

SAM Fitness: An Android Wellness Application

A Major Qualifying Project Report:

submitted to the faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

degree of Bachelor of Science

by

Haley Andrews

Nathan Ford

John Haas

Evan Safford

Date: March 27, 2015

Approved: _____

Keywords:

1. Software Engineering
2. Android
3. Wellness

Professor Emmanuel Agu, Major Advisor
Professor Elke A. Rundensteiner, Co-Advisor

Abstract

The emerging popularity of so-called “Wellness Apps” (mobile applications designed to assist users in pursuing a healthy lifestyle by encouraging them to make positive lifestyle decisions) has presented an interesting challenge to mobile application developers. Our application incorporates step and sleep tracking algorithms. In addition, the application tracks the user’s mood throughout the day, and, using this data, the user can monitor the correlation between his or her exercise, sleep habits, and overall mood. In this project a Wellness App for the Android platform, SAM Fitness, is developed and tested to track these factors.

Table of Contents

- Abstract 1
- Table of Contents 2
- List of Figures 4
- 1 | Introduction 5
- 2 | Background and Related work..... 6
 - 2.1 | Effect of Sleep and Exercise on Mood..... 6
 - 2.1.1 | Exercise and Mood..... 6
 - 2.1.2 | Sleep and Mood 6
 - 2.2 | Similar Applications 7
 - 2.2.1 | Moves 7
 - 2.2.2 | Argus 8
 - 2.2.3 | Sleep Time..... 10
 - 2.3 | Common Wellness App Features 11
 - 2.4 | Our Approach 11
 - 2.5 | Basic Layout 12
 - 2.6 | Achievements 12
- 3 | Methodology and System Architecture 15
 - 3.1 | The Application Design 15
 - 3.2 | Functionalities..... 18
 - 3.3 | User Interface 20
- 4 | Implementation 24
 - 4.1 | Developing for the Android Platform 24
 - 4.2 | The Development Process 24
 - 4.3 | Development Overview..... 24
 - 4.4 | Tab Design..... 25
 - 4.5 | Database 28
 - 4.6 | Real-time Graph Using Android Graphview 29
 - 4.7 | Testing..... 30
 - 4.7.1 | Information Gathering 30
- 5 | Results 33

5.1 Testing Results.....	33
6 Discussion	36
6.1 Challenges	36
7 Conclusion	38
7.1 Accomplishments.....	38
7.2 Future Work	38
8 References	39
Appendix A: Pre-Application Distribution Survey.....	41
Appendix B: Post-Application Distribution Survey	44
Appendix C: Pre-Testing Survey Results	46
Appendix D: Post-Testing Survey Results	51

List of Figures

- Figure 1: Moves App User Interface..... 7
- Figure 2: Argus App User Interface 9
- Figure 3: Sleep Time User Interface 10
- Figure 4: Simple Tabbed Interface 12
- Figure 5: Achievement List..... 13
- Figure 6: Main Screen 15
- Figure 7: Detail Tab Views 16
- Figure 8: SAM Fitness Application Screenshot Flow 17
- Figure 9: Saving mood data 20
- Figure 10: SAM Fitness Application Main Tab 22
- Figure 11: SAM Fitness Application Activity Tab..... 22
- Figure 12: SAM Fitness Application Sleep and Mood Tabs 23
- Figure 13: Android Fragment Code Snippet..... 26
- Figure 14: Android Fragment Lifecycle 27
- Figure 15: The Database Schema for Sleep and Mood Data 28
- Figure 16: Android GraphView Code Snippet 29
- Figure 17: SAM-Fitness Sleep Tracking vs. Sleep Time 34

1 | Introduction

Since the emergence and popularization of smartphones, many mobile applications that track and record data about their users have been created. The classic example of this is the pedometer which utilizes the mobile device's built-in accelerometer to track the number of steps the user takes each day. Applications in this category, that track and record health or activity data about their users, are typically called Wellness or Fitness Apps. These Wellness Apps are designed to assist the user in pursuing a healthy lifestyle by encouraging them to perform positive activities, and improve lifestyle choices. Factors that are typically targeted by such applications include exercise, sleep, and diet. Understanding the nature of this relationship is crucial when designing a Wellness App. Applications like this have the potential to motivate its users into maintaining a cycle of positive lifestyle decisions and/or breaking a cycle of negative lifestyle decisions. Diet, exercise and sleep can influence several physiological pathways associated with depression and a bidirectional relationship likely exists between depression and these lifestyle factors, thereby creating a potentially increasing cycle of influence (Lopresti et al., 2013).

The goal of this project: The purpose of this project was to create a wellness application for the Android platform capable of tracking, recording, and displaying data relevant to a user's sleep, activity, and mood habits. This application also enables individuals to become aware of deficiencies in their everyday habits and will hopefully encourage the user to self-regulate towards improvement. The addition of an avatar to represent the user's current wellness state, 3D graphics and "gamification" features further promotes user engagement and continued usage of the application. The overall goal is to show the user his or her daily habits and help them to make choices that will result in a healthier lifestyle, and therefore, a happier life. This document will comprehensively describe such an application's research, design, testing, and development.

2 | Background and Related work

2.1 | Effect of Sleep and Exercise on Mood

2.1.1 | Exercise and Mood

Studies have shown the positive effects that increased exercise can have on a person's overall mood, and it has been widely recommended that people suffering from a history of mental disorders such as depression and anxiety incorporate more exercise into their daily routines (Peluso Mam et al 62). There are several theorized reasons for this correlation between an increase in daily exercise and an improved state of mind. The first is the idea of distraction. A person placing his or her focus on exercise rather than the stressful or anxiety-inducing aspects of life, are more likely to temporarily forget about these depressants and have an improved state of mind. Another possible side effect of continuous daily exercise is a general improvement of a person's self-image due to becoming more physically fit, and healthier overall. Exercise can lead to an increase in self-esteem, and physical acceptance (i.e one's acceptance of his or her own physical body). In general, there is a direct correlation between a person's self-esteem and their overall happiness. Finally, when a person exercises, they are constantly overcoming obstacles and improving themselves. For example, a person that is only able to run for one mile at one point in time might be able to run two miles after spending several months running on a daily basis. This sense of achievement brought about by hurdling a personal obstacle, can also lead to an increase in overall mood (Sonstroem Morgan 329).

2.1.2 | Sleep and Mood

Similar to exercise, an increase in the number of hours slept on a nightly basis (eight or more) has been shown to be tied to an increased level of overall happiness. Conversely, getting less than the recommended amount of sleep per night can cause irritability, inability to properly cope with stress, and an overall impaired state of mind (WebMD *Sleepiness: Cognitive and Emotional Effects*, 2). Lack of sleep can also affect other areas of a person's life including exercise, dietary habits, and relationships with other people, all of which have an effect on a person's mood.

2.2 | Similar Applications

Applications that manage wellness have become some of the most popular downloads for smartphones today. These apps commonly track various types of user inputs such as exercise, sleep and other health habits. In theory, organizing this information in an easily interpretable way, motivates the user to continue a healthy lifestyle. Each of these apps has a unique way of expressing data to the user, and some methods have proven more effective than others.

2.2.1 | Moves

Moves is a mobile application that automatically tracks a user’s movement and location throughout the day and displays it in a timeline format. The app counts the number of steps taken per day, as well as the number of calories burned and the distance the user walked. Key places are recognized, and unrecognized places can be given a name by the user that will be remembered by the app. These travels are tracked and displayed on a map and timeline over the course of the day, as shown in Figure 1. To continue motivating a healthy lifestyle, the user is able to set goals for the number of steps he or she wishes to take throughout the day. In addition, a record can be kept of various exercises or activities that were performed that are unable to be automatically tracked by the app (such as gym workouts or various sports where the user may not carry their device). All of the information gathered about the user is easily sharable with other apps.

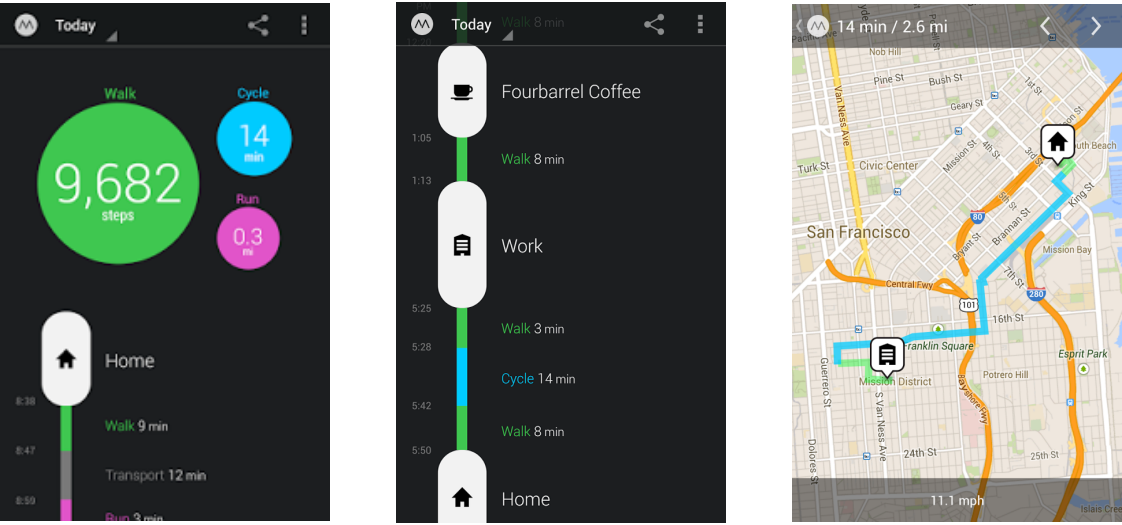


Figure 1: Moves App User Interface

Moves has a number of features from which we drew inspiration. One unique feature is the application’s ability to begin tracking movement as soon as the app is downloaded. There is not an initial setup of creating an account or need to go through a long calibration process. The main screen is very clean and simple. It contains a few circles displaying key information as soon as the application is opened. These circles are interactive and can be “tapped” to cycle through different information. The polished interface, paired with a quick setup process, creates a very solid movement tracking app. One limitation of the Moves app, is that it only tracks movement. This is not consistent with the plan for our wellness app, as our goal includes tracking sleep and mood habits as well.

From analyzing Moves, we decided to aim for a simple interface with a single initial view that displays the most important information. Fewer buttons and an informative display of the data, makes it very simple to see all of the necessary information on one screen. We found that the quick setup process, with no user registration, made the app extremely easy to dive into and start using immediately.

Pros	Cons
<ul style="list-style-type: none"> • Clean layout 	<ul style="list-style-type: none"> • Only collects movement data
<ul style="list-style-type: none"> • Doesn’t require an account 	

2.2.2 | Argus

Argus is another wellness app with a number of motion and life tracking features. Unlike Moves, which can only monitor movement, Argus features the ability for the user to create and track custom step and workout goals. It has a somewhat customizable main screen, where the user can display a pedometer and distance walked, reminders, as well as information about friends also using Argus. This app allows the user to set reminders to drink a glass of water or walk more throughout the day. Users are urged to connect with their friends on the app in order to gain motivation to strive towards a healthy lifestyle. This app can be linked to other “sister” apps that are able to track sleep and create more in-depth workout plans. All of these activities can be viewed in a graph format over days, weeks, months, or years. Although Argus has more features and trackers than many other wellness apps, the overall app seems less polished and somewhat disorganized.

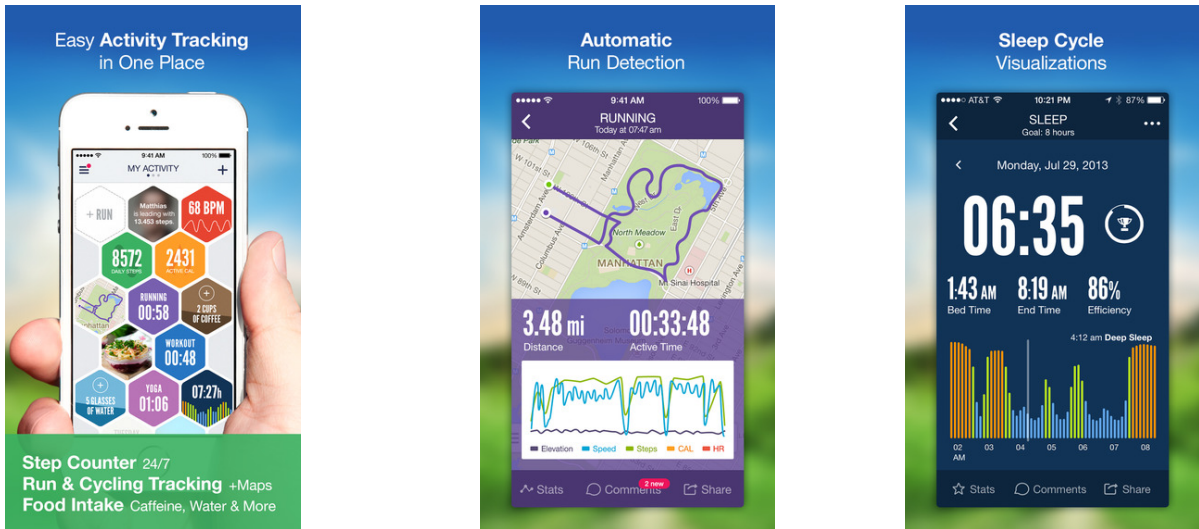


Figure 2: Argus App User Interface

The main display screen in the Argus app (Figure 2) is not easily interpretable. The daily workout information is displayed in hexagonal shapes that fit together as tiles. Although this keeps each piece of information separate, it makes it very difficult to see which information goes with which days, as they are all displayed together in this tiled format. In addition, Argus has so many features that it cannot display all of its information in a useful manner. Much of the information can only be viewed by navigating to different screens, which does not lend itself to comparing health habits and does not present a comprehensive glanceable display. There is also a long setup process associated with downloading the app. The user is required to create an account and profile, as well as take a number of steps so that the app can begin to calibrate itself.

Argus provided many examples of activities and information that could be stored about a user in a wellness app. The overall layout was messy and confusing, which reinforced our decision to push for a cleaner design. In addition, the long setup process made it far less appealing to begin using the app. We found it helpful to see charts and information about movement over time, which helped to give us ideas about how to display our graphs. Overall, Argus reinforced our simplistic design decisions and gave us inspiration for graph and chart layouts.

