Practicum Sprint #2

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 DESIGN

1.1 Project Summary

A primary motivation behind this project is the increasing prevalence of mental health issues in recent years, in addition to ongoing physical health issues. From the National Survey on Drug Use and Health, researchers estimate the number of adults with some degree of serious psychological distress has increased by 71% from 2006 to 2017, among young adults (Rosenberg, 2019). Furthermore, due to studies at Pew Research Center, it is known that more than 70% of people aged 18-24 use Snapchat and Instagram regularly (Ortiz-Ospina, 2019).

We wish to implement a project that would be relevant to both of these trends. A chatbot system resembling a social media interface can use machine learning to predict if the user is at risk of any mental illnesses or physical illnesses.

We define mental illnesses as illnesses mainly pertaining to mental health. We plan to support anxiety, depression, and bipolar disorder in our application. We define physical illnesses as illnesses mainly pertaining to physical health. We plan to support 41 physical illnesses (specified in the Disease Symptom Classification dataset (Patil, 2020)) in our application.

In terms of features, our application intends to provide:

- Interactive UI for a mobile application that takes in user input and visually returns feedback
- Machine learning/natural language processing techniques on the backend to analyze user symptoms
- A database connection to store user information and messages

- Accurate predictions pertaining to potential mental and physical illnesses

1.2 Tools and Technology

Through the project, we plan on using various different tools and techniques to achieve our goals.

- Mobile App
    - Mobile applications frontend shall be built in React Native to enable us to easily use the app on both Android and iOS
    - Further data for the mobile app shall be stored on SQL, using a hosting platform such as PostgreSQL with Heroku.
    - The mobile app shall also have a backend developed on Python Flask to enable Restful API development and ease of integration with other aspects of the app.
- Server
    - We plan on using a server hosted on the cloud through free-to-use services like Heroku, with data being stored in SQL.
    - The backend code for the mobile app shall be built in Python Flask and shall serve as the link between the frontend and the server.
- Machine Learning Models
    - The machine learning models shall also be built with Python, specifically using NumPy and pandas for data cleaning and data analysis and scikit-learn libraries for the ML models to be used.
    - We chose to use Python for both the backend and the ML models for better integration between the two segments.

1.3 Data Sources

Disease Symptom Classification (Patil, 2020)
The Kaggle dataset lists multiple symptoms and precautions to be taken along with weighted importance values for 41 unique diseases, including GERD, AIDS, diabetes, and gastroenteritis. We plan to train an ML model on this data to recognize these diseases from the user's descriptions of symptoms and predict if they are at risk of certain physical illnesses.

Twitter Emotion Analysis (Merin S, 2020)
This Kaggle dataset contains several tweets labeled by emotion (happy, sad, anger, etc.). We plan to develop and train an ML model on this data to predict the sentiment of the user from their messages and use that information to predict if they are at risk of certain mental illnesses.

Emotions dataset for NLP (Praveen, 2020)
This Kaggle dataset contains several sentences labeled by emotion (joy, sadness, fear, etc.). We plan to train an ML model on this data to predict the sentiment of the user from their messages and use that information to predict if they are at risk of certain mental illnesses.

1.4 Diagrams

Figure 1 below shows the architecture of the chatbot system and the flow of information between the user and the chatbot system.

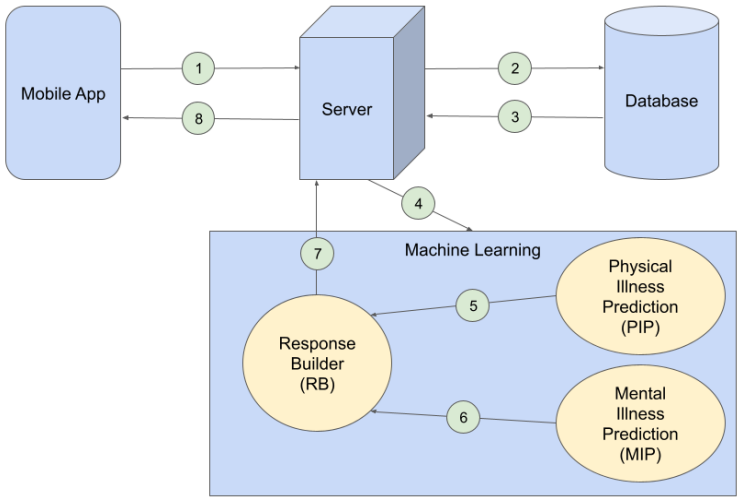


Figure 1—Architecture and information flow diagram for the chatbot system.

The chatbot system consists of a mobile app, a server, a database, and a machine learning suite. The ML suite contains an ML model for predicting physical illness (PIP), an ML model for predicting mental illness (MIP), and a response builder (RB) that builds a response to the user from the results of both models.

The information flow is described below as a series of numbered interactions that correspond to the numbered interactions shown in the diagram.

1. The user sends a message from the mobile app, which is received by the server.
2. The server stores the message in the database.
3. The server gets the last *X* messages from the database. The value of *X* will be tuned during the development and testing of the system.

4. The server sends the X messages to the ML suite.
5. The last message is passed to the PIP model to predict the physical illnesses the user may be at risk of.
6. All X messages are passed to the MIP model to predict the mental illnesses the user may be at risk of.
7. The RB builds a response to the user from the results from the PIP and MIP models.
8. The server sends the response to the user.

1.5 Screen Mockups

The screen mockups of the mobile app are shown in the following figures.

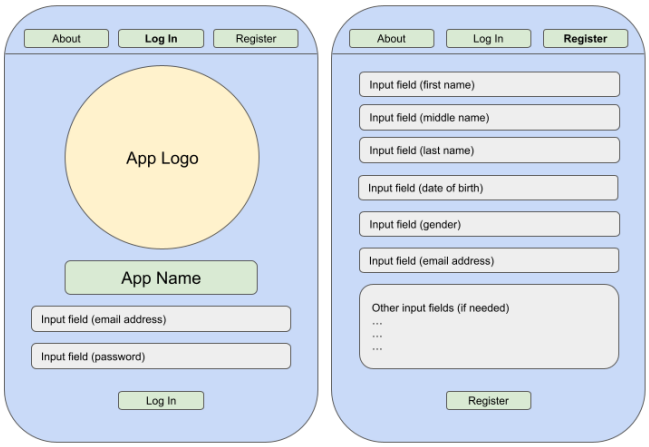


Figure 2—Log In and Register screens of the mobile app.

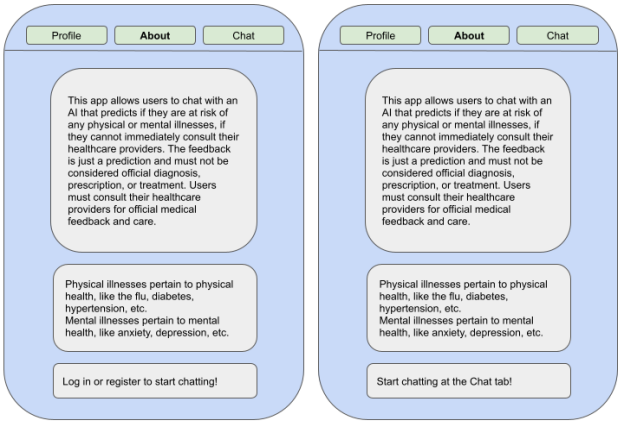


Figure 3—About screens of the mobile app, before (left) and after (right) logging in.

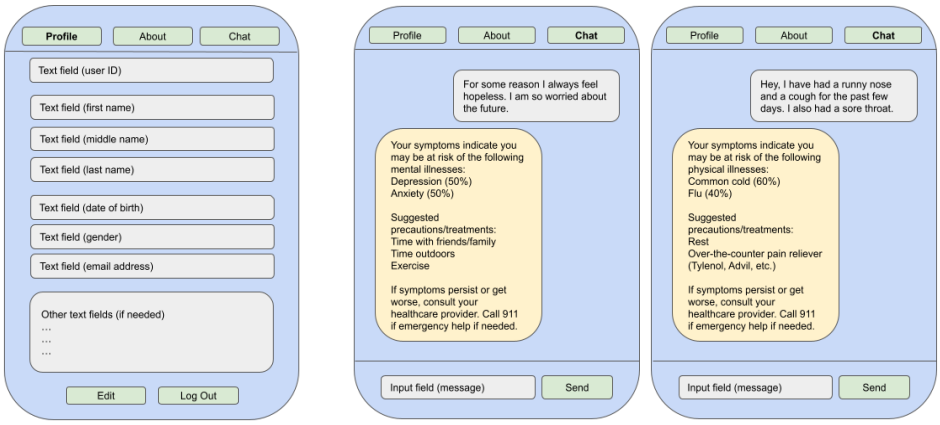


Figure 4—Profile screen (left) and Chat screens (right) of the mobile app. Chat screens show risk predictions for mental illness and physical illness, respectively.

