

SuppleMate



Flowlab • 12/05/2023





Idea

In a world where an increasing number of people rely on supplements to meet their specific health and wellness needs, the SuppleMate app emerges as a comprehensive solution. This project aims to address the diverse supplement requirements of individuals, offering a user-friendly platform to monitor, access, and optimize their supplement intake.



Competitive Landscape

MyFitnessPal:

- Strengths: A widely popular fitness and nutrition app with a large user base, it allows users to track not only supplements but also exercise, diet, and other health metrics.
- Weaknesses: Limited focus on supplements and less personalized guidance.

Cronometer:

- Strengths: Provides detailed nutrient tracking and micronutrient information, which is valuable for supplement users concerned about specific nutrients.
- Weaknesses: Less user-friendly for those primarily interested in supplement management.

Care/of:

- Strengths: Offers personalized supplement recommendations and delivers customized supplement packs to users' doorsteps.
- Weaknesses: Primarily a supplement retail and delivery service, which may not suit users who want to track existing supplement regimens.



Our Specialty

- Image reading: Use photos of physical exam results to build the user's profile based on the results, and the user's goals, the app will give the recommendation of supplements.
- Real world data: Using the real world data (the choice of other users with similar needs) and informing users to make plans.
- Supplement tracking: Reminder for users to take their supplements, with suggested dose and frequency. Track expiration date of supplements, and remind users to replace them.
- Integrations: Integrate with pharmacy or shopping apps, to subscribe and get autoship before supplements run out.



Scope

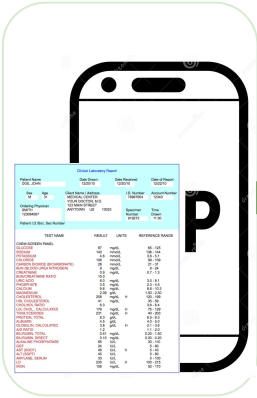
Prototype 1:

- Functional user interface
- Functional homepage
- Functional reminder

Prototype 2:

- Personalized supplement recommendations

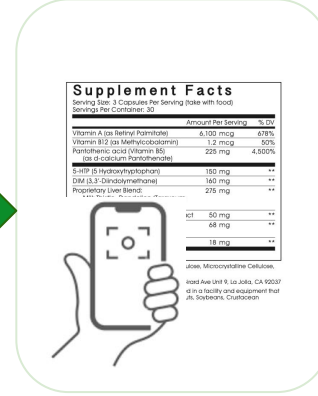
Our expectation



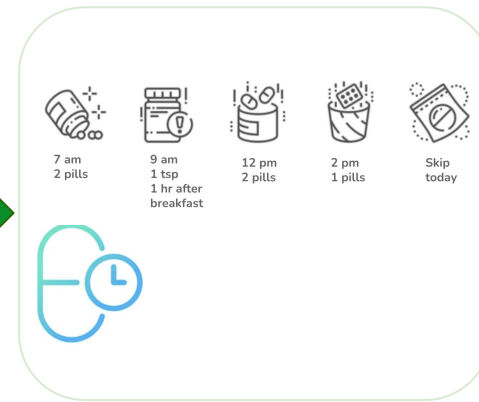
Input user's current nutritional concern/status

NUTRIENT	ID	IDEAL	CHANGE
Estimated Calorie Intake	2351 kcal	1851 kcal	▼ -500 kcal
① Carbohydrates	281 g (48% of daily calories)	195 g (48% of daily calories)	▼ -86 g
① Total Fat	97 g (37% of daily calories)	89 g (37% of daily calories)	▼ -7 g
① Protein	88 g (15% of daily calories)	74 g (15% of daily calories)	▼ -14 g
① Added Sugars	81 g	12 g	▼ -69 g
① Saturated Fat	37 g	21 g	▼ -17 g
① Sodium	4608 mg	2185 mg	▼ -2424 mg
① Dietary Fiber	14 g	30 g	▲ +16 g
① Cholesterol	401 mg	225 mg	▼ -176 mg
① Total Sugars	112 g	51 g	▼ -61 g

SuppleMate computes the required change for the user



Input user's existing supplements dosage facts



Creates a supplement routine/schedule for the user



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.

