NRI INSTITUTE OF TECHNOLOGY

BLACKBUCKS

AI VOICE ASSISTANT

Aspirants:

K. Divya

G. Pavani

N. Jaswanth

CONTENTS

- Introduction
- Problem Defnition
- Existing System
- Proposed System
- Software Requirements & Hardware Requirements
- Module Description
- Future Enhancement
- Architecture
- Conclusion

INTRODUCTION

Knowingly, or unknowingly, personal assistants have become an integral part of our lives these days. It is because of all the features and ease of use they provide. Personal assistants are also capable of automating some day-to-day tasks, so that a user can focus on what matters the most to them. Features like, making calls, writing messages, taking photographs, storing to-dos on the go, browsing internet etc., Are offered by personal assistants. So, utilization of these features of a virtual assistant will save an individual a lot of time, and effort. It is important to focus more on what matters the most for an individual, whether it could be personal work, or professional work. People often spend more time on doing routine tasks, and they can be automated with these types of personal assistants. When someone works in an environment with which he/she is not familiar with, they often find it difficult to locate applications that they need, like browser, any IDE or nay other software. Most of the time, they will end up wasting hours of time, searching for the application alone. This results in unnecessary time wastage. Therefore, a voice enabled personal assistant will help automating this process. User is expected just to give a voice command, and the assistant will take care of the rest. The paper indicates the usage of a voice enabled personal assistant and it can enable an individual to get things done with voice commands, and can save a lot of their time as well.

PROBLEM DEFNITION

The project titled “Intelligent Voice Assistant” is a standalone application that provides a unique user interface to perform everyday tasks on the system. This software can assist users to achieve tasks that they would normally do on their own. The reason for this, is that speaking the command will be easier and faster than typing or clicking. It offers a natural language user interface to customers. The application can accept input in the form of spoken words or phrases. This can be taken from the microphone, which most devices already have embedded in them. The appropriate command is obtained and executed. Once the command is done executing, appropriate feedback is provided for continuation of the recognition process.

EXISTING SYSTEM

- The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.
- If the system is not able to gather information from the user input it will keep asking again to repeat till the desired no. of times.
- The system can have both male and female voices according to user requirements.
- Features supported in the current version include playing music, emails, texts, search on Wikipedia, or opening system installed applications, opening anything on the web browser, etc.
- The system will keep listening for commands and the time for listening is variable which can be changed according to user requirements.

DISADVANTAGES OF EXISTING SYSTEM:

- Privacy concerns
- Accuracy
- Fragmeted Systems

PROPOSED SYSTEM

The proposed system is an adaptation of existing open-source voice assistants, with improved functionality, accuracy and security measures. It aims to parallel smaller ranks of Google Assistant in terms of accuracy and Apple's Siri in terms of security. The project is a standalone application that can integrate with any platform and will be released as an open-source software.

Through this application, the customer will be able to perform tasks such as searching online, opening applications and conversing with the application. The application analyzes data quickly and return outputs in a matter of seconds. Every time a command is spoken, it translates the speech into text and displays it on the screen. It provides appropriate feedback for the user to understand that it has listened and that the output has been displayed successfully.

The Google Speech Recognition package will be used in Python to convert the speech heard to text. The webpage will be displayed for the user to instruct them to start speaking. Then, the input will be received from the in-built microphone through the Python code and recognized words will be sent to the process. From there, the words will be sent to the webpage to display it on the screen. The user can validate recognized speech conversion to text, by clicking one of the two buttons on the screen. If correct, the words are sent back to the process, which sends it to a new Python child process that will perform the appropriate system tasks based on the words provided. Once it finishes, it will send back data to the process, which will display it once again on the webpage, before refreshing the page to take the next input command. Thus, there is a series of interactions between the main process its two child processes (Python), the webpage (GUI) and the user. There are appropriate instructions, error messages and animations on the display, to engage the user.

The resultant application will be runnable on personal computers of different operating systems and will avoid the general threat of security, since it doesn't save the user data in any way. The application will depend on certain technologies that have been discussed below in detail.