Pros	Cons
<ul style="list-style-type: none"> • Many tracking options 	<ul style="list-style-type: none"> • Long setup time
<ul style="list-style-type: none"> • Sister Apps 	<ul style="list-style-type: none"> • Unintuitive layout

2.2.3 | Sleep Time

Sleep Time is a “sister” app to the Argus app and tracks a user’s sleep habits and individual sleep cycles. By placing the device on the corner of the bed before sleep and pressing the “Start” button, the app is able to track a user’s movements throughout the night and infer the periods in which they are in a light sleep or a deep (REM) sleep (Figure 3). Before starting the app, the user chooses a time when he/she wishes to wake up and Sleep Time will set off the alarm at the user’s most shallow sleep point, within a half hour leading up to that time. This ensures the user is woken up at the optimum time and helps to eliminate morning grogginess. The overall goal of the app is to show the user his or her sleep cycles and wake up the user at a more optimum time than a standard alarm clock.

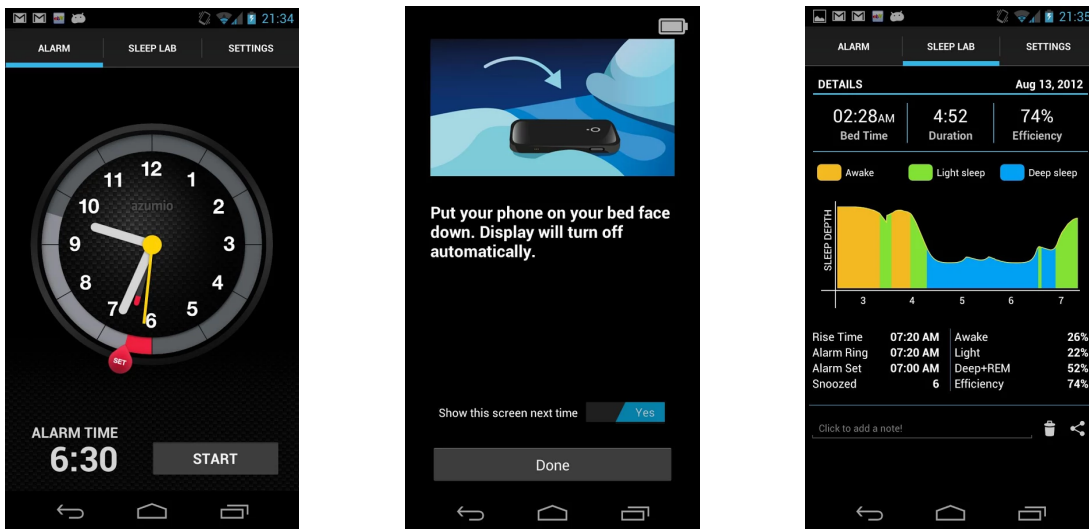


Figure 3: Sleep Time User Interface

This app strictly tracks only sleep, which is not consistent with our need for an overall wellness tracker. However, the idea of tracking sleep cycles and displaying the amount of time for which a user was asleep is a key feature of our app. We found the graphs simple to read and a good way to show the user’s habits over time. Unfortunately, it is unclear how accurate this app’s ability to track specific sleep cycles is. Overall, Sleep Time helped to inspire a clear display of a user’s sleep habits, as well as, shed light to possible future features.

Pros	Cons
<ul style="list-style-type: none"> Simple graphs 	<ul style="list-style-type: none"> Only collects sleep data
<ul style="list-style-type: none"> Has an alarm 	

2.3 | Common Wellness App Features

After researching the many existing wellness apps, we were able to identify several common features that we considered for our own design. We now list and expand on these features:

- **Step Counter:** One of the most common is the step counter. The implementation of which would require a subroutine that uses the device's accelerometer to record the number of steps that the user has taken. Moves takes this a step further with GPS tracking and records distance traveled differently, depending on whether the user is walking, running, biking, or riding in a vehicle. This presents us with an important design decision, as we have the choice between tracking activity with the accelerometer, GPS data, or both.
- **Goal Setting:** Another common feature is goal setting. This is a relatively simple way to encourage habitual use of the app. Including this is an important design consideration, not just because it can be implemented for any data that is recorded or provided by the user (such as with a calorie counter or time slept), but also because we could integrate game elements in order to further encourage habitual use of the app. It is also common for wellness apps to support integration with other apps. This is typically to allow data collected from one app to be imported into another app for analysis. This introduced us to another important design decision: whether or not to include data import/export (and, if so, which apps to support).
- **User Accounts:** Perhaps the most common feature is the use of user accounts. This allows users to be able to access their recorded data from any device capable of running the app. These apps typically require an account to use the software, but Moves is an important example of an app that requires no account. We were now faced with the interesting design decision between binding user data to user accounts, or binding user data to individual devices.

2.4 | Our Approach

Our approach to designing the app reflects our original objective: to create a wellness application for the Android platform capable of tracking a user's sleep, mood, and physical activity, and encouraging the continued use of the app. Many wellness apps have been created in the past, so the first step to accomplishing this was to research existing wellness apps and app design. We found a number of apps that tracked activity and sleep, so we drew inspiration from features and formats we found intuitive to design our app's interface.

2.5 | Basic Layout

Through research of similar apps we found a number of features that seemed useful in our design.

Simplistic Design: Overall, we found that the Moves app presented an extremely clean and simplistic layout that presented important information as soon as it was opened. We decided to model our app with this same mentality of keeping screens simplistic and present important data upfront.



Figure 4: Simple Tabbed Interface

Separate screens to display different lifestyle aspects: In addition, we appreciated how the Argus app had separate screens for a breakdown of activities and sleep, as well as the other data it tracked. Although we did not agree with the overall layout Argus presented, we drew inspiration for separate detailed screen for each activity, sleep, and mood. We modeled our tabbed interface after this idea (as shown in Figure 4). The Sleep Time app went much more in-depth with tracking sleep than our app is able to go, but we found it useful to see the data gathered by the app (such as the amount of time spent sleeping) and felt it left the door open to later expansions. Overall, we found inspiration in a number of different apps that helped to shape the various features of our project.

2.6 | Achievements

Early in the design process, we considered using achievements as a way to motivate and incentivize users to maintain wellness. These achievements would unlock

whenever a user completes a specific goal, for example, taking one thousand steps. Unlocking any achievement would trigger the appropriate in-app notification, and be added to a list of unlocked achievements available on the main tab.

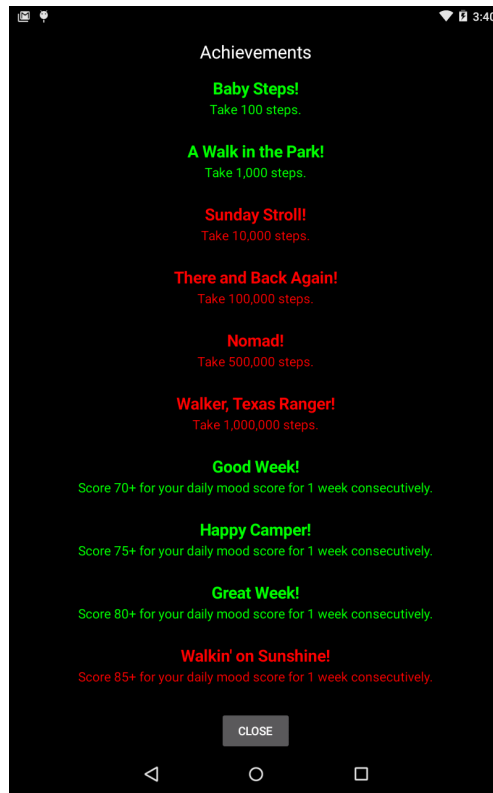


Figure 5: Achievement List

Figure 5 shows the achievement list available from the main page of our application. Unearned achievements are listed in red, while earned ones are shown in green. Achievements can only be earned once and are used as long term goals to incentivize healthy habits in the users.

After some discussion we ended up creating a total of 18, each with it's own clever name and description:

Achievement Title	Achievement Description
Baby Steps!	Take 100 steps.
A Walk in the Park!	Take 1,000 steps.
Sunday Stroll!	Take 10,000 steps.
There and Back Again!	Take 100,000 steps.
Nomad!	Take 500,000 steps.
Walker, Texas Ranger!	Take 1,000,000 steps.
Good Week!	Score 70+ for your daily mood score for 1 week consecutively.
Happy Camper!	Score 75+ for your daily mood score for 1 week consecutively.
Great Week!	Score 80+ for your daily mood score for 1 week consecutively.
Walkin' on Sunshine!	Score 85+ for your daily mood score for 1 week consecutively.
Best Week Ever!	Score 90+ for your daily mood score for 1 week consecutively.
Cloud Nine!	Score 95+ for your daily mood score for 1 week consecutively.
Power Nap!	Sleep at least once in a 24 hour period.
Good Night's Sleep!	Sleep for at least 8 hours in 1 day.
Enter Sandman!	Sleep for at least 56 hours in 1 week.
Sleeping Beauty!	Sleep 1000 hours.
Snorlax!	Sleep 2500 hours.
Rip Van Winkle!	Sleep 5000 hours.

3 | Methodology and System Architecture

3.1 | The Application Design

Early in the design process we considered having the “home screen” include a 3D room with a 3D character model in it, representing the user. This model would then change dynamically, based on the information collected about the user. After some discussion, however, we came to the realization that this particular design may have taken more time to implement than the six months that we had to complete the app, and we opted for a simpler tabbed layout.

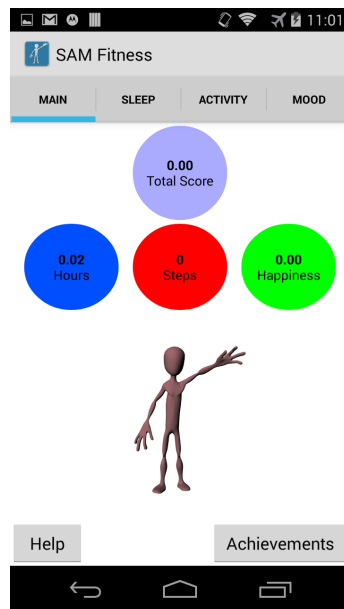


Figure 6: Main Screen

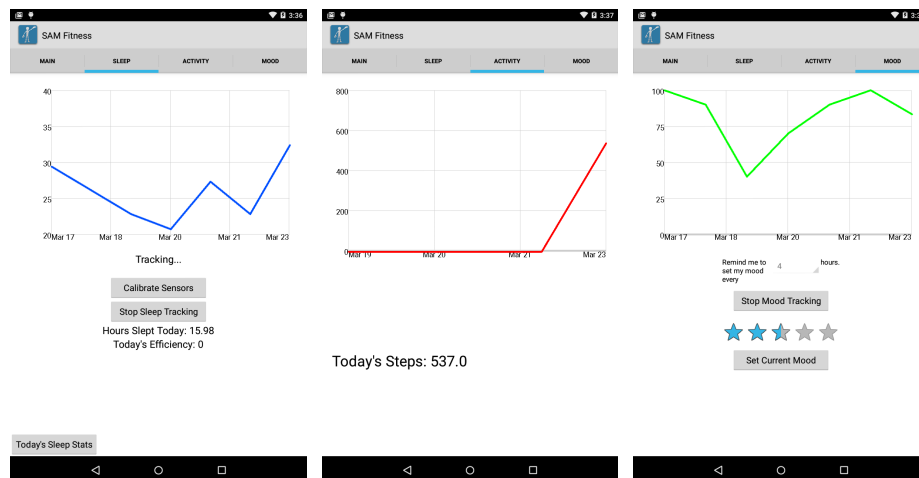


Figure 7: Detail Tab Views

The final design we chose consists of a single main screen with four additional tabs (Main, Sleep, Activity, and Mood, as seen in Figure 6). The main screen contains a video of the 3D avatar walking into view, buttons that contain scores for each metric that is tracked, a fourth button that contains a total score of all three, a button for displaying the achievements list, and a help button that displays a description of the application and its functions (Figure 6). Each of the next three tabs (shown in Figure 7) contain a graph and UI elements specific to the data collection of the given tab. In Figure 7 (left), is the sleep tracking tab. This tab consists of a graph displaying how many hours a day the user has slept for the past week, as well as, buttons for calibrating the sensors and viewing more detailed information about their sleep habits. The middle tab in Figure 7 shows the activity tab. This tab also contains a graph, but this one tracks how many steps the user takes throughout the day. The final tab (Figure 7, right) shows the mood tab. This tab is similar to the previous two, but the graph represents the users mood over the past week. Below the graph is the rating bar where the user enters his current mode by selecting stars based on how they are feeling. The rating bar increases at 0.5 star intervals.

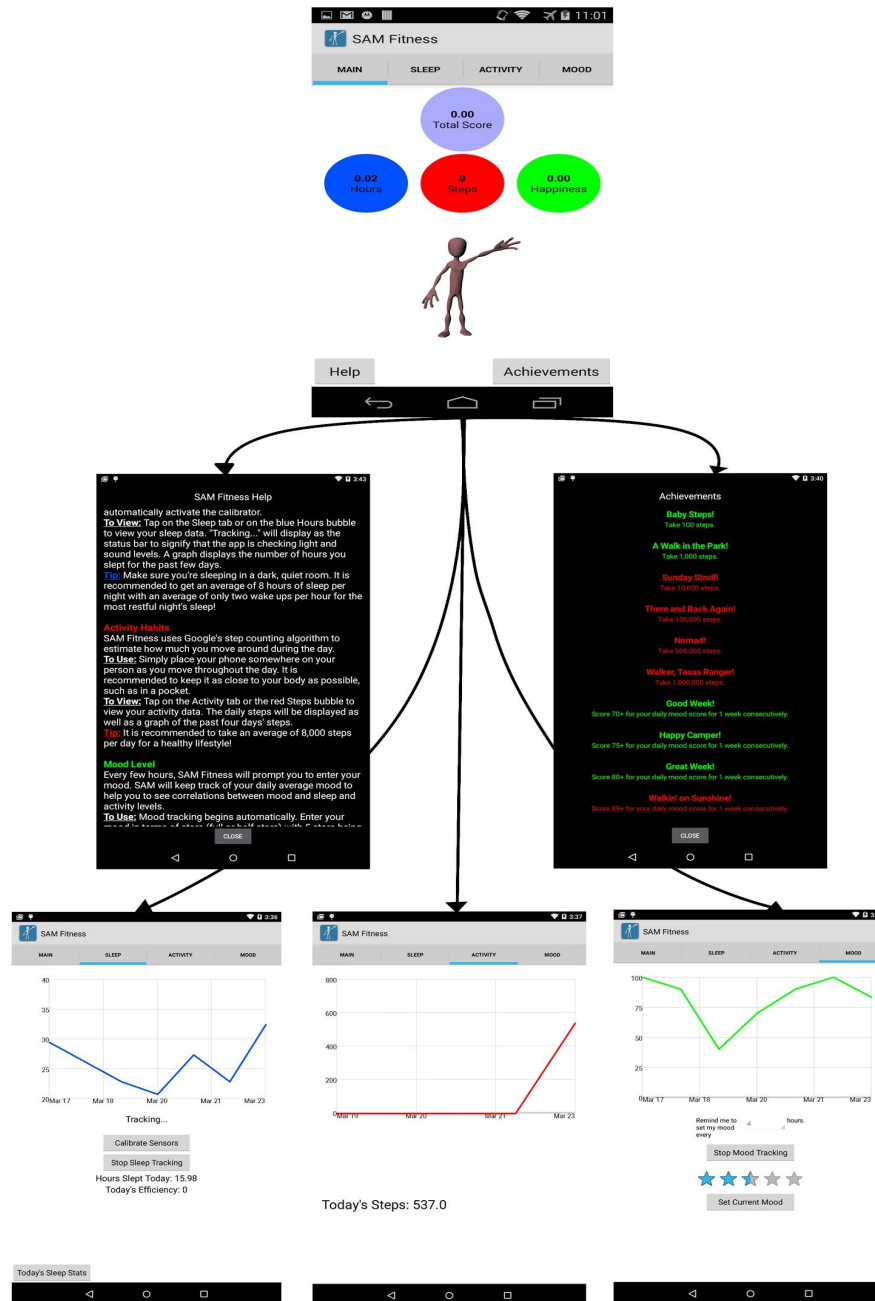


Figure 8: SAM Fitness Application Screenshot Flow

As shown in Figure 8, the main tab is the central hub of our application. It shows the most important information to the user immediately when they open the app. By simply tapping on piece of information, the app will take the user to a more detailed view. Also from the home page the user can view their achievement progress, as well as, view the in-app help.

