COA Progress Tracking App

Council of Aging

Dan Moore and Missy Rep

The OneStack Team

(Koji Natsuhara, Dileep Chokar, Luke Martinez)

03/02/2023

Table of Contents

1. Introduction

This document serves as a culmination of OneStack's work on the COA full stack app. In this project the team will design an extensive app which will be used to aid a client family in Colfax. The client family has a member that would benefit from a personalized app which tracks benchmarks, movement, and other postural progress. This document will provide details about the project's functional and non-functional requirements, specifications, and system evolution. The document will also describe the current state of the project and any future plans.

1.1 Background and Related Work

Currently, there are plenty of devices and apps that track the distance an individual has traveled. Fitbit, Garmin and other sport/fitness brands have developed GPS tracking devices which track the activity of the individual wearing the device. What will separate our COA app from these apps will be the reward and feedback aspect when you achieve a goal. Our goal is to do our best to encourage and motivate the user to complete their daily activities. To accomplish this, our app will include additional features that relate to tracking other aspects to target a user's needs.

Unfortunately, none of our team members have any experience in iOS app development. However, Koji has some experience in Android app development where the concepts of creating a mobile app will translate over. However, we may want to consider coding the project in Flutter as it can build multi-platform applications from a single codebase.

Flutter has seen a rise in popularity in the mobile development community. Our mentor recommended coding this full stack app in Flutter as it would allow us to create the app for Android and iOS using a single codebase. While all of our team members could code and build an Android app, none of us had an iOS workspace to develop Apple applications in. To develop an iOS app there is a considerable amount of materials needed to start. Starting with IOS app development you will need to download Xcode using a Mac computer with an Intel processor running the latest Mac OS version. To develop iOS apps you will need to learn to read and write Swift code. To build an iOS app, you will need to register as an Apple Application Developer to gain access to download Xcode and iOS SDK [8]. However, Flutter bridges this gap and allows us to cross-platform app development regardless of the OS workspace we are in.

The client does not have a specific direction for the app, so much testing and improvising will be needed. While we will reference other devices which implement trackers, our app will be structured uniquely to fit our client.

1.2 System Overview

The Council of Aging (COA) is seeking technological support for a client family in Colfax. One of the family members has special needs and they request a distance tracking app. This progress tracking app will record the distance a user has traveled and provide feedback in the form of positive animations and audio messages. The feedback prompts will be triggered when a user achieves goals such as meeting benchmarks, movement criteria, and other postural progresses. Ideally, the clients would like to have a screen embedded onto the aid equipment and can be configured to be in sync with the app.

The OneStack team, under the guidance of a team mentor will be addressing the designing, building, and delivering of the COA progress tracking app. While our efforts are towards a single family, we are aware that this application will be useful to many other families and educators for special needs children. To keep this in mind, the client has asked us to give the widest possible application to help others with similar challenges as our client's family.

Daily tracking of distance traveled will be the main feature the COA tracking app will need to cover. With this information, the app will be able to generate graphs from the daily data it collects in a viewable format for the user. The app will also provide feedback and progress of goals that the user can set and track.

To track the location of the user, we will be using the device's GPS location system. Within the app, we will require the user to give location access to the app so that we can then provide and collect data of the user's progress. As an additional option, we will need to find a tracking device that will record the walking progress of the user. When the user uses the walker, this tracking device can then be transferred to their backpack and monitor all of their movements. This tracking device will not only need to record the movements of the individual, but also send the information to our application so that it can be displayed on another's phone.

The animation and verbal phrases will need to be designed as well. Recognizing who our app is built for is important, while the client told us that firework animations and the song "Who let the dogs out" are our clients favorite, other users may prefer different positive exclamations of praise. Designing on how to implement the praise and congratulation feature will be an important factor to design and implement.

We will also have experimental features which we will implement or outline to the client and get their input. Such features are designed to assist our target individual in his daily activities, since we are aware of his struggles in walking and talking. These features might include, but are not limited to, communication features, mini games, and parental input. Communication features will allow our target individual to communicate better with his classmates and others by allowing basic and written phrases to be said through the app. Mini games will allow our target individual to play simple games such as tic-tac-toe with his peers.

Parental input may come from the parents devices and will allow them to monitor and encourage our target individual through the app.

1.3 Client and Stakeholder Identification and Preferences

Our client is Council on Aging & Human Services (COA Human Services) a non-profit organization dedicated to enhancing lives and supporting communities with transportation and nutrition services. Dan Moore is the primary contact who is the grandfather of a 7 year-old special needs child named JP. Dan will be helping us with what features should be developed. Initially, our team will develop a progress tracking app specifically JP, but as the project develops, we can streamline the app to be used for other families with special needs children.

The parents and the teacher of the special needs child will primarily be using the web app and will be able to track progress in the form of a graph. The parents and teacher can use the graphs and other data to measure the special needs child's ability to move over time.

2. System Requirements Specification

2.1. Use Cases

Source: Daniel Moore originated this user-story.

Story: JP is Daniel's 7 year-old grandson, who has been diagnosed with a chromosomal deficiency that has kept him from walking unassisted. His parents and teacher want to keep track of his movement throughout the day to track his progress. A tracking device will be placed on his walker or in his backpack that will relay his location to a database which will be outputted to any device supporting the app we have created. JP's guardians will be able to use this progress tracking app on their phones to view JP's distance and the activity he has done for each day. The app will display JP's data as distance traveled and places visited using graphs and maps that are easily readable to viewers.

Source: Daniel Moore originated this user-story

Story: JP is Daniel's 7 year-old grandson, who has been diagnosed with a chromosomal deficiency that has kept him from walking unassisted. His parents and teacher want to keep track of his movement throughout the day to track his progress. The app will be attached to JP's walker and automatically record movement related progress such as distance traveled, average speed, etc. After reaching a new daily, weekly, or monthly record in movement progress, a notification in the app appears. The notification plays triumphant or dance music, shows a graphic of silly dancing, balloons, or party poppers with the intention of motivating JP to move more, and train his own ability to walk.

Source: Daniel Moore originated this user-story

Story: JP is Daniel's 7 year-old grandson, who has been diagnosed with a chromosomal deficiency that has kept him from walking unassisted. JP also has a difficult time communicating and talking, which can be a barrier when trying to interact with his peers. In addition to audio features, the app will feature interactive mini games which JP can play with others. JP will click the mini games icon on the app which will take him to a random simple mini game which will be able to be played by JP and others. Some mini games available will be tic-tac-toe and rock paper scissors.

2.2. Functional Requirements

2.2.1. [Location Tracking]

[Tracking User Location]: [The application needs to be able to track the user's movement in order to calculate traveled distance for the user. The location will be tracked using a third party tracker such as a fitbit watch, and the data will be used in the application.]

Source: [The client Daniel Moore specified this requirement]

Priority: Level 0: Essential and Required

2.2.2. [Affirmations]

[Positive Affirmations]: [The application will give the user positive affirmations such as "congrats" or "well done" after specific actions which include walking and interacting with the app.] [9]

Source: [The client Daniel Moore specified this requirement]

Priority: Level 0: Essential and Required

2.2.3. [Interactive Features]

[Interactive Games]: [The application will have built in mini games which can be accessed by the user. The mini games will feature simple 2 player games such as tic tac toe and rock paper scissors for ease of access [10].]

Source: [A team member Dileep suggested this requirement.]

Priority: Level 1: Desirable

2.2.4. [Application Input/Output]

[App output]: [The application must be able to output results so that JP's parents or authorized individuals can access the gathered data. The output will contain information about JP's behavioral patterns such as movement and other interactions with the application]

Source: [The client Daniel Moore specified this requirement]

Priority: Level 0: Essential and Required

[App Input]: [The application must be able to take input. The input will come from a tracking device and from JP's teacher. The input will be used to record distance traveled and location of JP as well as his emotions for a particular day.]

Source: [The client Daniel Moore specified this requirement]

Priority: Level 0: Essential and Required

2.2.5. [User Authentication]

[User authentication]: [The application and all collected data should only be used by intended users.]

Source: [The client Daniel Moore specified this requirement]

Priority: Level 0: Essential and Required

2.3. Non-Functional Requirements

2.3.1. [Extensible and Versatile]

This app shall be extensible and versatile to allow a user to interact with multiple features that the app supplies. This tool should be able to easily support new environments beyond what was considered during development.

2.3.2. [Easy to Use]

This tool should be intuitive and have a UI that is friendly towards the user.

2.3.3. [Maintainable]

The application should be easy to maintain for the family of JP.

2.3.4. [Protected]

The application should protect the personal data of the user from outside interference. The security and protection of the user’s privacy and data is essential for this application.

2.3.5. [Portable]

This tool needs to be able to operate without an active network connection. The app should be able to communicate between the tracking device in a closed network.

2.3.6. [Updatable in-real time]

This app should be able to display the data of the user at any time. A user should be able to view their location and distance traveled while using their app and see their location being updated in-real time.

3. System Evolution

3.1 Devices in Early Development

The app is currently planned to be developed for all devices using the flutter plugin on android studios. All three of the developers on the OneStack team primarily use Windows to develop software, and Windows is capable of developing software for Android. The clients own iOS devices, so in order for the clients to use the app we must develop it for iOS as well.. Another discussed option for development is for the OneStack developers to acquire iOS devices to access an iOS development environment. This way the clients can use their existing devices to use the app. Discussion between the parties has led to the former to be less expensive and better for the developers.

3.2 Devices in Late Development

One of the goals of the project is to bring the app to other families with special children, so only having the app available to Android users alienates a large share of the mobile market. It is potentially worthwhile to develop for both Android and iOS users later in the project. For this reason flutter seems to be the best option.

3.3 Use of Fitbit

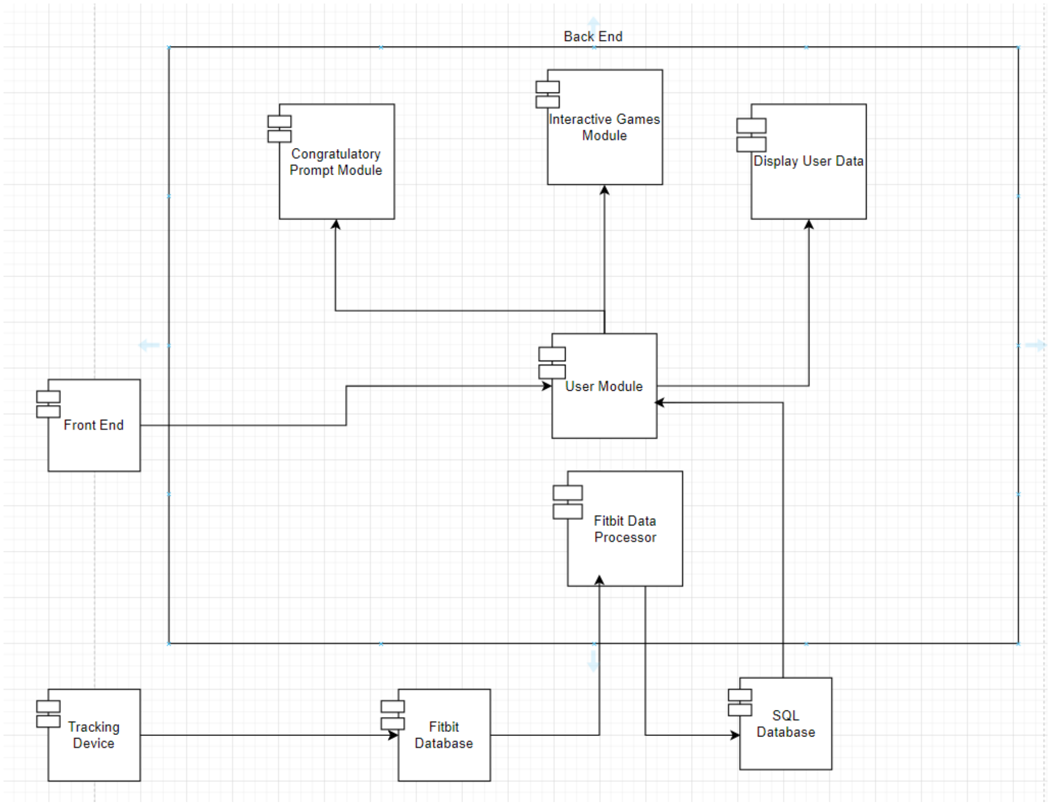
To aid the collection of data on the special child, the project could use a Fitbit as a supplementary means to track movement progress and health of the special child. There is potentially a lot of useful data to be collected from using a fitbit [7]. A few problems with this feature include: the special child does not consistently wear the fitbit, the developers cannot collect data from the Fitbit in a meaningful way, if a user does not own a Fitbit, they cannot access some of the features of the app.

4. Architecture Design

4.1. Overview

The OneStack team has decided on a progress tracking flutter application. This app will be based on a Client-Database architecture pattern. Based on Daniel Moore's input and preferences the team has concluded that this application would be best-suited as a mobile application. Thus there will be a front end which the user will interact and make requests with that will communicate to the Fitbit and our personal SQL database. Below is a diagram detailing a general view of our applications architecture using a component diagram. The Front End component of the application will deal with displaying and logging into our application and what the graphical interface the user will be interacting with. The User Module will receive data from our SQL database and transmit that data to the Display User Data Module. It will also be responsible for processing general user functions using data received from the front end. The user module will also call the Congratulatory Prompt Module which will display congratulatory and affirmation messages to the user when specific actions are performed by the user. The Interactive Games Module will be activated upon user request by the user module to provide simple interactive games that can be played by the user and a friend. The Display User Data component will be responsible for displaying location and tracking information to the UI that is processed from the user module. The Tracking Device component will be a fitbit device or the native location service that will collect the user's location and distances.

