Practicum Sprint #7

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

During this sprint, Akshay continued working on the PIP models, updated the dataset processing and model training scripts, fixed import errors, and worked on the response builder.

For the PIP models, Akshay added log files to save their performance results on training data and unseen data. The model that performs better on unseen data is the model that the user's messages should be passed to. There is a constant in the code that can be manually set by the team after inspecting the log files to ensure that the user's messages are passed to the better-performing PIP model in production, so the predictions are as accurate as possible. Akshay also improved the functionality of removing stopwords from the user message to ensure that the detected keywords (from which symptom information is extracted) are more likely to pertain to symptoms the user is experiencing. This improves the performance of the downstream PIP models.

Akshay also updated the PIP scripts for dataset processing and model training/evaluation can be run from any directory in the project directory structure, and the output files will always be produced in the same places.

Akshay also fixed import errors with functions from the PIP code that he ran into when working on the response builder. These fixes will prevent import errors for the backend when making predictions using the PIP models.

Akshay also wrote the response builder code to format the PIP results into an easily-readable message for the user. The response builder passes the user's message to the better-performing PIP model and selects the user's 3 (this value

can also be adjusted through a constant in the code) most likely physical illness risks from the prediction results. That information is compiled into a message that lists (from most to least likely) the physical illness risks by their respective names and probabilities.

Akshay's work is pushed to the GitHub repository and can be seen on the *main* branch. Next sprint, Akshay will write tests for the PIP code to observe how it performs across several test cases. This will help determine if any improvements need to be made to the PIP code before it is connected with the other parts 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. Afterward, 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:

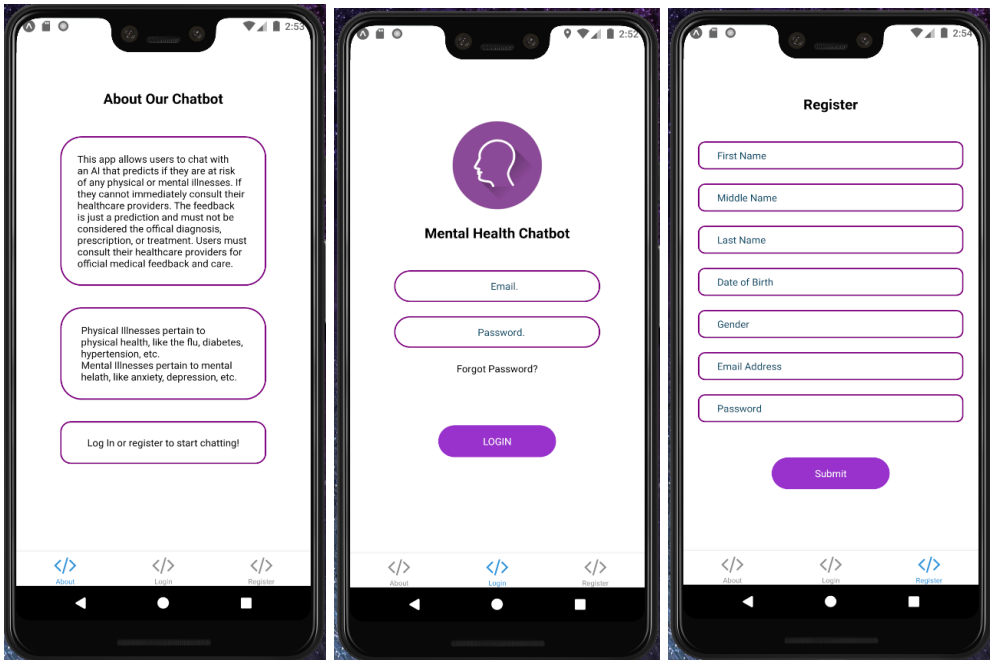

Figure 1: *These are the About Screen (left), Login Screen (middle), and Register Screen (right) of the application*

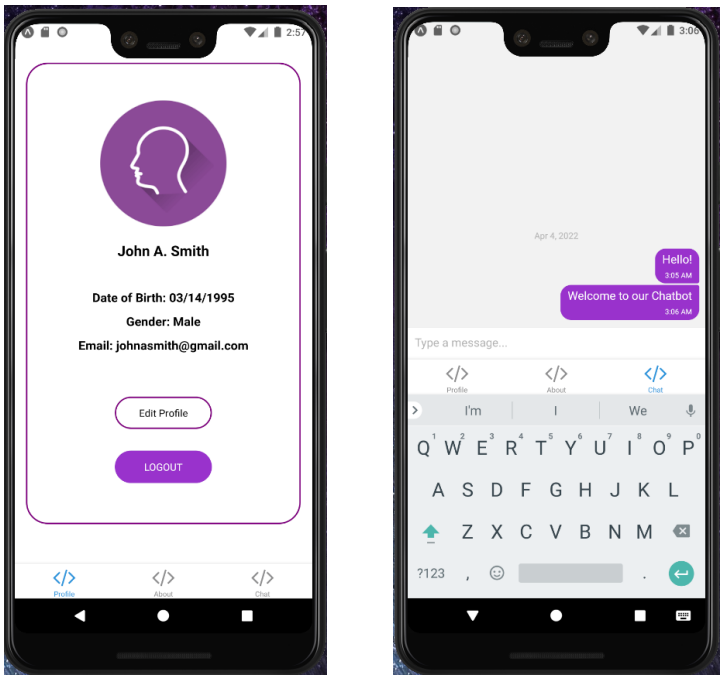

Figure 2: *After logging in, these are the Profile Screen (left) and Chat Screen (right) of the application*

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 worked on linking the backend service with the database and the machine learning model. He worked along with Tejas to integrate the backend API with the front-end UI buttons to properly relay information between the PostgresSQL database and the machine learning model of the physical illnesses prediction. 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. The database is currently empty so were unable to completely test this code.

Tusheet also helped with logging the chatbot conversations to the backend database to keep a record of conversations and the resultant illness classification from the machine learning model. Tusheet has thus set up the backend pipeline to relay information between the front end screen and the machine learning model to accurately transmit information to the appropriate services at each step and display the resultant illnesses classification. Next week, Tusheet is going to continue testing this and further integrate the mental illnesses prediction machine learning model to ensure similar functionality to the physical illnesses model.

1.5 Tejas Pradeep's Progress

This week Tejas worked heavily with Tushee on the backend of the system. Tejas worked on linking the front-end UI, the button, and inputs with the PostgreSQL database and the machine learning model. Specifically, Tejas worked on parsing input from the user and extracting relevant attributes from that input that can be passed onto the Machine Learning model which can then use those parameters to make a prediction. Next week Tejas shall work on transferring the output of the model onto the front end which can then be parsed to provide the users with useful predictions. Since the database is currently empty we are unable to test this code, next week we shall also populate the database to be able to test our code.

1.6 Sanket Manjesh's Progress

This week, Sanket pushed his progress with the Flask server so far and will let Tusheet fix the issues dealing with linking the Flask server with the PostgreSQL database as he has more experience in the area. Now, Sanket has shifted his focus to the ML side of the project and will be helping with developing the NLP models to discern sentiment from chats. These tasks will involve setting up models to detect bipolar disorders through random switches in positive/negative sentiment and also keyword detection for diseases mentioned within texts. Sanket will look into using the TextBlob library for sentiment analysis.

2 CHALLENGES ENCOUNTERED

None

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.