Software Requirements Specification

for

Photo Metadata Extractor Tool

Version 1.0 approved

Prepared by Bryan Portillo, Evan Putnam, Kay Vargas

Cal Poly Humboldt

09/23/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft	9/13/23	Initial Document	1.0
Revised	9/20/23	Made fonts and sizes more consistent. Updated document title.	1.1
Revised	10/01/23	Updating information and adding specifications	1.2

1. Introduction

1.1 Purpose

This documentation serves to help guide any users of the photo metadata extractor software which is described in detail. This software is designed to identify physical book and text information such as Superintendent of Documents (SuDoc) numbers, title, and publication data through the analysis of inputted images of the front page of the desired text. The software then returns the collected data to the user. Revision 1.0.0

1.2 Document Conventions

Throughout this text the word document will be used to describe various forms of written literature ranging from books, manuscripts, news articles, etc. The software title, Photo Metadata Extractor Tool (PMET), will be used throughout this documentation to describe the entire entity of the software which is being developed and described. The graphical user interface (GUI) will be used to describe the visual representation of the software which the user will interact with.

1.3 Intended Audience and Reading Suggestions

This documentation is intended for the users, testers, and developers of the PMET software. The first section lays out a fundamental description of the project along with information that will allow the readers to better follow the documentation. Section two has a more in depth description of the software itself. Section 3 addresses the specific interfaces of the software. Section 4 goes into the in depth process of the three main project features. Section 5 details more static aspects of the project such as security. Any remaining details and a glossary are included in chapter 6. Reading through the sections lineary is suggested however after reading the first two sections the reader will be able to understand any of the following sections. Testers may benefit the most from sections 3,4,5 while users of the software would likely benefit most from sections 1,2, and 4.

1.4 Product Scope

PMET is a tool intended to reduce the time spent cataloging and creating new records of physical documents. The goal of the tool is to automatically create a record entry from a photo of the cover and or title page, either through extracting the SuDoc and querying existing databases or extracting all relevant text from the photo(s) provided.

1.5 References

TBD(1)