Since a connection is already established between the tracking device and Fitbit (or native location service built into the device), the Fitbit Database component will be used as a middle-man to store the data the tracking device collects. The Fitbit Data Processor component is to export the data from the Fitbit database as a CSV file that we will then manipulate to store into our own SQL database. The SQL Database component will be what the progress tracking app will communicate with. The user module will actively connect to the SQL database and take and output data back into the SQL database so that other devices can then view the same data.



4.2. Subsystem Decomposition

4.2.1. [Congratulatory Prompt Module]

a) Description

The Congratulatory Prompt Module will be responsible for displaying affirmative and encouraging messages to the user for particular actions such as movement and position changes. This module will get the relevant data from the user module.

b) Concepts and Algorithms Generated

The Congratulatory Prompt Module will have one major class which will store all the encouraging messages and audio clip locations. These locations will be called upon the request of the user module. The class will have GET, SET, and PLAY functions. GET will retrieve the appropriate clip from its location. SET will set up the clip to be played in the UI. The PLAY function will enable the animation or audio of the clip in the front end UI.

c) Interface Description

Services Provided:

1. *GET*
    a. *Service name*: GET
    b. *Service provided to*: Congratulatory Prompt Module
    c. *Description*: The GET function takes input from the user module and retrieves the appropriate audio or video animation clip from its location.
2. *SET*
    a. *Service name*: SET
    b. *Service provided to*: Front End
    c. *Description*: The SET function takes the clip obtained from the GET function and loads it onto the front end
3. *PLAY*
    a. *Service name*: PLAY
    b. *Service provided to*: Front End
    c. *Description*: The PLAY function will access the clip from the SET function and play it.

Services Required:

1. *Service Name:* Send data from user module

    Service Provided From: User Module

4.2.2. [Interactive Games Module]

a) Description

The Interactive Games Module will be responsible for loading 2-player interactive mini games upon request of the user. This module will get the relevant request details from the user module.

b) Concepts and Algorithms Generated

The Interactive Games Module will have one major class which will store all the interactive game locations. These locations will be called upon the request of the user module. The class will have GET, SET, and PLAY functions. GET will retrieve the appropriate game from its location. SET will set up the game to be played in the UI. The PLAY function will enable the full function of the game in a separate UI.

c) Interface Description

Services Provided:

1. *GET*
    d. *Service name*: GET
    e. *Service provided to*: Interactive Games Module

f. *Description*: The GET function takes input from the user module and retrieves the appropriate game from its location.

2. SET

g. *Service name*: SET
h. *Service provided to*: Front End
i. *Description*: The SET function takes the game obtained from the GET function and loads it onto the game UI

3. PLAY

j. *Service name*: PLAY
k. *Service provided to*: Front End
l. *Description*: The PLAY function will access the game UI and run the game set by the SET function.

Services Required:

2. *Service Name:* Send data from user module

Service Provided From: User Module

4.2.3. [Display User Data]

a) Description

The Display User Data will be responsible for displaying user tracking data. This module will get the relevant tracking details from the user module.

b) Concepts and Algorithms Generated

The Display User Data will have one major UserData class. The class will have the DISPLAY function which will get user information from the user module and display it on the front end. Information displayed will be personal user information, location, and distance tracking.

c) Interface Description

Services Provided:

1. DISPLAY

a. *Service name:* DISPLAY
b. *Service Provided to:* Front End
c. *Description:* The DISPLAY function gets input from the user module about user information, location, and distance tracking. This data is outputted to a separate UI to the front end.

Services Required:

1. *Service Name:* Send data from user module

   Service Provided From: User Module

4.2.4. [Data Processor]

a) Description

Describe the subsystem and identify its responsibilities. The data processor subsystem will consist of exporting a CSV file from Fitbit's or the native device's database and manipulating the data to store it into a SQL database.

b) Concepts and Algorithms Generated

The Fitbit data processor subsystem will parse through a CSV file generated by the tracking device to then store it into the SQL database. The processor will contain ImportCSVFile, ParseCSVFile, and StoreCSVFile methods.

c) Interface Description

Services Provided:

1. ImportCSVFile (GET)
   a. *Service name:* ImportCSVFile (GET)
   b. *Service provided to:* SQL Database
   c. *Description:* ImportCSVFile will get the data that the tracking device collects from the location database as a CSV file.
2. ParseCSVFile
   a. *Service name:* ParseCSVFile
   b. *Service provided to:* SQL Database
   c. *Description:* ParseCSVFile will parse a CSV file from the tracking device and store the parsed data as insertion tables that will then be used to store into the SQL database.
3. StoreCSVFile (SET)
   a. *Service name:* StoreCSVFile
   b. *Service provided to:* SQL Database
   c. *Description:* StoreCSVFile will execute the insertion statements into the SQL database.

Services Required:

1. *Service name*: Send data to SQL Database

   Service provided from: Fitbit Database

4.2.5. [User Module]

a) Description

The user subsystem will contain and link all the info the user has with the SQL database. This subsystem will store all the information the user can change, view, and share to other users. The user module subsystem will rely heavily on the connection between the SQL database to retrieve all the data that the tracking device collects.

b) Concepts and Algorithms Generated

The user subsystem will have methods that will getters and setters for the user class. GetUser, SetUser, GetGroup, SetGroup, DisplayUserData, GameModule, CongratulatoryModule. The user module subsystem will be the main entry and exit point for data in the COA application.

c) Interface Description

<u>Service Provided:</u>

1. GetUser()
    a. *Service name:* GetUser (GET)
    b. *Service provided to:* SQL Database
    c. *Description:* Gets user information from SQL database
2. SetUser
    a. *Service name:* SetUser (SET)
    b. *Service provided to:* SQL Database
    c. *Description:* Sets user information from SQL database
3. GetGroup
    a. *Service name:* GetGroup (GET)
    b. *Service provided to:* SQL Database
    c. *Description:* Gets which groups the user is in from the SQL database
4. SetGroup
    a. *Service name:* SetGroup (SET)
    b. *Service provided to:* SQL Database
    c. *Description:* Sets which groups the user is in from the SQL database
5. DisplayUserData
    a. *Service name:* DisplayUserData
    b. *Service provided to:* SQL Database
    c. *Description:* Shows the DisplayUserData activity to the user
6. GameModule
    a. *Service name:* GameModule
    b. *Service provided to:* SQL Database
    c. *Description:* Shows the Game activity to the user
7. CongratulatoryModule

    a. *Service name:* CongratulatoryModule
    b. *Service provided to:* SQL Database
    c. *Description:* Shows the congratulatory activity to the user

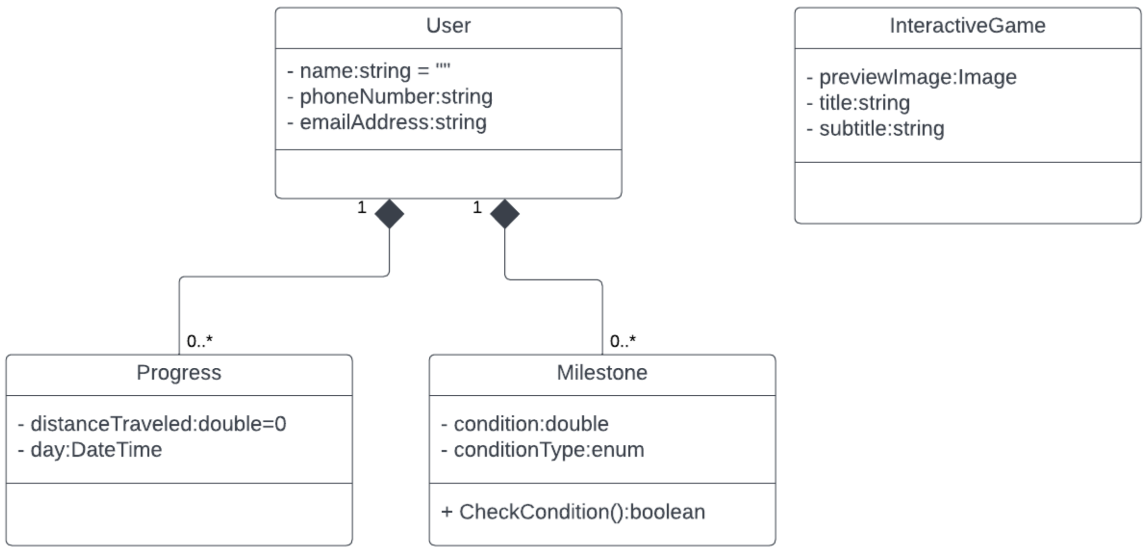
<u>Services Required:</u>

1. *Service name:* Connection to SQL Database

   Service provided from: Data Processor and SQL Database

5. Data design

This UML diagram shows the relationship between the user, progress, and milestone objects, as well as multiplicities.



6. User Interface Design

6.1. [Login Page]

When the app starts, the user will be routed to a login page. The login page contains a welcome message. Below the welcome message is a text input box for username and another text input box for password. Under the text input boxes is a login button, the button confirms that the password matches the login for the username and signs the user into the app.

6.2. [Home Page]

The home page contains a list of buttons that navigate to various pages. The buttons appear in two columns and can span multiple rows. The buttons have text that displays the name of the page as well as an associated image. The buttons navigate to User data, Milestone, Games, and Profile.

6.3. [Page Header]

After logging in to the app, there is a page header at the top of the page, and the page's content is displayed below the page header. On the left side of the header, there is a button that navigates the user back to the home page. On the right side, there is a gear button that navigates the user to the settings page.

6.4. [User Data Page]

The display data page consists of two tabs that the user can switch between. These two tabs display a line graph of the distance traveled by the special child, while the other displays a calendar showing distance traveled per day. The line graph page displays a graph that displays how many steps were taken on the y-axis, where the x-axis displays the time. The time can be represented in multiple ways depending on a dropdown input box, which allows the user to display the time in days, months, years, or all. The calendar tab displays the current month of the current year, as well as all of the days of that month displayed in calendar format. The distance traveled is shown on each of the days in the calendar. The user can press an arrow to switch to previous and following months. The user can also add comments to days.

6.5. [Milestone Page]

On the milestone page the user can view milestones or set milestones. Below the page header is a button to add milestones, the parameters for completing milestones have yet to be decided. Below the add milestone button, the user can view the currently created milestones. The milestones are displayed vertically in a list, where each button shows the condition for completion of the milestone.

6.6. [Congratulatory Prompt]

The user can view a congratulatory prompt when a milestone is reached. The prompt animates some balloons and party poppers, plays some celebratory music, and displays text such as “Milestone reached! Excellent job JP!” To view the prompt, the user should navigate to the milestone page and select the completed milestone.

6.7. [Interactive Games Page]

On the interactive games page, the user can select from a list of buttons of games to play. The buttons are listed vertically. Each button in the list contains a preview image of the game, displays in large bold text the title of the game, and a subtitle of the game in smaller text. When the user selects a button, the game starts and the user can play.

6.8. [Settings Page]

The settings page contains several togglable settings. The left side has a description of the setting, while the right side has a checkbox or slider depending on the settings. The settings are: A checkbox for Notifications, a slider for adjusting volume, and a checkbox for enabling location services.

6.9. [Profile Page]

This page allows the user to view and edit variables related to the user themselves. On the left side there is a list of variables to view and on the right side there is an edit button associated with each variable. The variables are: Username, Name, Password, and Group.

7. Work Testing and Acceptance

7.0.1 Test Objectives and Schedule

The OneStack team will test key features of our application to ensure proper functionality and integrity. Our test cases will cover all major application components and auxiliary components required by the main components. Since we will be following the continuous integration and continuous delivery models in testing, any updates to our application will automatically be able to be applied with our tests to ensure core functionalities of our application.

In order to conduct testing, we will be using the android studios testing framework to conduct testing on the application [5]. The framework includes a variety of tests such as unit tests, integration tests, instrumented tests, and local tests. We will also use a database management system to perform tests (SQL queries) on the database we will be using to ensure integrity of tables and data.

The milestones or process of testing will include writing code for a feature, writing test cases for the code, deployment, and user feedback. Since we are using the agile development process, the testing process will be performed many times for each deployment. The deliverables will include a working deployment of our application and updated documentation. Responsibility for each feature will be divided among the team and each team member will be responsible for testing the features they implement.

7.0.2 Scope

This document provides a general idea of the features that we will be testing in our application. The scope covers general developer testing strategies such as unit testing and also what testing frameworks we will be using, but specific test cases will not be specified.

7.1 Testing Strategy

The OneStack team will use both continuous integration and continuous delivery in our testing to gain confidence in our application integrity. Throughout the development of our application we will write tests for our system that will help determine the greatest points for failure in the system. The main purpose of our tests will address the functionality of our application in its location system, credential system, and setting changes. After finishing development for our project, we will provide a robust testing system that will allow for quick testing changes to ensure the functionality of our application.

In our testing and development process, we will follow an agile-like testing strategy in which tests will be written by the same developer writing the code. In this way, the tests will be produced concurrently with the code. Another reviewer may be assigned to make sure the tests are adequate and fully cover the functionality of the code being added. Milestones in our testing process will include: implementing code, unit testing, integration testing, system testing, and user acceptance testing.

