Psychosis iREACH: Reach for Psychosis Treatment using Artificial Intelligence

Jonathan Lee
Computing & Software Systems Division
University of Washington - Bothell
Bothell, WA
jjlee22@uw.edu

Sarah Kopelovich
Department of Psychiatry and Behavioral Sciences
University of Washington - Seattle
Seattle, WA
skopelov@uw.edu

Sunny Chieh Cheng
School of Nursing & Healthcare Leadership
University of Washington - Tacoma
Tacoma, WA
ccsunny@uw.edu

Dong Si*
Computing & Software Systems - Division
University of Washington - Bothell
Bothell, WA
dongsi@uw.edu

2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-6819-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIBM55620.2022.9995430

***Abstract*—Psychosis iREACH aims to optimize the delivery of evidence-based cognitive behavioral therapy to family caregivers who have a loved one with psychosis. It is an accessible digital platform that can utilize the user's intent and entities to determine the appropriate response. The platform is implemented based on an artificial intelligence and natural language understanding (NLU) framework, RASA. We developed the web application of the platform, and the chatbot has been integrated into the platform to collect data and evaluate the performance. The results showed that the NLU model's accuracy, precision, recall, and F1-score of the intent prediction are *88.31%, 89.80%, 88.22%,* and *88.65%* respectively. The link to the website is https://psychosisireach.uw.edu/.**

Keywords—artificial intelligence, natural language understanding (NLU), natural language processing (NLP), chatbot, web development, data exploration, psychosis, mental health, project management, RESTful API

I. Introduction

Psychosis refers to a transient experience in which individuals have difficulty distinguishing between consensus reality and their own distinctive experiences. Psychosis is commonly associated with schizophrenia spectrum disorders, which are associated with 2 to 3 times the risk of premature death compared to the general population [1]. Although these conditions affect roughly 1 % of the population, 1 in 100 Americans will experience psychosis in their lifetime. Approximately 60 million Americans are natural support for an individual with psychosis [2]. Individuals who are ill-equipped to deal with someone experiencing psychosis may feel highly distressed and unable to help due to the lack of knowledge of psychosis, let alone the steps to tackle the problems. Having a coach or an advisor who can walk through the steps on how to approach someone with psychosis could significantly increase the effectiveness of treatment [3]. However, coaching resources are generally hard to come by and are labor-intensive. Evidence-based Family Interventions for psychosis (FIp)[4] are incorporated into national schizophrenia treatment guidelines [5]. However, FIp is rarely delivered in practice; only 1.9% of the US families living with a member with psychosis receive proper psychoeducation [6]. Contributing to this deficient access is the fact that many behavioral health practitioners are not trained to provide FIp, families may struggle to attend in-person appointments to receive FIp, and some patients with schizophrenia refuse to permit their family members to be involved in aspects of their behavioral health care.

Psychosis iREACH was developed to address the challenges associated with delivering evidence-based FIp. Psychosis REACH (Recovery by Enabling Adult Careers at Home) is a FIp that delivers psychoeducation and evidence-based Cognitive Behavioral Therapy (CBTp)-informed illness management skills to family caregivers in the community. This psychotherapeutic intervention is proven to help individuals with psychosis improve their quality of life and fewer hospital admissions [7]. While Psychosis REACH utilizes a hybrid direct-to-families training model, thus bypassing the challenges associated with clinic-based FIp, neither the asynchronous nor synchronous digital training components enable repeated practice with feedback opportunities to families learning

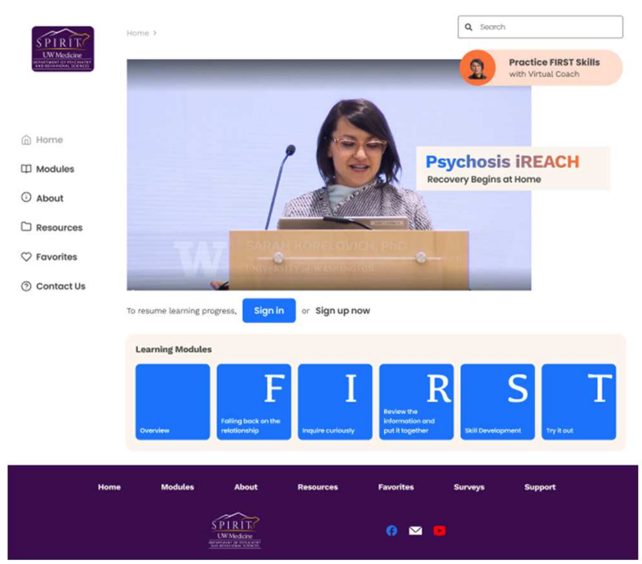

Fig.1. User Dashboard Design

* Correspondence to Dong Si at dongsi@uw.edu.

