**High Level Design (HLD)**

**PRESCRIPTION LABEL READING**



**Revision Number - 1.0**

**Last Date of Revision – 03-09-2023**

# Document Version Control

| **Date Issued** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 03-09-2023 | 1.0 | HLD Document | Asif Ali  Hiren Thakkar  Khushwant Singh  Sanket Maheta  Shritama Sengupta  Shreya mohanty |

**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……………………………………………………………………………………...6
   1. Functional Architecture………………………………………………………………...6
   2. Optimization………………………………………………………………………….….7
4. KPI……………………………………………………………………………………………….…8
5. Deployment……………………………………………………………………………………......8

# Abstract

* A doctor's prescription is a handwritten document written in the form of instructions that specifies a list of medications for patients during times of illness, injury, or disability.
* While receiving a new prescription from the doctor, the system is unable to recognise the medication name.
* The majority of the time, however, we wouldn't be able to read it because physicians use Latin abbreviations and medical terms that are not understood by the general public, making reading it extremely difficult.
* The National Academy of Sciences estimates that at least 1.5 million people are sickened, injured, or murdered annually as a result of prescription reading errors.

# 1 Introduction

## 1.1 Why this High-Level Design Document?

This High-Level Design (HLD) Document is intended to add the necessary detail to the current project description in order to represent a suitable coding model. This document is also intended to aid in the detection of contradictions prior to coding and can be used as a reference for the high-level interactions between modules.

**The HLD will:**

* Present and define all design elements in detail. Describe the user interface being implemented. Describe the hardware and software interfaces.
* Describe the required performance
* Include the design and architecture of the undertaking.
* List and describe the nonfunctional characteristics, including:

-Security

-Reliability

-Maintainability

-Portability

-Reusability

-Application compatibility

-Resource utilization

-Serviceability

## 1.2 Scope

The HLD documentation describes the system's architecture, which includes the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD employs non-technical to lightly technical terms that system administrators ought to be able to comprehend.

# 2 General Description

## 2.1 Product Perspective & Problem Statement

* Supporting elderly or vulnerable patients should be a priority for numerous organisations. This is notably true for health care professionals. Enabling voice messages can facilitate elderly individuals' comprehension of your message.
* Text-to-Speech can give you peace of mind by enabling you to provide superior service. You could even send voice messages that read prescription labelling, for instance.
* This can be quite problematic for those with reading difficulties, let alone the elderly and visually impaired. A label that speaks and is sent directly to your device makes it simple to learn everything about your medication. Additionally, dosage data can be monitored and shared with carers.

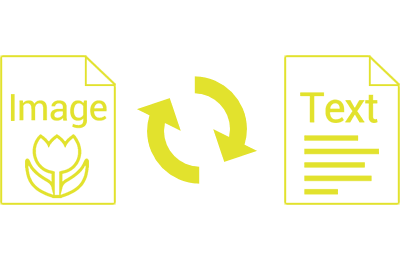
## 2.2 Tools used

The entire framework is constructed using Business Intelligence tools and libraries such as paddleOCR, paddlepaddle, gTT, google colab,pillow, and Python Programming Language.

1. PaddleOCR
2. PaddlePaddle
3. gTT
4. google colab
5. Pillow
6. Python

**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

* By materialising calculations, removing columns, and using accelerated views, optimise extracts to accelerate future queries.

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

* Employ guided analytics. There is no requirement to display everything in a single view. Compile related views and connect them with action filters to swiftly transition from overview to granular views.
* Remove dimensions from the detail shelf that are unnecessary.
* Exploration of. Consider displaying your data in various ways.

# 4 KPI

Output will be implemented to display and signify specific KPIs and pertinent image indicators.

As soon as the system begins to collect a user's data, the interfaces will be implemented to display time-series charts with progress on various indicators or factors.

# 5 Deployment

Deployment is done in the local system.