武汉纺织大学

Java应用开发课程设计

**基于MyBatis的超市收银系统**

**学 院： 数学与计算机学院**

**班 级： 软件11802**

**姓 名： 刘锐**

**学 号： 1804240409**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月20日**

目 录

[1 需求分析 1](#_Toc42352359)

[1.1登陆 1](#_Toc42352360)

[1.1.1 游客登录 1](#_Toc42352361)

[1.1.2 用户登录 1](#_Toc42352362)

[1.2 出题 1](#_Toc42352363)

[1.3 答题 1](#_Toc42352364)

[1.4 显示答题情况 1](#_Toc42352365)

[1.5 排行榜 1](#_Toc42352366)

[2 系统设计 1](#_Toc42352367)

[2.1用户用例图 1](#_Toc42352368)

[2.2 ER图 2](#_Toc42352369)

[2.3 UML类图（Class Diagram） 2](#_Toc42352370)

[2.4 UML时序图（Sequence Diagram） 4](#_Toc42352371)

[2.4.1.1老用户登录 4](#_Toc42352372)

[2.4.1.2游客登录 5](#_Toc42352373)

[2.4.2 出题模块 6](#_Toc42352374)

[2.4.3 答题模块（因为答题实则是将出的题渲染至界面上，用户填写答案，不涉及程序内部运行） 8](#_Toc42352375)

[2.4.4 显示答题详情模块 8](#_Toc42352376)

[2.5 UML活动图（Activity Diagram） 8](#_Toc42352377)

[2.5.1.1 游客登录 8](#_Toc42352378)

[2.5.1.2 用户登陆 9](#_Toc42352379)

[2.5.2 出题模块 10](#_Toc42352380)

[2.5.3 答题模块 11](#_Toc42352381)

[2.5.4 显示答题详情模块 12](#_Toc42352382)

[2.5.5 排行榜模块 14](#_Toc42352383)

[3 系统实现 15](#_Toc42352384)

[3.1 项目结构 15](#_Toc42352385)

[3.2 配置文件 15](#_Toc42352386)

[3.2.1 Config.java文件 16](#_Toc42352387)

[3.2.2 Path.java文件 16](#_Toc42352388)

[3.2.3 pom.xml文件 16](#_Toc42352389)

[3.3 DTO类 18](#_Toc42352390)

[3.3.1 DTO类User.java 18](#_Toc42352391)

[3.3.2 DTO实体类Paper.java 20](#_Toc42352392)

[3.3.3 DTO实体类Answer.java 21](#_Toc42352393)

[3.3.4 DTO实体类question.java 22](#_Toc42352394)

[3.4 服务类 23](#_Toc42352395)

[3.4.1 服务类UserService.java 23](#_Toc42352396)

[3.4.2 服务类文件QuestionService.java 27](#_Toc42352397)

[3.5 接口控制文件ProductApi.java 34](#_Toc42352398)

[3.6 用户现实层web 36](#_Toc42352399)

[3.6.1 mathTest.html 36](#_Toc42352400)

[3.6.2 rank.html 46](#_Toc42352401)

[3.7 demo层DemoApplication.java 48](#_Toc42352402)

[4 系统测试 49](#_Toc42352403)

[4.1.1游客登录功能 49](#_Toc42352404)

[4.1.2用户登录功能 49](#_Toc42352405)

[4.2出题功能 50](#_Toc42352406)

[4.3答题功能 51](#_Toc42352407)

[4.4显示题目详情功能 52](#_Toc42352408)

[4.5显示排行榜功能 52](#_Toc42352409)

[5 系统总结 53](#_Toc42352410)

[5.1项目不足 53](#_Toc42352411)

[5.2项目收获 53](#_Toc42352412)