Psychosis REACH skills. Thus, comes the emergence of iREACH--a digital augmentation of the standard Psychosis REACH training model (Kopelovich et al., 2021) [8] to deliver a web-based platform that provides repeated practice with feedback for people. This tool aims to respond to the critical need for broad access to evidence-based Flp by employing a participatory, iterative user-centered design process to develop Psychosis iREACH. Figure 1 shows the example interface of the application.

In short, a coach chatbot is tasked to help users understand the FIRST skills applied to individuals affected by psychosis. The FIRST acronym encompasses five methods: Falling back on the relationship, inquiring curiously, reviewing the information and putting it together, skill development, and trying it out. Additionally, the program includes another chatbot named Alfie acting as the affected individual. Users would practice the FIRST skills based on a common scenario family caregivers might encounter. After the practice, the coach chatbot debriefs the users. Debriefing has been found useful for skill acquisition [9]. All of these features are accessible on a public web platform.

II. Related Works

A. AI ChatBot Design during an Epidemic like the Novel Coronavirus

The paper written by Gopi Battineni, Nalini Chintalapudi, and Francesco Amenta are the researchers who created a chatbot service similar to the iREACH framework. Users can contact the chatbot for advice on symptoms and actions to take when a human operator is unavailable. The chatbot can recognize the user's distress and gives comforting words and directions depending on each situation. In addition, the chatbot can perform basic administrative tasks such as scheduling appointments and routines based on the patient's conversation. The chatbot can also identify if the patient may or may not have covid based on the symptoms listed by the user [10].

B. Acceptability of artificial intelligence (AI)-led chatbot services in healthcare: A mixed-methods study

The paper written by Tom Nadarzynski, Oliver Miles, and Aimee Cowie is a study that aims to explore how willingly humans accept talking to an AI. The article covers three main themes: understanding chatbots, AI hesitancy, and motivations for health chatbots. Human acceptability increases the more familiar an individual is with the chatbots. Furthermore, a good understanding of chatbots and their limitations can lower users' expectations and use the chatbot as designed. AI hesitancy depends on an individual's belief that the chatbot is knowledgeable and understanding to the user. It asks the question, "Do they trust the AI to understand what their problems are true?" [11].

C. Applying Existing Works to iREACH chatbot

Each work addresses two different aspects: technical and social. The first work (Covid-19 chatbot) follows a very detailed explanation of its chatbot framework. It includes a flow chart diagram that mentions the high-level architecture from the user interaction to administrative tasks the chatbot requests. Though the main focus was the chatbot functionality rather than the entire frontend and backend functionality of a website, it still highlights some of the critical features of the chatbot, namely, its threshold values assessment system, and the ten different quality evolution aspects. The two parts give a way to measure and quantify the rating of the chatbot and website. Since each functionality (e.g., visual look, knowledge base, personalization, etc.) has a rating between 1 - 5, we can use it to measure the quality of the iREACH chatbot. In addition, the threshold system is a great way to determine general yes/no binary decisions based on multiple binary data.

Adjunctive internet-based interventions have shown cost-effective and engaging (Buck et al.,2021) [12]. However, most online interventions still rely on human moderators to deliver the intervention. The second work showed the feasibility to develop a chatbot assistive tool to deliver tailored online therapy based on user data. The psychosis iREACH framework will leverage the existing works on chatbots and integrate user experience for more sophisticated models. However, there are ways to improve quality and engagement as best as possible. One way is to have the platform perceived as an assistive tool rather than a human substitute. This perception helps users focus more on the learning experience of the platform rather than the chatbot's performance. Additionally, providing complete anonymity and convenience for the user may encourage them to use the service more often without the fear of being known or potentially exposed.

III. Framework Statement

A. Problem Statement

Therapeutic interventions and resources for people experiencing psychosis are inaccessible due to a host of factors at the macro (system), meso (organizational), and micro (individual patients and clinician) levels. Therefore, family caregivers are often left to take on the responsibilities of caretaking [13]. However, caregivers, family members, and friends are often poorly educated about their loved one's diagnoses and are ill-equipped to address the symptoms appropriately for their loved ones. Therefore, Psychosis iREACH aims to support the skill acquisition for family caregivers using a chatbot AI on a publicly available web platform that is accessible anytime, anywhere.

B. User Requirements

The framework gave several user requirements to be implemented by the end of the development process. Based on user-centered designed activities, three design requirements were suggested by the end users. (Table I).

TABLE I. Functional Requirements

User story title	***Story description***	***Tasks***
Chatbot	Users can interact and receive feedback from the chatbot. There are two types of Chatbot: Our Coachbot (gives user	Create the model

User story title	*Story description*	*Tasks*
	feedback and progress) and AlfieBot (acts as the victim, and gives user response options)	Link model with backend using REST APIs Host a RASA server
User dashboard	Users can view their module progress and navigate to different help sections in their dashboard. The dashboard should include a learning module containing text, images, and videos of the FIRST methods.	Home Module Content About Dashboard contents must update to user's progress from the backend/database
Sign up and log in	Users are required to fill out a sign-up form to create their accounts. Logging in is required for users if they want to keep up their progress	Sign up form Sign up form makes a POST/PUT request to the backend (create a new user in the database) Login Form Fetch and authenticate user input when logging in and redirect to their dashboard
Onboarding	Step-by-step introduction to the website and function of the chatbot with skip options and provides estimated time needed per step	Overlay
FAQ/Support	Under the dashboard, include tabs for technical support or contact information should the user have any critical issues with the web app	FAQ page Support page
Community Resources	The user navigates to the resources tab to get a list of external resources they can use to supplement their learning experience	Generate a list of resources based on the user's preference Zipcode input for location-based resources
Favorites	Unknown	
Surveys	Unknown	
Content Customization	Users can select specific symptoms they are dealing with, and the chatbot will tailor the experience based on the user's selection	

C. Problem Analysis

The problem splits into two parts: designing the chatbot and building a user platform for the chatbot. Both parts maintain a high priority in the development process as the chatbot model cannot exist without a platform, and the platform is meaningless without the chatbot. Moreover, the framework designation is to be a web application platform. The reason for choosing this platform is mainly due to accessibility, as users can use it if they know the site's domain name. Furthermore, no desktop application downloads are required; users can access the site anytime and anywhere as long as they have access to the public internet.

An appropriate dataset is needed to build the model to develop the chatbot. Unfortunately, no existing dataset addresses this specific issue, so the data must be collected in parallel, meaning early responses from users will be collected and trained to improve the user experience of the future. The best option is to use an existing deep learning framework that can automatically perform these functions and integrate seamlessly with the web app. An option is to use the RASA open-source framework. The RASA framework provides a supervised learning platform where the developers can act as both the chatbot and the users while the software observes and collects conversational data [14]. The RASA model's most common data type is the natural language understanding (NLU) data, whose main features include intent. The model can derive context and intent, which is re-trained by the RASA framework and deployed into production.

Moreover, the chatbot interface provides a terminal-based platform where developers can test and interact with their newly trained models. Developers can also use the set of application programming interfaces (APIs) from the framework, which sends custom response data to any designated endpoints. These APIs will be ones that the web app will use to display the results to the users.

IV. Methods and Implementation

A. Tech Stack

The entire framework will combine several frameworks into one container application, the technology stack. Each of these frameworks represents a stack. Due to the differences in functionality, methods, purpose, and language for each framework, developers must write a custom API for the stacks to communicate with each other. For instance, wherever the user wants to resume their progress with the chatbot, the user must retrieve the previous state from the database by requesting the backend application to signal retrieval of the stateful data. Depending on the request, the backend application must retrieve the corresponding elements of the data, which is back to the frontend UI. Therefore, writing an API is crucial for the communication between each application, a function or script that serializes and deserializes object data into a readable format for the application. Fig.1 shows an example of the tech stack for the iREACH framework.

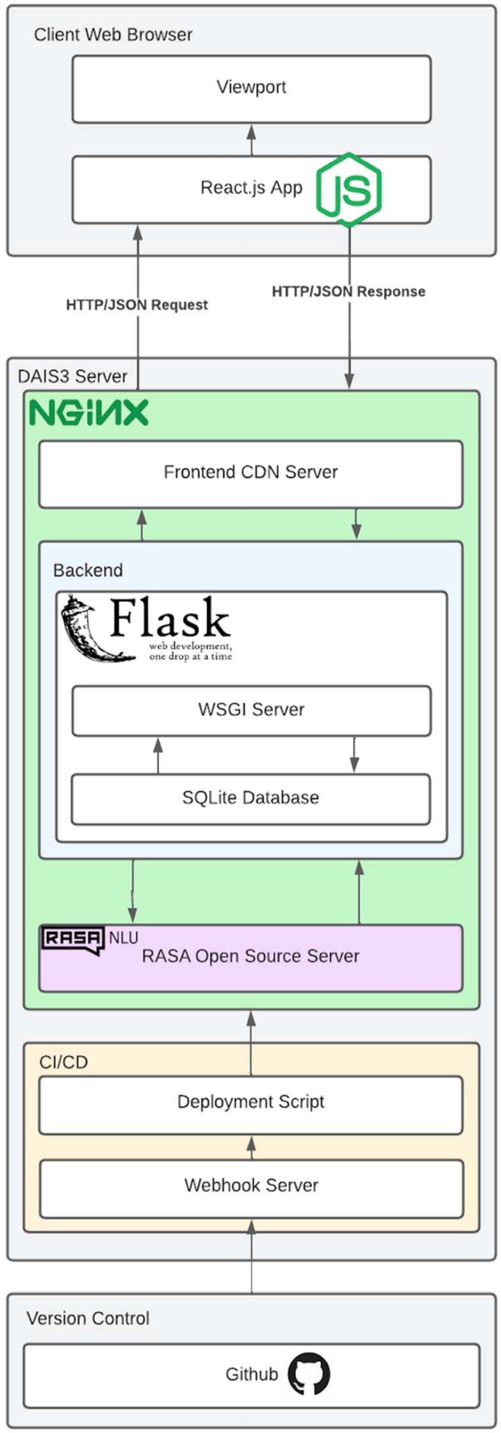

Fig. 2. Complete High-level Architecture of the Tech Stack

All frameworks used for the web platform are open-source, meaning the applications are free to use by the public as long as they are appropriately credited. If the platform is developed as a team effort, it is encouraged to use collaborative tools such as Git and continuous integration/deployment on a web server to ensure smooth-uninterrupted deployment. Lastly, a proxy/web server will serve all the web content requested from the users.

B. *Frontend Framework*

The frontend framework uses the React.js framework, developed by Meta. It is a JavaScript-based framework that provides all the necessary libraries specializing in user interface development. The advantage of React.js, compared to other web frameworks, is that it uses virtual DOM rather than having to manipulate it directly. This feature increases speed and flexibility for the browser's ever-changing contents, which is utilized heavily for interfacing user progress and chatbot interaction. Furthermore, combined with the Redux library, it allows for the development of powerful state management between UI components and significantly simplifies any dynamics updates to the UI.

In addition to Redux, the frontend framework uses a library called Materials UI (MUI) to easily create the professional-looking design of the website, as the modern style of the website will reflect the seriousness of the platform. MUI also automatically resizes the viewport of the contents, so users can browse on the phone browsers (although not recommended). The deployment of the front end uses NGINX, which will serve all the React elements in the client's browser. All clients are assumed to be using modern web browsers such as Chrome, Edge, Firefox, and Opera, so compatibility between web browsers should be a non-issue.

C. *Backend Framework*

The backend framework uses the Flask micro-framework, a Python-based web framework. Flask's advantages are its ease of use and minimal setup time. In addition, Flask relies only on the Python standard library and does not require any third-party libraries to run a backend server; therefore, it is relatively easy to construct and prototype different custom APIs. Another advantage of Flask is that it has a built-in SQLite database that can to queried directly from the Flask framework; thus, there is no need to create a separate database server to communicate with the backend.

The backend's function is to serve the appropriate data between the frontend and the database. For example, when a user wants to log into their iREACH account, the backend first verifies if the user ID exists in the database (or is registered); otherwise, it will redirect the user to the sign-up page. Once the user verification is complete, the backend will redirect the user to their home page along with their progress so far. The backend must process these requests concurrently as multiple users will be on the platform simultaneously.

D. *Database*

Flask's default built-in database uses SQLite. SQLite is a relational database that is serverless and self-contained. It can retrieve data instantaneously for small to medium-sized databases since it only loads the needed data rather than the entire set. However, as the database grows with more users, switching to a more dedicated database management system such as MongoDB may be necessary due to SQLite's limited storage capacity and the lack of scalability for expansive datasets.

For the storage itself, the database will contain all the user's basic information such as their user ID, first name, last name, password, birth date, email, sex, and the relationship with their significant other. In addition, there will be a table for the user's data and a table for the chatbot, which keeps track of the current chatbot's state and progress for users to resume their activity (Fig. 3).

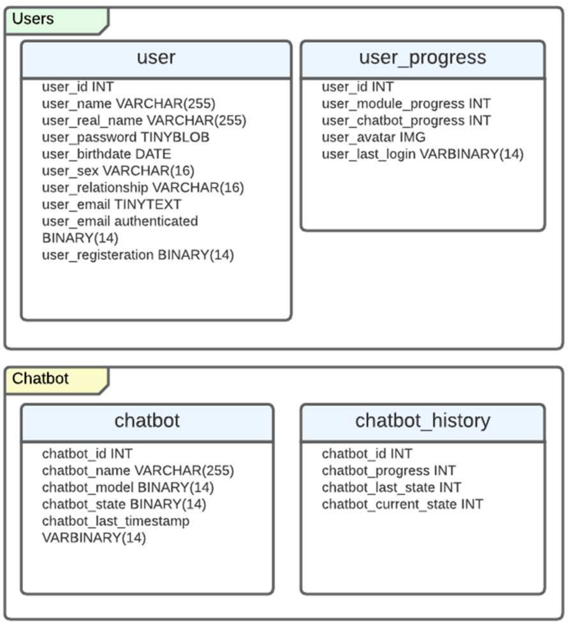

Fig. 3. Database Schema for Users and Chatbot

E. RASA Chatbot Design

The user opens the chatbot application, and the chatbot begins with a greeting, followed by options to choose which FIRST Skill module the user would like to work with. Next, the chatbot walks through with the user, explains the FIRST skill, and practice and debrief the FIRST skills based on a caregiving scenario developed by content experts. Figure 4 depicts the flowchart overview.

F. Server Implementation and Continuous Integration

The iREACH framework is a team-collaborative effort with multiple developers working on the framework simultaneously. Therefore, it is crucial to maintain version control and documentation throughout the development cycle. iREACH has two sites that serve different purposes: the test site and the production site. The test site is a private site only accessible through a private VPN. It serves as a precursor site to the production site. Developers can test their code for any bugs on the private site before deploying it to the production site. The production site serves as the site to which the users access and interact with the platform.

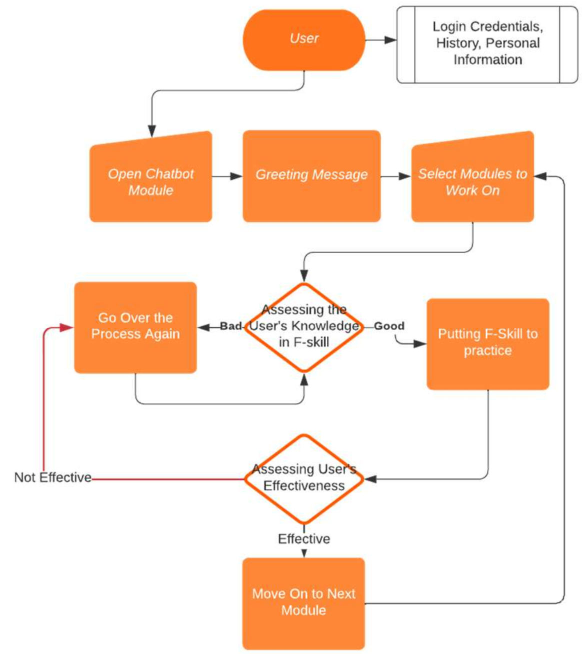

Fig. 4. Chatbot Flow Diagram

Whenever developers push a new commit (using Git) to the test site, a Github webhook is sent to the server, which signals the server to pull the latest changes and update the site. A custom bash script is responsible for the auto-deployment of the test and production sites.

V. CHATBOT IMPLEMENTATION

A. RASA Framework

According to the RASA framework documentation, the best features of RASA are its defined stories, intent, and action. Stories define the main task of the chatbot. For instance, the Alfie chatbot's current story teaches the user to show empathy when their loved ones are in a psychosis. This story is defined as the primary goal of Alfie's chatbot. Within the stories, we can determine how the chatbot should react (take action) depending on the user's intent (Fig. 5).

B. Data Format

The RASA's standard format of the NLU data would be the dataset. The NLU data (in YAML format) comprised four main

```
- intent: empathy
  examples: |
  -  i had no idea you felt that way
  -  am I right
  -  it sounds like everyone is ganging up
     on you
```

Fig. 5. Defining User's Intent

attributes: version, nlu, stories, and rules. The version states what training data format specifications the RASA framework should use for its model. The RASA framework will be using the RASA Open Source 3. X and higher.

C. NLU Dataset

The NLU training data comprises user responses, categorized by intent. The intent in the RASA framework is defined as the user's intention behind his or her's message. For example, when a user writes a message, "Where is the closest store?" the user intends to find the closest location from one's place. Therefore, we can define the NLU as check_closest_location and provide examples of what the messages would look like. To derive its intent, RASA will use the examples to gain more context and check which example matches more closely than others. The NLU data can also take synonyms, regular expressions, and lookup tables; however, the developer's methods are more hardcoded and are determinate than the RASA's intent-based modeling.

D. Stories and Rules

Stories and rules represent the conversations between the user and the chatbot. Stories help the machine learning model recognize the pattern from within the conversation, while rules define a specific path to where the chatbot must follow. For example, if the chatbot acknowledges that the user intended to greet the user. It would then execute an action where the chatbot would greet the user back. Rules are similar to stories, except they will execute particular messages when certain conditions are met, such as only greeting the user when the user's name is provided. In addition, the ability of the rules can chain in a series of conversations that would lead up to the chatbot's goal, e.g., providing all information on the FIRST skills for the people living with psychosis.

E. RASA Handling of Unexpected Input

Unlike the previous studies on the covid-19 chatbot, the study did not mention how the chatbot would respond to unexpected user input. Luckily, the RASA framework provides a feature that can define how the chatbot should respond if the input is unrelated to the current story; this is known as a fallback. There are two types of interjections: generic interjections and contextual interjections. Usually, for a generic interjection, the chatbot simply needs to remind the user of the current state of its story. However, the chatbot will need to recognize and respond to an interjection based on the previous conversation for contextual interjections. By creating a story that can handle interjections, the chatbot looks out for any influential conversations. Due to how the framework takes these conversations, the iREACH chatbot will most likely use the interjections for a more advanced chatbot implementation.

F. Script Options

Since there is no public NLU dataset that follows the FIRST skill conversations, all NLU data were manually crafted to resemble predicted user responses. The predicted responses are based on the provided script that describes all the possible

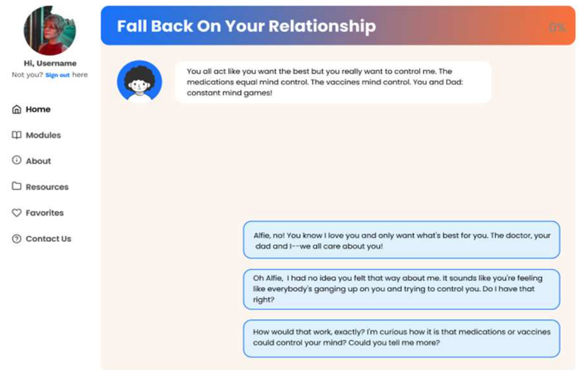

Fig. 6. Current chatbot module showing choices of user response of Alfie

responses the users are expected to type. For instance, the user is given the following dialogue:

Alfie Bot: *I feel respected when people stop trying to control my mind. I just wish everyone would leave me alone.*

The script provides three different options for user response:

Option A: *I hear you. I thought I was doing the right thing but I guess my intentions were being misunderstood. I promise to lay off from now on.*

Options B: *I wonder if we could maybe just take a break from talking about the meds, and your treatment, and just blow off some steam. It's been a while since I've done something just for the fun of it. I remember when we used to go to the trampoline park right when it opened--before it got crowded. Do you remember how fun that was?*

Option C: *I know you think you want everyone to leave you alone, but that sounds like a pretty lonely life to me. We share our concerns because we care.*

Depending on the user's choice, the coach bot will make corrections and provide feedback to the user.

Coachbot: *This is an example of a simple reflection with the added skill of a slight rephrase to emphasize that this is a feeling or interpretation rather than a statement of fact ("It feels like..."). Reflections are a type of active listening technique that restate, paraphrase, or uncover individuals' emotional reactions to situations, thoughts, behaviors, or interpersonal interactions. This simple technique of reflecting someone's emotions can help bring awareness of and ability to label their own emotions. It also provides assurance that you are listening without judging or jumping into problem-solving mode. Great work. This is a key "F" skill that should be used early and often.*

The script options will be added to the NLU dataset and various RASA models will be used to determine which custom responses align closest with each of the three responses. Fig 6 is an example of an interface for some of the options. Over time, live responses from test users will be collected and used to improve the reliability of the response prediction.

VI. Evaluation

A. Model Performance Results

RASA framework provides a built-in performance metric for its model. It has the option to either shuffle/split or perform cross-validation on the NLU dataset before performing a metrics test. The framework also gives the ability to fine-tune hyperparameters to improve the score. Once RASA performs a test, it outputs a log that reports each metric such as the number of correct predictions, F1-score, recall, precision, accuracy, confusion matrix (with or without normalization), and a predictive histogram. Results are split into the FIRST categories and the NLU dataset is evaluated at the global 80/20 train-test split. Each metric is evaluated on weighted averages. (Table II)

TABLE II. Model Performance Metrics Weighted Avg

	Accuracy	Precision	Recall	F1-score
F-skills	0.9119	0.9322	0.9193	0.9180
I-skills	0.9084	0.9171	0.9001	0.9085
R-skills	0.8657	0.8813	0.8651	0.8731
S-skills	0.9048	0.9281	0.9073	0.9175
T-skills	0.8248	0.8313	0.8196	0.8254

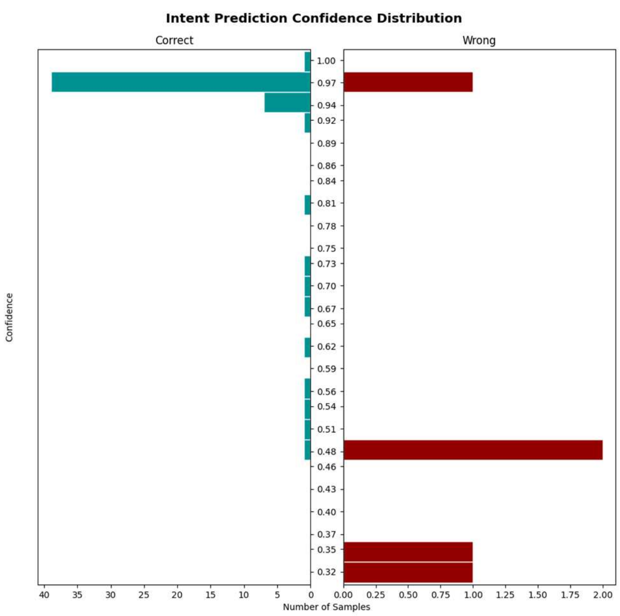

Fig. 7. Intent Prediction Confidence Distribution for F-Skill

The histogram for the NLU dataset for F-skill (Fig. 7) allows visualization of confidence for all predictions. The blue and red bars indicate correct and incorrect predictions respectively. The confusion matrix (Fig. 8) shows the frequency at which the intents are mistaken for others in the F-skill NLU dataset. From the results, the majority of the intents are correctly predicted with the weighted average of accuracy, precision, recall, and F1-score across all skills being *88.31%, 89.80%, 88.22%,* and *88.65%* respectively. There are certain dialogues in the F-skills in which the model had trouble predicting due to the close similarity of the words and sentence structure. The T-skill module suffers the lowest score mainly due to its short dialogue options (only 3 sections compared to 6 in F-skill). The built-in performance metrics in RASA also provide a good insight into how the model determines the intent, and the tool will be used to a greater extent during production when more NLU data is collected from the users. The results also demonstrate the viability of the model, as it can accurately predict the correct intents even with very little NLU data. Therefore, RASA open source should be highly considered for the iREACH framework.

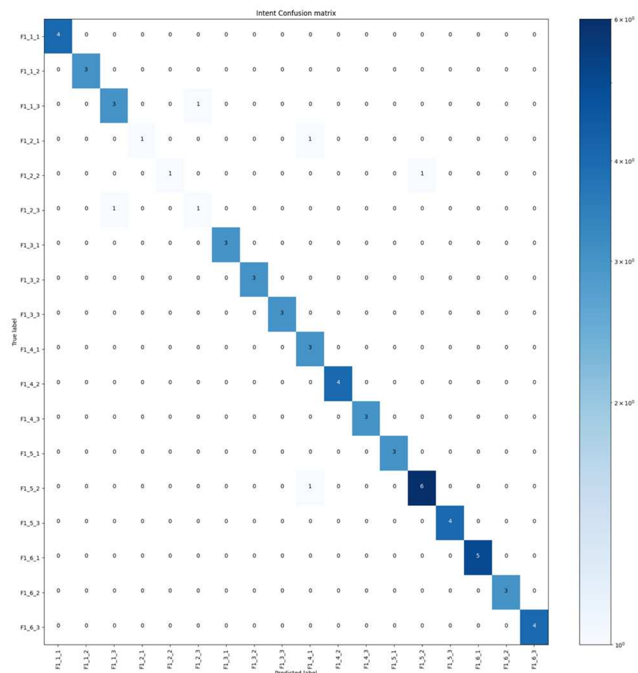

Fig. 8. Confusion matrix showing which intents are mistaken for others for F-skill. Both axis denotes the sections of the script

VII. Conclusion

Based on the implementation methods of the website/chatbot and model performance results, the iREACH framework has the potential to become a fully-fledged website that caters to family and friends to help people living with psychosis. Feedback is continuously received from the test users to make any changes in the UI and chatbot experience itself. The underlying technical framework is laid out to build the platform, and the model performance results will help improve the user experience.

During the initial development, many different frameworks were used or swapped until a suitable stack is found. We experimented with different frameworks, such as using Node.js for the backend and MongoDB on a separate database server. However, some were dropped and replaced using simpler and more robust frontend and backend framework. The choice for the RASA framework is due to the simplicity of training, building, and deployment of the model and utilizing existing web APIs that can communicate with the backend easily. Due to the large scale of the framework, the framework includes a CI/CD pipeline. The pipeline intends to abstract the complicated functions of website building and deployment into an automated process. This applies also to RASA since the chatbot AI must be able to collect user data and append information to its NLU dataset to be re-trained and re-deploy as soon as possible. As the framework is considered part of the healthcare field, it must be HIPPA compliant, carefully examining each tech stack to ensure that it does not violate any federal law. The HIPPA law includes maintaining the privacy of the patients, protecting the information of the patients, and keeping the patients safe at all costs. All hashed user information is stored in the database, and all connections between servers are encrypted. Although the codebase is private, the public page of the framework can be found (https://psychosisireach.uw.edu/).

The framework mentioned in the implementation is self-sustaining, meaning it does not rely on any paid third-party tools or service-based environments such as Firebase or various AWS services besides GitHub. Therefore, the upkeep cost solely depends on the hardware and resources the host is willing to provide for the site.

The framework's scale is relatively grand, with many components and technical outcomes to consider. Therefore, it is crucial not to be complacent early in the development process. The goal of this framework is to prove that technology, specifically machine learning and artificial intelligence, has the capacity and practicality to be used in healthcare. The iREACH psychosis is one of the pioneers to push technology into giving greater accessibility to those seeking to learn or improve their mental health conditions.

Acknowledgment

We would also like to thank all the additional team members that were part of the development of the iREACH framework. The student members are Victor Chee, Skye Tran, Victor Chee, Yinzhou Wang, Serene Gao, Mia Wang, Myra Divina, and Rene Gomez.

Lastly, we would like to thank the University of Washington Garvey Institute for Brain Health Solutions for funding this R&D effort.

References

[1] Hennekens, C. H., Hennekens, A. R., Hollar, D., & Casey, D. E. (2005). Schizophrenia and increased risks of cardiovascular disease. American heart journal, 150(6), 1115-1121.

[2] Health NIoM. Mental Illness 2020; https://www.nimh.nih.gov/

[3] Stowkowy J, Addington D, Liu L, Hollowell B, Addington J. (2012) Predictors of disengagement from treatment in an early psychosis program. 2012. Doi: 10.1016/j.schres.2012.01.027. PMID: 22336955

[4] George A. Keepers, Laura J. Fochtmann. (2020) The American Psychiatric Association Practice Guideline for the Treatment of Patients with Schizophrenia. Doi: 10.1176/appi.ajp.2020.177901. PMID: 32867516

[5] Melanie Claxton, Juliana Onwumere, Miriam Fornells-Ambrojo. (2017) Do Family Improve Outcomes in Early Psychosis? A Systematic Review and Meta-Analysis. Doi: 10.3389/fpsyg.2017.00371. PMID:28396643

[6] Committee ISMIC. (2017) The Way Forward: Federal Action for a System That Works for All People Living With SMI and SED and Their Families and Caregivers. Publication ID: PEP17-ISMICC-RTC

[7] Pitschel-Waltz G, Leucht S, Bauml J, Kissling W, Engel RR. *The effect of family interventions on relapse and rehospitalization in schizophrenia - a meta-analysis.* Schizophrenia bulletin. 2001;27(1):73-92

[8] Kopelovich**, S.,** Stiles, B., Monroe-DeVita, M., Hardy, K., Hallgren, K., & Turkington, D. (2021). Psychosis REACH: Effects of a brief CBT-informed training for family and caregivers of individuals with psychosis. *Psychiatric Services*. Doi: 10.1176/appi.ps.202000740. PMID: 34015942

[9] Bradley, C. S. (2019). Impact of training on use of debriefing for meaningful learning. Clinical Simulation in Nursing, 32, 13-19.

[10] Gopi Battineni, Nalini Chintalapudi, Francesco Amenta. (2020) Al ChatBot Design during an Epidemic like the Novel Coronavirus. *E-Health and Telemedicine Centre.* Doi: 10.3390/healthcare8020154. PMID: 32503298

[11] Tom Nadarzynski, Oliver Miles, Aimee Cowie, Damien Ridge. (2019) Acceptability of artificial intelligence (AI)-led chatbot services in healthcare: A mixed-methods study. Doi: 10.1177/2055207619871808/ PMID: 31467682

[12] Buck, B., Chander, A., Monroe-DeVita, M., Cheng, S. C., Stiles, B., & Ben-Zeev, D. (2021). Mobile health for caregivers of young adults with early psychosis: a survey study examining user preferences. *Psychiatric Services*, 72 (8), 955-959. https://doi.org/10.1176/appi.ps.202000193

[13] Cheng, S.C., Backonja U, Buck B, Monroe-DeVita M, & Walsh E. (2020) Facilitating pathways to care: A qualitative study of the self-reported needs and coping skills of caregivers of young adults diagnosed with early psychosis. *Journal of Psychiatric and Mental Health Nursing, 27*(4), 368-379. https://doi.org/10.1111/jpm.12591

[14] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, Alan Nichol. Rasa: Open Source Language Understanding and Dialogue Management. *NIPS Workshop on Converstaional AI*. https://doi.org/10.48550/arXiv.1712.05181