Graduation Project (1) Documentation

Department of Software Engineering, Computer Science &

Data Science & Artificial Intelligence

Faculty of Information Technology

Al-Ahliyya Amman University

The Uni-Bot

A project submitted in partial fulfillment of the requirements for the degree of Bachelor in Software Engineering, Computer Science & Data Science & Artificial Intelligence

By:

Tala Nidal Khudair (201820311)
Dana Mohammad Naser (201820346)
Raya Kamel Al-Taji (201720200)
Ismail Hussein Zubi (201910996)

Supervised by:

Dr. Mohammad Al Sharaiah

Abstract

An intelligent virtual assistant (IVA) may be a software agent which will perform tasks or services for private supported commands or questions. This robot acts like a chatbot, it has a specific answer to each question but replies verbally. Some virtual assistants are ready to interpret human speech and respond via synthesized voices. Users can ask the assistant questions, get their answers, and, also know the time with verbal commands.

Table of Contents

Chapter 1: Introduction **5**
1.1 Overview 5
1.2 Project Motivation 5
1.3 Review of Related literature and Systems 6
1.4 Objective of the project 6
1.5 Project Plan 7
Chapter 2: The Development Process **9**
Chapter 3: Requirements Engineering and Analysis **11**
3.1 Stakeholders 11
3.2 Functional Requirements 11
3.2.1 Table.2. shows the requirements needed 11
3.3 Non - Functional Requirements 13
3.3.1 Maintainability 13
3.3.2 Platform 13
3.3.3 Serviceability 13
3.3.4 Performance 13
3.3.3 Process 13
3.3.4 Usability 13
3.4 Use Case Diagram 14
3.5 Use Case Diagram Description 15
Chapter 4: Prototype Design **20**
4.1 Conceptual Class Diagram 20
4.2 Activity Diagram 21
4.3 User Interface Design 22
4.3.1 User Interface. 22
4.3.2 Robot Parts 23
Chapter 5: Partial System Implementation **27**
5.1 Overview 27
5.2 Tools and Technologies Used 27
5.2.1 Raspberry Kit 27
5.2.2 PyCharm IDE 27
5.2.3 Python 27
5.2.4 Gantt Chart 27
5.2.4 Dia Software 28
Chapter 6: Project Status **30**
References **31**

Chapter 1
Introduction

Chapter 1: Introduction

1.1 Overview

Gone are the days when humans depended on other humans for help or services. The digitalization of the world made sure that humans no need to contact anyone else to seek help, they could depend on a far more efficient and reliable device that can take care of their everyday needs. Computers, mobiles, laptops, and so on., became a part of us and our daily lifestyles, they may carry out simple calculations to complicated applications to lessen monotonous work and waste of manpower. virtual personal Assistant has nearly become a primary necessity in all electronic devices to execute the required problems easily. greater than just being a bot, VPA could make life easier for the person in numerous approaches. Speech recognition is one of the rather new integrations into the VPA. however, though it's moderately green, it isn't always very beneficial and isn't used by the consumer because of its high number of blunders. though the mistake percent of the upcoming VPAs is around 5 percent, it nevertheless isn't always pretty up to speed to where it turns into a primary part of the user's life. as a consequence, the task intends to build a VPA with a speech recognition that has a minimum error percentage. Voice recognition is a complicated procedure for the usage of superior ideas like neural networks and gadget studying. The auditory enter is processed and a neural network with vectors for each letter and syllable is created. that is known as the data set. while a person speaks the device compares it to this vector and the special syllables are pulled out with which it has the very best correspondence.

1.2 Project Motivation

The idea of substituting human beings with a chatbot like, that can help in directing, navigating, and producing information for the confused or lost users, inside the university and in each faculty seemed exciting to us, as it may reduce labor and time, meanwhile experiencing technologies in real life.

1.3 Review of Related literature and Systems

Technology has now become a part of our lives and without it, we're unable to be productive or create new ideas that might help us ease up daily activities. Our project's idea is extracted from the idea of small systems like the computerized appointments teller machine, that outputs a tiny paper that has the next turn inline or for any appointments that require waiting. We're also inspired by the mini-robots at EXPO, Dubai exhibition, which help and direct the tourists from all around the world inside the exhibition. There are Chatbots used to navigate through a website by using automated texts received from users. Our project is an outcome of these inspirations but what makes it special is adding the touch of speech recognition whilst with text which is already known.

1.4 Objective of the project

The main objective of this project is to provide services to the confused students or staff from other faculties, who enter the IT faculty, looking for someone or something specific inside the faculty such as:

- The Dean's Office and have an idea of who the dean of IT faculty is.
- Locations like the library, prayer rooms, and washrooms.
- Exams and terms schedules.
- Different lecture halls.
- Office/office hours of each professor in the faculty.
- Lecture timings
- The current time of that day (acts as a clock too).

1.5 Project Plan

Our project is divided into two main phases, phase one involves analysis and design, whereas phase two involves implementation and testing. However, the following Table shows phase one.