7.2 Test Plans

7.2.1 Unit Testing

Our team will follow a standard approach for unit testing. We will require all non-trivial methods to have at least two unit tests. One of these tests will test the valid use of the method and one for the invalid use of the method. Trivial methods that are a set of very simple operations such as adding two numbers or reading a string from a textbox will not require tests. A non-trivial method would be considered as logging into our application. In this case, a valid test will include using the correct credentials and an invalid test will attempt logging in using invalid credentials.We will defer to the individual developer to determine if a unit needs more testing and expect them to communicate with the developer responsible for the unit about each test.

7.2.2 Integration Testing

Given that our application has many complex entry points and paths a user can travel in our application, integration testing will be significantly more complex than unit testing. Our team will first identify all the major application entry points of the application and develop integration test classes to cover such points. Given that our team is not familiar with integration tests in Flutter and in Android Studios, we will be integrating on a per class basis rather than a per method basis. By doing this, we will hopefully reduce the complexity of writing tests while still providing good coverage in the terms of identifying bugs or defects in our software. If an entry point is vital in our application and complex, our team will consider a more rigorous approach to provide solid test coverage.

7.2.3 System Testing

7.2.3.1 Functional testing

All functional requirements detailed in the requirements specification section of the document will be tested. In our application some key features to be tested would be GPS location tracking, user and profiles, interactive games, encouraging audio/visual messages, database storage/retrieval, and UI displays. In system testing, we will assign our system tests to encompass all the functional requirements. The system tests will be updated if new features are implemented. The team member responsible for a specific feature will write tests for that feature. Any failed tests will be recorded as an issue in the github repository and will be fixed and retested.

7.2.3.2 Performance testing

Performance tests check whether the nonfunctional requirements and additional design goals from the design document are satisfied. In stress testing, the system is stressed beyond its specifications to check how and when it fails. The app will need to update in read time with a poor internet connection, we can test this by reducing the amount of data transfer to still be functional.

7.2.3.3 User Acceptance Testing

The users will perform acceptance testing with help by the developers. The app will need to be installed correctly on a compatible iOS or Android device that the users have access to. The users will then one-by-one compare the app to the initial requirements specified in this document. If the users approve the system, the tests pass and the features are complete; the features will not be modified unless the user requests a feature change. If the users do not approve of the system, the developers and users will meet and discuss whether the app meets the requirements in this document. The users can then reconsider whether the app meets the requirements or renegotiate the requirements.

7.3 Test Results

The environment testing specifies the test environment on the app and how it will be used when deployed. The app can be installed on iOS or Android devices that can use location services. We will use a database hosted on Amazon Web Services (AWS)[6] that the device will need to read and write from.

The app will be installed on at least one device and can be installed on more than one device. One device will be used for tracking movement and progress, and can be used to view progress and write comments as well. All other devices can only be used to view progress and write comments.

7.4 Test Cases

7.4.1 User Login

1. **Functional Test**
2. **Expected Results:** When a returning user inputs their accepted credentials into the COA Firebase, and logs in, the user will be shown to the home page.
3. **Observed results:** The user is shown the home page, when inputting valid credentials accepted by the COA Firebase.
4. **Test result:** PASSED
5. **Test case requirements:** User must have an account already registered into the COA Firebase before being able to login.

7.4.2 User Registration

1. **Functional Test**
2. **Expected Results:** Users should be able to create a new account using the registering profile page and the credentials should be added to the COA Firebase where they will be able to login with those same credentials.
3. **Observed Results:** After registering a new user. The COA Firebase is updated with the new user and the user is able to successfully login.
4. **Test Results:** PASSED
5. **Test Case Requirements:** User cannot register a new profile with an email that already exists in the database.

7.4.3 Progress Page

1. **Functional Test**

2. **Expected Results:** Sending a query to firestore to get distance tracking per day and displaying them as interactable tiles.
3. **Observed results:** Sending a query to firestore to get distance tracking per day and displaying them as interactable tiles.
4. **Test result:** PASSED
5. **Test case Requirements:** There is at least one document in the database for distance per day.

7.4.4 Location Tracking

1. **Functional Test**
2. **Expected Results:** User location is tracked and the accurate distance is output to the database
3. **Observed results:** User location is tracked and the accurate distance is output to the database
4. **Test result:** PASSED
5. **Test case requirements:** Location tracking page must be open.

7.4.5 Edit Comment

1. **Functional Test**
2. **Expected Results:** The user can press the edit button on the progress info page and type in or edit the current comment. After pressing the save button, the comment is updated and stored on the firebase firestore.
3. **Observed results:** The user can press the edit button on the progress info page and type in or edit the current comment. After pressing the save button, the comment is updated and stored on the firebase firestore.
4. **Test result:** PASSED
5. **Test case Requirements:** There is at least one document in the database for distance per day.

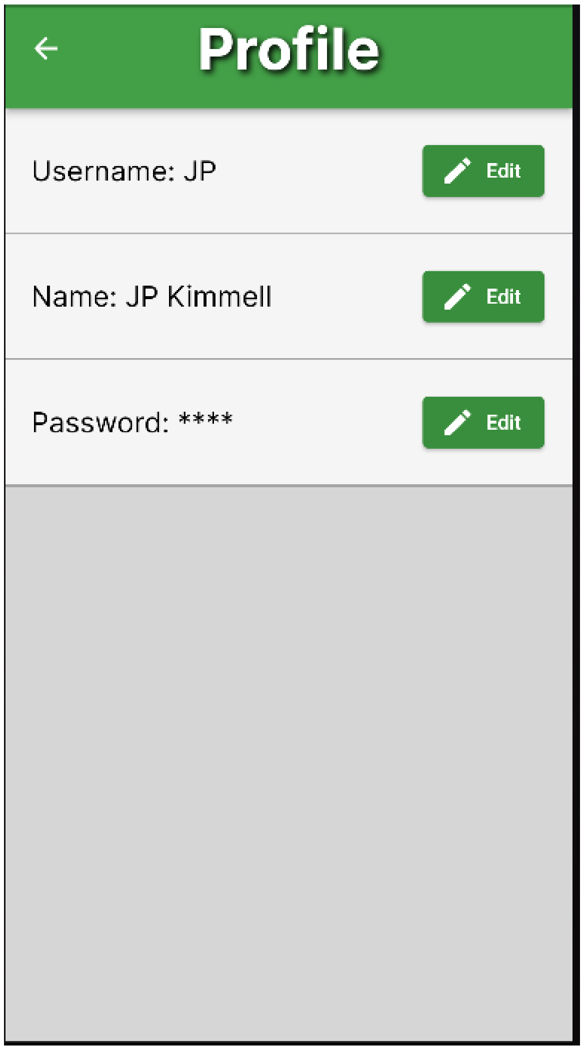
7.4.4 Location Tracking: UserID

1. **Nonfunctional Test**
2. **Expected Results:** Distance from Location Tracking is assigned to the appropriate current user
3. **Observed results:** Distance from Location Tracking is assigned to the appropriate current user as shown in firebase
4. **Test result:** PASSED
5. **Test case requirements:** Location tracking page must be open.

8. Alpha Prototype Description

The OneStack team's alpha prototype can be viewed with the images and subsystem descriptions below. Each subsystem has a description of the current state of the system and any remaining work to be done.

8.1 [Profile Page]



Username: JP

Name: JP Kimmell

Password: ****

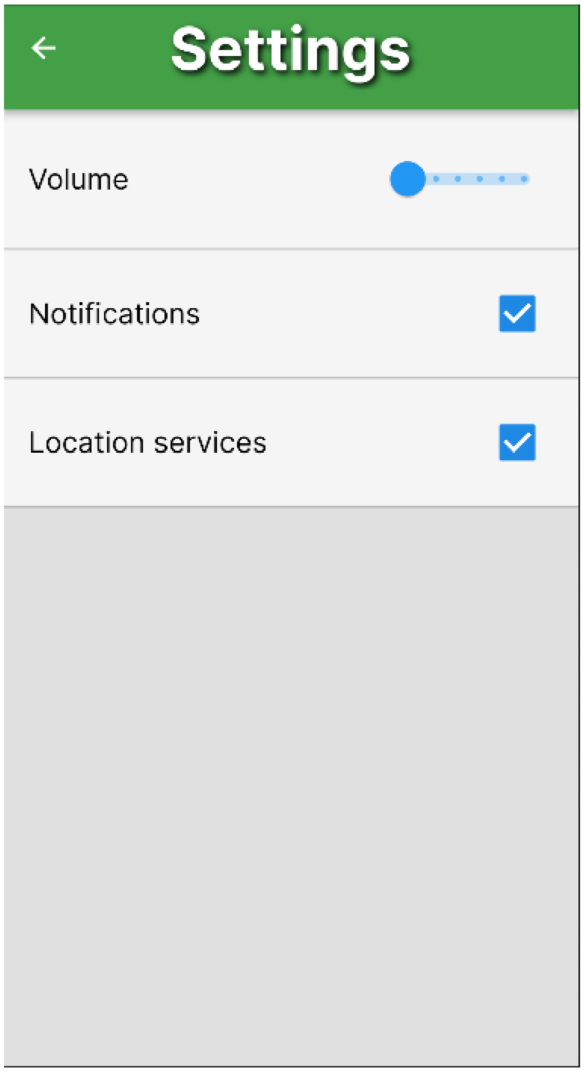
8.1.1 Functions and Interfaces Implemented

In the profile page, currently all features have been implemented. These features include displaying the username, name, password (hidden), and group of the current user. The user is also able to modify any of the fields. What remains is to connect our profile database to display data into this page for the user and any changes to the data to be reflected on the database.

8.1.2 Preliminary Tests

Tests have not been set up yet. We plan to include test cases for each feature to test the profile page. The team will mainly test if the changes to any of the fields on the profile page are updated properly on the database as well as testing if null checking is working as intended.

8.2. [Settings Page]



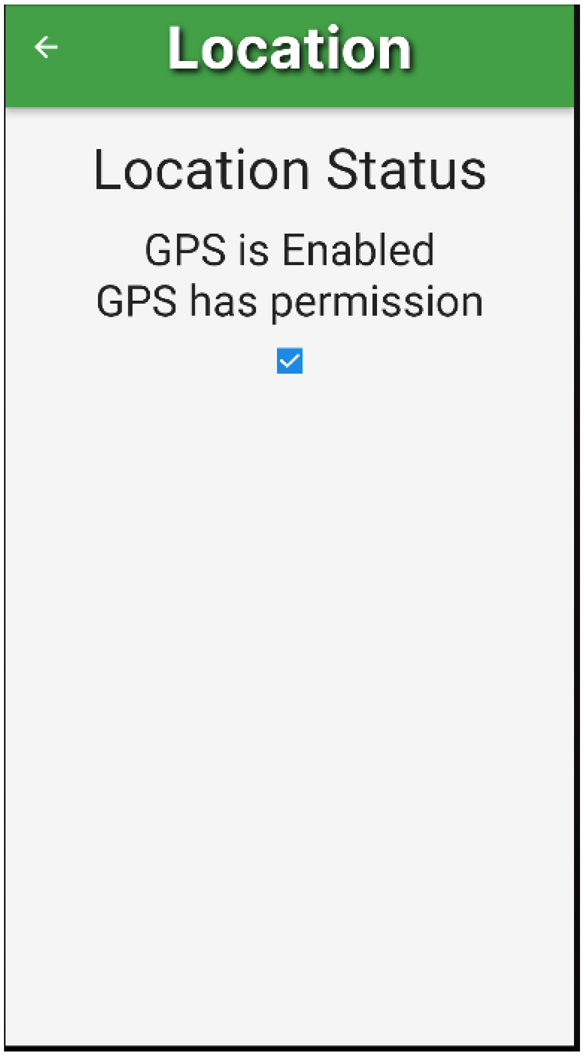
8.2.1 Functions and Interfaces Implemented

In the settings page we have implemented the volume, notifications, and location services features which are able to be toggled. What is remaining is to link all features to their representative components in the device.

8.2.2. Preliminary Tests

Tests have not been set up yet. We plan to include test cases for each feature to test the settings page. The team would need to manually test if notifications are sent to the device, volume adjustment works as intended. We can automate testing for location services by returning null if the location services are disabled.

8.3. [Location Module]



Location Status

GPS is Enabled
GPS has permission

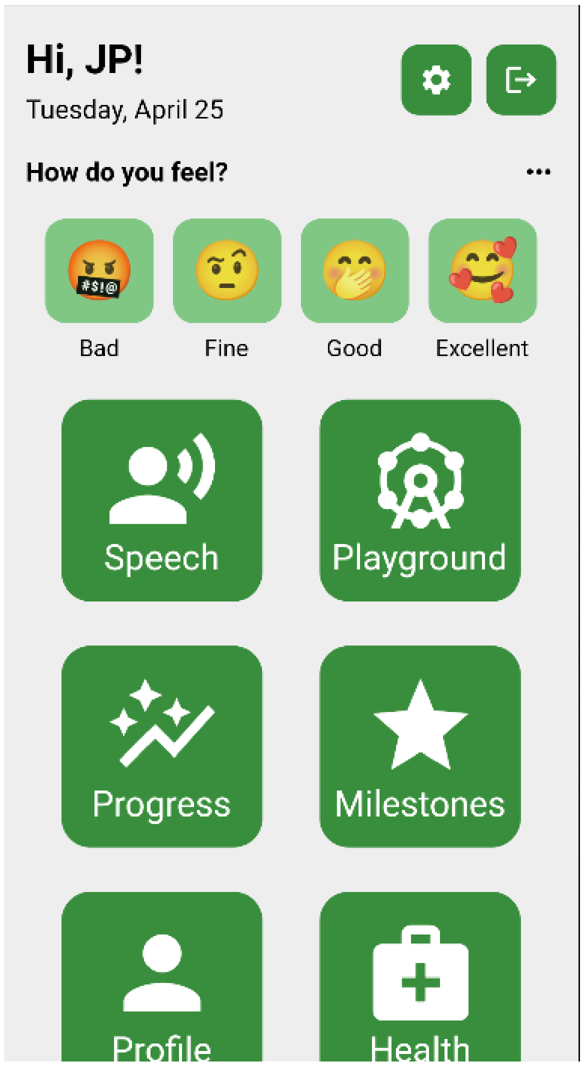
8.3.1. Functions and Interfaces Implemented

The location module is responsible for getting the coordinates of the device and tracking them. Currently the module can get the coordinates. What is remaining is to send the data in intervals to the database.