3.2 | Functionalities

Step Counting

When a user has their phone in their pocket or attached to their body in some form, and they take a step, the phone will move in a distinct manner. By using the sensors in the phone, we hoped to develop an algorithm to track this movement, and, in turn, track a user's steps.

After some research, we found a simple step counting algorithm that tested the change in the phone's position on the y-axis, and if this change in distance was greater than some user defined threshold, the algorithm would add a step (Michaud). After testing this algorithm for some time, we found that it was not as accurate as we had anticipated. We decided to adjust the algorithm to not only track the change in the y position of the phone, but to calculate the total change in distance of the x, y, and z coordinates. Unfortunately, this was not much more accurate in counting steps than our previous implementation so we began researching alternative methods for tracking exercise. We eventually came across the Google Fit API, which utilizes a phone's various sensors to implement algorithms that record and store different sets of fitness data, including step counting. That data is sent to the Google Fit Service and is then available to the developer to use as he or she needs.

There are several components to tracking fitness data using the Google Fit API. First, the app must be connected to Google Play Services, and the user must connect a Google account to the app. Next, the app is required to request the user's permission to record his or her fitness data. Once permission is granted, the app will subscribe to the Google Fit data stream or streams chosen by the developer and this data will be stored on the phone. The developer can then access the stored data to use in their own application. In our implementation, we read the step counting data every time the user opens the app or activity tab and use it to update the graph, the number of steps displayed to the user, and check for newly earned achievements.

Sleep Tracking

Sleep tracking occurs when a user places his or her phone in an environment where certain requirements are met. In most cases, users sleep in quiet, dark places at night. Our sleep tracking algorithm uses the phone's microphone to monitor sound levels, the phone's light sensor to monitor light levels, and the phone's clock to monitor the time of day. If all of these three factors fall within a dynamic calibrated threshold, sleep tracking begins automatically.

When sleep tracking begins, the app will continually check the environment based on given sound and light thresholds. The user can choose to click the 'Calibrate' button, which will take in the sound and light levels for a ten second period. After this time, it will average the values and add a buffer (to allow for daybreak/movement during sleeping), and a new threshold is created. This is helpful for users who sleep in a room with more light (perhaps in a city setting) or for users who sleep with background noise (such as a tv or soft music).

In addition to checking the noise and light levels, the app will only track sleep if it is between 8PM and 12PM. This helps to preserve the phone's battery life because it is not attempting to check sound and light levels continuously throughout the day. When the time of day requirement is met, the sound and light levels are checked continually. When all three conditions are satisfied, it is assumed that sleep has begun and the date and time are recorded. When the phone's sensors are no longer within the given thresholds, it is assumed that the user is no longer sleeping.

The efficiency of the sleep session acts as the scoring system for sleep habits. It is based on the amount of time the user was asleep (duration) and the number of times the user woke up within that amount of time. When the user has been asleep and a level changes, it is counted as a "wake up." However, if the conditions fall back into the required sleep thresholds before sleep tracking stops, the user is assumed to have fallen back to sleep and the duration is added cumulatively. Based on our testing, we found that the most effective sleep cycles consisted of an average of two "wakeups" per hour. Therefore, the number of expected wakeups is found by multiplying the number of hours slept by two. From this, the difference between the number of actual wakeups and expected wakeups is found, and then it is multiplied by a set point value of 0.625. If the user has been sleeping for less than an hour, the difference is instead multiplied by 6.25, as it is far less efficient to wake up more times in under an hour. This final value is subtracted from one hundred to get the efficiency percentage.

At any point in time the user is able to view the previous sleep session's statistics by clicking the "+" button in the upper lefthand corner. This brings up a popup that displays more specific information, such as the number of wakeups and the calibrated light and audio levels. If a user sees high calibrated light and audio numbers, as well as a high number of wakeups, he or she may be more motivated to create a more restful sleep environment.

The user is also able to manually stop sleep tracking if he or she wishes by tapping the "Stop Sleep Tracking" button. This stops the app from continually checking sleep and light sensors. The user can easily restart sleep tracking by tapping the "Start Sleep Tracking" button, which re-enables all sensor checking and begins tracking again, accordingly.

Mood Tracking

Due to the limitations of automatically inferring and tracking a complex qualitative metric such as mood, we chose a relatively simple user polling system for collecting mood data. We made a service that sends a notification to the phone every four hours (while sleep tracking is disabled) that reminds the user to input their mood. Figure 9 shows how when inputting their mood, users are presented with a 5 star slider that ascends in increments of $\frac{1}{2}$, where 0 stars represents a very bad mood, and 5 stars represents a very good mood. Each mood input is stored along with a timestamp in the database and, similar to step tracking, we read the mood data every time the user opens the app and use this data to update the graph on the mood tab.

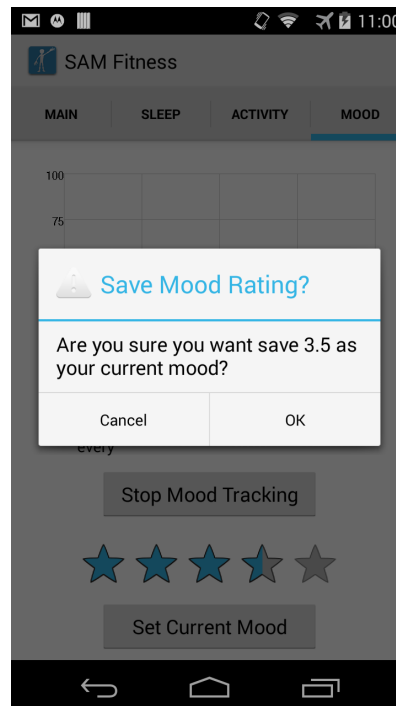


Figure 9: Saving mood data

3.3 | User Interface

In order to effectively incorporate each of the three focus areas of our application (sleep, exercise, and mood) into the user interface without creating too much clutter, we opted to use a tabbed design. When the app is initially opened, the user is taken to the “main” or “home” tab. There are several components displayed within this screen. At the top are four bubbles of varying colors with numbers contained within them. Each of the three lower bubbles are connected to one of the three activities being recorded within

the app, and the number contained within them represents a score attached to that activity. For example, if a user's total number of steps for the current day was greater than the previous day's step count, the user's exercise score would also increase from the previous day. The number contained within the bubble above the lower three bubbles represents an overall score. This score is calculated from the user's inputted mood, the number of steps, and number of hours slept based on the National Averages. "A value of 10 000 steps/day is gaining popularity with the media" (Tudor-Locke, 2004), so we based our scoring algorithm on 13,000 steps to earn a 100% activity score. The National Sleep Foundation recommend adults get 7 to 9 hours of sleep each night (National Sleep Foundation, 2015) , so we based our scoring algorithm on nine hours to earn a 100% sleep score. A user who has an average mood of 85 or higher (all 5-star ratings) is considered to have a 100% mood score for the day.

In the bottom half of this screen is an animated humanoid character that is used to give the user an interactive display of his or her current status based on the calculated scores. If, for example, the user is sleeping less than normal, the character will appear lethargic and sluggish. If he or she is lacking exercise, the character will begin to gain weight. The hope is that the user will form a connection to the character and will want it to appear "healthier" or more active. As a result, he or she may be more inclined to live a healthier lifestyle. Figure 10 shows the main screen that is displayed when the SAM Fitness is first opened.

Swiping two tabs to the right from the main screen brings the user to the activity tab. Figure 11 shows the user interface of this tab, which includes a graph that shows the user's step count for each day of the past week. Below this graph is the current number of steps the user has taken, so far, for that particular day which will update each time the user opens the app. The purpose of this screen is to allow the user to compare his or her exercise habits throughout the week, and make changes if these habits are below their general goals.

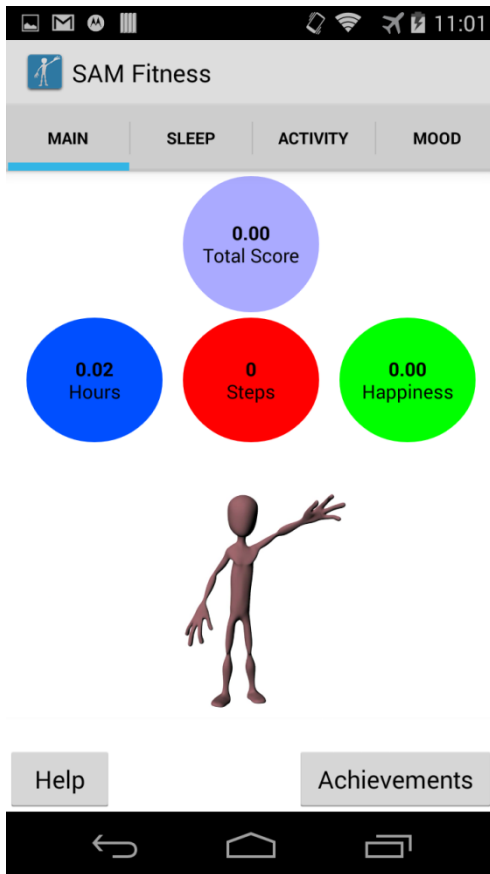


Figure 10:
SAM Fitness Application Main Tab

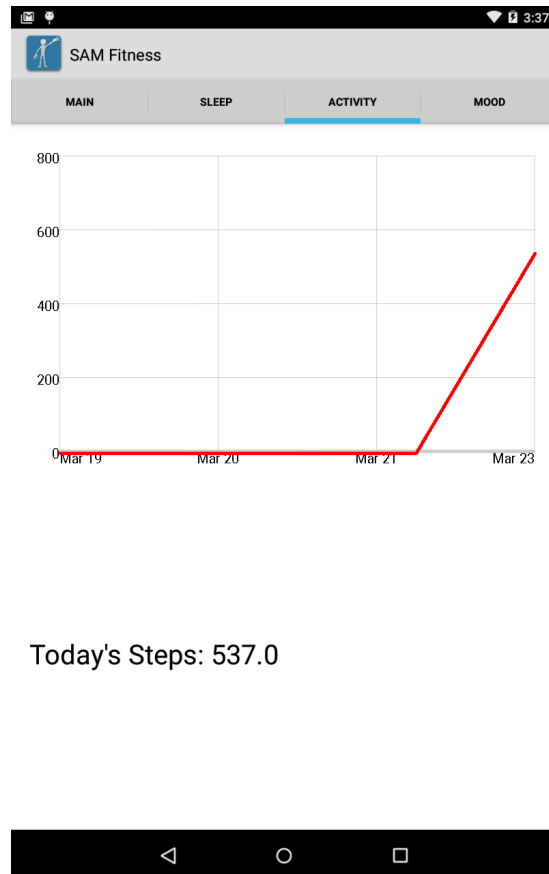


Figure 11:
SAM Fitness Application Activity Tab

The sleep tracking tab, located to the left of the activity tab, has a similar layout. Figure 12 (left) shows the user interface of this tab, which includes a similar graph displaying the number of hours slept for each night in the past week. Below this graph is a button that, when pressed, begins the process of checking if the user is asleep by testing various factors including audio levels, light levels, and any indication of movement of the phone. If all these variables are below a certain threshold, sleep tracking will begin. The next day, underneath the “Begin Sleep Tracking” button, the sleep tracking tab will display the time that the user fell asleep, the time that he or she woke up, and the number of cumulative hours slept the night before. In addition, the graph will be updated with the hours slept from that night.