# 1 需求分析

开发一个简单的算术运算测验程序，完成两位数以内两个随机整数的加减乘除运算作答（注：两个数相除时取整）。

## 1.1登陆

## 1.1.1 游客登录

用户初次使用需选择游客登录，系统随机分配昵称和ID，当完成答题提交成功时系统才会记录下该用户的所有信息

## 1.1.2 用户登录

用户再次登录时可选择用户登录，输入游客登录时系统随机分配的昵称或ID即可。

## 1.2 出题

用户可以输入获取题目的总数（不得大于100），系统即可随机生成对应数目的题目

## 1.3 答题

出题确认完毕，自动跳转至答题界面，用户可以在输入框中输入自己的答案，可对输入框进行清空等

## 1.4 显示答题情况

答题完毕，系统显示所出的题目，用户的答案及正确答案，同时显示自己的分数（采用百分制）

## 1.5 排行榜

用户可以在首页或答题结束界面点击查看排行榜按钮，进入排行榜界面显示所有用户的排名，排名涉及答题时间，答题者昵称、ID、分数和排名，排名取用户最好排名录入排行榜。

# 2 系统设计

## 2.1用户用例图

系统分为游客，老用员两个角色，各角色的用户用例如下图2-1所示：

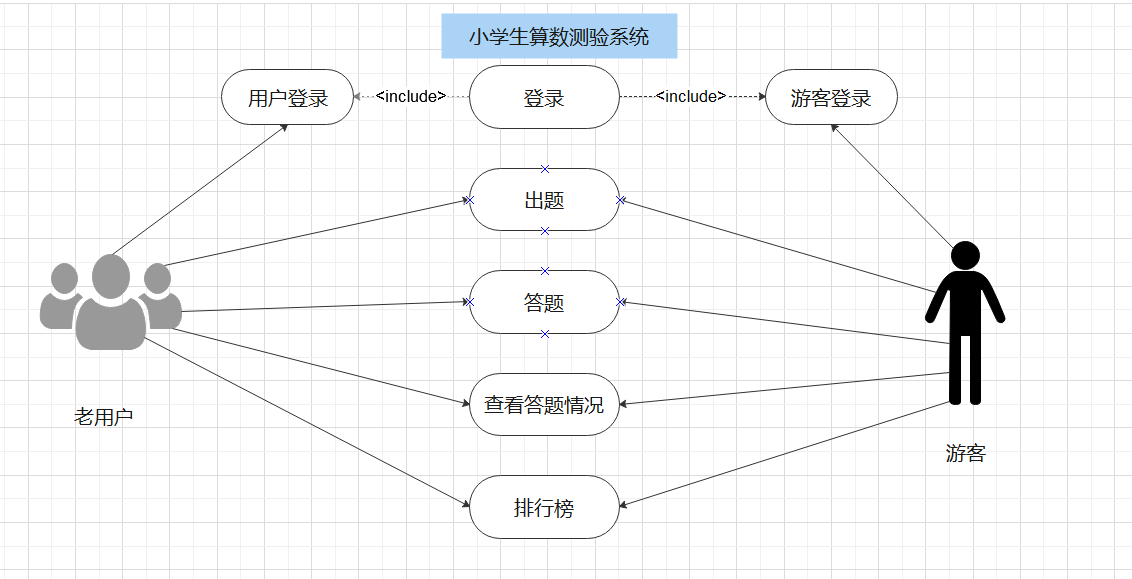


图2-1 用户用例图

## 2.2 ER图

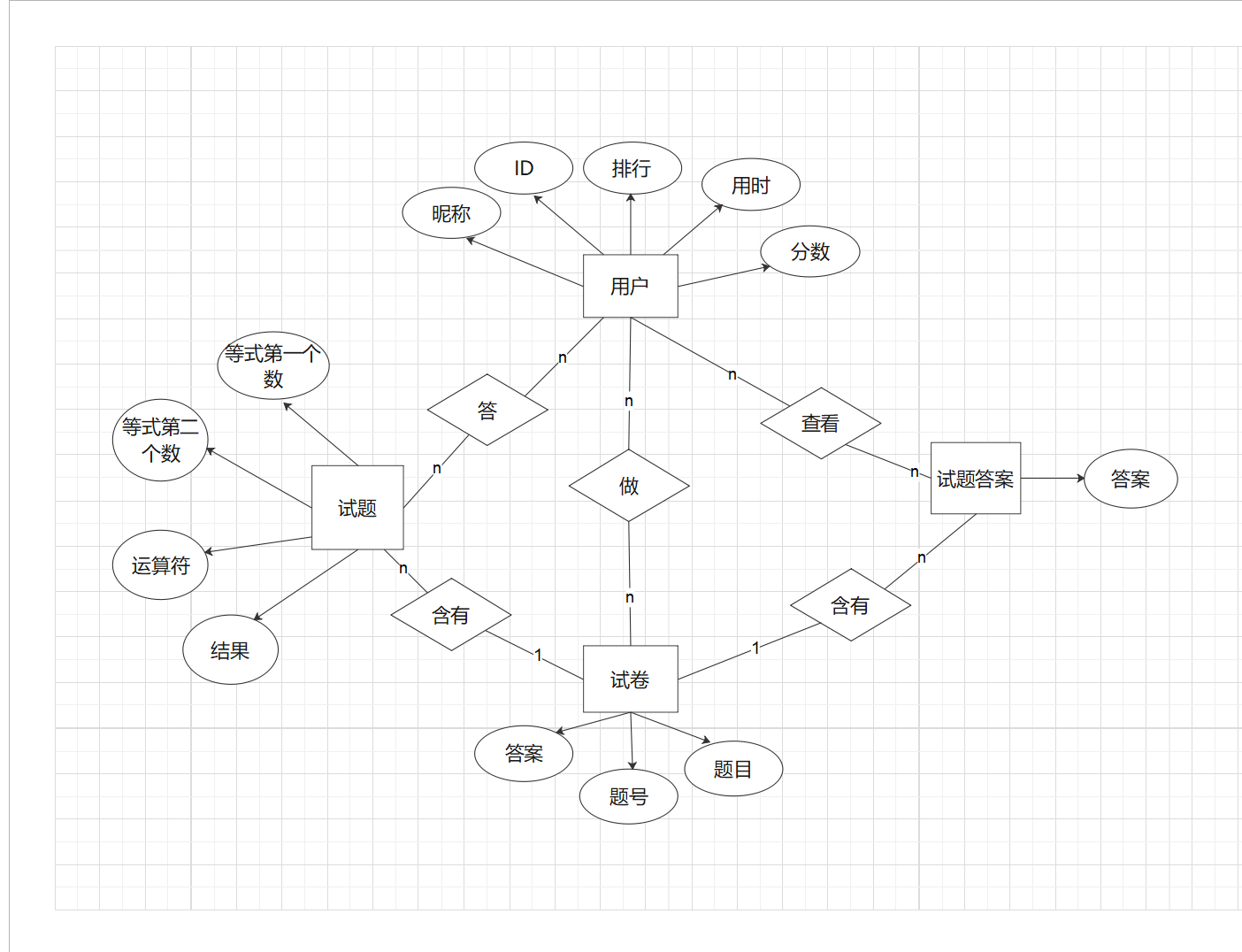


图2-2 ER图

## 2.3 UML类图（Class Diagram）

对于用户功能模块，共设计如下2个类。

* 用户服务类 UserService：生成随机昵称和ID，判断老用户登录ID或昵称是否正确，查看用户排行榜
* dto实体类User：与txt文件进行映射的类。主要由属性，setter, getter方法组成，dto类中的属性与文件中的字段相对应，每一个dto类的对象都表示表中的每一条记录
* txt 集合相互转换类 TxtSwitchArray:将txt转换成List,将List装换成txt
* controller类ProductApi:为外界提供接口类

各类的结构及类之间的关系如图2-3-1所示：

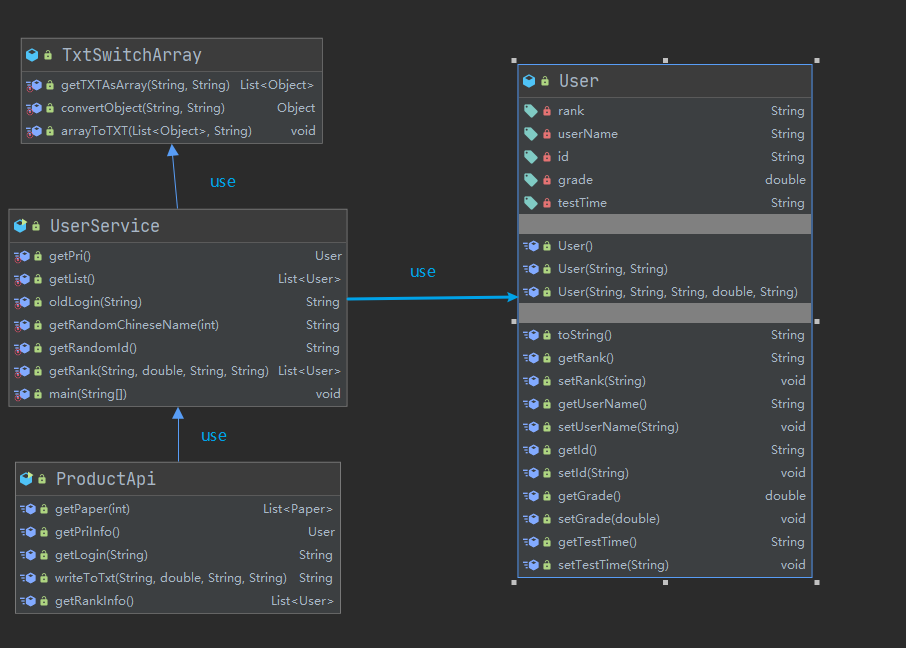


图2-3-1 用户功能类图

对于出题答题功能共设计如下个类。

* dto实体类question:内部为getter,setter,toString等为试卷的基本组成单元之一。
* Dto实体类paper: 内部为getter,setter,toString等是考试卷子完整的一体。
* Dto实体类answer:内部为getter.setter/toString 等为试卷的基本组成单元之一。
* Service服务类questionService:内部包含随机生成题目，将题目答案记录于答案集合中等
* Controller控制类productApi:为外界服务提供接口