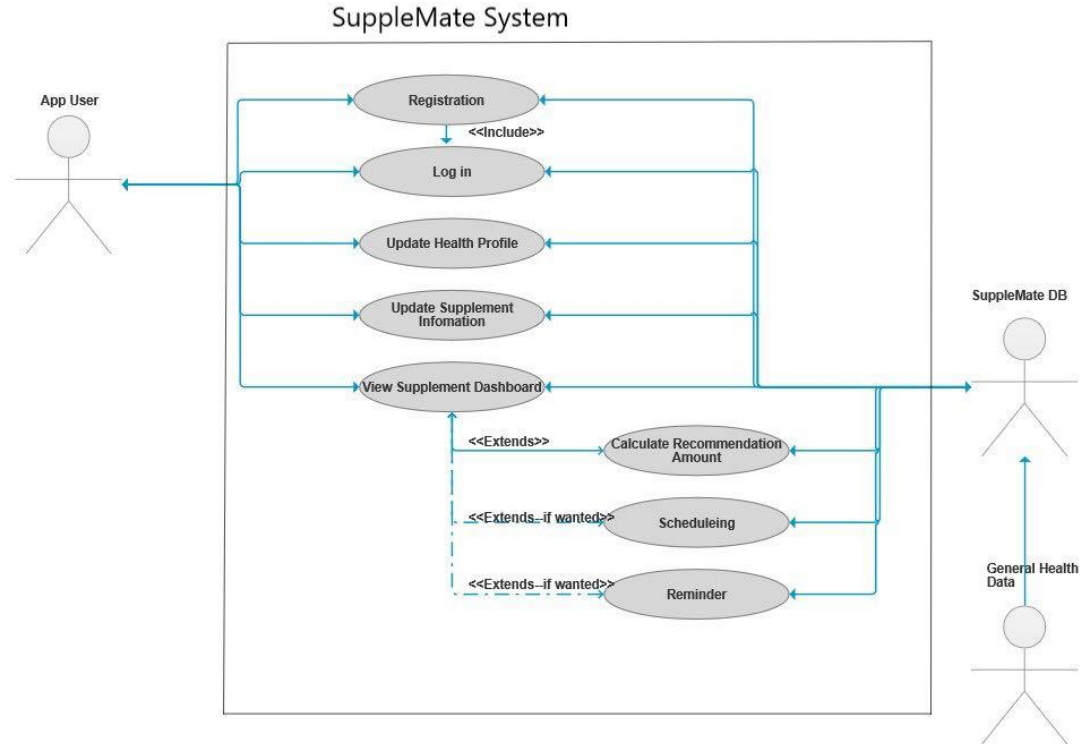


Non-Functional Requirments

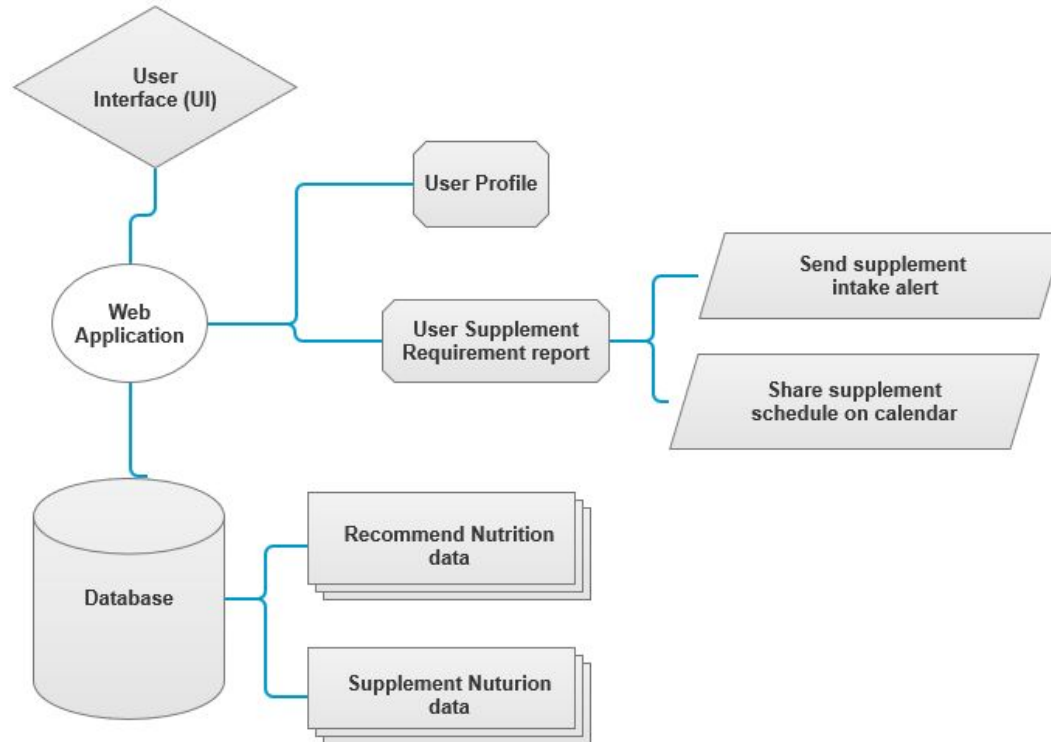
- **Response Time:** The application shall provide a response time of less than 2 seconds for all user interactions.
- **Concurrent Users:** The system should support a minimum of 1000 concurrent users without performance degradation.
- **Data Throughput:** The application should handle a minimum of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



