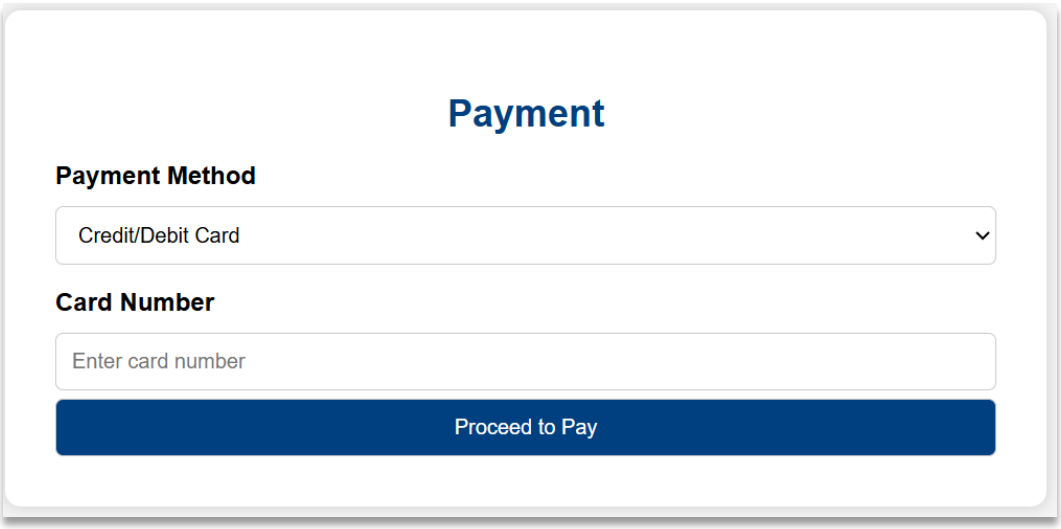
Payment

- **Layout:** Centered form titled “Payment,” with vertical arrangement.
- **Elements:**

➢ Dropdown: Payment Method (Credit/Debit Card, Bank Transfer, Mobile Wallet).
➢ Text Input: Card Number or account number.

- **Buttons:** Proceed to Pay (Submits payment).
- **Navigation:**

➢ Select method, enter details.
➢ Click "Proceed to Pay."
➢ System processes payment, confirms booking with reference and e-ticket.



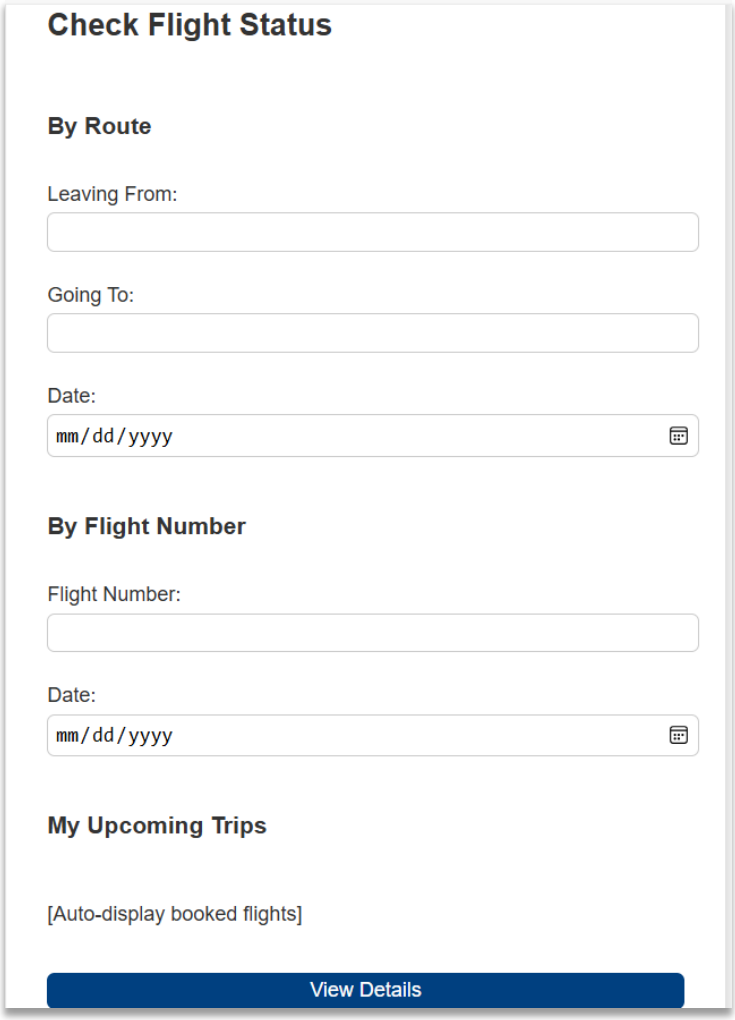
Flight Status

- **Layout:** Form titled “Check Flight Status,” divided into By Route, By Flight Number, and My Upcoming Trips sections.
- **Elements:**

➢ By Route: Departure/Destination cities, Travel Date.
➢ By Flight Number: Specific Flight Number and Date.
➢ My Upcoming Trips: Auto-displays logged-in user's bookings.

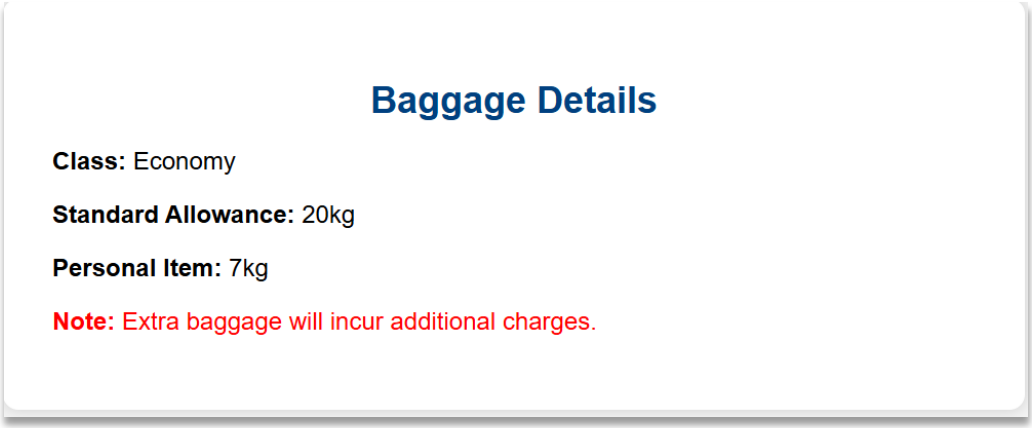
- **Buttons:** View Details (Fetches status).
- **Navigation:**

➢ Use route or flight number search.
➢ Click “View Details” for real-time status (on-time, delayed, cancelled).
➢ Logged-in users see upcoming trips automatically.



