PROJECT REPORT

ARTIFICIAL INTELLIGENCE

CSE-3013

FAITH AN ANTI-DEPRESSANT CHATBOT

BY

16BCE0765	RISHAB GUPTA
16BCE0782	SARTHAK DAHIYA
16BCE2157	SHARON SUSAN SHIBU

FACULTY- Annapurna Jonnalagadda

SLOT- B1+TB1

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

CERTIFICATE

This is to certify that the project work titled “FAITH AN ANTI-DEPRESSANT CHATBOT” that is being submitted by Rishab Gupta,Sarthak Dahiya,Sharon Susan Shibu for Artificial Intelligence (CSE-3013) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore
Date: April 10, 2019

Acknowledgements

We would first of all like to thank our faculty Annapurna Jonnalagadda, for her persistent motivation and guidance in regard to our project.

We also would like to thank the University Management and the Dean SBST for the successful completion of the project and giving us the opportunity to carry out our study at the University.

i.PROBLEM STATEMENT:

As per NHMS, in India, one in 20 people over 18 years of age have ever suffered from depression amounting to a total of over 45 million persons with depression in 2017. If we don't think, our anxiety, depression, sadness and stress will impact our physical health, think again. All of these emotions trigger chemical reactions in our body, which can lead to inflammation and a weakened immune system. This is where our Project "FAITH" kicks in.

Faith is an anti-depressant AI chat bot built with the sole purpose of cheering up users and help them decrease the number of health-related issues arising due to depression. Faith strives in performance and robustness, by fetching the latest songs, restaurants close to the user based on his taste. Faith keeps a tab on user's mind and churns out all the available resources close to him. The bots built in dialog flow automatically senses the mood of the user by his tone. The bot is intelligent enough and automatically changes its UI based on the user's mood.

The bot communicates and learns using inbuilt ML technologies and modules in dialog flow.

Till date, no similar app has been made and depression is a prevailing issue in today's world. It needs to be treated and talking with someone is the best way to subdue depressive or suicidal feelings according to research in the field. With this app, we aim to give everyone a cost-effective option to help them in difficult situations. It will not only suggest the person to take medical precautions when necessary but also suggest the person to otherwise indulge in activities based on their location. In a way, it is not just an anti-depressant Chabot but also helps people when they feel bored.

ii.Introduction

We don’t have a lot of digital tools and applications to cure depression. One application that we came across during the analysis of our project has been made by a Stanford researcher and psychologist, Alison Darcy.

Woebot, an artificial intelligence-powered chat bot designed using cognitive behavioural therapy is one of the most heavily researched clinical approaches to treat depression.

The results of the trial were published June 5 in the Journal of Medical Internet Research Mental Health. For the test, Darcy recruited 70 students who said they experienced symptoms of depression and anxiety and split them into two groups. One group spent two weeks chatting with Woebot; the other was directed to a National Institute of Mental Health e-book on depression. Over 14 days, people in the Woebot group not only reported chatting with the bot almost every day, but saw a significant reduction in their depressive symptoms.

That's a promising result for a type of treatment whose results have so far been tough to quantify. (We don't have a lot of research comparing bot-to-human therapy against traditional human-to-human therapy.)

Woebot uses a technique called cognitive behavioral therapy (CBT) to talk to patients, and several studies suggest the approach lends itself to being administered online. A review of studies published recently in the journal World Psychiatry compared people who received CBT online to people who did it in person, and found that the online setting was just as effective. One reason for this, according to Darcy, is that CBT focuses on discussing things that are happening in your life now as opposed to things that happened to you as a child. As a result, instead of talking to Woebot about your relationship with your mom, you might chat about a recent conflict at work or an argument you had with a friend. "A premise of CBT is it's not the things that happen to us, it's how we react to them," Darcy says. Woebot uses that methodology to point out areas where a person might be engaging in what's called "negative self-talk," which can result in seeing the environment around them in a distorted way and feeling bad about it.

If, for example, a friend forgot about your birthday, you might tell Woebot something like, "No one ever remembers me," or "I don't have any real friends."

Woebot might respond by pointing out that you're engaging in a type of negative self-talk called "all-or-nothing thinking," which is a distortion of reality. In reality, you *do* have friends and people *do* remember you. One of those friends simply forgot your birthday. "Self-talk is a part of being human," Darcy says. "But the kinds of thoughts that we have actually map onto the kinds of emotions we're feeling."

Darcy is quick to point out that Woebot is not a replacement for traditional therapy, but rather an addition to the toolkit of approaches to mental health.

i.Motivation

The statistics itself project the need to take up this project. There are more people, especially teenagers and young adults, entering into the stage of depression every year because of inadequate guidance or the feeling of insecurity in sharing their feelings with others. With this app, people will feel more comfortable in sharing their problems, because there is no person on the other end. Moreover, any kind of therapy even with a real human requires money while this is a free app, which anyone can use at any point in time. The bot is programmed to help divert the user from his/her concerns by giving the user some activities to indulge in. It is not just a regular chatbot but a chatbot enhanced to deal with cases of depression or loneliness.

ii.Scope and Applications of the Proposed System:

- An application which responds to the user based on his preferences.
- The bots built in dialog flow automatically senses the mood of the user by his tone. The bot is intelligent enough and automatically changes its UI based on the user's mood.
- Faith is an anti-depressant AI chat bot built with the sole intention of relieving users of their depression and thereby lightening the user's mood.
- The bot interacts with the user and when it sees any suicidal tendencies, it informs the user with nearby hospitals as well as nearby psychological clinics.
- Due to lack of digital tools and applications to cure depression, Faith becomes an exciting prospect even for all sorts of users.
- Dialog flow provides one-click integrations with such platforms as Facebook Messenger, Slack, Telegram, Amazon Alexa, Line. It helps to develop your bot for many platforms at once. Also, it has a set of SDKs for Node.js, Android, iOS, Python, etc., so you can use it not only for chat bots, but applications and devices.
- In truth, anything that exists as a web/mobile app can probably exist as a bot. It's just a different interface. As such, we still haven't truly explored all of its capabilities.

iii. Literature Survey:

We conducted a literature survey that gave us an idea of the existing approaches in today's world.

1. A Stanford researcher might be on the brink of a dramatic shift in how we treat depression.

We do know, however, that talking seems to help - especially under the guidance of a licensed mental health professional. But therapy is expensive, inconvenient, and often hard to approach. A recent estimate suggests that of the roughly one in five Americans suffering from mental illness, close to two thirds have gone at least a year without treatment. Several Silicon Valley-style approaches to the problem have emerged: There are apps that replace the traditional psychiatry office with texting, and chat-rooms where you can discuss your problems anonymously online. The newest of these tech-based treatments is Woebot, an artificial intelligence-powered chat bot designed using cognitive behavioural therapy, one of the most heavily-researched clinical approaches to treating depression. The results of the trial were published June 5 in the Journal of Medical Internet Research Mental Health. For the test, Darcy recruited 70 students who said they experienced symptoms of depression and anxiety and split them into two groups. One group spent two weeks chatting with Woebot; the other was directed to a National Institute of Mental Health e-book on depression. Over 14 days, people in the Woebot group not only reported chatting with the bot almost every day, but saw a significant reduction in their depressive symptoms.

2. Current research and trends in the use of smartphone applications for mood disorders. John Torous, Adam C. Powell.

Smartphone applications for mental illnesses offer great potential, although the actual research base is still limited. Major depressive disorder and bipolar disorder are both common psychiatric illness for which smartphone application research has greatly expanded in the last two years. We review the literature on smartphone applications for major depressive and bipolar disorders in order to better understand the evidence base for their use, current research opportunities, and future clinical trends. We conducted an English language review of the literature, on November 1st 2014, for smartphone applications for major depressive and bipolar disorders. Inclusion criteria included studies featuring modern smartphones running native applications with outcome data related to major depressive or bipolar disorders. Studies were organized by use of active or passive data collection and focus on diagnostic or therapeutic interventions. Our search identified 1065 studies. Ten studies on major depressive disorder and 4 on bipolar disorder were included. Nine out of 10 studies on depression related smartphone applications featured active data collection and all 4 studies on bipolar disorder featured passive data collection. Depression studies included both diagnostic and therapeutic smartphone applications, while bipolar disorder studies featured only diagnostics. No studies addressed physiological data.

3. Mobile, social, and wearable computing and the evolution of psychological practice.

Psychological assessment and intervention are extending from the clinic into daily life. Multiple forces are at play: Advances in mobile technology, constrained clinical care, and consumer demand for contextualized, nonstigmatizing, and low-cost alternatives are beginning to change the face of psychological assessment and interventions. Mobile, social, and wearable technologies are now enabling individuals to measure themselves and to integrate myriad forms of help and entertainment. The massive data sets generated by self- tracking of mood and passive sensing of voice, activity, and physiology may eventually reorganize taxonomies of mental health concerns. Compelling mobile therapies will also emerge, involving contextually appropriate, entertaining, and dynamic feedback to provide help in the context of daily life. The efficacy of such applications will be tested through citizen science as well as clinical trials. This article reviews technical advances that can be applied to enhance assessment and intervention and dramatically increase access to psychotherapy. It is recommended that, in addition to exploring clinically oriented products, practitioners should support patients' use of direct-to-consumer applications in ways that align with therapeutic objectives.

