Text Extraction from Documents into Elasticsearch

DESIGN DOCUMENT

sddec22-19

Client: Buildertrend
Advisor: Goce Trajcevski

Team Members - Roles:
Bruce Bitwayiki – Backend
Jared Hayashi – Backend
Rushal Sohal – Frontend
Tiffany Mayberry – Frontend

sddec22-19@iastate.edu
sddec22-19.sd.ece.iastate.edu

Revised: April 24, 2022 / V1

Executive Summary

Development Standards & Practices Used

IEEE-29119: We will write Unit Tests for our web application to ensure functionality in each separate piece.

IEEE-12207: We will follow the standard software lifecycle process.

IEEE-7.8: Code of ethics: We will follow ethical development practices.

Summary of Requirements

The application should...

1. be a desktop or web application (constraint)
2. utilize Elasticsearch to store the metadata (constraint)
3. be able to extract and index metadata from several different file types
4. able to support jpg, png, pptx, pdf, docx, xlsx, xls, and txt files
5. match search keywords with text fragments from the files
6. provide the ability for users to upload supported file types

Applicable Courses from Iowa State University Curriculum

- ComS 227, 228, & 327 - Basic programming skills
- ComS 309 - Project development skills
- ComS 319 - User Interface design
- SE 329 - Project management skills
- ComS 363 - Database management skills
- ComS 409 - Requirements discovery

New Skills/Knowledge acquired that was not taught in courses

- React framework
- Working with a client
- Elasticsearch setup

Table of Contents

1. Team **7**

1.1 Team Members 7

1.2 Required Skill Sets for Your Project 7

1.3 Skill Sets covered by the Team 7

1.4 Project Management Style Adopted by the team 7

1.5 Initial Project Management Roles 7

2. Introduction **7**

2.1 Problem Statement 7

2.2 Requirements & Constraints 8

2.2.1 Requirements 8

2.2.2 Constraints 9

2.2.3 Stretch Requirements 9

2.3 Engineering Standards 9

2.4 Intended Users and Uses 9

2.4.1 Use Case One 9

2.4.2 Use Case Two 9

3. Project Plan **11**

3.1 Project Management/Tracking Procedures 11

3.2 Task Decomposition 11

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria 12

3.3.1 Milestones 12

3.3.2 Metrics of Interest 13

3.3.3 Evaluation Criteria: 13

3.4 Project Timeline/Schedule 15

3.5 Risks And Risk Management/Mitigation 16

3.5.1. Planning Risks 16

3.5.2. Develop Web Application User Interface (UI) Risks 16
3.5.3 Develop REST API Risks 16
3.5.4 Setup Elasticsearch Node Risks 16
3.5.5 Develop File Text Extractor Risks 17
3.5.6 Application Testing Risks 17
3.6 Personnel Effort Requirements 17
3.7 Other Resource Requirements 18
4 Design **18**
4.1 Design Context 18
4.1.1 Broader Context 18
4.1.2 User Needs 20
Builders & Contractors 20
Homeowners 20
Buildertrend Staff 20
4.1.3 Prior Work/Solutions 20
4.1.4 Technical Complexity 21
4.2 Design Exploration 22
4.2.1 Design Decisions 22
4.2.2 Ideation 22
4.2.3 Decision-Making and Trade-Off 23
4.3 Proposed Design 23
4.3.1 Design Visual and Description 23
4.3.2 Functionality 25
4.3.3 Areas of Concern and Development 25
Uploading docs 25
UI/UX 25
Accuracy of results 26

Query speed 26

Scalability of doc types 26

4.4 Technology Considerations 26

4.5 Design Analysis 27

4.6 Design Plan 27

5 Testing **28**

5.1 Unit Testing 28

5.2 Interface Testing 29

5.3 Integration Testing 29

5.4 System Testing 29

5.5 Regression Testing 30

5.6 Acceptance Testing 30

5.7 Security Testing 30

5.8 Results 30

6 Implementation **31**

7 Professionalism **33**

7.1 Areas of Responsibility 33

7.2 Project Specific Professional Responsibility Areas 34

Work of Competence – High 34

Financial Responsibility – Medium 34

Communication Honesty - High 34

Health, Safety, Well-Being - Low 35

Property Ownership - High 35

Sustainability - Low 35

Social Responsibility - Medium 35

7.3 Most Applicable Professional Responsibility Area 35

8 Closing Material **36**

8.1 Discussion 36

8.2 Conclusion 36

8.3 References 36

8.4 Appendices 36

8.4.1 Team Contract 37

Team Members 37

Team Procedures 37

Participation Expectations 37

Leadership 37

Collaboration and Inclusion 38

Goal-Setting, Planning, and Execution 38

Consequences for Not Adhering to Team Contract 38

List of figures/tables/symbols/definitions

Figure 1: Use Case Diagram 10
Figure 2: User Case Persona Diagram of Buildertrend customers 10
Table 1: Task Decomposition 12
Figure 3: Milestone Gantt Chart 12
Table 2: Milestone Decomposition 13
Figure 4: Schedule Gantt Chart 15
Table 3: Personal Efforts Requirements 18
Table 4: Responsibility Considerations 19
Figure 5: A General Framework of IR Systems 21
Table 5: Decision Trade-off Considerations 23
Figure 6: TEDE Context Diagram 24
Figure 7: TEDE Component Diagram 25
Figure 8: Testing Section of Gantt Chart 30
Figure 9: Component Diagram with Testing Strategy 31
Figure 10.1: Mock-Up of the Results View of the User Interface 32
Figure 10.2: Mock-Up of the Upload File View of the User Interface 32
Figure 10.3: Mock-Up of the Filters View of the User Interface 33
Table 6: ACM Areas of Responsibilities 34

1. Team

This section describes the basic overview of our team and skills required for the project.

1.1 Team Members

- Bruce Bitwayiki
- Jared Hayashi
- Rushal Sohal
- Tiffany Mayberry

1.2 Required Skill Sets for Your Project

- Backend and database knowledge
- Frontend development for UI
- Communication and documentation
- Familiarity with Agile & waterfall methodology

1.3 Skill Sets covered by the Team

- Backend and database knowledge - Jared & Bruce
- Frontend development for UI - Tiffany & Rushal
- Communication and documentation - All
- Familiarity with Agile and waterfall methodology - Tiffany, Jared & Bruce

1.4 Project Management Style Adopted by the team

Agile + waterfall methodologies and bi-weekly sprints. More details are in the Project Plan.

1.5 Initial Project Management Roles

- Bruce Bitwayiki - Backend documenter, Backend Architecture Design
- Jared Hayashi - Client communication, Backend Architecture Design
- Rushal Sohal - Documentation and reports, Frontend and UI
- Tiffany Mayberry - Faculty Advisor communication, Frontend and UI

2. Introduction

We now describe the problem and discuss the requirements and constraints.

2.1 Problem Statement

Construction managers and workers using Buildertrend's software are unable to quickly find information inside documents they have uploaded quickly. Information may be spread across multiple files over a period of time, making it difficult and time-consuming to manually search through the documents. The time spent looking for the file could be better spent doing other, more important, tasks. A new way to search and store the file's content needs to be added to Buildertrend's application. This will enable the construction managers and workers to spend the time they would have spent looking for the file on more productive tasks.

2.2 Requirements & Constraints

We now enumerated the requirements, constraints, and stretch requirements. Stretch requirements are outside of the basic functionality of the project, but if time allows, some may be included in the final deliverables.

