Table of Contents

1. Introduction

1.1 Purpose

This SRS describes the requirements and specifications of **Voice Assistant**, **a voice-controlled personal assistant** designed for the **Linux** platform. It explains the **functional features** of the Personal Assistant, along **with interface details**, **user constraints** and related considerations. This SRS is targeted towards the team of developers involved in the development of Voice Assistant as a first step in the developmental process. The intended audience also includes users of Voice Assistant and other stakeholders of the project.

1.2 Definitions and abbreviations:

- NL: Plain English
- VI: Voice Input
- VO: Voice Output (outputted by the system)
- START: A special reserved key phrase used to bring system to Active state.
- STOP: A special reserved key phrase used to put system to Passive state.
- INTENT: An abstraction of the NL input given by the User, which represents what Action the user “intends” to achieve. It includes in its properties any information required to realize the INTENT.
- Action: The system operation to be performed /A realization of an INTENT through sequence of system operations.
- DESC: Description
- RAT : Rationale
- PLAYER : The default music/movie player(subject to constraint) as per context.
- BROWSER : The default web browser assumed to be Google Chrome or Mozilla Firefox.

1.3 Project Scope

The Voice Assistant is a simple stand-alone application designed to provide its users with a platform to execute and control some basic tasks on the Linux platform, entirely through voice input. This system is a simple, fast and robust tool that can understand

and execute actions described by the user in simple English. The scope of this project is well defined through its **Goals**.

1.4 Goals

The following are the goals of our project –

- To enable the user to control and execute some basic functionality associated with Linux from a distance and without the use of keyboard or mouse input.
- To enable the user to start and execute simple tasks to run concurrently in the background using voice, hence leaving his hands free to continue with whatever work they are already doing with the help of keyboard and mouse input.
- To enable the user to automate complex/ repetitive tasks through simple descriptions/ commands made in natural language.
- To make the Linux experience simpler, easier and more "natural" through voice control.

2. Overall Description

2.1 Product Perspective

The system will consist of a simple GUI as detailed later. The user will describe the Action through simple commands made in NL. The input will be in the form of speech inputted through an external microphone. The input will then be passed through voice processing technology (integrated into the system through external libraries) to convert it into text. The text will then be processed by the system, and the Action will be executed. The system will execute the Action by invoking commands in the default Linux Shell or by using external libraries that enable interaction with GUI of other applications via mouse and keyboard actions.

The system will exist in two states:

1. **Active**: In this state the system will process microphone input as commands.
2. **Passive**: In this state the system will monitor microphone input for the special keyword START, which will cause it to transition to Active state.

The system can transition from one state to another by use of a toggle switch in the GUI of the application or through voice commands as specified later in the document.

2.2 Product Functions

The user will be able to execute the following product functions –

- The user will be able to operate a Music/Movie player using voice commands in NL and navigate through the Music/Movie Library.
- The user will be able to operate certain basic functions of the BROWSER using voice commands in NL.
- The user will be able navigate the default File Browser, and perform basic file management operations using voice commands in NL.

- The user will be able to carry out some basic operations related to processes using voice commands in NL.
- The user should be able to access functionalities that are generally executable only through the terminal, and involve considerations of syntax and semantics, by simple descriptions of the task in natural language.

2.3 Operating Environment

The application will run on Ubuntu and Ubuntu–like Linux distros.

2.4 User Constraints

The following are the constraints imposed on the user –

1. The words being used in User speech should be restricted to the Vocabulary defined in the application. This vocabulary will at least contain the words of the English Dictionary.
2. Ambient noise should be minimum when the user is trying to input a voice command.
3. The user sets the system to passive state, if there is ambient noise and conversations going on which can be picked up by the audio input device.
4. The user sets the system to active state before issuing voice commands.
5. The user limits his speech to be Understandable. We define Understandable to include at least the set of all sentences correct under the English Grammar.

2.5 Hardware Constraints

The platform on which the system is running should at least have access to an audio input device such as a microphone for the most basic functionality.

3. Specific Requirements

3.1 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

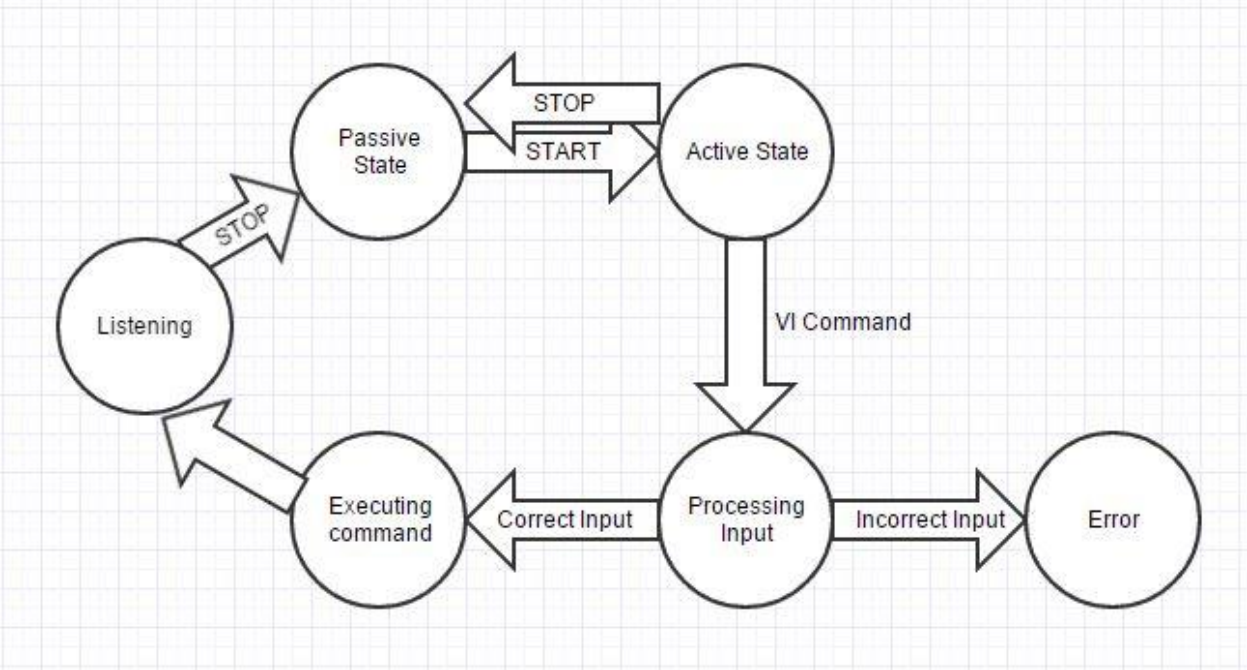


Fig. Flow diagram of System

ID: FR1

TITLE: VI Controlled Start

DESC: When the system is in Passive state, the user should be able to put the system into Active state by speaking a special keyword START into the microphone input.

ID: FR2

TITLE: VI Controlled Stop

DESC: When the system is in Active state, the user should be able to put the system into Passive state by speaking a special keyword STOP into the microphone input.

RAT: In order that ambient noise and conversations picked up by microphone unit are not misinterpreted as description of an Action.

Scenario: User says “STOP”. Thereafter user says “Update system”. The system should not take any action.

ID: FR3

TITLE: GUI Toggle Start/Stop

DESC: The user should be able to toggle between Active and Passive States using the GUI.

RAT: In case FR1/FR2 fail due to a lot of ambient noise, it would be useful to be able to turn the System into Passive state to prevent unwanted Actions. Also sometimes it might be simpler to use GUI to achieve Start/Stop functionality.

ID: FR4

TITLE: GUI Exit

DESC: The user should be shut down the system completely from the GUI.

ID: FR5

TITLE: GUI System Message

DESC: If the user INTENT requires return of some piece of information or text, the GUI should display this in human readable format. Error messages and suggestions should also be displayed in the GUI.

Scenario: The user attempts to search for a song that doesn't exist in the library. GUI should display "File Not Found".

ID: FR6

TITLE: VO System Message

DESC: The same functionality of FR5 should be read out by system through audio output devices such as speakers.

Scenario: The user attempts to search for a song that doesn't exist in the library. The system should say "File Not Found".

ID: FR7

TITLE: Stop VO

DESC: The functionality FR6 should be stopped and system should restrict itself to functionality FR5.

ID: FR8

TITLE: GUI Settings

DESC: The user should be able to change settings of the system through the GUI.

ID: FR9

TITLE: VI Settings

DESC: The user should be able to change settings of the system through VI.

ID: FR10

TITLE: VI Command

DESC: The user should be able to pass INTENT through voice, when the system is active.

ID: FR11

TITLE: Wrong Input

DESC: If the user's input cannot be processed correctly to an INTENT, the system should prompt the user to re input, along with suggestions of what could be a correct input for the user's INTENT.

ID: FR12

TITLE: Music/Movie Library Operations

DESC: If the user INTENT inputted is one of the following, the corresponding Action should be executed by the system (through PLAYER) –

1. Search library with keywords specified in the INTENT.
2. Play a specific song/video specified in the INTENT.
3. add song specified in the INTENT to playlist

ID: FR13

TITLE: Music/Movie In-Play Operations

DESC: If an audio/video file is being played by the PLAYER, and the user INTENT is one of the following, it should be realized –

1. pause song
2. change the volume (increase or decrease)
3. stop song
4. next song, previous song
5. shuffle
6. fast forward
7. rewind
8. increase/decrease speed

Scenario: Movie is playing on player. User says " Play it faster". Speed changes to 1.5x. User repeats "Play it even faster". Speed changes 2.5x.

