Voice Assistant

Revision Number: 1.0

Last date of revision: 25/08/2021

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 25/08/2021 | 1 | Initial HLD — V1.0 | Udaykiran Patnaik |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Contents

[Document Version Control](#_Document_Version_Control). 2

[Abstract](#_Abstract). 4

1. [Introduction](#_1_Introduction) 5
   1. [Why this High-Level Design Document?](#_Why_this_High-Level) 5
   2. [Scope](#_Scope). 5
   3. [Definitions](#_TOC_250016) 5
2. [General Description](#_General_Description). 6
   1. [Product Perspective](#_Product_Perspective) 6
   2. [Problem statement](#_Problem_statement) 6
   3. [Proposed Solution](#_Proposed_Solution) 6
   4. [Further Improvements](#_Further_Improvements) 6
   5. [Technical Requirements](#_Technical_Requirements). 6
   6. [Data Requirements](#_Data_Requirements) 7
   7. [Tools used](#_Tools_used). 7
      1. [Hardware Requirements](#_Hardware_Requirements). 7
   8. [Constraints](#_Constraints) 8
   9. [Assumptions](#_Assumptions). 8
3. [Design Details](#_Design_Details) 9
   1. [Process Flow](#_Process_Flow). 9
      1. [Model Training and Evaluation](#_3.1.1_Model_Training) 9
      2. [Deployment Process](#_Deployment_Process) 10
   2. [Event log](#_Event_log) 10
   3. [Error Handling](#_Error_Handling) 10
4. [Performance](#_Performance). 11
   1. [Reusability](#_Reusability). 11
   2. [Application Compatibility](#_Application_Compatibility) 11
   3. [Resource Utilization](#_Resource_Utilization) 11
   4. [Deployment](#_Deployment). 11
5. [Dashboards](#_Dashboards). 12
   1. [KPIs (Key Performance Indicators)](#_KPls_(Key_Performance) 12
6. [Conclusion](#_Conclusion) 12
7. [References](#_References) 12

# Abstract

In recent years, Artificial Intelligence (AI) has shown significant progress and its potential is growing. An application area of AI is Natural Language Processing (NLP). Voice assistants incorporate AI by using cloud computing and can communicate with the users in natural language. Voice assistants are easy to use and thus there are millions of devices that incorporates them in households nowadays. Most common devices with voice assistants are smart speakers and they have just started to be used in schools and universities.

Voice assistants use technologies like voice recognition, speech synthesis, and Natural Language Processing (NLP) to provide services to the users.

Voice assistants have several interesting capabilities such as:

* Answer to questions asked by users.
* Play music from streaming music services.
* Set timers or alarms.
* Play games.
* Make calls or send messages.
* Make purchases.
* Provide information about the weather.
* Control other smart devices (lights, locks, thermostats, vacuum cleaners, switches)

After the voice assistant hears its [signal](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) word, it starts recording. The [device](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) waits for a pause to know you’ve finished your [request](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us). The voice assistant then sends your recorded [request](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) over the Internet to its [database](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us).

Once in the [database](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us), your [request](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) is compared to other requests. It’s split into separate commands your voice assistant can understand. The [database](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) then sends these commands back to the voice assistant. Once it receives the commands, the voice assistant knows what to do next. The [device](https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us) might ask a question to make sure it understands what you want. If it thinks it understands, the voice assistant will carry out the task you asked for.

# **Introduction**

###### Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:
  + - * Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability

##### **Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

##### **Definitions**

*Term*

*Database*

*IDE AWS*

*Description*

Collection of all the information monitored by this system

Integrated Development Environment

Amazon Web Services

### **General Description**

#### Product Perspective

The basic idea of voice assistant is that the user makes a request through the voice-activated device, and then, the voice request gets streamed through the cloud, and here voice gets converted into text. Then, the text request goes to the backend and after processing, the backend replies with a text response. Finally, the text response goes through the cloud and gets transformed into voice, which will be streamed back to the user.

