

FAMBOT: SOFTWARE BASED FAMILY ROBOT

BY

HASIBUL KABIR EMON

ID: 181-15-10673

ANIMA PAUL

ID: 181-15-10588

MONJURUL ISLAM SHAKIL

ID: 181-15-10804

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Nazmun Nessa Moon

Associate Professor

Department of CSE

Daffodil International University

Co-Supervised By

Md. Tarek Habib

Assistant Professor

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2021

APPROVAL

This Project titled “**FamBot: Software Based Family Robot**”, submitted by **HASIBUL KABIR EMON ID:181-15-10673, ANIMA PAUL ID:181-15-10588, MONJURUL ISLAM SHAKIL ID:181-15-10804** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 2nd January 2022.

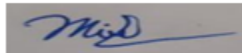
BOARD OF EXAMINERS



Dr. Touhid Bhuiyan
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

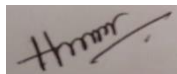
Chairman



Moushumi Zaman Bonny
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Mahfujur Rahman
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md Arshad Ali
Associate Professor

Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Nazmun Nessa Moon, associate professor, department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



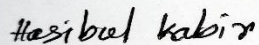
Ms. Nazmun Nessa Moon
Associate Professor
Department of CSE
Daffodil International University

Co-Supervised by:

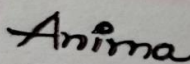


Md. Tarek Habib
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:



HASIBUL KABIR EMON
ID: 181-15-10673
Department of CSE
Daffodil International University



ANIMA PAUL
ID: 181-15-10588
Department of CSE
Daffodil International University

monjurul

MONJURUL ISLAM SHAKIL

ID: 181-15-10804

Department of CSE

Daffodil International University

ACKNOWLEDGEMENT

Firstly, we are expressing my heartiest thanks and gratefulness to almighty God for His divine blessing makes me possible to complete the final year project/internship successfully.

We are really grateful and wish our profound our indebtedness to **Ms. Nazmun Nessa Moon, Associate Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*web based*” to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Syed Akhter Hossain, Professor and Head, Department of CSE, for his kind help to finish my project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must be acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

This project “**FamBot: Software Based Family Robot**” carries a sole idea of ending the distance between family members and also as a leisure time chatting bot. This project is for a family to be used on a daily basis so that the members can stay in touch through this bot in this era of digital life. Now-a-days almost everyone has a smartphone or smart gadgets and people are busy with these most of the time. Even family bonding are also so fragile that there should be something which can reunite family members together in a digital way as we cannot just ignore the fact of importance of those devices in daily life. So why not make a bot and let it do the rest. This software based project will ensure that the time we use in the virtual world is somewhat used for a great purpose which is connecting the family members through the bot itself. This software will be helpful for both connection between family members and for getting to know useful information like what Google Assistant or Alexa does. To develop this project, we have used Excel, Python 3.9, initially in PyCharm and Google Collab for testing out different parts of the code.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iv
Abstract	v
List of Figures	ix
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
CHAPTER 2: BACKGROUND	4-7
2.1 Introduction	4
2.2 Related Works	4
2.3 Comparative Studies	6
2.4 Scope of the Problem	7

2.5 Challenges	7
CHAPTER 3: REQUIREMENT SPECIFICATION	
3.1 Business Process Modeling	8-16
3.2 Requirement Collection and Analysis	8
3.3 Use Case Modeling and Description	9
3.4 Logical Data Mode	15
3.5 Class Diagram Model	16
	17
CHAPTER 4: DESIGN SPECIFICATION	
4.1 Front-End Design	18-25
4.2 Back-End-Design	18
4.3 Interaction Design and UX	20
4.4 Implementation Requirements	24
	25
CHAPTER 5: IMPLEMENTATION AND TESTING	
5.1 Implementation of Data	26-28
5.2 Implementation of Front-end Design	26
5.2.1 User Module	27
5.3 Testing Implementation	27
5.4 Test Results and Reports	28
	28
CHAPTER 6: CONCLUSION AND FUTRUE SCOPE	
6.1 Discussion and Conclusion	29
6.2 Scope for Future Developments	29
	29

REFERENCES

PLAGIARISM

30

Plagiarism Report

31

31

LIST OF FIGURES

FIGURES	PAGE NO
Figure 2.1: Related works [Bocco, Alexa, Buddy]	5
Figure 3.1: Business Process Model for work process	8
Figure 3.2: Python Version Specification	13
Figure 3.3: Use Case Diagram of FamBot	15
Figure 3.4: ER Diagram of FamBot	16
Figure 3.5: Class Diagram of FamBot	17
Figure 4.1: Front Page of FamBot	18
Figure 4.2.1: Basic Functionality Used for Common Conversation	20
Figure 4.2.2: Voice Input Using Try and Except Block	21
Figure 4.2.3: Tokenizing Data	22
Figure 4.2.4: Algorithm Used on Tokenized Data	23
Figure 5.1: Visual of Stored Data Along with Date and Time	26

CHAPTER 1

INTRODUCTION

1.1 Introduction

This software installed on a hardware will be easy to use. We can see a large number of families out there are facing lack of connection between its members and facing this particular family issue is not a very new subject to us now. We thought out what is basically destroying this very pure relationship between family members and came to a conclusion that not consulting is one of the major problems. Consulting with family members on any major or minor issues in a regular basis would vanish this problem in a second. So we tried making a bot which would be interesting as well as helpful and acting as a connecting point of a family. This bot will listen to one family member and pass it to another; as simple as that but with a great purpose.

