Software Requirements Specification

for

Personalized Health and Wellness Assistant

Version 1.0 approved

Prepared by Vansh Kumar Payala

Vellore Institute of Technology, Chennai

3rd August 2024

Table of Contents

1. Introduction

1.1 Purpose

The "Personalized Health and Wellness Assistant" version 1.0 software requirements are outlined by this Software Requirements Specification (SRS) document. This document is aimed at providing a thorough description of the application's interfaces, constraints and functionalities. The whole scope of this SRS involves every single part of the system with all its components described and how they interact. They consist of artificial intelligence health recommendation engine, chat bot for medication and prescription guidance, user data input and analysis modules alongside overall user interface. This SRS makes it necessary for all stakeholders to have a proper understanding of what the system needs to be like before as well as during its development process.

1.2 Document Conventions

This paper observes all the standard IEEE conventions for software requirements specifications (SRS). Each requirement is uniquely identified by a specific identifier such as REQ-1, REQ-2, which ensures that there can be clear reference and tracking during development. Requirements are categorized according to their functionality, user classes and other relevant criteria that facilitate organized and comprehensible arrangement. Moreover, higher-level requirements are assigned with priorities (High, Medium, Low), which cascade down to more detailed requirements unless stated otherwise so that important functionalities may receive appropriate attention when developing and testing them. It is this perspective that facilitates effective communication among stakeholders and forms a strong basis for project planning and execution.

1.3 Intended Audience and Reading Suggestions

This document is meant to describe the basic functional and non-functional requirements of this waste management system, it serve as an SRS (Software Requirements Specification) for developers but also a reference point towards which our engineers might verify each time they need guidance throughout application development. Certain chapters pertain more relevant to specific groups depending on their roles and needs.

- **Developers:** Focus on detailed requirements & system features through the contents that are necessary for coding and implementation an overview of External Interface Requirements will be important to grasp system inter actions as well.
- **Project Managers:** It provides an insight into the project scope (objectives and constraints) as well as performance criteria which are of vital importance for carrying out planning activities from a Project Manager perspective.
- **Marketing Staff:** The product scope and overall description sections give a complete overview of the features of application which helps in deciding on marketing strategies.
- **Users:** User interfaces and system features sections explain how the application will function and what to expect in terms of usability and features.

- **Testers:** Detailed requirements, system features, and nonfunctional requirements sections help in designing test cases and ensuring the application meets all specified criteria.
- **Documentation Writers:** Overall description and user interfaces sections provide the context and details needed to develop comprehensive user manuals and documentation.

1.4 Product Scope

The scope of "Personalized Health and Wellness Assistant" involve creation of a mobile app that uses artificial intelligence to provide personalized health and wellness resources. This application will present different capabilities as well as comprehensive health support. What stands out in the application is its ability to give personalized health suggestions based on information provided by users such as their lifestyles, metrics of health or personal preferences among others. These recommendations will be in terms of dieting, exercise, lifestyle choices, among others depending on the user. It will also include a tracking component where users can assess progress towards their health goals and get feedback on how they are doing. The application will further include a chat bot that guides its users through medication issues including prescriptions that ensures quality medical help all the time. The aim of "Personalized Health and Wellness Assistant" is to increase active participation by end-users empowering them to make knowledgeable decisions regarding their own welfare concerning both physical aspects and psychological matters.

