Regression Testing

This document describes both an in-depth regression testing process which can take hours to complete, as well as an abbreviated smoke test process that can be performed in about 10 minutes.

Testing Started:

Testing Finished:

Environment:

Current Commit:

Test count :- 200 roughly

Table of Contents

[**Regression Test**](#_kllupw61td5f) **3**

[Functional Tests](#_ra1z8iu6arbm) 3

[User - Creation / Login / Logout](#_4yh9tobzg0z0) 3

[File Upload (sample documents link)](#_q5ahmsy07xkh) 4

[File support](#_e27cwt93g5oq) 5

[New Collection](#_2p81z8l4tt3) 5

[Organize (AI Architect Tool)](#_wooanr5pcab3) 5

[Ingestion (AI Architect Tool)](#_jr20kbo3oj0a) 6

[Dashboard / Collection / Cluster](#_co9t9ctr0qad) 8

[Search](#_lvxhpzo5py) 9

[Usage Limits](#_h58fmrrdip6s) 10

[Confirmation=False / Positive Credit Limit / No Credit Limit setup](#_w7pr60ph0pa7) 10

[Confirmation=True / Hard limits setup](#_927jj9szazzc) 11

[Licensing Summary Report](#_k27tbw3ozr1i) 12

[Permissions](#_lwyti4kweerg) 13

[Celery / Background workers](#_nnqe7ol0dc7g) 14

[Peripheral Pages / Features](#_sximon91l9xz) 14

[Analytics](#_16ot0lym2ddx) 14

[Visual tests](#_5960jz9gznm) 15

[Load tests](#_uah3k6wcy4n) 15

[Large Data Sets](#_tvjktbso8v77) 15

[Concurrent Usage (over a 20-30 minute timeframe)](#_7s1aquq686ml) 16

[Code tests](#_89hx9fkafz3c) 17

[Security tests](#_sgbnda5e6jii) 17

[Browser/Device Support tests](#_1gwpcovuzrnu) 18

[Accessibility tests](#_bwvb5qfcszgs) 18

[Chaos test](#_ujzib91dbool) 18

[**Smoke Test**](#_vwomba7kzqpv) **20**

# Regression Test

## Functional Tests

### User - Creation / Login / Logout

1. Login as Admin
   1. Create a new non-staff user account, set first/last name, company, and profile picture. Login as new user and ensure information displays correctly.
   2. Create a staff user account and assign appropriate permissions (by adding to the “System Administrator” group)
      1. Create “System Administrator” group if needed and assign permissions as mentioned in the “Launch Plan Template” document.
2. Ensure new users created as “not active” by default and cannot log in.
3. Trigger “New user/account email” from Django Admin for a new user.
   1. Ensure email received
   2. Ensure user account is activated and able to log in.
4. Trigger “Reset password email” from Django Admin for an active user.
   1. Ensure email received.
   2. Ensure Reset Password flow is working.
5. Login as a normal user (or reset “Terms accepted” on existing user)
   1. Ensure presented with first-time modal
   2. Ensure requires acceptance of terms to proceed.
   3. Logout and back in, ensure modal is shown every time until terms are accepted
   4. Once terms accepted, ensure modal is not shown again.
6. Login as a user, wait 1 hour, ensure that next click will redirect to the login page.
7. Login as a user, then logout and ensure that you cannot “deep link” into a part of the site and see any data. Site should always redirect to login page.
8. Logout then attempt to access a “deep link” to a Collection.
   1. Ensure application goes to login page
   2. Ensure after logging in (even with incorrect password failures), redirects to originally requested “deep link”
9. Ensure “only one login session per account” is active
   1. Login as a user in browser1.
   2. Open a Incognito window (or on a separate VM or computer), browser2
   3. Login as the same user on browser2.
   4. Go back to browser1, attempt to click on something in the application. Ensure that the original user session has been deleted/expired.
   5. Continue using the site on browser2, ensure that the session is still active.