2.2.1 Requirements

1. Functional Requirements:
    - 1.1. The application shall extract and index metadata from several different file types
    - 1.2. The application shall support jpg, png, pptx, pdf, docx, xlsx, xls, and txt files
    - 1.3. The application shall match search keywords with text fragments from the files
    - 1.4. The application shall allow users to upload supported file types
    - 1.5. The application shall provide search filters such as file type, file name, author, and date
2. User Interface and Aesthetics:
    - 2.1. The application shall have a simple UI that allows users to search files that are indexed
    - 2.2. The application shall display the results of a search query (ordered by best fit)
    - 2.3. The application shall indicate if no search results were found
    - 2.4. The application shall display the filename and path of the files that match the search criteria
    - 2.5. While the application is searching for results, the application shall indicate the results are loading
3. Resource Requirements:
    - 3.1. The application shall be easily deployable to a server(s) (potentially provided through Iowa State in the initial stages of development)
4. Environmental Requirements:
    - 4.1. All users will have the same view and permissions
    - 4.2. No authentication and authorization will be needed for this project

2.2.2 Constraints

1. The query keyword shall be up to 140 characters long
2. The application shall be a desktop or web application

3. The application shall be readable in windows larger than 500 by 600 size
4. The application shall utilize Elasticsearch to store the metadata
5. The application shall return a result within 10 seconds
6. If the application does not return a result within 10 seconds, the application shall limit the number of results returned or indicate an exception

2.2.3 Stretch Requirements

1. The application shall display a preview of selected files
2. Display search statistics such as load time and the number of keywords/indexes for specific files
3. The application shall limit the number of results displayed for performance
4. The application shall run and interact with Google Cloud Platforms such as Cloud Functions, Cloud Run, and Cloud Storage
5. The application shall automatically index uploaded files
6. The application shall include a feature to index a list of documents on demand
7. When a file is deleted from the server, the application shall delete the metadata and index from Elasticsearch

2.3 Engineering Standards

- **IEEE-29119**: We will write Unit Tests for our application to ensure functionality in each separate piece. This would help us ensure that the individual components function correctly.
- **IEEE-12207:** We will follow the standard software lifecycle process. This shall help us in the development of our application.
- **IEEE-7.8:** Code of ethics: We will follow ethical development practices as it defines the core values of our team.

2.4 Intended Users and Uses

The primary users are Buildertrend's customers and support staff. Below are some use cases:

2.4.1 Use Case One

Users will interact with the system via a web application. The web application will display a search bar and filters that the user can use to query for files. The results would be ordered by best fit, and the user can select the desired file (based on filename and path).

2.4.2 Use Case Two

Support staff shall have additional access to reports and queries made by regular users.

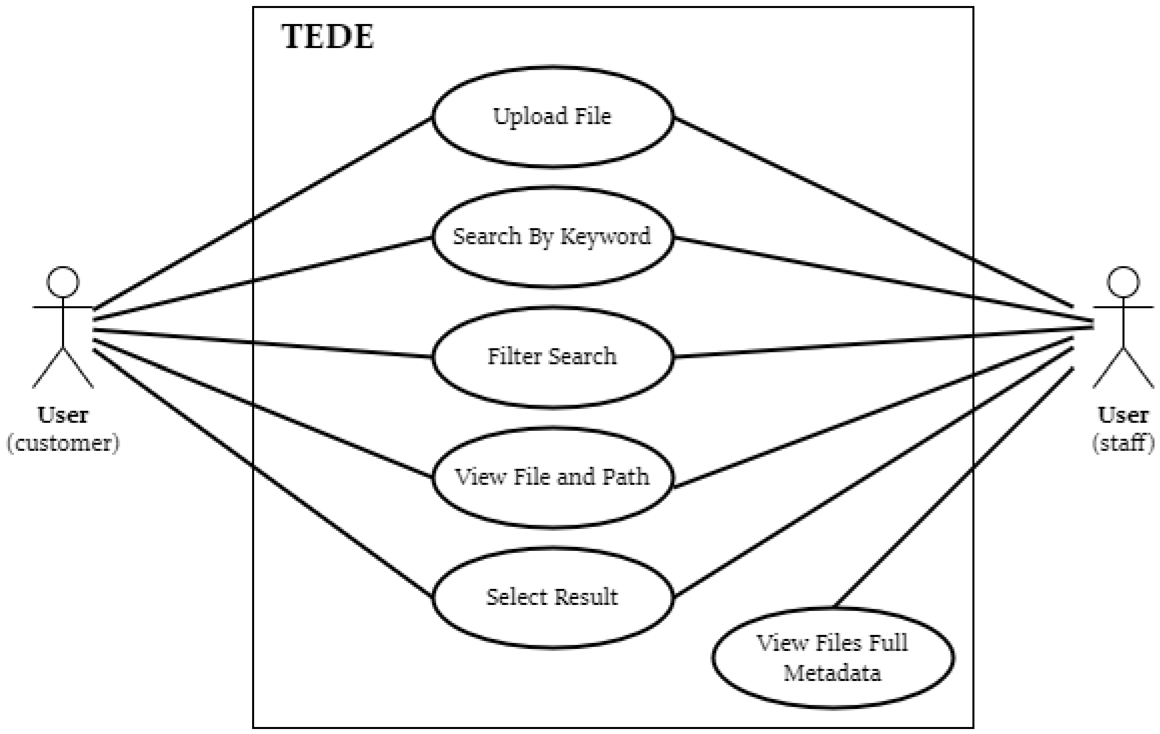
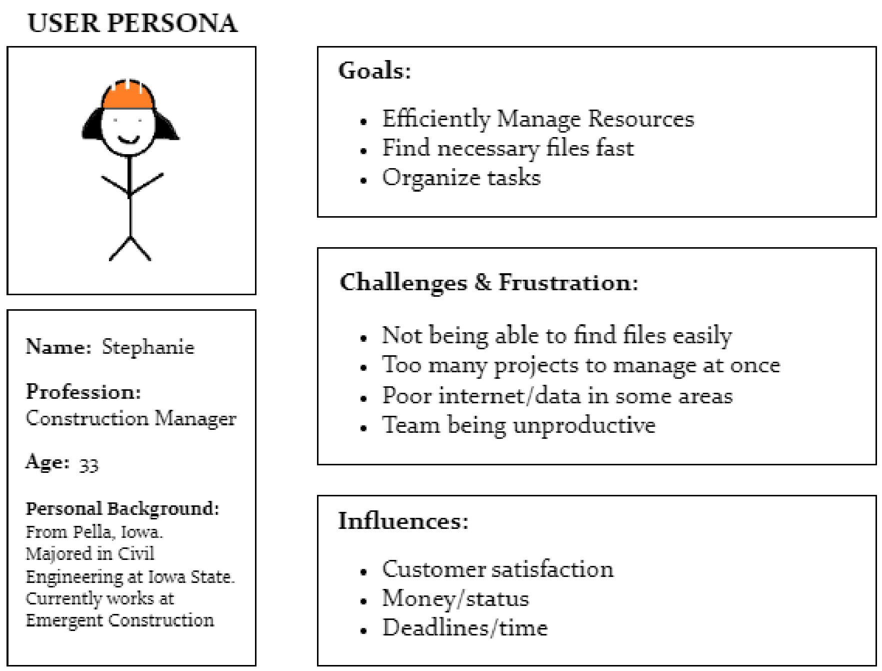


Figure 1: Use Case Diagram



USER PERSONA

Name: Stephanie

Profession: Construction Manager

Age: 33

Personal Background: From Pella, Iowa. Majored in Civil Engineering at Iowa State. Currently works at Emergent Construction

Goals:

- Efficiently Manage Resources
- Find necessary files fast
- Organize tasks

Challenges & Frustration:

- Not being able to find files easily
- Too many projects to manage at once
- Poor internet/data in some areas
- Team being unproductive

Influences:

- Customer satisfaction
- Money/status
- Deadlines/time

Figure 2: User Persona Diagram of Buildertrends customer base

3. Project Plan

We now discuss our project management plan and describe the breakdown of project tasks, milestones, and schedule.

3.1 Project Management/Tracking Procedures

We plan on using a combination of Waterfall & Agile management styles. The linear approach of the Waterfall process will allow us to take account of the initial requirements and help us in developing the application life cycle. The flexibility of the Agile Application Development process will allow our team to easily adapt to changes regarding features (documents support), timing constraints, new technologies, or, in an (unlikely but) extreme case, the overall scope of the project.

Our group plans on using a combination of GitLab and a shared Google Drive for this project. GitLab will be used to keep track of tasks and a repository for code. Our primary means of communication between team members will be through Discord. Microsoft Teams will be used for primary communication with BuilderTrend. Email will be used for communication with our faculty advisor.

3.2 Task Decomposition

The breakdown of the project's tasks is described in the following table.

Task	Task Description
1. Project Planning	Flush out requirements received from the client and figure out our team dynamic.
2. Develop Web Application User Interface (UI)	Create a UI for users to search using keywords and filters and see the results of related files. UI will also need to allow the uploading of files.
3. Develop REST API	Design and implement application functions that communicate between the frontend and Elasticsearch and between the text extractor and Elasticsearch.
4. Setup Elasticsearch Node	Setup Elasticsearch node to store uploaded data.
5. Develop File Text Extractor	Design and implement a way to extract text and metadata from various files.