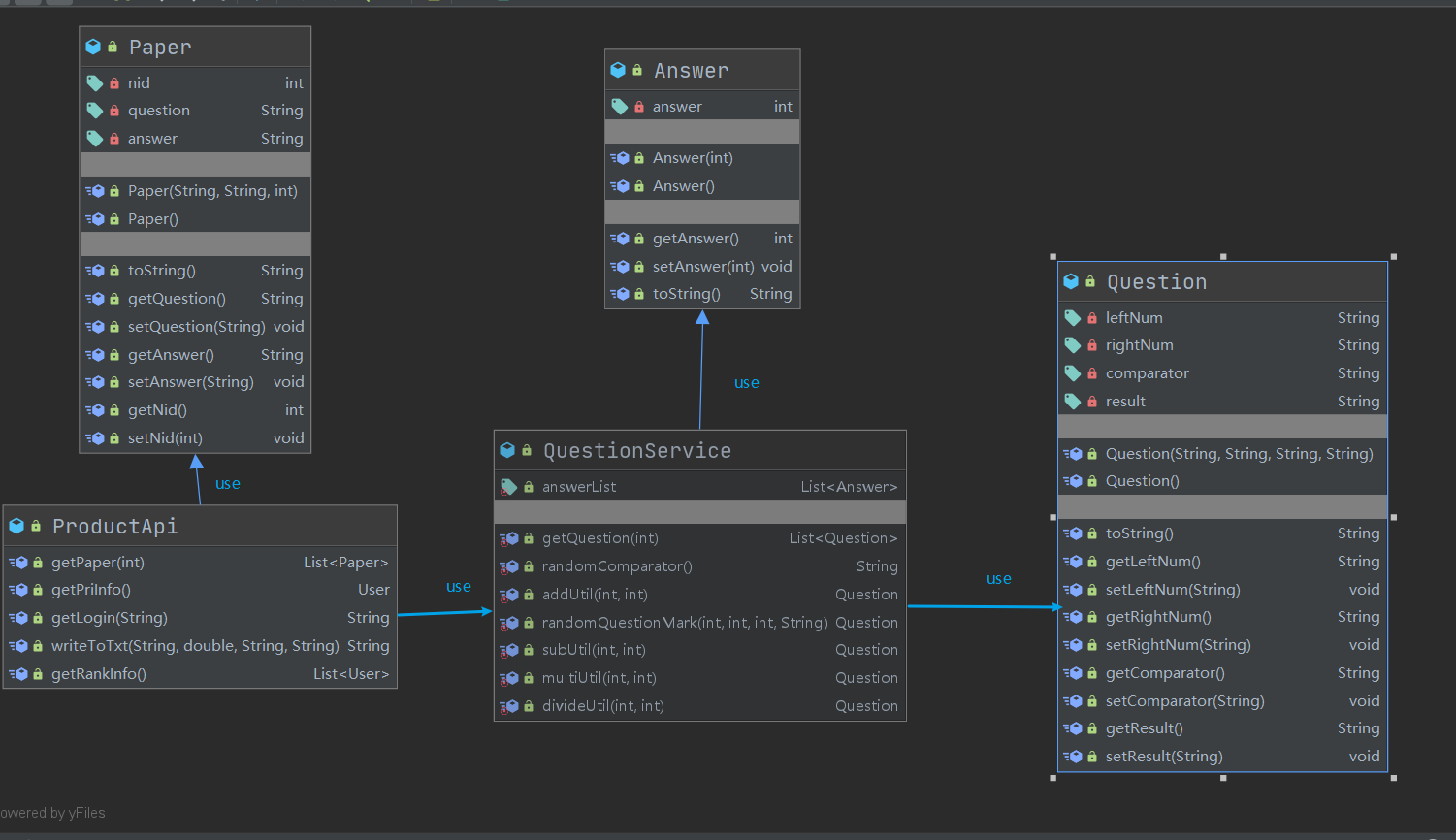
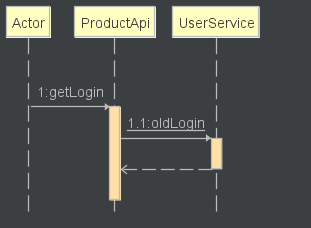


