“SuppleMate” App
Software Requirements Specification (SRS)

Team:

- Gregy Thomas Kokkaparampil – Project Manager
- Shuo Li– Assistant Project Manager/ Software Engineer
- Siddharth Santhanakrishnan – Software Engineer
- Wone Eui (Ella) Jung – Software Engineer

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirement Specification is designed to identify the requirements of SuppleMate Application. Requirements listed in this document are based on the discussion among team members of Flowlab. These requirements are expected to be implemented by the end of Fall quarter 2023 Software Engineering Course.

1.2 Scope

The SuppleMate is designed to help users manage their dietary supplements and vitamins effectively. It allows users to input and track their supplement intake, compute the exact dosage required each time, set dosage reminders, and receive personalized nutritional/ supplement recommendations based on their health goals and nutritional needs. The app provides insights into the benefits of each supplement, tracks expiration dates, and offers a comprehensive overview of the user's overall supplement regimen.

- Prototype 1: The first prototype will build an application that has a functional user interface that allows users to build a new profile, or log in to their account. A functional homepage where users can enter and see their nutrition details. A functional reminder which can remind users to take their supplement on time with the right dose. This will be our M.V.P.
    - In its prototype stage, SuppleMate will primarily assist health enthusiasts and fitness enthusiasts who already incorporate supplements into their dietary regimen in managing their supplement routine more effectively.

- Prototype 2: The second prototype will provide a recommendation function that users can get personalized supplement recommendations based on individual health goals and dietary needs.
    - In Prototype 2, while offering recommendations, it's important to emphasize that users will always receive a disclaimer advising them to consult a healthcare professional before making any significant lifestyle changes.

1.3 Definitions, Acronyms, and Abbreviations

Programming Language: A vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

SRS: This document, the software requirements specification (SRS) document lays out the functional and non-functional requirements of the SuppleMate App.

UML: Unified Modeling Language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system.

Use case diagram: A representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

User: Someone who uses the application or software specified in this software requirements specification.

MVP: A Minimum Viable Product (MVP) is the most basic version of a product that contains only essential features to meet the needs of early users. It is designed to gather feedback, validate ideas, and quickly bring a functional product to the market.

1.4 References

[1] IEEE Software Engineering Standard Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

1.5 Overview

The following sections offer a deeper insight into the requirements for the SuppleMate application.

The second section of this document furnishes an overarching description of the application's functionality. It encompasses details about system interfaces, core functions, user profiles, and any operational constraints.

In the third section of this document, we delve into the specific requirements in detail. This section covers interface specifications, performance benchmarks, design constraints, and system attributes to provide a comprehensive understanding of the application's functionalities and capabilities.