1.5 References

- IEEE Software Engineering Standards
- Existing health and wellness applications
- AI and machine learning research papers relevant to personalized health recommendations

2. Overall Description

2.1 Product Perspective

This is going to be in the mobile platform where the company is planning a new self-contained app that should be easy to connect up most modern smartphones and tablets known as the, "Personalized Health and Wellness Assistant." Unlike other health apps that offer generic advice, this one will be powered by Artificial Intelligence algorithms to analyze user data to provide personal advice on how individuals can maintain good health. As per how one prefers his/her foods include everything like personal tastes of foods one loves or does not like. The idea for the application is to address a significant issue in the present competition by providing comprehensive support for personal health management. A better way to monitor and track with a series of prevention and care plans like the Detailed Health Assessment, Personalized wellness Plan & Progress Tracker Interconnected Modules where work and Track in real-time), The Dynamic Chatbot in Medication/prescription inquiry. The use of artificial intelligence would mean that the app will be constantly updated and will incorporate the data of the user's current health issues to provide relevant advice and recommendations. Not only does this make the users more engaged, but it also has the added benefit of allowing people to better take care of their health and well-being in a manner that is quick and efficient from their portable electronic devices.

2.2 Product Functions

The "Personalized Health and Wellness Assistant" application will provide a range of functions designed to support comprehensive health management:

- **Collect and Analyze User Health Data:** The application will enable the user to feed it with several parameters, consisting of diet, exercise regime, sleep, and clinical measurements. Some of this data will be fed into A. I algorithms to adaptively analyze the user's health status and establish appropriate patterns, including mine, or correlations that will be useful in recommending certain products.
- **Provide Personalized Health and Wellness Advice:** Taking into account the data assumed, the application will also offer specific recommendations for improving health and acting in health-related areas that will be relevant to the user's profile of preferences. In this case, the advice will pertain to areas like diet, exercise, stress reduction, and general changes to the users' life to suit certain basic healthier lifestyle.
- **Track User Health Progress Over Time:** It will identify well-developed progress tracking elements to check users' compliance with the application-generated health plans and their effectiveness. The users will be able to enter their daily activity and daily goal and get the evaluation of their performance or of the observation of the improvements or the areas that deserve attention.
- **Offer Chatbot-Based Guidance for Medication and Prescriptions:** An integrated chatbot will give the user a pro-active perception into their medication and prescriptions. The chatbot will help the users easily get explanations concerning medication schedules, possible side effects, and other related concerns, given that it will be enjoined to the platform to ensure that they stick to their prescribed treatment plans.

2.3 User Classes and Characteristics

User classes can be created based on how frequently the product is employed, the specific part of the product's functionality which is utilized, the level of technical proficiency, security or access clearance, educational background, or past usage. Identify the main characteristics that are relevant for each of the described classes of users. This may apply certain requirements to the related classes of the users. Identify the user classes that are more vital for satisfying this product and separating them from the less important classes.

- ➢ **General Users:** It is noteworthy today that these people require individual counseling related to their health. Thus, they may differ significantly in terms of their technological skills ranging from informed computer users to no computer skills at all. Thus, their main concern is getting accurate and convenient health information that will suit their state.

- **Healthcare Professionals:** Patients that might need to complement the recommendations made by medical practitioners who might use the app in their practice. This user class is usually more trained on medical information and may need complex functions like seeing individual patient information or medical history; secure communication tools. Their comments may be useful for narrowing down the specifics of the product.

- **Technical Support:** Such people as programmers and users with technical knowledge are held responsible for maintenance of the app. This sector often possesses significant technical skill and is primarily interested in the app's operational efficiency, reliability, and capabilities. They may be involved in solving problems or addressing updates with the app or ensuring it is working properly.

2.4 Operating Environment

The application will be developed for portable devices, that is; the hardware platforms that support Android operating systems. It will be supported under the most current releases of operating systems, and several preceding releases for greater compatibility. This implies that the application will need to have a steady Internet connection that serves crucial operations such as data synchronization and working of the chatbot. Also, it has to share the same environment as other native applications on the device including health monitoring utilities, messaging services, and notifications.

