

**INTERNATIONAL INSTITUTE OF INFORMATION
TECHNOLOGY BANGALORE**

**PROJECT REPORT
ON**

MENTAL HEALTH ASSISTANT CHATBOT

Submitted by

KAVYA K PURANIK - MT2017060

MINNU ANN ABRAHAM - MT2017069

SAI GOWRI KUMAR -MT2017097

Under the guidance of

Prof. DIVYA RAJ

**(Principal Technical Strategist in the E-Health Research Centre (EHRC)
& Adjunct faculty)**



DEPARTMENT OF INFORMATION TECHNOLOGY

IIIT BANGALORE

DECEMBER 2018

Acknowledgements

We are grateful to our project guide, Prof. Divyaraj and our Dean, Dr. R Chandrashekhar, for their constant support and guidance through the duration of the project.

We would especially like to extend our gratitude to NIMHANS for coming up with this innovative project and giving us an opportunity to work on this chatbot.

We also would like to thank Mr. Ravi Shankar Chaudhary for giving us technical guidance and direction. A special mention to the PUSH-D team for helping us integrate the project with PUSH-D.

We are thankful to the college and the Director - Mr. S. Sadagopan for the opportunity that was given to us to pursue a project in a field of our choice.

A big thank you to all the teaching and non-teaching staff of the IIIT, Bangalore for their help over the course of our M.Tech course.

Abstract

Mental health is a level of psychological well-being or an absence of mental illness. It is the "psychological state of someone who is functioning at a satisfactory level of emotional and behavioural adjustment". It includes our emotional, psychological, and social well-being. It affects how we think, feel, and act. It also helps determine how we handle stress, relate to others, and make choices. Mental health is important at every stage of life, from childhood to adolescence through adulthood.

However, due to the stigma that often surrounds mental health, millions of people worldwide do not receive the help they need and often overlook this extremely prevalent health issue. Even with access to capable health care and social services, those afflicted with mental health conditions hesitate to avail the treatment due to this stigma surrounding the illness.

In response, this project aims to implement an artificial intelligence application - a chatbot. A chatbot is a computer program designed to simulate conversation with human users, which will be available over the internet, targeted to be the first line of support for mental health patients, while maintaining their privacy and anonymity.

This application is developed to proactively check on patients, be ready to listen and chat anytime, anywhere, and recommend activities that improve the users well being. It is not a replacement for a therapist and does not try to be.

Contents

1	Introduction	1
2	Scope of Work	1
3	PUSH-D application	1
4	Architecture and design of system	2
5	Key features	4
6	Key learning and Challenges	5
7	Code snippets	6
8	Screenshots	8
9	Future Enhancements	10
10	Conclusion	10

1 Introduction

Only a few years ago, visiting a mental health professional or even acknowledging the fact that one may have issues related to mental health had a large amount of stigma surrounding it. Fortunately, this has largely changed in the past decade or so. Yet, the fact remains that a person may assume that everything is fine with them even though they might be experiencing the beginnings of a problem, thus causing the problem to grow, which could easily have been avoided. The mental health chatbot (MeHA) that has been developed is the first step towards rectifying this. It helps a person identify problems, if any, which they may have ignored otherwise, and to resolve it as early as possible. The person is prompted to get professional help at the earliest if they show any symptomatic signs that MeHA is programmed to identify. In extreme cases where a person may injure themselves or others, there is also a provision to contact authorities to prevent any untoward incidents.

2 Scope of Work

The chatbot is developed to answer users' questions related to mental health especially regarding depression. It is a platform where people can freely interact and need not be hesitant. It is not intended to handle any query on a topic other than mental health and is not a replacement for a therapist.

3 PUSH-D application

The chatbot is integrated with the PUSH-D (Practice and Use Self-Help for Depression) application which is supported on an ongoing basis by the team at the NIMHANS Data Center. PUSH-D is a computer-based self-care program which is meant to empower individuals by offering a space to learn, strengthen and practice psychological skills that can be useful in dealing with depressive symptoms. Expectations from Chatbot integrated with PUSH-D:

- Self-driven and yet assisted : This program is primarily self-driven but the chatbot helps to keep up the motivation of the user along with PUSH-D through periodic mobile prompts and emails.
- Confidentiality : This program can be accessed in the privacy of your own chosen place. User will log-in using assigned user id and password. The information that provided by user while going through the program will be stored in a secure server. Identity of user will not be revealed to anyone other than the resource-team.
- Evaluation and feedback : The online evaluation during the registration process is meant to provide a basic/crude feedback to the user about their depressive symptoms and functioning, so that they may choose an appropriate course of action.