Baggage Details

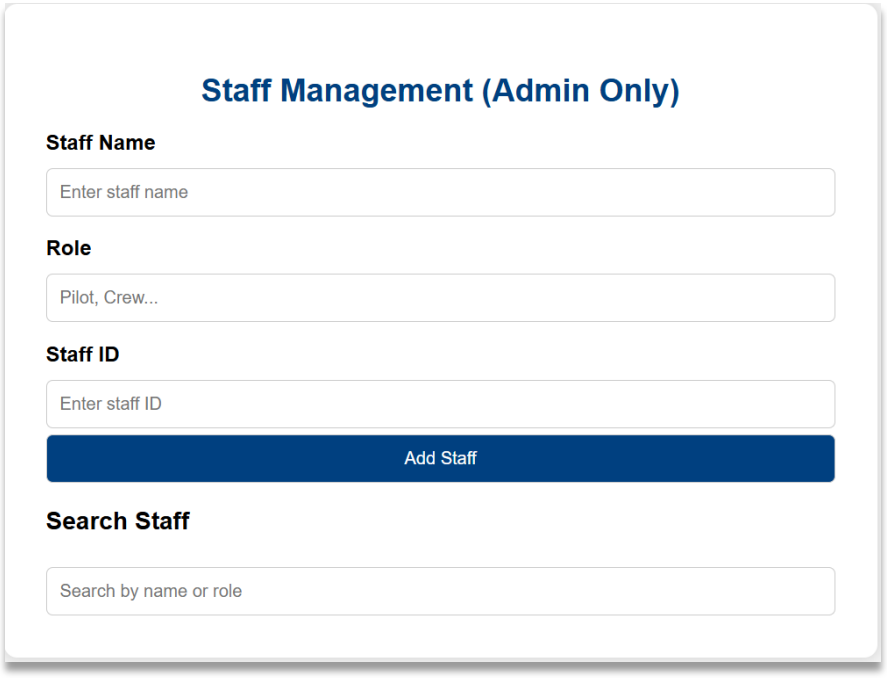
- **Layout:** Static informational panel in a centered module.
- **Elements:** Presents baggage allowance: Class (Economy), Standard Allowance (20kg), Personal Item (7kg). Note: Warning for extra charges.
- **Buttons:** None (informational page).
- **Navigation:** Accessed from flight booking confirmation or trip summary.



- **Layout:** Real-time weather and emergency alerts in structured modules.
- **Elements:** Current Weather Information (Status, temp, icon for selected airport), Emergency Alert Message (prominently styled warning, red text/icons).
- **Buttons:** None (informational display).
- **Navigation:** Accessible via flight booking or flight status section.

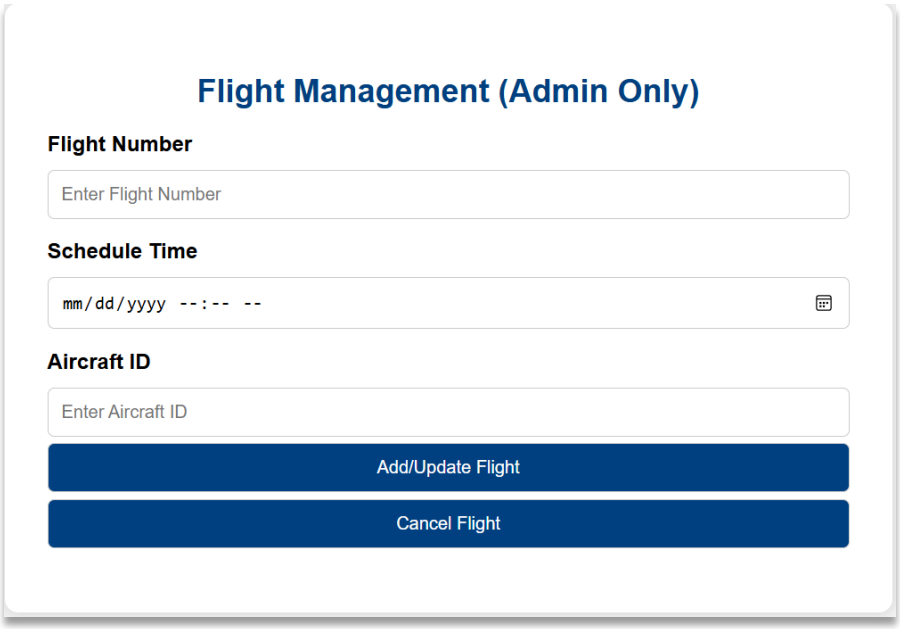
Staff Management (Admin Only)

- **Layout:** Divided into Add Staff and Search Staff sections. Admin-only access.
- **Elements:**
- Add Staff Form: Text Fields for Staff Name, Role, Staff ID.
- Search Section: Search Bar (by name or role).
- **Buttons:** Add Staff (Submits new entry).
- **Navigation:** Accessible from main dashboard/admin panel.



