COMP225 Section 004

Team Project (Final)

Team 4

Venzon Ariola, Angelo Caster Carino, Wonyoung Chung, Jan Rodel Escareses, Mar Jerico Lagmay, Albert Obcena, Andreas Themistocles

8-19-2022

Contents

Section 1 “Introduction” and subsections

1.1 Purpose

This product is designed to be a well-being check program that tracks a user’s screen time, hydration, heart rate, sleep cycle, number of steps taken daily, etc. so that they can work well with minimal fatigue and better health. This is version 0.1.0 of the program.

1.2 Document Conventions

Table of Typographic Conventions

Convention	**Description**
Boldface & Italicized	Identifies main titles and/or main subtitles
Boldface	Identifies headers and/or section titles
Italics	Used when referencing third-party media or third-party quotes

Table of Acronyms

Acronym	**Description**
CSS	Stands for Cascading Style Sheets. A computer language used in conjunction with HTML to augment the aesthetics of a webpage.
DBMS	Sands for Database Management System. A software system that provides users with the ability to store, manage, and manipulate data on computer hardware.
HTML	Stands for Hypertext Markup Language. A computer language used to design and create web pages.
IT	Stands for Information Technology. An industry field encompassing computer software, computer hardware, and the interactions between them and humans
JSON	Stands for JavaScript Object Notation. A file format derived from JavaScript.
OS	Stands for Operating System. A software system that manages the interactions between computer hardware and user-level software applications
PHP	Stands for Personal Home Page. A scripting language used in web development
SQL	Stands for Structured Query Language. A computer language used in numerous commercial database systems.

Table of Names

Name	Description
JavaScript	A scripting language used for client-side web applications
PostgreSQL	An open-source database system
Python	A scripting language used for data science, mathematics, and machine learning applications.

Table of Terms

Term	Description
Account	A private and secure compartment of the system database through which a user can access system features and personal information.
Notification	An automated message prompt that briefs the user of certain pieces of information.
Settings	A place in the software where a user can augment various aspects of their account and manipulate pieces of data and information associated with their account.
Sign Up	The process by which a user initializes an account within the system to access the software features.
Drop Out	The process by which a user terminates their account on the system database. All information about the user on the database is deleted.
Sign In	The process through which a user gains access to their account on their own personal device.
Sign Out	The process through which a user terminates access to their account on their own personal device.
User	A person who interacts with the software with the purpose of gaining utility from its features.
User Information	Within the context of this document, “Information” refers to personal user data (such as name, age, and occupation) and user health data (such as height, weight, sleep schedule, time spent using a computing device, and a number of steps taken in a day).

1.3 Intended Audience and Reading Suggestions

This document is intended for the project managers, developers, documentation writers, and marketing staff working on the program, as well as its target demographic (Office workers, IT workers, and Students) and testers. This document contains information regarding the development and specifications of the program.

1.4 Project Scope

Nowadays, most people are constantly on their devices, may it be their mobile phones, laptops, or desktop computers. Due to this, some health problems arise, such as obesity, sleep deprivation, and depression. HowAreYou is a new program designed for people who are on their devices, without forgetting about their wellbeing. HowAreYou runs on multiple platforms: Windows/Mac, iOS/Android, and WearOS. By using HowAreYou, the user can keep track of their screen time, hydration, sleep cycle, heart rate (exclusive to HowAreYou smartwatch app), and number of steps taken daily (exclusive to HowAreYou smartwatch and mobile app).

1.5 References

The negative effects of screen time for adults and children. Mosley. (2021). Retrieved from The Negative Effects of Screen Time for Adults and Children - Blog - Valleywise Health

*Screentime and the Brain***.** Ruder. (2019). Retrieved from Screen Time and the Brain

Section 2. Overall Description

2.1 Product Perspective

1. This new software will monitor the user's screen time and hydration level when using digital devices.
2. This new software will collaborate and use existing/available wellness data as a reference.

2.2 Product Features (Functions)

1. Collect and store information about the user
2. Generate the average screen time and hydration level of the user.
3. Provide historical data of the user.
4. Alert the user to stretch/move away from digital devices and hydration level.
5. Alert the user to stretch/move away from digital devices and hydration level.
6. This will connect to Wi-Fi and Bluetooth.
7. This will use the location services available to mobile devices.
8. Will connect to Google Maps.

2.3 User Classes and Characteristics

1. Students -will use the new software/application to validate their time from using any electronic device for studying and leisure activities.
2. Office Workers -will use the new software/application to have a time limit in case a person is over-working, which can cause some unexpected scenarios such as fatigue, dizziness, etc.
3. Programmers - will use the new software/application to make programmers divide their time between work and any physical activities that will help them maintain their fitness level to avoid any kind of sickness.
4. Professors/Teachers - will use the new software/application to help them have a scheduled agenda for their class and when making lecture notes. This will avoid stress and being tardy.

2.4 Operating Environment

- **Software:**
  - PostgreSQL will be used to store users’ information.
  - Will be developed in JavaScript, HTML, CSS, and Python.
  - Will support windows and iOS. For desktop operating systems.
  - Will support android and iOS for mobile devices.
  - The software can be used both in the developed application and on the web.
  - The application can work on any version but depends on the processor of the device.

2.7 Assumptions and Dependencies

- The software will operate using desktop operating systems through Windows. Some devices may not be updated to the most recent version of Windows. This could affect how the software will work on the device.

- Some old, outdated, mobile device processors may not be compatible with the software, thus, could affect the performance of the software.
- Incomplete or inaccurate existing wellness data could affect the historical data that the software will provide.

