Under graduate

Raspberry pi based Project

Query Bot

Under Guidance of:

Dr. Arun Kumar Sinha

(Associate Professor)

School of Electronics Engineering

VITAP, University

Team Members (Semester-III, 2019):

1. 18BCD7009 - Rishabh Singh Bais
2. 18BCN7057 - Paduchuru Abhishek
3. 18BCE7091 - Pulkit Joshi
4. 18BCE7116 - Rakesh Ranjan
5. 18MIS7152 - Kota V. V. L Naga Datta Manoj
6. 18BCE7280 - Pooly Vinay Kumar Reddy

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to my guide, Dr. Arun Kumar Sinha who introduced us to the methodology of work, and whose passion for the “underlying structures,” had lasting effect, as well as our university VIT-AP who gave us the golden opportunity to do this wonderful project on the topic “Query Bot” which also helped us in doing a lot of research and we came to know about so many new things, we are really thankful to them. Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our project. We thank all the people for their help directly and indirectly to complete our project.

Date: 02-11-2019

CONTENTS

1. INTRODUCTION

Today, Alexa lets us control our lights with voice, Google assistant suggests places we would want to go when we are discussing dinner plans with our friends and Tesla can drive for us. Siri and Cortana live inside our phones and take commands. Screen less conversations are expected to dominate. Today, due to the penetration of social media and internet along with the progress in artificial intelligence, not only are bots coming up as a way to reach out to the users but also, they are making way for conversational interfaces such as Alexa to be omnipresent and go screen-less. In the case of chat bots, with the launch of messenger bots and platforms such as slack, bots have gotten a boost. Facebook bots have grown from 34000 bots in November 2016 to 100,000 bots in April 2017 . As social media widens its penetration, companies will shift their focus to bots for reaching and serving users [3].

MOTIVATION

We sometimes need professors help for various projects, assignments or any academic related help. But there are many professors and we don't know at what time they are available, what is their cabin no. So making a query bot is indeed an interesting idea as we can save students time also. We also have thought of integrating it with chat bot features.

2. LITERATURE REVIEW

The ones that are rule based or fetch responses from a predetermined set of responses. Retrieval Based models can be used when the data is limited and the conversation domain is constricted to a few conversation scenarios. Hence, if all the possible conversations for a use case with the bot are imaginable, a retrieval based model works well. Also, when the bot is not expected to display intelligence in all scenarios but can afford to deny knowing the answer, such retrieval based models work well. Booking systems, FAQ systems or any other systems that fetch information could work well even with a retrieval based bot [2] .

3. COMPONENTS REQUIRED

- A Raspberry pi.
- Android Device.

4. TELEGRAM BOT ARCHITECTURE

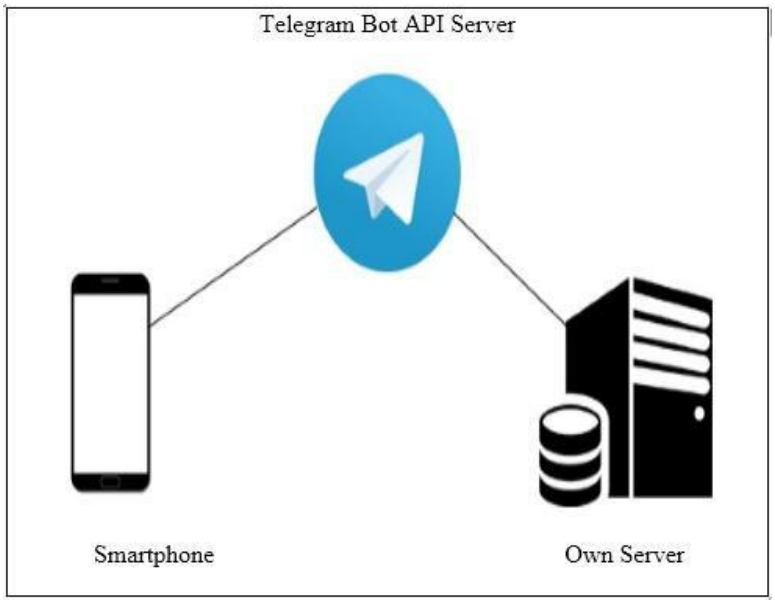

Figure 1: Telegram bot architecture[3]

5. SOFTWARE AND HARDWARE

5.1 Hardware:

Raspberry Pi: The raspberry pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python [2].

5.2 Software:

Python: Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming. Many other paradigms are supported via

extensions, including design by contract and logic programming. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management [1].

6. CONSTRUCTION AND TESTING

6.1 Assembly Instructions

- Collecting the faculty information.
- Integrating the weather API with Telegram bot code.
- Reading news headlines using python web scrapping.
- Integrating Wikipedia library with bot.
- Uploading the code in raspberry pi using python programming language.
- Uploading faculty information in Telegram bot.

6.2 Testing

- Testing and debugging the code.
- Integrating with Telegram bot.
- Testing bot in Telegram bot.
- Testing bot for various queries.
- Bot is ready to use.

7. RESULTS

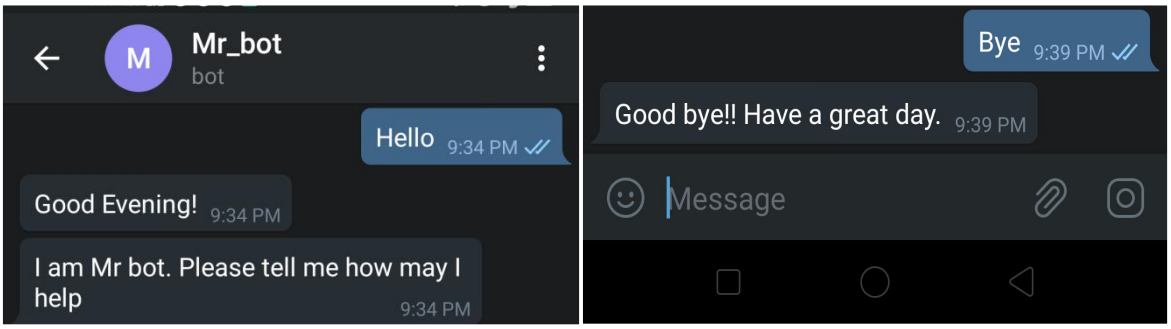

Figure 2: General reply

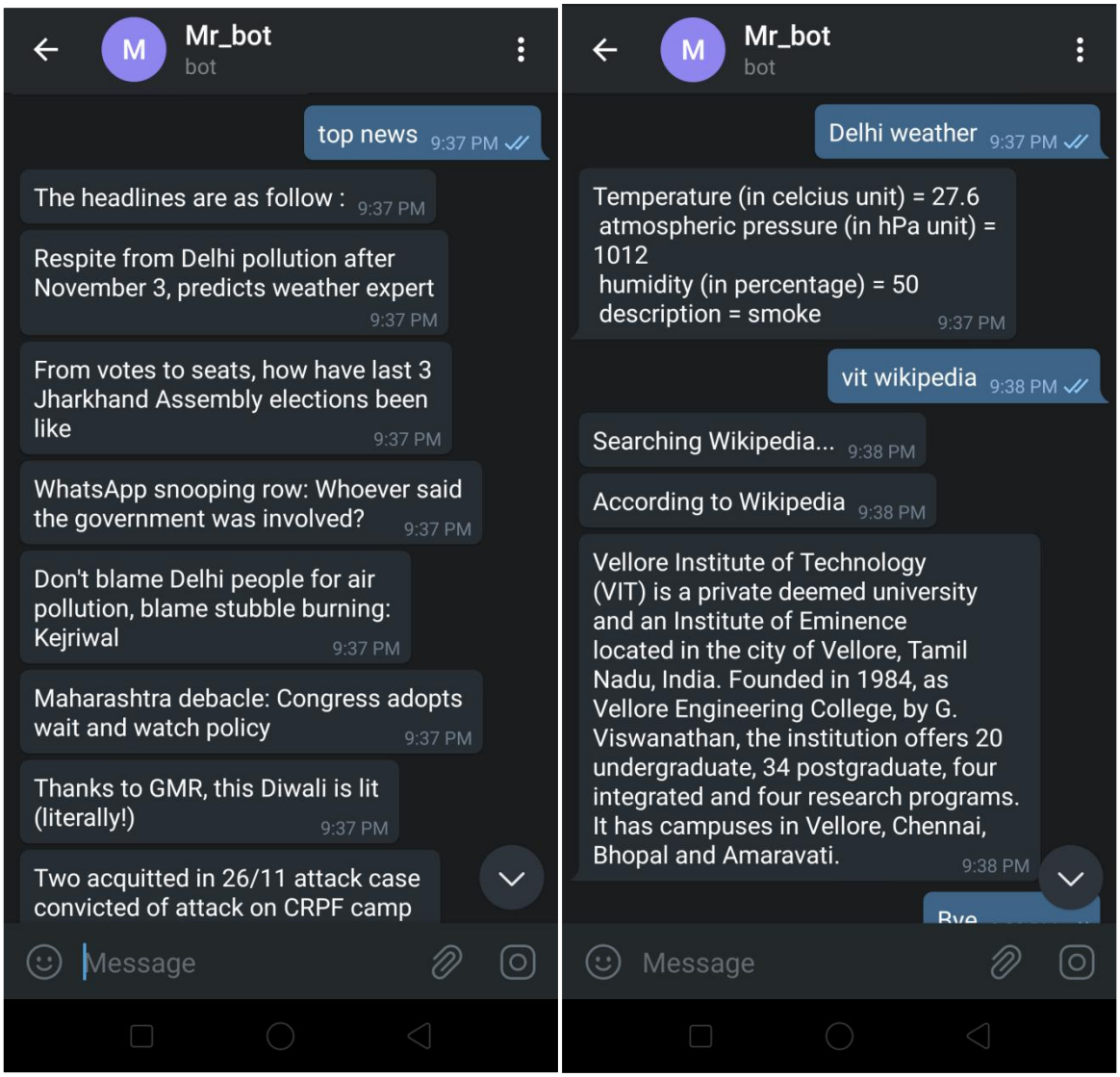

Figure 3: Top headlines, weather & Wikipedia search queries

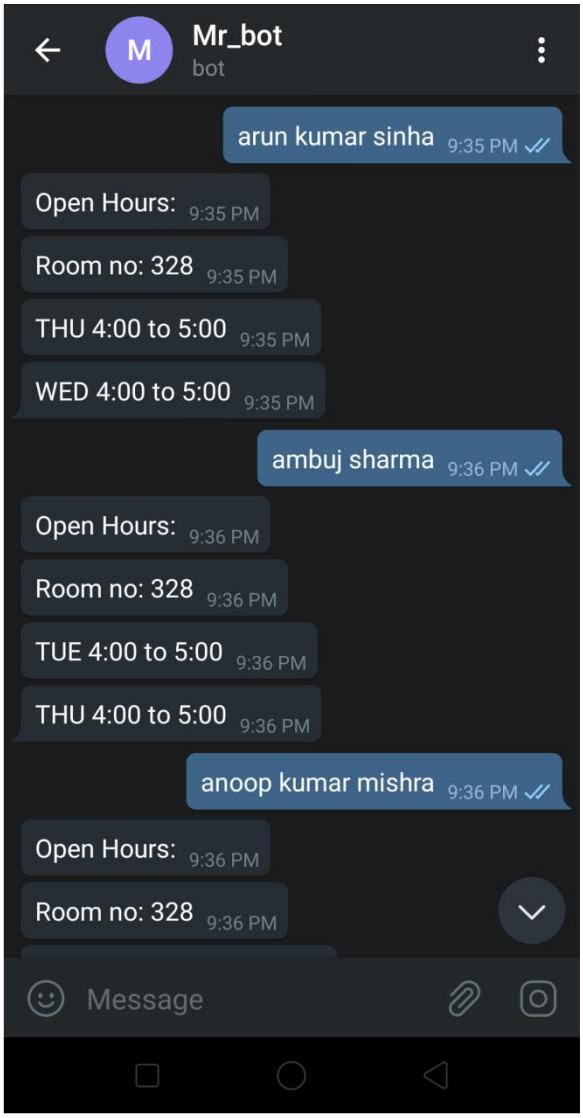

Figure 4: Faculty details

8. WORK PLAN

Timeline (Week)	5th Aug	14th Aug	5th Sep	10th Sep	11th Sep.	2nd Oct	12th Oct	27th Oct	5th Nov	18th Nov
Delivery										
Overall Outcome										
Testing										
Connections										
Coding										
Hardware analysis										
Problem identify										
Literature survey										

9. REFERENCES

[1]Available online: https://www.hackster.io/Salmanfarisvp/telegram-bot-with-raspberry-pi-f373da

[2]Available online: https://create.arduino.cc/projecthub/ShebinJoseJacob/telegram-bot-with-esp8266-dbada8

[3]Available online: https://www.instructables.com/id/Make-a-Python-Powered-ChatBot-Raspberry-Pi/

[4]Available online: http://kamilslab.com/2018/09/15/how-to-create-a-simple-python-chatbot-on-the-raspberry-pi/

[5]Available online: https://www.raspberrypi.org/

[6]Available online: https://en.wikipedia.org/wiki/Raspberry_Pi

[7]Book: Raspberry Pi For Dummies by Mike Cook and Sean McManus

[8]Book : Getting Started with Raspberry Pi: Electronic Projects with Python, Scratch, and Linux by Matt Richardson and

[9]Chatbots Magazine by Octane AI, “Machine Learning,Neural Networks and Algorithms – Chatbots Magazine.”

[10] Oracle, “Chatbots 101,” pp. 1–4, 2017.

APPENDIX:

```
import sys
import time
import ssl
import random
import datetime
import telepot
def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
    print ('Got command: %s') % command
    if 'Bye' in command:
      bot.sendMessage(chat_id,"Good bye!! Have a great day.")
    elif 'anoop kumar mishra' in command:
        bot.sendMessage(chat_id,"Open Hours:")
        bot.sendMessage(chat_id,"Room no: 328")
        bot.sendMessage(chat_id,"FRI 10:30 to 11:30")
        bot.sendMessage(chat_id,"SAT 3:00 to 4:00")
    elif 'amogh katti' in command:
        bot.sendMessage(chat_id,"Open Hours:")
        bot.sendMessage(chat_id,"Room no: 328")
        bot.sendMessage(chat_id,"TUE 2:00 to 3:00")
        bot.sendMessage(chat_id,"THU 4:00 to 5:00")
    elif 'ambuj sharma' in command:
        bot.sendMessage(chat_id,"Open Hours:")
        bot.sendMessage(chat_id,"Room no: 328")
        bot.sendMessage(chat_id,"TUE 4:00 to 5:00")
        bot.sendMessage(chat_id,"THU 4:00 to 5:00")
    elif 'alluri' in command:
        bot.sendMessage(chat_id,"Open Hours:")
        bot.sendMessage(chat_id,"Room no: 328")
        bot.sendMessage(chat_id,"THU 1:00 to 2:00")
        bot.sendMessage(chat_id,"WED 1:00 to 2:00")
```

```
elif 'nagaraju devarakonda' in command:
    bot.sendMessage(chat_id,"Open Hours:")
    bot.sendMessage(chat_id,"Room no: 328")
    bot.sendMessage(chat_id,"SAT 9:00 to 10:00")
    bot.sendMessage(chat_id,"THU 12:00 to 1:00")
elif 'arun kumar sinha' in command:
    bot.sendMessage(chat_id,"Open Hours:")
    bot.sendMessage(chat_id,"Room no: 328")
    bot.sendMessage(chat_id,"THU 4:00 to 5:00")
    bot.sendMessage(chat_id,"WED 4:00 to 5:00")
elif 'rama satish' in command:
    bot.sendMessage(chat_id,"Open Hours:")
    bot.sendMessage(chat_id,"Room no: 328")
    bot.sendMessage(chat_id,"SAT 12:00 to 1:00")
    bot.sendMessage(chat_id,"WED 1:00 to 2:00")
elif 'anupama namburu' in command:
    bot.sendMessage(chat_id,"Open Hours:")
    bot.sendMessage(chat_id,"Room no: 328")
    bot.sendMessage(chat_id,"FRI 11:00 to 12:00")
    bot.sendMessage(chat_id,"WED 11:00 to 12:00")
elif 'weather' in command:
    import requests, json
    api_key = "449a334d62ee40a08c18d99ba97249db"
    base_url = "http://api.openweathermap.org/data/2.5/weather?"
    command = command.replace("weather", "")
    city_name = command
    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()
    if x["cod"] != "404":
        y = x["main"]
        current_temperature = y["temp"]
        current_temperature = current_temperature - 273.15
```

```
            current_pressure = y["pressure"]
            current_humidiy = y["humidity"]
            z = x["weather"]
            weather_description = z[0]["description"]
            bot.sendMessage(chat_id,"Temperature (in celcius unit) = " +
                    str(current_temperature) +
        "\n atmospheric pressure (in hPa unit) = " +
                    str(current_pressure) +
        "\n humidity (in percentage) = " +
                    str(current_humidiy) +
        "\n description = " +
                    str(weather_description))

        else:
            bot.sendMessage(chat_id,"City not found")
    elif 'top news' in command:
        import requests
        from bs4 import BeautifulSoup
        url='https://www.indiatoday.in'
        resp=requests.get(url)
    #http_respone 200 means OK status
        if resp.status_code==200:
            bot.sendMessage(chat_id,str("The headlines are as follow :"))
        soup=BeautifulSoup(resp.text,'html.parser')

        l=soup.find("ul",{"class":"itg-listing"})

        #now we want to print only the text part of the anchor.
        #find all the elements of a, i.e anchor
        for i in l.findAll("a"):
            bot.sendMessage(chat_id,str(i.text))
#           print(i.text)
```

```
    elif 'wikipedia' in command:
        import wikipedia
        bot.sendMessage(chat_id,'Searching Wikipedia...')
        command = command.replace("wikipedia", "")
        results = wikipedia.summary(command, sentences=3)
        bot.sendMessage(chat_id,"According to Wikipedia")
        bot.sendMessage(chat_id,str(results))
    elif 'Hi'or'Hello' in command:
        hour = int(datetime.datetime.now().hour)
        if hour>=0 and hour<12:
            bot.sendMessage(chat_id,"Good Morning!")
        elif hour>=12 and hour<18:
            bot.sendMessage(chat_id,"Good Afternoon!")
        else:
            bot.sendMessage(chat_id,"Good Evening!")

        bot.sendMessage(chat_id,"I am Mr bot. Please tell me how may I help")
bot = telepot.Bot('964727150:AAFQnxDIGbD1nKJP9gvFwPdt-mIDypBHeRQ')
bot.message_loop(handle)
print ('I am listening...')
while 1:
    time.sleep(10)
```