6. Application Testing	Test individual components of the web application, text extractor, and REST API. See Section 5 for more details.

Table 1: Task Decomposition

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

We now discuss our project milestones and how we plan on measuring the success of the project.

3.3.1 Milestones

The project milestones for each task can be seen in a Gantt chart shown in Figure 3. The corresponding descriptions of the activities are provided in Table 2.

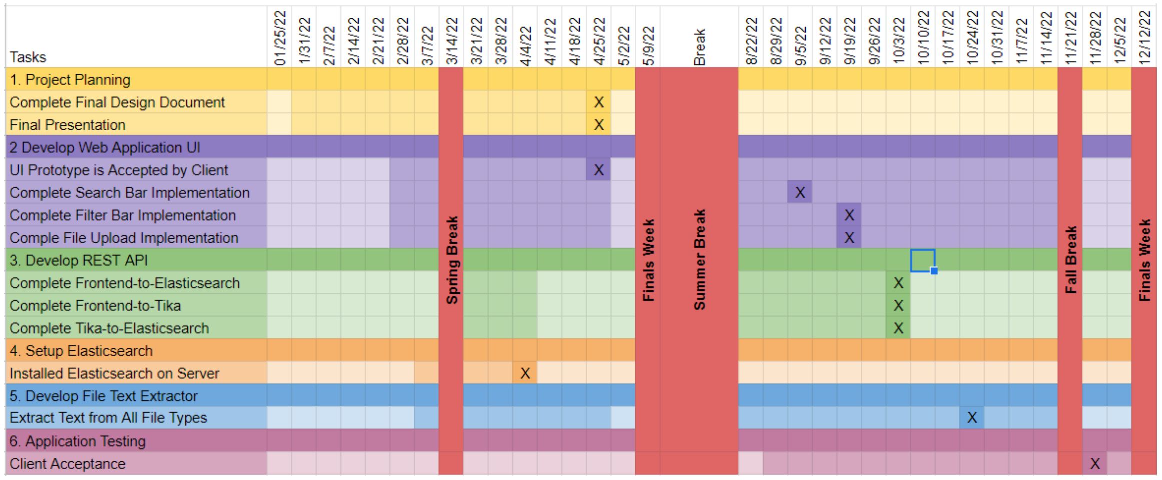


Figure 3: Gantt chart for milestones

Task	Milestone Description
1. Project Planning	The main milestones for this phase are completing this document and presenting our project. This will allow us to have a solid base when beginning implementation phases after the summer break.
2. Develop Web Application User Interface (UI)	The first milestone for this phase is to complete a UI prototype and present it for approval from our client, Buildertrend. This enables us to verify the requirements and what they are expecting from the final deliverable.

	The other milestones of this phase are implementation end goals for the core components of the UI: Search Bar, Filter Bar, and File Uploader. The milestones are concurrent due to how closely related the functionality of the components are.
3. Develop REST API	The milestone of this phase is the complete functionality of the interfaces between the core components: Web Application to Elasticsearch, Web Application to Text Extractor, and Text Extractor to Elasticsearch. More information can be found in Section 5.2 (Interface Testing).
4. Setup Elasticsearch Node	Elasicsearch is a core system of our application. The main milestone for this phase is a successful installation of an Elasticsearch node on our development server.
5. Develop File Text Extractor	The main milestone for this phase is the ability of our application to extract text and metadata from the required files (see requirement 1.2) with at least 90% accuracy.
6. Application Testing	The main milestone of this phase is to receive approval from the client. More information about our testing plan can be found in Section 5.

Table 2: Milestone Decomposition

3.3.2 Metrics of Interest

- User Interface Usability
- Text extraction accuracy
- Query speed and accuracy

3.3.3 Evaluation Criteria:

- User Interface Usability
    - Responsiveness of UI
    - Good user experience (get feedback from different users)
- Text extraction accuracy
    - Accuracy of the results with keyword matching
    - Meets user expectations

- Query speed and accuracy
    - Time needed for the search
    - Our implementation vs. brute force

3.4 Project Timeline/Schedule

Our project schedule is broken down as illustrated in the Gantt chart shown in Figure 4.

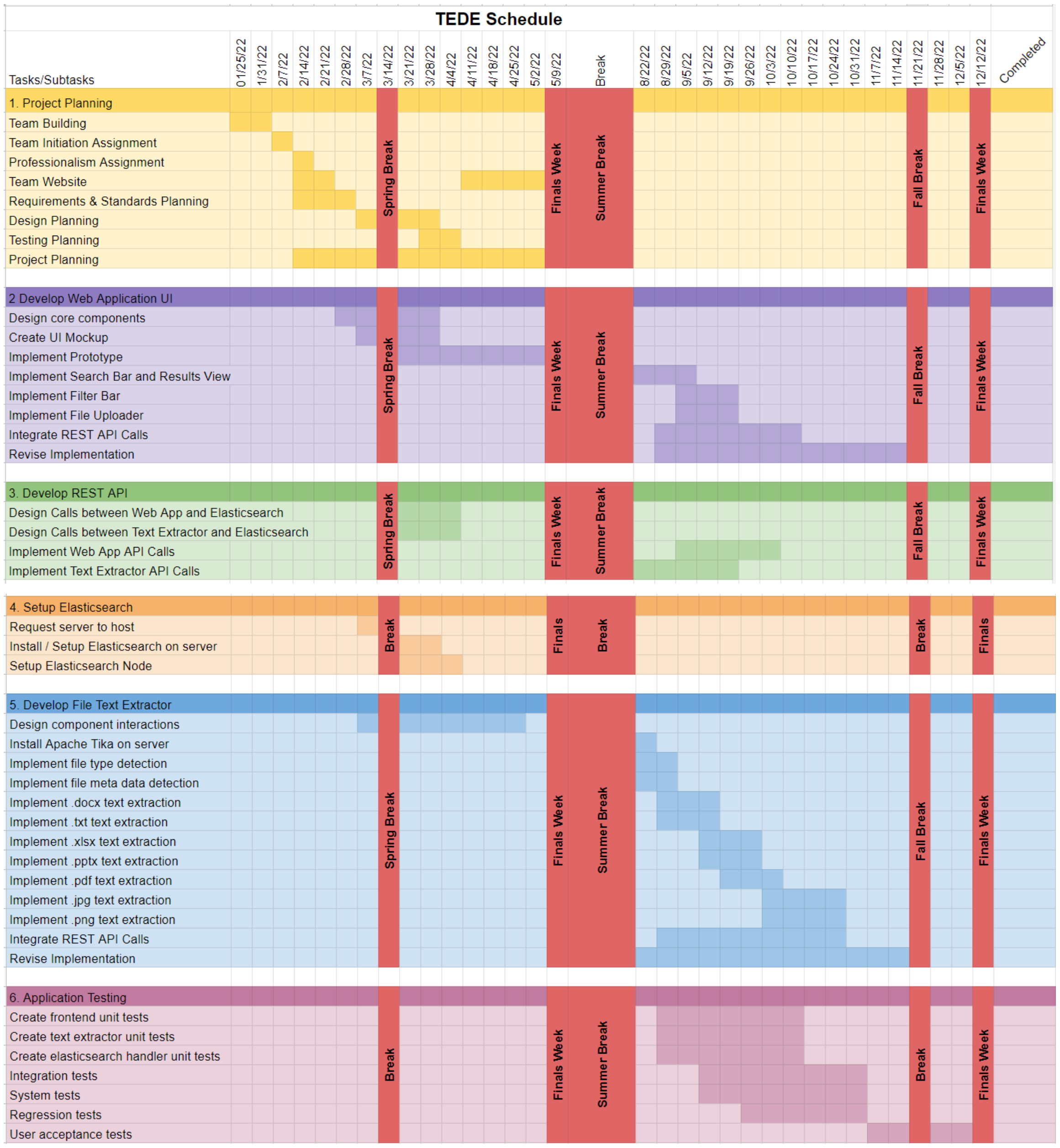


Figure 4: Gantt chart for schedule

3.5 Risks And Risk Management/Mitigation

In this section, we elaborate on the risks related to each project task and detail our plan to mitigate them.

3.5.1. Planning Risks

Risk probability: 0.3

The major risk involved with our project planning is missing requirements/constraints or having to make major changes to our project requirements and constraints. Our initial project document did not contain a lot of detail regarding the project, so we had to invest a lot of time asking our client for the requirements before making any major design decisions. If we end up missing a major requirement, it could result in us having to invest a lot of extra time fixing the issue.

