

# **AUCares**

## **AU Healthcare Mobile Application**

Anthony Baron, Isabella Sims, Reika Kitano, Tami Yousafi, Nathan Kennedy, and Elisabetta  
Gabriele

# **1. Abstract**

## **1.1. Problem**

During this time of the pandemic and COVID-19, many people have concerns regarding their health and well-being. Many people also seek resources and knowledge regarding their health and how to maintain their well-being. Students, in particular, may not have much knowledge of their health and the resources available to them. Many students might want to pay more attention to their health, but they do not know how to start or have the motivation to do so.

Health is a broad subject. There are many different sides to one's health that it becomes difficult to observe it all. Even if one does know what they need to do and what they are looking for, taking action to improve their health may seem like tedious tasks with no clear benefit in the near future.

## **1.2. Project Definition**

We wanted to give students a means of paying attention and keeping track of their health while learning more about their well-being and raising their awareness towards their health.

We have developed a mobile application catered towards health and wellness. This student wellness app will be tailored to the AU student body. Our motivation and rationale behind this application are to provide education and assistance, better the lives of students, facilitate long-term growth and development, and learn about students at AU. Several goals of the application include supporting positive behavior change while collecting data, share personalized educational advice, and give the user insight into groups of people connected to AU.

This application also connects the user to exercise, support, and social opportunities. This application caters to the issue of motivation as it pertains to keeping track of one's progress and areas of improvement by providing daily truths and dares for the user to complete. A user's progress is visually represented through wheels and pie charts, which the user could find more satisfying and more clearly understand their progress.

## **1.3. Purpose of Report**

In this report, we will walk through the thought process of the application, including where we began, the tools and techniques we used, analysis and review of the current state of the project, and descriptions of all the features and their importance. We will also discuss where the application currently stands in terms of completion and where it has yet to go.

## **1.4. The Aftermath**

We knew from the beginning that this mobile application will require multiple semesters to complete. The application has yet to have several features and user interface components but provides a solid foundation for future implementations and improvements. Several features that have yet to be implemented are the progress wheel, several components on the "Settings" page, user personalization, and the appearance of daily truths and dares.

## 2. Introduction

This pandemic has given professionals and potential users alike the opportunity to reflect on both mental and physical well-being as well as what that means to them. Though the issue of mental health, in particular, has been especially relevant during such times of distress, we are also reminded of the fact that it is a standing problem - one that will not dissipate with the concession of the pandemic.

With this, we have teamed up with health professionals here at AU to develop an app to address these problems. We, Professor Owrang, Professor Yamamoto, and Professor Ulrich see this as a time to capitalize on what matters - and with mental and physical health being cornerstones to all other facets of life, we are thrilled to be a part of something with the ability to improve both.

The areas of health and well-being we are focusing on are the following:

- Sleep
- Physical Activity
- Occupational Health
- Fruit and Vegetable Consumption
- Social Health
- Emotional Health
- Hydration

The app we are developing will help to educate students on mental and physical health, but more importantly, is designed to be an avenue for improvement. With specific challenges poised as “truths” or “dares”, we feel we can help students reach their goals in a way that is familiar to them. Students will also be able to keep track of their tasks in an organized matter through a to-do list type of feature. Being able to see a visualization of their progress will also help encourage students to continue to improve. Students should also be able to learn and understand new practices that will help them lead healthier lives.

The application is later expected to have Organization, which will allow groups to be created and for people to compete amongst each other for fun. Other than developing healthier habits as a group competition can also encourage students to do more tasks and continue to improve.

First, we will discuss existing applications that pertain to health-related activities and how they compare to our application. Then, we will move on to the requirements of the application and its features, going into detail about expectations and functionalities. Next, we will evaluate the application in terms of its sustainability, feasibility, and other qualities. Finally, we will discuss what we have completed so far and where this application has yet to go in the future.

### **3. Review**

There are several similar existing applications. One example is Apple Health, which is an iOS application that keeps track of a user's sleep, steps, walking and running distance, and headphone audio levels. It shows data from the current day, past week, and past year. The application shows the data through bar graphs, line graphs, bubble plots, and scatter plots.

Apple Health provides basic numerical data of tasks that the user completes. It is also a platform for a user to view their health records and suggestions for improvement based on their data. For example, if the application reads that the user's audio volume is too loud to the point where it will damage the user's hearing, then the application recommends that the user turn down the volume and enable headphone notifications for when the audio goes too high.

While the application offers suggestions, it does not offer daily random tasks for the user to complete. Rather it forces the user to analyze their data and set their own goals. For example, the user may want to increase their number of daily steps because they feel that they should get more.

Google Fit is another commonly-used application. It is an Android application that allows users to manage their workouts, such as offering heart rate sensors to record their speed, pace, route, and calories burnt. It also helps the user track their sleep duration and bedtime patterns. There is a journal feature that allows users to keep track of the user's progress, such as daily and weekly goals. There are heart points and steps goals.

Much of the data is displayed using radial bar charts, which encourage users to complete the circles. This application only tracks a user's physical activity and sleep, not their mental health, food and water consumption, or other areas of their well-being.

Another health-related mobile application is Fabulous: Self Care, which is available on both Android and iOS. This application mainly focuses on helping users keep track of their self-care activities, such as water consumption, exercise, productivity, and relaxation time. There is a scheduling feature to help users plan their day and remind them of what they need to do to ensure that they are practicing healthy habits. The application suggests goals for the user.

This application requires the user to already know healthy practices and what they need to improve on. However, a user may not know some things or they may not understand the importance of certain healthy practices. It's important to educate and encourage users as they work to improve themselves.

## **4. Design Requirements and Details of Project**

### **4.1 User Requirements**

- Functional app for both major platforms
- User spinnable task wheel
- A way for the user to track their tasks

- User profile
- Login and sign up

The customer required a functional app written for both iOS and Android systems. The user has to be able to spin a wheel on the home screen to get a task (truth, dare, or question). From there the task would be saved to the user profile where they can view their current and completed tasks in a challenge screen. From the challenge screen, the user has the option to mark a task complete. The user also has access to a list of resources in their profile for different situations. Another important requirement is user-profiles and the ability to log in or sign up for new accounts. To save the task a user received from spinning the wheel it needed to be tied to an account on the backend.

Due to our project being a mobile application, there is no important central function, instead, each screen within the application is split into its container with its functions based on the requirements for that screen. The most important of these functions would be the ability for a user to log in or sign up. The creation of user accounts allows us to store and track data prevalent to that user efficiently and quickly. An example of this would be the user's tasks gained from spinning the wheel on the home screen and stored in a list for them to view in the challenge screen.

Performance-wise, our goal was to create a smooth, lag-free user experience while a user is using our application. This means all data fetch calls need to be small and specified for the exact amount of information needed to complete the task. It also means loading only the relevant data about the screen the user is currently on and not for a screen that the user is not looking at. Our goals have been met so far with the functionality currently implemented in the application, no current fetch calls take a long time that the user would notice in the form of lag in the user interface and no other screen is doing any processing than the one the user is currently using.

Some of the user interface requirements for the login and sign-up pages are that the user must be able to retrieve their password if they forgot it, as well as user authentication using the AU email address. The following images are the current prototypes for the user interface of the application.

(Figma file:

<https://www.figma.com/file/xfPLEaXnY6BM49gxgEMBya/AUCares-Prototype?node-id=0%3A1>

(Main color scheme: <https://coolors.co/d7263d-f46036-2e294e-1b998b-c5d86d>)

Color scheme for the seven areas of health:

- Social health - Pink - #ffd5e5
- Emotional health - Yellow - #ffffdd
- Fruit and Vegetable consumption - Green - #a0ffe6
- Hydration - Blue - #81f5ff
- Occupational health - Orange - #f9d89c
- Sleep - Purple - #c3aed6
- Physical activity - Maroon - #cd5d7d

## **Description of the app:**

### **About**

AUCares is a platform to help American University students to stay on top of their health and keep track of their well-being. We support positive behavior change, share educational tidbits appropriate for your personal development, give insight to resources at AU, and connect you to exercise and social opportunities.

### **Target Areas**

AUCares strives to assist you in:

- Sleep
- Physical Activity
- Occupational Health
- Fruit and Vegetable Consumption
- Social Health
- Emotional Health
- Hydration

### **Our Goal**

Our goal is to provide education and assistance, better the lives of the AU student body, and facilitate long-term growth and development.

### **How to Use**

Upon creating an account using your AU email, you will answer a series of initial questions so we can get an idea of your current state of well-being.

Through a daily log-in, you can spin a wheel to receive a dare that pertains to your current progress. You will also answer a question about your previous dare to determine how much you've improved.