8.3.2. Preliminary Tests

Tests have not been set up yet. We plan to include test cases for each feature to test the location page. We plan to manually test whether the location pings are accurate.

8.4. [Home Page]



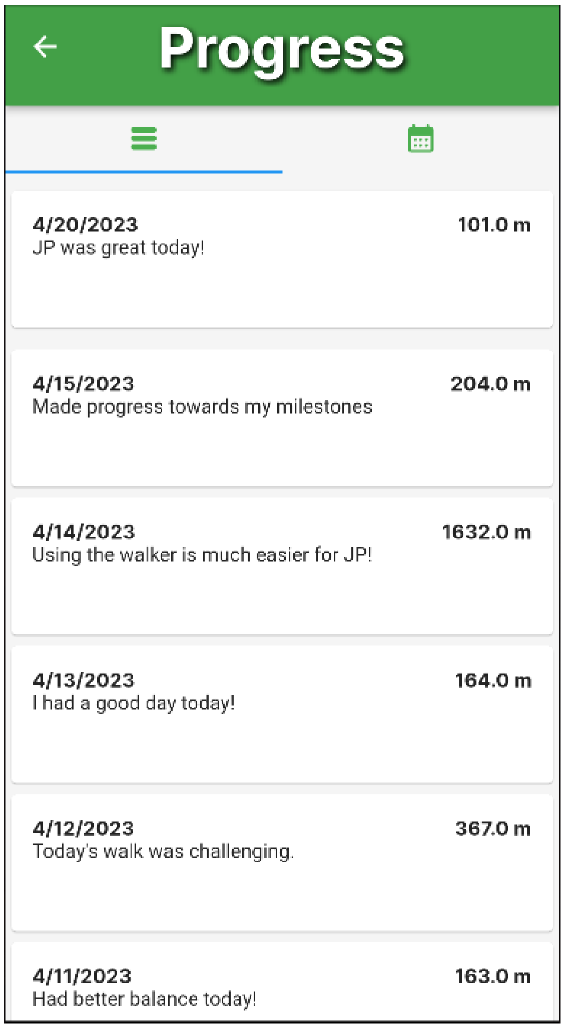
8.4.1. Functions and Interfaces Implemented

The homepage will be the main hub that the user will be in when they login. The home page will have access to view all the pages within the app. The user will be greeted with a message in the home page as well as be able to select how they are feeling on that particular day. Some features that still need to be implemented is a navigation sidebar, the search bar may be removed, and the feature to record your well-being.

8.4.2. Preliminary Tests

Testing is still in progress for the homepage as it is not yet completed. However, navigation between the pages and the homepage are all working accordingly. However, when testing the application on a smaller screen, their is a pixel overflow error that will need to be fixed and retested.

8.5. [Progress Page]



Progress

4/20/2023 101.0 m
JP was great today!

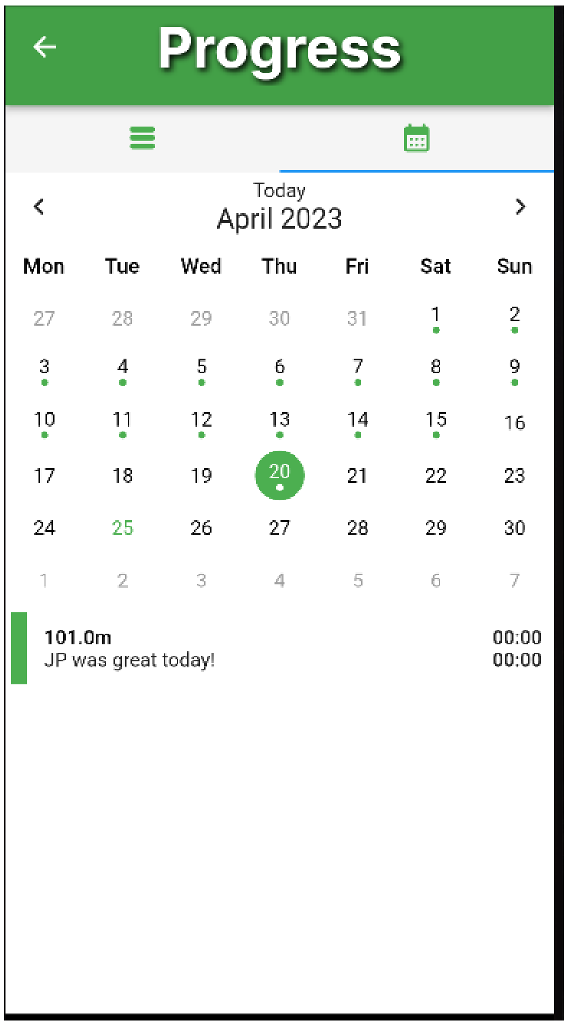
4/15/2023 204.0 m
Made progress towards my milestones

4/14/2023 1632.0 m
Using the walker is much easier for JP!

4/13/2023 164.0 m
I had a good day today!

4/12/2023 367.0 m
Today's walk was challenging.

4/11/2023 163.0 m
Had better balance today!



Progress

Today
April 2023

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

101.0m 00:00
JP was great today! 00:00

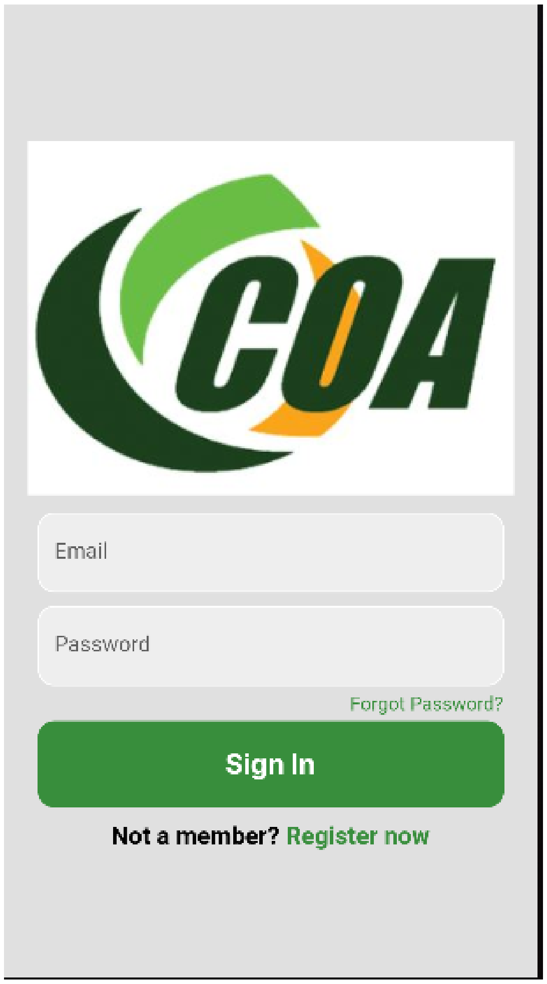
8.5.1. Functions and Interfaces Implemented

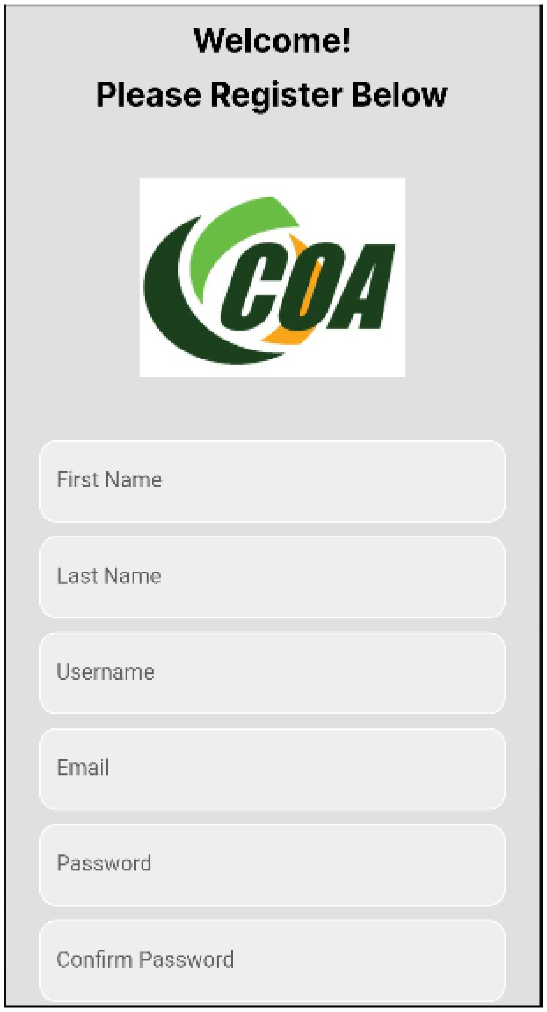
The page is a vertical list view of days and the distance traveled for each day as well as an optional comment written by the user. Each day has an add/edit comment button that allows the user to add/edit the comment. The list of days are built by reading from the firebase database.

8.5.2. Preliminary Tests

Currently there are no running tests for the progress page. We can test it by using a mock database specifically tailored for executing unit tests.

8.6. [Welcome Page]





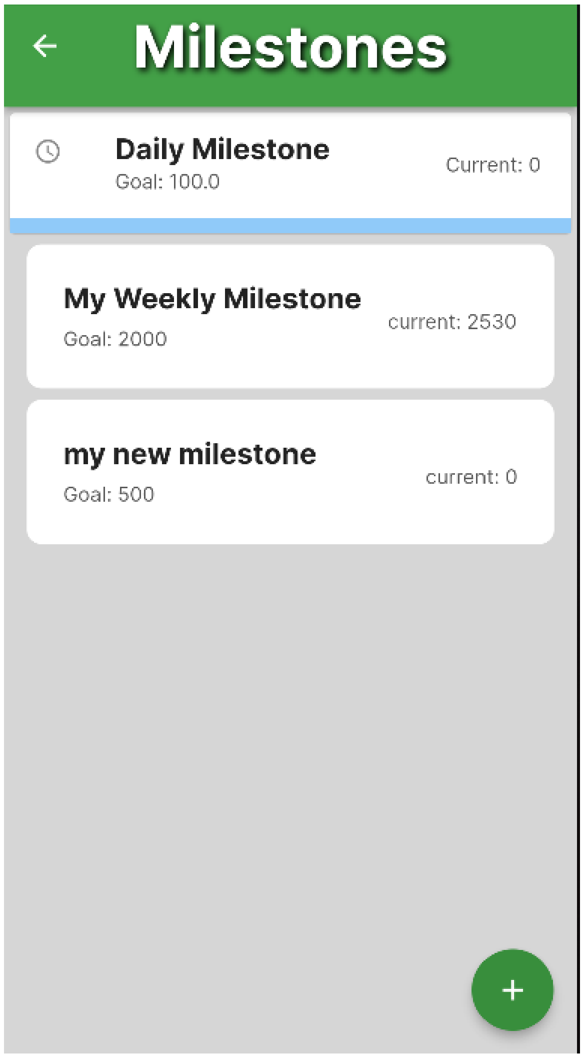
8.6.1. Functions and Interfaces Implemented

The welcome page and the register page both allow the user to sign into the application. If a user has already made an account they can login or if they forgot their password they can get a reset password link sent to them via email. If a user does not have a preexisting account they can register a new profile and then be immediately brought to the homepage afterwards.

8.6.2. Preliminary Tests

After pushing our branches into master, we have encountered an error where the sign in and register buttons no longer bring you to the homepage if you have logged in successfully. We will need to fix this issue and then retest if this bug has been fixed.

8.7. [Milestones Page]



8.7.1. Functions and Interfaces Implemented

The milestones page will allow the user to create daily or weekly goals. Each milestone will have a target distance which the user will be motivated to complete. When a milestone is created, a congratulatory message will appear.

8.7.2. Preliminary Tests

To test the milestone page, we will create a test milestone and attempt to complete the target distance set in the milestone. If the distance covered by the user satisfies the milestone requirement, the milestone will be labeled as completed.

