Low Level Design (LLD)

TEXT TO SPEECH

Revision Number: 1.8

Deepranjan

Document Version Control

Date Issued	Version	Description	Author
18th May 2020	1.1	First Draft	Amit K Gupta
20th May 2020	1.2	Added Workflow chart	Amit K Gupta
20th May 2020	1.3	Added Exception Scenarios Overall, Constraints	Vrunda Patel
21st May 2021	1.4	Added KPIs	Sukritha Joshi
26th May 2021	1.5	Added user I/O flowchart	Amit K Gupta
26th May 2021	1.6	Added FAC, VGG16 model diagrams	Nagesh
31st May 2021	1.7	Added dataset overview and updated user I/O flowchart.	Amit K Gupta
08th Sep 2021	1.8	Restructure and reformat LLD	Deepranjan

Contents

Abstract

The text-to-speech (TTS) is the process of converting words into a vocal audio form. The program, tool, or software takes an input text from the user, and using methods of natural language processing understands the linguistics of the language being used, and performs logical inference on the text. This processed text is passed into the next block where digital signal processing is performed on the processed text. Using many algorithms and transformations this processed text is finally converted into a speech format. This entire process involves the synthesizing of speech.

1 Introduction

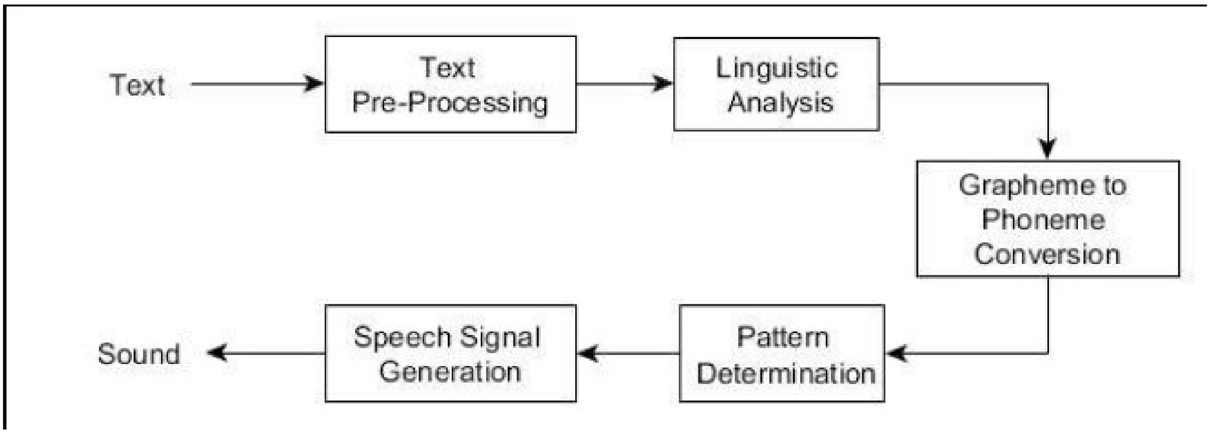
1.1 Why this Low-Level Design Document?

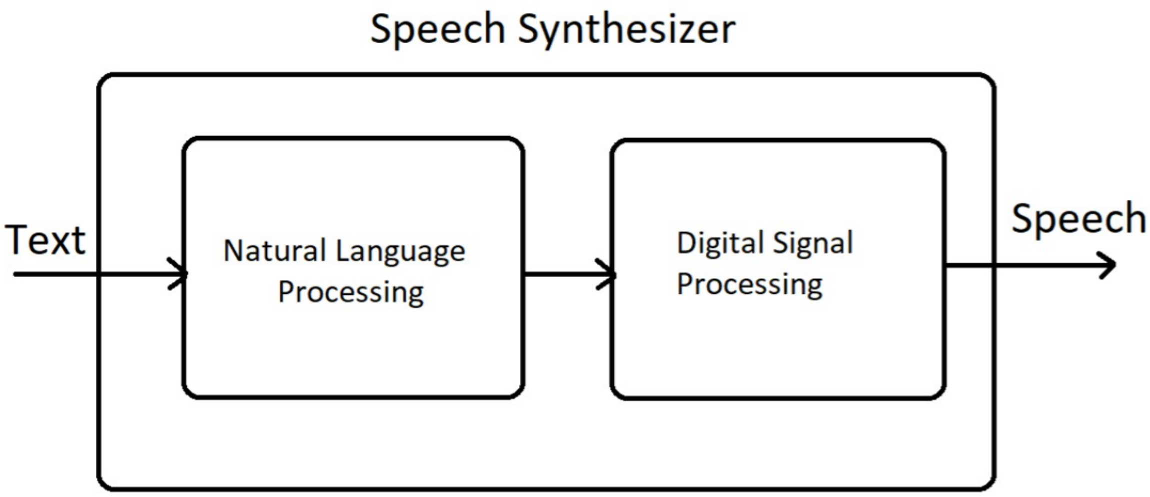
The purpose of this document is to present a detailed description of the Text to Speech . It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to take words on computers, smartphones, tablets and convert them into audio.