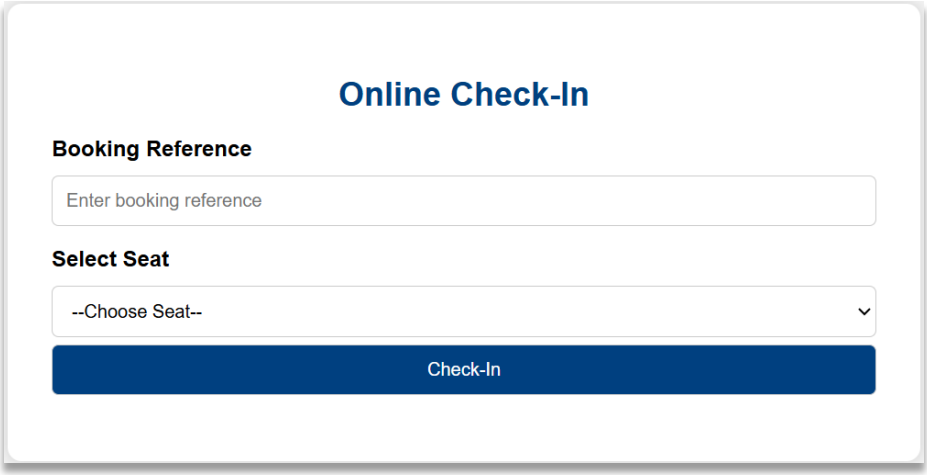
Flight Management (Admin Only)

- **Layout:** Structured form to add, update, or cancel flight schedules.
- **Elements:**
- Text Field: Flight Number.
- Date-Time Picker: Schedule Time.
- Text Field: Aircraft ID.
- **Buttons:** Add/Update Flight (Submits data), Cancel Flight (Triggers cancellation).
- **Navigation:** Accessible from dashboard/admin panel.



Online Check-In

- **Layout:** User-friendly page with a clean form.
- **Elements:** Booking Reference Field (text input), Seat Selection Dropdown (Window, Aisle, Middle).
- **Workflow:** After form submission, shows "Check-In Complete" message and a "Download Boarding Pass" button.
- **Behavior:** Input validation; boarding pass section hidden until successful check-in.



GUI Standards

The design and implementation of the Flight Management System's user interface follow key industry-recognized GUI standards to ensure consistency, usability, and accessibility:

Material Design Guidelines

- **Applied Concepts:** Layout grid, elevation, card-based modules, and intuitive form components.
- **Reason:** Material Design provides a clean and responsive structure that enhances user engagement and consistency across all modules.

Bootstrap Guidelines (Grid and Responsiveness)

- **Usage:** Although custom styling is used, layout design principles such as spacing, alignment, and form responsiveness are inspired by Bootstrap's 12-column grid system and mobile-first design.
- **Benefit:** Ensures responsive behavior across desktops, tablets, and mobile devices.

WCAG Accessibility Guidelines

- **Compliance Aspects:**
  - Clear labels and instructions on all forms.
  - High contrast for alerts and important messages (e.g., red warnings for storm delays).
  - Logical navigation order for screen readers.
  - Form fields marked with required attributes for validation.
- **Purpose:** Makes the system usable for individuals with visual or motor impairments and ensures inclusive access.

Hardware Interfaces

The Flight Management System (FMS) is a web-based application that interacts with various hardware components to ensure smooth and real-time operations. The software interfaces with the following hardware systems:

Web Server

- **Role:** Hosts the front-end and back-end components of the FMS, including the user interface, authentication, booking, and check-in modules.
- **Interaction:** Handles HTTP/HTTPS requests and responses between users and the system.

Database Server

- **Role:** Stores and manages all critical data such as user information, flight schedules, booking records, baggage details, and staff profiles.
- **Interaction:** Communicates with the FMS through SQL queries for data retrieval, insertion, and updates.

Airport Systems (Optional Integration)

- **Examples:** Flight display boards, announcement systems.
- **Interaction:** Can receive real-time updates from the FMS for flight delays, gate changes, and weather alerts (if integrated via APIs or middleware).

Barcode Scanners (Check-In Counters)

- **Role:** Used for scanning boarding passes or booking references during manual check-in at the airport.
- **Interaction:** Sends scanned data to the FMS to verify check-in status and retrieve booking details.

Printer (Boarding Pass / Receipt)

- **Role:** Prints physical boarding passes or payment receipts if check-in is done via airport kiosk or admin panel.
- **Interaction:** Accepts data from the FMS and prints it using connected printer hardware.

LAB # 06

Traceability Matrix:

Requirement ID	Requirement Description	Source (Stakeholder)	Test Case ID	Test Case Description	UAT
FR-001	User can register with personal details	Passengers	TC-001	Verify that all required fields are displayed	
	User can register with personal details		TC-002	Submit empty form and check for error messages	
	User can register with personal details		TC-003	Submit valid form and verify redirection to booking page	
FR-002	Registered user can log in	Passengers, Support Staff	TC-004	Verify login form contains email and password fields	
	Registered user can log in		TC-005	Attempt login with wrong password 3 times	
	Registered user can log in		TC-006	Login with “Stay Logged In” option	
	Registered user can log in		TC-007	Click on “Forgot Password” and verify reset process	
FR-003	Search and book a flight	Passengers, Support Staff	TC-008	Search flights by trip type, locations, and dates	
	Search and book a flight		TC-009	Sort or update search criteria and check response	
	Search and book a flight		TC-010	Book selected flight and verify booking confirmation	
FR-004	Complete booking with payment	Passengers, Admin Users	TC-011	Test card, bank, and mobile wallet payments	
	Complete booking with payment		TC-012	Submit invalid or missing payment info	
	Complete booking with payment		TC-013	Use “Hold My Fare” option and verify reservation	
FR-005	Check flight status	Passengers, Support Staff	TC-014	Search flight status by flight number	
	Check flight status		TC-015	Verify displayed details like gate, terminal, status	
FR-006	View baggage allowance	Passengers, Support Staff	TC-016	View baggage limits based on ticket type and route	
	View baggage allowance		TC-017	Try exceeding baggage and check for warnings	

FR-007	Show weather and emergency alerts	All Users	TC-018	Check current weather conditions at departure/arrival	
	Show weather and emergency alerts		TC-019	Verify alerts for delayed or emergency-affected airports	
FR-008	Manage staff	Admin Users	TC-020	Add a new staff member and verify entry	
	Manage staff		TC-021	Update an existing staff record	
	Manage staff		TC-022	Delete a staff member	
	Manage staff		TC-023	Try accessing staff page as non-admin	
FR-009	Manage flight information	Admin Users	TC-024	Add a new flight with all required details	
	Manage flight information		TC-025	Update an existing flight's schedule	
	Manage flight information		TC-026	Cancel a flight and verify removal from booking list	
FR-010	Online check-in	Passengers	TC-027	Try checking in within allowed time window	
	Online check-in		TC-028	Attempt check-in after cutoff time	
	Online check-in		TC-029	Select seat and download boarding pass	

LAB # 07

Software Interfaces

This section describes how the Flight Management System (FMS) connects with other software components, including databases, operating systems, libraries, and third-party services used in the system.

