DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SUMMER 2021

OPTICAL PROFILIERS
DOCUMENT CLASSIFIERS

XAVIER WELLS
KOSHISH KHADKA
BISHWAMITRA SAPKOTA
MUHAMMED DAUD

Revision History

Revision	Date	Author(s)	Description
0.1	6.28.2021	XW, KK, BS, MD	First Release
0.2	8.15.2021	BS	Final Release

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The Document Classifier system is a document identification solution. The system allows users to upload documents and classify them based on their type. Document Classifier will also store the results of classifications on database to be retrieved by privileged users later. So far, Document Classifier can recognize multiple pdf document classes and the new varieties of documents are yet to be added. Document Classifier in its currently planned state is not intended to be available publicly or commercially. Document Classifier is designed specifically for CSE 4317: Senior Design class sponsored by UTA. It could be of value for any corporation that has to process large volumes of electronically scanned documents. This Detailed Design Specification (DDS) document provides information about specific implementation details like libraries, frameworks, or any other dependencies on each subsystem. A detailed description of the overall structure of the system and data flow can be found on the Architectural design document. Likewise, details on requirements on this project can be found on the System Requirements Specifications(SRS) document.

2 SYSTEM OVERVIEW

The Document Classifier consists of two major components:- front end and a back end. A revision from identidoc, the back end and front end are distinctly separate and are not highly dependent on one another. For the front end, we have made a significant change in User Interface part.

A new back end concept has been implemented. The back-end uses a module called PDPDF2 . The module scans uploaded pdf documents and extracts the text from the pdf formatted file, which is used to search for key pair values to check the type of the document. Also AWS database has been used to store classified documents and files can be downloaded when needed.

