**SOFTWARE REQUIREMENTS SPECIFICATION**

**(SRS DOCUMENT)**

**FOR**

**<Health Reach>**

**Version 1.0**

**By**

**Student 1: BROWNIA Nji**

**Student 2: Eyong Marion**

**Student 3: AKIJIKA Randolph**

**Supervisor**

**Mr. ANDREW AJUATA**

TABLE OF CONTECT

[1. Introduction 3](#_Toc171933614)

[2. System Scope 3](#_Toc171933615)

[3. Functional Requirements 6](#_Toc171933616)

[3.1 User Requirements 6](#_Toc171933617)

[3.2 Admin Requirements 7](#_Toc171933618)

[4. Non-Functional Requirements 8](#_Toc171933619)

[4.1 Performance 8](#_Toc171933620)

[4.2 Security 9](#_Toc171933621)

[4.3 Usability 9](#_Toc171933622)

[4.4 Reliability 9](#_Toc171933623)

[5. System Architecture 10](#_Toc171933624)

[6. Data Model 11](#_Toc171933625)

[7. Development Environment 11](#_Toc171933626)

[8. Testing and Deployment 12](#_Toc171933627)

[9. Maintenance and Support 12](#_Toc171933628)

[10. Future Enhancements 13](#_Toc171933629)

[11. Conclusion 13](#_Toc171933630)

1. Introduction

This document outlines the Software Requirements Specification (SRS) for a comprehensive web management system designed to assist users in finding, comparing, and booking appointments with hospitals in a specified geographical area. The system will provide detailed information about hospitals, their services, user reviews, and facilitate online appointment scheduling.

2. System Scope

The system will contain the following functionalities:

* **Hospital Listing and Information:**
* Display a comprehensive list of hospitals in a defined area (e.g., city, state, region).
* Allow users to filter hospitals by specialty, location, insurance coverage, etc.
* Provide detailed information about each hospital, including:
* Name, address, contact information, website, social media links.
* Operating hours, emergency services availability.
* Specialties and services offered (e.g., cardiology, oncology, pediatrics, specific diseases).
* Hospital staff and doctor’s profiles (with qualifications and experience).
* Facility details, including bed capacity, equipment availability, and amenities.
* Integrate with external sources for updated information (e.g., official hospital websites, healthcare databases).
* **User Review System:**
* Allow registered users to submit reviews and ratings for hospitals they have visited.
* Include fields for user experience, waiting times, staff friendliness, cleanliness, etc.
* Display review summaries and average ratings for each hospital.
* **Search and Navigation:**
* Implement robust search functionalities for hospitals, specific services, and diseases.
* **Online Appointment Booking:**
* Allow users to book appointments with specific doctors or departments at hospitals.
* Integrate with hospital appointment systems or provide a dedicated system for scheduling.
* Enable users to select preferred dates and times for appointments.
* Send appointment confirmations and reminders via email or SMS.
* Provide a mechanism for users to cancel or reschedule appointments.

**Admin Panel:**

* Provide a dedicated admin panel to:
* Manage hospital information, including updates and edits.
* Moderate user reviews and ratings.
* Monitor website usage and analytics.
* Manage user accounts and permissions.
* Control appointment booking settings and availability.

3. Functional Requirements

3.1 User Requirements

* Users should be able to easily browse and filter hospital listings based on various criteria.
* Users should be able to access comprehensive information about individual hospitals.
* Users should be able to submit reviews and ratings for their experiences at hospitals.
* Users should be able to view and interact with other user reviews and ratings.
* Users should be able to search for specific doctors or departments within hospitals.
* Users should be able to book appointments online with preferred doctors and departments.

3.2 Admin Requirements

* Admins should be able to add, edit, and delete hospital information.
* Admins should be able to moderate user reviews and ratings.
* Admins should be able to manage user accounts and permissions.
* Admins should be able to control appointment availability and settings.
* Admins should be able to manage and update hospital data through an API or integration.

4. Non-Functional Requirements

4.1 Performance

* The system should be responsive and load quickly, even with a large number of users.
* Search functions should return relevant results efficiently.
* The website should be optimized for various devices (desktop, mobile, tablet).

4.2 Security

* User data should be secured and protected from unauthorized access.
* Secure login mechanisms should be implemented.
* The website should be protected from hacking attempts and cyber-attacks.

4.3 Usability

* The system should be easy to use and navigate.
* The user interface should be interactive and user-friendly.
* The system should be accessible to users with disabilities.

4.4 Reliability

* The system should be reliable and available at all times.
* Appointment booking system should be robust and reliable to avoid scheduling conflicts.

5. System Architecture

The system will be developed using a modular architecture based on the following components:

* Front-End: HTML, CSS, JavaScript, React (or similar framework).
* Back-End: Python (Flask or Django), PHP.
* Database: MySQL, phpMyAdmin.
* API Integration: Use APIs for data sources like hospital websites, healthcare databases, map services, and appointment scheduling systems.

6. Data Model

The system will utilize a database to store information about:

* Hospitals: Name, address, contact details, ratings, reviews, appointment availability.
* Users: User accounts, login details, review history.
* Doctors/Departments:
* Profile information, specialties, appointment availability, etc.
* Reviews: Review text, ratings, user ID, hospital ID, etc.
* Appointments: Date, time, user ID, hospital ID, doctor/department ID, confirmation status, etc.

7. Development Environment

The development environment will include:

* Code Editor: Visual Studio Code, Sublime Text, etc.
* Version Control System: Git, GitHub.

8. Testing and Deployment

* Testing: The system will undergo thorough testing to ensure functionality, performance, security, and user experience.

9. Maintenance and Support

The system will require regular maintenance to ensure its stability and security. A dedicated support team will be available to address user issues and provide technical assistance.

10. Future Enhancements

Future enhancements to the system could include:

* Integration with insurance providers for coverage verification.
* Personalized recommendations based on user preferences and medical history.
* Integration with wearable health devices to track user health data.
* Advanced analytics and reporting for hospital performance and user behavior.
* Implementation of telehealth features for virtual consultations.

11. Conclusion

This SRS document provides a comprehensive overview of the requirements for the web management system. This system will be a valuable resource for users to compare and select hospitals, book appointments online, and share their experiences. It aims to improve access to healthcare services and empower users to make informed decisions about their medical care.