4 Architecture and design of system

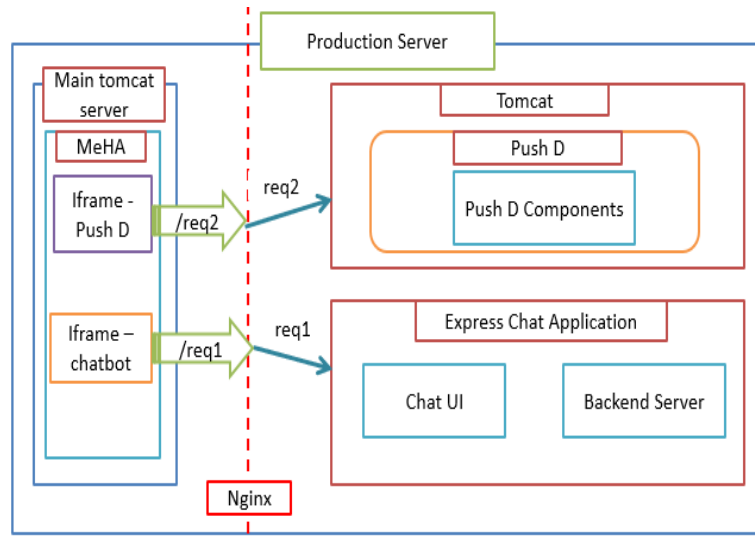


Figure 1: **The Big picture**

PUSH-D and chatbot are 2 independent applications brought together using MeHA project. User requests are made to main Tomcat server (MeHA), which contains 2 iframes - one for PUSH-D and the other for the Chatbot. Since PUSH-D and chatbot run on 2 different servers, (Tomcat and Express respectively) in order to get common base URL, nginx server is used.

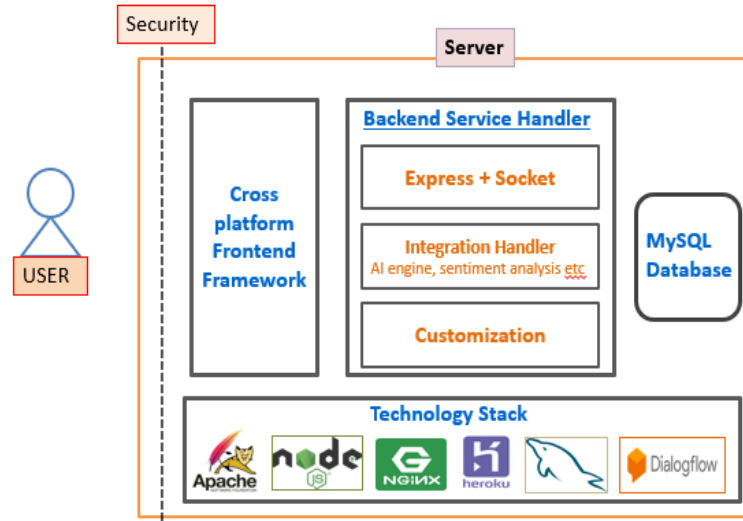


Figure 2: **Overall architecture of the system**

Overall architecture includes front-end, back-end and database layer architecture. The idea is to develop a minimum viable chatbot framework. Front-end uses HTML 5, CSS, Bootstrap, jQuery and JavaScript to facilitate cross OS support. Back-end is developed using NodeJS. REST API calls are made from the back-end to PUSH-D and webhook. Database is MySQL and the AI Engine is

