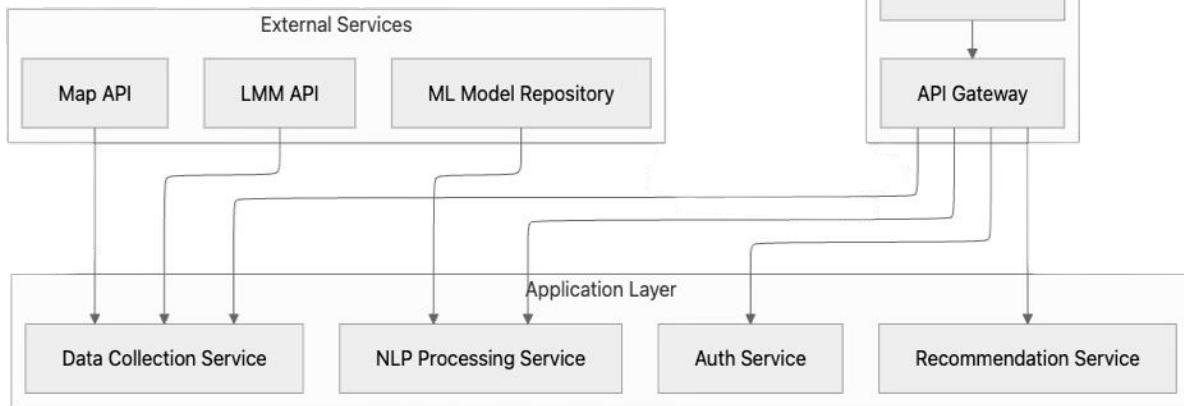


Cureat

Personalized Gastronomy Exploration and Recommendation Platform

System Architecture

Modularized core functionalities (Authentication, Map, Data Collection, NLP Processing, Recommendation) to ensure system flexibility and scalability. User requests are distributed via an API Gateway, and a caching mechanism implemented to maximize the efficiency of the response. Vector DB (ChromaDB) is utilized as a core repository to implement the AI-based personalized gastronomy recommendation system.



Core Technology

AI-based Natural Language Analysis and Recommendation System

Natural language queries input by the user are tokenized and refined via the Okt morphological analyzer. The refined text is deeply analyzed and converted into high-dimensional vectors.

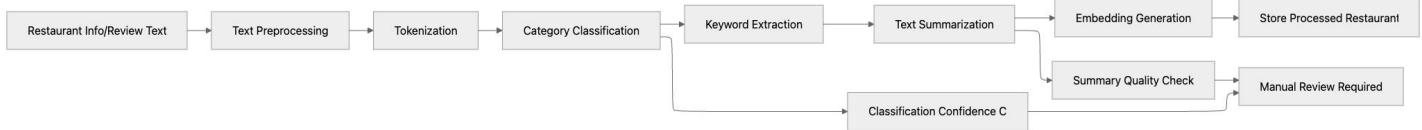
Restaurant information and review texts collected via external APIs and web crawling are also refined.

The Cosine Similarity algorithm is used to perform similarity analysis between the user input vector and the stored document vectors, providing a personalized list of gastronomic recommendations.

Efficient Data Collection and Management System

Implemented a real-time information collection logic through external RESTful API integration and Web Crawling or RSS Feeds to secure highly reliable, up-to-date gastronomy information.

Applied a logic to identify and filter promotional content from collected reviews to enhance the quality (Data Quality) and reliability of recommendation data.



Problem Solving

The technical problem was defined as API response Latency, which occurred because LLM communication and complex embedding lookup processes acted as major I/O-Bound tasks in the initial recommendation system.

The solution involved leveraging the asynchronous features of the Python FastAPI framework and the asyncio library to parallelize the processing of LLM API calls and Vector DB I/O-Bound tasks, thereby improving Concurrency.

A strategy was adopted to utilize the Vector DB as an embedding data cache to minimize redundant calculations for similar inputs.

Retrospective

Acquired integrated Full-Stack development capabilities by leading the entire system from design to implementation, including Front-end and Back-end development.

Experienced project scheduling management using Jira in an Agile/Scrum methodology and active collaboration and version control through Git/GitHub. The experience of handling various technology stacks and resolving complex system stability issues contributed to establishing the foundational knowledge and problem-solving skills necessary for growth as a future Solution Engineer or Integrated Architect.



FastAPI



Hugging Face