2. Overall Description

2.1 Product Perspectives

2.1.1 System interfaces (deployment diagram)

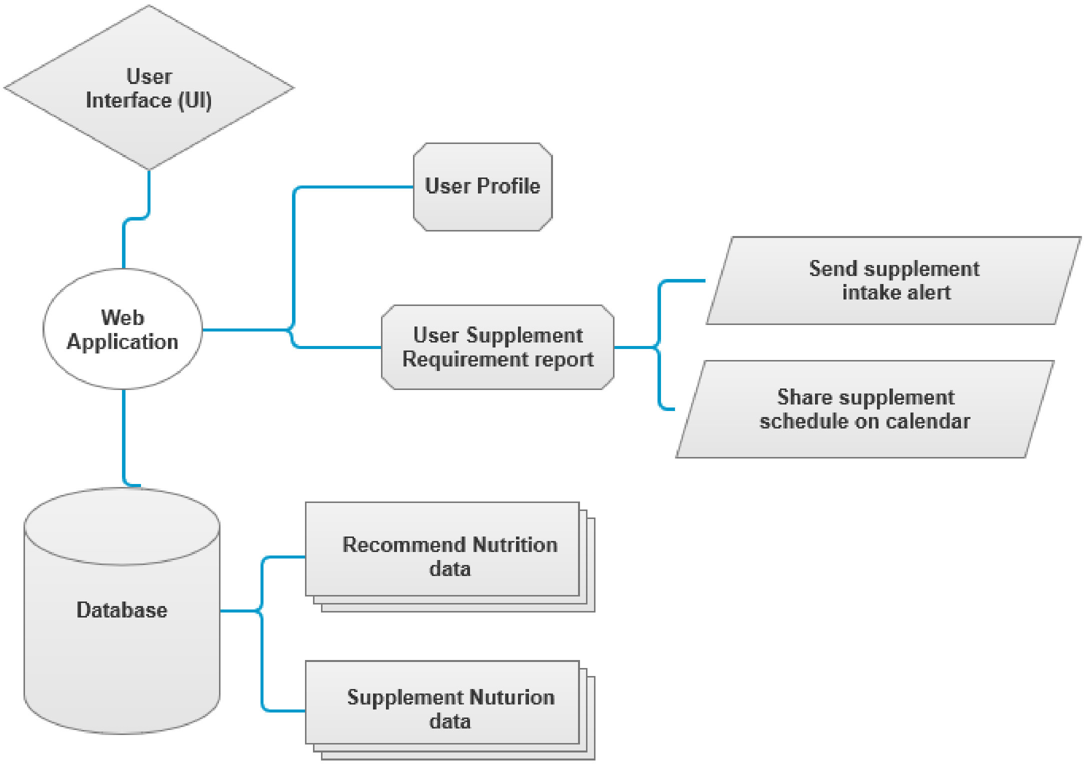


Figure 1. System Architecture

2.1.2 User interfaces

- For a first-time user setting up their supplement plan, SuppleMate offers a user-friendly onboarding experience. Users begin by creating an account or logging in, followed by the entry of personal details, which include factors like age, height, and weight to calculate their BMI. Subsequently, users input their “Current health profile”, detailing vitamin and nutrient levels. The app then compares their health profile to a reference "Ideal Health Profile" for their BMI group, pinpointing disparities. Users are then prompted to assess their current supplement intake, which can be selected from a provided list or manually entered. Throughout this process, important disclaimers underscore the significance of consulting healthcare professionals when necessary, ensuring a safe and effective supplement plan.

- On consecutive logins, users of SuppleMate will be greeted by a familiar and intuitive interface. The home page provides a snapshot of their supplement plan, displaying current supplements, the intake schedule, and personalized recommendations. Users can seamlessly navigate to their profile, offering options to edit preferences, manage supplements, view usage history, access supplements usage reports, and set reminders for consistent intake. The analytics dashboard delivers insightful data visualization, tracking supplement and nutrient intake, as well as overall health progress, ensuring informed choices and a user-centric experience.
- The front-end of the web app is responsible for user interaction.
- Use Figma to create a responsive and user-friendly interface.
- Utilize a front-end framework or library like React, Angular, or Vue.js for building a dynamic and interactive UI.

2.1.3 Software interfaces

There are two major software interfaces that are going to be used in this project.

1) Web Server:

Web server serves as the entry point for client requests in an app system, directing those requests to the appropriate components within the application, handling security and encryption, and improving performance through various mechanisms like caching and load balancing. It plays a pivotal role in ensuring the availability, reliability, and security of web applications.

- Deploy a web server to serve static assets and handle client-side requests.
- Our options include Nginx, Apache, or a CDN (Content Delivery Network) for static content distribution.

2) Application Server:

The Application Server in a mobile app architecture acts as the backend component responsible for handling data, business logic, security, and communication with external services. It serves as the intermediary between the mobile app's user interface and the data sources and services needed to provide a seamless and functional user experience.

- The application server (Heroku) handles the business logic, user authentication, and API requests.
- Use a server-side language or framework like Node.js, Ruby on Rails, Django, or Express.js.
- Implement RESTful or GraphQL APIs to communicate with the front-end.

Web servers and Application servers are different components that work together in a typical web application architecture to provide static content delivery and dynamic application logic processing, respectively.