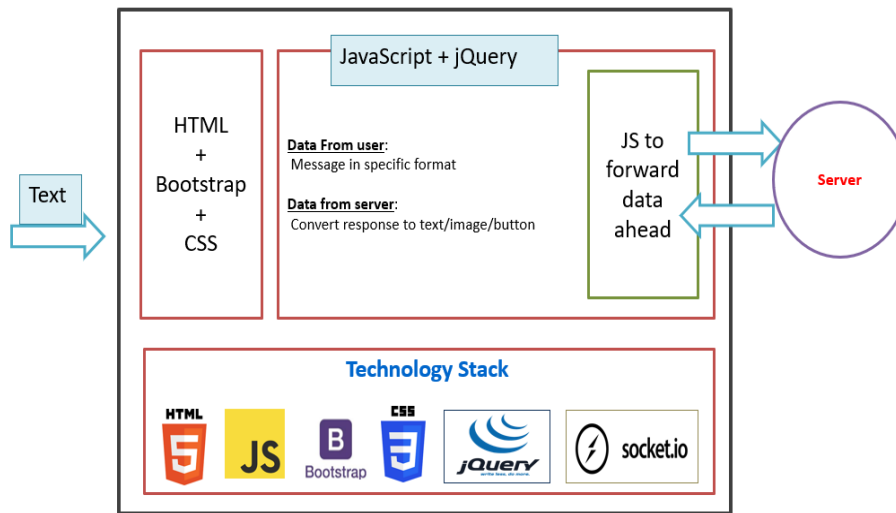


Figure 3: **Front-end architecture of the system**

Dialogflow. Servers deployed are Nginx, Apache Tomcat and Express server.

Front-end architecture has components of HTML, CSS and Bootstrap. They are used to build Chatbot UI - Chatbox. JavaScript and jQuery are used to facilitate -

- 1) Formatting outgoing message
 - 2) Provide interface to convert json to button and links
- Front-end will pass the request to back-end.

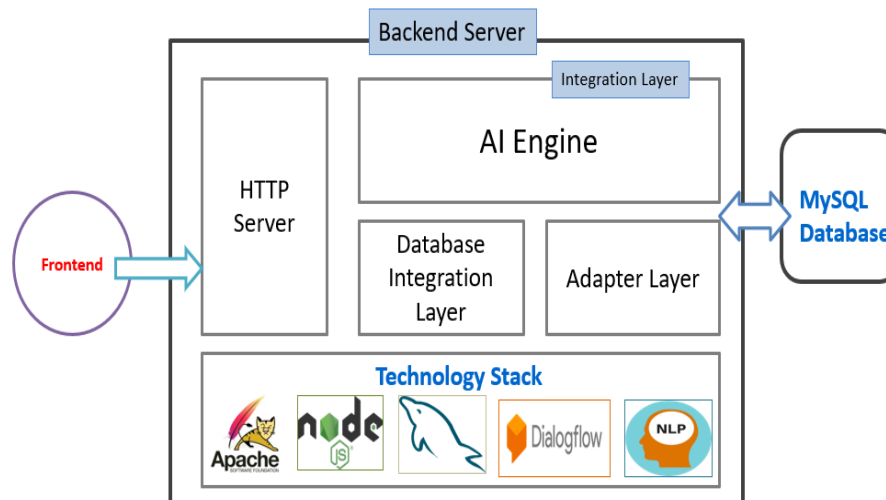


Figure 4: **Back-end architecture of the system**

Back-end server has 4 major components:

- 1) Exposed HTTP server To take incoming request
- 2) Integration Layer It contains AI engine
- 3) Database Layer To handle DB interaction
- 4) Adapter Later To handle post AI operation and customize replies

5 Key features

Key features of the Mental Health Assistant Chatbot are as follows:

- The chatbot provides a simulated conversation regarding mental health which includes:
 - Evaluation of mental wellness of user using standard World Health Organisation (WHO) questionnaire
 - Evaluation of depression level
 - Remedy for any problems faced by the user
 - Lightening mood of user by suggesting approved list of music, books and videos
 - Details about mental health institutions and doctors nearby to user using user's browser geolocation
 - Providing information about illnesses
- Conversational flow along with context is maintained with the user, allowing them to either enter free-text or choose from provided options.
- Conversation flow for PUSH-D registered users is different from that of un-registered users.
- For every user, information collected during the conversations is stored as history.
- Security aspects of chatbot include:
 - Privacy of users is maintained and their information is not disclosed to anyone except the doctor registered with the chatbot.
 - Users can also choose to remain anonymous during the conversation.
 - Registration and log-in of the user can be done only through PUSH-D application.
 - In cases where email is needed from unregistered users, OTP verification is done through e-mail.
- Sentiment Analysis is done on few messages to assess the mood of the user. According to the analysis, the further flow is decided.
- In cases where the user is showing suicidal tendencies, an emergency notification is immediately sent via e-mail to a registered doctor.
- At the end of every conversation, rating and feedback is collected from the user to help improve the chatbot as well as PUSH-D application.
- A report is generated daily summarizing the conversation details with the users that interacted with the chatbot. It also includes the feedback provided for the chatbot and related applications by various users.