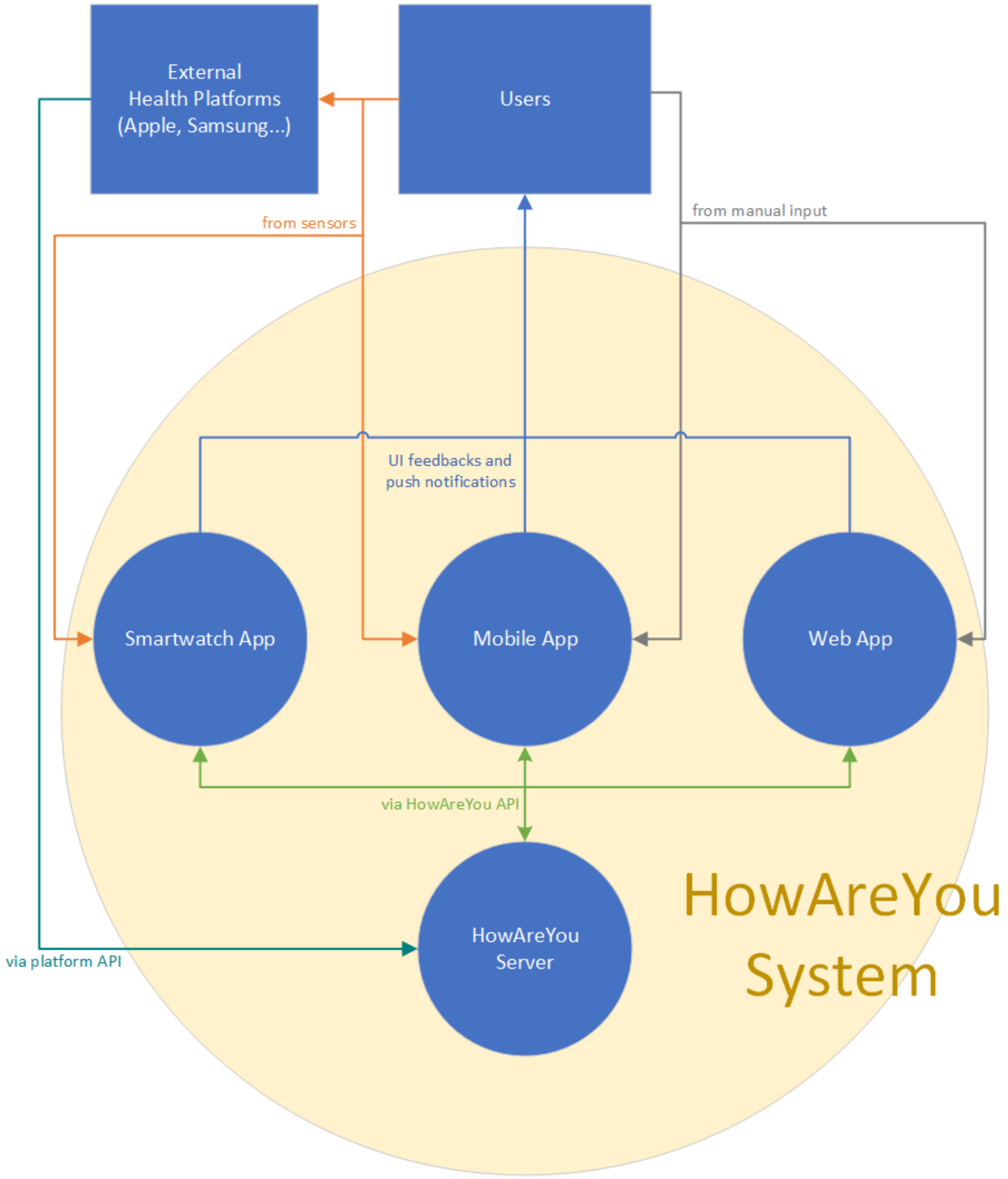
Section 3. External Interface Requirements

3.1 User Interfaces

- **Web App:**
    - In the web app, users can do the most interactions, including
        - Sign up / drop out.
        - Sign in / sign out.
        - Manually record health information including water intake, computer usage time, blood pressure, etc.
        - Manually edit/delete health information.
        - Get/track/manage notifications from the HowAreYou app.

- **Mobile App:**
    - In the mobile app, in addition to what users can do in the web app, users also can do mobile-specific actions, including
        - Set notification options.
        - Check pairing status with the smartwatch app.

- **Smartwatch App:**
    - In the smartwatch app, users can do the following actions.
        - Sign in / sign out.
        - Decide which data will be shared with the HowAreYou app.
        - Get notifications from the HowAreYou app.


External
Health Platforms
(Apple, Samsung...)
Users
from sensors
from manual input
UI feedbacks and
push notifications
Smartwatch App
Mobile App
Web App
via HowAreYou API
HowAreYou
Server
via platform API
HowAreYou
System

3.2 Hardware Interfaces

- **Web App Interface**
  - Input from either touch input or keyboard and mouse.
  - Output to device screen.
  - Output to web browser notification.

- **Mobile App**
  - Input from either touch input or keyboard and mouse.
  - Input from hardware buttons.
  - Input from various medical sensors in each device, including but not limited to work out measurements and sleep patterns.
    We can gather these data directly from the sensors or platform APIs.
  - Input from other sensors, including microphone and photoresistor.
  - Output to device screen.
  - Output to push notification (may include vibration and sound alarm).

- **Health Monitoring Device (including Apple Watch and Galaxy Watch)**
  - Input from touch interfaces on the watch face.
  - Input from hardware buttons.
  - Input from various medical sensors in each device, including but not limited to heart rate, heart rhythms, blood oxygen, workout measurements, and sleep pattern.
    We can gather these data directly from the sensors or platform APIs.
  - Input from other sensors, including microphone and photoresistor.
  - Output to device screen.
  - Output to push notification (may include vibration and sound alarm).

3.3 Software Interfaces

- **Web/Mobile Common Pages**
  - **Title**
    - Sign in
    - Sign up
  - **Main page**
    - Health info dashboard
    - Notifications
  - **Manage Health info**
    - Add / remove / modify data type
    - Add / remove / modify data items of each type
  - **Settings**
    - Sign of dialogue
    - Drop out dialogue
- **Mobile-only Pages**
  - **Settings**
    - Notification options
    - Check pairing status with the smartwatch app
- **Smartwatch Pages**
  - **Title**
    - Sign in
  - **Main page**
    - Health info dashboard (simple)
    - Notifications
  - **Settings**
    - Sign out dialogue
    - Notification options
    - Data sharing options

3.4 Use Case Table

Use cases			
Use Case Name	**List of related Requirements ID**	**Actor(s)**	**Brief Description**
Pause All Notifications	FR04	Daily Health Check Users	The actor will click on the settings button and navigate to a menu option that stops all notifications during the actor does not want the software to interfere with their work.
Create User Profile	FR02, FR03	Initialization Users	The actor will select create an account and the software/application will present a profile form that needs to be completed to create the profile. The software/app will send an email to confirm profile creation. The actor will need to access the email verification link to complete the profile creation.
Grant Permission Access	FR07, FR08, FR11, FR13	Daily Health Check Users	The actor will be prompted by the app to accept the system access requests for certain functionalities, such as the internal clock, GPS location, internet connectivity, and data storage in the device. After accepting these permissions, the app can now provide full functionality to the actor.
Connect to Social Media	FR02, FR11	Daily Health Check Users	The actor will link their social media account with the program. After the account is linked, the actor can now share their information through the social media platform, such as the number of kilometers they have walked.
Set Alert Notifications	FR01	Daily Health Check Users	The actor will click on the settings button. In the settings, there will be options for the user to augment the type of notifications that they will receive on their device(s). The content of these notifications, and the frequency that they are sent to the actor will depend on what settings the user changes and allows.
Report Feedback	FR05, FR06	Daily Health Check Users	The actor can find a button titled “Contact Us” in the FAQ tab of the program. When the actor clicks on it, it will lead them to a webpage containing questions regarding their feedback. After the actor fills out the forms, the answers will be sent to the developers.

Download and Install the Software	NFR03, NFR04	Initialization Users	The actor will download the application from an app store or other website. The software application will be installed and run on whatever operating system that the actor uses. The actor can also use the application to interact with whatever third-party web browser they choose.
Set Privacy Settings	NFR01	Initialization Users	The actor will click on the settings button and navigate to a section of the menu that gives the actor options on what type of personal data will be stored on the system servers, and what type of personal data will be stored in the on-device storage system.
Set Language Setting	NFR05	Initialization Users	The actor will download the software from an app store or other website. Prior to installation, the actor will be prompted with a small menu, whereby the actor can select the language they wish to use the software with. After installation, the user can click on the settings menu, and find an option to change their language, should they wish to do so.
Set Accessibility - Low Vision Setting	NFR06	Initialization Users	The actor will click on the settings menu and navigate to a series of menu options that can augment that scale of the content on the screen, including text, graphics, and other user aspects of the user interface. The actor can adjust these settings to help their interactions with the application.
Set Accessibility - Color Blindness Setting	NFR07	Initialization Users	The actor will click on the settings menu and navigate to a series of menu options that can augment the colour of the contents on screen, including graphics and other user aspects of the user interface. The actor can adjust these settings to help their interactions with the application.
Update App	NFR02	Daily Health Check Users	The actor will click on the update menu, which will notify the actor if there is a software update available. The actor can choose to install the update or ignore it. The actor can also select an option that automatically installs any software update whenever they become available.