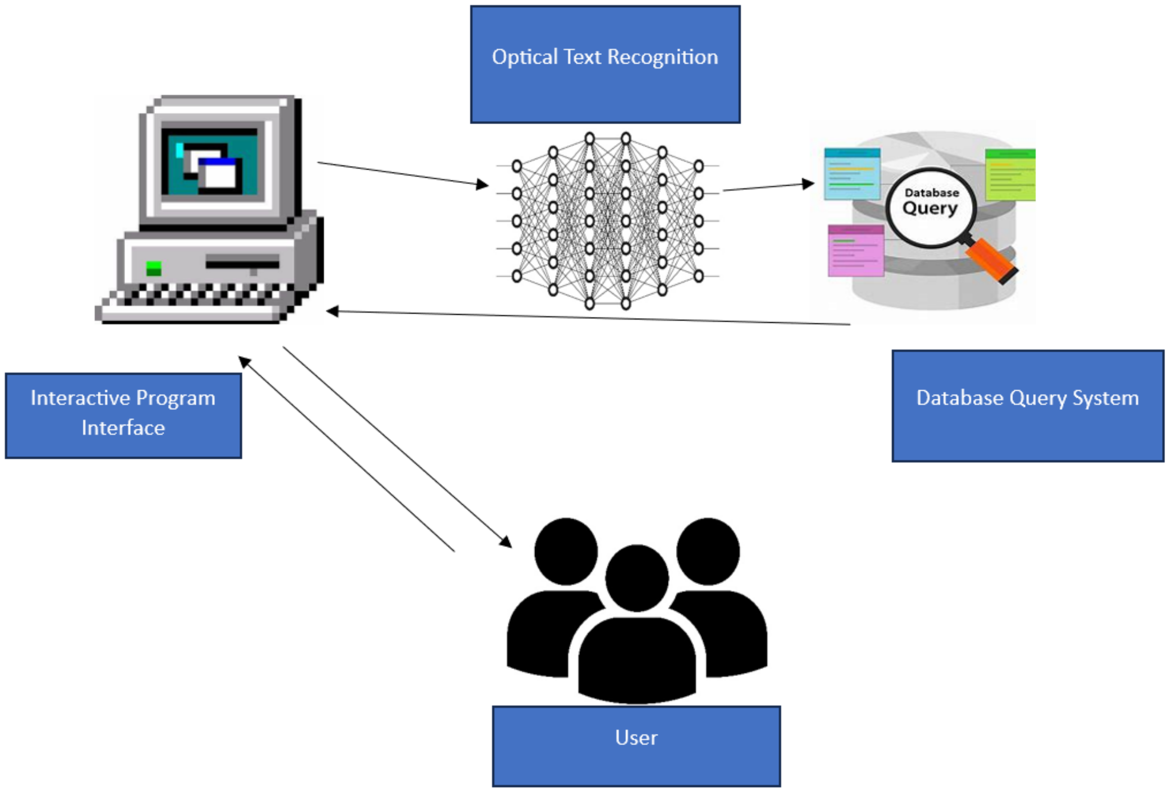
2. Overall Description

2.1 Product Perspective

This self-contained product is designed to extract data from uploaded images of documents to be used to search and match pre-existing database entries. In particular, this tool is intended to extract metadata from photos of the title page of a given government document and use captured data to search library resources for existing records.

2.2 Product Functions

I. Text Recognition Model
II. Interactive Program Interface
III. Database Query System



2.3 User Classes and Characteristics

The sole user class for this product is librarians or any individual who needs to keep track of large amounts of government documents that may contain a combination of typed and handwritten identifying information. No technical expertise will be needed for any user of this program. There will be no distinction between privilege or security for any users as all individuals will have equal access and accessibility to the product features.

2.4 Operating Environment

This software is intended to run on windows or mac OS and is designed solely for computers. The program will run in python.

2.5 Design and Implementation Constraints

TBD(2)

2.6 User Documentation

TBD(3)

2.7 Assumptions and Dependencies

TBD(4)

3. External Interface Requirements

3.1 User Interfaces

TBD(5)

3.2 Hardware Interfaces

TBD(6)

3.3 Software Interfaces

The PMET takes images as inputs in the interactive interface which are then fed into the optical text recognition tool for analysis. After the analysis the derived information will be sent to the database query system which will then conduct a query. The results of the query are then written to a file and sent back to the interactive interface where the user can then download the file. The database query system will be working directly with one of the following application programmer interfaces (API): HathiTrust, historic shipping lists, or Internet Archive.

3.4 Communications Interfaces

TBD(7)

4. System Features

4.1 System Feature 1

Optical Text Recognition

4.1.1 Description and Priority

This machine learning model will work to identify and characterize text through inputted images of documents. This text will then be analyzed and ordered in a manner such that the documents SuDoc can be recognized and then relayed to the other systems of this software. This feature is a high priority as all other features are dependent on this one's functionality.

4.1.2 Stimulus/Response Sequences

The user will input text through the interactive program interface. This will then activate the optical text recognition process where the inputted data will then be reformatted into an acceptable format for the machine learning model. The model will then output the corresponding extracted text which will then be fed through the parsing algorithm and fed to the database query system.

4.1.3 Functional Requirements

O_REQ-1: Inputted user data must be formatted into an acceptable format for the machine learning model to be able to successfully process the data.

O_REQ-2: The underlying model must be able to identify and extract image text from the images inputted by the user.

O_REQ-3: The identifying text must be parsed text into recognizable categories such as SuDoc, title, publish date, etc….

O_REQ-4: If the data inputted by the user is not acceptable for the model the users will be notified and the data will not be processed.

4.1.4 Data Design

-

4.2 System Feature 2

Database Query System

4.2.1 Description and Priority

This query system will access the needed database (TBD)(8) and retrieve data points on text information as specified by the customer (TBD)(9). This system has a high priority for the overall project as it collects the needed data for the desired output thus its overall functionality is mandatory for successful project completion. Overall the risk associated with this system is low as the data accessed is publicly accessible and no data will be saved or retained throughout the process.

4.2.2 Stimulus/Response Sequences

When the user uploads an image which is then processed by the optical text recognition feature the resulting text will then be used to query the system.

- If the query fails the user will be notified and the database query system will automatically populate the data fields for the desired data while notifying the user of the potential error and requesting validation of all guessed fields. The then verified data will be written to a text file and sent to the interactive program interface for user download
- If the query succeeds the collected data will be written to a text file and sent to the interactive program interface for user download

4.2.3 Functional Requirements

D_REQ-1: The database query system will be able to connect to the necessary API (TBD)(10).

D_REQ-2: The query system will be able to use the SuDoc to find the corresponding catalog entry in the system for the given document.

D_REQ-3: The query system will extract specific data points from the query and write them to a secondary file.

