**Software Requirements**

**Specification**

**for**

**Research Paper Summarizations using RAG Model**

**Version 2.0 approved**

**Prepared by**

**S. Satwika (23BD1A665R)**

**N. Gayatri (23BD1A665B)**

**B. Vaishnavi (23BD1A664A)**

**Keshav Memorial Institute of Technology**

**08-04-2025**

**Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.**

**Software Requirements Specification for Research paper summarization Page ii**

# Table of Contents

**Table of Contents**........................................................................................................................... **ii**

**Revision History** ............................................................................................................................ **ii**

1. **Introduction** ............................................................................................................................. **3**
   1. Purpose ............................................................................................................................................. 3
   2. Document Conventions .................................................................................................................... 3
   3. Intended Audience and Reading Suggestions .................................................................................. 3
   4. Product Scope.................................................................................................................................... 3
   5. References ........................................................................................................................................ 3
2. **Overall Description** ................................................................................................................. **4**

2.1 Product Perspective ............................................................................................................................. 4

2.2 Product Functions ............................................................................................................................... 4

2.3 User Classes and Characteristics ........................................................................................................ 4

2.4 Operating Environment ...................................................................................................................... 4

2.5 Design and Implementation Constraints ............................................................................................ 4

2.6 User Documentation ........................................................................................................................... 4

2.7 Assumptions and Dependencies ..........................................................................................................4

1. **External Interface Requirements** ...........................................................................................**5** 
   1. User Interfaces....................................................................................................................................5
   2. Hardware Interfaces ...........................................................................................................................5
   3. Software Interfaces ............................................................................................................................5
   4. Communications Interfaces ...............................................................................................................5
2. **System Features** ........................................................................................................................**5**

4.1 System Feature 1 ..................................................................................................................................5

4.2 System Feature 2 ..................................................................................................................................6

* 1. System Feature 3 ..................................................................................................................................6

4.4 System Feature 4 ..................................................................................................................................6

1. **Other Non-functional Requirements** ..............................................................................................**7**
   1. Performance Requirements ................................................................................................................7
   2. Safety Requirements ..........................................................................................................................7
   3. Security Requirements .......................................................................................................................7
   4. Software Quality Attributes ...............................................................................................................7
   5. Business Rules ...................................................................................................................................7
2. **Other Requirements** .......................................................................................................................**7**

**Appendix A: Glossary** ....................................................................................................................**8**

**Appendix B: Analysis Models**.........................................................................................................**8**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for

“Research paper summarization using RAG model” a web-based application designed to assist users in summarizing and querying research papers in PDF and DOCX formats using a Retrieval-Augmented Generation (RAG) model powered by Large Language Models (LLMs). This SRS pertains to Version 2.0 of the application and covers the full system, including frontend, backend, and database components.

## 1.2 Document Conventions

* **Font Styles:** Bold text denotes section headings; italic text highlights key terms or emphasis.
* **Priority:** Unless explicitly stated, all requirements inherit the priority of their parent section (High by default).
* **TBD:** Indicates information to be determined later.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for :

**Developers**: To understand the system's functional and non-functional requirements.

**Project Managers:** To oversee the project's scope and deliverables.

**Testers:** To create test cases based on the requirements.

**End Users:** To understand the system's capabilities and limitations.

## 1.4 Product Scope

Our project enables users to upload research papers, receive AI-generated responses to queries (e.g., summaries), and manage chat sessions. It aims to streamline research analysis by leveraging RAG and LLMs, offering benefits like time savings and enhanced understanding. The software aligns with goals of providing accessible AI-powered research tools. No separate vision and scope document exists.

## 1.5 References

**MongoDB Atlas Documentation:** [**https://www.mongodb.com/docs/atlas/**](https://www.mongodb.com/docs/atlas/)

**Flask Documentation:** [**https://flask.palletsprojects.com/**](https://flask.palletsprojects.com/)

**React Documentation:** [**https://reactjs.org/docs/**](https://reactjs.org/docs/)

**LangChain Documentation:** [**https://python.langchain.com/docs/**](https://python.langchain.com/docs/)

# 2. Overall Description

## 2.1 Product Perspective

The product is a new, self-contained web application integrating a React frontend, Flask backend, and MongoDB database. It interfaces with external LLM services (e.g., Together API) for text generation and uses FAISS for vector storage in the RAG model. It replaces manual research paper analysis with an automated, AI-driven solution.

## 2.2 Product Functions

* User registration and login with JWT authentication.
* Upload and processing of PDF/DOCX research papers.
* Creation, management, and deletion of chat sessions linked to documents.
* Query processing with LLM-generated responses, including document metadata queries.
* Chat history persistence and pinning functionality.
* Document preview for uploaded files.

## 2.3 User Classes and Characteristics

* Researchers: Frequent users who need quick summaries of research papers.
* Students: Users who require summaries for academic purposes.
* Professionals: Users who need to quickly understand technical documents.

## 2.4 Operating Environment

* **Front End**: ReactJS with Bootstrap.
* **Back End**: Python with Flask.
* **Database**: MongoDB.
* **Model**: RAG, Llama 3.3 70B Instruct Turbo Free.
* **Vector Storage**: FAISS.
* **Framework**: Langchain.

## 2.5 Design and Implementation Constraint

* File uploads limited to 10MB and PDF/DOCX formats.
* JWT token expiration set to 1 hour.

## 2.6 User Documentation

* Instructions for uploading documents, generating summaries, and using the Q&A feature.
* Inline tooltips and welcome messages in the UI.
* FAQs within the application.

## 2.7 Assumptions and Dependencies

* **Assumption:** The system assumes that users will upload valid research papers in PDF or Docx format; Users have stable internet access; documents contain extractable text.
* **Dependencies:** Together API uptime, MongoDB Atlas connectivity, third-party libraries (e.g., PyPDF2, pdfplumber).

# 3. External Interface Requirements

## 3.1 User Interfaces

* **Landing Page:** A homepage with an overview of the application, including a brief description, key features (summarization and Q&A), and a call-to-action to log in or sign up.
* **Login Page:** A secure form for users to enter credentials (username/email and password) to access the application, with options for registration and password recovery.
* **Chat Interface:** A section for uploading research papers in PDF/Docx format, to display the generated summary. And a text box for users to input questions and a section to display answers.

**3.2 Hardware Interfaces**

* N/A: The system is web-based and does not require specific hardware interfaces.

## Software Interfaces

## Front End: ReactJS with Bootstrap.

## Back End: Python with Flask.

## Database: MongoDB.

## Model: RAG, Llama 3.3 70B Instruct Turbo Free.

## Vector Storage: FAISS.

## 3.4 Communications Interfaces

* **HTTP/HTTPS**: For communication between the front end and back end.
* **API Calls**: For interactions between the back end and the RAG model.

# 4. System Features

## 4.1 User Authentication

### 4.1.1 Description and Priority

* Enables user registration and login with JWT-based authentication. High priority for securing user data and chat history.
* Priority: High

### Stimulus/Response Sequences

* **Signup:** User submits username, email, password → System hashes password, stores user, returns success.
* **Login:** User submits credentials → System verifies, issues JWT, updates last login.

### 4.1.3 Functional Requirements

* **REQ-1**: System must hash passwords using bcrypt before storage.
* **REQ-2**: System must issue a JWT token valid for 1 hour upon successful login.
* **REQ-3**: System must reject duplicate email registrations with a 400 error.

## 4.2 Document Upload and Processing

### 4.2.1 Description and Priority

* **Description:** Allows users to upload PDF/DOCX files for processing and querying. High priority for core functionality.
* **Priority**: High

### 4.2.2 Stimulus/Response Sequences

* **Upload:** User selects file → System validates format/size, extracts text/metadata, stores file.
* **Preview:** User requests preview → System displays PDF embed or DOCX text excerpt.

### Functional Requirements

### REQ-4: System must accept only PDF and DOCX files up to 10MB.

### REQ-5: System must extract text and metadata (title, author, sections) using PyPDF2/pdfplumber or python-docx.

* **REQ-6:** System must store uploaded files in the "uploads" folder with unique filenames.

## Chat Management

### Description and Priority

* Manages chat sessions linked to documents, including creation, deletion, renaming, and pinning. Medium priority for user experience.
* **Priority**: High

### Stimulus/Response Sequences

* **New Chat:** User clicks "New Chat" → System creates a chat session.
* **Delete Chat:** User selects delete → System removes chat and associated queries.
* **Pin Chat:** User toggles pin → System updates chat status.

