## 一 系统分层图

外部接口或第三方平台

通用处理层（Manager层）

数据持久层（DAO层）

业务逻辑层（Service层）

请求处理层（Web层）

开放接口

终端显示层

## 二、系统架构图

层

微信端

PC端

展示层

静态资源

zqls-ui

基础框架

zqls-common

代码生产器

zqls-dao

反馈中心

zqls-feedback

用户服务

zqls-ucenter

日志分析

zqls-log

数据分析

zqls-data

配置中心

zqls-config

支付中心

zqls-pay

Zookeeper

Dubbo

安全控制

文件

Mysql

数据管理

基础服务

公共服务

微信平台

PC平台

业务系统

权限管理

zqls-umps

用户角色

集中权限

权限服务

分布式事务

注册和发现

授权和监控

服务治理

接口服务

接口系统

zqls-api

加密解密

## 三、数据库结构图

### 实体：

人、食物

### 表结构：

#### zqls-ums

|  |
| --- |
| **用户表**：id号（主键），账户id（外键），工号（学工号与职工号），用户名（姓名），密码，性别，头像，电话号码，邮箱，描述，状态，类型，创建时间，最近一次操作时间，最近一次操作ip地址，操作者。  **账户表**：id号（主键），账户id，账户余额，流水类型，流水id（外键）。  **消费表**：id号（主键），消费时间，消费地点，消费金额。  **充值表**：id号（主键），充值时间，充值方式，充值金额。  **角色表**：id号（主键），角色名，类型，状态，描述，最近一次操作时间，最近一次操作ip地址，操作者。  **权限表**：id号（主键），权限码，权限名，请求url，类型，状态，描述，最近一次操作时间，最近一次操作ip地址，操作者。  **用户角色表**：id号（主键），用户id（外键），角色id（外键），最近一次操作时间，最近一次操作ip地址，操作者。  **权限角色表**：id号（主键），权限id（外键），角色id（外键），最近一次操作时间，最近一次操作ip地址，操作者。 |

#### zqls-fms

|  |
| --- |
| **食物表**：id号（主键），名称，价格，描述，口味类型，基本特征，状态，图片，所在餐厅，类型，创建时间，最近一次操作时间，最近一次操作ip地址，操作者。  **食物消耗表**：id号（主键），食物id（外键），工号（外键），食物质量，类型（早中晚），创建时间，更新时间。  **用户喜好表**：id号（主键），用户id（外键），民族，地区，口味（酸辣甜等口味），忌口（葱姜蒜香菜等忌口），喜好菜类，创建时间，更新时间。  **食物推荐表：**id号（主键），用户id（外键），食物id（外键），类型（膳食推荐与喜好菜品推荐），创建时间，更新时间。  **膳食推荐参数表（待定）**：id号（主键），用户id（外键），身高，体重，创建时间，更新时间。  **生产量推荐表（待定）**：id号（主键），食物id（外键），生产量，创建时间，更新时间。  **生产量推荐参数表（待定）**：id号（主键），食物消耗id（外键），创建时间，更新时间。  **时间表（待定）**：id号（主键），日期，星期，类型（周末，节假日），描述。 |

#### zqls-log

|  |
| --- |
| **日志表**：id号（主键），类型，目标id，旧值，新值，状态，最近一次操作时间，最近一次操作ip地址，操作者。 |

#### zqls-feedback

|  |
| --- |
| **反馈表**：id号（主键），用户id（外键），反馈类型（食物，食堂环境，系统改进），反馈内容，状态，是否有图片，图片url，创建时间，更新时间。 |

#### zqls-system

|  |
| --- |
| **系统配置表（待定）**：id号（主键），系统配置名，系统配置值，状态，创建时间，最近一次操作时间，最近一次操作ip地址，操作者。 |

#### zqls-pay(待定)

|  |
| --- |
|  |

## 四、应用技术

### 后端：

开发环境：

Jdk1.8+

Tomcat8.5+

Maven3.6+

Redis5.0+

Mysql8.0+

Zookeeper3.4+

Dubbo-admin2.6+

Nignx1.12+

Apache24 2.4.41+：高并发压力测试

技术：

|  |  |
| --- | --- |
| SpringBoot | 微服务框架 |
| Spring | 容器 |
| SpringMVC | MVC框架 |
| Mybatis | ORM框架 |
| mybatis-generator | Mybatis代码生成器 |
| Druid | 数据库连接池框架 |
| mysql-connector-java | Mysql连接驱动 |
| ZooKeeper | 分布式协调服务 |
| fastjson | Json框架 |
| dubbo | 分布式服务框架 |
| redisson | Redis分布式框架 |
| fastdfs | Fastdfs文件系统客户端 |
|  |  |

