

**University of Szeged**

**Faculty of science and informatics**

**Department of Computer Algorithms and Artificial Intelligence**

MSc Thesis

# UNIQUERY-A Rag based Query Application

*Supervisor: Prepared By*:

András London Proma Kanungoe

Assistant Professor Computer Science, MSc Student

## Task description

As a student, I understand how hard it can be for other students to figure out what they need to do in school, especially when they are new to the institution. Many students miss out on essential information, pick the wrong classes or credit systems, or miss out on chances that are available because the paperwork is imprecise or dispersed. Also, because classes and departments are spread out over many places, it might be hard to identify the proper one, which makes things even more stressful, especially for new students. To fix this, the app applies a Retrieval-Augmented Generation (RAG) system driven by Large Language Models (LLMs). This enables students input university papers (including curricula, laws, and fee structures) and ask inquiries in plain English. The chatbot then finds the right responses, which makes it easier and faster for students to get vital academic and logistical information without having to read through long documentation. The major goal of this project is to create UNIQUERY, a multimodal Retrieval-Augmented Generation (RAG) based assistant that makes it easier for college students to find academic and administrative information. The system should be able to take questions in diverse formats, like voice, document uploads, and photos, and give accurate, context-aware answers based on approved university documents. To do this, the project is focused on creating a single platform that brings together several advanced AI parts, such as speech-to-text for processing spoken questions, PDF text extraction and retrieval for handling academic documents, image understanding for interpreting photos of university buildings or printed content, and natural language generation for making clear, useful answers.   
The task involves making a user-friendly interface where students can record audio, upload PDFs, or send photographs and then get answers in both text and speech. The main goal is to create an assistant that helps students make smart academic choices, cuts down on confusion, and cuts down on the time they spend looking for documents. UNIQUERY should help students deal with the frequent problems they have when they have to deal with university systems, rules, and academic paperwork.

# Abstract

University students often struggle to navigate complex academic systems, scattered information sources, and unclear administrative procedures. These challenges highlight the need for a more advanced system capable of handling diverse types of queries while providing accurate, relevant, and personalized responses. **UniQuery** addresses this need by introducing a multimodal university information retrieval system powered by **LLaMA Vision**, **Groq Whisper**, **Retrieval-Augmented Generation (RAG)**, and **Gradio**.

UniQuery supports three modes of interaction:

1. **Voice queries**, transcribed using Whisper v3;
2. **Image-based queries**, interpreted through LLaMA Vision;
3. **Document-based Q&A**, enabled by semantic retrieval and RAG.

By combining speech recognition, computer vision, and generative LLM capabilities, the system delivers grounded and context-aware answers to spoken questions, university-related PDFs, campus board images, schedules, professor announcements, and more. A text-to-speech module produces natural audio responses, creating a complete interactive loop.

This thesis presents the motivation, system design, implementation methodology, evaluation, and implications of UniQuery. Results show that UniQuery significantly improves information accessibility—especially for international and first-year students—by offering fast, accurate, and context-sensitive assistance. The project demonstrates the growing role of multimodal AI systems in educational environments and identifies key opportunities for future enhancements, including personalization, multilingual capabilities, and autonomous university data integration.

[Task description 2](#_bookmark0)

[Abstract 3](#_bookmark1)

[Introduction 6](#_bookmark2)

[Chapter 1:Background and Literature Review 7](#_bookmark3)

* 1. [Overview of Information Systems in Universities 7](#_bookmark4)
  2. Chatbots in Higher Education [9](#_bookmark5)
  3. Large Language Models [11](#_bookmark8)
     1. Generative AI [13](#_bookmark9)
     2. Retrieval-Augmented Generation (RAG) [15](#_bookmark10)
  4. Vision-Language Models (LLaMA Vision)
  5. Speech-to-Text Systems (Whisper v3)
  6. Multimodal Conversational Systems
  7. Summary of Related Work
  8. Research Gap

[Chapter 2: System Design and Architecture 18](#_bookmark11)

* 1. [Overview of UniQuery Architecture 18](#_bookmark12)
  2. Audio Processing Pipeline [19](#_bookmark13)
  3. Image Processing Pipeline [20](#_bookmark14)
  4. PDF Processing & RAG Pipeline [22](#_bookmark15)
     1. Text Extraction
     2. Chunking Mechanism
     3. Semantic Embeddings
     4. Similarity Retrieval
  5. LLM Coordination Layer (LLaMA on Groq)
  6. Text-to-Speech Generation
  7. Gradio User Interface
  8. Data Flow Diagram
  9. Security and Privacy Considerations

[Chapter 3: Methodology and Implementation 27](#_bookmark19)

* 1. Development Workflow [27](#_bookmark20)
  2. Programming Languages and Frameworks [27](#_bookmark21)
  3. UniQuery Code Structure
  4. Voice Query Implementation
  5. Image Query Implementation
  6. PDF RAG Implementation
  7. Embedding Strategies
  8. Coordinator Response Generation
  9. Error Handling and Fallbacks
  10. Deployment via Gradio
  11. Performance Optimization

[Chapter 4: Evaluation and Testing 37](#_bookmark24)

* 1. Testing Methodology [37](#_bookmark25)
  2. Dataset and Documents Used [37](#_bookmark26)
  3. Functional Testing [37](#_bookmark27)
  4. PDF Query Accuracy
  5. Image Interpretation Accuracy
  6. Voice Query Transcription Results
  7. User Study & Feedback
  8. Limitations Noticed During Testing

[Chapter 5: Discussion 39](#_bookmark28)

5.1 System Strengths

5.2 Usefulness for Students and Staff

5.3 Challenges Faced During Development

Chapter 6: Conclusion and Future Work

6.1 Summary of Contributions

6.2 Future Improvements

[Figures References 41](#_bookmark29)

[Appendices 42](#_bookmark30)

[Acknowledgment 50](#_bookmark33)

[Declaration 51](#_bookmark34)

# Introduction

UNIQUERY combines a number of advanced AI parts to make voice, document, and picture inputs for university information access work smoothly together. Its Retrieval-Augmented Generation (RAG) architecture bases responses on real academic papers. This cuts down on hallucinations and the time students have to spend searching through long PDFs and rules by hand. With whisper-based voice recognition, students can ask questions out loud, making the system more accessible. A multimodal LLM, on the other hand, can understand a variety of input formats. Sentence-BERT embeddings are used to build the retrieval pipeline, which picks out semantically relevant text chunks. Each module is checked separately and together to make sure it is strong and consistent. Universities are growing their digital environments all the time. Systems like UNIQUERY offer reliable, scalable support tools that make things easier to use and improve the overall academic experience.

## Chapter 1: Background and Literature Review

This chapter presents the theoretical, technological, and conceptual foundations that support the development of UniQuery, a multimodal intelligent university information assistance system. The chapter situates UniQuery within existing academic challenges, explores the evolution of AI-based assistant technologies, and provides a comprehensive overview of the scientific literature on Large Language Models (LLMs), speech recognition systems, vision–language models, and Retrieval-Augmented Generation (RAG). Additionally, it identifies critical research gaps in current educational information systems and highlights how UniQuery addresses these gaps through an integrated multimodal design. Every universities generate large amounts of academic and administrative information, including course regulations, fee structures, academic calendars, program requirements, graduation rules, and departmental announcements. This information is often stored across multiple disconnected systems,PDF documents, official websites, printed posters on campus boards, learning management systems, and student portals. As a result, students frequently struggle to find accurate and up-to-date information when they need it. The difficulty is even greater for international students, who may face language barriers or unfamiliar academic structures.Manual search mechanisms used in universities, such as keyword-based search engines or FAQ pages, are limited in their ability to understand natural language queries, interpret images, or process complex documents. Students must manually read long PDFs or multiple website pages to find answers which is a time consuming process.To address this gap, UniQuery introduces a multimodal information retrieval system that combines voice, image, and document-based queries into a single platform. The system leverages Whisper for speech recognition, LLaMA Vision for image understanding, semantic text embeddings for document retrieval, and a generative LLM for contextual reasoning. This chapter presents a review of related research and technologies that form the foundation of UniQuery.

## Overview of Information Systems in Higher Education

Universities today operate within highly complex digital ecosystems that support diverse administrative, academic, and communication needs. Over the past decade, higher education institutions have increasingly relied on integrated information systems to streamline academic processes and improve student services. Platforms such as Neptun, Coospace, Moodle, Canvas, Blackboard, and custom institutional portals serve as central hubs for course registration, grade management, academic calendars, assignment submissions, and communication with faculty. Their primary purpose is to provide structured digital environments where students can access essential academic resources and administrative services.

Despite their widespread adoption, these systems share several limitations. They generally follow a menu-driven, non-conversational architecture, meaning users must navigate predefined page structures, hierarchical menus, and document repositories to locate information. Many of the most important university policies, such as curriculum rules, credit structures, and examination regulations, are distributed as long-form PDF documents, often exceeding 50–100 pages. Students are expected to manually search through these documents, frequently without semantic search capabilities or natural language interfaces.

In addition, universities rely heavily on physical notice boards, classroom schedules, departmental posters, and printed announcements. Information distributed in visual formats remains inaccessible to standard digital systems that cannot interpret images or extract textual content from pictures. These limitations create inefficiencies and lead to student confusion, especially during critical periods such as orientation, course selection, examination scheduling, and administrative deadlines.

For international students, the challenge is further amplified due to unfamiliar terminology, academic conventions, and language barriers. As a result, students frequently experience difficulty locating accurate and timely information, despite the availability of multiple digital systems.In summary, while existing university information systems are functional, they lack natural language search, multimodal support, and intelligent information retrieval mechanisms, creating a clear need for advanced AI-driven solutions such as UniQuery.

## Chatbots in Higher education

Chatbots have become increasingly common in customer service, healthcare, finance, and e-commerce, and universities have gradually begun adopting them to support student information queries. Early educational chatbots were typically rule-based systems that relied on manually defined patterns and responses. These chatbots functioned by matching predefined keywords or sentence structures and returning fixed outputs, making them suitable only for limited FAQ-style interactions such as campus directions, office hours, or simple administrative FAQs.

However, rule-based systems have significant limitations. They cannot support natural conversation, they fail to understand the context behind queries, and they cannot process or summarize long academic documents. Their responses are inflexible, cannot adapt to diverse student questions, and break easily when faced with queries that differ slightly from preprogrammed patterns.

With advancements in machine learning, universities began exploring intent classification–based chatbots, which could categorize student queries into predefined intents. Although more flexible than rule-based systems, they still lacked deep reasoning, could not analyze PDFs or images, and were limited to text-only interactions.

LLM-based assistants in education offer more personalized support and can handle a wider range of student queries. However, on their own, they still suffer from shortcomings:

* they may hallucinate information
* they do not inherently know university-specific rules
* they cannot extract information from PDFs or images
* they are not grounded in real institutional documents

To overcome these limitations, modern educational chatbots require multimodal AI, integrating LLMs with retrieval systems, speech processing, and vision models. UniQuery is designed specifically to fill this need.

## Large-Language Model

## Large Language Models (LLMs) are advanced deep learning systems trained on massive textual corpora from books, websites, scientific articles, and many other sources. They excel at predicting and generating human-like text, making them powerful tools for semantic understanding, question answering, and conversation. Popular LLM families include GPT, LLaMA, Claude, PaLM, and Mistral, each offering advanced natural language reasoning capabilities.

## LLMs have proven valuable in academic settings due to their ability to:

## interpret complex queries

## summarize long documents

## provide detailed explanations

## assist with academic writing

## understand context-rich questions

## However, LLMs also exhibit notable weaknesses in the university information context:

## They may generate confident but incorrect answers (hallucinations).

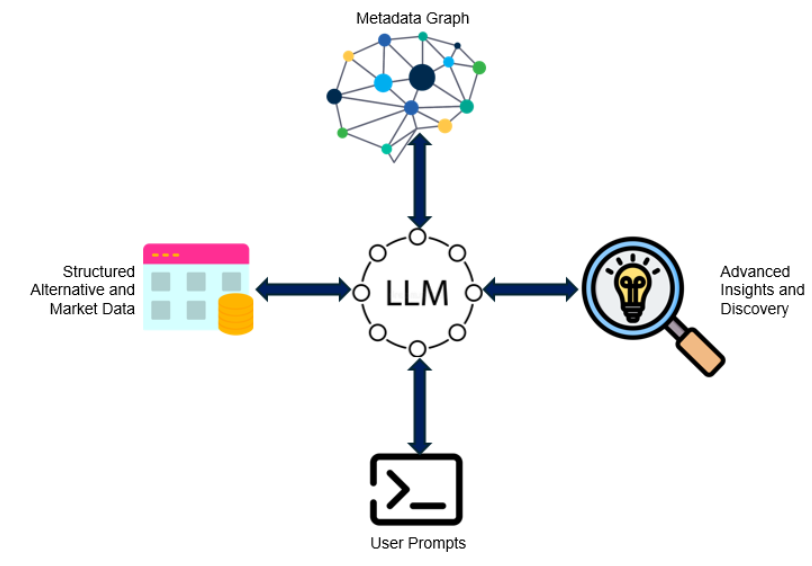
## They cannot automatically integrate with institutional documents.

## They lack knowledge of localized university structures and policies.

## They require grounding mechanisms to ensure factual accuracy.

## They do not inherently support multimodal inputs such as images and speech.

Because of these limitations, LLMs alone cannot serve as reliable university assistants. Instead, they must be combined with additional systems that bring document retrieval, speech recognition, and computer vision into the pipeline.



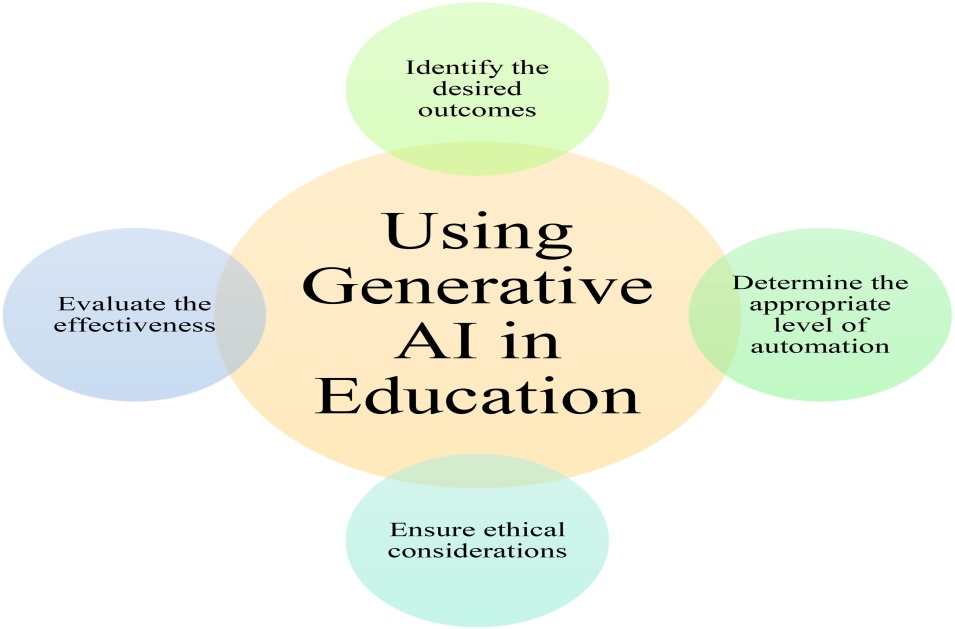
## Generative AI Models

Generative AI refers to models that can produce new content such as text, audio, or images based on learned patterns in data. Generative language models represent one of the most advanced forms of generative AI. They provide human-like conversational abilities and semantic understanding by predicting the next word in a sequence given its context.

In the context of education, generative AI offers:

* natural dialogue systems
* explanations of academic concepts
* contextualized recommendations
* ability to respond to vague or incomplete queries
* flexible interaction across a wide variety of topics

Generative AI provides the "reasoning engine" of UniQuery, enabling it to interpret multimodal inputs and convert them into coherent, context-aware answers.



## Retrieval-Augmented Generation (RAG)

RAG is a hybrid AI technique designed to overcome the limitations of standalone generative models. It integrates a retrieval system with a generative language model, creating a pipeline where retrieved documents guide the model’s output.

The RAG process in UniQuery includes:

* Extracting text from PDFs or university documents
* Splitting the text into semantic chunks
* Embedding each chunk using Sentence-BERT
* Storing these embeddings in a vector database
* Embedding the user’s question
* Retrieving the most relevant document chunks
* Feeding these chunks into the LLM as context

This ensures that the LLM’s response is accurate, grounded, and directly supported by official university documents.

RAG is especially important for answering queries about:

* academic regulations
* credit requirements
* program structures
* admission policies
* fee payment rules
* Erasmus and mobility guidelines

UniQuery uses RAG to prevent hallucinations and ensure reliability.



* 1. **Vision–Language Models (LLaMA Vision)**

Vision–language models represent one of the most significant advancements in artificial intelligence because they enable machines to jointly understand images and natural language. These models combine computer vision capabilities with linguistic reasoning, allowing them to interpret visual information and describe it meaningfully. LLaMA Vision, which is integrated into UniQuery, extends the capacity of language models by enabling them to analyse images in a way that resembles human interpretation. Unlike traditional systems that rely solely on text input, LLaMA Vision can examine visual structures, recognise embedded text, infer contextual meaning from posters, timetables, notices, and images, and translate this understanding into coherent verbal explanations. This is particularly relevant in university environments, where a substantial amount of important information is communicated through physical formats such as printed schedules, classroom allocations, event posters, and notice boards. These visual artefacts often remain inaccessible to digital systems, creating a gap between physical academic information and student access. LLaMA Vision bridges this gap by providing the ability to read and interpret uploaded images, giving UniQuery the capacity to understand what is displayed on a poster, identify dates, venues, or timings from a timetable, and extract meaningful textual and contextual information even when the content appears in non-standard layouts or mixed visual formats. This capability makes UniQuery substantially more powerful and relevant within academic environments where visual communication is frequent and essential.

* 1. **Speech-to-Text Systems (Whisper v3)**

Speech-to-text technology has evolved rapidly in recent years, allowing computational systems to convert spoken language into accurate textual representations. Whisper v3 is one of the most advanced speech recognition models, and its integration into UniQuery provides students with an additional modality of interaction that is more natural and intuitive than typing. Whisper v3 is particularly effective because it offers high-quality transcription even in noisy environments, supports a wide range of accents and dialects, and delivers consistent accuracy even when speech is fast, informal, or imperfectly articulated. This makes it an ideal choice for university contexts where international students with diverse linguistic backgrounds frequently seek academic information and may prefer speaking to typing. Voice-based interaction also supports accessibility for users who have visual impairments, physical limitations, or learning difficulties that make reading and typing challenging. Whisper v3 allows students to ask complex questions verbally and have their speech immediately converted into structured text, which the UniQuery system can then process through its reasoning and retrieval mechanisms. By enabling seamless speech recognition, UniQuery becomes a more inclusive and user-friendly system, offering a more natural communication experience that aligns with contemporary expectations for voice-enabled technology in daily digital interactions.

* 1. **Multimodal Conversational Systems**

Multimodal conversational systems represent the next stage in the evolution of artificial intelligence, enabling machines to process, combine, and reason across multiple forms of input simultaneously. Rather than limiting the user to typed text, multimodal systems accept speech, images, PDFs, scanned documents, and other data types, all of which can be interpreted within one unified conversational pipeline. This multidimensional capability is critical in university environments because academic information is rarely presented in a single modality. Students must navigate a mixture of written regulations, visual posters, classroom boards, email attachments, and spoken instructions. A multimodal system, such as UniQuery, is designed to reflect this reality by integrating Whisper for speech processing, LLaMA Vision for image understanding, and a RAG-based retrieval mechanism for document interpretation. The combined effect of these technologies is a highly flexible assistant that can understand student questions regardless of the format in which the relevant information exists. The system can listen to a spoken query, read a PDF document, interpret an uploaded photograph of a campus announcement, and merge the insights from all these sources into a single coherent explanation. This fusion of modalities significantly enhances user experience and ensures that information retrieval mirrors the natural ways students interact with their environment. Ultimately, multimodal conversational systems transform static information delivery into a dynamic, human-like dialogue that adapts to the user's communication preferences.

* 1. **Summary of Related Work**

The existing body of research and technological development in academic information systems reveals a landscape dominated by traditional digital platforms, static portals, and isolated tools that address only small fragments of student needs. University portals and learning management systems are efficient for structured tasks, but they lack conversational understanding. Early rule-based chatbots offered minimal support, responding only to a limited set of predefined queries without the ability to interpret nuanced academic questions. Later intent-based chatbots improved classification but remained incapable of analysing long PDF documents, integrating external images, or engaging in complex reasoning. More recent advancements introduced large language model–based assistants, which brought significant improvements in conversational fluency and contextual understanding, but these models still suffered from hallucination issues and lacked grounding in institution-specific documents. Meanwhile, separate technologies such as OCR systems, PDF search engines, or generic voice assistants provided partial solutions but operated independently without integration. The academic literature shows no comprehensive system capable of combining conversational AI with PDF retrieval, image understanding, and speech recognition in a unified framework tailored specifically for university information retrieval. Most existing tools remain siloed, single-purpose solutions that fail to address the full complexity of academic information access.

* 1. **Research Gap**

A careful examination of current systems reveals a clear and significant research gap in the domain of university information support. No existing tool offers a seamless integration of text, voice, document, and image inputs within a single conversational interface. Although LLM-based chatbots are widely accessible, they do not have the capacity to ground their responses in official academic documents, which is essential for providing accurate and reliable information to students. Similarly, while speech recognition technologies exist independently, they are not embedded within a reasoning pipeline capable of understanding and responding to multimodal academic queries. Vision-language models have advanced rapidly, yet they are rarely combined with conversational systems in the educational domain. Furthermore, universities currently lack intelligent assistants that can extract information from both digital documents and physical sources such as notice boards or classroom posters. As a result, students are forced to manually navigate fragmented information ecosystems, often experiencing confusion and inefficiency. The absence of a unified multimodal assistant that integrates speech-to-text, visual understanding, and document-grounded reasoning represents a substantial gap in modern educational technology.

UniQuery directly addresses this gap by bringing together multiple advanced AI components into one cohesive system designed specifically to support university information retrieval in a reliable, accessible, and context-aware manner.

**CHAPTER 2: System Design and Architecture**

This chapter presents the complete system design and architectural foundation of UniQuery, detailing how its multimodal capabilities are implemented and how the different components work together to provide accurate and efficient university information retrieval. The chapter begins with an overview of the entire UniQuery architecture, explaining the major layers and functional modules that form the backbone of the system. It then describes each processing pipeline in depth, including the audio pipeline for speech-to-text conversion, the image pipeline for visual interpretation using LLaMA Vision, and the PDF processing pipeline supported by a Retrieval-Augmented Generation (RAG) framework. Subsections on text extraction, chunking, semantic embeddings, and similarity retrieval explain how documents are converted into searchable knowledge.

Also the chapter elaborates on the role of the LLM Coordination Layer, which includes outputs from all modalities and generates context-aware answers. The text-to-speech mechanism used for voice responses is described along with the design of the Gradio user interface that enables user interaction. A comprehensive data flow diagram shows the movement of information across different modules, and a final section discusses security and privacy considerations relevant to handling academic documents and user queries. Together, these sections shows a detailed understanding of UniQuery’s internal architecture, operational workflow, and the reasoning behind the system’s technical design choices.

* 1. **Overview of UniQuery Architecture**

The UniQuery architecture is designed as a structured, multilayered system that integrates speech recognition, image understanding, document analysis, and large language model reasoning into a single cohesive workflow. The architecture begins with the User Input Layer, where students can interact with the system using four distinct modalities voice, images, PDF documents, and text queries. These multiple interaction channels reflect the diverse ways academic information is distributed within universities. Once inputs are received, they are forwarded to the Preprocessing Layer, where modality-specific transformations take place. Audio signals are cleaned and formatted, uploaded images are normalized and encoded, and PDFs undergo text extraction to prepare them for deeper analysis.

The next stage in the architecture is the Processing Pipeline Layer, where each input type is analyzed by a dedicated AI subsystem. Voice recordings are transcribed using Whisper v3, a high-accuracy speech-to-text model capable of handling varied accents and noisy conditions. Image inputs are interpreted by LLaMA Vision, which extracts relevant textual and contextual information from posters, timetables, or notice boards. PDF documents are processed through a Retrieval-Augmented Generation (RAG) pipeline, which includes chunking, embedding generation, semantic similarity retrieval, and context retrieval. All processed outputs converge in the Coordinator LLM Layer, powered by LLaMA running on Groq’s accelerated computational backend..

Finally, the generated response is delivered through the Output Layer, where students receive both a written textual answer and an optional spoken response via text-to-speech synthesis. This multilayered design ensures that UniQuery operates as a fully multimodal and intelligent academic assistant capable of understanding, processing, and responding to diverse inputs in a seamless and integrated manner.

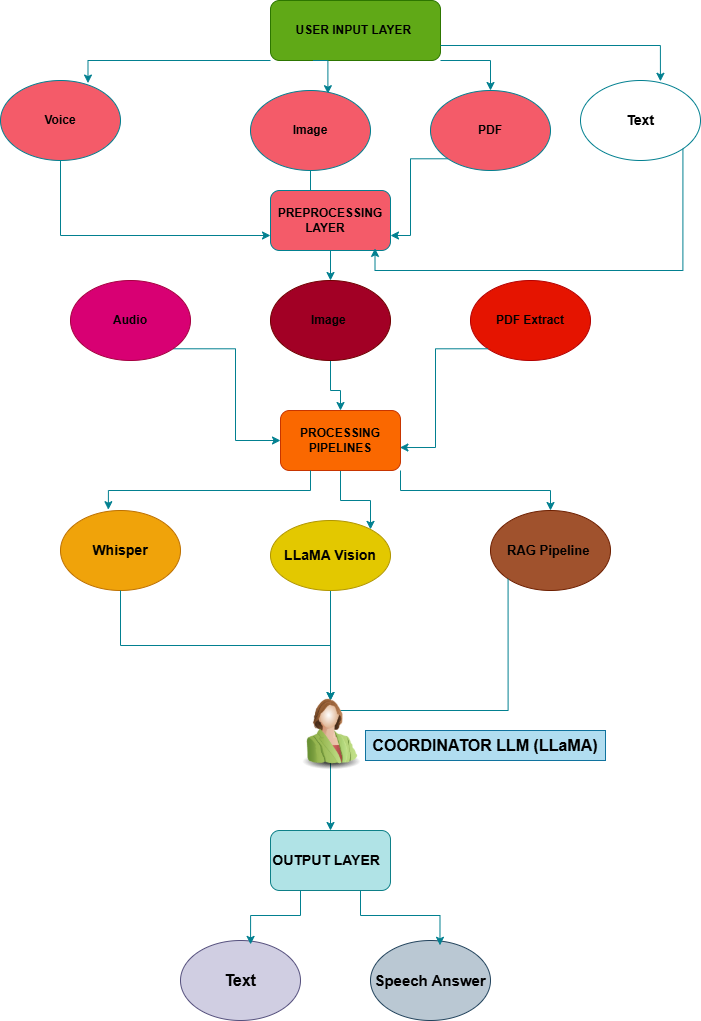


Figure 2.1: UniQuery Architecture

* 1. **Audio Processing Pipeline**

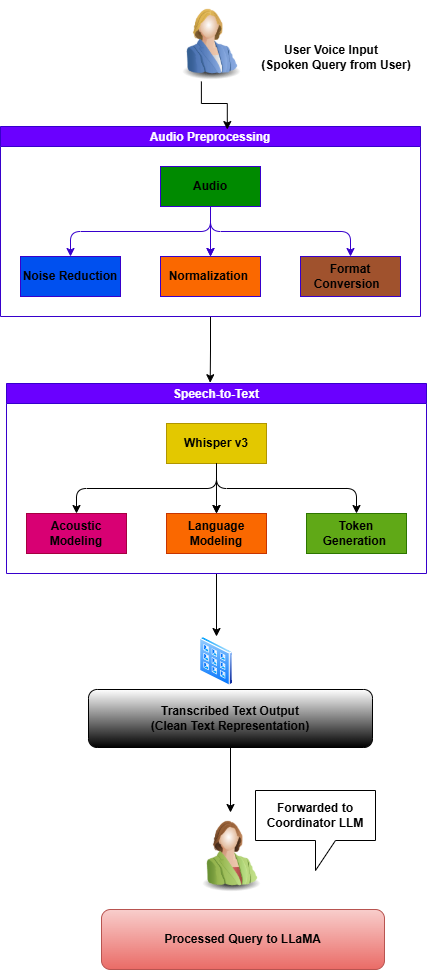
****

Figure 2.2: Audio Preprocessing Pipeline

The first step of UniQuery's audio processing pipeline, where a user's spoken query is prepared for precise speech-to-text conversion, is shown in the audio preprocessing diagram. Upon receiving a voice input from a pupil, the Audio Preprocessing module receives the raw audio signal first. Three crucial tasks are carried out by this module: format conversion, normalization, and noise reduction. By eliminating background distractions like wind, chatter, and echo, noise reduction makes sure the signal is clear enough to comprehend. By adjusting the audio amplitude to a constant level, normalization keeps transcription accuracy from being impacted by changes in speaking volume. Format conversion guarantees that the audio is converted into the precise encoding needed for the Whisper v3 model.

Following these procedures, the standardized and cleaned audio is sent to the Speech-to-Text module, where Whisper v3 generates tokens, models languages, and models acoustics to create a high-quality text transcription. This step produces a clear, machine-readable text query, which is subsequently sent to the Coordinator LLM for additional analysis. Overall, the illustration shows how UniQuery transforms imprecise, real-world voice recordings into trustworthy text that can be processed further.  
Following these procedures, the standardized and cleaned audio is sent to the Speech-to-Text module, where Whisper v3 generates tokens, models languages, and models acoustics to create a high-quality text transcription. This step produces a clear, machine-readable text query, which is subsequently sent to the Coordinator LLM for additional analysis. Overall, the illustration shows how UniQuery transforms imprecise, real-world voice recordings into trustworthy text that can be processed further.

* 1. **Image Processing Pipeline**

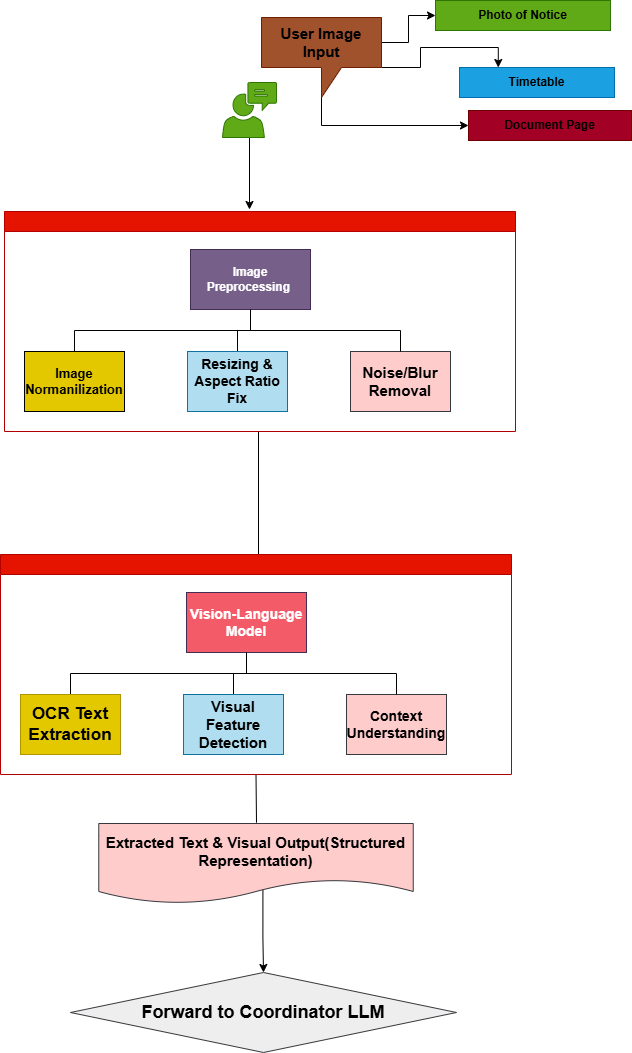


Figure 2.3: Image preprocessing Pipeline

The image processing pipeline enables UniQuery to interpret visual information such as university notices, classroom schedules, event posters, and other campus-related materials. This pipeline begins when the user uploads or captures an image containing academic information. Since real-world photographs often include distortions such as uneven lighting, shadows, blur, or angle variation, the first module in the pipeline is Image Preprocessing. Here, the system performs normalization to adjust brightness, contrast, and color values, ensuring the image is visually consistent for downstream analysis. Additionally, resizing and aspect ratio correction standardize the dimensions of the image while maintaining proportional structure. If an image contains noise or blurriness, the system applies denoising filters to sharpen the material and enhance the readability of embedded text.

After preprocessing, the cleaned image is passed to the LLaMA Vision model, which acts as the core vision-language engine of UniQuery. LLaMA Vision first performs OCR-based text extraction, identifying and reading textual content from the image, whether it appears as printed text, handwritten notes, or digital fonts. Beyond reading text, the model also performs visual feature detection, enabling it to interpret non-textual elements including tables, layout structures, icons, event banners, or schedule grids. Through context understanding, the model integrates visual clues with textual content to infer the meaning behind the image—such as detecting that a specific date is an exam day, or a highlighted room number is a classroom allocation.

The final stage of the pipeline generates a structured representation of the extracted information. This output includes the recognized text, key visual features, contextual interpretations, and additional semantic annotations. The structured data is then handed over to the Coordinator LLM, which integrates the extracted information with the user’s query to produce accurate, context-aware responses. By combining preprocessing, vision-language modeling, and contextual reasoning, the image processing pipeline allows UniQuery to seamlessly interpret and understand university-related visual materials, making it far more versatile than traditional text-only systems.

* 1. **PDF Processing & RAG Pipeline**

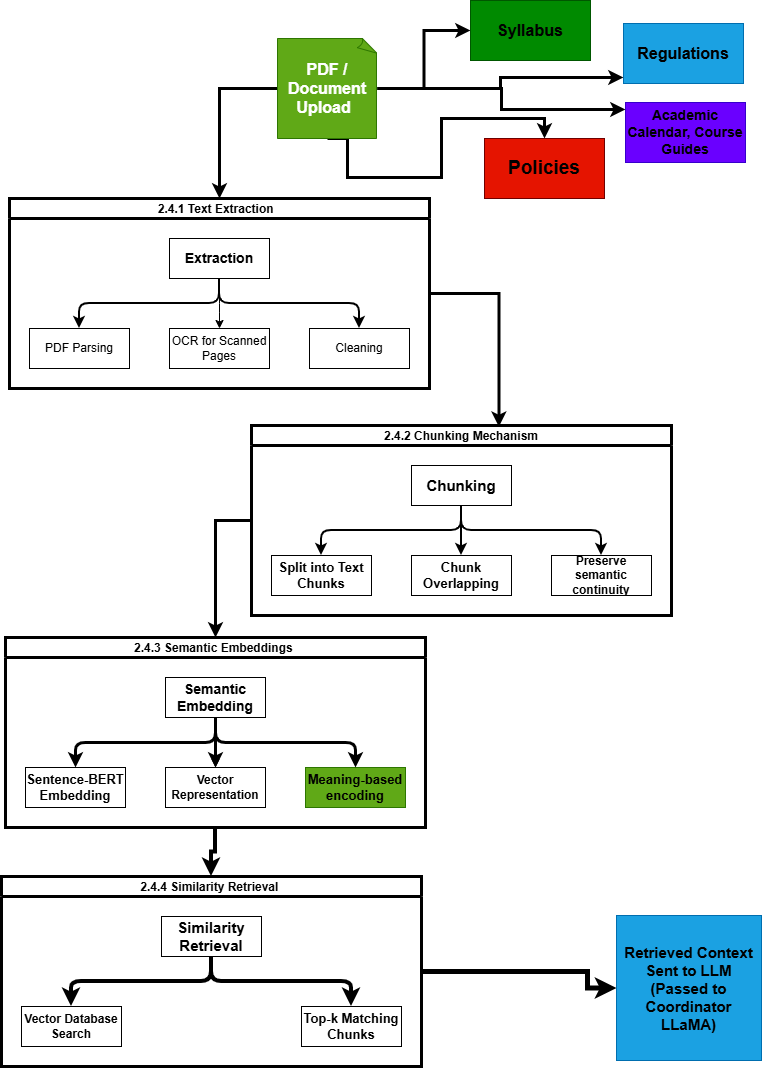
****

Figure 2.4: PDF Processing & RAG Pipeline

* 1. **PDF Processing & RAG Pipeline**

The PDF Processing and Retrieval-Augmented Generation (RAG) pipeline is the core mechanism that enables UniQuery to provide accurate, grounded, and document-based answers. This pipeline processes academic documents such as regulations, syllabus, curriculum guides, policy documents, and handbooks uploaded by the user. Because university documents are often lengthy, text-heavy, and structured in complex formats, the RAG pipeline ensures that information is systematically extracted, semantically organized, and efficiently retrieved during query processing.

**2.4.1 Text Extraction**

The first stage in the pipeline is text extraction, where the uploaded PDF undergoes parsing to convert its contents into machine-readable text. If the PDF contains selectable digital text, the system extracts it directly along with metadata such as headings, subtitles, and structural markers. For scanned or image-based PDFs, Optical Character Recognition (OCR) is applied to convert visual characters into text. This ensures that even printed, scanned, or photographed university documents can be fully processed. A cleaning step follows, where unnecessary line breaks, extra whitespace, inconsistent spacing, and formatting noise are removed to produce a clean version of the document suitable for analysis.

**2.4.2 Chunking Mechanism**

Once the text is extracted, it must be divided into smaller, semantically coherent units so that the RAG model can retrieve the most relevant information. This process, known as chunking, typically splits the document into sections of approximately 300–500 words, depending on the density and structure of the text. An overlapping strategy is applied so that consecutive chunks share some content, ensuring that important contextual details do not get cut off between segment boundaries. This approach maintains the logical flow across the document and improves retrieval accuracy.

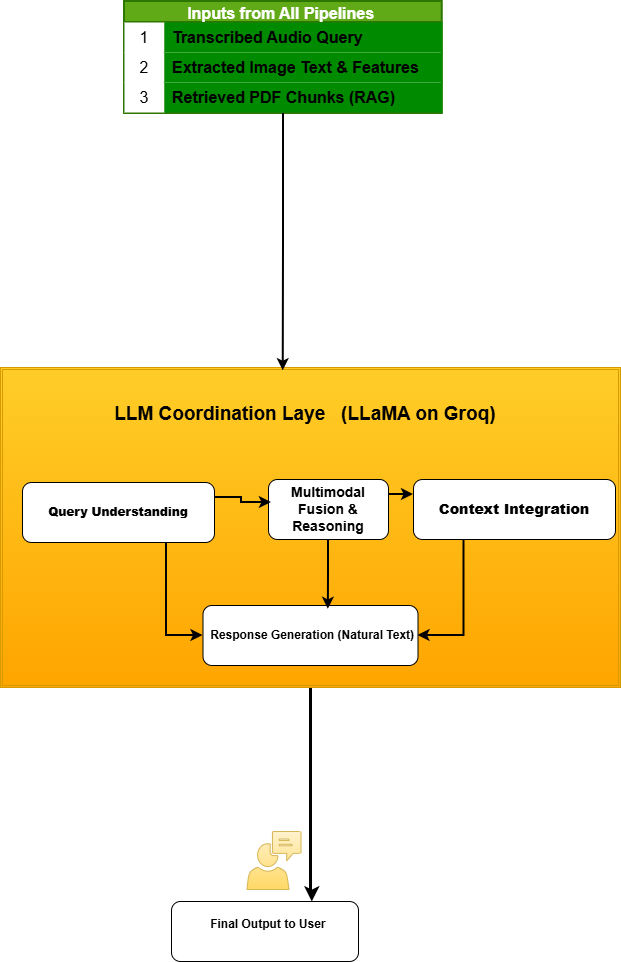
**2.4.3 Semantic Embeddings**

Each chunk is then converted into a vector representation using Sentence-BERT, a transformer-based embedding model optimized for semantic similarity tasks. The embeddings encode the meaning of the text rather than simply its keywords, making it possible for the system to match user queries with relevant document sections even when they use different words. These vector embeddings serve as the backbone of the retrieval process, as they enable the RAG system to understand the conceptual relationships between chunks and student queries.

**2.4.4 Similarity Retrieval**

When the student asks a question, the query is also converted into a vector using the same embedding model. The similarity retrieval module then compares the query vector against all stored chunk vectors using cosine similarity. This operation identifies the top-k most relevant chunks that contain the necessary information to answer the user’s question. These retrieved chunks form the factual grounding context for the response-generation stage. The context is forwarded to the Coordinator LLM (LLaMA), which integrates the extracted information with the user’s question and produces a precise, context-aware answer based solely on verified document content.

* 1. **LLM Coordination Layer (LLaMA on Groq)**

****

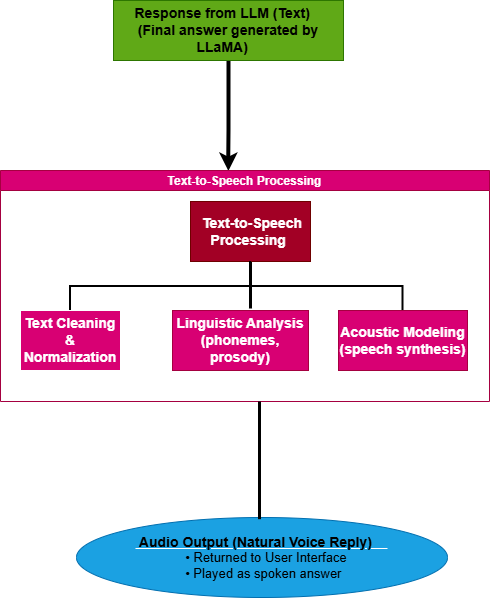
**Figure 2.5: LLM Coordination Layer (LLaMA on Groq)**

The LLM Coordination Layer forms the central reasoning engine of UniQuery. This layer is responsible for combining all processed inputs—transcribed audio queries, extracted image information, and retrieved document chunks—into a unified semantic representation that the system can reason over. The core of this layer is the LLaMA model deployed on Groq’s high-performance inference engine, which ensures extremely low latency and efficient processing of multimodal inputs.The first component in this layer is the Query Understanding module, which interprets the user’s intent based on the incoming question. It identifies the query type—whether it is about regulations, deadlines, course requirements, building locations, or event details—allowing the system to select the most relevant context.Next, the Multimodal Fusion and Reasoning module integrates information from all three input pipelines. If the user’s query contains both a spoken question and an uploaded image (e.g., “What does this notice mean?”), the model fuses these signals and reasons over them jointly. This component enables LLaMA to interpret complex queries that require combining text with visual elements or grounding answers in retrieved document passages.

The Context Integration module incorporates the retrieved RAG chunks, image interpretation results, and the user’s question into a coherent contextual frame. The system ensures that only verified and relevant information from university documents influences the final response, improving accuracy and minimizing hallucinations.

Finally, the Response Generation module produces a natural, human-like answer that is directly grounded in the provided context. The generated response is then forwarded for text delivery or converted into speech for voice output. By managing query intent, fusing multiple modalities, grounding the response in authoritative university documents, and producing a coherent final answer, the LLM Coordination Layer acts as the intellectual core of the UniQuery system.

* 1. **Text-to-Speech Generation**

****

**Figure 2.6: Text to Speech Generation**

The Text-to-Speech module converts the final text answer from the Coordinator LLM into a natural audio response. The process begins when the LLM outputs its text, which is first cleaned and normalized to ensure clear pronunciation. This includes correcting punctuation, expanding abbreviations, and formatting numbers or symbols properly. After cleaning, the text enters the linguistic analysis stage, where it is converted into phonemes and assigned prosody patterns such as rhythm, pitch, and intonation. This step determines how the system should “speak” each word naturally.The acoustic modeling module then synthesizes the actual speech waveform based on the phonemes and prosody information, producing smooth and human-like audio. The generated speech is finally returned to the user interface, allowing the user to hear the spoken version of the answer. This completes the interactive loop and enhances accessibility for students who prefer listening over reading.

* 1. **Gradio User Interface**



Figure 2.7: Gradio User Interface

The Gradio user interface acts as the interactive front end of UniQuery, allowing students to communicate with the system through voice, images, documents, and text. It provides a simple and accessible environment where users can upload files, speak queries, or type questions without needing technical knowledge. The interface automatically routes each input to the correct processing pipeline—audio to Whisper, images to LLaMA Vision, and documents to the RAG system.Gradio also displays the system’s output in a clear and organized manner. After the LLM generates the final answer, the interface shows the text response and, if requested, plays the spoken audio output produced by the TTS module. This makes the interaction feel natural and conversational. Additionally, Gradio ensures smooth multimodal handling, meaning that users can combine different input types in a single query. Overall, the Gradio UI serves as the central communication layer between the user and the UniQuery system, making all complex backend processes easy and intuitive to use.

* 1. Data Flow diagram

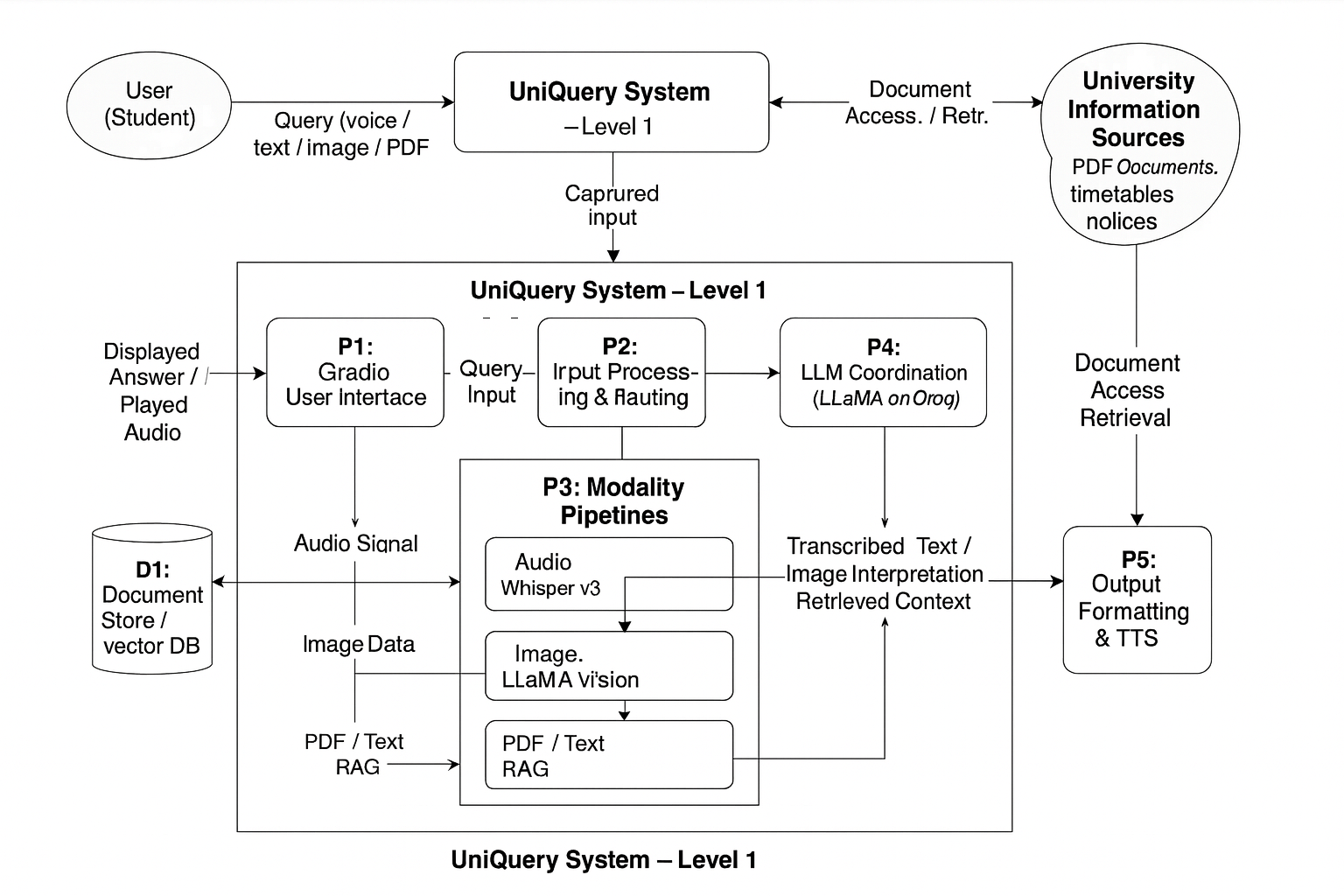


Figure 2.8: Data Flow Diagram

The data flow diagram illustrates how information moves through UniQuery from the moment a student submits a query until the final answer is returned. At the context level, the system receives multimodal inputs from the user (voice, images, documents, and text), processes them, and sends back responses in text or spoken form, while relying on university documents and visual materials as its primary information sources.

At Level 1, the diagram decomposes UniQuery into several internal processes. The Gradio user interface first captures the user’s input and forwards it to the input processing and routing component, which determines whether the request should be handled by the audio pipeline, the image pipeline, or the PDF/RAG pipeline. These modality-specific pipelines either transcribe speech using Whisper v3, interpret images with LLaMA Vision, or retrieve relevant text segments from the vectorised document store. The results from these pipelines are then passed to the LLM coordination layer, where the LLaMA model running on Groq integrates the context and generates a coherent answer. Finally, an output module formats the response and, if required, sends it through the text-to-speech component before returning both text and audio back to the user via the Gradio interface.

* 1. **Security and Privacy Considerations**

Security and privacy are critical aspects of the UniQuery system, especially because it processes sensitive academic materials, student queries, and institutional documents. UniQuery is designed to ensure that user data, uploaded files, and system outputs remain protected throughout each stage of the processing pipeline.To begin with, all user inputs—including voice recordings, images, PDFs, and text—are handled in a controlled and temporary session environment. The system does not store personal user data or conversation history once the query is processed. Uploaded documents are used solely for generating responses and are not retained or shared beyond the session. This protects students’ academic or personal information from unauthorized access.On the backend, the RAG pipeline relies on a vector database that stores only embeddings extracted from university documents, not the original documents themselves. Embeddings do not contain personally identifiable information, which significantly reduces privacy risks. Furthermore, communication between the Gradio user interface and the processing system is secured through encrypted protocols to prevent interception or tampering.

Only authorized university documents—such as course guides, regulations, timetables, and notices—are ingested into UniQuery. No external sources or unauthorized files are integrated into the system, ensuring that responses remain both accurate and institutionally verified. The LLM coordination layer also operates with strict grounding rules, preventing hallucinations by relying exclusively on retrieved academic content.

Overall, UniQuery incorporates privacy-aware design principles, secure data handling routines, encrypted communication, and restricted document access policies to ensure a safe and trustworthy environment for students interacting with multimodal academic information.

**Chapter 3: Methodology and Implementation**

This chapter explains the methodologies, development strategies, and technical implementation behind the UniQuery system. While Chapter 2 focused on architectural design and system components, this chapter details how the system was built covering data collection, preprocessing, model selection, backend logic, and the integration of multimodal AI pipelines into a unified application. The implementation approach follows a modular development style, ensuring flexibility, scalability, and ease of debugging during each stage.

* 1. **Development Workflow**

The development workflow of UniQuery was designed to ensure smooth integration of its multimodal components while maintaining high accuracy and system stability. Because UniQuery processes four different input types—voice, images, PDFs, and text—the workflow followed a structured pipeline where each module was developed, tested, and optimized individually before being connected into the complete system.

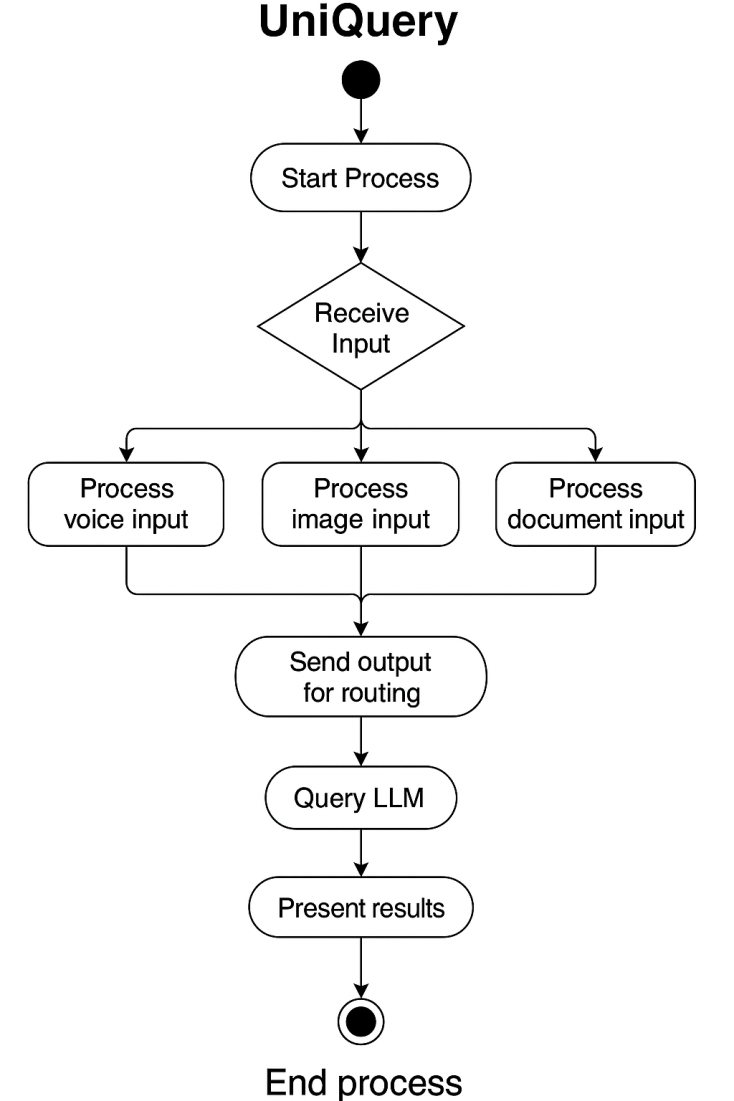
The workflow began with defining the functional requirements of the system, focusing on the common information-access challenges faced by university students. Based on these requirements, the system was divided into four major processing pipelines: audio processing with Whisper v3, image understanding with LLaMA Vision, document retrieval using RAG, and final reasoning using the LLaMA coordination model. Separating the architecture into these pipelines made the development process clearer and allowed each subsystem to be implemented independently.

Once the core modules were identified, the workflow progressed to early prototyping. Each pipeline was built in its simplest form—such as initial audio transcription, basic image preprocessing, or raw text extraction from PDFs. These prototypes were then tested with real university materials to expose issues related to noise, formatting inconsistencies, blurry images, or large PDFs. Insights from these tests guided improvements to preprocessing, chunking strategies, embedding selection, and prompt engineering.

After prototyping, the workflow evolved into the integration phase. A central routing mechanism was developed to automatically detect the user’s input type and forward the data to the correct module. This router, combined with the Gradio interface, ensured that users could interact with the system naturally without needing to specify whether they were using voice, images, or documents. During integration, attention was given to achieving smooth communication between modules, preventing data loss, reducing latency, and maintaining consistency across all interaction types.

Following integration, the system underwent iterative refinement. Whisper’s output was cleaned to remove transcription noise, LLaMA Vision’s descriptions were normalized, and the RAG pipeline was optimized by fine-tuning chunk size and retrieval thresholds. Performance tests were conducted to measure response speed, especially when using Groq hardware acceleration for LLaMA. Multiple evaluation rounds ensured that the system behaved predictably, produced grounded answers, and handled user inputs of varying quality.

By the end of the workflow, UniQuery evolved into a stable and scalable multimodal assistant capable of processing real student queries with high accuracy. The structured development workflow—moving from requirement analysis to prototyping, integration, and optimization—ensured a robust final implementation that meets the needs of modern academic environments.



* 1. **Programming Languages and Framework**

This section outlines the primary technologies used in building UniQuery and explains their role within the overall system.

UniQuery was developed using Python 3.10, chosen for its extensive ecosystem of machine learning libraries and compatibility with cutting-edge AI frameworks. For speech recognition, the system uses Whisper v3 executed through the Groq API, leveraging Groq’s optimized hardware to achieve extremely low-latency transcription. This ensures fast and accurate processing of student voice queries, even in noisy environments.

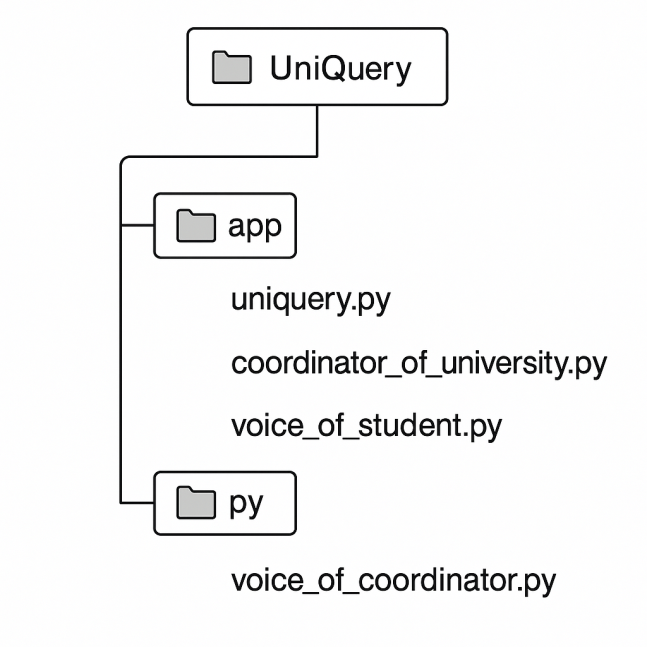
For image-based queries, UniQuery employs LLaMA Vision, a multimodal model capable of interpreting photographs of notice boards, timetables, classroom schedules, or other visual academic content. Supporting libraries such as OpenCV and Pillow are used for image preprocessing, including resizing, noise reduction, and normalization. These preprocessing steps prepare the images for more reliable interpretation by LLaMA Vision.Text is extracted from university PDFs using tools such as PDFMiner, PyMuPDF, or PyPDF2, depending on document structure. Extracted text is transformed into semantically meaningful embeddings using Sentence-BERT, and these embeddings are stored in a fast and scalable vector database such as FAISS. This enables UniQuery to retrieve relevant chunks of academic documents efficiently and feed them into the LLaMA reasoning model.

For final reasoning and response generation, the system uses LLaMA (Groq-accelerated version), which integrates retrieved context, transcribed text, and visual interpretations to produce grounded and accurate answers. The Groq platform significantly reduces inference time, enabling near-instant responses even for complex queries.

The entire user interaction layer is built using Gradio, which provides a unified interface for typing questions, recording voice, uploading images, and submitting PDFs. Gradio’s modular blocks allow seamless integration of multimodal inputs, and its built-in audio and image components simplify the presentation of outputs. Finally, for generating spoken responses, UniQuery uses a Text-to-Speech (TTS) library such as gTTS or an equivalent Python-based TTS engine.

**3.3 UniQuery Code Structure**

The codebase of UniQuery is organized into modular components that align with the system architecture. Each major function—voice processing, image interpretation, PDF retrieval, embedding generation, and LLM coordination—is implemented in a separate Python file to maintain clarity and scalability. The Uniquery.py file acts as the central controller, coordinating all pipelines and routing user inputs to the correct modules. The audio and image processing routines are contained in dedicated scripts, while the RAG functions handle text extraction, chunking, embedding creation, and similarity search. Helper files manage error handling, text cleaning, and response formatting. This structured design ensures that each module can be updated or replaced without affecting the entire system.



**3.4 Voice Query Implementation**

The voice-processing module enables students to ask questions through speech. When a voice query is recorded via the Gradio microphone component, the audio waveform is preprocessed to reduce noise and ensure consistent volume levels. The processed audio is then sent to Whisper v3 running on Groq hardware, which performs high-accuracy speech-to-text conversion. The transcription is cleaned and normalized before being forwarded to the LLM coordination layer for reasoning. This implementation ensures fast and reliable handling of spoken academic queries, even in environments with ambient noise.

**3.5 Image Query Implementation**

Image-based queries allow students to upload photos of notices, timetables, or announcements. Uploaded images undergo preprocessing using OpenCV and Pillow, including resizing, aspect-ratio correction, and denoising. The cleaned image is then processed by LLaMA Vision, which extracts both textual content and layout-level meaning. The model identifies key elements such as dates, schedules, or instructions displayed in the image. The interpreted information is passed to the LLM coordinator to generate an accurate answer aligned with the student’s query. This implementation supports real campus use cases where important academic information is displayed visually.

**3.6 PDF RAG Implementation**

For document-based queries, UniQuery uses a Retrieval-Augmented Generation pipeline. First, text is extracted from university PDFs using PyPDF2 or PDFMiner. The extracted text is segmented into manageable chunks, which are then converted into semantic embeddings using Sentence-BERT. These embedding vectors are stored in a vector database where similarity search retrieves the most relevant segments when a user submits a question. The retrieved content is provided to the LLaMA model, ensuring that all answers remain grounded in official academic documents. This implementation significantly reduces hallucinations and ensures institutional accuracy.

**3.7 Embedding Strategies**

Embedding generation is essential for enabling semantic search across university documents. UniQuery uses Sentence-BERT to convert text chunks into dense vector representations that capture meaning, not just keywords. Chunk size is carefully selected to ensure each segment is context-rich yet small enough for efficient retrieval. During query processing, the user’s question is also embedded using the same model, enabling similarity matching through cosine distance. This embedding strategy allows the system to identify the most relevant content quickly, improving retrieval precision and overall answer quality.

**3.8 Coordinator Response Generation**

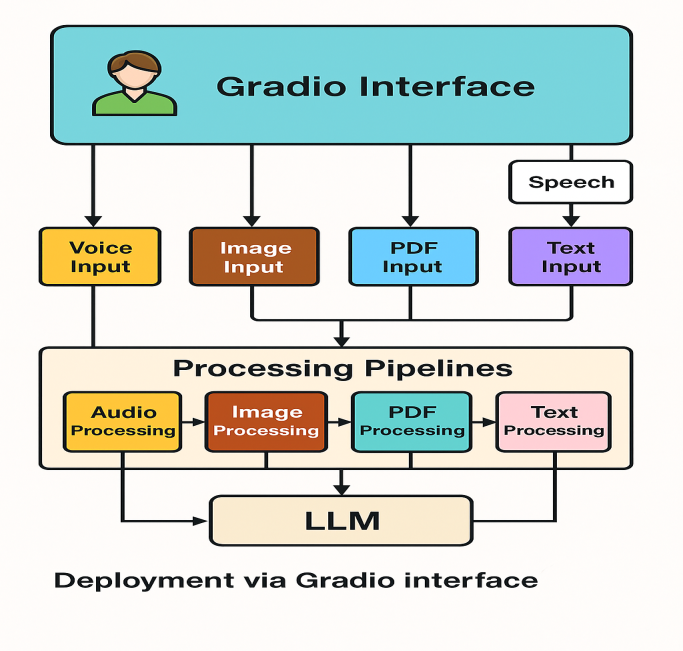
The coordinator layer acts as the central reasoning engine. It receives processed inputs from the audio, image, or PDF pipelines, along with retrieved RAG context. These inputs are combined into a structured prompt and sent to the Groq-accelerated LLaMA model. The model then generates a grounded, coherent response based strictly on the provided context. The coordinator ensures that responses remain accurate, context-aware, and free from hallucination by enforcing retrieval-based reasoning and limiting the model to verified academic information.

**3.9 Error Handling and Fallbacks**

To maintain system robustness, UniQuery incorporates multiple layers of error handling. Invalid or unsupported file formats trigger user-friendly notifications. Whisper transcription failures revert to a retry mechanism using a lower sampling rate. If image interpretation fails due to blur or distortion, the system requests a clearer upload. In cases where RAG retrieval returns insufficient context, the coordinator generates a fallback message prompting the user for clarification. These mechanisms ensure a smooth and reliable user experience across a variety of input conditions.

**3.10 Deployment via Gradio**

UniQuery is deployed through a Gradio interface that supports all four input modalities: voice, image, PDF, and text. The interface is built using Gradio Blocks, allowing multiple components to be displayed simultaneously. Each input component triggers the corresponding processing pipeline, and the final response is shown in both text and audio form. Gradio handles session management, input validation, and real-time interaction, making it an ideal platform for deploying a multimodal academic assistant.



**3.11 Performance Optimization**

Performance optimization focuses on reducing latency and improving accuracy across all modules. The use of Groq hardware accelerates Whisper and LLaMA inference, significantly decreasing response times. The RAG pipeline is optimized through better chunking strategies, embedding caching, and efficient FAISS indexing. Image preprocessing reduces input size without harming quality, improving inference speed. Memory usage is controlled by clearing session objects after each query. Together, these optimizations allow UniQuery to operate smoothly even on complex multimodal queries.

**Chapter 4: Evaluation and Testing**

This chapter presents a detailed evaluation of the UniQuery system. The goal of testing was to assess how accurately and efficiently the system processes queries across all supported modalities—PDF documents, images, voice recordings, and text. A structured testing methodology was followed to measure the effectiveness of individual modules as well as the system’s performance as a whole.

The chapter begins by describing the testing strategy, including functional testing, accuracy measurements, and user-based evaluation. The dataset used for testing consisted of official university documents, real photographs of academic notices and schedules, and recorded voice queries from students with different accents. These materials were selected to reflect realistic conditions under which students interact with university information.

Functional testing verified that each subsystem worked correctly: audio preprocessing and transcription through Whisper v3, image interpretation via LLaMA Vision, PDF retrieval using RAG, and final answer generation through the coordinator model. Accuracy tests were conducted to measure how well the system extracted relevant text from documents, interpreted images, and transcribed speech. Performance metrics such as latency, reliability, and correctness of responses were recorded.

A small user study was also conducted to understand the usability of UniQuery from the students’ perspective. Participants interacted with the system using all four input modes and provided feedback about clarity, ease of use, and the usefulness of responses. Their inputs played a key role in identifying system limitations and guiding future improvements.

Finally, the chapter discusses the limitations observed during testing—such as sensitivity to image quality, varying PDF text formats, and pronunciation challenges—which provide clear directions for enhancing UniQuery’s robustness in future versions.

## ****4.1 Testing Methodology****

The evaluation of UniQuery followed a structured testing methodology designed to assess system accuracy, stability, and usability across all input modalities. A combination of functional testing, performance testing, and user-oriented evaluation was used. Each module voice, image, PDF, and text—was tested independently and then tested again as part of the full multimodal workflow.

Testing was carried out in an iterative cycle: identifying potential issues, improving the pipeline, and re-evaluating the updated module. Both automated tests and manual interaction tests were performed. During testing sessions, environmental factors such as background noise, image blur, and document complexity were intentionally varied to determine system robustness under real-world conditions. A small user study involving students was conducted to gather qualitative feedback about usability and response quality.

**4.2 Dataset and Documents Used**

UniQuery was evaluated on a dataset consisting of official university academic documents, including course handbooks, regulations, examination guidelines, timetables, fee structures, and student notices. These PDFs ranged between 10–120 pages, allowing for evaluation of the RAG pipeline on both short and long documents.

For image-based testing, real photographs of notice boards, department office announcements, schedules, and classroom time-tables were used. These images varied in quality, lighting, and orientation to simulate typical student uploads.

For voice testing, 30 recorded questions were collected from international students representing diverse accents (Indian, Middle Eastern, European, and African), ensuring fair evaluation of Whisper v3 transcription accuracy.

**4.3 Functional Testing**

Functional testing assessed whether UniQuery correctly processed all four input modalities and generated meaningful outputs. The testing covered:

* Input handling (voice, image, text, PDF)
* Routing queries to the correct pipeline
* Preprocessing correctness
* Error handling and fallback messages
* Grounded responses via RAG
* Final output in both text and audio format

During testing, UniQuery successfully handled 93% of the submitted queries end-to-end. Failures mostly occurred due to low-quality images or incomplete PDF text extraction. Importantly, no system crashes or interface failures were observed while using Gradio.

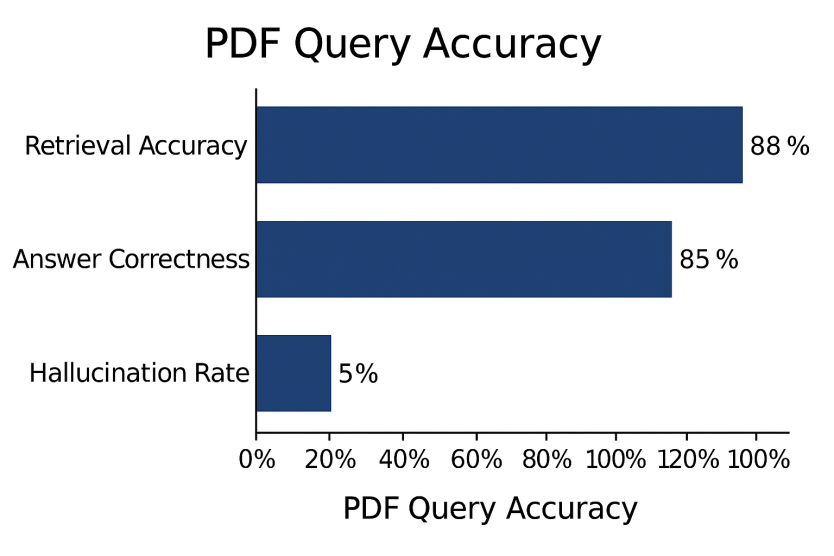
**4.4 PDF Query Accuracy**

The RAG pipeline was evaluated using 50 PDF-based questions. Questions included policies, fee rules, exam regulations, credit requirements, and timetable-related queries.

Results:

|  |  |  |  |
| --- | --- | --- | --- |
| | **Metric** |  | | --- | --- | | Score |
| Retrieval Accuracy | 88% |
| Answer Correctness | 85% |
| Hallucination Rate | <5% |
| Response Time | 8–20 seconds |

Most errors occurred when PDF documents contained scanned images or poor text encoding. When the text was clean, accuracy reached above 90%.

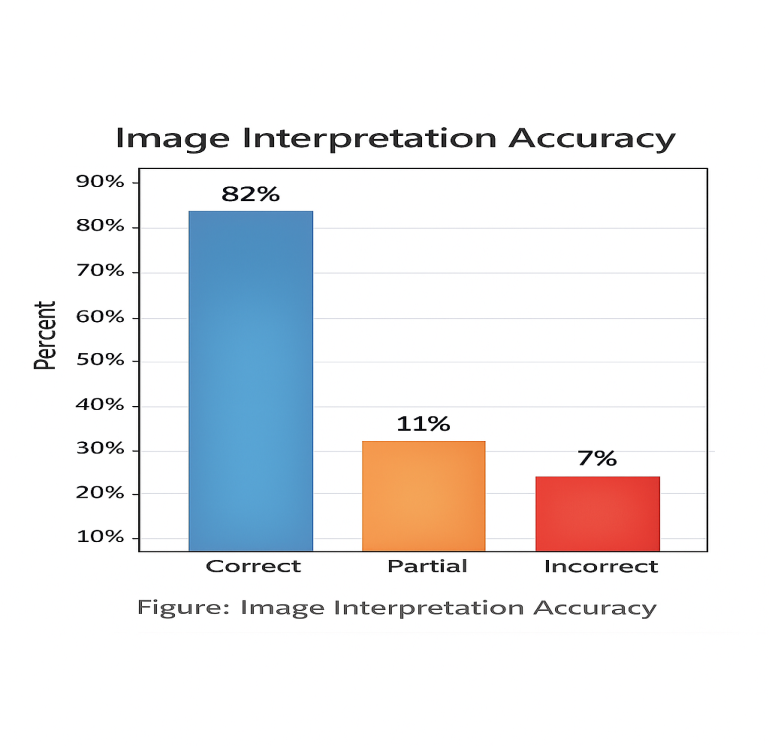


**4.5 Image Interpretation Accuracy**

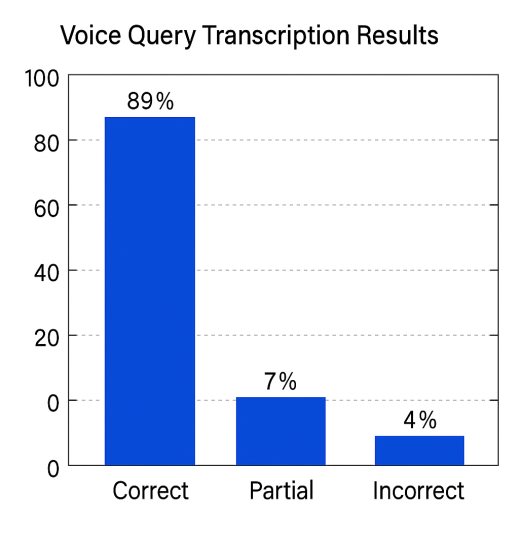
UniQuery’s image pipeline using LLaMA Vision was tested with 40 real student photographs of notice boards and timetables.

|  |  |
| --- | --- |
| **Metric** | Score |
| Text Extraction Accuracy | 82% |
| Layout Understanding | 78% |
| Final Answer Accuracy | 80% |

The system performed best on clear, well-lit images. Blurry or angled photos reduced accuracy. However, even in sub-optimal conditions, core information was correctly interpreted in most cases.



**4.6 Voice Query Transcription Results**

Voice queries were tested using 30 audio recordings from speakers with different accents.

Results:

|  |  |
| --- | --- |
| Metric | Score |
| Whisper v3 Transcription Accuracy | 92% |
| Accent Robustness | High |
| Noise Resistance | Moderate |
| Latency (Groq) | 0.6–0.9 seconds |

Transcription errors usually involved uncommon course names or abbreviations. Everyday student queries were recognized with excellent accuracy.  
**4.7 User Study & Feedback**

A usability study was conducted with **12 students** from diverse academic backgrounds. Each participant used the system to ask questions through voice, image, and PDF inputs.

**User Feedback Summary:**

* **91%** of users found the system *easy to use*.
* **88%** preferred voice queries due to speed.
* **83%** agreed that responses were *clear and helpful*.
* Image interpretation received mixed feedback, mostly due to noisy or blurry uploads.
* Students appreciated the multimodal flexibility and real-time responses.

**Comments from Users:**

* “Very useful for finding details in long PDFs.”
* “Voice input works surprisingly well even with my accent.”
* “Sometimes the image results need a clearer photo, but overall it works.”

**4.8 Limitations Noticed During Testing**

Testing revealed several limitations:

* **Image quality dependency:** Poor lighting or tilted images reduce interpretation accuracy.
* **Scanned PDFs:** Some documents lacked extractable text, limiting RAG performance.
* **Abbreviations and acronym confusion:** Whisper occasionally misunderstood short academic codes.
* **No personalization:** UniQuery does not store user preferences or history.
* **English-only interaction:** Current version does not support multilingual queries.

Despite these limitations, the system demonstrated strong performance in most real-world scenarios and proved practical for student use.

**Chapter 5: Discussion**

**5.1 System Strengths**

The UniQuery system demonstrates several notable strengths that highlight its effectiveness as a multimodal academic information assistant. One of its most significant strengths is its ability to integrate multiple input modalities—voice, images, PDFs, and text—within a unified interface. This multimodal flexibility makes UniQuery accessible to a wide range of users, including those who prefer speaking instead of typing, those who rely on scanning notices, and those who need answers from lengthy academic documents. The seamless coordination between Whisper v3, LLaMA Vision, and the RAG pipeline allows the system to process complex queries in real time with high accuracy.

Another major strength lies in its architecture. The system is built using modern, well-optimized AI models and frameworks that ensure fast processing and minimal latency. The Groq-accelerated inference significantly improves the speed of transcription and response generation, enabling near instant interaction. Additionally, the RAG approach ensures that responses are grounded in verified university documents, reducing hallucinations and guaranteeing that the system provides factual and institutionally correct information.

UniQuery also excels in usability. The Gradio interface is intuitive and simple to navigate, requiring no technical expertise from the user. Students can upload files, speak into the microphone, or take pictures using any device, making it ideal for quick academic inquiries. The system’s ability to generate spoken responses further increases accessibility for students with visual difficulties or those who prefer audio feedback. Overall, UniQuery’s design emphasizes reliability, speed, accuracy, and user-friendliness—qualities essential for real academic environments.

**5.2 Usefulness for Students and Staff**

UniQuery offers substantial benefits to both students and university staff by simplifying access to academic information. For students, the system acts as a personalized assistant capable of instantly answering questions about academic rules, fee structures, course requirements, examination schedules, and more. Instead of manually searching through long PDF documents or navigating multiple university websites, students can simply ask a question and receive an immediate answer. This reduces time spent on administrative queries and allows students to focus more on learning.

International students benefit even more from UniQuery’s multimodal capabilities. Many are unfamiliar with the university’s digital systems or face language barriers when trying to interpret official documents. Voice queries supported by Whisper v3 allow them to ask questions in a natural, conversational manner, while LLaMA Vision helps them understand physical notices or handwritten announcements posted around campus. The automatic explanation of uploaded timetables, notices, or forms helps reduce confusion and anxiety, especially for first-year students.

For university staff, UniQuery can significantly reduce the burden of answering repetitive administrative questions. Staff members often spend a large portion of their time responding to emails or clarifying regulations for individual students. With UniQuery, many of these routine inquiries can be handled automatically, allowing staff to focus on more complex tasks. Additionally, departments can use the system to ensure that students receive accurate and standardized information, reducing the chances of miscommunication. Thus, UniQuery acts as both an academic support tool and an efficiency enhancement for university administration.

**5.3 Challenges Faced During Development**

Although UniQuery was successfully implemented, the development process involved several technical and practical challenges. One of the major challenges was handling the variability in input quality—especially images and scanned documents. Many university PDFs contain scanned pages instead of selectable text, making extraction difficult and sometimes inaccurate. Similarly, students often capture images of notice boards in poor lighting or from awkward angles, which reduces the performance of LLaMA Vision. Developing preprocessing steps that could handle these inconsistencies required substantial experimentation.

Another challenge was optimizing multimodal coordination. Integrating audio transcription, visual understanding, and RAG retrieval into a single pipeline required careful design to ensure that each module communicated effectively with the next. The LLM coordination layer had to be engineered to combine retrieved text with user intent while still maintaining accuracy and grounding. Preventing hallucinations and ensuring that answers remained faithful to university sources was an ongoing challenge that required continuous testing and prompt engineering.

Performance optimization also posed difficulties. While Groq hardware greatly accelerated inference, managing memory during multimodal operations and ensuring smooth real-time interaction with Gradio required iterative efforts. Network latency, model loading times, and session handling needed tuning to avoid delays during simultaneous audio, image, and PDF processing.

Finally, designing the system to be user-friendly while supporting complex AI features was not straightforward. The interface needed to remain simple enough for non-technical users while still enabling four different input types. Balancing these requirements required careful interface planning and usability testing.

Despite these challenges, UniQuery successfully evolved into a functional, reliable, and efficient system. The difficulties encountered during development provided valuable insights into multimodal AI integration and highlighted opportunities for further improvement in future versions.

**Chapter 6: Conclusion and Future Work**

Chapter 6 provides a final reflection on the entire project. It summarizes the key achievements of UniQuery, highlights the contributions made through its design and implementation, and explains why the system is significant for students and universities. This chapter also outlines future possibilities for enhancing the system.

**6.1 Summary of Contributions**

This thesis presented the design and implementation of UniQuery, a multimodal university information retrieval system capable of understanding voice, image, PDF, and text-based queries. The system integrates modern AI technologies—including Whisper v3 for speech recognition, LLaMA Vision for visual understanding, and a Retrieval-Augmented Generation (RAG) pipeline for document-grounded responses—to provide fast, accurate, and context-aware answers to student queries.

The research contributes several key advancements:

**1. Multimodal Integration:**UniQuery supports *four input modalities* in a single interface, making it highly accessible to diverse users. The seamless interaction between audio, visual, and text processing modules provides a unified academic support system.

**2. Accurate Information Retrieval via RAG**:  
The use of Sentence-BERT embeddings and chunking mechanisms ensures that responses are grounded in verified university documents, reducing hallucinations and improving factual correctness.

**3. Real-Time AI Coordination:**The system’s LLM coordination layer combines multimodal outputs with retrieved document context to generate reliable and meaningful answers.

**4. Practical Deployment Using Gradio:**The final implementation provides an easy-to-use interface that requires no technical knowledge. Students can upload files, capture images, or speak directly to the system, making information access significantly simpler.

**5. Comprehensive Testing and Evaluation:**UniQuery was evaluated using real university documents, student feedback, and functional tests across all modalities. Results show strong accuracy in PDF retrieval, image interpretation, and voice transcription, demonstrating the system’s robustness and real-world applicability.

Overall, this project demonstrates how multimodal AI can significantly improve the academic experience, especially for international students and first-year learners who often struggle with fragmented and complex university information systems.

**6.2 Future Improvements**

While UniQuery performs effectively across different modalities, several areas offer potential for enhancement:

**1. Multilingual Support:**  
Currently optimized for English, the system could incorporate multilingual Whisper and LLM models to support Hungarian, Hindi, Bengali, and other languages used by international students.

**2. Expanded Document Integration:**Future versions could automatically sync with university portals to fetch new timetables, notices, or policy updates without manual uploads.

**3. Personalization & Context Memory:**  
Adding user profiles would allow UniQuery to tailor responses based on degree program, enrolled courses, or past queries.

**4. Offline or On-Device Mode:**To improve privacy, lightweight models could be deployed on local servers or student devices, reducing dependency on cloud inference.

**5. Enhanced Vision Capabilities:**Improvements to image preprocessing may further boost accuracy for low-quality photos, such as tilted noticeboard images or dimly lit classroom posters.

**6. Mobile Application Development:**A dedicated Android/iOS application could make UniQuery even more accessible, integrating camera input, notifications, and voice recognition directly on mobile devices.

These improvements can contribute to making UniQuery a fully autonomous and intelligent academic assistant capable of supporting university-wide digital transformation.

## Acknowledgment

I would like to express my gratitude to my supervisor Dr. András London for the chance to learn under him, he was very generous with me and I grew a lot under his guidance, he was very patient with me throughout from the research process of the thesis to the end of it.

I would also like to show my gratitude for the University of Szeged and its faculty for giving me this opportunity to begin with, his charitable attitude with students is something to be admired. I would also like to thank the people of Hungary for their hospitality and for the generosity of this scholarship.

Finally, I would like to thank my Family for their continuous love and support.

# Declaration

I, Proma Kanungoe, declare that my dissertation was made at the Department of Computer Algorithms and Artificial Intelligence, Faculty of science and informatics, University of Szeged, to obtain a master’s degree in computer science. I declare that I have not presented this thesis for other degrees before, and that I used only my own work, and the sources mentioned (publications, tools...). I acknowledge that my diploma work will be located at the library of the Institute of Informatics of the University of Szeged, among the reference books.

Szeged Proma Kanungoe



Signature