iNeuron

HIGH LEVEL DOCUMENT

Project Title : Neuron Desktop Assistant

Created by: iNeuron Private Limited

Date: 12/4/2022

Contents

Abstract

Managing desktops manually sometimes irritates when we see there are many Automatic desktop assistants available over the internet people are being used. We all must be familiar with the Iron Man movie. There we saw an artificial intelligence voice based assistant program that had been used by Tony Stark. It was amazing right! Wondering if we would have the same kinds of assistants in our day to day life. Voice technology within your company can help to improve your employees' productivity as voice assistants can remember important dates or deadlines, schedule appointments and in general, keep the relevant information updated. With integrating AI innovations and Deep Learning, the voice assistants work without stopping! In this project we have implemented a Python based desktop assistant system which can perform some specific conditional tasks defined by the user like opening notepad, VS code, Pycharm, setting up alarm, sending email to friends & family and so on.

1. Introduction

1.1 Why this High-Level Design Document ?

The purpose of this High-Level Design (HLD) Document is to add the important details about this project. Through this HLD Document, I’m going to describe every small requirement about this project.

2. General Description

2.1 Product Perspective

To automate & make interesting desktops. Whatever tasks we do manually it can be done using voice commands in a very easy way.

2.2 Problem Statement

The current voice assistant system basically exists on Windows OS is Cortana which is a completely online-based system and requires high-speed fast internet and also a regular Microsoft account for login and another existing system is Ok-Google voice assistant which is browser dependent. But our assistant is fully customizable. You can add your own tasks what you want to do by your assistant.

2.3 Proposed Solution

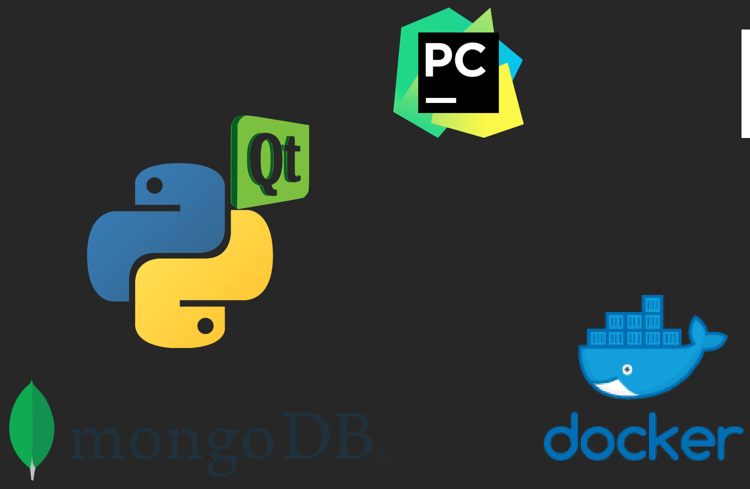
This solution gives an outline thought of a personal-assistant system. The framework draws its motivation from virtual assistants like Cortana for Windows and Siri for iOS. It has been planned to supply a user-friendly interface for carrying out an assortment of errands by utilising certain well-defined commands. As an individual right hand, this project centres upon the assistance of the end-user with day-to-day exercises like general human discussion, looking at questions on Google, searching for sons, recovering pictures, live climate conditions, looking for IP addresses, opening notepad, vs code and so on. The user statements/commands are analysed with the help of a machine learning module to give an optimal solution.

2.4 **Technical Requirements**

Here are some requirements for this project.