- All free-text messages collected from users' chat are sent to the chat administrator at the end of each day. This is used as the data to train and improve the chatbot.

6 Key learning and Challenges

Key learning from this project is as follows:

- A chatbot UI was needed to be built from scratch and be linked with Dialogflow where Natural Language Processing was done and messages were modified to create a conversation-like flow.
- Custom payload feature of Dialogflow was used to add buttons, links and customized replies to messages sent by the user.
- Maintaining context in the conversation was an important learning. It enabled continuity of the conversation - even by using pronouns and not the noun over and over.
- Collection and aggregation of data via chat with user was done and it was persisted to a database to maintain history of the users. This prevents same questions from being asked to a returning user.
- Integration of APIs and usage of features like browser geolocation was another learning.
- Dialogflow requests were handled in NodeJS code. The requests were processed in NodeJS and appropriate responses were displayed to user.
- Sentiment analysis was an important feature that was learned where the flow of the conversation was determined using the score given by this analysis.

Challenges faced in the duration of the project were as follows:

- Choosing the appropriate technologies to deliver this chatbot was a challenge. Usage of various languages like Python, Java, NodeJS were an option and it was time consuming to test and select the perfect fit.
- Integration of the chatbot with the PUSH-D application was a challenge due to the cross-origin domains. Also, since the chatbot is a part of PUSH-D, it needed to use the log-in service of PUSH-D. This was difficult as access to PUSH-D database was not available and APIs had to be created for this purpose.
- Chatbot was integrated in PUSH-D as an iframe and this prevented the browser geolocation access in the chatbot. This was solved by creating an event for the same.
- Already available libraries for Sentiment Analysis were not sufficient for this project so appropriate modifications were made to fit it to our needs. Also, free-text messages that were not correctly recognized by the library are sent to chat administrator to manually train this library.

- Since the chatbot and PUSH-D applications were running on 2 different servers(ports), Express and Tomcat respectively, providing a common base URL for the two was also a challenge. This was solved by creating a reverse proxy provided by Nginx server.

7 Code snippets

Server side code snippet

```
var apiai = require('apiai');

//Enter API Key
var app = apiai(apikey);

//returns speech from api.ai
var getRes = function(query,options)
{
    var request = app.textRequest(query, options);
    const responseFromAPI = new Promise(
    function (resolve, reject)
    {
        request.on('error', function(error)
        {
            reject(error);
        });
        request.on('response', function(response)
        {
            resolve(response);
        });
    });
    request.end();
    return responseFromAPI;
};

module.exports = {getRes}
```

Client side code snippet

```
<body class="chat-main">
  <div style="background-color: #fcf8e3;text-align:center" class="
    col-sm-12 username pl-2">
    <span class="text-muted">
      <small>      To restart the conversation, click on the
        HOME icon</small>
    </span>
  </div>
  <div class="col-md-12 chat-header bg-primary text-white">
    <div style="height:25px;padding-top:3px" class="row">
      <div class="col-sm-9 username pl-2">
        <i class="fa fa-circle text-success" aria-hidden="
          true"></i>
        <span>NIMHANS Mental Health Assistant</span>
      </div>
      <div class="col-sm-3 options text-right pr-2">
        <i style="color:white;font-size:15px;padding-bottom:3
          px" class="tooltipicon" onclick="home()">
          <span class="fa-fw select-all fas"> </span>
          <span class="tooltiptext" style="white-space:
            nowrap;">Home</span>
        </i>
        <i style="color:white;font-size:15px;padding-bottom:3
          px" class="tooltipicon" onclick="usefulLinks()">
          <span class="fa-fw select-all fas"> </span>
          <span class="tooltiptext" style="white-space:
            nowrap;">Useful Links</span>
        </i>
        <i id="minimize-icon hide-chatbox" class="fa fa-
          window-minimize" aria-hidden="true"></i>
        <i id="maximize-icon hide-chatbox" style="display:
          none;" class="fa fa-window-maximize" aria-hidden="
          true"></i>
      </div>
    </div>
  </div>
  <div class="chat-content">
    <div id="chat-scroll" class="col-md-12 chats border">
      <ul id="response" class="p-0">
      </ul>
    </div>
    <div class="col-md-12 message-box border pl-2 pr-2 border-top
      -0">
      <input id="input" style="margin:0px;width:310px" type="
        text" class="row" placeholder="Type a message..." />
    </div>
  </div>
</body>
```

8 Screenshots

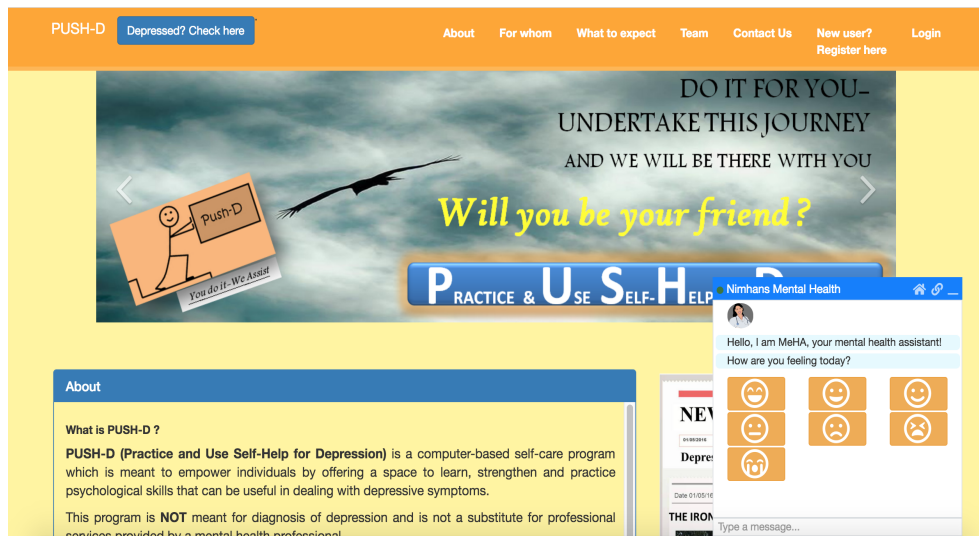


Figure 5: Chatbot integrated with PUSH-D - Unregistered User

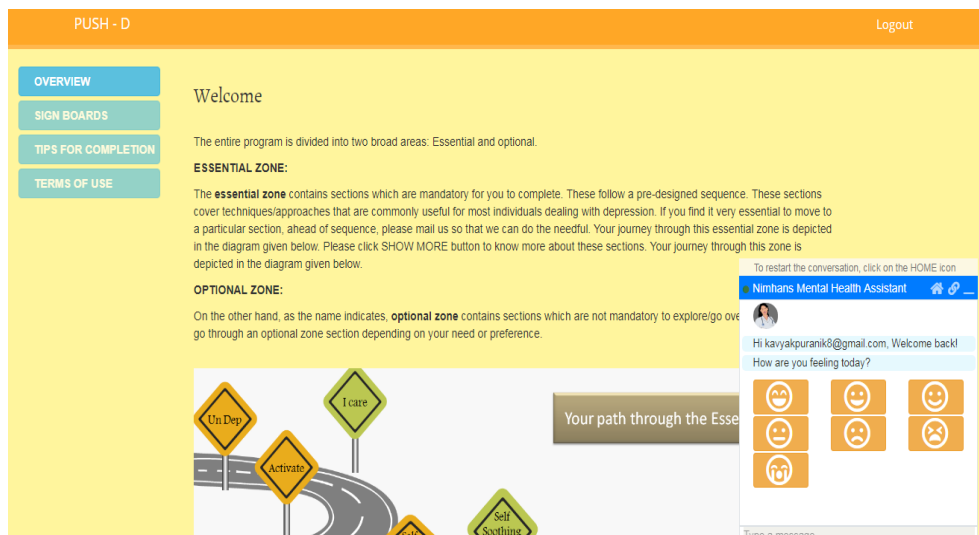
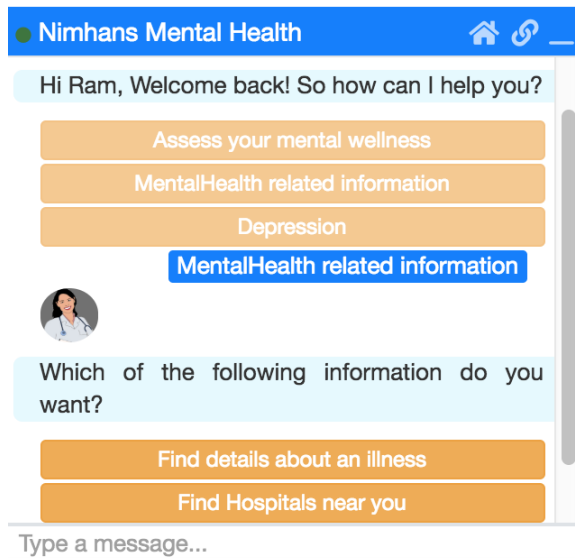