图2-3-2

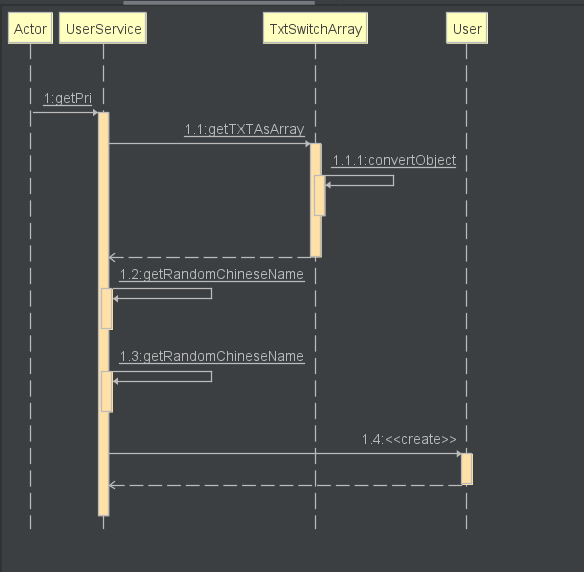
## 2.4 UML时序图（Sequence Diagram）

### 2.4.1.1老用户登录



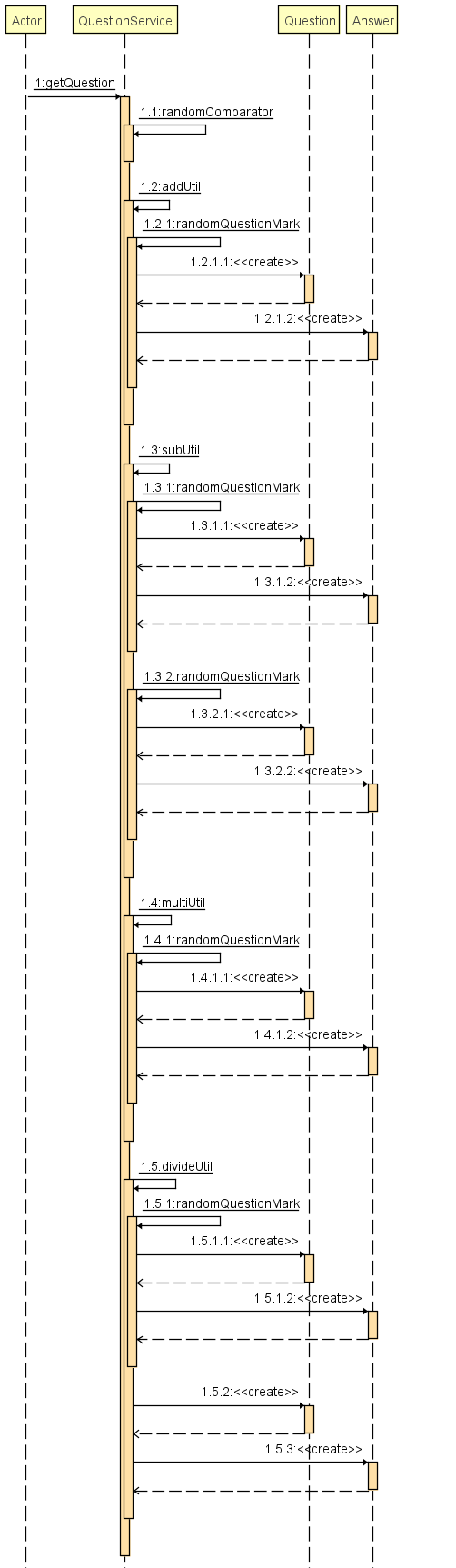
2-4-1-1

### 2.4.1.2游客登录



2-4-1-2

### 2.4.2 出题模块

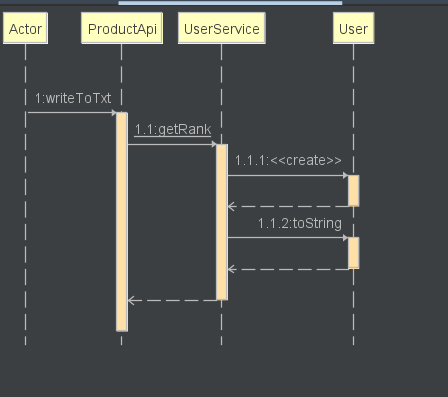


2-4-2

### 2.4.3 答题模块

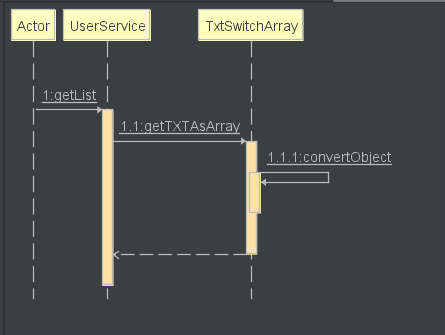
因为答题实则是将出的题渲染至界面上，用户填写答案，不涉及程序内部运行

### 2.4.4 显示答题详情模块



2-4-4

2.4.5 排行榜模块



2-4-5

## 2.5 UML活动图（Activity Diagram）

### 2.5.1.1 游客登录

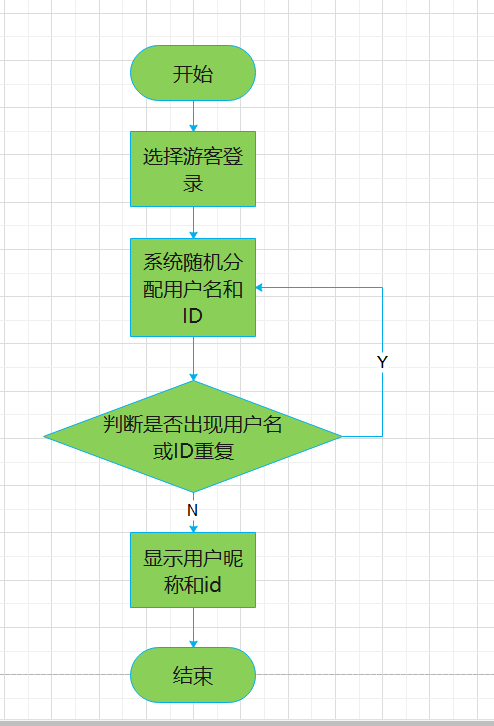
游客登录系统随机分配用户名和id,并显示给用户。

图2-5-1-1 游客登录活动图

### 2.5.1.2 用户登陆

用户登录操作需要验证昵称或id，只有当昵称或id存在即用户存在的情况下才能成功登录系统，否则会提示错误信息，具体流程如图2-6所示。

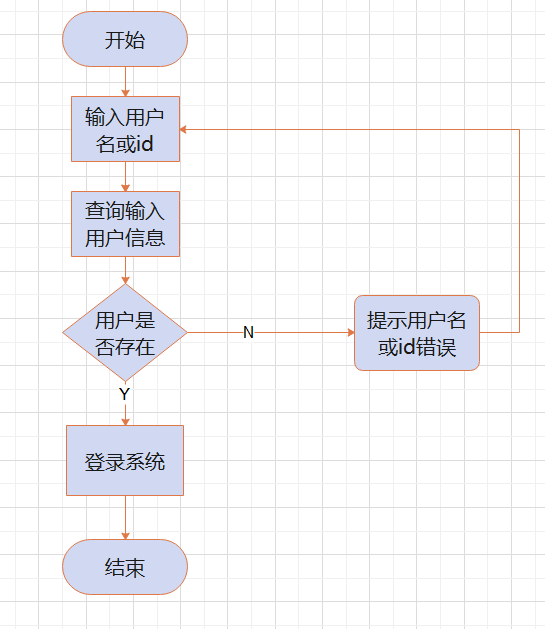


图2-5-1-2 用户登录流程图

### 2.5.2 出题模块

在出题模块下，用户需输入题目的数量，确认后，系统随机分配对应数量的数学题

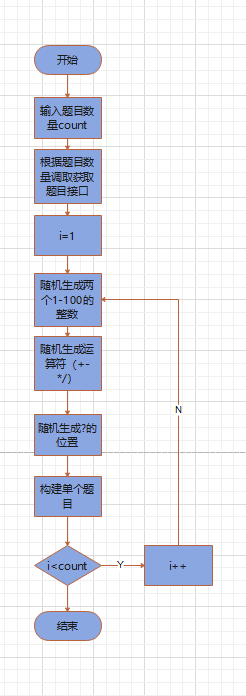


图2-5-2 出题功能活动图

### 2.5.3 答题模块

在答题模块下用户需要填入自己的答案，点击提交按钮即可，如需要大批次修改可以选择清空答题文本框

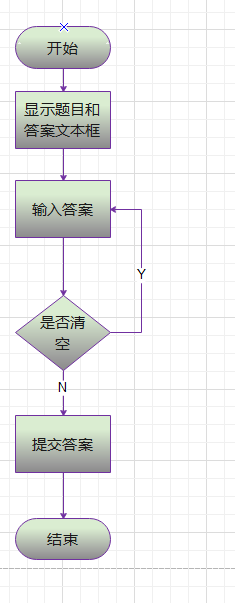


图2-5-3 答题功能活动图

### 2.5.4 显示答题详情模块

在显示答题详情模块中，系统需要获取用户所答的题目，和用户所填写的答案，并根据后端接口获取正确答案，根据百分制计算用户的分数，同时记录下用户答题的时间，后端将文件写入

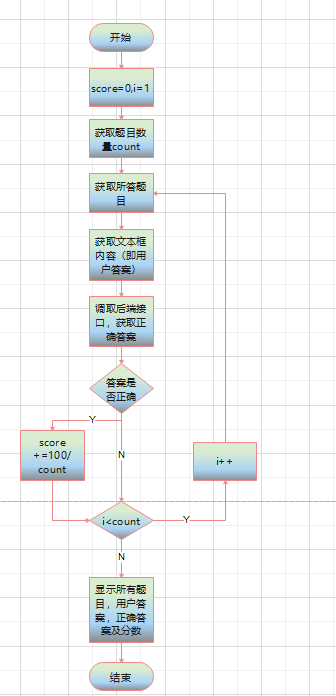


图2-5-4 显示答题详情活动图

### 2.5.5 排行榜模块

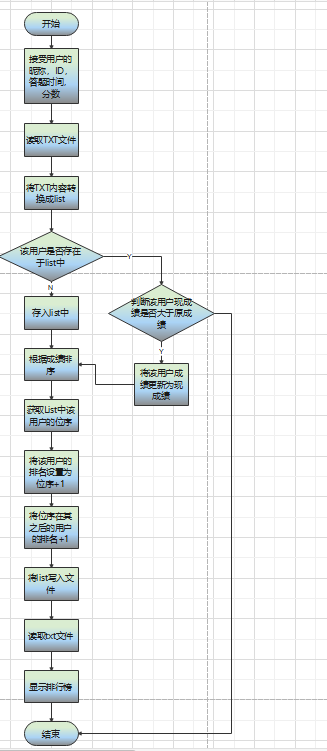
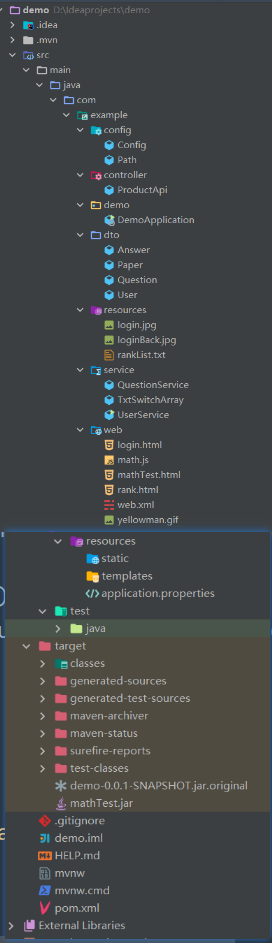


图2-5-5 排行榜功能活动图

# 3 系统实现

## 3.1 项目结构



## 3.2 配置文件

在项目下新建配置包，名为config

### 3.2.1 Config.java文件

该配置文件主要是解决CORS跨域跨端口问题

package com.example.config;  
  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.CorsRegistry;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
  
@Configuration  
public class Config implements WebMvcConfigurer {  
 public void addCorsMappings(CorsRegistry registry) {  
 registry.addMapping("/\*\*")  
 .allowedOrigins("\*")  
 .allowedMethods("GET","HEAD","POST","PUT","DELETE","OPTIONS")  
 .allowCredentials(true)  
 .maxAge(3600)  
 .allowedHeaders("\*");  
 }  
}

### 3.2.2 Path.java文件

将路径存放于文件中，便于今后改动文件路径

package com.example.config;  
  
public class Path {  
 public static String *pathName*="src/main/java/com/example/resources/";  
}

### 3.2.3 pom.xml文件

全局配置文件，主要用于配置spring-boot web的相关内容

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.3.0.RELEASE</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>demo</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>demo</name>  
 <description>Demo project for Spring Boot</description>  
  
 <properties>  
 <java.version>1.8</java.version>  
 </properties>  
  
 <dependencies>  
 <dependency> <!--建立spring-boot依赖-->  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 <exclusions>  
 <exclusion>  
 <groupId>org.junit.vintage</groupId>  
 <artifactId>junit-vintage-engine</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

3.2.4 web.xml文件

该文件主要为servlet相关的配置

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns:javaee="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">  
 <display-name></display-name>  
 <context-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath:applicationContext.xml</param-value>  
 </context-param>  
 <listener>  
 <listener-class>  
 org.springframework.web.context.ContextLoaderListener  
 </listener-class>  
 </listener>  
 <servlet>  
 <servlet-name>jersey-serlvet</servlet-name>  
 <servlet-class>  
 org.glassfish.jersey.servlet.ServletContainer  
 </servlet-class>  
 <init-param>  
 <param-name>javax.ws.rs.Application</param-name>  
 <param-value>com.kendy.filter.RestJaxRsApplication</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>jersey-serlvet</servlet-name>  
 <url-pattern>/\*</url-pattern>  
 </servlet-mapping>  
</web-app>

