Software Requirements Specification

for

Smartdocs

Mentor : Sandeep Kankal sir

Prepared by Anil Jangid (202201103142)

Members : Aryan Chaudhari (202201103135)

Tushar Harkal (202201103139)

Sahil Lakhase (202101103191)

Date : 29-07-2025

Table of Contents

1. Introduction

SmartDocs is a modern, AI-based document tool created to help users deal with long or complex documents more efficiently. Whether it's a student reading research papers or a business analyst reviewing reports, this system helps users quickly understand the content using summarization, Q&A, and visual tools.

1.1 Purpose

SmartDocs is an AI-powered web platform that helps users analyze complex documents (PDFs, PPTs, CSVs, Word files) by providing summaries, visualizations, quizzes, and AI-generated images.

1.2 Intended Audience

The primary intended audiences are:

- **Developers:** Implement AI integrations and file parsing.
- **Admins:** Manage users and documents.
- **End Users:** Students, professionals, educators.

1.3 Project Scope

- Generate short, readable summaries.
- Convert tables into clear charts.
- Generate images based on prompts.
- Allow question-answer interaction based on the document.
- Convert slides into quizzes or question banks.

1.4 Definitions and Acronyms

- **GPT:** Generative Pre-trained Transformer, a large language model used for understanding and generating text.
- **LangChain:** A framework used to process document chunks and retain context

for language model interactions.

- **CSV/DOCX/PDF/PPTX:** Common file types supported by SmartDocs.
- **Firebase:** A platform developed by Google for creating mobile and web applications, used for storing user data and documents.
- **QnA:** A question-answer feature based on document content.
- **DALL·E:** An AI system that generates images from text descriptions.
- **UI:** User Interface.

1.5 References

- OpenAI API Documentation
- Firebase Documentation
- LangChain Tutorials
- Chart.js Documentation
- pdfplumber Documentation
- python-docx Documentation
- pandas Documentation
- python-pptx Documentation

2. Overall Description

2.1 Background and Product Perspective

The product, SmartDocs – Intelligent Document Assistant, is an independent product and does not directly depend on any other major product or system for its core functionality, though it integrates with various third-party APIs. The product aims to automate various tasks associated with handling digital document details, better organizing the stored information, and achieving optimum performance in document comprehension. This helps users reduce the burden of reading through lengthy or technical documents by using AI to present key points, answer questions, and visualize data clearly.

2.2 Major Features and Functions

SmartDocs will provide the following primary functions:

- **Document Upload:** Allow users to upload documents in .pdf, .docx, .csv, and .pptx formats.
- **AI-Powered Summarization:** Generate concise summaries and extract key points from uploaded documents using OpenAI GPT.
- **Data Visualization:** Automatically detect and extract tabular data from documents and convert it into interactive charts using Chart.js.
- **Question Answering:** Enable users to ask natural language questions related to the content of the uploaded document and receive AI-generated responses via LangChain and GPT.
- **Quiz and Question Bank Generation:** Automatically create quizzes or custom question sets from PowerPoint slides using OpenAI GPT.
- **AI Image Creation:** Generate contextual illustrations or figures based on text prompts using DALL·E.
- **Secure Data Storage:** Save user data, document metadata, and generated content securely using Firebase.