4. Can Mobile Phone Apps Influence People's Health Behaviour Change? An Evidence Review.

Globally, mobile phones have achieved wide reach at an unprecedented rate, and mobile phone apps have become increasingly prevalent among users. The number of health-related apps that were published on the two leading platforms (iOS and Android) reached more than 100,000 in 2014. However, there is a lack of synthesized evidence regarding the effectiveness of mobile phone apps in changing people's health-related behaviors. The aim was to examine the effectiveness of mobile phone apps in achieving health-related behavior change in a broader range of interventions and the quality of the reported studies. We conducted a comprehensive bibliographic search of articles on health behavior change using mobile phone apps in peer-reviewed journals published between January 1, 2010 and June 1, 2015. Databases searched included Medline, PreMedline, PsycINFO, Embase,

Health Technology Assessment, Education Resource Information Center (ERIC), and Cumulative Index to Nursing and Allied Health Literature (CINAHL). Articles published in the *Journal of Medical Internet Research* during that same period were hand-searched on the journal's website. Behavior change mechanisms were coded and analyzed. The quality of each included study was assessed by the Cochrane Risk of Bias Assessment Tool.

5. The CES-D Scale

A Self-Report Depression Scale for Research in the General Population.

The *CES-D* scale is a short self-report scale designed to measure depressive symptomatology in the general population. The items of the scale are symptoms associated with depression which have been used in previously validated longer scales. The new scale was tested in household interview surveys and in psychiatric settings. It was found to have very high internal consistency and adequate test- retest repeatability. Validity was established by pat terns of correlations with other self-report measures, by correlations with clinical ratings of depression, and by relationships with other variables which support its construct validity. Reliability, validity, and factor structure were similar across a wide variety of demographic characteristics in the general population samples tested. The scale should be a useful tool for epidemiologic studies of depression.

6. National Institute of Mental Health Treatment of Depression Collaborative Research Program General Effectiveness of Treatments.

M. Tracie Shea, PhD; John T. Watkins, PhD.

We investigated the effectiveness of two brief psychotherapies, interpersonal psychotherapy and cognitive behavior therapy, for the treatment of outpatients with major depressive disorder diagnosed by Research Diagnostic Criteria. Two hundred fifty patients were randomly assigned to one of four 16-week treatment conditions: interpersonal psychotherapy, cognitive behavior therapy, imipramine hydrochloride plus clinical management (as a standard reference treatment), and placebo plus clinical management. Patients in all treatments showed significant reduction in depressive symptoms and improvement in functioning over the course of treatment. There was a consistent ordering of treatments at termination, with imipramine plus clinical management generally doing best, placebo plus clinical management worst, and the two psychotherapies in between but generally closer to imipramine plus clinical management. In analyses carried out on the total samples without regard to initial severity of illness (the primary analyses), there was no evidence of greater effectiveness of one of the psychotherapies as compared with the other and no evidence that either of the psychotherapies was significantly less effective than the standard reference treatment, imipramine plus clinical management.

ii. & iii.Gaps Identified & Drive to the present work

Existing Systems that incorporate chatbots do not have mechanisms to deal with depression. Other regular chatbots do not take initiative to know what kind of day the user had or do not take the initiative to suggest activities they can take part in, if they are feeling low. These systems would only suggest them restaurants when the user explicitly asks it to. FAITH on the other hand, will understand the tone of the user through its conversations and try to suggest activities that will help him improve his mood.

Other chatbots' reply:

User: 'Suggest me some nearby restaurants'
Chatbot: 'There is a great Chinese restaurant within 2 km ...'

FAITH's reply:

User: 'I am very angry at the moment!'
Chatbot: 'Here is a list of your favourite songs... Maybe later you can have dinner at the new Chinese restaurant opened up just within 2 km ...'

Stanford has made a chatbot that works on cognitive behavioural therapy. However, this chatbot only points out negative self-talks and warns the user not to repeat these mistakes as it can be harmful for their health. However, there is no mention about what a user can do if they are in depression. It might aware the user of their critical condition, but our chatbot might take further steps to improve the user's condition, while at the same time making the user's relatives aware about the user's condition. This chatbot is also like any other chatbot. It performs all the other actions that other chatbots perform like setting reminders, calling someone etc.

iv.Implementation

Software details

1. Xcode

Xcode is an integrated development environment (IDE) for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS. First released in 2003, the latest stable release is version 10.0 and is available via the Mac App Store free of charge for macOS High Sierra users.Registered developers can download preview releases and prior versions of the suite through the Apple Developer website.

2.Dialogflow

Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant (by Speaktoit), a virtual buddy for Android, iOS, and Windows Phonesmartphones that performs tasks and answers users'

question in a natural language. Speaktoit has also created a natural language processing engine that incorporates conversation context like dialogue history, location and user preferences.

i.Proposed System

Overview of the Proposed system

Since chat bots are fundamentally virtual robots they never get worn out and keep on obeying your order. They will keep on operating each day during the time without any breaks. This enhances your client UX and causes you rank profoundly in your division.
Faith is an anti-depressant AI chat bot built with the sole purpose of cheering up users and help them decrease the number of health-related issues arising due to depression. Faith strives in performance and robustness, by fetching the latest songs, restaurants close to the user based on his taste. Faith keeps a tab on user's mind and churns out all the available resources close to him. The bots built in dialog flow automatically senses the mood of the user by his tone. The bot is intelligent enough and automatically changes its UI based on the user's mood.

The bot communicates and learns using inbuilt ML technologies and modules in dialog flow.

System Architecture

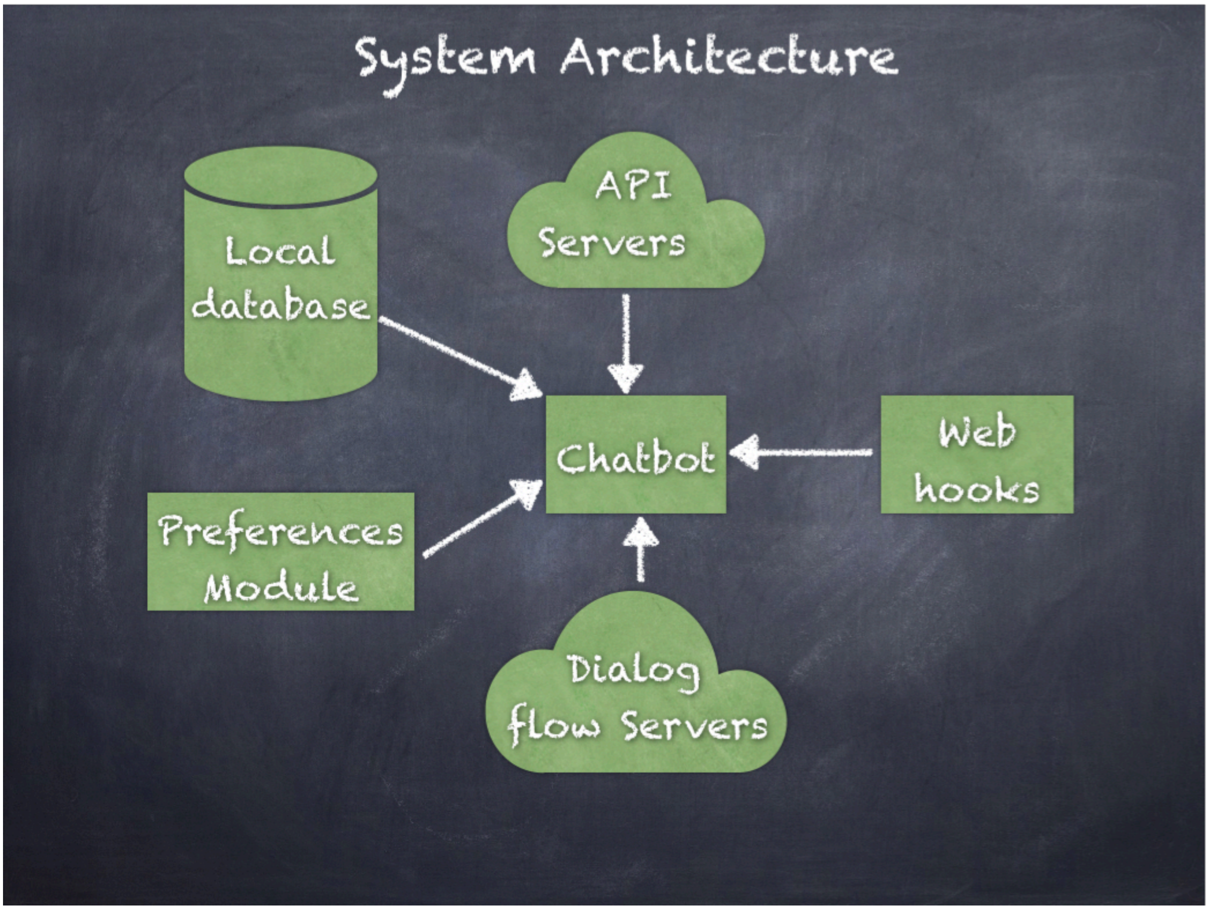


Fig. 1: System Architecture of the proposed system

Fig. 1 represents the main modules of the system in the form of a system architecture. The system architecture involves all the connections with the chatbot. The Chatbot is central and interacts with all the other modules. It is connected with the local database to store the preferences. It is connected with the API servers to fetch user wanted data. It also has connections with web hooks which get data from the external APIs. DialogFlow servers are used for text to speech and speech to text conversion and understanding of user natural language query.

