

**A Minor Project Report on
“VIRTUAL ASSISTANT”**

Submitted to



**Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal
Towards Partial Fulfillment for the Award of
Bachelor of Engineering
(Computer Science and Engineering)**

Submitted By:

Dipesh Paul (0827CS151066)

Gayyur Qureshi (0827CS151073)

Harshdeep Totlaney (0827CS151077)

CS-II, III Year

Guided By

Mr. Deepak Agrawal



**Department of Computer Science and Engineering
Acropolis Institute of Technology and Research, Indore
January-June 2018**

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

RECOMMENDATION

The Project entitled “ **Alan**” submitted by **Dipesh Paul (0827CS151066)**, **Gayyur Qureshi (0827CS151073)**, **Harshdeep Totlaney (0827CS151077)** , is satisfactory on account of the bonafide work done under our supervision and is recommended towards partial fulfilment for the award of **Bachelors of Engineering** (Computer Science and Engineering) degree by **Rajiv Gandhi Proudlyogiki Vishwavidyalaya**, Bhopal.

Date:

Mr. Deepak Agrawal

Project Guide

Computer Science & Engineering Department

Acropolis Institute of Technology &

Research, Indore

Mr. Anil Patidar

Project Coordinator

Computer Science & Engineering Department

Acropolis Institute of Technology &

Research, Indore

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

CERTIFICATE

The project entitled “**Virtual Assistant**” submitted by **Dipesh Paul (0827CS151066)**, **Gayyur Quershi (0827CS151073)**, **Harshdeep Totlaney (0827CS151077)** has been examined by us and is hereby approved for the award of degree **Bachelor of Engineering** in **Computer Science and Engineering** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

Internal Examiner

Date:

Place: Indore

External Examiner

Date:

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

PROJECT APPROVAL SHEET

The project work entitled “**Virtual Assistant**” submitted **Dipesh Paul (0827CS151066)**, **Gayyur Quershi (0827CS151073)**, **Harshdeep Totlaney (0827CS151077)** is approved as partial fulfilment for the award of the **Bachelor of Engineering (Computer Science and Engineering)** degree by **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**.

Date:

Mr. Deepak Agrawal

Project Guide

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

Mr. Anil Kumar Patidar

Project Coordinator

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

STUDENT DECLARATION

We the student of **Bachelors of Engineering** (Computer Science and Engineering), hereby declare that the work presented in this project synopsis entitled “Virtual Assistant” submitted towards completion of Minor Project in 6th semester of B.E. (Computer Science and Engineering) at Acropolis Institute of Technology & Research, Indore, is an authentic record of our own work. Due acknowledge have been made in the text to all other material used. The project was done in full compliance with the requirement and constraints of the prescribed curriculum.

Date:

Dipesh Paul

(0827CS151066)

Gayyur Qureshi

(0827CS151073)

Harshdeep Totlaney

(0827CS151077)

ACKNOWLEDGEMENT

We take this opportunity to express our deepest and sincere gratitude to our guide **Mr. Deepak Agrawal, Assistant Professor, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his constant encouragement and advice.

We express our deep gratitude to **Mr. Anil Kumar Patidar, Assistant Professor, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** for his regular support, co-operation, and co-ordination.

We express our hearty gratitude to **Prof. Sanjay Bansal, Professor & Head, Computer Science & Engineering Department, Acropolis Institute of Technology & Research, Indore** his support, and help provided during the project tenure.

We also express our deep gratitude to **Dr. S. C. Sharma, Principal, Acropolis Institute of Technology & Research, Indore** for all the help they provided for the completion of project.

The in-time facilities provided by the department throughout the Bachelors program are also equally acknowledgeable.

We would like to convey our thanks to the teaching and non teaching staff of the Department of Electronics & Communication and Computer Engineering, acme for their invaluable help and support throughout the period of Bachelors Degree. We are also grateful to all our classmates for their help, encouragement and invaluable suggestions.

Finally, yet more importantly, we would like to express our deep appreciation to our grandparents, parents, sister and brother for their perpetual support and encouragement throughout the Bachelors degree period.

Dipesh Paul(0827CS151077)

Gayyur Qureshi(0827CS151077)

Harshdeep Totlaney(0827CS151077)

ABSTRACT

Artificial Intelligence has emerged as the new computing technology and is converging into a powerful technology that can efficiently work on each computing platform. This article envisions a Virtual Assistant as a new dimension in such a technology to further optimize the overall computing experience for users. Virtual Assistant are software agents that interact with user in a conversation. The main goal of its creation is to resemble a human being in the way they perform said interaction trying to make the user think he/she is writing to another human being. Other important features of Virtual Assistant is to answer any question of human being or displaying weather forecast or setting reminder and many more. Machine Learning is an application of artificial intelligence(AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Python is an interpreted high-level language utilized for all important logics written for Virtual Assistant. Naturally this Virtual Assistant can be extended in daily life for several purposes.

Contents

Recommendation	2
Certificate	3
Project Approval Sheet	4
Log Note	5
Student Declaration	6
Acknowledgement	7
Abstract	8
List of Figures	10
List of Tables	11
Chapter 1 Introduction	12-14
1.1 Rationale	12
1.2 Problem Definition and Proposed Solution	13
1.3 Objective and Scope	13
1.4 Report Organization	13-14
Chapter 2 Literature Surveys/Existing System	15
2.1 Background	15
Chapter 3 System Analysis Requirement Specification	16-17
3.1 Functional and Non Function Requirement	16
3.1.1. Functional	
Functional Requirement will contain your project modules like login page, registration page.	
3.1.2. Non- Functional Requirement	
I. Performance Requirement	
II. Reliability	
III. Scalability	
IV. Security	

V. Testability	
3.2 Software /Hardware Requirements	17
3.3 Feasibility Study	17
Chapter 4 System Design	18-23
4.1 Use Case Diagram	18
4.2 Activity Diagram	19
4.3Sequence Diagram	20
4.4 Class Diagram	21
4.5 Data flow Diagram	22-23
Chapter 5 Project Implementation and Output Screens	24-34
5.1 Screenshots of your Projects	24-28
5.2 Important coding	28-34
Chapter 6 Testing	35-40
6.1 Testing Strategy Adopted	35
6.2 System Testing	36
6.3 Unit Testing	37
6.4 Test Plan & Test Cases	38-40
Chapter 7 Conclusion	41
7.1 Conclusion	41
7.2 Future Enhancement	41
A. Appendix A	42
• References	42
• Bibliography	42

LIST OF FIGURES

LIST OF FIGURES		
Figure Number	Figure Name	Page Number
Figure 1.4 (a)	Incremental Model	14
Figure 4.1.5 (a)	Data Flow Diagram (Level 0)	22
Figure 4.1.5 (b)	Data Flow Diagram (Level 1)	23
Figure 4.1.1 (a)	Use Case Diagram	18
Figure 4.1.2 (a)	Activity Diagram	19
Figure 4.1.3 (a)	Sequence Diagram	20
Figure 4.1.4	Class Diagram	21
Figure 5.1(a)	Home	24
Figure 5.1 (b)	Web Search	25
Figure 5.1 (c)	Weather Forecast	26
Figure 5.1 (d)	Reminder	27
Figure 5.1 (e)	Game	28
Figure 13.3 (a)	System Testing Types	37

LIST OF TABLES

LIST OF TABLES		
Table Number	Table Name	Page Number
Table 1	Test Cases	37

1. INTRODUCTION

1.1 Rationale

“Alan” is a web application. A virtual assistant (typically abbreviated to VA, also called a virtual office assistant) is generally self-employed and provides professional administrative, technical, or creative (social) assistance to clients remotely from a home office. A virtual assistant is a software agent that can perform tasks or services for an individual. A Virtual assistant is based on Artificial Intelligence techniques. Sometimes the term "chat bot" is used to refer to virtual assistants generally or specifically those accessed by online chat. Our virtual assistant can perform the various tasks such as set a reminder, weather forecasting, and it can communicate with you like a human. It can perform the internet search and also provide you the latest news. We will also create a login page through which any user can log in. When next time user will search for the same problem he will get the solution easily. Basically, it is a web app so it is available on every platform.

