ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

Design of Ordering System based on Spring Boot Framework

Zhijie Li

College of Electrical and Engineering, Southwest Minzu University, Chengdu, Sichuan, 610041, China

*2894011152@qq.com

Abstract

With the rise of Internet food delivery services, online ordering business is growing. New business model realized through online and offline interaction[1,2]. Online take-out has the advantages of convenience and speed, On the basis of online ordering, online payment and other main functions, combined with a variety of preferential activities, the traditional catering industry's business model has been innovated,but also brought new business growth points,Therefore,in order to better meet the needs of customers at any time,the back end of the ordering system based on the We Chat service number platform built by Spring boot[3,4,5] and JPA is a Web APP running on the We Chat public number constructed by VUE.JS[6,7,8] .The system follows the MVC.development pattern.The front and rear ends are separated and connect to each other through REST[9,10] interfaces.From the point of view of technology development,the overall design of the ordering system,to achieve user registration and login,menu view,ordering generation, intelligent ordering and other functions.

Keywords

Spring Boot; JPA; Order System; Restful Interface.

1. System Development Environment and Overall Framework

The system uses IDEA as a development tool,which is different from the common Eclipse development tool:

1) Download the Java Development kit(JDK).After the installation,configure the bin directory in the JDK installation directory in the system environment variable,run the CMD console,and enter Java-v to check whether the configuration is successful.

2) Download IDEA.IDEA is a Java development tool developed by Jet Brains.Compared with the original Eclipse development tool,IDEA is more convenient and intelligent.

3) Download MySQL.MySQL is a common relational database based on disk storage data,it has good application advantages,installation files are very small,support flexible configuration ,running speed can meet user needs Especially as a free open source tool,it greatly reduces the cost of project development.

4) Download Redis.Redis is an in-memory database based on key-value pairs.Because Redis stores all data in memory,its read and write performance is amazing.Redis also has the feature of persistence,which can save the memory data to the hard disk in the form of snapshots and logs,so as to avoid the problem of memory data loss in the event of power failure or machine failure.

5) Download Spring Boot[11].Spring Boot is currently a popular framework system,it is the Spring framework family of derivatives,annotated development,support AOP programming.It can provide a container for dependency injection.It introduces the concept of automatic configuration,abandon the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

original MANUAL configuration of XML,reduces the complex and error-prone configuration problems,and is more efficient and easy to use.

The whole system is based on Java language,and the main functions of the development are user login,commodity view,menu generation and other functions.The main functional architecture of the system is shown in Figure 1.

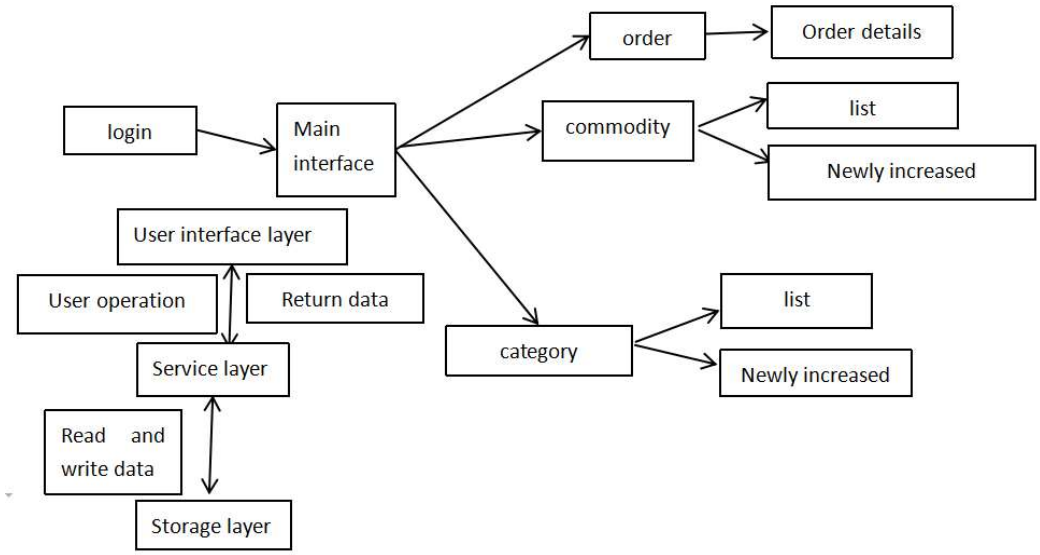

Figure 1. The diagram of the main functions system

2. System Implementation

2.1 User Login

When the user logs in to management system,he/she has obtained the unique OpenID of the user for this website.This OpenID will be passed in as the request parameter along with the user's request path.Then these are three more steps:

Step1:We need to look for this OpenID in the database user information table.if the database does not have any,the user cannot log in and enter the second step.