2.1.4 Communication interfaces

In SuppleMate, several communication interfaces using APIs are established within the system and external systems to obtain nutritional and statistical data. Here's an overview of these interfaces:

- **Nutritional Databases:** The Institute of Medicine has determined upper limits for 24 nutrients
- **Supplement Manufacturers website:** Collects product details directly from manufacturers.

- **User Authentication and Authorization APIs:** Ensures secure user access.
- **Push Notification Services:** Sends reminders and alerts.
- **Geolocation Services (in future App expansion):** Provides location-based recommendations.
- **E-commerce Integration (in future App expansion):** Facilitates online supplement purchases.
- **Wearable Devices and Health App Integration (in future App expansion):** Syncs user health data from wearables and health apps.

2.1.5 Memory

- Store user profiles, preferences, supplement information and supplement intake history.
- .MySQL is an open-source relational database management system (RDBMS) that enables users to store, manage, and retrieve structured data efficiently. It is widely used for various applications, from small-scale to large-scale data .
- Ensure data integrity and implement proper indexing for efficient queries.

2.1.6 Operation

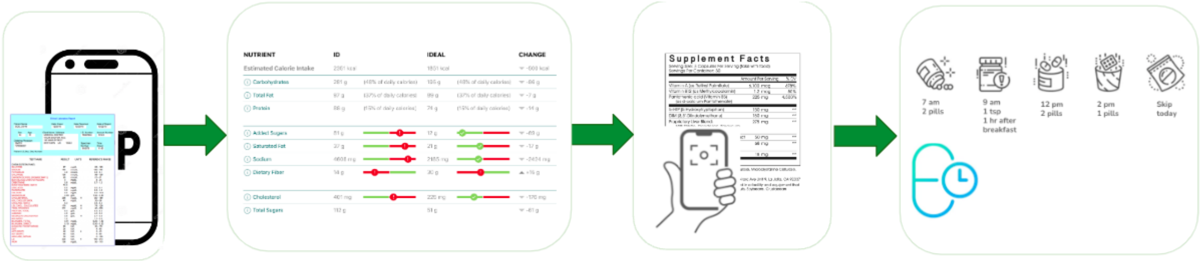
Users login to the application via Authentication and feed the current nutritional status into the user interface, then the data is sent to the application server through RESTful API. When the report request is sent the model-based reports are created from the MongoDB-based server. When the user updates information the server processes the data and saves it in the database.

2.1.7 Site adaptation requirements

SuppleMate app is primarily a Web App. As the app matures, it will be designed to work on multiple platforms or devices, such as mobile, or desktop. These requirements ensure that SuppleMate functions optimally and looks appropriate on different platforms. Here are some potential site adaptation requirements:

- Mobile responsiveness for various screen sizes.
- A responsive web version for different browsers and devices.
- Cross-browser compatibility.
- Platform compatibility (Web).
- Accessibility standards compliance.
- Localization for international users.
- Performance optimization.
- Security considerations.
- Consistent UI/UX.
- Well-functioning push notifications.
- Seamless integration with external services.
- Rigorous testing on all target platforms to address specific issues.

2.2 Product Functions



User feed their current nutritional concern/ status into the app manually

The app runs the user fed data against medically approved standards (or their personal goals[1]), and computes the change required at a macro and micro nutrient leve (image used just for demo)

If the user have existing supplements, they can feed the dosage facts by:

- selecting from the list of supplements in the app
- Or manually entering

The app using AI/rule based approach creates a supplement routine/schedule for the user .. with other features like

- reminders
- dosage recommendations

Figure 2. Product functions

- **Manage user accounts:** This use case allows the administrator to create and manage user accounts.
- **Manage supplements:** This use case allows the administrator to add and remove supplements, and to manage the supplement information.
- **Track supplement usage:** This use case allows the user to track their supplement usage, such as the supplement name, dosage amount, and scheduling information.
- **Set supplement reminder:** This use case allows the user to set reminders for themselves to take their supplements.
- **View supplement usage data:** This use case allows the healthcare provider to view the user's supplement usage data. MVP will not cover healthcare provider integration!
- **Provide supplement recommendations:** This use case allows the healthcare provider to provide recommendations to the user about their supplement usage. MVP will not cover healthcare provider integration!
- **Generate supplement usage report**: This use case allows the user to generate reports on their supplement usage, such as daily, weekly, and monthly reports. This is not an MVP feature.