Text to Speech can convert the textual information into audio and it can :

- gTTS (Google Text-to-Speech) is a Python library and CLI tool to interface with Google Translate text-to-speech API .
- The text being passed is firstly preprocessed with the help of natural language processing, and then using digital signal processing is converted to speech.
- gTTS supports multiple languages.
- When ever textual information you will provide it will convert it into audio file.
- Allow access to convert into audio just by passing text.
- Automate the audio conversion problem.





This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

1.2 Scope

This software system will be a Web application This system will be designed to convert textual information into audio, improved interventions, and more efficient audio transformation in which you can get audio of different text so that it will be easy to understand.

1.3 Constraints

We will only be selecting few lines of text.

1.4 Risks

Document specific risks that have been identified or that should be considered.

1.5 Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

2 Technical specifications

2.1 Language Supported

- Afrikaans (South Africa)
- Arabic
- Bengali (India)
- Bulgarian (Bulgaria)
- Catalan (Spain)
- Chinese (Hong Kong)
- Czech (Czech Republic)
- Danish (Denmark)
- Dutch (Netherlands)
- English (Australia)
- English (India)
- English (United Kingdom)
- English (United States)
- Filipino (Philippines)
- Finnish (Finland)
- French (Canada)
- French (France)
- German (Germany)
- Gujarati (India)
- Hindi (India)
- Hungarian (Hungary)
- Icelandic (Iceland)
- Indonesian (Indonesia)
- Italian (Italy)
- Japanese (Japan)
- Kannada (India)
- Korean (South Korea)
- Latvian (Latvia)
- Malayalam (India)
- Mandarin (China)
- Norwegian (Norway)
- Polish (Poland)
- Portuguese (Brazil)
- Portuguese (Portugal)
- Romanian (Romania)
- Russian (Russia)
- Serbian (Cyrillic)
- Slovak (Slovakia)
- Spanish (Spain)
- Spanish (United States)
- Swedish (Sweden)
- Tamil (India)
- Telugu (India)
- Thai (Thailand)
- Turkish (Turkey)
- Ukrainian (Ukraine)
- Vietnamese (Vietnam)

2.2 Predicting

- The system will ask you for enter text.
- The User will write textual information.
- The system will try to extract features from text and it will convert it into audio files.

2.3 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User will write text.

2. The system stores each and every information given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

2.4 Logging

We should be able to log every activity done by the user.

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

2.5 Deployment

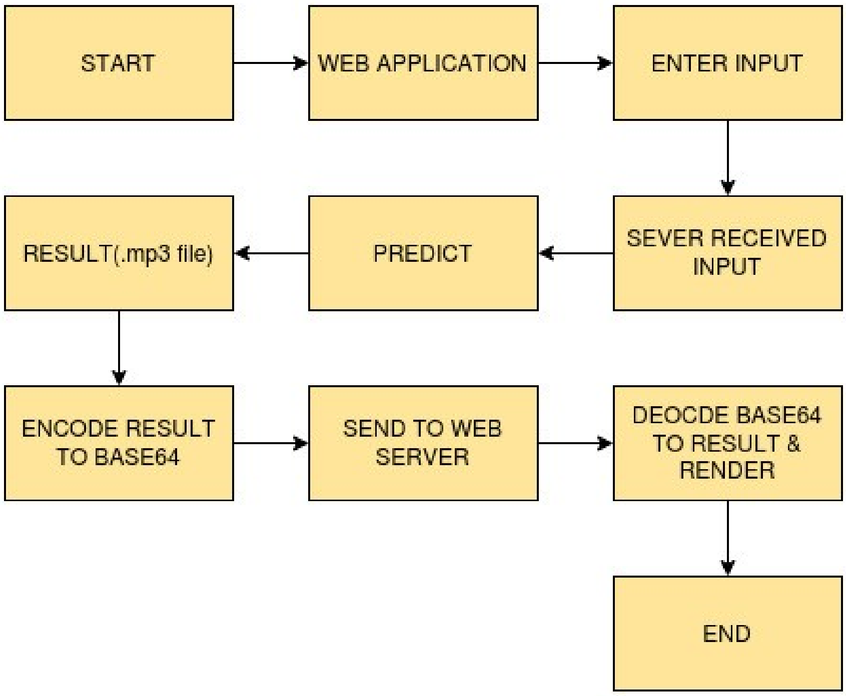
1. GCP

Google Cloud Platform

3 Technology stack

Front End	HTML/CSS/JS/React
Backend	Python Flask
Database	MongoDB/MySql
Deployment	GCP

4 Model training/validation workflow



5 Exceptional scenarios

Step	Exception	Mitigation	Module

5 Test cases

Test case	Epochs to train	Module	Pass/Fail

6 Key performance indicators (KPI)

- Time and workload reduction using the Text to Speech
- Accuracy of model prediction with time.
- Number of times you need to check the product.
- Just take textual information and give audio file.