1. Nosql database MongoDb to store the logs and exceptions.
2. PyQt5 for GUI

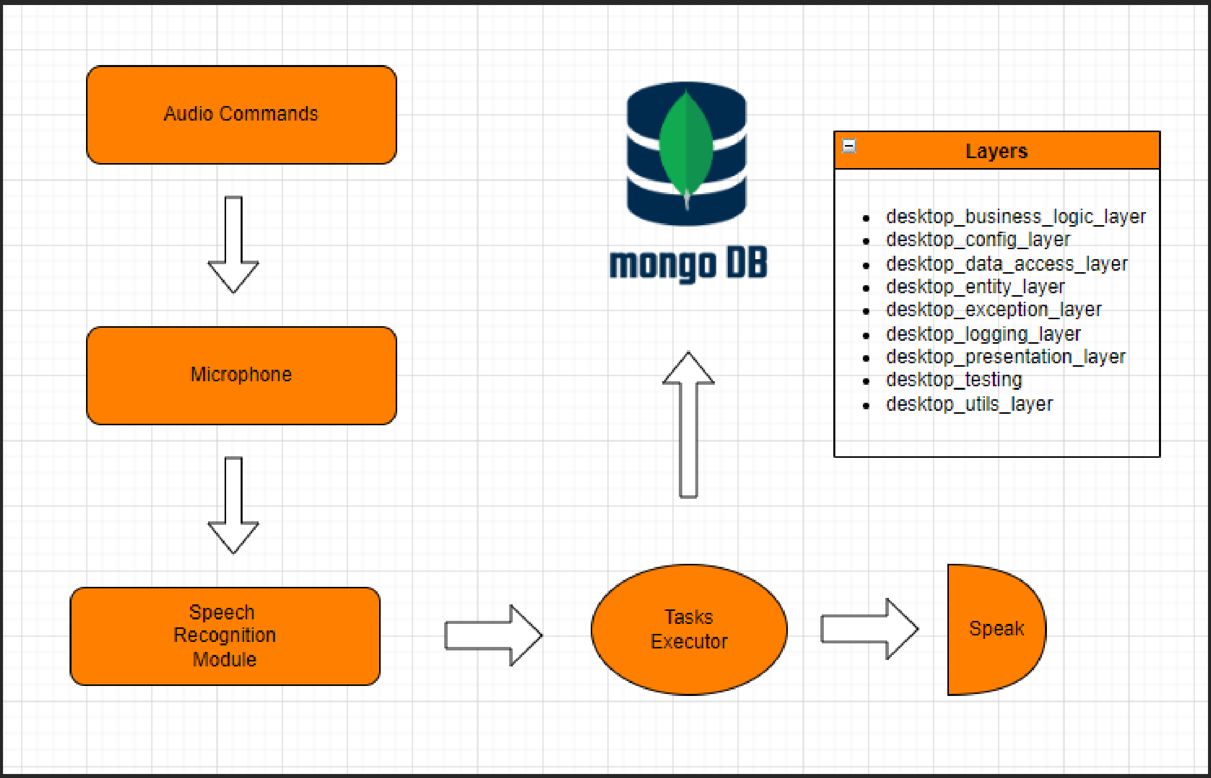
2.5 **Tools Used**



- PyCharm is used as an IDE.
- PyQt5 for GUI
- Mongodb is used to retrieve, insert, delete, and update the database.
- Front end development using PyQt5
- Python is used for the backend.
- GitHub is used as a version control system.

3. Design Details

3.1 Process Flow



3.2 Event Log

In this Project we are logging every process so that the user will know what process is running internally. We have defined logging for every function, class.

By logging we can monitor every insertion and flow of data in the database, monitor every step which may create problems or every step which is important in the file system.

3.3 Error Handling

We have defined our own exception handling class so we can modify exception errors thrown by any event. If any exception occurs we catch its file name, line number and error message.

4. Performance

4.1 Re-usability

We have followed industry standards while coding and also maintained an object oriented programming approach so in future you can change components or add without any issues.

We have given initial setup script which uses can edit and add components to replicate project

4.2 Application Compatibility

This application is compatible with python 3.7 and above, versions of required libraries are mentioned in the requirements.txt which users can look and twick accordingly.

Code is compatible with both Windows and Linux operating systems. Development system was Windows 10 while the Testing Environment was ubuntu.

4.3 Resource Utilisation

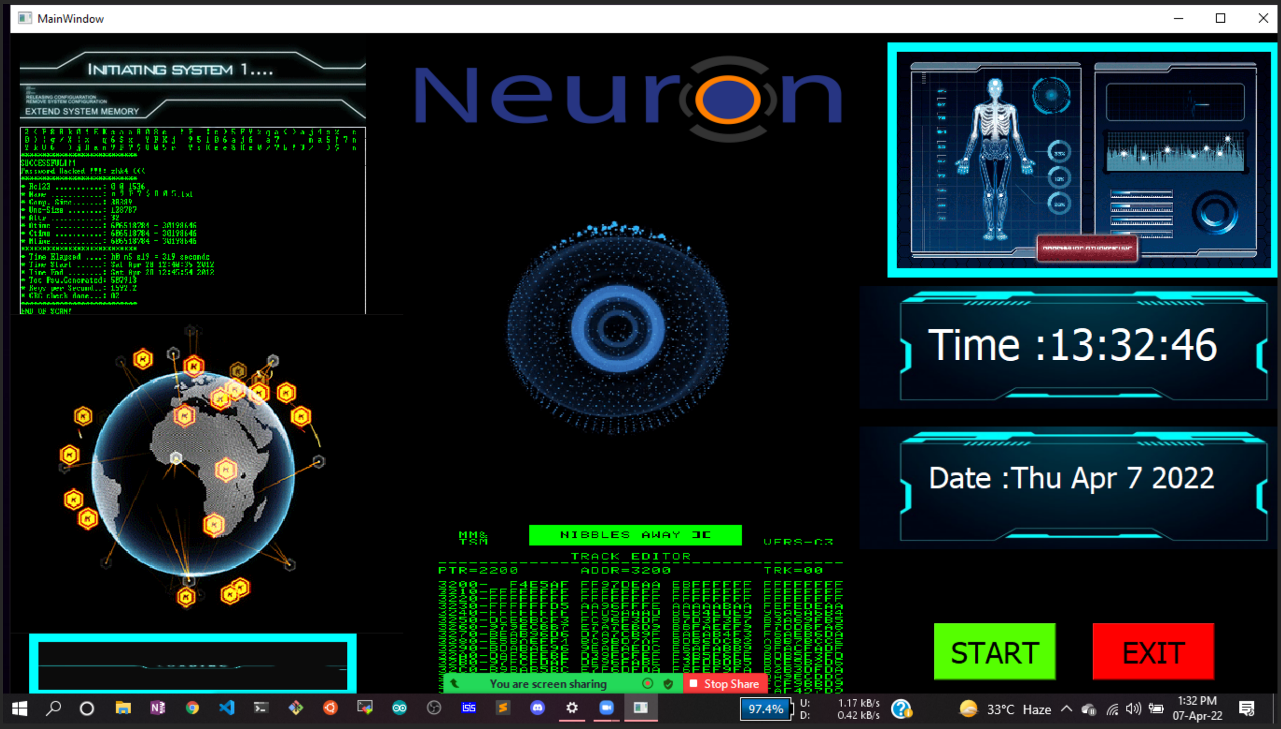
Code is written with proper optimization with the aim in mind to use less resources and perform better. Code to access Database is written with functionality to remove data from if needed. Once a command is spoken by the user it will save, execute & save all the logs , exceptions in mongo db.

4.4 User Interface

There is 1 main UI

- Main UI

Main UI



5. Conclusion

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, IP addresses, opening notepad, vs code and so on. The future plans include integrating our software with mobile to provide a synchronised experience between the two connected devices.