Section 4. Functional Requirements List

Requirement ID	Requirement Title	Short Description	Priority	Requester
FR01	Alerts	The application should give the user mobile notifications / alerts.	High	End User (Stay-at-home Father)
FR02	Wellness App Connect Integration	The system should allow the integration of "Wellness App Connect" functionality for advising and notifying users of any new up-coming online sessions through text messages.	Medium	End User (Marketing Officer)
FR03	Profiles	The application should have profiles based on what specific features the users would like to use.	Medium	End User (Marketing Officer)
FR04	Non-Interference	The app should not disrupt any of the work being done by the end user.	High	End User (Systems Analyst)
FR05	Feedback Form	The new app should allow the users to give their feedback.	Medium	End User (Programmer)
FR06	Frequently asked questions (FAQ)	The system should have the capability to allow the administrator to set up and manage frequently asked questions.	Low	End User (Customer Services Representative)
FR07	Usability Without Internet Connection	The app must function as intended with or without an internet connection.	High	End User (Professor)
FR08	Fitness Tracking	The app allows users to track fitness data.	Low	End User (Student)
FR09	Voice Command	The app allows the users to set up voice commands to conveniently use the app.	High	End User (Customer Service Representative)

FR10	Data storage and interoperability	The app allows users to save and store fitness information in various standardized data formats to be sent to doctors that serve the user for improved healthcare delivery.	Low	End User (Stay-at-home Father)
FR11	Social Media integration	The app allows users to share various fitness information and fitness progress on social media.	Low	End User (Student)
FR12	GPS Integration	The app allows users to track their running / cycling routes throughout their environment and record the time it takes to navigate it.	Medium	End User (Professor)

4.1 Use Case Formal Descriptions

Use Case: Set Alert Notifications

Primary actor: Daily Health Check Users

Goal in context: Health check review for the week using your mobile app or web

Preconditions: 'Health Check Users' and 'Initialization Users' must configure the proper set-up of the “Wellness Check Program” for accurate data to be retrieved.

Trigger: User wants to know the previous week activity for health evaluation

Scenario:

1. The user opens up HowAreYou through the app/program or through their web browser.
2. The user enters their user credentials (email, password)
3. The system will then display all the trackers the user has previously used.
4. The user selects a specific feature that they would like to see previous logs of.
5. After selecting a feature, the user will scroll down and select the “see previous activity” tab.
6. The system will then display the user’s previous activity for that specific feature.
7. The system will then display their current activity levels and compare it to the previous week.

Exceptions:

1. Email or password is incorrect
2. The user is using the feature for the first time - it would display no data

Priority: High Priority

When available: Second Increment

Frequency of use: Frequent

Channel to actor:

1. Web browser (PC and mobile)
2. Mobile App

Secondary actors:

1. Smart watches with health monitoring features
2. Smartphones

Channels to secondary actors:

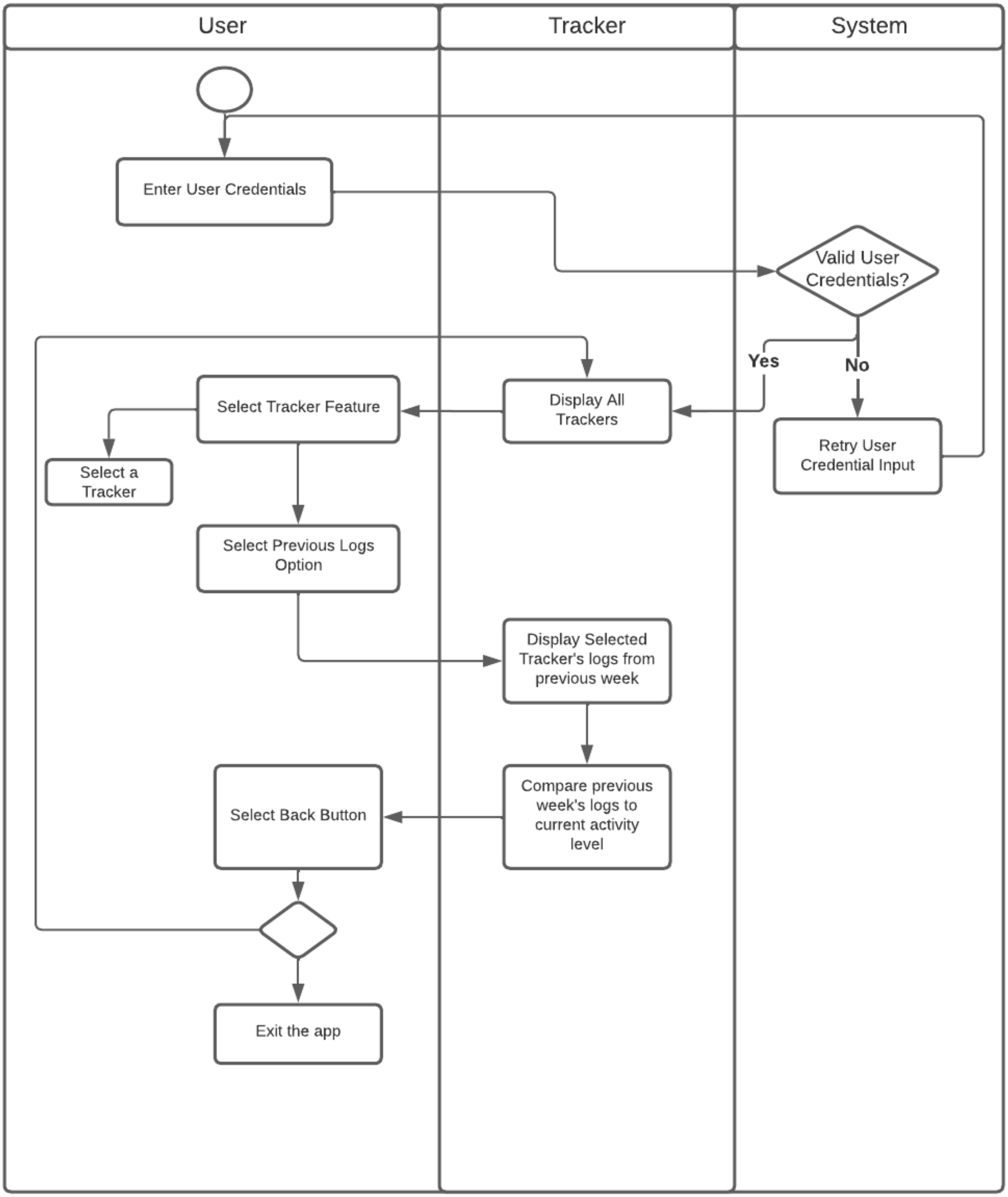
1. Smart watches: APIs (REST, WebSocket) between HowAreYou server and watch app
2. Smartphones: APIs (REST, WebSocket) between HowAreYou server and mobile app