9. Projects and Tools Used

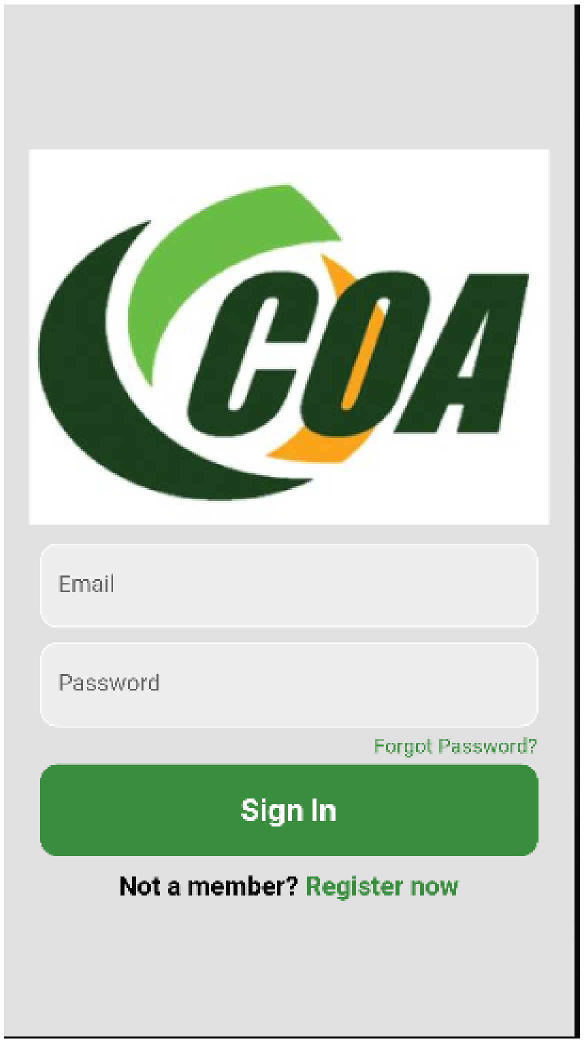
Tool/library/framework	Quick note on what it was for
Flutter	Framework for making a cross platform app
Geolocator plugin	Used to get the user’s location
Firebase	Database used to store information about users and distances
Android studios	IDE used to make the app and built-in emulator for testing

9.1 Programming Languages Used

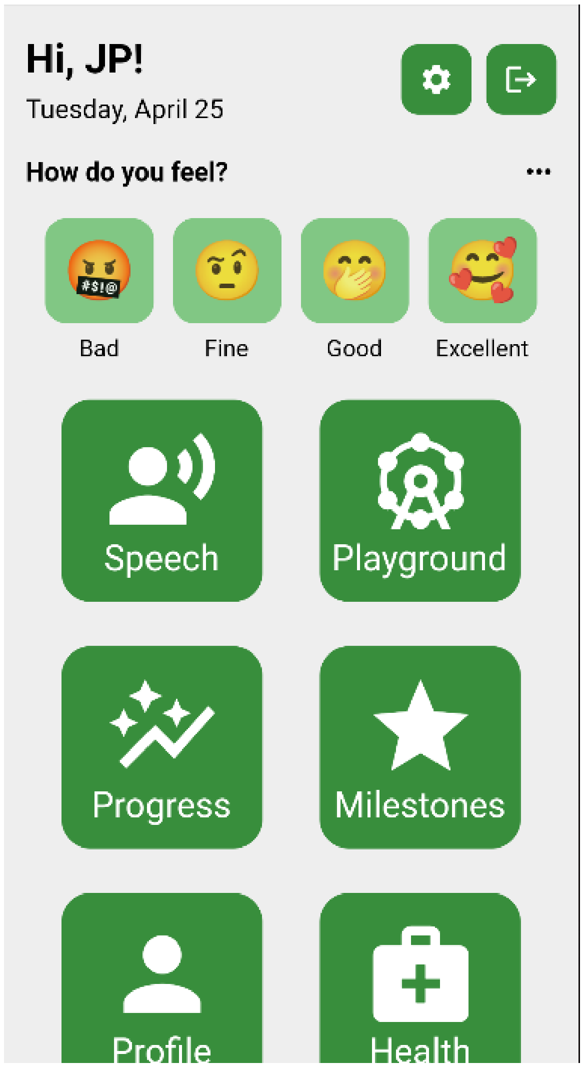
Dart	The main language used to create the full stack app.
SQL	Query language used to update, delete, and add data to our Firebase database.

10. Description of Final Prototype

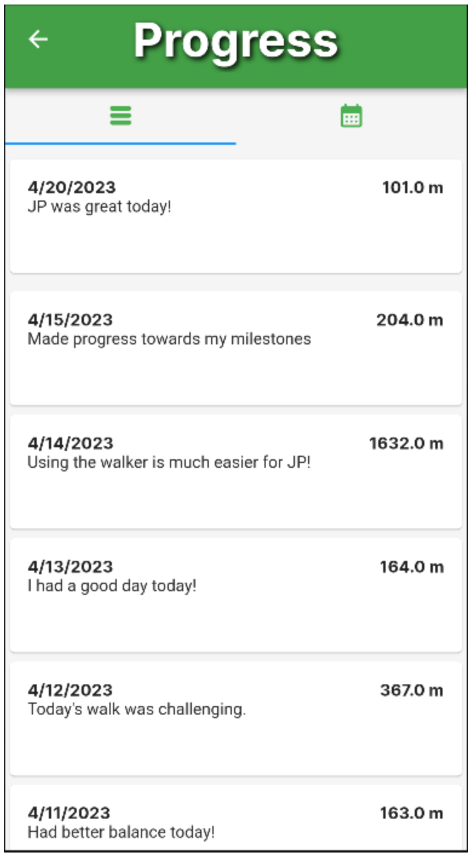
To use the COA progress app, the user will need to login or register an account to use the software.



The user will then be brought to the home page where they can submit how they feel that day.



The user will also be able to view and update their profile as well as view their progress. The progress page will show how much distance the user traveled that day as well



4/20/2023 **101.0 m**
JP was great today!

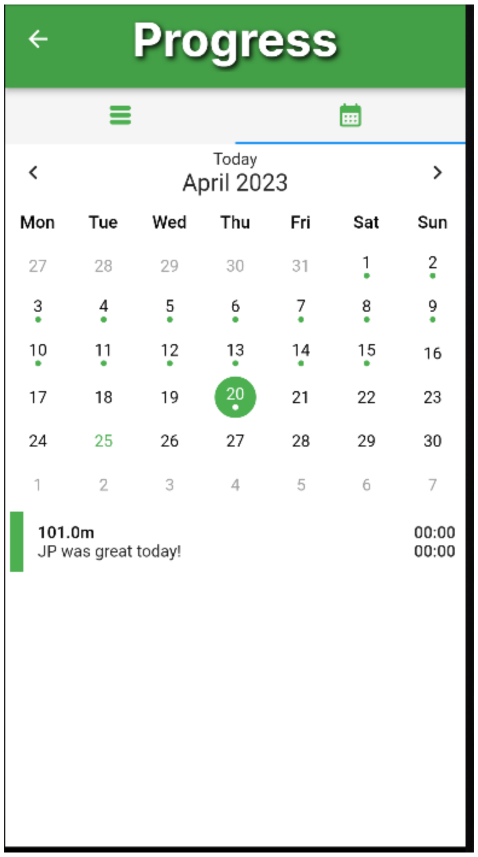
4/15/2023 **204.0 m**
Made progress towards my milestones

4/14/2023 **1632.0 m**
Using the walker is much easier for JP!

4/13/2023 **164.0 m**
I had a good day today!

4/12/2023 **367.0 m**
Today's walk was challenging.

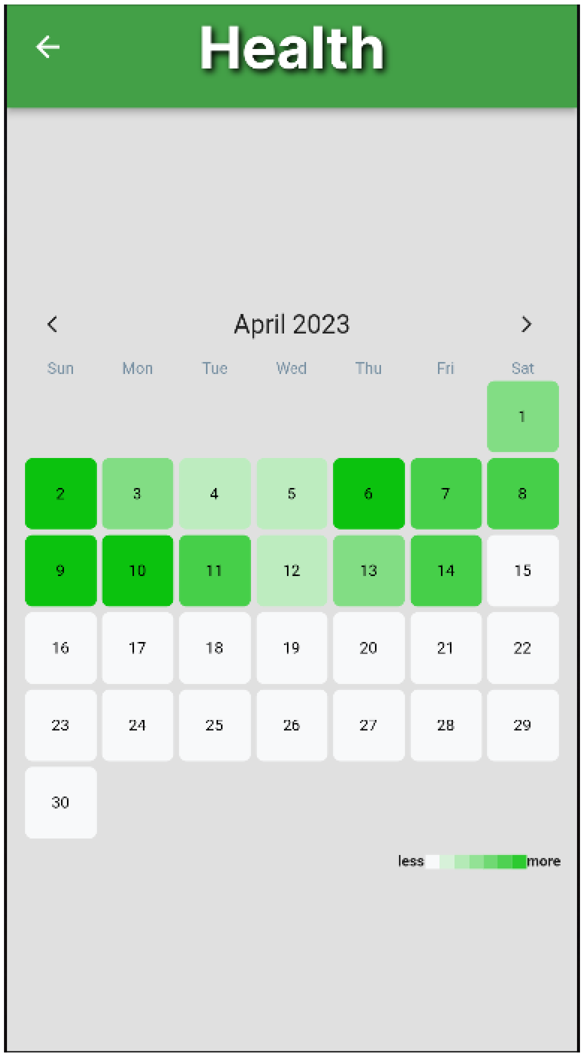
4/11/2023 **163.0 m**
Had better balance today!



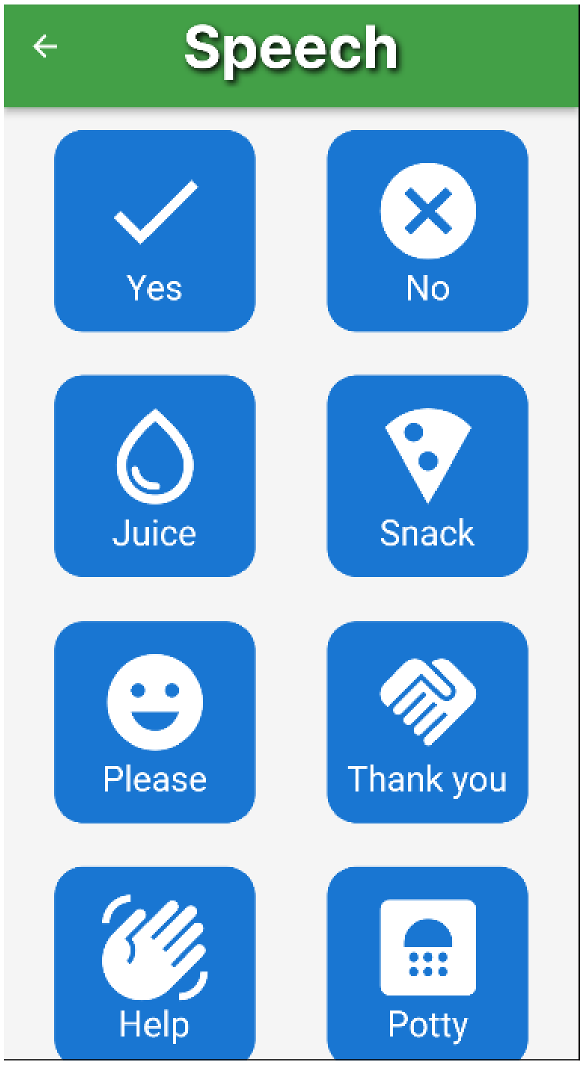
Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

101.0m 00:00
JP was great today! 00:00

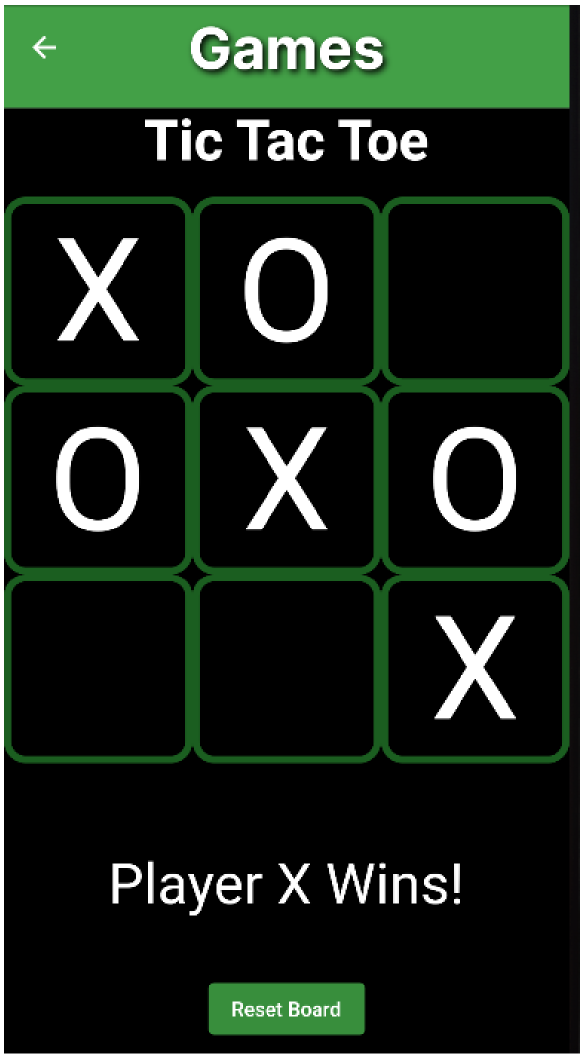
The user will also be able to view their well-being on the health page. The more opaque a calendar day is colored in, the better the user was feeling that day.



The user can open the Speech module and select a button to play an audio requesting the message they have chosen.



The game page allows a user to play Tic Tac Toe with another player. Whenever a player wins, confetti will appear to celebrate their victory.


Games
Tic Tac Toe
X
O
O
X
O
X
Player X Wins!
Reset Board

10.1. Product Delivery Status

The product will be delivered to the client on May 2nd. The source code will be given to the clients as well as instructions on how to install the application.