依赖关系

Zqls-xxx-web：xxx模块web端

Zqls-xxx-service：xxx模块service端

Zqls-web-api：web端依赖

Zqls-service-api：service端依赖

Zqls-common-api：通用依赖

Zqls-parent：依赖版本管理

Zqls-parent：

|  |
| --- |
| <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>  <!-- 项目版本管理 -->  <fastjson.version>1.2.60</fastjson.version>  <dubbo-starter.version>1.1.2</dubbo-starter.version>  <dubbo.version>2.6.6</dubbo.version>  <zkclient.version>0.11</zkclient.version>  <mybatis.version>2.1.0</mybatis.version>  <nekohtml.version>1.9.22</nekohtml.version>  <xml-apis.version>1.4.01</xml-apis.version>  <batik-ext.version>1.9.1</batik-ext.version>  <jsoup.version>1.11.3</jsoup.version>  <httpclient.version>4.5.8</httpclient.version>  <commons-lang3.version>3.9</commons-lang3.version>  <mybatis-generator.version>1.3.7</mybatis-generator.version>  <jedis.version>2.9.1</jedis.version>  <jest.version>5.3.4</jest.version>  <jna.version>5.3.1</jna.version>  <beanUtils.version>1.9.3</beanUtils.version>  <druid.version>1.1.18</druid.version>  <netty.all.version> 4.1.38.Final</netty.all.version>  <guava.version>28.0-jre</guava.version>  <redisson.version>3.11.1</redisson.version>  <fastdfs.version>1.27-SNAPSHOT</fastdfs.version>  <pagehelper.version>1.2.10</pagehelper.version>  <lombok.version>1.18.8</lombok.version>  <joda-money.version>LATEST</joda-money.version>  </properties>  <!-- 引入依赖 -->  <dependencyManagement>  <dependencies>  <dependency>  <groupId>org.joda</groupId>  <artifactId>joda-money</artifactId>  <version>${joda-money.version}</version>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <version>${lombok.version}</version>  </dependency>  <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper-spring-boot-starter</artifactId>  <version>${pagehelper.version}</version>  </dependency>  <!-- 添加fastdfs -->  <dependency>  <groupId>org.csource</groupId>  <artifactId>fastdfs-client-java</artifactId>  <version>${fastdfs.version}</version>  <exclusions>  <exclusion>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  </exclusion>  </exclusions>  </dependency>  <!-- redisson -->  <dependency>  <groupId>org.redisson</groupId>  <artifactId>redisson</artifactId>  <version>${redisson.version}</version>  </dependency>  <!-- guava -->  <dependency>  <groupId>com.google.guava</groupId>  <artifactId>guava</artifactId>  <version>${guava.version}</version>  </dependency>  <!-- https://mvnrepository.com/artifact/com.alibaba/fastjson -->  <!-- json -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>fastjson</artifactId>  <version>${fastjson.version}</version>  </dependency>  <!-- 分布式服务 -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <version>${dubbo.version}</version>  </dependency>  <dependency>  <groupId>com.gitee.reger</groupId>  <artifactId>spring-boot-starter-dubbo</artifactId>  <version>${dubbo-starter.version}</version>  </dependency>  <!-- Zookeeper之开源客户端 -->  <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  <version>${zkclient.version}</version>  </dependency>  <!-- netty -->  <!-- https://mvnrepository.com/artifact/io.netty/netty-all -->  <dependency>  <groupId>io.netty</groupId>  <artifactId>netty-all</artifactId>  <version>${netty.all.version}</version>  </dependency>  <!-- mybatis -->  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>${mybatis.version}</version>  </dependency>  <!-- druid -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid-spring-boot-starter</artifactId>  <version>${druid.version}</version>  </dependency>  <!-- html解析 -->  <dependency>  <groupId>net.sourceforge.nekohtml</groupId>  <artifactId>nekohtml</artifactId>  <version>${nekohtml.version}</version>  </dependency>  <!-- xml -->  <dependency>  <groupId>xml-apis</groupId>  <artifactId>xml-apis</artifactId>  <version>${xml-apis.version}</version>  </dependency>  <!-- 操作svg工具包 -->  <dependency>  <groupId>org.apache.xmlgraphics</groupId>  <artifactId>batik-ext</artifactId>  <version>${batik-ext.version}</version>  </dependency>  <!-- html解析器 -->  <!-- https://mvnrepository.com/artifact/org.jsoup/jsoup -->  <dependency>  <groupId>org.jsoup</groupId>  <artifactId>jsoup</artifactId>  <version>${jsoup.version}</version>  </dependency>  <!-- http -->  <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  <version>${httpclient.version}</version>  </dependency>  <!-- 工具包 -->  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>${commons-lang3.version}</version>  </dependency>  <!-- Mybatis代码生成器 -->  <dependency>  <groupId>org.mybatis.generator</groupId>  <artifactId>mybatis-generator-core</artifactId>  <version>${mybatis-generator.version}</version>  </dependency>  <!-- redis操作 -->  <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  <version>${jedis.version}</version>  </dependency>  <!-- 测试 -->  <!-- https://mvnrepository.com/artifact/io.searchbox/jest -->  <dependency>  <groupId>io.searchbox</groupId>  <artifactId>jest</artifactId>  <version>${jest.version}</version>  </dependency>  <!-- 访问系统本地库 -->  <!-- https://mvnrepository.com/artifact/net.java.dev.jna/jna -->  <dependency>  <groupId>net.java.dev.jna</groupId>  <artifactId>jna</artifactId>  <version>${jna.version}</version>  </dependency>  <!-- javaBean操作工具包 -->  <dependency>  <groupId>commons-beanutils</groupId>  <artifactId>commons-beanutils</artifactId>  <version>${beanUtils.version}</version>  </dependency>  </dependencies>  </dependencyManagement> |