Tasks	Description	Start time	End time	Duration	Dependency
T1	Planning	17/3/22	24/3/22	1 week	
T2	Research	25/3/22	1/4/22	1 week	T1
T3	Analysis	9/4/22	16/4/22	1 week	T2
T4	Design	17/4/22	1/5/22	2 weeks	T3
T5	Implementation	7/5/22	21/5/22	2 weeks	T1,T2,T3,T4

Table 1. Project Plan

The above table is represented in a Gantt chart as shown in Figure.1.

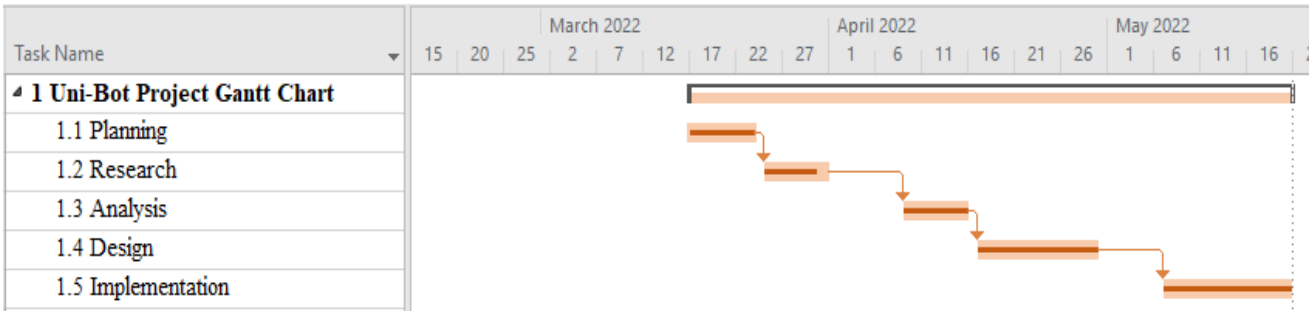

Figure 1. Gantt Chart

Chapter 2
The Development Process

Chapter 2: The Development Process

As for this project we are working on using the Agile Development process, as it's easier when it comes to planning, designing, developing, testing, deployment, reviewing, and then launching as shown in Figure.2. As studied, this software development process is simply helpful to engage with all the stakeholders including the project team and, can make changes anytime to the project even after moving to the next phase. We almost took a week to plan the major idea of how the Uni-Bot should work and what services it may produce. After two weeks of planning, we decided to add more services to the services we already listed. Then came the Research part, where we had to do a few researches on how to make a virtual assistant using Python libraries.

Figure 2. Agile Methodology

Next comes the Analysis phase, where we collected all the information about the faculty, the professors, lectures, and different locations in the faculty like washrooms, prayer rooms, and the library, afterwards we collected the requirements needed to build the Uni-Bot, from services to hardware requirements. Then comes the design phase, here we sketched what the Uni-Bot should look like, including how we can implement it, using classes and dictionaries. Lastly, comes the implementation phase, where we started coding, building the bot piece by piece, and coding the Raspberry kit. In the later documentation (2), we shall mention the rest of the development processes.

Chapter 3

Requirements Engineering and Analysis

Chapter 3: Requirements Engineering and Analysis

3.1 Stakeholders

- University staff who are responsible for helping the students inside the faculty.

- The dean of IT faculty and supervisors of this project.

- Future sponsors who may like this project.

-Project team.

3.2 Functional Requirements

3.2.1 Table.2. shows the requirements needed

Requirement ID	Requirement Description
Req-1	The Uni-Bot shall open the asking session only if the User says 'Uni'.
Req-2	The Uni-bot shall look and move towards the User who called it.
Req-3	The Uni-bot shall have the ability to introduce itself and greet the User depending on the time of the day, before listening to the User.
Req-4	The Uni-Bot shall provide the User the ability to speak to it, after displaying 'Listening...', on its screen.
Req-5	The Uni-Bot shall provide the User the ability to read what the User said on the bot's screen.
Req-6	The Uni-Bot shall provide the User the ability to ask more than one question once it's turned on.
Req-7	The Uni-Bot shall provide the User the ability to ask specific questions about the dean and other professors' offices and can answer back, by specifying 'who'.
Req-8	The Uni-Bot shall provide the User the ability to ask about the lecture's time and location, by specifying 'when' & 'where'.
Req-9	The Uni-Bot shall have the ability to listen to the User's questions.
Req-10	The Uni-bot shall look for keywords in the User's questions, and respond from the dictionaries set after searching through them.

Req-11	The Uni-bot shall be able to move around in all directions, especially around people.
Req-12	The Uni-bot shall provide the user the ability to know when each semester ends.
Req-13	The Uni-bot shall provide the user the ability to know where are the library, washrooms, auditoriums, restaurants and prayer rooms.
Req-14	The Uni-bot shall prevent objects like walls and staircases.
Req-15	The Uni-bot shall provide the user the ability to know where are the exam halls.
Req-16	The Uni-bot shall provide the user the ability to know when are the Final exams and Midterm exams.
Req-17	The Uni-bot shall be able to provide the ability to ask about the time.
Req-18	The Uni-Bot shall close the asking session only if the User said "No".
Req-19	The Uni-bot shall have the ability to say "have a nice day" when the user ends the session by saying the "no" command.

Table 2 (Functional Requirements)

3.3 Non - Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system in particular conditions, rather than specific behavior.