REQUIREMENTS

HARDWARE

- Processor : Any processor above

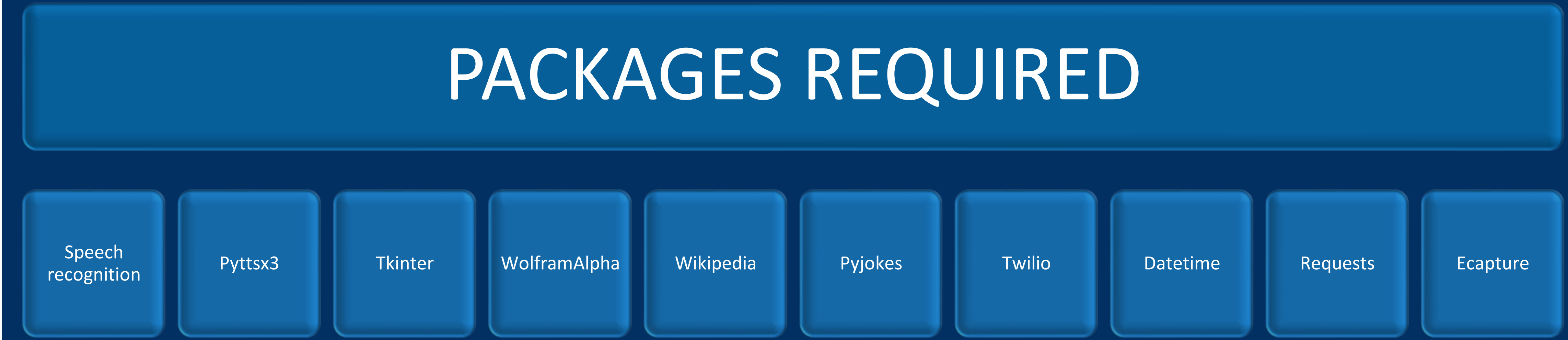
 1 GHz

- RAM : 1 GB
- Hard Disk : 10 GB
- Input device : In-built microphone, mouse and keyboard
- Output device : Monitor or display

SOFTWARE

- Operating system : No restrictions
- Front End : Chromium or Chrome (written in JavaScript)
- Back End : Node.js
- Server : Google Speech Recognition API

MODULES DESCRIPTION



PACKAGES REQUIRED

- Speech recognition
- Pyttsx3
- Tkinter
- WolframAlpha
- Wikipedia
- Pyjokes
- Twilio
- Datetime
- Requests
- Ecapture

GOOGLE SPEECH API: This application uses the Google Speech Recognition API through Python.

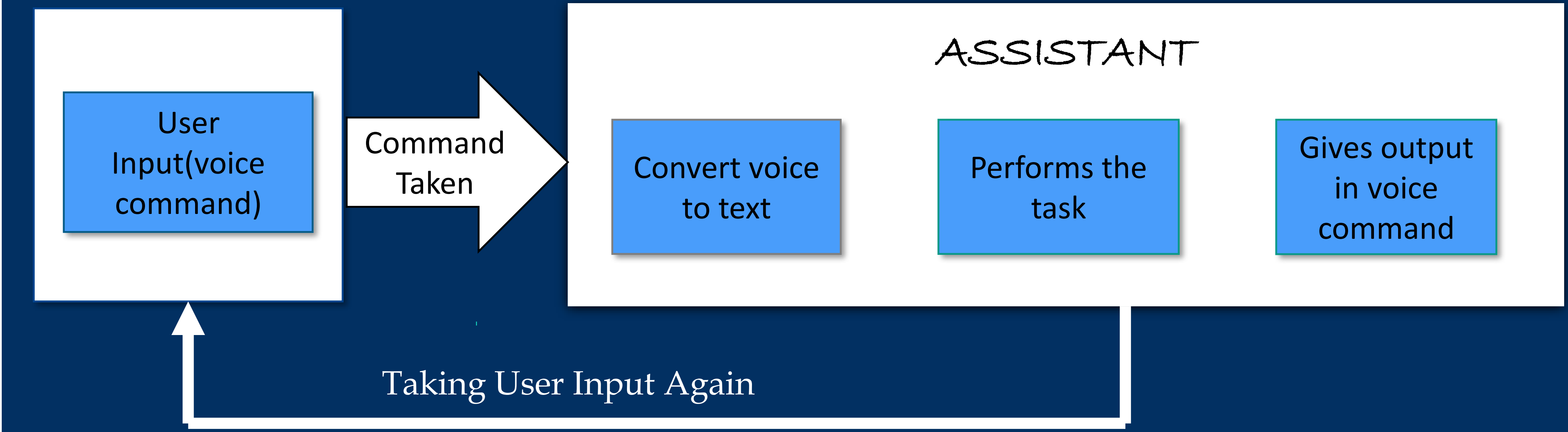
WEB TECHNOLOGIES: The concepts of HTML, CSS and JavaScript are major modules of this project.

PYTHON PROCESS: In this project, the automation is done using Python.

SYSTEM ARCHITECTURE

This Assistant consists of three modules. First is, assistant accepting voice input from user. Secondly, analysing the input given by the user, and mapping it to the respective intent and function. And the third is, the assistant giving user the result all along with voice.

Initially, the assistant will start accepting the user input. After receiving the input, the assistant will convert the analog voice input to the digital text. If assistant was not able to convert the voice into text, it will start asking user for the input again. If converted, it will start analyzing the input and will map the input with particular function. And later, the output will be given to user via the voice command.



FUTURE ENHANCEMENT

In the future versions of this project, some attributes that can be added are:

- Facility for extended conversation (like a chatbot)
- Features for persistent data storage
- Personalization for different users
- Voice biometrics and security

CONCLUSION

The final outcome of this project is an intelligent voice assistant, as described in the title. It combines natural language processing techniques with web programming to present an efficient digital personal assistant. It can give answers for some of the user questions. It can perform everyday tasks on the system based on the spoken command of the user. It can recognize the words, map the speech into text and decide what task to execute accordingly. The application can perform operations on the device such as open applications like settings, calculator, Microsoft Word, media players, etc. It can also change volume settings, open websites, perform online searches and minimize all open windows on the computer.

THANK YOU 👍