Software Requirements Specification
for
Airline Management system

BY:
Rishabh Kothari
Ayush Lokre
Roll no:54
Roll no:60

Table of Contents

1.Introduction

1.1 Abstract

This Software Requirements Specification (SRS) outlines the functional and non-functional requirements for an Airline Management System. It covers passenger booking, flight management, crew scheduling, administrative tasks, reporting, and feedback mechanisms. The document ensures data integrity, security, reliability, and maintainability using modern technologies. This SRS serves as a guide for developing a scalable and efficient airline management solution aligned with industry standards.

1.2. Purpose

The purpose of this Software Requirements Documentation is to provide high-level and detailed descriptions of the Airline Management Web Site. This Software Requirements Documentation will provide quantifiable requirements of the web site for use by the designer and the users of the Airline Management Website.

1.3. Scope

The Airline Management System shall provide airline companies the ability to efficiently manage their flight schedules, reservations, passenger information, and operational data. It will allow passengers to search for flights, make bookings, and manage their travel details in a user-friendly manner.

The system will be designed to cater to the specific needs of airline companies, offering comprehensive administrative rights to manage their flights, staff, and passenger data. It will support multiple airlines and be scalable to accommodate the requirements of various airline businesses in the aviation industry.

1.4. Overview

The main purpose of the Airline Management System is to provide a comprehensive digital platform for airline companies to manage their flight operations and passenger services effectively. The system will empower airline administrators to oversee and control various aspects of their operations, including flight scheduling, reservations, passenger information, and reporting.

The system will also incorporate modern communication channels, allowing passengers to interact with the airline via social media platforms such as Instagram or Facebook. Additionally, it will offer detailed and specific search capabilities, enabling users to find flights based on criteria such as date, destination, airline, and fare class.

Overall, the Airline Management System aims to streamline airline operations, enhance passenger experiences, and provide a robust platform for efficient management and growth in the aviation industry.

2. Overall Description

2.1. Product Perspective

The Airline Management System will be a web-based application designed to serve two primary user groups: airline administrators and passengers. The system will utilize Python as the scripting language and MySQL as the relational database management system (RDBMS). It will be hosted on a web server to ensure stability, security, and scalability.

2.1.1. User Interfaces

The Airline Management System will feature user interfaces tailored for both passengers and airline administrators.

Passenger Interface:

Homepage: The homepage will display featured flights, quick search options, and links to essential functions like flight booking, reservation management, and contact information.
Flight Search: Passengers can perform detailed flight searches based on criteria such as date, destination, class, and price range. The search results will be displayed in a user-friendly format.

Booking Process: The booking interface will guide passengers through the flight selection, seat reservation, and payment process. It will include validation checks for accurate data entry and provide appropriate messages for any errors.

User Profile: Passengers can create and manage their profiles, view booking history, and update personal information securely.

Admin Dashboard: The administrator dashboard will provide a comprehensive overview of flight schedules, reservations, passenger data, and system analytics. It will include quick access to essential management tasks.

Flight Management: Administrators can add, update, or cancel flights, assign crew members, and manage aircraft availability through a dedicated interface.

Reservation Management: The system will offer tools for managing passenger reservations, seat allocations, ticketing, and payment processing.

Reporting and Analytics: The admin interface will include reporting tools to generate insights on flight performance, passenger demographics, revenue analysis, and other key metrics.
User Management: Administrators can manage staff accounts, roles, permissions, and access controls securely.

2.1.2. Software Interfaces

Hosting Service:
Source: http://www.localhost.com
The hosting service provided by http://www.localhost.com will host the MySQL relational database and the Streamlit web application developed in Python for the Airline Management System.
The hosting service will meet the necessary software requirements to run MySQL and Python code effectively.

MySQL Database:
Version: MySQL 5.7 or higher
The MySQL relational database will serve as the backend for storing and managing data related to flights, reservations, passenger information, and administrative tasks in the Airline Management System.
Python's MySQL Connector library will be used to interact with the MySQL database, allowing for efficient querying, data retrieval, and updates.

Streamlit Web Application:
The Streamlit library in Python will be utilized to develop the user interfaces (UI) for the Airline Management System.
Streamlit provides a straightforward way to create interactive web applications with Python, enabling the display of dynamic data, forms, and visualizations seamlessly.
The web application will generate HTML and CSS elements dynamically based on user inputs and backend data retrieved from the MySQL database.
Streamlit's capabilities will be leveraged to create a user-friendly interface for both passengers and airline administrators, allowing for tasks such as flight search, booking management, reporting, and administrative controls.
By using Python with Streamlit for the web application and MySQL as the backend database, the Airline Management System will benefit from a modern, efficient, and scalable architecture for managing airline operations and passenger services.

2.1.3. Communications Interfaces

Chrome: Specification Number: 3.6.13 Source: http://www.chrome.com/ chromeis a free and open source web browser that will interpret the XHTML markup the PHP parser produces andapply the necessary styles as defined in the various CSS pages to create the overall look and feel of the web site.
Internet Explorer: Specification Number: 8.0 Source: http://www.microsoft.com Internet Explorer is a web browser that will interpret theXHTML markup the PHP parser produces and apply the necessary styles as defined in the various CSS pages to create the overall look and feel of the web site

2.2. Product Functions

The Airline Management System is designed to provide efficient functionality for airline administrators and passengers alike, ensuring a user-friendly experience with standardized web flow and intuitive design. Passengers can easily search for and book flights, manage their profiles, check real-time flight statuses, and access customer support, while administrators can efficiently manage flights, reservations, reporting, staff, and system configurations. The user interface is designed for easy navigation, intuitive interaction, responsiveness across devices, and a guided user experience that accommodates both novice and experienced users, making it a seamless platform for airline operations and passenger services.

3. Specific Requirements

3.1. Functional Requirements

Admin Control Panel Login:
Verify administrator's credentials against the database for login access to the control panel.

Password Reminder:
Verify administrator's information for sending login credentials via email.

Registration:
Insert user-provided registration information into the database.

View Recent Updated Flights:
Retrieve and display recent admin actions and edits related to flights throughout the system.

View/Update Admin Contact Information:
Retrieve and display contact information, allowing for updates saved in the database.

Update Flight details
Save flight information provided by the airline in the database.

3.2. Quality Requirements

Reliability:
The system should maintain high availability, with a maximum downtime of 1 hour per six months for maintenance.
Server-side validation of user input ensures data integrity and prevents malicious input.
Data normalization using primary and foreign keys in the database prevents data discrepancies.

Efficiency:
Search queries and page loads should be processed and formatted efficiently by the system's backend, providing timely responses based on user requests.

Security:
Admin passwords will be encrypted within the database for enhanced security.
Access to pages within the system will be restricted to authorized users only, ensuring that admin tasks are performed by privileged users only.
Files and directories within the system will be secured to prevent unauthorized access and maintain the intended structure.

Maintainability:

PHP pages used by the system will be organized centrally within the file structure, allowing for easy updates that affect all relevant areas.
Administrators will have the capability to edit aspects of the system directly related to their account, ensuring maintainability and customization options for admin users.

3.3. User Interface Requirements

Home Page:
The home page should provide quick access to essential functions such as flight search, booking, and account management.
Users should have clear navigation options to explore flight listings, view their bookings, and access support services.
The layout should be clean, intuitive, and responsive, adapting to different screen sizes and devices for a seamless user experience.
Important announcements, promotions, or alerts should be prominently displayed on the home page.

Flight Search and Booking:
The flight search interface should allow users to specify criteria such as date, destination, class, and number of passengers.Search results should be displayed in a user-friendly format, showing relevant flight details, prices, and availability.Users should be able to select and book flights easily, with clear instructions for seat selection, payment processing, and itinerary management.Confirmation messages and booking summaries should be provided after successful booking.

User Profile Management:
Users should have a dedicated profile section to manage personal information, preferences, and past bookings.Profile editing should be straightforward, allowing users to update contact details, password, and notification settings.Users should be able to view their booking history, upcoming trips, and loyalty program benefits in the profile dashboard.

Admin Control Panel (for airline administrators):
The admin panel should provide comprehensive tools for managing flights, reservations, staff, and system configurations.
Dashboard widgets or summaries should display key metrics such as flight occupancy, revenue, and customer feedback.
Admins should have options to add, edit, or remove flights, assign crew members, and configure pricing and discounts.
Reporting and analytics features should be accessible, allowing admins to generate insights on flight performance, passenger demographics, and revenue trends.

Accessibility and Responsiveness:
The user interface should adhere to accessibility standards, ensuring compatibility with screen readers, keyboard navigation, and color-contrast settings for users with disabilities.
The interface should be responsive, optimizing layout and functionality across devices such as desktops, tablets, and smartphones.
Text, buttons, and interactive elements should be appropriately sized and spaced for easy interaction on touch screens.

3.4. Logical Database Requirements

Entities:
Flights: Store information about flights such as flight number, departure and arrival airports, departure and arrival times, aircraft type, seat availability, and ticket prices.
Passengers: Maintain passenger details including passenger ID, name, contact information, booking history, and loyalty program status.
Bookings: Record booking details such as booking ID, passenger ID, flight number, seat assignment, booking status, and payment information.
Aircraft: Store data related to aircraft including aircraft ID, model, seating capacity, maintenance history, and availability status.
Crew: Manage information about flight crew members such as crew ID, name, position, qualifications, and assigned flights.

Relationships:
Flights-Bookings: One-to-Many relationship between flights and bookings, where one flight can have multiple bookings but each booking is associated with only one flight.
Passengers-Bookings: One-to-Many relationship between passengers and bookings, indicating that one passenger can have multiple bookings but each booking is linked to one passenger.
Aircraft-Flights: One-to-Many relationship between aircraft and flights, signifying that one aircraft can operate multiple flights but each flight is associated with one aircraft.
Crew-Flights: Many-to-Many relationship between crew and flights, as multiple crew members can be assigned to multiple flights, and each flight can have multiple crew members.

Attributes:
Flight Attributes: flight_number, departure_airport, arrival_airport, departure_time, arrival_time, aircraft_type, seat_capacity, ticket_price.
Passenger Attributes: passenger_id, name, contact_info, booking_history, loyalty_status.
Booking Attributes: booking_id, passenger_id, flight_number, seat_assignment, booking_status, payment_info.
Aircraft Attributes: aircraft_id, model, seating_capacity, maintenance_history, availability_status.
Crew Attributes: crew_id, name, position, qualifications, assigned_flights.

Normalization:
The database should be normalized to eliminate data redundancy and maintain data integrity. This involves organizing data into separate tables and establishing relationships between them using primary and foreign keys.

Integrity Constraints:
Implement integrity constraints such as primary keys, foreign keys, unique constraints, and check constraints to ensure data accuracy, consistency, and reliability.

Database Management System (DBMS):
Choose a suitable DBMS like MySQL, PostgreSQL, or Oracle that supports the required features, scalability, and performance for managing the airline management system database effectively.

3.5. Use Cases Diagram