Since existing assistants are platform dependent so this is the foremost problem we identified. So we are making a virtual assistant which is available on web so it is platform independent.

The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including natural language processing and speech recognition platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated algorithms to learn from data input and become better at predicting the end user's needs.

Features of “Alan”:

- Any human can interact with Assistant
- Weather Forecast
- Stay better informed for any information
- Human can use assistant for entertainment.
- It bridges the gap among humans and technology.
- Gaining extra knowledge from the internet is just one click.

1.2 Problem Definition and Proposed Solution

Many unpredictable factors may affect any human's life. One of the major factor that can affects one's life is loneliness. In today's modern world everybody has a very busy schedule so it is very difficult for them to go to another place and interact with someone. So, the proposed solution is to build a Virtual Assistant that provides various facilities such as to interact or chat, set a remainder, weather forecasting, and it can communicate with you like a human. It can perform the internet search and also you can set a reminder and finally you can also play games.

1.3 Objective and Scope

The main objective of this application is to facilitate a user-friendly environment for all users and reduces the manual effort. In further resolution of the technology we can generate the Virtual Assistant. The functional requirements includes all the questions for which user wants an answer or for which user wants to perform google search. Application will interact or chat, set a remainder, weather forecasting, and it can communicate with you like a human. It can perform the internet search and also you can set a reminder and finally you can also play games.

The major highlights and scope of this project is that it provides an online solution to the questions asked by individuals. This web app is easy to understand and has catchy interface. The system handles all the operations and generates efficient answers to each question or performs the desired features very efficiently

1.4 Report Organization

The software model used in our project is Incremental model. The incremental model applies the waterfall model incrementally. The series of releases is referred to as "increments", with each increment providing more functionality to the customers. After the first increment, a core product is delivered, which can already be used by the customer. Based on customer feedback, a plan is developed for the next increments, and modifications are made accordingly. This process continues, with increments being delivered until the complete product is delivered. The incremental philosophy is also used in the agile process model.

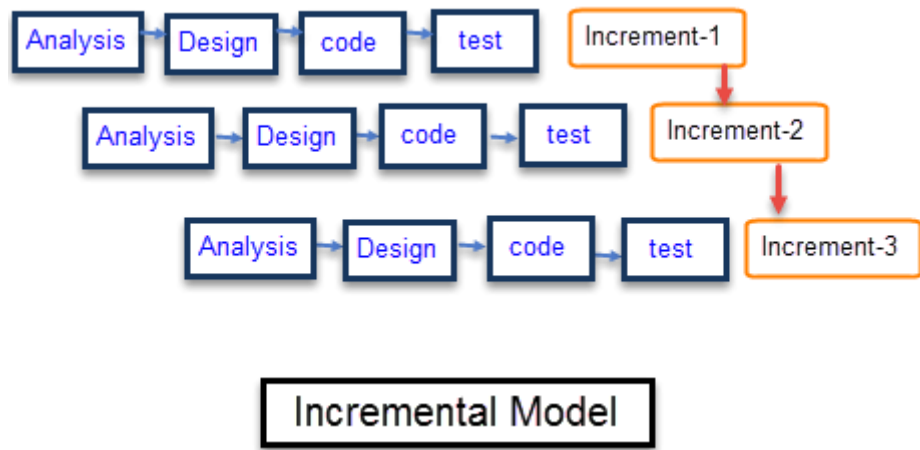


Figure1.4 (a): - Incremental Model

2. Literature Survey/Existing System

2.1 Background

A virtual assistant (typically abbreviated to VA, also called a virtual office assistant) is generally self-employed and provides professional administrative, technical, or creative (social) assistance to clients remotely from a home office. Increasingly Virtual Assistants are utilizing technology such as Skype as well as Google Voice. As of now existence virtual assistant are Google Assistant and Siri are most commonly found. They neck-to-neck when it comes to organising and planning the user's tasks. While Siri has an advantage seamless connection with third party apps such Twitter, Uber, etc, Google Assistant has the support of all its in-house apps. When it comes to choosing a winner, Google Assistant wins by a long shot, while Microsoft's Cortana has a long way to go, seeing how the race between AI-powered VAs is increasing day by day. But all assistants available can only be run on particular platform and this is the major disadvantage which is needed to overcome.

3. System Analysis Requirement

3.1 Functional and Non-Functional Requirement

3.1.1 Functional Requirement

In this section we describe the functional capability of the system for each functional requirement the require inputs, desire output and processing requirement are specified.

They describe the relationship between the input and output of the system.

3.1.2 Non-Functional Requirement

a. Reliability :

High Reliability is the measure of how a product behaves in varying circumstances and our project is reliable because there are less chances of errors and exceptions and works well in varying circumstances. The probability that application will perform required function without failure would depend on server as well as the good internet speed.

b. Availability :

The application is available to users. The users are able to use the application The services are available to the users as long as there is a good internet connectivity.

c. Security :

Security is the ability of the software to remain protected from unauthorized access. This includes both change access and view access. This application is well secured using multiple levels of security constraints. It would be taken care that a person does not access data in an unauthorized manner.

d. Maintainability :

Maintainability of system is done by admin. Admin adds or deletes dataset according to the conditions.

e. Portability :

This project works on an web application, mobile phones are highly portable and can be easily carried anywhere by users while laptops and computers are difficult to carry.

f. Performance :

Performance requirement is concerned with the speed of operations of features and their accuracy. For optimum performance of application some non-functional requirements are recommended:

Good Internet Connectivity

Minimum Processor Required i3 (5th Generation)

Firefox (59.0.2))

3.2 Software/Hardware Requirements

HARDWARE:

- a) Operating System: Android API 17 or higher
- b) Hard disk: 200 MB or higher
- c) RAM: 512 MB or higher
- d) Processor: 1GHz or higher

SOFTWARE:

- a) Text Editor(Brackets or Notepad++)
- b) Pycharm IDE
- c) Dataset

3.3 Feasibility Study

- a. A chat activity is interacting to user on any topic that user likes to talk about. User can communicate with alan on basic topics.
- b. User can basically search any question on web if he or she is not satisfied with the answer given by alan while chatting.
- c. User can know about weather of any city by entering city's name.
- d. User can also play interesting game.
- e. User can also listen music.
- f. User can also watch videos.
- g. User can also set reminder