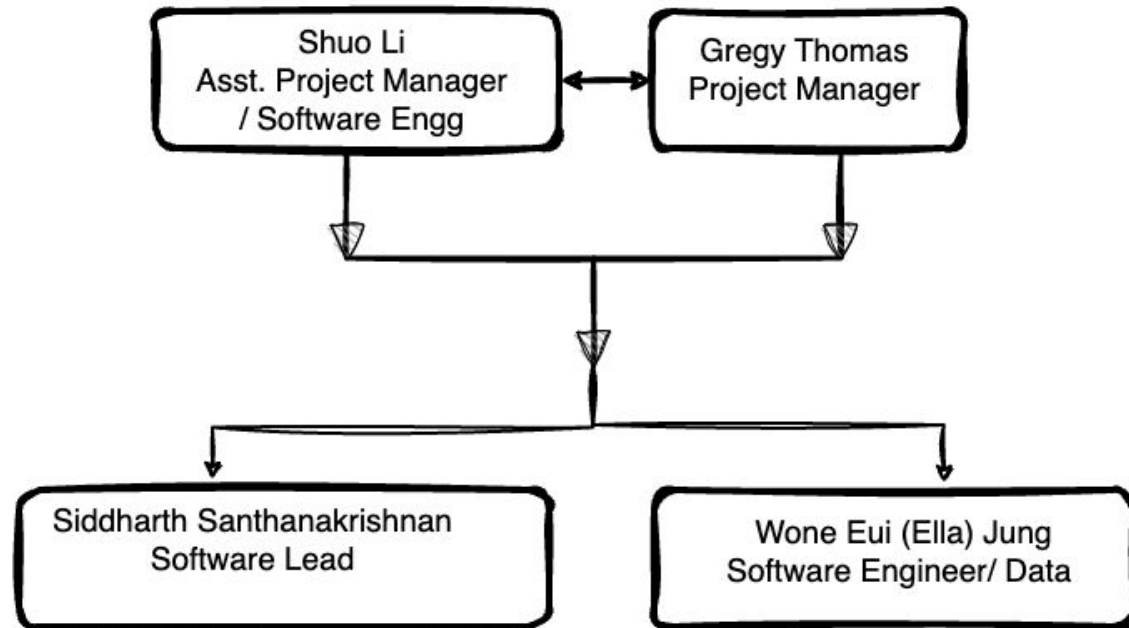
Use case



System Architecture

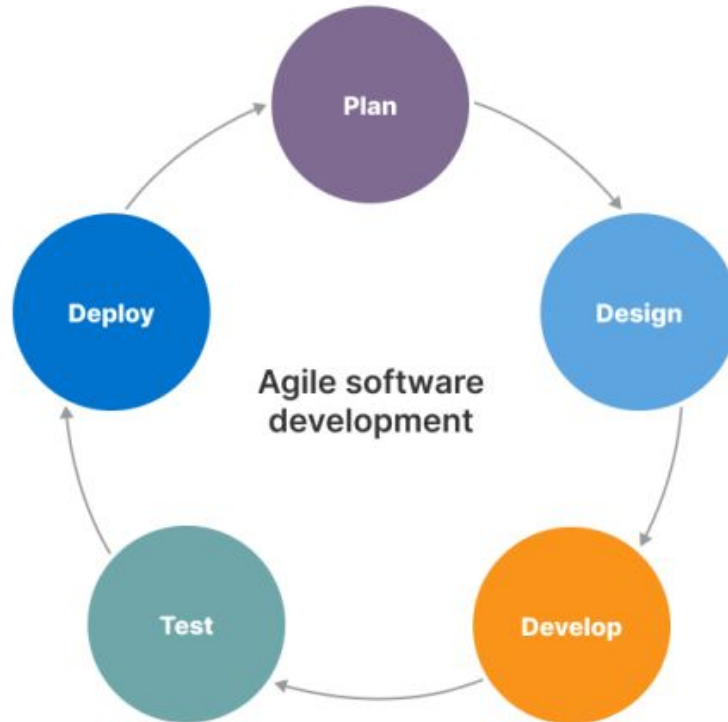


Organization





Technical Process model





Development Plan

10/30/2023	11/05/2023	Prototype #1 Development
11/06/2023	11/12/2023	Prototype #1 Development
11/13/2023	11/19/2023	Prototype #1 Development and testing
11/20/2023	11/26/2023	Prototype #1 Development and testing
11/27/2023	12/02/2023	Prototype #1 Final testing and delivery
12/02/2023	12/10/2023	Prototype #1 Final testing and delivery



Unexpected changes

In light of the limited time available for project development, we have revised our expectations to prioritize essential features in SRS 2.0.