You will be able to manage your goals and challenges in an organized manner. We will keep track of your overall progress through the Wheel of Wellness and provide you with advice and information that will help you to continue to advance in your areas of improvement.

### **Initial Questionnaire: (optional)**

1. Birthday
  - a. Note - There should be an automatic setting where anyone who indicates that they are <18 years of age at the time of application is automatically removed from data collection.
  - b. Note - Possibly we can consider putting in a birthday pop-up for everyone, but also include a pop-up asking individuals turning 18 after they begin using the application, if they would like to opt in for data collection.
2. Gender
  - a. Male (cisgender)
  - b. Female (cisgender)

- c. Transgender Female
  - d. Transgender Male
  - e. Gender-fluid, non-binary
  - f. Prefer to self-describe: \_\_\_\_\_
  - g. Prefer not to say
3. Which category (or categories) best describes you? Please select all that apply.
- a. White
  - b. Hispanic, Latino or Spanish origin
  - c. Black or African American
  - d. Asian
  - e. American Indian or Alaska Native
  - f. Middle Eastern or North African
  - g. Native Hawaiian or Other Pacific Islander
  - h. Other (Please specify): \_\_\_\_\_
  - i. Prefer not to say
4. What is your classification?
- a. Freshman
  - b. Sophomore
  - c. Junior
  - d. Senior
  - e. Master's student
5. On average, how many hours do you sleep at night?
6. How many times a day do you eat a handful size serving of fruits? (Note: two handfuls would count as two times)
7. How many times a day do you eat a handful size serving of vegetables? (Note: two handfuls would count as two times)
8. On average, how many 8oz glasses of water do you consume per day? (Note: an average size water bottle has 16oz of water and would therefore count as two 8oz glasses)
9. How many hours/minutes a week would you say that you exercise at an intensity where it becomes slightly difficult to carry on a conversation?
10. Others?

## **Resources**

- AU Health Center
  - [Link](#)
  - [shc@american.edu](mailto:shc@american.edu)
  - 202-885-3380
- AU Health and Wellness
  - [Link](#)
  - [dos@american.edu](mailto:dos@american.edu)
  - 202-885-3300
- Health Promotion and Advocacy Center
  - [Link](#)
  - [hpac@american.edu](mailto:hpac@american.edu)
  - 202-885-3276

- Counseling Center
  - [Link](#)
  - 202-885-3500
- Step Up
  - [Link](#)
- Center for Diversity and Inclusion
  - [Link](#)
  - [cdi@american.edu](mailto:cdi@american.edu)
  - 202-885-3651
- Kay Spiritual Life Center
  - [Link](#)
  - [kslc@american.edu](mailto:kslc@american.edu)
  - 202-885-3320
- OASIS: Office of Advocacy Services for Interpersonal and Sexual Violence
  - [Link](#)
  - 202-885-7070
  - [OASIS@american.edu](mailto:OASIS@american.edu)
- Public Safety
  - [Link](#)
  - [dps@american.edu](mailto:dps@american.edu)
  - Non-Emergency: 202.885.2527
  - Emergency: 202-885-3636
- Student Conduct & Conflict Resolution Services
  - [Link](#)
  - [conduct@american.edu](mailto:conduct@american.edu)
  - 202-885-3328
- Student Activities
  - [Link](#)
  - 202-885-3390
- Office of Campus Life
  - [Link](#)
  - [campuslife@american.edu](mailto:campuslife@american.edu)
  - 202-885-3310
- Women's Health Services
  - [Link](#)
  - [shc@american.edu](mailto:shc@american.edu)
  - 202-885-3380
- COVID-19 Resources
  - [Link](#)
- Health Equity
  - [Link](#)
  - [cdi@american.edu](mailto:cdi@american.edu)
  - 202-885-3651
- Psychiatric Services
  - [Link](#)
  - [shc@american.edu](mailto:shc@american.edu)

- 202-885-3380
- Health Insurance
  - [Link](#)
  - [shc@american.edu](mailto:shc@american.edu)
  - 202-885-3380
- Immunization
  - [Link](#)
  - [shc@american.edu](mailto:shc@american.edu)
  - 202-885-3380
- AhealthyU
  - [Link](#)
  - [ahealthyu@american.edu](mailto:ahealthyu@american.edu)
  - 202-885-3742
- Academic Support and Access Center
  - [Link](#)
  - [asac@american.edu](mailto:asac@american.edu)
  - 202.885.3360

The image displays two wireframe prototypes for a log-in/sign-up interface. Both screens feature a large circular placeholder for a profile picture at the top center. Below the placeholder, the word "Welcome!" is centered in bold black font.

**Login Screen:** On the left, there are two input fields: "Email" with an envelope icon and "Password" with a lock icon. Below these fields is a red "Submit" button. To the right of the "Submit" button, the text "Don't have an account?" is displayed above a red "Sign-up" button. At the bottom, the text "Forgot password?" is followed by a red "Send Email" button. The bottom of the screen is divided into two green buttons: "Login" on the left and "Sign-up" on the right.

**Signup Screen:** On the right, there are two input fields: "Email" with an envelope icon and "Password" with a lock icon. Below these fields is a red "Submit" button. To the right of the "Submit" button, the text "Already have an account?" is displayed above a red "Login" button. The bottom of the screen is divided into two green buttons: "Login" on the left and "Sign-up" on the right.

Figure 1.0 Log-in/Sign-up Prototype

When the user first signs up, they are met with a description of the app which is easily navigable, and an initial questionnaire to get a sense of their current health status and their current strengths and areas of improvement.

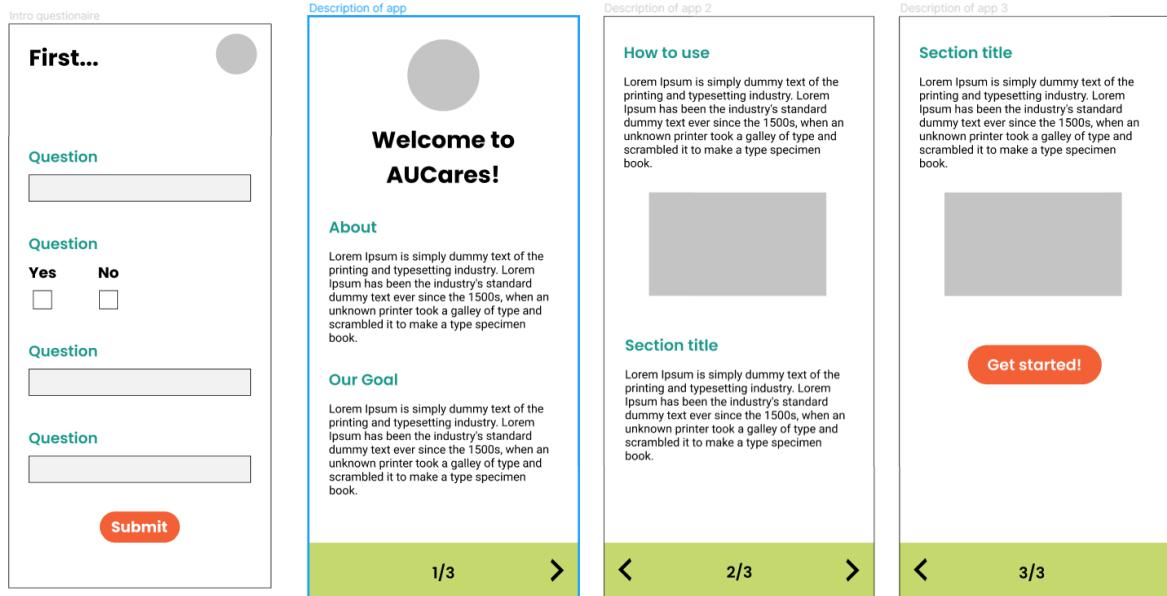


Figure 2.0 Description of the App and Initial Questionnaire Prototypes

Then, the user is met with the main application screens. The home screen consists of the Wheel of Wellness, which the user can spin and receive a new daily challenge. They will also receive a daily check-in of how their current challenge is going.