Zqls-common-api：

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.joda</groupId>  <artifactId>joda-money</artifactId>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  </dependency>  <dependency>  <groupId>org.csource</groupId>  <artifactId>fastdfs-client-java</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>fastjson</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <!-- 项目热部署 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <optional>true</optional>  </dependency>  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  </dependency>  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  </dependency>  <dependency>  <groupId>commons-beanutils</groupId>  <artifactId>commons-beanutils</artifactId>  </dependency>  <dependency>  <groupId>commons-codec</groupId>  <artifactId>commons-codec</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  </dependency>  <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  <exclusions>  <exclusion>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  </exclusion>  <exclusion>  <groupId>io.netty</groupId>  <artifactId>netty</artifactId>  </exclusion>  </exclusions>  </dependency>  <dependency>  <groupId>io.netty</groupId>  <artifactId>netty-all</artifactId>  </dependency>  <!--todo-->  <dependency>  <groupId>com.gitee.reger</groupId>  <artifactId>spring-boot-starter-dubbo</artifactId>  <!--<exclusions>  <exclusion>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-api</artifactId>  </exclusion>  </exclusions>-->  </dependency>  </dependencies> |

Zqls-web-api：

|  |
| --- |
| <dependencies>  <!-- 引入通用工具包 -->  <dependency>  <groupId>com.duanxin.zqls</groupId>  <artifactId>zqls-common-api</artifactId>  <version>1.0-SNAPSHOT</version>  </dependency>  <!-- 引入thymeleaf -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <!-- html解析 -->  <dependency>  <groupId>net.sourceforge.nekohtml</groupId>  <artifactId>nekohtml</artifactId>  </dependency>  <!-- xml -->  <dependency>  <groupId>xml-apis</groupId>  <artifactId>xml-apis</artifactId>  </dependency>  <!-- 操作svg工具包 -->  <dependency>  <groupId>org.apache.xmlgraphics</groupId>  <artifactId>batik-ext</artifactId>  </dependency>  </dependencies> |

Zqls-service-api：

|  |
| --- |
| <dependencies>  <!-- 引入通用工具包 -->  <dependency>  <groupId>com.duanxin.zqls</groupId>  <artifactId>zqls-common-api</artifactId>  <version>1.0-SNAPSHOT</version>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-jdbc</artifactId>  </dependency>  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  </dependency>  <dependency>  <groupId>org.mybatis.generator</groupId>  <artifactId>mybatis-generator-core</artifactId>  </dependency>  <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper-spring-boot-starter</artifactId>  </dependency>  <!-- 引入druid -->  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid-spring-boot-starter</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>  <!-- redis -->  <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  </dependency>  <!-- guava -->  <dependency>  <groupId>com.google.guava</groupId>  <artifactId>guava</artifactId>  </dependency>  <!-- redisson -->  <dependency>  <groupId>org.redisson</groupId>  <artifactId>redisson</artifactId>  </dependency>  </dependencies> |

### 前端：

### 软硬件数据交互：

#### 软件：

|  |
| --- |
| 采用java编写服务端，以SpringBoot+Mybatis为主编写服务端程序，采用json格式数据传输给客户端。参考：<https://github.com/BuffettCui/arduino> |

#### 硬件：

|  |
| --- |
| 采用Arduino开发板，使用c语言编写http客户端程序，利用Arduinojson库实现与服务端进行json数据格式交互，参考： 。 |

### 服务端调用深度学习模型

#### 方法一：

|  |
| --- |
| 采用tcp/ip通信方式，利用python编写服务端程序运行深度学习模型，java编写客户端程序进行数据请求，客户端服务端使用json数据格式进行数据交互，参考：<https://www.cnblogs.com/maosonglin/p/9397257.html> |

#### 方法二：

|  |
| --- |
| 利用java的Runtime.getRuntime().exec（）方法来执行python深度学习模型，参考：<https://blog.csdn.net/David_Yang_7/article/details/86156716> |

## 五、软件结构图

## 六、接口约定

### 编程规范：

|  |
| --- |
| - service类，需要在叫名`service`的包下，并以`Service`结尾，如`XxxServiceImpl`  - controller类，需要在以`controller`结尾的包下，类名以Controller结尾，如`XxxController.java`，并继承`BaseController`  - mapper.xml，需要在名叫`mapper`的包下，并以`Mapper.xml`结尾，如`XxxMapper.xml`  - mapper接口，需要在名叫`mapper`的包下，并以`Mapper`结尾，如`XxxMapper.java`  - model实体类，需要在名叫`model`的包下，命名规则为数据表转驼峰规则，如`Xxx.java`  - 类名：首字母大写驼峰规则；方法名：首字母小写驼峰规则；常量：全大写；变量：首字母小写驼峰规则，尽量非缩写  - 配置文件放到`src/main/resources`目录下  - 模块命名为`项目`-`子项目`-`业务`，如`zqls-ums-service`  - 数据表命名为：`子系统`\_`表`，如`ums\_ user`  - 更多规范，参考【阿里巴巴java开发手册】  - 分层领域模型规约： |

### 接口约定：

|  |  |  |
| --- | --- | --- |
| - 前后端命名规则按照驼峰命名法（参照编程规范），前后端数据命名应保持一致性。  - 前后端通过json数据格式进行传输数据，数据形式要求   |  | | --- | | {  code: 状态码：1成功，其他为失败,  message: 成功为success，其他为失败原因,  data: 数据结果集  } |   - 前端与后端路由接口格式（可参考RPC架构风格）   |  | | --- | | - 查询 user----GET  - 添加 user----POST  - 修改 user/{id}-----PUT  - 删除 user/{id}-----DELETE | |

用户：管理员（超级管理员，普通管理员），学生，职工（教师和工作人员）

管理员：用户名（姓名），密码，头像，电话号，邮箱，描述，状态，类型，最近一次登陆ip地址，最近一次登陆时间，操作者，创建时间，更新时间，职工号。

学生：账户，用户名（姓名），密码，头像，电话号，邮箱，描述，状态，类型，创建时间，更新时间，学工号，账户余额，消费时间，消费地点，消费金额，充值时间，充值方式，充值金额。

职工：账户，用户名（姓名），密码，头像，电话号，邮箱，描述，状态，类型，创建时间，更新时间，职工号，账户余额，消费时间，消费地点，消费金额，充值时间，充值方式，充值金额。