2.5 Design and Implementation Constraints

The application development will be subject to several constraints:The application development will be subject to several constraints:

- **Regulatory Compliance:** The program should include the legal requirements as per confidential data regulation like GDPR and HIPAA, through which the app will have specific direction as to how user data is going to be collected, stored, and processed.
- **Development Framework:** The app will be developed using Kotlin to ensure it will run on the Android since it is one of the popular operating systems for smartphones. This choice restricts the usage of elements and optimizations specific to a particular platform.
- **AI and Machine Learning:** Tensor flow will be used in creating the AI models within the application; this brings the need to learn how TensorFlow works on the mobile frameworks.
- **Hardware Limitations:** The app must be adaptive to the diverse mobile devices that it is being installed in to avoid reaching the maximum physical memory or the processing power of the specific devise.
- **Security Considerations:** There is a need for effective security the most sensitive user information through techniques such as, encryption and online security protocols among others that addresses SECU10D.
- **Design and Programming Standards:** SDFE practices will be followed at the development phase of the solution to facilitate its maintainability and scalability for future upgrades by the customer's organization.

2.6 User Documentation

The user documentation will include the following components:

- **Comprehensive User Manual:** A detailed guide covering all aspects of the application's functionality, available in both digital and printable formats.
- **Online Help Guides:** Context-sensitive help accessible within the app, providing users with quick access to relevant information based on their current activity.
- **Tutorials:** Step-by-step tutorials designed to help users familiarize themselves with the app's features. These will be available both within the app and on the product website.

2.7 Assumptions and Dependencies

The project is based on the following assumptions and dependencies:

- **User Access:** It is assumed that all users will have access to mobile devices with reliable internet connectivity. The app's functionality heavily depends on this assumption, as many features require an active internet connection.
- **AI Framework Availability:** The project depends on the availability and continued support of a reliable AI framework, specifically TensorFlow, for implementing machine learning models within the app.
- **Mobile Development Framework:** The project relies on Kotlin as the mobile development framework for Android compatibility. Any significant changes or limitations in Kotlin could impact the development process and final product.
- **Third-Party Dependencies:** The project assumes that any third-party components or APIs used within the app will remain available, supported, and compatible throughout the development and operational phases.

3. External Interface Requirements

3.1 User Interfaces

The user interface (UI) will be designed to be intuitive and user-friendly, with a focus on a clean design and easy navigation. The UI will include the following components:

- **Dashboards for Health Metrics:** Users will have access to personalized dashboards displaying their health metrics, such as step count, heart rate, and other relevant data. The layout will be customizable to allow users to prioritize the information most important to them.
- **Chatbot Interface:** A conversational interface will be integrated for users to interact with the app's AI-driven chatbot. This interface will support both text and voice inputs and will include quick access to frequently used functions.
- **Settings for Personalization:** Users will be able to access and modify app settings, such as notification preferences, data sharing options, and personalized health goals. The settings interface will follow a standard layout for ease of use.

- **Standard Buttons and Functions:** The UI will include standard buttons like "Help," "Back," and "Home," which will be consistently placed across screens for easy access. Error messages will be displayed in a user-friendly manner, clearly explaining the issue and providing actionable steps to resolve it.
- **Screen Layout and Style Guides:** The design will adhere to established GUI standards and product family style guides to ensure a consistent and cohesive look and feel across the app. Screen layouts will be responsive, adapting to various screen sizes and orientations.

3.2 Hardware Interfaces

The application will interface with various hardware components of mobile devices to support its functionality. The key hardware interfaces include:

- **Data and Control Interactions:** The interaction between the software and hardware will involve continuous data collection, filtering, and processing. For instance, the accelerometer data will be used to monitor physical activity levels, while GPS data will track location for activity mapping and distance calculations.
- **Supported Device Types:** The app will support a wide range of mobile devices running Android operating systems. It will be compatible with both newer devices with advanced sensors and older models with basic sensor capabilities.
- **Communication Protocols:** The app will utilize standard mobile OS communication protocols, such as Bluetooth, for potential connectivity with external health devices (e.g., wearables). It will also adhere to the OS-specific APIs for sensor data access and processing.

