Product Requirements

Project: E-7 LeaseAnalytics

Automated Paragraph Bounding Box Visualizer

Team: Null_biters

Brief problem statement

Lease Analytics data recovery company plans to use deep learning to extract valuable information from databases for their clients. However, Lease Analytics often scans documents that are difficult to extrapolate data from due to poor alignment, mold, and other problems. In order to perform data analysis on these special case documents, Lease Analytics needs a web app that can efficiently and reliably generate bounding boxes around paragraphs. We need to research an efficient and accurate solution as a team and build a small application for housing, selecting, and browsing pdfs as well as research an algorithm for making the bounding boxes. Taking into account the current technology stack and design, the frontend of the application will integrate with pspdfkit to browse and display sets of PDFs.

System requirements

What system configuration needs to run your proposed system (including anything third party that is needed to run your system).

- Access to Factz.io
- PostGreSQL, Javascript, Rails 7 (with temp active storage), VS Code, Node, a package manager (npm/yarn/bundler), Vue 3, AXIOS, Pinia, vite

Users profile

Who is the system intended for? What characteristics should the users have (this can also be a range of things such as reading level, etc.).

This system is intended to be used by technical employees of Lease Analytics and clients trying to browse, scan, and extrapolate data. Users are assumed to be technologically literate and have experience with data analysis.

List of Features

Provide a numbered list of features (F1, F2, etc.) that concisely, clearly, and accurately describe that which constitutes your project.

F1. Bounding box algorithm

F2. PDF Viewer

- ID, filename, page count, processing time

F3. Next and Previous Buttons

F4. Drawer for PDF browsing

- Contains a list of documents

F5. Sample Button

F6. Request PDF by id

Functional requirements (user stories)

List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

No. User Story Name Description Priority

No.	User Story Name	Description	Priority	Sprint No.
R1	Generate bounding boxes for paragraphs	As an employee of Lease Analytics or a client, I want to be able to see documents divided into sections by bounding boxes so I can analyze data contained in PDF files.	1	
R2	PDF viewer	As an employee of Lease Analytics or a client, I want to be able to see documents in the web app as you would be able to do like in any typical PDF viewer-type application so I can extract the data from these documents. Dev notes: download psPDFkit to get PDF viewer functionality	1	
R3	Next and previous buttons	As an employee of Lease Analytics or a client, I want to be able to press next and previous buttons located at the top of the left sidebar so I can move between the next and previous PDF files in the database. Dev notes: See meeting recording for page layout. Recording part 1 at about 4 min. in.	2	

R4	Menu that lists documents	As an employee of Lease Analytics or a client, I want to be able to see a menu located in the left sidebar that lists documents stored in the database so I can understand and navigate all the documents in the database.	2	
R5	Sample Button	As an employee of Lease Analytics or a client, I want to be able to press the Sample button (located on the top left corner of page) to fetch ~50 random pdf documents so that I can test the bounding box capabilities on these documents. Dev Notes: This will involve a web scraper	3	
R6	Request PDF by id	As an employee of Lease Analytics or a client, I want to query a database of ids and have it return a pdf	3	
R7	Basic Data about PDF	As an employee of Lease Analytics or a client, I want to be able to see ID, filename, page count, and processing time for each PDF	2	

Non-Functional Requirements

Describe any constraints or cross-cutting characteristics of the system in a manner that is clear, specific, and testable. Each requirement should have a unique identifier (e.g. NF1, NF2,..). Only present those which are applicable to your system. Categories include but are not limited to:

Accessibility:

Application should be usable for people with colorblindness.

Efficiency:

Original project was $O(n^2)$. Since improving on this is difficult, we need to achieve about this same complexity.

Security:

Authentication between backend and frontend via JWT's.

Usability:

A user who has never used this application should be able to easily learn to view PDF documents and switch to the next or previous PDF document.

Accuracy:

The new solution must accurately translate the text in the scanned pdf to text in a text file. 100% accuracy is expected for “clean” paragraphs. Margin of Error needs to be regular throughout the program.

Reliability:

Normal pages (text in regular paragraphs) should have 100% accuracy for bounding boxes

Pages with mold or incorrectly scanned should have a semi-reliable accuracy for bounding boxes.

Sponsor Requirements

I have read and approved the material in this document. If there is no external sponsor, the TA or the instructor will sign it for accuracy/scope.

_____ _____ _____

Print Name Signature Date