Software Requirement Specification (SRS) Document

Project Name: Medi-Finder
Date: 13/01/2025
Version: 1.0
By: Nyuysemo Carlglain

Version History:

Version	**Author**	**Updated Date**
1.0	Carlglain	/

Review History:

Version	Author	Updated Date
1.0	Carlglain	/

Approval History:

Version	Author	Updated Date
1.0	Carlglain	/

Index

1. Purpose of Document

The purpose of this document is to provide a description a Medical App (Medi Finder) that will help users (persons) to be able to access essential medications and healthcare services during emergencies or in unfamiliar locations.

Medi Finder is a platform designed to simplify healthcare access by providing real-time information on pharmacies, their operating hours, and the availability of medications. With additional features like user reviews, emergency contacts, and medication details, Medi Finder ensures that users can make informed decisions and access the healthcare services they need, precisely when they need them.

2. Scope of Document

Medi-Finder is a web application designed with a mobile-first approach that assists users in locating essential medications and healthcare services during emergencies or when they are in unfamiliar areas.

3. Project Definition

3.1 Objective

To develop a system that helps users locate medication and medical facilities to minimize confusion and panic in emergency health situations.

3.2 Target Audience

Buea population for the time being and Cameroon at large with time.

3.3 System Platform

The system will be hosted on vercel and available on web browsers.

3.4 Platform Orientation

Web application with mobile first approach will be made available on Desktops and mobile phones.

3.5 Language

This software will be developed using html, CSS and java script with the REACT.js framework.

4. Project Strategy

4.1 Define Project Work Strategy

The system development life cycle I will use is Agile. I have less than a month to build this website so I’ll have to deliver it faster or atleast have some functionality working before the end of February and Agile methodology permits this rapid development. Also, there are speculations of future improvements hence flexibility is required and Agile from its name permits flexibility.

4.2 Milestones of the Project

No. of Milestones	Delivery Date
1 Deployment to Vercel	13/01/2025
2	
3	
4	

5. Tools and Technology to Use

Tools and Technologies in the Project	
Programming Language	JavaScript
Web Server	Node.js
Deployment Platform	Vercel
Version Control platform	GitHub
Server OS	Windows
Frameworks	REACT

Intergrations

1. Geolocation Services

- **Google Maps API**: For mapping, geolocation, and route optimization.
- **OpenStreetMap**: An open-source alternative for mapping and location data.

2. Authentication and Authorization

- **Firebase Authentication / Auth0**: To manage user accounts and secure login processes for both passengers and drivers.

3. Notification Services

- **Firebase Cloud Messaging (FCM)**: For push notifications to mobile devices.

4. Backend Services

- **Node.js/Express**: To handle the server-side API requests and data processing.

5. Data Storage

- **Firebase Firestore**: As an alternative NoSQL database for real-time data syncing.

6. Analytics and Monitoring

- **Google Analytics**: To track user behaviour and interactions within the app.
- **Sentry**: For error tracking and performance monitoring.

7. Deployment and Hosting

- **Vercel**: For hosting the application, providing scalability and reliability.

8. Testing and Quality Assurance

- **Jest / Mocha**: For unit testing the backend logic.
- **Cypress / Selenium**: For end-to-end testing of the application.

6. Functional Requirements

a. Requirements

- **User Registration/Login**: Allow users to create accounts for a personalized experience.
- **Guest Mode Access**: Provide core features without requiring registration.
- **Pharmacy Information Display**: Show operating hours, contact details, and available medications.
- **Pharmacy Search**: Enable users to search for nearby pharmacies using location services or manual entry.
- **Emergency Contacts**: Provide a feature to save and access local emergency contact information.
- **Notifications**: Enable alerts for medication availability or updates related to nearby pharmacies.
- **User Reviews and Ratings**: Allow users to submit and view reviews and ratings for pharmacies.
- **Real-Time Medication Data**: Provide up-to-date information on the availability of specific medications at pharmacies.

Future Features

- **Medication Information**: Offer detailed descriptions of medications, including indications, dosages, and side effects.
- **Favourites List**: Allow users to save preferred pharmacies and medications for quick access.
- **Accessibility Features**: Support for visually impaired users (e.g., voice navigation).

8. Non-functional Requirements

8.1 Security

- Input/output Validation

- Ensure that all user inputs are validated and sanitized to prevent malicious data from being processed.

-This helps prevent injection attacks like SQL injection and cross-site scripting (XSS). This will involve any input form and process that generates output.

- ➢ Authentication and Authorization

  - Securely verify user identities and define what resources they can access.

  - Implement multi-factor authentication (MFA) for added security. Use role-based access control (RBAC) to ensure users have access only to resources necessary for their role. It's mostly recommended for third party to handle this operations like Google, Apple, Facebook but this will depend.

- ➢ Data Encryption

  - Protect data both in transit and at rest to prevent unauthorized access.

  - Use HTTPS (SSL/TLS) to encrypt data sent between the client and server.

  Make use of AES algorithm for server side encryption protect users Data and financial information.

- ➢ Regular Security Audits

  - Regularly assess the application for vulnerabilities.

  - Conduct penetration testing and vulnerability assessments to identify and remedy security flaws.

- ➢ Error Handling

  - Properly handle errors to avoid exposing sensitive information.

  - Display generic error messages to users and log detailed errors for developers. This prevents attackers from gaining insights into the system.

- Third-Party Libraries

  - Keep third-party libraries and dependencies up to date to prevent vulnerabilities.

  - Regularly review and update libraries, and avoid using deprecated or unsupported ones.

- Database Security

  - Secure database interactions to prevent unauthorized access.

  - Use parameterized queries to prevent SQL injection. Limit database user permissions to only what is necessary.

- Logging and Monitoring

  - Keep track of application activities to detect suspicious behavior.

  - Implement logging for all critical operations and monitor logs for signs of unauthorized access or attacks. There are lots of tools available and even coding activities to perform this operations.

<u>**8.2 Compatibility**</u>

- This Website will be able to run on any device that has a web browser.
- It will be able to run on windows OS, MAC OS, LINUX, Android, just to name few.
- Responsiveness will be ensured in this website so that it has good looks in any type of screen it's viewed from.

<u>**8. 3 Reliability**</u>

- User data cannot be accessed by any other user or unauthorized person hence there is data integrity
- The website handles errors and provide user friendly error messages.

8.6 Scalability

- The website is designed to handle multiple users without failing or degrading in performance.

8.7Maintainability

- Clear and comprehensive documentation (user guide, developer guide) will be available for the website's code, design, and infrastructure.
- Git will be used to track code changes and manage updates.

8.8 Usability

- The website’s design is user friendly and easy to use as it’s self-explanatory.

8.9 Capacity

- The website will be optimized for speed and efficiency to provide a fast user experience.
- Regular load testing will be conducted to assess the website's capacity and identify potential bottlenecks.

9. Definitions and Acronyms

- DB: Database
- OS: Operating System.