2 IMPLEMENTATION PLAN

2.1 Project Tasks

Task 1: Preprocess Data from Datasets

The first part of our project will deal with obtaining data from each of the main datasets in order to train our models. To read and preprocess our data, we can use Python and its CSV library to read in the data, filter out specific features and rows with missing/null values if needed, and format it so that it can be used for training the ML models.

Task 2: Splitting Training and Test Data

The second part of this project will involve splitting our data into training data and testing data. Initially, we will utilize a basic 80/20 split (80% training data and 20% testing data), and we will consider other splits down the line as well. We will also attempt to balance our dataset by randomly shuffling the order of our data so that neither the training nor testing set is biased.

Task 3: Creating, Training, and Validating ML Models

To create our disease classification predictor, we will attempt to use simple regression models, Bayesian classifiers, random forests, and neural networks and pick the one that provides the most accurate classification results. After creating our models and training them with data from our datasets, we can validate them

using k-fold cross validation through various metrics, such as testing accuracy, F1-score, precision, and recall.

Task 4: Designing a Mobile Application

The front end of the system will be a mobile app. This mobile app will allow users to create a profile with their information and message the chatbot. To create this front-end application, we can utilize React Native.

Task 5: Create Database to Store Data

We will also need to build a database to store user messages. We can build this database utilizing PostgreSQL.

Task 6: Integrate Database with Models and Front-end Application

Our final task will be to integrate our database with our front-end application and ML models. We plan to create a server that will take a request from the front-end application, retrieve data from the database, feed this data to the ML models for predictions, build to a response, and send it back to the front-end application to display to the user. We can create the server using Flask.

2.2 Project Timeline

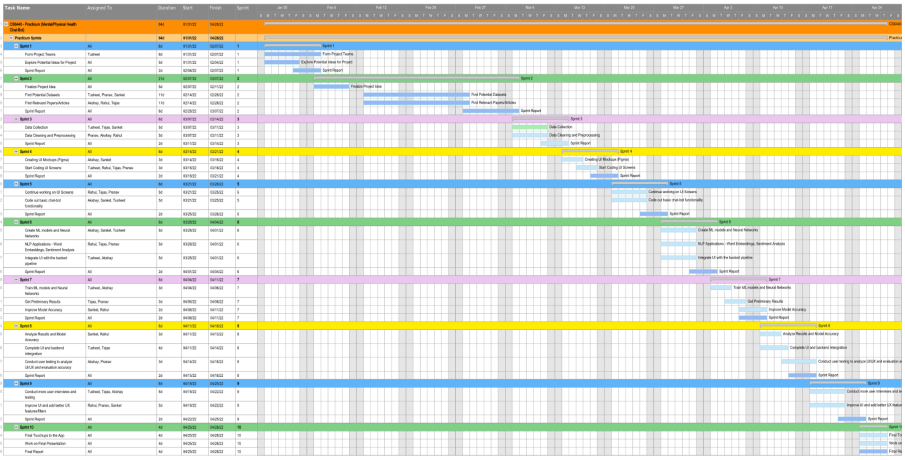


Figure 5—Gantt chart outlining project timeline. See page 10 for an enlarged image.

As you can infer from the Gantt chart, we have divided our project into 10 weeklong sprints. These phases are logically structured and tasked appropriately

to make them doable within a week while making meaningful progress towards the final project.

Here is a quick breakdown of what we plan on doing in the 10 sprints. The Gantt chart provides a more detailed view of the individual tasks assigned to each sprint.

Sprint 1 – Form project teams and explore potential project ideas.
Sprint 2 – Finalize project idea, and find potential datasets and other relevant research papers.
Sprint 3 – Data collection, cleaning, and preprocessing.
Sprint 4 – Creating UI mockups and starting coding out UI screens.
Sprint 5 – Finish working on UI screens and integrate chatbot functionality into the application.
Sprint 6 – Create ML models, NLP applications (LSTM, embeddings, etc.), and integrate backend and UI.
Sprint 7 – Train ML models and get preliminary results for the model.
Sprint 8 – Improve models and result accuracy, complete UI and backend integration, and start conducting user testing and interviews.
Sprint 9 – Prioritize more user interviews and improve the UI and UX features of the application.
Sprint 10 – Work on final touchups to the application and work on the final presentation and report.

