KRATOS

Virtual Assistant

Version 1.0

May 15, 2022

Team M4lware

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Creation v1.0	Kabeer Ahmed	Fully meets the functional requirements	05/13/22

Review and Approval

Project Plan Approval History

Approving Party	Version Approved	Signature	Date
Sir Wahabuddin	v1.0		

Project Plan Review History

Approving Party	Version Approved	Signature	Date
Sir Wahabuddin	v1.0		

Contents

1. Introduction

1.1. Overview

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behavior and trying to become his personalized assistant.

Virtual assistants are software programs that help you ease your day-to-day tasks, such as showing weather reports, creating remainders, making shopping lists etc. They can take commands via text (online chatbots) or by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana.

This system is designed to be used efficiently on desktops. Personal assistants’ software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user.

This project was started on the premise that there is a sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

1.2. Deliverables

Along with the final project, the customer will be provided with following things:

- Desktop application will be provided with the data set.
- It will be open source as All source code will be available.
- A document having all the details of the project i.e. User Manual.
- A tutorial for usage present on the web.

1.3. Assumptions and Constraints

The mass adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user’s life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years. The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee usually reserved for the rich and famous.

There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

1.4 Definitions and Acronyms

Quepy: Quepy is a python framework to transform natural language questions to queries in a database query language.

Pyttsx: Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis.

Speech Recognition: This is a library for performing speech recognition, with support for several engines and APIs, online and offline.

SQLite: SQLite is a capable library, providing an in-process relational database for efficient storage of small-to-medium-sized data sets.

Content Extraction: Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents.

Speech Recognition module: The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system.

2. Management Structure

2.1. Project Lifecycle

This project is naturally based on extreme programming, because we need high level of research and development knowledge. Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. When a user asks a question to personal assistant to perform a task, then natural language audio signal is converted into executable command or digital data that can be analyzed by the software. Then this data is compared with a data of the software to find a suitable answer. Virtual Assistant is used to run machines on your own commands. For making virtual assistant we use some python installer packages like Speech recognition, gTTS, pipwin, etc. Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called Speech Recognition to allow us to convert voice or audio command into text for further processing. By above diagram, firstly users give the command to the interaction entities like laptop, PC's this interaction entities listen the command and recognize it. For further analyzing process compare this command with cloud in which we already store data. After matching request, the output is generated in the text as well as voice form if the request is match with cloud data. Look up for the function or a logic to be executed based on request and send output of the backend process as a response.

2.2. Project Organization

This project will be carried out by a three-person team consisting of developer, architect, and researcher. One of the members is also the project manager. All team members participate while interacting with any of the external organizational entities. The identities and roles of each member of this four-person team are documented below.

2.2.1. External Interfaces

The external interfaces that will interact with the project team includes:

- Dr. Sheikh Muhammad Wahabuddin Usmani: He has approve and review this project For any queries, any of the team members can contact him either directly by face to face meeting or by contacting him via google classroom by posting the comments or WhatsApp by the help of CR.
- FYP Coordinator: He is the final person who validates the project and approves the project on the basis of its validity and working. All the team members can directly contact him via his email or face to face meeting.
- Internal Supervisor: He is the project coordinator and any member can contact this interface via email or WhatsApp contact or face to face meeting.
- External Supervisor: He is the person who specifies the requirements and validates the project for user satisfaction. Team leader will contact this person via email.

2.2.2. Internal Structure

2.2.2.1. Roles and Responsibilities

Role	Responsibility
Project Manager	The project manager serves as a liaison between the client and the team. He or she creates a time schedule with the rest of the team, and is responsible for supervising the work of the team.
Architect	The role of the software architect is to design the software being created for this project.
Researcher	Research about the new technology about Voice-based intelligent assistants.
Developer	This person is responsible for developing the code used to build the software.

2.2.2.2 Staffing

Role	Staff Member	Start Date	End Date
Project Manager	Kabeer Ahmed	05/01/2022	05/13/2022
Architect 1	Kabeer Ahmed	05/01/2022	05/04/2022
Architect 2	Nizam Ali	05/01/2022	05/04/2022
Architect 2	Rehan Mumtaz	05/01/2022	05/04/2022
Researcher 1	Kabeer Ahmed	05/05/2022	05/07/2022
Researcher 2	Rehan Mumtaz	05/05/2022	05/07/2022
Researcher 3	Nizam Ali	05/05/2022	05/07/2022
Developer 1	Kabeer Ahmed	05/08/2022	05/13/2022
Developer 2	Rehan Mumtaz	05/08/2022	05/13/2022
Developer 3	Nizam Ali	05/08/2022	05/13/2022

2.3. Communication

Weekly meetings will be held to review project updates, accomplished milestones, and upcoming milestones. The team will also communicate via email with one another. The team will engage with the customer via the project manager, who will be in charge of requesting resources and keeping the client up to date.