3.5.2. Develop Web Application User Interface (UI) Risks

Risk probability: 0.3

One big risk for developing our frontend application is having problems integrating the application with the backend and retrieving data from it. This could result from several factors - one being an incompatibility between languages.

Risk probability: 0.1

Another risk involving the frontend interface is, that with the change or addition of requirements, we may need to redesign the user interface or make amends to it. If this occurs, it will not be a major issue since our design for the UI is relatively simple and can be redesigned without much effort.

3.5.3 Develop REST API Risks

Risk probability: 0.4

Since the REST API will be handling a lot of the communication between our applications, there is a lot of room to run into issues. Whether it be between the Web Application and Elasticsearch or Elasticsearch and Apache Tika, it is crucial that our implementation is carefully designed to ensure that we will not run into any major issues that will result in a large setback.

3.5.4 Setup Elasticsearch Node Risks

Risk probability: 0.2

We should not run into many issues with this. The major risk would be having technical problems setting the node up on the server, especially if it's from their side. In that case, we might have to contact Elasticsearch to resolve it.

3.5.5 Develop File Text Extractor Risks

Risk probability: 0.4

The file extractor is a major part of the project, and we could run into multiple issues. If the extractor is not able to detect the file type, get metadata (within our accuracy %), or format and insert files into Elasticsearch, we will have to re-design from the very basics and dive deep down to solve the issue. As such, we need to work closely on designing this part of the project.

3.5.6 Application Testing Risks

Risk probability: 0.1

There isn't much risk associated with testing. The biggest risk would be issues in our test code/implementation.

3.6 Personnel Effort Requirements

A time estimate to complete each task is included in Table 3 and a brief description of our reasoning.

Task Name	Est. Hours	Explanation
1. Project Planning	75	Including most of the assignments for the first semester, research for learning how to utilize the APIs, and asking our client questions. The estimated time should be between 75 and 100 hours.
2. Develop Web Application User Interface (UI)	50	Our frontend requirements are not that rigorous, so it should take less time relative to our backend applications
3. Develop REST API	100	The REST API will interface with all our applications and can result in many issues transmitting messages between our applications, so the most time will be spent here
4. Setup Elasticsearch Node	75	Elasticsearch is one of the core parts of the backend, so a large amount of time will be

		allocated to developing and refining it for maximum performance
5. Develop File Text Extractor	75	Apache Tika is the other core part of the backend. We need to ensure we get the correct info from the files as efficiently as possible.
6. Application Testing	25	We will need to spend some time testing some edge cases and the entire system's performance. Although there should not be too many test cases depending on how many subsystems, we implement from the stretch goals

Table 3: Personal Efforts Requirements

3.7 Other Resource Requirements

There should not be any need for additional resources for our project apart from the server. Since it is a software project, we won't need any parts or materials. However, we might need some software licenses in the future.

4 Design

We now describe the context of our project at different scales. We also describe our design process/reasoning and present our proposed design.

4.1 Design Context

In this section, we elaborate on the context of our project at a broad scale and a user scale. We also discuss the current solution of Buildertrend's search feature and the technical complexity of our project.

4.1.1 Broader Context

Buildertrend provides construction project management software and solutions for construction companies, homeowners, and builders. Our search feature will enable them to save time and increase productivity by reducing the time spent searching through various documents.

Area	Description	Project Relevance
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or maybe indirectly affected (e.g., a solution is implemented in their communities)	Our project tries to make information inside uploaded files more accessible to the different users. It also allows users to find all the files relevant to their search and cross-reference them. The goal is to save the user's time.
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	The main contextual groups are professionals in the construction industry. This project will enable them to have much greater search capabilities in terms of not only technical specifications but also contextual bindings, user requests, etc. Our project does not conflict with any ethnic or racial group and is transparent to different groups. However, the product is for now limited to the construction industry.
Environmental	Direct and indirect environmental impacts such as deforestation or unsustainable practices related to materials manufacture or procurement.	By having the capability and access to different search filters, construction workers can gain environmental awareness (e.g., municipality constraints on forestry practices near construction sites). Should future software versions scale beyond a threshold, resources such as energy consumption may be a concern.
Economic	Possible financial viability of your product within the team, company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	The reduced search time will allow Buildertrend customers to save on labor/billed time costs and work on more projects. Giving them an edge over the competition and attracting more customers for Buildertrend increasing sales and profits. In

		addition, other constraints with financial impacts may be searchable.

Table 4: Responsibility Considerations

4.1.2 User Needs

Our application has three different user groups; their needs are defined below.

Builders & Contractors

Builders and workers need a quick and efficient method to search for information on site, such as measurements, designs, orders, etc., to complete the project according to requirements.

Homeowners

Homeowners need a way to search for documents regarding the construction of their home at any time. The documents could include important information such as previous contracts, costs, or design parts/specifications as necessary in order to make a decision regarding the project.

Buildertrend Staff

Buildertrend staff need a way to support the users on any potential issues they face with the search feature. They will have access to the full entries saved in Elasticsearch to understand why certain results were returned for a query.

4.1.3 Prior Work/Solutions

Text extraction from files has been explored in various solutions. Other solutions typically provide support to a subset of file types our solution aims to provide support to. These solutions aim to solve the common goal of information retrieval systems (IRS) which is “finding relevant information or a document that satisfies user information needs” (Sharma). To achieve this goal, IRS’s usually implement three processes: 1. Indexing, 2. Filtering, 3. Searching (Sharma). These three steps can be shown in the following diagram (Figure 5) from A Survey on Information Retrieval Models, Techniques, and Applications and is the common workflow of other solutions.

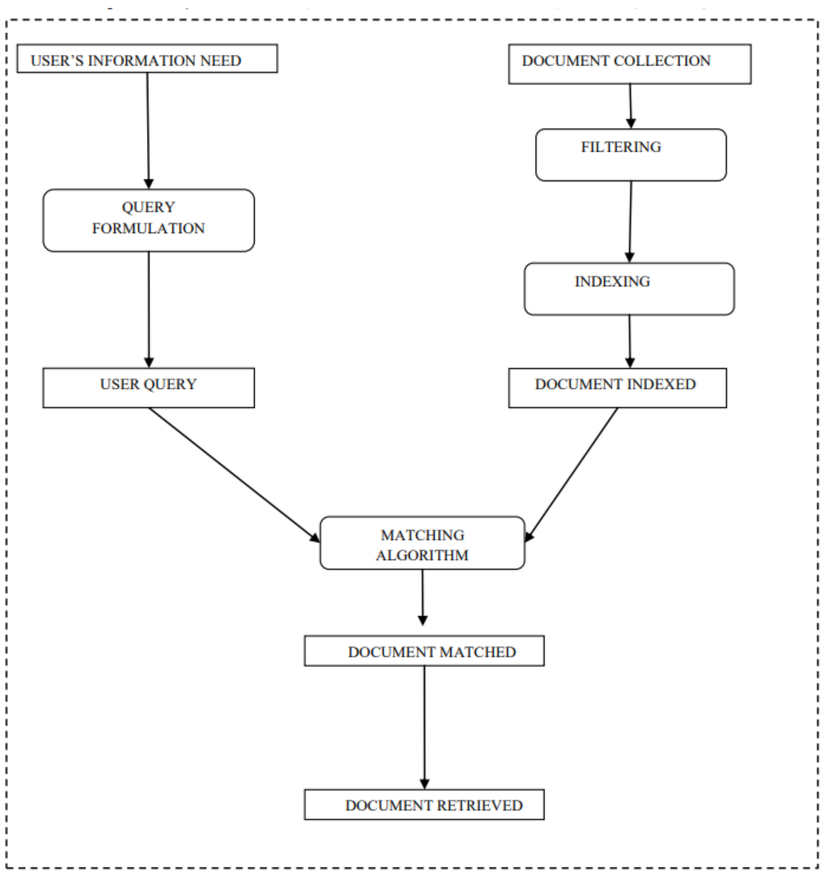


Figure 5: A general framework of IR System

Buildertrend has a current implementation of Elasticsearch that searches files only by file name. There is also an implementation that allows staff members to search for files based on their SQL representation.

Our solution will follow a similar flow as other IRS's (as shown above in Figure 5). It will allow Buildertrend to integrate the tool into their existing application, which would not be possible with other solutions. It will also support the file types that are relevant to Buildertrend's work. It is also important to note that other solutions utilize different database backends; however, we are constrained to using Elasticsearch for storing our data.

4.1.4 Technical Complexity

The design consists of several subsystems of various complexity, each requiring background knowledge and/or additional research. These subsystems include:

- **Searching algorithm:** Implement a search algorithm to find a return the best-fit result within 10-seconds.
  - *Requirements:* Algorithm analysis and implementation, CI/CD

    - How can we maximize query result precision and recall?
- **Simple frontend UI/UX:** User-friendly web app with the ability to navigate, search, and upload files
    - *Requirements:* Web development, React, CI/CD
- **Backend architecture:** Use decided tech stack to implement a system to extract, and index provided documents to be served to the frontend UI when queried.
    - *Requirements:* Database management, Tika, Elasticsearch, Investigate the impact of "mediator" that translates the text from different file types
- **Test cases:** Design proper testing suite to check for project requirements as well as optimal functionality
    - *Requirements:* Experience with previous testing libraries(e.g., JUnit)
- **Project management:** We must combine all components of the project which requires the frontend, backend, and other components to be properly implemented and tested to enable them to communicate seamlessly and efficiently.

4.2 Design Exploration

Our client gave us flexibility with some of the technologies we could use. We elaborate on the reasoning behind our design decisions in the sections below.

4.2.1 Design Decisions

We elected to use the following technology stack to accomplish our task following research and interactions with the client:

- Frontend Framework (React)
- Backend Communication (REST)
- Backend APIs (Apache Tika, Elasticsearch)
- Application type (Web application)

4.2.2 Ideation

We were given the flexibility to use various tools and technologies for this project by the client. Some of the options considered for the tech stack include:

- React JS (frontend/UI)
- Apache Lucene (file search)
- Apache Tika (type detection and data extraction)
- Elasticsearch (file indexing and search)

Upon researching and exploring various options, our decisions for the tech stack was based on the given requirements and basic functionality as this project would serve as a standalone feature to be implemented in another application at a later date.

4.2.3 Decision-Making and Trade-Off

Tools and Technologies	Pros	Cons
Frontend Framework (React)	- Simple to learn - Well-supported - Can easily be expanded to add more functionality	- Unfamiliar to group members
Backend Communication (REST)	- Easy to implement - Logistically simple - Group members already familiar	- Possibly difficult to pick up for new developers
Backend API's (Apache Tika, Elasticsearch)	- Scalable - Fast indexing and search responses	- (Elasticsearch) Not atomic (can read and write indexes at the same time)
Application type (Web application)	- Easily accessible - Compatible on any platform (e.g., Windows, Mac, Linux)	- Accessible to anybody with URL unless secured (user login, network-specific accessibility)

Table 5: Decision Trade-off Considerations

4.3 Proposed Design

We now discuss the proposed project design using several diagrams and describe how it meets the requirements.

4.3.1 Design Visual and Description

The following diagram, Figure 6, shows a high-level overview of the interactions of our application with relevant external systems. The following are brief explanations of the numbered interactions:

1. Receives a list of files relevant to the query made and displays them to the user
2. Triggers a query using a keyword filter
3. Add other filters to a query
4. Upload a file to have the text content extracted
5. Store a copy of the upload file on server

6. Returns a list of files from Elasticsearch that match the query filters
7. Query the data stored in Elasticsearch that matches the given filters
8. Stores the metadata and content of the uploaded file into Elasticsearch

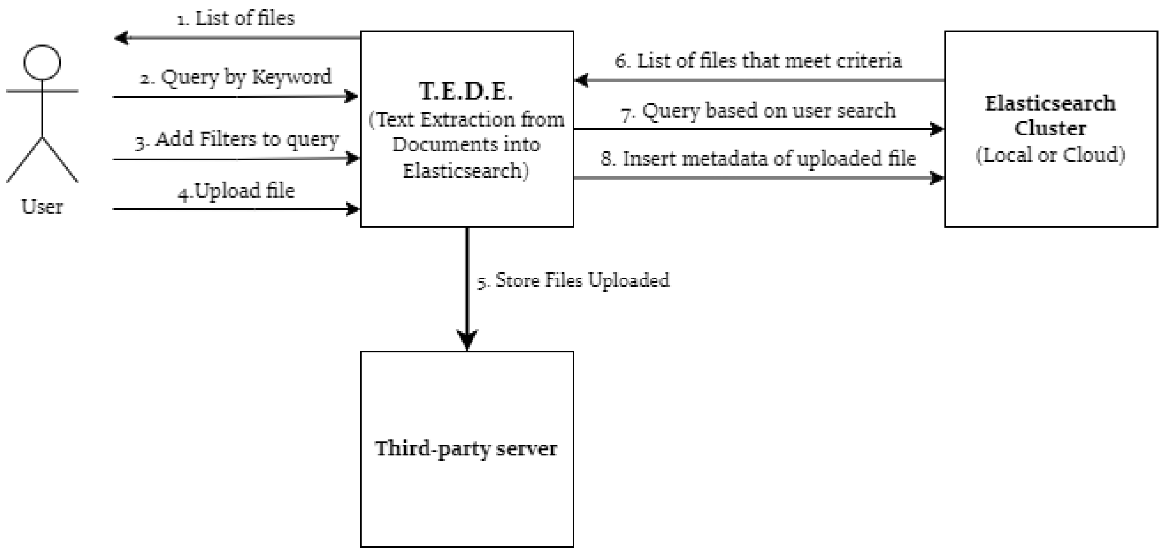


Figure 6: TEDE Context Diagram

The following diagram, Figure 7, shows a breakdown of the components of TEDE. The components are responsible for the following responsibilities:

- **Web Application System:**
    - *Search & Filter Bar:* Handles the applying keyword and other filters from the user
    - *Upload Cache:* Initializes an upload of a file to store the file on the server and triggers a text extraction/ index of the file's contents
- **Apache Tika System:**
    - *Tika Text Extractor:* Extracts the metadata and text content from the given file and calls the Tika Handler to index/store the data
- **Elasticsearch System:**
    - *Web-App Handler*: Creates a query call from user input and retrieves data from the Elasticsearch cluster
    - *Tika Handler:* Inserts data into the Elasticsearch cluster

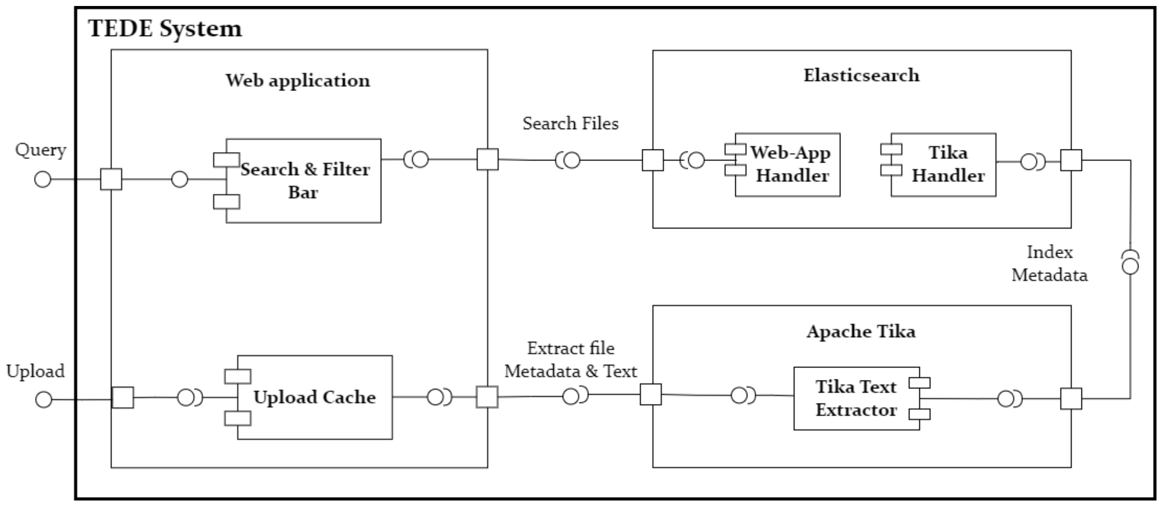


Figure 7: TEDE Component Diagram

4.3.2 Functionality

Users of our application will be able to search using keywords and search filters to find relevant documents. It will also be able to extract text from uploaded files and store the important information from the file. The application would support various file types like txt, docx, pdf, pptx, etc.

