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.

Intended Audience: Our app is targeted primarily towards young adults, but is not limited to seniors, younger children, or middle-aged adults. Today, almost everyone spends time in front of a mobile device. Teenagers and people in their 20s and 30s use smart phones very often and receiving help in making healthy habits regarding screen time will help them in the future.

Resources

1. Android Studio
    a. Android Studio is the official program for developing applications for Android OS. This allows for easy integration and native support for lower level accessibility into the OS, which is what we need to catch notifications from system wide apps.
2. Microsoft SQL Server 2019
    a. This easy to use and intuitive at-home solution for creating a SQL server to run on a Linux or Docker containerized node. This way, we allow users to access their information anywhere in the world, without having to use their phone.
3. Docker
    a. Containerizing the SQL server inside a Docker container is the best way to run the server efficiently and independent of other programs running on the server machine. It also allows for easy and secure storage of data.

Main Features

There are 3 main features/processes that our app will have to accomplish.

1. Report
    a. This will be the primary and backbone feature that the other 2 will pivot from. With report, the application will need to extract data from the phone itself, reading values such as notifications, pickups, etc. It will need to extract and store this data to be able to depict these trends correctly and aesthetically in graphs to the user. It needs to *report* the user's current condition of nomophobia, depicting the extent of dependence that the user has with their phone. Due to the amount of times that this feature will likely called by the user, the most, simple yet appealing UI needs to be implemented in this section (simple perhaps with a button click but revealing a lot of digestible information).
2. Recommend
    a. With recommend, after the user has generated a report of their phone use, at the bottom, there will be recommendations to decrease usage if the user wishes to see it. Recommend will need to gather data from both report and alert to gauge what apps need to be adjusted. Adjustments could take the form of limiting the number of notifications per hour that an app is allowed to show, or even just negate an app from being able to send notifications. With the user's consent, it could also take more direct action like setting a time limit for usage on the app.

3. Alert
    a. Alert is a feature that focuses not on the user's raw usage of their phone, but rather how they are using the phone. Alert gauges the user's interactions by monitoring harmful keywords in common entertainment or social media apps such as Snapchat, WhatsApp, and Instagram. Potentially, alert could even work to analyze specific chains of conversation to detect malicious intent that could cause the user's mood/mental health to deteriorate.

Test Cases

For Processes

Test Cases for Report:

- Returns correct number or time usage of the app (ensuring that this is separate from the overall screen time usage)
- Only returns apps which have time (does not return apps with 0 --> unnecessary information)
- Check to ensure data is stored --> will be needed to calculate trends (the data is not being overwritten with each use of report but rather added onto for that day)
- Makes sure to account for background processes, and doesn't include apps running in the background into their overall time usage (Ex: If user leaves Instagram open for an hour, but isn't actively using it, then that statistic is irrelevant).

Test Cases for Recommend:

- Ensure that the changes to app notifications being displayed are for the correct app that it got its data from (ex: is not suggesting lowering Facebook when given Instagram's data).
- Since recommend will be taking data from Alert and Report to make its suggestions, the algorithm that determines what is more important will need to be continually tweaked. For example, spending what is more harmful to the user (spending 1 hour on Instagram daily or 30 minutes on Snap every 2 days, if while using Snap the user has a higher "Alert" score [bad mental state keywords].
- Does not recommend for apps that were not mentioned by either report of alert.
- Can be filtered by day or week (so as to prevent the same recommendation to be provided by each use).

Test Cases for Alert:

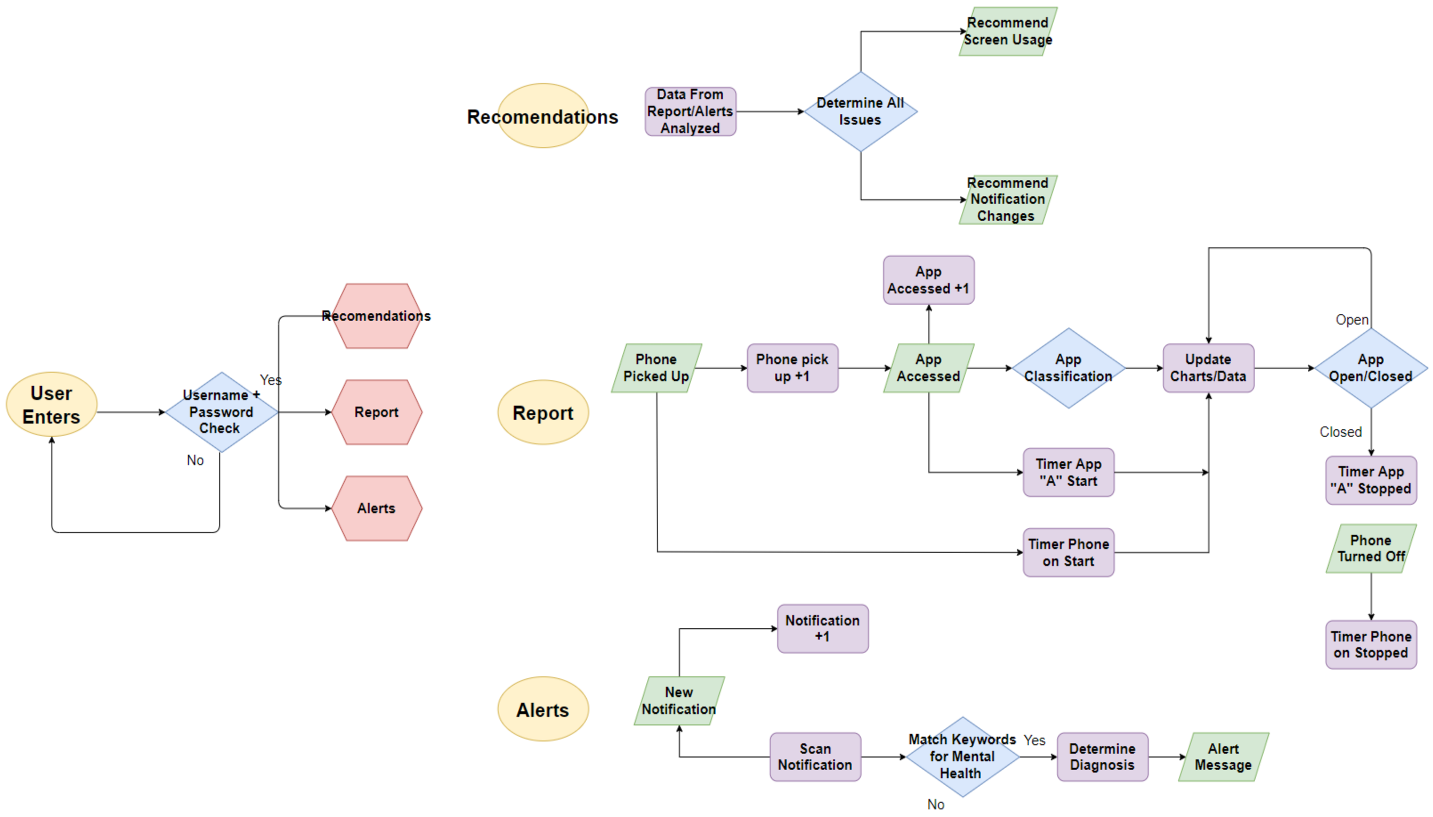
*Note with Alert, due to its word-specific nature, getting the most "correct classifications" for malicious thoughts or harmful text that can directly trouble the user is necessary. The app needs to catch as many cases as possible.

- Method accurately checks the word itself [all caps, common misspells, asterisks (attempted censorship), swear words, or common slurs]
- Method understands normal text (the negative case)
- Accurately Keep track (stores) which apps are most impactful/harmful --> if possible then check talking to one user constantly (bullying), and notes for suicidal/depression thoughts

Make sure the Android application:

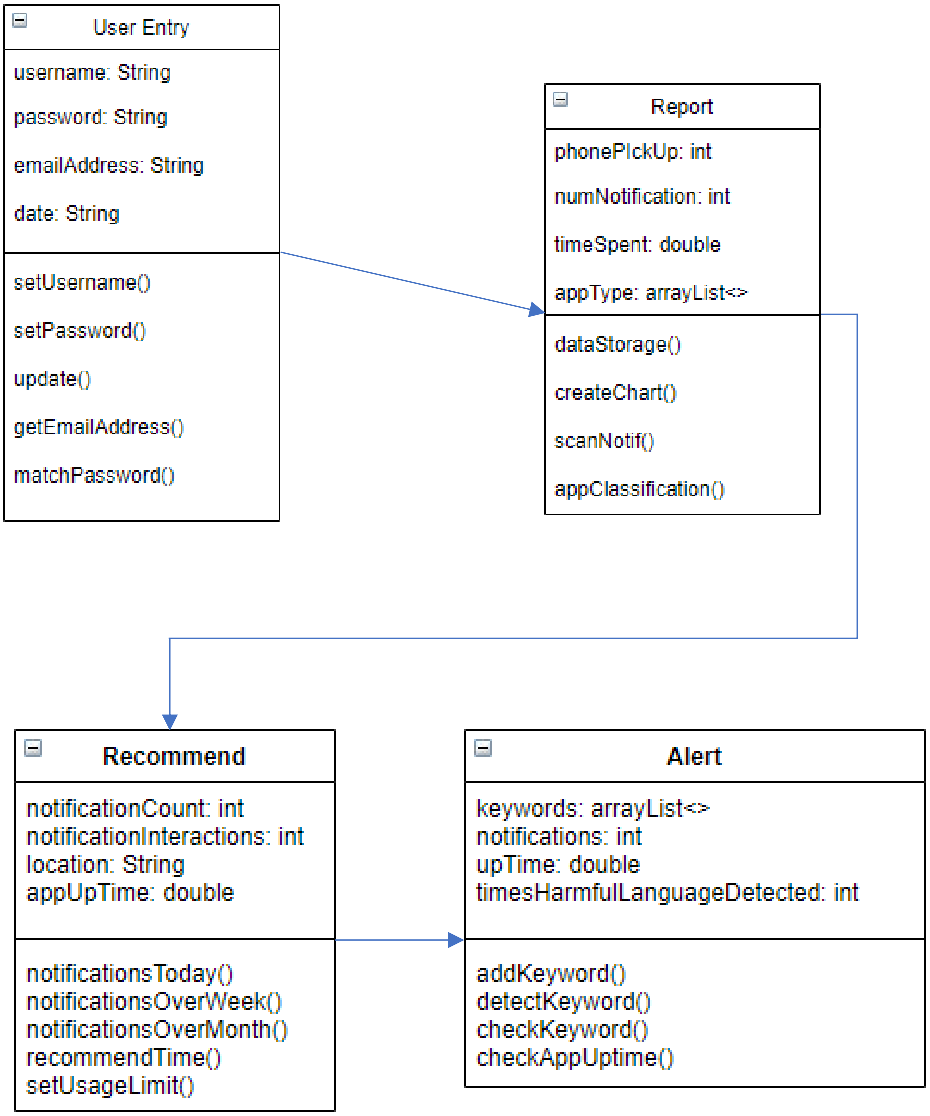
- launches successfully and in a reasonable amount of time
- can scale to multiple screen sizes and aspect ratios
- behaves correctly when a call or text is encountered
- only displays the desired information on each page
- behaves correctly when connected and disconnected from Wi-Fi
- has multilingual support
- uses phone sensors correctly
- obeys all android privacy guidelines
- does not corrupt any data when device is turned off
- supports multitouch

Flowchart



- *Report Clarification: Timer starts when phone turned on and when app is started. Charts/Data is continually updated till phone is tuned off or app is closed.

UML Diagram


User Entry
username: String
password: String
emailAddress: String
date: String
setUsername()
setPassword()
update()
getEmailAddress()
matchPassword()
Report
phonePIckUp: int
numNotification: int
timeSpent: double
appType: arrayList<>
dataStorage()
createChart()
scanNotif()
appClassification()
Recommend
notificationCount: int
notificationInteractions: int
location: String
appUpTime: double
notificationsToday()
notificationsOverWeek()
notificationsOverMonth()
recommendTime()
setUsageLimit()
Alert
keywords: arrayList<>
notifications: int
upTime: double
timesHarmfulLanguageDetected: int
addKeyword()
detectKeyword()
checkKeyword()
checkAppUptime()

Timeline

October 1st – 8th

Finish pseudocode

Begin code development and write test cases

Begin working on UI framework

October 8th – 15th

Finish UI framework and begin working on elements

Work on code development and writing test cases

October 15th – 16th

Finish initial version of code

Finish UI elements

Finish test case writing

October 16th – 23rd

Test code and fix bugs

Polish UI elements as necessary

Write UI code to accompany the program

October 23rd

Finish UI code

Finish UI elements

Finish code testing and have the working program ready

Finish commenting code

October 24th - 25th

Fix minor issues with code and UI (if any)

Test working application extensively for any possible bugs

Resolve any potential problems

October 26th

Program completed

Useful Links

https://en.wikipedia.org/wiki/Nomophobia

https://undepress.net/nomophobia/

https://www.healthline.com/health/anxiety/nomophobia