Connections and Components:

- **Database:**
  - MySQL (version 8.0 or compatible)
    Stores all user data, flight information, bookings, payment records, staff details, and system logs.
    The backend interacts with MySQL using standard SQL queries.
- **Operating System:**
  - The backend server runs on a windows/Linux-based OS (e.g., Ubuntu 20.04 LTS) which provides a stable environment for the web server and database.
- **Backend Environment:**
  - The server-side logic is developed using Java (JDK 11+) or PHP 7.4+, depending on implementation preference.
    These languages handle processing, database interaction, and communication with external services.

- **Frontend Components:**

  - The user interface is built with HTML5 and CSS3, ensuring compatibility across modern browsers (Chrome, Firefox, Edge, Safari).
  - JavaScript is optionally used for client-side validation and interactivity.

- **External APIs:**

  - Payment Gateway APIs (e.g. PayPal) accessed over HTTPS for secure payment processing.
  - Weather and Flight Status APIs providing real-time data used to update flight statuses and alerts.

- **Libraries and Tools:**

  Common libraries used include:

  - Database connectors/drivers (e.g., MySQL Connector/J for Java or mysqli for PHP)
  - JSON parsers for API data handling
  - SSL/TLS libraries for secure communications.

- **Integrated Commercial Components:**

  - Payment gateways (third-party services) for transaction handling.
  - Email services (like SMTP servers) for sending booking confirmations and alerts.

Communications Interfaces

This section outlines how the Flight Management System handles communication functions, including protocols, message formats, and security.

Communication Requirements:

- **Web Browser Interaction:**

  - Users access the FMS through standard web browsers (Chrome, Firefox, Edge, Safari).
  - The frontend and backend communicate using the HTTP/HTTPS protocols, with HTTPS enforced to secure data transmission.

- **Data Transfer and Protocols:**

  - All user input and system responses are exchanged via HTTP POST and GET requests.
  - Data formats are primarily in JSON or standard form-urlencoded formats for easy parsing.

- **Email Communication:**

  - The system sends emails (booking confirmations, password resets, alerts) through SMTP protocol using a configured mail server or third-party email provider.
  - Emails follow standard MIME formatting.

- **External API Communication:**

- Backend interacts with third-party services (payment gateways, weather data providers) using HTTPS REST APIs.
- Requests and responses are formatted in JSON.

- **Security and Encryption:**

  - All communication between client browsers and the server is secured using TLS/SSL encryption (HTTPS).
  - Sensitive data such as passwords and payment details are encrypted during transmission.
  - Backend servers use secure connections to external APIs.

- **Data Transfer Rates and Synchronization:**

  - The system operates on standard internet connections, with typical web page loads expected within 2-5 seconds depending on network speed.
  - Flight status and weather data are updated on demand or via scheduled backend polling.
  - Synchronization between modules (e.g., bookings and payments) occurs immediately after transactions to maintain data consistency.

LAB # 08

System Features

This section outlines the major **functional components** (modules) of the Flight Management System (FMS). Each module represents a core service offered by the system. The features are organized **module by module** to follow the logical and functional flow of the system.

For each feature, the following details are provided:

- **4.1.1 Description and Priority:** A short description of the feature and its implementation priority (High, Medium, Low).
- **4.1.2 Stimulus/Response Sequences:** The sequence of user actions (**Stimulus**) and corresponding system responses (**Response**) based on related functional and non-functional requirements.

System Feature 1

4.1 User Registration

4.1.1 Description and Priority

Description: Users can create an account by providing their basic personal details.

Priority: High

4.1.2 Stimulus/Response Sequences

- **Stimulus:** User enters personal details into the registration form and clicks **Register**.
- **Response:** System validates input, creates the user account, and redirects the user to the booking/login page.

4.2 User Login

4.2.1 Description and Priority

Description: Registered users can log in using their email and password.

Priority: High

4.2.2 Stimulus/Response Sequences

- **Stimulus:** User enters credentials and clicks login.
- **Response:** System verifies credentials and grants access.

4.3 Flight Booking

4.3.1 Description and Priority

Description: Users can search for flights by entering travel details and book available options.

Priority: High

4.3.2 Stimulus/Response Sequences

- **Stimulus:** User enters travel details (e.g., trip type, locations, dates, passengers, class) and clicks "Search Flights."
- **Response:** System displays available flights matching the criteria, allowing users to select and proceed to booking.

4.4 Payment

4.4.1 Description and Priority

Description: Users can complete their flight booking using different payment methods.

Priority: High

4.4.2 Stimulus/Response Sequences

- **Stimulus:** User selects a payment method and confirms transaction details.
- **Response:** System processes payment, displays booking confirmation, generates an e-ticket, and provides a reference number.

4.5 Flight Status

4.5.1 Description and Priority

Description: Users can check the status of flights using various methods.

Priority: Medium

4.5.2 Stimulus/Response Sequences

- **Stimulus:** User searches for flight status by route, flight number, or accesses their upcoming bookings.
- **Response:** System displays the current status, including departure/arrival times, terminal, and gate information.

4.6 Baggage Details

4.6.1 Description and Priority

Description: Users can view baggage allowance based on their ticket class and route.

Priority: Medium

4.6.2 Stimulus/Response Sequences

- **Stimulus:** User navigates to the baggage details section or views information within a booking summary.

- **Response:** System displays standard and personal baggage limits, along with warnings for extra charges.

4.7 Weather and Emergency Alerts

4.7.1 **Description and Priority**

Description: Shows weather conditions and emergency alerts for departure/arrival airports.