### File Upload (sample documents [link](https://drive.google.com/open?id=1mDdIcU4RCoS3MgNYONzKeUyiLl1B16Ok)) - need precondition

1. Upload a single supported image file (jpg, png, bmp, multi-page tiff, gif)
2. Upload a single PDF file
3. Upload multiple PDF files with tables
4. Upload a single DOCX file
5. Upload a single XML file
6. Upload a single CSV file
7. Upload a single XLSX file
8. Upload a single TXT file
9. Upload a large image (100MB, but less than 101M pixels)
10. Upload a small ZIP file (5-10MB)
11. Upload a medium ZIP file (100-200MB)
12. Upload a large, but supported ZIP file (495MB)
13. Upload a small ZIP created from a Windows machine.
14. Upload a TAR file (5-10MB)
15. Upload 100 files in a single “drop”, including supported files and ZIP/TAR files.
16. Upload 10 files, including an unsupported file type.
17. Special cases
    1. Upload a too large ZIP file (502MB), ensure proper user messaging of file being too large.
    2. Upload a single large image (more than 101M pixels), ensure that the file is uploaded (TransferBatch/TransferFile created), but then ignored (no UnpackFile created) with a message added to the TransferBatch object.
    3. Upload a large file (or throttle browser network speed) and cancel mid-way through the upload. Ensure UI messaging is suitable and that no TransferBatch was created.
    4. Upload a large file (or throttle browser network speed), try to close the browser tab mid-way through the upload. Ensure the browser-based warning about leaving the page is shown.
    5. Upload a ZIP with some supported files and some unsupported files. Ensure that the unsupported files are “ignored” (no UnpackFile created).
18. Ensure emails on initial upload and preprocess completion are sent to the support email for each upload (except file-too-large example), and the contents (subject/body) are correct.

### File support

1. Perform the steps in the File Upload section and check the following
2. Login as a staff user.
3. Ensure ZIP/TAR file extractions (creation of UnpackFile) succeeded
4. Ensure preview\_image created/available for each page of each file type: image file, PDF file, and DOCX file.
5. Ensure reasonable parse\_tree created for DOCX file.
6. Ensure “has\_header”, “rows”, and “cols” detected for CSV (TODO - XLSX) and set on UnpackFilePart metadata attribute.
7. Ensure reasonable “full\_text” extracted for all file types.

### New Collection

1. After collection created
   1. Ensure collection naming dialog is shown
   2. Ensure file(s) show in processing queue while “QUEUED”
   3. Ensure file(s) are removed from processing queue after they are UNPACKED, and are shown in the “Routing” tab inside Ingested Documents
2. Ensure no clusters shown
3. Ensure no certainty or error information shown
4. Ensure User is unable to “Download” the collection data.
5. Ensure User is able to upload more files, and that it creates a new “queued” transfer

### Organize (AI Architect Tool)

1. Log in as a staff user
2. Create a Cluster (status=Pending) through the Django admin for a new Collection with any files that have images created. Use the default cluster icon
3. Create a Cluster (status=Pending) through the Django admin for a new Collection with PDF files that have tables included. Use the non-default cluster icon
4. Create a Cluster (status=Pending) through the Django admin for a new Collection with DOCX files. Use a non-default cluster icon.
5. Go to the Architect tab and It will navigate to AI tool
6. For both Collections:
7. Ensure Collections are listed
8. Ensure Cluster for Collections are listed
9. Ensure TransferBatches for Collections are listed.
10. Ensure able to move a file from a TransferBatch to a Cluster.
11. Ensure able to move a (non-INGESTED) file between Clusters in a Collection.
12. Ensure TransferBatch is not displayed once all files are moved to a Cluster
13. Go to /admin and set one of the Cluster’s status to “Proposed”, one to “Queued”, and one to “In Progress”
14. Logout and login as a non-staff user.
15. Ensure that the Pending Cluster is not shown on the Collection.
16. Ensure that the files moved into the Pending cluster are shown in the “New Doc. Type Identified” tab of the “Ingested Documents” area
17. Ensure that the “Proposed”, “Queued” and “In Progress” clusters ARE shown on the Collection’s Cluster list.
    1. Ensure that all the files for these Clusters are NOT shown in any of the Ingested Document tabs, and that all the files for the Clusters ARE shown in their respective Cluster detail pages.
    2. Ensure the Cluster detail page behaves appropriately for these Clusters (no certainty displayed, no “download structured data” button, etc.)

