

Contents

Introduction	2
1.1 Purpose	2
1.2 Intended Audience and Reading Suggestions.....	2
1.3 Product Scope	2
1.4 Features:.....	3
1.5 References	3
Overall Description.....	3
2.1 Product Perspective	3
2.2 Operating Environment	4
2.3 Design and Implementation Constraints.....	4
2.4 Product Functions	4
2.5 User Classes and Characteristics	4
2.6 Design and Implementation	5
External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	7
3.4 Communications Interfaces	7
Modules-:.....	8
4.1 Division Of Modules between Team Members-:.....	8
Input Query:.....	12
5.1 Reminders and Alarms:	12
5.2 To-do and Shopping lists:	12
5.3 Opening Apps:	12
5.4 Conversations:	12
Other Nonfunctional Requirements	13
6.1 Usability:	13
6.2 Reliability:	13
6.3 Supportability:	13

Introduction

1.1 Purpose

Maya is designed in order to give its users an interactive user experience by socially communicating with the user along with providing useful information and services whenever required by the user. It can help its user in ways such as making todo lists, opening apps present in the system by name, reminding its users of dates and alarms set by the user etc. Maya can also help its users surf the web better by searching online for queries entered by the user and showing relevant results.

1.2 Intended Audience and Reading Suggestions

Maya is intended for all kind of audiences from kids to the adults. Its features can be an entertainment for kids with some parental control. For adults, it can be a useful asset as a day to day manager along with covering the aspects of reminder pads and alarm clocks. So, the spectrum of audience it has is quite large and considerably bulky which gives an opportunity for people to experience a new way of searching and managing their personal devices.

1.3 Product Scope

Maya is an AI chatbot that receives questions from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, when it receives the question "What time does the gym close today?", it will give a response "The gym closes at 10pm today." The main objective is creating a Web API, and sample web, mobile, and text messaging interfaces that demonstrate the use of the API. The goal is to provide Maya students and faculty a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate Maya into their projects.

1.4 Features:

- Intelligent Chatbot
- Web Scrapping
- Text to speech
- Speech to text
- Wikipedia crawling

1.5 References

Referred Websites

<https://www.google.com/>

<https://www.tutorialspoint.com/html/>

<https://www.w3schools.com/>

Referred Books

Machine Learning for OpenCV

Overall Description

2.1 Product Perspective

Most of the search engines today, like Google, use a system (The Page rank Algorithm) to rank different web pages. When a user enters a query, the query is interpreted as keywords and the system returns a list of highest ranked web pages which may have the answer to the query. Then the user must go through the list of webpages to find the answer they are looking for. Maya Chatbot, however, will try to understand the query and provide a definitive answer. There will be four main units to the system working together to understand the question and return an appropriate answer:

- Generic question construction - capable of taking a natural language question and making it more generic.
- Generic answer construction - capable of taking a generic question template and

providing a generic answer template. • Generic answer population - capable of taking a generic answer template and populating it with information from the database to form an answer. • Information extraction - capable of finding information through structured or unstructured websites, and storing that information in a database.

2.2 Operating Environment

Maya is developed in open source and user-friendly environment of python with machine learning implementation. Basic libraries such as numpy, pandas are also used. Source code has been coded in python which supports the application as well as database.

2.3 Design and Implementation Constraints

- **Language Requirement:** Python
- **Technologies:** Machine Learning
- **Tools:** GTTS, Google Speech to text, Wikipedia API ,Selenium, IDLE

2.4 Product Functions

The major features for Maya Chatbot will be the following: • Speech to text input: capability to take speech as input and convert it into text and later can be used as a web search query

- Language Processing: the system will take in questions written in standard English.
- Natural Language Responses: the answer to the question will be written in standard and understandable English.
- Opening Apps: ability to open applications present in the system.
- Reminders and Alarms: It can be used to set reminders and alarms for the user on a specific date or time.

2.5 User Classes and Characteristics

The two classes of users for this system are described below:

- API users API users consist of application developers who want to incorporate Maya Chatbot API into other software applications.

- app users these users consist of non-technical users who want to get answers for their questions. These users ask questions and get answers with mobile, web, or text messaging interfaces. This class of users includes Maya's current and prospective users.