Priority: Medium

4.7.2 **Stimulus/Response Sequences**

- **Stimulus:** System fetches weather data for upcoming flights or detects operational issues.
- **Response:** System displays current weather, potential delays, or emergency notifications for affected airports to relevant users.

4.8 Staff Management

4.8.1 **Description and Priority**

Description: Admins can manage staff records including pilots, crew, and ground staff.

Priority: High

4.8.2 **Stimulus/Response Sequences**

- **Stimulus:** Admin accesses the staff management panel and performs an add, update, or delete action on a staff profile.
- **Response:** System updates the staff database and displays confirmation of the change.

4.9 Flight Management

4.9.1 **Description and Priority**

Description: Admins manage flight information such as schedules, crew, and aircraft.

Priority: High

4.9.2 **Stimulus/Response Sequences**

- **Stimulus:** Admin accesses the flight management panel and performs an add, update, or cancel flight action.
- **Response:** System updates flight records, and changes are reflected in user-facing modules like booking and flight status.

4.10 Online Check-In

4.10.1 **Description and Priority**

Description: Users can check in online, select seats, and get their boarding pass.

Priority: Medium

4.10.2 **Stimulus/Response Sequences**

- **Stimulus:** User enters booking reference and selects a seat on the online check-in page.
- **Response:** System verifies booking, assigns the seat, and generates a downloadable e-boarding pass.

LAB # 09

4.1.3 Functional Requirements

- **REQ-1**: User Registration
- **REQ-2**: User Login
- **REQ-3**: Flight Booking
- **REQ-4**: Payment
- **REQ-5**: Flight Status
- **REQ-6**: Baggage Details
- **REQ-7**: Weather and Emergency Alerts
- **REQ-8**: Staff Management
- **REQ-9**: Flight Management
- **REQ-10**: Online Check-In

Ambiguous Requirements (TBD)

These features are non-essential for initial system operation but can be considered for future upgrades.

TBD-1: Multi-language Support

- The system currently operates in English only. Future versions may allow users to select their preferred language (e.g., Urdu, Arabic, etc.) for a better user experience.

TBD-2: Travel History and Saved Preferences

- A future version may allow users to view past bookings, save preferred destinations, and frequent flyer details for quicker booking.

TBD-3: Notification Customization

- Users may later be able to choose how they receive alerts (SMS, email, app notifications) and what type of notifications they want to get.

LAB # 10

Other Nonfunctional Requirements

Performance Requirements

- **Requirement ID:** NFR-002
- **Description:** The system should respond quickly and perform efficiently under normal and peak usage.

- FMS should respond quickly during normal and peak usage.
- Search and booking must complete within seconds.
- System must handle multiple users without slowing down.

Safety Requirements

- **Requirement ID:** NFR-001
- **Description:** The system must protect user data and payment information using encryption and secure authentication.

- **Requirement ID:** NFR-003
- **Description:** The system should be accessible and functional most of the time without unexpected downtime.

- Sensitive data is encrypted in storage and transmission.
- User passwords are hashed for secure storage.
- Login attempts are limited to block brute-force attacks.
- Payment system complies with security standards (e.g., PCI-DSS).
- System maintains 99.5% uptime to avoid unexpected downtime.
- Recovery from failures occurs within 15 minutes.
- Users are notified 24 hours before scheduled maintenance.

Security Requirements

- **Requirement ID:** NFR-001
- **Description:** The system must protect user data and payment information using encryption and secure authentication.

- Encrypt data at rest and in transit.
- Secure login with password hashing.
- Limit login attempts to stop brute-force attacks.
- Admin access restricted; payments follow PCI-DSS standards.

LAB # 11

Software Quality Attributes

- Reliable and easy to use on all devices.
- Modular design for easier updates.
- Fast response even under heavy use.
- Scalable to support growing users and data.
- Data protection through encryption and secure login.

Business Rules

- Admins manage flights/staff; customers book and check-in.
- Limited ticket booking per user.

- Bookings confirm after payment; fare hold available briefly.
- Check-in allowed within set time before flight.
- Real-time flight updates shown.
- User data remains private and secure.

Other Requirements

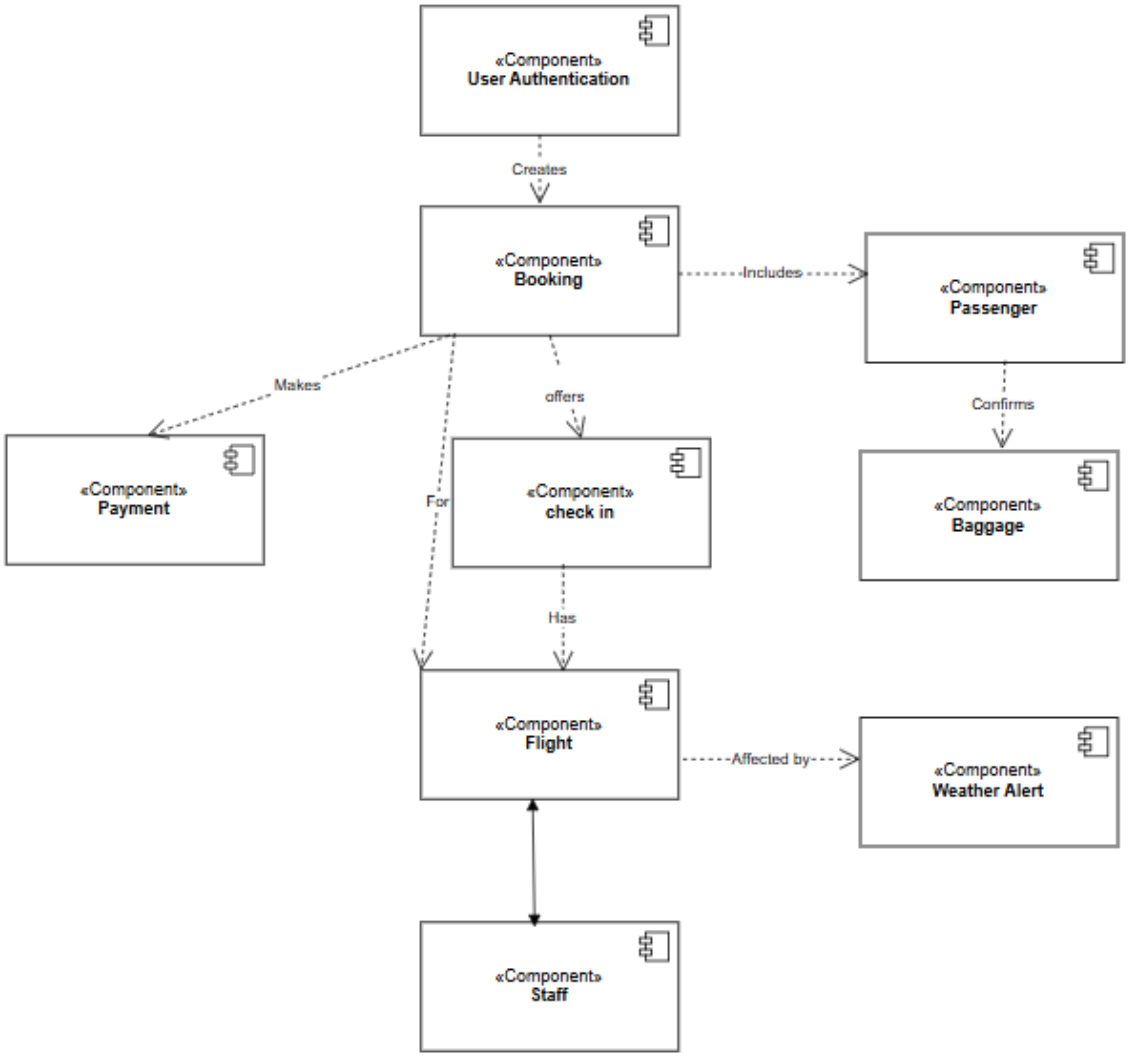
- No extra requirements at this stage.
- Future updates may introduce new needs.

