About this project

In the mortgage world, every lender has their own way of naming documents. One lender might call it a “W2”, while another might call it a “Form W-2”. This process of converting names from 1 to another is done through translations and will be covered in the “Classifications” section of this test.

Each document has various data fields. Think of your Bank Statement. It has information regarding your Account Number, the Account Owner, the Statement Balance and so on. Each of these values that can be extracted from the document is called a data element. The “Extraction” section of this test will cover some simple concepts during extraction.

Main Page

1. The front page has the following buttons:
    a. EXTRACTION
    b. CLASSIFICATION
2. When the user clicks on CLASSIFICATION: redirect to classifications page.
3. When the user clicks on EXTRACTION: redirect to extractions page.

Classifications Page

1. User can see 2 drop downs:
    a. Customer Names
    b. Document Types
2. User can select from the customer name drop-down
    a. 1 or more customer name
    b. All customer names
    c. Clear all selections.
3. User can select from the Document Types drop-down:
    a. 1 or more customer name
    b. All customer names
    c. Clear all selections.
4. Based on the selection from the dropdown, display a table that has the document types as the row and the customer name as the column. Select the value for a customer from the customers file (refer to customers-file-map.json) corresponding to the document types that have been selected.

5. Example,
   Selections in customers list (Customer 1, Customer 2, Customer 3)
   Selections in Documents list (Document 1, Document 2).

   Find value for Document 1 in Customer 1's file.
   Find value for Document 1 in Customer 2's file.

Find value for Document 1 in Customer 3's file.
Find value for Document 2 in Customer 1's file.
Find value for Document 2 in Customer 2's file.
Find value for Document 2 in Customer 3's file.

Expected Visual:

Document Type	Customer 1	Customer 2	Customer 3
Document 1	val1	val2	val3
Document 2	val4	val5	val6

6. Users have an ability to update any values in the table.
   Example:

   User can change val1 to val2, val3 to val9

   Expected Visual:

Document Type	Customer 1	Customer 2	Customer 3
Document 1	val2	val2	val9
Document 2	val4	val5	val6

7. When you hover over each document name in the table, it pulls up the sample document for that document name (refer to documents-file-map.json).
   E.g. Mouse over Document 1 shows sample for Document 1. If it's a multi page pdf should be able to scroll through pages.
8. Users can export the updated table as a CSV (see classification-export-example.csv in examples).
9. Ability to define new document types and add new samples.
10. Ability to define new customer names.

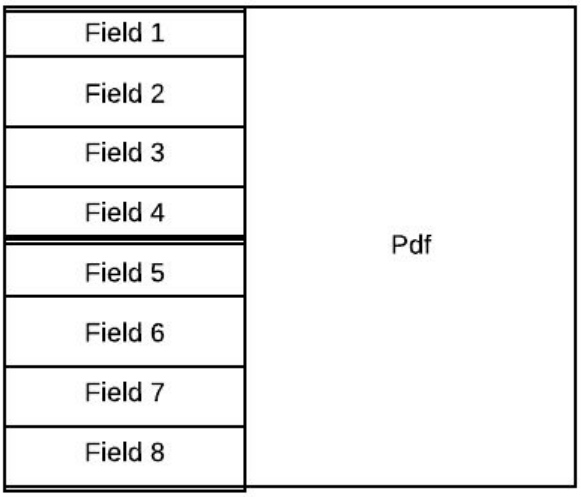
Extractions Page

1. User can see 1 drop down:
   a. Document Types
2. Based on document types selected:

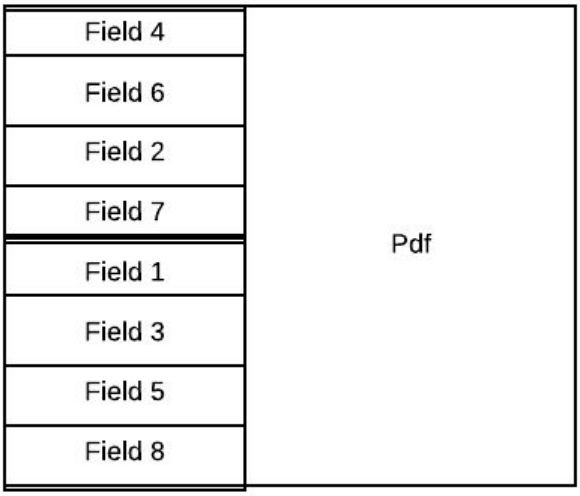
Read the data elements CSV (see extractions/data-elements.csv) and pull up all data elements for that particular document type.

a. Show all data elements (see column labelled “Data Elements”) as a list on the UI.
b. The sample document should be displayed next to the data fields. If it’s a multi page pdf should be able to scroll through pages.
c. The fields should be grouped together based on group number (see column labelled “Group Number”).
d. WIthin a group the data element order (see column labelled “Datapoint Order”) determines the order of each field within the group.

Example: Something like this:



3. Ability to rearrange data elements within groups and across groups.
   Example:



4. Ability to export a list of data fields for each document type as a file.

   Example file (see fields.txt in examples):

   Field 4
   Field 6
   Field 2
   Field 7
   Field 1
   Field 3
   Field 5
   Field 8

5. Ability to export all fields for all documents types as a csv. Each Column header indicates the document type. Each row for that column is 1 data element each. Example (see all-extraction-fields-export-example.csv in examples):

fx

	A	B	C	D	E
1	Document 1	Document 2	Document 3	Document 4	Document 5
2	Field 4	Field 1	Field 1	Field 4	Field 4
3	Field 1	Field 2	Field 4	Field 1	
4	Field 2	Field 3		Field 2	
5	Field 3	Field 15		Field 15	
6	Field 5	Field 9		Field 9	
7	Field 9			Field 10	
8	Field 15			Field 3	
9					
10					

Important Technical Requirements

1. For each functionality created, communication with backends MUST be through APIs.
2. Multiple Users should be able to use the UI simultaneously.
3. Ensure consistency in changes in both the CLASSIFICATIONS and EXTRACTIONS pages.
    a. Any changes made by 1 user should be visible to the other user and vice versa.
    b. Avoid conflicting changes by multiple users.
4. Should allow big files up to 100 m.b. to be uploaded as samples and scroll through all pages.