Project Information:
Source Code: WSUCptSCapstone-Fall2022Spring2023/coa-fullstackapp: OneStack Team COA Progress Tracking App (github.com)
Firebase: Firebase console (google.com)
Android Studio: Download Android Studio & App Tools - Android Developers
Flutter: Flutter - Build apps for any screen

To recreate the project:

1) Download Android Studio
2) Install the Flutter plugin to Android Studio
3) Clone source code from github and open the project in Android Studio

11. Conclusions and Future Work

11.1. Limitations and Recommendations

At this time, there is only about 3 months of development remaining, the app is working and needs additional features. The features are, a milestone page to view milestones, a games module for playing games, and potentially a text-to-speech feature.

11.2. Future Work

Although we've completed the main features of our project, we need to conduct further testing to identify any remaining bugs or defects in our app. We also need to test its scalability for a larger audience, which requires purchasing a subscription to increase our data storage in Firebase. We're also interested in adding user-controlled customization to the app, which would allow users to personalize their experience. Currently, the app is tailored to JP's needs, but we want to give users the ability to change the audio that plays when they reach a milestone goal, customize the app's color theme and home tile size, and display their progress in a linear graph.

12. Glossary

IDE: Integrated Development Environment

Xcode: The IDE containing tools for developing iOS applications for all Apple products.

Fitbit: A watch-like device made by a Fitbit, a fitness company, that measures various activities of the person wearing it.

Front end: Everything in which the user interacts with. This is the client-side of the application.

Back end: Everything in which the user cannot use or interact with. This deals with the server-side of things.

Full Stack Development: Developing an application front end and back end which are combined in full-stack development.

Unit Testing: Is a software testing technique which determines if individual units in a software are suitable for use or not. Also known as white-box testing.

Functional Testing: Is the process which quality assurance determines if a piece of software is acting in accordance with its intended purpose. Also known as black-box testing.

Flutter: An open-source software development kit with cross-platform capability to build apps for Windows, macOS, Android, iOS, Linux, and more.

13. References

[1] "What's new for Apple Developers," One Apple Park Way, Cupertino, CA Apple Inc. 2022, https://developer.apple.com/whats-new/.

[2] Patel, Bhavel, "How to develop an iOS App [A Complete Guide]," *Space-O Technologies*, September 2, 2022 https://www.spaceotechnologies.com/blog/how-to-develop-an-ios-app/.

[3] "Fitbit Help." Fitbit MyHelp, https://help.fitbit.com/articles/en_US/Help_article/1133.htm#:~:text=Export%20Fitbit%20Account%20Data%201%20From%20the%20fitbit.com,to%20download%20your%20Fitbit%20account%20data.%20See%20More.

[4] Mijacobs. (n.d.). *What is continuous delivery? - azure DevOps*. Azure DevOps | Microsoft Learn. Retrieved October 31, 2022, https://learn.microsoft.com/en-us/devops/deliver/what-is-continuous-delivery

[5] Android Studios, “Test in Android Studio : Android Developers”, Android Developers. https://developer.android.com/studio/test/test-in-android-studio (Accessed: 31-Oct-2022).

[6] Amazon Web Services, *“Explore the AWS platform, cloud products, and capabilities”,* Amazon. Free Cloud Computing Services - AWS Free Tier (amazon.com)

[7] Fitbit, “Fitbit Device API Reference”, Fitbit. https://dev.fitbit.com/build/reference/device-api/ (Accessed October 31, 2022).

[8] Apple Developer, “Develop apps for iOS”, Apple. https://developer.apple.com/tutorials/app-dev-training (Accessed October 31, 2022).

[9] Positive Affirmations, *“Positive Affirmations: Definitions, Examples, and Exercises"*, Betsi Sites & Tchiki Davis, MA, PhD. Positive Affirmations: Definition, Examples, and Exercises - The Berkeley Well-Being Institute (berkeleywellbeing.com)

[10] The Exploratorium, “Tic Tac Toe”, The Exploratorium. https://www.exploratorium.edu/brain_explorer/tictactoe.html (Accessed October 31, 2022).

[11] Ryan Black, “System Testing”, TechTarget. https://www.techtarget.com/searchsoftwarequality/definition/system-testing#:~:text=System%20testing%2C%20also%20referred%20to,full%2C%20integrated%20system%20or%20application (Accessed October 31, 2022).

[12] Alin Turcu, “A view on testing Android apps”, ProAndroidDev. December 18, 2018. https://proandroiddev.com/writing-integration-tests-in-android-b0436978ed7b . (Accessed October 31, 2022).

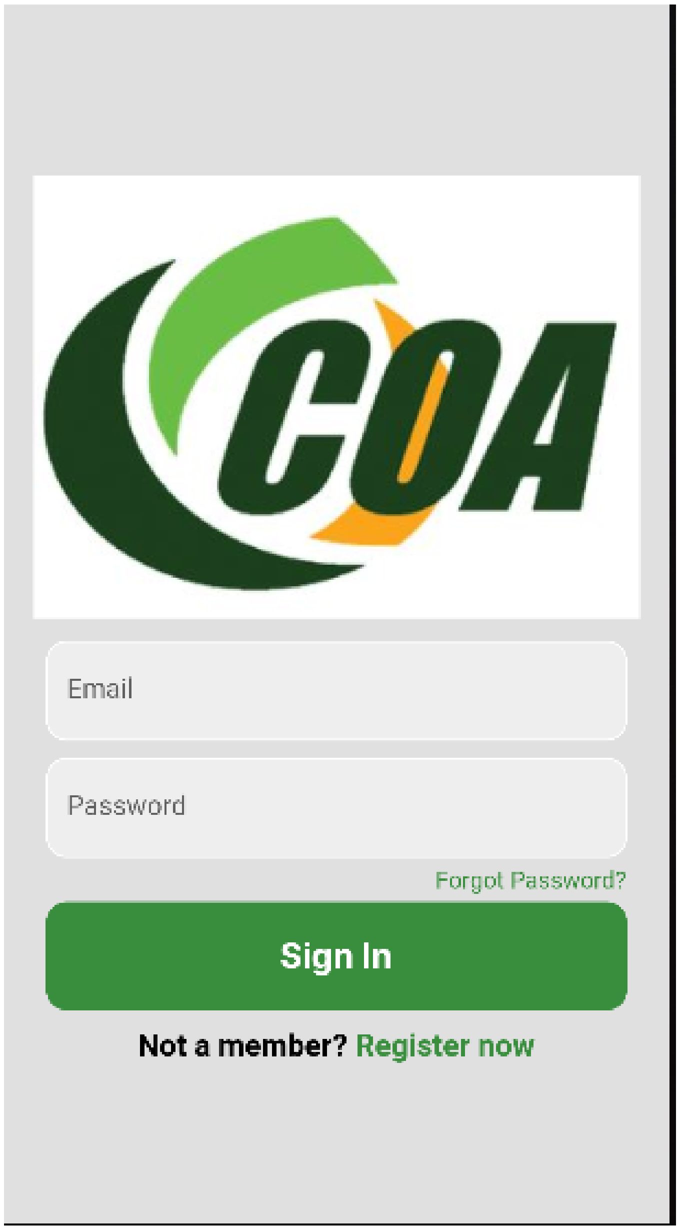
[13] chitrasingla2001, “Functional vs Non-functional requirements”, GeeksForGeeks. https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/ (Accessed October 31, 2022).

[14] Flutter, “Flutter Documentation,” Flutter. https://docs.flutter.dev/ (Accessed October 26, 2022).

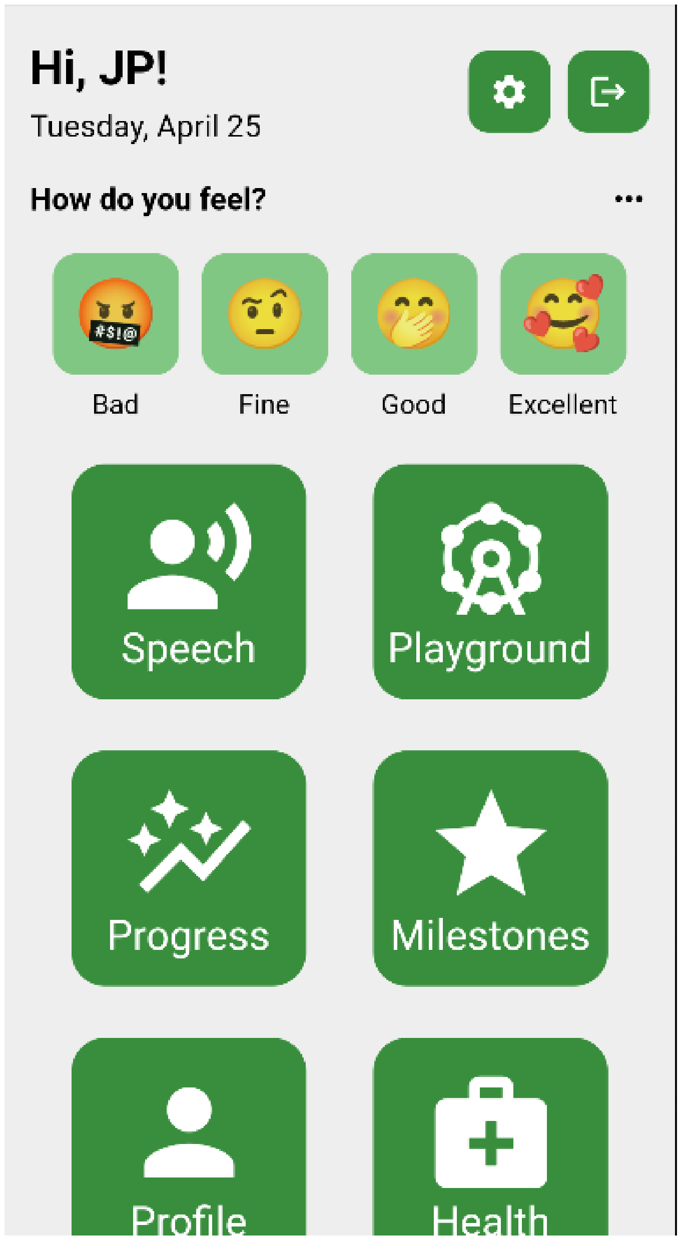
[15] Practical Coding, “How to upload an android studio project to GitHub. YouTube. https://www.youtube.com/watch?v=GhfJTOu3_SE&list=PL5z-9TsivGUOQ7z6y5jD_gZczr3V5FLmG&index=4. January 7, 2021. (Accessed October 22, 2022).

14. Appendices

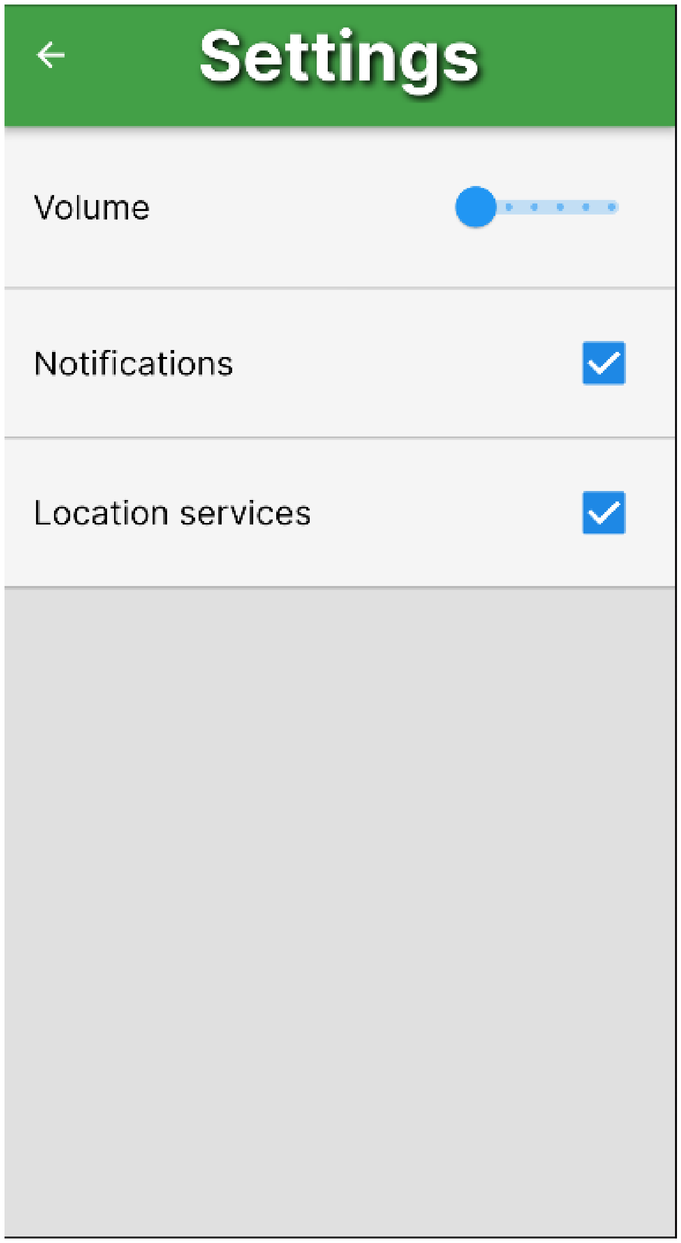
14.1 Appendix A - App Pages



(Image 1)


Hi, JP!
Tuesday, April 25
How do you feel?
Bad
Fine
Good
Excellent
Speech
Playground
Progress
Milestones
Profile
Health

(Image 2)