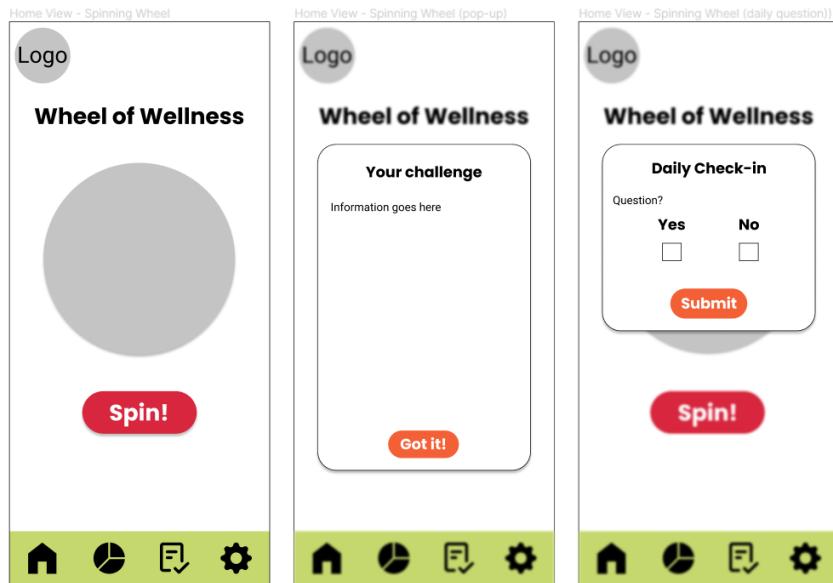


Figure 3.0 Home Screen Prototypes

The user can more specifically track their progress using the progress wheel, which displays their biggest strength areas as a larger pie slice. The user would also be able to see information about each section.

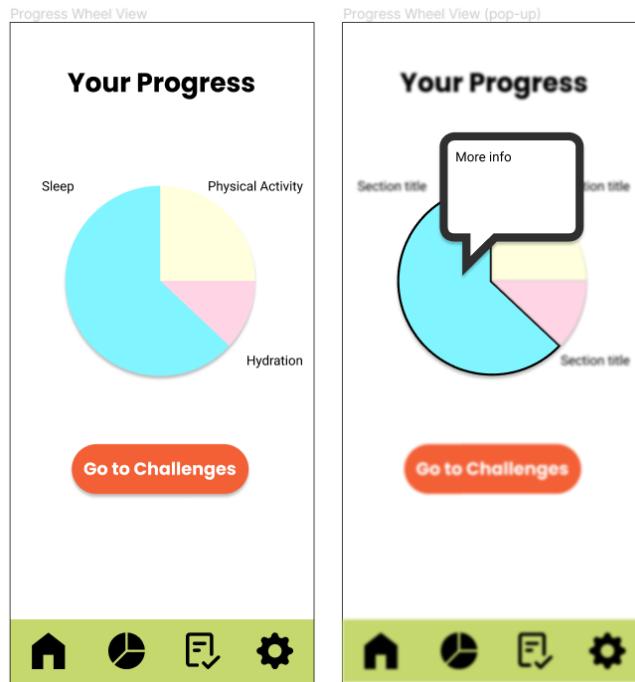


Figure 4.0 Progress Wheel Prototype

The user will be able to see their current and completed goals and objectives on the challenges page, which is a simple list that they can modify.

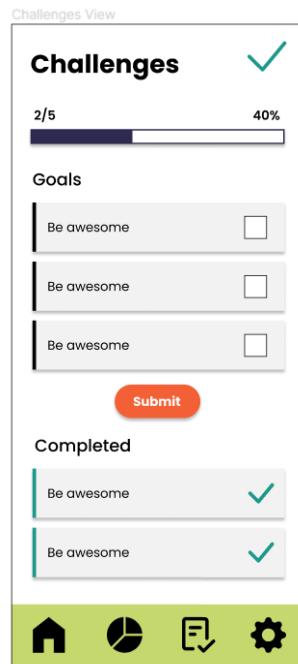
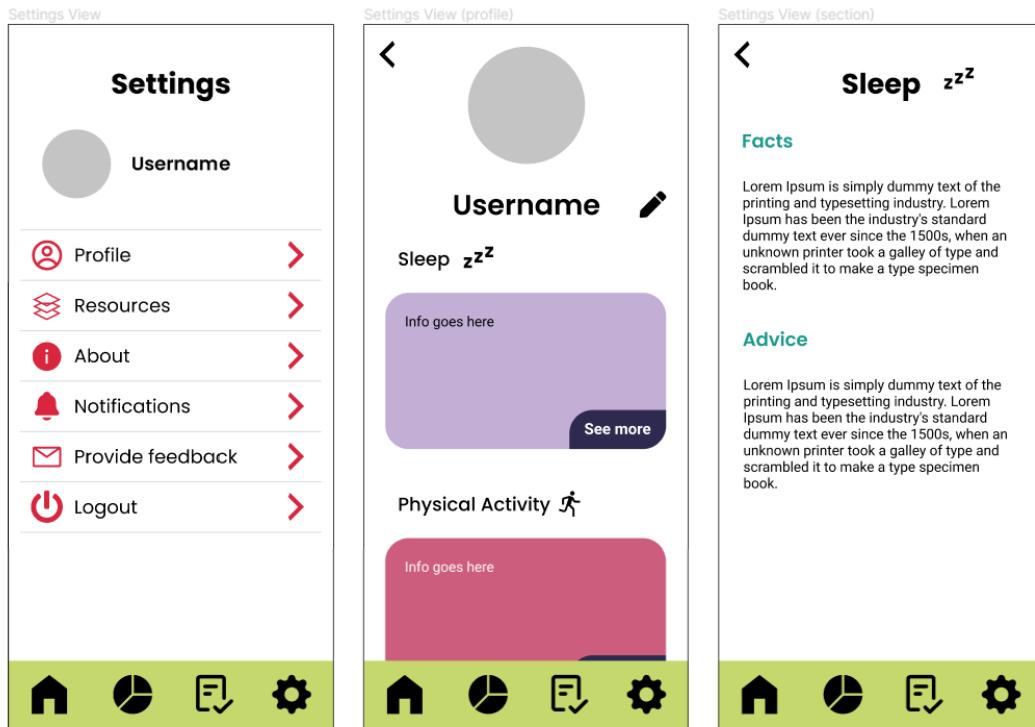


Figure 5.0 Challenges Prototype

The setting page holds the user's personal information, such as their account information and statements regarding their progress, as well as resources available to them on and off-campus, facts about each of the seven areas of health that the application targets, general information about the application, and a form that they can fill out to provide their feedback on the application.



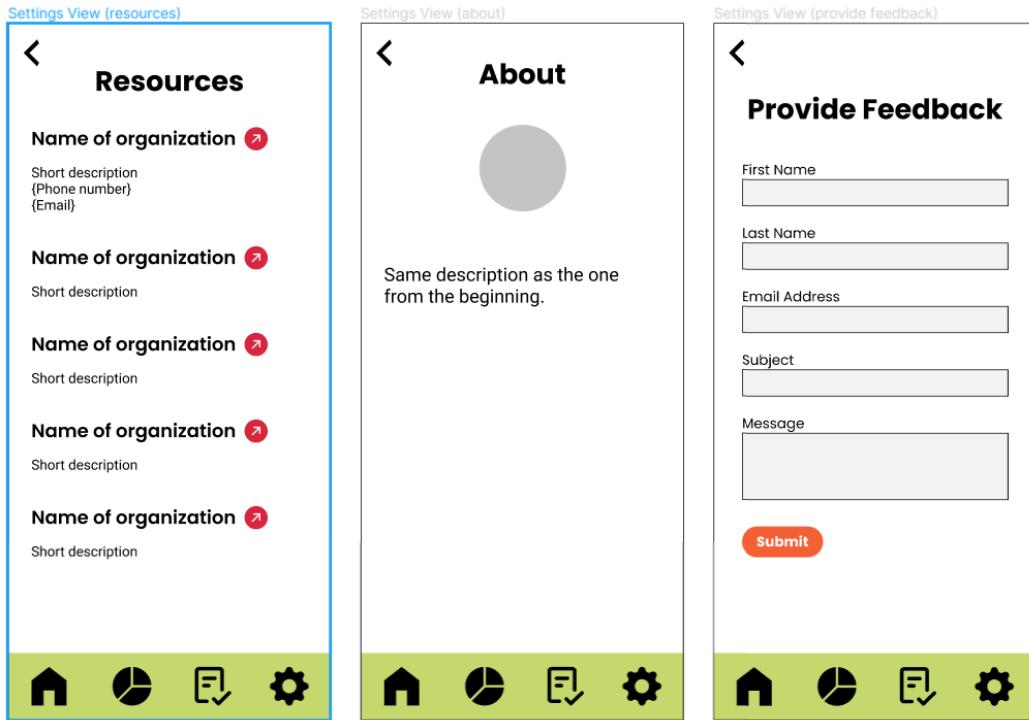


Figure 6.0 Other Main Pages Prototypes

## 4.1. Feasibility Discussion

Upon creating the application and starting the implementation of it, there are many factors we need to consider beforehand, including costs, what to ensure to complete and not complete by the end of the project, what should be included in the application, tools, time to learn new technologies, and how the application overall should be implemented.

When it comes to the use of the application, our goal is to promote students to practice healthier habits in the short and long term. With this application, we hope that students throughout the American University community will promote healthy habits while also bringing some fun and possibly competition in the near future.

When it comes to the economics of creating the application, we did not have much of a budget, other than a stipend given to our clients which can be used to maintain the application. Due to this, we had to ensure that the technologies we used were cheap or free. Luckily for us, the technologies we are using to implement our application are both completely free to use and open source, React-Native on the front end, Express with Node.Js on the backend, and MariaDB as the database. Even though we did not have any costs in building the application, our clients will need to pay for a cloud service, such as Amazon Web Services (AWS) or Google Cloud, upon the application going into production. Prices could vary based on which service is used, but for AWS the cost of storing data in the database is \$0.10 per GB month with \$0.20 per 1 million requests,

and a backend server of ~\$30 a month with a free AWS tier. Both of these costs will increase as the number of users increase and more data is stored. Deploying the mobile application the Apple Store and Google Play store will also cost money. Apple requires paying \$99 per year to have an application on the Apple store while Google Play will be a one-time \$35 fee. These are the expected costs currently, but these could also increase if the website is later decided to be used along with the application as well as have a professional programmer to look over the project since this project was implemented by seniors in college. Also, maintainability could be a cost depending on who maintains it.

When it comes to sustainability, as this is an application, as long as the code is maintained and updated, this application will go for a long time and reach more users. The application itself and the many directions it can go, however, can go a very long way. When it comes to the application itself, currently we are only looking at challenges such as truth, dares, and questions, but new games can easily be added to the scheme, such as races, activity bingo, etc. Streaks can also be added, and for completing streaks some rewards could be given, such as a double challenge, more spins, or even a coupon. When it comes to health categories, currently there are eight in our system, but more could be added if needed. Another way this application could grow is by going into other topics. Currently, this application only focuses on health, but there can be many other aspects in which people can gradually improve or maintain, such as financial spending, education, career growth, etc. Lastly, this application is currently only used by American University, however, if successful it can grow to other schools as well and even workplaces. To get to this point, there would have to be many advancements, but it is also an aspect to look at.

When it comes to access and environment of the application, this is a mobile application, and due to this, almost anyone with a smartphone will be able to download the application. The application does not use serious technologies or require a lot of storage and phone energy, so the newest phone will not be needed for its use. However, as the application does get updated and the codebase gets maintained, old operating systems could become obsolete. Due to this, people with very low phones might see a lower performance when using the application. It is also possible that in the future this application could also be used via a browser, which if this were to occur, even people without a good smart phone or no smartphone will be able to use the application.

When it comes to risk and technical risk, this project is practically risk-free. The only things that could cause an issue when it comes with dares is having a dare that is best not to complete given someone's personal circumstances, such as for example daring someone to run when they have a broken bone, or asking a dare which itself has any minuscule danger to it (ex: simply walking). Outside of this, the project will be using simple tasks which have already been used before and are not using anything new.

When it comes to ethicality, it is very ethical in that it is trying to promote healthy habits and encourage people to complete good things. The only caveat that can come to truths and dares when it comes to the ethical side is that users are not checked for whether their responses actually correspond to their doing. Due to this, the application would reward a user for lying in the application if the user does not use the application to their own benefit. Also, even though the data stored from users is anonymous, it is still viewable by those who have access to the

database. Even though this should not be an issue, it is hope this data is not used for any specific advantages, but only for the benefit of seeing how students are progressing.

## 5. Final Implementation

### 5.1. Navigation

AUCares is an application built on top of a navigation container. The first instance where a user will interact with this navigation container is after they log in and are presented with an onboarding experience consisting of a few pages explaining the purpose of the application and ending with a questionnaire. At the very end of this start-up sequence the user will be directed to a login page where, after successfully logging in, the user will be redirected to a home screen. Figure 7.0 shows the home screens navigation container consisting of a bottom tab bar. This bottom navigation bar is used by the user to switch between the four main screens in the application.

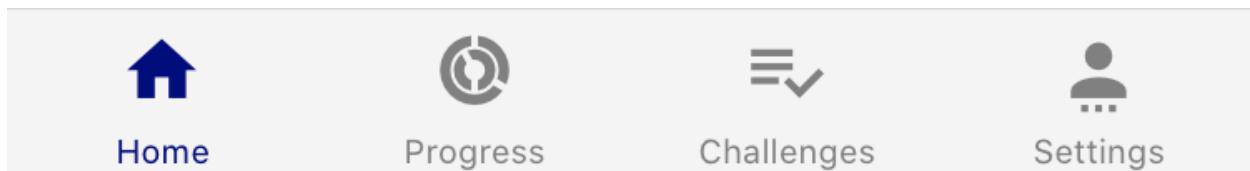


Figure 7.0 Bottom Navigation Bar

These nested navigation containers are important because one handles the user's experience as they enter the app while the second handles the user's experience while they are in the app. Figure 8.0 shows the main file of this app and how it is just a navigation stack with the different screens the user will click through before being allowed to login and land at the home screen. Towards the bottom of the stack, you will notice a screen with the name 'HomeApp' and component 'HomeApp'. This line is what controls the navigation to the home screen, and in the home screen is the code for the bottom navigation bar shown in figure 9.0.

```
//App
export default function App() {
  return [
    <Provider store={store}>
      <NavigationContainer>
        <Stack.Navigator screenOptions={{ header: ()=>null }}>
          <Stack.Screen name='Onboarding_1' component={Onboarding_1}/>
          <Stack.Screen name='Onboarding_2' component={Onboarding_2}/>
          <Stack.Screen name='Questions' component={Questionnaire}/>
          <Stack.Screen name='Mood' component={Mood}/>
          <Stack.Screen name='Login' component={preLogin} />
          <Stack.Screen name='HomeApp' component={HomeApp}/>

        </Stack.Navigator>
      </NavigationContainer>
    </Provider>
  ];
}
```

Figure 8.0 App.js Navigation Container

```
<Tab.Screen name="Home" component={HomeScreen} />
<Tab.Screen name="Progress" component={ProgressScreen} />
<Tab.Screen name="Challenges" component={ChallengeScreen} />
<Tab.Screen name="Settings" component={Settings} />
</Tab.Navigator>
</NavigationContainer>
```

Figure 9.0 Home Screen Navigation Container

Finally, there is a third nested navigation stack responsible for controlling the settings view. When the user clicks on the settings icon in the bottom navigation bar seen in Figure 7.0 they are directed to a screen, shown in figure 10.0, that shows different settings that the user can then click on:

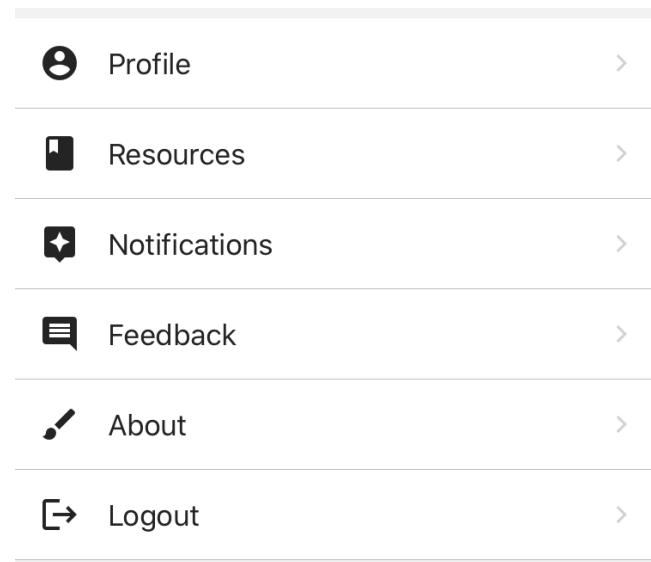


Figure 10.0 Settings Navigation Stack

Each option is shown in figure 10.0 links to a different screen within settings with its purpose. The profile shows the user profile and displays cards that show information about all eight different categories with the app (sleep, physical activity, etc etc), Resources links to a page that displays a list of recommended resources in the area, notifications are responsible for turning application notifications on or off, feedback is a form that sends user feedback to the database, about rehashes the application info shared during the onboarding experience, and logout is a button for logging the user out of the application.