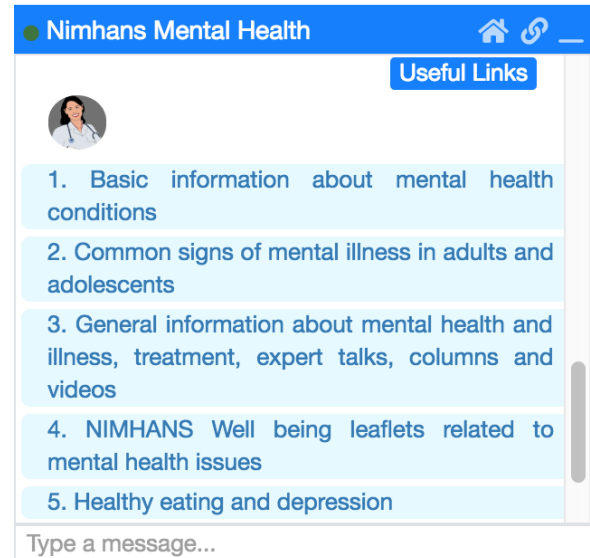


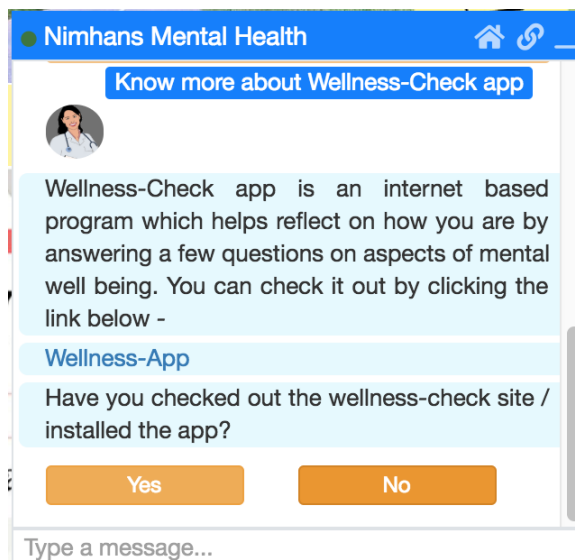
Figure 6: Chatbot integrated with PUSH-D - Registered User



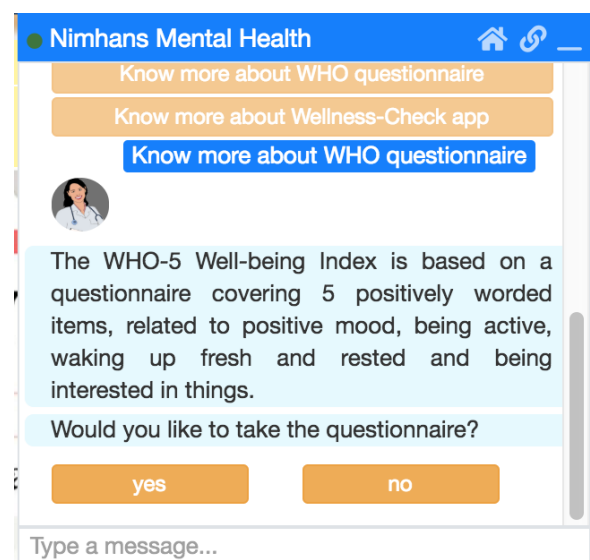
(a) Registered user returning to chatbot



