

## 1.1 PROBLEM STATEMENT

What problem is your project trying to solve? Use non-technical jargon as much as possible.

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 that they would've spent looking for the file on more productive tasks.

## 1.2 REQUIREMENTS & CONSTRAINTS

List all requirements for your project . This includes functional requirements (specification), resource requirements, qualitative aesthetics requirements, economic/market requirements, environmental requirements, UI requirements, and any others relevant to your project. When a requirement is also a quantitative constraint, either separate it into a list of constraints, or annotate at the end of requirement as "". Other requirements can be a single list or can be broken out into multiple lists based on the category.

### Requirements:

- Functional Requirements:
  - The application should be able to extract and index metadata from several different file types
  - The application should be able to support jpg, png, pptx, pdf, docx, xlsx, xls, and txt files
  - The application should match search keywords with text fragments from the files
  - The application should provide the ability for users to upload supported file types
- User Interface and Aesthetics:
  - The application should have a simple UI that allows users to search files that are indexed
  - The application should display the results of a search query (ordered by best fit)
  - The application should indicate if no search results were found
  - The application should display the filename and path of the files that match the search criteria
  - While the application is searching for results, the application shall indicate the results are loading
- Resource Requirements:
  - The application should be easily deployable to a server(s) (potentially provided through Iowa State in the initial stages of development)
- Environmental Requirements:
  - All users will have the same view and permissions
  - No authentication and authorization needed for this project

### Constraints:

- The query keyword should be up to 140 characters long
- The application should be a desktop or web application
- The application should be able to be readable in windows larger than 500 by 600 size
- The application should utilize Elasticsearch to store the metadata
- The application should return a result within 10 seconds
- If the application does not return a result within 10 seconds, the application should limit the amount of results returned or indicate an exception

### Stretch Requirements:

- Application should display preview of selected files
- Specific search filters such as file type, file name, author, date
- Display search statistics such as load time, number of keywords/indexes for specific files
- (Constraint?) Application should limit number of results displayed for performance
- The application should be able to run and interact with Google Cloud Platform such as Cloud Functions, Cloud Run, and Cloud Storage
- The application should automatically index uploaded files
- The application should include a feature to index a list of documents on demand
- When a file is deleted from the server, the application shall delete the metadata and index from elasticsearch

### 1.3 ENGINEERING STANDARDS

What Engineering standards are likely to apply to your project? Some standards might be built into your requirements (Use 802.11 ac wifi standard) and many others might fall out of design. For each standard listed, also provide a brief justification.

- IEEE-29119: We will write Unit Tests for our 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

### 1.4 INTENDED USERS AND USES

Who benefits from the results of your project? Who cares that it exists? How will they use it? Enumerating as many “use cases” as possible also helps you make sure that your requirements are complete (each use case may give rise to its own set of requirements).

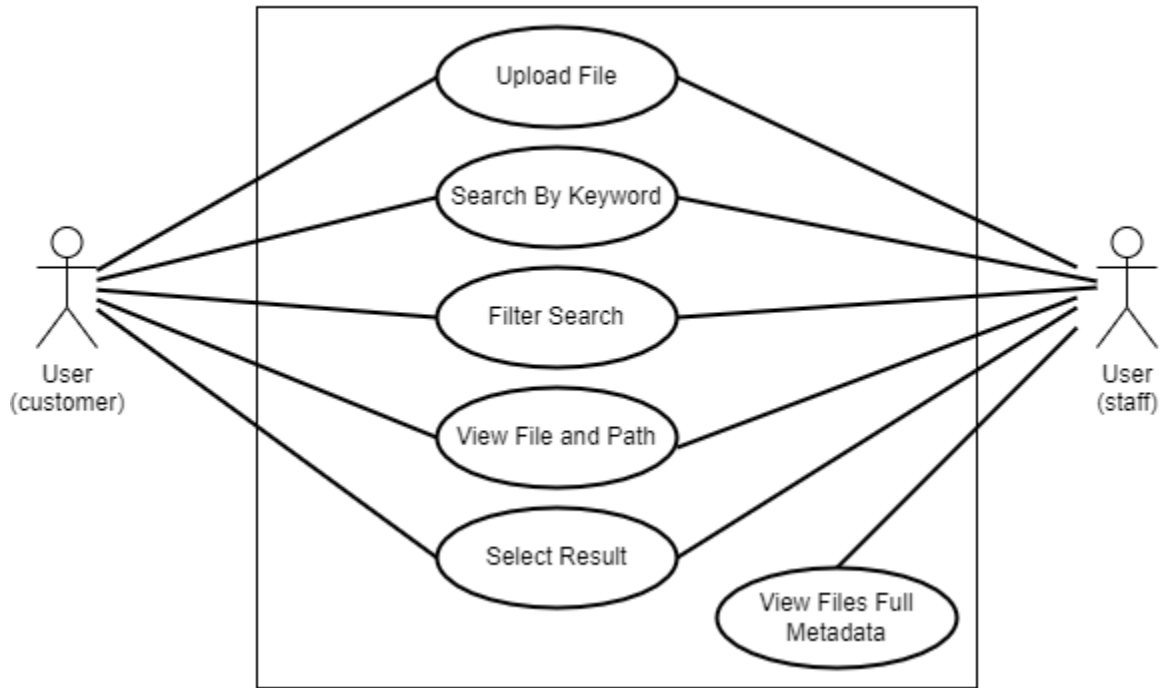
- BuilderTrend customers and support staff are the primary users

Below are some use cases.

- Use Case 1: 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).
- Use Case 2: Support staff shall have additional access to reports and queries made by regular users.

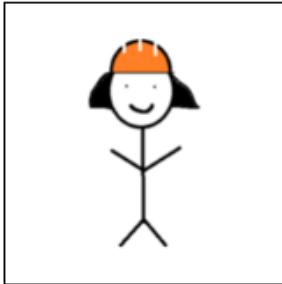
Use Case Diagram

**USE CASE DIAGRAM**



## User Persona 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