D_REQ-4: The system will notify user in the case the query fails even after the user has verified the SuDoc and automatically populate data fields with anticipated data based on the machine learning model text parsing.

D_REQ-5: The system will ping the user for verification in the case of a failed query. The user will be able to verify the SuDoc and re-initiate the program to query the database again with the verified SuDoc in the case of a correction.

D_REQ-6: Data must be formatted into an easily readable file and return the file to the interactive program interface

4.2.4 Data Design

4.3 System Feature 3

Interactive program interface

4.3.1 Description and Priority

This feature is designed to allow the user to interact with the underlying machine learning model and access information within the database. It represents a medium priority within the overall project as it adds the benefit of accessibility for the intended user. The highest priority within the GUI is the functionality with the graphical design having the lowest significance. The GUI is a low-risk feature as it will run locally and interact with no critical data. The benefit offered by the GUI itself is significant as it encompasses the usability of the software itself.

4.3.2 Stimulus/Response Sequences

The user will upload either an image file in large batches or individually which will then prompt the program to locally save and classify the files.

The user will be able to begin the image processing by selecting start.

The user will be able to interactively fill out any missing fields as prompted by the system during the query process.

The user will be prompted and easily able to verify the correct classification in the case of any uncertainty throughout the classification and query process.

The user will then be able to download the resulting classification of inputted images by simply downloading the output file.

4.3.3 Functional Requirements

I_REQ-1: The IPI will offer a input field which allow the user to load batches or individual document images which can then be processed by the program.

I_REQ-2: The IPI will be able to accept images in many formats

I_REQ-2:The IPI will Output the results of searching the database in a formatted file which is easily accessible to the user and in the specified form csv with labeled data points.

I_REQ-3:When a database search is unsuccessful the IPI will notify the user and ping them to verify the recognized SuDoc from the text. In the case the search fails after verification the user will then be notified and prompted to manually fill in the needed data fields.

I_REQ-4: The user will be easily able to download the resulting data at the end of the entire process

4.4.4 Data Design

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The tool in optimal circumstances should be able to create a data entry in under fifteen seconds. In a failed query search a data entry should take no more than a single minute however failed query should be limited to less than 10%. With these requirements the program will be able to optimally support its users in the task of document classification while minimizing the potential for incorrectly classified documents.

5.2 Safety Requirements

The tool will display confidence in its extracted text and possibly request the user to verify the extracted text, i.e. display a section of the photo next to the extracted text and ask if it matches. This will minimize the risk of outputting incorrect document information to the users of the program.

5.3 Security Requirements

The tool will create a unique folder to ensure that no other files are overwritten when creating a CSV to store the generated entries. The resulting CSV will be deleted upon a complete cycle of the program along with the inputted images in order to prevent memory misuse and uphold data security.

5.4 Software Quality Attributes

The IPI will be easy to read and simple so that it is optimally accessible to its users. The tool should include tooltip explanations for buttons, fields, and other inputs that allow for ease of use. The program will be reliable as it will be verified to be correct at a minimum of 90% of the time to minimize the need for user intervention or incorrect classification. The program will be flexible as it will allow inputted text images to range over several formats (jpg, png, etc) and in singular or large quantities. The program will be effective in being able to process up to 100 images in one batch.

5.5 Business Rules

BR1: The program will be efficient and decrease text processing time for its users.

BR2:The program will be easy to use and widely accessible to those who access it.

BR3: The program will notify the user in the case of system failures or uncertainties to minimize incorrect classification