Appendix A: Glossary

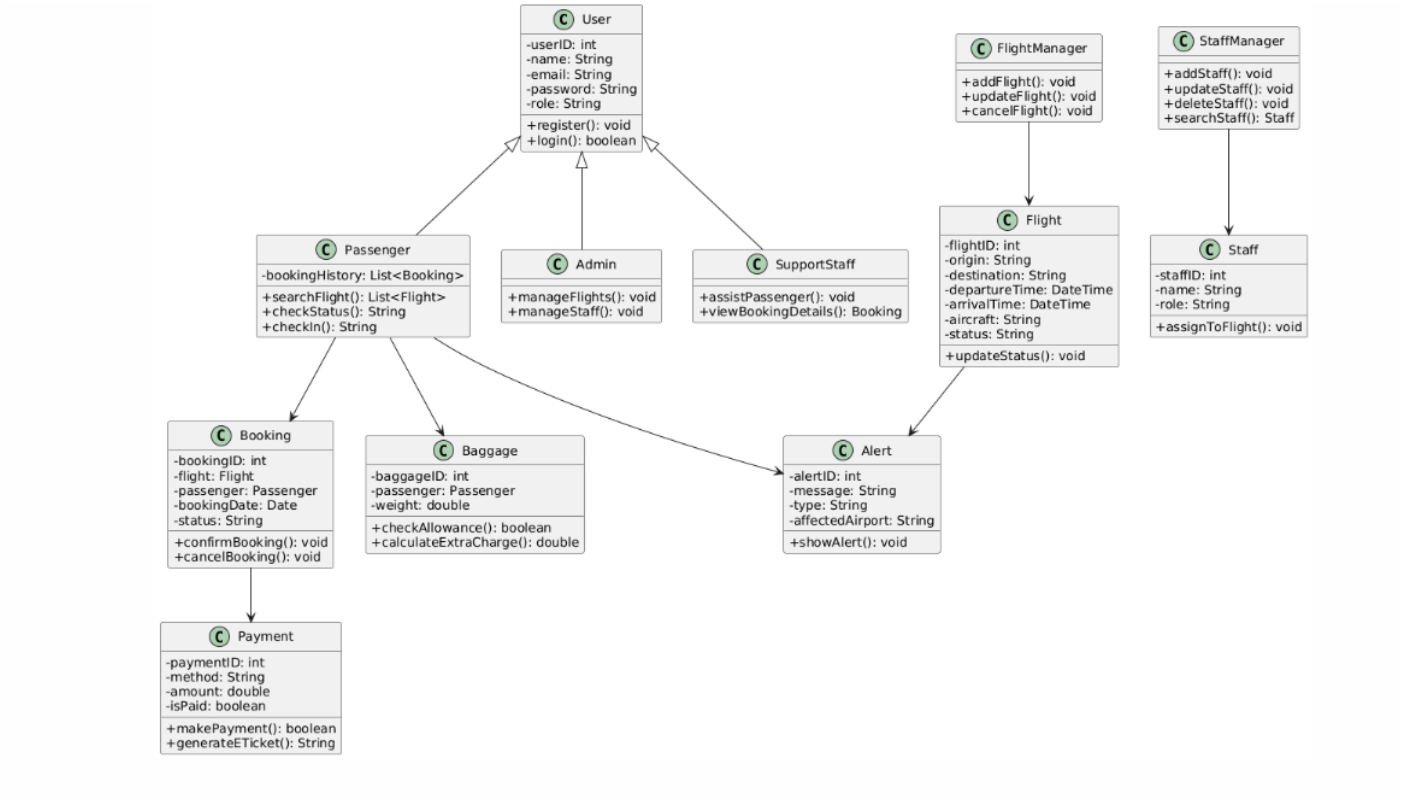
- **API (Application Programming Interface):** A set of rules and protocols for building and interacting with software applications.
- **FMS (Flight Management System):** The software system described in this document.
- **HTTPS (Hypertext Transfer Protocol Secure):** A secure version of HTTP, used for secure communication over a computer network.
- **JSON (JavaScript Object Notation):** A lightweight data-interchange format.
- **PCI-DSS (Payment Card Industry Data Security Standard):** A set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.
- **REST API (Representational State Transfer API):** An architectural style for designing networked applications.
- **SRS (Software Requirements Specification):** A document that describes the requirements for a software system.
- **SSL/TLS (Secure Sockets Layer/Transport Layer Security):** Cryptographic protocols designed to provide communications security over a computer network.
- **UAT (User Acceptance Testing):** The final stage of software testing where real users test the software to ensure it can handle required tasks in real-world scenarios.