There will be no login feature or anything in this family bot. It will be a screen on a device and all one has to do is come to it and talk to it. It will reply to one's basic questions and most importantly remember what is said to it with the permission of the user and will like to pass the information when another one is engaging with it.

1.2 Motivation

There are many chatting bots which includes so many attractive features. But most of them out there serves for one single user. For this reason, we tried to make one bot that works with a family where it has a goal of reducing the distance between family members.

In this platform one family member can pass something so valuable that maybe he or she cannot say in real because of the distance already there is. It will eventually help out in being practical and gaining confidence. A simple message sometimes can mean a lot to other specially when inside a family.

.

1.3 Objectives

- To act as a blessing in a family.
- To get necessary information and get them stored.
- Not to let anything rest unspoken which should have been discussed.
- To get an assumption which information should be passed when asked for.
- To reduce complexities, anxiety or confusion of every family member.
- To save family bonding.
- To enjoy other available facilities the bot provides.

1.4 Expected Outcome

1. By using this software family members will be able to manage the relationship between themselves.
2. Family members will get the opportunity to use all the facilities provided by the FamBot.
3. Family members will be able to check what important note any other specific member has passed to them through bot
4. Communicating with the family member will be so much easier in between our daily busy life.
5. A member will not have to message every single person rather let the bot know if there is something that should be known to others.
6. It will save their time, reduce complexities and other heavy factors.

1.5 Report Layout

Chapter 1: Introduction

Motivation, objectives and expected outcome.

Chapter 2: Background

Related works, comparison, scope of the problem and challenges.

Chapter 3: Requirement Specification

Business process modeling, requirement collection and analysis, use case model, logical relational database model and design requirements.

Chapter 4: Design Specification

All the designs of the project; front-end design, back-end design, interaction design and UX and the implementation requirements.

Chapter 5: Implementation and Testing

Implementation of database, front-end designs and interactions, testing implementation, test results and reports.

Chapter 6: Conclusion and Future Scope

Conclusion and future scope of the project.

CHAPTER 2

BACKGROUND

2.1 Introduction

We had to study and research about the related projects and works done previously, for starting this development project. We have got some relevant works information which was actually turned to be helpful to complete this project. By identifying their limitation, we worked on their limitation and improvised. And finally, we built a new bot for family of our own. Without researching related works it would be a messed up situation finding where to start and where to focus.

2.2 Related Work

Following figure 2.1 shows some of the related works which provide family support through its features:

- Bocco [1]
- Buddy [2]
- Alexa [3]



Figure 2.1: Related works [Bocco, Alexa, Buddy]

Features: These are basically robots and software along with some cool features:

- Bocco mainly can be accessed online by parents to send voice messages to home and acts like an alternative to smartphones. Sensors on the door is one of the features which is actually helpful and very protective for family's safety. [1]
- Buddy tries to recognize family members and fulfill their personal needs. [2]
- Alexa helps surfing the internet with voice commands and do many other interactive works. [3]

Problems:

- Most of them doesn't have a concern in family bonding.
- Can be very expensive and not improve the status of bonding.
- Privacy concern can be an issue.

2.3 Comparative Studies

In our software we have many features. Some are similar and some are different comparing to the mentioned works done before. Moreover, we have developed it with a sole purpose and added additional features which are more user friendly to users.

An authorized user can take full access to the software including its insides and develop it according to his or her own preferences.

- A family member can chit chat with the bot continuing with normal conversation.
- One can drag some informative conversation like asking for a solid reply with correct attributes.
- We have used excel sheet to put the information into as this bot will remain in the family area and doesn't need a high level security.
- Commonly asking time or ordering it to do something is possible with this bot. More other ideas are attached in the string to meet its features as promised.

2.4 Scope of the problem

We have faced many problems to complete this project. But we have overcome these difficulties. For database choice we choose excel sheet. PyCharm studio as editor. We used python coding for designing the view of the software. As like HTML, CSS we have used some library function of python for designing this software. We have chosen this programming language because it will be easy to handle for developers as well as for us. As this software is super user friendly, we hope many families will be using it in current time if fueled properly.

2.5 Challenges

Here are some of the challenges which we have faced mentioned below:

- Python libraries had so much to provide us and we had to learn about them.
- Looking for the perfect idea to come up with.
- Developing a user friendly interface.
- Handling code and data saving area.
- Tried to learn speech recognition but couldn't really overcome.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Processing Model

Business processing model defines the flow of the data included. A structural view of the project is built through the process of this model. Following figure 3.1 describes the work process of business process model.