2.6 Design and Implementation

Constraints Creating : An AI that can answer every single question is not possible to implement with current technology and within the duration of the project, so the system will be able to answer questions about limited topics. The system will only support questions in standard English.

Each feature will be assigned an importance value. The project will be complete if all the features of Priority 1 and at least 50% of features of Priority 2 are implemented. No Priority 3 requirements are necessary.

Priority Meaning

- 1 Speech to text input, Web surfing
- 2 Reminders, Alarms
- 3 Casual and random talks

External Interface Requirements

3.1 User Interfaces

Application GUI Framework used:- TKINTER

1. The GUI will have a textbox that will accept inputs from a keyboard.
2. Text box will originally contain a suggestive text question, to guide the user to the format of an appropriate question.
3. The GUI will have a “Send” button which sends text from the textbox to the API when clicked.
4. The GUI will have a chat window displaying questions sent to the system and responses from the API.
5. The chat window will contain all questions and answers from the current session, with a scroll bar if all messages can’t fit on the screen.
6. If there is a network issue, the chat window will display an error message.
7. There will be a designated phone number that users can send texts to.
8. Texts sent to that number will be sent to the API, then the system will reply to the user with the answer as another text message.
9. If a question is not understood by our API, the system will send a text containing an example question after the text with the API response.

3.2 Hardware Interfaces

1. A touch screen or any other pointing device is required by the user in order to interact with the system.
2. A microphone is required to implement speech to text.
3. Speakers are required to receive audio outputs.
4. Keyboard is required to Input search queries and other text inputs.

3.3 Software Interfaces

1. Libraries such as numpy, pandas etc.
2. Python 3
3. PYTHON IDE such as spyder or PyCharm.
4. System Features.

3.4 Communications Interfaces

The communication between the different parts of the system is important, since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems. The most important communication takes place between the database and the microphone as the storing and retrieving process are taking place whenever the user is using the software.

Modules-:

.git	11/10/2018 12:00	File folder	
bin	11/10/2018 10:58	File folder	
lib	11/10/2018 11:08	File folder	
src	14/10/2018 00:15	File folder	
aigui1	24/10/2018 19:37	Python File	2 KB
assiatant	11/10/2018 23:10	Python File	1 KB
README.md	11/10/2018 10:57	MD File	1 KB

Fig 1.0 Main Module

__pycache__	11/10/2018 11:58	File folder	
mail	11/10/2018 13:47	File folder	
other	14/10/2018 00:15	File folder	
SearchWeb	11/10/2018 23:02	File folder	
SpeechRecognition	11/10/2018 11:27	File folder	
text2speech	11/10/2018 11:27	File folder	
wiki	11/10/2018 11:27	File folder	
wolframalpha	11/10/2018 11:27	File folder	
__init__	11/10/2018 11:56	Python File	0

Fig 1.1 Supporting Modules

4.1 Division Of Modules between Team Members-:

1. Manzar Nadeem

- Mail-:** The **Mail Library** of python is used to send mails according to instruction given. It is initiated by calling maya mail. Then it will ask for the sender, receiver email ids and message. It will authenticate the sender email and send the message.
- Search Web-:** The **Webbrowser Library** is used to open the chrome and search for specific query on google as instructed by the user. This is mainly when the **Maya** is not able to identify the keyword in its on database.
- Others-:** The **OS Library** is used to implement normal operating system related activities by the virtual assistant such as ShutDown, opening apps etc.

2. Amitabh Kumar

- Speech Recognition-:** The google's **speech_recognition library** is used to recognize the audio tokens from the audio source such as microphone and then convert them to text. It is one of the core component of **Maya**. This module is used in other modules.

- b. **Text to speech:-** The **Google text to speech Library** is used to convert the text to audio then this audio is stored as mp3 file in the folder. To run this audio file **Pyglet library** is used and then this file is deleted using **OS library**.
- c. **Scraping:-** The **Wikipedia library** and **selenium webdriver** is used to automate scraping and find results on the web. **Wikipedia library** is used to fetch the summary of the query asked by users from Wikipedia. The **selenium webdriver** is used to automatically search and play videos from the youtube on the request of user.
- d. **GUI:-** The **Tkinter library** is used to create the Buttons, labels, and textfields for the GUI for virtual assistant. That will facilitate user's interaction with the application, in other words making it more user-friendly.