Functional Architecture

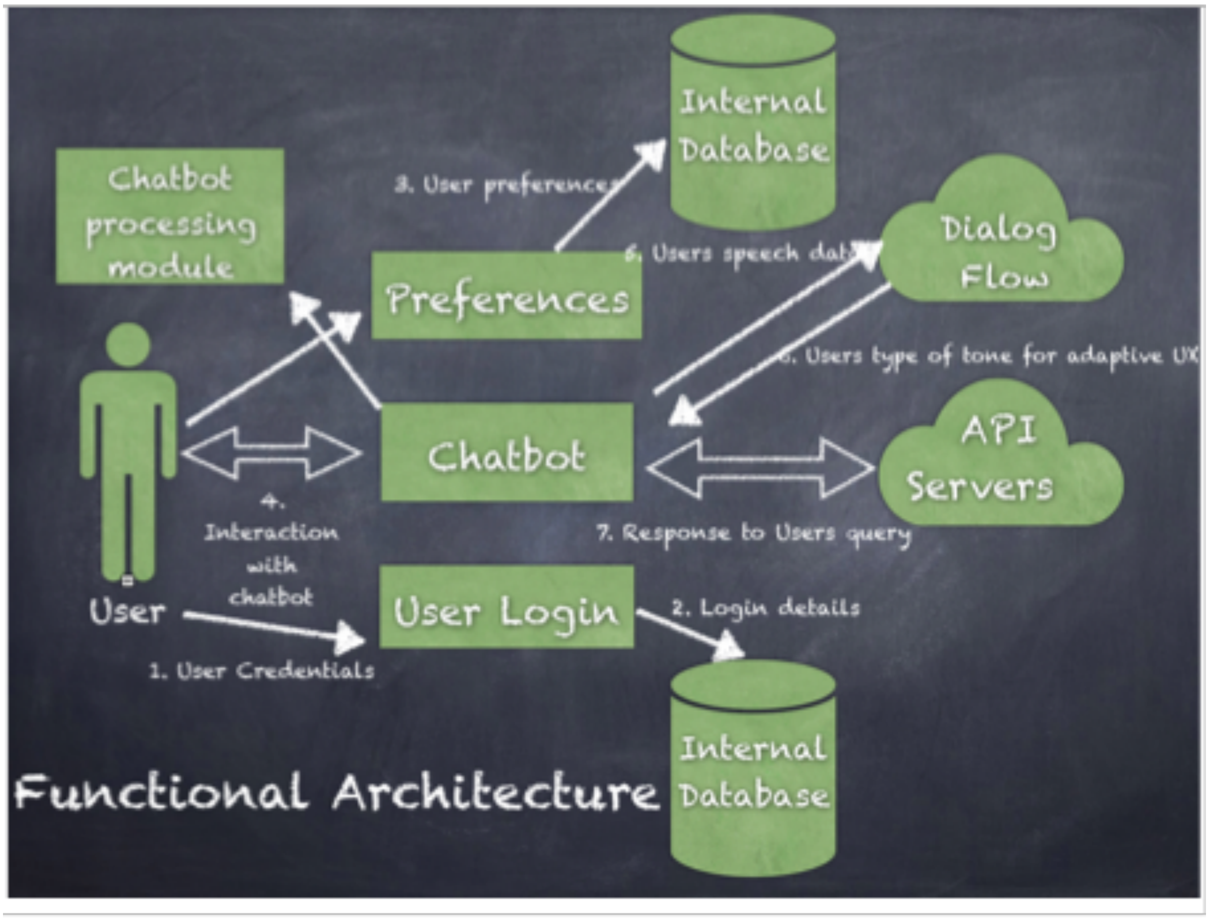


Fig. 2: Functional Architecture of the proposed system

Fig. 2 represents the functional architecture which highlights the major functions of all the modules. The Preferences module will gather the user preferences and store them in an internal database. The chatbot will interact with the user and send the user query to dialog flow servers for understanding the user's natural language. After chatbot understands the query and processes it using the chatbot processing module, it will fetch corresponding data from API servers and form the response and give it to the user. The User must also login before he/she can interact with the user so that the correct preferences are loaded by the system.

Modular Design

1. Data Flow Diagram

A data flow diagram is a graphical representation of the flow of data through theproposed system and modelling its process aspects.

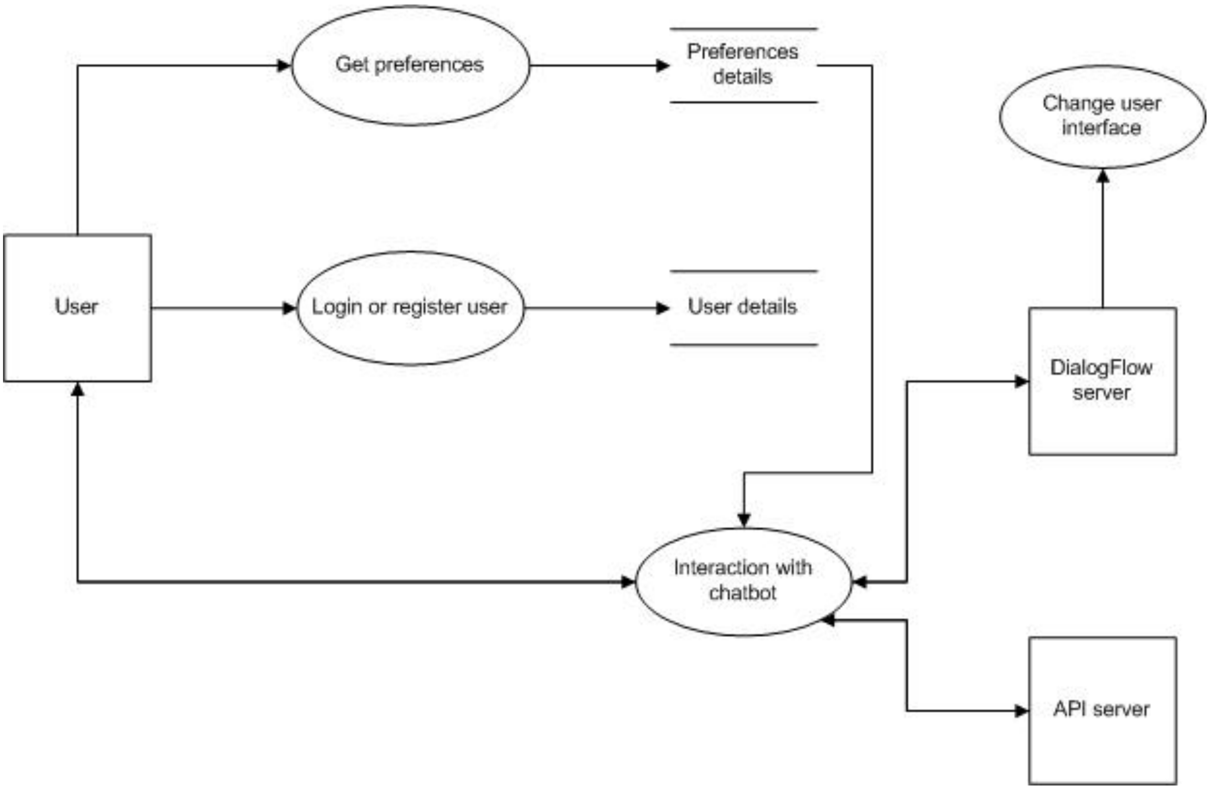


Fig. 3: Level 1 DFD diagram

Fig. 3 shows the level 1 DFD which notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