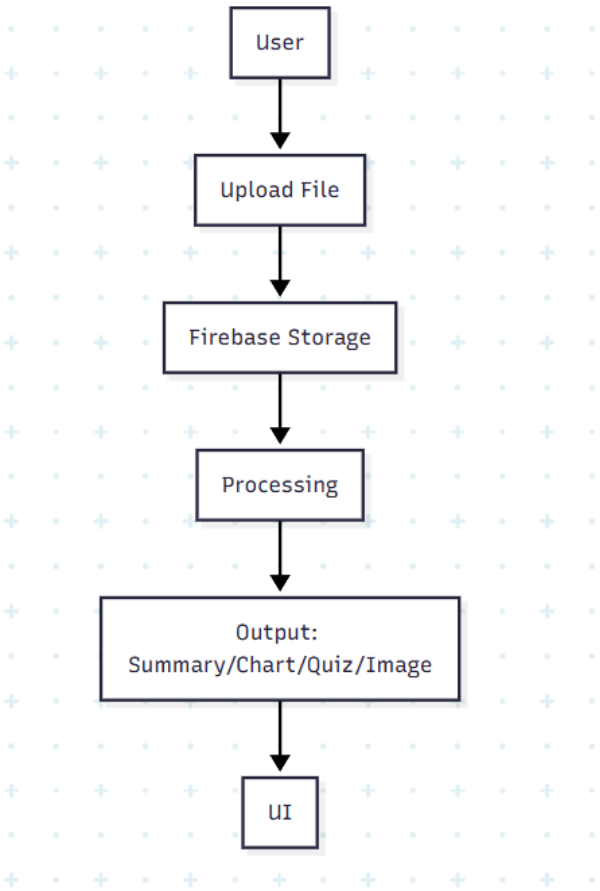
2.3 User Profile

- **Technical Proficiency:** Users are assumed to be able to use web browsers and perform basic file operations (e.g., uploading files). No coding or specific technical background is required to operate the application.
- **Target Audience:** Students, educators, researchers, and professionals who frequently interact with and need to extract insights from various document types.

2.4 Assumptions and Limitations

- **Internet Connectivity:** The system requires an active internet connection for full functionality, as it relies on cloud-based AI APIs and Firebase services.
- **API Access:** Continuous and stable API access to OpenAI (GPT and DALL·E) and Firebase is required for core features to function.

- **Document Format Quality:** The accuracy and effectiveness of summarization, Q&A, and data extraction are dependent on the clarity and machine-readability of the uploaded documents. Scanned image-based files are not fully supported for text extraction (OCR is a future scalability option).
- **PPT Content Structure:** Quiz generation relies on clear, structured, and sufficiently detailed text content within PowerPoint slides.
- **Data Entry (if applicable):** It is assumed that any manual data entry (e.g., user profile updates) by office personnel will be based on correct values.
- **Administrator Responsibility:** Users with administrator access (if implemented for system management) are assumed to exercise caution when modifying or deleting information to prevent database inconsistency.



3. External Interface Requirements

3.1 User Interfaces

The SmartDocs user interface will be a modern, responsive Graphical User Interface (GUI) developed with React.js and styled with Tailwind CSS. It will feature:

- **Intuitive Navigation:** GUI along with meaningful frames and buttons for easy navigation and interaction.
- **File Upload Area:** A clear and accessible interface for uploading various document types.
- **Content Display:** Dedicated panels for displaying summaries, extracted key points, interactive charts, and AI-generated images.
- **Interactive Input:** Text fields for Q&A, and controls for quiz generation parameters.

- **Feedback and Reports:** Reports (e.g., quiz results, document analysis overviews) will be generated as per user requirements.
- **Error Handling:** When invalid inputs are given or incomplete information is provided, appropriate error messages will be popped up to inform the user.

3.2 Hardware Interfaces

The SmartDocs system operates as a web application and primarily interfaces with standard client-side computing hardware.

- **Client System Configuration:**
  - **Processor:** Intel i5 11th generation or equivalent (minimum recommended for client-side browser performance).
  - **RAM:** 4 GB (minimum).
  - **HDD:** 80 GB (minimum for local storage/browser cache).
- **Operating System (Client):** Windows 11 or any modern operating system compatible with current web browsers.

3.3 Software Interfaces

SmartDocs interacts with various software components to deliver its functionality:

- **Frontend:** React.js
- **Backend:** Flask (Python) or Node.js (Express.js)
- **Database/Storage:** Firebase (Firestore for structured data, Storage for document files).
- **External APIs:**
  - OpenAI API (for GPT models and DALL·E).
- **Python Libraries (Backend):**
  - pdfplumber: For PDF content extraction.
  - python-docx: For DOCX content extraction.
  - pandas: For CSV processing and tabular data manipulation.
  - python-pptx: For PPTX content extraction.
  - LangChain: For orchestrating language model interactions.

- **JavaScript Libraries (Frontend):**
    - Chart.js: For rendering interactive data visualizations.

3.4 Communications Interfaces

The SmartDocs system leverages standard communication protocols for data exchange:

- **HTTP/HTTPS Protocols:** All data transfer between the client (frontend), server (backend), and external APIs will utilize HTTPS to ensure secure and encrypted communication.
- **RESTful API:** The backend will expose RESTful endpoints for the frontend to consume, and it will make RESTful calls to external AI APIs.
- **Local Area Network (LAN) / Internet:** The machine hosting the backend server and client devices will need to be part of a network (LAN or internet) to access the central database (Firebase) and external APIs.

4. Functional Requirements

This section details the specific functional requirements that SmartDocs must fulfill.

- **Accept File Uploads:** The system shall accept user uploads of documents in PDF, DOCX, CSV, and PPTX file formats.
- **Generate Document Summaries:** The system shall utilize the OpenAI GPT API to generate concise and readable summaries from the text content of uploaded documents.
- **Visualize Tabular Data:** The system shall automatically detect and extract tabular data from uploaded documents (PDF, DOCX, CSV, PPTX) and convert it into interactive charts using Chart.js on the frontend.
- **Generate Custom Images:** The system shall generate contextually relevant images based on user-provided text prompts or automatically extracted topics from documents, using the DALL·E API.
- **Handle Q&A Interactions:** The system shall allow users to ask natural language questions about the content of an uploaded document and provide accurate

answers by leveraging LangChain for context management and OpenAI GPT for response generation.

- **Build Quizzes/Question Banks from Slides:** The system shall automatically extract content from PowerPoint slides and generate quizzes (e.g., MCQs, short answers) or custom question banks using the OpenAI GPT API.
- **Secure Data Storage:** The system shall securely save uploaded files in Firebase Storage and associated metadata, summaries, chart data, quiz data, and image links in Firebase Firestore.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Processing Speed:** The system shall process and generate a summary for a typical 10-page document (approx. 5000 words) in under 5 seconds.
- **Query Response Time:** The system shall respond to user Q&A queries within 3 seconds.
- **Chart Generation Time:** The system shall generate and display charts from detected tabular data within 2 seconds.
- **Image Generation Time:** The system shall generate an AI image based on a prompt and display it within 10 seconds.
- **UI Responsiveness:** The user interface shall load and respond to user interactions (e.g., button clicks, form submissions) within 1 second.

5.2 Safety Requirements

- **Data Integrity:** The system shall ensure the integrity of uploaded documents and generated data, preventing corruption or unintended modification.
- **Error Handling:** The system shall gracefully handle errors during file processing or API interactions, providing informative messages to the user without crashing.
- **Resource Management:** The system shall implement mechanisms to prevent resource exhaustion (e.g., memory leaks, excessive API calls) that could lead to system instability.

5.3 Security Requirements

- **Authentication:** The system shall implement robust user authentication using Firebase Authentication to ensure only authorized users can access the application.
- **Authorization:** The system shall ensure that users can only access and manage documents they have uploaded or have been explicitly granted access to.
- **Data Confidentiality:** All sensitive user data and document content shall be encrypted during transit (HTTPS) and at rest (Firebase's default encryption).
- **API Key Management:** API keys for external services (OpenAI) shall be securely managed on the backend and never exposed to the client-side.
- **Input Validation:** The system shall perform rigorous input validation to prevent common web vulnerabilities such as injection attacks.

5.4 Software Quality Attributes

- **Usability:** The system shall be highly intuitive and user-friendly, requiring minimal learning curve for users familiar with web applications. The interface should be clean, consistent, and provide clear status updates.
- **Reliability:** The system shall operate consistently and correctly, minimizing failures. It should include retry mechanisms for external API calls and robust error logging. The system should aim for 99.9% uptime.
- **Maintainability:** The codebase shall be modular, well-commented, and adhere to established coding standards to facilitate easy understanding, modification, and debugging by developers.
- **Scalability:** The architecture shall be designed to accommodate an increasing number of users and documents without significant performance degradation, leveraging cloud services (Firebase) and scalable backend frameworks (Flask/Node.js).
- **Accuracy:** Summaries, Q&A responses, and data visualizations shall accurately reflect the content and intent of the original documents.