3.3 Software Interfaces

The application will interface with the following software components:

- **Cloud-Based Databases:** The app will connect to cloud-based databases (e.g., Firebase, AWS DynamoDB, SQL) for data storage and retrieval. User data, such as health metrics, preferences, and activity logs, will be stored in these databases. The app will send and receive JSON-formatted data over secure HTTP/HTTPS protocols to ensure data integrity and security during transmission.
- **AI Models for Analysis:** The app will interface with AI models hosted in the cloud, implemented using TensorFlow. These models will process user data to provide personalized health recommendations and insights. The data sent to the AI models will include user activity metrics, health data, and other relevant information. The AI models will return analyzed data and predictions, which will be displayed within the app.
- **Operating Systems:** The app will run on Android operating systems, utilizing OS-specific APIs to access device sensors, manage data storage, and handle user interactions. It will also need to interact with system services like notifications and background processing.
- **Tools and Libraries:** The app will rely on various third-party libraries and tools for UI development and RESTful APIs for server communication. Integration with these tools will involve standard API calls and adherence to their usage guidelines.
- **Data Sharing and Communication:** Data shared across software components will be primarily user data and analysis results. The communication between the app and cloud

services will be stateless, relying on RESTful API calls. Authentication tokens will be used to ensure secure access to user data.

- **Implementation Constraints:** All data sharing between software components must adhere to security and privacy standards, including encryption of data in transit and at rest. The use of global data areas in multitasking environments will be avoided to prevent concurrency issues.

3.4 Communications Interfaces

The application will require the following communications functions:

- **Secure Communication Protocols:** The application will use HTTPS for secure communication between the client and server. This ensures that all data transmitted between the user's device and the cloud services is encrypted and protected from interception. The TLS protocol will be used to establish a secure connection.
- **Real-Time Data Synchronization:** Data synchronization will occur in real-time to ensure that users have access to the most up-to-date information. This will involve continuous background communication between the app and the cloud-based databases. WebSocket or HTTP/2 may be used to facilitate real-time communication and reduce latency.
- **Message Formatting:** Data exchanged between the client and server will be formatted in JSON, which is lightweight and easy to parse. This will be the standard format for all API requests and responses.
- **Communication Standards:** In addition to HTTPS, other communication standards may include OAuth 2.0 for secure user authentication and authorization. This will manage access to the user's data and ensure that only authorized clients can interact with the server.
- **Security and Encryption:** All communications will be encrypted using industry-standard encryption techniques. This includes encryption of data in transit using HTTPS/TLS and encryption of sensitive data at rest. The application will also implement measures to prevent common security threats, such as man-in-the-middle (MITM) attacks and data breaches.
- **Data Transfer Rates:** The application will optimize data transfer rates to ensure efficient use of bandwidth, particularly when syncing large volumes of health data. The communication protocol will be designed to handle variable network conditions, such as low bandwidth or intermittent connectivity.
- **Synchronization Mechanisms:** The app will implement synchronization mechanisms to ensure that data is consistently updated across all user devices. This will include conflict resolution strategies in case of concurrent updates and efficient data synchronization techniques to minimize redundant data transfers.

4. System Features

4.1 Personalized Health Advice

4.1.1 Description and Priority

- **Description and Priority:** This feature analyzes user data to provide tailored health and wellness advice. It is of **High** priority as it constitutes the core functionality of the application.
- **Priority Component Ratings:**
  - **Benefit:** 9 (High) - Directly impacts user engagement and satisfaction.
  - **Penalty:** 2 (Low) - Minor impact if not implemented.
  - **Cost:** 7 (High) - Significant development and maintenance costs.
  - **Risk:** 6 (Medium) - Potential risks related to data accuracy and privacy.

4.1.2 Stimulus/Response Sequences

- **User Action:** User inputs personal health information and preferences into the app.
  System Response: The system processes the input data and updates the user profile.
- **User Action:** User requests health advice via the app's dashboard or chatbot interface.
  System Response: The system analyzes the user data and provides personalized health recommendations based on the latest AI model outputs.
- **User Action:** User interacts with the advice provided, such as setting new health goals or providing feedback.
  System Response: The system updates recommendations and goals based on user interaction and feedback.