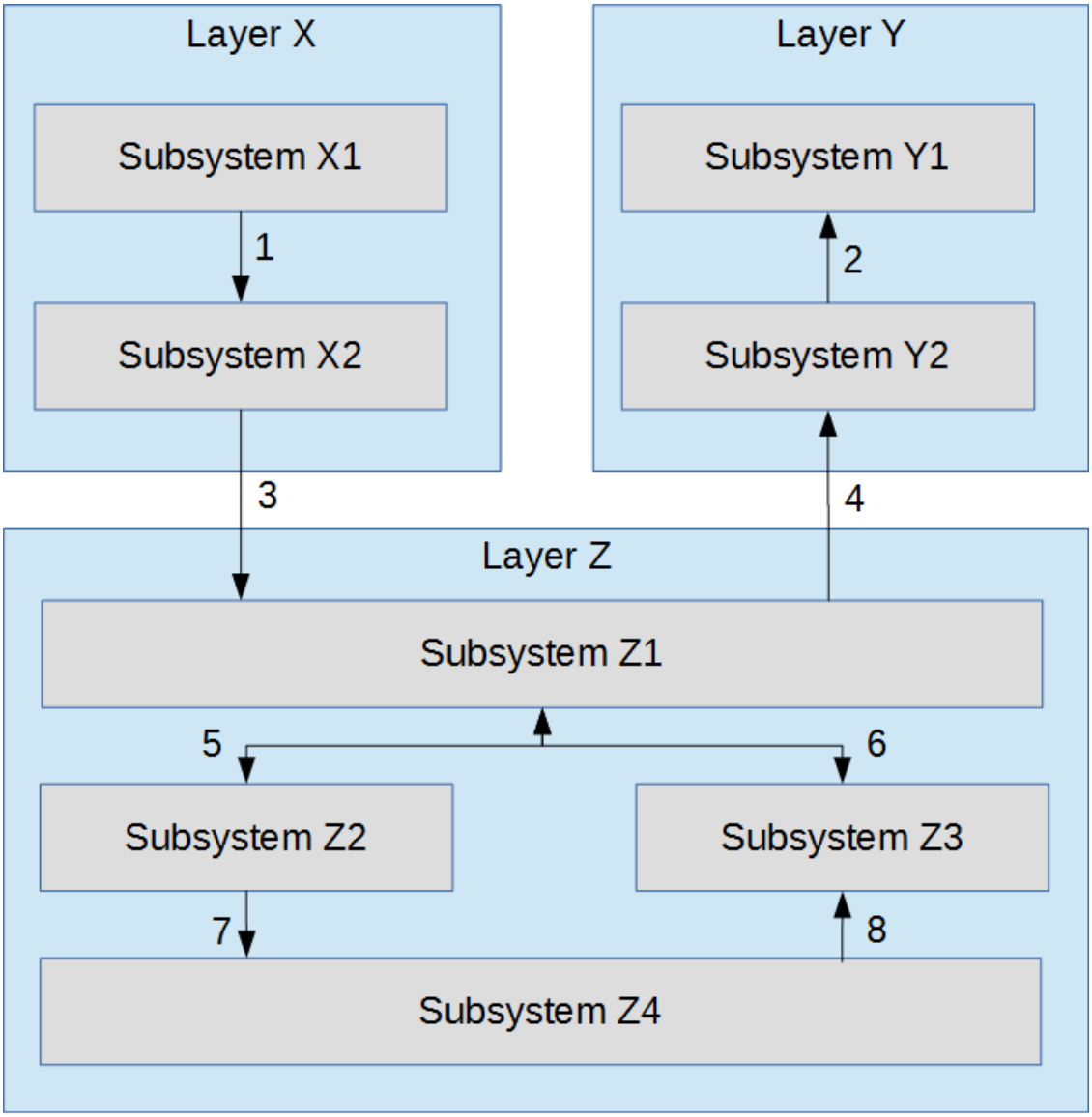


Figure 1: System architecture

3 Front End Subsystems

In this section, the layer is described in terms software design of the front end alone. What the purpose of the front end is and what it accomplishes.

3.1 Layer Operating System

The Front End layer is primarily developed and hosted using Ubuntu 20.04. Since identiDoc is a web based service, it should work on any other operating systems.

3.2 LAYER SOFTWARE DEPENDENCIES

Bootstrap 4.4.1, Data Tables 1.10.23, jQuery 1.11.3, pdf.js 2.11.9, popper.js, heic2any

3.3 User Interface

The User Interface allows the user to interact with the document classifier system through a web browser. Through the User Interface, the user will be able to upload a file for classification and also retrieve previous classification based on classification data,uploaded document type including voting registration documents.

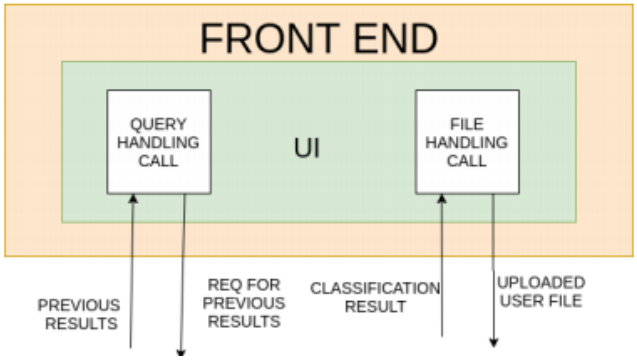


Figure 2: Front End subsystem Example Diagram

3.3.1 Subsystem Hardware

No special hardware needed

3.3.2 Subsystem Operating System

It is compatible with Windows, Mac OS X, and Linux

3.3.3 Subsystem Programming Languages

JavaScript, Flash with python, HTML

3.3.4 Subsystem Data Structures

Does not apply

3.3.5 Subsystem Data Processing

The file handling call performs HTTP POST request with appended file to upload and the query handling call performs HTTP GET request with selected data and document class

