1) The purpose of this document is to build an app to help motivate and organize a person's long and short-term goals into manageable timelines. It will allow users to visualize tasks and set short term goals to keep them on track. An AI will be implemented to recommend tasks to help users progress.

Examples of different possible themes or goals that users could have are; fitness, education, skill learning, or personal mental wellness. The UI will allow users to have multiple timelines set up that they can click into and focus on one specific task.

2.1) A wellness application will retain the following information:

Wellness Themes and Goals:

The app will store a database of different goals and wellness themes that have a variety of different activities associated with them. This will include the name of the activity and a brief description.

User Progress:

The app will store the user's progress in the goals and themes they have selected for the time span the user determined. This progress will be connected to the user's account.

User Information:

Web account information will be stored in order to connect the user's progress with their friends so they can be mutually visible. The ability to network with others through the app will not be public but private, where the only way to connect is to send a friend request.

2.4) The operating environment for the app is listed below:

Unity Engine

Android OS

iOS

AI LLM Software interface

3) A wellness app is something that every single user could use, every single day. It is designed to keep users productive, happy, and healthy with a built-in calendar reminder system to complete tasks of your choosing, or generated by AI. A user will have to create an event/task, it will be added to their feed for completion, and will be ultimately removed from the feed once completed. There will be a tab for your feed, AI Search page, Friends, Goals (past, current, and future), and Account/Settings. Most, if not all of the code will be done in C#.

4. External Interface Requirements

4.1 User Interface

Our app will use a dock style interface at the bottom listing the multiple options they have when using the app. The page on startup will be the chatbot section. This will include a place to type and examples of what you can prompt specific to the theme of our app. The dock will have tabs listing the features such as: chatbot, calendar, friends, goal progress tab.

4.2 Hardware Interface

Our hardware interface will communicate with the phone's CPU and WIFI chip. IOS and Android OS will be compatible with the environment we are using-Unity.

4.3 Software Interface

Each tab will be a software interface with distinctive features specific to that tab. The AI will be a software interface connecting to an online generative api.

4.4 Communication Interface

The friends tab will have connectivity features with adding friends and tracking their goals. We will have network management and security protocols to handle traffic between users.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

- Response Time: The app should support fast response time to all user input. This can be achieved through efficiency and proper optimization.
- Scalability: The app must be built to support more than what the potential maximum concurrent users could reach without any performance issues.

5.2 SAFETY REQUIREMENTS

- Data Recovery: There must be protection to fight potential data loss. The app must be able to restore goals and progress when necessary.

5.3 SECURITY REQUIREMENTS

- Encryption: All user data, including goals, progress, and personal information, must be encrypted both in transit and at rest.
- Privacy Options: The user should have access to the ability to make any goals private in the social scene.
- Secure API: Any third-party API used shall be trusted and uphold industry standards.

5.4 SOFTWARE QUALITY ATTRIBUTES

- Availability: The app should be available 99.9% of the time with the exception of maintenance.
- Correctness: The app should provide effective solutions using AI, and function without bugs.
- Maintainability: The code must be readable with proper documentation and comments to ensure updates and bug fixes are more easily possible.
- Usability: The app should be user-friendly and properly formatted for iOS and Android OS.