2.3 Needs/Risks

Some of the major preliminary things we need for this project are datasets, reputable and relevant research papers and articles, and a software architecture plan. We have done extensive research and have found all the datasets and papers that we plan on using and can help us in our approach. We also have a robust software architecture plan and data pipeline for our project which has been explained in our technologies and architecture diagrams. In the later stages, we are going to need users for testing and validating our application as well as suggest better UI/UX upgrades to the application.

Some of the general concerns of the team regarding this assignment arise from us being successfully able to identify mental illnesses like depression, anxiety, stress, etc. just from a few short conversations with a chatbot. Our hypothesis is

that these mental illnesses come along with a certain characteristic tone and language that can be picked up in a conversation with the chatbot. But we are afraid that lacking the non-verbal aspects of conversation such as body language, facial expressions, etc. might negatively impact our model's accuracy.

We have been able to find some good research papers that enable us to perform novel NLP applications like sentiment analysis, and other speech/text recognition patterns to identify and classify mental illness. But most of these papers use complex word-embeddings and other complex models to accurately perform these NLP applications. We are afraid that we might not be able to accurately replicate the works of these papers while adding our novelty to this algorithm.

With busy schedules for our team members, we are afraid if we have either over-scoped or under-scoped our project. We all are of the opinion that the scope of this proposed project is reasonable, while at the same time is an interesting idea that could use some additional novelty to the algorithm. This is not a major risk/concern per se, but this is something that could arise as a potential problem in the future with our team members' busy and changing schedules.

Thus, scope creep, unable to implement paper techniques, low accuracy for our algorithm is some of the potential risks for our plan.

