fsd-demo-SRS

December 30, 2024

0.1 Software Requirements Specification (SRS) for full stack development example - Flask and SQLite application (fsd-demo):

---

1 Software Requirements Specification (SRS)

1.1 1. Introduction

1.1.1 1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for a simple full-stack application based on Flask and SQLite. This application will manage user data, including creating, reading, updating, and deleting user information.

1.1.2 1.2 Scope

This application is a web-based system that allows users to manage a list of users. The system will provide functionalities to add new users, view existing users, update user information, and delete users. The application will use Flask as the web framework and SQLite as the database.

1.1.3 1.3 Definitions, Acronyms, and Abbreviations

- **Flask**: A micro web framework written in Python.
- **SQLite**: A C-language library that provides a lightweight, disk-based database.
- **CRUD**: Create, Read, Update, Delete.

1.1.4 1.4 References

- Flask documentation: Flask
- SQLite documentation: SQLite

1.2 2. Overall Description

1.2.1 2.1 Product Perspective

The application is a standalone system that will be accessed via a web browser. It will interact with an SQLite database to store and retrieve user data.

1.2.2 2.2 Product Functions

- **Create User**: Add a new user to the database.
- **Read Users**: Display a list of all users.
- **Update User**: Modify the details of an existing user.
- **Delete User**: Remove a user from the database.

1.2.3 2.3 User Classes and Characteristics

- **Admin**: The primary user who will manage the user data. The admin will have full access to all functionalities of the application.

1.2.4 2.4 Operating Environment

- **Server**: The application will run on a server with Python and Flask installed.
- **Database**: SQLite will be used as the database.

1.2.5 2.5 Design and Implementation Constraints

- The application must be developed using Flask and SQLite.
- The application should follow the MVC (Model-View-Controller) architecture.

1.2.6 2.6 Assumptions and Dependencies

- The user has a web browser to access the application.
- The server environment supports Python and Flask.

1.3 3. Specific Requirements

1.3.1 3.1 Functional Requirements

3.1.1 Create User

- **Description**: The system shall allow the admin to add a new user.
- **Inputs**: User name, email, and age.
- **Outputs**: Confirmation message and updated user list.

3.1.2 Read Users

- **Description**: The system shall display a list of all users.
- **Inputs**: None.
- **Outputs**: List of users with their details.

3.1.3 Update User

- **Description**: The system shall allow the admin to update user information.
- **Inputs**: User ID, new name, new email, and new age.
- **Outputs**: Confirmation message and updated user list.

3.1.4 Delete User

- **Description**: The system shall allow the admin to delete a user.
- **Inputs**: User ID.
- **Outputs**: Confirmation message and updated user list.

1.3.2 3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- The system should respond to user actions within 2 seconds.

3.2.2 Security Requirements

- The system should validate all inputs to prevent SQL injection attacks.

3.2.3 Usability Requirements

- The user interface should be intuitive and easy to navigate.

1.3.3 3.3 Database Requirements

- The database shall store user information including ID, name, email, and age.
- The database shall be an SQLite database.

1.4 4. External Interface Requirements

1.4.1 4.1 User Interfaces

- The home page shall display a list of users.
- The update page shall provide a form to update user information.
- The delete functionality shall be accessible via a button next to each user.

1.4.2 4.2 Hardware Interfaces

- The application shall run on standard server hardware.

1.4.3 4.3 Software Interfaces

- The application shall interact with the SQLite database using SQL queries.

1.4.4 4.4 Communications Interfaces

- The application shall use HTTP/HTTPS for communication between the client and server.

[ ]:

```

```