### Functional Requirements

* **REQ-7:** System must persist chat sessions in MongoDB for registered users.
* **REQ-8:** System must allow renaming chats with non-empty names.
* **REQ-9:** System must support pinning chats for easy access.

## 4.4 Query Processing with LLM

### Description and Priority

* Processes user queries using RAG and LLM, providing responses based on uploaded documents. High priority for core AI functionality.
* Priority: High
  + 1. **Stimulus/Response Sequences**
* **Query Submission:** User enters query → System processes document, calls LLM, displays response.

### Functional Requirements

### REQ-10: System must split documents into 1500-character chunks with 200-character overlap.

### REQ-11: System must use FAISS and HuggingFace embeddings for document retrieval.

### REQ-12: System must integrate with Together API for LLM responses, limiting context to 8000 characters.

# 5. Other Non-functional Requirements

**5.1 Performance Requirements**

* System must process a 10MB document and return a response within 30 seconds under normal load.

**5.2 Safety Requirements**

• **N/A**: The system does not involve physical safety concerns.

## 5.3 Security Requirements

• User passwords must be hashed with bcrypt.

• JWT tokens must be validated for all protected routes.

• Uploaded files must be stored securely with unique names to prevent overwriting.

## 5.4 Software Quality Attributes

* **Usability**: The system must be easy to use, with clear instructions for uploading documents and asking questions.
* **Reliability**: The system must generate accurate summaries and answers.
* **Maintainability**: The code must be well-documented for future updates.

## 5.5 Business Rules

* Only registered users can persist chat history.
* Guest users’ chats are ephemeral.

**9 Other Requirements**

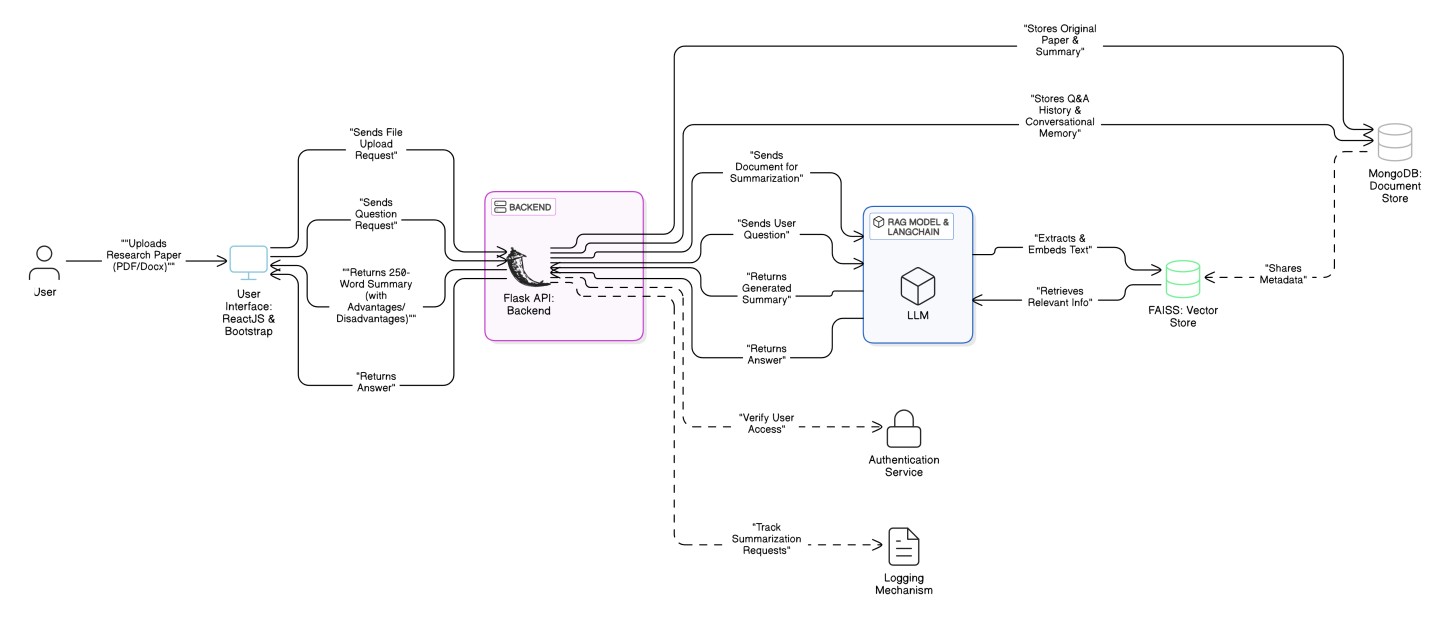
•  **N/A:** No additional requirements at this time.