3.3.1 Maintainability

The code written for the Uni-bot must be maintainable. Adding documentation will improve the maintainability scale of the system. From an implementation perspective, we will use python code using PyCharm IDE.

3.3.2 Platform

The code must run on Raspberry PI Screen, Windows, and even on Mac.

3.3.3 Serviceability

The Uni-bot shall respond to the user, once called, and offer the services available in it.

3.3.4 Performance

The Uni-bot shall have a response time of not more than two seconds after users speak to it.

3.3.3 Process

- The Uni-bot shall have a female voice, of sound speed not too slow nor too fast.
- The Uni-bot shall have speech recognition installed from the PyAudio library.
- The Uni-bot shall have speech-to-text installed from the Pyttx3 library.

3.3.4 Usability

The Uni-bot interface has to be easy to use and doesn't require more than an hour of training. As it only requires the user to touch or just speak to it, and read the responses on it.

3.4 Use Case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. The Figure below represents the use case for the Uni-Bot.

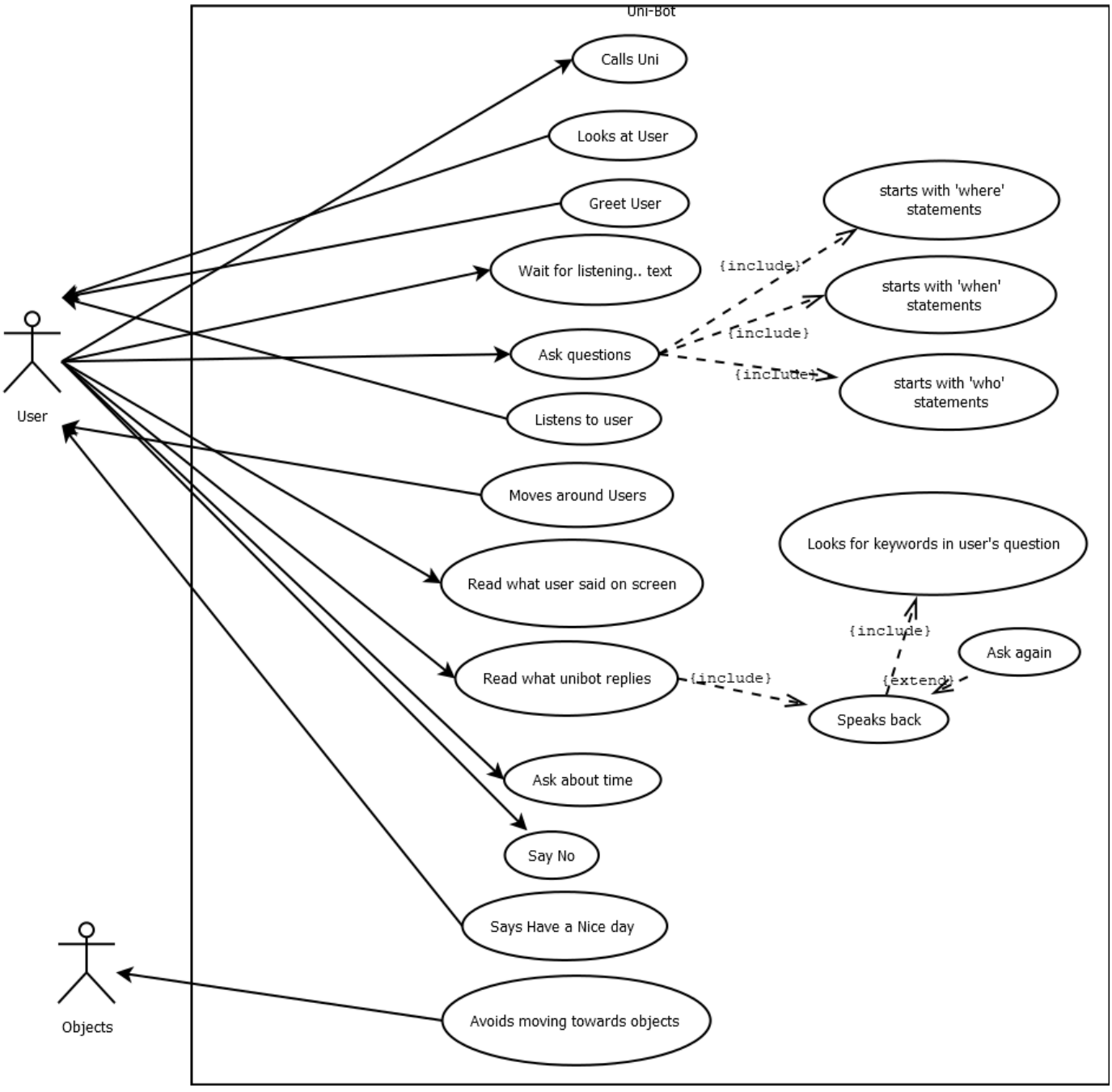

Figure (3) Use a case diagram

3.5 Use Case Diagram Description

The blank use case diagram has 13 main activities to make the User practically interact with the Unit-Bot. The Table below shows all the use cases.

USE CASE NAME:	Call Uni	
USE CASE ID:	1	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User shall call on the Uni-bot to answer him/her.	
PRE-CONDITION	The User must say Uni.	
POST-CONDITION	The Uni-bot turns towards the User	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. calls Uni	1.1. listens to User.