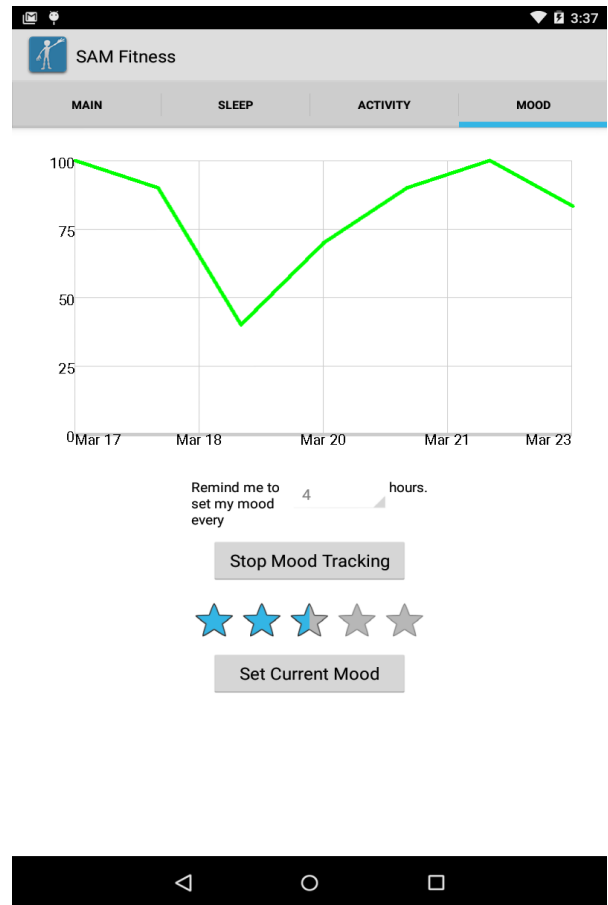
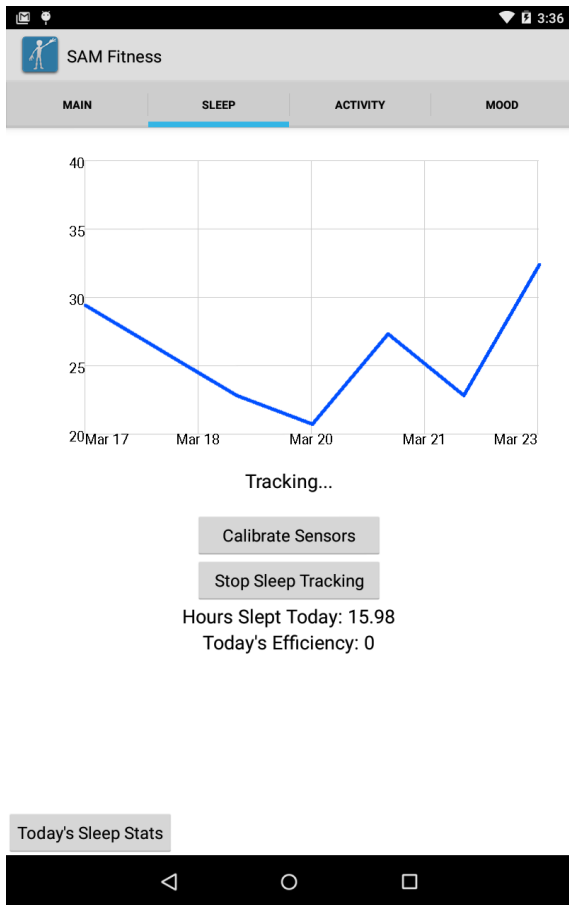


Figure 12: SAM Fitness Application Sleep and Mood Tabs

The final tab, the mood tab, is also very similar to the exercise tab. Figure 12 (right) shows the graph at the top of the tab that displays the past week of mood data. Below this graph is a button that allows the user to start and stop mood tracking. Next, there is a 5 star slider and a set mood button that allows users to set and input a mood on a scale from 1 to 5 in increments of $\frac{1}{2}$.

4 | Implementation

4.1 | Developing for the Android Platform

In developing our Android application, we utilized the Eclipse integrated development environment bundled with the Android Development Tools plugin for approximately three quarters of the development time. About halfway through our project, Google released Android Studio, an IDE created specifically for designing Android applications. Due to some complications integrating the Google Fit API into our application using Eclipse, we decided to transfer our project to Android Studio. This switch not only alleviated our problems with the Fit API, but also assisted in making the removal of other bugs a more simple process due to Android Studio's relatively seamless integration with Android development (specifically, the appcompat support library).

4.2 | The Development Process

The development process consisted of weekly meetings, in which the entire group met to come up with goals for the next week and review the progress of current programming tasks that were assigned to each team member. We also used GitHub (GitHub, 2014) to manage our collective changes to a shared repository. Our goals were often driven by the obstacles we had faced during the previous week and were prioritized by how severely they impacted further development of the app.

4.3 | Development Overview

At the beginning, much of the work was individual, as we were all learning the syntax and architecture of Android development. As time progressed, we split into two main teams; one that worked with the development of the app itself and one that worked with the development of the character that would personify the information collected by the app. Later this was broken down further into more individual tasks within each team, which typically exemplified each team member's individual strengths. More specifically, one team member worked on the process of tracking exercise, one on sleep tracking, and a third on the process of recording a user's mood. Once we had all of the pieces we worked to integrate them to make one app.

4.4 | Tab Design

Relatively early in the development process, we came to the decision to separate each data tracked aspect (steps, sleep, and mood) into individual tabs within the app. After researching several different methods for implementing this type of multi-page app design, we came to the decision to use the *ViewPager* layout manager, an Android class that is “most often used in conjunction with *Fragment*, which is a convenient way to supply and manage the lifecycle of each page” (*ViewPager*). Each of the four tabs is implemented in its own class, and is a subclass of the *Fragment* class. In addition, the tabs share a single Android activity, which is the Android class that handles all user interaction with the app. The main activity of the application manages the tab layout and tells the application which tab view to display when a user selects a certain tab. The purpose of using fragments is to eliminate the need to create a new activity every time the user switches between tabs, thus increasing both the temporal and spatial efficiency of the application. Structurally, each fragment is comprised of two components. The first is an XML file that defines the visual layout of the fragment. The second is the logic of the fragment, which contains various functions defining certain actions to take at different times. Each time a certain fragment is switched to or away from, these functions are called in a particular order, all of which make up what is called the fragment’s “lifecycle”. For example, when a particular fragment is displayed on the screen, the function `onCreateView()` is called, which is generally where most of the initialization process occurs. A diagram of this lifecycle can be seen in Figure 13.

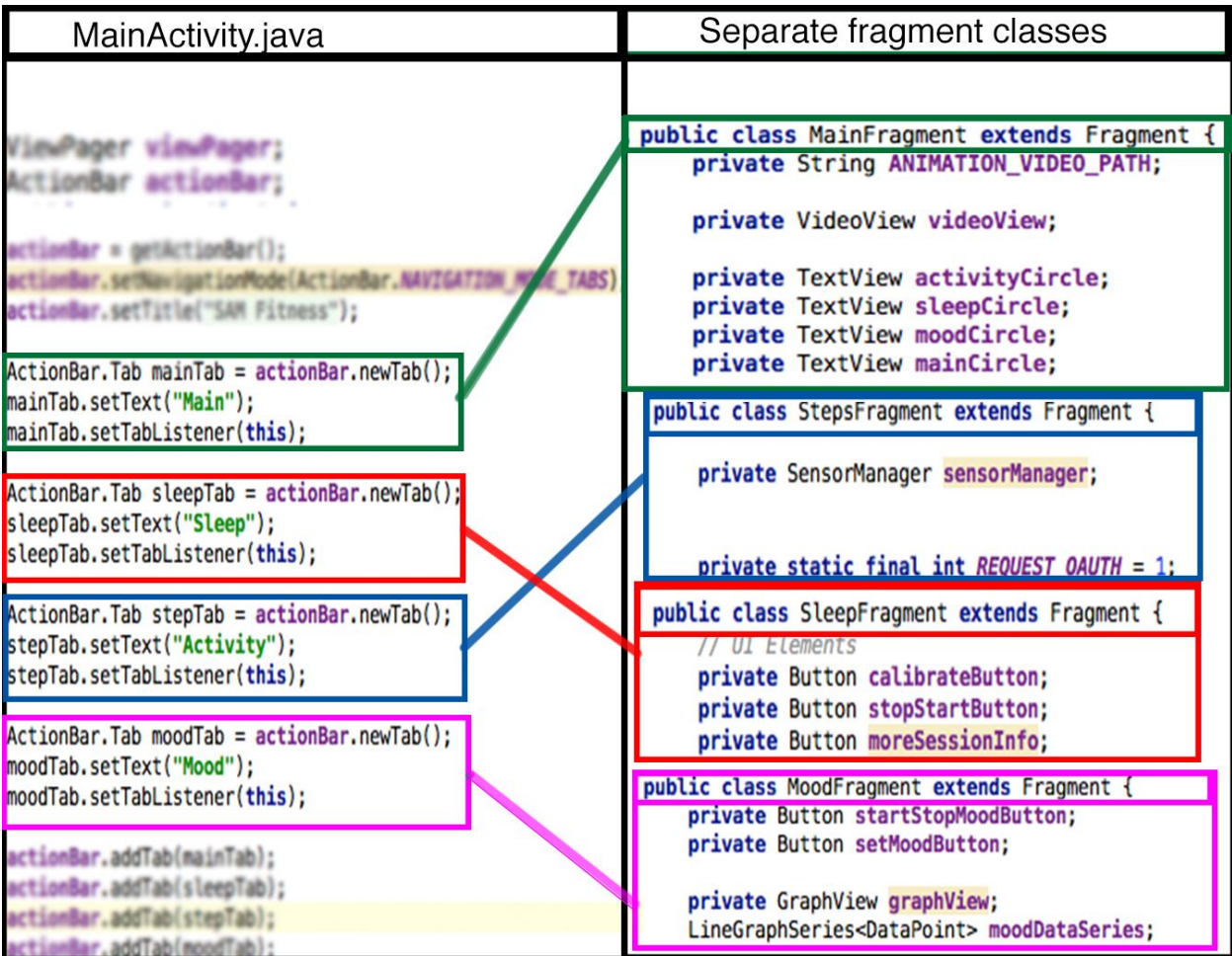


Figure 13: Android Fragment Code Snippet

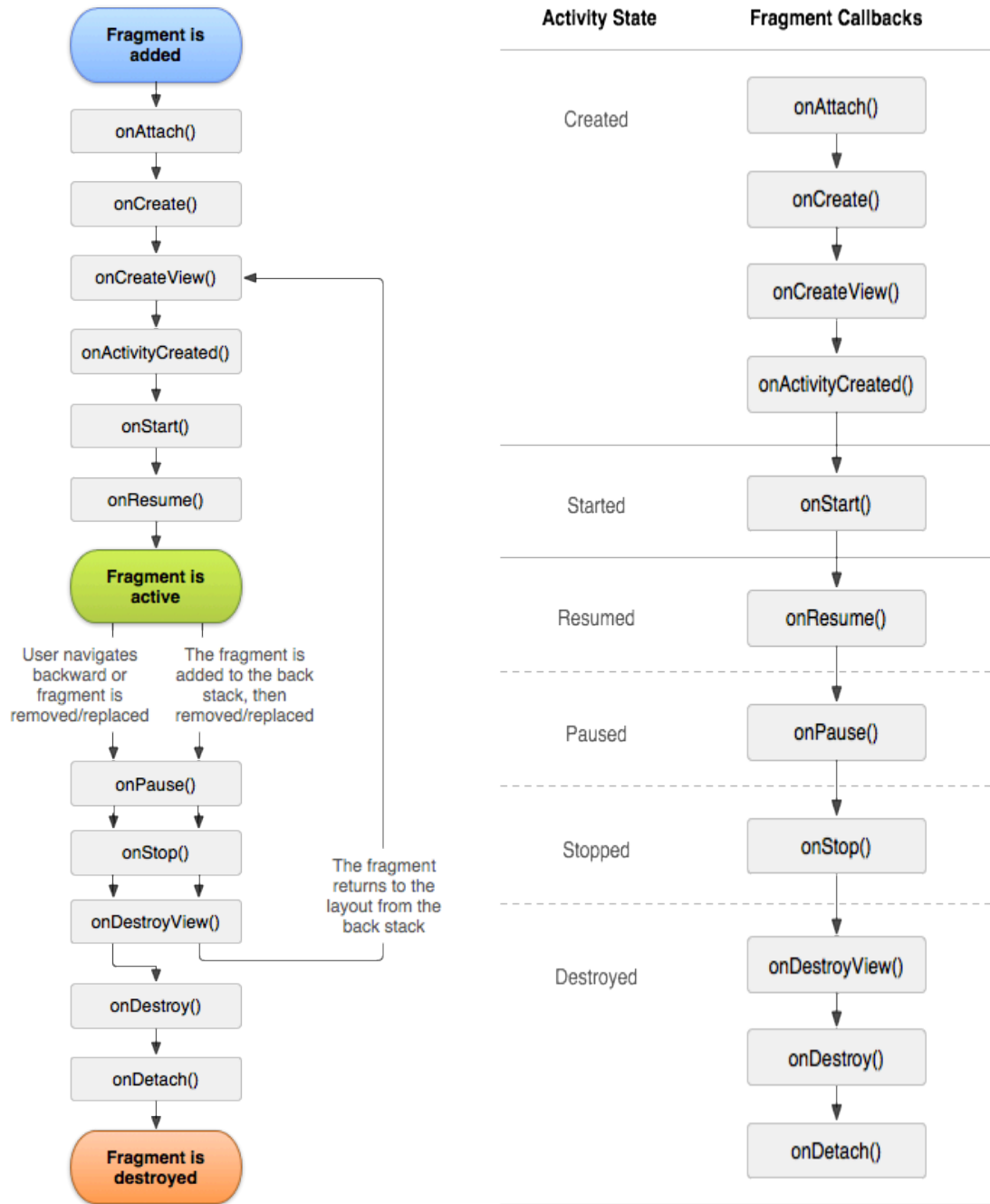


Figure 14: Android Fragment Lifecycle

4.5 | Database

In order to effectively store the user's step counting and sleep tracking history, we decided to use a SQLite database, which is already built into the Android operating system. Because Google Fit integration automatically manages the step counting data, our database only has two tables, one for mood data and another for sleep data.

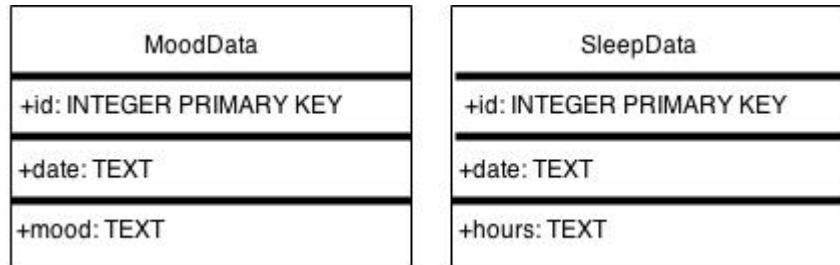


Figure 15: The Database Schema for Sleep and Mood Data

The first step in implementing the database was to create objects that could be used to store and retrieve information for each data type. More specifically, we created a sleep tracking class and a mood tracking class. Each class has two variables, as well as methods for setting and getting these variables. As seen in Figure 15, the sleep counting class contains a variable for the number of hours slept the night before, and the mood tracking class stores a number ranging from one to ten (where one represents the least happy, and ten represents the happiest). Each class also contains a variable representing the date at the time the data was stored. We then use these classes to periodically store the numerical data. For example, when the application detects that the user has woken up in the morning, it will calculate the number of hours that he or she has slept. It then creates an instance of the sleep counting class with that number of hours, and the date of the day before. It will then use that instance to create a SQL query that inputs the data into the database. Finally, it will use that same information to update the information displayed to the user.