Open Issues:

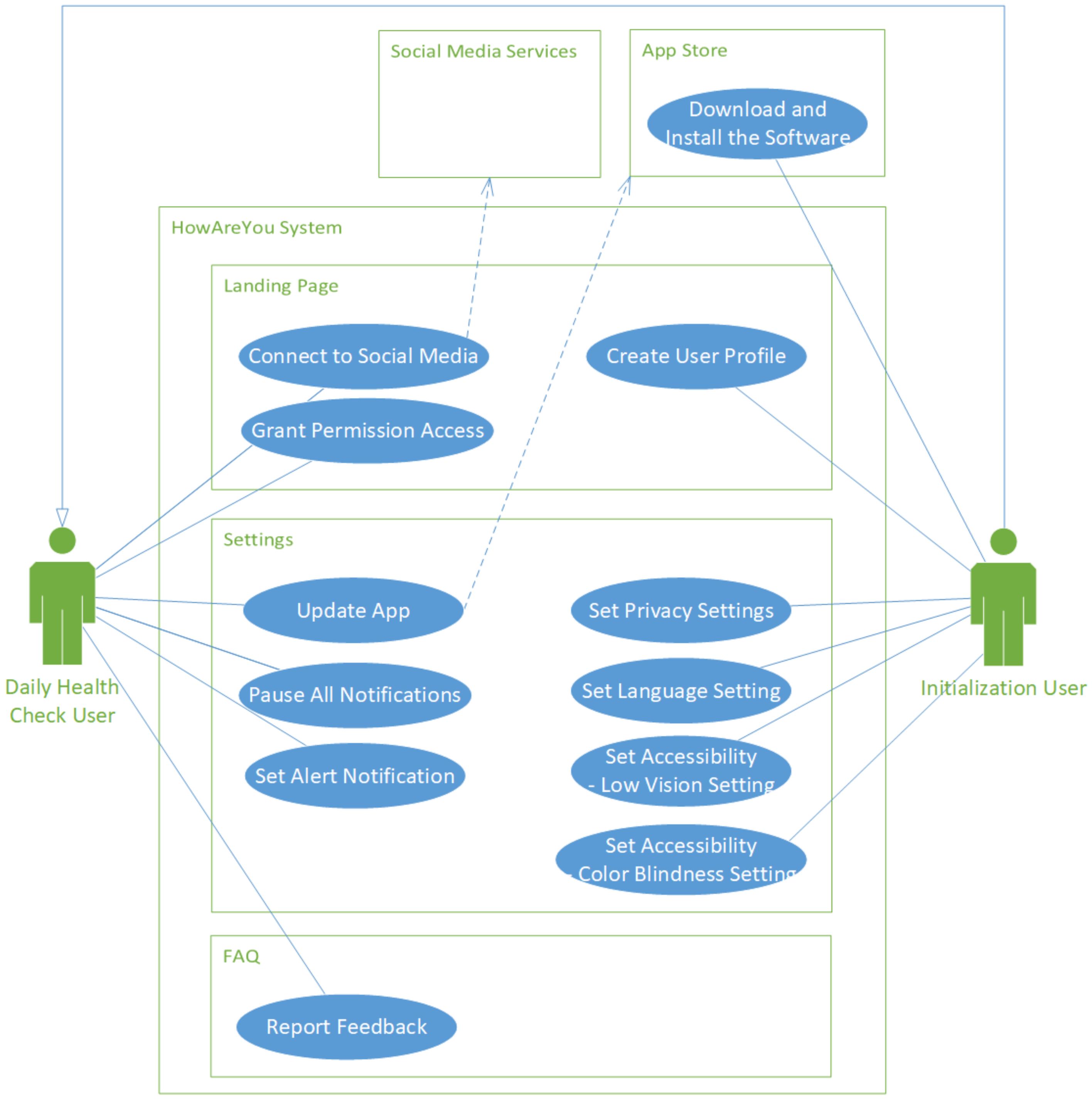
1. Will we develop a more in-depth tracker system for each feature?

2. Will we add a feature which can show activities longer than a week?
   e.g.) Two weeks, months, or a year before?
3. What can protect the data of the users from unauthorized access?
4. Is this feature easy enough for every user? As the HowAreYou system's main feature, we should endeavour to make it easy for everyone.
5. When the time zone of the mobile phone is changed, what should we do?
   e.g.) Automatically change each time-based alert to the current time zone or prompt the user to adjust each alert?

4.2 Activity Swim Lane Diagrams


User
Tracker
System
Enter User Credentials
Valid User Credentials?
Yes
No
Retry User Credential Input
Display All Trackers
Select Tracker Feature
Select a Tracker
Select Previous Logs Option
Display Selected Tracker's logs from previous week
Compare previous week's logs to current activity level
Select Back Button
Exit the app

4.3 Use Case Diagram



Section 5. Non-functional Requirements List

Requirement ID	Requirement Title	Short Description	Priority	Requester
NFR01	User Privacy	The app will not need to access any personal information of the user at any given time.	High	End User (Stay-at-home Father)
NFR02	Updates	The app should receive an update once completed and must adhere to modern system requirements.	High	End User (Systems Analyst)
NFR03	Mobile system compatibility	The new system should be compatible with the following mobile operating systems: to run of iOS, Android, WearOS	Medium	End User (There were various smartphones and smartwatches the users were using)
NFR04	Web browser compatibility	The new system should be compatible with the following browsers: Chrome, Firefox, Microsoft Edge, Safari	High	End User (There were various web-browsers the users were using)
NFR05	Language Settings	The application should have different language settings for users who don't understand English.	Medium	End User (Student)
NFR06	Font Size Settings	The apps should provide font size settings for users with low vision.	High	End User (Professor)
NFR07	Color Blind Settings	The apps should provide settings for users with color blindness.	Medium	End User (Programmer)

Appendices

Appendix C. Stakeholder Register

Stakeholder Register					
Stakeholder Name	Stakeholder Position	External /Internal	Stakeholder contact details	Operational /Executive	Interest
David Smith	End User (Programmer)	External	david.smith@gmail.com	Operational	high
Jerome Langston	End User (Professor)	External	jlangston@college.ca	Operational	medium
Larry Guzman	End User (Systems Analyst)	External	larry.guzman@systems.com	Operational	high
Michael Adams	End User (Customer Service Representative)	External	m.adams@gmail.com	Operational	medium
Jennifer Nguyen	End User (Student)	External	jnguyen@college.ca	Operational	high
Mathew Nava	End User (Marketing Officer)	Internal	mathew.nava@com.ca	Operational	high
Kareem Alford	End User (Stay-at-home Father)	External	alfordkareem@gmail.com	Operational	high
Philip James	End User (Office Worker)	External	pjames@officework.com	Operational	low