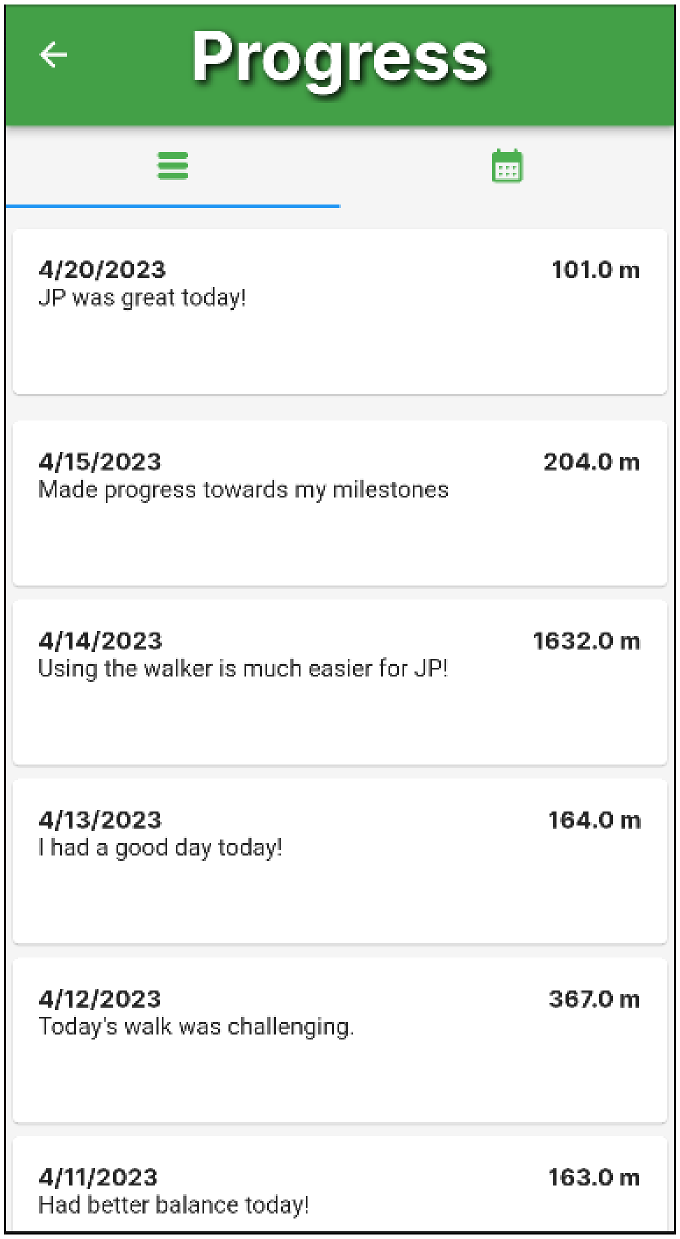
Settings

Volume

Notifications ✓

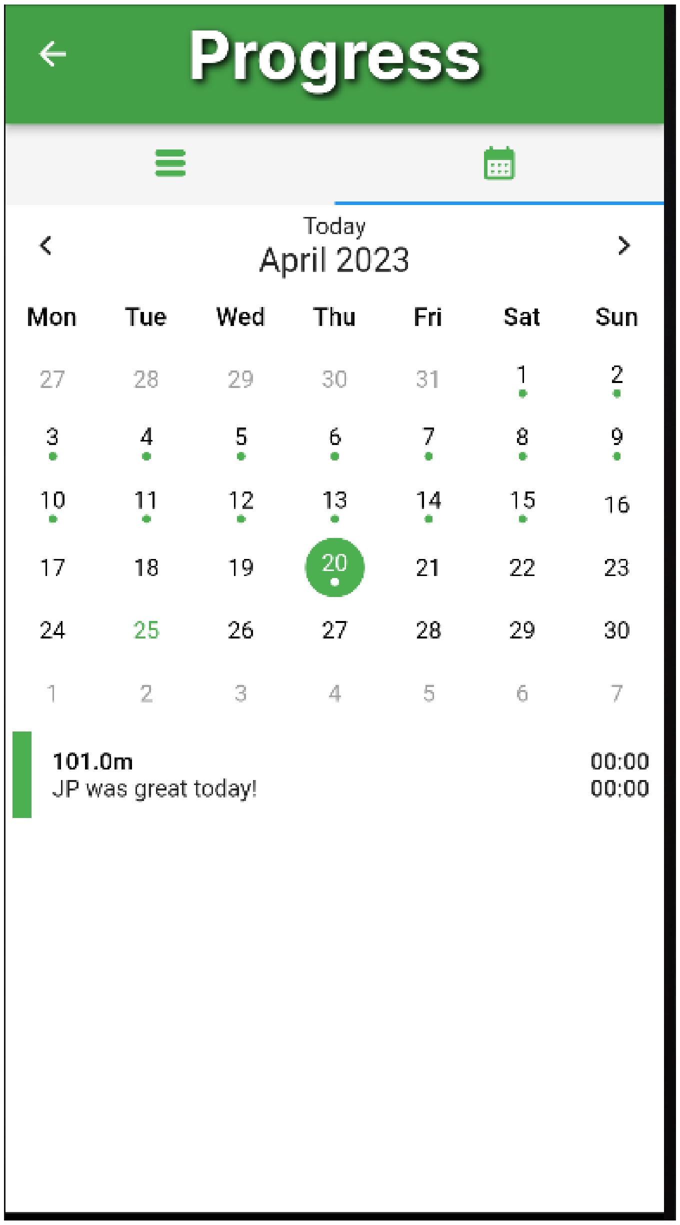
Location services ✓

(Image 3)



4/20/2023 JP was great today!	101.0 m
4/15/2023 Made progress towards my milestones	204.0 m
4/14/2023 Using the walker is much easier for JP!	1632.0 m
4/13/2023 I had a good day today!	164.0 m
4/12/2023 Today's walk was challenging.	367.0 m
4/11/2023 Had better balance today!	163.0 m

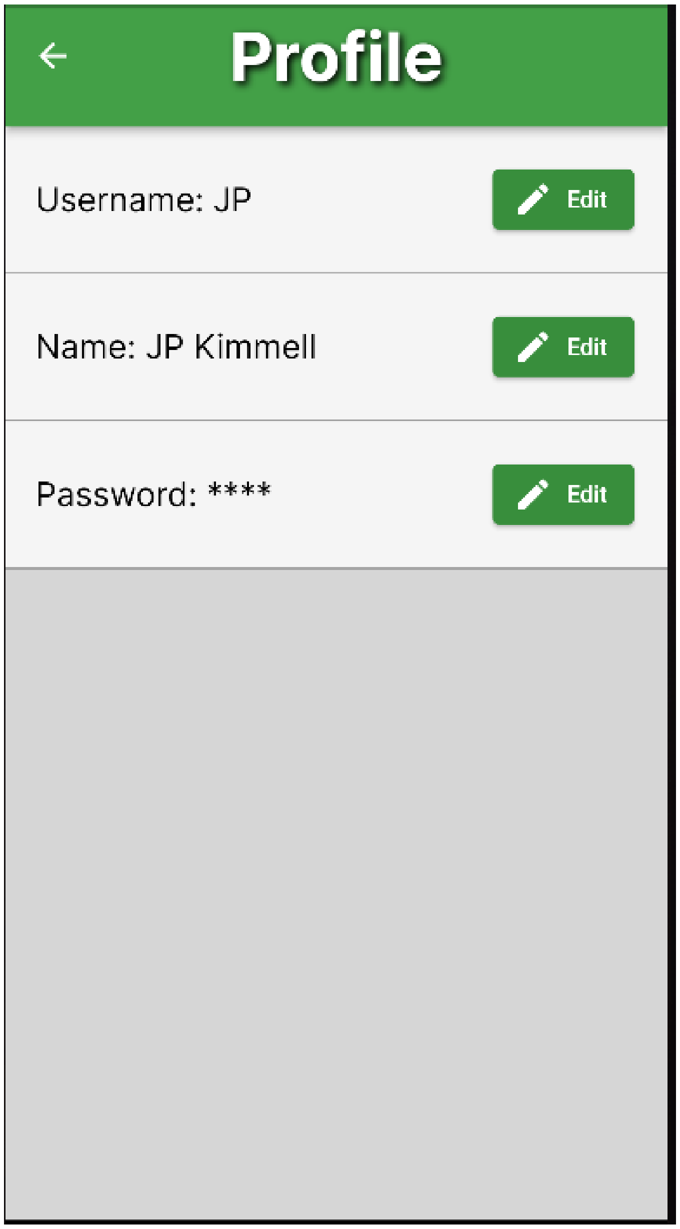
(Image 4)



Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

101.0m 00:00
JP was great today! 00:00

(Image 5)