## 5.2 Application Screens

Another way this application is split is by ‘screens’. A screen refers to what is currently being shown to the user. Usually, this means what page of the application they are on. The main part of the application consists of four screens, Home, Progress, Challenges, and Settings. The

home screen displays a central component of the application, which is the spinning wheel. The sequence of interactions on this page leads to a series of pages layered on top of the home page that allows a user to dive deeper into their selected challenge and task. The login and signup screens allow users to create new accounts and stay logged in so that their information is remembered securely. In addition, the progress screen highlights the user's current progress in the form of a circular graphic, whose bars are filled based on the level of completion of tasks they have been assigned. The challenges screen displays the user's current challenges, and the settings contain a myriad of options that will allow the users to complete items like changing account details, changing app preference settings, sending feedback about the app, etc. There is only one instance of a screen having multiple screens hosted within it in AUCares and it is the settings screen. As mentioned above the setting screen essentially is a third navigation stack where each option in the stack is connected to a screen with that setting option. For the most part, all other screens consist of just one screen and some custom components. For organizational purposes, each screen is given its file and exported so it can be called from any other file when needed.

AUCares consists of several screen-specific functions used to enhance the user's experience. An example of this would be the fetch calls for the challenge screen and resource page within settings. Both these functions fetch data from the backend and process it in the best way for the user.

### 5.3 Database

This mobile application uses a Maria DB system with SQL code, a MySQL administration tool called phpMyAdmin, and a web server platform called XAMPP to operate the database. Using these methods, the application stores different tables and functions to save user and frontend data. First, the SQL code is written for each table along with any relations to other tables and restrictions, and then imported into phpMyAdmin (Figure 11). The application administrators/developers can browse various information by clicking on different tables on the left-hand side of the screen. For example, if an administrator clicks on the "Truths" table, a list of truths that are given to the user appears with detailed information like its "TruthID" and the number of points of each truth (Figure 12).

Screenshot of phpMyAdmin showing the database structure for 'au\_cares\_db'. The left sidebar shows tables like ActivityCompletedTypes, ApplicationFeedback, CategoryLevels, etc. The main area shows a table of all tables with columns: Table, Action, Rows, Type, Collation, Size, Overhead. A search bar at the top says 'Containing the word:'.

Table	Action	Rows	Type	Collation	Size	Overhead
ActivityCompletedTypes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	3	InnoDB	utf8mb4_general_ci	32.0 Kib	-
ApplicationFeedback	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	16.0 Kib	-
CategoryLevels	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
CategoryPoints	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
CategoryPointsHistory	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
CategoryTypes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	7	InnoDB	utf8mb4_general_ci	48.0 Kib	-
CurrentLeaders	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
Dares	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_general_ci	48.0 Kib	-
DaresHistory	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	80.0 Kib	-
DaresResponses	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
GenderTypes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	6	InnoDB	utf8mb4_general_ci	48.0 Kib	-
HistoryQuestionnaireResponses	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	48.0 Kib	-
Organization	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
OrgUserInfo	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
QuestionResponseAnswer	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
QuestionResponseChoices	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
Questions	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_general_ci	64.0 Kib	-
QuestionsHistory	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	80.0 Kib	-
QuestionsResponses	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_general_ci	32.0 Kib	-
QuestionTypes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
RaceTypes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	7	InnoDB	utf8mb4_general_ci	48.0 Kib	-
ResourcePage	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	20	InnoDB	utf8mb4_general_ci	16.0 Kib	-
TruthsHistory	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_general_ci	48.0 Kib	-
TruthsResponses						
Console	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_general_ci	48.0 Kib	-

Figure 11.0 phpMyAdmin screenshot of the database

Screenshot of phpMyAdmin showing the 'Truths' table. The left sidebar shows tables like Category, CategoryPointsHistory, CurrentLeaders, Dares, etc. The main area shows the 'Truths' table with columns: TruthId, Description, Points,CategoryId, MinPointsNeeded, HoursToComplete, SentNum, CompleteNum. Two rows are listed:

TruthId	Description	Points	CategoryId	MinPointsNeeded	HoursToComplete	SentNum	CompleteNum
4	another test	2	1	4	5.0	0	0
5	say hello	2	1	3	4.0	0	0

Figure 12.0 The “Truths” table in phpMyAdmin

As much as tables are used to store the user’s data, a big goal of SQL Databases is to create relationships between data and have as least redundancy as possible. We ensure this by planning out how each table will look by following the steps of Normalization. Below is a map of our resulting database. As much as it is hard to read, it shows relationships between tables, allowing easier and more efficient search queries.

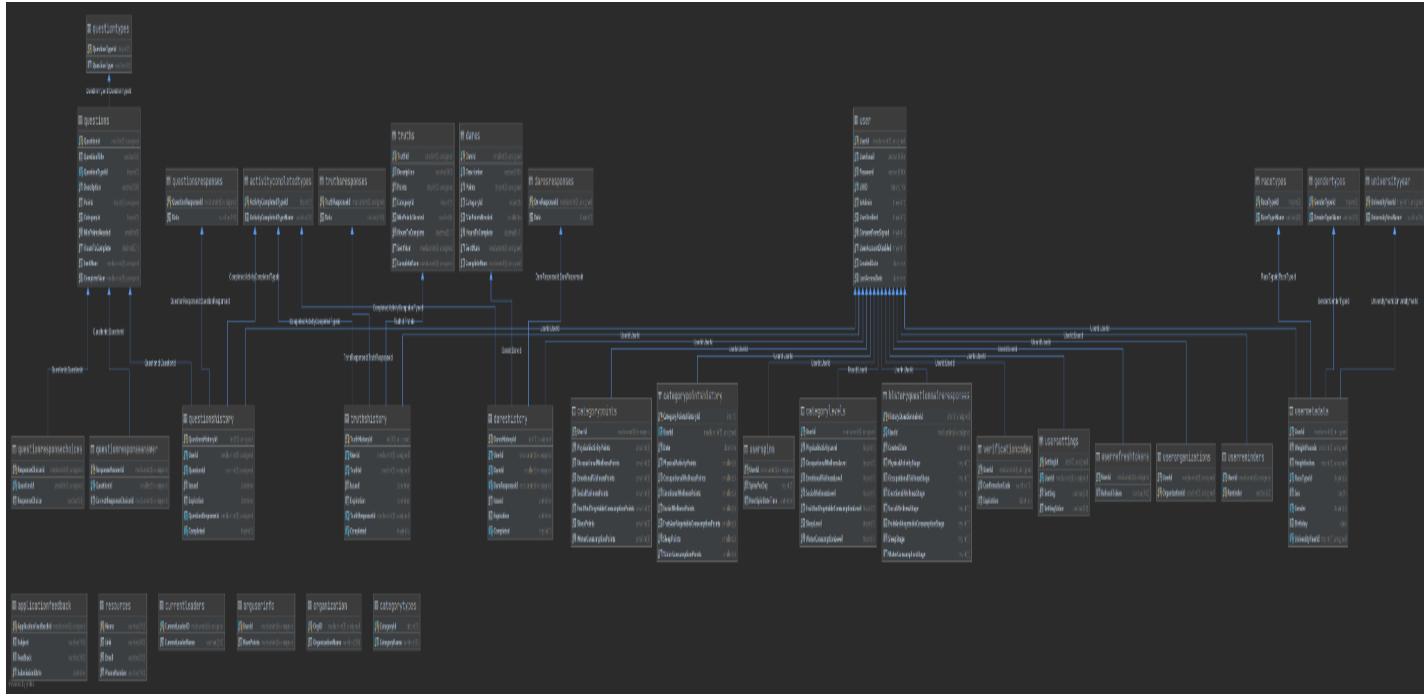


Figure 13.0 Database Tables and their relationship

## 5.3 Backend

The backend of this application uses Node.JS as its runtime environment, and JavaScript is used as the programming language used in Node JS. We are using Express as our backend framework to handle user requests to the server and respond to them. The main requests of our application are logging a user end and tracking users' progress, along with giving new truths, dares, and questions upon spinning the wheel daily. The code and the back end are organized based on a structure similar to the MVC model, with each service being in its layer. Data is sent between the layers, however, only the service function layer has access to the database to ensure consistency. Each layer also has its functions divided into separate files based on the tasks they provide. For example, all API routes that have to do with authentication are in their file, and API requests that have to do with accessing resources are in another file. Figure 14 is the map of the layers showing communication between them and where information travels.

