Practicum Sprint #6
Mental/Physical Illness Chatbot

Tusheet Goli, Tejas Pradeep, Akshay Sathiya, Pranav Khorana, Sanket Manjesh, Rahul Chawla
tgoli3@gatech.edu, tpradeep8@gatech.edu, asathiya6@gatech.edu, pkhorana3@gatech.edu, smanjesh3@gatech.edu, rchawla36@gatech.edu

1 ACCOMPLISHMENTS THIS WEEK

1.1 Akshay Sathiya's Progress

This week, Akshay continued working on the physical illness prediction (PIP) ML models for determining the user's risk of illnesses affecting their physical health, as described in the physical diseases dataset (Patil, 2020).

Akshay wrote the code to evaluate the PIP models using the multi-class F1 score, which is a weighted (by the number of positive/true samples for each class) average of the F1 scores for all classes (scikit-learn developers, 2022, *sklearn.metrics.f1_score*), and K-fold cross-validation (scikit-learn developers, 2022, *sklearn.model_selection.Kfold*). The PIP models are first trained on 80% of the dataset and evaluated on the remaining 20% of the dataset before being evaluated on five (K=5) folds of the dataset. Both PIP models (random forest and neural network) had good accuracy and F1 score across the training set, testing set, and all K-folds.

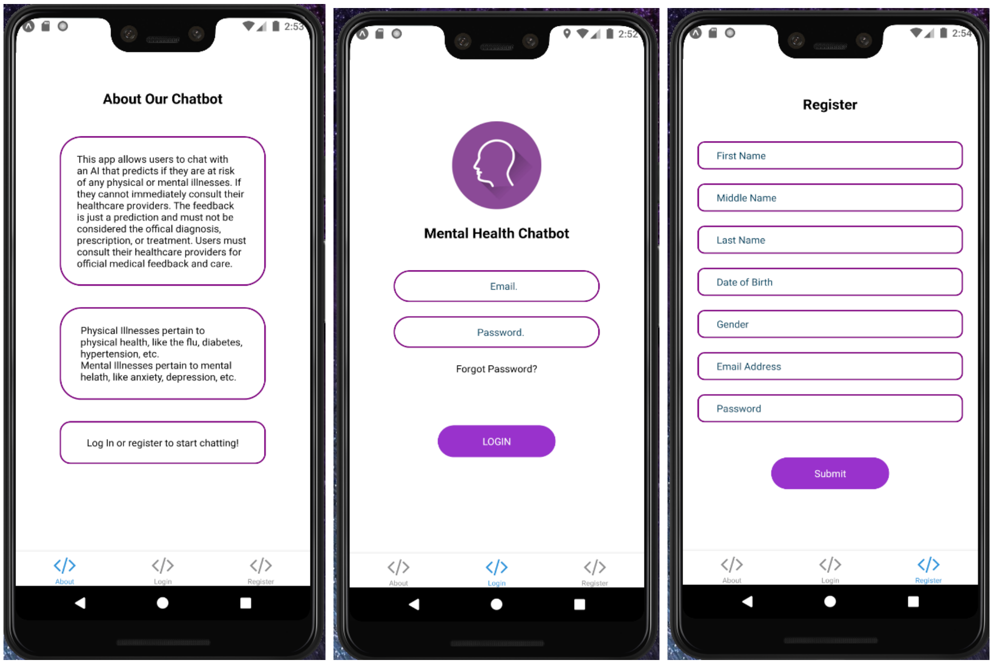
Akshay also wrote code to extract symptom information from user message strings using the Rapid Automatic Keyword Extraction (RAKE) technique (Saxena, 2020) and TF-IDF vectorization (scikit-learn developers, 2022, *sklearn.feature_extraction.text.TfidfVectorizer*), construct feature vectors, and pass those feature vectors to the PIP models for predictions. RAKE was used to extract key phrases of around one to three words from the user message string. Each unique symptom in the physical diseases dataset is converted to a TF-IDF vector and compared with the key phrases, which are also converted to TF-IDF vectors. The difference between symptoms and key phrases is measured as the Euclidean distance between the symptom and key phrase TF-IDF vectors. If the difference is at most 1.1 (a threshold determined via trial and error), then the user is considered to have the symptom. A feature vector of length equal to the number of unique symptoms in the physical diseases dataset is constructed, where each element corresponds to a unique symptom. The value for an element is 0 if the user's message does not indicate that they have the corresponding symptom, 1 if the message does indicate that they do have the corresponding symptom. The feature vector is then passed to the PIP models for predictions of the physical illness(es) the user may be at risk of.

Akshay also refactored code to eliminate redundancies and make the code more concise.

Akshay has committed and pushed his work to the *pip_model* branch in the project's GitHub repository. For the next sprint, Akshay will write code to format the output (class/PIP probabilities) in a way that is easily understandable by the user. Akshay will also further test and update the PIP model code as needed to ensure that it works correctly before it is integrated with the rest of the project.