Appendix D. Interview Questions

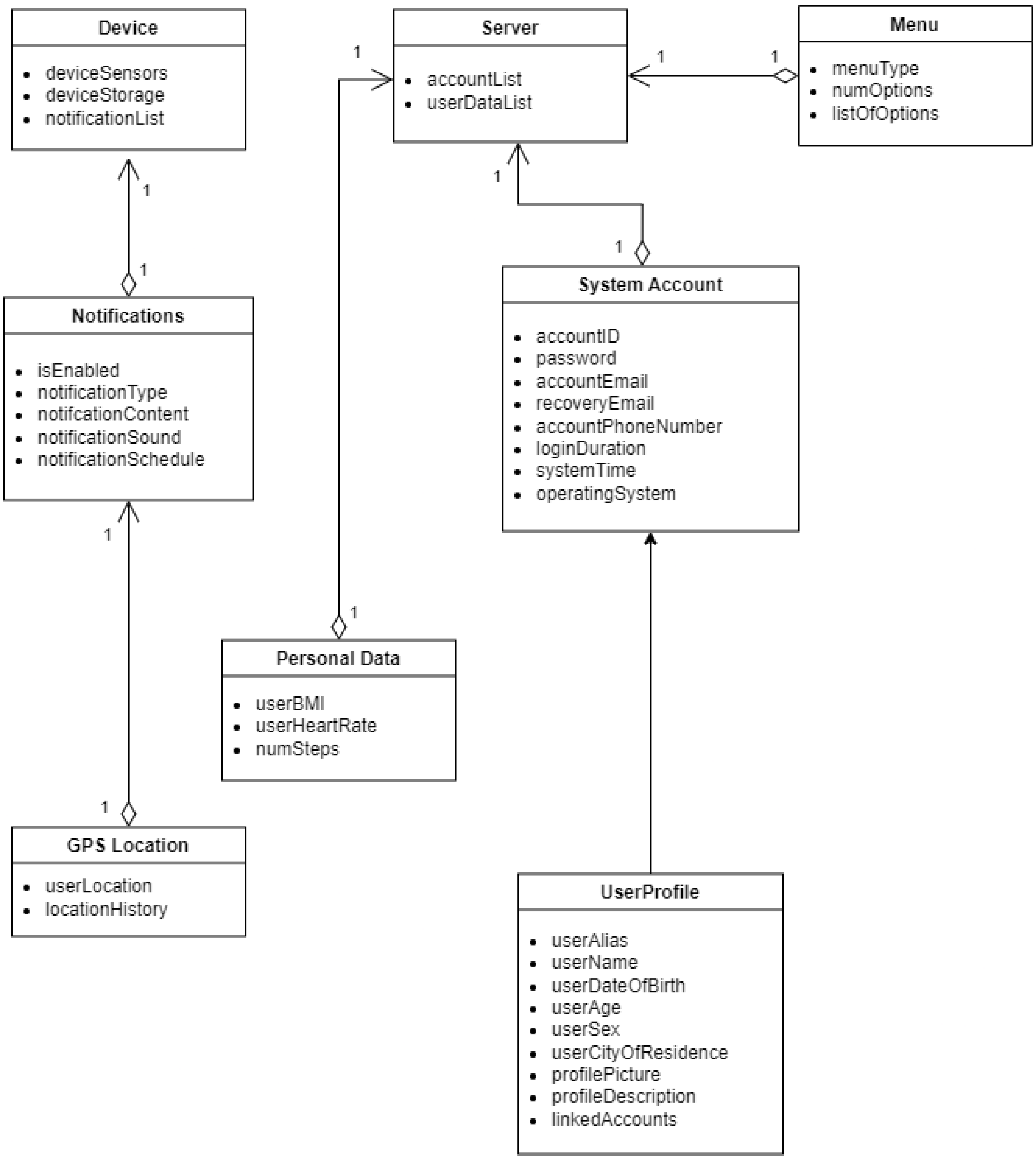
Interview Questions		
Question	Stakeholder position	Answer
1) In terms of marketing the software, what do you think is something that stands out in this software compared to other software that is already available in the market?	End User (Marketing Officer)	“I think reminding the user to hydrate while using digital devices is something new that we can market to the end user.”
2) At what time during the day do you think you’d be using the app most often?	End User (Student)	“Probably most often in the evenings after school.”
3) What specific features would you like to see?	End User (Office Worker)	“The screen-time notifications and the hydration reminders.”
4) Where do you plan on using this app?	End User (Customer Service Representative)	“I think I’d mostly use it when I’m at my local gym.”
5) What features would you like to see and where do you see yourself using them?	End User (Office Worker)	“I’d see myself using the screen-time scheduler mostly while I’m at work.”
6) What type of features would you like to see?	End User (Stay-at-home Father)	“Dark Mode, mealtime alerts, and light exercise suggestions such as stretches and a short walk.”
7) What are the system preferences that will help your application/software attract more users?	End User (Customer Service Representative)	“An easy setting that is straightforward and organized so I can use the application without difficulty.”
8) How would you like to pay for the App services? A one-time payment/a subscription service/or a “free” service with advertisements.”	End User (Stay-at-home Father)	“It would be comfortable with a baseline “free” service with a paid service for extra/additional features”
9) If there was a “premium” version of the App, how much would you be willing to pay for it?	End User (Office Worker)	“Probably $5 per month maximum, though I’d be more inclined to pay for the service if there was a yearly payment plan instead of monthly.”

10) What type of features would you like to see in the software?	End User (Stay-at-home Father)	“I would like to store various pieces of information that I’ve recorded so that I can share it with my family doctor”
11) What kind of features would you like to see?	End User (Student)	“I’d like a way to track my fitness progress and share it with my friends on social media.”
12) How can the app make your life easier?	End User (Student)	“I’d like an easy way to organize and schedule when I need to go to the gym.”
13) How can we make the app better suited for you?	End User (Professor)	“I hope I could enlarge the fonts from the settings menu. The fonts are a bit small for my eyes.”
14) How can the App be made more accessible to you?	End User (Systems Analyst)	“I’d like the App to be easy to navigate, with all the features being easily accessible rather than hidden away in various nested menus. I’d also like the text to be easily readable.”
15) What specific features would you like to see?	End User (Marketing Officer)	“I’d like a way to keep track of my hydration and provide periodic hydration reminders/notifications.”
16) What specific features would you like to see?	End User (Professor)	“I’d like a GPS integration feature that would be able to track my cycling routes and how fast I’ve done them. That way I can observe my training progress.”
17) What specific features would you like to see?	End User (Systems Analyst)	“I would like a feature that I can use to track the amount of time I look at computer screens. I’d also like the ability to set screen-time breaks/notify me if I’m looking at them for too long.”
18) What type of features would you like to see in future iterations of the program?	End User (Professor)	“I would like a feature that notifies me about my physical activity, such as stretching and light exercises in between classes.”
19) What type of features would you like to see in future iterations of the program?	End User (Programmer)	“I’d like a fitness tracker extension, so that I can also monitor my fitness and activity levels.”
20) In what kind of devices would you want to use the software?	End User (Office Worker)	“I would like to use the app mostly in smart watches and in small digital devices.

21) Do you prefer free apps with ads or paid apps without ads?	End User (Customer Service Representative)	“I don’t prefer apps with ads. When provided service is satisfactory, I pay a maximum of $80 USD per year for each app.”
22) Any other improvements you hope for the apps?	End User (Programmer)	“I am color blind. Some pages in your mobile app needs improvement for color blind users.”