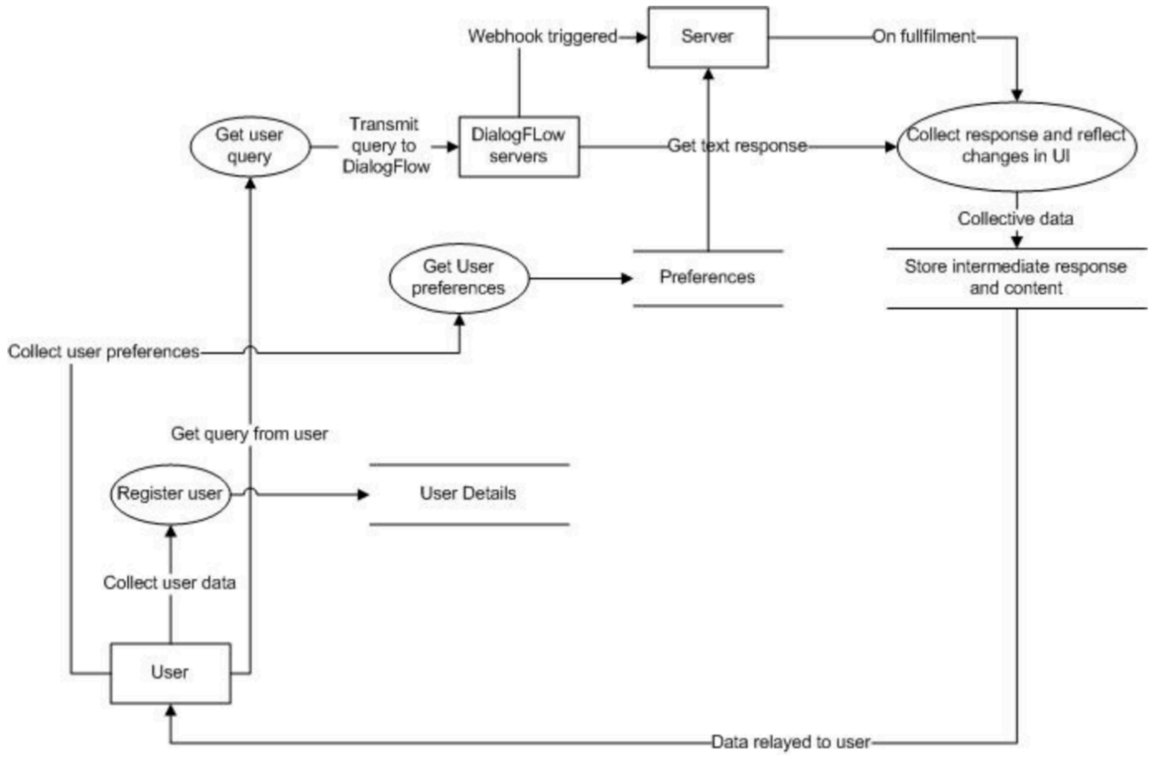


Fig. 4: Level 2 DFD diagram

Fig. 4 represents the level 2 DFD which is an expanded context level diagram for each process in level 1 DFD.

2. Activity Diagram

It is another important diagram in UML to describe the dynamic aspects of the

system. Activity diagram is basically a flowchart to represent the flow from

one activity to another activity.

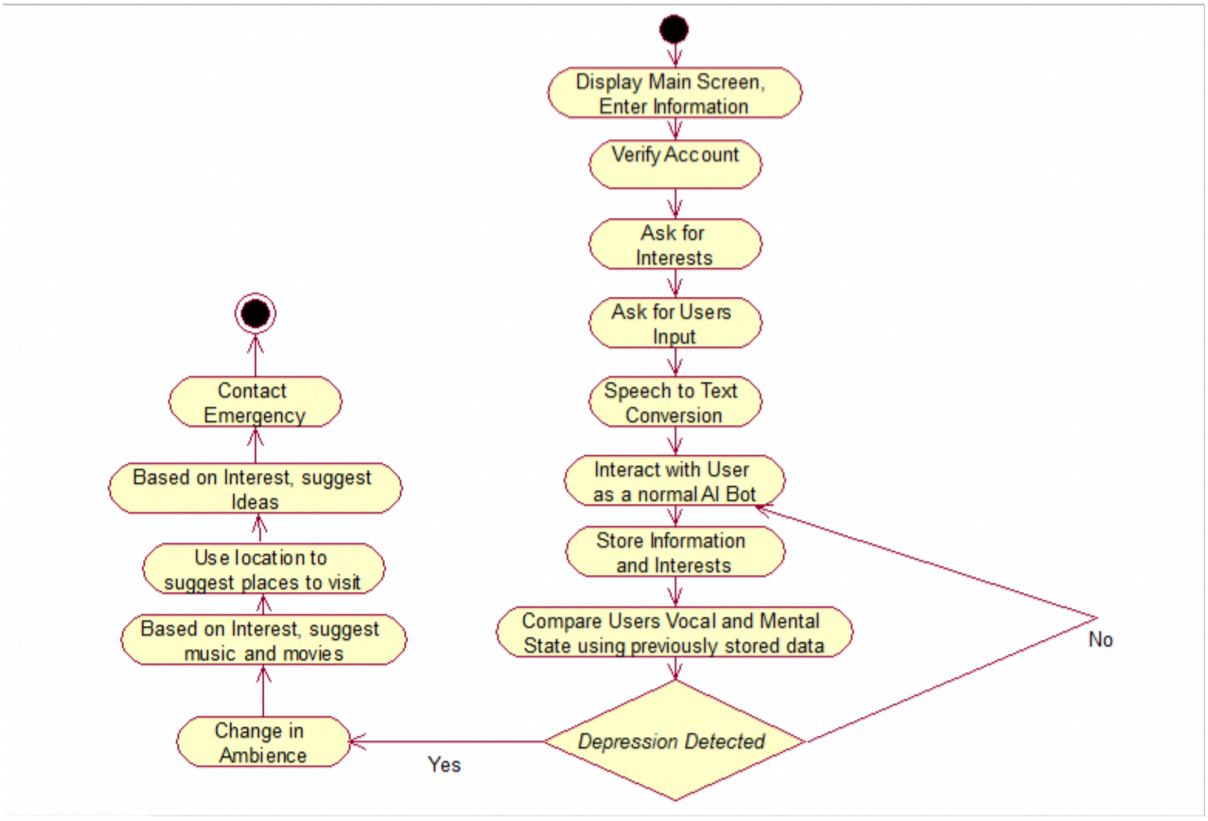
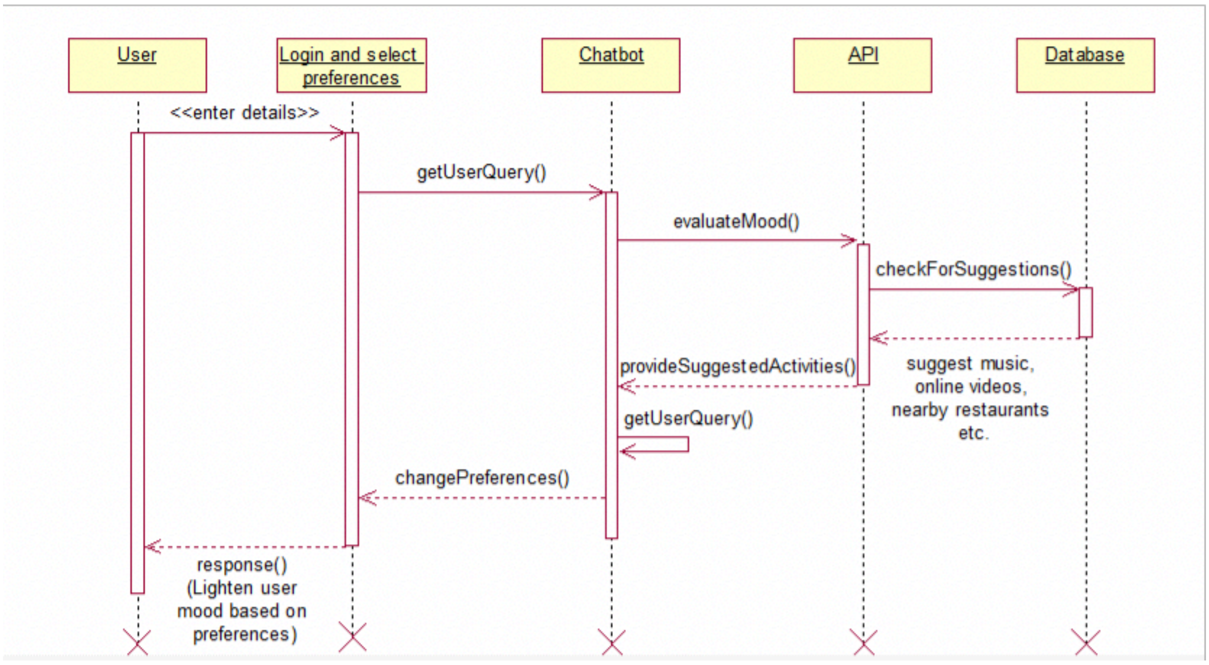

Display Main Screen, Enter Information
Verify Account
Ask for Interests
Ask for Users Input
Speech to Text Conversion
Interact with User as a normal AI Bot
Store Information and Interests
Compare Users Vocal and Mental State using previously stored data
Depression Detected
No
Yes
Change in Ambience
Based on Interest, suggest music and movies
Use location to suggest places to visit
Based on Interest, suggest Ideas
Contact Emergency

