Overview

Technical Areas

Full-stack development and deployment, data acquisition and analysis, machine learning

- Frontend - React Native
- Backend and Database - Firebase
- Machine Learning Algorithm - Python

Description

This project involves the design of a mobile application that aims to bridge the gap between those suffering from mental health disorders (focusing on depression/anxiety with college students on college campuses first) and their friends, family, and medical professionals. Specifically, this app will contain a monitoring and journaling system for an individual to gauge their own mental wellbeing while bringing awareness to this progress for third-party users associated with the individual. The latter will be done through machine learning, in which the application will be able to pick up information from an individual's interaction with the app, and signal a trusted medical professional to reach out to their patient. In addition, with the existing lack of understanding for how to help those with mental illnesses, this app acts as a platform to connect with and keep their family and friends updated with interactive functionalities such as a community feed, messaging system, meditation, etc.

Current State and Future Goals

We have fully developed the frontend of our app using React Native. However, some of the screens do not have their functionalities fully implemented such as the Daily Check-Up screen, Health Index screen, and the Meditation screen. The ideal functionalities for these screens are the following:

- **Daily Check-Up**: In this screen the user should be able to choose an index (Sad, Neutral, Happy) that represents their mood for the day. This index would then be passed to the database as it would be used in the Machine Learning Algorithm. Moreover, it will contain a button to take a quick survey to further address the user's feeling and another button to redirect the user to the Journal Entry screen.
- **Health Index**: In this screen the user should be able to track their Daily, Weekly and Monthly progress. The screen should display a graph which shows the users mood progression based on the Health Index generated by the Machine Learning Algorithm which analyzes the user's journal entry.

- **Meditation**: In this screen the user will be able to access a variety of helpful exercises to help them relax and feel better. To do this, we planned to communicate with the developers of headspace and see if they have an API that we could integrate with our app to offer our users their services.

On the other hand, the screens that do have its functionalities fully if not mostly implemented are the SignUp/LogIn screens, Journal Entry/Past Entries screens, Community Feed/Post to Feed screens and the Messaging screen. Additional functionalities that can be added to these screens are the following:

- **Sign-Up/Login**: Add other methods of login such as using Google, Facebook, Instagram, etc.
- **Journal Entry/Past Entries**: Allow the user to be able to edit/share their journal entries. Also, if the user allows it, pass the journal entry to the Machine Learning Algorithm in order to generate the Health Index.
- **Community Feed/Post to Feed**: Implement the functionality for the search bar and the like/comment button. Currently the comment button redirects the user to a screen where they can write a comment, but it doesn't post correctly.
- **Messaging**: Implement the ability to create group chats and start live conversations with multiple distinct users. Currently it only works as a one to one conversation between the two users, but they are restricted to talking to one another and no additional chat can be started.

Although we decided to use Firebase as both our backend and database service, it is highly suggested that the group implements their own backend and database integrations as it allows for more flexibility and freedom for the developers. For the backend, we suggest that the group use Node.js as it will be easy to integrate with a frontend based on React as they are both based on javascript. For the database, we suggest that the group use MongoDB as it allows for more flexibility on what can be stored and since the app stores different kinds of data it would be greatly beneficial.

The main component of the project that needs to be worked on is the Machine Learning Algorithm. Due to the time constraint and the lack of data, we were not able to implement the machine learning portion of the project, however, we will explain our idea behind it. For this portion of the project, we planned to do sentiment analysis on the user's journal entry if the user permits it. This will allow us to generate a Health Index that should indicate with high accuracy how the user is feeling on that specific day. This health index will then be part of the user's profile and will allow those in their close community (friends and family) to be aware of the user's mental health. In order to implement this portion, we needed to gather data and finish the implementation of the algorithm. The plan to gather the necessary data was to publish a beta version of our app to the public, where the users would be asked to truthfully write a journal

entry about how their day went and choose the index that matches their mood (sad, neutral, happy), which will help in the training process. For the implementation of the algorithm, we planned to use Recurrent Neural Networks, especially the type of Long Short Term Memory Networks, which are greatly used in Natural Language Understanding.

Things To Look Out For

The main challenge that we faced when developing our app was the constant version updates that our tech stack faced. At the start of the project, we were using the current version of React Native at that time being SDK 40.0 and towards the end, the version came to be SDK 44.0. This change in version hindered our progress a lot as when upgrading from an older version to a newer version made most of the libraries/packages used deprecated. Therefore, after upgrading versions, we spent a lot of time rewriting code to meet the new syntaxes or correct implementations for the current version. Our advice for future developers is to stick with a version that works for them, or to be aware if some of the libraries/packages used will be deprecated in the near future, if so be prepared to quickly implement the new standards.