USE CASE NAME:	Looks at User	
USE CASE ID:	2	
PRIMARY ACTOR:	Uni-Bot	
DESCRIPTION:	The Uni-bot looks at the User who called it.	
PRE-CONDITION	Called on Uni-bot	
POST-CONDITION	The Uni-bot starts detecting who called it.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. detects user who called	1.1. walks towards it

USE CASE NAME:	Greet User	
USE CASE ID:	3	
PRIMARY ACTOR:	Uni-bot	
DESCRIPTION:	The Uni-bot shall greet the user depending on the days' time.	
PRE-CONDITION	Must be called on, then looks at the user who called it.	
POST-CONDITION	The Uni-bot shall greet the user and introduces itself before listening to the User.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. Greet User.	1.1 moves towards User and greet. 2.1 provides listening... text to listen to the User
	2. Introduces itself to the User.	

USE CASE NAME:	Wait for listening... text	
USE CASE ID:	4	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User shall wait to see the 'listening...' text before speaking.	
PRE-CONDITION	The User must call on the Uni, get greeted, and then speak after seeing the 'Listening...' Text.	
POST-CONDITION	The User then speaks.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. speaks after listening.	1.1. starts listening to the User.

USE CASE NAME:	Ask Questions	
USE CASE ID:	5	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User shall ask about the services provided by the Uni-bot like, about the lecture's locations & timings, or locations like the washroom, prayer room, library or about the dean and other professor's offices, or even about a specific exam's time and place, or university calendar.	
PRE-CONDITION	The User must start with the words 'when' or 'where' or 'who'.	
POST-CONDITION	The Uni-bot shall be able to respond to the User.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. ask, who is the dean of IT.	1.1 respond, dean's name, and office no.
	2. ask, when is the logic lecture.	2.1 respond, when is the logic lecture.
	3. ask, where is the washroom	3.1 respond, where the washroom is located

USE CASE NAME:	Listens to user	
USE CASE ID:	6	
PRIMARY ACTOR:	Uni-bot	
DESCRIPTION:	The Uni-bot shall have the ability to listen to the user.	
PRE-CONDITION	The user must ask the Uni-bot.	
POST-CONDITION	The Uni-Bot must listen and recognize what the user says.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. User asks a question.	1.1 Bot Listens. 1.2 If not heard properly, the Bot shall tell the user 'Sorry I could not hear that'.

USE CASE NAME:	Moves around user	
USE CASE ID:	7	
PRIMARY ACTOR:	Uni-bot	
DESCRIPTION:	The Uni-bot shall move around people.	
PRE-CONDITION	None	
POST-CONDITION	None	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	None	1.Moves around until someone calls for it.

USE CASE NAME:	Read what the User Said	
USE CASE ID:	8	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User shall read what he/she asked the Uni-bot as a means of reassurance, that the Bot heard what the user wanted to ask.	
PRE-CONDITION	Must speak after the 'listening...' text	
POST-CONDITION	The Uni-bot then transforms from speech to text.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. reads their speech to the bot.	1.1 transforms from speech to text on the screen of the bot. 1.2 looks for keywords in user's question. 1.3 speaks back according to the User's question statement. 1.4 asks the user if there anymore questions.

USE CASE NAME:	Read what Uni-bot replies	
USE CASE ID:	9	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User can also read what the bot replied, after reading what the bot heard from the user, thus be assured that he/she got answered back to the questions they needed.	
PRE-CONDITION	Must be already asked the bot a question.	
POST-CONDITION	The Uni-Bot then replies accordingly.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. Reads bot's answers. 2. Reads bot's question.	1.1 Transforms from text to speech & speech to text. 1.2 shows text on the screen. 2.1 asks User if they have any more questions, after answering the User.

USE CASE NAME:	Ask about Time	
USE CASE ID:	10	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User can also ask about the time provided by the bot.	
PRE-CONDITION	After seeing the 'Listening...' Text	
POST-CONDITION	The Uni-bot shall tell the User the time	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. ask, what's the time	1.1 tells time.

USE CASE NAME:	Say No	
USE CASE ID:	11	
PRIMARY ACTOR:	User	
DESCRIPTION:	The User shall answer no after the Uni-bot asks them 'Do you have any more questions?', to end the interaction session between the User and the bot.	
PRE-CONDITION	After listening to 'Do you have any more questions?'.	
POST-CONDITION	The User must say any reply with the word 'No' in it.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. User says no.	1.1 Bot ends this user's session and waits until another User calls on at it to begin a new session.

USE CASE NAME:	Says have a nice day	
USE CASE ID:	12	
PRIMARY ACTOR:	Uni-bot	
DESCRIPTION:	The Uni-bot shall tell the user 'Have a nice day' statement, after the user says 'No'.	
PRE-CONDITION	After listening to 'No'.	
POST-CONDITION	The Uni-Bot must say 'Have a nice day'.	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	1. User says no.	1.1 Bot replies with 'Have a nice day' statement.

USE CASE NAME:	Avoids moving towards objects	
USE CASE ID:	13	
PRIMARY ACTOR:	Uni-bot	
DESCRIPTION:	The Uni-bot shall avoid moving towards objects and staircases.	
PRE-CONDITION	None	
POST-CONDITION	None	
MAIN SUCCESS SCENARIO:	**Actor Action**	**System Action**
	None	1.The bot avoids objects like walls and staircases.

Table 3 (Use case diagram discerption)

Chapter 4
Prototype Design

Chapter 4: Prototype Design

4.1 Conceptual Class Diagram

A class diagram in the Unified Modeling Language (UML) As shown in figure.4, is a form of static structural diagram in software engineering that depicts the structure of a system by displaying the system's classes, properties, operations (or methods), and relationships among objects. The following classes are shown below in Figure.4.

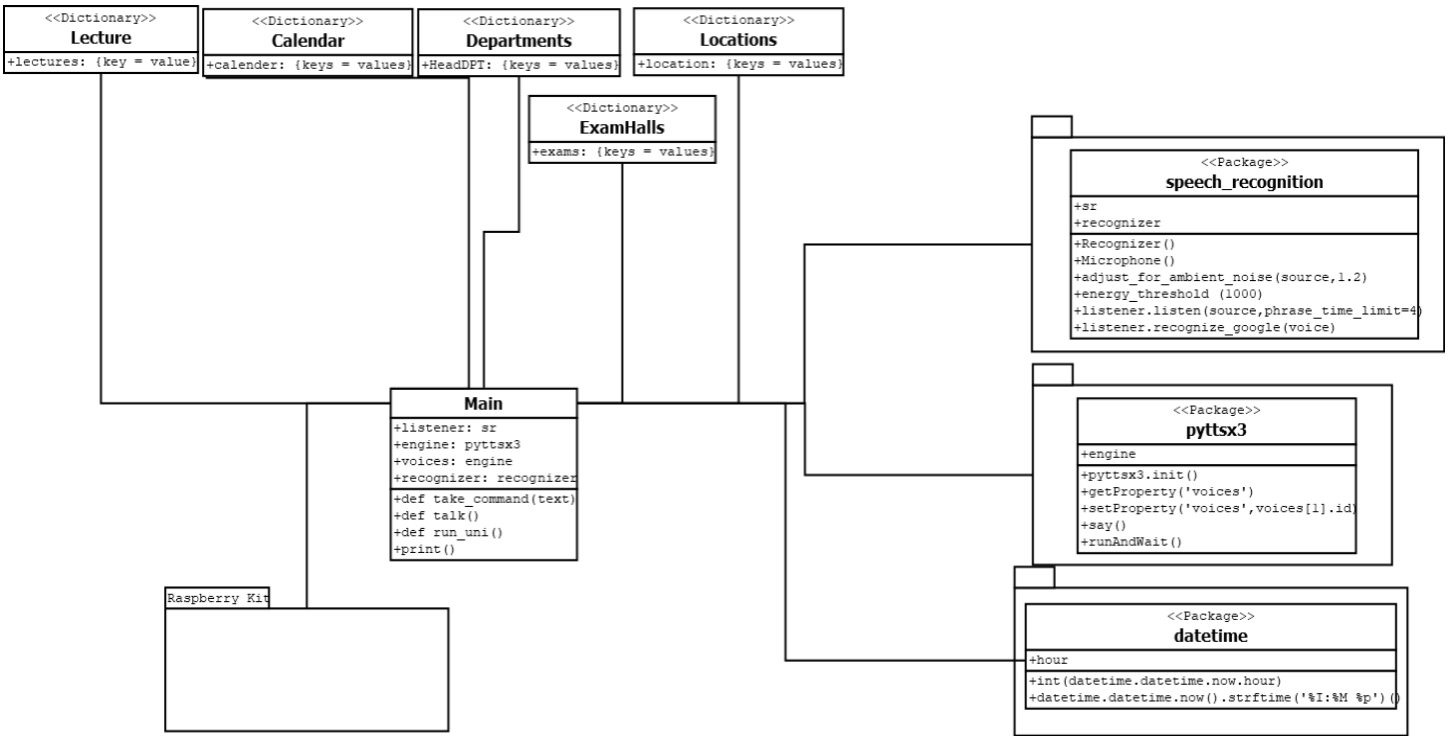

Figure (4) Class diagram

4.2 Activity Diagram

Another significant behavioral diagram in the UML diagram for describing dynamic characteristics of the system is the activity diagram. An activity diagram is a more complex version of a flow chart that depicts the flow of information from one activity to the next. As shown in Figure.5.