#### Problem statement

To create an AI solution for deep learning NLP based voice assistance system to implement the following use cases.

* Answer to questions asked by users.
* Play music from streaming music services.
* Set timers or alarms.
* Provide information about the weather.

#### Proposed Solution

The solution proposed here is deep learning and natural language processing-based voice assistance system that can be implemented to perform above mention use cases.

Case 1, Whenever a user asks a question to voice assistance system, then the system should be able to recognize the voice, extract the meaning, reply to the user accordingly.

Case 2, Whenever a user requests voice assistance system to play an online music streaming platform like YouTube or amazon prime, then the system should be able to play the music.

Case 3, When a user asks voice assistance system to set alarm at a particular time, then the voice assistance system should be able to do so and play alarm at set time.

Case 4, When a user asks about the weather information to voice assistance system then the voice assistance system should be able to pick the required information from internet and reply back user with the correct information.

#### Further Improvements

The voice assistance system could further be enhanced to play various predefined games. To make calls or send messages from a smart phone as requested by user, should be able to do online purchases as requested by user and also should be able to control other smart devices like lights, locks, thermostats, vacuum cleaners, switches.

#### Technical Requirements

This document addresses the requirements to capture human voice in real time voice and process them which is an essential part of a voice assistance system. A voice assistance system should essentially include a good quality audio input device like boya, audio output device like usb speaker and a high-speed edge computing device like raspberry pi.

#### Data Requirements

* Audio files with File format as WAV.
* Text files that will allow Kaldi to communicate above audio data.

#### Tools used

Python programming language and frameworks such as NumPy, Pandas, Pytorch, Kaldi, ESPnet are used to build the whole model.





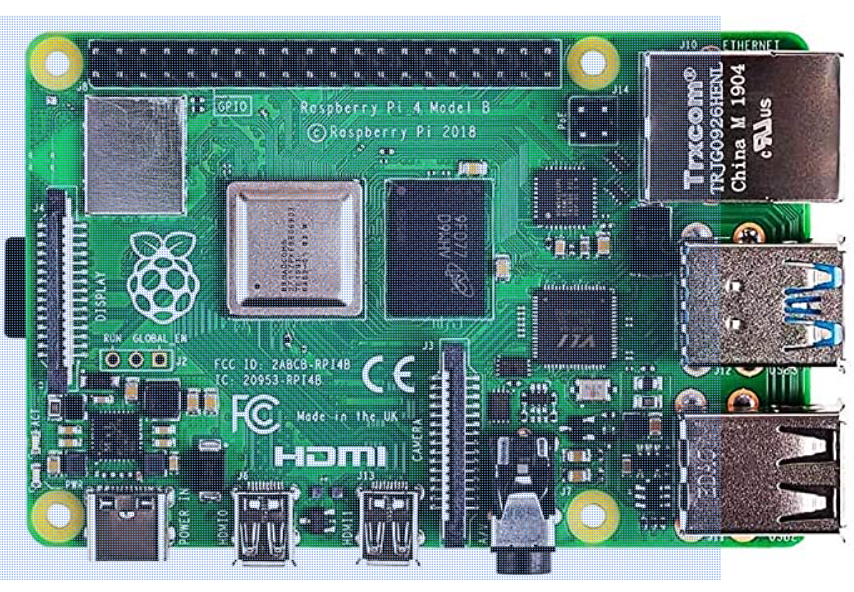




* + - Vscode is used as IDE.
    - For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
    - AWS is used for deployment of the model.
    - Tableau/Power BI is used for dashboard creation.
    - Cassandra is used to retrieve, insert, delete, and update the database.
    - Front end development is done using HTML/CSS
    - Python Flask is used for backend development.
    - GitHub is used as version control system.
    - Espnet for ASR and TTS

#### Hardware Requirements

* + - * GPU Enabled PC Environment
      * Raspberry Pi
      * Audio Input and output device.