Appendix E. Class Diagrams

E.a. Domain Class Diagram (First Cut)


Domain Class Diagram (First Cut)

E.b. CRC Index Cards

Class: Device	
Responsibility:	**Collaborator:**
Knows sensors	
Knows storage	
Knows notificationList	Notification
Sends message	Server
Notifies user	Notification
Creates Notification	Notification
Deletes Notification	Notification
Updates Notification	Notification

Class: Notification	
Responsibility:	**Collaborator:**
Knows isEnabled	
Knows type	
Knows content	
Knows sound	
Knows schedule	
Update	Device
Set Enable	Device
Notify	Device

Class: GPSLocation	
Responsibility:	**Collaborator:**
Knows currentLocation	
Knows locationHistory	
Refreshes	Device

Class: Server	
Responsibility:	**Collaborator:**
Knows accountList	UserAccount
Sends message	Device
Creates account	UserAccount
Deletes account	UserAccount
Updates account	UserAccount
Authenticates account	UserAccount
Records TrackerData	UserAccount
Gets menu options	Menu
Sets menu options	Menu

Class: UserAccount	
Responsibility:	**Collaborator:**
Knows ID	
Knows password	
Knows primaryEmail	
Knows recoveryEmail	
Knows deviceList	
Knows loginDuration	
Knows timezone	
Knows OS	
Authenticates	Server
Updates self	Server
Gets TrackerData	TrackerData
Gets UserProfile	UserProfile

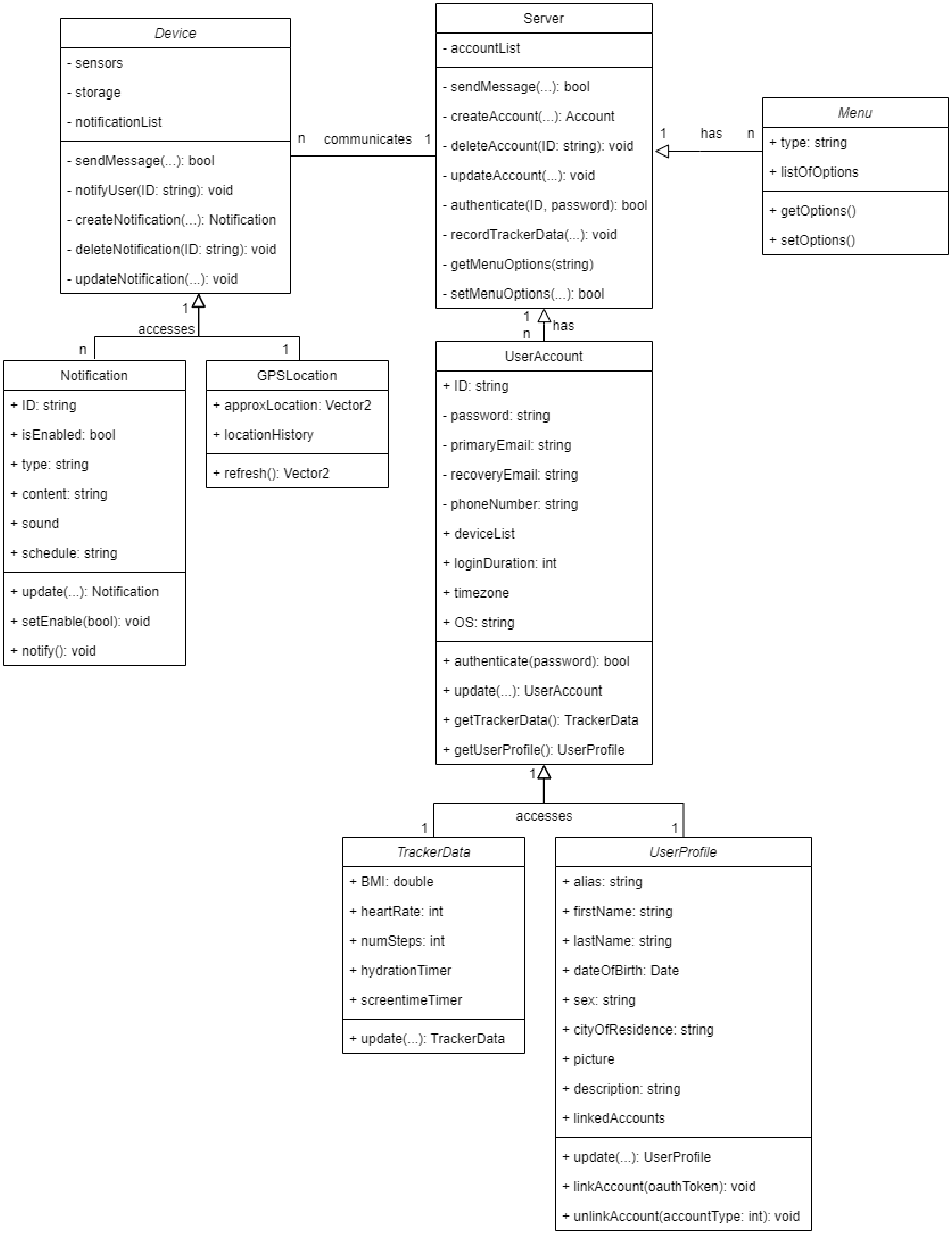
Class: Menu	
Responsibility:	**Collaborator:**

Knows type	
Knows options	
Knows listOfOptions	
Gets options	Server
Sets options	Server

Class: TrackerData	
Responsibility:	**Collaborator:**
Knows BMI	
Knows heartRate	
Knows numSteps	
Knows hydrationTimer	
Knows screentimeTimer	
Updates self	UserAccount

Class: UserProfile	
Responsibility:	**Collaborator:**
Knows alias	
Knows firstName	
Knows lastName	
Knows dateOfBirth	
Knows sex	
Knows cityOfResidence	
Knows picture	
Knows description	
Knows linkedAccounts	
Updates	UserAccount
Links account	UserAccount
Unlinks account	UserAccount

E.c. Class Diagram



Class attribute descriptions and visibilities

Notifications	
Attribute	**Description**
isEnabled	A boolean variable that indicates whether an instance of a notification is active or not. If true, the notification will send prompts to the user as specified by the user. If false, the notification will not send any prompts.
notificationType	A string that indicates the functionality of the notification. Functionality can include whether it sends “push” prompts to the user's lockscreen, whether it texts the user, whether it uses the device sounds etc.
notificationContent	A string that stores that content of the message as specified by the user.
notificationSound	A string that indicates the type of sound the prompt will make.
notificationSchedule	A method that determines at what times the notification will be prompted for the user. Requires access to the system clock on the device
Device	
Attribute	**Description**
deviceSensors	A list variable that stores all the various aspects of the sensors on the device
deviceStorage	A variable that interfaces/accesses the storage on the user device
notificationList	A class list that stores all of the notifications that are currently active
GPS Location	
Attribute	**Description**
approxLocation	A variable that stores the user’s current approximate geolocation derived from the user’s device
locationHistory	A variable that stores previous “userLocation” entries.
Server	
Attribute	**Description**
accountList	A class list that stores all of the instances of “System Account”
Menu	