Fig. 5: Activity Diagram of the proposed system

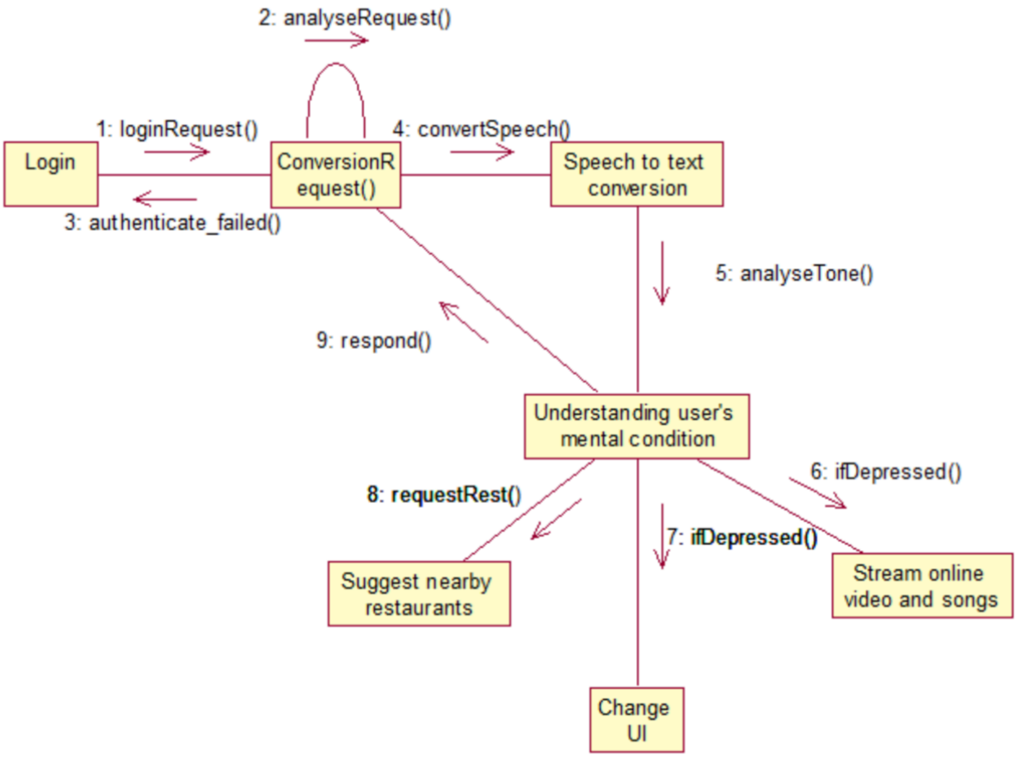
Fig. 5 represents the Activity Diagram which describes the dynamic aspects of a proposed system. It signifies the flow from one activity to another.

3. Sequence Diagram

It shows object interactions arranged in time sequence. Also, it depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.


User
Login and select preferences
Chatbot
API
Database
<<enter details>>
getUserQuery()
evaluateMood()
checkForSuggestions()
suggest music, online videos, nearby restaurants etc.
provideSuggestedActivities()
getUserQuery()
changePreferences()
response() (Lighten user mood based on preferences)

4. Collaboration diagram



It illustrates the relationships and interactions among software objects in the UML.

Fig. 7: Collaboration diagram of the proposed system.

5. Class Diagram

It describes the structure of a system by showing the systems classes, their attributes, operations, and the relationship among objects.

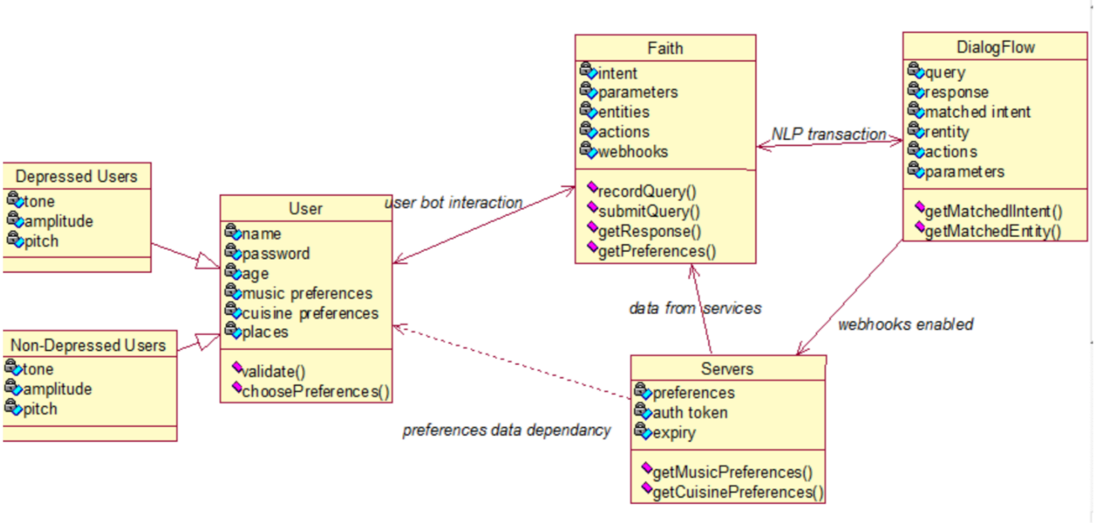


Fig. 8: Class Diagram of the proposed system

6. Use Case Diagram

It is a representation of the user's interaction with the system and shows the relationship between the user and the different use cases in which user is involved.

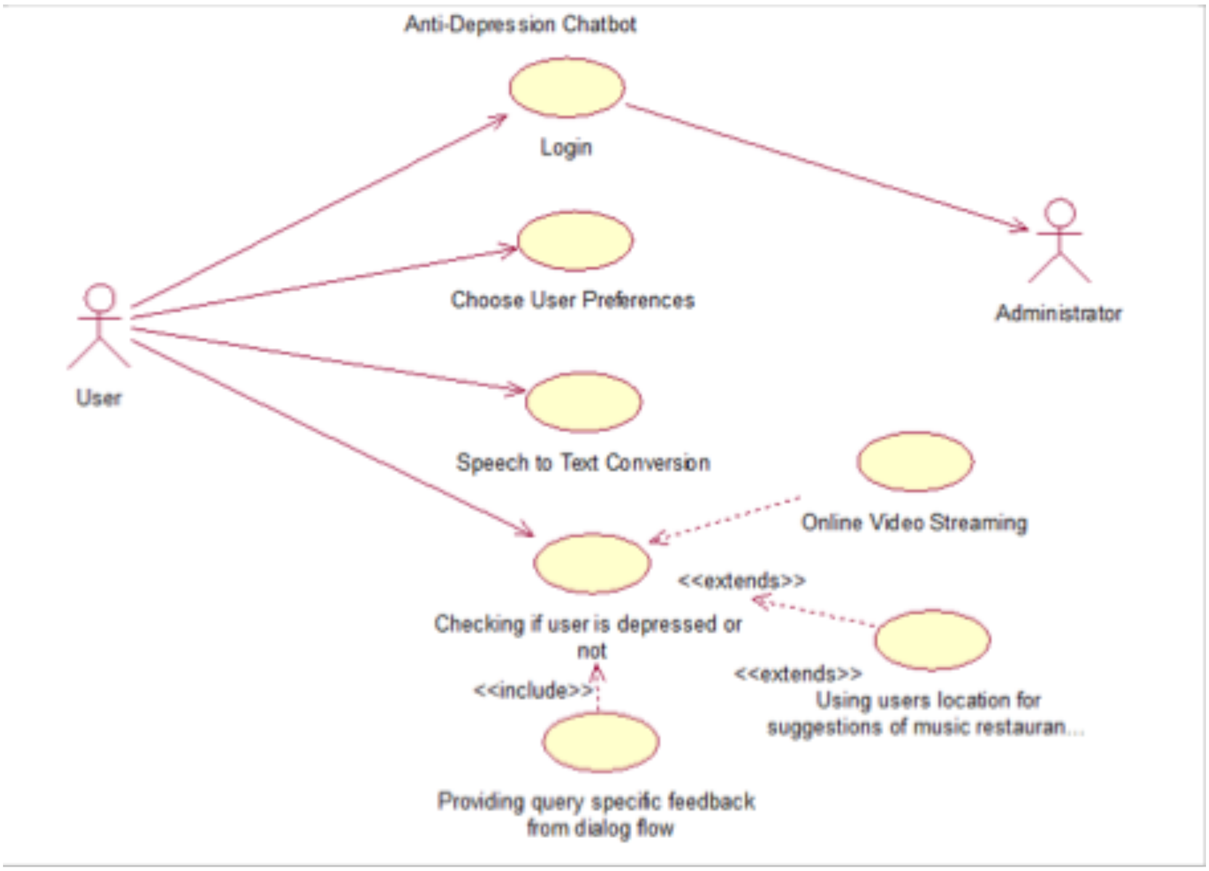

Anti-Depression Chatbot
Login
Choose User Preferences
Administrator
User
Speech to Text Conversion
Online Video Streaming
<<extends>>
Checking if user is depressed or not
<<extends>>
<<include>>
Using users location for suggestions of music restauran...
Providing query specific feedback from dialog flow

Fig. 9: Use Case Diagram of the proposed system

ii.Algorithm

For every sort of inquiry, an extraordinary example must be accessible in the database to give a reasonable reaction. With heaps of mix on examples, it makes a various leveled structure. We use calculations to diminish the classifiers and create the more sensible structure. PC researchers consider it a "Reductionist" approach-so as to give an improved arrangement, it diminishes the issue.

Global Naive Bayes is the exemplary calculation for content arrangement and NLP. For an occurrence, how about we expect a lot of sentences are given which are having a place with a specific class. With new information sentence, each word is meant its event and is represented its shared trait and each class is alloted a score. The most astounding scored class is the destined to be related with the information sentence.

For instance Sample Training set

class: welcoming

"How you doing?"

"hello"

"hello"

Barely any example Input sentence characterization:

input: "Hi hello"

term: "hi" (no matches)

Term: "great" (class: welcoming)

term: "morning" (class: welcoming)

arrangement: welcoming (score=2)

With the assistance of condition, word matches are found for given some example sentences for each class. Arrangement score distinguishes the class with the most astounding term coordinates however it additionally has a few constraints. The score means which goal is well on the way to the sentence yet does not promise it is the ideal match. Most astounding score just gives the relativity base.

Counterfeit Neural Networks

Neural Networks are a method for computing the yield from the information utilizing weighted associations which are determined from rehashed cycles while preparing the information. Each progression through the preparation information corrects the loads bringing about the yield with precision.

Neural systems - chatbots

As talked about before here likewise, each sentence is separated into various words and each word at that point is utilized as contribution for the neural systems. The weighted associations are then determined by various emphasess through the preparation information a great many occasions. Each time improving the loads to making it exact. The prepared information of neural system is a tantamount calculation more and less code. At the point when there is a similarly little example, where the preparation sentences have 200 distinct words and 20 classes, at that point that would be a grid of 200×20. Be that as it may, this lattice measure increments by n times all the more progressively and can cause countless. In this sort of circumstances, preparing velocity ought to be extensively high.

There are various varieties in neural systems, calculations just as examples coordinating code. Unpredictability may likewise increment in a portion of the varieties. Be that as it may, the key continues as before, and the essential work is that of arrangement.

NLU (Natural Language Understanding)

It has 3 explicit ideas like:

Substances: Entity fundamentally speaks to an idea in your Chatbot. It may be an installment framework in your Ecommerce Chatbot.

Goals: It is fundamentally the activity chatbot ought to perform when the client state something. For example, plan can trigger same thing if client types "I need to arrange a red pair of shoes", "Do you have red shoes? I need to arrange them" or "Demonstrate to me some red pair of shoes", these client's content show trigger single direction giving clients alternatives for Red pair of shoes.

Setting: When a NLU calculation dissects a sentence, it doesn't have the historical backdrop of the client discussion. It implies that in the event that it gets the response to an inquiry it has quite recently asked, it won't recall the inquiry. For separating the stages amid the talk discussion, it's state ought to be put away. It can either be banners like "Requesting Pizza" or parameters like "Eatery: 'Dominos'". With setting, you can without much of a stretch relate aims with no compelling reason to realize what was the past inquiry.

NLP (Natural Language Processing)

Normal Language handling (NLP) Chatbot finds a way to change over the client's content or discourse into organized information that is utilized to choose the related answer. A portion of the Natural Language Processing steps are:

Assessment Analysis: Tries to learn if the client is having a decent encounter or if the after some point the visit ought to be sent to the human.

Tokenization: The NLP isolates a series of words into pieces or tokens that are semantically representative or are distinctively valuable for the application.

Named Entity Recognition: The chatbot program demonstrate searches for classifications of words, similar to the name of the item, the client's name or address, whichever information is required.

Standardization: The Chatbot program show forms the content with an end goal to discover basic spelling botches or typographical blunders that may the client aim to pass on. This gives increasingly human like impact of the Chatbot to the clients.

Reliance Parsing: The Chatbot searches for the items and subjects-action words, things and normal expressions in the client's content to discover needy and related expressions that clients may attempt pass on.

Like the majority of the Applications, the Chatbot is likewise associated with the Database. The learning base or the database of data is utilized to sustain the chatbot with the data expected to give an appropriate reaction to the client. Information of client's exercises and whether your chatbot had the capacity to coordinate their inquiries, is caught in the information store. NLP makes an interpretation of human language into data with a blend of examples and content that can be mapped in the ongoing to discover appropriate reactions.

There are NLP administrations and applications programming interfaces that are utilized to fabricate the chatbots and make it workable for all sort of organizations, little. Medium and vast scale. The primary concern here is that Smart Bots can possibly help increment your client base by improving the client bolster administrations and accordingly supports the deals just as benefits. They are an open door for some little and moderate sized organizations to achieve an enormous client base.

As individuals talk with our bot, they may at times become irritated: possibly they wish our business hours were unique, or are disappointed with our administrations. We will break down the tone (estimation) of every one of their announcements. In the event that we see a negative pattern, we'll contribute a few reactions to enable them to feel somewhat better, improving client commitment and maintenance.

To keep things clean, we'll put these reactions into a different Topic, so hit that "New Topic" catch and call this one "slants". Presently include this code, which contains the reactions we'll contribute when we identify changes in their disposition:

```
+ cheerful

- Great! We should continue.\n\n{topic=faq}{@<get input>}

+ furious

- Sorry things aren't working for you. I'll do my best to fix it! \n\n{topic=faq}{@<get input>}

- Stick with me. We'll make sense of this together.\n\n{topic=faq}{@<get input>}

+ veryangry

- I understand this is baffling. Would i be able to apologize with a coupon for 20% off your next purchase?\n\n{topic=faq}{@<get input>}

- It sounds like I'm not working admirably of fixing the issue. Would it be simpler for us to call you?\n\n{topic=faq}{@<get input>}

+ alright

- {topic=faq}{@<get input>}
```

In contrast to our prior contents, these Triggers don't contain "*", in light of the fact that these Triggers won't be shot by the client's real composing. Afterward, they'll be activated expressly by come code we're going to add to the "default" Topic. At the point when any of these Triggers is hit, it will answer with one of the reactions (picked arbitrarily) from straightforwardly underneath the Trigger.

After we react with an expression to console the client, we need to keep handling their unique info. Luckily, we as of now have this put away in the client variable "input", (which we set in the "default" content). Along these lines, we essentially include "{topic=faq}{@<get input>}" to every reaction, demonstrating that we'll change them over to the "faq" Topic and send it the estimation of the "input" variable.

Each time the client communicates something specific, we'll dissect its tone. This is known as Sentiment Analysis, and is too mind boggling to even consider implementing straightforwardly inside a chatbot content. Luckily, Dexter makes it simple to call APIs, so we can utilize Algorithmia's SentimentAnalysis calculation to decide how positive or negative the tone of every approaching message is.

Since we need to recognize at changes in the client's temperament after some time, we'll additionally need to complete a tad of time arrangement investigation. For this, we can utilize Algorithmia's ExponentialMovingAverage. This API enables us to go in a few qualities (for this situation, the notion dimension of the client's last couple of articulations) and get back a weighted moving normal in which the latest qualities have a lot more noteworthy significance than more established ones.

Calling an API from Dexter requires only one line of code, set after any Trigger. Snatch this precedent, and spot it in your "default" subject, supplanting the code which is as of now there:

```
+ *
```

```
$ POST:SA https://api.algorithmia.com/v1/algo/nlp/SentimentAnalysis/1.0.3 {"headers": {"Content-Type": "application/json", "Approval": "Basic YOUR_API_KEY"}, "body": {"document":"<star>"}}

- <set input=<star>> ${{SA.result.0.sentiment}}
```

Before you can utilize this, you need to roll out one manual improvement. Calls to Algorithmia's API require an API key, which you can get for nothing by making a beeline for Algorithmia.com/ information exchange. When you've joined, head to your profile page. Under the "Accreditations" tab, duplicate your API key (beginning with "sim… "). Utilize this to supplant "YOUR_API_KEY" in the code above.

