**PRESCRIPTION LABEL READING**

**High Level Design (HLD)**

**Revision Number - 1.1**

**Last Date of Revision – 01/01/2024**

**Mary Ara N.**

**Document Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** |  | **Version** | **Description** | **Author** |
| 12/22/2023 |  | 1.0 | First draft | Mary Ara N. |
| 01/01/2024 | 1.1 |  | HLD added | Mary Ara N. |

**Contents**

Document Version Control…………………………………………………………………………1

Abstract……………………………………………………………………………………………….2

1. Introduction……………………………………………………………………………………….3
   1. Why this High-Level Design Document?..............................................................3
   2. Scope……………………………………………………………………………………3
2. General Description……………………………………………………………………………...4
   1. Product Perspective ………………………..…………………………………………4
   2. Problem Statement………………………………………….…………………………4
   3. Proposed Solution……………………………………………………………………...4
   4. Tools Used……………………………………………………………………………...4
   5. Data Requirements …………………………………………………………………….6
   6. Assumptions ……………………………………………………………………………6
3. Design Detail……………………………………………………………………………………...7
   1. Functional Architecture………………………………………………………………...7
   2. How Model Works ……………………………………………………………………...9
   3. Optimization………………………………………………………………………….…. 9
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 receive 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

The Prescription Label Reading solution is an important tool that will help the elderly and visually impaired patients to easily get the necessary details about their prescriptions.

## 2.2 Problem Statement

People, like the elderly and visually impaired, with reading difficulties find it difficult to accurately read and get the right dosage and other information from their prescription labels. Supporting elderly or vulnerable patients should be a focus for any business in the health care sector.

## 2.3 Proposed Solution

To build a text-to-voice system that will create voice messages from the prescription label. This voice message can then be sent to the patient’s phone. Enabling voice messages can make it easier for elderly people to understand the doctor’s message. Here are the benefits of this solution:

* Text-to-voice can provide peace of mind by empowering the doctor to give better services. For example, you could even send voice messages that read prescription labels.
* For the patient, a talking label sent straight to his device makes it easy to know everything about his medication. Dosage info can also be tracked and shared with caregivers.

## 2.4 Tools used

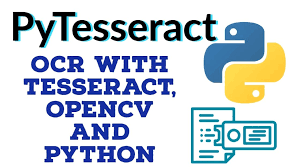
The programming language here that is used to build the whole framework is Python. The libraries and other business intelligence tools applied include Pytesseract OCR, OpenCV, Github, FastAPI, Pillow, Uvicorn.

Pycharm is used as the IDE.

FastAPI is used for local deployment.

Github is used as version control system.

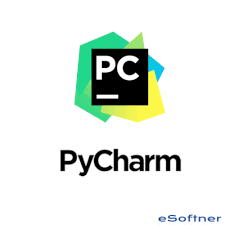
Postman is used for testing.



 A logo with orange circles and grey dots

Description automatically generated

## 2.5 Data Requirements

* This model only accepts images in .jpg format.A label must be saved in a .jpg format before being loaded into the model.
* The model takes in only one label at a time
* Only labels from Gateway Pharmacy, Edmonton, Alberta can be used with this model, since the Regex code is based on thieir labels. Any other kind of label will require a modification of the Regex code in order to fit its needs.

## 2.6 Assumptions

It is assumed that all prescription labels that will be used on this model will come from Gateway Pharmacy, Edmonton, Alberta.

It is also assumed that all aspects of this project have the ability to work together in the way the author is expecting.

# 3 Design Details

Saving output voice in local repo

Testing

Application start

Data selection

Data conversion: image to text

Local storage

Local deployment with FastAPI

text extraction

Data preprocessing

Data import

Start

## 3.1 Functional Architecture

Text to speech conversion

End



## 3.2 How Model Works

Step 1: upload an image.

Step 2: Preprocesses image

Step 3: Extracts Text from pre-processed image.

Step 4: Text recognition is performed by OCR.

Step 5: Relevant text is extracted using Regex.

Sep 6: Text is converted to speech and stored in local repository a .MP3 file.

## 3.3 Optimization

Built-in functions and libraries were used instead of reinventing the wheel to boost the performance of the Python code.

Data structures were optimized by using a dictionary where appropriate to optimize code execution. This is especially seen in the file, parser\_prescription.py.

Code refactoring was applied to reduce the size of the code, thereby improving it and making it easy to maintain. This is especially seen in the file, parser\_prescription.py

# 4 Deployment

Deployment is done in the local system using FastApI.