

Design report

Table of Contents

Contents

Table of Contents

Design Strategies

[Pre-Development Planning](#)
[Development Approach](#)
[Maintainability Focus](#)
[System Enhancements](#)
[Maintainability Best Practices](#)

Architectural Patterns

[Core Components of SOA](#)
[Service Registry](#)
[Service Repository](#)
[API Gateway](#)
[Services](#)
[Communication Protocols](#)
[Benefits of SOA in NTU Note-Taking System](#)
[Challenges and Considerations](#)

Software Configuration Management Tools

[MediaWiki](#)
[GitHub](#)

Design Strategies

Pre-Development Planning

- **Scalability and Flexibility:** Prioritizing scalability to handle increased traffic and data as NTU's system usage grows, ensuring performance remains unaffected.
- **Modular Architecture:** Employing a modular approach, such as Service-Oriented Architecture (SOA) or Microservices, to facilitate independent development, maintenance, and scaling of system components.

Development Approach

- **Agile and Test-Driven Development (TDD):** Adopting Agile methodologies complemented by TDD to ensure high code quality and facilitate maintenance.
- **User-Centric Design:** Designing with a focus on user experience, incorporating continuous feedback from students and faculty to enhance system usability.

Maintainability Focus

- **Corrective Maintainability:** Implementing robust error logging and monitoring to quickly identify and fix system faults. Emphasis on automated testing to detect errors early.
- **Preventive Maintainability:** Ensuring the code is clean, understandable, and well-documented, with regular code reviews and refactoring sessions to maintain code health.

System Enhancements

- **Adaptive Maintainability:** Designing the system to be flexible, allowing for easy adaptation to technological advancements in AI and speech-to-text technologies.
- **Perfective Maintainability:** Establishing CI/CD pipelines for quick iterations and enhancements post-launch, enabling rapid adaptation to user feedback and changing requirements.

Maintainability Best Practices

- **Readable and Well-Documented Code:** Adhering to coding conventions and extensive documentation, including in-line comments and API docs, to facilitate understanding and maintenance.
- **Modularity and Reusability:** Ensuring system components are reusable and loosely coupled, minimizing the impact of changes in one module on others, thereby reducing error introduction risk.
- **Automated Testing:** Implementing comprehensive automated testing, including unit, integration, and system tests, to ensure system reliability and ease defect identification and correction.
- **Continuous Learning and Improvement:** Promoting continuous skill development within the team to stay abreast of best practices in software design and maintainability, through regular training and workshops.

Architectural Patterns

SOA is a design pattern that allows services to communicate with each other over a network through a loose coupling of components, promoting interoperability and flexibility. This architectural style is particularly suited for the NTU note-taking system due to its modular and scalable nature.

Core Components of SOA

Service Registry

- The **Service Registry** acts as a directory for all available services in the system. It allows services to register themselves and discover other services, facilitating dynamic binding and lookup.

Service Repository

- The **Service Repository** stores the actual implementations of the services along with their respective versioning information. This component ensures that the correct version of a service is available for consumption.

API Gateway

- The **API Gateway** serves as the single entry point for all client requests. It routes these requests to the appropriate service, handles load balancing, enforces security policies, and can provide additional functionalities like caching and request transformation.

Services

- **Automated Note-Taking Service:** Converts lecture audio to text using speech-to-text technology.
- **Transcript Management Service:** Manages the storage, retrieval, and management of lecture transcripts within a database.
- **AI-Generated Summaries Service:** Creates summaries of transcripts using generative AI technologies.
- **Interactive Chatbot Service:** Offers a personalized tutoring experience by answering user queries based on lecture materials.

Communication Protocols

Services communicate using standard protocols such as HTTP/REST or messaging queues, ensuring that the communication is stateless, and services remain loosely coupled. This allows for greater flexibility in service development, deployment, and maintenance.

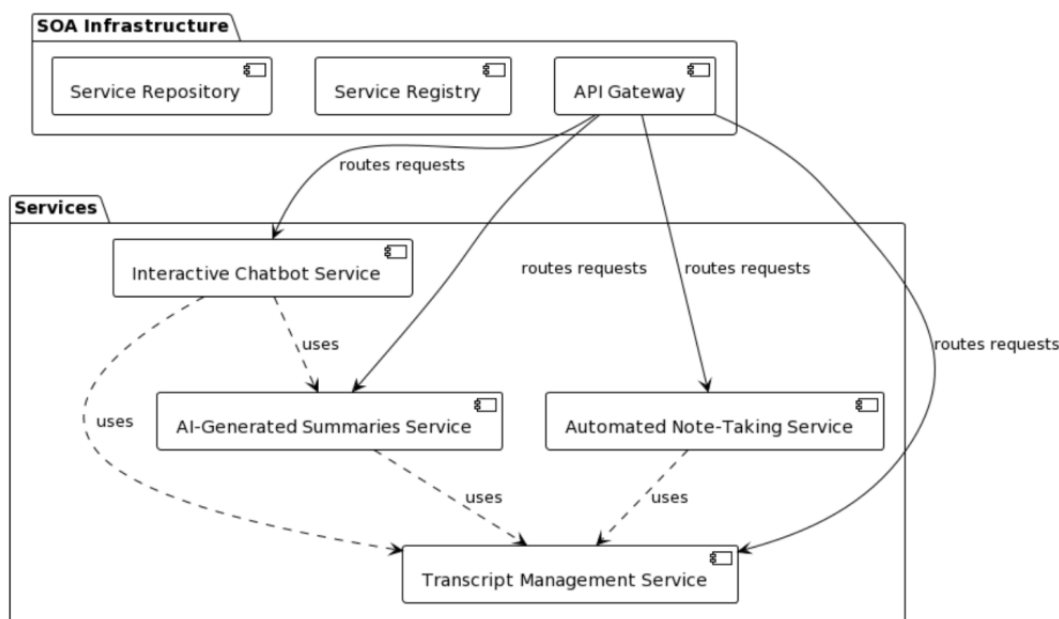
Benefits of SOA in NTU Note-Taking System

- **Scalability:** Services can be scaled independently based on demand, improving the system's overall responsiveness and efficiency.
- **Flexibility:** New services can be added or existing services updated without disrupting the entire system, allowing the system to evolve with changing requirements.
- **Reusability:** Services can be reused across different parts of the system or in other systems, reducing development time and costs.
- **Interoperability:** SOA facilitates integration with external systems and services, enhancing the system's capabilities through third-party services.

Challenges and Considerations

- **Service Governance:** Effective governance mechanisms are required to manage service lifecycle, maintain service quality, and ensure compliance with standards.
- **Security:** Robust security measures must be in place to protect sensitive data and ensure safe communication between services.
- **Performance:** Proper monitoring and optimization strategies must be implemented to mitigate any potential performance overhead introduced by the SOA infrastructure.

By leveraging SOA, the NTU note-taking enhancement system can achieve high levels of modularity, maintainability, and scalability, making it well-suited to meet the dynamic needs of the university's academic environment.



Software Configuration Management Tools

This section discusses the tools used for version control management, enabling tracking of modifications, who made them, and when they were made.

MediaWiki

MediaWiki is a free and open-source platform, renowned for its user-friendliness, especially for beginners. It provides extensive FAQs to guide users through its functionalities, offering a variety of formatting options for content creation. One of its key features is the support for concurrent edits on the same page, ensuring that no information is lost during collaborative modifications.

GitHub

GitHub serves as a source code hosting platform, leveraging Git's distributed version control and source code management capabilities. Chosen for its widespread use and integration with numerous IDE applications, GitHub facilitates issue tracking akin to a ticketing system. Users can report software bugs, suggest code enhancements, or highlight documentation needs through issues, which can be labeled and assigned to team members for resolution, keeping all stakeholders updated on the issue's progress.

Retrieved from "http://155.69.100.30/s22324SCELMagnificent7/index.php?title=Design_report&oldid=166"

This page was last edited on 2 April 2024, at 22:43.