Our current design addresses these function needs for the specified file type but may need to be adjusted based on what is attainable. For the UI/UX requirements, we might need further client approval.

4.3.3 Areas of Concern and Development

We have the following concerns about our current design and our plan of handling them.

Uploading docs

Concern: No clear specification of what this feature entails.

Plan: The plan is to be able to upload document types that we can search. As we scale our search document types, we also diversify our upload document types.

UI/UX

Concern: No detailed specifications for this one either. We will have to reiterate our design until we get approval.

Plan: Propose a UI design, receive feedback, and reiterate until final approval.

Accuracy of results

Concern: No qualitative measures were specified by the client.

Plan: For now, we will just follow the traditional measure - if the result fits the user's need or what the user was actually looking for.

Query speed

Concern: The client has not specified anything yet. Nonetheless, we want to return our results within 10s. Anything longer than this, we will have to rework and optimize our algorithm.

Plan: Develop a search algorithm and optimize it until an acceptable speed is attained.

Scalability of doc types

Concern: The list of document types is quite open-ended. Less of concern and more of a development.

Plan: We would be starting with txt and docx files. As we find success, we will progress to other types like pdf, ppt, jpeg, etc.

Overall, most of our concerns can be resolved with better client specifications.

4.4 Technology Considerations

- **Elasticsearch** - Searching and Indexing
  - *Pros:*
    - Relatively easy to learn and use with good documentation
    - Good scalability and easily deployed to the cloud
    - Open source and free
  - *Cons:*
    - Nobody in the group has any experience with it
  - *Solutions and Design Alternatives:*
    - We have allocated more time to research and learn Elasticsearch to ensure that we have plenty of time to complete our implementation
    - Considered Apache Solr, but Elasticsearch seems to be more feature-rich and better supported
- **Apache Tika** - Text Extraction
  - *Pros:*
    - Fast and efficient processing
    - Natively deployed in Java which we are familiar with

        - Able to extract text from images with Optical Character Recognition (OCR) from Tesseract
    - *Cons:*
        - No one in the group has previous experience with it
        - Not very well documented
    - *Solutions and Design Alternatives:*
        - Similarly to Elasticsearch, we have allocated more time to research and familiarize ourselves with Tika. Since our use case is very simple, we should not have to spend too much time learning how to use Tika.
- **React JS** - Frontend
    - *Pros:*
        - Fast and scalable library
        - Components are easily redesigned for revisions specified by the client
        - Cross-platform compatible
    - *Cons:*
        - Only some members of our group are familiar with React
    - *Solutions and Design Alternatives:*
        - We divided up our group into teams for the frontend and backend. This should minimize the time spent learning how to use React.
        - Considered desktop application using Electron, but it does not seem to have many benefits since we can not support the application when offline. It would require more time to develop, and users would have to take time to download the program.

4.5 Design Analysis

While we are before the implementation stage of this project, we have done experiments on the different tools and are confident that our design will work. As we begin the implementation phase, we may tweak the different components for them to integrate. The most important part of our design is to adapt to the needs of Buildertrend.

4.6 Design Plan

Our design process will follow our current schedule o utlined in Section 3.4. We will be developing most of the components concurrently and getting them approved during our weekly meetings with the client. We allocated some of the time in the implementation sections to making changes and refining the components as we develop them based on the feedback we receive. We will adjust and update our schedule accordingly based on the progress that we make throughout the semester as well. We plan to finish the overall implementation before fall break. This will give us a couple of weeks to make adjustments and implement any stretch goals previously specified by the client.

5 Testing

Some of the unique testing challenges in our project include the complexity of testing communications between the frontend and backend. As well as testing the accuracy of our text extractor module. Below is our testing plan to handle these challenges.

5.1 Unit Testing

As shown in our design diagrams in section 3, there are several distinct modules in our system. Each of these modules has identifiable units of work that are planned to be a part of the final deliverable.

Below is a list of the identifiable elements:

1. **Web Application**
   a. All components and functionality displayed to the user such as:
      i. Search bar queries and filters
      ii. Proper presentation of results
      iii. Uploading of new documents
2. **Tika Text Extractor**
   a. Extraction of metadata from newly uploaded files. It should be able to handle all of the file types specified in Requirement 1.2.
   b. Be able to identify file types and determine what pieces of metadata to send to Elasticsearch. (e.g., text body for pdf, resolution for jpg, etc.)
3. **Elasticsearch Handler**
   a. Proper indexing of the metadata from Tika Text Extractor
   b. Building queries to search the Elasticsearch cluster based on inputs received from the web application

Below is a list of tools that will help us test the above elements:

1. **Web Application**
   a. We plan on using Jest and React Test Library packages for unit testing. Since they are React's recommended testing tools, they will integrate well into the project. Jest will allow us to see the code coverage of our tests and will allow us to mock data inputs. React Testing Library provides virtual DOMs for testing React components without a browser.
2. **Tika Text Extractor**
   a. We plan on using JUnit for unit testing. Since the primary package we are using to support this component is a Java package, JUnit gives us the ability to easily test our code. JUnit is also well documented and will integrate into our development environment easily.
3. **Elasticsearch Handler**

    a. Since the primary purpose of this component is to handle communication between components. See 5.2 Interface Testing for more information.

5.2 Interface Testing

There are several identifiable levels of interface testing:

1. **Web Application to Elasticsearch Handler**
    a. Is responsible for fetching and retrieving query data
2. **Web Application to Tika Text Extractor**
    a. Is responsible for sending a selected file to be extracted by the Tika Text Extractor component
3. **Tika Text Extractor to Elasticsearch Handler**
    a. Is responsible for inserting a file's metadata and text content into Elasticsearch

Elasticsearch provides a REST API for inserting data. We plan on using Postman to validate data insertions into Elasticsearch are successful. To test our components' interaction with Elasticsearch, interfaces 1 and 3, we plan on using Jest and JUnit, respectively.

5.3 Integration Testing

Integration testing will be critical for our project as we will have several components communicating with each other through various endpoints. Incremental integration testing will be the approach for this project, allowing us to test individual modules prior to integrations. It will be important to test and validate the functionality between the Elasticsearch endpoint and the tika application as well as the integration between the frontend and Elasticsearch; this will be the core of the application. Additionally, if the group reaches the stretch goal, integration of the application into a cloud-based environment would require additional integration testing as there would be additional components to take into consideration during development (VPC, Cloud DB, etc.). However, our project will primarily be focused on the local implementation prior to integrating with the cloud. As this project serves as a proof-of-concept, we anticipate that Buildertrend will also be integrating this product into their own application, implying additional testing during this stage.

5.4 System Testing

Considering that our overall system is composed of three major components with minimal communication among them, our system testing should be fulfilled by our unit and integration tests. The individual components should be tested via the unit tests to ensure that they are functional before testing the communication among them. Following the unit tests, there are three interprocess communications that are specified in the interface testing section that need to be tested. These communications will be tested using the methods specified in the integration testing section which will complete our system testing.

5.5 Regression Testing

It is important that new additions made to the system should not break it and instead enhance the user experience. Our most essential features are web application, Elasticsearch handler, and tika text extractor, and we need to make sure they are robust enough. We would do enough unit and integration testing on our critical components and then add a new feature and see if we still get the same results. This way, we can have our core system to be robust and can easily figure out what new feature is breaking the old functionality.

5.6 Acceptance Testing

Requirements agreed upon by the client and the team should be fully visible and usable in our minimal viable product - the search web application. We plan on having several iterations of acceptance testing as we expect to receive feedback from the client while making progress toward the final deliverable. The final iteration will include documentation in order to aid users and developers, as well as the working application based on the feedback received, showcasing the requirements and features.

5.7 Security Testing

Security testing is now more important than ever, however, it will not be applicable to our project as it's just a proof of concept. We expect Buildertrend to work on the security when they integrate it into their system.

5.8 Results

We won't have any results this semester, as we are not performing any tests. Next semester, we will follow the testing schedule in Figure 8. An overview of how the testing strategies overlay the project's components are shown in Figure 9.

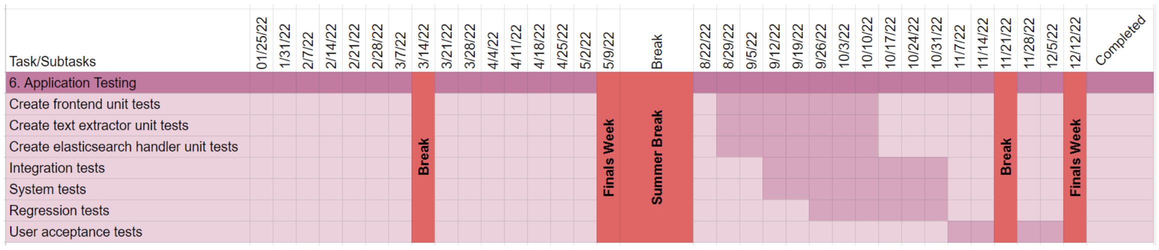


Figure 8: Testing section of Gantt chart

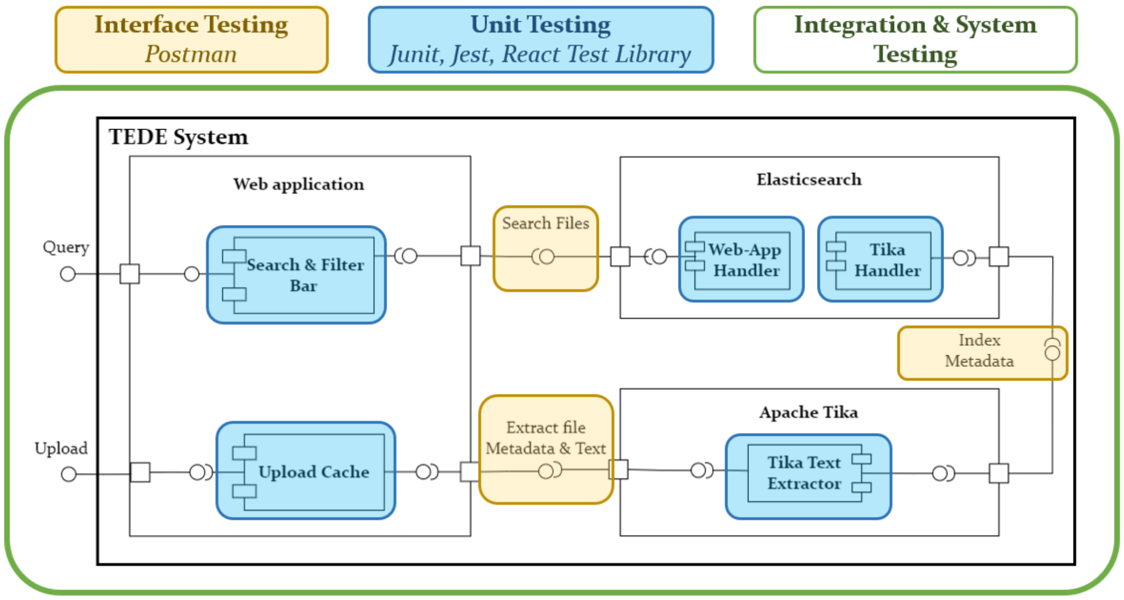
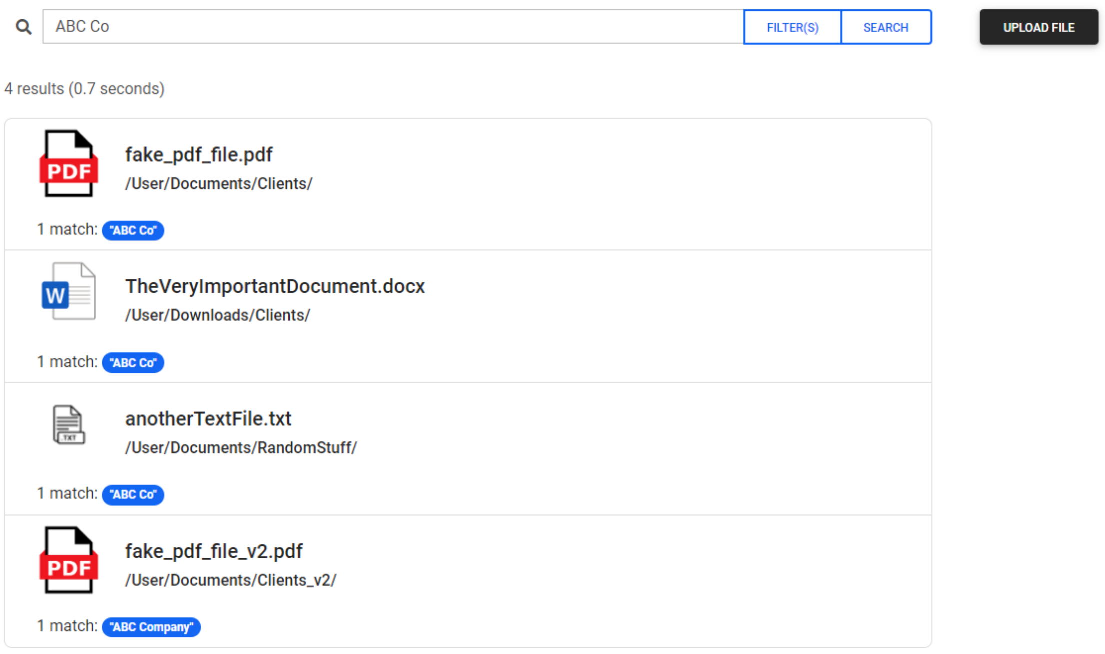


Figure 9: Annotated Component Diagram with the testing strategy

6 Implementation

As a part of the project planning phase, we implemented a simple prototype of the user interface. Screenshots of our client-approved mockup can be found in Figure 10.1, Figure 10.2, and Figure 10.3 For this project, implementation is inseparable from the design activities. See the design section for more details.



4 results (0.7 seconds)

fake_pdf_file.pdf
/User/Documents/Clients/
1 match: "ABC Co"

TheVeryImportantDocument.docx
/User/Downloads/Clients/
1 match: "ABC Co"

anotherTextFile.txt
/User/Documents/RandomStuff/
1 match: "ABC Co"

fake_pdf_file_v2.pdf
/User/Documents/Clients_v2/
1 match: "ABC Company"

Figure 10.1: Mock-Up of the Results View of the User Interface.

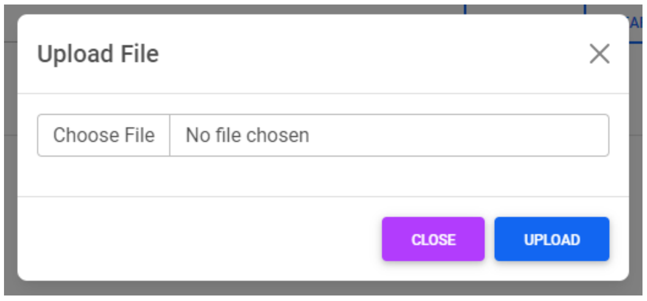
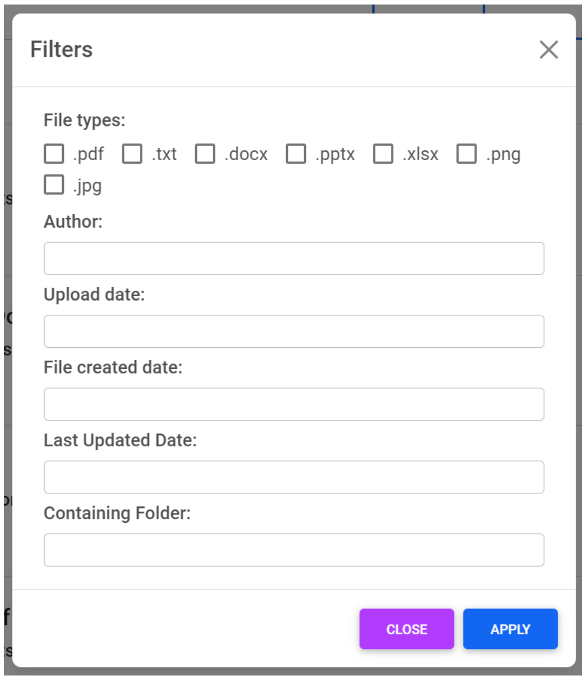


Figure 10.2: Mock-Up of the Upload File View of the User Interface.


Filters

File types:

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

Author:

Upload date:

File created date:

Last Updated Date:

Containing Folder:

Figure 10.3: Mock-Up of the Filters View of the User Interface.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

The table below, Table 6, addresses how the ACM Code of Ethics and Professional Conduct addresses each of the seven professional responsibilities from the paper.