# Appendix A: Glossary

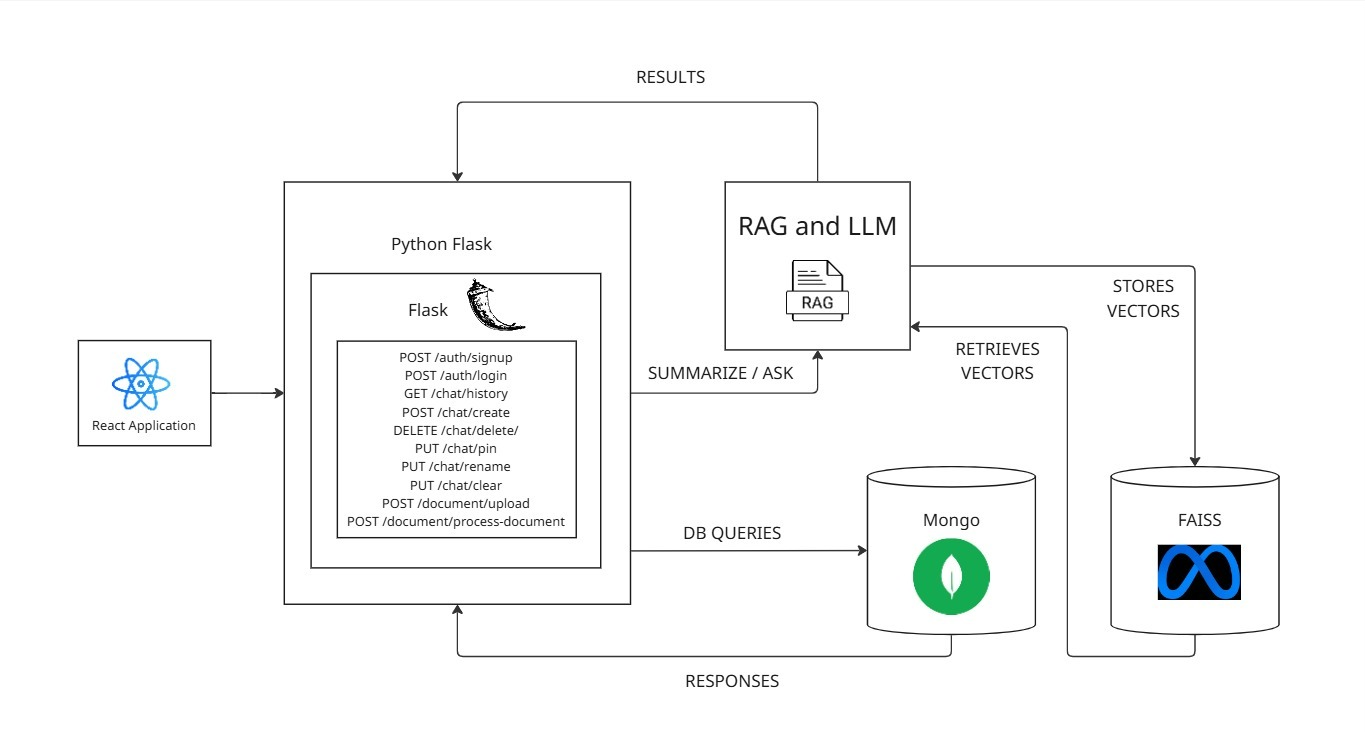
* **FAISS**: A vector database used for storing and retrieving document embedding
* **JWT:** JSON Web Token, used for authentication.
* **Langchain**: A framework for building applications with language models.
* **Llama 3.3 70B Instruct Turbo** **Free**: A transformer-based model used for text generation.
* **RAG Model**: Retrieval-Augmented Generation model used for summarization and Q&A.