## 3.3 DTO类

### 3.3.1 DTO类User.java

对记录存储用户信息起着重要作用

package com.example.dto;  
  
import com.fasterxml.jackson.annotation.JsonFormat;  
  
public class User {  
 @JsonFormat //使接口转换成Json形式  
 private String rank;  
 @JsonFormat  
 private String userName;  
 @JsonFormat  
 private String id;  
 @JsonFormat  
 private double grade;  
 @JsonFormat  
 private String testTime;  
  
 public User() {  
 super();  
 }  
  
 @Override  
 public String toString() {  
 return rank+"\t"+userName+"\t"+id+"\t"+grade+"\t"+testTime+"\n";  
 }  
  
 public User(String userName, String id) {  
 this.userName = userName;  
 this.id = id;  
 }  
  
 public User(String rank, String userName, String id, double grade, String testTime) {  
 this.rank = rank;  
 this.userName = userName;  
 this.id = id;  
 this.grade = grade;  
 this.testTime = testTime;  
 }  
  
 public String getRank() {  
 return rank;  
 }  
  
 public void setRank(String rank) {  
 this.rank = rank;  
 }  
  
 public String getUserName() {  
 return userName;  
 }  
  
 public void setUserName(String userName) {  
 this.userName = userName;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public double getGrade() {  
 return grade;  
 }  
  
 public void setGrade(double grade) {  
 this.grade = grade;  
 }  
  
 public String getTestTime() {  
 return testTime;  
 }  
  
 public void setTestTime(String testTime) {  
 this.testTime = testTime;  
 }  
}

### 3.3.2 DTO实体类Paper.java

对试卷的生成和呈现起着重要作用

package com.example.dto;  
  
import com.fasterxml.jackson.annotation.JsonFormat;  
  
public class Paper {  
 @JsonFormat  
 private int nid;  
 @JsonFormat  
 private String question;  
 @JsonFormat  
 private String answer;  
  
 public Paper(String question, String answer,int nid) {  
 this.question = question;  
 this.answer = answer;  
 this.nid=nid;  
 }  
  
 public Paper() {  
 }  
  
 @Override  
 public String toString() {  
 return nid+" "+question+" "+answer;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
  
 public String getAnswer() {  
 return answer;  
 }  
  
 public void setAnswer(String answer) {  
 this.answer = answer;  
 }  
  
 public int getNid() {  
 return nid;  
 }  
  
 public void setNid(int nid) {  
 this.nid = nid;  
 }  
}

### 3.3.3 DTO实体类Answer.java

对答案的设置与显示起着重要作用

package com.example.dto;  
  
public class Answer {  
 private int answer;  
  
 public Answer(int result) {  
 this.answer = result;  
 }  
  
 public Answer() {  
 }  
  
 public int getAnswer() {  
 return answer;  
 }  
  
 public void setAnswer(int result) {  
 this.answer = result;  
 }  
  
 @Override  
 public String toString() {  
 return String.*valueOf*(answer) ;  
 }  
}

### 3.3.4 DTO实体类question.java

对题目的设置与显示起着重要作用

package com.example.dto;  
  
import com.example.service.QuestionService;  
  
public class Question {  
 private String leftNum;  
 private String rightNum;  
 private String comparator;  
 private String result;  
  
 public Question(String leftNum, String rightNum, String comparator, String result) {  
 this.leftNum = leftNum;  
 this.rightNum = rightNum;  
 this.comparator = comparator;  
 this.result = result;  
 }  
  
 public Question() {  
 }  
  
 @Override  
 public String toString() {  
 return leftNum+comparator+rightNum+"="+result;  
 }  
  
 public String getLeftNum() {  
 return leftNum;  
 }  
  
 public void setLeftNum(String leftNum) {  
 this.leftNum = leftNum;  
 }  
  
 public String getRightNum() {  
 return rightNum;  
 }  
  
 public void setRightNum(String rightNum) {  
 this.rightNum = rightNum;  
 }  
  
 public String getComparator() {  
 return comparator;  
 }  
  
 public void setComparator(String comparator) {  
 this.comparator = comparator;  
 }  
  
 public String getResult() {  
 return result;  
 }  
  
 public void setResult(String result) {  
 this.result = result;  
 }  
}

## 3.4 服务类

### 3.4.1 服务类UserService.java

描述涉及到用户服务的相应业务逻辑，如随机生成用户昵称和ID，校验用户输入的昵称或ID是否正确，将用户写入排行榜文件等

package com.example.service;  
  
import com.example.config.Path;  
import com.example.dto.User;  
  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileWriter;  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class UserService {  
 public static User getPri() throws Exception{ //随机生成用户名昵称函数  
 List<Object> list1 = TxtSwitchArray.*getTXTAsArray*("src/main/java/com/example/resources/rankList.txt", "com.example.dto.User");  
 List<User> userList=*getList*();  
 list1=new ArrayList<>();  
 System.*out*.println("游客登录");  
 String id=*getRandomId*();  
 String userName = *getRandomChineseName*(6);  
 for (int i=0;i<userList.size();i++){  
 if (userList.get(i).getId().equals(id)||userList.get(i).getUserName().equals(userName)){  
 id=*getRandomId*();  
 userName = *getRandomChineseName*(6);  
 }  
 }  
 User user = new User(userName,id);  
 return user;  
 /\* System.out.println(userList.toString());  
 for (User user1:userList){  
 Object object = (Object) user1;  
 list1.add(object);  
 }  
 System.out.println(list1.toString());\*/  
 //TxtSwitchArray.arrayToTXT(list1,"src/main/java/com/example/resources/rankList.txt");  
 }  
 public static List<User> getList(){ //获取排行榜信息函数  
 List<Object> list1 = TxtSwitchArray.*getTXTAsArray*("src/main/java/com/example/resources/rankList.txt", "com.example.dto.User");  
 List<User> userList=new ArrayList<>();  
  
 for (Object object:list1){  
 User user = (User)object;  
 userList.add(user);  
 }  
 return userList;  
 }  
  
 public static String oldLogin(String info){ //判断用户登录输入的信息是否正确函数  
 List<User> userList = *getList*();  
 for (int i=0;i<userList.size();i++){  
 if (userList.get(i).getId().equals(info)||userList.get(i).getUserName().equals(info)){  
 return "成功";  
 }  
 }  
 return "用户名或id错误";  
 }  
 public static String getRandomChineseName(int len) { //获取随机中文昵称函数  
 String randomName = "";  
 for (int i = 0; i < len; i++) {  
 String str = null;  
 int hightPos, lowPos; // 定义高低位  
 Random random = new Random();  
 hightPos = (176 + Math.*abs*(random.nextInt(39))); // 获取高位值  
 lowPos = (161 + Math.*abs*(random.nextInt(93))); // 获取低位值  
 byte[] b = new byte[2];  
 b[0] = (new Integer(hightPos).byteValue());  
 b[1] = (new Integer(lowPos).byteValue());  
 try {  
 str = new String(b, "GBK"); // 转成中文  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 randomName += str;  
 }  
 return randomName;  
 }  
 public static String getRandomId(){ //获取随机ID函数  
 UUID uuid = UUID.*randomUUID*();  
 String uuidStr=uuid.toString();  
 uuidStr = uuidStr.replace("-","");  
 return uuidStr;  
 }  
 /\*将信息排序写入文件函数\*/  
 public static List<User> getRank(String duringTime,double grade,String userName,String id) throws Exception{  
 List<User> userList =new ArrayList<>();  
 userList=*getList*();  
 User user = new User("0",userName,id,grade,duringTime);  
 int index=userList.size();  
 int flag=0;  
 for (int i=0;i<userList.size();i++){  
 if (userList.get(i).getUserName().equals(userName)&& userList.get(i).getGrade()>=grade){  
 return userList;  
 }  
 if (userList.get(i).getUserName().equals(userName)&&userList.get(i).getGrade()<grade){  
 userList.get(i).setGrade(grade);  
 flag=1;  
 }  
 }  
 if (flag!=1){userList.add(user);}  
 File file = new File(Path.*pathName*+"rankList.txt");  
 FileWriter fw =new FileWriter(file);  
 BufferedWriter bw = new BufferedWriter(fw);  
 String str="rank\tuserName\tid\tgrade\ttestTime\n";  
 String rank="0";  
 userList = userList.stream().sorted(Comparator.*comparing*(User::getGrade).reversed()).collect(Collectors.*toList*());  
 System.*out*.println(userList);  
 for (int i = 0; i < userList.size(); i++) {  
 if (userList.get(i).getUserName().equals(userName)) {  
 rank = String.*valueOf*(i + 1);  
 userList.get(i).setRank(rank);  
 }  
 if (userList.get(i).getGrade() < grade) {  
 userList.get(i).setRank(String.*valueOf*(Integer.*parseInt*(userList.get(i).getRank()) + 1));  
 }  
 str += userList.get(i).toString();  
 }  
 bw.write(str);  
  