1.2 Pranav Khorana's Progress

This week, Pranav worked on adding components to each of the UI screens including TextInput, TouchableOpacity, React-Native-Gifted-Chat, StyleSheet, and more. He also applied styling to each of the screens to make it appealing to the user. Afterwards, Pranav pushed all his changes to the Github repository. Later, Pranav plans to work on adding more functionality to the components, the authentication process for login, sending the chats to the database, or establishing a connection between the frontend and the flask server. He included images below on the next page to show what the application screens look like:



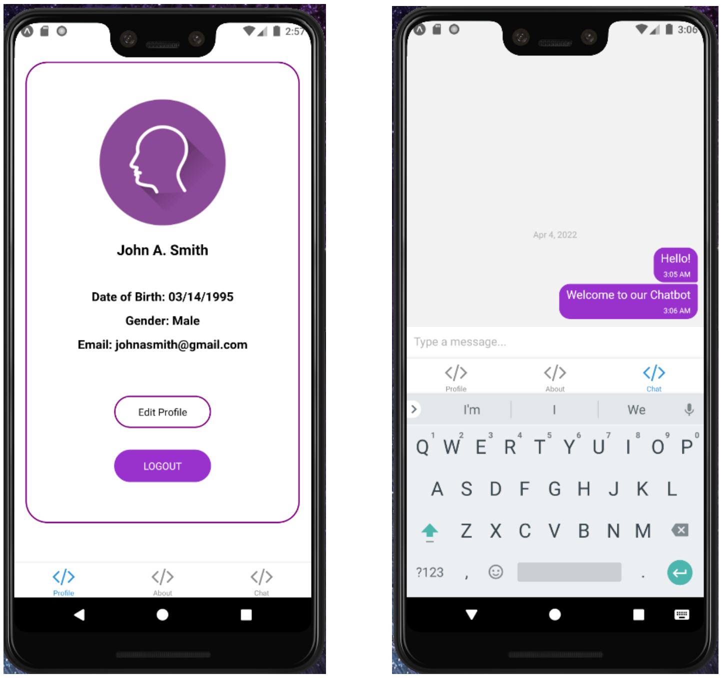
About Our Chatbot

This app allows users to chat with an AI that predicts if they are at risk of any physical or mental illnesses. If they cannot immediately consult their healthcare providers. The feedback is just a prediction and must not be considered the offical diagnosis, prescription, or treatment. Users must consult their healthcare providers for official medical feedback and care.

Physical Illnesses pertain to physical health, like the flu, diabetes, hypertension, etc.
Mental Illnesses pertain to mental helath, like anxiety, depression, etc.

Log In or register to start chatting!

Figure 1: *These are the About Screen (left), Login Screen (middle), and Register Screen (right) of the application*



John A. Smith

Date of Birth: 03/14/1995
Gender: Male
Email: johnasmith@gmail.com

Apr 4, 2022

Hello!
3:05 AM

Welcome to our Chatbot
3:06 AM

Figure 2: *After logging in, these are the Profile Screen (left) and Chat Screen (right) of the application*

1.3 Rahul Chawla's Progress

This week, Rahul continued work with the PostgreSQL database, hosted on Heroku, in order to prepare it for integration with the flask server and other backend components. Currently, after looking into more of the design for the application, Rahul has identified changes that need to be made for the tables in the database and plans to implement those in the following week.

1.4 Tusheet Goli's Progress

This week, Tusheet continued his work with the API design and implementation of the endpoints for the respective services to properly determine how these services are going to interact with each other and use which endpoints. He came up with detailed documentation of the several services and endpoints that are available (on GitHub) that will be used all over our application to send and receive information between the services. He was successfully able to link it with a Heroku-hosted PostgresSQL database and tested the functionality of the services and their endpoints to match the expectations. Designing the API architecture schema, implementing the endpoints, and testing it on dummy data were the main tasks Tusheet accomplished this week. The front-end screen is in progress. Next week, Tusheet is going to link these backend endpoints to the front-end screen and ensure proper data storage and retrieval using the UI for the chatbot.

1.5 Tejas Pradeep's Progress

This week Tejas worked on linking the backend service with the database and the machine learning model. Since the backend is the backbone of the system it must be linked to all aspects of the systems and this week I worked on linking the backend to the Heroku Postgresql database, next week I shall work on uploading data from the database, this week I worked on reading from the database. I also further tested that functionality. I also linked the backend to the Physical Illness model to be able to communicate between the model and the backend. Without much data in the database, I was not able to test that functionality very well but I shall be able to next week.

1.6 Sanket Manjesh's Progress

Sanket is attempting to connect the Flask server we have built with the PostgreSQL database that was created. However, Sanket is running into some issues with connecting with the database and is not able to receive data properly from the database to test and display. This could be due to problems with the URL/password for the database or security issues and Sanket hopes to resolve these issues this week.

2 CHALLENGES ENCOUNTERED

2.1 Sanket Manjesh's Challenges

Sanket hopes to resolve issues dealing with connecting the Flask server with the database as mentioned above and possible security concerns preventing the connection.

3 FUTURE PLANS

The team members made significant progress on their tasks. In the future, each team member will continue working on their respective tasks and prepare for their respective parts of the project to be integrated with one another.

4 REFERENCES

1. Patil, P. (2020, May 24). *Disease Symptom Prediction*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/itachi9604/disease-symptom-description-dataset
2. Saxena, N. (2020, September 6). *Extracting Keyphrases from Text: RAKE and Gensim in Python*. Extracting Keyphrases from Text: RAKE and Gensim in Python | by Nikita Saxena | Towards Data Science. Retrieved April 3, 2022, from https://towardsdatascience.com/extracting-keyphrases-from-text-rake-and-gensim-in-python-eefd0fad582f
3. scikit-learn developers. (2022). *sklearn.feature_extraction.text.TfidfVectorizer*. sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.0.2 documentation. Retrieved April 3, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
4. scikit-learn developers. (2022). *sklearn.metrics.f1_score*. sklearn.metrics.f1_score — scikit-learn 1.0.2 documentation. Retrieved April 3, 2022, from

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

5. scikit-learn developers. (2022). *sklearn.model_selection.Kfold*. sklearn.model_selection.KFold — scikit-learn 1.0.2 documentation. Retrieved April 3, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html