# **Software Requirements Specification (SRS) - Online Flight Ticketing System**

Student Name: Mwape Chanda

Student ID: SE2023005

College: Lusaka Institute of Technology (LIT)

Date: June 20, 2025

## **1. Introduction: The Big Idea**

Alright, this document is all about what we need for our online flight booking system. Think of it as the blueprint. We're building something that lets people find flights, book seats, and pay for their tickets, all online, nice and easy. It's also gonna have some behind-the-scenes stuff for the airline staff to manage flights and bookings. The goal is to make flying less of a headache for everyone involved.

## **2. Who's This For? (And the Environment)**

### **2.1 The Users**

Basically, two main types of folks will be using this:

* **The Travelers (Passengers):** These are the people buying tickets. They'll need to search, view flight info, make reservations, and access their booking details. They'll mostly be using it on their phones or home computers.
* **The Airline Crew (Admin Staff):** These are the people working for the airline. They'll need specific tools to add new flights, change existing ones, check passenger lists, and see sales figures. Their access will be controlled, of course.

### **2.2 Where It'll Live**

This system is going to be a website. So, it has to work perfectly on any modern web browser – Chrome, Firefox, Safari, Edge, you name it. And it needs to look good and be usable whether you're on a big desktop screen or a small phone. We'll host it on a reliable cloud server, so it's always up and running.

### **2.3 What We're Counting On (Assumptions & Constraints)**

We're assuming most users will have decent internet. And for payments, we're relying on external payment services to work smoothly with our system. A good, solid database is a must to keep all the flight and customer data safe and organised.

The biggest constraint is **security**. People are trusting us with their personal details and money, so we need to lock it down tight. Everything from login to payment has to be secure. And yeah, we have to follow all the local (Zambian) and international aviation and privacy rules.

## **3. What It Needs To Do (The Functions, Step-by-Step)**

### **3.1 Your Account Stuff**

* **F.1:** New users should be able to create an account. This means giving their email, making a password, and filling in their basic contact info (name, phone).
* **F.2:** If you have an account, you can log in securely.
* **F.3:** Once logged in, you can update your profile details if something changes.
* **F.4:** Forgot your password? No worries. There needs to be a way to reset it through your registered email.

### **3.2 Finding Your Flight**

* **F.5:** You can search for flights. You'll need to tell it where you're flying from, where you're going, your travel dates (one way or return), and how many people are tagging along.
* **F.6:** The system will show you all the available flights. For each one, you should see the airline, flight number, departure and arrival times, how long the trip is, how many stops, and the price.
* **F.7:** It needs filters! So you can narrow down results by price, airline, direct flights only, or specific times.

### **3.3 Booking Your Seat**

* **F.8:** Once you pick a flight, you click it to start the booking process.
* **F.9:** You'll then need to enter all the passenger details for everyone travelling on that booking.
* **F.10:** Before you hit "pay," you'll get a summary page. This is your chance to double-check everything (flights, passengers, total cost).
* **F.11:** After a successful booking, the system will give you a unique booking reference number. Keep this safe!

### **3.4 Paying Up**

* **F.12:** The system has to link up with reliable payment platforms, probably credit/debit cards (Visa/MasterCard) and maybe even local mobile money options. This connection has to be super secure.
* **F.13:** As soon as the payment goes through, your booking status needs to change to "Confirmed" immediately.
* **F.14:** An electronic ticket (e-ticket) and a booking confirmation will be sent straight to your email.

### **3.5 The Admin Panel (Airline Side)**

* **F.15:** Admin staff need to be able to add new flights, change details for existing ones (like times or seat availability), or remove flights from the system.
* **F.16:** They also need to view customer bookings, make changes (like changing names or dates), or cancel them if a customer requests it.
* **F.17:** The system should generate useful reports for the admins – stuff like how many seats are booked on a flight, total revenue, or passenger lists.

## **4. How It Interacts (Interfaces)**

### **4.1 What You See (User Interface)**

It's a website, so the design needs to be clean, modern, and easy to navigate. It must look good and work well on any screen size – from a tiny phone to a big desktop monitor. Everything should be clear and straightforward.

### **4.2 What It Talks To (Software Interfaces)**

* **Payment Gateway API:** This is critical for secure financial transactions.
* **Email Sending API:** For sending out all those confirmations and e-tickets automatically.
* **Future (maybe):** An API to link directly with the airline's internal flight operation systems for real-time data accuracy.

### **4.3 How It Communicates (Security, Again!)**

All data sent between your browser and our server, especially sensitive stuff like personal info or payment details, must be encrypted using HTTPS. No exceptions.

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

### **5.1 Speed (Performance)**

* **NFR.1:** Flight searches should show results in under 2.5 seconds, almost every time (95% of the time).
* **NFR.2:** The system needs to handle a good number of users at once – let's say 75 people actively using it – without slowing down.

### **5.2 Rock-Solid Security**

* **NFR.3:** User passwords will be securely hashed and salted in the database.
* **NFR.4:** Only authorized admin staff can get into the admin sections based on their roles.
* **NFR.5:** All private data (like credit card numbers) must be encrypted, both when it's moving across the internet and when it's sitting in our database.
* **NFR.6:** We'll have strong session management to prevent anyone from hijacking a user's session.

### **5.3 Always On (Reliability)**

* **NFR.7:** The system needs to be available 99.6% of the time. Planned maintenance is okay, but unplanned downtime should be super rare.
* **NFR.8:** We need a solid backup and recovery plan for all data, especially bookings and payments, so nothing ever gets lost.

### **5.4 Easy to Use (Usability)**

* **NFR.9:** Someone who's never used a flight booking site before should be able to complete a booking from scratch in about 6 steps or less.
* **NFR.10:** If you make a mistake, the system should give you clear, helpful messages to tell you what went wrong and how to fix it.

### **5.5 Ready for Growth (Scalability)**

The way we build this system should make it easy to add more flights, more users, and more features in the future without needing a complete overhaul. It should be able to grow with the airline.

### **5.6 Easy to Maintain (Maintainability)**

The code will be well-written, clearly commented, and organized. This means new developers can quickly understand it, and it'll be easy to fix bugs or add new functions down the line without causing new problems. Updates should be smooth and quick to deploy.