One of the features of the Google Fit API is that it conveniently stores the collection of fitness data that it records in the Google Fit Store. This is a cloud server that the data is sent to. The data can be easily accessed by sending a data read request to the server. If the phone is offline when the request is sent, it will store the request and wait until it has once again been reconnected to the internet to retrieve the data. Due to this integration of data storage built directly into the API, the SQLite database, which we had previously used, was no longer necessary for the storage of step tracking data. Instead, the app periodically retrieves aggregated step counting data from the Google Fit Store, and uses it to update the graph as well as the current number of steps for that day.

4.6 | Real-time Graph Using Android Graphview

We decided that our app would be most effective in encouraging users to live a healthier lifestyle if they could compare the fitness data from the current day to the fitness data of previous days. This way, they could try to improve over time. We concluded that the best way to display this data would be to use a real time graph. After looking into several different methods for integrating a graph into an Android application, we decided that the best option would be to use an open source third party library. We found a few graph libraries that had real time capabilities, but ended up using the Android GraphView library due to its thorough documentation and ease of implementing within a fragment (Gehring). The graph is first initialized in the onCreateView() method of the particular fragment that it is being displayed on. This method is called each time the fragment is redisplayed, so the graph gets reinitialized each time a user swipes to that particular fragment. In the onResume() method (which is called after the onCreateView() method), the application checks if there is any new data since the latest update, and if there is, it will add this data to the current dataset. The graph is then completely reset and redisplayed with the updated dataset. Generally, this dataset consists of seven data points (one for each day), each point consisting of an x-value represented by a date, and a y-value represented by some numerical fitness data (e.g number of steps).

```
private void updateGraphWithStepCounter(final ArrayList<DataPoint> values) {
    getActivity().runOnUiThread(new Runnable() {
        @Override
        public void run() {

            DataPoint[] getTimeStamps = values.toArray(new DataPoint[values.size()]);

            if (values.size() >= 1) {
                textViewSteps.setText("Today's Steps: " + String.valueOf(getTimeStamps[values.size() - 1].getY()));
                graphView.getGridLabelRenderer().setNumHorizontalLabels(values.size());
                graphView.getViewport().setXAxisBoundsManual(true);
                graphView.getViewport().setMinX(getTimeStamps[0].getX());
                graphView.getViewport().setMaxX(getTimeStamps[values.size() - 1].getX());
            }
            exampleSeries.resetData(values.toArray(new DataPoint[values.size()]));
        }
    });
}
```

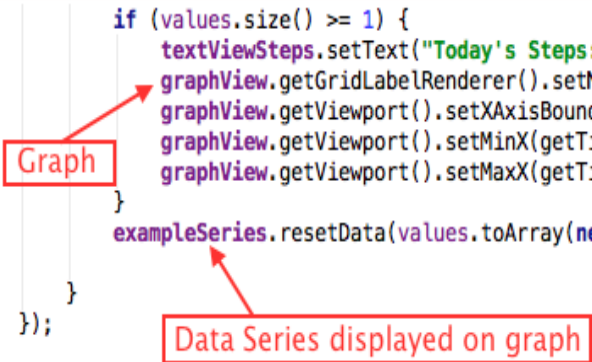


Figure 16: Android GraphView Code Snippet

4.7 | Testing

Initial testing was done on an Android simulator which ran on our computers. This aided in development because we were able to immediately load changes and view how they affected the app. This also eliminated the immediate need for a physical Android device and we were able to begin developing and testing our app immediately. Eventually, we needed to test the efficacy of our sleep and exercise tracking implementations, two processes which require the sensors of a physical phone, and so we purchased a Motorola Moto G Android Phone to complete the remainder of our tests.

4.7.1 | Information Gathering

Testing Step Counting

We used several different methods to test the efficacy of our step counting algorithm. We had to check that it successfully counted the correct number of steps in multiple different situations. For example, we considered how the algorithm would react if the user placed their phone in a backpack or in a purse, versus simply placing it in their pocket. To test each of these situations, we placed the phone in a different location for each test, and then walked 100 steps down a straight line (counting manually as we walked). We then compared the phone's results with the number we counted. Our original algorithm tested a change in position of the accelerometer's y vector to see if it moved past a certain threshold. We needed to calculate what this threshold should be in order to accurately track if the user took a step. To do this, we walked around with the phone and tweaked the threshold until it was as close as possible to the number of steps that we had counted. Unfortunately, we had difficulty finding a threshold that was neither too sensitive (i.e it tracked a step with every small movement of the phone) or not sensitive enough (i.e it rarely tracked any change in y-position as a step).

Due to the results of these tests, we decided to incorporate the Google Fit API's step counting algorithm into our application. After testing Google's algorithm using the methods above, we found that it was effective for counting steps. Next, we had to test if it was properly sending and receiving data from the Google Fit Service. To do this, we pulled step counting data from the past seven days from the Fit Store and displayed it on the activity tab.

Testing Sleep Tracking

There are a number of different factors associated with sleep tracking. To ensure that the overall algorithm was working properly, each piece needed to be first tested separately, then tested in conjunction with the other factors.

The light sensor was tested by setting a threshold and printing a boolean as to whether or not the light level fell within this threshold. Once the sensor was functioning we were able to print the light sensor values to the screen in real time. Next, we started testing the audio sensor. The audio levels are taken from a media recorder that takes short sound clips and analyzes the noise levels. To test this we again set a threshold and printed a boolean when the audio level satisfied the threshold. In ten second intervals, we were then able to print the audio levels, which helped us to choose a default audio calibration of 350.

The final factor in sleep tracking is the time of day. We found that most people sleep for a period of time between the hours of 8pm and 11am. When the time of day is within that threshold, the app automatically begins checking the audio and light levels to see if the user is asleep. To test this, the threshold was set to encompass the current time of day and to exclude it while we checked to see if sleep tracking had begun or stopped appropriately.

The calibration feature was another important piece of sleep tracking. Due to the fact that people sleep in all different environments (city vs. country) and with different external factors (radio, tv, night lights, etc), it is important for the user to be able to have the option to set audio and light levels that are more tailored to their particular environment. To ensure that the algorithm accurately averaged the levels over the course of ten seconds and added a buffer, we printed the audio values to the screen every second. From this, we were able to add up the displayed audio values, plus the known buffer, and get an expected light and audio calibrated value. In doing this we found that the first value stored for each sensor is zero, therefore, the number of seconds had to be increased to eleven to ensure there were ten entries stored before calculating the average. The buffer values for audio and light were chosen by carrying the phone in a pocket to make sure the sensors were sensitive enough and placing it on a night table to make sure the sensors were not overly sensitive.

To ensure the newly calibrated values were accurate, we printed them to the screen while also printing the current light and audio values in real time. When the values both fell within their respective thresholds, a boolean would be displayed to show the device was in sleep conditions.

The sleep efficiency is based on the overall time the user slept and the number of times he or she woke up in that time period. To test this, thresholds were set to allow sleep tracking to take place for one hour. Within the hour, it was expected that the user wake up two times, and any subsequent “wakeups” over this threshold would result in a 0.625 deduction of an efficiency point. In addition, if the user woke up more than twice in under a full hour of sleep, this would result in a 6.25 deduction of efficiency points. The number of times a user wakes up is displayed within the Sleep Status popup, which allows the user to compare the number of wakeups with their overall amount of sleep. A

user with a high number of wakeups could use this information to consider improving their sleep efficiency by changing their sleep environment to have less light and sound.

Testing Sleep Scoring

Sleep scoring is based on the number of hours a user slept. This was tested by keeping track of the time we went to bed and the time we woke up and comparing how accurate the time reported seemed to be. In addition, we tested our app against the Sleep Timer app by Azumio. It was usually within a half an hour difference from Sleep Timer's reported amount of sleep.

Testing the Graphs

The graphs within our application are an important element for displaying fitness data to the user so that they may view their progress over time, and so we needed to test that they were displaying accurate and appropriate information. We began by simply inserting a set of fake data to ensure the graph was formatted properly and displayed the appropriate values. Next, we had the various fitness elements push their data to the database and let the graph pull its data directly from there. Once we were able to successfully make a connection to the database we changed the device's date to ensure the graph would update each day and display previous data points correctly.

5 | Results

5.1 | Testing Results

Once SAM Fitness was completed and polished, we sent it to 10-15 Android phone users who assisted us in testing. Before sending them the application, we sent them a pre-testing survey that assessed general information about the user (See Appendix A). Once all users had completed the initial survey, we sent them the Android Application Package (APK) used for installing SAM Fitness on their phones. In addition, we sent them another survey (See Appendix B), used for assessing any bugs that might be present within the app, as well as gathering data regarding the effectiveness of the application's sleep and step tracking algorithms. We received 14 responses from the pre-testing survey and 10 responses from the post-testing survey. Please see Appendix C for response data from the pre-testing survey, and Appendix D for response data from the post-testing survey.

Through the feedback gathered from the surveys, we discovered several bugs. For example, several users reported issues with the sleep tracking algorithm:

"It thought I slept 200 hours last night....I don't feel like I slept more than 7"

"The sleep tracker said I slept 114 hours the first night I used the app... And woke up 50 times"

The high number of wakeups were produced by the sound and light sensors being too sensitive. This bug was fixed by increasing the thresholds to account for more light and sound without deeming the user as "awake."

We were also able to reproduce the bug regarding the highly unrealistic sleep times and found that the issue was associated with the way the sleep hours were being stored in the database. Initially, there was a total duration that was stored within the sleep fragment that was pushed to the database each time a user woke up. Eventually, we discovered that we had inadvertently been totaling all of these entries within the database, resulting in an exponential sum of sleep data. By making a simple change to the sleep fragment we were able to just push new duration (time in between wakeups) to the database and keep a running sum of the hours slept from there. Although these changes did appear to fix the bugs, we were not able to receive feedback in time to include here.

After these improvements were made, SAM Fitness was tested again in conjunction with the Sleep Time app (Section 2.2.3) to check its accuracy (as shown in Figure 17). On the same night, Sleep Time was run on an iPhone5S while SAM-Fitness was run on our Android test phone. Sleep Time reported 5:56 hours of sleep. On the same night, our app reported 5.97 hours of sleep, which works out to 5:58 hours of

sleep. Sleep Time reported a lower efficiency (83%) than our app, which reported 98%, but this is likely due to the fact that Sleep Time keeps track of actual sleep cycles, rather than just overall time spent asleep.

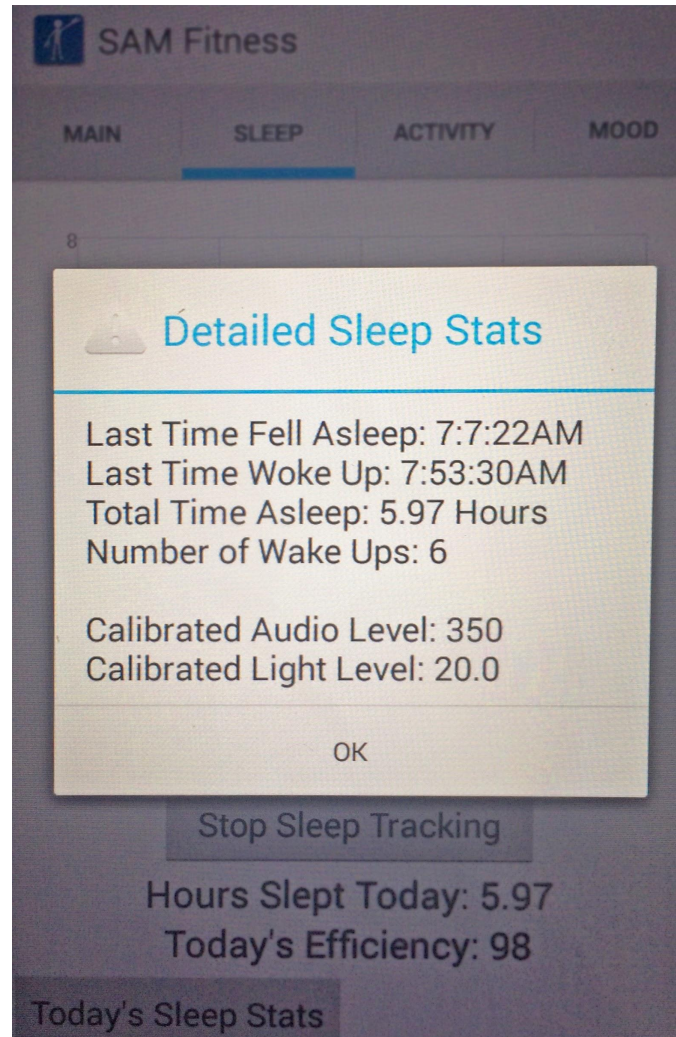


Figure 17: SAM-Fitness Sleep Tracking vs. Sleep Time

There were also two minor bugs reported, both of which we were unable to replicate ourselves. The first:

"I was asked several times to log back into google fit."

Though we were unable to reproduce this issue, we speculated that this could have been due to loss in network connection. We have noticed through general testing of the application over time that most of the issues we have encountered with Google Fit have been due to this factor.

The second minor bug noted by two users was with the rate at which they were prompted to update their mood:

“It did not ask me (to update my mood) every 4 hours.”

“It only reminded me to update my mood while I was asleep”

We believe this could be an ongoing compatibility issue with different Android devices and versions handling our notification intent code in unexpected ways. Action buttons, for example, only appear on Android 4.1 and higher.

6 | Discussion

6.1 | Challenges

IDE Issues

For roughly the first two and a half terms of development, we were using Eclipse with the ADT (Android Developer Tools) plugin suite. While many of the features worked perfectly, we had a lot of trouble getting the v7 appcompat support library to link properly on each team member's project. We initially thought that this was an issue with our GitHub integration, but the issue seemed to be machine specific as not every team member was affected. We attempted to reinstall all of the development tools, but the issues persisted. This affected the development of the app until we migrated the project into Android Studio in early February 2015. With this IDE integrated into GitHub, we never experienced any further linking issues with the v7 appcompat support library.

Test Devices

Another issue that affected the early development was our lack of a (relatively) modern Android device on which to test our project. For the first term and a half of development, we could only test our code on the Android emulator packaged with ADT. This worked for simple functions and designs, such as tabs buttons, but the real issue was that we were not sure how well our OpenGL code would work on an actual device. Luckily, we received a device on which to test this code in our second term of development. Unfortunately, our subsequent testing revealed an issue with OpenGL and Android's fragment lifecycle.

OpenGL and Fragment Lifecycle

Early in the development process we decided to use OpenGL ES 1.0, an API for hardware accelerated graphics on embedded systems, along with jPCT and the Bones animation library. These tools were used in order to render and animate the 3D avatar. Testing was difficult even when we had the 3D environment set up and the 3D model loaded into the scene. Once we were able to start testing on a physical device, we encountered a bug in which any animations playing on the main tab would pause and not resume if the user selects either the Activity or Mood tab before returning to the Main tab.