4. System Design

4.1.1 Use Case

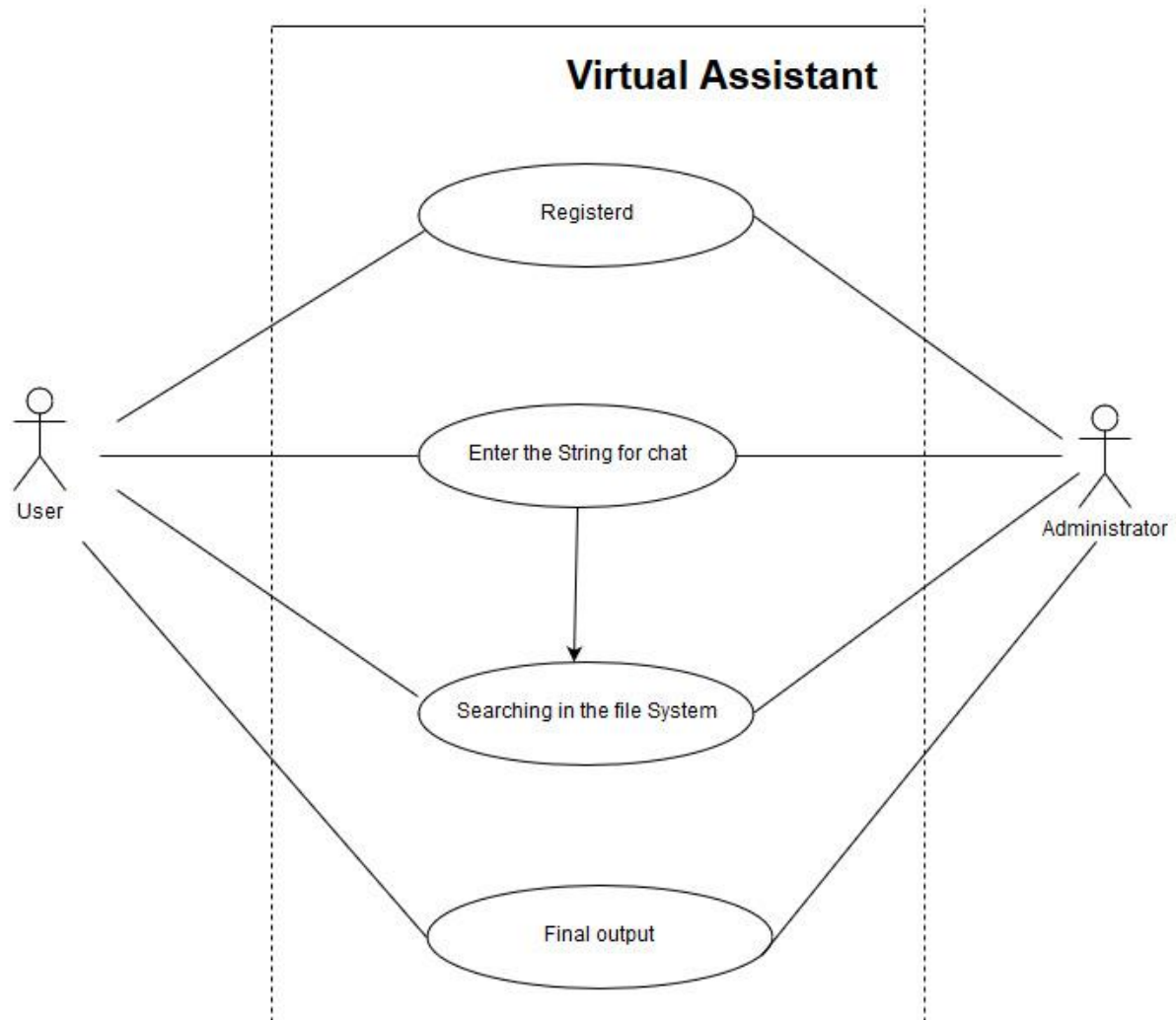


Figure 4.1.1: - Use Case Diagram

4.1.2 Activity Diagram

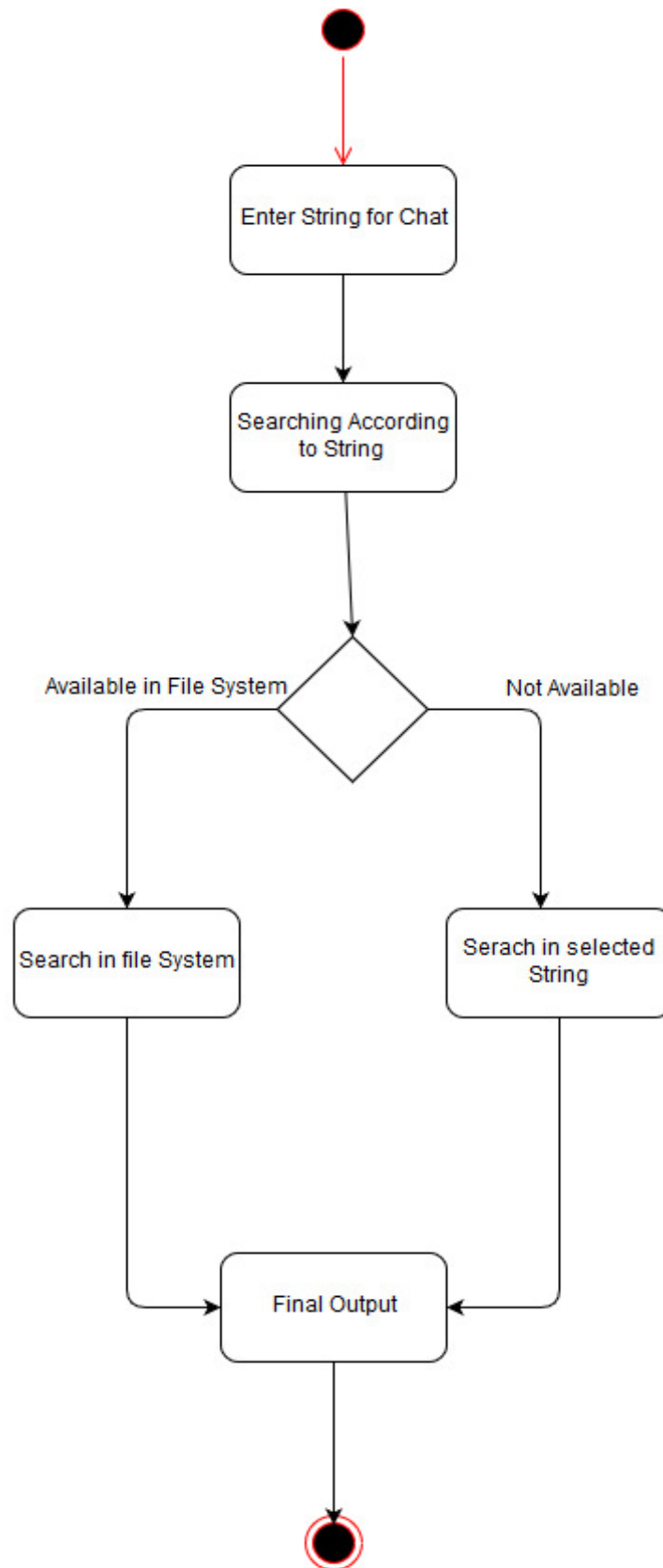


Figure 4.1.2 :- Activity Diagram