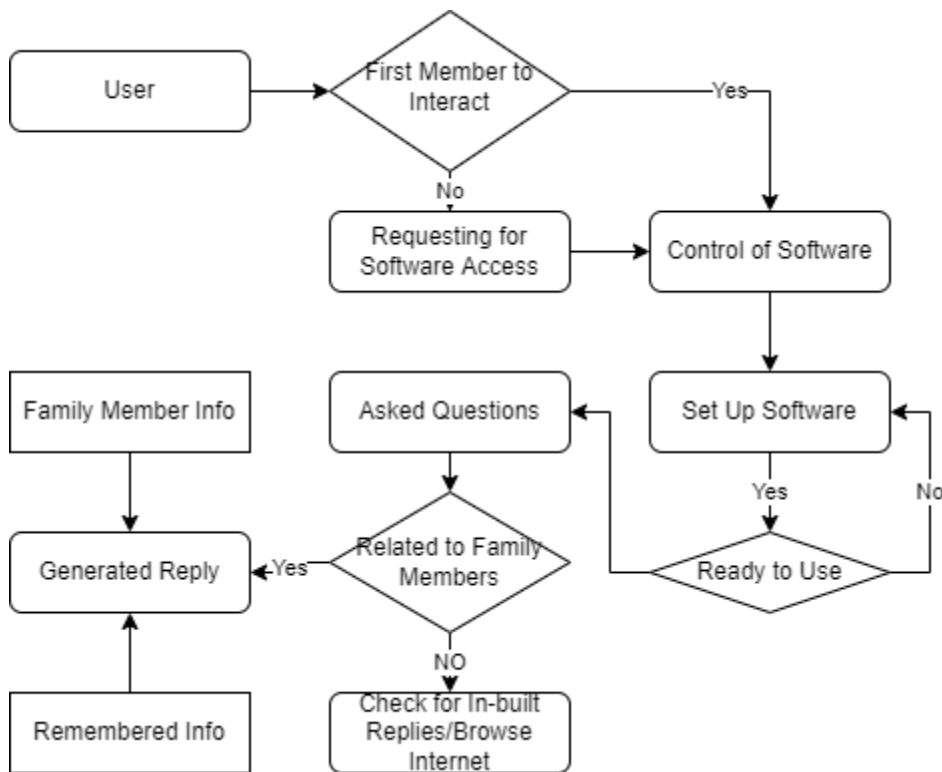


Figure 3.1: Business Process Model for Work Process

3.2 Requirement Collection & Analysis

We need to collect specifying the requirement for the initial work of this developing project. We had to fulfill the requirements of drawing a graphical view of the project for better understanding. We collected information from many different sources. Searched online for related works. We had to think of a basic plot and implement ideas on that plot and stitching them together. We analyzed our project by sharing our idea with friends. We received warm feedbacks almost from everyone. Thus we were able to fix our project destination and could work with confidence.

Programming Language Python:

Python is an interpreted programming language. It is a high-level general-purpose programming language. Code readability is its main concern. It is designed by following the code readability philosophy with the use of significant indentation. Its object-oriented approach as well as its language constructs helps programmers write simple, less error, less syntax, clear, logical code for small and large projects.

Python consistently ranks as one of the most popular programming languages.

In this project we used python 3.9. It is perfectly compatible for our work. There some package that we used in this project are described below.

speech_recognition: Perform speech recognition. Support online and offline engines and api.

Its support:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (works offline)

By using this library the project able to take speech input from the microphone, and it can transcribe any audio file or audio data, it can view results, can calibrate the recognizer energy threshold for ambient noise levels, it can listen through microphone in background.

Requirement:

- Python 2.6,2.7 or 3.3+
- PyAudio 0.2.11+
- PocketSphinx
- Google API Library Client
- FLAC encoder

PyAudio is important for getting input through by microphone. Without this Item everything will be work fine but microphone won't work. In our project the main goal is to take information from human voice so it is may be mentioned as the most important part of our work. So it must be ensure first that the pyaudio is working Properly.

pytt3: It does the opposite work of speech_recognition library. It actually make the speech from the text. It can do it by itself offline.

Python 2 and 3 both are compatible for this.

It requires pypwin32 module. It should be in the project or it will show error as "No module named win32".

pywhatkit: It has some special feature I can send message and image in WhatsApp, Convert image to ASCII art, convert string to handwriting, can play youtube videos, can send mails with html code.

datetime: By this library we obtain the time of something happening and can measure the laps between multiple event.

Wikipedia: it is a library that make a span between code and the site wikipedia. By this library program parse important info from there then send it for analyzing for other activities.

Pyjokes: This is a project for online code for joke. We have added some interesting feature in our project that some one don't get bother by using it. As a smile is helpful for sound mind so user can hear joke by using our complete project.

Openpyxl: Openpyxl is used to read/write excel 2010 xlsx/xlsm/xltx/xltn files. Use defusedxml can guard against quadratic blowup or billion laughs xml attack.

Warnings: Warning is an important library for developer to get notified about any situation of his work.

nltk: NLTK is a python library that process natural language. Our project is about processing natural language of human and process them according to user satisfaction reply from there. So nltk is the most important part of our project.

It requires python 3.6,/3.7/3.8/3.9+

String: String is a datatype of python introduced as a package.

random: built-in module. It can generate something randomly or choose randomly or finish randomly. [4]

sklearn.feature_extraction.text.TfidfVectorizer : The module extraction from raw data. It includes image and text. feature_extraction.DictVectorizer transform lists of feature value mapping to vector. The .text submodule collect to build feature vector. text.CounterVectorizer create matrix of token from document. text.TfidfTransformer create normal tf-idf transformed from matrix. Text.TfidfVectorizer make a collection of raw documents to Tf-IDF features. To make it smart don't going to deep description.