Presently test this out, utilizing the bot analyzer alongside your content editorial manager on Dexter. On the off chance that everything is wired up appropriately, you can type in a positive articulation like "This is Amazing!!" and get back a positive number (about 0.5859). Composing in a negative explanation, for example, "That is awful", should yield a negative number (appx - 0.4767).

iv.Program:

Faith View Controller:

```
import UIKit
import ApiAI
import AVFoundation import NRSpeechToText import SwiftyJSON

class FaithViewController: UIViewController,
UIGestureRecognizerDelegate {

@IBOutlet weak var logo: UIImageView!
var query = ""
@IBOutlet weak var labelText: UILabel!
@objc func longTap(_ sender: UILongPressGestureRecognizer) {

if(sender.state == .began) { animate()

NRSpeechToText.shared.authorizePermission { (authorize) in if
authorize {

if !NRSpeechToText.shared.isRunning { print("start")

self.startRecording(completion: { (result) in self.query = result
self.processQuery(completion: { (text) in

self.textToSpeech(text: text) })

}) }

} }

}
else if(sender.state == .ended) {

print("end")
stopAnimation() NRSpeechToText.shared.stop()

} }
```

```
func processQuery(completion : @escaping(String) -> ()) { let request
= ApiAI.shared().textRequest() request?.query = [query]
var speech = ""

response) in

request?.setMappedCompletionBlockSuccess({ (request,

let response = response as! AIResponse
let resp = response.result.fulfillment.messages[0] speech =
resp["speech"] as? String ?? "" print(speech)
completion(speech)

}, failure: { (request, error) in print(error as Any)

}) ApiAI.shared().enqueue(request)

}
```

21

```
var preferences : [MyPreferences] = []

override func viewDidLoad() {
super.viewDidLoad()
let longGesture = UILongPressGestureRecognizer(target: self,

action: #selector(longTap(_:))) longGesture.minimumPressDuration =
0.3 logo.addGestureRecognizer(longGesture)

}
override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning() }

func tapped() { }

func startRecording(completion : @escaping(String) -> ()) { var x = 0

NRSpeechToText.shared.startRecording {(result: String?, isFinal:
Bool, error: Error?) in

if error == nil { self.labelText.text = result
}
if(isFinal && x == 0) {

x=1

completion(result!) }

} }

func animate() { }

func stopAnimation() { self.logo.layer.removeAllAnimations()
self.logo.layoutIfNeeded()

}

func textToSpeech(text : String) {
let speechSynthesizer = AVSpeechSynthesizer()
let speechUtterance = AVSpeechUtterance(string: text)
speechUtterance.rate = AVSpeechUtteranceMaximumSpeechRate /
2.2 speechUtterance.voice = AVSpeechSynthesisVoice(language: "en-

US") }

}

speechSynthesizer.speak(speechUtterance)
```

Landing View Controller:

```
import UIKit
class LandingViewController: UIViewController {

override func viewDidLoad() { super.viewDidLoad()

Services.shared.fetchUserLoggedIn() { (preferences) in
if(preferences.count > 0) {

Any?.self)
}
self.performSegue(withIdentifier: "loggedIn", sender:

else {
self.performSegue(withIdentifier: "logIn", sender:

Any?.self)
}

} }

override func didReceiveMemoryWarning()
{ super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.

} }
```

Preferences View Controller:

```
import UIKit
import CoreLocation import Magnetic

class PreferencesViewController: UIViewController,
MagneticDelegate {

@IBOutlet weak var bttnContainer: UIView!
var selected : [Node] = []
@IBOutlet weak var nextBttn: UIButton! @IBAction func
nextBttnTapped(_ sender: Any) {

if(validate()) { clear()

x=x+1 print(x) setUpView()

} }

func validate() -> Bool {
if(x==0 && selected.count == 3) {

return true

}
if(x==1 && selected.count == 6) {
```

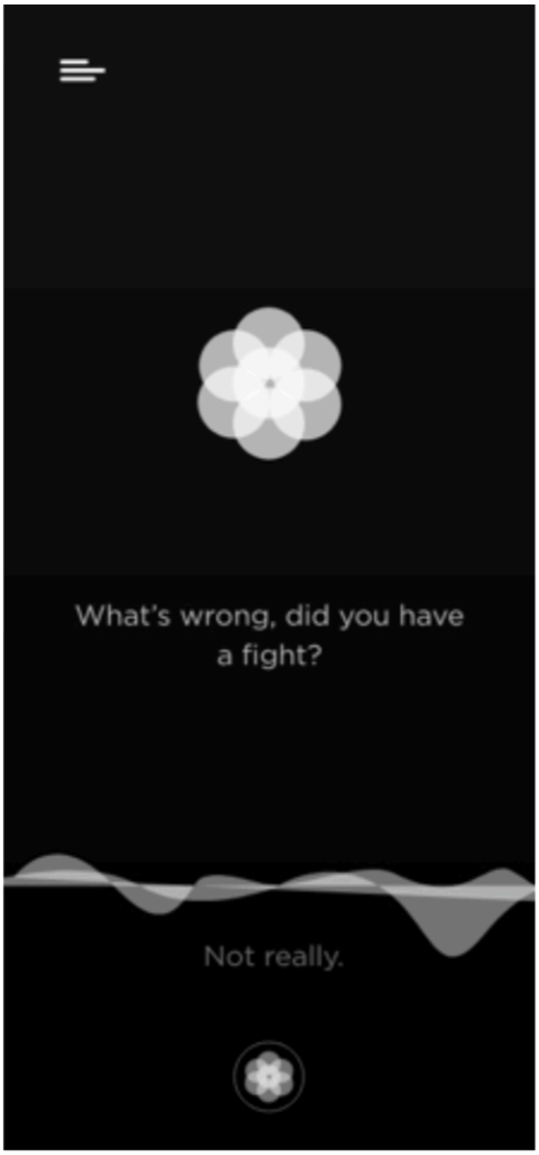
```
return true

}
if(x==2 && selected.count == 9) {

return true

}
else {

return false

} }

@IBOutlet weak var cardText: UILabel! var locationManager :
CLLocationManager! var username = ""
var password = ""

var music =
["Pop","Rock","Indie","Trance","EDM","Punk","HipHop","Bollywoo
d","Tamil ","Hindi","Telugu","Guitar","Rap"]

var places =
["Parks","Temple","Church","Fuel","Atm","Cafes","Hospitals","Stati
ons", "Mosques","Restaurants"]

var cuisine =
["Chinese","Indian","Mexican","Italian","Mediterranean","South
Indian","North Indian","Thai","BBQ","Pizza"]

var data :[String] = []
var refined : [String] = []

var x = 0
var magnetic: Magnetic? override func viewDidLoad() {

super.viewDidLoad()
let magneticView = MagneticView(frame: self.view.bounds)
```

```
music"
```

```
visit"

cuisine"

cardText.text = "Choose exactly 3 different genres of addData(data:
music)

magnetic = magneticView.magnetic
self.view.addSubview(magneticView)

view.bringSubview(toFront:

view.bringSubview(toFront: view.bringSubview(toFront:
magnetic?.magneticDelegate setUpView()

}

cardText)

bttnContainer) nextBttn)
= self

func setUpView() {
var pref : [PrefData] = []
var mypref = PrefData(pref: [], username: "", password: "") if(x==0) {

}
else if(x==1) {

cardText.text = "Choose exactly 3 different places of addData(data:
cuisine)

}
else if(x==2) {

cardText.text = "Choose exactly 3 different types of addData(data:
places)

}
else {

print(username) mypref.username mypref.password for i in 0...2 {

= username
= password
mypref.pref.append(MyPreferences(music: selected[i].text!, place:
selected[i+3].text!, cuisine:

selected[i+6].text!)) }

pref.append(mypref)

Services.shared.persistUserData(data: pref) { (status) in if(status) {

self.performSegue(withIdentifier: "SignUpSuccessful", sender:
Any?.self)

} }

} }

func clear() { magnetic?.removeAllChildren()
magnetic?.reloadInputViews()

}
override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning() }


func magnetic(_ magnetic: Magnetic, didSelect node: Node)
{ if(selected.count <= 3*(x+1)) {

selected.append(node) }

}

func magnetic(_ magnetic: Magnetic, didDeselect node: Node) { }

func addData(data : [String]) { for item in data {

let node = Node(text: item, image: nil, color: .red,
magnetic?.addChild(node)

} }

}
```

v.Result analysis

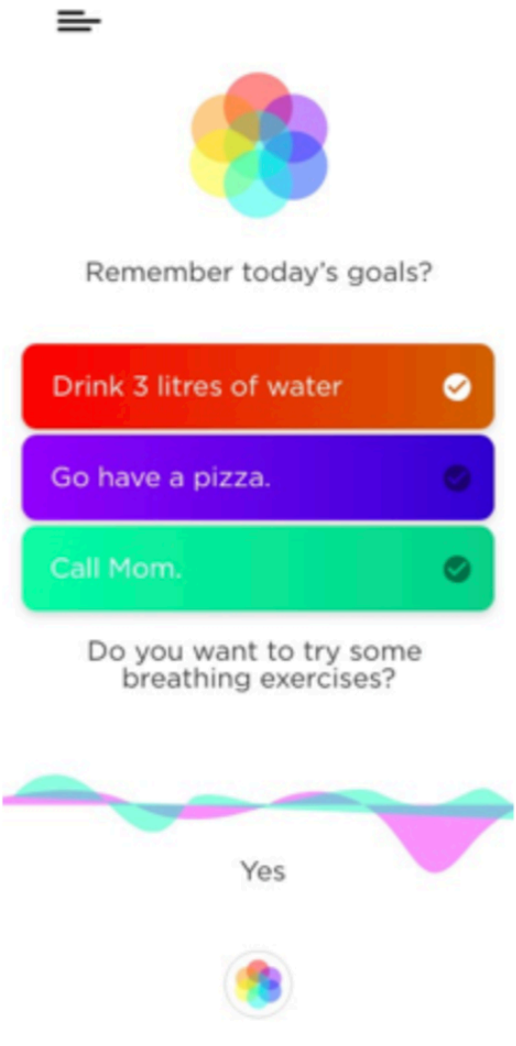
<u>Screenshots</u>



What's wrong, did you have a fight?

Not really.

Fig. 10: Main page



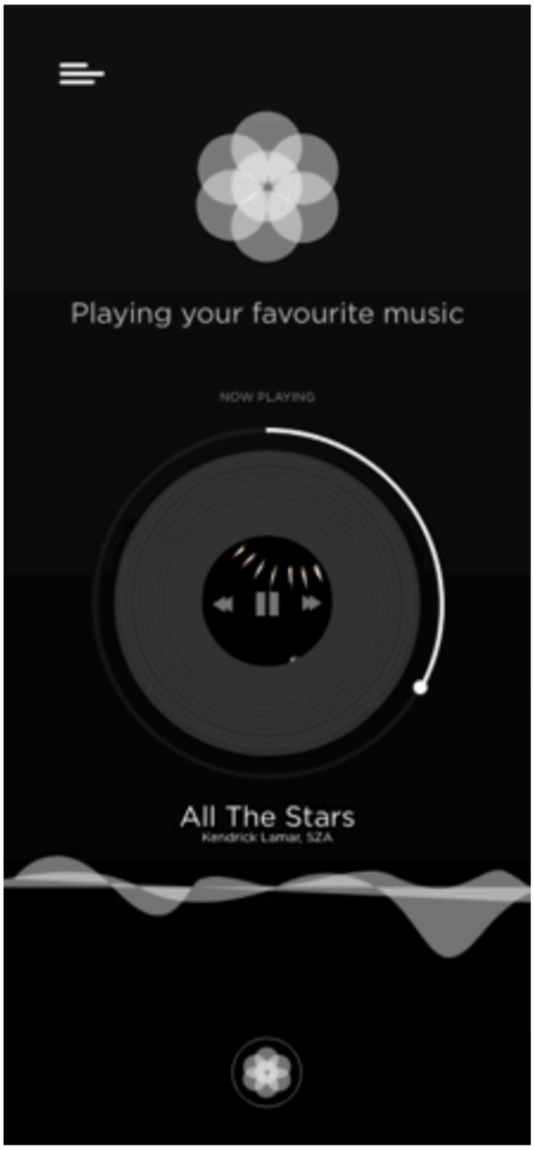
Remember today's goals?

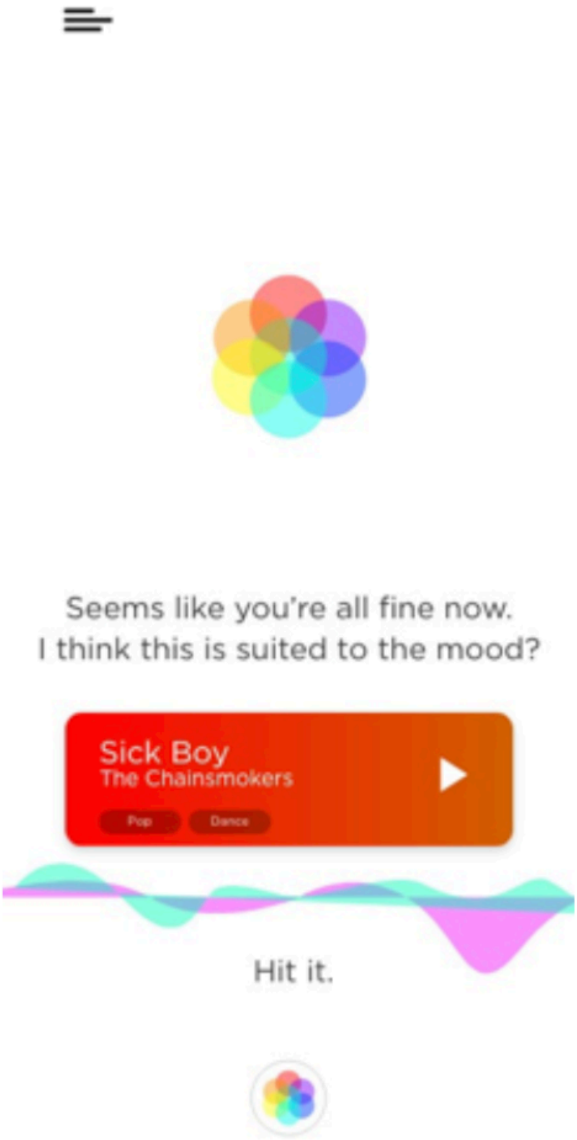
- Drink 3 litres of water ✓
- Go have a pizza. ✓
- Call Mom. ✓

Do you want to try some breathing exercises?

Yes

Fig. 11: Preferences module


Playing your favourite music


Seems like you're all fine now.
I think this is suited to the mood?

Hit it.

Fig. 12: Music player

Test Cases and Analysis

Test Case/Script Identifier	Test Case/Script Title	Execution Priority
Launch Application	Launch of Bot	*High*
Enter correct email and password and hi the login button	Credentials validation	*High*
Voice recognition of the user’s voice	Voice extraction	*High*
Suggestions for users	Suggestions for depressed user	*High*
Suggestions execution	Proper execution of suggestions	*Low*
Regular talking	Bot regular interaction	*Low*
Gathering of users interests	Gathering of interests	*Low*
Change of preferences	Preferences changes	*High*
Checking mental status of user	Users mental status	*High*
Proper detection of the users depression	Depression detection	*High*
Checking output in the console window	Console output checking	*Low*
Checking Location API	Location API (Google)	*High*
Checking Zomato API	Food Restaurants API	*High*
Checking users favorites songs	Song Favorites	*Low*
Logout from the system	Logout	*High*

Table 1: Test cases and analysis

Table 1 represents the different test cases and their analysis involved in our project. This can help us in proper testing of the system.

Performance Analysis

Necessity Traceability Matrix or RTM catches all prerequisites proposed by the customer or programming advancement group and their traceability in a solitary report conveyed at the finish of the life-cycle.

At the end of the day, it is a report that maps and follows client necessity with experiments. The primary motivation behind Requirement Traceability Matrix is to see that all experiments are secured with the goal that no usefulness should miss while doing Software testing.

Requirements Traceability Matrix			
Project Report: Anti-depression Bot			
Functional Requirement Document (FSD)			**Test Case Document**
Functional Requirement ID#	**Functional Requirement/ Use case**	**Priority**	**Test Case ID#**
FR_1	Launch Application	*High*	TC#001
FR_2	Enter correct email and password and hit the login button	*High*	TC#002
FR_3	V oice recognition of the user's voice	*High*	TC#003

FR_4	Suggestions for users	*High*	TC#004
FR_5	Suggestions execution	*Low*	TC#005
FR_6	Regular talking	*Low*	TC#006
FR_7	Gathering of users interests	*Medium*	TC#007
FR_8	Change of preferences	*High*	TC#008
FR_9	Checking mental status of user	*High*	TC#009
FR_10	Proper detection of the users depression	*High*	TC#010
FR_11	Checking output in the console window	*Low*	TC#011
FR_12	Checking Location API	*High*	TC#012
FR_13	Checking Zomato API	*High*	TC#013
FR_14	Checking users favorites songs	*Low*	TC#014
FR_15	Logout from the system	*High*	TC#015

Table 2: Performance Analysis

Table 2 represents the

Requirement Traceability Matrix or RTM which captures all requirements proposed by the client or software development team and their traceability in a single document delivered at the conclusion of the life-cycle. This can help us analyse the performance of the system.

vi.Conclusion and Future Work

This project is just a stepping stone in the development of this idea. We are still further looking to make this chatbot a complete application in every manner. One additional feature and scope for future work is to make the bot understand sarcasm. This is because when the user is frustrated or tired or depressed, they tend to give sarcastic answers, which us humans can interpret using facial cues or expressions or body language, but the bot cannot. However, we will try to add this feature to the bot as well. We also plan to allow the bot to speak out jokes, to lighten the mood whenever required. This should be sensed by the bot when the user is sad and accordingly send jokes to the user.

vii.References

[1] https://www.businessinsider.in/A-Stanford-researcher-might-be-on-the-brink-of-a- dramatic-shift-in-how-we-treat-depression/articleshow/59022778.cms.

[2] https://www.businessinsider.in/I-spent-2-weeks-chatting-with-a-robot-about-my- anxiety-heres-how-it-went/articleshow/62715617.cms.

[3] Internet Interventions, Volume 2, Issue 2, May 2015, Pages 169-173. Current research and trends in the use of smartphone applications for mood disorders. John Torous, Adam C. Powell.

[4] RECORD DISPLAY CITATION Morris, M. E., & Aguilera, A. (2012). Mobile, social, and wearable computing and the evolution of psychological practice. *Professional Psychology: Research and. Practice, 43*(6), 622-626.

[5] J Med Internet Res. 2016 Nov; 18(11): e287.
Can Mobile Phone Apps Influence People's Health Behavior Change? An Evidence Review.
Gunther Eysenbach.

[6] SAGE JOURNAL, The CES-D Scale Volume: 1 issue: 3, page(s): 385-401 A Self-Report Depression Scale for Research in the General Population. Lenore Sawyer Radloff.