4.1.3 Functional Requirements

- **REQ-1:** The system must analyze user health data, including activity levels, biometrics, and user-reported symptoms, to generate personalized health advice.
- **REQ-2:** The system must ensure that personalized health advice is updated in real-time based on new user data and interactions.
- **REQ-3:** The system must handle invalid inputs or incomplete data by prompting the user for correct information and ensuring data integrity.
- **REQ-4:** The system must comply with data privacy regulations (e.g., GDPR, HIPAA) when processing and storing user data.
- **REQ-5:** The system must provide clear and actionable feedback when the advice cannot be generated due to insufficient data or other issues.

4.2 User Data Management

4.2.1 Description and Priority

- **Description and Priority:** This feature allows users to manage their personal health data, including updating, deleting, and viewing historical data. It is of **High** priority, as effective data management is crucial for user engagement and accuracy of health advice.
- **Priority Component Ratings:**
  - **Benefit:** 8 (High) - Enhances user control and engagement.
  - **Penalty:** 3 (Medium) - Moderate impact if not implemented.
  - **Cost:** 5 (Medium) - Moderate development effort.
  - **Risk:** 4 (Medium) - Risk related to data integrity and user privacy.

4.2.2 Stimulus/Response Sequences:

- **User Action:** User adds or updates personal health data (e.g., weight, activity levels).
  System Response: The system updates the user profile and stores the new data securely.
- **User Action:** User requests to view historical data.
  System Response: The system retrieves and displays historical health data in a user-friendly format.
- **User Action:** User deletes personal health data.
  System Response: The system removes the data from the database and confirms deletion to the user.

4.2.3 Functional Requirements:

- **REQ-1:** The system must allow users to add, update, and delete personal health data through a secure interface.
- **REQ-2:** The system must provide users with access to their historical data, ensuring that data retrieval is accurate and timely.
- **REQ-3:** The system must handle data deletion requests by removing data securely and updating all relevant records.
- **REQ-4:** The system must ensure data integrity and privacy during all operations.

4.3 User Authentication and Security

4.3.1 Description and Priority

- **Description and Priority:** This feature handles user authentication and security, including login, registration, and secure access to personal data. It is of **High** priority due to its critical role in protecting user data and ensuring secure access.
- **Priority Component Ratings:**
  - **Benefit:** 9 (High) - Essential for user trust and data protection.
  - **Penalty:** 9 (High) - High impact if security is compromised.
  - **Cost:** 6 (High) - Significant development and maintenance costs.
  - **Risk:** 8 (High) - High risk related to security vulnerabilities.

4.3.2 Stimulus/Response Sequences:

- **User Action:** User logs in to the app using credentials.
  System Response: The system verifies credentials and grants access if valid, or prompts for re-entry if invalid.
- **User Action:** User registers a new account or resets their password.
  System Response: The system processes the request, verifies identity, and updates account information securely.

4.3.3 Functional Requirements:

- **REQ-1:** The system must provide secure user authentication using encryption and secure protocols (e.g., OAuth 2.0).
- **REQ-2:** The system must support account registration, password recovery, and multi-factor authentication to enhance security.
- **REQ-3:** The system must protect user data through encryption and secure storage methods.
- **REQ-4:** The system must log security-related events for audit and monitoring purposes.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Response Time for Health Recommendations:** The appearance of the result based on the input of the user's data must be no longer than 2 seconds; the application should provide individualized health management advice. This requirement helps to guarantee that users get appropriate feedbacks and recommendations and consequently the effectiveness and the satisfaction in the functionality of the app. Any delay beyond 2 seconds can be less attractive or even discourage further use as users lose their interest.