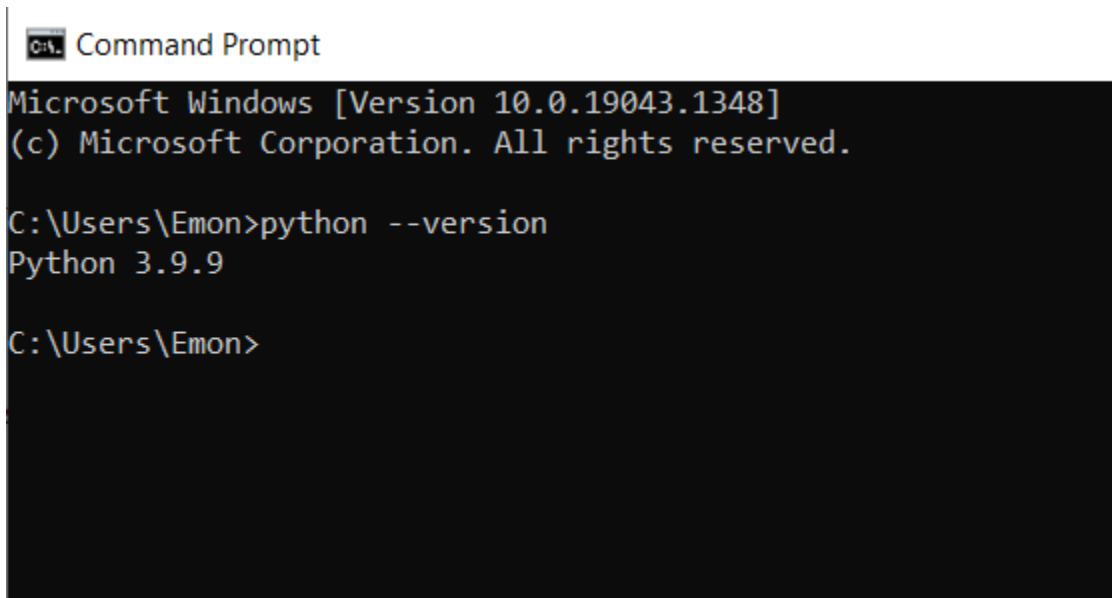
cosine_similarity: L2-normalized dot product calculation vector. If a and b are row vectors, their cosine similarity k is defined as:

$$k(a,b)=\frac{ab}{||a|||b||}$$

This is mentioned as cosine similarity, onto the unit sphere Euclidean normalization projected, dot products is their cosine of angle between points pointed by the vector. This kernel represent tf_idf vector is a popular similarity computing method. [5]

PySimpleGUI: 2018 python lunched GUI. With 330+ demo programs & cook book for start super simple 2021 pysimpleGui is an awesome project. It doesn't become a barrier to limit you to simple problem as it is simple to understand.

This figure 3.2 shows the version of python we used in the development of our project.



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Emon>python --version
Python 3.9.9