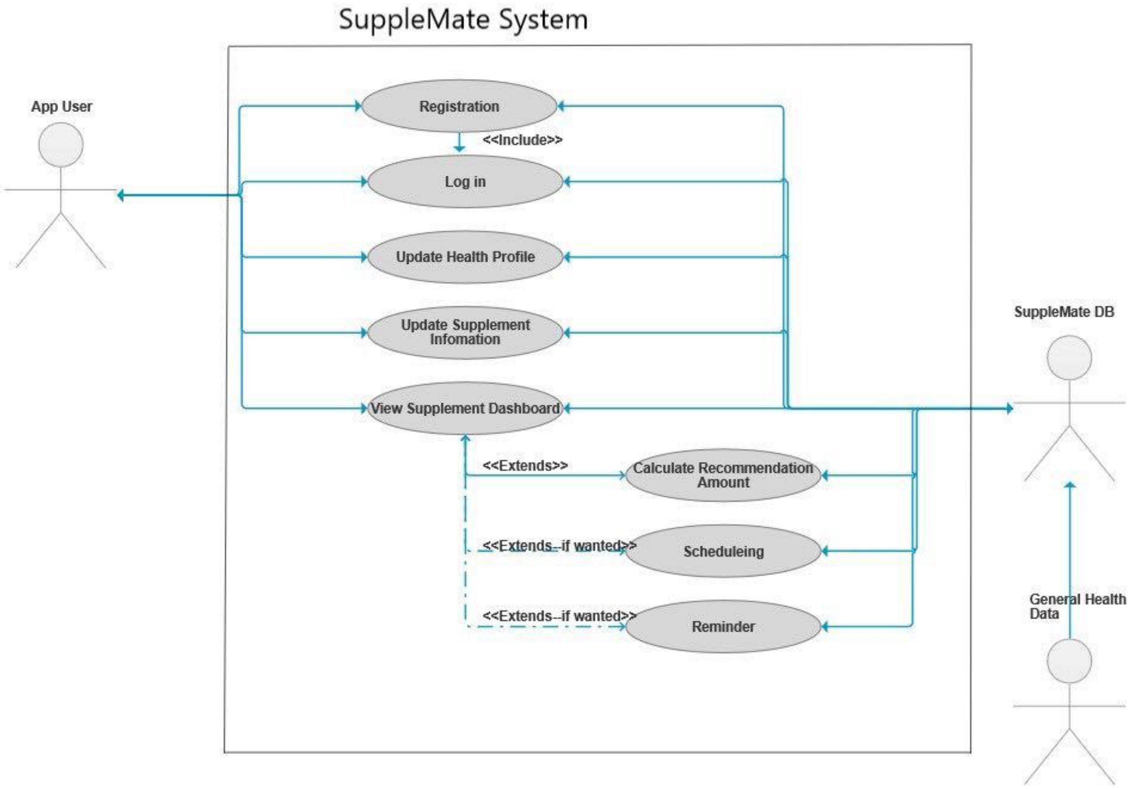


Figure 3. User Case diagram

2.3 User Characteristics

- **User:** This is the primary actor, and is the one who uses the system to track their supplements. Some key primary users of SuppleMate are:
    - Health Enthusiasts: People who proactively manage their health and well-being through nutritional supplements and vitamins.
    - Athletes and Fitness Enthusiasts: Individuals engaged in sports and fitness activities who use supplements to support their training and recovery.
    - Seniors: Older adults who may rely on supplements for improved health and vitality.
    - People with Specific Dietary Requirements: Individuals with dietary restrictions or specific nutritional needs, such as vegetarians or vegans.