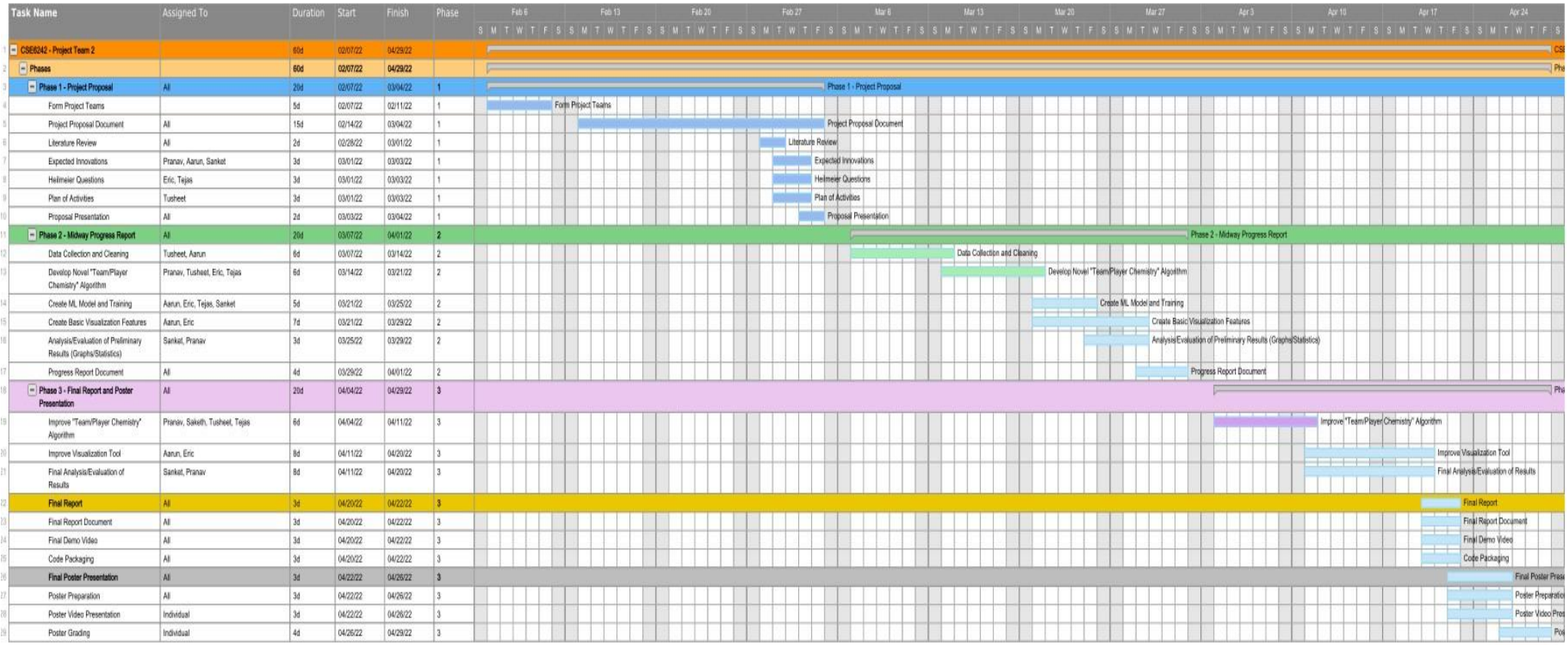
3 REFERENCES

1. Merin S, S. (2020, April 17). *Twitter Emotion Analysis*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/shainy/twitter-emotion-analysis/data
2. Ortiz-Ospina, E. (2019, September 18). *The rise of Social Media*. Our World in Data. Retrieved February 7, 2022, from https://ourworldindata.org/rise-of-social-media
3. Patil, P. (2020, May 24). *Disease Symptom Prediction*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/itachi9604/disease-symptom-description-dataset
4. Praveen. (2020, April 16). *Emotions dataset for NLP*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp
5. Rosenberg, J. (2020, July 30). *Mental health issues on the rise among adolescents, young adults*. AJMC. Retrieved February 7, 2022, from

https://www.ajmc.com/view/mental-health-issues-on-the-rise-among-adolescents-young-adults

Gantt Chart

Link - https://drive.google.com/file/d/16W77ProdMKOBTJszDbdj3EDJZtWW_ixr/view?usp=sharing



Task Name	Assigned To	Duration	Start	Finish	Phase
CSE6242 - Project Team 2		60d	02/07/22	04/29/22	
Phases		60d	02/07/22	04/29/22	
Phase 1 - Project Proposal	All	20d	02/07/22	03/04/22	1
Form Project Teams		5d	02/07/22	02/11/22	1
Project Proposal Document	All	15d	02/14/22	03/04/22	1
Literature Review	All	2d	02/28/22	03/01/22	1
Expected Innovations	Pranav, Aarun, Sanket	3d	03/01/22	03/03/22	1
Heilmeier Questions	Eric, Tejas	3d	03/01/22	03/03/22	1
Plan of Activities	Tusheet	3d	03/01/22	03/03/22	1
Proposal Presentation	All	2d	03/03/22	03/04/22	1
Phase 2 - Midway Progress Report	All	20d	03/07/22	04/01/22	2
Data Collection and Cleaning	Tusheet, Aarun	6d	03/07/22	03/14/22	2
Develop Novel "Team/Player Chemistry" Algorithm	Pranav, Tusheet, Eric, Tejas	6d	03/14/22	03/21/22	2
Create ML Model and Training	Aarun, Eric, Tejas, Sanket	5d	03/21/22	03/25/22	2
Create Basic Visualization Features	Aarun, Eric	7d	03/21/22	03/29/22	2
Analysis/Evaluation of Preliminary Results (Graphs/Statistics)	Sanket, Pranav	3d	03/25/22	03/29/22	2
Progress Report Document	All	4d	03/29/22	04/01/22	2
Phase 3 - Final Report and Poster Presentation	All	20d	04/04/22	04/29/22	3
Improve "Team/Player Chemistry" Algorithm	Pranav, Saketh, Tusheet, Tejas	6d	04/04/22	04/11/22	3
Improve Visualization Tool	Aarun, Eric	8d	04/11/22	04/20/22	3
Final Analysis/Evaluation of Results	Sanket, Pranav	8d	04/11/22	04/20/22	3
Final Report	All	3d	04/20/22	04/22/22	3
Final Report Document	All	3d	04/20/22	04/22/22	3
Final Demo Video	All	3d	04/20/22	04/22/22	3
Code Packaging	All	3d	04/20/22	04/22/22	3
Final Poster Presentation	All	3d	04/22/22	04/26/22	3
Poster Preparation	All	3d	04/22/22	04/26/22	3
Poster Video Presentation	Individual	3d	04/22/22	04/26/22	3
Poster Grading	Individual	4d	04/26/22	04/29/22	3