C:\Users\Emon>
```

Figure 3.2: Python Version Specification

We have used this PyCharm IDE to build our full project and finally make it a windows software. PyCharm is awesome platform for doing simple and long-term project.

Sometime we use google Colab for testing our mini part of work.

PyCharm:

System requirements for PyCharm:

1024×768 minimum screen resolution

Python 2.7, or Python 3.5 or newer

4 GB RAM minimum, 8 GB RAM recommended

64-bit versions of Microsoft Windows 10, 8, 7 (SP1)

1.5 GB hard disk space + at least 1 GB for caches

Primary release: 30 October 2014

Developers: JetBrains

Stable version: 2021.2.2

Using version in this project: 2021.2.1

Written in language: Java Python

Operating system: Mac, Linux, Windows

File size: 315-450

Type: IDE

License name: Apache License 2.0

Website: www.jetbrains.com/pycharm-edu/

Google Colab: Colaboratory, or "Colab" for short, write and execute Python in browser, with Zero configuration required Free access to GPUs Easy sharing.

As a developer colab made my work easier and help me to overcome from critical barrier of covid-19 lockdown situation.

3.3 Use Case Modeling and Description

Use case diagram includes an overall view of the system being used in the field section. And it identifies some of the internal and external factors. As it is a development project use case diagram helps understanding the work process and indicates the level of development. This figure 4.1 represents the use case diagram of our 'FamBot' project.