4.1.3 Sequence Diagram

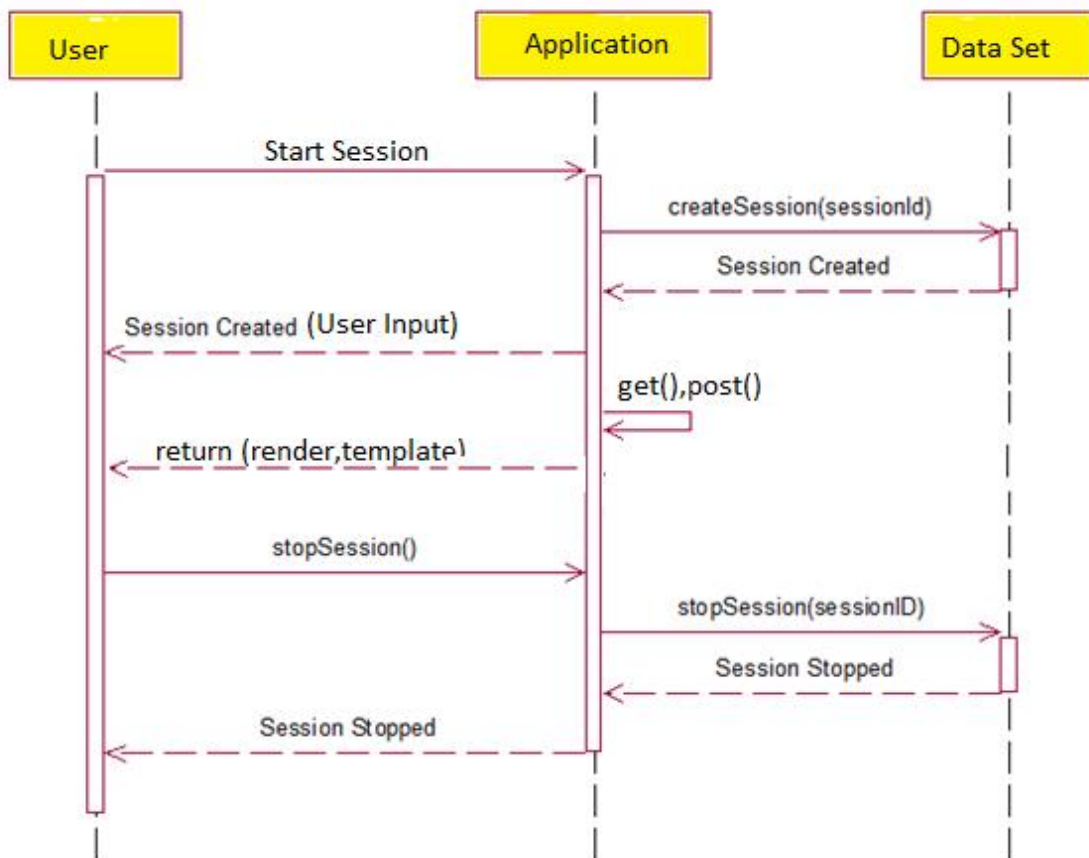


Figure 4.1.3: - Sequence Diagram

4.1.4 Class Diagram

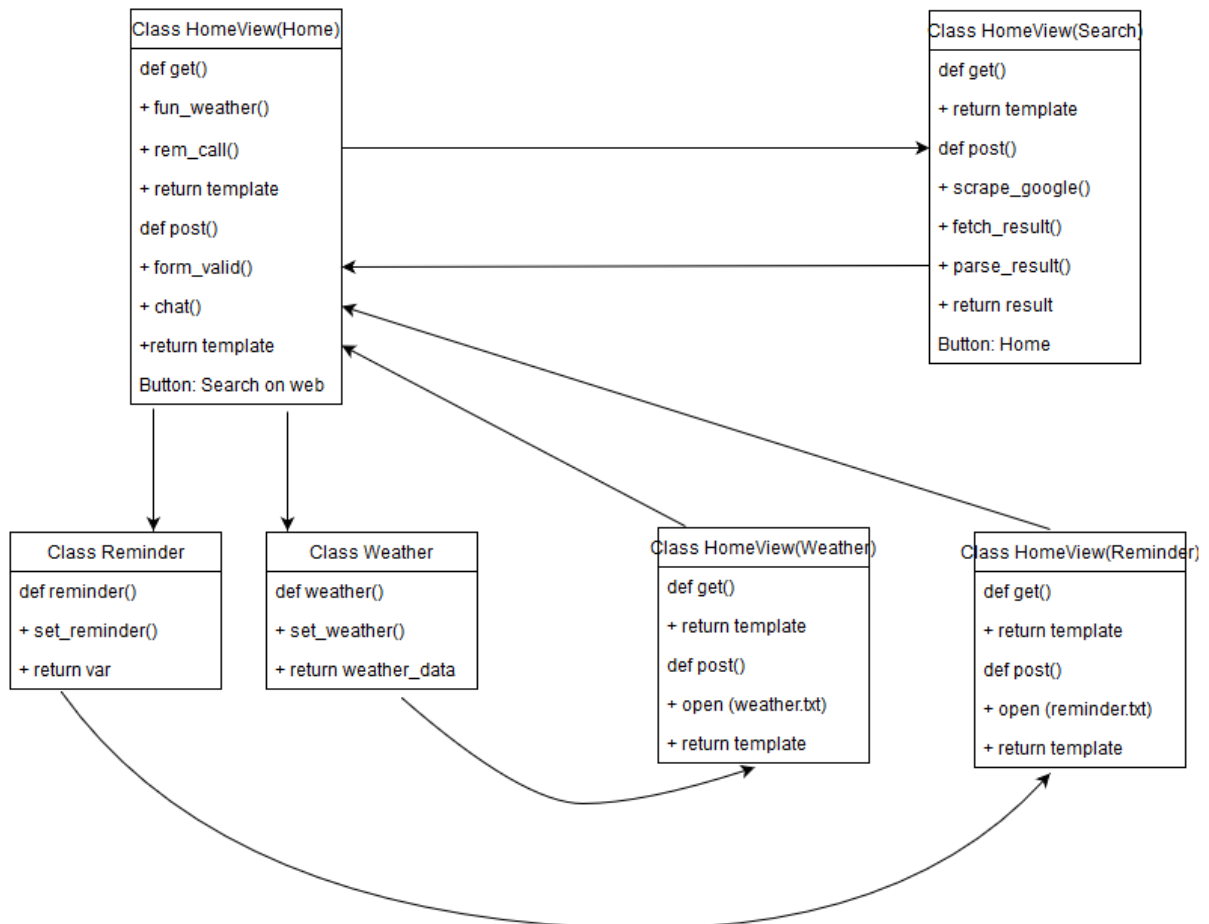


Figure 4.1.4:-Class Diagram

4.1.5 Data Flow Diagram