Attribute	**Description**
menuType	A list that stores all of the various health data trackers that are either taken from the user device or inputted by the user themselves.
listOfOptions	Contains a list of all the different types of data trackers.
User Account	
accountID	A string variable that stores an identifier associated with the user’s account
password	A string variable that stores the encrypted key associated with the user’s password
primaryEmail	A string variable that stores the main email address associated with the account
recoveryEmail	A string variable that stores a second email address.
phoneNumber	A string variable that stores a phone number associated with the account
loginDuration	An int variable that stores the length of time that a user is active in their account in seconds
timezone	User’s local timezone
OS	A variable that stores the operating system that the app is running on.
User Profile	
userAlias	A string variable that stores the alias of a user as inputted by the user
firstName	A string variable that stores the first name of user
lastName	A string variable that stores the last name of user
dateOfBirth	A string variable that stores the birth date of the user as inputted by the user
sex	A string variable that stores the user sex as inputted by the user
cityOfResidence	A string variable that stores the city that the user is residing in as inputted by the user.
picture	A variable that stores a picture inputted by the user
description	A variable that stores a fixed-length string as inputted by the user
linkedAccounts	A list variable that stores a series of associated social media accounts

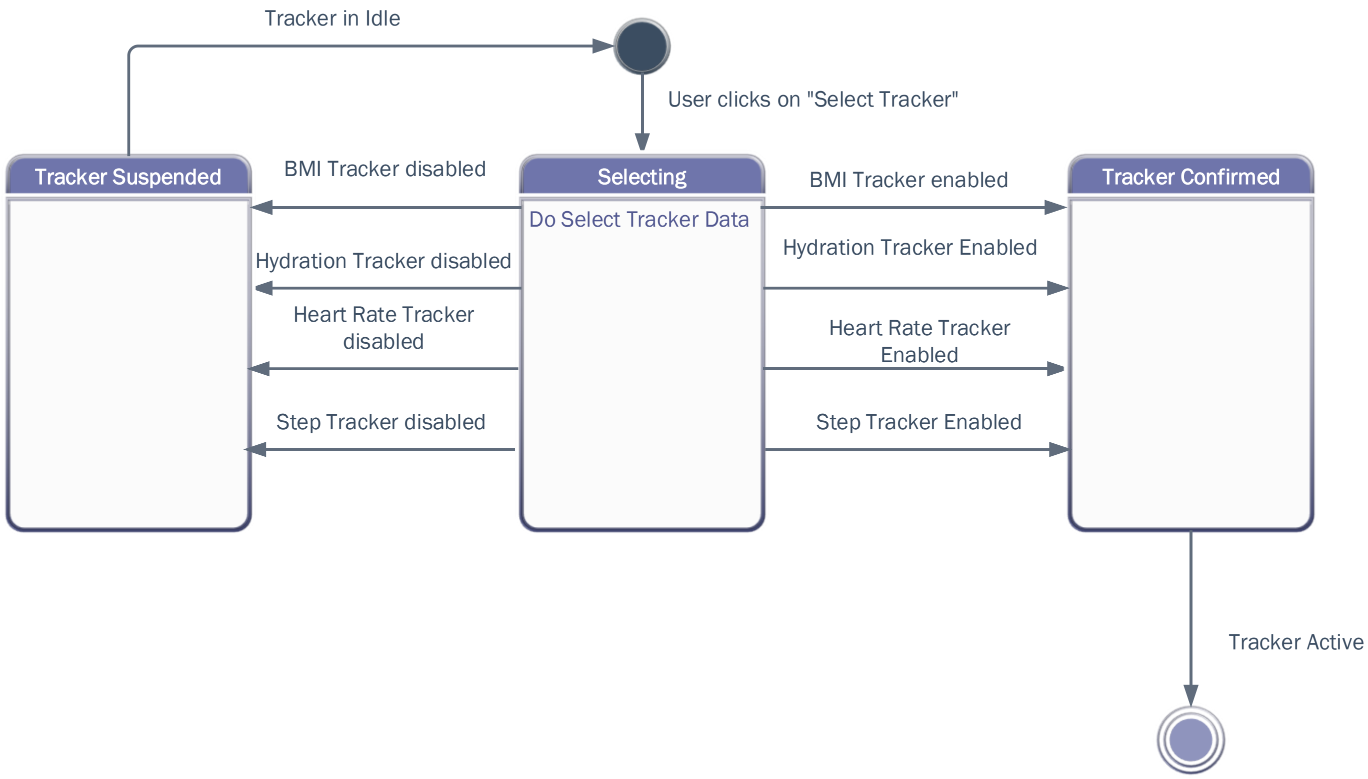
Methods with the respective parameters and visibilities

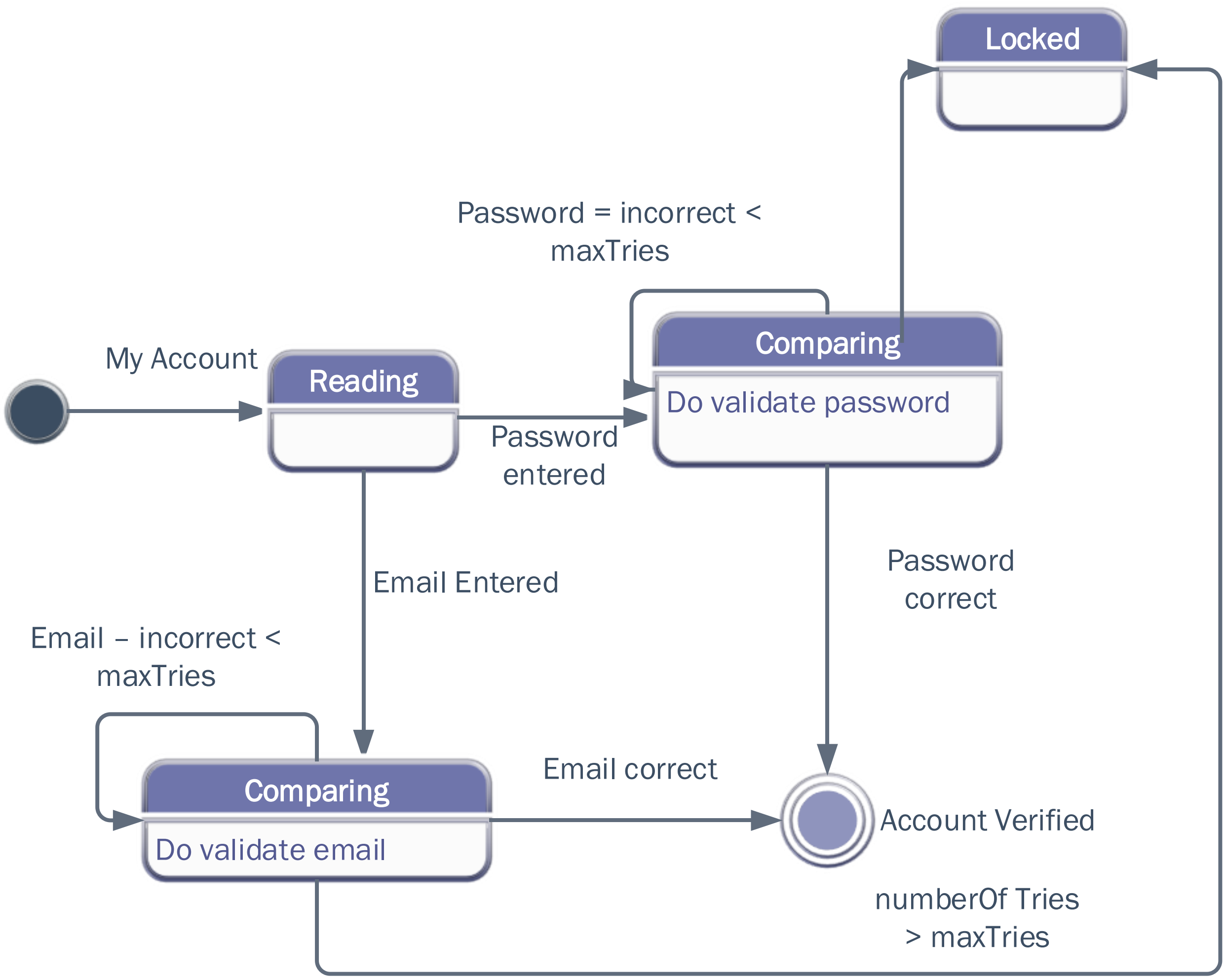
Method	**Parameters**	**Visibility**
Device		
sendMessage()	type: string params: string[]	private
notifyUser()	id: string	private
createNotification()	type: string content: string sound schedule: string	private
deleteNotification()	id: string	private
updateNotification()	id: string type: string content: string sound schedule: string	private
Notification		
update()	content: string sound schedule: string	public
setEnable()	value: bool	public
notify()		public
GPSLocation		
refresh()		public
Server		
sendMessage()	type: string params: string[]	private
createAccount()	id: string password: string primaryEmail: string recoveryEmail: string phoneNumber: string timezone	private
deleteAccount()	id: string	private
updateAccount()	id: string	private