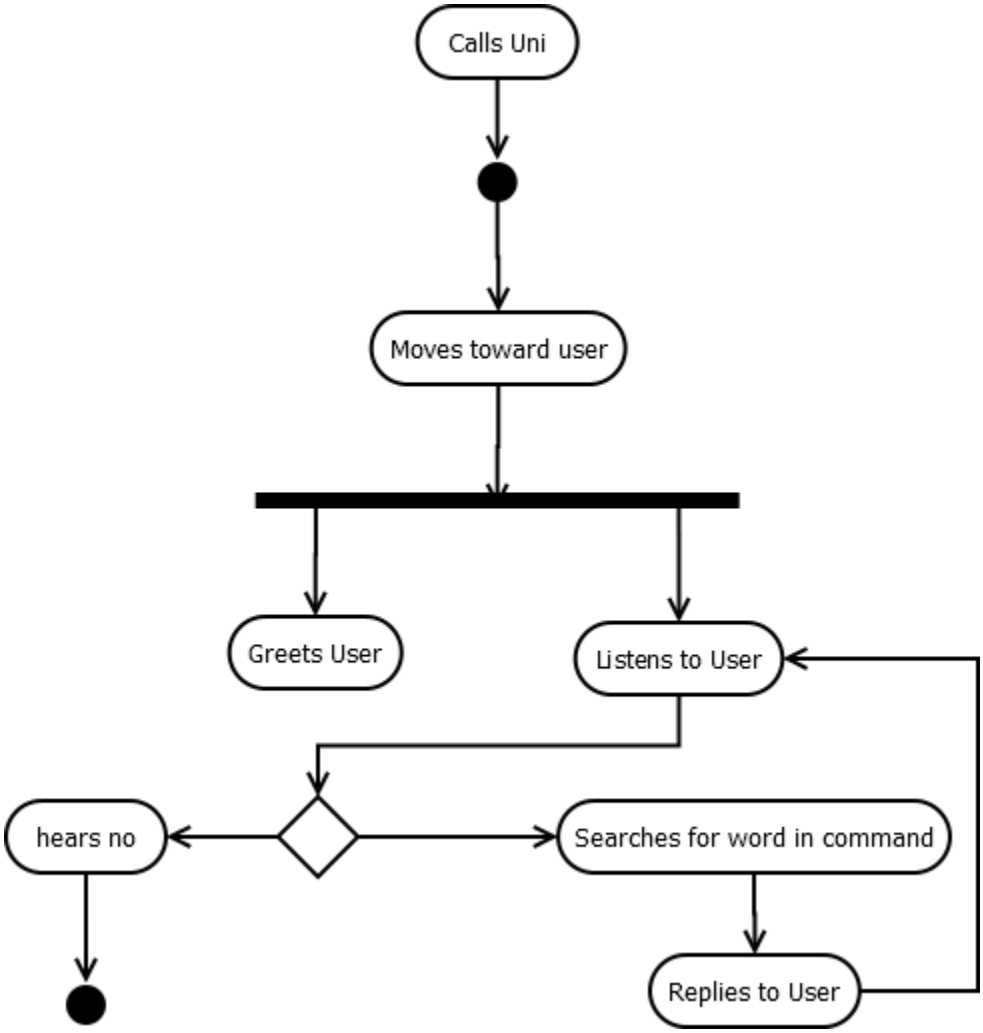

Figure (5) Activity Diagram

4.3 User Interface Design

4.3.1 User Interface.

The following Figure.6, is the interface that will be on the Uni-bot's screen.

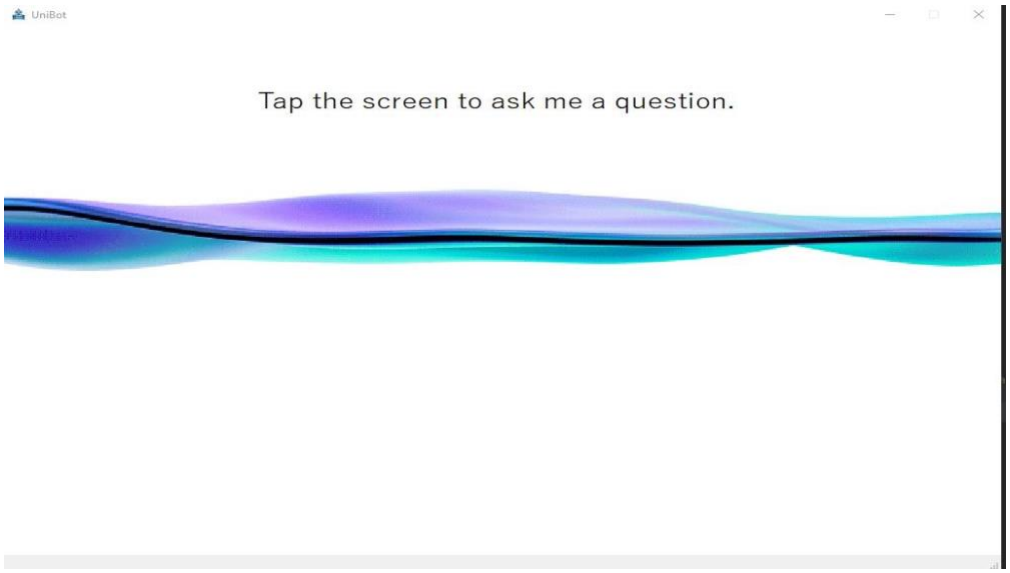

Figure (6) User Interface Design main screen

While the Figure.7. shows the user's questions and the Uni-bot responses.