Art Assets

Originally, we had an Interactive Media and Game Development artist on the team. For the first term and a half he worked on designing, modeling, and animating the

main avatar. Due to unforeseen personal issues, he unfortunately had to drop the project halfway through. This presented a big challenge: what were we going to do for artwork? For a while, we continued to develop the application using an animated placeholder ninja model that came with the jPCT OpenGL tutorials. This worked for a while until we were able to find a solution.

Before the artist had to leave, he completed modeling the avatar, but did not rig or animate it yet. We reached out to various peers in the WPI IMGD community to see if someone could finish it, but everyone was either busy with their own academic work, or already contracted for other projects. We turned to the more professional side and started talking with experienced 3D animators on game development forms and freelance websites. We found one 3D artist who works at Pixar and freelances on the side. We opened a dialog with him to see if he would be able to complete the avatar, but we did not end up moving forward due to pricing and time constraints. We instead decided to create the animations ourselves, and used Autodesk Maya 2015 to create the walk cycle and wave animations visible in the final version of the app.

7 | Conclusion

7.1 | Accomplishments

While it was a challenge to develop, our project team successfully created a prototype wellness application for the Android platform capable of tracking, recording, and displaying data relevant to a user's sleep, activity, and mood habits. While the full scope of the initial app design was not realized, all of the core data tracking functionality has been successfully implemented.

7.2 | Future Work

While we feel that we successfully implemented the desired tracking functionalities, there are still several elements that could be added to improve the effectiveness of the app. The primary example of such an improvement is to resolve the issue with OpenGL and the fragment lifecycle. Once this issue is resolved, the video of the 3D character on the main page of the app could be replaced with an actual rendered environment, meaning it would be possible to re-render the 3D character depending on the scores of the user (a low sleep score, for example, could make the 3D character appear tired). We believe that providing this type of feedback to the user would make the user experience much more compelling than it currently is.

Another possible improvement is to enhance the accuracy of our tracking algorithms. While we have already devoted a significant portion of our development time to writing and testing these algorithms, theoretically, they can always be improved. The sleep tracking algorithm would likely be the most in need of improvement, due to the difficulty of analyzing sleep simply with sound and light levels. In this case, a biometric device that could track and record pulse or movement, would contribute tremendously to the accuracy of the collected sleep data.

8 | References

Azumio Inc. (n.d.). Argus - Pedometer, Run, Cycle achieve your fitness and weight loss goals with the ultimate activity tracker by Azumio on the App Store on iTunes. Retrieved from <https://itunes.apple.com/us/app/argus-pedometer-run-cycle/id624329444?mt=8>

Azumio. (n.d.). Sleep Time - Alarm Clock. Retrieved from https://play.google.com/store/apps/details?id=com.azumio.android.sleeptime&feature=search_result#?t=W251bGwsMSwxLDEslmNvbS5henVtaW8uYW5kcm9pZC5zbGVlcHRpbWUiXQ

Fitbit Inc. (n.d.). Fitbit. Retrieved from <http://dev.fitbit.com/>

Fragments. (n.d.). Retrieved March 23, 2015, from <http://developer.android.com/guide/components/fragments.html>

Gehring, J. (n.d.). Documentation. Retrieved March 27, 2015, from <http://www.android-graphview.org/documentation>

Google. (n.d.). E. Automation and developer API - Sleep as Android. Retrieved from <https://sites.google.com/site/sleepasandroid/doc/developer-api>

Lopresti, A. L., Hood, S. D., & Drummond, P. D. (2013). A review of lifestyle factors that contribute to important pathways associated with major depression: Diet, sleep, and exercise. *Journal of Affective Disorders*, 148, 12-27

Michaud, C. (n.d.). Pedometer Using Accelerometer Sensor. Retrieved March 26, 2015, from http://nebomusic.net/androidlessons/Pedometer_Project.pdf

PELUSO MAM et al. Physical activity and mental health: the association between exercise and mood. *CLINICS* 60(1): 61-70, 2005.

ProtoGeo. (n.d.). Moves. Retrieved from <https://play.google.com/store/apps/details?id=com.protogeo.moves&hl=en>

Recording Fitness Data. (n.d.). Retrieved March 23, 2015, from <https://developers.google.com/fit/android/record>

Sleepiness: Cognitive and Emotional Effects. (n.d.). Retrieved March 23, 2015, from <http://www.webmd.com/sleep-disorders/excessive-sleepiness-10/emotions-cognitive?page=3>

Sonstroem RJ, Morgan WP. Exercise and self-esteem rationale and model. *Med Sci Sports Exerc* 1989;21:329-37.

ViewPager. (n.d.). Retrieved March 23, 2015, from <http://developer.android.com/reference/android/support/v4/view/ViewPager.html>

Working with the Fitness History. (n.d.). Retrieved March 23, 2015, from <https://developers.google.com/fit/android/history>

Build software better, together. (n.d.). Retrieved December 15, 2014, from <https://github.com/>

Tudor-Locke, C., & Bassett Jr, D. (n.d.). How Many Steps/Day Are Enough? Retrieved November 15, 2014, from [http://www.health.utah.edu/peak/docs/Tudor Locke Paper.pdf](http://www.health.utah.edu/peak/docs/Tudor%20Locke%20Paper.pdf)

National Sleep Foundation Recommends New Sleep Times. (n.d.). Retrieved February 27, 2015, from <http://sleepfoundation.org/media-center/press-release/national-sleep-foundation-recommends-new-sleep-times>

Appendix A: Pre-Application Distribution Survey

1. Age *
2. Weight (in lbs) *
3. Height (in feet and inches) *
4. Gender *
 - Male
 - Female
 - Other:
5. Race/Ethnicity *
 - American Indian or Alaska Native
 - Asian
 - Black (or African American)
 - Hispanic or Latino
 - Native Hawaiian or Other Pacific Islander
 - White (or Caucasian)
 - Other
6. How many smartphones do you own? *
 - 0
 - 1
 - 2
 - 3
 - Other:
7. What type(s) of smartphone do you own? *

If you are unsure, please put the device name(s) in the "Other" text field.

 - iPhone & iPad & iPod Touch (using iOS)
 - Android phone (using Android OS)
 - Blackberry phone (using BlackBerry OS)
 - Windows phone (using Windows Phone OS 7 or higher)
 - Symbian phone (using Symbian OS)
 - Palm phone (using webOS)
 - Other:
8. How many apps have you installed on each of your mobile devices (smartphone, tablet, etc) on average? *

Please do not include games.

 - 0 - 10
 - 10 - 20
 - 20 - 50
 - 50 - 100
 - Other:
9. Have you ever used any health, fitness, or wellness apps on a mobile device? *

(For example: Moves, Argus, Sleep Time, etc.)

 - Yes

- No

10. Please list the health, fitness, and wellness apps you have used. *

(For example: Moves, Argus, Sleep Time, etc.)

11. On a scale of 1 -10, how healthy do you think you are? *

"1" is very unhealthy, and "10" is very healthy.

1 2 3 4 5 6 7 8 9 10

12. What exercises do you currently do? *

Select all that apply.

- Jogging and/or Running
- Bicycling
- Swimming
- Basketball
- Football
- Soccer
- Volleyball
- Badminton
- Bowling
- Golf
- Dancing
- Hiking
- Other:

13. At which time(s) of the day do you typically exercise?

Select all that apply.

- Morning (05:00 - 12:00)
- Noon (12:00 - 13:00)
- Afternoon (13:00 - 18:00)
- Evening (18:00 - 20:30)
- Night (20:30 - 24:00)
- Other:

14. On a scale of 1 -10, how healthy do you think your sleep habits are? *

"1" is very unhealthy, and "10" is very healthy.

1 2 3 4 5 6 7 8 9 10

15. On average, how many hours of sleep do you get each night?

- 0-2
- 2-4
- 4-6
- 8-10
- 10+

16. Do you typically try to sleep in silence or with ambient noise?

- Silence

- With ambient noise
17. Do you typically try to sleep in darkness or with ambient light?
- Darkness
 - With ambient light
18. On a scale of 1 -10, how healthy do you think your mood is? *
- "1" is very unhealthy, and "10" is very healthy.
- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
19. On a scale of 1 -10, how much does your mood affect your behavior?
- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
20. Do you see yourself as more of an optimist or pessimist?
- Optimist
 - Pessimist
 - Both/Neither
21. Do you think mobile apps can help you better manage your physical activity?
- Yes
 - No
22. Would you use a mobile app that tracks your sleep, mood, and physical activity?
- Yes
 - No

Appendix B: Post-Application Distribution Survey

Please complete the following four questions immediately after opening the app for the first time.

Open up the application titled SAM Fitness. You should see a screen with 4 bubbles titled "Total Score", "Hours", "Steps", and "Happiness". Tapping each of the "Hours", "Steps", and "Happiness" bubbles should bring you to a different tab. Below these bubbles should be a humanoid character that walks across the screen and waves at you. On the bottom left of this screen should be a button that says "Help", and clicking it should cause a screen to pop-up with instructions on how to use the application. On the bottom right of the screen should be a button that says "Achievements", and clicking it should cause a screen to pop-up with a list of possible achievements you can earn, and all achievements should initially be in red. Please check below all that apply:

- 4 bubbles appear as described above
- Tapping on each of the 4 bubbles takes me to a different tab
- There is an animated character that walks across the screen and waves at me
- There is a help button that functions as described above
- There is an achievement button that functions as described above

Navigate to the "Sleep" tab. An empty graph should initially appear at the top of the screen, and below it there should be 2 buttons: 1 that says "Calibrate Sensors", and another that allows you to Start/Stop sleep tracking. Below that should be two text fields: "Hours Slept Today", and "Today's Efficiency". At the bottom right of the screen should be a button that says "Today's Sleep Stats" that, when tapped should cause a screen to pop-up providing you with statistics regarding how well you slept. Please check below all that apply:

- An empty graph appears at the top of the screen
- All the buttons are displayed as described above
- A pop-up appears when you tap "Today's Sleep Stats" as described above

Navigate to the "Activity" tab. An empty graph should appear at the top of the screen. Below this, a text field should appear that says "Today's Steps: 0". Please check below all that apply:

- An empty graph appears at the top of the screen
- A text field appears as described above

Navigate to the "Mood" tab. At the top of the screen, there should be an empty graph. Below this should be a button that says "Start Mood Tracking". Underneath this button should be a text field that says "Remind me to set my mood every __ hours", with a selection box that allows you to select a number of hours. Below this should be a set of 5 stars that, when tapped, fills with blue up to the location of the tap. Finally, toward the bottom of the screen should be a button that says "Set current mood", that when tapped causes a pop-up asking whether you would like to set the current mood. Please check below all that apply:

- There is an empty graph at the top of the screen
- There is a "Start Mood Tracking" button
- There is a set of stars with functionality as described above
- There is a "Set Current Mood" button with functionality as described above

Please read the information described when clicking the "Help" button on the main screen to understand how the app is used. After using the app for several days, please answer the questions below.

Navigate to the main tab and check all that apply below:

- The "Hours" bubble has updated with a score other than 0.0
- The "Steps" bubble has updated with a score other than 0
- The "Happiness" bubble has updated with a score other than 0
- The "Overall" bubble has updated with a score other than 0

Please describe below any issues and/or bugs you had while using the Main tab of the app

Initiate sleep tracking and check all that apply below:

- The sleep tracking algorithm correctly, or closely, recorded the number of hours I slept on a nightly basis
- The "Sleep Stats" screen correctly calculated statistics related to my sleep behavior
- The graph updated with the correct number of hours slept the night before

Please describe below any issues and/or bugs you had while using the sleep tracking portion of the app

Please walk around with the phone in your pocket for several days and check all that apply below regarding the "Activity" portion of the app.

- The app consistently tracked the number of steps that I was taking on a daily basis
- The graph consistently updated with this number
- The text field titled "Today's Steps" consistently updated

Please describe below any issues and/or bugs you had while using the activity tracking portion of the app

Please consistently update your mood as described in the "Help" portion of the app and check all that apply below.

- The app prompted me to update my mood at time intervals specified by the text field labeled "Remind me to set my mood ever ___ hours"
- The graph was updated when I updated my mood

Please describe below any issues and/or bugs you had while using the mood tracking portion of the app

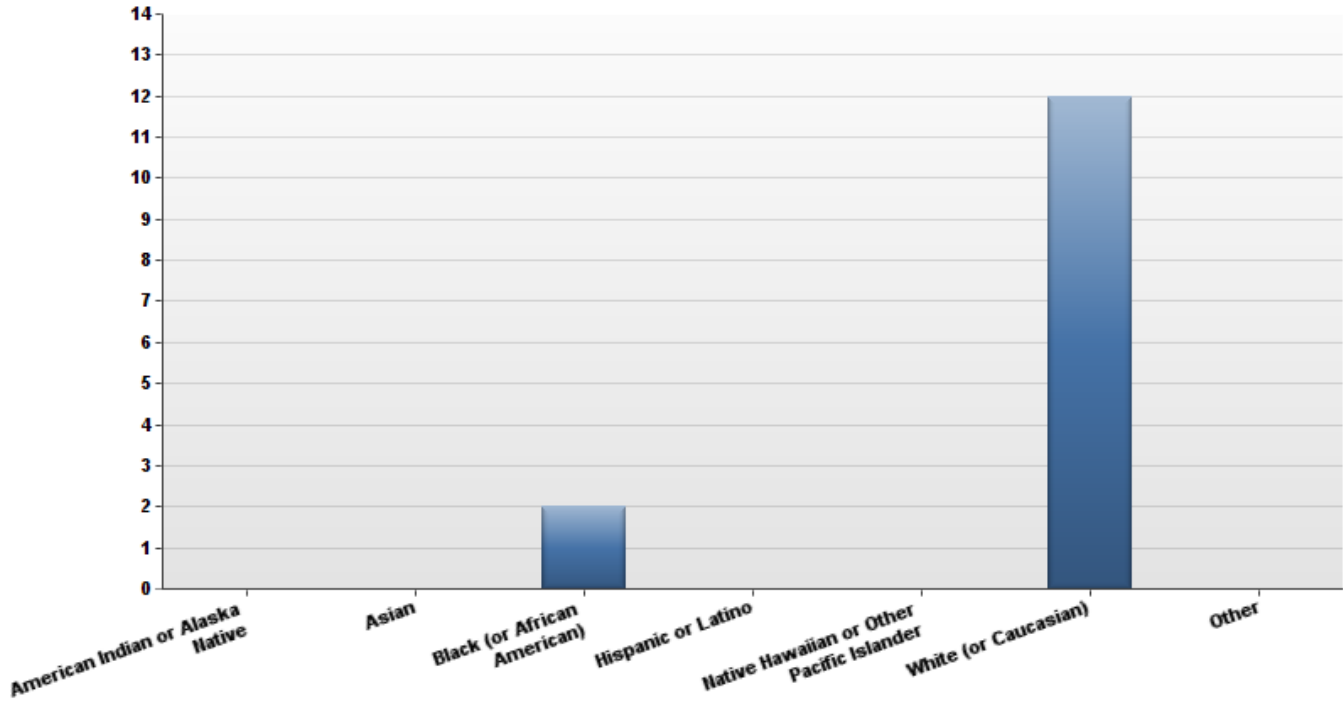
Please describe any general issues/bugs you found while using the app. In addition, please list any improvements (if any) you would like to see to the functionality and/or aesthetics of the app.