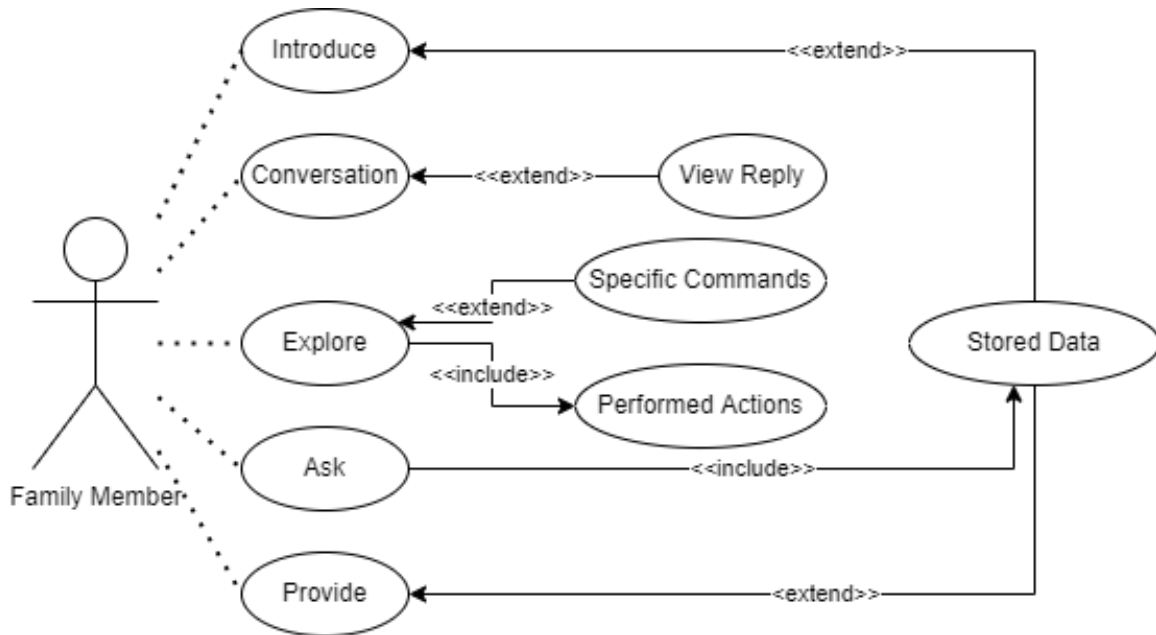


Figure 3.3: Use Case Diagram of FamBot

3.4 Logical Data Model

To report for business sector, logical data model serves the database elements. ER model centralizes entities and relationships shared by elements. Figure 3.4 is representing the ER diagram of FamBot.

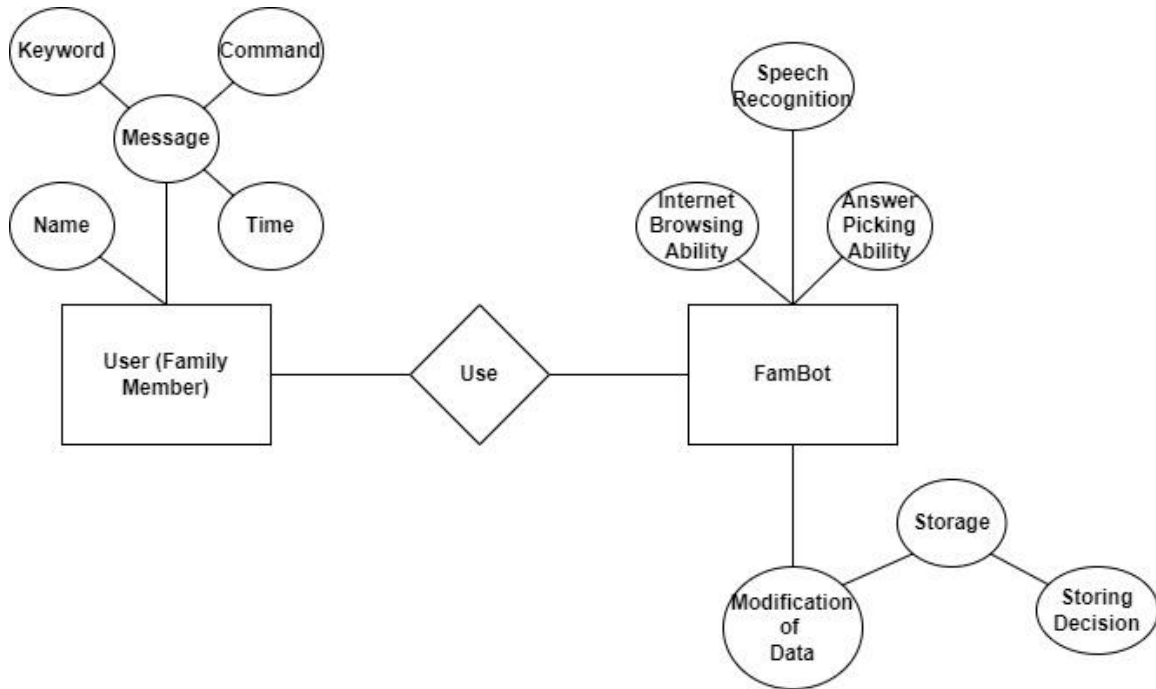


Figure 3.4: ER Diagram of FamBot

3.5 Class Diagram Model

Class diagram is used to model the objects of a system and displays the relationships between them. Figure 3.5 describes relationship between the objects.

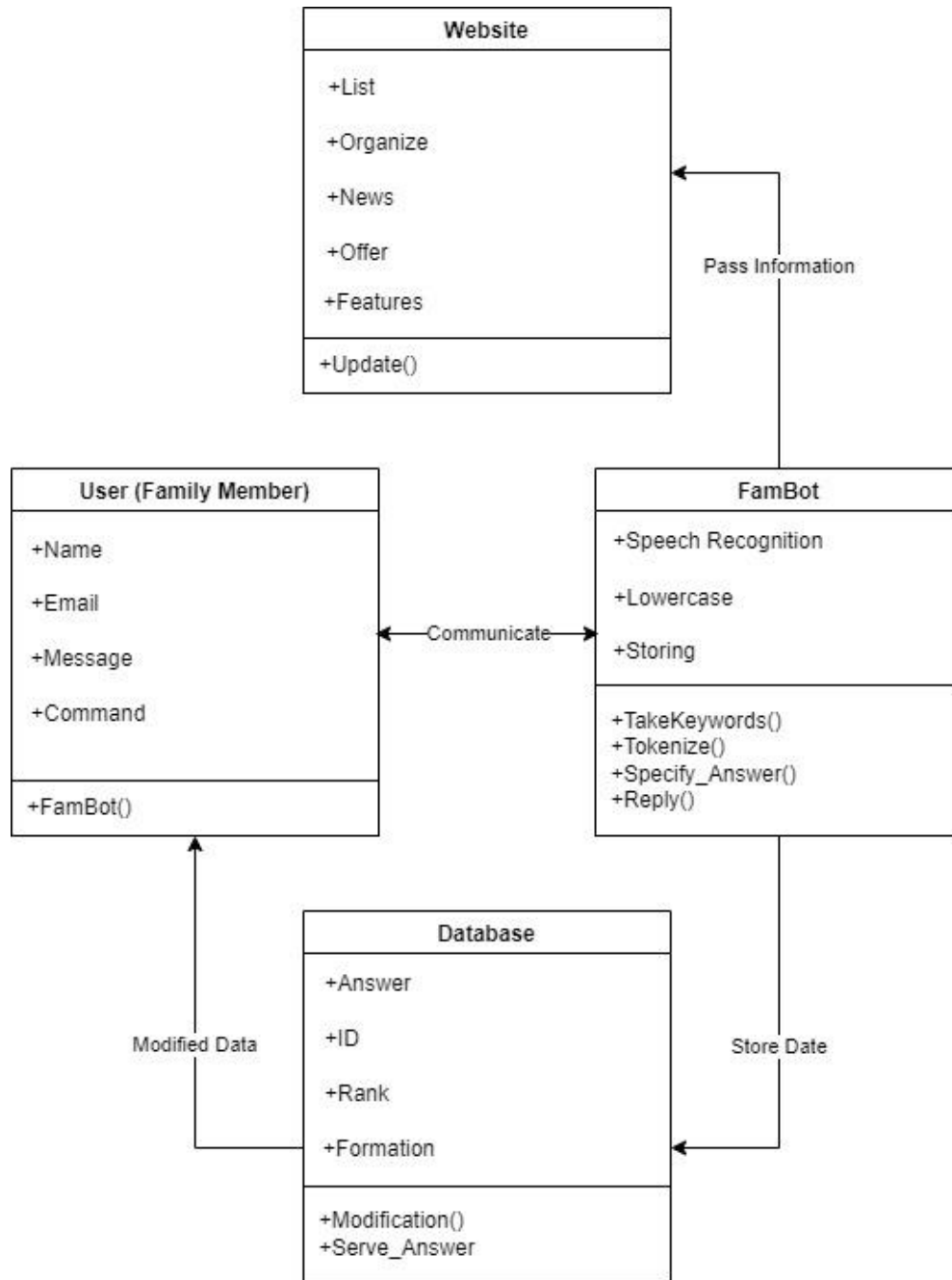


Figure 3.5: Class Diagram of FamBot

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-End Design

Front-end-design is very necessary as users do interact with the software through this. We've tried to keep it user-friendly and understandable easily.

For the user interface, we have used python 3.9. Our software has one visual page to interact with its users. And so a single page had to be designed. Let's preview the page with its design and code. Here figure 4.1 shows us the front end view.

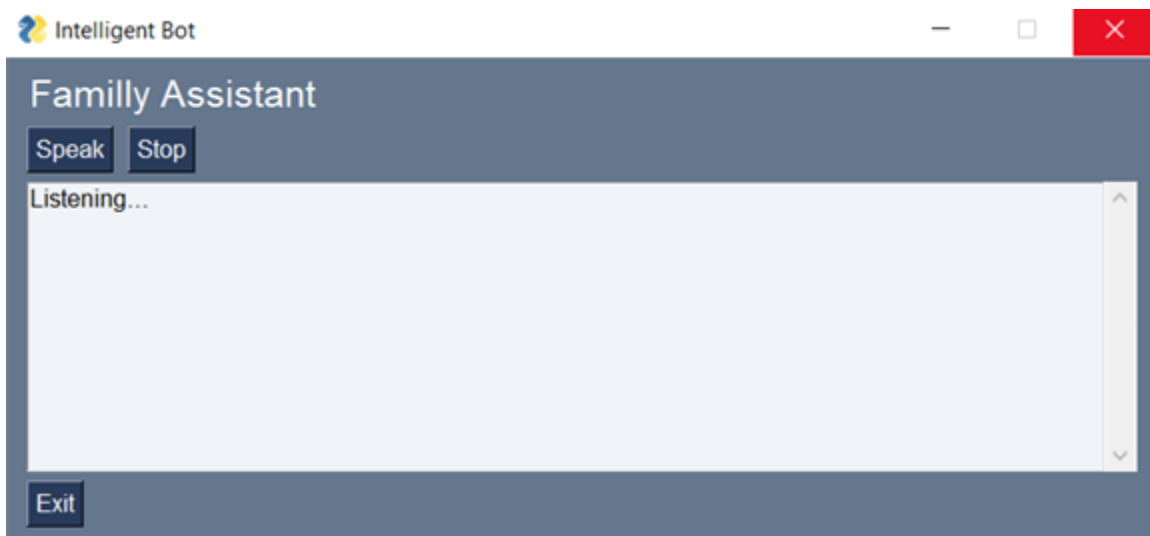


Figure 4.1: Front Page of FamBot

This front end uses the user voice as command and also as data. It determines according to its button click. When the user click the speak button then it start to take voice command and then reply accordingly. The front end is very simple and its's coding is also simple.

Front end Design

