1. Introduction

1.1 Purpose

This document specifies the software requirements for a doctor-patient portal, which is a new, self-contained product. The portal aims to streamline communication and interaction between patients and doctors, facilitating appointment booking, health record management, and feedback.

1.2 Document Conventions

All requirements in this document are assumed to have a High priority unless otherwise specified. The text "TBD" is used as a placeholder for information that is not yet available.

1.3 Intended Audience and Reading Suggestions

This document is intended for various readers, including developers, project managers, testers, and marketing staff. It provides an overview of the system's functions and detailed requirements.

1.4 Product Scope

The **Doctor-Patient Portal** is a web-based application that allows patients to find and book appointments with doctors, and allows doctors to manage their appointments and profiles. It also includes an admin panel for managing users and viewing system summaries. The primary goal is to provide an efficient and user-friendly platform for healthcare-related interactions.

1.5 References

This SRS references the IEEE template provided.

2. Overall Description

2.1 Product Perspective

The Doctor-Patient Portal is a new, self-contained product. It is not a replacement for any existing system.

2.2 Product Functions

The portal will perform the following major functions:

- **Patient Functions:**
  - User registration and login.
  - Searching for doctors by category.
  - Filtering search results.
  - Booking and managing appointments.
  - Viewing prescriptions and reports.
  - Editing personal profiles.

    - Providing feedback to doctors.
    - Identifying and seeking care for emergency conditions.
- **Doctor Functions:**
    - Checking patient requests.
    - Viewing and managing upcoming appointments.
- **Admin Functions:**
    - Approving or deleting doctor profiles.
    - Viewing patient feedback and updating doctor ratings.
    - Generating and viewing a summary of doctor appointments (as a chart).
    - Managing user profiles (both doctors and patients).

2.3 User Classes and Characteristics

The product is intended for three main user classes:

- **Patients:** Individuals seeking medical care. They need a simple, intuitive interface to find doctors and manage appointments.
- **Doctors:** Healthcare professionals. They require a clear view of their schedule and patient requests.
- **Admin:** System administrators responsible for managing the platform. They need elevated privileges to manage user profiles and view system analytics.

2.4 Operating Environment

The software will operate on a standard web browser (Chrome, Firefox, Safari, Edge) on various operating systems. The application will be built using **React**.

2.5 Design and Implementation Constraints

The project will use **React** for development. Other constraints include:

- A specific database technology (TBD).
- Compliance with data privacy and security regulations (TBD).

2.6 User Documentation

The software will be delivered with a user manual and an online help section.

2.7 Assumptions and Dependencies

- Third-party components for payment gateways or mapping services (TBD) will be available and functional.
- An external database management system will be used.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be web-based. It will feature a clean, intuitive layout with standard buttons and functions. The design will be consistent across all pages to ensure a seamless user experience.

3.2 Hardware Interfaces

The software will interact with standard hardware components such as monitors, keyboards, and mice. No specialized hardware interfaces are required.

3.3 Software Interfaces

The portal will interface with:

- **Operating Systems:** Standard operating systems such as Windows, macOS, and Andriod
- **Databases:** A database to store user information, appointments, and other data (TBD).
- **Other Applications:** Potential interfaces with other applications (e.g., electronic health records, payment systems) will be defined as needed.

3.4 Communications Interfaces

The portal will use standard web protocols like

HTTP or **HTTPS** for secure communication. All data transfers will be encrypted to ensure security and privacy.

4. System Features

This section organizes the functional requirements by system features.

4.1 Patient Management

This feature covers all functionalities related to the patient user class.

4.1.1 Description and Priority

This is a High priority feature as it is the core of the patient experience. It includes everything from user registration to viewing their health history.

4.1.2 Stimulus/Response Sequences

- **Stimulus:** Patient navigates to the registration page.
  - **Response:** System displays the registration form.
- **Stimulus:** Patient submits the login form.
  - **Response:** System authenticates the user and redirects to the dashboard or displays an error message.
- **Stimulus:** Patient searches for a doctor.
  - **Response:** The system displays a list of doctors based on the search criteria.
- **Stimulus:** Patient books an appointment.

    - **Response:** The system records the appointment and notifies the doctor.
- **Stimulus:** Patient submits a feedback form.
    - **Response:** The system saves the feedback and notifies the admin.

4.1.3 Functional Requirements

- **REQ-P1:** The system must allow new patients to register with a valid email and password.
- **REQ-P2:** The system must allow registered patients to log in and log out securely.
- **REQ-P3:** The system must provide a search function to find doctors by category.
- **REQ-P4:** The system must allow patients to filter doctor search results.
- **REQ-P5:** The system must allow patients to book appointments with a selected doctor.
- **REQ-P6:** The system must display prescriptions and reports to the patient.
- **REQ-P7:** The system must allow patients to edit their profile information.
- **REQ-P8:** The system must store previous health history for each patient.
- **REQ-P9:** The system must allow patients to submit feedback for a doctor.

4.2 Doctor Management

This feature covers all functionalities for the doctor user class.

4.2.1 Description and Priority

This is a High priority feature as it is essential for doctors to interact with patients.

4.2.2 Stimulus/Response Sequences

- **Stimulus:** Doctor logs into their account.
    - **Response:** System displays a dashboard with upcoming appointments and new patient requests.
- **Stimulus:** Doctor accepts or rejects a patient request.
    - **Response:** The system updates the appointment status and notifies the patient.
- **Stimulus:** Doctor manages an appointment.
    - **Response:** The system updates the appointment schedule.

4.2.3 Functional Requirements

- **REQ-D1:** The system must allow doctors to check patient requests.
- **REQ-D2:** The system must provide a view of all upcoming appointments for a doctor.
- **REQ-D3:** The system must allow a doctor to accept or decline an appointment request.

4.3 Admin Management

This feature covers the functionalities of the admin panel.

4.3.1 Description and Priority

This is a High priority feature that provides the necessary tools for system administration and quality control.

4.3.2 Stimulus/Response Sequences

- **Stimulus:** Admin approves a new doctor profile.
  - **Response:** The system changes the doctor's status to "approved" and makes them visible to patients.
- **Stimulus:** Admin views a patient's feedback.
  - **Response:** The system displays the feedback details. If approved, the system updates the doctor's rating.
- **Stimulus:** Admin requests a summary of appointments.
  - **Response:** The system generates a chart showing appointment summaries.
- **Stimulus:** Admin edits a user profile.
  - **Response:** The system updates the user's information.

4.3.3 Functional Requirements

- **REQ-A1:** The system must allow the admin to approve or delete doctor profiles.
- **REQ-A2:** The system must allow the admin to view patient feedback on doctors.
- **REQ-A3:** The system must update a doctor's rating if the admin approves the feedback.
- **REQ-A4:** The system must generate a chart summary of doctor appointments.
- **REQ-A5:** The system must allow the admin to manage (change information of) doctor and patient profiles.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must be able to handle at least 1,000 concurrent users without significant performance degradation. All common actions (login, search, book appointment) should complete within 3 seconds.

5.2 Safety Requirements

The system must not store sensitive health information in an unencrypted format. Data backups must be performed daily to prevent data loss.

5.3 Security Requirements

All data transmission between the client and server must be encrypted using

HTTPS. User authentication will be required for all user classes to access their respective features.

5.4 Software Quality Attributes

- **Usability:** The interface should be intuitive and easy to navigate for all user classes.
- **Reliability:** The system must be available 99.9% of the time.

- **Maintainability:** The codebase should be well-documented to allow for easy maintenance and future updates.

5.5 Business Rules

- Only an admin can approve or delete a doctor's profile.
- Only a registered patient can book an appointment or provide feedback.
- Feedback will only be added to a doctor's rating if approved by an admin.

6. Other Requirements

This section is a placeholder for any other requirements not covered in the previous sections, such as legal or internationalization requirements.

Appendix A: Glossary

- **SRS:** Software Requirements Specification
- **TBD:** To Be Determined

Appendix B: Analysis Models

Appendix C: To Be Determined List

- **TBD-1:** Specific database technology to be used.
- **TBD-2:** Details on data privacy and security regulations to comply with.
- **TBD-3:** Specific third-party components for payment gateways or mapping services.