    - General Wellness Seekers: People who take vitamins and supplements for overall health and wellness.

Note: SuppleMate is designed to cater primarily to users who maintain an existing supplement regimen and should not be utilized for self-diagnosis. This application is not intended for individuals falling into the following categories:

- **Patients:** This includes individuals under the care of healthcare professionals who are prescribed specific medications or supplements as part of their treatment plan.
- **Individuals with Chronic Conditions:** Supplemate is not intended for individuals managing chronic health conditions that require specific supplement regimens.

Disclaimer:
SuppleMate is a supplement management tool and is not a substitute for professional medical advice, diagnosis, or treatment. Individuals with specific medical needs, patients, and those managing chronic conditions must consult their healthcare provider or medical specialist for personalized guidance on supplement intake and treatment plans. While SuppleMate can assist in managing supplement regimens, it is not a self-diagnostic or treatment tool for medical conditions. Always prioritize professional healthcare guidance for your specific health needs.

2.4 Constraints

- **Platform and Technology Constraints:**
    - Supported platforms (primarily web).

  - Programming languages and frameworks (e.g., Java, React Native).
  - Compatibility with screen sizes.
  - Integration with external systems or APIs.
- **Performance Constraints:**
  - Maximum response time for user interactions.
  - Minimum and maximum system load and capacity.
  - Targeted performance metrics, such as latency or throughput.
- **Security Constraints:**
  - Compliance with data protection laws (e.g., GDPR).
  - Encryption requirements for data in transit and at rest.
  - User authentication and authorization mechanisms.
  - Data access controls and permissions.
- **Regulatory Constraints:**
  - Compliance with regulatory agency requirements for supplement tracking (if applicable).
  - Requirements for storing and handling user health data.
- **Data Constraints:**
  - Data storage and retention policies.
  - Data synchronization across devices.
  - Database technology and constraints (e.g., MySQL).
- **User Interface Constraints:**
  - Accessibility requirements for users with disabilities.
  - Design and layout guidelines for consistent user experience.
- **Usability Constraints:**
  - Target user demographics and skill levels.
  - Localization and internationalization requirements.
  - User training and onboarding constraints.
- **Legal and Copyright Constraints:**

    - Intellectual property rights for the application and its content.
    - Compliance with copyright and trademark laws.
- **Resource Constraints:**
    - Availability of server resources (e.g., hosting, cloud services).
    - Network bandwidth limitations.
    - Hardware and software dependencies.
- **Budget and Cost Constraints:**
    - Budget limitations for development and maintenance.
    - Constraints on third-party services or licenses.
- **Time Constraints:**
    - Project timeline and deadlines.
    - Phases or milestones for development and testing.
- **Scalability Constraints:**
    - Expected user growth and scalability requirements.
    - Ability to handle increased data and user load.
- **Third-Party Integration Constraints:**
    - Dependencies on third-party APIs or services.
    - API rate limits and usage constraints.
- **Operational Constraints:**
    - System availability and maintenance windows.
    - Backup and disaster recovery plans.
- **Testing and Quality Assurance Constraints:**
    - Test environments and data constraints.
    - Testing methodologies and frameworks.
- **Documentation Constraints:**
    - Requirements for user manuals, technical documentation, and support materials.
- **Support and Maintenance Constraints:**
    - Support hours and response time constraints.
    - Software update and patch release schedule.
- **Compliance Constraints:**

    - Industry-specific compliance requirements (e.g., HIPAA for healthcare apps).
    - Adherence to specific coding standards or best practices.
- **Environmental Constraints:**
    - Environmental conditions for hardware or IoT integration (e.g., temperature or humidity).
- **User Privacy Constraints:**
    - Privacy policies and regulations governing user data collection and usage.
    - When documenting constraints in an SRS, it's crucial to be specific and ensure that they align with the project's goals and constraints. These constraints will guide the development process and help ensure that the final SuppleMate Application meets its intended purpose within specified limitations.