Appendix C: Pre-Testing Survey Results

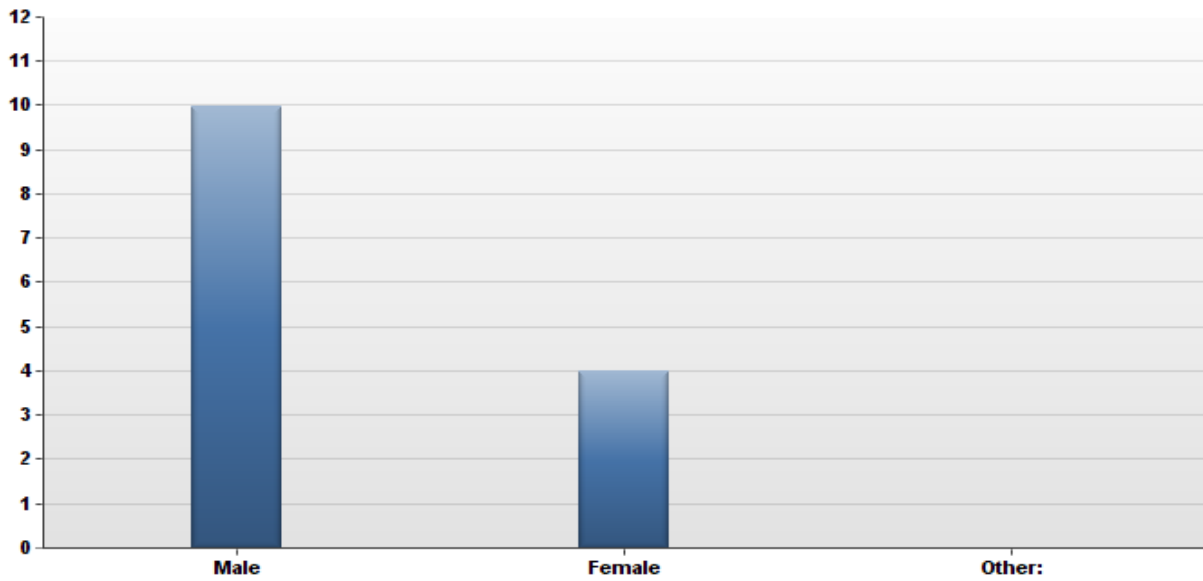
Test Subject General Demographics

Height	Age	Weight (lbs)
5' 10"	18	178
6' 2"	19	186
6' 2"	20	150
6' 7"	21	260
6' 1"	21	180
6' 0"	21	165
5' 5"	21	155
6' 2"	22	280
5' 6"	22	220
5' 5"	22	142
6' 0"	22	215
6' 1"	22	210
5' 9"	22	170
5' 5"	43	125

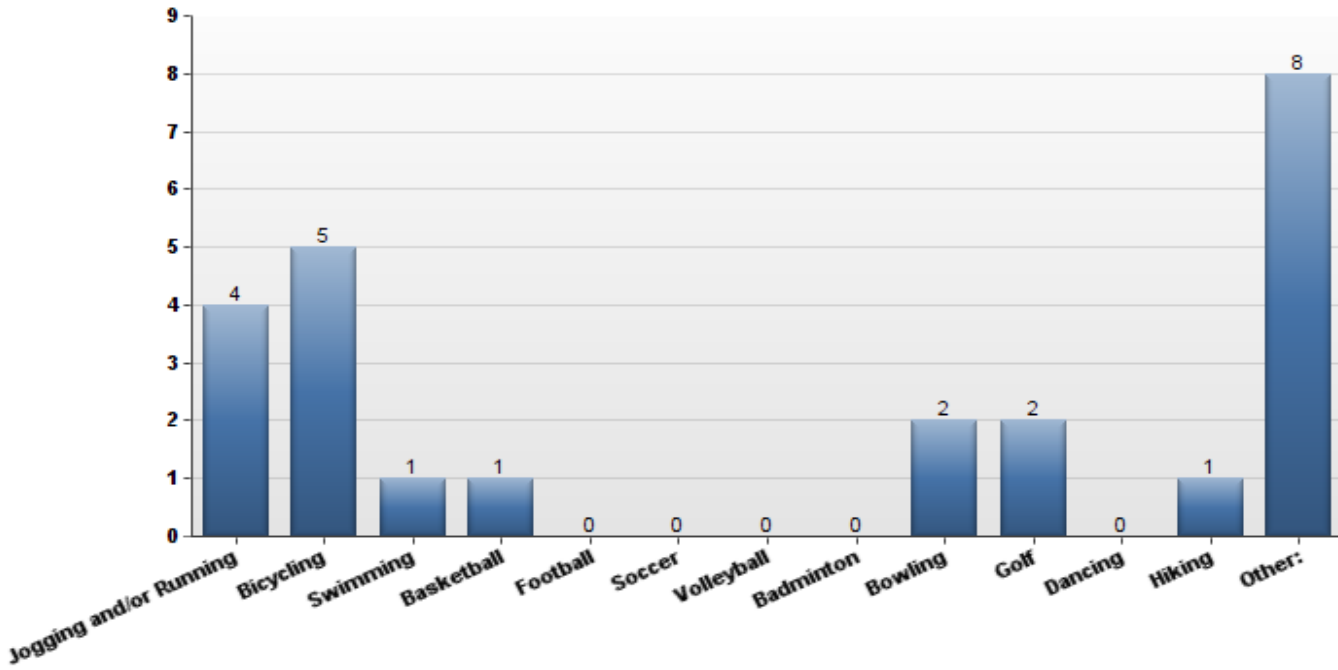
Race/Ethnicity Distribution



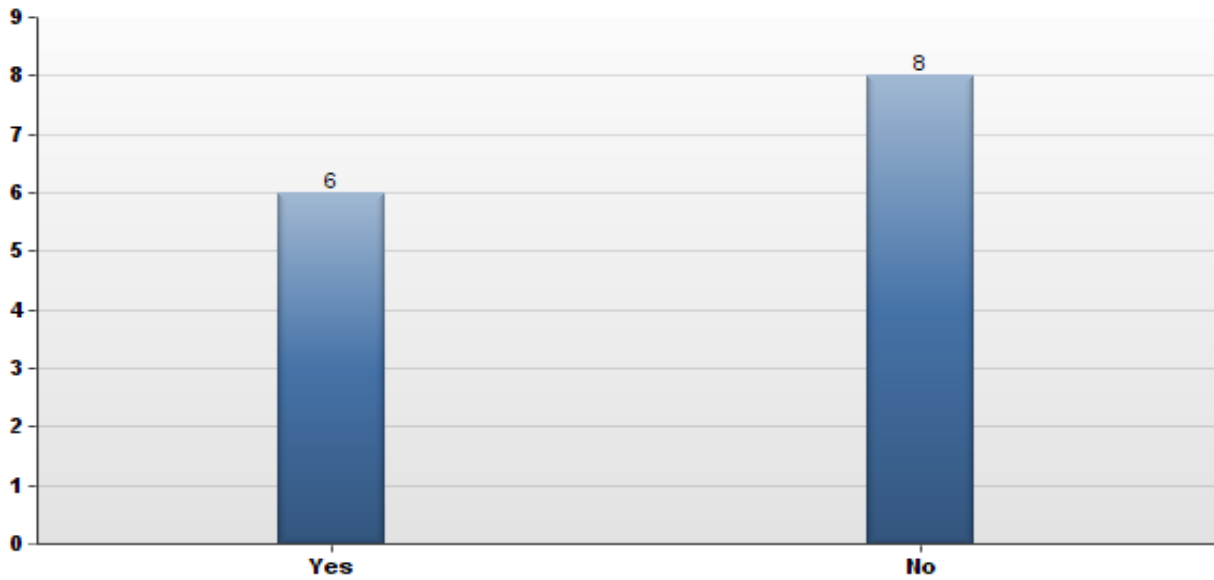
Gender Distribution



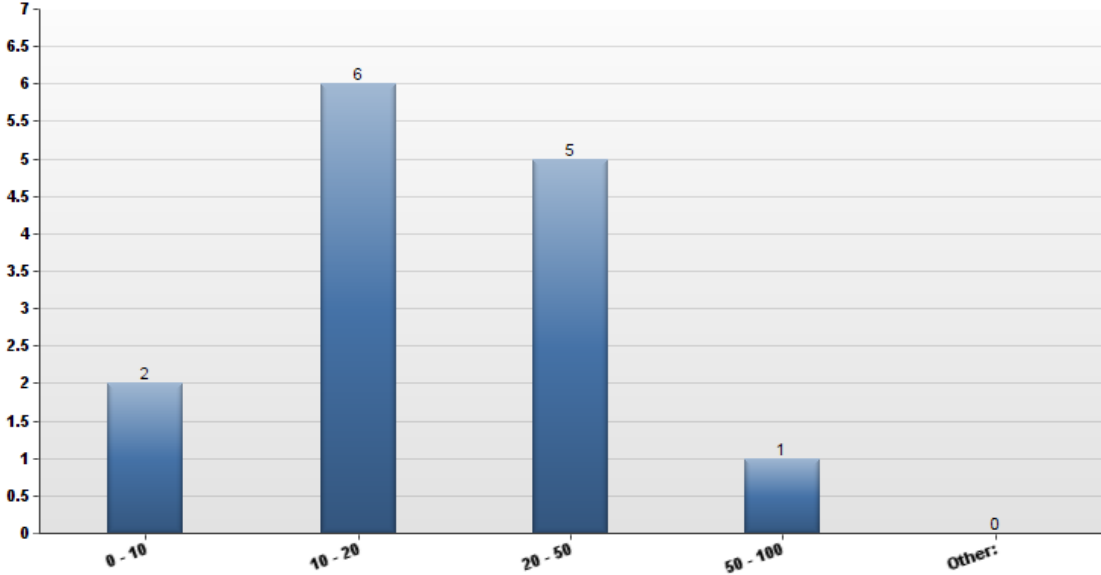
What exercises do you currently do?



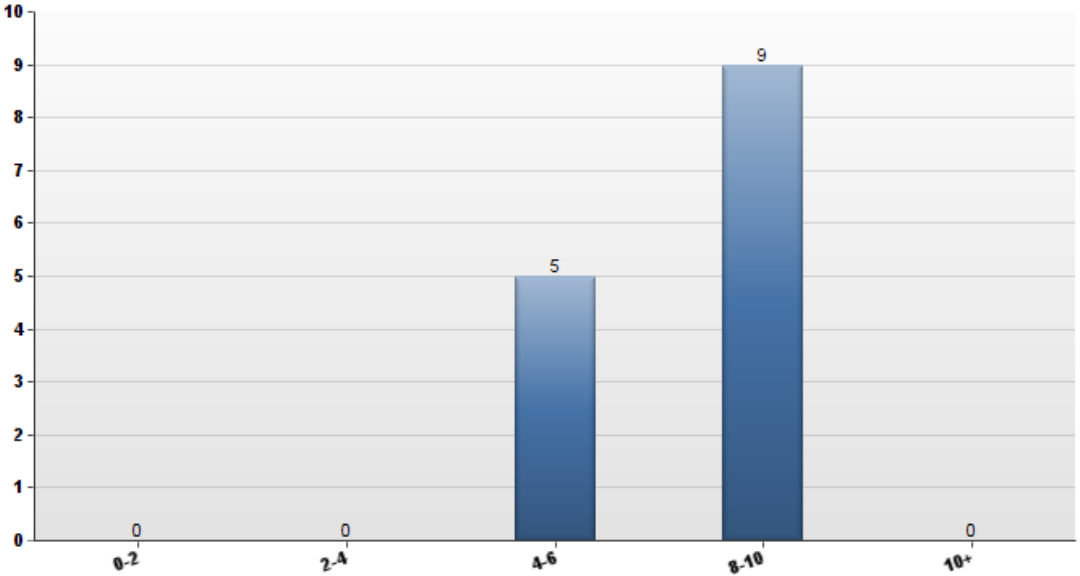
Have You Used Wellness Apps Previously?



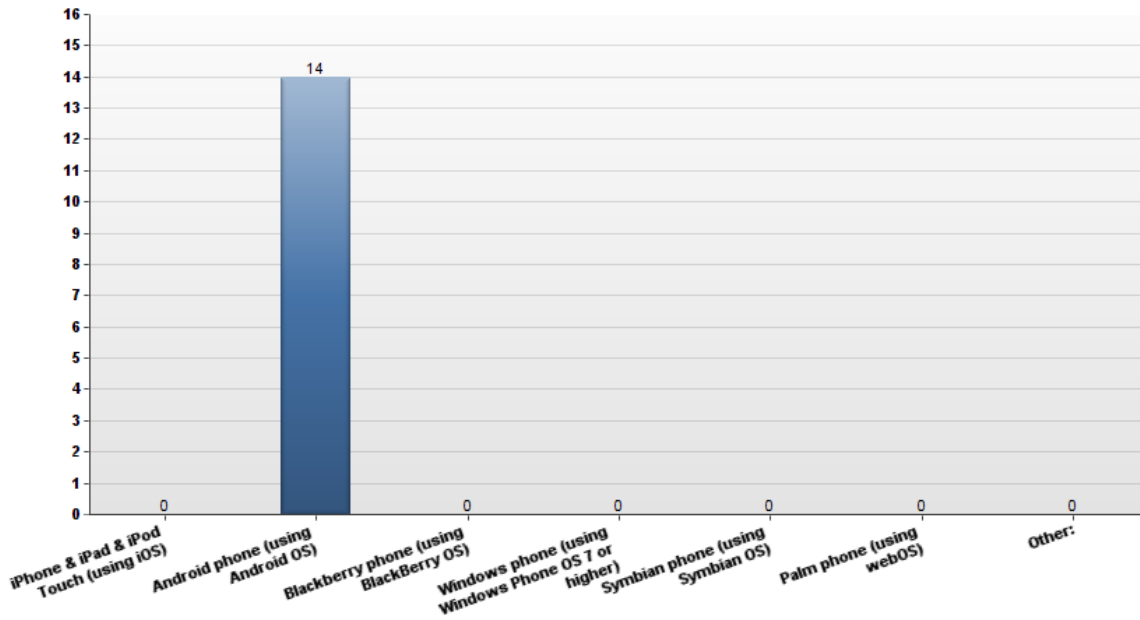
How many apps have you installed on each of your mobile devices (smartphone, tablet, etc) on average?



