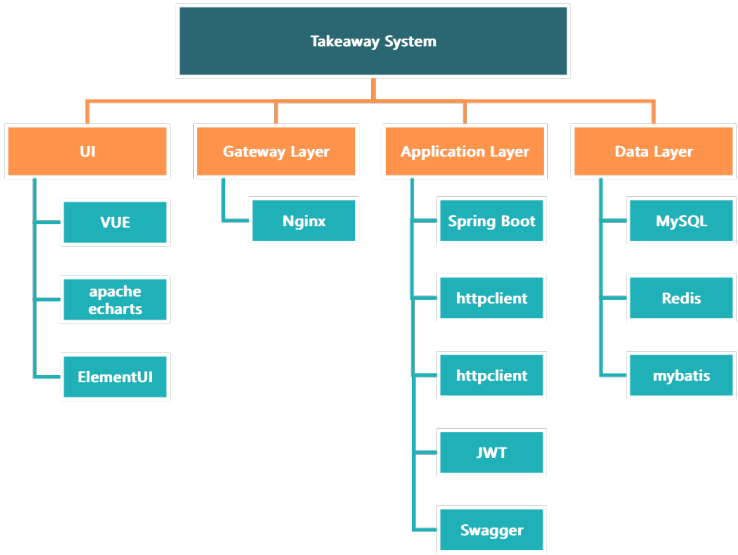
Team member: Zhixun Liu(zl5203), Siyue Zheng(sz4562)

Project Overview

In the final project, we will design a takeaway system. This project is a food delivery application based on a microservices architecture, with front-end and back-end development separated. The project implements core features such as registration and login, menu display, cart management, order processing, and payment functionalities (Merchant Management Portal for Takeaway System).

Technology Stack

1. The project is developed in Java, leveraging Java object-oriented programming concepts, and includes the use of collections, multithreading, and more.
2. The back-end stack includes SSM (Spring, Spring MVC, MyBatis) framework, MySQL database, RabbitMQ for messaging, and Spring Cloud.
3. We also use the tools like Maven, JUnit, and Swagger are extensively used to improve development efficiency and ensure software quality.



Problem-Solving Highlights

We addressed several key challenges:

1. Adopted a microservices architecture, breaking the system into independent services (e.g. order service, product service), which can be developed and deployed independently to enhance scalability and maintainability.
2. Incorporated distributed transaction control in core business areas such as order processing and inventory management, ensuring data consistency and reliability even in cases of system failure or network delays.
3. Optimized the order processing workflow using Redis caching and message queues, resulting in improved system performance.

4. Leveraged Spring Cloud for service registration, discovery, and load balancing to guarantee efficient communication between microservices, improving availability and fault tolerance.

Project Development Plan

Environment Setup

1. Set up the back-end environment.
2. Set up the front-end environment.
3. Use Git for version control of both front-end and back-end projects.

Back-End Development

Development based on different system business modules:

1. Order management and payment interface.
2. Combo (package) management.
3. Dish management.
4. Category management.
5. Employee management.
6. Data analytics.

Front-End Development

1. Login page.
2. Dashboard UI: Today's data, order management, dish overview, combo overview.
3. Order management UI.
4. Combo management UI.
5. Dish management UI.
6. Category management UI.
7. Employee management UI.
8. Data analytics UI.

Front-End and Back-End Integration (Testing)

1. Test login to ensure correct flow.
2. Perform CRUD tests on employees, dishes, combos, and categories to ensure back-end data is correctly updated and reflected in the front-end.
3. Simulate user ordering behavior and test the order workflow to ensure proper updates to inventory, order volume, and dish quantities.
4. Bulk import of test data to validate proper data aggregation and display on the dashboard.