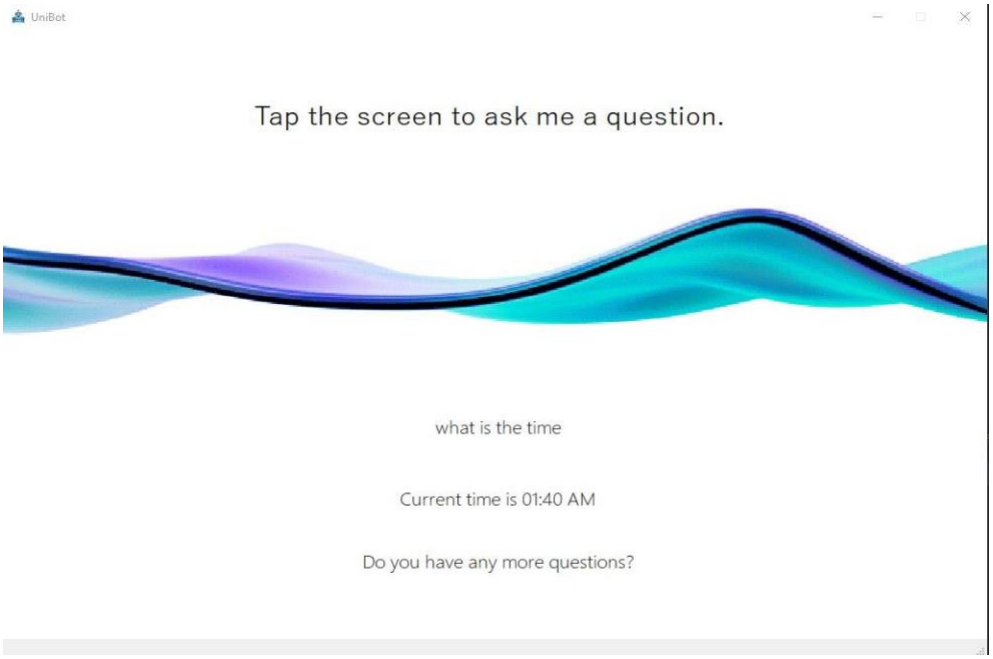

Figure (7) User Interface Design after listening and replying to the user

4.3.2 Robot Parts

Raspberry Pi 7 2GB Kit – This kit will be mainly used for the Uni-Bot Project. The following Figure.8. shows what the kit looks like.

Figure (8) Raspberry Pi 7 2GB Kit

Raspberry Screen – The screen will be used to display the UI made for this project to let the User view what they & the Bot said. The screen, as shown below in Figure.9.

Figure (9) Raspberry Pi 7 screen 7 inches.

Servo motor driver, Dc gear motor, and servo motor- The motors are used to help the Uni-Bot function and move its wheels and the inner circuit. The Figure.10. below shows how the motors look like.

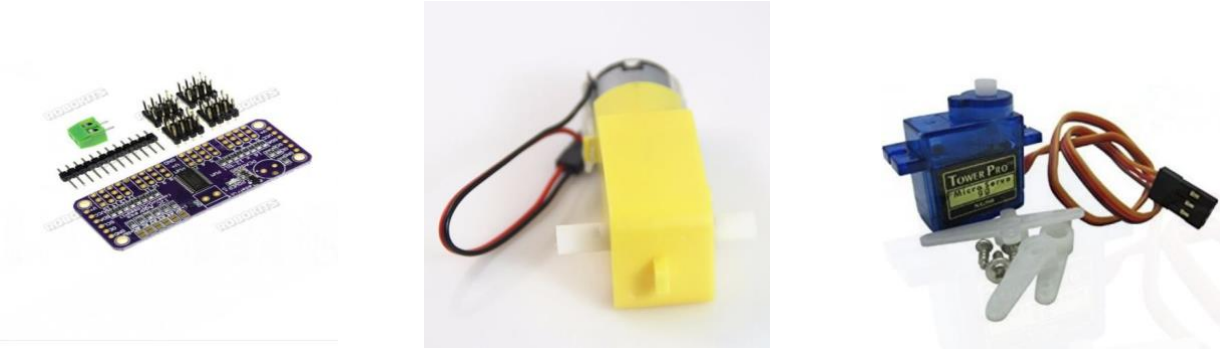

Figure (10) Servo motor diver, Dc gear motor, Servo motor.

The ultrasonic sensor and sound detector sensor- These sensors will help the Uni-bot to detect where the sound came from so it could move towards it and the ultrasonic is used to help prevent the Uni-Bot to hit objects like walls and avoid staircases. As shown in Figure.11. below are the sensors.

Figure (11) Ultrasonic sensor and sound detector sensor.

The Microphone and speaker – These are also used within the Uni-bot in order to help it listen to the user and talk through the speaker. The Figure.12. below shows what they look like.

Figure (12) Microphone and Speaker.

The Solar panel and the wheel- The solar panel will be used to recharge the Uni-bot so it doesn't run out of charge, while the wheel will help the Uni-bot to move around, 4 wheels will be used. The Figure.13. below shall represent them.

Figure (13) The solar panel and the Wheel.

Chapter 5
Partial System Implementation

Chapter 5: Partial System Implementation

5.1 Overview

This robot will be functioned using Raspberry PI, coded in PyCharm IDE. The parts mentioned above are also used to structure the Uni-bot.