#### Constraints

The voice assistant must be user friendly, as automated as possible and users should not be required to know any of the workings.

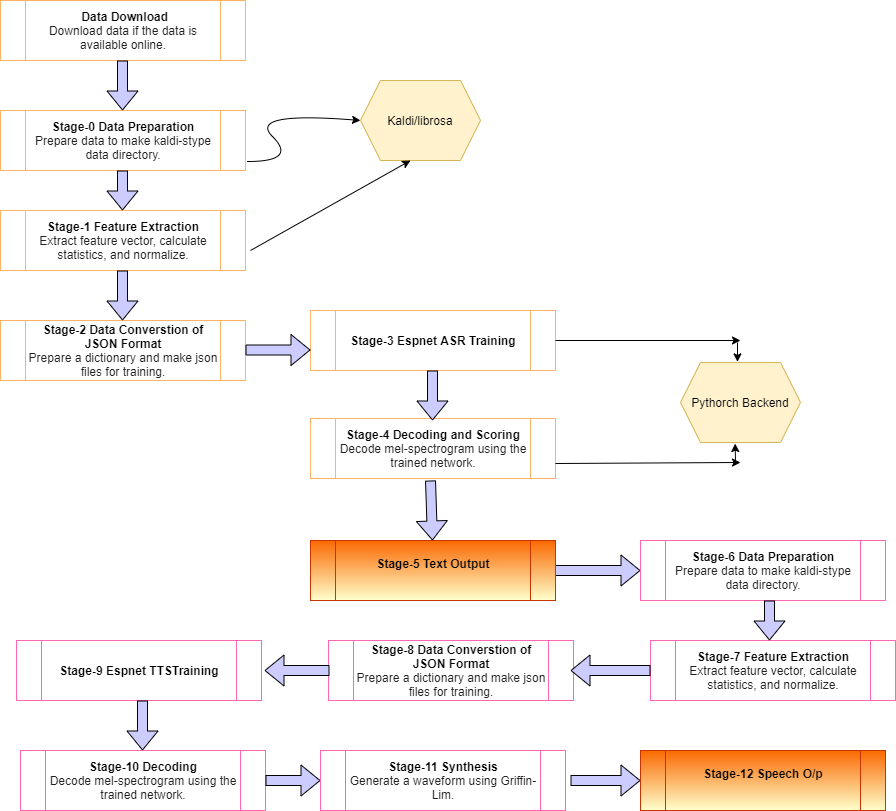
#### Assumptions

The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes through voice assistant which has mic installed for capturing the voice input. Deep Learning based NLP is used for detecting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

## Design Details

##### **Process Flow**

##### **Model Training and Evaluation**



##### 

##### **Deployment Process**

TO BE DECIDED

##### **Event log**

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

##### **Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

## **Performance**

Quality analysis with corpus data such as unwanted text, duplicate test, spam text analysis and noise reduction for the better performance

#### Reusability

The code written and the components used should have the ability to be reused with no problems.

#### Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

#### Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

#### Deployment



## **Dashboards**

Dashboards will be implemented to display and indicate certain KPls and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

#### KPls (Key Performance Indicators)

1. Good processing speed, accuracy and handling of special case like instant speech recognition.
2. Identification of the correct voice command.
3. Response should be based upon the context of speech.
4. Should be able to take the intended action based upon voice command.

## **Conclusion**

The designed voice assistant should be able to recognize user voice, process the question or request or command and should be able to take necessary action or reply back the user with response or further queries based on the use cases mentioned above. For example: voice assistance system should be able to play an online music player when requested, should be able to tell current location weather when asked, should be able to set an alarm when user asks to set alarm at a particular time etc.

## **References**

1. [espnet/espnet: End-to-End Speech Processing Toolkit (github.com)](https://github.com/espnet/espnet)