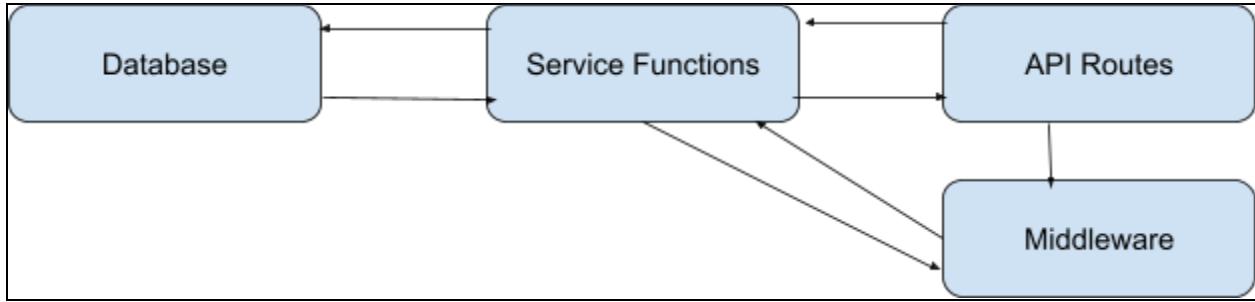


Figure 14.0 Diagram of where the information travels

Data security is done by using JSON Web Tokens (JWTs) as the authentication and authorization method used in the application. Personal data, such as emails, are hashed to ensure data is anonymous. The way that a JWT works is that upon a user logging in and being authorized, they will receive an access token. An access token is an encoded JSON that holds information to ensure that a user is valid as well as information about the user and is required upon accessing secured request endpoints. Upon each request, the access token is verified to authorize the user, and then it is decoded to determine the user, and then respond with the user's wanted data. For security reasons, the tokens only have a lifespan of around 30 minutes. To allow a user to receive a new access token as well as stay logged in even after closing the application, a refresh token is sent to the database, which once verified will send a new access token, allowing the user to access the database again. Below is a diagram showing the logic of JWT

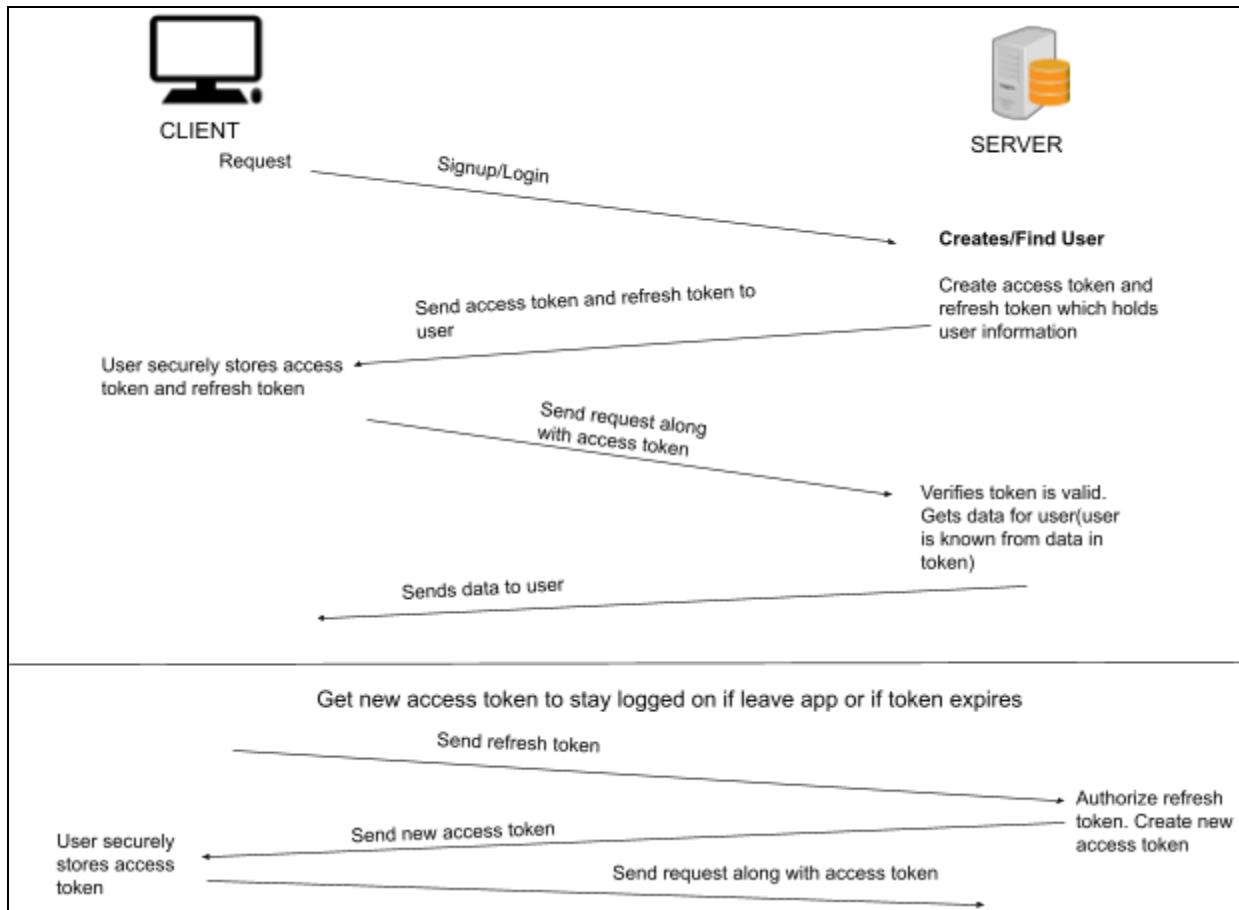


Figure 15.0 JWT diagram

Personal data is hashed to ensure data security as well as for the user to feel safe to store data in the database, knowing that it can not relate to them in any way.

The backend is tested by testing endpoints and creating/holding fake data in the database. Upon each endpoint being created, we test the functionality by running various queries and testing whether we get our expected results. The application which we use to test our application endpoints is called Insomnia, which allows us to send requests to mock requests to the database and see the responses in a good user interface. Below is an example of the login request.

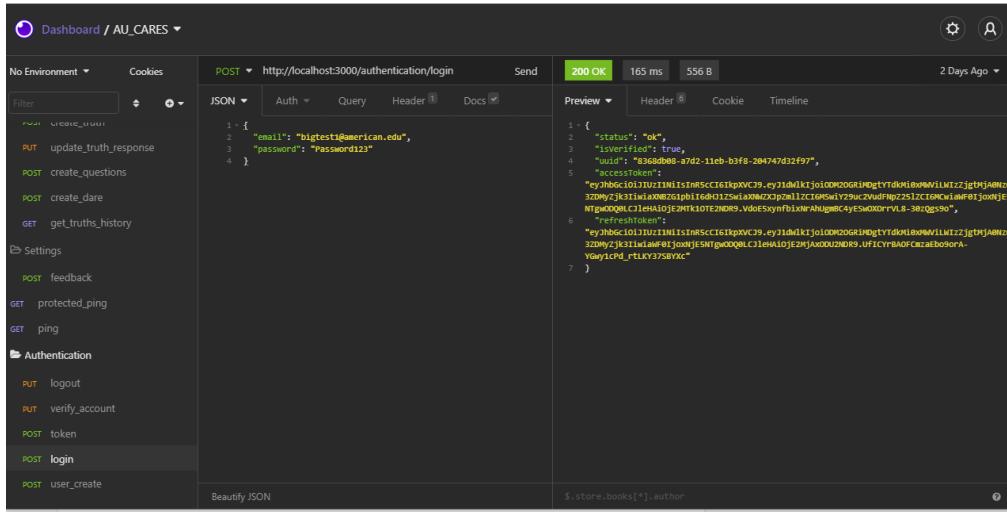


Figure 16.0 Insomnia demonstration

We do not have any big calculations in our backend.

## 6. Results

We estimated to have a functional app for both major platforms, a user-spinnable task wheel, a way for the user to track their tasks, a user profile, and a login and sign up page. We were able to implement all of these features successfully in our final application.

When the user opens the mobile application, a welcome page appears with a brief explanation of the application and the main goal (Figure 18). After the user clicks “continue,” instructions on how to use the application and the target audience appear on the screen (Figure 19). The last introductory screen comes up with a questionnaire that will help tailor the application to the user based on his or her answer (Figure 20). The questionnaire consists of six graphics, each composed of six original and curated designs requested by the AU Health professors. They represent different health components: water, fruits and vegetable intake, social activity, exercise, and career standing. Within these wheels, each graphic represents a stage of personal development: pre-contemplation, contemplation, preparation, action, maintenance, and relapse. We have also provided an alternative design of five phases which excludes relapse for future devs to keep on hand when they decide which implementation is better after some user testing.



Figure 17.0

Each of these files is directly editable, and we have provided everything a future developer needs to make adjustments: every individual graphic as an AI file as well as the majority of them in a combo file, a shell of the circle shapes along with a tutorial of how to create clipping masks to get images to fit in them, and multiple versions of each character for ease of making changes. Additionally, with the help of a bit of code, we were able to plan for future devs who may prefer to edit directly using code rather than using Adobe applications.