```
layout = [[sg.Text('Family Assistant', font='Helvetica
15')],
          [sg.ReadButton('Speak'), sg.ReadButton('Stop')],
          [sg.Output(size=(80, 10))],
          [sg.Exit()]]
window = sg.Window('Intelligent Bot').Layout(layout)
```

```
while True:
    listener = sr.Recognizer()
    event, values = window.Read()
    if event is None or event == 'Exit':
        break
    elif event == 'Speak':
        talk('hi, What\'s your name?')
        user = voice_input()
        user = user.lower()
```

4.2 Back-End Design

All logical parts of our project has been performed in the back-end part. Internet, data connection, .xlsx files, txt file maintaining and all the related codes are performed in the back-end part. In our project python 3.9 has been used for the back-end coding.

To maintain the back-end part, we have chosen .xlsx for storing data for this project. For primary command analysis there are some simple while, for loop, if else condition to determine the task automatically to do what to do in next step working procedure to continue the process. Figure 4.2.1 shows a glimpse of FamBot tackling common questions.

[6]

```
def run_intelligent():
    time = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')[:-3]
    command = voice_input()
    command = command.lower()
    print(command)

    for member in entry:
        if 'play' in command:
        elif 'time' in command:
        elif 'who' in command:
        elif 'date' in command:
            talk('sorry, I have a headache')
        elif 'are you single' in command:
            talk('I am in a relationship with wifi')
        elif 'joke' in command:
            talk(pyjokes.get_joke())
        elif 'remember' in command:

    elif 'bye' in command:
        exit()

    else:
        while True > elif event == 'Speak' > if 'yes' in comment > while True
```

Figure 4.2.1: Basic Functionality Used for Common Conversation

The function `run_intelligent()` is called over and over from the main function while the user doesn't leave. Then it calls a small function named `voice_input()`. Figure 4.2.2 is representing the blocks used for voice input.

A screenshot of a code editor showing the definition of a Python function named `voice_input()`. The function starts with a `try:` block followed by an ellipsis `...`, and an `except:` block followed by `return 'ok'`. The code is displayed in a light blue font on a white background with a yellow highlight behind the `try:` line. On the left side of the code editor, there are icons for a minus sign, a plus sign, a lightbulb, and another minus sign.

```
def voice_input():  
    try: ...  
    except:  
        return 'ok'
```

Figure 4.2.2: Voice Input Using Try and Except Block

The function `voice_input()` takes the voice command from the microphone and processes it to make a sentence. Then the process starts working on the command. Here we have added some interesting features like other chat bots. They can play YouTube videos, inform time, ask to say a joke, etc. But the unique task of this project happened here if the command does not include the above conditions and if it is about himself or about any other member of his family, then the system takes the data and starts another process.

If it is about a query about any of his family members, then it checks about the family member information from the loop process. It saves the data about the family member which was given by other members of the family and stores it in an Excel sheet. It collects the data and saves it temporarily in a `temporary.txt` file. When the `temporary.txt` file is created completely, it sends the command and the data of the `txt` file to another function named `final_response()`. Following figure 4.2.3 summarizes the tokenization of data.

```

def finalresponse(command):
    user_response = command
    user_response = user_response.lower()

    warnings.filterwarnings('ignore')

    # download package from nltk
    nltk.download('punkt', quiet=True)
    nltk.download('wordnet', quiet=True)

    userdata = open('temporary.txt', 'r')
    converteduserdata = userdata.read()

    text = converteduserdata
    global sent_tokens
    sent_tokens = nltk.sent_tokenize(text)
    ...
    global remove_punct_dict
    remove_punct_dict = dict((ord(punct), None) for punct in string.punctua

    # user_response=user_response.lower()
    if (user_response != 'bye'):
    else:
finalresponse()

```

Figure 4.2.3: Tokenizing Data

This function convert command and text file to process as they can do the perfect job. Here all data turns into lower case then they tokenize according to algorithm and depending on the command line it goes through the appropriate algorithm.

If the command line carry simple welcome message then the system reply simply welcome message. But if the command line ask for other information about the family member then it start query on the tokenize files. Here it use artificial neural networking to find out the best answer. And the algorithm used here is the tf-idf vector. It find the best match for reply according to query command. Figure 4.2.4 represents a complex algorithm of performed action on the tokenized data.


```

def response(user_response):
    ...
    robo_response = ''
    sent_tokens.append(user_response)
    # print(sent_tokens)
    tfidfvec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = tfidfvec.fit_transform(sent_tokens)
    ...
    val = cosine_similarity(tfidf[-1], tfidf)
    # print(val)
    idx = val.argsort()[0][-2]
    flat = val.flatten()
    flat.sort()
    score = flat[-2]
    # print(score)
    if score == 0:
        robo_response = robo_response + "sorry,i dont understand"
    else:
        robo_response = robo_response + sent_tokens[idx]