Level 0: -

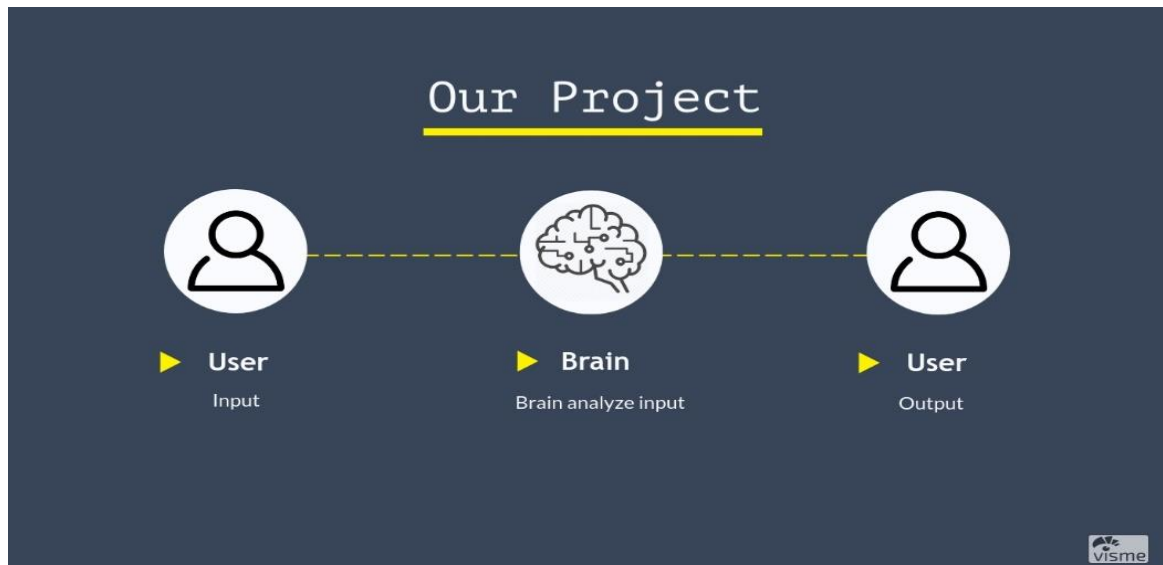


Figure 4.1.5 (a): - Data Flow Diagram

Level 1: -

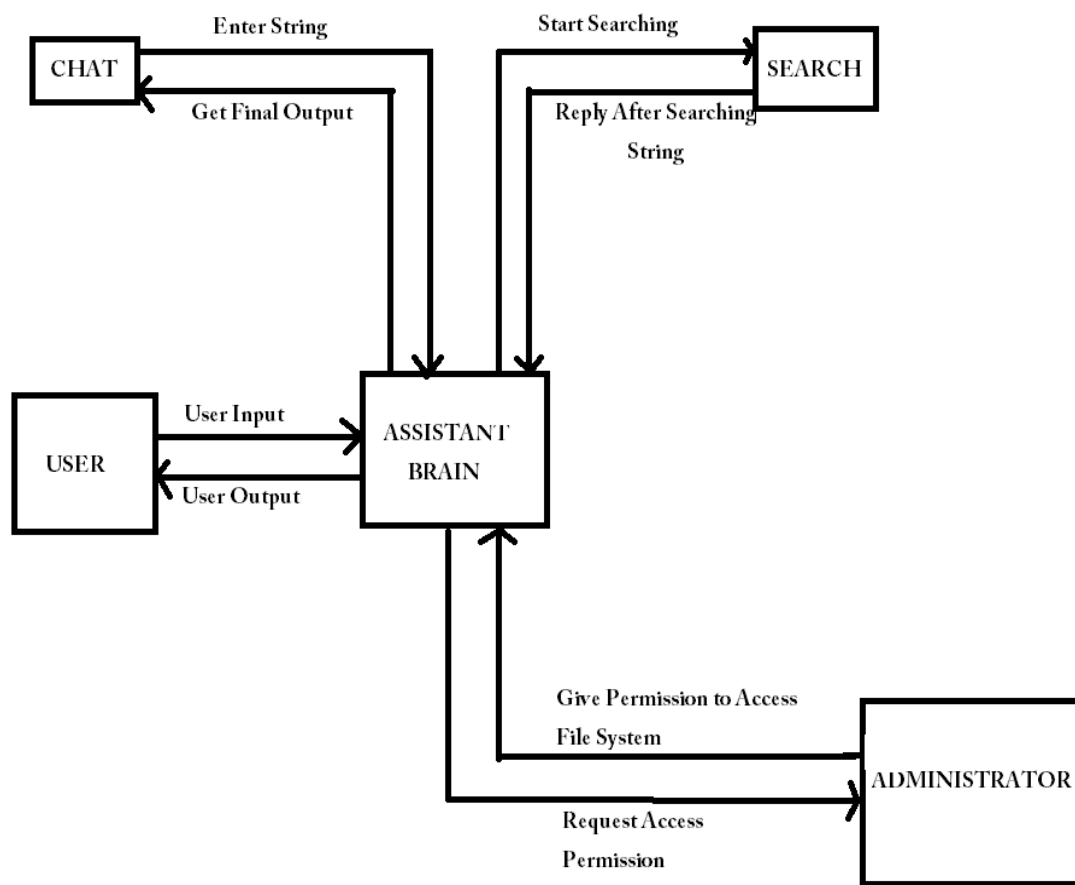


Figure 4.1.5 (b): - Data Flow Diagram