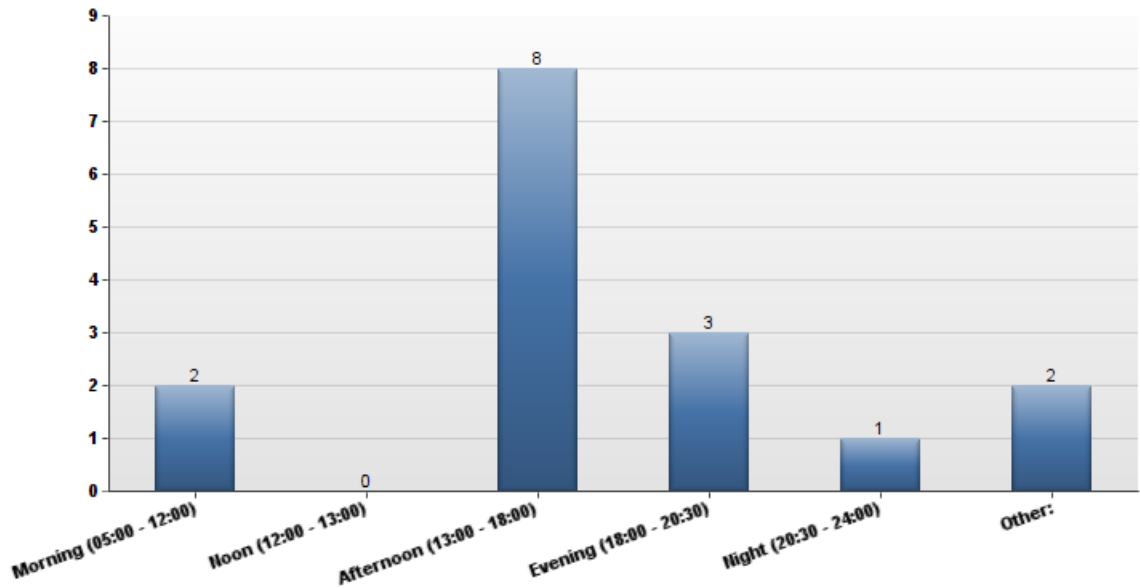
On average, how many hours of sleep per night do you get?



What type(s) of smartphones do you own?

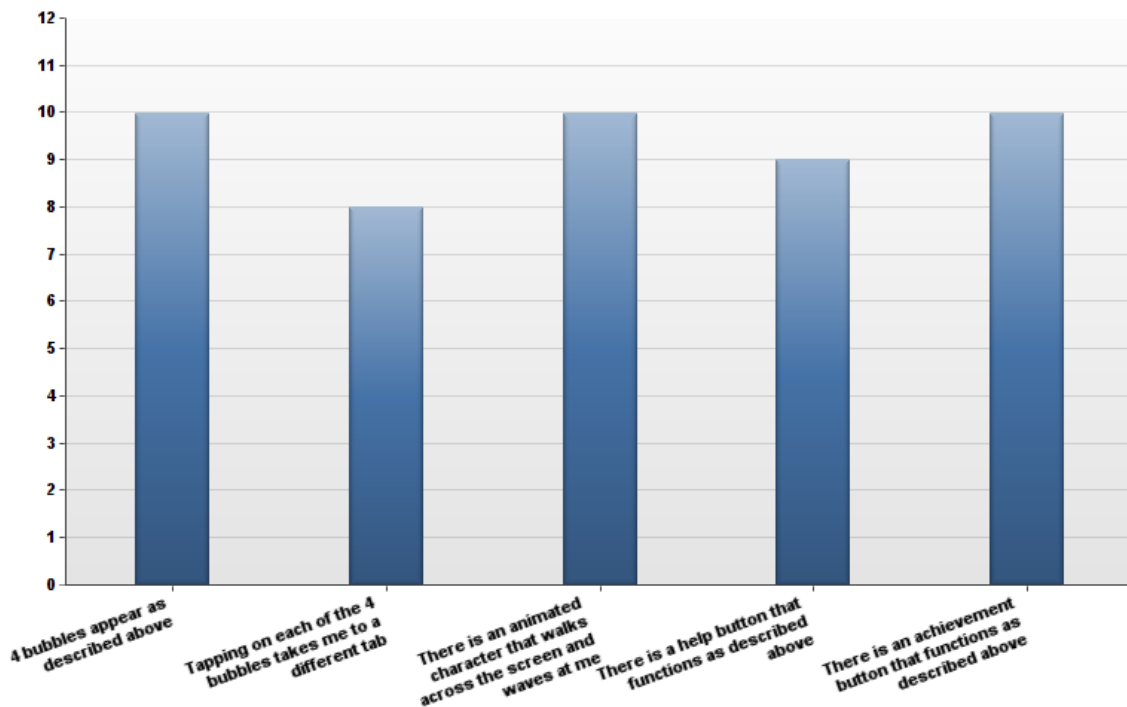


When do you normally exercise?

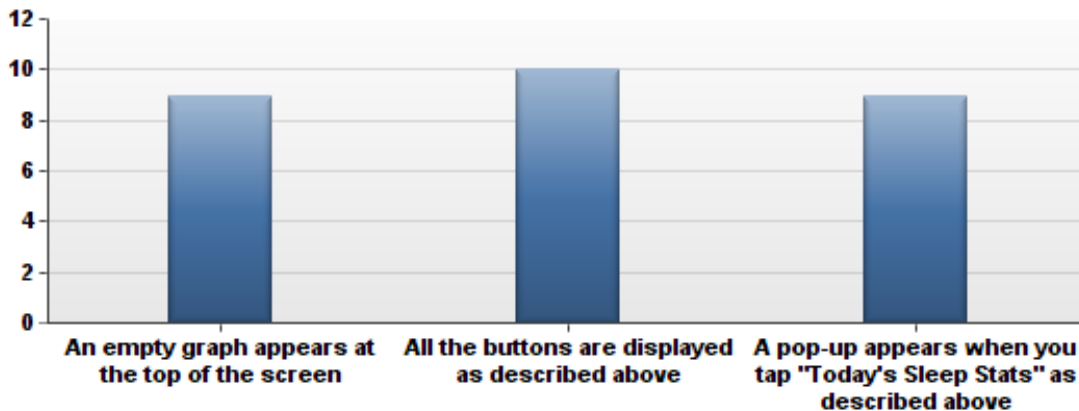


Appendix D: Post-Testing Survey Results

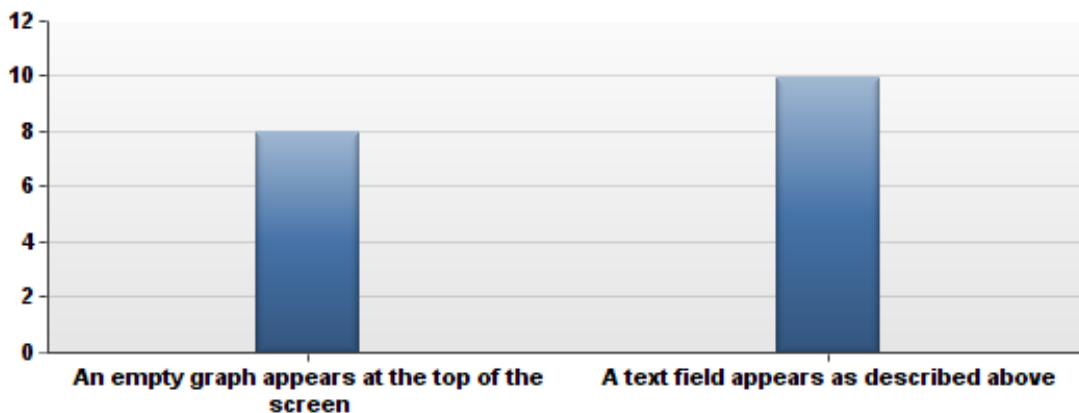
Open up the application titled SAM Fitness. You should see a screen with 4 bubbles titled "Total Score", "Hours", "Steps", and "Happiness". Tapping each of the "Hours", "Steps", and "Happiness" bubbles should bring you to a different tab. Below these bubbles should be a humanoid character that walks across the screen and waves at you. On the bottom left of this screen should be a button that says "Help", and clicking it should cause a screen to pop-up with instructions on how to use the application. On the bottom right of the screen should be a button that says "Achievements", and clicking it should cause a screen to pop-up with a list of possible achievements you can earn, and all achievements should initially be in red. Please check below all that apply:



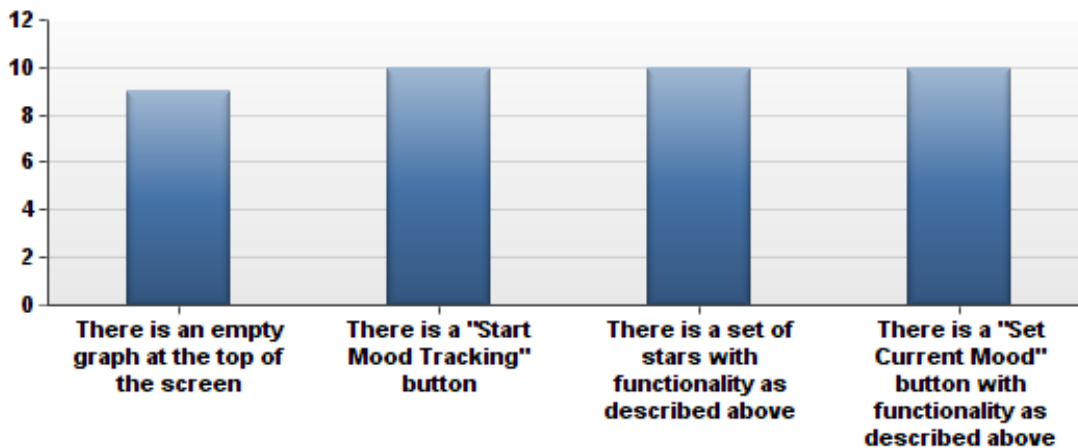
Navigate to the "Sleep" tab. An empty graph should initially appear at the top of the screen, and below it there should be 2 buttons: 1 that says "Calibrate Sensors", and another that allows you to Start/Stop sleep tracking. Below that should be two text fields: "Hours Slept Today", and "Today's Efficiency". At the bottom right of the screen should be a button that says "Today's Sleep Stats" that, when tapped should cause a screen to pop-up providing you with statistics regarding how well you slept. Please check below all that apply:



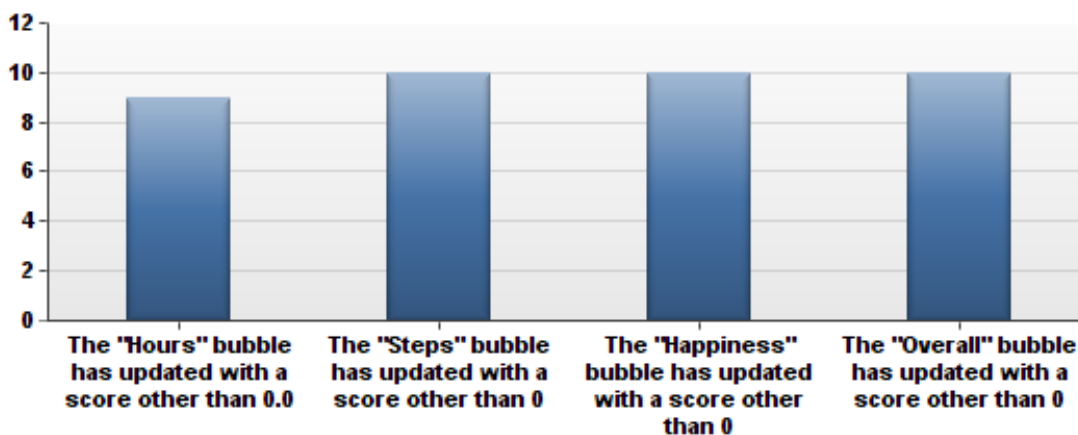
Navigate to the "Activity" tab. An empty graph should appear at the top of the screen. Below this, a text field should appear that says "Today's Steps: 0". Please check below all that apply:



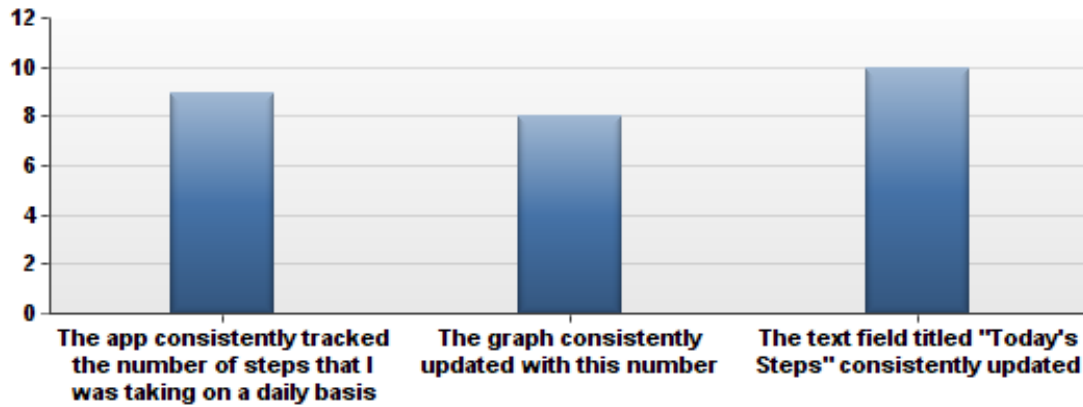
Navigate to the "Mood" tab. At the top of the screen, there should be an empty graph. Below this should be a button that says "Start Mood Tracking". Underneath this button should be a text field that says "Remind me to set my mood every ___ hours", with a selection box that allows you to select a number of hours. Below this should be a set of 5 stars that, when tapped, fills with blue up to the location of the tap. Finally, toward the bottom of the screen should be a button that says "Set current mood", that when tapped causes a pop-up asking whether you would like to set the current mood. Please check below all that apply:



Navigate to the main tab and check all that apply below:



Please walk around with the phone in your pocket for several days and check all that apply below regarding the "Activity" portion of the app.



Please consistently update your mood as described in the "Help" portion of the app and check all that apply below.

