Information Systems and Economics (2022)
Clausius Scientific Press, Canada

DOI: 10.23977/infse.2022.030410
ISSN 2523-6407 Vol. 3 Num. 4

Design and Implementation of Online Food Ordering System Based on Springcloud

Yu Yang*

College of Information Engineering, Guangzhou Vocational and Technical University of Science and Technology, Guangzhou, Guangdong, 510555, China
**Corresponding author*

Keywords: Micro-service, springcloud, vue.js, meal ordering system, load balancing

Abstract: Due to the severe form of the epidemic, most of the living communities in China have been closed, which brings great inconvenience to the residents' life. Based on this social background, an online meal ordering system is particularly important, which can not only meet the dining needs of different residents, but also accurately determine the amount of food ingredients to avoid waste. The system adopts the development mode of front and rear end separation, the front end adopts the current popular vue.js framework technology, the back end adopts the springcloud micro service architecture, provides four services: account service, menu service, order service, user service, using feign technology to complete the service load balancing, not only can expand the scope of service, but also can improve the work efficiency.

1. Background

1.1. Design Background

As the epidemic spread around the world, home quarantine and homework have become the norm, and online services are developing rapidly. In order to better facilitate the life of residents, the proposal of online meal ordering system is imminent. Traditional monomer structure development mode has shown many deficiencies. Although it can operate normally temporarily, the system has poor scalability, low efficiency, complex operation and other conditions that make the later maintenance become difficult. This paper investigates the requirements of online ordering system and designs an online ordering system based on springcloud micro-service architecture.

1.2. Technological Choices

1.2.1. Front-End Framework Technology

The front-end framework of this system adopts vue.js, Vue.js framework, which can run separately from the back-end and can be more focused on front-end development. Therefore, this system selects Vue.js as the front-end framework technology, combined with Element-UI template technology, to complete the design of the front-end page.

1.2.2. Back-End Frame Technology Selection

This system uses springcloud micro-service plus it as the back-end framework. Compared with the single architecture such as springmvc and springboot, the microservice architecture has obvious advantages of [1-4]. Single architecture in small micro enterprise is common, a typical representative is an application, a database plus a Web container can run, single structure development after a period of time, the company's business model was recognized, trading volume is slowly up, the enterprise in order to cope with greater traffic, you need to split the original business, such as background system, front-end system, trading system, etc., and such split will make late maintenance work. And micro service architecture emphasizes the business system needs complete component and service, a component is a product, can independently provide services, it no longer emphasize the traditional SOA architecture inside heavy ESB enterprise service bus [5], instead emphasize each micro service has its own independent operating space, including database resources. On the other hand, the micro service architecture itself comes from the idea of the Internet, so the services released by the components use HTTP Rest API, so that the micro service will be smaller, so now most developers now choose springcloud as the back-end technology [6-10].

1.3. The Reason for Selecting the Springcloud

(1) Spring Cloud comes from Spring, and its quality, stability and sustainability can be guaranteed.
(2) Spring Cloud naturally supports Spring Boot, which is easier for business landing.
(3) The Spring Cloud iterates very fast.
(4) Spring Cloud is the most suitable framework for micro-services in the Java field.
(5) Compared to other frameworks, Spring Cloud supports the most around the microservice frameworks.
(6) For small and medium-sized enterprises, the threshold of use is relatively low.
(7) Spring Cloud is the best landing solution for the microservice architecture.

2. Overall Structure of the System

2.1. Requirement Analysis

The system is divided into two interfaces: client and background management system. The client is mainly for ordinary users. The functions include user login, user exit, food ordering and my order. The background management system is for administrators, including administrator login, administrator exit, add dishes, query dishes, modify dishes, delete dishes, order processing, add users, query users, delete users and other functions.

2.2. Overall Design of the System

According to the requirements analysis, the system is divided into two main parts: client and background management system. The client includes: user login, food ordering, my order and other modules. The background management system includes administrator login, dish management, order management, user management and other modules, as shown in Figure 1.

In Figure 1, the client part is, mainly the customer part. Customers can complete the login function, order dishes, add, delete, modify, query dishes and other operations, and also manage orders. The back-end management system part is mainly the administrator part. The administrator can complete the login function, and can add, delete, modify, query and other operations on the menu, add, delete, modify, query orders and other operations, and can manage users.

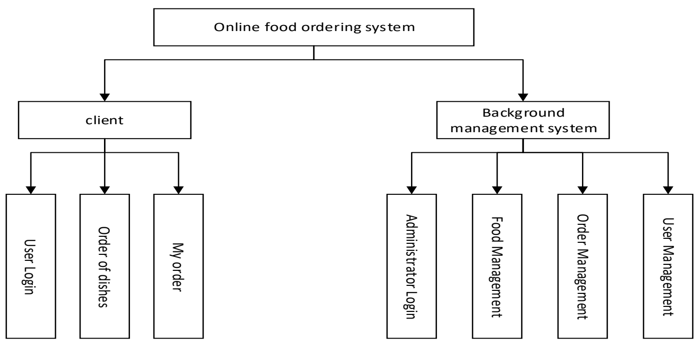


Figure 1: Overall Design Drawing

According to the demand analysis of the online meal ordering system, the system is divided into 7 modules: service consumer module, registration center module, 4 service provider module, and configuration center module. The service provider module assigns four service providers: account, menu, order, and user, as shown in Figure 2.

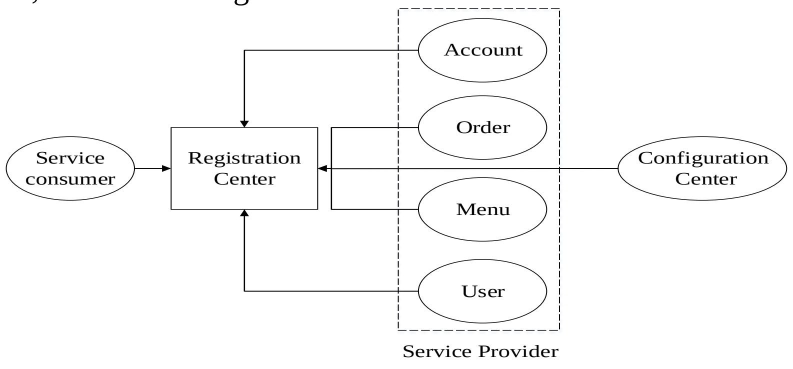


Figure 2: Overall Design drawing of the system

As shown in Figure 1, the system consists of 7 modules, including the registry center, the configuration center, the service consumers, and 4 service providers.

2.3. Module Function Design

2.3.1. Service Consumer Module

The system will assign a service consumer, including the front-end page and background interface of the client, the front-end page and background interface of the background management system. The resources directly accessed by the user / administrator are stored in the service consumer.

2.3.2. The Service Provider Module

There are four service provider modules assigned: account, menu, order, and user.account mainly provides account services, such as user and administrator logins. The menu provides food services: adding, checking, modifying and deleting dishes. The order provides order services: adding orders, checking orders, deleting orders, and processing orders. Finally user provides user services including adding users, querand deleting users. Service consumers call the corresponding interfaces of the four service providers to complete the business logic, and complete the load balancing through feign.

2.3.3. Registry Center Module

One service consumer and four service providers have to complete the registration before running, so the system has designed the registry module, service consumers and service providers need to complete the registration in the registry, and register the configuration center to provide remote configuration information reading.

2.3.4. Configure the Central Module

The Configuration Center module saves the configuration information for the four service provider modules and provides the configuration information for use by the registry.

3. Realize Core Technology

3.1. Database-Specific Implementation

The system will set up the database entity through the pojo layer, using the mybatis technology and the database association.

The implementation code is as follows:

```
@MapperScan("edu.gkd.repository")
//This annotation allows you to associate the mybatis with the repository layer.
public class MenuApplication {
public static void main(String[] args) {
SpringApplication.run(MenuApplication.class,args);
```

In the code, by noting the associated mybatis files, add and delete the data.

3.2. Service Consumer Settings

The service consumer is configured as follows:

```
spring:
cloud:
config:
name: user #Corresponding profile name
label: master #gitWarehouse branch name
discovery:
enabled: true
serviceId: configserver #The configuration center name for the connection
eureka:
client:
serviceUrl:
defaultZone: http://localhost:8971/eureka/
```

After the user configuration of service consumption, the eureka service interface is used to complete the service discovery function.

3.3. Login Identity Switch

Since there is only one login interface, you need to switch to two identities: ordinary customer and administrator.