After the survey graphics, the user is taken to the login and signup page. Upon signing up, the user's credentials including the username and password are stored in the database. The password is encrypted with a hash. When signing up, the user gets access and refresh tokens as a security measure and gets verified through the backend. Eventually, the user is taken to the main screen in the application (Figure 23), where they are allowed to spin a wheel. The wheel may land on one of seven categories — Physical Activity, Occupational Wellness, Emotional Wellness, Social Wellness, Fruit & Vegetable Consumption, Sleep, and Water Consumption. Upon landing on a category, the user is given information on the meaning of the category and is asked if they would like to proceed or spin again (Figure 24). If they choose to proceed, they will be prompted to select a truth, dare, or question as a 'challenge' within their selected category. They can choose to accept the challenge, which will add the task to the user's task list, marking it as currently uncompleted, or they may opt to switch the type of challenge (e.g. truth or question) and proceed. The truths, for example, are connected to the database, and they are randomly assigned and stored for the user.

While the application is being used, the database is either storing new data or keeping track of ones already stored. For instance, once a user creates an account, that user data is saved in the database and is constantly referenced while the user interacts with the application. Meanwhile, the data that is initially stored, as the list of truths, is also used when the user chooses a truth challenge.

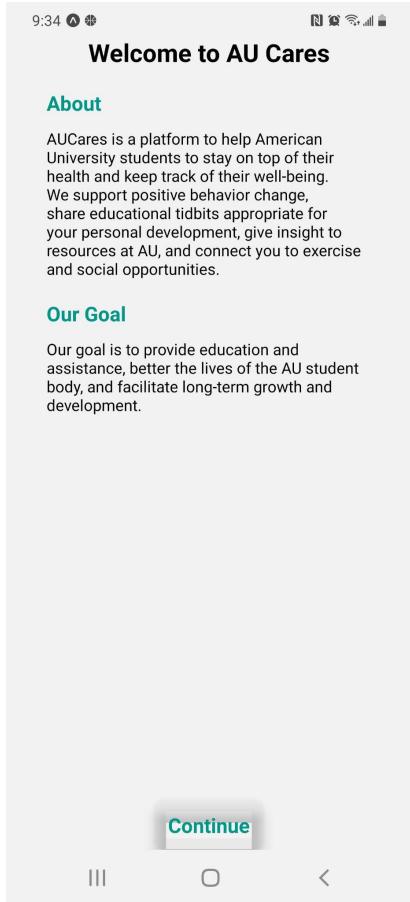


Figure 18.0 Welcome Page

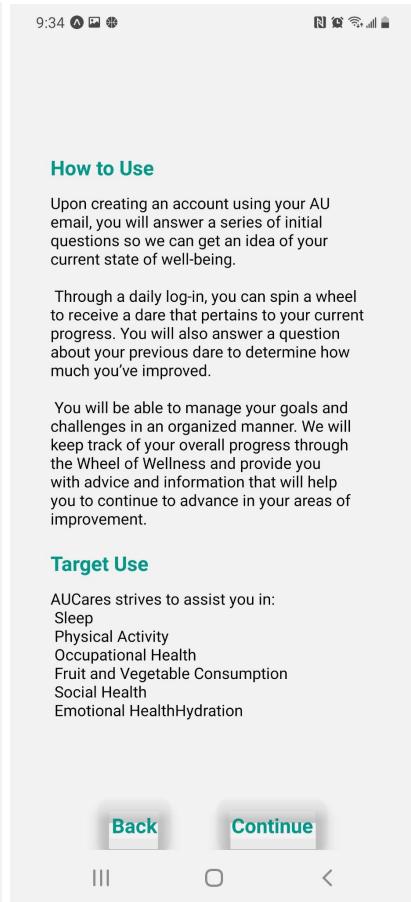


Figure 19.0 Second Introductory Screen

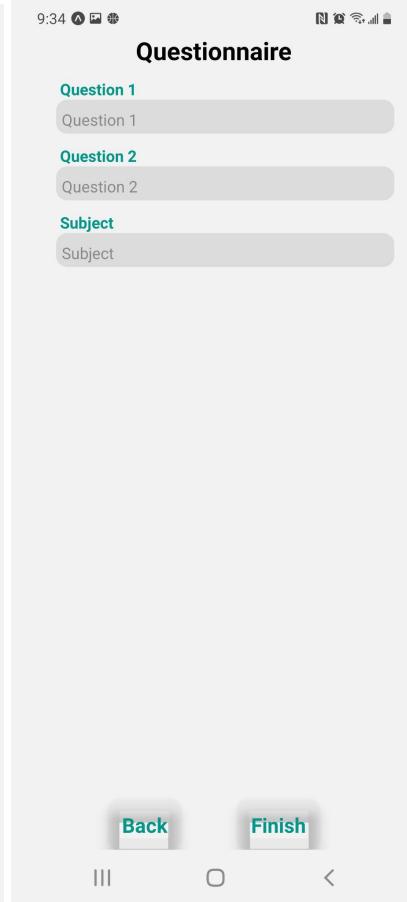


Figure 20.0 Questionnaire

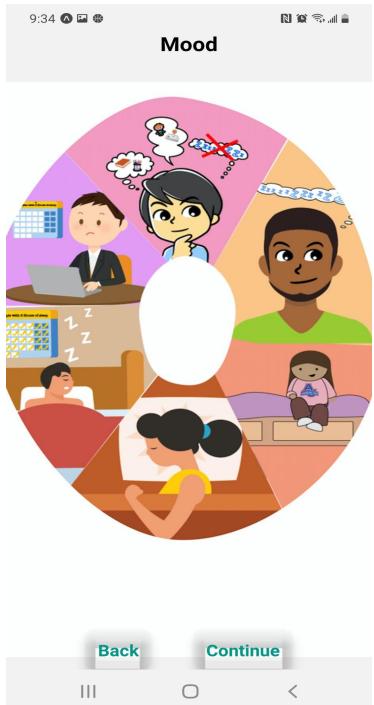


Figure 21.0 Mood Page

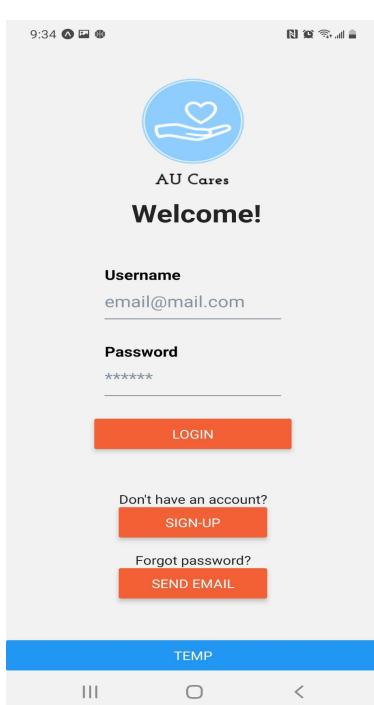


Figure 22.0 Sign-in Page

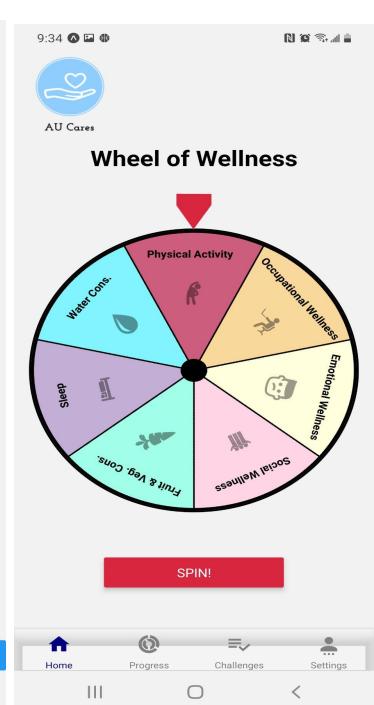


Figure 23.0 Wheel of Wellness



Figure 24.0 Challenge Pop Up



Figure 25.0 Truth, Dare, Question Selection

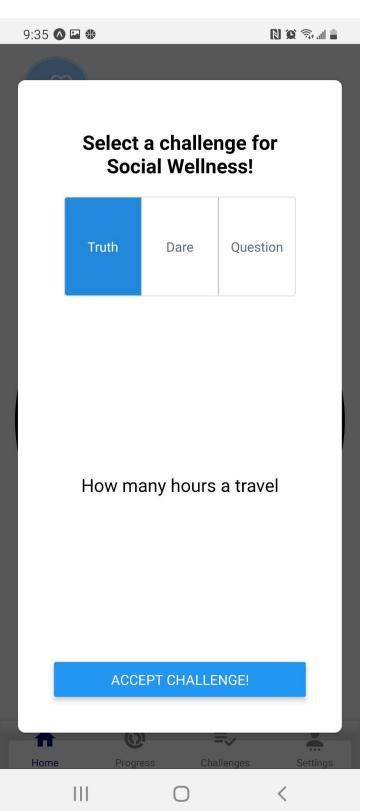


Figure 26.0 Getting a truth

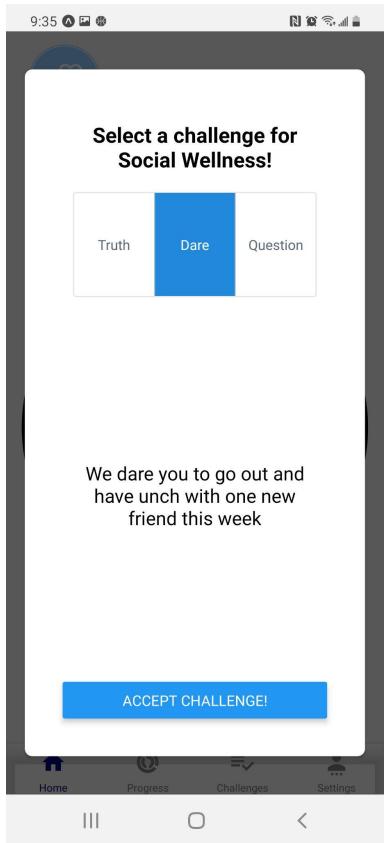


Figure 27.0 Getting a dare

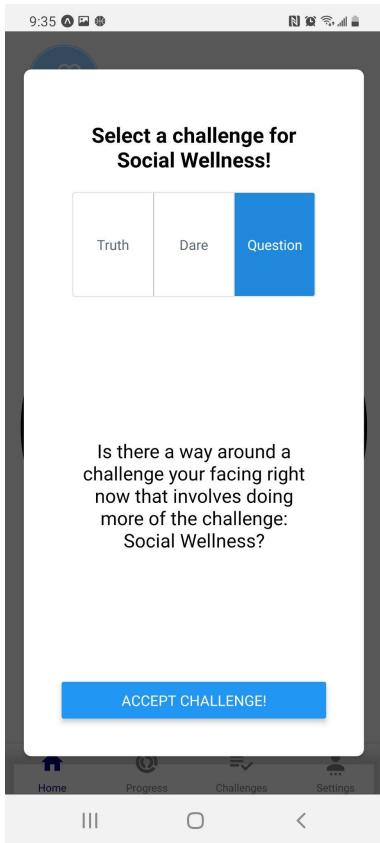


Figure 28.0 Getting a question

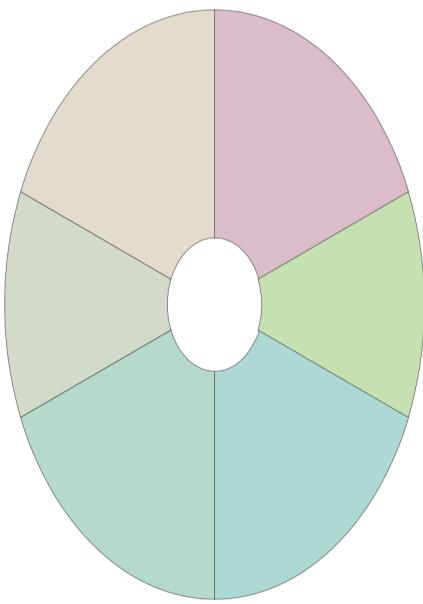


Figure 29.0 wheel shell

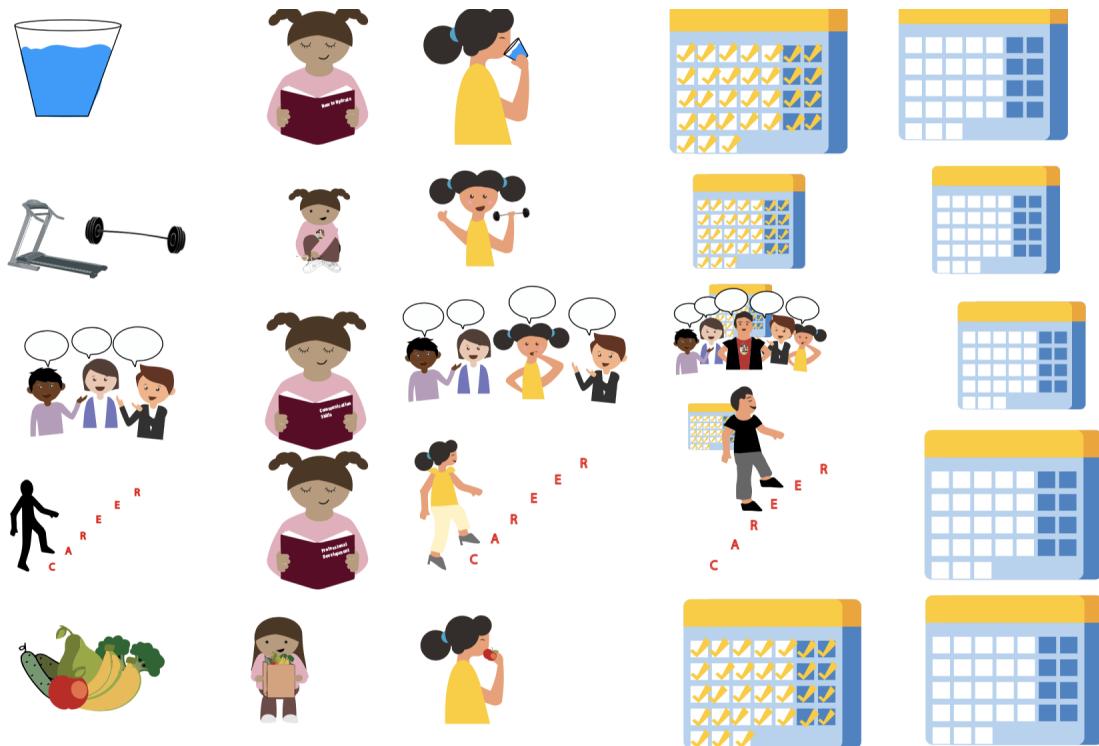


Figure 30.0 One of our graphics AI files for future devs

```
<svg id="Layer_1" data-name="Layer 1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 2972.15
2204"><defs><style>.cls-1,.cls-131,.cls-132,.cls-139,.cls-14,.cls-145,.cls-146,.cls-148,.cls-23,.cls-39,.cls-57,.cls-82,.cls-84,.cls-87,.cls-92,.cls-97{fill:none;}.cls-2{fill:#d0db
c9;}.cls-19,.cls-2,.cls-27,.cls-40,.cls-58,.cls-9{stroke:#231f20;}.cls-19,.cls-2,.cls-27,.cls-40,.cls-58,.cls-82,.cls-87,.cls-9{stroke-
linecap:round;}.cls-14,.cls-146,.cls-148,.cls-19,.cls-2,.cls-23,.cls-27,.cls-36,.cls-37,.cls-38,.cls-39,.cls-40,.cls-57,.cls-58,.cls-9{stroke-miterlimit:10;}.cls-3{clip-
path:url(#clip-
```

Figure 31.0 Snipped of Responsive code for graphics

## 7. Conclusions

The rising concerns of people's health and well-being during the COVID-19 pandemic have inspired the six of us to take action. As people strive for resources and knowledge to better protect themselves and keep safe, we wanted to provide the best and easy platform for everyone to reflect on their health, especially within the AU community.

While there are other health applications available, we found that many only display graphs for users to analyze and do not offer daily random tasks for them to interact with. That is why we were inspired to create one that allows users to educate and encourage themselves to improve their health.

The application performs smoothly without any errors, loads relevant data about the screen the user is currently on and does not have any lag since the fetch data does not take a long time. The application works how it was initially designed; it operates on two major platforms, spins a task wheel, tracks user tasks and profile, and has a login and sign-up page.

One improvement that could have been made in the planning phase is to gather information from the clients earlier so we could have started developing faster. However, with the timeframe we were given, we made major developments to this mobile application. We hope that the next group continues to expand this project and eventually release it to the AU community for everyone to use.

Everyone on this project learned how to work together, manage time, and understand through discussions and explanations. We collaborated, either as the frontend/backend group or one whole group altogether, on complex problems and tasks, delegated roles, and held each other accountable. We also were able to effectively balance other classes, work on this project, and attend client meetings in an online environment. And lastly, we shared diverse perspectives as we all had different backgrounds and experiences in mobile app development. Overall, we are grateful that we were able to experience these essential skills during this project since they are crucial in the workforce.

## 8. References

- Ulrich, T. U., & Yamamoto, J. Y. (2021). *AU Wellness App Concepts* [Slides]. Google Docs.  
<https://docs.google.com/presentation/d/1qSY8a28e3DCvADSjqHAD2gvz8ATp4HcpKIwgpVqvdrU/edit#slide=id.p1><sup>1</sup>