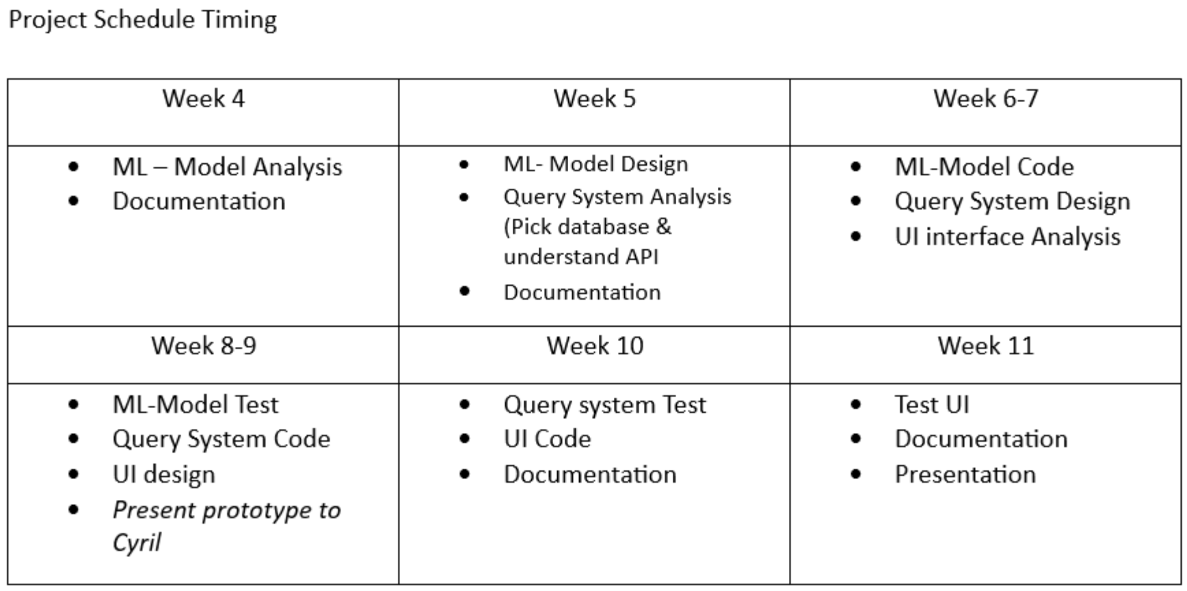
6. Other Requirements

Appendix A: Glossary

Team Roles

- **Bryan Portillo - Lead Designer & Quality Assurance Leader(Testing)**
- **Evan Putnam - Lead Programmer & Quality Assurance (Testing)**
- **Kay Vargas - Analyst/ Requirements Engineer & Quality Assurance (Testing) & Team Leader**

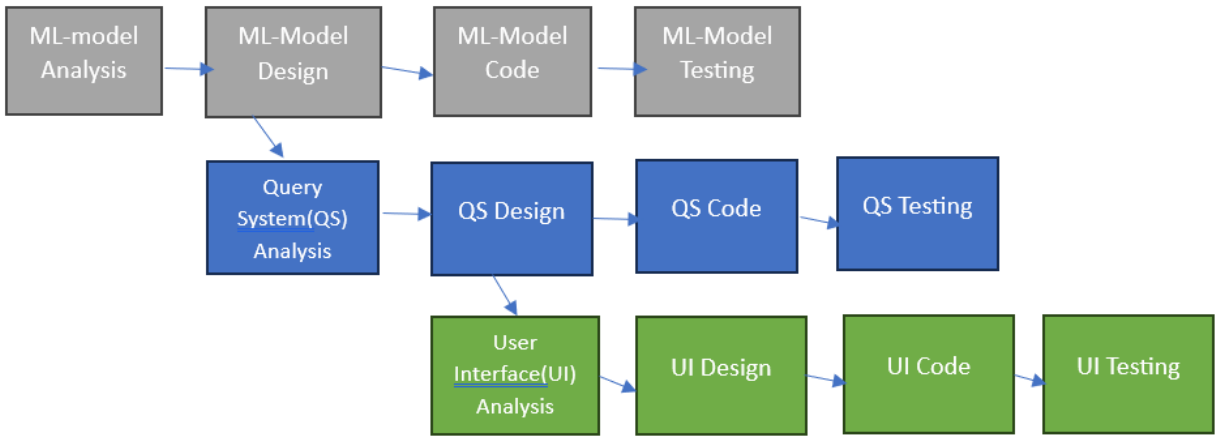
Appendix B: Analysis Models



Project Schedule Timing

Week 4	Week 5	Week 6-7
• ML – Model Analysis • Documentation	• ML- Model Design • Query System Analysis (Pick database & understand API • Documentation	• ML-Model Code • Query System Design • UI interface Analysis
Week 8-9	Week 10	Week 11
• ML-Model Test • Query System Code • UI design • *Present prototype to Cyril*	• Query system Test • UI Code • Documentation	• Test UI • Documentation • Presentation

Software Process Model (Incremental)



Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

1. *TBD - References*
2. *TBD - Design and Implementation Constraints*
3. *TBD - User documentation*
4. *TBD - Assumptions and Dependencies*
5. *TBD - Operating environment*
6. *TBD - User Interfaces*
7. *TBD - Hardware requirements*
8. *TBD - Communication Interfaces*
9. *TBD - Database which will be used to pull data*
10. *TBD - Datapoints pulled from specified database*
11. *TBD - API to be connected to*
12. *TBD - Communication Interfaces*