BUILDERTREND

Text Extractions from Documents into Elasticsearch

Advisor: Goce Trajcevski | **Team :** Bruce Bitwayiki, Jared Hayashi, Rushal Sohal, Tiffany Mayberry | sddec22-19

The Client

Industry-Leader in construction management software

The Problem

Supports search only by filename

Extra time manually searching results

Our Goals

Easier Search

Quicker Search

User Interactions

Client's Customers

Primary Interactions:

- Web Application
- Search Filter Options
- File Uploader

Client's Staff

Primary Interactions:

- Web Application
- Reports & Queries made by users

Design Requirements

.txt .docx .xlsx .pptx .pdf .png .jpg

Functional requirements:

- Support above file types
- Extract & index file metadata
- Allow file upload
- Search filters - keyword, file type, file name, author, date

Non-functional requirements:

- Display search results ordered by best fit
- Display filename and path
- No authentication and authorization
- Easy to deploy and scale

Engineering Constraints:

- Query keyword up to 140 characters
- Utilize Elasticsearch to store metadata
- Return result in 10s (or a subset of it)
- Desktop / web app
- Readable in windows larger than 500x600

Relevant Standards:

- IEEE-29119: write unit tests for functionality
- IEEE-12207: standard software development lifecycle
- IEEE-7.8: code of ethics

Operating Environment:

- Web browser / Node env.
- Elasticsearch in Docker
- Spring Server on Ubuntu VM

Design and Implementation

Frontend:

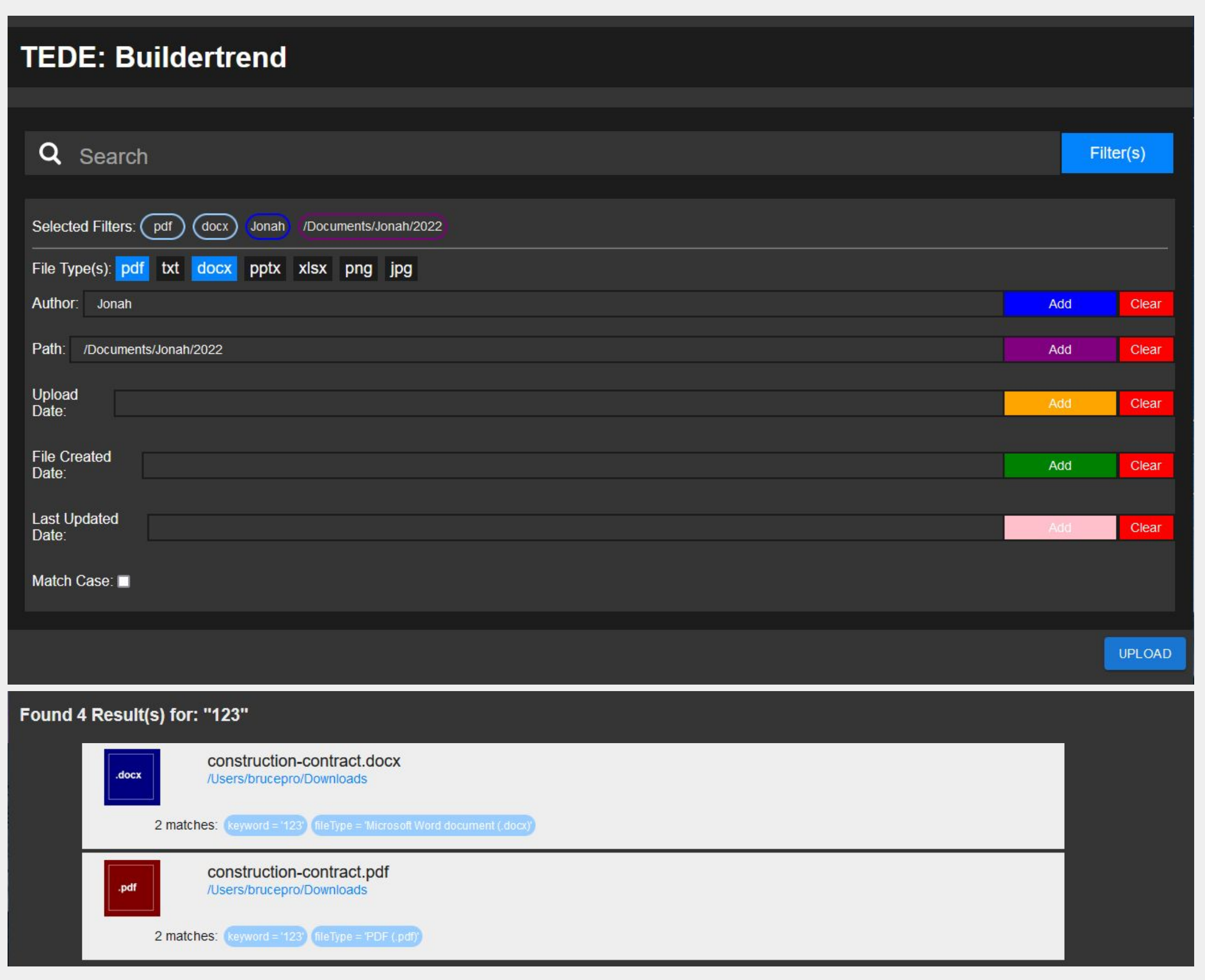
- Web Application: allows users to easily make and filter searches, view results & upload files
- Elastic Query Handler: formats the search parameters into a Elasticsearch query to retrieve data
- File Upload Handler: Stores received files & triggers the Text Extractor for those files

Content and Metadata Extraction:

- Apache Tika API is used to extract relevant data from various source documents
- Extracted data is formatted into a JSON and sent to Elasticsearch using REST API

Data Indexing and Querying

- Elasticsearch takes in structures JSON from the extracted files, stores and indexes it according to a specified mapping
- Distributed architecture of Elasticsearch allows users to to search and analyze the data in near real time



TEDE: Buildertrend

Search | Filter(s)

Selected Filters: pdf docx Jonah /Documents/Jonah/2022

File Type(s): pdf txt docx pptx xlsx png jpg

Author: Jonah | Add | Clear

Path: /Documents/Jonah/2022 | Add | Clear

Upload Date: | Add | Clear

File Created Date: | Add | Clear

Last Updated Date: | Add | Clear

Match Case: ☐

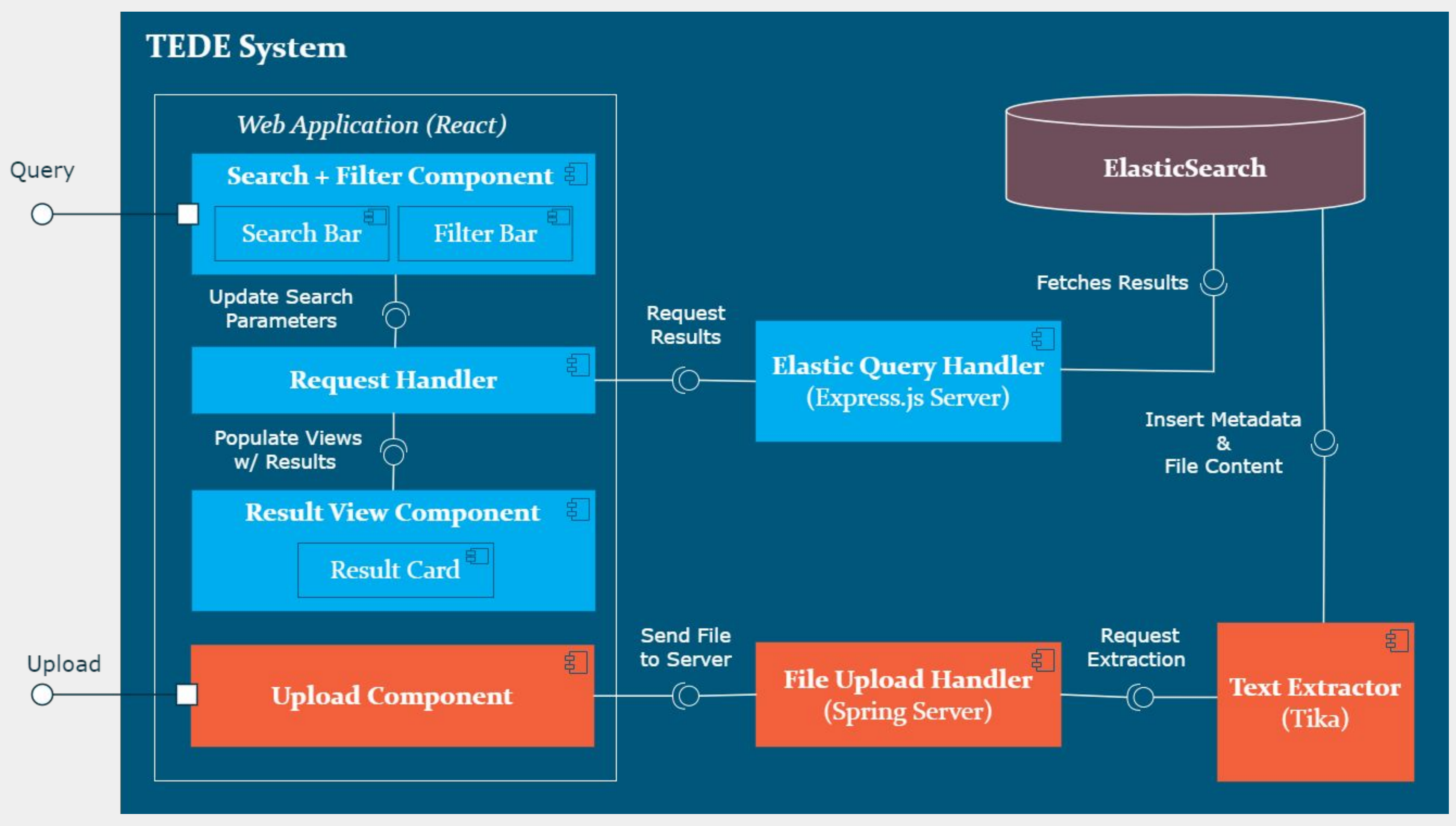
UPLOAD

Found 4 Result(s) for: "123"

.docx construction-contract.docx
/Users/brucepro/Downloads
2 matches: keyword = '123' fileType = 'Microsoft Word document (.docx)'

.pdf construction-contract.pdf
/Users/brucepro/Downloads
2 matches: keyword = '123' fileType = 'PDF (.pdf)'

Web Application Screenshots



Context Diagram

Technology Stack

Frontend

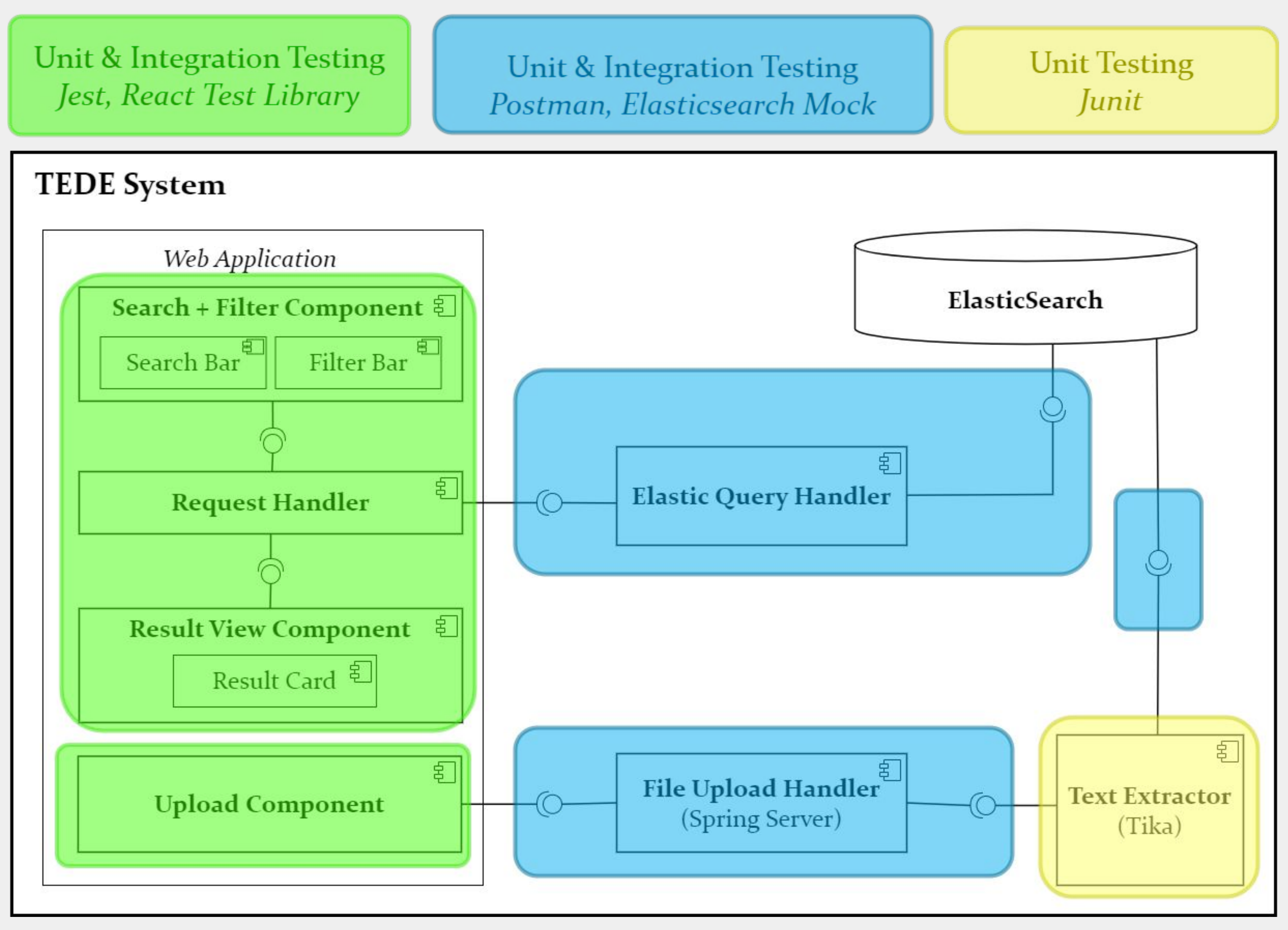
ReactJS Javascript, CSS

Backend

Spring Apache Tika Elasticsearch

Testing

- Interface Testing:
  - Elasticsearch API for inserting and querying data
  - Apache Tika API for extracting relevant metadata
- Regression Testing
  - Implement features incrementally with unit testing
  - Use Git for code management/version control
- End-to-End Testing
  - Successfully perform round trip:
    - Upload doc -> Extract data -> Send data to Elasticsearch -> Query data from new doc
- Acceptance Testing
  - Demo progress and receive feedback from client as we progress
  - Provide Minimal Viable Product(MVP) based on requirements


Unit & Integration Testing
Jest, React Test Library
Unit & Integration Testing
Postman, Elasticsearch Mock
Unit Testing
Junit
TEDE System
Web Application
Search + Filter Component
Search Bar
Filter Bar
Request Handler
Result View Component
Result Card
Upload Component
ElasticSearch
Elastic Query Handler
File Upload Handler (Spring Server)
Text Extractor (Tika)