****

**Doctor on Call**

**Software Requirements Specifications**

-Rohan Topno

Contents

|  |  |
| --- | --- |
| 1. Introduction |  |
| 2. General Description |  |
| 3. Software Requirements |  |
| 4. User Interface |  |
| 5. Security |  |
| 6. Use Case Models and Sequence Diagrams |  |

1. **Introduction**
   1. Purpose
      1. This Android application is designed to offer the users a simplified and accessible way to have a preliminary diagnosis based on a multiple-choice based questions and answers system. This will be aim of this application will be to provide the users with an easy-to-use tool which can resolve small medical problems without the immediate need of a medical professional to diagnose them. It is not intended to replace medical advice but help the user to get a better understanding of their issues, and if needed guide them towards a medical professional to diagnose them. If the user choses to do so, it can also provide the medical professional with the data gathered by the application to help them diagnose faster.
         * This application will provide the users the ability to:
         * Register and create a personal account.
         * Log in to their account
         * Access a dashboard, which will be used for all the features of the application currently present and those which will be added in the future.
         * Access the core functionality of the app which is the diagnosing system.
         * Review their own data, where they will be in control of the data and will have the ability to store or remove their data according to their choices.
   2. Scope of the Application
      1. The application is going to be modular, allowing us to expand the scopt of the application in future. As of now, the application will focus on the following:
         * **User Authentication**: An account registration and a login system that will securely manage the user’s credential and sessions.
         * **Dashboard system**: The dashboard system will allow the user to navigate through different sections of the applications. It will act as the central navigation system.
         * **Question-Based Diagnosis**: The core feature of the application is to provide diagnosis to the user by asking them questions which are already created and stored in a database, with multiple-choice options which the user can select, and it will ultimately lead the user to a diagnosis of their problem. It is hoped that it will also be able to connect the user with a medical professional if the user feels so, or if it’s required.
         * **Data Storage**: User information including diagnosis history will be stored securely. The exact method (local or cloud) is yet to be decided.
         * **Security and Data Privacy**: The app is built with security in mind, so any data transfer and storage will try to protect the user data.
      2. The application is planned to be updated and maintained, and further updates to the diagnostic system will also be made.
   3. Target user/audience
      1. This application target user’s who need fast and basic health advice without the need of visiting a doctor / medical professional. The user base could include:
         * Medical professionals to provide a quick help tool for their patients
         * Individuals who are experiencing symptoms not serious enough to warrant a visit to the doctor.
         * Potentially users from remote locations where health centres, hospitals and clinics are limited.
      2. User interface will be designed as such so that even a person who’s not technologically pro efficient will be able to use the application.
2. **Overall Description**
   1. Thi**s** section provides a high-level overview of the application, its objectives, functionalities, and context within the broader system. This section will help to understand how the different components of the app work together to achieve the intended goals.
      1. The application will be designed to be easily maintainable and scalable, enabling future developers to extend its functionalities.
      2. The application will be divided into three layers:
         * **User Interface**: A clean and intuitive interface that facilitates user registration, login, navigation via the dashboard and the diagnostic process.
         * ****Business Logic Layer****: Handles the processing of user input, guides the question flow, and computes preliminary diagnoses based on user responses.
         * ****Data Management Layer****: Responsible for storing and retrieving user information, such as credentials, diagnosis history, and preferences.
   2. **Product Functions**
      1. The core functionalities of the application are centred around user management, diagnostic capabilities, and security. The following summarizes these key functions:
      2. User Registration:
         * New users can create accounts by providing basic information such as email and password.
         * The registration process ensures that valid, non-duplicate accounts are created, with secure storage of user credentials.
      3. Dashboard:
         * Once logged in, users are directed to the main dashboard. This acts as the central navigation point, from which users can initiate a new diagnosis, view past diagnosis results, or update personal details.
      4. Question-Based Diagnosis:
         * The core feature of the app is a dynamic diagnostic process. The user is asked a series of questions related to their health concerns.
         * The questions follow a decision-tree model, where each answer leads to a subsequent, more specific question.
         * At the end of the question flow, the app generates a preliminary diagnosis based on predefined rules and data patterns.
      5. Data Management:
         * The app stores user data, including diagnosis history, securely. The storage solution (either local or cloud-based) is not yet finalized but will comply with security standards.
         * Users can review their past diagnoses and track their health insights over time.
      6. Security Management:
         * User credentials are stored securely, and personal information is protected through encryption.
   3. Constraints
      1. Mobile-Only Access
         * As of now the application is mobile only, and cannot be accessed through a website. Future version may expand and be a cross-platform application.
      2. Limited Capabilities
         * As of now the software can provide only a certain about of diagnosis, in the future it can expand and provide more help.
3. **Requirements**
   1. Functional Requirements
      1. User Registration
         * The system must allow new users to register by providing their email address, username, and password.
         * Passwords should be validated for a minimum length and complexity (e.g., at least 8 characters, including one uppercase letter and one number).
         * The system must ensure no duplicate registrations with the same email address.
         * After successful registration, users should receive a confirmation email (if integrated with email services).
      2. User Login
         * Registered users must be able to log in with their email and password.
         * The system must authenticate users based on valid credentials, with an error message for incorrect inputs.
      3. Dashboard Access
         * Upon login, users must be redirected to the dashboard, which serves as the primary interface for navigation.
         * The dashboard must display key options: start a new diagnosis, view past diagnoses, or update account settings.
         * The dashboard should also show any recent activity or updates relevant to the user.
      4. Diagnostic Question Flow
         * The app must present users with a series of multiple-choice questions designed to assess their medical condition.
         * The questions should be presented sequentially, with the next question determined based on the user’s previous answers (decision-tree logic).
         * The system must calculate a preliminary diagnosis based on predefined rules and user responses.
         * The diagnostic result should be displayed clearly, with advice on potential next steps (e.g., "Consult a healthcare provider" or "Monitor symptoms").
      5. Data Storage
         * The app must securely store user data, including login credentials, diagnostic history, and any personal health information.
         * The data must be encrypted both at rest and during transmission.
         * The system should allow users to review past diagnostic sessions via the dashboard.
      6. Account Management
         * Users must be able to update their account details (e.g., change password, update email).
         * Users should have the option to delete their account, which will permanently remove all associated data from the system.
      7. Notifications
         * If integrated, the app should send users notifications (e.g., reminders to complete a diagnosis or updates to terms and conditions).
         * Users must have the option to enable/disable notifications through settings.
   2. Non-Functional Requirements
      1. TBD
   3. Software Interfaces
      1. TBD
4. **User Interface**
   1. TBD
5. **Security**
   1. TBD
6. **Use Case Models and Sequence Diagrams**

