The Crash Test Dummies present…

Software Requirements Specification

for

Recovering Early Hollywood

Client: Samuel Backer

Prepared by Aspyn Call, Chimezie Ugbuaja, Jimmy Ocaya, Gabe Fitzpatrick, and Michael Wilkinson

University of Maine

October 10, 2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirements for an online database of copyright documents for films in early Hollywood. This database aims to provide a comprehensive repository of copyright information that allows researchers, filmmakers, historians, and enthusiasts alike to easily locate and analyze these documents.

1.2 Project Scope

The software being specified is a database management system designed to catalog copyright documents for films produced in early Hollywood. This platform will serve as a repository for copyright information, including registrations and renewals. The software will use Optical Character Recognition (OCR) to develop transcripts of the documents to better preserve them for the future and be more accessible to those who use screen readers. Above all, we hope to make something accessible and easy to use for those interested in early Hollywood.

1.3 References

What is OCR? - Optical Character Recognition Explained - AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/ocr/

2. Description

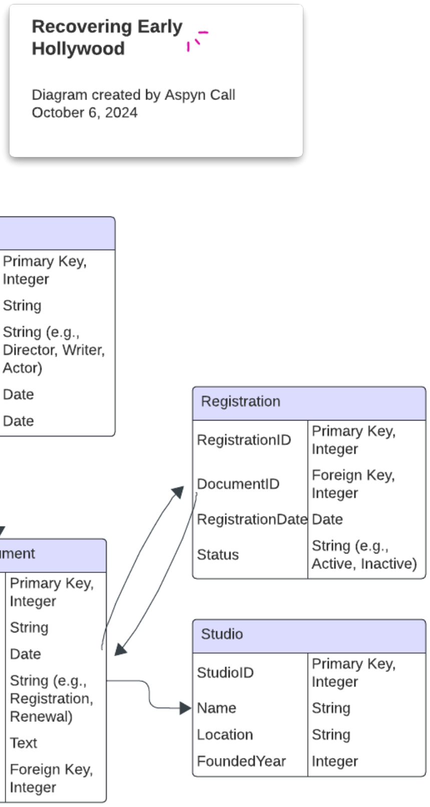
2.1 Product Perspective

The early Hollywood database system stores the following information:

- **Digitized Version of Documents:** It will include a digitized version of each document in PDF format.
- **Transcript of the Document:** Each entry will contain a transcript of the document that is generated by OCR. This is for better readability.
- **Description of the Document:** A description of the document, including the film's title, a summary of the film's synopsis, and the names of the cast and crew.

2.2 Product Features

The major features of the product are shown in the entity-relationship diagram below:



Recovering Early Hollywood

Diagram created by Aspyn Call
October 6, 2024

Creator	
CreatorID	Primary Key, Integer
Name	String
Role	String (e.g., Director, Writer, Actor)
BirthDate	Date
DeathDate	Date

Registration	
RegistrationID	Primary Key, Integer
DocumentID	Foreign Key, Integer
RegistrationDate	Date
Status	String (e.g., Active, Inactive)

Copyright Document	
DocumentID	Primary Key, Integer
FilmTitle	String
DateRegistered	Date
DocumentType	String (e.g., Registration, Renewal)
OCRTranscript	Text
CreatorID	Foreign Key, Integer

Studio	
StudioID	Primary Key, Integer
Name	String
Location	String
FoundedYear	Integer

2.3 User Stories

1. As a film historian, I want to search for specific copyright documents by title or director, so that I can easily access and analyze copyright information related to specific films or filmmakers.
2. As a researcher, I want to filter search results by year of release or registration, so that I can narrow down my research to specific time periods and understand the evolution of copyright in early Hollywood.
3. As an archivist, I want to upload scanned documents to the system using OCR technology, so that I can convert physical documents into digital, preserving the information for future access.
4. As a legal scholar, I want to download copyright documents in various formats, so that I can easily reference them in my academic work without any formatting issues.

2.4 Operating Environment

The operating environment for Recovering Early Hollywood is as follows:

- Database management system
- Client/server system
- OCR software
- Operating systems: Windows, Mac
- Database: MySQL
- Programming Language: Python

2.5 Design and Implementation Constraints

Optical Character Recognition Accuracy: The accuracy of OCR technology can vary based on the quality of the scanned documents. Ensuring high accuracy for historical documents with varying fonts and conditions can be challenging.

Integration with Existing Systems: Integrating with existing databases can pose compatibility issues.

Scalability: The system must handle a large volume of data. Performance may degrade if not designed to scale effectively.

Response Time: The large volume of data can also affect response time for users. Optimization will be needed for faster loading times.

2.6 Assumption Dependencies

Assumption: The chosen OCR technology will reliably convert scanned documents into machine-readable text.
Dependency: The accuracy of the document and OCR processing feature are reliant on the performance of the OCR technology.

Assumption: The scanned documents uploaded to the system will be of sufficient quality for OCR processing.
Dependency: High-quality documents are essential for accurate text interpretation and transcription.

Assumption: The system will use a reliable cloud storage solution for documents.
Dependency: The performance and accessibility of the stored documents depend on the cloud service provider's reliability.

3. System Features

3.1 Document Database

3.1.1 Description and Priority

This feature allows users to view the documents uploaded to the database. This feature is of high priority because it is the most essential feature for the software.

3.1.2 Stimulus/Response Sequences

1. The user selects a document to view.
    a. The system displays the document as well as its description.
2. The user clicks a button to show the full transcript of the document.
    a. The system shows the full transcript of the document on the screen.

3.1.3 Functional Requirements

REQ-1: The system shall allow users to select a document to view.
REQ-2: The system shall allow users to see a full description of the document.
REQ-3: The system shall allow users to read a transcript of the document in plain text.

3.2 Search and Filtering

3.2.1 Description and Priority

This feature enables users to perform searches of the database for documents based on a number of criteria, including title, director, year of publication, and document type. This feature is of high priority, as it makes finding documents easier and allows research to be done quickly.

3.2.2 Stimulus/Response Sequences

1. The user navigates to the search interface from the main dashboard.
    a. The system displays the search options with a text input field and button filters.
2. The user enters search criteria and/or filtering options.
    a. The system accepts the input and activates the “Search” button.
3. The user clicks the “Search” button.
    a. The system processes the search and retrieves relevant documents from the database.
4. The user selects a document from the search results.
    a. The system displays the document details, including metadata and transcript.

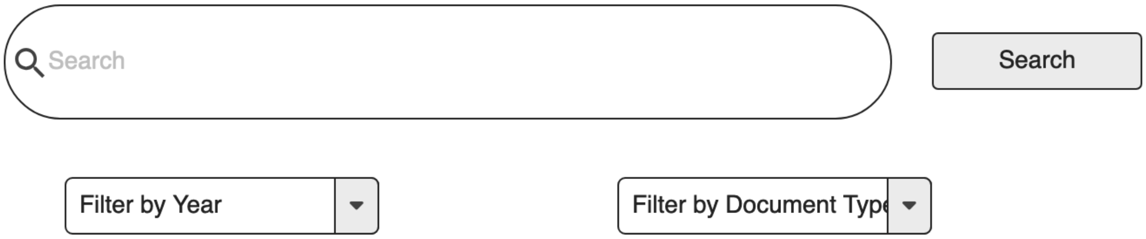
3.2.3 Functional Requirements

REQ-1: The system shall provide a search interface with filters for title, director, year of release, and document type.
REQ-2: The system shall allow users to combine multiple search criteria to refine results.
REQ-3: The system shall return search results that match the specified criteria.
REQ-4: The system shall provide clear error messages if no results are found based on search criteria.
REQ-5: The system shall ensure that the search functionality is responsive and performs efficiently, even with a large dataset.

4. External Interface Requirements

4.1 User Interfaces

The end user is expected to be able to interface with the database by using a GUI. The user will be expected to have a keyboard and mouse in order to specify which documents are to be retrieved and to filter documents by type, year published, and other specifications. Each screen will feature a search bar and corresponding button as a standard for the user to submit their specifications. Screens should fit a standard 16x9 ratio for the software.



4.2 Hardware Interfaces

4.2.1 Supported Device Types

- Desktops and laptops
    - Operating Systems: Windows, macOS
    - Browsers: Chrome, Safari and Edge

4.2.2 Nature of data controls and Retrieval

1. Data Retrieval

    a. A request is sent to a centralized database server using HTTP protocols to collect copyright documents based on the queries provided by the user.
2. User Input
    a. Input data is collected through forms and transmitted to the server for processing.
3. Communication Protocols
    a. HTTP - used to exchange data in a secure manner between the client and server.
    b. RESTful API - used for data retrieval.

