**High Level Design (HLD)**

**PRESCRIPTION LABEL READING**



**Revision Number - 1.0**

**Last Date of Revision – 08-07-2022**

**SRIPHANI**

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 30-08-2023 | 1.0 | HLD Document | Prashant Kadam |
| 01-09-2023 | 1.1 | HLD Document | Praful Chalakh |
|  |  |  |  |

**Contents**

Document Version Control…………………………………………………………………………2

Abstract……………………………………………………………………………………………….3

1. Introduction……………………………………………………………………………………….4
   1. Why this High-Level Design Document?..............................................................4
   2. Scope……………………………………………………………………………………4
2. General Description……………………………………………………………………………...5
   1. Product Perspective & Problem Statement…………………………………………5
   2. Tools Used……………………………………………………………………………...5
3. Design Detail……………………………………………………………………………………...7
   1. Functional Architecture………………………………………………………………...7
   2. Optimization………………………………………………………………………….….8
4. Deployment……………………………………………………………………………………......9

# Abstract

* A Doctor's prescription is a handwritten document written by doctors in the form of instructions that describes list of drugs for patients in time sickness, injuries and other disability problems.
* While we receiving a new prescription from doctor, it is unable to understand what drug name is prescribed on it.
* In most cases, however, we wouldn't be able to read it anyway because doctors use Latin abbreviations and medical terminologies on prescriptions that are not understandable by the general persons which make reading it very difficult.
* According to the National Academy of Sciences estimates that at least 1.5 million peoples are sickened, injured or killed each year by errors while reading prescription.

# 1 Introduction

## 1.1 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 before 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

## 1.2 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.

# 2 General Description

## 2.1 Product Perspective & Problem Statement

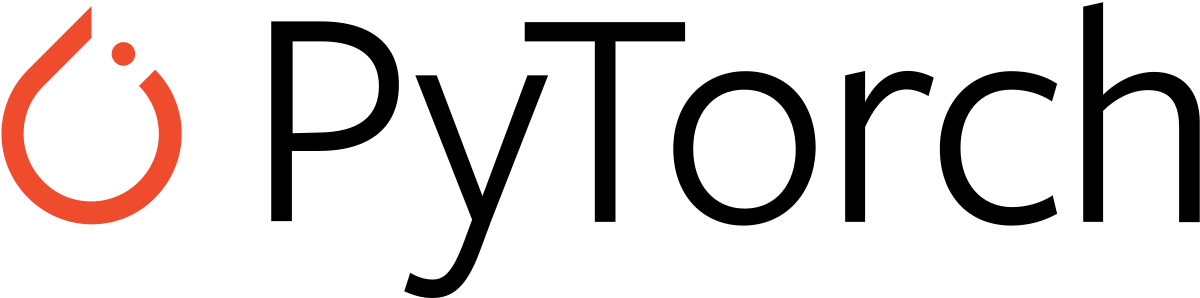
* Support elderly or vulnerable patients should be a focus for many businesses. This is especially true for those in the health care sector. Enabling voice messages can make it easier for elderly people to understand your message.
* Text-to-Speech can provide peace of mind by empowering you to give better services. For example, you could even send voice messages that read prescription labels.
* This can be a real challenge for anyone with reading difficulties, not to mention the elderly and visually impaired. A talking label, sent straight to your device, makes it easy to know everything about your medication. Dosage info can also be tracked and shared with caregivers.

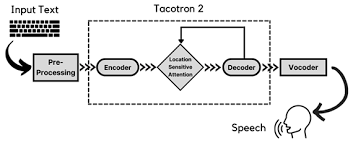
## 2.2 Tools used

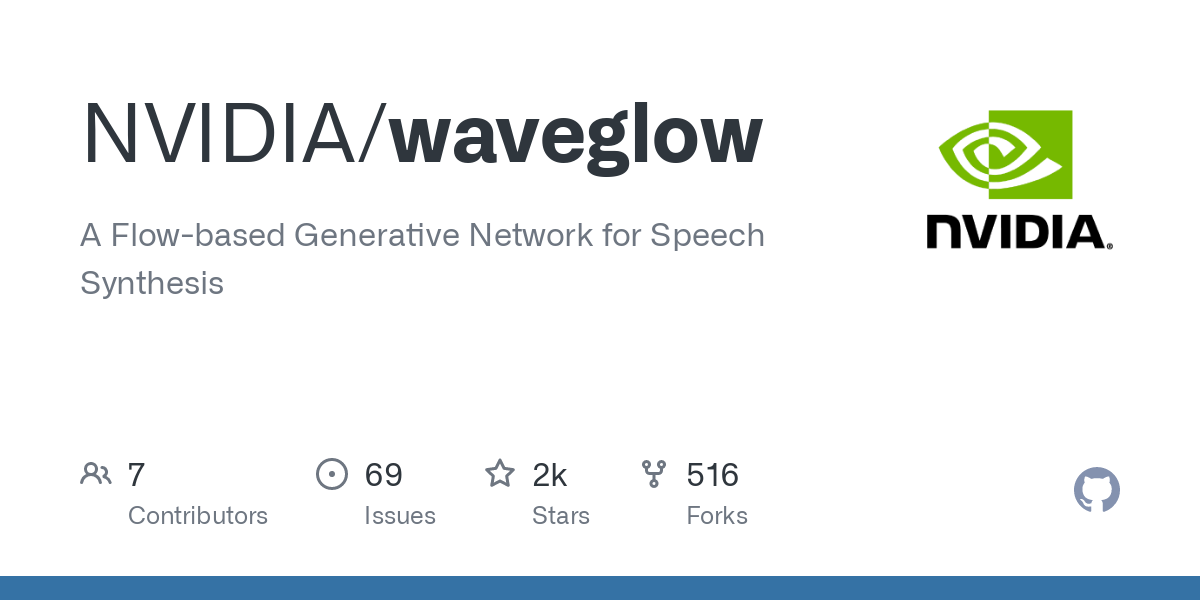
Business Intelligence tools and libraries works such as Tesseract OCR, Tocotron2 Model, PyTourch, Jupyter Notebook,Waveglow and Python Programming Language are used to build the whole framework.