	password: string primaryEmail: string recoveryEmail: string phoneNumber: string timezone os: string	
authenticate()	id: string password: string	private
recordTrackerData()	bmi: double heartRate: int numSteps: int hyrationTimer screentimeTimer	private
getMenuOptions	type: string	private
setMenuOptions	type: string params: string[]	private
UserAccount		
authentificate()	password: string	public
update()	password: string primaryEmail: string recoveryEmail: string phoneNumber: string timezone os: string	public
getTrackerData()		public
getUserProfile()		public
TrackerData		
update()	bmi: double heartRate: int numSteps: int hyrationTimer screentimeTimer	public
UserProfile		
update()	alias: string firstName: string lastName: string dateOfBirth: string sex: string cityOfResidence: string	public

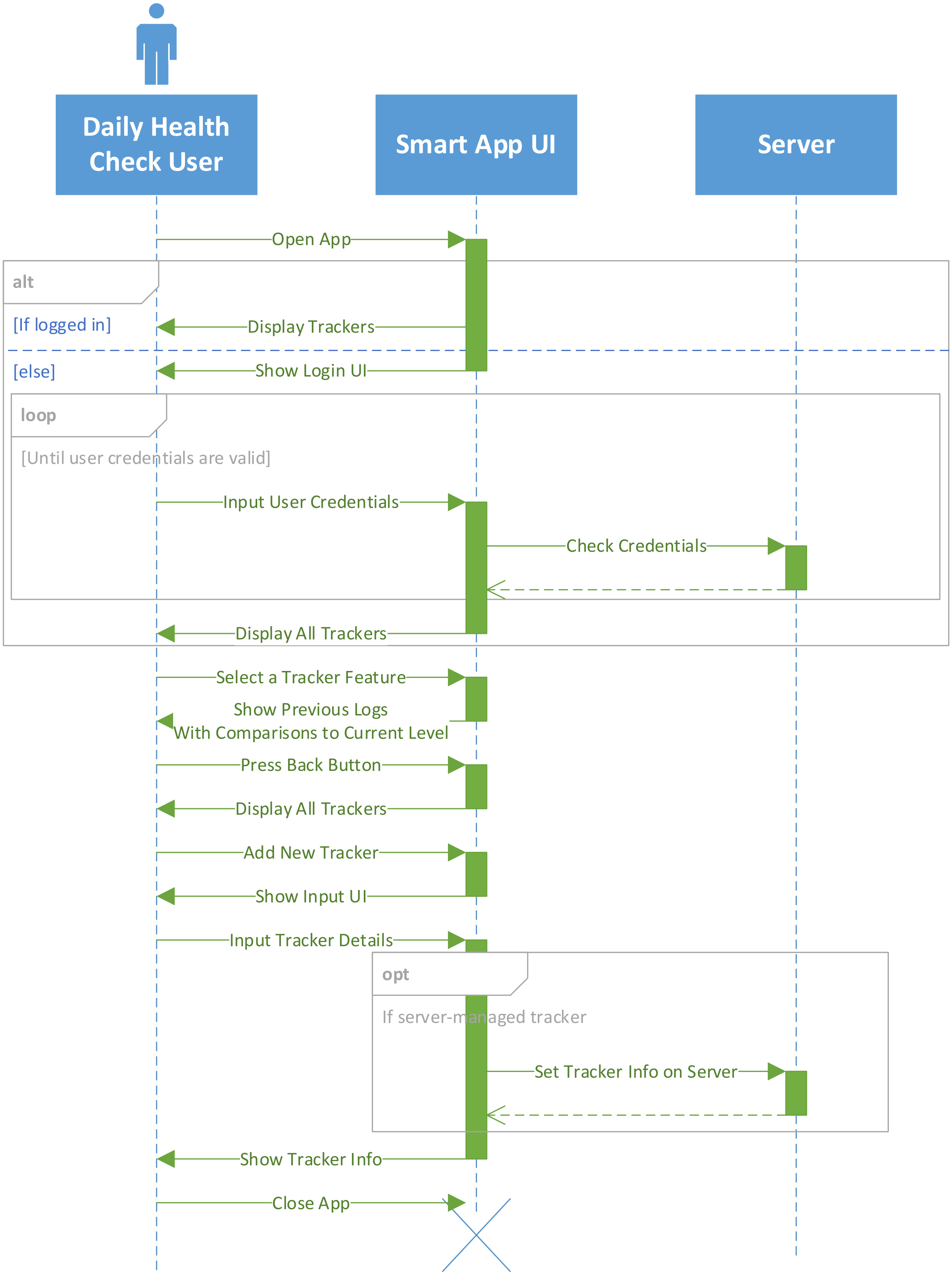
	picture description: string	
linkAccount()	oauthToken: string	public
unlinkAccount()	accountType: int	public
Menu		
getOptions()		public
setOptions()	params: string[]	public

E.d State Diagrams




Locked
Password = incorrect < maxTries
My Account
Reading
Comparing
Do validate password
Password entered
Email Entered
Password correct
Email – incorrect < maxTries
Comparing
Do validate email
Email correct
Account Verified
numberOf Tries > maxTries

E.e Sequence Diagram



E.f Party Analysis Pattern

We choose not to apply the Party Analysis Pattern for our project even though it can offer value because a party or supertype is not required. We designed this app to be used by individual users. Thus, the system does not provide features for a group of users as a party.

We also considered setting a supertype for our classes, like how the 'Party' class substituted the 'Person' and 'Organization' Classes in the example. However, our class designs are already concise at this stage, so adding an abstract pattern might make the system harder to understand.