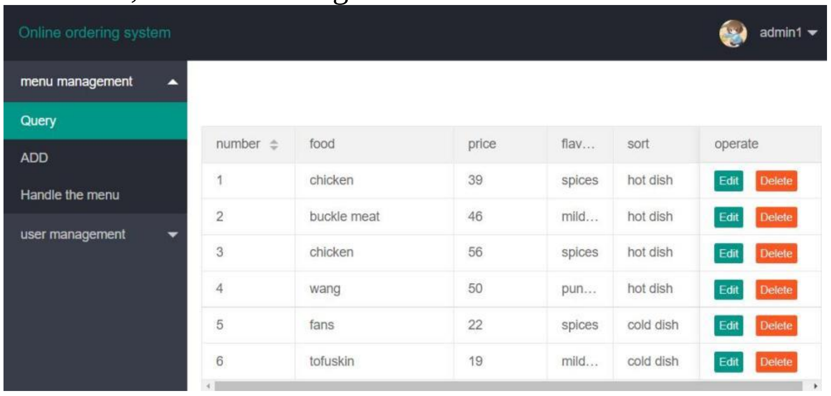
```
@GetMapping("/login/{username}/{password}/{type}")
public Object login(@PathVariable("username") String username,@PathVariable("password") String password, @PathVariable("type") String type){
Object object = null;
switch (type){             //The type is the data coming over from the front desk
```

```
    case "user":
    object = userRepository.login(username,password);
    break;
    case "admin":
    object = adminRepository.login(username,password);
    break;
    }
    return object;          //If the user name and password are correct, the user is query and //returned,
otherwise the login fails
    }
```

In code a method needs to return two objects and the parameter type is used to switch identities. The return type is Object, with either a normal user type or an administrator type.Object can represent both user or admin, enabling the multiplexing of objects.

4. Test

Test this system, first start the registry, then start the Configuration Center, and then start the service providers: account, menu, order, user. Start the client again, log in to the administrator account, and enter the main interface, as shown in Figure 3.



number	food	price	flav...	sort	operate
1	chicken	39	spices	hot dish	Edit Delete
2	buckle meat	46	mild...	hot dish	Edit Delete
3	chicken	56	spices	hot dish	Edit Delete
4	wang	50	pun...	hot dish	Edit Delete
5	fans	22	spices	cold dish	Edit Delete
6	tofuskin	19	mild...	cold dish	Edit Delete

Figure 3: Query menu results diagram

As shown in Figure 3, in the upper left corner of the figure, the name of the system, and the right is the name of the login user. Since the login is the administrator, the left menu bar shows menu management and user management. The menu management module is mainly responsible for food increase, deletion, modification, query and other functions, while the user management module is mainly responsible for customer increase, delete, modification, and query and other functions. The content part shows the content of the query menu, which can be edited and deleted. In the upper right corner of the interface, click the drop-down button to opt out.

5. Conclusion

This paper designs an online meal ordering system based on the springcloud microservice architecture, which can not only support extended services, but also ensure continuous and high-quality service delivery. This system adopts the producer and consumer design mode, separating the customer and the service side with the service registry, which not only expands the service, but also greatly improves the work efficiency. This system combines Eurake, zuul, feign and other components together, using the development environment for IDEA, using Mysql as the main storage medium, using micro service ideas, the system into four service providers: account, menu, order, user, a registration center, a configuration center and a service consumer: client, realize the detailed functions including login, order, dish selection, user management, not only to meet the design needs, obtain good practicability, It also solves the problem that people can not eat normally in home isolation.

References

[1] Suryotrisongko H, Jayanto D P, Tjahyanto A. Design and development of backend application for public complaint systems using microservice spring boot. Procedia Computer Science, 2017, 124: 736-743.
[2] Teng F, Wu Q. Design and Implementation of the Information System of Retired Veteran Cadres Bureau Based on SpringBoot Framework[C]//2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). IEEE, 2021: 87-92.
[3] Mythily M, Raj A S A, Joseph I T. An Analysis of the Significance of Spring Boot in the Market//2022 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2022: 1277-1281.
[4] Menezes G, Cafeo B, Hora A. How are framework code samples maintained and used by developers? The case of Android and Spring Boot. Journal of Systems and Software, 2022, 185: 111146.
[5] Pytel K, Marcinkowska R, Rutkowska M, et al. Recent advances on SOA formation in indoor air, fate and strategies for SOA characterization in indoor air-A review. Science of the Total Environment, 2022: 156948.
[6] Quiña-Mera A, Flores Landeta D, Rea-Peñafiel X M, et al. Quality Evaluation of a Spring Cloud Microservices Architecture Implementation//Communication, Smart Technologies and Innovation for Society. Springer, Singapore, 2022: 745-754.
[7] Zhou Z. Integration and Optimization of O2O System for College Art Guiding Plartform Implementation based on Spring Cloud Framework//2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2022: 1298-1302.
[8] Xiong Q, Li W. Design and Implementation of Microservices Gateway Based on Spring Cloud Zuul//CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications. VDE, 2022: 1-5.
[9] Jiang R, Wang W, Xie Y, et al. Research and Design of Infrastructure Monitoring Platform of Intelligent High Speed Railway//2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2022, 6: 2096-2099.
[10] Zhao J, Cui J, Yuan T. Design and implementation of school-based curriculum resource System based on micro-service architecture//2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE, 2022: 1357-1360.