One month into the project, a team member withdrew from the course. Additionally, unforeseen tragedies befell two other team members during the project's development. Consequently, more than half of the team faced challenges contributing their full efforts to the project. Despite our best efforts, only a portion of the features are functional as we conclude development.



Technical Problems

- Lack of technical expertise in Front end development and short development timeline make it hard to have a functional user interface on website.
- MySQL and MongoDB had obstacles to share data such as need cloud server and difficulties.
- Difficult to integrate multiple APIs during short timeline.
- Debugging issues taking longer than expected due to less developer resources.



Solutions

- Use Figma to make a mock user interface, to show the logic and function of our application
- Bigquery is similar to SQL and easy for data sharing through Google API
- Longer development time can make the job done
- More developers and resources needed to complete all requirements



Instructions 1

- Application can run
 1. Local Server
 2. Emulating it on a Windows Computer
- To run the application
 1. Use a laptop or desktop computer that is connected to the internet
 2. Use Google Chrome to access or emulate the application.
- To access functionalities, user can log in with this account
 1. Email: User1@gmail.com
 2. Password: 1234



Instructions 2

- Run the application through a URL generated and hosted by local host, the following steps must be taken:
 1. Use a laptop or desktop computer that is connected to the internet.
 2. Actively run Google Chrome (other browsers will not work).
 3. Type the URL: <https://supple-mate.web.app/index.html> into the address bar.
 4. Press enter



Test Report

SuppleMate Prototype1	
Test Case Description	General Information
<u>shouldPassIfStringMatches()</u> Test Case - This test case tests the response from application UI if the string matches the given test data(expected) with the string displayed in the given URL Endpoint/page.	Name: Sid
	Date: 11/28/2023
	Pass/ Fail: Pass
	Resolved?: Overall check -Pass
	Comments: No issues here.
Standard Code Compliance	Compliant/ Not Compliant
Meets the HTTP Method Standards and Status Codes	Compliant

SuppleMate Prototype1	
Test Case Description	General Information
<u>testGetAllSupplementEntriesSorted()</u> Test Case - This test case is a part of the Supplement Entry Data Access Objects Class checks whether the Supplement Entry Objects data injection and sorting operations for the customer are performed correctly or not.	Name: Sid
	Date: 11/29/2023
	Pass/ Fail: Pass
	Resolved?: Overall check - Pass
	Comments: Pass
Standard Code Compliance	Compliant/ Not Compliant
Meets the Java Code, Database and REST API Standards	Compliant

SuppleMate Prototype1	
Test Case Description	General Information
<u>testAddSupplementType()</u> Test Case - This test case creates a sample supplement type test data for a customer, stores it in the DB and checks whether the new Supplement Type added to the DB exists or not.	Name: Sid
	Date: 11/29/2023
	Pass/ Fail: Fail
	Resolved?: Overall check - Fail
	Comments: <u>java.sql.exception</u> at runtime due to access issues to DB. Working on resolving this error.
Standard Code Compliance	Compliant/ Not Compliant
Meets the Java Code, Database and REST API Standards	Compliant



Our Achievements

- The application has a Figma prototype, which shows the application is pleasant in appearance.
- The app has a simple layout and clear logic on each page.
- The application contains error handling that should prevent unexpected crashes/undesired output.
- The application has code developed for customer registration, log in, daily log for users to add notes including Supplement Entry and Supplement Types, security protocols.
- The application has a database and uses bigquery for easy data sharing through Google API.



What did we learn

From the project, we learned:

- There's always change happening, we need to be more prepared for changes to happen and get more resources when we can.
- Tools like IntelliJ, Figma, MongoDB, MySQL, Bigquery and how to build a software system.
- There are many tools to use, but we need to select the one that suits our project best.



Needs Improvements

- The application has a Figma Prototype to show how the UI looks like, but it does not have a user interface to work on a website yet.
- The application is not mobile compatible.
- Initially mySQL was planned to use, however due to data sharing issues we used bigquery.



Overall Recommendations

If a future development team were to work on our application, we would recommend:

- Develop a functional user interface with websites and mobile devices.
- Making the database functional with FDA data to be useful for dosage recommendation.
- Making the application compatible on different browsers and mobile devices.