- **Concurrent User Sessions:** The application should be able to handle 10,000 simultaneous users as there will be high user traffic. This requirement is necessary if a large number of users are expected as well as in order to guarantee that the application will perform well when it is most valued by users. The system should be designed as the scalable one in order it to distribute the loads among the servers as it can easily become a bottleneck.

- **Data Synchronization:** The application should ensure real-time response to the cloud engine or within 5 seconds of the data change by the users. This means that for any information that users need to work on, they can always be sure that they are working on the most updated version of that information regardless of the device they are using. Delayed synchronization of the data will lower the app's reliability of the recommended health decisions as there may be disparities between the local and original data held at Google.

- **API Response Time:** Incoming API calls related to external devices or health services are to respond within 3 seconds. It is crucial to keep the interaction between subsystems and external systems to the minimum, in terms of response time, so that this requirement is achieved.
- **System Load Handling:** The application must be able to process 500 TPS at the times of higher loads. This encompasses such tasks like data entry, recommendation or responses to users' inquiries and other features that should not compromise system speed.
- **Error Handling and Recovery:** The system has to be capable of self-healing to any minor mistakes or failures (ethernet breakouts) to take no more than 30 seconds of the user's time. In cases of error, users should be informed of the problem and offered decisions to make such as retry or get help.

5.2 Safety Requirements

- **Data Privacy and Compliance:** The application has to provide protection to the user data and that also adheres with the healthcare data standards like HIPAA and GDPR. This encompasses taking measures to ensure the users' data is secure from access or breaches on the side of the application, misuse, etc.
- **Secure Access Controls:** There are also security requirements that need to be met by the application such as; The application should have proper access controls to prevent any user from accessing other user's data. This involves such steps as utilizing MFA for accounts, including different types of users' accounts, and using RBAC for their access to resources, as well as data encryption in-transit and at-rest.
- **Data Loss Prevention:** The application must contain backups that refer to data protection in case of their loss, as well as data redundancy. Contingency plans and mechanisms must be able to restore data back to a functional state, so the impact on the users is minimal, especially in cases where important records of the patients' health are compromised.
- **Error Handling and Notifications:** The errors encountered in the application must be well managed and the possible notifications/ messages to the user should be well displayed. This entails the need to notify the users of any problem that is likely to affect their data or any interactions and as well directing them on how to report the problems or even solve the problems by themselves.
- **External Policies and Regulations:** The application must obey external safety standards and guidelines that shape the application's design and utilization. This implies following the standard practices concerning the security of software and data as well as any particular instructions of the health authorities or those regulating this sphere.
- **Safety Certifications:** The application must meet the safety certifications that include the ISO/IEC 27001 that deals with information security management, and the ISO/IEC 27701 Information protection management. These certifications show that the application of the safety and security levels is something that is highly valued by the application in question.

- **User Consent and Control:** The application must ensure that users are fully informed about how their data will be used and must obtain explicit consent before collecting or processing any personal health information. Users should also have control over their data, including the ability to review, update, and delete their information.