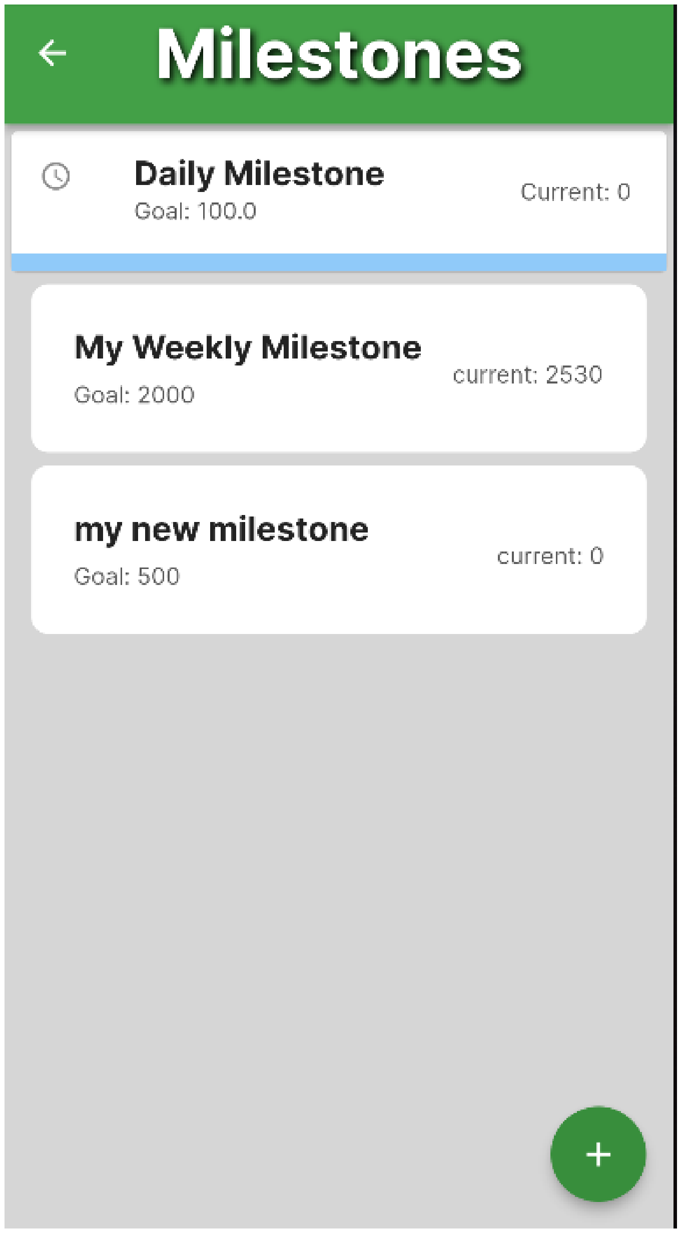

Profile

Username: JP

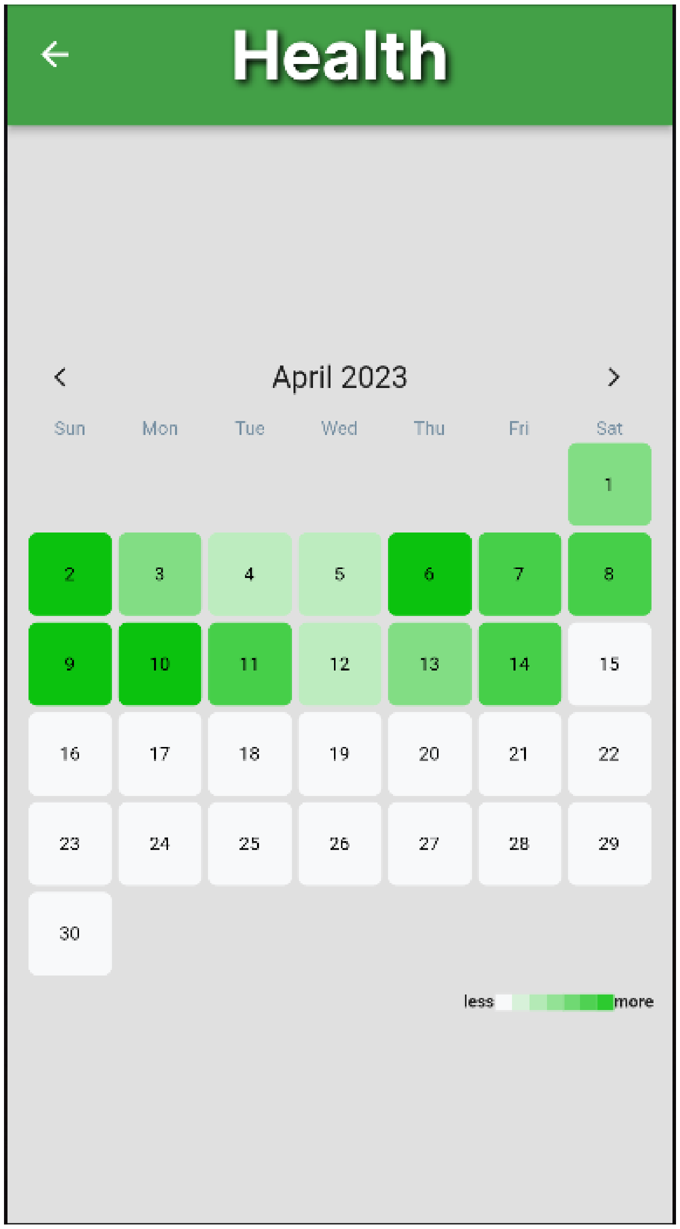
Name: JP Kimmell

Password: ****

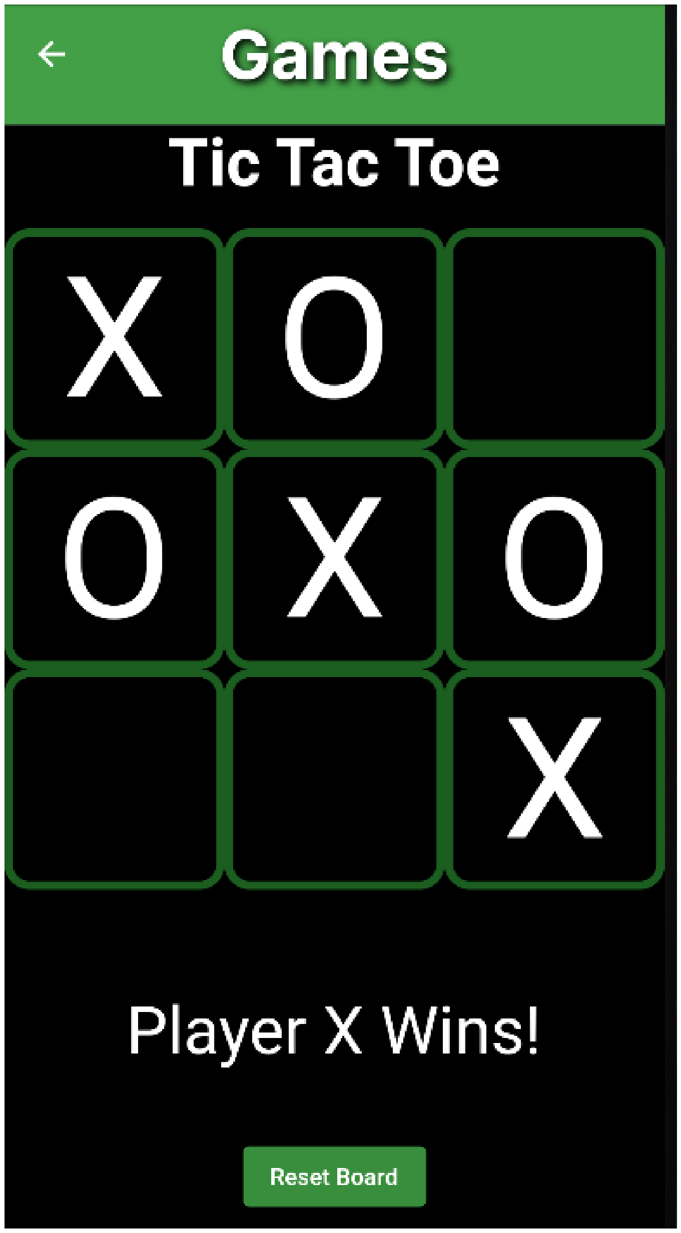
(Image 6)


Milestones
Daily Milestone
Goal: 100.0
Current: 0
My Weekly Milestone
Goal: 2000
current: 2530
my new milestone
Goal: 500
current: 0
+

(Image 7)

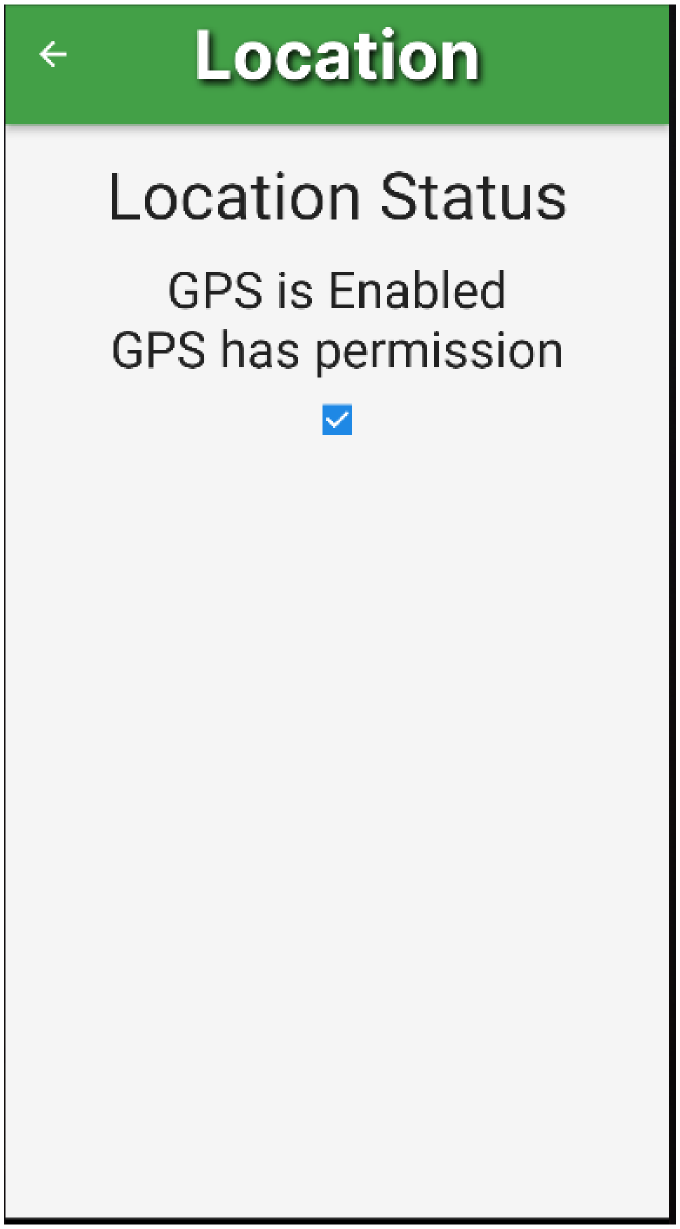

Health
April 2023
Sun
Mon
Tue
Wed
Thu
Fri
Sat
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
less
more

(Image 8)



Player X Wins!

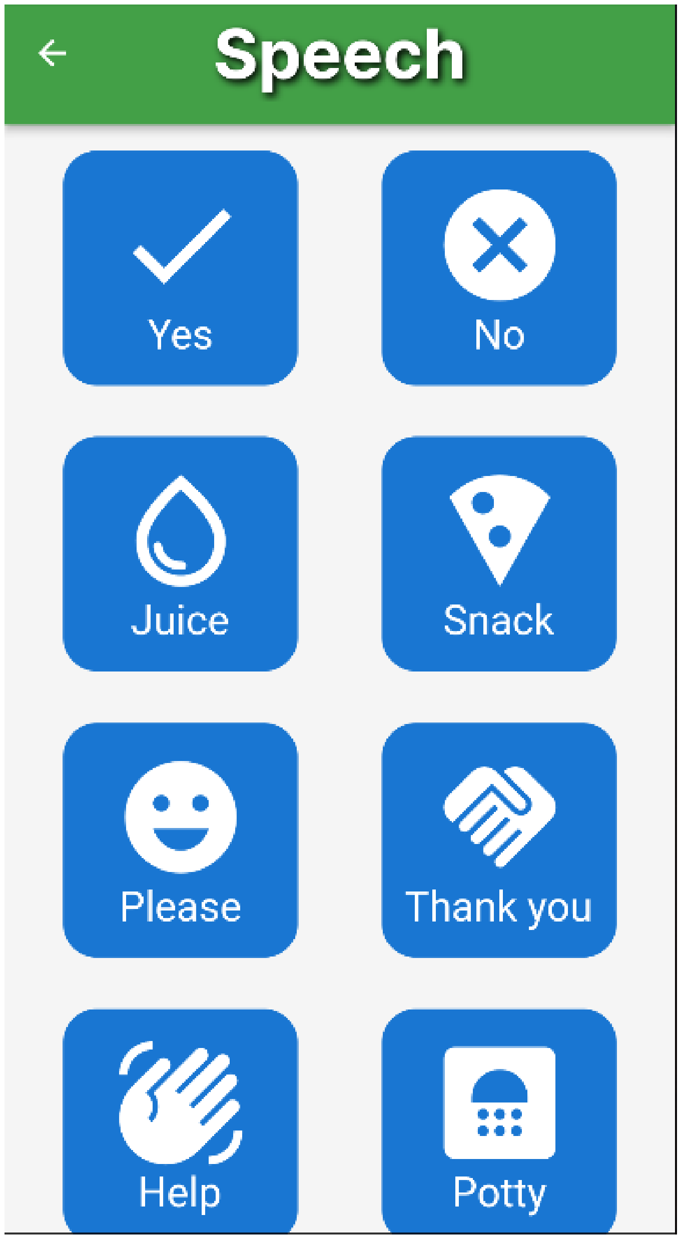
(Image 9)



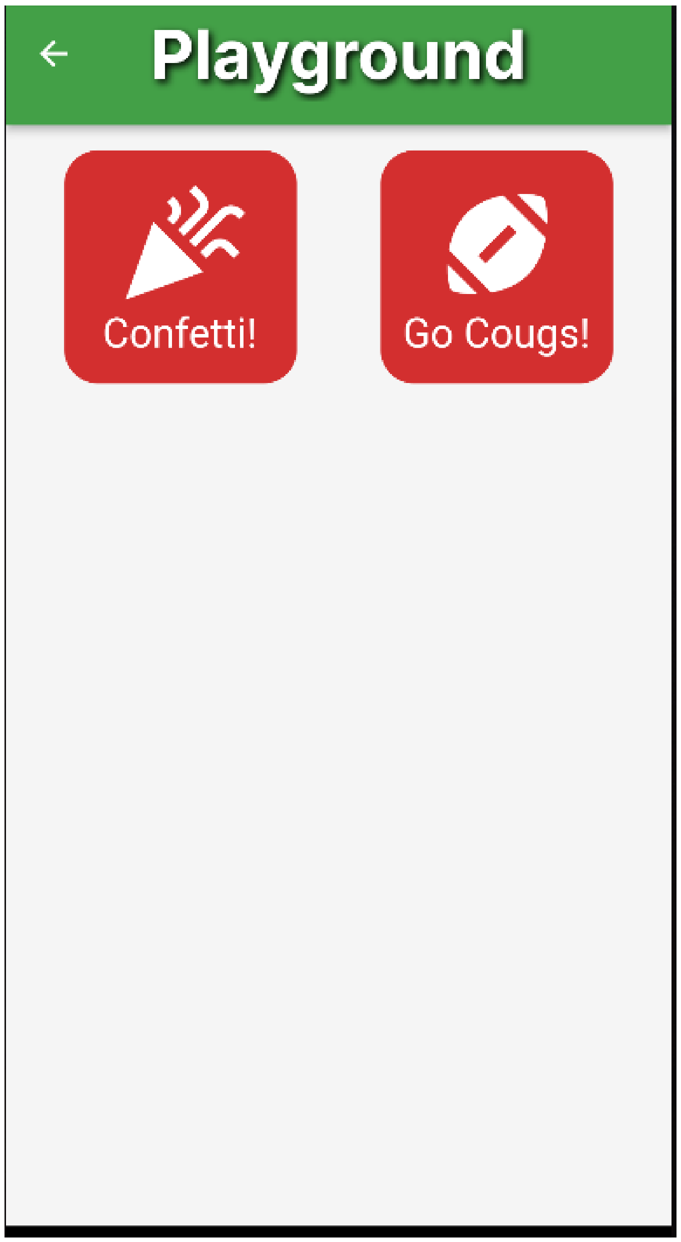
Location Status

GPS is Enabled
GPS has permission

(Image 10)



(Image 11)



(Image 12)

14.2 Appendix B - Team Information

Team Leader

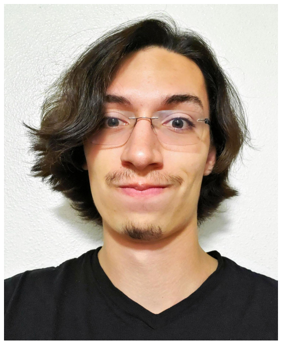
Name: Koji Natsuhara
Major: Computer Science

Team Member

Name: Dileep Chokar
Major: Computer Science

Team Member

Name: Luke Martinez
Major: Computer Science



14.3 Appendix C - Project Management

Our team followed a weekly schedule where we would meet together to discuss our progress and to work on the project. We would also host a bi-weekly with our clients to share our progress and receive any feedback that they have. We would usually meet a day before each week's sprint ended, during which we evaluated our project's progress, demonstrated our work, and discussed challenges. The meeting typically entailed a report on the current sprint's progress, answering work-related questions, and discussing future work for the next sprint. We also addressed significant blockers or issues. This approach proved effective in keeping our clients informed and addressing our concerns. The second weekly meeting functioned as a sprint planning meeting, which occurred after the client meeting. During this meeting, we reviewed progress from the previous sprint, resolved any pending issues, and created the sprint board for the upcoming sprint. We also discussed any obstacles team members were facing and scheduled follow-up meetings as necessary.