# Appendix B: Analysis Models

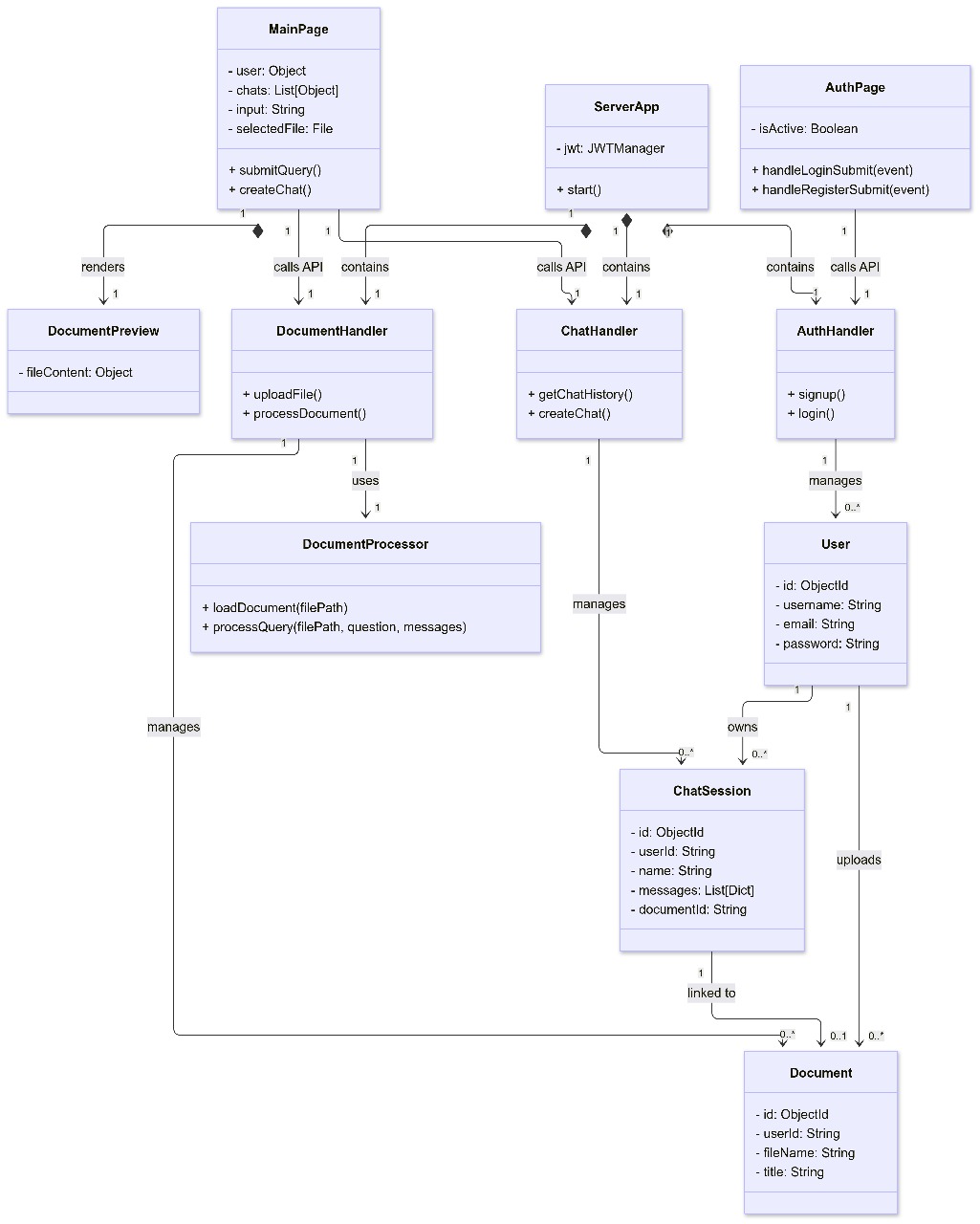
WORK FLOW DIAGRAM:



ARCHITECTURE DIAGRAM:



CLASS DIAGRAM:



DATABASE DESIGN:

