Group 2 Requirements Assignment

Preface: This document defines the requirements for the Website, GrabDocs. GrabDocs is a platform designed to simplify storage, retrieval, sharing of digital documents, and communication amongst team members.

Introduction:

GrabDocs serves as an AI powered document management and video meeting tool. It provides real-time AI meeting transcriptions, document management, form building, collaborative workspaces and more. GrabDocs can work standalone or be paired with other systems based on the service required. GrabDocs offers a variety of services as a platform. Businesses might seek GrabDocs' services to streamline document management, transcriptions and gain AI insights on meetings and files.

Glossary:

- Transcription: A written or printed representation of something
- OCR (Optical Character Recognition): The process of turning images into text.
- AI Chat: A feature that lets users ask questions about a document and get answers.
- Entity Recognition: A natural language processing technique for extracting key information, such as people, organizations, locations, dates, and other specific entities, from unstructured text
- Workspace: A shared folder or project where multiple people can collaborate.

User requirements definition:

1. Scan and upload documents with text automatically recognized.
2. Ask questions about a document and get AI-based answers/summaries.
3. Record a meeting and receive a transcript and notes afterward.
4. Share documents and set who can view/edit them.
5. Work with documents offline and sync them when the internet is back.

- **Presentation Layer**: Developed using Java frameworks and provides responsive UI for desktop and mobile.
- **Application Layer**: Handles authentication, document management, and search indexing.
- **Data Layer**: Database for user accounts, metadata, and access controls. Object storage for documents. Search indexing.

System Requirements Specification:

Functional Requirements:

1. Users shall be able to register, login, and logout
2. User shall be able upload file from local machine and cloud storage
3. User shall be able to create collaborative workspace
4. User shall create and select the form templates
5. User shall be able to view real-time AI transcriptions and insights.

Non-Functional Requirements:

1. Scalability: As user presence/file sizes increase, the system should be able to effectively handle and perform essential tasks
2. Performance: The system should consistently & efficiently perform to standard and provide users with ease of use
3. Security: The system should be secure against possible cyber-attacks and data leaks. Users should trust that their information will remain confidential
4. Availability: System should have at least 99.9% uptime.
5. Accessibility: Must be usable by people with disabilities and follow accessibility standards.
6. Storage/file size: The maximum file size must be specified.

System Evolution:

GrabDocs is expected to evolve as technology and user needs change. Future updates could include multi language support, more advanced AI features like risk detection in contracts, and integration with tools like google docs, google meet, slack, Microsoft teams, Microsoft word and more. The system may also need updates to keep up with new or existing security standards or privacy laws.

Appendices:

Appendix A: Use Cases

- UC1: Scan and upload a document.
- UC2: Ask AI questions about a contract.
- UC3: Record and transcribe a meeting.

Appendix B: User Stories

1. As a manager I want AI summaries of contracts, so I don't miss key points.
2. As a team member I want meeting transcripts, so I don't have to actively take notes.
3. As an admin I want to control permissions so only the authorized people see certain files.
4. As a user I would like offline access so I can work on the go.
5. As a worker, I want to scan

Index:

Non-functional Requirements Performance

<u>Ethical</u>: This platform should have all kinds of consumers in mind and be accessible for those with disabilities. Features should be put in place to ensure that all clients have the same experience when using the platform.

<u>Legislative</u>: This platform should rightly secure itself against cyberattacks, malware, and bad faith actors to ensure the safety of the client data. To protect the integrity of the platform, safeguards should be in place.

<u>Operational</u>: The platform should be scalable and able to handle a large volume of data and/or client accounts. The platform must be able to perform to standard while handling high-volume requests without shutting down or producing errors.

User Stories

1. As a user, I want to upload written documents for AI analysis so I can gain insights on what I wrote
2. As a user, I want to upload documents with different file types effectively, so I won't have to worry about document conversion
3. As a user, I want to access high-quality video conferencing so I can clearly meet with colleagues
4. As a user, I want real-time AI transcription on my video meetings so I can understand what is being said
5. As a user, I want to use a form builder so I can create forms through the platform without accessing an external service

Tasks

1. *<u>"As a user, I want to upload written documents…"</u>*

   Develop back-end upload functionality

   Outsource AI model for document analysis

2. *<u>"I want to upload documents with different file types…"</u>*

   Develop Java code to accept different file-types

   Develop back-end code to efficiently accept file uploads

3. *“I want to access high-quality video conferencing…”*

    Develop 4K/HD video conferencing integration into platform

    Find a server to handle video streaming

4. *“I want real-time AI transcription…”*

    Integrate AI model for transcription

    Develop back-end code to integrate model into video-conferencing

5. *“I want to use a form builder…”*

    Develop JavaScript presentation layer code

    Develop back-end code to store form responses

6. *“I want to reset my password easily so that I can easily regain access if I forget it.”*

    Develop password reset functionality with secure email verification.

    Implement password strength validation rules during reset.

7. *“I want to search across all my documents so that I can quickly find the one I need.”*

    Develop search indexing functionality for uploaded documents.

    Create a front-end search bar with filters (by date, file type, tags)

8. *“ I want to collaborate with others in real time so that we can work on the same document together.”*

    Implement document version control and change tracking.

Develop a real-time collaborative editing.

9. *<u>" I want to save my process automatically so I don't lose work if I close the app."</u>*

Implement auto-save functionality with periodic checkpoints.

Build local storage fallback in case the internet disconnects.

10. *<u>"I want to access the platform on mobile devices so I can work on the go."</u>*

Create responsive design layouts for small screens.

Conduct mobile usability testing on iOS and Android devices.

Manual Equivalent of Application

The manual equivalent of GrabDocs would be traditional handling and filing system for documents. Since GrabDocs is an online tool, it's manual counterpart would involve people handing some of the following tasks:

- Filing and Storing Documents
- Organizing Documents
- Manually searching
- Searching
- Information Extraction
- File Sharing

Architectural Diagram

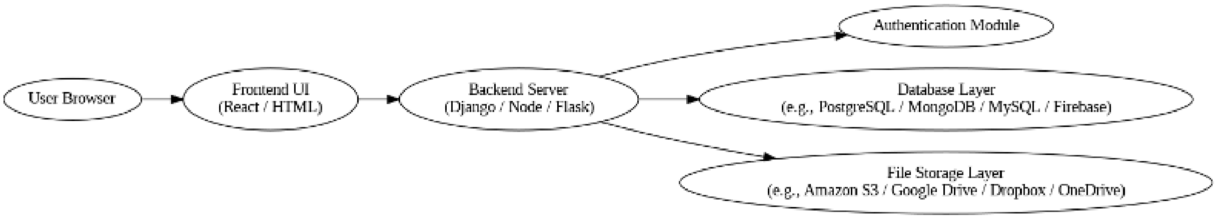

Figure 1: Architecture with optional databases and cloud storage services shown as examples.

Use Case Diagram

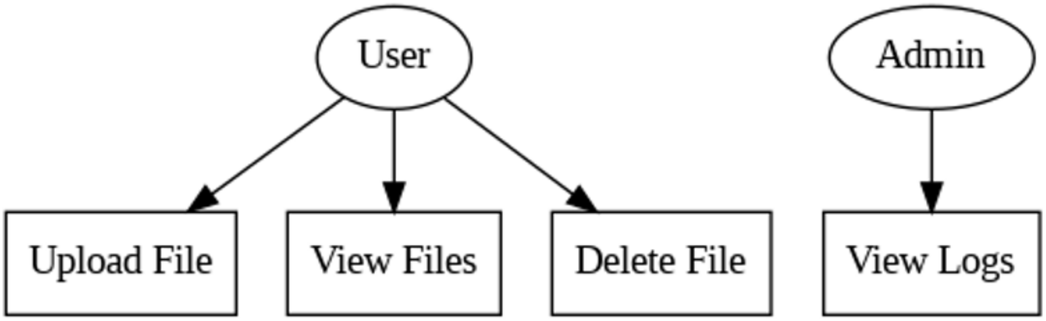

Figure 2: Use Case Diagram showing user interactions