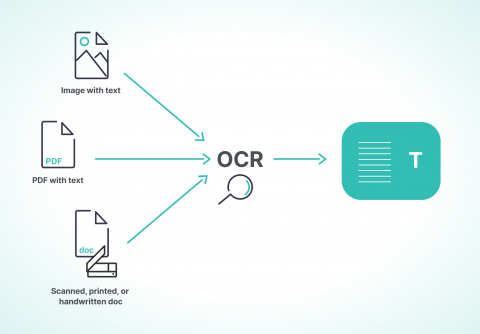
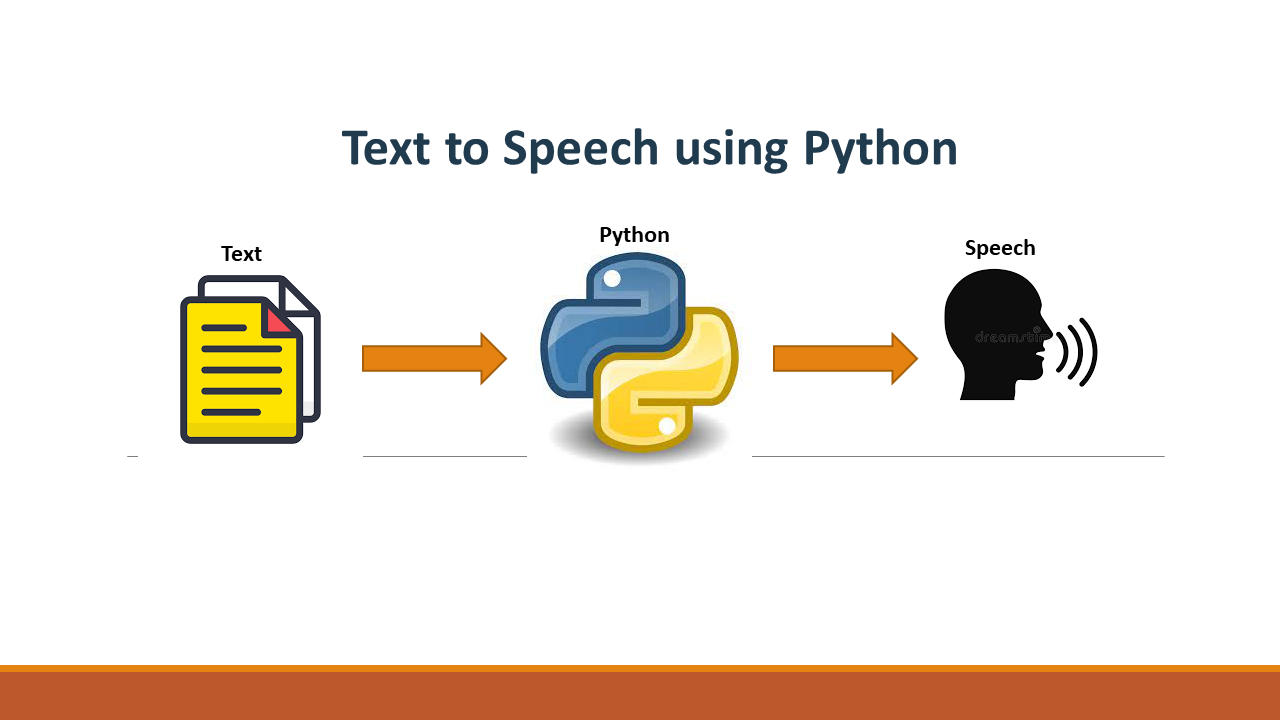






**3 Design Details**

## 3.1 Functional Architecture

** **

|  |  |
| --- | --- |
|  |  |

HOW MODEL WORKS:

STEP 1: upload an image.

STEP 2: Extracts Text Region.

STEP 3: Extracted text region undergoes text binarization and recognition.

STEP 4: Text recognition is performed by OCR.

STEP 5: Displaying label in the form of text.

STEP 6:Text is converted to speech .

## 3.2 Optimization

#### 1. Your data strategy drives performance

* Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views

#### 2. Reduce the marks (data points) in your view

* Practice guided analytics. There’s no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly granular views at the speed of thought.

* Remove unneeded dimensions from the detail shelf.

* Explore. Try displaying your data in different types of views.

# 4 Deployment

Deployment is done in the local system.