### Ingestion (AI Architect Tool)

1. Log in as a staff user
2. Ensure Collections are shown
3. Ensure Cluster is listed under the Collections
4. Create a new draft Ingest Plan for the clusters
   1. Add 4 or 5 different Data Labels under the schema
      1. For Cluster with PDFs including tables
         1. Include a “Table” Data Label.
   2. For Cluster with files with images
      1. Add a bounding box step for each data label created.
   3. For Cluster with PDFs including tables
      1. Add a TableSelector & Table Iterator & RowValue step chain
   4. For Cluster with DOCX files:
      1. Add a ParseTreeQuery step for each data label created.
   5. Under the bounding box step, add a CreateDataAtom next step which saves the value to one of the data labels.
   6. Other Step tools:
      1. Regex Step
      2. Next Step Selector
      3. Passthrough Step
   7. Ensure previewing the plan on a few documents works as expected.
   8. Ensure the “Available” / “Plan” / “Preview” bounding box checkboxes work as expected.
5. Log out and in as a non-staff user.
   1. Ensure that the “Pending” Cluster is not visible, and that no document/extraction/certainty information are shown for any of the Clusters and do not affect overall Collection metrics.
6. Login as staff user
   1. Modify the Ingest Plan, overriding the bounding box or ParseTree query step for a particular unpackfile\_id
   2. Modify the Ingest Plan, overriding a CreateDataAtom step for a low certainty / high data error extraction by providing a corrected value.
   3. Ensure previewing the plan on a few documents works as expected.
   4. Ensure right-clicking on a file in the “Unstaged” folder shows the “Stage File” option and moves it to the “Staged”. Ensure the reverse action works from Staged to Unstaged.
   5. Leave at least one file in “Unstaged”
   6. Execute the “Ingest and Activate” action on the Clusters / IngestPlans.
   7. Ensure that a IngestResult object is created, is executed in a background process, and gets a status value of complete.
   8. Ensure DataSet/DataRecord/DataAtoms were created as expected. Ensure that the UnpackFiles were set to status=INGESTED.
   9. Ensure that files still in “Unstaged” where not ingested.
7. Login as normal user.
   1. Ensure that the Pending Cluster is not visible, and that no document/extraction/certainty information from ANY of the Clusters in Pending/Proposed/Queued/InProgress are displayed or affect the Collection metrics.
8. Login as staff user
   1. Set the Cluster status to Active for one of the Clusters
9. Login as normal user.
   1. Ensure that the Active Cluster is visible, uses the expected icon, and that all the document/extraction/certainty information reflects the ingested files.
   2. Upload additional files to the Active Cluster.
   3. Upload an “ERROR” file (like a corrupt PNG), and a file that will be marked later as “Ingestion Unsupported”.
10. Login as staff user
    1. Add the new files to the Active Cluster
    2. Set the UnpackFile.status for one of the non-ERROR files to “INGESTION\_UNSUPPORTED”.
11. Login as normal user.
    1. Ensure the new files are displayed in the “Routing” tab of Ingested Documents area, and identify the Active Cluster that they are associated with.
    2. Ensure the ERROR and INGESTION\_UNSUPPORTED files are shown in the “Exceptions” tab of the Ingested Documents area
12. Login as staff user
    1. Go to the Ingest area and preview the IngestPlan for the new files.
    2. Execute the “Ingest new Files” action.
    3. Ensure that a new IngestResult was created, but re-used the previous DataSet object and added new DataRecords for the newly added files and set the UnpackFiles to status=INGESTED.
13. Login as normal user
    1. Ensure that the newly ingested files are represented in the Active Cluster.
    2. Ensure that the certainty is correctly calculated.
    3. Ensure that the data error graph correctly buckets the data error counts at the different likelihood levels.
    4. Download the data CSV at the Cluster level.
    5. Download the data ZIP (assuming multiple Clusters) at the Collection level

