**Research Paper Summarization using RAG Model**

|  |  |  |
| --- | --- | --- |
| N. Shruti | – | 23BD1A059B |
| K. Niharika | – | 23BD1A662V |
| B. Karunya | – | 23BD1A05A6 |
| B. Vamshi Deepak | – | 23BD1A05A7 |
| S.V.S Ram Maanas | – | 23BD1A05BJ |
|  |  |  |
|  |  |  |

**Abstract:**

**Problem Statement: Research Paper Summarization using RAG Model**

Research papers contain vast amounts of data, making it difficult for users to extract key insights quickly. The Research Paper Summarization System is designed to help users generate concise summaries while retaining the original document’s context.

This system will allow users to upload research papers and receive automated summaries that highlight key points, advantages, and disadvantages. Additionally, the system will enable interactive Q&A, where users can ask queries related to specific topics within the paper.

The goal is to reduce the time and effort required for analyzing research papers while ensuring that critical information is preserved. The system will provide an efficient, user-friendly platform for students, researchers, and professionals to quickly grasp complex research topics.

**Software Requirements Specification for Research Paper Summarization using RAG Model**

**Prepared by:**

1. N Shruti – 23BD1A059B

2. B.Vamshi Deepak– 23BD1A05A7

3. B.Karunya – 23BD1A05A6

4. S.V.S Ram Maanas – 23BD1A05BJ

5. K.Niharika– 23BD1A0662V

**Keshav Memorial Institute of Technology**

27-02-2025

### Table of Contents

[1.Introduction 5](#_Toc21806)

[1.1 purpose 5](#_Toc21807)

[1.2 Document Convention 5](#_Toc21808)

[1.3 Intended Audience and Reading Suggestions 5](#_Toc21809)

[1.4 Product Scope 5](#_Toc21810)

[1.5 References 5](#_Toc21811)

[2.Overall Description 6](#_Toc21812)

[2.1 Product Perspective 6](#_Toc21813)

[2.2 Product Functions 6](#_Toc21814)

[2.3 Operating Environment 7](#_Toc21815)

[2.4 User Characteristics 7](#_Toc21816)

[2.5 Design and Implementation Constraints 8](#_Toc21817)

[2.6 User Documentation 9](#_Toc21818)

[2.7 Assumptions and Dependencies 9](#_Toc21819)

[3. External Interface Requirements 10](#_Toc21820)

[3.1 User Interfaces 10](#_Toc21821)

[3.2 Hardware Interfaces 10](#_Toc21822)

[3.3 Software Interfaces 10](#_Toc21823)

[3.4 Communication Interfaces 10](#_Toc21824)

[4. System Features 10](#_Toc21825)

[4.1 Research Paper Uploading 10](#_Toc21826)

[4.2 Summarization using RAG Model 11](#_Toc21827)

[4.3 Chatbot for Q&A Interaction 11](#_Toc21828)

[4.4 Download Options 12](#_Toc21829)

[4.5 User Account & History 12](#_Toc21829)

[5. Nonfunctional Requirements 13](#_Toc21831)

[5.1 Performance Requirements 13](#_Toc21832)

[5.2 Safety Requirements 13](#_Toc21833)

[5.3 Security Requirements 13](#_Toc21834)

[5.4 Software Quality Attributes 13](#_Toc21835)

[5.5 Business Rules 13](#_Toc21836)

[6.Other Requirements 14](#_Toc21837)

[6.1 User Registration and Authentication: 14](#_Toc21838)

[6.2 Search and Filter Functionality: 14](#_Toc21839)

[6.3 Research Ppaer Summarization Portal: 14](#_Toc21840)

[6.4 AI Chatbot for Research Assiatance: 14](#_Toc21841)

[6.5 Reviews and Feedback System: 14](#_Toc21842)

[6.6 Document Processing and Cleanup: 14](#_Toc21843)

[6.7 Mobile Accessibilityt: 14](#_Toc21844)

[6.8 Notifications and Alerts: 15](#_Toc21845)

[6.9 Integration with External Libraries: 15](#_Toc21847)

[Appendix A: Glossary 15](#_Toc21848)

[Appendix B System Diagrams: 16](#_Toc21849)

[Appendix C: To Be Determined List 19](#_Toc21850)

### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reasons for Changes** | **Version** |
| Week-1 | 28-02-2025 | SRS Template Creation  (Introduction) | na |
| Week-2 | 28-02-2025 | SRS Documentation – Use Case and Class Diagram | na |

## 1.Introduction

# 1.1 purpose

# The purpose of this document is to define the requirements for the Research Paper Summarization System using the RAG Model. This system aims to simplify the process of understanding complex research papers by generating concise summaries and providing an interactive chatbot for users to ask queries related to the document. The traditional approach to research paper reading is time-consuming and requires extensive effort to extract relevant insights. By integrating Retrieval-Augmented Generation (RAG) and Conversational AI, the system will automate summarization while retaining contextual relevance. This will benefit researchers, students, and professionals who need quick and accurate insights from academic papers.

# 1.2 Document Convention This document follows a structured format for readability and consistency. The primary text is written in Times New Roman, size 12, while headings are in bold, size 14, and subheadings are in bold, size 12. Technical terms are explained in a glossary at the end of this document. Key system components such as API calls, database structures, and model functionalities are described in detail, ensuring clarity for developers and stakeholders. Figures and tables are included where necessary to improve understanding. All references to external documents, research papers, and AI frameworks follow IEEE citation standards for proper attribution.

# 1.3 Intended Audience and Reading Suggestions

This document is intended for multiple stakeholders, including **researchers, developers, educators, and project managers**. Researchers and students can use this system to generate summarized versions of research papers and ask questions to extract relevant information. Developers working on AI-based systems can refer to this document to understand the technical implementation and system architecture. Educators can use it to create educational material, while project managers can assess the feasibility and requirements of the system. The document is structured logically, beginning with an introduction, followed by system requirements, design considerations, and implementation details.

# 1.4 Product Scope

# The Research Paper Summarization System provides an AI-powered text summarization and Q&A assistant that processes academic research papers efficiently. Users can upload PDF/DOCX files, and the system will extract the most relevant information using RAG (Retrieval-Augmented Generation) Model. The system will generate structured summaries with key takeaways, advantages, and disadvantages. Additionally, a chatbot feature will allow users to ask queries regarding the research content and receive relevant answers. The goal is to reduce the effort required for understanding research papers, making academic research more accessible and efficient.

# 1.5 References

This document references multiple academic and technical sources, including **IEEE research papers on NLP and AI**, official **LangChain and ChromaDB documentation**, and existing literature on **Retrieval-Augmented Generation (RAG)**. The references also include relevant API documentation, industry-standard best practices for AI model deployment, and guidelines on ethical AI usage. Further citations provide insights into previous research on text summarization and knowledge retrieval to demonstrate how this system improves upon existing methods.

## 2.Overall Description

# 2.1 Product Perspective

# This system is designed to address the inefficiencies of traditional research paper reading. Academic papers are often lengthy and filled with complex terminology, making it difficult for users to extract useful information quickly. Current summarization tools lack contextual understanding, leading to generic summaries that fail to retain key details. This system overcomes these challenges by using RAG Model, ChromaDB for vector retrieval, and an AI chatbot for interactive Q&A. The system provides an engaging, intelligent, and context-aware way of summarizing and querying research papers, ensuring users get meaningful and precise information efficiently.

# 2.2 Product Functions

# The system offers three primary functions: document uploading, summarization, and chatbot-based Q&A. Users can upload research papers in PDF or DOCX format, which are then processed by text extraction algorithms to retrieve relevant content. The summarization module, powered by the RAG Model, generates structured summaries focusing on essential information, including advantages and limitations. The Q&A chatbot allows users to ask specific questions, retrieving precise, context-aware answers from the document’s indexed data. The system ensures high efficiency in providing meaningful insights while minimizing the time spent reading full research papers.

The Research Paper Summarization System provides three **core functionalities** that improve the way users interact with academic papers:

1. **Document Uploading**
   * Users can upload **PDF/DOCX** files containing research papers.
   * The system extracts **text content** while maintaining document structure.
   * It processes images, figures, and tables separately (if needed).
2. **Summarization Using RAG Model**
   * The uploaded document is divided into **logical text chunks**.
   * The **Retrieval-Augmented Generation (RAG) Model** retrieves **relevant sections** and generates a **structured summary**.
   * The summary contains key findings, methodologies, **advantages, and limitations** of the research.
   * Summarization results are presented in a **readable format** with bullet points for clarity.
3. **Chatbot-Based Q&A Interaction**
   * Users can ask questions about specific **concepts, sections, or data** in the research paper.
   * The **LLM-powered chatbot** fetches responses based on the indexed text.
   * The chatbot **remembers previous questions** for context-aware responses.
   * Users can request additional explanations, definitions, or **comparisons with other research papers**.

# 2.3 Operating Environment

# The system will operate as a web-based platform accessible through modern browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It will feature a ReactJS-based frontend, a Flask-powered backend, and will be hosted on cloud infrastructure to ensure scalability. The system’s AI models will be deployed using GPU-enabled servers to speed up inference times for both summarization and chatbot responses. Data storage will be managed using MongoDB for structured data and ChromaDB for vector-based document retrieval. The platform will be optimized for both desktop and mobile devices, ensuring accessibility across different user groups.

The system is designed as a **cloud-based web application**, accessible via:

* **Desktop browsers (Chrome, Firefox, Edge)** and **mobile devices** (Android, iOS).
* **Frontend:** ReactJS for an interactive UI.
* **Backend:** Flask API for handling requests.
* **Database:**
  + **ChromaDB** for document indexing & retrieval.
  + **MongoDB** for structured storage of user-generated queries & summaries.
* **AI Model:** RAG for text summarization and Q&A.
* **Hosting:** The system will be deployed on a **cloud-based infrastructure** with GPU support for efficient AI processing.

# 2.4 User Characteristics

# The system is designed for a diverse range of users, each with varying levels of technical expertise. Students and researchers can use the summarization and Q&A features to quickly grasp research content. Developers and data scientists can use the platform’s API to integrate its functionalities into other applications. Educators and academic professionals can utilize the chatbot to create learning modules based on research papers. Given these different user profiles, the system is built with an intuitive interface while providing advanced customization options for more experienced users.

This system is designed for a wide range of users, including:

* **Students & Researchers**
  + Need **quick summaries** to understand complex papers.
  + Use the **chatbot to clarify research concepts**.
* **Educators & Professors**
  + Extract key points for **teaching and academic discussions**.
  + Use summaries to **create simplified lecture materials**.
* **Developers & AI Enthusiasts**
  + Interested in **fine-tuning AI models** using the system.
  + Can integrate system **APIs for further research automation**.

# 2.5 Design and Implementation Constraints

# There are certain constraints to consider in the system’s design and implementation. The model needs to handle large research papers efficiently while maintaining low response time for both summarization and chatbot queries. The backend should support multiple concurrent requests, ensuring scalability for widespread adoption. Data privacy regulations must be followed to protect sensitive research data, meaning files should be encrypted and auto-deleted after processing. The system should also adapt to different research domains, requiring a flexible approach to training and fine-tuning the AI model.

The system must address the following constraints:

1. **Performance Constraints**
   * The AI model should **generate summaries in under 5 seconds**.
   * Chatbot responses should be **delivered within 2 seconds**.
2. **Scalability Constraints**
   * The system must support **multiple users simultaneously**.
   * The backend should handle **large document processing** without slowing down.
3. **Security Constraints**
   * Uploaded research papers must be **stored securely**.
   * User **authentication is required** to access chatbot features.
4. **Data Privacy Constraints**
   * Documents should be **deleted automatically** after processing.
   * No user data should be **stored permanently** without consent.

# 2.6 User Documentation

# User documentation will include step-by-step guides on how to use the platform effectively. This includes instructions for uploading research papers, accessing generated summaries, and interacting with the chatbot. The documentation will also provide troubleshooting steps for common issues such as file format errors, slow response times, and API connectivity problems. Developers will have access to a technical API guide for integrating the system’s functionalities into their own applications.

To help users navigate the system, **detailed documentation** will be provided, covering:

* **Step-by-step guide** on how to:
  + Upload research papers.
  + Access AI-generated summaries.
  + Use the chatbot for Q&A.
* **FAQs Section** to address common issues like:
  + Why is my document not uploading?
  + How do I ask a chatbot a follow-up question?
* **API Documentation** for developers who want to:
  + Integrate the system into **third-party applications**.
  + Use the **summarization and chatbot services** via API requests.

# 2.7 Assumptions and Dependencies

It is assumed that **users will provide well-structured research papers** for processing, as poorly formatted documents may affect summarization accuracy. The system relies on **continuous internet connectivity** for API-based interactions and cloud-based model execution. Dependencies include **Flask for backend services, ChromaDB for document indexing, and LangChain for AI-driven conversations**. The success of the system also depends on periodic updates to the AI model to ensure accuracy across various academic fields.

* The system assumes that:
  + Users will upload **structured, machine-readable documents**.
  + Internet access is available for **API-based model inference**.
* Dependencies include:
  + **Flask & ReactJS** for web application functionality.
  + **ChromaDB** for document retrieval and storage.
  + **Cloud-based AI models** for text summarization and chatbot interactions.

**3. External Interface Requirements**

# 3.1 User Interfaces

The platform will feature an intuitive and visually engaging user interface, consisting of:

* A **document upload section** with drag-and-drop functionality.
* A **summary display panel** where users can view AI-generated research paper summaries.
* A **chatbot interface** allowing users to ask questions and receive context-aware responses.

# 3.2 Hardware Interfaces

* The system is optimized for **desktops, laptops, and mobile devices**.
* It requires a **minimum 1.7 GHz processor and 2GB RAM** for basic operation.
* A **GPU-enabled server** is necessary for handling AI processing efficiently.

# 3.3 Software Interfaces

* **Backend:** Flask API for managing requests and processing research papers.
* **Database:** MongoDB and ChromaDB for storing documents and retrieval-based queries.
* **Frontend:** ReactJS for dynamic content rendering

# 3.4 Communication Interfaces

* Uses **RESTful APIs** for seamless interaction between frontend and backend.
* Secure **HTTPS protocols** to encrypt data transmission.
* Chatbot queries are processed using **LangChain’s API**, ensuring context-aware responses.

**4. System Features**

**4.1 Research Paper Uploading**

##### 4.1.1. Description and Priority

This feature allows users to upload **PDF or DOCX** research papers for summarization. The system processes the document and extracts meaningful content while maintaining structural integrity. This is a **High Priority** feature, as it forms the **core functionality** of the system.

##### 4.1.2. Stimulus/Response Sequences

* Stimulus: A user uploads a research paper.
* Response: The system processes the document, extracts text, and prepares it for summarization.

##### 4.1.3. Functional Requirements

* **REQ-1:** The system must support **PDF and DOCX file formats** for document uploads.
* **REQ-2:** The system should extract **structured text while preserving important formatting**.
* **REQ-3:** The system must notify the user once **file processing is complete**.

# 4.2 Summarization using RAG Model

###### 4.2.1. Description and Priority

This feature enables **automated text summarization** using the **Retrieval-Augmented Generation (RAG) Model**. The system retrieves **relevant sections** of the research paper and generates a structured summary, highlighting key findings, methodologies, and conclusions. It is a **High Priority** feature for ensuring effective knowledge extraction.

###### 4.2.2. Stimulus/Response Sequences

* Stimulus: A user requests a summary of an uploaded document.
* Response: The system processes the document, retrieves relevant sections, and generates a summary

###### 4.2.3. Functional Requirements

# REQ-4: The system must divide long research papers into meaningful text chunks for efficient processing.

# REQ-5: The RAG Model should retrieve and summarize key findings, advantages, and limitations.

# REQ-6: The generated summary must be structured and easy to read.

# 4.3 Chatbot-Based Q&A Interaction

###### 4.3.1. Description and Priority

This feature allows users to ask **research-related questions** and receive **context-aware answers** based on the uploaded paper. The chatbot uses **retrieval-based AI** to fetch precise responses, helping users understand complex concepts. This is a **High Priority** feature for enhancing user interaction.

###### 4.3.2. Stimulus/Response Sequences

* Stimulus: A user asks a question related to the uploaded document.
* Response: The chatbot retrieves relevant information and provides a contextually accurate answer.

###### 4.3.3. Functional Requirements

* **REQ-7:** The system must allow users to enter **queries in natural language**.
* **REQ-8:** The chatbot should provide **accurate, document-specific answers**.
* **REQ-9:** The chatbot must support **follow-up questions and maintain conversation flow**.

**4.4 Download Options**

Users can download the generated summary in PDF or Word formats for use.

**4.5 User Account & History**

Users can create accounts to track their uploads and summaries.

They can view past uploads and summaries in their account history.

## 5. Nonfunctional Requirements

# 5.1 Performance Requirements

The system must ensure **fast and efficient performance** when processing research papers. Summarization should be **completed within 5 seconds**, and chatbot responses should be **generated in under 2 seconds**. The system should handle **multiple concurrent users** without noticeable slowdowns.

# 5.2 Safety Requirements

To maintain **system integrity and user trust**, all uploaded documents should be **securely stored and auto-deleted after processing**. The system must also have **automatic backups** of chatbot interactions to prevent data loss

# 5.3 Security Requirements

The system should implement **end-to-end encryption** for all **user-uploaded documents**. Role-based access control must be used to ensure **only authorized users** can access research content. All chatbot interactions must be **logged for security auditing**.

# 5.4 Software Quality Attributes

* **Reliability:** The system should maintain **99.8% uptime**.
* **Scalability:** It must **support increasing numbers of users** without affecting performance.
* **Usability:** The interface must be **intuitive and user-friendly**.
* **Maintainability:** The system should have a **modular design for easy updates**.

# 5.5 Business Rules

* Users must **cite AI-generated summaries** when using them in academic work.
* The chatbot should provide **fact-based answers**, avoiding speculative or misleading responses.
* All AI-generated responses must **include references to document sections** for credibility..

**6. Other Requirements**

# 6.1 User Registration and Authentication:

The system should allow users to **register** with their email and password and **authenticate securely** before accessing features like research paper summarization and chatbot interaction. To enhance security, **two-factor authentication (2FA)** should be available. Users should also have an option to reset passwords securely.

# 6.2 Search and Filter Functionality:

# Users must be able to search uploaded documents and filter summaries by keywords, research topics, or date. The system should allow filtering based on research domain, author, publication year, and key concepts extracted from the document.

# 6.3 Research Paper Summarization Portal:

# The system should provide a dedicated portal where users can view all previously uploaded research papers and their corresponding summaries. Users should have options to download summaries and delete old documents after processing.

# 6.4 AI Chatbot for Research Assistance:

# A chatbot should assist users by answering document-specific queries. The chatbot should use RAG-based retrieval to provide accurate, research-driven responses. Additionally, it should allow users to ask follow-up questions while retaining conversational context.

# 6.5 Review and Feedback System:

Users should be able to **rate and provide feedback** on chatbot responses and AI-generated summaries. This feedback will help improve **the accuracy of future AI responses** and enhance system usability.

# 6.6 Document Processing and Cleanup:

To maintain **data security and privacy**, uploaded research papers should be **automatically deleted after processing** unless the user chooses to save them for future reference. The system should also allow users to **manually delete documents** at any time.

# 6.7 Mobile Accessibility:

The system should provide **real-time alerts and notifications** for various actions, including:

* **File upload completion.**
* **Summary generation notification.**
* **Chatbot follow-up reminders.**
* **Document deletion confirmations.**

Users should be able to **customize notification preferences** within their account settings.

# 6.8 Notifications and Alerts:

The research summarization system should be fully **optimized for mobile devices**, allowing users to upload documents, view summaries, and interact with the chatbot on **smartphones and tablets**. The mobile version should maintain the same usability as the desktop version.

# 6.9 Integration with External Libraries:

## The system should have the potential to integrate with online research databases such as Google Scholar, ArXiv, and IEEE Xplore. This will enable users to directly fetch research papers from external sources for summarization.

## Appendix A: Book Bank Glossary

* **Research Paper Summarization:** The process of extracting key insights, findings, and conclusions from academic research papers.
* **RAG Model (Retrieval-Augmented Generation):** A deep learning model that combines retrieval-based search with text generation for improved response accuracy.
* **ChromaDB:** A vector database used for indexing and retrieving relevant text from research papers.
* **Contextual Memory:** A feature that allows the chatbot to remember previous interactions to improve multi-turn conversations.
* **Fine-tuning:** The process of optimizing an AI model for specific tasks by training it on domain-specific data.

## Appendix B:

**Use Case template:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | RPS-20250227 | | |
| Use Case Name | Research Paper Summarization and Chatbot Q&A | | |
| End Objective | To enable users to upload research papers, generate concise AI-driven summaries, and interact with a chatbot for detailed explanations. | | |
| Created by | 1.Nalla Shruti    2.Vamshi Deepak  3.Karunya  4.Maanas  5.Niharika | On (date): | February27, 2025 |
| User/Actor | * Student/Researcher: Uploads documents, reads summaries, asks chatbot questions. * Administrator: Manages document processing and monitors AI performance. * System: Processes research papers, generates summaries, and provides chatbot responses. | | |

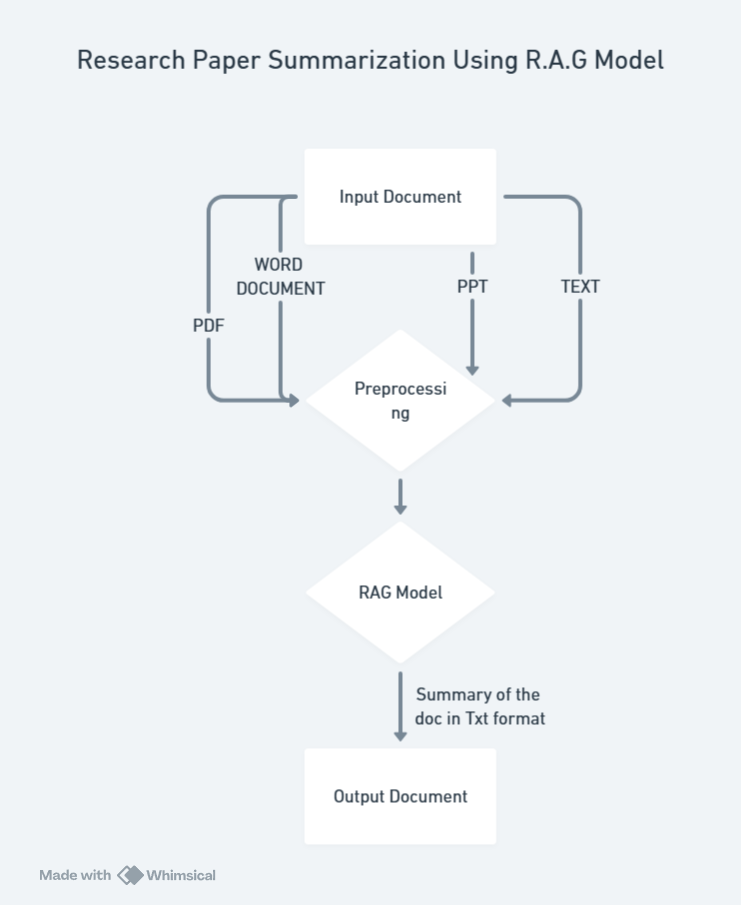
|  |  |  |  |
| --- | --- | --- | --- |
| Trigger | A user initiates an action, such as uploading a research paper, requesting a summary, or asking the chatbot a question. | | |
| Basic/Normal Flows | | | |
| User Actions | | Admin Actions | System Actions |
| The user uploads a research paper in PDF/DOCX format.  The user requests an AI-generated summary.  The user interacts with the chatbot, asking research-related questions.  The user downloads the generated summary. | | The admin monitors system performance and data processing.  The admin ensures AI model updates are applied correctly.  The admin reviews chatbot performance metrics.  The admin ensures compliance with data security policies. | The system extracts text and processes the document.  The system uses the RAG model to retrieve relevant sections and generate a summary.  The chatbot retrieves and generates context-aware responses.  The system allows users to save or delete their summaries. |
| Exception Flows | | | |
|  | |  |  |
| User Actions | | Admin Actions | System Actions |
| The user tries to upload an unsupported file format.  The user requests a summary, but the document contains unreadable text.  The user requests a summary, but the document contains unreadable text.  The user requests a summary for a document exceeding system size limits. | | The admin troubleshoots user complaints about errors.  The admin identifies issues with OCR processing.  The admin reviews chatbot performance logs.    The admin reviews chatbot performance logs. | The system displays an error message: "Invalid file format. Please upload PDF or DOCX.”  The system alerts the user: "Text extraction failed. Please upload a clearer document.”  The chatbot responds: "I can only answer questions related to this research paper."  The system notifies the user: "File size too large. Please upload a document under 10MB." |

## 

## Appendix C: To Be Determined List

* **Customization Features:** Allow users to modify summarization length or highlight key sections.
* **Multi-language Support:** Implement support for summarization in **multiple languages** beyond English.
* **External Database Integration:** Connect with **Google Scholar, ArXiv, and IEEE Xplore** for automatic paper retrieval.
* **Advanced Chatbot Capabilities:** Enable **follow-up query recommendations** based on user interactions.
* **Summarization Metrics:** Display AI **accuracy scores** and **confidence levels** for generated summaries.
* This **list helps track important decisions that need further development** as the project progresses.

WORK FLOW DIAGRAM:



**ARCHITECTURE** DIAGRAM:

