System Requirement Specification

Table of Contents

Contents

Table of Contents

Problem Statement

Overview
- Background
- Overall Description

Investigation & Analysis Methodology
- System Investigation
- Analysis Methodology
  - Feasibility Study and Requirements Elicitation
  - System Analysis and Requirement Specification
  - Object-oriented design using UML
  - Prototyping

Constraints
- Scalability
- API Dependency
- Project Schedule

Operational Requirements
- Help Desk Support
- Application Services & Technical Support
- Administration Features

Functional Requirements
- Automated Note-Taking Process
- Database Utilization for Transcript Management
- Summary Generation with Generative AI
- Interactive Chatbots for Enhanced Learning

Input Requirements
- Lecture Recordings Submission
- Search Parameters
- Feedback Mechanism
- Summary Customisation Options
- Optional Presentation Slides Upload
- Chatbot Integration Context

Process Requirements
- SQL Transaction
- Consistency Priority (CAP Theorem)
- Data Security
- Data validation

Output Requirements
- Generated Transcript
- Customisable Summaries
- Chatbot Interactions

Hardware Requirements
- Network Infrastructure
- Client Devices
- Servers
- Peripheral Devices

Software Requirements
- Operating Systems
- Web Browsers
- Speech-to-Text and AI Services
- Database Management
- Development Tools and Libraries
- Security and Compliance Software
- Collaboration and Version Control
- Licenses

Deployment Requirements

Problem Statement

Nanyang Technological University (NTU) is facing significant challenges with its current approach to note-taking for academic lectures. The university employs student note-takers at a cost of $18 per hour to document lectures across a wide array of subjects. With hundreds of modules being conducted on a weekly basis as part of approximately 140 courses, the financial burden on the university is substantial, amounting to tens of thousands of dollars each month. This traditional method of note-taking not only incurs a high financial cost but also lacks reliability and efficiency for several reasons:

1. **Accuracy and Completeness**: There is no guarantee that the notes taken are accurate or complete. The quality of the notes can vary significantly depending on the note-taker's understanding of the lecture material, attentiveness, and note-taking skills. This variability can lead to inconsistent quality of notes, potentially affecting students' learning outcomes.

2. **Contextual Understanding**: The student note-takers, often undergoing the same course, may not possess the necessary depth of knowledge or context to capture the nuances and key points of the lectures accurately. This lack of expertise can result in notes that are superficial or miss critical information, making them less useful for study and revision purposes.

3. **Scalability and Efficiency**: The manual process of note-taking is not scalable to the volume of lectures conducted at NTU. It is labor-intensive and does not leverage technology to streamline the process, leading to delays in note availability and increased workload on student note-takers.

Overview

Background

Nanyang Technological University (NTU) is currently facing challenges with its traditional approach to lecture note-taking. The existing method relies on student note-takers and is characterized by significant financial costs, inefficiencies, and inconsistencies in note quality. These issues stem from variations in note-takers' accuracy, understanding, and completeness of the material, which in turn affects the reliability and scalability of the note-taking process. The university's reliance on this manual and labor-intensive method has resulted in substantial monthly expenses and has proven to be unsustainable given the volume of lectures and the wide array of subjects taught at NTU.

Overall Description

Magnificent 7 proposes a comprehensive and innovative solution to overhaul its note-taking system by incorporating advanced technological tools. The strategy includes the deployment of state-of-the-art speech-to-text and AI technologies aimed at automating the note-taking process, managing lecture transcripts efficiently, generating AI-powered summaries, and enhancing student learning through interactive chatbots.

The proposed system will introduce automated note-taking using advanced speech recognition technology to convert lecture audio into precise transcripts, thus enhancing accuracy and reducing manual labor. A sophisticated database will be developed for storing, retrieving, and managing these transcripts, complete with an advanced search function and a user feedback mechanism. Additionally, generative AI will be employed to create tailored summaries of lecture transcripts, and AI-powered chatbots will be implemented to provide a personalized tutoring experience.

Investigation & Analysis Methodology

System Investigation

The Automated Note-Taking System is designed to revolutionize the way lecture content is captured and utilized, offering a seamless, efficient, and interactive educational experience. At its core, the system employs Whisper API technologies to convert lecture audio into precise

transcripts, accommodating both live recordings and user-uploaded files in various audio and video formats. This process ensures the rapid availability of transcripts, which are editable for accuracy enhancement. To manage these transcripts, a sophisticated PostgreSQL database system is established, enabling users to easily store, search, and retrieve content based on specific academic parameters. The system also integrates a feedback mechanism to maintain transcript quality, relying on a community-driven upvote/downvote system. Further enriching the educational toolset, generative AI technology of ChatGPT API is harnessed to craft tailored summaries of the lecture transcripts, offering customizable difficulty levels and lengths to suit diverse learning needs. The inclusion of an option to upload lecture slides enriches the summaries with additional context, making them more comprehensive. The system's interactive dimension is then embodied in AI-powered chatbots, designed to provide a personalized tutoring experience. These chatbots can engage with users conversationally, drawing on the vast repository of transcripts, summaries, and slides to offer accurate, context-aware responses to inquiries, thereby enhancing understanding and facilitating deeper learning.

Analysis Methodology

Feasibility Study and Requirements Elicitation

The initiative will be spearheaded by a dedicated team, adept in the nuances of existing note-taking methodologies, ensuring a collaborative and informed approach through regular strategic meetings. To gain a comprehensive understanding of the current landscape and to sculpt the future system's blueprint, a series of targeted interviews will be conducted with both the architects and the stewards of the NTU note-taking ecosystem. These dialogues will extend to the operational staff, whose day-to-day interactions with the system offer invaluable insights into its strengths and areas for enhancement.

System Analysis and Requirement Specification

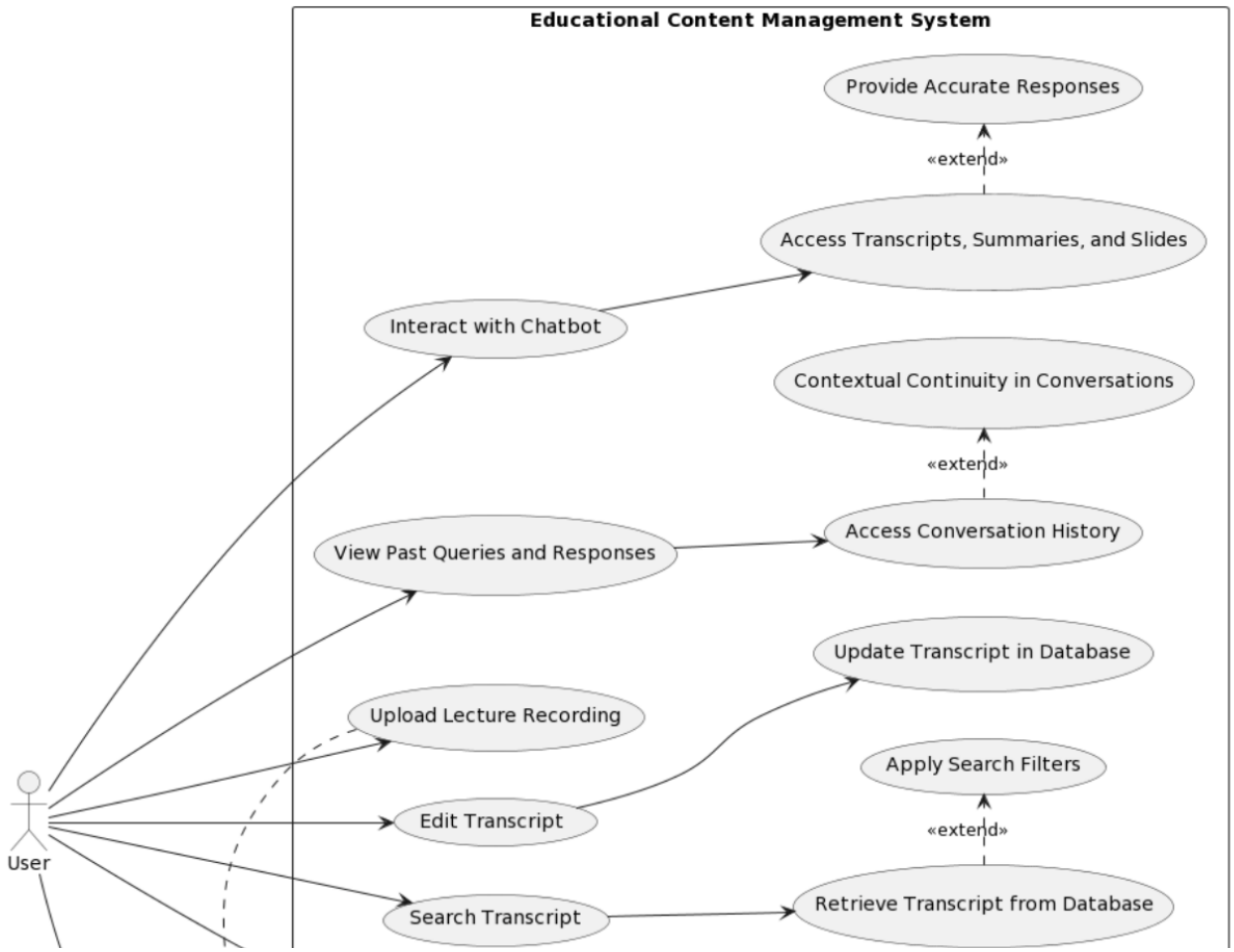
The system incorporates a variety of functionalities designed to streamline and enhance the learning experience through advanced automation and artificial intelligence technologies.

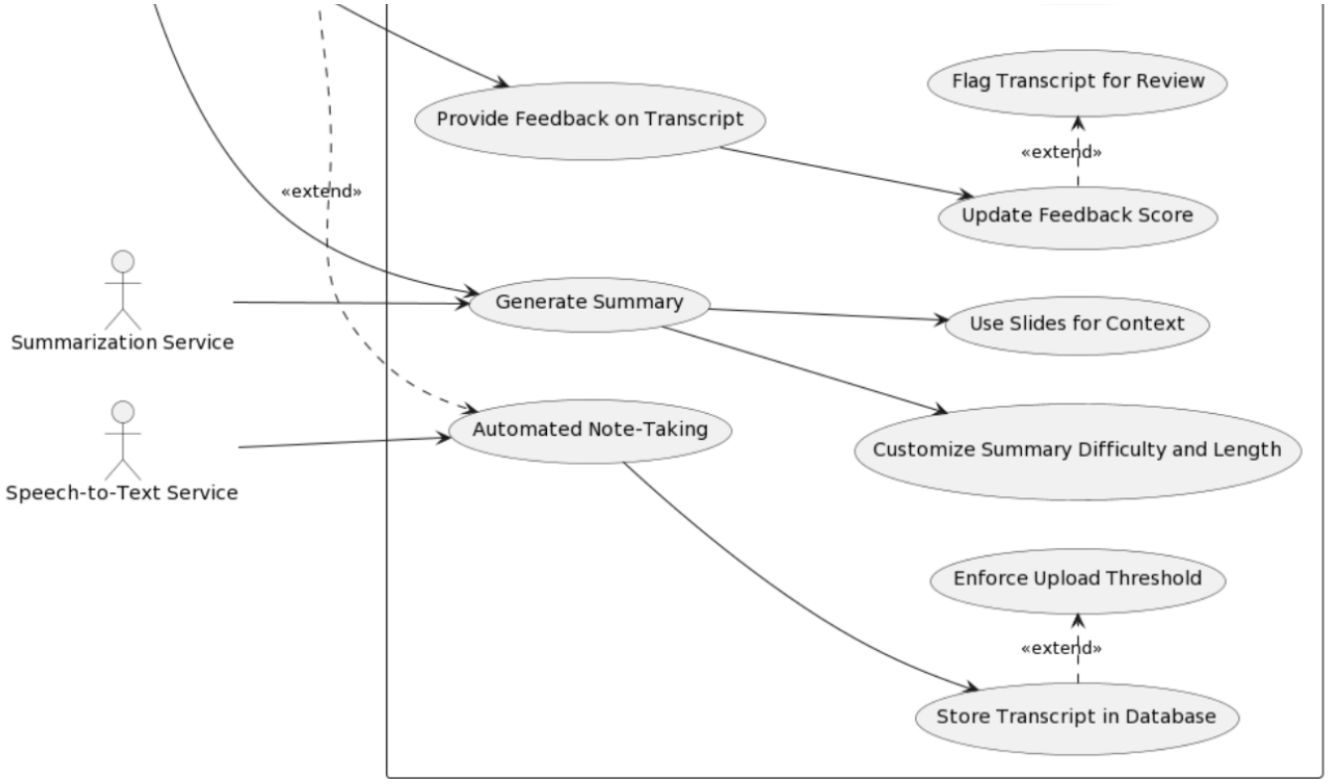
- **Upload Lecture Recording**: This feature allows users to upload audio or video recordings of lectures, which are then processed by a Speech-to-Text Service to create accurate transcripts. This ensures comprehensive and accessible documentation of lecture content for future reference.
- **Edit Transcript**: Users are empowered to edit or correct the automatically generated transcripts, ensuring the content's accuracy and reliability. These modifications are updated in the system's database, maintaining the integrity of the educational content.
- **Search Transcript**: This functionality enables users to efficiently locate specific transcripts using a variety of search parameters. It facilitates easy access to relevant lecture content, enhancing the user experience and supporting effective study practices.
- **Provide Feedback on Transcript**: Users can contribute to content quality control by rating the transcripts through an upvote/downvote system. This feedback mechanism helps in maintaining high standards of educational material by identifying areas for review or improvement.

- **Generate Summary**: Leveraging an AI-powered Summarization Service, this feature produces concise summaries of lecture transcripts tailored to user preferences for difficulty level and length. It provides customizable content that aids in efficient studying and revision.
- **Interact with Chatbot**: This interactive feature allows users to engage with an AI-powered chatbot for clarifications, further explanations, or additional information based on the lecture content. It simulates a personalized tutoring experience, enhancing understanding and engagement with the material.
- **View Past Queries and Responses**: Users can review their past interactions with the chatbot, including queries posed and responses received. This functionality ensures continuity in learning and allows users to revisit and reflect on previously discussed topics and clarifications.

Together, these features create a comprehensive system that not only facilitates the automated transcription of lectures but also enriches the educational experience with customizable content, interactive learning tools, and quality assurance mechanisms, catering to a wide range of learning preferences and needs.

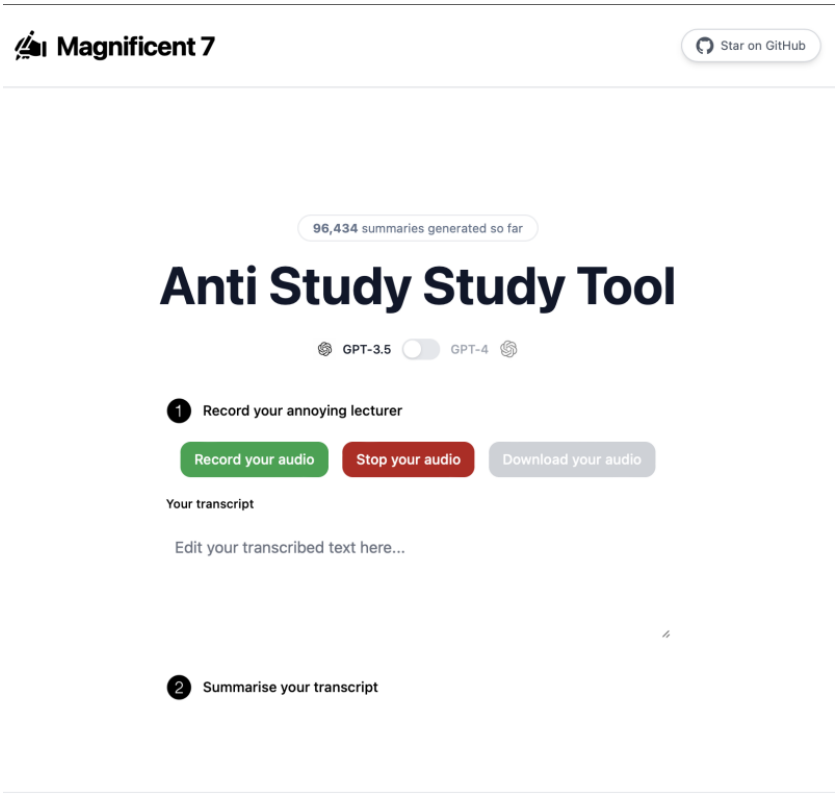
Object-oriented design using UML





In the proposed system, users can upload lecture recordings, which are then transcribed by a Speech-to-Text Service, ensuring detailed and accessible lecture documentation. The system also offers the ability to edit these transcripts, enhancing accuracy and reliability. A search function allows for quick retrieval of specific content, improving study efficiency. Users contribute to quality assurance through a feedback mechanism, rating transcripts to signal the need for review or adjustment. An AI-powered service generates tailored summaries, facilitating customized study aids. Additionally, an interactive chatbot provides personalized assistance, deepening understanding of lecture material. Users can also revisit their interactions with the chatbot, reinforcing learning and continuity.

Prototyping



The Object Oriented Rapid Prototyping (OORP) method will be used to implement a limited and functional prototype for the note-taking system. The prototype will be a working example of part of the system for demonstration and proof of concept purposes only. It will include web-based forms as an end-user interface with the PostgresSQL database. The prototype will be presented to the implementation team.

Constraints

Scalability

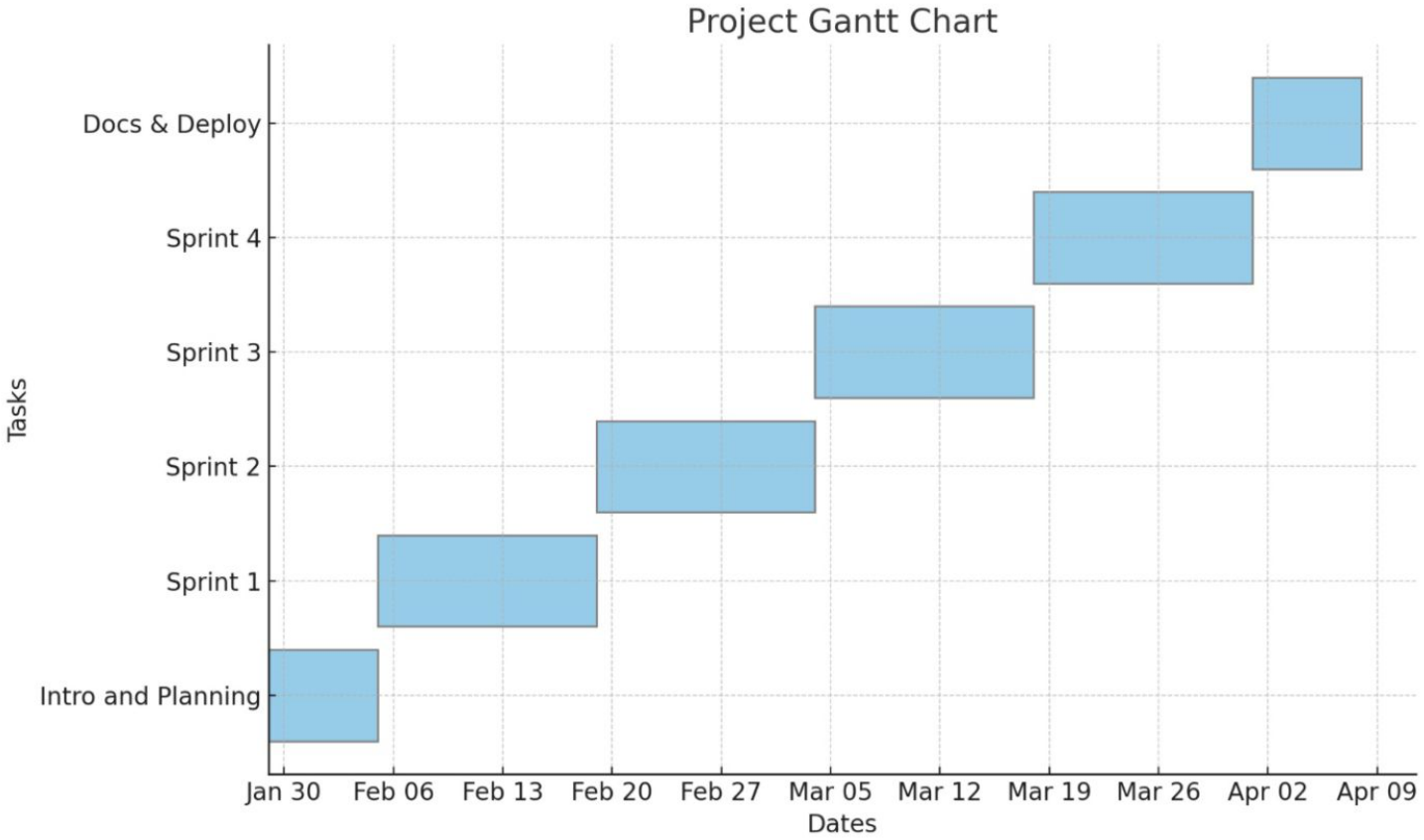
Scaling the system is hindered by the need for substantial manual input to configure each course, including semester year and week number, and the continuous database administration required to manage transcripts flagged by users, often due to excessive downvotes.

API Dependency

Separately, the system's dependency on API requests, particularly to services like Whisper for speech-to-text processing and ChatGPT for interactive functionalities, introduces challenges related to potential lag and increased operational costs, complicating scalability further.

Project Schedule

The project will adopt an Agile project management approach, allowing for iterative development, frequent reassessment, and adaptation of plans. This approach will enable the team to respond to changes in project requirements and priorities effectively. The project will be divided into two-week sprints, with each sprint focusing on a specific set of features and tasks.



Gantt Chart illustrating the project timeline and milestones.

Operational Requirements

Help Desk Support

To ensure the reliability and appropriateness of the content, the system includes a dedicated Help Desk Support feature. Users can reach out directly to report any inappropriate transcripts or notify about service disruptions, such as a down server. This support mechanism is crucial for maintaining the integrity of the note-taking system and ensuring that it remains a safe and reliable resource for all users.

Application Services & Technical Support

Recognizing the importance of continuous improvement and bug resolution, the system's framework includes comprehensive Application Services & Technical Support. During the production phase, access to the source code is provided to key personnel, enabling them to identify

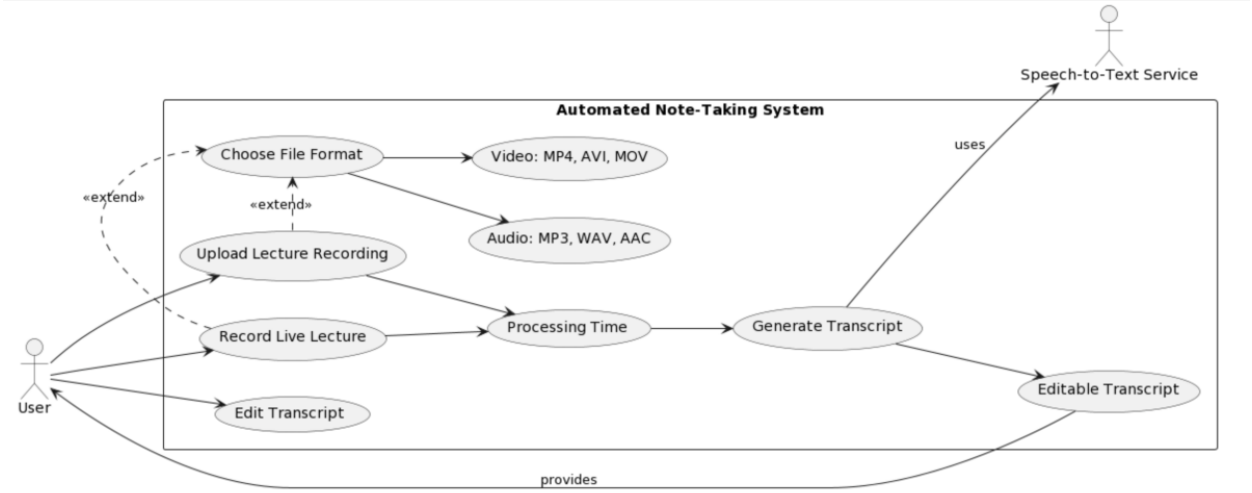
and rectify any bugs swiftly. This proactive approach to technical support ensures the system's robustness and reliability, enhancing user experience and system performance.

Administration Features

To uphold the quality of the content and address any concerns raised by the user community, the system is equipped with robust Administration Features. Database administrators are specifically tasked with reviewing transcripts that have been flagged by users. This might include content flagged for inaccuracies, inappropriateness, or any other issues that contravene the platform's guidelines. Through this feature, the platform ensures that the content remains accurate, appropriate, and of high quality, reflecting the system's commitment to excellence and user satisfaction.

Functional Requirements

Automated Note-Taking Process

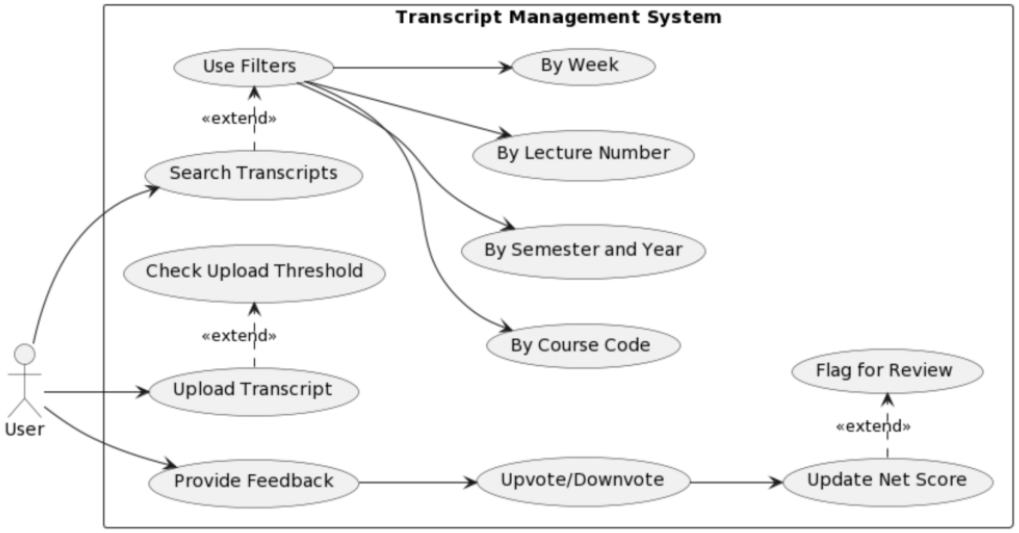


Implement advanced speech-to-text technologies to automate the conversion of lecture audio into accurate transcripts. This automation will be facilitated through two primary methods: direct recording of live lectures and the option for users to upload existing lecture recordings. By leveraging cutting-edge speech recognition and processing capabilities, this objective aims to eliminate the need for manual note-taking and reduce associated costs, while ensuring comprehensive and accurate documentation of lecture content.

- **Facilitation Option**: Provide users with the versatility to submit content in either audio or video formats.
- **Wide Range of Compatible Audio Formats**: Transcripts should be able to be generated from three of the most popular audio formats: MP3, WAV, AAC
- **Wide Range of Compatible Video Formats**: Transcripts should be able to be generated from three of the most popular video formats: MP4, AVI, MOV
- **Processing Time**: Ensure that the time taken from the end of a lecture recording/upload to the availability of the transcript does not exceed 10 minutes for a 1-hour lecture.

- **Editable Transcript**: Enable users to interactively edit transcripts, ensuring they can accurately correct or refine key terms and content, enhancing the overall quality and reliability of the generated text.

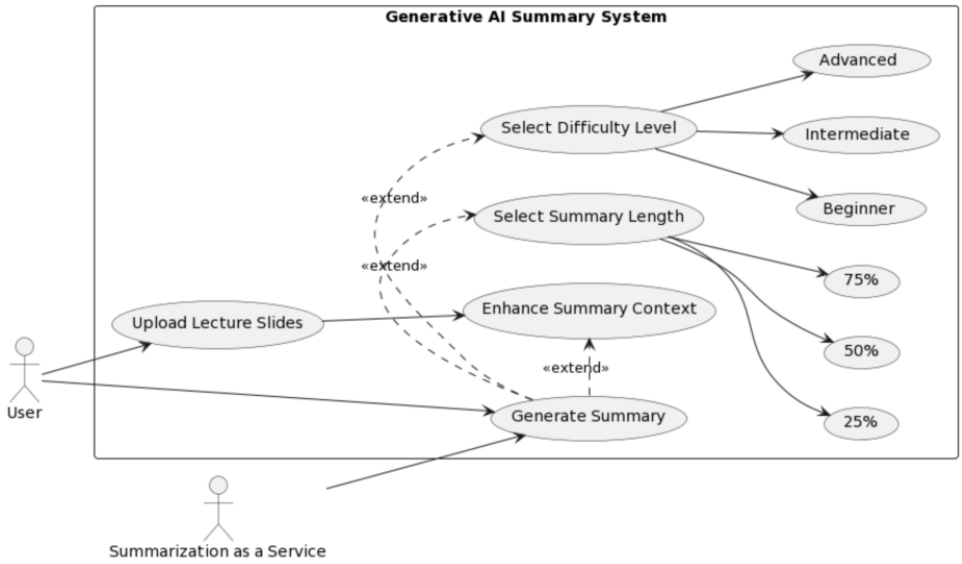
Database Utilization for Transcript Management



Develop a robust database system to facilitate efficient storage, retrieval, and management of lecture transcripts. This system will allow users to search for specific transcripts based on multiple parameters such as course code, semester and year, lecture number, and week. Additionally, the database will incorporate a user feedback mechanism through an upvote/downvote system to assess the quality and accuracy of the generated transcripts. Transcripts with a net score (upvotes minus downvotes) below -100 will be flagged for review or removal, ensuring the maintenance of high-quality content within the database.

- **Upload Threshold Enforcement**: The system shall automatically check the existing count of available transcripts for a given course code, semester & year, lecture number, and week. If there are 3 or more transcripts with a net score greater than -100, the system will not permit additional uploads for that specific lecture
- **Search Functionality**: Enable precise search results within 2 seconds of query submission, using filters such as course code, semester & year, lecture number, and week.
- **Feedback Mechanism**: Implement an upvote/downvote system for each transcript, with real-time updates to the net score (upvotes minus downvotes). Automatically flag transcripts with a net score below -100 for review and hide it from selection.

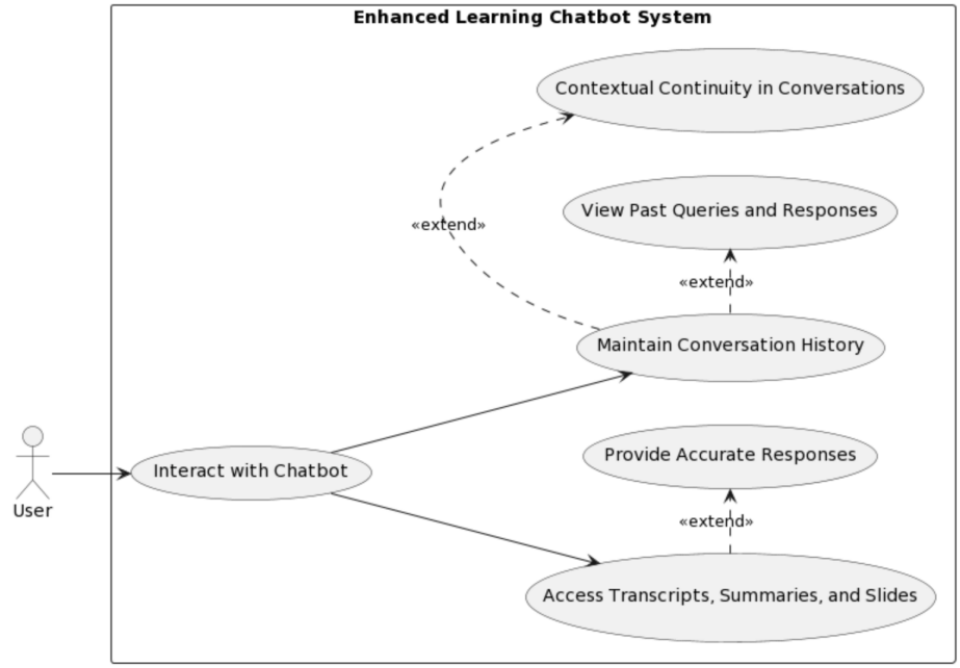
Summary Generation with Generative AI



Utilize generative AI technologies to produce concise and coherent summaries of the generated transcripts. These summaries will be tailored based on varying levels of difficulty (beginner, intermediate, advanced) and desired length (25%, 50%, 75%), catering to different student needs and preferences. The system will also allow users to upload relevant lecture slides, which the AI can use as additional context to enhance the accuracy and relevance of the summaries. This feature aims to provide students with quick and customizable access to the essence of lecture content, facilitating efficient study and revision processes.

- **Difficulty Customisation**: The Generative AI system must be capable of adjusting the complexity of summaries according to three predefined difficulty levels: beginner, intermediate, and advanced in the choice of vocabulary, concept depth, and assumed prior knowledge
- **Length Customisation**: The system shall offer users the option to specify the length of the summary as a percentage of the original transcript's length, with selectable options at 25%, 50%, and 75%. The generated summary must adhere to the chosen length with a tolerance of ±5% to accommodate natural language constraints.
- **Optional Slides Upload for Added Context**: Users shall have the option to upload relevant presentation slides alongside the lecture transcript. The Generative AI will analyze these slides to incorporate key points, terms, and concepts into the summary, providing a richer and more contextually accurate output. The system must ensure compatibility with common presentation formats such as PPT, PPTX, and PDF.

Interactive Chatbots for Enhanced Learning



Implement interactive AI-powered chatbots that can engage with users in a conversational manner, allowing them to ask questions related to the lecture summaries. These chatbots will utilize the information from the transcripts, summaries, and any uploaded slides to provide accurate and helpful responses. This interactive component is designed to mimic a personalized tutoring experience, enabling students to clarify doubts, explore topics in depth, and reinforce their understanding of the lecture material.

- **Content Integration**: The chatbots will have integrated access to the transcripts, summaries, and uploaded slides. This enables them to draw upon the comprehensive content repository to provide accurate and contextually relevant responses to user inquiries.
- **Contextual Memory & Conversation History**: The chatbot shall maintain a memory of previous interactions with each user, allowing for continuity in conversations and enabling users to view their past queries and the chatbot's responses for seamless learning and reference.

Input Requirements

Lecture Recordings Submission

Each lecture recording submitted to the system must be in a supported audio or video format. The system will automatically detect and process these formats to generate transcripts. Users can upload recordings directly from live lectures or pre-recorded sessions, ensuring comprehensive coverage of all lecture content. The system supports a wide range of audio and video formats to ensure user convenience and system accessibility. For audio submissions, supported formats include MP3, WAV, and AAC. For video submissions, the system is compatible with MP4, AVI, and MOV formats.

Search Parameters

To facilitate efficient retrieval of specific transcripts, the system incorporates advanced search functionality. Users can search for transcripts based on multiple parameters such as course code, semester, year, lecture number, and week.

Feedback Mechanism

The system includes a user feedback mechanism through an upvote/downvote system for each transcript. This feature allows users to rate the quality and accuracy of transcripts, contributing to content quality control.

Summary Customisation Options

The system allows users to customize summaries based on their preferred level of difficulty and length. Difficulty levels include beginner, intermediate, and advanced, which influence the choice of vocabulary and depth of concepts in the summaries. For length customisation, users can select summaries to be a specific percentage of the original transcript's length, with options such as 25%, 50%, and 75%.

Optional Presentation Slides Upload

Users have the option to upload presentation slides alongside lecture transcripts to provide additional context for summary generation and chatbot interactions. The system supports common presentation formats such as PPT, PPTX, and PDF.

Chatbot Integration Context

For the interactive AI-powered chatbots, the system requires input in the form of user queries related to lecture content. The chatbots utilize information from transcripts, summaries, and uploaded slides to provide accurate and contextually relevant responses.

Process Requirements

SQL Transaction

The system employs SQL transactions to ensure that all database operations related to the note-taking process are executed safely and reliably. This includes the creation, updating, and deletion of transcripts, user feedback, and summary data. The use of transactions guarantees the atomicity, consistency, isolation, and durability (ACID) of data operations, preventing partial updates and maintaining the integrity of the database in the event of system failures or concurrent access conflicts.

Consistency Priority (CAP Theorem)

In aligning with the principles of the CAP theorem, which posits that a distributed system can only simultaneously ensure two out of three properties—Consistency, Availability, and Partition Tolerance—the note-taking system places a priority on Consistency. This approach ensures that all users receive the most recent and accurate version of any data, such as transcripts or summaries, regardless of which node in the system they access it from. By prioritizing consistency, the system aims to maintain the reliability and accuracy of educational content, crucial for the integrity of the learning experience. While availability and partition tolerance are also important, the system is designed to slightly compromise on these aspects when necessary to uphold data consistency, especially in scenarios that demand real-time updates and edits to shared educational materials.

Data Security

To prevent SQL injection in the note-taking system, stringent input validation and parameterized queries are employed. User inputs, particularly those influencing database queries, are rigorously validated against a defined set of criteria to ensure they contain only expected values. Parameterized queries are utilized across the system, where inputs are treated as parameters rather than direct SQL code, effectively separating the data from the command. This approach, coupled with the use of prepared statements, ensures that user inputs cannot alter the structure or intent of SQL statements, thereby safeguarding the database from malicious injections and maintaining the integrity and security of the system's data.

Data validation

In the context of the note-taking system, it is crucial to meticulously handle data errors that may arise from user interactions or during backend database processing. The system incorporates comprehensive data validation mechanisms and robust error-handling routines to ensure the integrity and reliability of inputs, such as lecture uploads, transcript edits, and summary requests. These safeguards are designed to gracefully manage any discrepancies or issues, maintaining the system's operational fluency and ensuring a seamless user experience. By preemptively addressing potential data inaccuracies and system anomalies, the note-taking platform upholds its commitment to delivering accurate, reliable, and high-quality educational content.

Output Requirements

Generated Transcript

The core functionality of the system is to provide accurate and comprehensive transcripts of lectures. These transcripts can be generated in two primary ways: through real-time speech-to-text processing during live lectures, and by converting uploaded audio or video recordings into text. The system ensures that these transcripts are readily available and accessible on the server for users, facilitating immediate or future reference.

Customisable Summaries

To aid in efficient study and revision, the system offers a feature to generate concise summaries of the detailed transcripts. Users can tailor these summaries according to their preferred level of complexity (beginner, intermediate, advanced) and length (25%, 50%, 75% of the original transcript). This customization allows users to engage with the content in a manner that best suits their learning needs and preferences.

Chatbot Interactions

An AI-powered chatbot provides an interactive output, offering personalized responses to user queries based on the lecture content. This feature simulates a tutoring experience, allowing users to delve deeper into topics, clarify doubts, and reinforce their understanding. The chatbot's responses are dynamically generated, drawing from the information within the transcripts and summaries to provide accurate and relevant information.

Hardware Requirements

Network Infrastructure

High-speed Internet connectivity to support real-time data processing and cloud-based services. Secure and reliable network infrastructure to facilitate seamless communication between client devices and servers.

Client Devices

Compatibility with a wide range of devices including smartphones, tablets, laptops, and desktop computers. Support for both wired and wireless devices to ensure accessibility from various locations.

Servers

Robust server infrastructure capable of handling speech-to-text processing, database management, and AI computations. Redundant server setups to ensure high availability and data backup solutions for disaster recovery.

Peripheral Devices

Support for audio recording devices and cameras for direct lecture recording within the application environment.

Software Requirements

Operating Systems

Compatibility with major operating systems such as Windows, macOS, Linux for desktop clients, and iOS, Android for mobile clients.

Web Browsers

Support for the latest versions of major web browsers including Chrome, Firefox, Safari, and Edge to ensure wide accessibility.

Speech-to-Text and AI Services

Integration with advanced speech-to-text services with Whisper APIs. Use of ChatGPT API and machine learning platforms for summary generation and chatbot functionalities.

Database Management

Use of robust database systems of PostgreSQL for efficient data storage and retrieval. Implementation of secure database management practices to protect sensitive information and ensure data integrity.

Development Tools and Libraries

Utilization of development frameworks and libraries compatible with the application's technology stack, including React for frontend development and Node.js for backend development.

Security and Compliance Software

Implementation of security software and services to protect against threats, including firewalls, anti-virus, and intrusion detection systems. Compliance with data protection regulations through the use of encryption libraries and secure data handling practices.

Collaboration and Version Control

Use of version control systems like Git for collaborative development and source code management. Integration with project management tools like Jira or Trello to streamline the development process and track progress.

Licenses

Necessary licenses for proprietary software, development tools, and third-party services used within the application ecosystem.

Deployment Requirements

For the deployment of the note-taking application, the system leverages Vercel, a cloud platform optimized for ease and efficiency. Vercel provides seamless deployment capabilities, enabling instant pushes of new code versions to production with minimal configuration. Its support for a range of modern development frameworks ensures that the application can be deployed swiftly, with automatic scaling to handle varying loads. This approach facilitates rapid iteration and continuous delivery, allowing for quick updates and enhancements to the application while ensuring high availability and performance.