5.3 Security Requirements

- **Data Encryption:** The application must use encryption to protect data both at rest and in transit. This includes encrypting sensitive health data stored in the cloud and ensuring that data transmitted between the app and servers is encrypted using TLS/SSL protocols. The encryption standards must comply with industry best practices, such as AES-256 for data at rest and TLS 1.2 or higher for data in transit.
- **User Authentication:** User authentication must be required to access personal health data. The application must implement robust authentication mechanisms, including:
  - **Multi-Factor Authentication (MFA):** To provide an additional layer of security, MFA must be used for user login, requiring users to provide two or more verification factors.
  - **Password Policies:** Passwords must meet strong complexity requirements and be securely hashed using modern algorithms (e.g., bcrypt or Argon2).
- **Access Controls:** The application must enforce role-based access controls (RBAC) to ensure that users can only access data and functionalities that they are authorized to use. Different levels of access should be defined for general users, healthcare professionals, and administrative staff.
- **Data Protection Regulations:** The application must comply with relevant data protection regulations and standards, including:
  - **General Data Protection Regulation (GDPR):** For users in the European Union, ensuring that personal data is collected, processed, and stored in compliance with GDPR requirements.
  - **Health Insurance Portability and Accountability Act (HIPAA):** For handling protected health information (PHI) in the United States, ensuring compliance with HIPAA's privacy and security rules.
- **Security Certifications:** The application must obtain relevant security and privacy certifications, such as:
  - **ISO/IEC 27001:** For information security management, demonstrating a commitment to protecting information assets.
  - **ISO/IEC 27701:** For privacy information management, ensuring compliance with privacy regulations and best practices.
- **Data Integrity and Privacy:** The application must implement measures to ensure data integrity and privacy, including:
  - **Audit Trails:** Maintaining detailed logs of data access and modifications to support auditing and forensic investigations.
  - **Data Anonymization:** Where appropriate, data should be anonymized or pseudonymized to protect user identities in analytical or research contexts.
- **Incident Response and Reporting:** The application must include an incident response plan to address and mitigate security breaches or data leaks. Users and relevant authorities must be promptly notified of any significant security incidents affecting their data.

5.4 Software Quality Attributes

- **Usability:**
  - **Ease of Navigation:** The application must enable users to complete common tasks (e.g., viewing recommendations, entering data) within **3 clicks** or screen interactions.
  - **User Feedback:** The application must provide clear, context-sensitive feedback for user actions, such as confirming data entries or notifying errors, with **90%** of feedback delivered within **2 seconds** of user interaction.
  - **Learnability:** The app should have an average learning time of no more than **10 minutes** for new users to understand basic functionality.
- **Reliability:**
  - **Error Handling:** The application must handle errors gracefully, with **99%** of system errors resulting in user-friendly error messages rather than application crashes.
  - **System Availability:** The application must achieve an uptime of **99.9%** over any 30-day period, excluding scheduled maintenance.
  - **Failure Recovery:** The system must recover from failures or crashes within **5 minutes**, ensuring minimal disruption to users.
- **Maintainability:**
  - **Modularity:** The application code must be modular, with no more than **10%** of code dependencies crossing module boundaries to facilitate independent updates and maintenance.
  - **Documentation:** The codebase must be documented with comments covering **95%** of functions and modules, providing clear explanations of purpose, inputs, and outputs.
  - **Update Frequency:** The system should support updates with **minimal downtime** (less than **15 minutes**) and without requiring significant user intervention.
- **Portability:**
  - **Android Compatibility:** The application must function consistently across **Android 11 and above**, with UI and performance adhering to platform-specific guidelines.
  - **Installation Size:** The application must have an installation size of **less than 100 MB** to ensure it is feasible for users with limited storage.
- **Interoperability:**
  - **Third-Party Integration:** The application must integrate with major third-party health devices and services using standard protocols (e.g., Bluetooth, APIs), with **95%** compatibility with devices that meet specified criteria.
- **Robustness:**
  - **Data Validation:** The application must validate all user inputs with **99%** accuracy, rejecting invalid inputs and providing corrective guidance to users.
  - **Stress Testing:** The application must handle up to **10,000 concurrent users** without significant degradation in performance.
- **Testability:**
  - **Automated Testing:** The application must support automated testing for **90%** of critical functionalities, including unit tests, integration tests, and regression tests.

- **Test Coverage:** The codebase should achieve a test coverage of **85%** or higher, ensuring comprehensive validation of application features.

