iNeuron

Desktop Assistant Synopsis

Title: Desktop Assistant

Author: Boktiar Ahmed Bappy

Created date: 14 April 2020

Table of Contents

1. Title: Neuron Desktop Assistant (Desktop Application)

2. About the Problem:

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.

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.

3. The primary reason to choose this particular topic:

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.

4. The main objective of the project:

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

5. Scope of the Project:

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.

6. Working Methodology:

This Desktop assistant will be a desktop application design using PyQt5. The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. To build our application we will be using the python PyQt5 framework.

7. Details about the Hardware & Software used:

- Processors: Any two or higher core processor including Intel® Core™ i5 @2.60GHz, new-gen Xeon® processor @2.30 GHz or AMD Ryzen 5 CPUs running at higher frequency
- RAM: 4GB of system memory from any decent manufacturer
- Disk space: 2-3GB of SEAGATE Hard Drive
- Operating System: Windows 10 Official, Mac OS 10.12.6 (and up), or Linux/Ubuntu 16.10+
- Software: Python, Pycharm or Visual Studio Code, Google Chrome.

8. Listing out the Testing Technologies:

- Unit Testing: Testing individual pieces of software code to ensure that code is flawless.
- Blackbox testing: Testing outcomes without worrying about internal implementation.
- Integration Testing: Integrating all individual pieces of software code as whole and testing whether it is working as per expectation.
- System Testing: System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.
- Stress Testing: Stress testing is the process of determining the ability of a computer, network, program or device to maintain a certain level of effectiveness under unfavourable conditions.

9. Limitations of the system proposed:

This is just a python based assistant so it will be only working on conditional basis.

10. Specifying the contribution that the project would make

This project will allows user to automate & make interesting desktops. Whatever tasks we do manually it can be done using voice commands in a very easy way.

11. Future enhancement:

The future plans include integrating our software with mobile to provide a synchronized experience between the two connected devices.

12. 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 synchronized experience between the two connected devices.