- **Extensibility:** The system should be designed to allow for the easy addition of future features, such as support for new file formats (e.g., markdown) or integration with other AI models (e.g., OCR for scanned documents).

6. Other Requirements

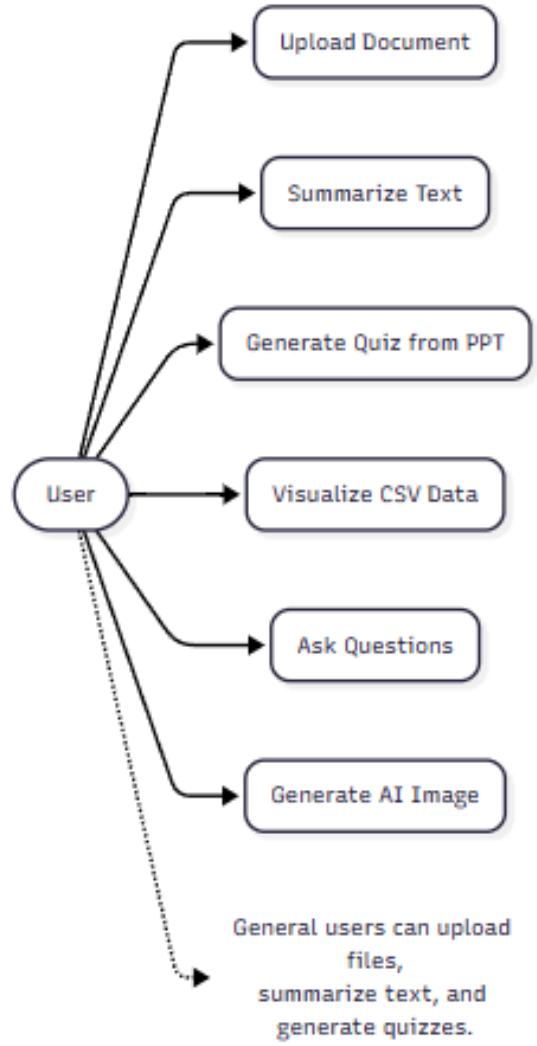
- **User Account Management:** The system shall allow users to register, log in, and manage their basic profile information.
- **Document History:** The system shall provide a history of uploaded documents and the results of processing (summaries, charts, quizzes) for easy retrieval.
- **Feedback Mechanism:** The system should include a way for users to provide feedback or report issues.

Appendix A: Glossary

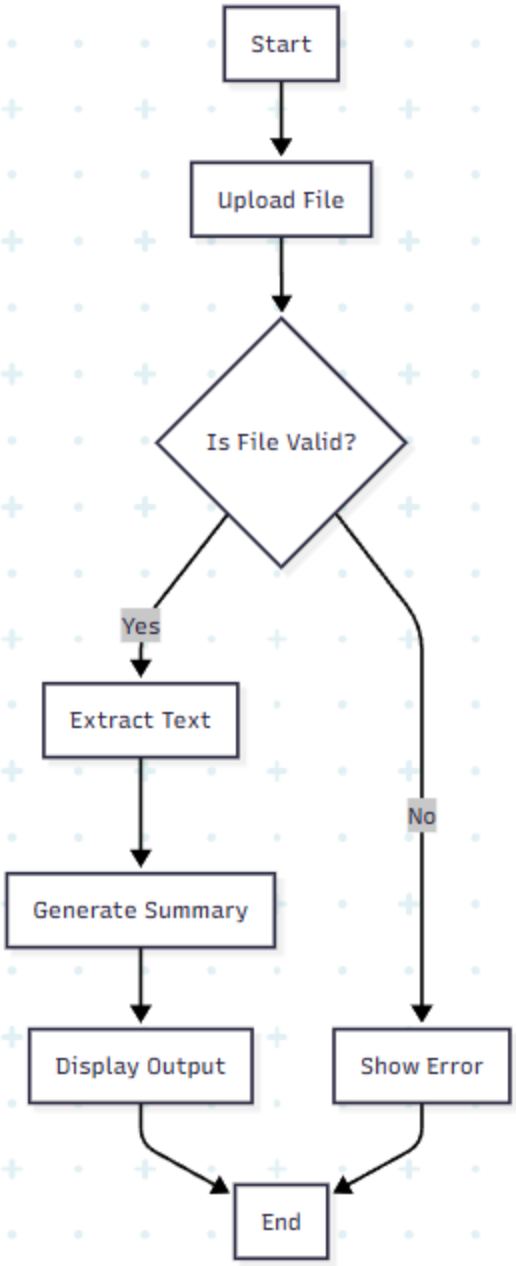
- **Summarization:** The process of reducing a text to its key points or main ideas.
- **DALL·E:** An artificial intelligence system capable of creating realistic images and art from a description in natural language.
- **LangChain:** A development framework designed to simplify the creation of applications using large language models.
- **Firebase:** A comprehensive mobile and web application development platform by Google, providing services like authentication, database, and storage.
- **OCR:** Optical Character Recognition, a technology that converts different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera, into editable and searchable data.