5.5 Business Rules

- **Health Recommendations Only:** The app provides health recommendations but does not offer medical advice. It should include disclaimers clarifying this limitation.
- **User Agreement Required:** Users must agree to the terms of service and privacy policy before accessing the app. This agreement must be accepted during registration or first-time login.
- **Role-Based Access:**
  - **General Users:** Access personal health recommendations and manage their data.
  - **Healthcare Professionals:** Access additional features for detailed analytics and patient recommendations.
  - **Administrative Staff:** Manage user accounts and system maintenance with restricted data access based on role.
- **Data Privacy Compliance:** Ensure compliance with data privacy regulations (e.g., GDPR, HIPAA) by securing user consent and managing data access.
- **User Data Access:** Users can view, update, and manage their personal health data.
- **Incident Reporting:** Users can report issues or security concerns via a built-in reporting mechanism.
- **Content Moderation:** User-generated content must be moderated to prevent inappropriate or harmful material.

6. Other Requirements

- **Scalability:** The application must be designed to scale effectively, accommodating a growing user base. This includes supporting horizontal scaling of the backend infrastructure to handle increased traffic and data load without performance degradation.
- **Internationalization:** The application must support multiple languages to cater to a global audience. This includes providing user interfaces, health recommendations, and support materials in various languages, with localization tailored to regional norms and conventions.
- **Database Requirements:**
  - **Data Storage:** The application must use a scalable cloud-based database solution to handle user data and health information efficiently.
  - **Data Backup:** Regular data backups must be performed to prevent data loss and ensure recovery in case of failures.
- **Legal Requirements:** The application must comply with relevant laws and regulations related to data protection and privacy (e.g., GDPR, HIPAA). This includes implementing necessary legal safeguards and ensuring that user consent and data handling practices meet legal standards.

- **Reuse Objectives:** Where possible, the project should aim to reuse existing components, frameworks, or libraries to reduce development time and leverage proven technologies. This includes reusing open-source libraries or APIs for standard functionalities.
- **Accessibility:** The application should comply with accessibility standards (e.g., WCAG) to ensure it is usable by individuals with disabilities. This includes providing features like screen reader support, keyboard navigation, and customizable text sizes.
- **Performance Monitoring:** The application must include performance monitoring tools to track system performance, user interactions, and potential issues. This data will help in optimizing the app and maintaining a high-quality user experience.

Appendix A: Glossary

- **AI (Artificial Intelligence):** The human aspects such as learning, reasoning, and problem solving being emulated on a computer or any other machine.
- **Chatbot:** A programmed system which can engage in dialog with the users through text and/or voice input and output to deliver information or aid.
- **GDPR (General Data Protection Regulation):** A regulation in EU law pertaining to data protection and privacy of all EU citizens from the EU and EEA.
- **HIPAA (Health Insurance Portability and Accountability Act):** A legal instrument in the United States of America for the purpose of offering legal standards of privacy concerning patients' medical records and other health information submitted to health plans, clinicians, hospitals among others.
- **ISO/IEC 27001:** An ISMS standard that provides guidance on how to develop, implement, monitor and update a company's ISMS.
- **ISO/IEC 27701:** It is another international standard for the management of Privacy Information it generally describes how to initiate, carry out, analyze, and enhance the organization's Privacy Information Management System (PIMS).
- **MFA (Multi-Factor Authentication):** Two or more identification pieces that are offered by the user in order to have access to a certain resource in order to increase the security of an account.
- **RBAC (Role-Based Access Control):** A process of reducing access of an organization system or applications to only those personnel who are supposed to do so depending on their rank.
- **TLS (Transport Layer Security):** You've got an SSL that is intended to offer communications security in a computer network, succeeding the earlier SSL or Secure Sockets Layer.
- **UI (User Interface):** It refers to how a user can use a software application and or how elements like buttons, menus and screens are created.
- **WCAG (Web Content Accessibility Guidelines):** Guidelines developed to ensure web content is accessible to people with disabilities, including recommendations for making web content more perceivable, operable, understandable, and robust.
- **TensorFlow:** An open-source machine learning model that has been designed, developed and implemented by Google used in designing, training, and deploying of the machine learning models.

Appendix B: Analysis Models

No analysis models as of now

Appendix C: To Be Determined List

No TBD references as of now