    sent_tokens.remove(user_response)
    return robo_response

```

Figure 4.2.4: Algorithm Used on Tokenized Data

And it returns if the response is found.

There are many more function and task I need to add for do this process perfectly. As we have describe about the modelling before here we make short the explanation.

4.3 Interaction Design and UX

User Page Design: As our aim is to interact user with the voice. So we need not to make many design for its representation. So as simple as we can we made the design and its satisfied user completely. Because when one start using it he will be busy in talking and hearing. He need not to input anything by his hand.

UX

For UX (User Experience) we have tried making a user friendly system. Different computers have been used to justify its behavior. With positive reviews from them we are actually satisfied in a basic level with its performance till now. But without launching FamBot publically, it is hard to evaluate user experience clearly.

4.3 Implementation Requirements

In this rapid changing world, the method of working is changing day by day. So one can implement the idea in different way that we done already. We have used an open source language and it has many similar method to work on. As it is open source it has a vast library for working. Working area is also big. But there are some main base which cannot be change they are the flexibility in it input output method and the policy of choosing decision.

- Different libraries of python make our work simple and help us to think about the next in a sort moment.
- We need not any hosting and think about any domain because it can do all thing into its own area. As it is made with a view to make shorter the distance between family members so it doesn't need to go through the whole world in details.
- Colab helped a lot to test single snap of different part to make it finally possible to done the project.
- PyCharm is a good emulator like everything in one place for pc software develop.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Data

Data storing analysis is an important part of this project. Due to local operation we need simple storing environment. To do the task we used xlsx file and txt file. We store data about user with their identity and along with time. We make simple change in the main data before storing. The change is about cutting the extended or unimportant part of the information. We manage it to record it on the xlsx file automatically with clear identity and time. Then we give user output again from this data. This is no as like as the other chatbot. Here the working procedure is quite different. As many times the system will be used it will gather more knowledge and will able to response more accurately. Figure 5.1 shows an example of the way data being stored in excel.

1	shakil	i am going to ride cycle	2021-11-18 18:24:37.554
2	emon	i am trying to complete project	2021-11-18 18:25:34.999
3	shakil	say emon he is good	2021-11-18 18:34:19.746
4	shakil	i am going outside	2021-11-18 18:36:16.218
5	shakil	i am going to pump my cycle	2021-11-18 18:36:38.020
6	shakil	i am watching earth2	2021-11-18 21:12:21.438
7	emon	i have learned a lot about python	2021-11-18 21:31:54.042
8	shakil	emon went outside of the room	2021-11-18 21:59:10.202
9	shakil	i feel suffocated	2021-11-18 22:02:40.983
10	nadim	1234 get on the dance floor	2021-11-18 22:54:31.543
11	imo	I am going to home in next January	2021-11-30 22:08:02.609

Figure 5.1: Visual of Stored Data Along with Date and Time

When there is a request for information about any member included in the sheet then it returns according to the target user and the information is delivered to the txt file. Then the algorithm of AI find out the best answer among them, Which is interactive. For example we add a screenshot off the sheet in the fig: which was used during testing period. [7][8]

5.2 Implementation of Front-End Design

We have added the front end design of our software project in the main code section. Front end design is very important part of any project or work. It make a spawn between user and back end code. We design our font end with pysimplegui library. We have added all the possible update time to time in the front end output. It make the work more interactive for user. He can realize easily how the software is reacting with his behavior. If there need any repeat of command the user can easily understand the failure of process. Afterall it is the most interactive part of our project.

5.2.1 User Module

In the user module it is very simple part for use. One can easily introduce himself only by his name and the guardian name of his family in this software by providing valid mail address and phone number. After that every step of his work is as usual. He can add other members easily.

In the other hand the host of this software can make other member of his family as host and can add other member in the software by providing valid mail address and phone number.

5.3 Testing Implementation

After successfully running the application, we tested it in many different ways. Here are some of the testing we have went through:

Table 5.1: Testing Information and Date:

Test Case	Test Input	Expected Outcome	Obtained Outcome	Result	Tested on
1. Can listen	Can you hear me	can you hear me	can you hear me	pass	10/10/2021
2. Can speak	How can I help you?	How can I help you?	How can I help you?	pass	10/10/2021
3. Save data	I am going to ride cycle	I am going to ride cycle	I am going to ride cycle	pass	11/10/2021
4. User identify	Hi, I am shakil	Hi I am shakil	Hi I am shakeel	fail	11/10/2021
5. User identity	Hi I am john	Hi I am john	Hi I am john	pass	11/10/2021

5.4 Testing Result and Reports

In order to find bugs we needed to test the application and found many bugs and have fixed many of them. We need more testing on the software and reduce more complexities.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

The software “FamBot” is especially designed as a virtual family member. The unique feature in it is this member of the family never die. After generation it will carry the history of their previous family member like grandfather grandmother grand grandfather and grand grandmother. But it has not started its journey yet. If it get chance to start its journey it will help to continue the family with their unique identity and introduce the newborn child about their history of previous generation.

6.2 Scope for Further Developments

Any development project has no end to develop. So, we have also to develop this project in future. Here some of the plans are given below:

- We want to publish it among the world with unique device like other Gadgets.
- We want to add a feature that would be able to combine multiple families in one place.
- We want to enable it to read the user reaction.
- Make it available in all language.
- Make it compatible for all kind of device.

REFERENCES

[1] Bocco,

<https://www.kickstarter.com/projects/1477302345/bocco-the-family-robot-that-brings-your-loved-ones>; last accessed on 2nd January, 2022

[2] Buddy, <https://buddytherobot.com/en/buddy-the-emotional-robot/>; last accessed on 2nd January, 2022

[3] Alexa,

<https://www.gadgetsnow.com/faqs/how-does-alexa-work/articleshow/78238756.cms>; last accessed on 2nd January, 2022

[4] Python packages, <https://www.udacity.com/blog/2021/01/what-is-a-python-package.html>; last accessed on 15th October, 2021

[5] TF-IDF Vectorizer,

https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[6] Base Project, <https://github.com/ProgrammingHero1/romantic-alexa>; last accessed on 2nd October, 2021

[7] Handling .xlsx files, <https://www.marsja.se/your-guide-to-reading-excel-xlsx-files-in-python/>; last accessed on 22nd December, 2021

[8] Data storage, <https://www.geeksforgeeks.org/reading-excel-file-using-python/>; last accessed on 19th October, 2021

PLAGIARISM

Plagiarism Report

Fambot

ORIGINALITY REPORT

18%

SIMILARITY INDEX

17%

INTERNET SOURCES

2%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

10%

2

en.wikipedia.org

Internet Source

1%

3

Submitted to Daffodil International University

Student Paper

1%

4

www.msvu.ca

Internet Source

1%

5

codeloop.org

Internet Source

1%

6

Vinayak Mathur, Yannis Stavrakas, Sanjay Singh. "Intelligence analysis of Tay Twitter bot", 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016

Publication

1%

7

sourceforge.net

Internet Source

1%

8

anaconda.org

Internet Source