2.5 Assumptions and Dependencies

Assumptions clarify conditions or expectations taken for granted during the project, such as hardware availability or user behavior. Some key assumptions are:

- Hardware Availability: Assumptions regarding the availability and compatibility of required hardware components or devices for the software to function correctly.
- Software Dependencies: Assumptions about the presence and compatibility of other software, libraries, or frameworks necessary for the operation of the system.
- User Behavior: Assumptions about how users will interact with the software, including their technical knowledge, frequency of use, and adherence to guidelines.

- Data Integrity: Assumptions about the quality and reliability of input data, including data provided by external systems or users.
- Stakeholder Availability: Assumptions regarding the availability of key stakeholders for requirements gathering, reviews, and decision-making.
- Regulatory Compliance: Assumptions about the software's compliance with industry standards, legal regulations, or specific certifications.
- Third-Party Services: Assumptions about the availability and reliability of third-party services or APIs used by the software.
- Network Conditions: Assumptions about network connectivity and conditions, such as latency, bandwidth, and reliability.

Dependencies are external factors that can affect the project, such as reliance on specific software components, third-party services, or team resources.

- Hardware Dependencies: Document dependencies on specific hardware components, configurations, or infrastructure.
- Software Dependencies: List any external software components, libraries, or frameworks on which the project relies.
- Third-Party Services: Identify dependencies on third-party services, including service availability and API specifications.
- Data Sources: Specify dependencies on external data sources, databases, or data providers.
- Regulatory Dependencies: Document dependencies on compliance with legal or industry-specific regulations.
- .. etc

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

Some key user interfaces are

- **Registration and Onboarding:**
    - This interface allows new users to create accounts, set up profiles, and provide initial information about their health goals and supplement preferences.
- **User Dashboard:**
    - The dashboard serves as the central hub, providing users with a summary of their supplement intake and reminders. In future app releases, it will also include personalized recommendations.
    - Users can manage their profiles, update personal information, set privacy preferences, and configure app settings, including notification preferences and data sharing.
- **Supplement Dashboard:**
    - Users can enter information about their supplements, which may encompass brand, dosage, and intake frequency. This interface offers both manual input options and potential enhancements for more automated data entry methods in forthcoming app updates.
    - This interface allows users to search for supplements and access information from external databases, such as The Institute of Medicine or nutritional databases.
- **"For You" Dashboard**
    - Dosage Reminders: Users can set and manage dosage reminders, ensuring they take their

    supplements at the right times. This interface includes scheduling and notification settings.
  - Personalized recommendations (Future feature): This interface displays personalized recommendations based on a user's health goals and nutritional needs. Users can explore and accept or reject these recommendations.
  - Analytics and Insights (Future feature): Users can view analytics and insights about their supplement intake, helping them make informed decisions about their regimen.
- **Support and Help Center:**
  - This section provides access to FAQs, support resources, and a help center to assist users with any questions or issues.

3.1.3 Software interfaces

- **MongoDB:** defines how software interacts with databases, allowing applications to retrieve, store, and manipulate data stored in stored procedures. When a report is requested this will grab correct information and calculate the result.
- **Command Line Interface (CLI):** A CLI is a text-based interface that allows users to interact with a software application by entering text-based commands.
- **Authentication:** When a user accesses certain pages this will check if the user is authenticated or not, after authentication this will redirect the user to the login page.

3.1.4 Communication interfaces

Communication interfaces for SuppleMate play a crucial role in connecting the app with external data sources, APIs, and services. Here are some of the key communication interfaces for SuppleMate:

- Institute of Medicine Nutrition guideline: SuppleMate has essential data about intake guidelines for dietary supplements
- Supplement product data: SuppleMate retrieves specific product information, including dosage recommendations, ingredient lists.
- User Authentication and Authorization APIs: These interfaces manage user accounts, authentication, and authorization, ensuring secure user access to their data and settings.
- Geolocation Services (Future feature): In the future when the app offers location-based recommendations, it communicates with geolocation services to suggest nearby supplement stores, clinics, or wellness centers.
- E-commerce Integration (Future feature): For users who wish to purchase supplements through the app, it communicates with e-commerce platforms or APIs to enable secure and convenient online shopping.
- Wearable Devices and Health App Integration (Future feature): If SuppleMate integrates with wearable devices or health apps, it establishes communication interfaces to sync user health data, such as exercise and biometric information.

These communication interfaces enable SuppleMate to access external data sources, enhance user experiences, provide real-time information, and offer personalized features while ensuring data security and regulatory compliance.

3.2 Functional requirements

- The system must be able to track a wide variety of supplements, including their ingredients, dosage amounts, and scheduling information.
- The system must be able to set reminders for users to take their supplements.
- The system must be easy to use and navigate.
- The system must be secure and protect user data.

3.2.1 Mode 1: User Onboarding and Support Mode

- **User Registration:** Users should be able to create accounts with unique usernames and passwords.
- **Profile Setup**: Users need to set up their profiles, including personal information (name, age, gender, etc.).
- **Help Center:** Users can access a help center with FAQs and resources.
- **Customer Support**: Provide a way for users to contact customer support for assistance.

3.2.2 Mode 2: Supplement Tracking Mode

- **Supplement Input:** Users should be able to add supplements to their regimen, including product name, brand, dosage, and frequency.
- **Editing and Deletion:** Allow users to edit or delete supplements from their regimen.

3.2.3 Mode 3: Reminder/ Recommendation Mode

- **Dosage Reminders:** App should auto generated or if preferred users should be able set up dosage reminders, specifying the time and frequency for each supplement.

3.2.4 Mode 4: Insight and Analytics Mode

- **Database Search:** Implement a search feature that allows users to look up supplements and access detailed information from external sources like the Institute of Medicine or nutritional databases.

3.2.5 Mode 5: Safety Mode

- **Data Encryption:** Ensure secure storage and transmission of user data, especially personal health information.
- **Privacy Preferences:** Allow users to configure data sharing and notification settings in line with their preferences.

3.3 Performance requirements

- **Response Time:** The application shall provide a response time of less than 2 seconds for all user interactions.
- **Concurrent Users:** The system should support a minimum of 1000 concurrent users without performance degradation.
- **Data Throughput:** The application should handle a minimum of 1000 supplement entries per minute.

3.4 Design constraints

- **Cross-Platform Compatibility:** The application must be designed to work on any web and should be platform agnostic
- **Device Compatibility:** The app should be compatible with smartphones and tablets, with a minimum Android version of 5.0 and iOS version 11.
- **Database Integration:** The system should integrate with a specific database system (e.g., MySQL) for data storage.

- **Security Compliance:** The application must adhere to specific security standards (e.g., HIPAA, GDPR) if applicable.
- **Scalability:** The system should be designed to easily scale to accommodate future feature additions and a larger user base.

3.5 Software system attributes

- **Reliability:** The application shall have a 99.99% uptime, ensuring users can access their data without interruption.
- **Security:** User data shall be securely encrypted both in transit and at rest. User authentication shall follow industry best practices.
- **Data Backup and Recovery:** Regular automated data backups shall be performed, with the ability to recover user data in case of a system failure.
- **Usability:** The system should have an intuitive user interface with accessibility features, and it must be designed following user experience (UX) best practices.
- **Maintainability:** The system should be designed to facilitate updates, bug fixes, and feature additions without major disruptions to the user base.
- **Scalability:** The system should be scalable to accommodate a growing user base and data volume.

3.6 Other requirements

- **Data Privacy:** The application should have a comprehensive privacy policy, and users should have control over their data.

- **Notifications:** The app should provide users with optional reminders for supplement intake.

- **User Support:** Provide user support through email, in-app chat, or a dedicated support portal.