2.4. Risk and Asset Management

Risk	Category	Probability	Severity	Mitigation Plan
False recognition	TE	40%	2	Integration of more datasets
Software Uninstallable	TE	30%	1	Adding Different rules to give a green signal
Inexperience with virtual Assistant	TE	20%	3	Proper training needed for individual to proper recognition
slow performance	PS	60%	2	using voice API to access models
lack of coding experience	ST	30%	1	giving the training to the developers in the early phases
server overload	PS	20%	2	Design the system by considering the most probable number of users that could visit a site at a time

2.5. Startup

To startup this project, the following things are required

- The system uses Google's online speech recognition system for converting speech input to text.
- The python backend gets the output from the speech recognition module.
- An API is a software intermediary that allows two applications to talk to each other.
- Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents.
- Text-to-Speech (TTS) refers to the ability of computers to read text aloud.

2.6. Closeout

The project will ramp down or end, when

- All the requirements of the stakeholder are completed without any fault or bug.
- The user is successfully able to view the result.
- the desktop application is always working and provides responses to users in less time.

3. Planning and Control

3.1. Estimate

Python requires 10 mins to write 50 lines of code and in this project we required 21.000 lines of code so that we completed this project in 2 week by working 5 hours a day.

Prior to testing, about 30 hours should be spent developing the code for the various parts of the project. This is a 20-hour estimate for the architect and 20-hour estimate for the developer.

3.1.1. Estimation Process

The pilot project will be tested after a minor implementation of the model and application to check that it can meet all of the criteria within the predicted time frame. Estimates will alter somewhat if the test succeeds. The original estimates contain some buffer time in case any revisions are necessary or an unforeseen occurrence happens. If the test fails and a high number of modifications are necessary, the projected time and line of code values will need to be adjusted to meet the new requirements.

The project will be tested early in the development process so that if the customer is dissatisfied or new needs are discovered, there will be no complications later on and a precise and acceptable estimate and timetable can be created. Because of the project's importance, secure coding will be followed.

3.2. Resource Identification

3.2.1. Staff

The staffing of the project changes according to the requirements at the moment. During the starting phase, the workload is less and each member will divide the work among them, later on work load will increase so more than one member will have to work on one particular task. So, for these reasons, we have organized different levels of staffing.

Staff Group	Roles	Team Member
Beginning Staff	Developer	Kabeer Ahmed
	Architect	Rehan Mumtaz
	Manager	Kabeer Ahmed
	Document Author	Nizam Ali
Peak Staff	Developers	Nizam Ali
	Architect	Kabeer Ahmed
	Manager	Kabeer Ahmed
	Document Author	Reahn Mumtaz
Final Staff	Testers	Kabeer Ahmed
	Manager	Kabeer Ahmed

	Document Author	Rehan Mumtaz
	Document Assistant	Nizam Ali

3.2.2. Time

A semester lasting few weeks will be the calendar time available for the execution of this project. The team will contribute an average of 35 hours weekly to this project.

3.2.3. Cost

For this project, no financial budget is available. All the resources and tools used are free of cost available online & it is made open source so anyone will be able to give pull request and that will be welcomed

The time budget for a team of 3 people with weekly 11.5 hours will be 35 hours weekly man-hour.

3.2.4. Materials

No specialized equipment for the project will be required. All the tools and material required for the project are available online and used either by installing locally into the devices or on the online tools.

3.3. Resource Allocation

3.3.1. Milestones

Milestone	Due date
Planning and estimation of project	05/01/2022
Subprocess	05/01/2022
WolframAlpha	05/02/2022
pyttsx3	05/02/2022
JSON	05/03/2022
Speech recognition	05/04/2022
gTTS	05/05/2022
Datetime	05/06/2022
Wikipedia	05/07/2022
webbrowser	05/08/2022
OS	05/09/2022
Winshell	05/10/2022
Pyjokes	05/11/2022
Pyaudio	05/12/2022
Pyaudio	05/12/2022

ctypes	05/13/2022
Requests	05/13/2022

3.3.2. Work Breakdown Structure

Team Member	Tasks
Kabeer Ahmed	Communicate with external interfaces to give a report or to resolve any kind of query.
	Done the project planning and estimation
	Technical feasibility
	Setup the working environment for python
	Write code for the model training
	Economic feasibility
	Review all the final version of documents
Rehan Mumtaz	Collect the dataset
	Done the project planning and estimation
	Operational feasibility
	Complete draft of SDD.
	Complete draft of project documentation.
	Complete final version of project plan document and all other documents.
	Improve the usability and interface of the desktop application
	Organizational feasibility
Nizam Ali	Done the project planning and estimation
	Clean the dataset
	Setting up working environment for desktop app
	Cultural feasibility
	Design the desktop application
	Deployment of the desktop application
	Does the system and deployment testing

