# **Software Requirements Specification (SRS) for Online Flight Management System Project**

Student Name: Peter Nkole

Student ID: SE2023006

College: Lusaka Institute of Technology (LIT)

Date: June 20, 2025

## **1. Introduction – Getting Started**

This document is basically my attempt to lay out exactly what we need this online flight system to do. We're calling it the "Zambian Airlink Portal" (ZAP). The main goal is to build a web platform where people can easily find flights, make bookings, and pay for their tickets without all the hassle. It also needs to have some tools for the airline’s own staff to manage things behind the scenes, making their job smoother too. It’s all about bringing flight booking into the modern age for Zambia.

## **2. Who Uses It and How It Works**

### **2.1 Our Users**

There are two main groups who will interact with the ZAP:

* **The Passengers:** These are ordinary people who want to travel. They’ll be using the public-facing part of the website to search for flights, book their seats, and handle their personal booking details. They’ll likely be using a mix of smartphones, tablets, and desktop computers.
* **The Airline Administration Team:** This includes anyone from ground staff managing bookings to operations personnel updating flight schedules. They’ll need special secure access to a different part of the system to carry out their duties. Their level of access will vary depending on their role.

### **2.2 Where the System Lives**

The ZAP will be a web application, meaning you access it through any standard internet browser (like Chrome, Firefox, Safari, Edge). It must display correctly and function well on all device types, from large desktop monitors to smaller mobile phone screens. For reliability, it will be hosted on a cloud server, ensuring it’s almost always available.

### **2.3 What We’re Banking On (Assumptions) and What We Can’t Ignore (Constraints)**

We're going into this assuming that users will have a stable internet connection. Crucially, we also need to assume that the various online payment services (credit/debit cards, maybe even some local mobile money solutions) will integrate smoothly with our system. A robust and secure database is a must-have to store all the flight and customer data reliably.

The biggest constraint, hands down, is **security**. We’re dealing with sensitive personal and financial information, so the system *has* to be extremely secure. This means encryption for everything. Also, we must stick to all the relevant aviation regulations and data privacy laws here in Zambia and internationally.

## **3. The Features – What It Does**

### **3.1 Account Management for Users**

* **F1.1:** New users should have an easy way to sign up. This means providing an email address, setting a strong password, and giving basic contact information (name, phone number).
* **F1.2:** Existing users will log in securely with their email and password.
* **F1.3:** Once logged in, users should be able to update their personal details in their profile.
* **F1.4:** If someone forgets their password, there needs to be a straightforward and secure process to reset it via their registered email.

### **3.2 Finding and Displaying Flights**

* **F2.1:** Users need to search for flights by inputting their departure and arrival locations, selecting their travel dates (for one-way or round-trip journeys), and specifying the number of passengers.
* **F2.2:** Search results must clearly show essential flight details: the airline name, flight number, exact departure and arrival times, the total duration of the flight, any layover information, and the ticket price.
* **F2.3:** Filters are essential. Users should be able to narrow down results by criteria such as price range, specific airlines, direct flights only, or preferred times of day.

### **3.3 The Booking Process**

* **F3.1:** Users will select their preferred flight(s) from the search results to initiate the booking.
* **F3.2:** The system will guide the user to input comprehensive details for each passenger on the booking (e.g., full name, date of birth, and passport details if required for international travel).
* **F3.3:** Before the final payment, a clear summary of the selected flights, passenger information, and total cost must be presented for review and confirmation.
* **F3.4:** Upon successful completion of the booking, a unique booking reference number will be generated and displayed.

### **3.4 Handling Payments**

* **F4.1:** The system will integrate with a recognized and secure online payment gateway (e.g., for Visa/MasterCard transactions, and possibly local mobile money options in Zambia). This connection must be fully encrypted.
* **F4.2:** Once payment is successfully processed, the booking status needs to be immediately updated to "Confirmed."
* **F4.3:** An electronic ticket (e-ticket) and a detailed booking confirmation will be automatically sent to the user's registered email address after payment clears.

### **3.5 Tools for Airline Staff (Admin Functions)**

* **F5.1:** Authorized administrative staff will have the ability to add new flights to the system, modify details of existing flights (e.g., times, capacities), or completely remove flights.
* **F5.2:** Administrators should be able to search for, view, make changes to, or cancel passenger bookings as needed.
* **F5.3:** The system must generate various reports for administrative purposes, such as flight occupancy rates, revenue summaries, and passenger manifests for specific flights.

## **4. How It Connects to Everything Else (Interfaces)**

### **4.1 The User's View (User Interface)**

The ZAP will have an intuitive, web-based graphical user interface (GUI). It needs to be designed to adapt seamlessly across different screen sizes (desktop, tablet, mobile) and be visually appealing while remaining functional and easy to navigate.

### **4.2 Talking to Other Software (Software Interfaces)**

* **Payment Gateway API:** This is critical for all financial transactions, ensuring secure communication with payment processors.
* **Email Service API:** For automated sending of booking confirmations, e-tickets, and password reset links.
* **Airline Internal Systems (Potential Future):** There's a potential need to integrate with the airline's existing operational databases for real-time flight data and inventory management.

### **4.3 Communication Methods (Security!)**

All data communication between the user's browser and our servers, especially any sensitive personal or payment information, must be secured using HTTPS protocols to prevent interception.

## **5. The "Behind-the-Scenes" Must-Haves (Non-Functional Requirements)**

### **5.1 Performance – How Fast It Is**

* **NFR5.1.1 (Search Speed):** When a user searches for a flight, the results should pop up within 2.8 seconds for at least 90% of requests, even when the system is busy.
* **NFR5.1.2 (Capacity):** The system must be able to handle at least 65 simultaneous users making searches or bookings without slowing down noticeably.

### **5.2 Security – Keeping Things Safe**

* **NFR5.2.1 (Passwords):** All user passwords will be stored in a highly secure, encrypted format (hashed and salted).
* **NFR5.2.2 (Access Control):** Access to any administrative features will be strictly controlled based on user roles and permissions, ensuring only authorized personnel can perform specific actions.
* **NFR5.2.3 (Data Encryption):** All sensitive data, whether moving across the network (e.g., credit card details during payment) or stored in the database, must be encrypted.
* **NFR5.2.4 (Session Security):** User sessions will be managed securely to prevent unauthorized access or session hijacking.

### **5.3 Reliability – Always There When You Need It**

* **NFR5.3.1 (Uptime):** The system should be operational 99.5% of the time, not counting planned maintenance. This means it rarely goes down unexpectedly.
* **NFR5.3.2 (Data Integrity):** The system will ensure that all booking and payment data remains consistent and accurate. Robust backup procedures will be in place to prevent data loss.

### **5.4 Usability – Easy to Figure Out**

* **NFR5.4.1 (Ease of Booking):** A first-time user should be able to complete a flight booking (from search to confirmation) in no more than 7 steps.
* **NFR5.4.2 (Error Messages):** If something goes wrong, the system will provide clear, helpful, and polite error messages that guide the user on how to correct the issue.

### **5.5 Scalability – Growing with Demand**

The architecture of the ZAP should allow it to easily expand its capacity to handle a significant increase in users, flights, and data volume in the future without needing a complete redesign.

### **5.6 Maintainability – Easy to Manage and Update**

* **NFR5.6.1 (Code Quality):** The system's code will be well-documented, follow established programming standards, and be modular. This makes it easier for new developers to understand, maintain, and extend the system.
* **NFR5.6.2 (Deployment):** Updates and new features should be deployable quickly and smoothly, minimizing any disruption to service.