Area of Responsibility	Definition	ACM Code of Ethics and Professional Conduct
Work of Competence	Deliver high quality, professional work on time	2.2 Perform high-quality work and take responsibility for your work 2.6 Perform work in your knowledge field
Financial Responsibility	Deliver valuable, reasonably cost products	3.3 Manage the team and resources
Communication Honesty	Communicate the true state of the work clearly	1.3 Be honest and trustworthy 2.5 Evaluate the application functionality and their impacts

Health, Safety, Well-Being	Establish a safe, healthy work environment	1.2 Avoid harm 2.9 Create secure but usable systems 3.7 Take care of systems that are widely used
Property Ownership	Respect all aspects of clients and others	1.6 Respect Privacy 1.7 Honor confidentiality 2.8 Access authorized resources
Sustainability	Minimize impact on the environment	1.2 Avoid actions which result in negative consequences (unjustified damage to property, animals, humans, and the environment)
Social Responsibility	Deliver beneficial products	1.1 Contribute to society; and all people are stakeholders 3.1 Ensure that the public good is the main concern

Table 6: ACM Areas of Responsibilities

7.2 Project Specific Professional Responsibility Areas

We now describe

Work of Competence – High

Work competence plays a key role in our project as we, as a team, agreed to deliver an optimal product to the client based on some agreed upon requirements and guidelines. We are required and committed to having the technical knowledge and skills to understand the project's requirements, create new ideas/designs, and implement such ideas based on the predetermined expectations set forth by both parties (team and client).

Financial Responsibility – Medium

This responsibility applies to our project because the project aims to produce a product that enables users to find their documents more quickly, and it does so in a very straightforward fashion since it only requires the implementation of a search engine. By allowing users to find their documents more quickly, our software will cut down on time they spend searching for documents and enable them to spend more time doing other tasks that will benefit the company.

Communication Honesty - High

Being transparent with our progress on the project is of high responsibility. Design decisions and functionality must be disclosed to show the reasoning behind them and their true state. The end product should function as we have claimed. We must fully utilize our resources, such as our Faculty Advisor and Buildertrend contact, for our project to succeed. To avoid failure, our team members will also need to communicate honestly with each other.

Health, Safety, Well-Being - Low

Health, safety, and well-being do not apply in a very traditional sense to our project. Since our project is entirely based in software and does not interact with any physical objects, there is not any physical harm that can be caused by our project. However, it is important for us to make sure that our system takes fundamental security precautions (e.g., SQL injections) into consideration to ensure that our system is secure.

Property Ownership - High

Since we are developing a project for a company working in the industry, it is important to understand the boundaries that will be established and to honor those conditions. Documents such as NDA's will be presented before us, and it will be important for everyone to understand the conditions of the contracts that we sign and the repercussions of breaking the contracts.

Sustainability - Low

Since our project is completely software and is not expected to be computationally intensive, the impact on the environment is low. The potential use of cloud platforms for storage and execution would increase the electricity requirements for the application. It is our responsibility to keep the amount of computation and storage required low for the application to function.

Social Responsibility - Medium

As professionals, we should always strive to make people (including users, customers, colleagues, and others affected directly or indirectly) and their experiences our primary concern. It is important to find a balance between technical complexity and user experience, such as in our project. We are to share information and knowledge acquired with clients and other stakeholders appropriately and maintain open communication channels to deliver the optimal solution. Our team must also consider social responsibilities when developing the product(user privacy, interactions, risks, and limitations).

7.3 Most Applicable Professional Responsibility Area

Communication honesty is especially important to the success of our project, and our team demonstrates a high level of proficiency in it. It will be crucial to be transparent with the project's progress, and as such, we shall be consistently communicating with our advisor and the client. Besides, we need to be honest about what is attainable and what is not. We have had regular meetings with our faculty advisor and kept him updated on our progress. Additionally, we have regularly communicated to and tried contacting our clients (it is just a little delayed from their end). Overall, being honest in our communications would lead to better relations and a more attainable successful project.

8 Closing Material

After describing our project in detail, we now highlight the main results of our project. We also reiterate the work done so far and our goals for the next phases of our project.

8.1 Discussion

We have designed our solution to meet our client's requirements. We have included support for the request file types for extracting text. As requested, we have also included a user interface that allows the user to query by keywords and filters.

8.2 Conclusion

We started by researching potential packages to help us extract text from files and develop a simple user interface. Then based on that, we decided on our technology stack. After this, we continued to learn more about how to use the technologies we used.

To reiterate, our goal is to design a prototype application that uploads the text content and metadata of supported file types into Elasticsearch and develop a simple user interface to query and filter uploaded content. Our application will serve as a proof of concept as part of Buildertrend “Global Search” initiative.

The best course to achieve these goals is to follow our plan developed this semester. We have researched the capabilities of our selected technologies to produce our application. We have created a prototype of the user interface portion of our application to verify Buildertrends' needs. Next semester, we will begin implementing our application with our understanding of the project gained this semester.

8.3 References

Sharma, Manish, and Rahul Patel. “A Survey on Information Retrieval Models, Techniques and Applications.” *IJETAE*, vol. 3, no. 11, Nov. 2013.

8.4 Appendices

This section includes the contract our team agreed to at the beginning of our project.

8.4.1 Team Contract

Team Members

1) Bruce Bitwayiki
2) Jared Hayashi
3) Rushal Sohal
4) Tiffany Mayberry

Team Procedures

1) Day, time, and location (face-to-face or virtual) for regular team meetings:
    a) Monday after 5 - virtual
2) Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
    a) Discord
3) Decision-making policy (e.g., consensus, majority vote):
    a) Open discussions and consensus regarding key decision
4) Procedures for record-keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
    a) Meeting notes will be stored in our shared Google drive
    b) Team members will alternate who takes meeting notes

Participation Expectations

1) Expected individual attendance, punctuality, and participation at all team meetings:
    a) Team members are expected to attend all meetings unless they give prior notice of absence
    b) Team members should not miss more than three meetings in a row.
2) Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
    a) Team members should complete work within the allotted time unless communicated prior if they can't
3) Expected level of communication with other team members:
    a) Team members should maintain high levels of communication
4) Expected level of commitment to team decisions and tasks:
    a) Every team member should be equally involved in the decision-making process

Leadership

1) Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
    a) Bruce Bitwayiki - Meeting notes
    b) Jared Hayashi - Client Interaction

    c) Rushal Sohal - Individual component design, testing, documentation and report submitting
    d) Tiffany Mayberry - Team organization
2) Strategies for supporting and guiding the work of all team members:
    a) Attend team meetings, participate(by asking/answering questions), update team regarding your progress
3) Strategies for recognizing the contributions of all team members:
    a) At the start of the meeting, briefly summarize what you worked on and how much progress was made
    b) Go over what you plan to make progress on by next meeting and how the team can support you

Collaboration and Inclusion

1) Describe the skills, expertise, and unique perspectives each team member brings to the team.
    a) Bruce Bitwayiki – IOS/Swift mobile development, Google Colab/Jupyter Notebook, AWS
    b) Jared Hayashi - SQL, Java, REST
    c) Rushal Sohal - C, Java, Jupyter Notebook, Firebase
    d) Tiffany Mayberry - Java, Python, Git, SQL
2) Strategies for encouraging and support contributions and ideas from all team members:
    a) Brainstorm ideas and possible solutions as a team
    b) All team members will have the opportunity to explain their ideas
3) Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
    a) Discord channel (of just team members) for effective communication
    b) Could report to the instructor

Goal-Setting, Planning, and Execution

1) Team goals for this semester:
    a) Have sufficient and detailed documentation for project plan from start to finish
2) Strategies for planning and assigning individual and teamwork:
    a) Discuss and assign responsibilities to every individual weekly
3) Strategies for keeping on task:
    a) Hold each other responsible for deadlines and distribute work evenly amongst team members

Consequences for Not Adhering to Team Contract

1) How will you handle infractions of any of the obligations of this team contract?

    a) Issue will be discussed in a team meeting and the member will be given a warning
2) What will your team do if the infractions continue?
    a) If the issue does not resolve, we will contact the class instructor or our faculty advisor to discuss next steps

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Bruce Bitwayiki____________________________________ DATE: 2/12/22

2) Rushal Sohal______________________________________ DATE: 2/12/22

3) Jared Hayashi_____________________________________ DATE: 2/12/22

4) Tiffany Mayberry__________________________________ DATE: 2/12/22