### Dashboard / Collection / Cluster

1. Login as normal user
2. Rename a Collection.
3. Rename a Cluster.
4. Ensure sorting Collections by all options works as expected.
5. After sorting, and clicking into a Collection, the LeftNav should show the Collections according to the “index” order (1,2,3,4) regardless of sorting on the Dashboard.
6. Collections from Ingest steps above
   1. On the dashboard
      1. Ensure that the certainty/documents/extractions/data errors numbers are correctly calculated.
         1. Certainty = average across the certainty values of all the data atoms.
         2. Total documents = number of UnpackFiles
         3. Total extractions = total number of data atoms
         4. Total data errors = total count of data atoms with data\_error\_confidence above 0.5
   2. On the Collection detail page
      1. Ensure that the certainty number and documents number (under list of Clusters) is correctly calculated.
         1. Certainty = same as above
      2. Ensure that the data error graph correctly buckets the data error counts at the different likelihood levels.
      3. Ensure the Cluster list correctly calculates/shows documents/extractions-per-doc/certainty/data errors
         1. Total documents = same as above
         2. Extractions Per Doc = number of data labels in the Cluster data set.
         3. Total data errors = same as above
   3. On the Cluster detail page
      1. Ensure the documents/extractions-per-doc/certainty/data errors numbers are shown correctly for the Cluster.
         1. Same as above on Collection Detail -> Cluster list
      2. Ensure the files are listed correctly.
      3. Ensure the file list lazy loads additional sets of files.
      4. Ensure the certainty/data errors show correctly for the files
         1. Certainty = same as above
         2. Total data errors = same as above.
      5. Ensure clicking on a file row updates the preview image
      6. (currently disabled in the code, don’t test for now) ~~Ensure clicking on a file row shows the bounding boxes for the extracted text from the document.~~
      7. Ensure clicking download icon on a file row initiates the browser to download the original uploaded file.
   4. Download the data CSV at the Cluster level.
      1. Ensure the output contains all the data labels (including certainty, data error), and values for each row.
      2. Ensure the values, certainty and data\_error are correct.
   5. Download the data ZIP (assuming multiple Clusters) at the Collection level
      1. Ensure each CSV file is correct as above.
7. (currently disabled in the code, don’t test for now) ~~Any collection~~
   1. ~~Delete a collection~~
   2. ~~Try providing incorrect password.~~
   3. ~~Provide correct password~~
   4. ~~Ensure Collection, Cluster, UnpackFiles, UnpackFileParts, TextRenderings, TransferFiles, TransferBatches, IngestPlans, IngestResults, DataSets, DataRecords, DataAtoms, DataSetExports are all deleted.~~

### Search

1. From the Dashboard page (or any location other than Collection or Cluster detail)
   1. Perform a search for an “abundant” search term like “test”
   2. Ensure results are grouped by Collection
   3. Ensure pagination occurs within each Collection (load more results loads 10 more in that Collection.
   4. Ensure “x of y results” counts make sense
2. Ensure that the search itself (AJAX call) doesn’t fire until you’ve stopped typing for 1-2 seconds.
3. From a Collection page
   1. perform a search for a term that doesn’t exist in the documents in the Collection, ensure no results are returned.
   2. Unselect the “this collection only” checkbox, ensure that results outside of the Collection are returned.
   3. Perform a search for a term that exists multiple Clusters within the Collection, ensure the results show across the multiple clusters.
4. From a Cluster page.
   1. perform a search for a term that doesn’t exist in the documents in the Cluster, ensure no results are returned.
   2. Unselect the “this cluster only” checkbox, but keep the “this Collection only” checkbox, ensure that matching results from the Collection level are shown.
   3. Unselect the “this Collection only” checkbox, ensure that results from across the platform are returned.
5. From any search results
   1. Ensure that clicking on the Collection name in the results take you to the cluster.
   2. Ensure that clicking on a result takes you to the Cluster where the document is (or Ingested Documents area)

### Usage Limits

#### Confirmation=False / Positive Credit Limit / No Credit Limit setup

1. Will need to update the Django settings file to set DocumentDownloadRecords AND DocumentDownloadRecordPages licensing to CONFIRMATION=False
2. Positive Credit Limits
   1. Set a specific credit limit per Liability account.
   2. Make it some number that you can reasonably stay “under” based on the existing balance on the account. Also should be a number that you can easily then attempt to surpass.
   3. Perform an action that uses the remaining balance on each Liability account, and proceeds into “negative balance” territory, thus utilizing some (but not all) of the set credit limit.
   4. Confirm that the account shows a negative balance.
   5. Perform an action that would attempt to go past the credit limit on each Liability account.
   6. Confirm that the system blocks/prevents the action (either through a modal for Downloads, or with a system email notification for Uploads, etc.)
3. No Credit Limit
   1. Remove the credit\_limit value from the Liabilities accounts. (Just clear out the text field and save the object.)
   2. Perform an action that uses for values for each Liability account.
   3. Confirm that the balance continues to go into negative territory without blocking the user.

#### Confirmation=True / Hard limits setup

1. Will need to update the Django settings file to set EITHER DocumentDownloadRecords or DocumentDownloadRecordPages licensing to CONFIRMATION=True
2. Credit “spend” tracking
   1. Start with enough credits for all operations.
   2. Ensure that creating a new user “spends” a User credit.
   3. Ensure that uploading documents accurately “spends” UploadDocumentPage credits at the correct rate.
   4. Ensure that downloading document accurately “spends” DownloadDocumentRecord credits.
      1. Credits should only be spent when first downloading a record. Once “unlocked” ensure that the record can be accessed/downloaded without spending credits.
   5. Ensure that downloading document accurately “spends” DownloadDocumentRecordPages credits. Should be based on the number of UploadPage credits used to original upload each document.
      1. These credits should ALWAYS be spent, even when re-downloading previously unlocked/downloaded records.
      2. These credits should NOT be spent when a staff user is downloading.
3. Partial credit restrictions (DownloadDocumentRecords)
   1. Attempt to download a Cluster where there are not enough remaining credits to unlock all records.
   2. Ensure the system properly describes how many records can be unlocked, and gives the option to just download previously unlocked files (if any), and how many total rows (previously unlocked + newly unlocked, which would be less than all records) would be in the CSV.
   3. Attempt to download a Collection, it should only provide an option to download previously unlocked files, and no option to spend the remaining credits.
4. Partial credit restrictions (DownloadDocumentRecordPages)
   1. Attempt to download a Cluster where there are not enough remaining credits download the documents.
   2. Ensure that the system notifies the (non staff) user that they are not able to download the Cluster at all due to not enough credits.
5. Partial credit restrictions (Uploads)
   1. Attempt to upload files/pages that would need more than the remaining UploadDocumentPage credits. The system should send an email to the user and delete the attempted upload files, but retain the TransferBatch record with a description of the problem in the “error\_log” attribute.
6. No credit remaining restrictions
   1. Start with no remaining credits for any operations.
   2. Ensure that new uploads result in the error email to the user as above.
   3. Ensure that all Cluster/Collection downloads only allow downloading previously unlocked records.
   4. Attempt to create a new user account (through Django admin), it should fail due to lack of credits.
7. Credit expiration
   1. Create a LiabilityAccount with an expiration date of the following day (tomorrow) at midnight.
   2. Fund the account with some number of credits.
   3. On the next day, check back that the LiabilityAccount is “Closed” and has no balance.
   4. Check that the credits from the LiabilityAccount have been transferred to the corresponding “Lapsed Account”.

#### Licensing Summary Report

1. After performing all the various licensing operations above…
2. As staff user, go to /admin and “Periodic Tasks”
3. Create, or re-use a “one off” periodic task that runs the “noble.usage\_limits.tasks.create\_and\_send\_licensing\_report” task.
   1. Set the “Enabled” checkbox to False.
   2. Set the “Arguments” text area to “[8,2019]”, where the first number is the Month you want to report on, and the second number is the year.
4. Previous Month report
   1. Set the “Arguments” value to a previous month, with some amount of licensing activity.
   2. On the list of “Periodic Tasks”, check the task from the list and select “Run selected tasks” from the Actions list
   3. Confirm the report is emailed to the “support” email group/list
   4. Confirm all the licensing types, accounts, starting/ending balance, etc. are accurate.
5. Current Month report
   1. Set the “Arguments” value to the current month, which should include all the activity from testing above.
   2. On the list of “Periodic Tasks”, check the task from the list and select “Run selected tasks” from the Actions list
   3. Confirm the report is emailed to the “support” email group/list
   4. Confirm all the licensing types, accounts, starting/ending balance, etc. are accurate.

### Permissions

1. Login as UserA, create a CollectionX with some documents
2. Login as staff and create a Cluster for that Collection
   1. Ingest some (but not all) documents into that Cluster, leave some in IngestedDocuments.
3. Login as UserA, go to CollectionX
   1. Download the records for the Cluster
   2. \*\*Make a record of a URL to a document preview image for next section\*\*
4. Login as UserB, for CollectionX, ensure you can NOT
   1. See the collection on the Dashboard
   2. See the collection on the LeftNav
   3. “Deep link” to the collection
   4. “Deep link” to the collection upload page.
   5. “Deep link” to the Ingested Documents page
   6. “Deep link” to the cluster
   7. See the documents in global search
   8. Use a document preview URL for a document in CollectionX
   9. [TBD]
      1. Find other ways to ‘attack’ CollectX-related API endpoints as UserB and confirm access is blocked - Upload Docs, Get List of Docs for Cluster / ‘Ingested Documents, etc.
5. Login to /admin as staff user
   1. Create a group
   2. Add UserB to group
   3. Add group to CollectionX
   4. Assign permissions to CollectionX
6. Login as UserB
   1. Ensure you can now do all the actions on CollectionX
   2. Ensure all Collections are visible on the Dashboard
   3. Ensure loading of any Collection or Cluster page.
   4. Ensure all files accessible through global Search
   5. Ensure able to download records
   6. Ensure file preview images can load
   7. Ensure able to download any file.
7. Login (to FE) as staff user
   1. Ensure you can do all the actions that UserA/UserB can
8. Login to /admin as staff user
   1. Assign designated “Everyone” group to CollectionX (create a group named exactly “Everyone” if there is none)
9. Login as UserC
   1. Ensure you can do all actions on CollectionX

### Celery / Background workers

TODO -

### Peripheral Pages / Features

1. Ensure the User Profile menu / drop-down is operational
2. Ensure the “User Manual” link opens a new browser tab showing the UserManual.
   1. Ensure that clicking on an item in the left nav scrolls the user to that item.
   2. Ensure that an item in the User Manual can be “deep linked” to from a new window
3. Ensure the “FAQ” link opens a new browser tab showing the FAQ.
   * 1. Ensure that clicking on an FAQ item expands the answer
     2. Ensure that a FAQ item can be “deep linked” to from a new window
4. Ensure the “Onboarding” link brings up the onboarding wizard flow.
5. Ensure the “Contact Support” link triggers an email send.
6. Ensure the “Terms of Service” link opens the terms in a new browser tab.
7. Ensure you can create a “util” ECS task and ssh into it.

## Analytics

1. Ensure that the Google Analytics tracking code is loading properly
2. Ensure that events are being sent to the Google Analytics server.

## Visual tests

TBD - For now should be someone from the Design team looking over the site to ensure all visual elements are (still) implemented

## Load tests

### Large Data Sets

1. If they don’t already exist on the environment
   1. Create a Collection with 10 Clusters
   2. Create a Cluster with 1000+ files.
   3. Otherwise, is ok to re-use existing Collections/Clusters for this test, as long as they are of substantial size to test appropriate handling of the large data set for various platform actions.
2. Confirm AI Training Tool performs appropriately dealing with 1000+ files in Collection/TransferBatch and Cluster.
3. Create an IngestPlan with 20 labels/steps
4. Confirm IngestResult (DataSet) creation completes within reasonable time (10-15 minutes)
5. Download Cluster CSV, ensure preparation status is reasonably displayed to user (can judge status of completion within a minute or so) and download completes successfully.

Metrics from last “large data set” run:

1. Uploaded 1280 files
   1. All PDF files
   2. 45.4MB Zip file (52MB uncompressed).
   3. Contained 8,192 pages total
      1. Avg = 6.4 pages/document.
2. Data transfer to server based on past experience would have been <30 seconds.
3. Server resources
   1. 2 worker servers in parallel
   2. 1 vCPU, 4GB RAM per server
4. ***PDF*** Preprocessing
   1. Started May 7, 2019, 5:44 p.m.
   2. Completed May 8, 2019, 5:06 p.m.
   3. Total time ~ 24 hours (86,400 seconds)
   4. Work performed
      1. Split multi-page PDFs into single-page PDFs
      2. Create Image rendering of each PDF page.
      3. Perform OCR on each page image
   5. OCR’d 2,598,065 individual words
      1. Avg = 317 words / page
      2. SQL for getting this:
         1. select count(tr.id)
         2. from transfer\_transferfile as tf, unpack\_unpackfile as upf, unpack\_unpackfilepart as upfp, unpack\_textrendering as tr
         3. where tf.transfer\_batch\_id=69 and tf.id=upf.transfer\_file\_id and upf.id=upfp.unpack\_file\_id and upfp.id=tr.unpack\_file\_part\_id;
   6. PDF Preprocessing metrics
      1. ~ 10.5 seconds / page
      2. ~ 30 words / second
      3. ~ 27 minutes / ***PDF***-megabyte
5. Manual “IngestPlan”
   1. Manually created an IngestPlan with 10 data labels / 10 bounding boxes
   2. “Preview” ingestion computation response time was nominal (~1 second)
6. Data extraction processing
   1. ~10 minutes to run the IngestPlan against each 1280 documents
   2. ~ 0.5 second / document
7. CSV preparation
   1. After user requests to download a Cluster, background job prepares the CSV file
   2. ~90 seconds to prepare the file.
   3. 14.2 rows/records/documents per second.
   4. Generated 4.5MB CSV file
8. CSV file download
   1. Somewhat dependant on user’s network quality.
   2. ~15 seconds to download on WeWork network.

### Concurrent Usage (over a 20-30 minute timeframe)

1. Start with 2 or 3 Collections with 2 or 3 Clusters, each with 50-60 documents.
2. 5 users should log in at the same time
3. User 1 - Create a new Collection with a Upload a mid-sized ZIP file (200MB). Repeat (either with new Collection or existing collection) every 5 minutes.
   1. Ensure admin notification emails are received at a reasonable rate.
4. User 2 - Login as staff user. Go to one of the new Collections, create a Cluster and add files. Create an IngestPlan, preview a number of documents. Publish and Ingest the plan/documents. Repeat on additional documents
5. User 3 - View Collection Dashboard, click between Collections every 20 seconds or so. Download the entire Collection every 4th or 5th time.
6. User 4 - View a Collection, click between Clusters every 20 seconds or so. Within a Cluster, click between 5-10 of the documents.
7. User 5 - Execute a site search that returns at least 10 documents, wait 10 seconds. View a Collection, perform a search that returns at least 10 documents, wait 10 seconds. View a Cluster, perform a search for that returns at least 10 documents.

## Code tests

* [todo] Check test code coverage levels for appropriate levels.

## Security tests

1. Ensure HTTPS uses valid certificate
2. Ensure forced redirects from HTTP to HTTPS
3. Ensure HTTP Auth is in place and operates across all items delivered by our servers.
   1. Main React app
   2. /api
   3. AI Training Tool
   4. /admin
   5. All static assets across the above
4. Ensure access to /api is blocked without a Token in request header or active Cookie-based session.
5. Check the following is not accessible from the outside world
   1. S3 buckets with files
   2. RDS database
   3. Redis cache
6. Check the AWS Security Hub for notifications on security vulnerabilities in the infrastructure.
7. Run passive pen-test monitor on site.
8. Review AWS Trusted Advisor report
9. TODO
   1. Check ddos protections

## Browser/Device Support tests

1. Log into site from Windows OS (can be in a VM) + Chrome Browser
   1. Perform “Smoke test” script
2. [Future] - Perform “Smoke test” script with Mac+Safari, Windows+Firefox, iPhone, Android
3. Using the following hardware scenarios, perform a visual check of pages and hover/focus/selected states to ensure contrasts are sufficient
   1. Using a non-Mac laptop and laptop screen
   2. Using a non-Mac laptop connected over HDMI to a TV monitor
   3. Using a Mac laptop connected over HDMI to a TV monitor
4. Use a 1336x768 resolution display (VM-level preferred to account for browser URL field, etc.)
   1. Note: Current code support level is 1440x??? (corresponding on Mac)

## Accessibility tests

1. Attempt to use the site purely with the keyboard.
2. Check for alt text and screen reader items (if possible)
3. Check for font sizes / contrast
4. Check for color-only feature designations (color blindness, visibility, etc.)

## Chaos test

1. Anywhere a normal user can supply text input, try unusual values:
   1. Numbers, special characters, unicode, emojis
   2. All Spaces, Short values, really long values
   3. HTML, JavaScript
2. Try random URLs
3. Try loading site without JavaScript
4. Turn down / throttle network in browser and navigate site.
5. Delete Collections
6. Try to access deleted items through URLs
7. Upload really small images/documents
8. Upload empty files
9. Upload ZIP files with supported and unsupported files.
10. Upload corrupt files
11. Ingest files/text with values that could cause problems with CSV export.
12. Upload files with weird characters in the filename
13. Try using the forward/back buttons and browser reload in various locations to test navigability.
14. Use a browser that has visited the site before, checking for incorrect file caching methods.

# 

# Smoke Test

An abbreviated version of regression tests, meant to check latest code on test.internal.noble.ai before starting a release/\* branch for full regression.

1. Use an Incognito Browser
2. Login with normal user
3. Create a Collection with a single file.
4. View “processing” Collection Detail, Ingested Documents queued and complete - refresh page and see it’s complete
5. Do a multi-file upload to an existing Collection, including a ZIP file and supported file types.
6. View “processing” Collection detail with additional queued items
   1. Confirm upload received and processing complete emails are sent to staff.
7. Log out and back in as staff user.
8. In the AI Training Tool (<server>/training), confirm all uploaded files are “Unpacked”, and available to be placed in a Cluster.
9. Create a new “pending” Cluster in Django Admin
10. The In Organize area of the AI Training tool, place uploaded files into the created cluster
11. Create draft Ingest Plan

{

"steps": [

{

"step\_type": "FixedBoundingBoxStep",

"name": "bbox - label1",

"unpack\_file\_part": "0",

"x": "0.1",

"y": "0.2",

"w": "0.4",

"h": "0.05",

"next\_steps": [

{

"step\_type": "CreateDataAtomStep",

"name": "atom - label1",

"data\_label\_name": "label1"

}

]

}

],

"schema": {

"data\_labels": [

{

"name": "label1",

"displayName": "label1",

"type": "STRING"

}

]

}

}

1. Confirm previewed ingestion
2. Stage one or more documents for ingestion
3. “Ingest&Activate” the plan (which creates an Ingest Result)
4. Confirm Ingested Result - check in Django Admin Ingested Results is Complete
5. Confirm the status of all UnpackFiles is “Ingested” - check in Django Admin, Unpack Files and confirm file is Status=Ingested
6. Update status of Cluster to “Active”
7. Log out back in as normal user.
8. Confirm Dashboard shows certainty/extractions/etc for Collection
9. View Collection Detail, confirm same numbers
10. Confirm Cluster shows, with numbers
11. Go into Cluster, confirm document preview images show ~~with bounding boxes (feature currently disabled)~~
12. Confirm export/download of CSV at Cluster level and Collection level.