5. Project Implementation and Output Screens

5.1 Screenshots of our Project

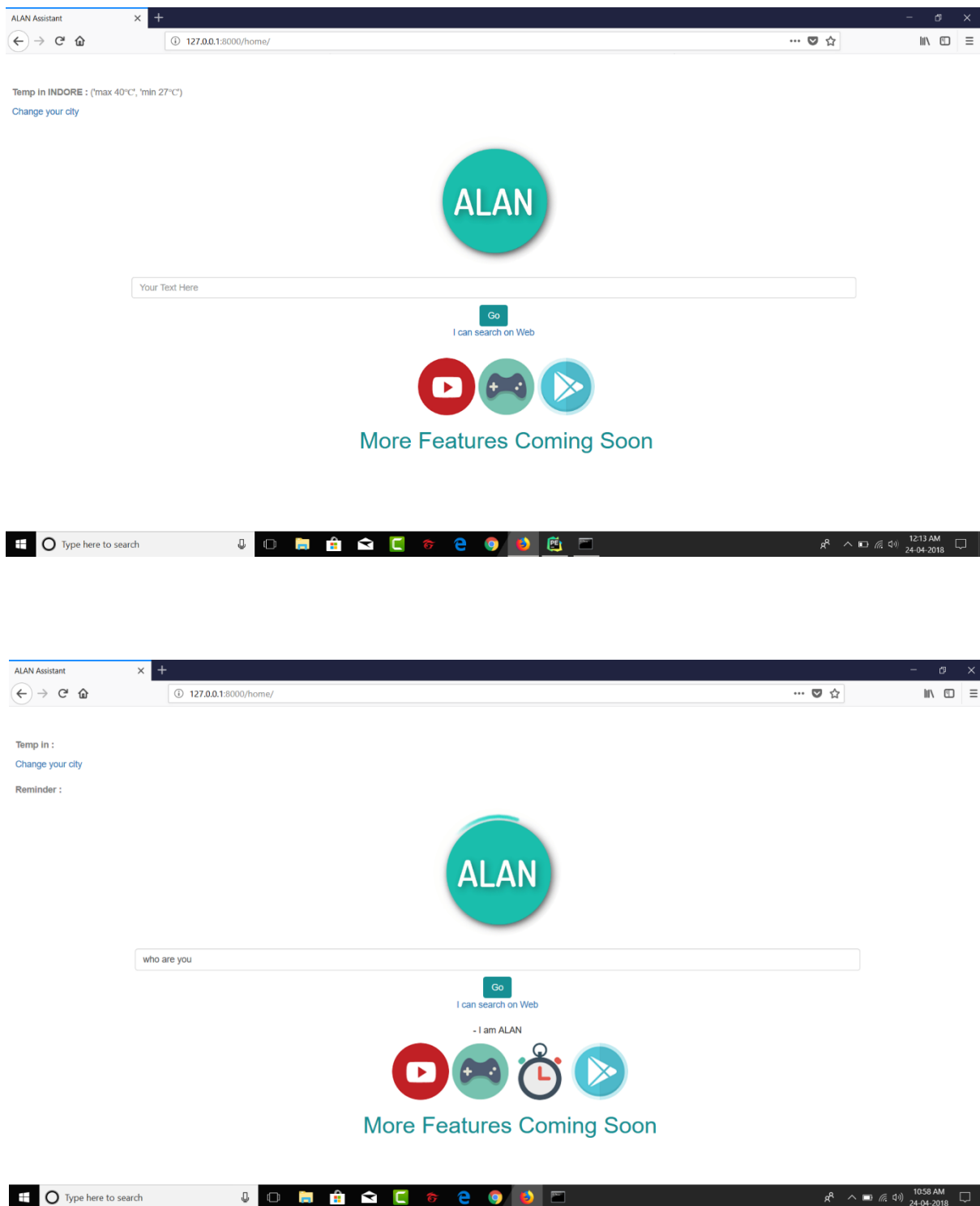


Figure 5.1(a)-Home Page

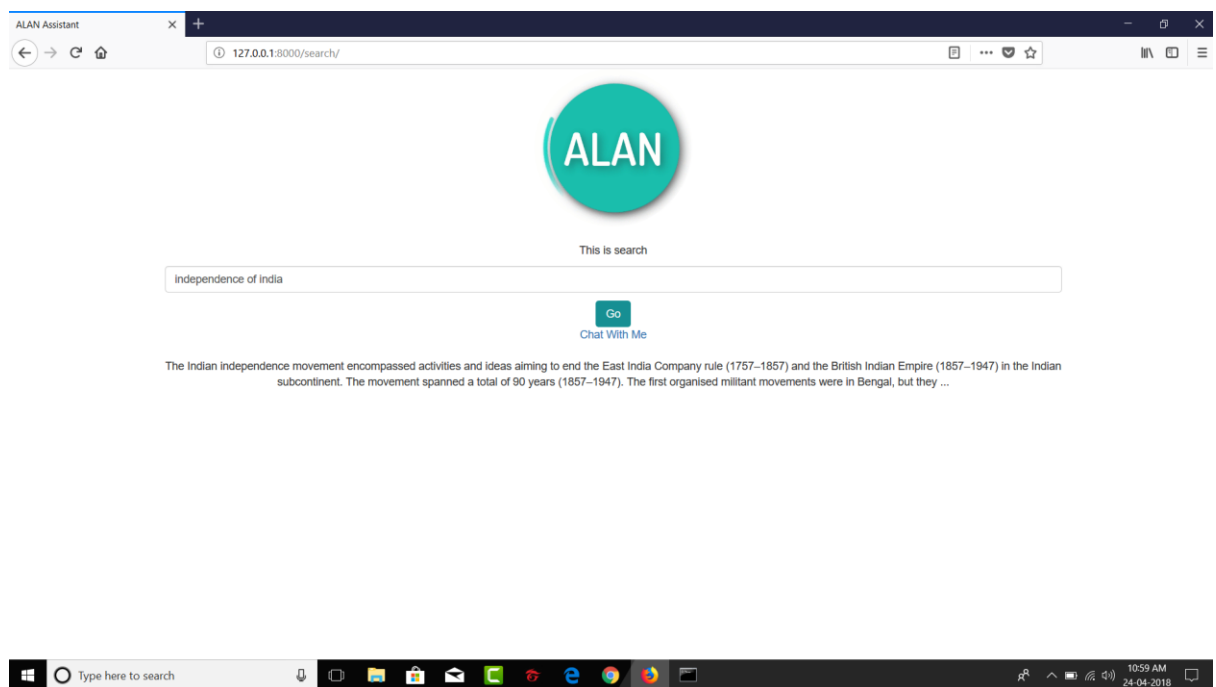
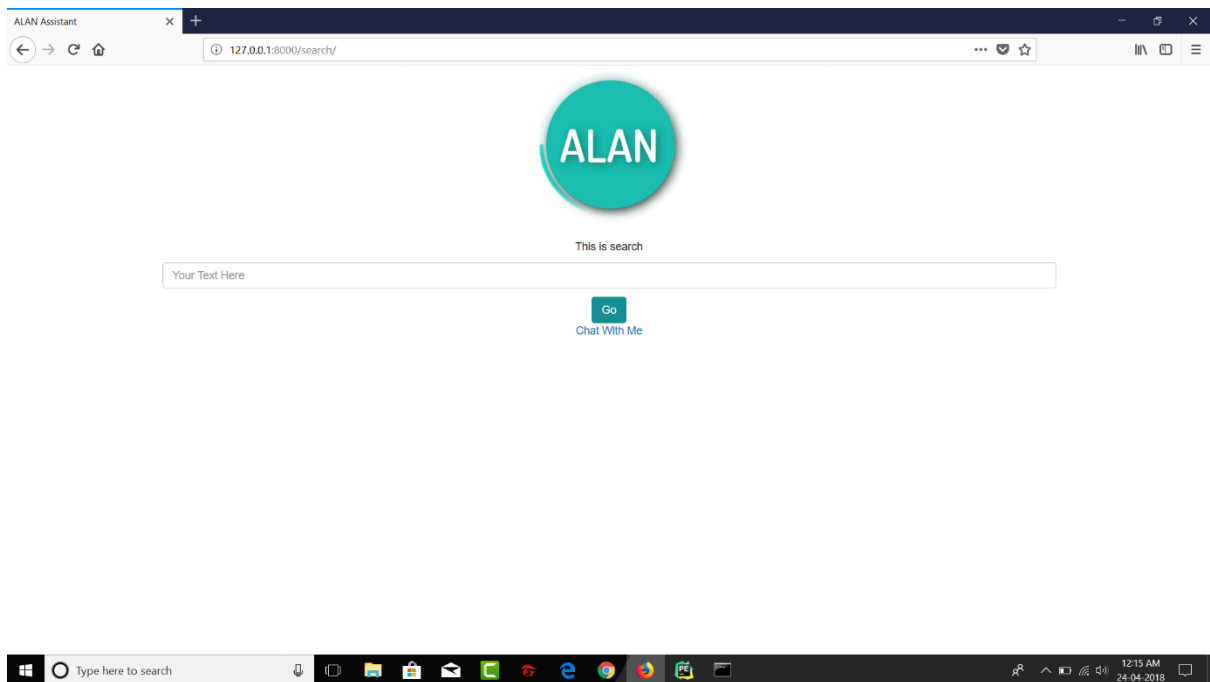


Figure 5.1(b)-Web Search

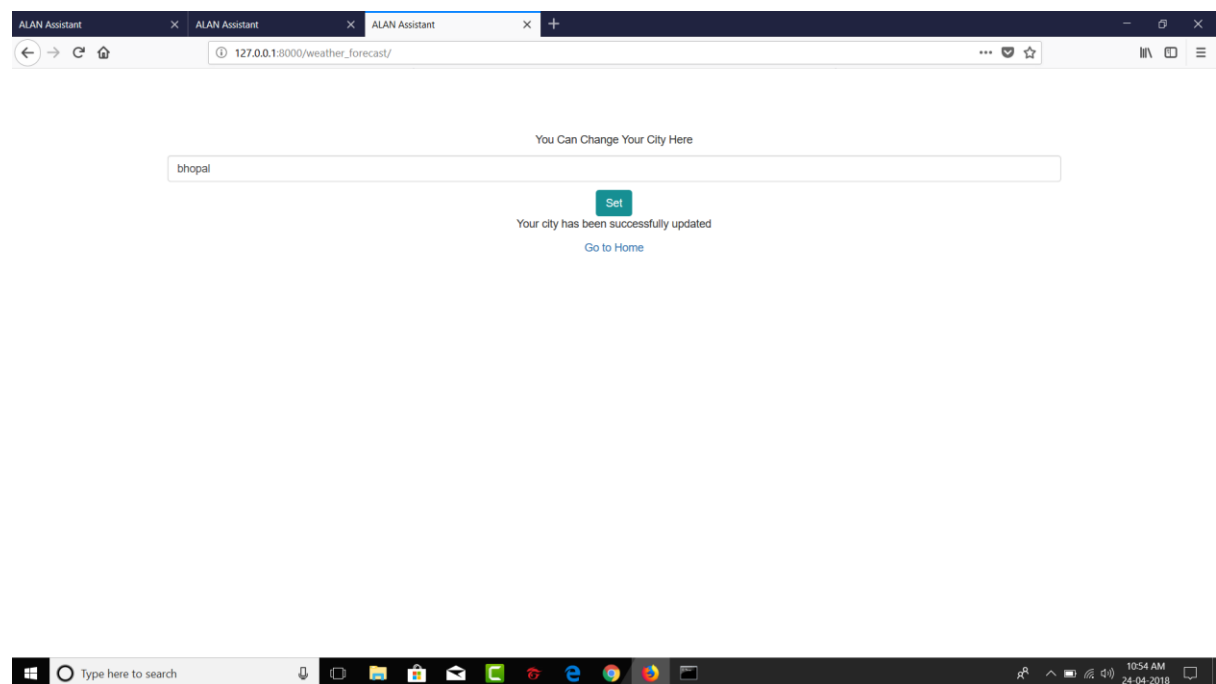
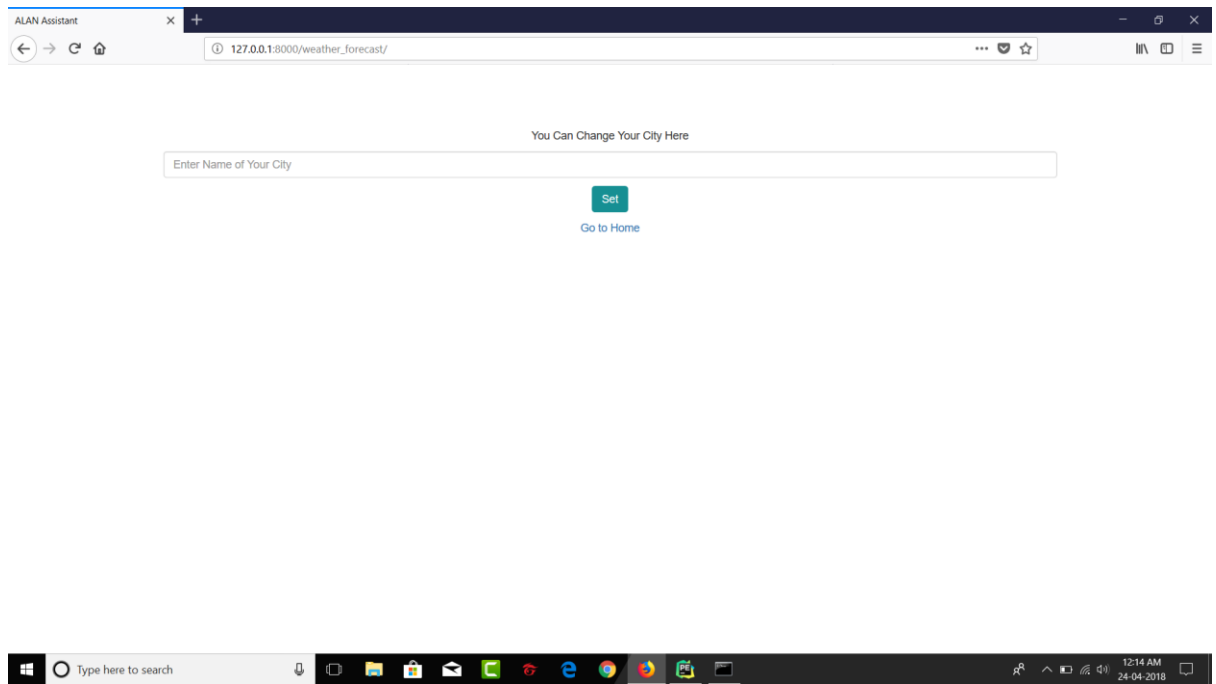


Figure 5.1(c)-Weather Forecasting



You can set your reminder here

24/04/2018
11:04
This is test reminder

Set

Your city has been successfully updated

[Go to Home](#)



Temp in BHOPAL : ('max 41°C', 'min 29°C')

[Change your city](#)

Reminder : This is test reminder



Your Text Here

Go

I can search on Web



More Features Coming Soon



Figure 5.1(d)-Reminder



Figure 5.1(e)-Game

5.2 Important Coding

Web Deployment

```

from django.views.generic import TemplateView
from home.forms import HomeForm
from django.shortcuts import render
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
from home.weather import Weather
from home.remainder import rem_call
obj = Weather()
class HomeView(TemplateView, Weather):
    template_name = 'home/home.html'

    def get(self, request):
        template_name = 'search/search.html'
        form = HomeForm()

```

```

## Weather Forecast Start
temp, city = obj.fun_weather()
## Reminder Start
reminder_result = rem_call()
args1 = {'form': form, 'temp': temp, 'city': city, 'reminder_result': reminder_result}
return render(request, self.template_name, args1)
## Reminder Stop
##Weather ForeCasting Stop
def post(self, request):
form = HomeForm(request.POST)
if form.is_valid():
    text = form.cleaned_data['post']
    bot.set_trainer(ListTrainer)
    while True:
        message = text
        if message.strip() != 'Bye':
            reply = bot.get_response(message)
            args2 = {'form': form, 'text': reply}
            return render(request, self.template_name, args2)
        break
    if message.strip() == 'Bye':
        break

```

Search Code

```

from django.shortcuts import render
from django.views.generic import TemplateView
from search.forms import HomeForm2
import requests
from bs4 import BeautifulSoup

class HomeView(TemplateView):
    template_name = 'search/search.html'

```

```

def get(self, request):
    #template_name = 'search/search.html'
    form = HomeForm2()
    return render(request, self.template_name, {'form': form})

```

```

def post(self, request):
    form = HomeForm2(request.POST)
    if form.is_valid():
        text = form.cleaned_data['post']
        data = []
        query = text
        results = self.scrape_google(query, 100, "en")
        for result in results:
            data.append(result)

        args2 = {'form': form, 'text': data[0]['description']}
        return render(request, self.template_name, args2)

```

```

def fetch_results(self, search_term, number_results, language_code):
    USER_AGENT = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36'}
    assert isinstance(search_term, str), 'Search term must be a string'
    assert isinstance(number_results, int), 'Number of results must be an integer'

    temp = search_term.split(" ")
    escaped_search_term = []
    for index in range(len(temp)):
        if temp[index] == "":
            pass
        else:
            escaped_search_term.append(temp[index])
    print(escaped_search_term)

```

```

        escaped_search_term = "+".join(escaped_search_term)
        print(escaped_search_term)

    google_url =
'https://www.google.com/search?q={ }&num={ }&hl={ }'.format(escaped_search_term,
number_results, language_code)
    response = requests.get(google_url, headers=USER_AGENT)
    response.raise_for_status()
    return search_term, response.text

def parse_results(self, html, keyword):
    soup = BeautifulSoup(html, 'html.parser')

    found_results = []
    rank = 1
    result_block = soup.find_all('div', attrs={'class': 'g'})
    for result in result_block:
        link = result.find('a', href=True)
        title = result.find('h3', attrs={'class': 'r'})
        description = result.find('span', attrs={'class': 'st'})
        if link and title:
            link = link['href']
            title = title.get_text()
            if description:
                description = description.get_text()
            if link != '#':
                found_results.append({'keyword': keyword, 'rank': rank, 'title': title,
'description': description})
            rank += 1
    return found_results

def scrape_google(self, search_term, number_results, language_code):
    try:

```

```

        keyword, html = self.fetch_results(search_term, number_results, language_code)
        results = self.parse_results(html, keyword)
        return results
    except AssertionError:
        raise Exception("Incorrect arguments parsed to function")
    except requests.HTTPError:
        raise Exception("You appear to have been blocked by Google")
    except requests.RequestException:
        raise Exception("Appears to be an issue with your connection")

obj = HomeView()

```

Reminder Code:

```

import datetime
import time
def remainder():
    now = datetime.datetime.now()
    today = now.strftime("%d/%m/%Y")
    totime = now.strftime("%I:%M")
    with open("Reminder.txt",'r') as file:
        #print(today,totime)
        for line in file:
            line = line.strip().split(" ")
            if today == line[0]:
                if totime == line[1]:
                    var = (" ".join(line[2:]))
                    return var
def rem_call():
    reminder_result = remainder()
    sleeptime = 60 - datetime.datetime.now().second
    time.sleep(sleeptime)
    return reminder_result

```

.....

