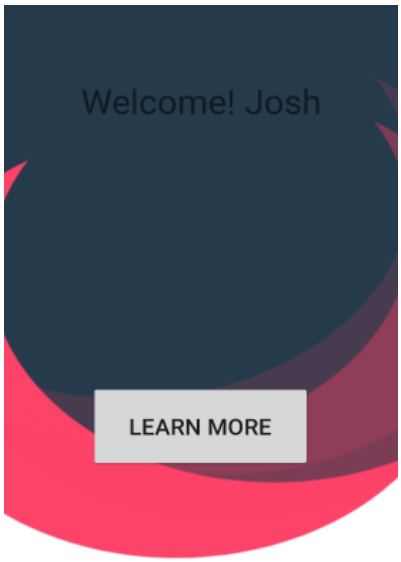
Project Nomophobia

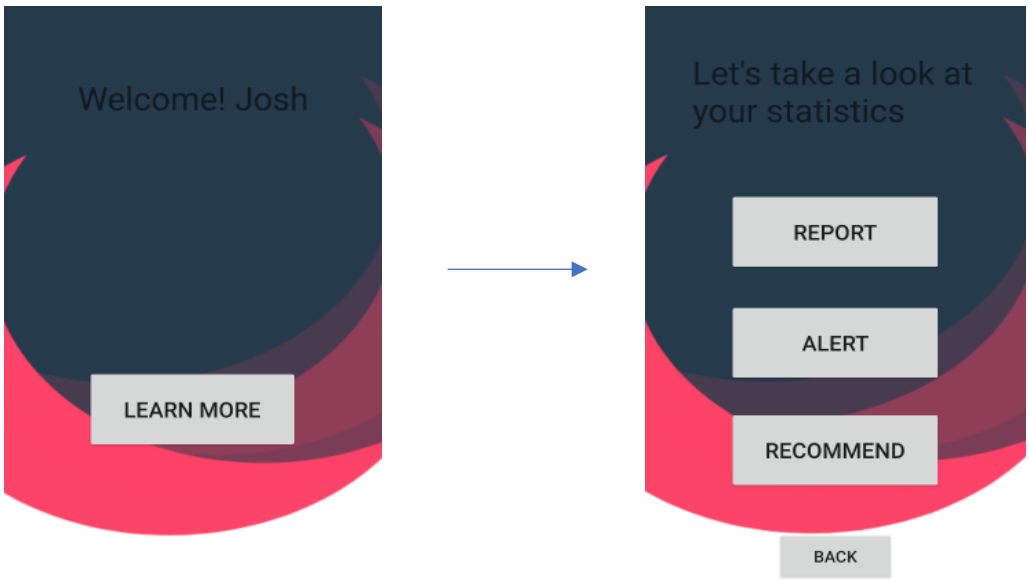


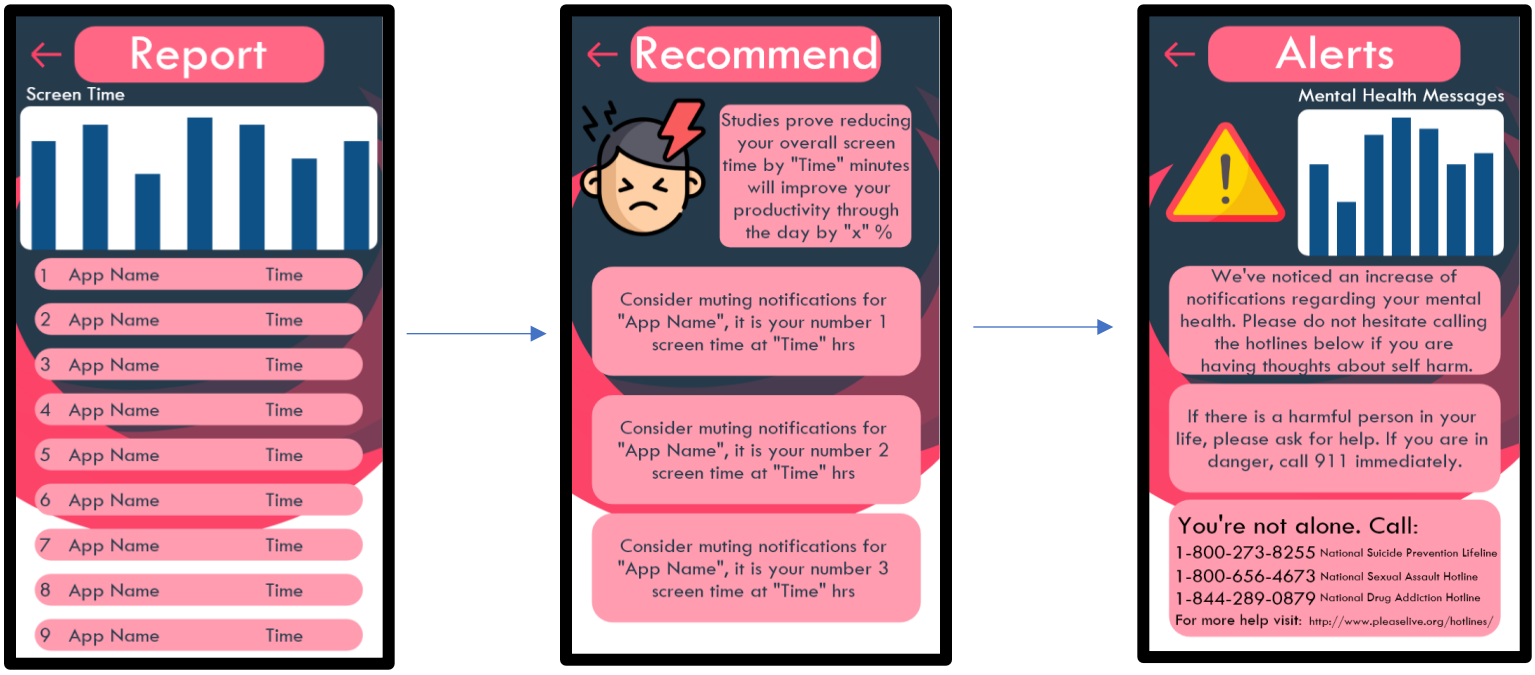
Goal of Project:

Nomophobia is the irrational fear of being without your phone. While most people would deny this fear, statistics show otherwise. We spend an average of 2 hours 15 minutes a day on smart phones. A whopping 75% of smart phone users bring their phones to the bathroom, which makes us ask ourselves, are we too addicted to our cellular devices? There is not one answer for all of us, but this project will help individuals find out their own personal answer.

This application will monitor smart phone usage and analyze trends in different things you do on your phone. Whether that be watching cat videos or texting with your friends, the app will analyze your usage and build reports that can then be used to recommend changes to your usage patterns and alert you when harmful actions are detected such as anxiety, depression, bullying and self-harm. With the help of this application, smart phone users will be able to make the most of their day and improve their social lives.

UI Pathway:


Welcome! Josh
LEARN MORE
Let's take a look at your statistics
REPORT
ALERT
RECOMMEND
BACK


Report
Screen Time
1 App Name Time
2 App Name Time
3 App Name Time
4 App Name Time
5 App Name Time
6 App Name Time
7 App Name Time
8 App Name Time
9 App Name Time
Recommend
Studies prove reducing your overall screen time by "Time" minutes will improve your productivity through the day by "x" %
Consider muting notifications for "App Name", it is your number 1 screen time at "Time" hrs
Consider muting notifications for "App Name", it is your number 2 screen time at "Time" hrs
Consider muting notifications for "App Name", it is your number 3 screen time at "Time" hrs
Alerts
Mental Health Messages
We've noticed an increase of notifications regarding your mental health. Please do not hesitate calling the hotlines below if you are having thoughts about self harm.
If there is a harmful person in your life, please ask for help. If you are in danger, call 911 immediately.
You're not alone. Call:
1-800-273-8255 National Suicide Prevention Lifeline
1-800-656-4673 National Sexual Assault Hotline
1-844-289-0879 National Drug Addiction Hotline
For more help visit: http://www.pleaselive.org/hotlines/

Library of Useful Methods

1. void onSaveInstanceState(Bundle outstate)

This method was one of the first methods that we attempted to use to store data after the closing of an app. Through the test-driven process however we realized that this method only functions on if the app is closed and doesn't retain the specific data inputted but rather the UI state. However, this may prove to be useful for graphical elements that we choose to implement in the future. The parameter provided is the state of the UI at the time the user moves away from actively using the app.

2. void onPause()

onPause() was one of the key methods used in our app to store the user's date. In addition to the use of SharedPreferences, onPause() like onSaveInstanceState(), represents a part of the activity cycle in android studio. onPause() is the method that studio calls right before the app is terminated, and as a result, and attempt to store data or UI states can also be made in onPause(), since it's the final call before termination.

3. Shared Preferences/getSharedPreferences()

This object and method allows the users to store their data dynamically and locally on their phone in what are known as "preferences". For any future development, this will be crucial to implement, as it is the only reliable way at the currents state for us to store data with a "key", and then extract it with that same key.

Features/Classes:

*Note: Following methods are how they would look like in finished product. The logic is present in app, perhaps not in the same structure.

1. Report

a. Screen Time and Phone Pickup Detector (2)

- long calculateTime()

This method uses Android Studio's internal clock to compare the current time (determined by a button click) to the time that the application was launched to determine the amount of time the user has spent using their phone. With a set procedure, the same

system can be used to determine how often the user uses each individual app. The method returns the long value that represents the amount of time the user has spent on the app/phone.

Due to the UI implementation of the method, the User is able to visually determine the amount of phone pickups that they had over the course of the day.

b. Notification Detector

- getNotifications()

Accesses a pre-determined list of social media apps the user is using and determines the number of notifications received for each app, returning a list stored with the app and the number of notifications for each app when called.

2. Recommend

a. Addiction Detector (This method is what we wanted to achieve, but is modified for now)

- ArrayList calculateRisk (ArrayList notifInfo, ArrayList timeInfo)

This method would take both the notification and screen time information that the methods in report returned and then using a set of criteria, weighing the importance of notifications and time appropriately, assigns each app a risk identifier, and returns an arraylist for the same.

Future Plans:

1. Alert

We plan to incorporate Alert in the future but were unable to due to time and legal restrictions.

a. Addiction Detector

-scanNotification()

Accesses notification received into the user's phone from a list of pre-determined social media apps. The method will return a list of strings parsed by spaces in the notification.

b. Cross-checks words

- detectKeyword()

Using the returned list from scanNotification(), this method will compare the words from the notification against a pre-made array list of common mental health indications. It will return the number of similar words as well as categorize the types of mental issue each word is indicating to.

2. Graphical Representation

In the future, we plan to display our data for both the Report and Alert pages using graphs. Report would show how much time the user has spent overall on their phone throughout that week as well as show graphs for individual apps. Alert would include graphs to display the number of mental health related messages scanned throughout the week. Graphs require several technical skills which we were not able to adapt to given our time constraint.