Appendix B: Analysis Models

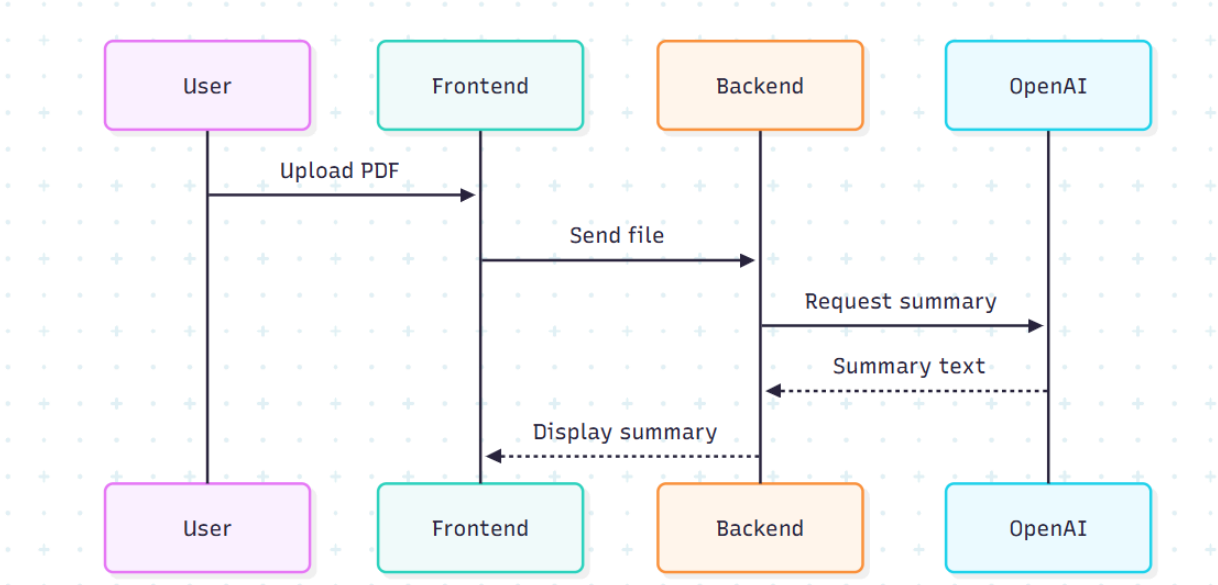
- **Use Case diagram:**


Upload Document
Summarize Text
Generate Quiz from PPT
User
Visualize CSV Data
Ask Questions
Generate AI Image
General users can upload files,
summarize text, and
generate quizzes.
Manage Users
Admin
Moderate Content
Admins manage user accounts
and moderate content.

- **Activity Diagram:**



- **Sequence Diagram:**



Appendix C: Issues List

- Specific limits for file size and daily processing quotas for users to manage API costs.
- Detailed error message specifications for each anticipated error condition.
- Strategy for handling very large documents that exceed typical token limits even with chunking (e.g., specialized summarization models or different processing approaches).
- Detailed design for user account management and profile updates.

5. Summary

SmartDocs is an AI-based tool created to make document interaction easier and faster. With support for smart features like summarization, charting, Q&A, and quiz creation, it's designed to serve the real-world needs of students, educators, and working professionals. This document defines the expected features and helps guide development and testing.