5.2 Tools and Technologies Used

5.2.1 Raspberry Kit

The Raspberry Pi is a low-cost computer the size of a bank card that connects to a computer monitor or television and utilizes a conventional keyboard and mouse. It's a capable small device that allows individuals of all ages to learn about computing and programming in languages such as Scratch and Python. Several interfaces (HDMI, multiple USB, Ethernet, onboard Wi-Fi and Bluetooth, many GPIOs, USB powered, etc.), Supports Linux and Python (making it simple to build apps). Developing an embedded board of this caliber will take a significant amount of time and money.[3]

5.2.2 PyCharm IDE

PyCharm is a Python Integrated Development Environment (IDE) that includes a variety of key tools for Python developers that are tightly integrated to create a pleasant environment for effective Python, web, and data science development.[2]

5.2.3 Python

Python is a dynamically semantic interpreter-based object-oriented high-level programming language. Python's concise, easy-to-learn syntax prioritizes readability, which lowers software maintenance costs. Python facilitates program flexibility and code reuse by supporting modules and packages. The Python interpreter and its substantial standard library are free to download and distribute in source or binary form for all major platforms.[1]

5.2.4 Gantt Chart

This chart is used to pin out when we started and when we are expected to finish our project and also helps in identifying where we reached with our project progress. We used it to represent our work as in Figure.1. above.[4]

5.2.4 Dia Software

Dia features a modular design with a variety of form packages for various requirements, including flowcharts, network diagrams, circuit diagrams, and more. It doesn't prevent symbols and connections from different groups from being combined. We used this tool to draw the class and activity diagrams.[5]

Chapter 6
Project Status

Chapter 6: Project Status

The Project's Status is partially done, as for now, we'll put our focus on building the robot and put it into action, and then test it and complete the rest of the documentation for Project (2). Meanwhile, the coding part is fully done and is being tested, including setting the GUI for the Raspberry tablet. As shown in Figure.14. The document in blue and the implementation in orange.

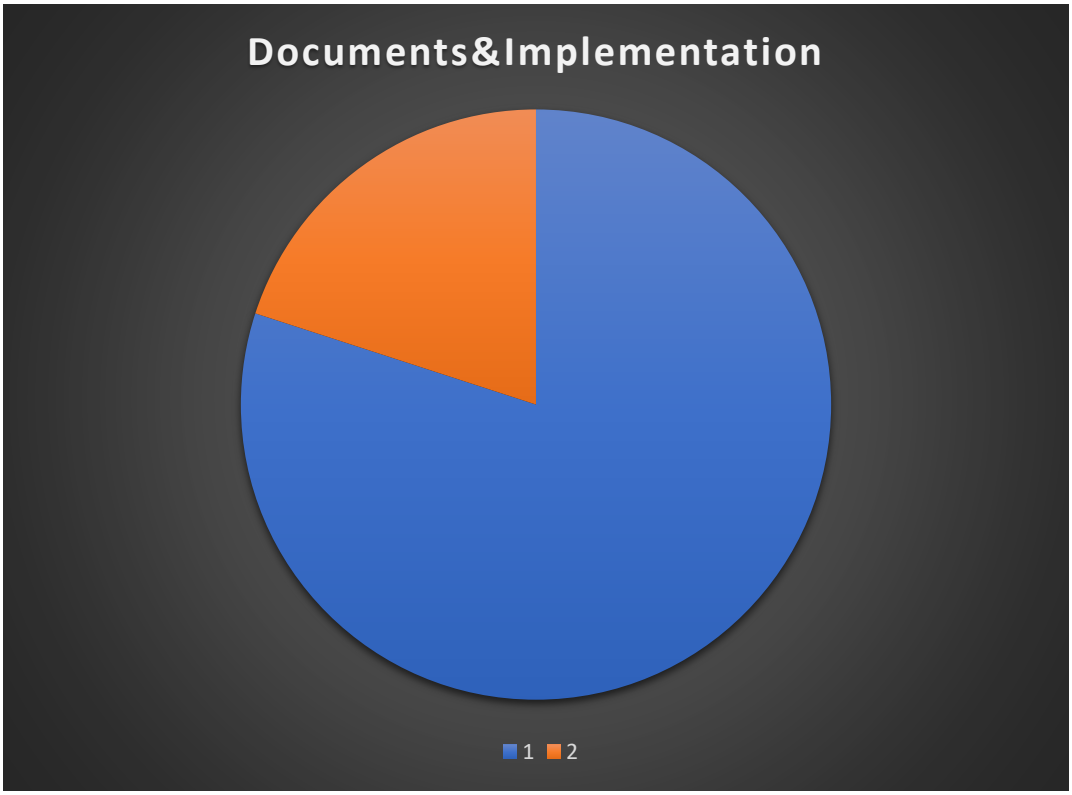

Figure (14) Project Status.

References

[1] https://www.onbenchmark.com/blog-detail/python-and-its-future-potential.12/4/2022

[2].https://en.wikipedia.org/wiki/PyCharm.30/4/2022

[3] https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/#:~:text=The%20Raspberry%20Pi%20is%20a,languages%20like%20Scratch%20and%20Python.30/4/2022

[4] https://www.bitrix24.com/uses/free-gantt-chart-maker-gantt-template-excel-google-sheet.php?gclid=Cj0KCQjw-JyUBhCuARIsANUqQ_LiSCTWkh2sMi_zt_ITxOqem-kuZkrt5r76_czCs-IZm2_9-F2Vx14aAku1EALw_wcB 1/5/2022

[5] https://en.wikipedia.org/wiki/Dia_(software)1/5/2022