Appendix B: Analysis Models

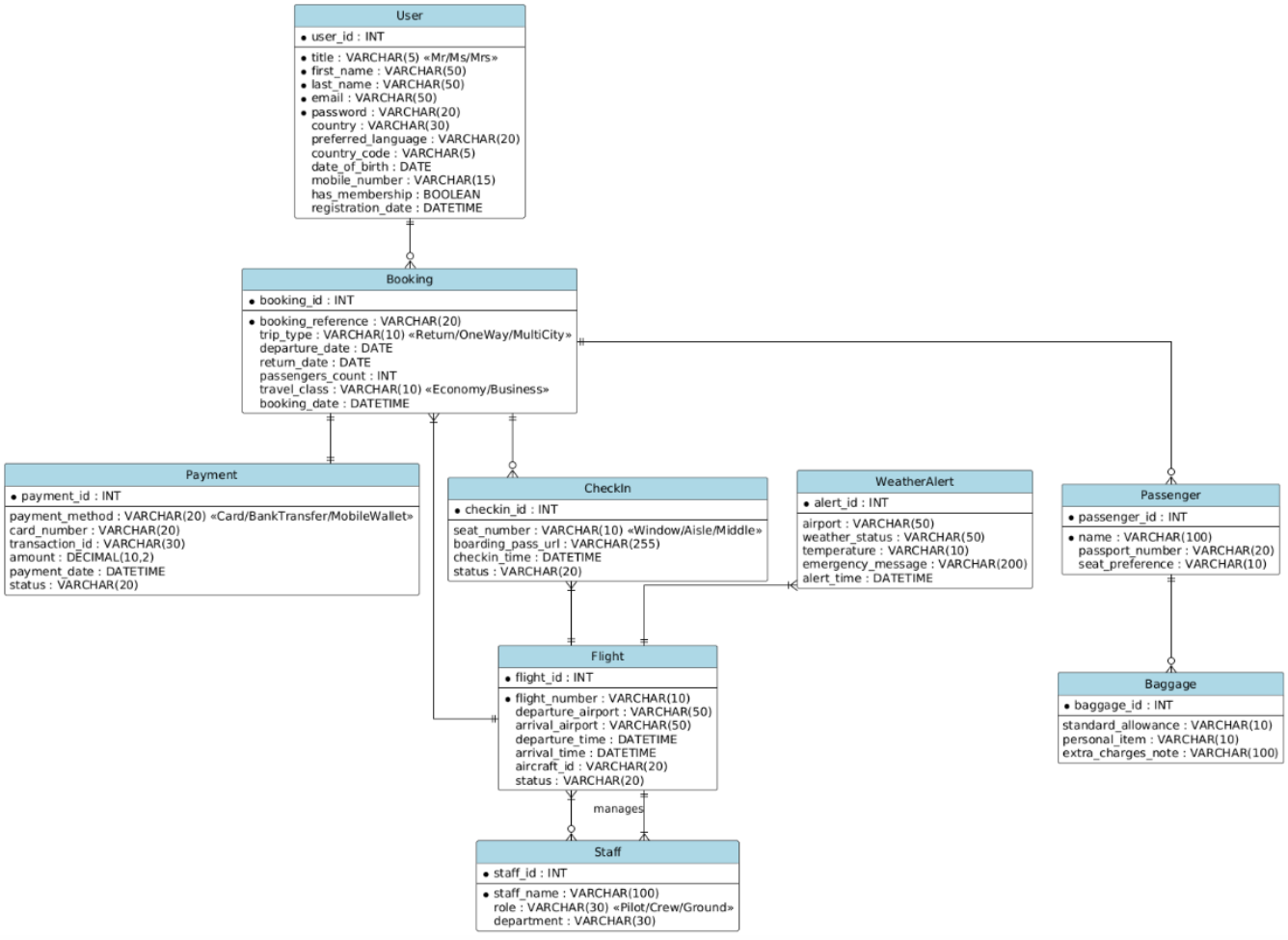
COMPONENT DIAGRAM



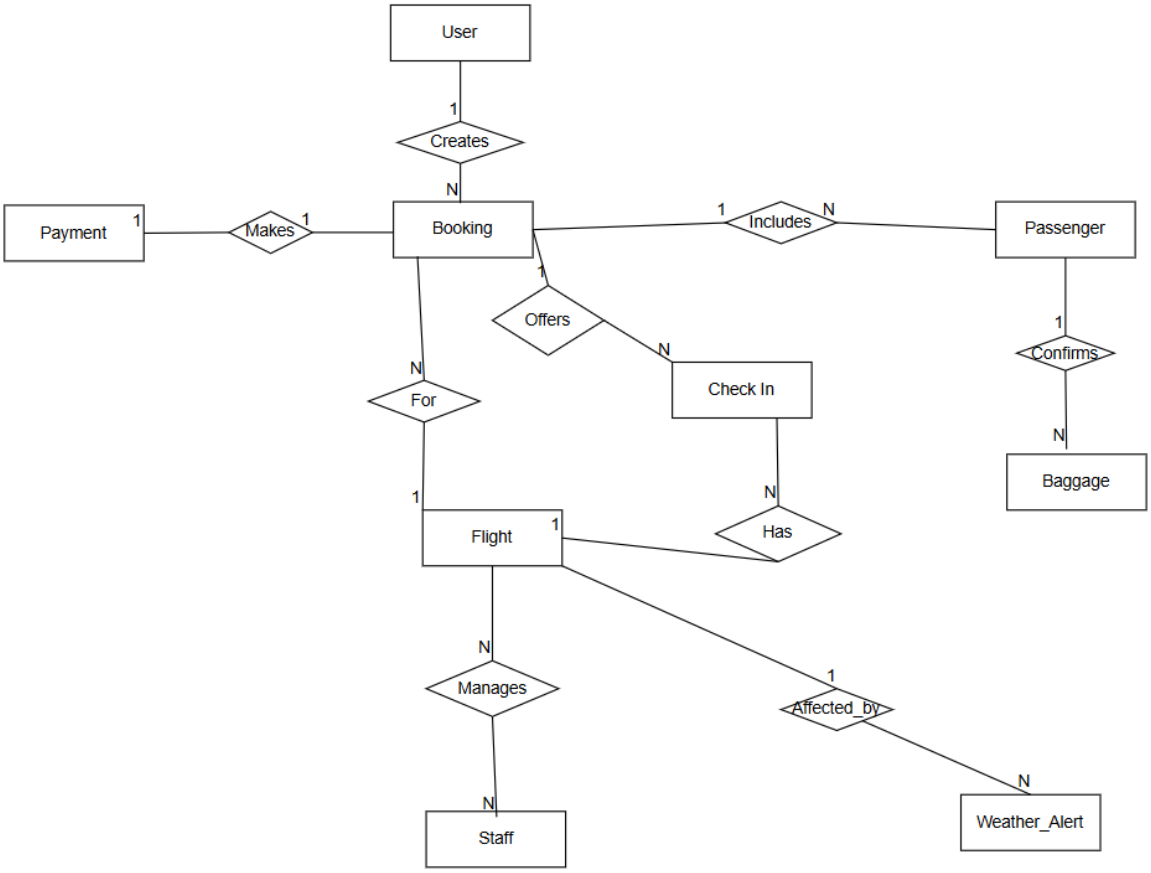
UML CLASS DIAGRAM


User
-userID: int
-name: String
-email: String
-password: String
-role: String
+register(): void
+login(): boolean
FlightManager
+addFlight(): void
+updateFlight(): void
+cancelFlight(): void
StaffManager
+addStaff(): void
+updateStaff(): void
+deleteStaff(): void
+searchStaff(): Staff
Passenger
-bookingHistory: List<Booking>
+searchFlight(): List<Flight>
+checkStatus(): String
+checkIn(): String
Admin
+manageFlights(): void
+manageStaff(): void
SupportStaff
+assistPassenger(): void
+viewBookingDetails(): Booking
Flight
-flightID: int
-origin: String
-destination: String
-departureTime: DateTime
-arrivalTime: DateTime
-aircraft: String
-status: String
+updateStatus(): void
Staff
-staffID: int
-name: String
-role: String
+assignToFlight(): void
Booking
-bookingID: int
-flight: Flight
-passenger: Passenger
-bookingDate: Date
-status: String
+confirmBooking(): void
+cancelBooking(): void
Baggage
-baggageID: int
-passenger: Passenger
-weight: double
+checkAllowance(): boolean
+calculateExtraCharge(): double
Alert
-alertID: int
-message: String
-type: String
-affectedAirport: String
+showAlert(): void
Payment
-paymentID: int
-method: String
-amount: double
-isPaid: boolean
+makePayment(): boolean
+generateETicket(): String

ERD MODEL


User
• user_id : INT
• title : VARCHAR(5) «Mr/Ms/Mrs»
• first_name : VARCHAR(50)
• last_name : VARCHAR(50)
• email : VARCHAR(50)
• password : VARCHAR(20)
country : VARCHAR(30)
preferred_language : VARCHAR(20)
country_code : VARCHAR(5)
date_of_birth : DATE
mobile_number : VARCHAR(15)
has_membership : BOOLEAN
registration_date : DATETIME
Booking
• booking_id : INT
• booking_reference : VARCHAR(20)
trip_type : VARCHAR(10) «Return/OneWay/MultiCity»
departure_date : DATE
return_date : DATE
passengers_count : INT
travel_class : VARCHAR(10) «Economy/Business»