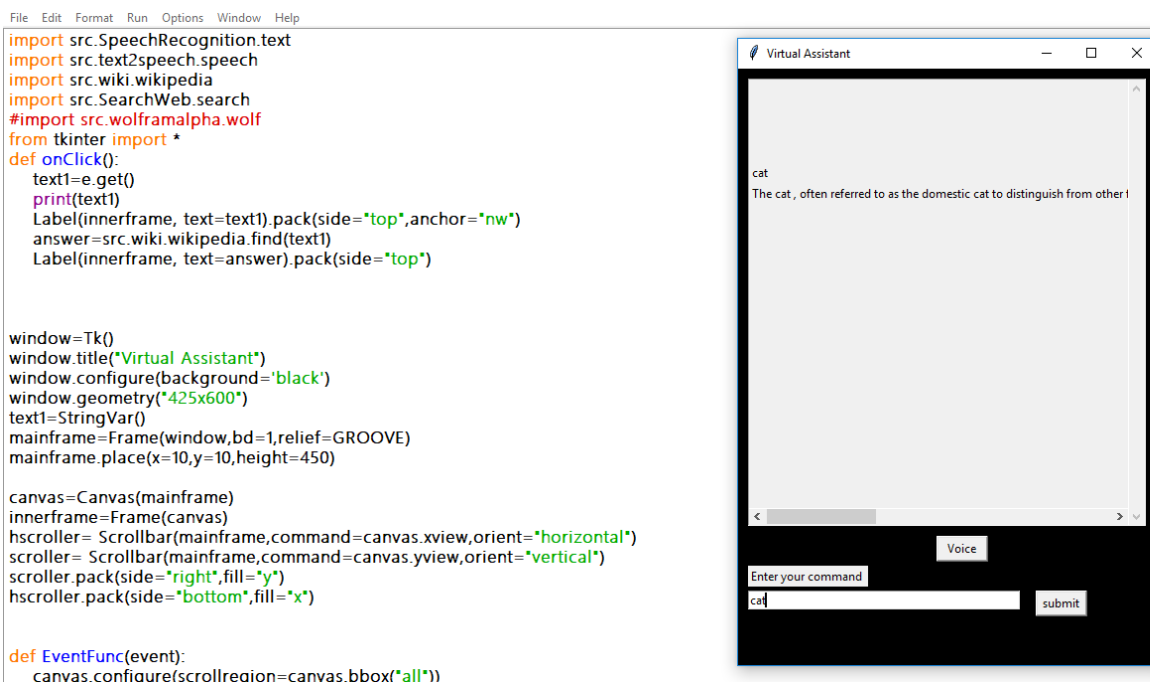


Fig 2.0 GUI implementation

```

import wikipedia
import os,re
def find(query):
    summary=wikipedia.summary(query,sentences=2)
    summary=re.sub(r"\([^\)]*\)", "", summary)
    return summary
  
```

Fig 2.1 Wikipedia module

```

from gtts import gTTS
import pygame
import time, os
def speak(text):
    file=gTTS(text, 'en')
    filename='temp.mp3'
    file.save(filename)
    music= pygame.media.load(filename, streaming=False)
    music.play()
    time.sleep(music.duration)
    os.remove(filename)

```

Fig 2.2 Text to speech Module.

```

File Edit Format Run Options Window Help
import speech_recognition as sr
def ask():
    r=sr.Recognizer()
    r.energy_threshold = 2000
    with sr.Microphone() as source:
        print("say something")
        audio=r.listen(source)
    try:
        result=r.recognize_google(audio)
        print(result)
        return result
    except:
        pass

```

Fig 2.3 Speech Recognition Module

```

File Edit Format Run Options Window Help
import webbrowser as wb
def find(query):
    chrome_path="C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s"
    query='https://www.google.co.in/search?q='+query
    wb.get(chrome_path).open(query)

```

Fig 2.4 Web search module.

```
File Edit Format Run Options Window Help
import os
def shutdown():
    os.system('shutdown -s')
def offline():
    quit()
```

Fig 2.5 others Module.

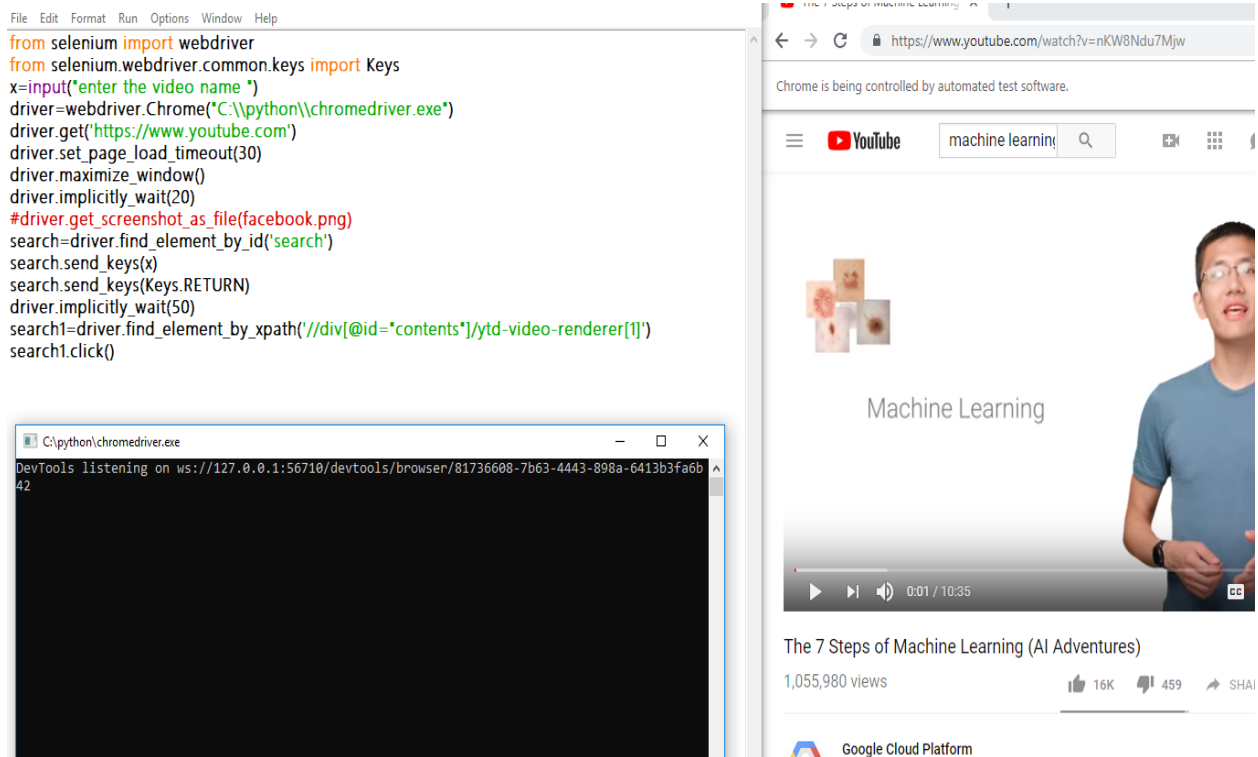


Fig 2.6 Web scarping using selenium library.

Input Query:

The input query can be in form of text or speech. It uses Google's Speech to text input API in order to understand the speech of the user and converts it into text format.

This can later be used as a web search query and be searched for using any browser.

5.1 Reminders and Alarms:

Maya can be used in order to set reminders and alarms for the user on a specific date or time. Dates like birthdays, anniversaries and other special occasions will never be forgotten.

The schedule will be strictly followed by the user as data stored by the user in Maya will constantly remind the user.

5.2 To-do and Shopping lists:

Shopping lists will be converted from boring pen and paper stuff to digitally corrected lists with checkboxes to indicate whether that particular item has been purchased or not. Same goes for to-do lists. Checkboxes indicate whether the task corresponding to it has been completed by the user or not.

5.3 Opening Apps:

Maya can be used to open installed apps on the system. It recognises the app by its name which is provided by the user in form of text or speech.

5.4 Conversations:

Maya is capable of doing some casual conversations with the user at times as per the will of the user.

Other Nonfunctional Requirements

6.1 Usability:

Maya is easy to use with well defined GUI interface and visible controlling systems.

6.2 Reliability:

This app is reliable and is capable of handling bugs and errors made by the user. Along with the autocorrection features implemented by the in-system software, it is highly reliable and transparent in its working.

6.3 Supportability:

This app is supported on a variety of machines, those of which supports python-based application software and the capability to implement them.