Weather Forecast

```
import re

import urllib.request

class Weather:

    def fun_weather(self):

        file = open("WeatherData.txt", 'r')

        var_city = file.read()

        var_city = str(var_city)

        city = var_city.upper()

        file.close()

        url = "https://www.weather-forecast.com/locations/"+city+"/forecasts/latest"

        data = urllib.request.urlopen(url).read()

        data = data.decode('utf-8')


        m = re.search('span class="phrase"', data)

        start = m.end()

        end = start + 100

        maxtemp = re.search('max',str(data[start:end]))

        mintemp = re.search('min',str(data[start:end]))

        temp1 = maxtemp.start()

        temp2 = mintemp.start()

        newString = data[start+temp1:start+temp1+6]

        newString1 = data[start+temp2:start+temp2+6]
```

```
#print(newString + "\u2103",newString1 + "\u2103")  
  
output = newString + "\u2103",newString1 + "\u2103"  
  
return output, city
```

6. Testing

6.1 Testing Strategy Adopted

Various software-testing strategies have been proposed so far. All provide a template for testing. Things that are common and important in these strategies are: Testing begins at the module level and works “outward”: tests which are carried out are done at the module level where major functionality is tested and then it works towards the integration of entire system. Different testing techniques are appropriate at different point of time: Under different circumstances, different testing methodologies are to be used which will be the decisive factor for software robustness and scalability. The developer of the software conducts testing and if the project is big then there is a testing team: All programmers should test and verify that their results are according to the specification given to them while coding. In cases where programs are big enough or collective effort is involved for coding, responsibilities for testing lies with the team as a whole.

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

Reactive - An approach in which the testing is not started until after design and coding are completed.

There are many strategies that a project can adopt depending on the context and some of them are:

- a) Dynamic and heuristic approaches
- b) Consultative approaches
- c) Model-based approach that uses statistical information about failure rates.
- d) Approaches based on risk-based testing where the entire development takes place based on the risk
- e) Methodical approach, which is based on failures.
- f) Standard-compliant approach specified by industry-specific standards.

Testing Objectives

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates

that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

Objectives of testing are: -

- a) Finding defects which may get created by the programmer while developing the software.
- b) Gaining confidence in and providing information about the level of quality.
- c) To prevent defects.
- d) To make sure that the end result meets the user requirements.
- e) To ensure that it satisfies the SRS that is System Requirement Specifications.
- f) To gain the confidence of the customers by providing them a quality product.

6.2 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

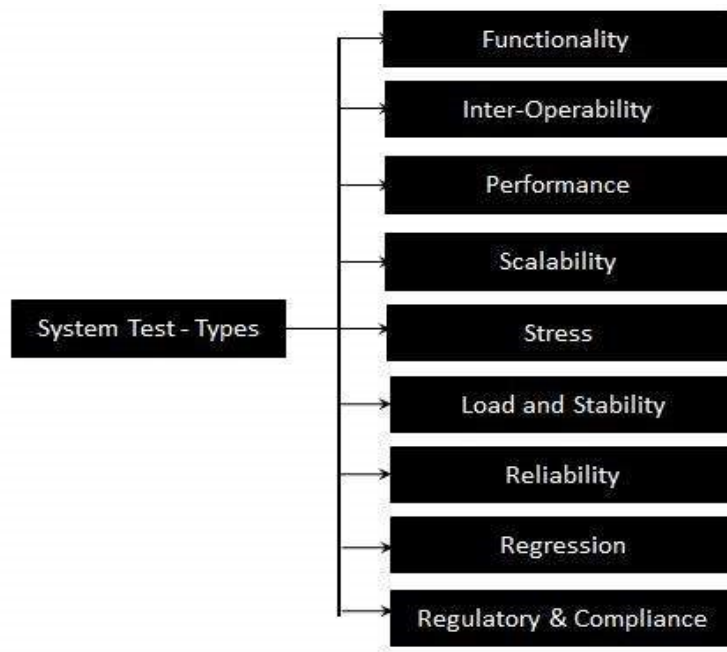


Figure 6.2:- System Testing Types

6.3 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

The following are the tests that are performed during the unit testing:

Data Fetch from server using Async Methods: - To check bus session and coordinates information fetched through server.

- a) **Network Failures:** - To check network failures if any while data transmission between client and server.
- b) **Boundary Conditions:** - It is observed that much software fails at boundary conditions. That's why boundary conditions are tested to ensure that the program is properly working at its boundary conditions.
- c) **Independent Paths:** - All independent paths are tested to see that they are properly executing their task and terminating at the end of program.

6.4 Test Plan and Test Cases

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software/product in a project. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice and any risks requiring contingency planning.

Features to be Tested

Features are changes that add new functionality or significantly modify existing functionality. They are designed to be useful, effective, and attracting. Though, users are inventive and use the functionality unexpected ways. Also the application is complex and all pieces have to cooperate with each other. Any change might have surprising side effects in the related parts. Finally, developers are just humans and they might simply forget to implement a piece.

Feature testing should catch the above mentioned problems and make sure that users will be happy with the new stuff.

Features to be tested:-

Chat Activity:-A chat activity is interacting to user on any topic that user likes to talk about. User can communicate with alan on basic topics which is needed to be tested.

Web Searching Activity:-User can basically search any question on web if he or she is not satisfied with the answer given by alan while chatting which is needed to be tested.

Weather Forecasting Activity:-User can know about weather of any city by entering city's name which is needed to be tested.

Reminder Activity:- User can set reminder which is needed to be tested.

Features not to be Tested

This is a listing of what is 'not' to be tested from both the user's viewpoint of what the system does and a configuration management/version control view. This is not a technical description of the software, but a user's view of the functions.

Some features are not to be tested and they include:

Games Activity:-User can also play interesting gamewhich is not needed to be tested.

Music Activity:-User can also listen music which is not needed to be tested.

Video Activity:-User can also watch videos which is not needed to be tested.

Test Cases

Test Case Number	Test Case	Description	Test Result
T-001	Chatting Module	It can interact with user.	Successful
T-002	Web Searching Module	It can search on web for user.	Successful
T-003	Weather Forecasting Module	It will display Weather report for a city entered by user.	Successful
T-004	Games Module	It can make user play game for entertainment.	Successful
T-005	Video Module	It helps you watch videos.	Successful
T-006	Audio Module	It helps you listen music.	Successful

Table 1:- Test Cases

7. Conclusion

7.1 Conclusion

This is to conclude that the project that we undertook was worked upon with a sincere effort working on the Web application we would like to conclude that most of the work in this Domain is currently worked upon by the Students, but still there are many things which are need to be addressed like for the application to work. This project would satisfy all the requirements of the college and would be beneficial for the users.

7.2 Future Scope

- Can be integrated for home automation.
- Advancing with help of advanced machine learning techniques.
- Payment modes.
- Assistant can be used in IOT(Internet of Things).

A. Appendix A

References

1. Available at <https://www.coursera.org/>
2. Available at <https://www.coursera.org/specializations/python>
3. Available at <https://www.djangoproject.com/>
4. Available at http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
5. Available at <https://chatterbot.readthedocs.io/en/stable/>

Bibliography

1. Python Programming for the Absolute Beginner, 3rd Edition
2. Learning Python, 5th Edition
3. Two Scoops of Django
4. Web Scrapping with Python