booking_date : DATETIME
Payment
• payment_id : INT
payment_method : VARCHAR(20) «Card/BankTransfer/MobileWallet»
card_number : VARCHAR(20)
transaction_id : VARCHAR(30)
amount : DECIMAL(10,2)
payment_date : DATETIME
status : VARCHAR(20)
CheckIn
• checkin_id : INT
seat_number : VARCHAR(10) «Window/Aisle/Middle»
boarding_pass_url : VARCHAR(255)
checkin_time : DATETIME
status : VARCHAR(20)
WeatherAlert
• alert_id : INT
airport : VARCHAR(50)
weather_status : VARCHAR(50)
temperature : VARCHAR(10)
emergency_message : VARCHAR(200)
alert_time : DATETIME
Passenger
• passenger_id : INT
• name : VARCHAR(100)
passport_number : VARCHAR(20)
seat_preference : VARCHAR(10)
Flight
• flight_id : INT
• flight_number : VARCHAR(10)
departure_airport : VARCHAR(50)
arrival_airport : VARCHAR(50)
departure_time : DATETIME
arrival_time : DATETIME
aircraft_id : VARCHAR(20)
status : VARCHAR(20)
Baggage
• baggage_id : INT
standard_allowance : VARCHAR(10)
personal_item : VARCHAR(10)
extra_charges_note : VARCHAR(100)
manages
Staff
• staff_id : INT
• staff_name : VARCHAR(100)
role : VARCHAR(30) «Pilot/Crew/Ground»
department : VARCHAR(30)

ERD DIAGRAM


User
1
Creates
N
Payment
1
Makes
1
Booking
1
Includes
N
Passenger
1
Offers
N
Check In
1
Confirms
N
For
N
Baggage
1
Flight
1
N
Has
N
Manages
1
Affected_by
N
N
Staff
Weather_Alert

Appendix C: To Be Determined List

- **TBD-1: Multi-language Support:** The system currently operates in English only. Future versions may allow users to select their preferred language (e.g., Urdu, Arabic, etc.) for a better user experience.
- **TBD-2: Travel History and Saved Preferences:** A future version may allow users to view past bookings, save preferred destinations, and frequent flyer details for quicker booking.

- **TBD-3: Notification Customization:** Users may later be able to choose how they receive alerts (SMS, email, app notifications) and what type of notifications they want to get.