3.3.3. Schedule

Days	Members involved	Tasks
1	Group	Project planning and estimation is done
2	Nizam Ali	Dataset cleaning
	Kabeer Ahmed	Analyze the dataset and the documents
	Rehan Mumtaz	Dataset collection and begin documenting SRS
3	Group	Report the progress to the internal supervisor.
	Kabeer Ahmed	Setup the Flask and colab environment
	Rehan Mumtaz	Start finalizing SRS
	Nizam Ali	Setup the Desktop app coding environment
4	Kabeer Ahmed	Review SRS
	Rehan Mumtaz	Finalize the SRS
	Nizam Ali	Help with SRS
5	Kabeer Ahmed	Create design and architecture of model
	Rehan Mumtaz	Document all the designs and architectures and design decisions
	Nizam Ali	Create design of the desktop application
6	Group	Discuss the design decisions with the internal supervisor.
	Kabeer Ahmed	Code the model for the malware detection and do the pre training
	Rehan Mumtaz	Speech Recognition
	Nizam Ali	Speech Recognition
7	Kabeer Ahmed	Integrate the model with Flask Api and deploy
	Rehan Mumtaz	Improve the desktop application responsiveness
	Nizam Ali	Make the front end of the desktop application
8	Group	Meeting to demonstrate the progress
	Kabeer Ahmed	Help with documentation
	Rehan Mumtaz	Begin writing the project report and documentation
	Nizam Ali	Call api and display results on desktop application
9	Kabeer Ahmed	Test the system
	Rehan Mumtaz	Test the system
	Nizam Ali	Test the system

10	Group	Meeting to demonstrate the working
	Kabeer Ahmed	Help with documentation
	Rehan Mumtaz	Complete the documentation
	Nizam Ali	Deploy the desktop application
11	Kabeer Ahmed	Deployment Testing
	Rehan Mumtaz	Deployment Testing
11	Group	Present the functional desktop application to the external interfaces
	Kabeer Ahmed	Code the model for the malware mitigation and do the pre training
	Nizam Ali	Speech Recognition
12	Kabeer Ahmed	Integrate the model with Flask Api and deploy
	Rehan Mumtaz	Improve the desktop application responsiveness
	Nizam Ali	Make the front end of the desktop application
12	Group	Meeting to demonstrate the progress
	Kabeer Ahmed	Help with documentation
	Rehan Mumtaz	Begin writing the project report and documentation
	Nizam Ali	Call api and display results on desktop application
13	Kabeer Ahmed	Test the system
	Rehan Mumtaz	Test the system
	Nizam Ali	Test the system
13	Kabeer Ahmed	Help with documentation
	Rehan Mumtaz	Complete the documentation
	Nizam Ali	Deploy the desktop application
14	Kabeer Ahmed	Deployment Testing
	Rehan Mumtaz	Deployment Testing
	Nizam Ali	Deployment Testing
14	Group	Present the functional desktop application to the internal supervisor
	Kabeer Ahmed	Supervise the documentation
	Rehan Mumtaz	Finalize the documentation
	Nizam Ali	Help with documentation
14	Group	Present the functional desktop application to the external interfaces

3.4. Tracking and Control

Each member tracks the time they have spent on each task and reports it to the team leader or team manager. The team lead will make a weekly report and track the progress against the schedule to ensure that there is no or little diversion from the schedule.

Each member after completing their work will perform the testing before completing the milestone and ensure that the quality is maintained and the requirements are fulfilled. They will provide their progress to the manager who will make reports and present it in the meeting with the external interfaces.

Manager will ensure that all the milestones are successfully completed and the schedule is followed. She will also ensure that if there is any need to make changes in the schedule or plan then those changes should be accommodated.

For version control, the code will be pushed to the GitHub repository and after completion all the branches will be merged together.

4. Technical Process

4.1. Engineering

The entire project will consist of four main areas.

