**PRESCRIPTION LABEL READING**

**Architecture Design**

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 |  | Added architecture | Mary Ara N. |

**2**

**Contents**

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

1. Introduction……………………………………………………………………………………….4
   1. Purpose of Architecture Design Document?....................................................................4
   2. Scope………………………………………………………………………………..……………4
2. Architecture and Data Flow……….…………………………………………………….……...5

# 1. Introduction

## 1.1 Purpose of the Architecture Design Document

The architecture design document (ADD) is a comprehensive description of the software architecture for a system. It outlines the high-level structure of the system, including its components or modules, their relationships, and how they interact to achieve the system's functionality and goals. The purpose of the architecture design document is to provide a blueprint that guides the development team throughout the implementation phase.

**1.2 Scope**

The scope of an architecture design document typically includes the following aspects:

1. **Introduction of the project:**

This project recognizes and identifies the text in the prescription labels and reads out the name of medicine, the dosage limits, the number of refills prescribed and the expiry date of the refills, to the visually impaired or elderly patients.

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.

1. **Key Components and Technology Stack:**

This model leverages a host of Python libraries and order in the following order:

Computer Vision’s adaptive thresholding to process the prescription label .jpg image.

Python's Tesseract OCR (Optical Character Recognition) to convert the image to text.

It then uses Regular Expressions (regex) to extract the relevant text.

It uses Google Cloud’s texttospeech module to convert the text to voice.

FastAPI is used for local deployment.

And finally, Postman for testing.

1. **Data Storage and Handling:**

No particular database was built for this project as the data consists of just one .jpg prescription label stored in the working repository. The reason behind this is that there is a variety of labels used by different hospitals, and the regex code will be different for each of them. This project only uses one example label and will work for any hospital that uses that example label.

1. **Error Handling and Logging:**

A manual exception is in-built to raise an exception each time an invalid file format is fed into the system.

Also, a custom exception, takes care of errors with the FastAPI app.

Once the app is run, running logs are registered in the logs repository.

1. **Deployment Architecture:**

Deployment is done locally using FastAPI and Uvicorn.

1. **Testing Strategy:**

Testing is done using Postman.

# 2. Architecture and Data Flow



## 

