Software Requirements Specification (SRS) for Online Flight Management System

Student Name: Chanda Lungu
Student ID: SE2023001
College: Lusaka Institute of Technology (LIT)
Date: June 20, 2025

1. Introduction

1.1 Purpose and Scope

This document outlines the software requirements for an Online Flight Management System (OFMS). The primary purpose of the OFMS is to provide a comprehensive, user-friendly platform for managing flight operations, including flight search, booking, payment, and administrative tasks. The scope covers the functionalities accessible to both registered passengers and airline administrative staff. It aims to streamline the process of flight reservation and information dissemination, ultimately enhancing efficiency for the airline and convenience for its customers.

1.2 Intended Audience and Document Conventions

This SRS is intended for developers, testers, project managers, and key stakeholders involved in the OFMS project. Key terms and acronyms will be defined as they appear or in an appended glossary (not included in this submission). All requirements are written in a clear, unambiguous language.

1.3 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

2. Overall Description

2.1 Product Perspective

The OFMS is a standalone web-based application, but it will integrate with existing external systems such as payment gateways and potentially airline operational databases for real-time flight data. It's designed to reduce manual overhead and improve data accuracy compared to traditional methods.

2.2 User Characteristics

The system will primarily serve two main user groups:

- **Passengers:** Individuals seeking to search for, book, and manage flight

reservations. They are expected to have basic computer literacy and internet access.
- **Airline Administrative Staff:** Employees responsible for managing flights, routes, passenger data, and system configurations. They require appropriate access levels and will be trained on the system's administrative modules.

2.3 General Constraints

- **Operating Environment:** The system must be deployable on a cloud-based server environment (e.g., AWS, Azure) and be accessible via modern web browsers (Chrome, Firefox, Edge, Safari).
- **Design Constraints:** Prioritize a modular architecture for ease of maintenance and future scalability. Adherence to web security best practices is crucial.
- **Legal/Regulatory:** Must comply with relevant civil aviation regulations and data protection laws (e.g., GDPR-like principles for passenger data).

2.4 Assumptions and Dependencies

- **Assumptions:** Users have reliable internet access. External payment gateways will be available and provide necessary APIs.
- **Dependencies:** Availability of a robust database management system. Cooperation from third-party API providers (e.g., for flight data or payment processing).

3. System Features

3.1 User Authentication and Profile Management

- **SR1.1:** The system shall allow new users (passengers) to register by providing a unique email address, password, and personal details (name, contact info).
- **SR1.2:** The system shall allow registered users to log in using their email and password.
- **SR1.3:** The system shall allow logged-in users to view and update their profile information.
- **SR1.4:** The system shall allow users to reset forgotten passwords via their registered email.

3.2 Flight Search and Information Display

- **SR2.1:** The system shall allow users to search for flights based on:
  - Departure and destination airports.
  - Departure and return dates (for round-trip).
  - Number of passengers (adults, children, infants).
- **SR2.2:** The system shall display available flights with details including airline, flight

number, departure/arrival times, duration, number of stops, and price.
- **SR2.3:** The system shall allow users to filter search results by criteria such as price range, airline, number of stops, and departure/arrival times.

3.3 Booking and Reservation

- **SR3.1:** The system shall allow logged-in users to select a flight from search results.
- **SR3.2:** The system shall prompt the user to enter passenger details for each traveler.
- **SR3.3:** The system shall allow users to review booking details before confirmation.
- **SR3.4:** The system shall generate a unique booking reference number upon successful reservation.

3.4 Payment Processing

- **SR4.1:** The system shall integrate with at least one major online payment gateway (e.g., Visa/MasterCard through a third-party processor) for secure transactions.
- **SR4.2:** The system shall process payments and confirm the booking status upon successful payment.
- **SR4.3:** The system shall send an electronic ticket (e-ticket) and booking confirmation to the user's registered email address after successful payment.

3.5 Administrative Functions (for Airline Staff)

- **SR5.1:** The system shall allow authorized administrative staff to add, modify, or delete flight schedules.
- **SR5.2:** The system shall allow administrative staff to manage passenger bookings (e.g., view, modify, cancel bookings).
- **SR5.3:** The system shall provide reporting features for flight occupancy and revenue generation.

4. External Interface Requirements

4.1 User Interfaces

- The system shall provide an intuitive, web-based graphical user interface (GUI) accessible via standard web browsers.
- The UI shall be responsive, adapting to various screen sizes (desktop, tablet, mobile).
- Common UI elements like navigation menus, search forms, and booking forms will be consistently designed.

4.2 Software Interfaces

- **Payment Gateway API:** Integration with a secure third-party payment processor

API for handling credit/debit card transactions.
- **Airline Data API (Optional but desirable for real-time data):** Interface with the airline's internal flight database or a Global Distribution System (GDS) for real-time flight availability and pricing.
- **Email Service API:** Integration with an email service provider for sending confirmations and notifications.

4.3 Communication Interfaces

- The system shall communicate over HTTPS for all secure transactions and data transfer to ensure data integrity and confidentiality.

5. Non-Functional Requirements

5.1 Performance

- **NFR5.1.1 (Response Time):** Flight search results shall be displayed within 3 seconds for 90% of requests under normal load.
- **NFR5.1.2 (Throughput):** The system shall support at least 50 concurrent users performing flight searches and bookings without significant degradation in performance.

5.2 Security

- **NFR5.2.1 (Authentication):** User passwords shall be securely hashed and salted before storage.
- **NFR5.2.2 (Authorization):** Access to administrative functions shall be restricted based on user roles and permissions.
- **NFR5.2.3 (Data Encryption):** All sensitive data (e.g., payment information, personal details) transmitted between the client and server shall be encrypted using TLS/SSL.
- **NFR5.2.4 (Session Management):** User sessions shall be securely managed to prevent session hijacking.

5.3 Reliability

- **NFR5.3.1 (Availability):** The system shall be available 99.5% of the time, excluding scheduled maintenance windows.
- **NFR5.3.2 (Data Integrity):** The system shall ensure data consistency and integrity, especially for booking and payment records. Mechanisms like transaction logging and database backups will be in place.

5.4 Usability

- **NFR5.4.1 (Ease of Use):** The system shall be intuitive and easy to navigate for users with basic internet proficiency, allowing flight bookings to be completed

within a maximum of 5 steps from search.
- **NFR5.4.2 (Error Handling):** The system shall provide clear and helpful error messages for invalid inputs or system failures.

5.5 Scalability

- **NFR5.5.1:** The system architecture shall be designed to allow for horizontal scaling to accommodate a growing number of users and flight data.

5.6 Maintainability

- **NFR5.6.1:** The codebase shall be well-documented and follow established coding standards to facilitate future modifications and bug fixes.
- **NFR5.6.2:** The system shall support easy updates and deployments with minimal downtime.