Step2:We need to store this OpenID in Redis.First a unique token is generated as the key.

String token=UUID.randomUUID().toString();

Then save it to Redis with key and value pairs(token,OpenID).

Integer expire=RedisConstant.EXPIRE;

redisTemplate.opsForValue().set(String.format(RedisConstant.TOEKN_PREFIX,token),openid,expire,TimeUnit.SECONDS);

Step3:Set the token into the cookie.

CookieUtil.set(response,CookieConstant.TOKEN,token,expire);

Thus the(CookieConstant Token,the Token)such key and the value to save the user's cookies. CookieUtil is the tool class we wrote.the set method for setting cookies is as follows :

public static void set(HttpServletResponse response,String name,String value,

int maxAge){

Cookie cookie = new Cookie(name,value);

Cookie.setPath("/");

Cookie.setMaxAge(maxAge);

response.addCookie(cookie);

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

```
}
```

After these three steps,the user can successfully in to the system.Below is the complete login process:

```
@GetMapping("/login")
public ModelAndView login(@RequestParam("openid")String openid,
HttpServletResponse response,
Map<String,Object> map){
SellerInfo sellerInfo = sellerService.findSellerInfoByOpenid(openid);
If(sellerInfo==null){
map.put("msg",ResultEnum.LOGIN_FAIL.getMessage());
map.put("url","/sell/seller/order/list");
Return new ModelAndView("common/error");
}
String token=UUID.randomUUID().toString();
Integer expire=RedisConstant.EXPIRE;
redisTemplate.opsForValue().set(String.format(RedisConstant.TOKEN_
    PREFIX,token),openid,expire,TimeUnit.SECONDS);
CookieUtil.set(response,CookieConstant.TOKEN,token,expire);
return new ModelAndView("redirect:"+projectUrlConfig.getSell()+
"/sell/seller/order/list");
}
```

2.2 User Logout

User logout also requires three steps:

Step1:When the user logout request is made,go to the request object HttpServletRequest and look for the cookie whose key is Cookieconstant.TOKEN.

```
Cookie cookie = CookieUtil.get(request,CookieConstant.TOKEN);
```

Step2:Clear user login information in Redis.

```
redisTemplate.opsForValue().getOperations().delete(String.format(RedisConstant.
    TOKEN_PREFIX,cookie.getValue()));
```

Step3:Clear the user's cookie information.

```
CookieUtil.set(response,CookieConstant.TOKEN,null,0);
```

Then the user has logged out successfully. Here is the complete logout process: @GetMapping ("/logout");

```
public ModelAnd View logout(HttpServletRequest request,
HttpServletResponse response,
Map<String,Object>map){
Cookie cookie = CookieUtil.get(request,CookieConstant.TOKEN);
if(cookie != null){
redisTemplate.opsForValue().getOperations().delete(String.format(RedisConstant.TOKEN_PREFI
X,cookie.getValue()));
CookieUtil.set(response,CookieConstant.TOKEN,null,0);
}
map.put("msg",ResultEnum.LOGOUT_SUCCESS.getMessage());
map.put("url","/sell/seller/order/list");
```

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

return new ModelAndView(“commone/success”,map);

}

2.3 AOP Implement Authentication

Before the user did not log in,it still can access the commodity management system resources,and now we can use Spring AOP to implement some preposition of the access.

First defining a class SellerAuthorizeAspect,with @Aspect note.Then declare all controllers starting with Seller as pointcuts,excluding SellerUserController,because that Controller is the one that validates the user’s login.

@Pointcut(“execution(public * com.imocc.controller.seller*.*(..))”+

”&&!execution(public *com.imocc.controller.SellerUserController.*(..))”)

public void verify(){}Finally,some preposition is done for this tangent.After the user logs in,cookie and Redis should contain the user’s information according to the logic written before.Therefore,these two places are now queried to verify whether the user logs in.

@Before(“verify()”)

public void doVerify(){

ServletRequestAttributes attributes=(ServletRequestAttributes)

RequestContextHolder.getRequestAttributes();

HttpServletRequest request=attributes.getRequest();

Cookie cookie = CookieUtil.get(request,CookieCXonstant.TOKEN);

if(cookie == null){

Log.warn(“can’t find token in Cookie”);

Throw new SellerAuthorizeException();

}

}

The cookie object is obtained through the CookieUtil utility class.If the current cookie does not exist,there is no token in the current cookie and a custom SellerAuthorizeException is thrown.

String tokenValue = redisTemplate.opsForValue().get(String.format.

(RedisConstant.TOKEN_PREFIX,cookie.getValue()))

if(StringUtils.isEmpty(tokenValue)){

Log.warn(“can’t find token in Redis”);

Throw new SellerAuthorizeException();

}

As we can see from the code above,if the user is not logged in,a SellerAuthorizeException is thrown,which is a custom exception.This exception is simple,with a simple definition,runtime exception.

Public class SellerAuthorizeExcception extends RuntimeException{}

After that,we need to define a handler for this exception,SellerExceptionHandler.

when this exception is caught,it means that user is not logged in,and then jump to the login interface again.

2.4 Distributed Session Problem

When a user request an application Web page for the first time,the Web server automatically creates a Session object for the user to store information about the user,such as the user’s login information,so that multiple requests can be located to the same context.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

As Web sites become more functional and more accessible.Web applications deployed as a single machine service are no longer supported.At this point,you need to optimize or adjust the current architecture,how to optimize,or which parts to optimize first,depending on the specific situation of the site.According to usage:

if the database is under heavy pressure,you can set read/write separation and separate database and tables.There are vertical and horizontal partitions for database and tables(for example,if the user table has a large amount of data,you can design the tables according to certain rules,but the table structure remains the same).

If the Web application server is under heavy pressure,you can add a service deployment application,that is,change a singe service to a cluster.After the cluster,which server does the user choose to visit the website?this need to be solved by adding load balancing devices in front of the application server.

In the second case, the problem of session management, which we will discuss in a moment, arises: sessions on distributed systems.the solution includes Session Stick.

Session replication,centralized Session management,and Cookie based management.

2.4.1 Session Stick

In the case of a single machine,the session is stored on the single machine and requests are sent to the single machine,so there is no problem.if yo can ensure that every request goes to the same service,it will be the same as a single machine.this parameter needs to be modified on the load balancing device,there are several problems with this approach:

1) If a server breaks down or restarts,the session data on the server is lost,if the session data contains login status information,users need to log in again.

2) Loading balancing processes the mapping between specific sessions and servers.

2.4.2 Session Replication

Session replication,as the name implies,means that each application service stores Session data,which is supported by common application containers.Session replication has less load balancing requirements than Session Stick.But there are drawbacks:

1) Synchronizing session data incurs network overhead.As long as session data changes,it needs to be synchronized to all machines.The more machines,the more network overhead.

2) Because each server stores session data,if the cluster has a large number session data,such as 900,000 people visiting a website,the content used to store session data on each machine will be heavily occupied.This solution relies on the application container to complete,does not depend on the application,if the number of application services is not very many,can be considered.

2.4.3 Centralized Session Management

This is also easy to understand.Add another service to manage session data.Each application service takes session data from the dedicated session management service.you can use databases,NOSQL database,etc.Compared with Session replication,it reduces the memory usage of each application service and network overhead caused by Session synchronization.but there are drawbacks:

1) Read/write sessions introduce network operations.Compared with local read/write sessions,they bring latency and instability.

2) If the Session centralized service is faulty,the application may be affected.

2.4.4 Cookie-based management

The last one is Cookie based management,where we store session data in cookies,and then when the request comes in,we get session from the Cookie.Compare with centralized management,this solution does not rely on external storage systems,and does not have the network operation delay and instability caused by reading and writing session data.but there are still drawbacks:

1) Cookies have a length limit,which affects the length of session data.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

2) Security. Session data is originally stored on the server, but in this solution, session data is transferred to the external network or the client, so there are security issues. However, session data written to cookies can be encrypted.

3) Bandwidth consumption. Since each Http request carries session data,bandwidth of course increases a bit.

4) Performance consumption. Each Http request and response carries Session data,and the Web server supports more concurrent requests with less output of the response for the same processing.

2.5 Integrate Redis Using the Spring Catching Mechanism

The cached classes must be serialized,implementing the Serializable interface so that they can be stored in the corresponding encoding via Spring's serialize and cached in Redis,or read back into the corresponding encoding via Redis and deserialize into the corresponding Java objects.

In the Spring project it provides an interface called CacheManager to define the CacheManager so that different caches can implement it to provide the manager's functionality.In the spring-data-redis.jar pckage,the CacheManager interface is RedisCacheManager,so the Bean RedisCacheManager needs to be defined ,but RedisTemplate needs to be defined first.The following uses the annotation-driven RedisCacheManager definition:

```
@Configuration;
@EnableCaching;
public class RedisConfig{
@Bean(name=”redisCacheManager”)
Public CacheManager initRedisCacheManager(@Autowired;
RedisTemplate redisTemplate){
RedisCacheManager cacheManager = new RedisCacheManager();
List<String> caceNames = new ArrayList<String>();
cacheNames.add(“redisCacheManager”);
cacheManager.setCacheNames(cacheNames);
return cacheManager;
}
}
```

With the cache manager bean in place,Spring allows caching to be annotated.There are three common annotation:@cacheable,@cacheput,and @cacheevit.

@cacheable:if there is a value in the cache,the cached data is returned,otherwise the access method gets the data.

@cacheput:Means that the method is executed anyway,and finally the return value of the metho is saved to the cache,and the database data is updated simultaneously with the cache,right.

@cacheeevit:For mainly remove key/value pairs from the cache.For deleted operations.remove the cache after the method is completed.

2.6 Uses Redis for Cache in Spring Boot

In heavy traffic scenarios,caching can effectively improve data read speed.The following three concepts are commonly encountered in caching:

1) Hit: An application retrieves data from the cache and returns it.

2) Invalidation: When the cache time expires, the cache is invalidated.

3) Update: The application stores the data to the database and then puts it back into the cache.

The following describes the process of using annotations to implement Redis caching.

1) Annotate @enablecaching on the startup class.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

2) With the @cacheable annotation on the Controller method that returns front-end data that needs to be cached,the data is cached the first time the method is accessed and read from the cache the next time the interface is accessed.

3) The @cachePut(cachNames=”product”,key=”123”)annotation is used to modify the data.Each time a method under this annotation is accessed,the returned data is written to Redis.the data must be serializable.However,if our method does not return a custom type,we can use the @cacheevict (cacheNames=”product”,key=”123”).

Annotation.when accessing the method under this annotation, Redis will erase the cached data.

2.7 Wechat Related Webpage Authorization and Payment

This system involves logging in the wechat public number for relevant operations. if a user accesses a third-party web page in the wechat client,the public account can obtain the user’s basic information through the wechat page authorization mechanism,so as to realize the business logic.

2.7.1 Description of Web Page Authorization Callback Domain Name

Before the wechat public account requests the user’s webpage authorization,the developer needs to modify the authorization callback domain name in the configuration option of “development-interface permission-Webpage service-Webpage account -Webpage authorization to obtain the user’s basic information”in the official website of the public platform.it is important to note that the domain name(which is a string)is not a URL,so do not include protocol headers such as http://.

The authorization callback domain name configuration specification is the full domain name.Such as need web authorized domain is :www.qq.com,configuration after this model under the domain name http://www.qq.com/music.html, can OAuth2.0 authorization However,http://pay.qq.com,cannot perform OAuth2.0 authentication.

If the login of the public account is authorized to a third party developer for management,there is no need to do any settings,and the third party public account can realize webpage authorization.

2.7.2 The Difference between the Two Scopes of Web Page Authorization

1) The web page authorization initiated by snsapi_base as scope is used to obtain the openID of the user who enters the page, and is silently authorized and automatically jumps to the callback page. What users perceive is that they are going straight to the callback page (usually the business page).

2) Web page authorization initiated by snsapi_userinfo as scope is used to obtain user basic information. However, this authorization requires the user's manual consent, and since the user has agreed, the basic information of the user can be obtained after authorization without concern.

3) The "Access to Basic User Information interface" in the user management interface can obtain basic user information according to the user OpenID only after the user and the public account have message interaction or concern. This interface, including other wechat interfaces, requires the user (namely, OpenID) to pay attention to the public account before it can be successfully called.

2.7.3 About the Difference between Access_Token and Ordinary Access_Token for Web Page Authorization

Wechat web page authorization is realized through OAuth2.0 mechanism. After the user authorizes the public, the public account can obtain a unique interface call certificate (access_token) for web page authorization.Other wechat interfaces need to use the "Access Access_Token" interface in basic support to obtain common access_token invocation.

The main steps of the web authorization process are divided into four steps:

(1) Guide the user to enter the authorization page to agree the authorization and obtain the code. On the premise that the wechat public account has the authorization scope, guide followers to open the following page: https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&state=STATE #wechat_redirect if the message "This link cannot be accessed" is displayed, check whether the parameters are correctly specified and whether you have the authorization scope permission corresponding to the scope parameter.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

(2) Exchange web authorization access_token with code (different from access_token in base support). What you're exchanging for with code is a special Web page authorization Access_token, unlike the Access_token in base support, which is used to invoke other interfaces. Public accounts can obtain web page authorization access_token through the following interface. If the scope of web authorization is SNsapi_base, the access_token and OpenID of web authorization are obtained in this step. The snsapi_Base web authorization process ends here.

(3) If necessary, developers can refresh the web license access_token to avoid expiration. Because the access_token has a short validity period, when the access_token times out, the refresh_token can be refreshed. The refresh_token validity period is 30 days. When the refresh_token becomes invalid, users need to re-authorize the access_token.

(4) Access to basic user information through Web authorization Access_Token and OpenID (support UnionID mechanism) If the scope of web authorization is SNSAPi_userinfo, developers can access user information through access_Token and OpenID.

2.7.4 Wechat Pay

The business flow chart is as follows:

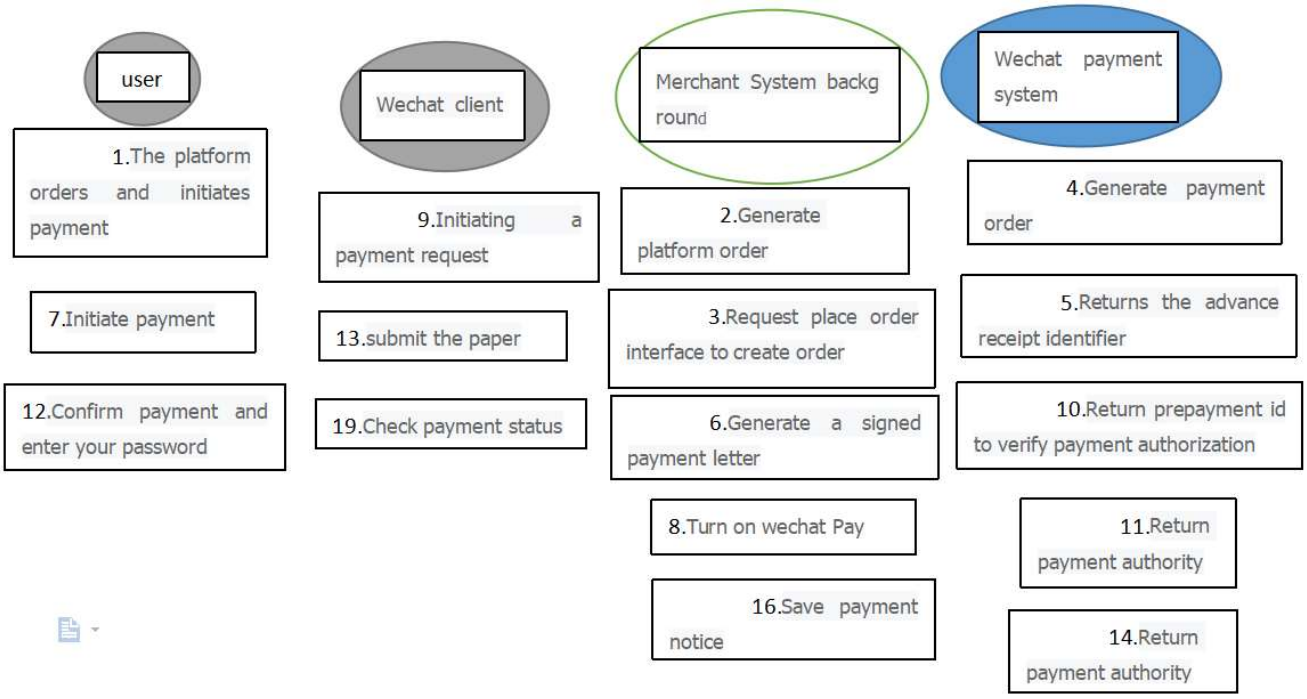

Figure 2. Wechat Payment business flow chart

Main interactions between the commodity system and wechat Payment system:

① Merchant server invokes unified ordering interface to request orders.

② Merchant server can activate wechat Payment through JSAPI to initiate payment request.

③ Merchant server receives payment notification.

④ Merchant server queries payment results.

3. Mode of Operation

1) Import the project to IDEA. In IDEA, File->open... , and then select the project folder (Spring Boot-Project). If you are using Spring Boot for the first time, this process may take a bit longer and require downloading many dependent JAR packages.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

2) The configuration file of the project is in the resources directory, application.yml file. Example Modify the connection information of the MySQL database.

3) Run the SQL script for table construction on the MySQL database terminal (or use the Navicat for MySQL graphical tool just downloaded). The table construction sentence of this project is sqmax.sql under the root path of the project.

4) Finally, you can start the project. Run the SellApplication main class in IDEA as Spring Boot. As you can see, this is different from the way our traditional Web projects start up. We didn't configure a server like Tomcat, because Spring Boot has already introduced the server into the startup dependency.

5) After the above steps, our project should be ready to start. Can come to our seller side of the business management system interface.

4. Conclusion

The ordering system is based on Java language, MySQL database, RPC framework, using Spring Boot as the business logic framework, hierarchical structure, combined with JPA, Lombok, Redis cache cluster and other technologies. To realize the configurable and calculable category service of ordering software.

The operation system supporting the service system, based on Spring and Spring MVC framework, realizes the flexible, strict and traceable operation configuration function of category service and supports category configuration, graph title configuration and other functions. 、By integrating Redis with Spring cache, it effectively realizes the data reading speed in high concurrency scenarios, improves the traceability of the operation system, and makes the system have the advantages of quick response, perfect function and convenient operation, which has high use value.

References

[1] Bo Zhang, Business revolution int the area of mobile Internet, Mechanical Industry Press, 2013.

[2] Zhao Zhiwei, Zhang Shengyue, JIANG Yingju, Tu Xiaoguang. Design and implementation of innovation capability evaluation Platform for high-tech enterprises based on SpringBoot [J]. Modern information technology, 2021, 5 (15) : 40 and 42. DOI: 10.19850 / j.carol carroll nki. 2096-4706.2021.15.011.

[3] LI Peng. Implementation of Fast Development Platform based on SpringBoot [J]. Electronic Technology and Software Engineering,2021(12):36-37.

[4] Kaiping Ye, Weisheng CAI, Jiamin Chen, Sini Deng. Research and design of Integrated Visual Management System based on SpringBoot [J]. Computer knowledge and technology, 2021 (12) : 100-104. The DOI: 10.14004 / j.carol carroll nki CKT. 2021.1155.

[5] Xiong Baixiang. Design and implementation of examination resource service platform based on Springboot and Vue framework [J]. Information & Computer (Theory Edition),202,34(01):97-99+103.

[6] Wang Wei, Chang Qingli, Wu Zhaoxia. Research and implementation of precise helping System based on Springboot+Vue [J]. Henan Science and Technology, 201,40(27):12-14.

[7] Xiang Fuchuan, Fang Yu, Liu Lang, Tang Zhenyun, Lian Yao. Design and Development of Collaborative Talent Education System based on SpringBoot+Vue Framework [J]. Modern information technology, 2021, 5 (14) : 5-7 + 12. DOI: 10.19850 / j.carol carroll nki. 2096-4706.2021.14.002.

[8] Shan Shuqian, Ren Jiaxun. Web Design and Implementation based on SpringBoot and Vue Framework [J]. Computer knowledge and technology, 2021 (30) : 41 + 40-50. DOI: 10.14004 / j.carol carroll nki CKT. 2021.2868.

[9] Huo Fuhua, Han Hui. An MVVM Model based on SpringBoot Microservice Architecture [J]. Electronic Technology and Software Engineering,2022(01):73-76.

[10] ZUO Hao. Design and Development of Visual Platform based on MVC Pattern [J]. China New Communication, 201,23(19):46-48.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202205_8(5).0074

[11] Yu Xin, LIAO Chenling, ZHOU Weili. Based on MVC software architecture reconstruction and optimization research [J]. Journal of jilin institute of chemical technology, 2021, 38 (7) : 49-52. DOI: 10.16039 / j.carol carroll nki cn22-1249.2021.07.010.

[12] Chen Heng, Lou Oujun, GONG Qingzhi, Zhang Lijie. Spring MVC development technology guide [J]. Journal of computer education, 2021 (7) : 194. DOI: 10.16512 / j.carol carroll nki jsjjy. 2021.07.043.