Speech Recognition module: The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information Centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

Python Backend: The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

API calls: API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

Content Extraction: Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

Text-to-speech module: Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

4.1.1. Environment

The project is built using open source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

4.1.2. Methods, Tools and Techniques

Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. When a user asks a question to personal assistant to perform a task, then natural language audio signal is converted into executable command or digital data that can be analyzed by the software. Then this data is compared with a data of the software to find a suitable answer. Virtual Assistant is used to run machines on your own commands. For making virtual assistant we use some python installer packages like Speech recognition, gTTS, pipwin, etc. Speech recognition is the process of converting audio into text. This is commonly used in voice assistants like Alexa, Siri, etc. Python provides an API called Speech Recognition to allow us to convert voice or audio command into text for further processing.

The following software are needed:

- Windows 7(32-bit) or above.
- Python 2.7 or later
- Chrome Driver
- Selenium Web Automation
- SQLite

4.2. Technology

4.2.1. Environment

Multiple technologies for different elements of our system have been chosen. Python environment for the frontend will be set up by installing the required python modules. For api, Flask will be used as API in our system. Pytorch code will be written for models and the dataset will be loaded in the zip format which can easily be extracted using Pytorch.

4.2.2. Methods, Tools, and Techniques

Visual Studio Code and Pycharm are going to be used for the project development. The model will be trained on Google Colab and the dataset will be stored in Google Drive. During model training, our model will also be saved in drive from time to time.

4.3. Infrastructure

Our development environment is mostly based on the google services so for this reason, a google account is needed which we already have. Thus, we just need to sign in to services such as Google Colab, Google Drive, Google Docs and GitHub.

For the frontend, we will have to install VS code and Python. Same case for the backend where we will have to start the Flask development server after installing Python in our system. From Python we will use the pip command to install the Flask. Our all work will be done on the localhost server.

A git repository will be created and different branches for each member will be created where the members will push their data.

5. Supporting Plans

5.1. Configuration Management

There will be a .config file to save all the configurations such as secret key, port no, max size etc which will be used to ensure that all the constraints are always applied.

The changes can be tracked and version history can be controlled using GitHub as all the code will be timely pushed. This helps to ensure that all the developers are working on the latest files and dependencies. Each member will push the code in their branch and the team manager will merge all those branches into the master branch.

5.2. Quality Assurance

Whenever a new functionality is built, it is tracked against the SRS and SDD to ensure that the specified requirements are set true. Also the Gantt Chart is timely updated to mark the milestones achieved and to track the progress of the project which also ensures that the schedule is followed. Weekly meetings with the internal supervisor will be held to get the reviews and ensure that the quality is maintained.

There will be a list of test cases to be performed on each functionality and after the completion of the functionality, it will be tested against those test cases and a file will be maintained where the result of those tests will be mentioned. If any test case fails then the developer is informed of those errors.

5.3. Testing

Test cases will be made for each type of functionality and different tests will be performed so as to have the maximum test coverage.

Unit Testing: Each developer will have to perform the unit testing on the component they have built before marking the milestone. In unit testing, the functionality will be tested along with the code on various test cases to ensure that function works well on each kind of scenario.

Integration Testing: Whenever the component has passed the unit test, it is then merged with the previous work and then an integration test is conducted which ensures that the components still work well independently and also with each other.

System Testing: When the whole system will be developed, then system testing would be performed to ensure that the frontend, backend and model all are working well and response from the model is passed to the frontend from the api. Also, it is ensured that the request from the frontend is passed to the model successfully.

User Acceptance Testing: It is to ensure that the desktop interface is usable and users have no issue in performing the actions and viewing the data.

Deployment Testing: Deployment testing is performed after the deployment of the desktop application and the api and the model to ensure that all functionalities are still working fine and performance of the application is not decreased even on the load.

5.4. Deployment

The system will be provided to the user as a form of desktop application. The application will be deployed on the AWS platform. The Flask application will also be deployed on the same platform. We will use Docker for the model deployment and securely integrate the Flask application with the model.

5.5. Integration

Our system will be divided into three parts. The desktop interface, api and the model. The api will be integrated with the model by passing a call to the model function. Then the desktop interface will call the api to interact with the model.

Our team will have 3 members and each will be working on different tasks. Each member will push their work in their respective branches on the git repository. The team lead will merge all the git branches from time to time and each member will pull the work from the master branch. In this way, the work of each member will be integrated together.

5.6. Product Acceptance

The product can only be accepted if the model that is trained has accuracy over 90%. If the model is not giving accuracy more than 90% then the data set needs to be changed or some other data augmentation techniques should be applied to get the required accuracy otherwise the project will not be considered successful.