ID: FR14

TITLE: Web Browser Operations

DESC: If the BROWSER is running and in the foreground, and the user INTENT is one of the following, it should be realized –

1. open new tab
2. close tab
3. change tab
4. close browser
5. scroll
6. search page
7. zoom in, zoom out
8. add a bookmark
9. open a bookmark
10. back, forward, refresh
11. open history
12. save page
13. print page
14. search a keyword using default search engine

ID: FR15

TITLE: Webpage Content Operations
DESC: If the BROWSER is running and in the foreground, and the user INTENT is one of the following, it should be realized –

1. search a website using its own search dialog with keywords specified in the INTENT
2. open a link on a webpage by specifying keywords that relate to the link in the INTENT

Scenario : Movie is playing on player. User says " Play it faster". Speed changes to 1.5x. User repeats "Play it even faster". Speed changes 2.5x.

Scenario : On Google search results page. 3rd link is from Quora. User says " Open 3rd link"/"Open Quora's link". System opens 3rd link in same tab.

ID: FR16

TITLE: File Browser Operations
DESC: If the default File Browser is running and in the foreground, and the user INTENT is one of the following, it should be realized –

1. copy and move files from one location to another
2. move back to previous directory
3. create new directory
4. open directory within current directory
5. open file in current directory
6. delete files and directories in current directory
7. search current directory

Scenario : User says " Delete/Remove/Scrap all pdf files/all files with pdf extension" . System should delete all files ending with .pdf.

ID: FR17

TITLE: Process Start
DESC: If the user INTENT is to start a particular application or create another instance of it, it should be realized.

ID: FR18

TITLE: Process Management Operations
DESC: If a process instance is running , and the user INTENT relates to it , and is one of the following, it should be realized-

1. close a program instance
2. bring to foreground
3. send to background
4. minimize

Scenario : The user says " Bring Chrome to front". Google chrome comes to the front of the screen.

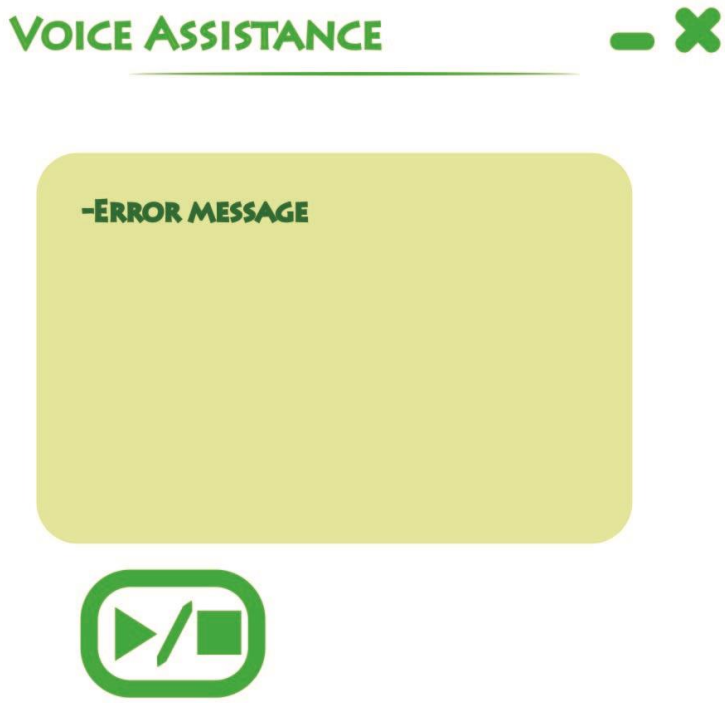
ID: FR19

TITLE: Command Line Operations
DESC: The user should be able to access functionalities that are generally accessible only executable through terminal, and involve considerations of syntax and semantics.
Scenario : The user says "Update system ". The system runs apt-get update

3.2External Interface Requirements

1. User Interfaces

There will be two user interfaces – a Graphical User Interface (GUI) and a Voice Based Interface.



- Graphical User Interface:

The minimum requirements that the GUI must fulfill are –

- It should consist of a small window, not occupying more than 1/12th of the total screen space.
- It must have buttons, toggle switches etc required to implement the relevant FRs.

- Voice Based Interface:

Its requirements are specified by the relevant FRs.

2. Hardware Interfaces:

The system needs to interface with the computer's sound input and output devices. This is achieved through the use of external python libraries.

3. Software Interfaces :

- The system interfaces with the inbuilt Linux shell
- It interfaces with GUI of other applications by simulation of clicks and keyboard input, realized through external libraries.

4. Communications Interfaces:

For realizing the some of the functional requirements pertaining to FR14 and FR15 it requires access to the internet. Otherwise internet connectivity is not required

3.3Non-functional requirements

1. Reliability :

   Although certain user constraints are imposed, in the interests of reliability and robustness, the following reliability requirements need to met -

- If the noise levels are high, than the system switches to passive state, with a system message informing the user of the same.
- If the user attempts to execute a voice command which is erroneous (for ex – closing an application when it is not running) the system should generate an error message.
- The error messages generated by the system should be informative and enable the user to properly troubleshoot the situation.

2. Legal Requirement:

   The application should be released under GNU GPL v3.