4 Back End Subsystems

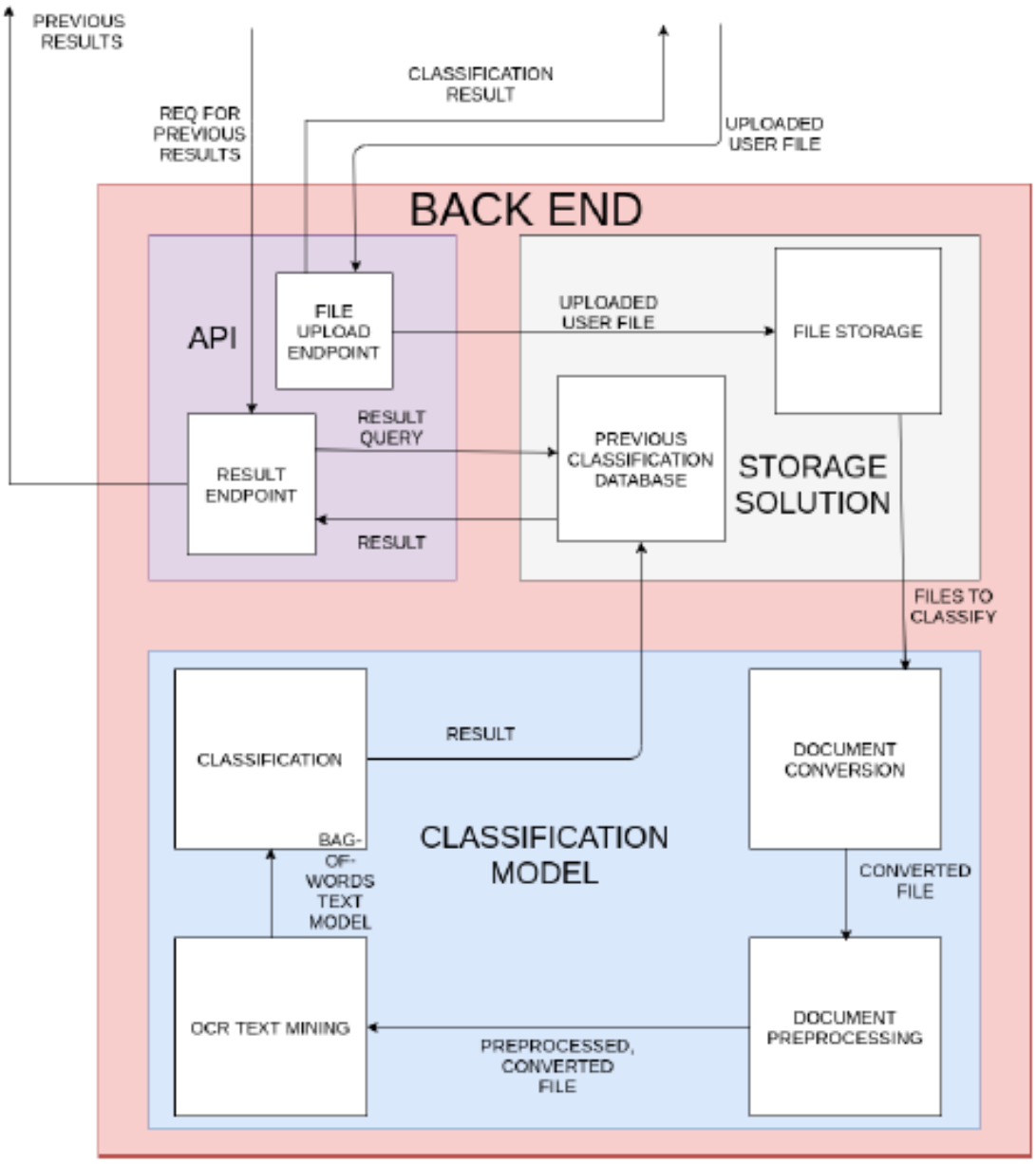
4.1 Layer Hardware

Not applicable.

4.2 Layer Operating System

Linux-based OS.

4.3 Layer Software Dependencies



4.4 API

The API serves as the access point of the application. The API handles all requests from the Front End of the identiDoc system. The user, through different API calls, will be able to upload files for classification and query for previous classification results.

4.4.1 Subsystem Hardware

Not applicable.

4.4.2 Subsystem Operating System

Linux-based OS.

4.4.3 Subsystem Software Dependencies

This sub-system uses python web framework Flask to create a API.

4.4.4 Subsystem Programming Languages

Python3

4.4.5 Subsystem Data Structures

Not applicable

4.4.6 Subsystem Data Processing

Not applicable

4.5 Storage Solution

The Storage Solution stores all static data needed for the identiDoc system. This includes aws database of records of classification results, and the actual files that were classified.

4.5.1 Subsystem Hardware

Not applicable.

4.5.2 Subsystem Operating System

Linux-based OS.

4.5.3 Subsystem Software Dependencies

4.5.4 Subsystem Programming Languages

Python3

4.5.5 Subsystem Data Structures

Not Applicable

4.5.6 Subsystem Data Processing

Not applicable.

4.6 Classification Model

The classification model is responsible for classifying documents. The text can then be classified using a similarity score for classification. A similarity threshold will be set to deal with unrecognized documents. The classification model will pass off the resulting classification result to the storage solution.

4.6.1 Subsystem Hardware

Not applicable.

4.6.2 Subsystem Operating System

Linux based OS.

4.6.3 Subsystem Software Dependencies

System Uses the module called PDPDF2 to scan pdf documents and perform OCR(Optical character reading). The pdf formatted file is used to search key pair values to check the type of the document.

4.6.4 Subsystem Programming Languages

Pyhton3

4.6.5 Subsystem Data Structures

Not applicable

4.6.6 Subsystem Data Processing

The user uploaded file is the input for the system, which it receives from the Storage Solution subsystem.The input file goes through document conversion,document pre-processing and the OCR text mining units and finally is passed to the classification unit.The classification unit uses the pre-trained model to predict the label of the file passed and passes the predicted label to the storage solution

5 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES