

Product Design

Team 41

Members - Karsh, Manvith, Mrinal, Shanmukh

Design Overview

Architectural design

The subsystems will be:

1. Opening and annotating of documents. The modules in this will be:
 - a. A module for uploading files
 - b. A module for opening files in the browser
 - c. A module for highlighting/annotation/commenting on the file
2. Searching for comments/annotations. The module(s) in this will be:
 - a. A module for elastic search on the comments/annotations
3. Continuous integration/deployment. The module(s) in this will be:
 - a. A module for continuous integration

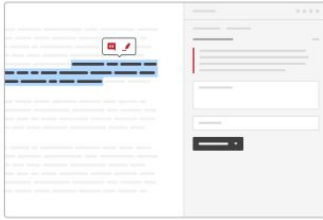
Relations between subsystems:

- The second subsystem will perform elastic search on the database created by subsystem1 to find posts/annotations/comments
- The 3rd subsystem will take the API given by the second subsystem and deploy the product, providing continuous integration and deployment

System interfaces

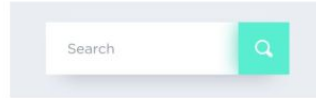
User Interface

- User will be able to upload a file (eg: PDF) via GUI.
- User can open any of the previously uploaded files (eg: PDF).
- Users can highlight text and add annotations/comment to the file (eg: PDF).
- Users can search the annotations/comment added.
- Users can then discuss/respond to the annotations/comment added.
- Users can upvote/downvote a topic.
- Users can delete/edit their comments



Users can upload papers to the website and add annotations.

The annotations and papers can be searched for using a search bar (using elastic search).



Users can comment, discuss, and interact on the specific paper.

APIs

An API which uses the built in APIs of Elastic search like:

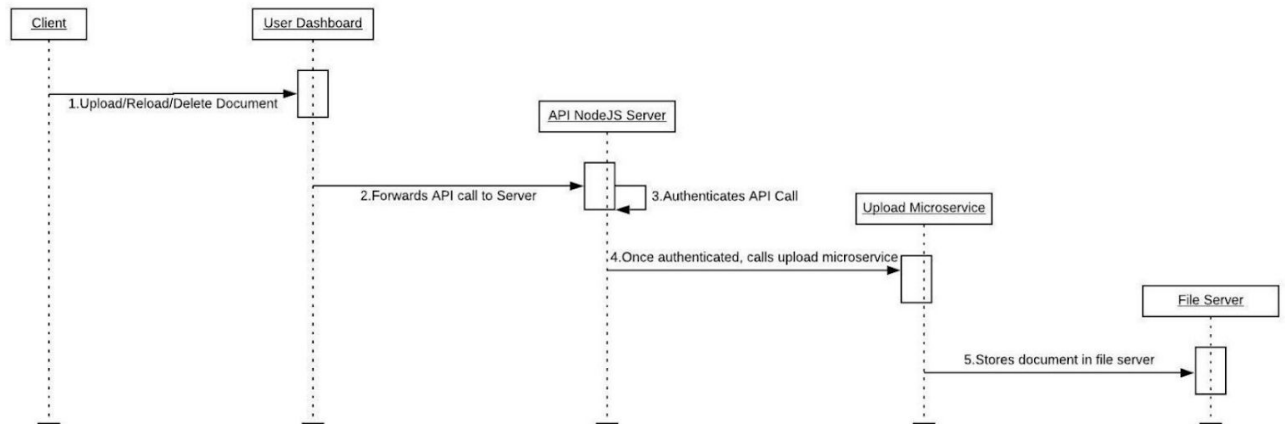
- Index API: Adds a JSON document to the specified index and makes it searchable. If the document already exists, it updates the document and increments its version
- Get API: Retrieves the specified JSON document from an index
- Update API: Removes a JSON document from the specified index
- Delete API: Updates a document using the specified script.

Model

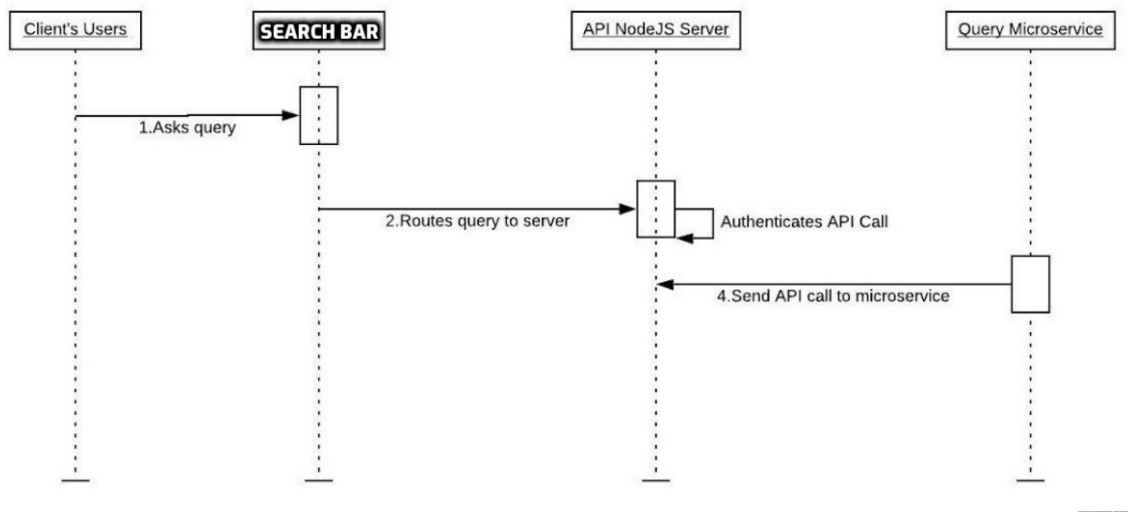
A class for handling files	<p>Class state</p> <ul style="list-style-type: none"> • It will hold information about files <p>Class behavior</p> <ul style="list-style-type: none"> • A method in the class will handle the uploading of files • A method in the class will handle maintaining the annotations
A class for doing elastic search	<p>Class state</p> <ul style="list-style-type: none"> • It will hold the method for elastic search <p>Class behavior</p> <ul style="list-style-type: none"> • The class will have a method for doing elastic search on the database of annotations
A class for doing continuous integration	<p>Class state:</p> <ul style="list-style-type: none"> • It will maintain information on whether or not the current system has been deployed <p>Class behavior:</p> <ul style="list-style-type: none"> • It will do continuous integration

Sequence Diagram(s)

Upload/post sequence:



Search sequence:



Design Rationale

- How to handle the case where multiple users annotate a pdf at the same.
- How to personalize the search results according to a specific user.
- Deciding whether to include the title of the post in the search or just search the annotations.
- Whether to enable feedback functionality for non-logged in users.
- What is the annotation limit for one post? (if any)
- Deciding whether expletives should be auto moderated.