# SOFTWARE REQUIREMENTS SPECIFICATION

AIRZAMBIA.COM

2026

# 1. INTRODUCTION

## 1.1. Purpose

This document outlines a project plan for the development of the AirZambia.com Flight Booking System (Platform) by a prospect employee of YUAI Soft Solution. The System is designed to demonstrate comprehensive capabilities, utilizing Django/DRF for the backend and React for the frontend. It aims to provide a streamlined, secure, and real-time flight booking experience. I will build this system with key features including user registration with admin approval workflow, efficient flight search and booking functionalities, and real-time flight status updates. Finally, this document serves as a reference for stakeholders, developers, and testers to ensure that the system meets its core value propositions and operational goals.

## 1.2. Scope

This SRS describes a web-based Flight Booking System developed using Django/DRF for the backend and React for the frontend. The system shall provide functionality for user registration, administrative approval of users, secure authentication, flight search, booking management, and real-time flight status updates. The system shall ensure secure handling of user data and controlled access through role-based authorization. The product is intended to demonstrate full-stack web application development and does not aim to replace or integrate with real-world airline reservation or payment systems.

## 1.3. Definitions, Acronyms, and Abbreviations

**ADMIN:** Administrative User with full Access
**API**: Application Programming Interface
**Config CORS**: Cross-Origin Resource Sharing (Security feature implemented in web browsers to control how resources are shared between origins)
**CRUD Operations**: The fundamental functions to manage data in a software application through Creation, Read, Update, and Delete
**DB**: Database
**DFD:** Data Flow Diagram
**DRF**: Django Rest Framework
**GDPR:** General Data Protection Regulation
**JWT**: JSON Web Token
**PCI DSS:** Payment Card Industry Data Security Standard
**SRS:** Software Requirements Specification
**UI**: User Interface
**USER**: Customer or travel agent
**UX**: User Experience

### 1.4. Overview

This document presents the Software Requirements Specification (SRS) for the AirZambia.com Flight Booking System, a web-based platform developed as a demonstration of full-stack web application capabilities. The system is being developed using Django and Django Rest Framework (DRF) for backend services and React for the frontend user interface. It is designed to deliver a secure, efficient, and real-time flight booking experience.

It is a detailed description of the system's intended functionality, constraints, and requirements, serving as a reference for stakeholders, developers, and testers throughout the development lifecycle. It defines the system scope, overall description, functional and non-functional requirements, and external requirements to ensure clarity and consistency in implementation.

The remainder of the document is organized as follows:

Section 2: Overall Description: This section details the overall description of the product by outlining the product perspective, production functions, User classes and characteristics, operating environment, design and implementation constraints, and then assumptions and dependencies.

Section 3 – System Specific Requirements outlines the functional and non-functional requirements as well as the external interface requirements.

Section 4 – Appendices: This section lists the list of APIs, Use case diagram, DFDs.

## 2. OVERALL DESCRIPTION

### 2.1. Product Perspective

The System functions as a comprehensive web-based platform that seamlessly integrates with various external service to provide an end-to-end flight booking experience. It establishes secure connections with airline databases (AirLabs, OpenSky Network) and APIs to retrieve real-time flight information including schedules, availability, and pricing. Through these integrations, users can access up-to-date flight options across multiple airlines, ensuring accurate and current data for their travel planning.

Moreover, the platform interfaces with payment gateways to facilitate secure and reliable transactions, allowing users to compete bookings with confidence. It supports various payment methods, such as credit or debit cards, and ensures sensitive payment data is handles with strict security measures.

Furthermore, the system may connect with external travel services such as hotel booking and taxi services, offering users comprehensive travel packages and additional services. This interconnected approach aims to enhance user convenience by providing a unified interface for most of travel-related needs.

In essence, the system acts as an intermediary that consolidates data from multiple external sources, ensuring users have a seamless, real-time booking experience, while maintaining data security and operational reliability.

### *2.2. Product Functions*

- Flight Search and Filtering
- Seat Selection
- Passenger Information and Issuance
- Payment Processing
- Booking management (View, Modify, Cancel)
- Reporting and Analytics
- User Account Management

### *2.3. User Classes and Characteristics*

- Travelers: End-users booking flights with ease of use
- Travel Agents: Manage multiple bookings for clients and need advanced management features
- Airline Staff/Admin: Manage flights, schedules, and bookings, user account creation, which will be done with full administrative access.
- Payment Processors: Handle secure payment transactions.

### *2.4. Operating Environment:*

- Web browsers (Chrome, Edge)
- Mobile Support (Android, iOS)
- Backend Server with secure APIs
- Integration with airline, Hotel, taxi, and payment gateway APIs

### *2.5. Design and Implementation Constraints*

- Compliance with PCI DSS payments
- Integration with multiple airline data sources
- Scalability to handle high traffic during peak seasons

### *2.6. Assumptions and Dependencies*

- Reliable internet connectivity
- Up-to-date airline data feeds
- External payment gateway availability

## 3.0. SYSTEM SPECIFIC REQUIREMENTS

### 3.1. Functional Requirements

| ID | Description | Priority | Remarks |
|---|---|---|---|
| FR1 | Search flights based on origin, destination , date, and passenger count | High | Multiple filters supported |
| FR2 | Display available flights with details (airline, time, price) | High | Real-time data retrieval |
| FR3 | Seat Selection during booking | Medium | Seat maps visualization |
| FR4 | User registration and login | High | Profile management |
| FR5 | Book a selected flight and generate PNR | High | Confirmation email sent |
| FR6 | Secure payment processing for bookings | High | Support credit/debit cards, e-wallets |
| FR7 | View, modify, or cancel bookings | Medium | Policy compliance considerations |
| FR8 | Admin panel for managing flights, schedules, and bookings | High | Role-based access |
| FR9 | Generate reports on bookings, cancellations, revenue | Low | Data export options |

### 3.2. Non-Functional Requirements

| Category | Requirement | Description |
|---|---|---|
| Performance | System shall multiple concurrent users without degradation (estimate of 1000) | During peak booking hours |
| Security | All user data shall be encrypted and complaint with GDPR/PCI DSS<br>Authenticated user registrations with JWT | Secure login, data storage, and transactions |
| Usability | Interface shall be user-friendly with intuitive navigation | Support multiple Operating systems |
| Availability | System uptime 99% | Minimal downtown during maintenance |
| Scalability | Ability to scale during high demand seasons | Cloud-based infrastructure |

### 3.3. External Interface Requirements

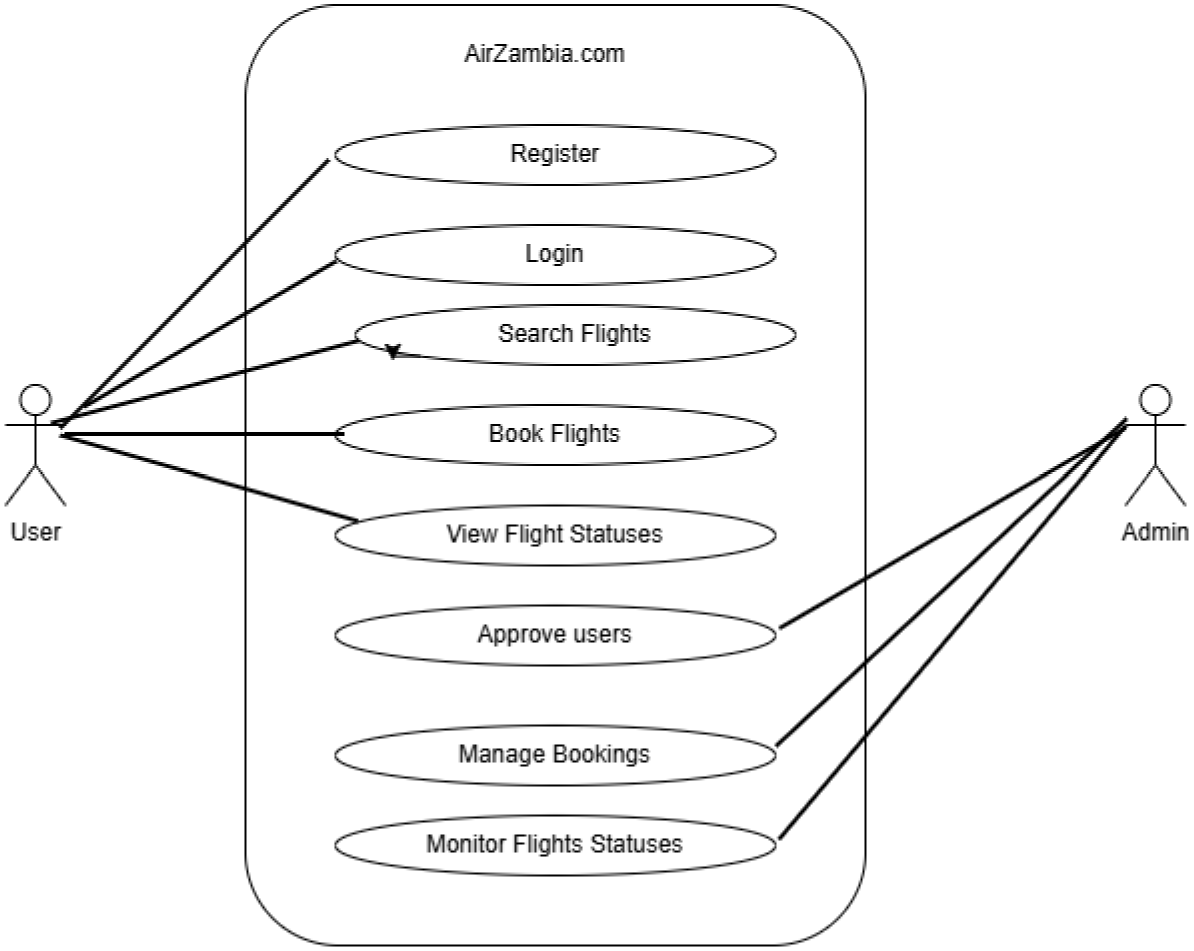
- Integration with airline APIs for flight data
- Integration with payment gateways (Stripe, Paypal, Square)
- Email notification services
- User authentication via JWT

## 4. APPENDICIES

### 4.1. List of APIs

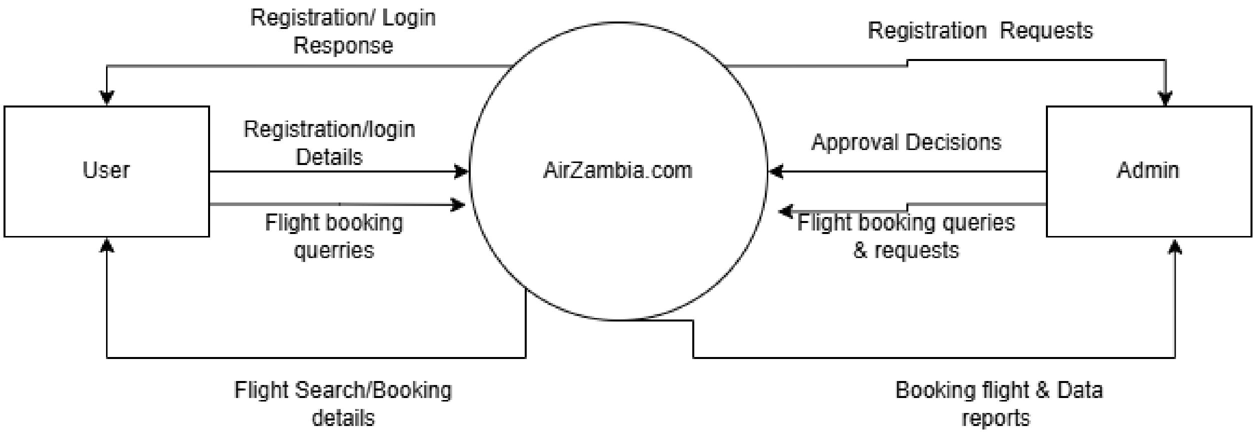
| API Type | API and Purpose | Functionality |
|---|---|---|
| External Airline APIs | Amadeus, OpenSky Network, AirLabs, & Sabre APIs - Access flight data and process booking | Search flights, book, modify, cancel flights, update flight statuses |
| Payment Gateway APIs | Handle secure financial transactions | Process payments, refunds, verify transactions |
| Authentication and Authorization APIs | JWT handlers – Manage user login and permissions | Register users, login, assign roles |
| Internal Application APIs | Enable frontend and backend communication | Fetch flight, search data, manage bookings, and user profiles |
| External Travel Services APIs | Offer additional travel services | Books hotels, taxi services |
| Notification and Messaging APIs | Communicate with Users | Send email alerts and push notifications |

### 4.2. Use Case Diagram

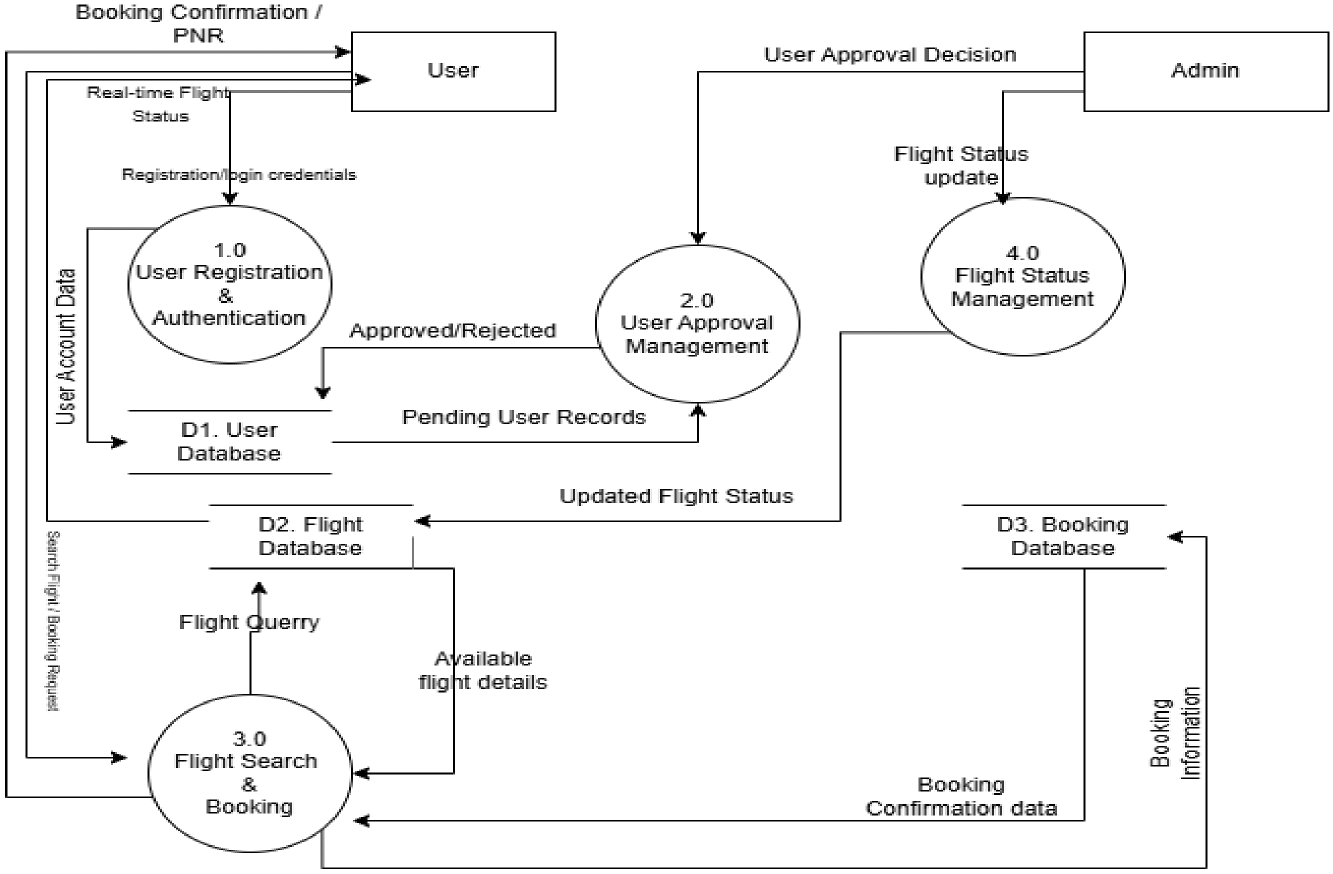


## 4.3 Data Flow Diagrams (Using the Yourdon and De Marco formats)

### 4.3.1. Contextual Level


Registration/ Login Response
Registration Requests
User
Registration/login Details
AirZambia.com
Approval Decisions
Admin
Flight booking querries
Flight booking queries & requests
Flight Search/Booking details
Booking flight & Data reports

### 4.3.2. 1st level DFD


Booking Confirmation / PNR
User
User Approval Decision
Admin
Real-time Flight Status
Registration/login credentials
Flight Status update
User Account Data
1.0 User Registration & Authentication
2.0 User Approval Management
4.0 Flight Status Management
Approved/Rejected
Pending User Records
D1. User Database
Updated Flight Status
D2. Flight Database
D3. Booking Database
Search Flight / Booking Request
Flight Querry
Available flight details
Booking Information
3.0 Flight Search & Booking
Booking Confirmation data