4.2.3 Server Infrastructure

1. Database Server; A server to store and manage copyright documents and metadata.

4.2.4 User Device Specifications

1. Desktops and Laptops
    a. Dual-core processor of 4GB RAM as minimum Requirement.

4.2.5 Network

1. Users are required to have a stable internet connection.

4.3 Software Interfaces

4.3.1 Description

This section describes the relationship between the database and other software components like the operating systems, tools, libraries and other integrated components.

4.3.2 Software components

1. Database
    a. PostgreSQL, version: latest
2. Operating system
    a. Ubuntu server, version: latest
3. Tools and Libraries
    a. Django for backend

4.3.3 Data Flow and Communication

1. User Inputs; Captures data from the user and sent to the backend for processing.

4.3.4 Outgoing Data Items

1. Search Results; Shows the film and everything related to it that the user is looking for.
2. Error message; This communicates issues encountered during user interactions.

4.3.5 Shared Data

1. Document/film metadata; shared the database and the frontend for search and display purposes.

4.3.6 Implementation Constraints

1. Data Consistency; This makes sure that any updates to the database are immediately updated on all the interfaces.

4.4 Communications Interfaces

The application will communicate with the end user via a web browser, and all information sent and received to/from the application will be routed this way. The end user will be expected to have a working internet connection and can connect with HTTP.

4.4.1 Communication Requirements

1. Web Browser
    a. Primary Protocol
        i. HTTP/HTTPS
    b. Message Format
        i. JSON
        ii. HTML
        iii. CSS/Javascript
        iv. AJAX/Fetch API
        v. WebSockets
2. Network Server
    a. Backend to Frontend
        i. RESTful API
        ii. GraphQL
        iii. Message Format
            1. JSON
            2. XML
    b. Backend to Database
        i. SQL
        ii. ORM
    c. Other

        i. WebSockets
        ii. FPT/SFTP

3. Forms
    a. HTML Forms
    b. Message Format
        i. JSON

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Search Queries

All search queries must return results within 3 seconds for datasets of up to 100,000 documents. For larger datasets, the response time should not exceed 6 seconds.

5.1.2 Load Time

The web interface, which includes both the search and filtering features, should load within 3 seconds on stable broadband connection (minimum 5 Mbps download)

5.1.3 Concurrent Users

The system should support up to 500 concurrent users without any degradation in performance due to the load. The performance will be tested under load to ensure scalability.

5.1.4 Data Processing

OCR operations for document uploads should complete within 1 minute for an average document size of 10 pages, with processing distributed to background tasks to avoid blocking the user interface.

5.2 Safety Requirements

5.2.1 Data Backup/Local Access

In case of database access interruption, users will have the ability to download the entire or parts of the entire database.

5.2.2 Database Backups

Full backups of the database will occur daily or whenever a document is added to the database.

5.2.3 Error Handling

If OCR fails due to document quality issues, the system will notify the user with a clear error message and offer options to re-upload or manually review the document.

5.2.4 Disaster Recovery

A disaster recovery plan will be in place to ensure that the system can be restored within 24 hours of a major failure.

5.3 Security Requirements

5.3.1 Data Modification

1. Only users with access to the backend of the database will be able to modify or add data to the database.

5.3.2 Server Access

1. In case of massive access requests, caching will be used to prevent an overload and shutdown of the main server(s).

5.4 Software Quality Attributes

5.4.1 Usability

The software shall provide an intuitive user interface, ensuring that users with varying levels of technical competence can easily navigate, search for, and retrieve documents. The search functionality will be straightforward, with minimal user input required to achieve desired results.

5.4.2 Maintainability

The system shall be designed to facilitate easy updates and maintenance. The architecture will support modular updates to ensure new features and fixes can be added without extensive rework.

6. Other Requirements

6.1 Update Integrity

Updates or changes to the database must not result in data loss.

Glossary

Optical Character Recognition: The process of converting an image of text into a machine-readable format (*What Is OCR? - Optical Character Recognition Explained - AWS*, n.d.).