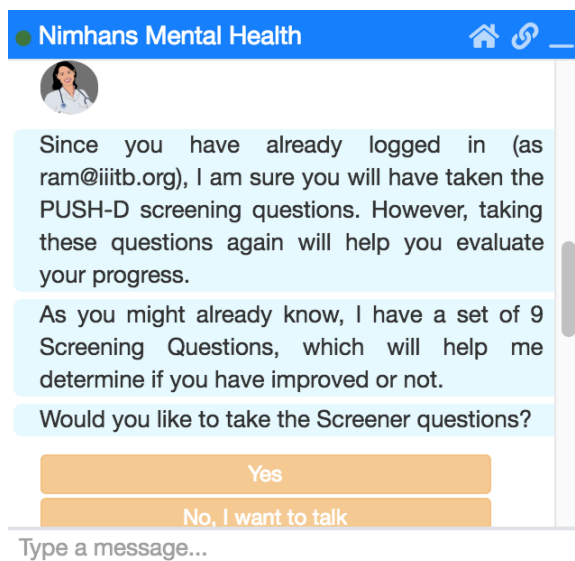
(b) Useful Links



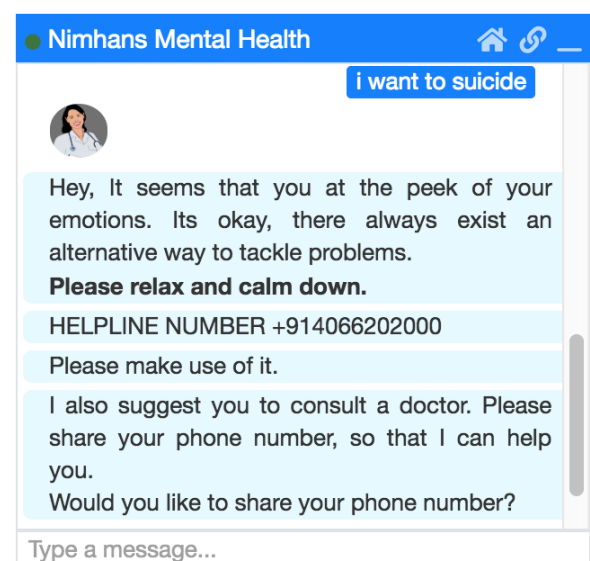
(c) Wellness-Check Application



(d) WHO questionnaire



(e) Screener start



(f) Depression check

Figure 7: Various screenshots

9 Future Enhancements

This project was developed in a short time-frame by a small team and hence, many potential enhancements are possible.

- The hospital dataset that is currently being used is limited and extension of the same would be very useful.
- The conversation flow has limited free-text entry points at this point in time. Handling these free-texts is not an easy task so finding a way to manage this and adding more free-text entry point to the flow would increase the user friendliness of the chatbot.
- Creation of chat admin screen could be done. Here, integration with any new application could be implemented for which a generalized interface could be developed where basic necessary details of the new app could be collected and incorporated accordingly in the chatbot flow.
- List of registered doctors is not available presently and integration of EHRC registered doctors database would make the chatbot more informative.
- The chatbot can process messages in English only right now. A potential improvement would be to include Indian regional languages like Hindi, Kannada, etc to make the chatbot usable by larger audience.
- Sentiment Analysis library has minimal rules currently. Adding complex rules and handling different scenarios would make the analysis more accurate and help users gain better insight on their condition. Using professional help, more effective progress would be made.
- Incorporating domain knowledge, better analysis and more useful replies could be provided.
- Users' history is being maintained in the database. Using this information for further analysis and comparison could help track the progress of each user and give them better advice.
- Sometimes, users can play around with the chatbot and raise unnecessary alarms. This situation is not handled and could cause some havoc. Keeping it under check would make the chatbot more efficient.

10 Conclusion

In this era where use of NLP and AI is growing, this project tries to make a realistic implementation of these technologies. Using various methods, it assesses the user's mental state and helps them gain insight on how to deal with their problems.

The chatbot aims to be a preliminary step in the mental health care field, where assistance is provided to users at any time. It provides a space for people to open up and vent their feelings.