 bw.close();  
  
 return userList;  
 }  
 /\*测试函数\*/  
 public static void main(String[] args) throws Exception{  
 *getRank*("0小时0分钟4秒333毫秒",50,"恢卸趴乌仗锚","8e438837e5f84ae6be017959927141af");  
 }  
}

### 3.4.2 服务类文件QuestionService.java

该文件主要是实现随机出题，设置对应的答案功能

package com.example.service;  
  
import com.example.dto.Answer;  
import com.example.dto.Question;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Random;  
  
public class QuestionService {  
 public static List<Answer> *answerList*=new ArrayList<>();  
  
  
 public static List<Question> getQuestion(int count) { //获取问题函数  
 List<Question> questionList = new ArrayList<>();  
 *answerList*=new ArrayList<>();  
 Random rand = new Random();  
 for (int i=0;i<count;i++) {  
 Question question = null;  
 int firstNum = rand.nextInt(99) + 1;  
 int secondNum = rand.nextInt(99) + 1;  
 String comparator = *randomComparator*();  
 switch (comparator) {  
 case "+":  
 question=*addUtil*(firstNum,secondNum);  
 if (question==null){  
 i=i-1;}  
 break;  
 case "-":  
 question=*subUtil*(firstNum,secondNum);  
 break;  
 case "\*":  
 question=*multiUtil*(firstNum,secondNum);  
 if (question==null){  
 i=i-1;}  
 break;  
 case "/":  
 question=*divideUtil*(firstNum,secondNum);  
 break;  
 }  
 if (question!=null) {  
 questionList.add(question);  
 }  
 }  
 return questionList;  
 }  
  
 public static String randomComparator() { //获取随机运算符  
 Random rand = new Random();  
 String comparator = "";  
 int index = rand.nextInt(4);  
 switch (index) {  
 case 0:  
 comparator = "+";  
 break;  
 case 1:  
 comparator = "-";  
 break;  
 case 2:  
 comparator = "\*";  
 break;  
 case 3:  
 comparator = "/";  
 break;  
 }  
 return comparator;  
 }  
 public static Question addUtil(int firstNum,int secondNum){ //加法控制在两位数以内  
 Question question = null;  
 int result = firstNum+secondNum;  
 if (result<100){  
 question = *randomQuestionMark*(firstNum,secondNum,result,"+");  
 }  
 return question;  
 }  
 /\*问号位置随机函数\*/  
 public static Question randomQuestionMark(int firstNum,int secondNum,int result,String comparator){  
 Question question = new Question();  
 Random random = new Random();  
 int index = random.nextInt(3);  
 Answer answer = new Answer();  
 switch (index){  
 case 0:  
 answer.setAnswer(result);  
 question.setResult("?");  
 question.setLeftNum(String.*valueOf*(firstNum));  
 question.setRightNum(String.*valueOf*(secondNum));  
 question.setComparator(comparator);  
 break;  
 case 1:  
 answer.setAnswer(secondNum);  
 question.setRightNum("?");  
 question.setLeftNum(String.*valueOf*(firstNum));  
 question.setComparator(comparator);  
 question.setResult(String.*valueOf*(result));  
 break;  
 case 2:  
 answer.setAnswer(firstNum);  
 question.setLeftNum("?");  
 question.setRightNum(String.*valueOf*(secondNum));  
 question.setComparator(comparator);  
 question.setResult(String.*valueOf*(result));  
 break;  
 }  
 *answerList*.add(answer);  
 return question;  
 }  
 /\*控制减法在两位数以内函数\*/  
 public static Question subUtil(int firstNum,int secondNum){  
 Question question = null;  
 int result =Math.*abs*(firstNum-secondNum);  
 if (firstNum<secondNum){  
 question=*randomQuestionMark*(secondNum,firstNum,result,"-");  
 }  
 else{  
 question = *randomQuestionMark*(firstNum,secondNum,result,"-");  
 }  
 return question;  
 }  
 /\*控制乘法在两位数以内函数\*/  
 public static Question multiUtil(int firstNum,int secondNum){  
 Question question = null;  
 int result=firstNum\*secondNum;  
 if(result<100){  
 question=*randomQuestionMark*(firstNum,secondNum,result,"\*");  
 }  
 return question;  
 }  
 /\*控制除法在两位数以内函数\*/  
 public static Question divideUtil(int firstNum,int secondNum){  
 Question question = null;  
 int flag;  
 int result;  
 if (firstNum<secondNum){  
 flag = firstNum;  
 firstNum=secondNum;  
 secondNum=flag;  
 }  
 result=firstNum/secondNum;  
 int rest = firstNum%secondNum;  
 if (rest==0){  
 *randomQuestionMark*(firstNum,secondNum,result,"/");  
 }  
 else{  
 question=new Question();  
 Answer answer = new Answer();  
 question.setResult("?");  
 question.setLeftNum(String.*valueOf*(firstNum));  
 question.setRightNum(String.*valueOf*(secondNum));  
 question.setComparator("/");  
 answer.setAnswer(result);  
 *answerList*.add(answer);  
 }  
 return question;  
 }  
  
  
}

3.4.3 服务类文件TxtSwitchArray.java

该文件主要实现txt文件与list的转换，利用类反射机制达到迅速且通用

package com.example.service;  
  
import java.io.\*;  
import java.lang.reflect.Field;  
import java.lang.reflect.Method;  
import java.util.ArrayList;  
import java.util.List;  
  
public class TxtSwitchArray {  
 /\*将txt转换成list函数\*/  
 public static List<Object> getTXTAsArray(String path, String className) {  
 List<Object> resultList = new ArrayList<Object>();  
 File file = new File(path);  
 BufferedReader br = null;  
 try {  
 br = new BufferedReader(new FileReader(file));  
 String title = br.readLine();  
 String titles[] = title.split("\\s+");  
 String aLine = "";  
 Class aClass = Class.*forName*(className);  
 while ((aLine = br.readLine()) != null) {  
 String datas[] = aLine.split("\\s+");  
 Object obj = aClass.newInstance();  
 for (int i = 0; i < titles.length; i++) {  
 String fieldName = titles[i];  
 Field field = aClass.getDeclaredField(fieldName);  
 String getMethodName = "set" + fieldName.substring(0, 1).toUpperCase() + fieldName.substring(1);  
 Method method = aClass.getMethod(getMethodName, field.getType());  
 method.invoke(obj, *convertObject*(datas[i], field.getType().getSimpleName()));  
 }  
 resultList.add(obj);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 if (br != null) {  
 try {  
 br.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 return resultList;  
 }  
 /\*类型转换函数\*/  
 public static Object convertObject(String data, String type) { //类型转换函数  
 Object obj = null;  
 switch (type) {  
 case "String":  
 obj = new String(data);  
 break;  
 case "double":  
 case "Double":  
 obj = new Double(data);  
 break;  
 case "float":  
 case "Float":  
 obj = new Float(data);  
 break;  
 case "int":  
 case "Integer":  
 obj = new Integer(data);  
 break;  
 //后面可以扩充转换为其他数据类型  
 }  
 return obj;  
 }  
 /\*将list写入txt文件函数\*/  
 public static void arrayToTXT(List<Object> list, String path) { //list转TXT  
 File file = new File(path);  
 FileWriter fw = null;  
 BufferedWriter bw = null;  
 try {  
 if (list != null && list.size() != 0) {  
 fw = new FileWriter(file);  
 bw = new BufferedWriter(fw);  
 String str = "";  
 String keyValue = "";  
 Object obj = list.get(0);  
 Field[] fields = obj.getClass().getDeclaredFields();  
 for (int j = 0; j < fields.length; j++) {  
 str += fields[j].getName() + "\t";  
 }  
 str += "\n";  
 bw.write(str);  
 for (int i = 0; i < list.size(); i++) {  
  
 Object obj1 = list.get(i);  
 for (int j = 0; j < fields.length; j++) {  
 fields[j].setAccessible(true);  
 keyValue += fields[j].get(obj1).toString() + "\t";  
 }  
 keyValue += "\n";  
 }  
 bw.write(keyValue);  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 if (bw != null) {  
 try {  
 bw.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 if (fw != null) {  
 try {  
 fw.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
 }  
  
}

## 3.5 接口控制文件ProductApi.java

该文件主要用于前端调用不同的接口，接受get，post等请求

package com.example.controller;  
  
import com.example.dto.\*;  
import com.example.service.QuestionService;  
import com.example.service.UserService;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@RestController  
public class ProductApi {  
 @GetMapping("/PrimarySchoolMathTest")  
 public List<Paper> getPaper(int count){  
 List<Paper> paperList = new ArrayList<>();  
 List<Question> questionList = new ArrayList<>();  
 List<Answer> answers = new ArrayList<>();  
 questionList = QuestionService.*getQuestion*(count);  
 answers = QuestionService.*answerList*;  
 for (int i=0;i<questionList.size();i++){  
 Paper paper = new Paper();  
 paper.setNid(i);  
 paper.setQuestion(questionList.get(i).toString());  
 paper.setAnswer(answers.get(i).toString());  
 paperList.add(paper);  
 }  
 return paperList;  
 }  
 @GetMapping("/randomUserInfo")  
 public User getPriInfo() throws Exception{  
 return UserService.*getPri*();  
 }  
 @GetMapping("/login")  
 public String getLogin(String loginInfo){  
 return UserService.*oldLogin*(loginInfo);  
 }  
 @GetMapping("/write")  
 public String writeToTxt(String duringTime,double grade,String userName,String id){  
 try {  
 UserService.*getRank*(duringTime,grade,userName,id);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return "已成功写入文件";  
 }  
 @GetMapping("/rank")  
 public List<User> getRankInfo(){  
 List<User> userList = null;  
 userList= UserService.*getList*();  
 return userList;  
 }  
}

## 3.6 用户现实层web

### 3.6.1 mathTest.html

显示网页，将后端的数据渲染至web服务器，采用vue+axios方式实现前后端分离

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>小学生算数测验</title>  
</head>  
<style>  
 \*{  
 margin: 0;  
 padding: 0;  
 }  
 .login{  
 position: absolute;  
 top: 30%;  
 left: 50%;  
 }  
 .loginButton{  
 border: solid 3px cornflowerblue;  
 border-radius: 5px;  
 width: 160px;  
 height: 40px;  
 margin-top: 10px;  
 background-image: linear-gradient(to top, #a8edea 0%, #fed6e3 100%);  
 }  
 #loginText{  
 margin-top: 90px;  
 border: solid 3px #333456;  
 border-radius: 5px;  
 height: 30px;  
 }  
 .getTest{  
 position: absolute;  
 top: 20%;  
 left: 40%;  
 }  
 .getTest input{  
 height: 50px;  
 width: 200px;  
 text-align: center;  
 border-radius: 10px;  
 border: solid 5px #569801;  
 }  
 .userInfo input{  
 border: solid 4px #132920;  
 border-radius: 50%;  
 width: 80px;  
 height: 80px;  
 }  
 .doTest{  
 margin-top: 200px;  
 margin-left: 90px;  
 }  
 .doTest p{  
 color: aquamarine;  
 font-size: 15px;  
 padding-top: 10px;  
 }  
 .testButton{  
 margin-left: 90px;  
 width: 50px;  
 height: 30px;  
 border: solid 2px azure;  
 border-radius: 3px;  
 margin-top: 50px;  
 background-image: linear-gradient(to right, #eea2a2 0%, #bbc1bf 19%, #57c6e1 42%, #b49fda 79%, #7ac5d8 100%);  
 }  
 .testResult h3{  
 margin-left: 50px;  
 color: darkblue;  
 }  
 .answers{  
 color: darkblue;  
 }  
 .testResult input{  
 width: 180px;  
 height: 30px;  
 background-image: linear-gradient(to top, #a8edea 0%, #fed6e3 100%);  
 }  
 .jump{  
 position: relative;  
 font: normal 500 16px 'Varela Round', sans-serif;  
 color: goldenrod;  
 animation: bounce 0.75s cubic-bezier(0.05, 0, 0.2, 1) infinite alternate;  
 margin-left: 53%;  
 margin-top: 2%;  
 }  
 .letter {  
 animation: bounce 0.75s cubic-bezier(0.05, 0, 0.2, 1) infinite alternate;  
 display: inline-block;  
 transform: translate3d(0, 0, 0);  
 margin-top: 0.5em;  
 text-shadow: rgba(255, 255, 255, 0.4) 0 0 0.05em;  
 font: normal 500 32px 'Varela Round', sans-serif;  
 }  
 #message{  
 color :#61992f;  
 border: 4px solid #c4c4c4;  
 background:#ddd;  
 -moz-border-radius:5px;  
 -webkit-border-radius:5px;  
 border-radius:5px;  
 min-height:1em;  
 padding:5px;  
 top:-45px;  
 position:absolute;  
 text-align:center;  
 width:auto **!important**;  
 color:#61992f;  
 z-index:10000;  
 -moz-box-shadow:0 0 15px #eeeeee;  
 -webkit-box-shadow:0 0 15px #eeeeee;  
  
 box-shadow:0 0 15px #eeeeee;  
 outline:none;  
 }  
 #spig {  
 display:block;  
  
  
 right:90px**!important**;  
 bottom: 400px;  
 position: absolute;  
 }  
 #mumu{  
 display: block;  
 width:409px;  
 height:200px;  
 cursor:move;  
 right:400px**!important**;  
 background:url(yellowman.gif) no-repeat;  
 }  
  
 @keyframes bounce {  
 0% {  
 top: 0;  
 text-shadow: rgba(255, 255, 255, 0.4) 0 0 0.05em;  
 }  
 100% {  
 top: -1em;  
 text-shadow: rgba(255, 255, 255, 0.9) 0 1em 0.35em;  
 }  
 }  
 .scroll{  
 position: absolute;  
 bottom: 0px;  
 }  
 .scroll\_span{  
 font-size: 17px;  
 color: #ffffff;  
 }  
</style>  
<body style="background: url('../resources/login.jpg');opacity:70%">  
<div id="app" >  
 <!--字节跳动开始-->  
 <div class="jump">  
 <h1 class="letter">算宝</h1>  
 </div>  
 <!--字节跳动结束-->  
 <!--滚动条开始-->  
 <marquee class="scroll" scrolldelay="4" scrollamount="10">  
 <span class="scroll\_span">本网站已限制外网IP，如正在使用网络代理或外网IP，请切换国内IP使用&nbsp;</span>  
 <span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span>  
 <span class="scroll\_span">第一次进入本站的用户请选择游客登录，如想再次使用同一账户，可记录下自己游客登录分配的昵称或id，下次登录时填写对应的昵称或ID选择用户登录即可</span>  
 </marquee>  
 <!--滚动条结束-->  
 <!--登录模块-->  
 <div class="login" v-show="login" >  
 <input type="button" value="游客登录" @click="jump" class="loginButton">  
 <br>  
 <input type="text" placeholder="输入用户名或id" id="loginText" >  
 <br>  
 <input type="button" value="用户登录" @click="oldLogin" class="loginButton">  
 <p>{{loginInfo}}</p>  
 <input type="button" value="查看排行榜" @click="jumpRank" class="loginButton"/>  
 </div>  
 <!--用户信息模块-->  
 <div class="userInfo" style="position: absolute;right: 20px">  
 <input type="button" @click="newLogin" value="个人信息" v-show="infoButton">  
 <p v-show="userDetail">你的个人信息<br>{{userName}} <br>{{id}}</p>  
 </div>  
 <!--出题模块-->  
 <div class="getTest" v-show="getButton">  
 <input type="text" placeholder="请输入题目数量" id="count">  
 <input type="button" value="获取题目" @click="getQuestion" >  
 </div>  
 <!--答题模块-->  
 <div class="doTest" v-show="testButton">  
 <form >  
 <p v-for="(item,i) in questions">{{i+1}}、{{item}}<input type="text" :id="'question\_'+i" style="position:absolute; left: 300px;" placeholder="你的答案"/></p>  
 <div class="tests" v-show="tests">  
 <input type="button" value="提交" @click="verifyAnswer" class="testButton" >  
 <input type="reset" value="清空" class="testButton" >  
 </div>  
 </form>  
 <!--答题结果模块-->  
 <div class="testResult" v-show="testResult">  
 <h3>你的分数为{{score}}!</h3>  
 <br>  
 <div v-for="(item,i) in testDetail" class="answers" > {{i+1}}、 {{item}}</div>  
 <br>  
 <input type="button" value="查看排行榜" @click="jumpRank">  
 </div>  
  
 </div>  
 <!--小黄人模块-->  
 <div id="spig" class="spig" >  
 <div id="message">……</div>  
 <div id="mumu" class="mumu"></div>  
 </div>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
 <script src="math.js"></script>  
  
 <script type="text/javascript">  
 var ***isindex*** = true;  
 var ***title*** = "";  
 var ***visitor*** = "这位同学";  
 </script>  
 <!--精灵结束-->  
</div>  
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>  
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>  
<script>  
 var ***app***=new Vue({  
 el:"#app",  
 data:{  
 userName:"",  
 id:"",  
 questions:[], /\*问题数组\*/  
 answers:[], /\*答案数组\*/  
 getButton:false, /\*显示出题按钮\*/  
 question:"", /\*单个问题\*/  
 answer:"", /\*单个答案\*/  
 result:"", /\*getQuestion传来的数据\*/  
 score:0, /\*得分\*/  
 testButton:false, /\*测试栏显示\*/  
 count:0, /\*题数\*/  
 testResult:false, /\*测试结束显示\*/  
 testDetail:[],  
 infoButton:false,  
 userDetail:false,  
 startTime:"",  
 endTime:"",  
 duringTime:"",  
 loginInfo:"",  
 login:true,  
 tests:false  
 },  
 methods:{  
 getQuestion:function(){  
 var that=this;  
 that.count= ***document***.getElementById("count").value;  
 axios.get("http://127.0.0.1:8080/PrimarySchoolMathTest?count="+that.count).then  
 (function(response) {  
 let backInfo = response.data;  
 if (backInfo.length == that.count) {  
 that.startTime = new ***Date***();  
 that.startTime.setTime(that.startTime.getTime());  
 ***console***.log(that.startTime)  
 that.result = response.data;  
 that.getButton = false;  
 that.testButton = true;  
 that.tests=true;  
 for (var i = 0; i < that.result.length; i++) {  
 that.question = that.result[i].question;  
  
 that.questions.push(that.question);  
 that.answers.push(that.result[i].answer);  
 }  
 }  
 if (that.userName == "") {  
 axios.get("http://127.0.0.1:8080/randomUserInfo").then  
 (function (response) {  
 that.infoButton = true;  
 that.userDetail = true;  
 that.userName = response.data.userName;  
 that.id = response.data.id;  
 }, function (err) {  
 ***console***.log(err);  
 })  
 }  
 })  
 },  
 verifyAnswer:function () {  
 var that=this;  
 that.testResult=true;  
 that.tests=false;  
 that.endTime=new ***Date***();  
 that.duringTime=that.endTime-that.startTime;  
 ***console***.log(that.duringTime)  
 var duringHour=***Math***.floor(that.duringTime/3600000);  
 var duringMin = ***Math***.floor(that.duringTime%3600000/60000);  
 var duringSec = ***Math***.floor(that.duringTime%3600000%60000/1000);  
 var duringBitSec=***Math***.floor(that.duringTime%3600000%60000%1000);  
 that.duringTime=duringHour+"小时"+duringMin+"分钟"+duringSec+"秒"+duringBitSec+"毫秒";  
 ***console***.log(that.duringTime)  
  
 that.endTime.setTime(that.endTime.getTime());  
 for (var i=0;i<that.answers.length;i++){  
 var res= ***document***.getElementById("question\_"+i).value;  
  
 if (res==that.answers[i]){  
 that.score+=(100.0/that.count)\*1.0;  
 }  
 that.testDetail.push(that.questions[i]+"\xa0 \xa0 \xa0\xa0\xa0\xa0你的答案:"+res+"\xa0 \xa0\xa0\xa0\xa0\xa0正确答案："+that.answers[i])  
  
 }  
 that.score=***Math***.round(that.score);  
  
 axios.get("http://127.0.0.1:8080/write?duringTime=" + that.duringTime +  
 "&grade=" + that.score + "&userName=" + that.userName + "&id=" + that.id).then  
 (function (response) {  
 ***console***.log(response.data)  
 }, function (err) {  
 ***console***.log(err);  
 })  
  
  
 },  
 newLogin:function(){  
 let that=this;  
 that.infoButton=true;  
 that.userDetail=true;  
 },  
 oldLogin: function () {  
 let that = this;  
 var info = ***document***.getElementById('loginText').value;  
 ***console***.log(info)  
 axios.get("http://127.0.0.1:8080/login?loginInfo="+info).then  
 (function (response) {  
 ***console***.log(response.data)  
 if (response.data != "用户名或id错误") {  
 that.getButton=true;  
 that.loginInfo="";  
 that.login=false;  
 that.infoButton=true;  
 that.userName=info;  
 axios.get("http://127.0.0.1:8080/rank").then  
 (function (response) {  
 var results=response.data;  
 ***console***.log(results);  
 for (var i=0;i<results.length;i++){  
 if (results[i].userName==that.userName){  
 that.id=results[i].id;  
 }  
 }  
 ***console***.log(that.id)  
 })  
 } else {  
 that.loginInfo = response.data;  
 that.userName=***document***.getElementById('loginText').value;  
  
 }  
 }, function (err) {  
 ***console***.log(err);  
 })  
 },jump: function () {  
 let that=this;  
 that.login=false;  
 that.getButton=true;  
 that.infoButton=true;  
 },  
 jumpRank:function () {  
 ***window***.location.href='rank.html'  
 }  
 }  
 })  
</script>  
</body>  
</html>

### 3.6.2 rank.html

该网页实现对后端txt文件在服务器端的渲染

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>rank</title>  
</head>  
<style>  
 .rankButton{  
 margin-left: 50%;  
 }  
 .info{  
 height: 700px;  
 overflow: auto;  
 }  
</style>  
<body>  
<div id="app">  
 <input type="button" value="点击查看排行榜信息" @click="getRank" class="rankButton">  
  
 <p>排名&nbsp;&nbsp; 昵称&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
 ID&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;成绩&nbsp;&nbsp;  
 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;用时</p>  
<div class="info">  
<p v-for="(item,i) in ranks">  
 {{item}}  
</div>  
</p>  
</div>  
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>  
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>  
<script>  
 var ***app***=new Vue({  
 el: "#app",  
 data:{  
 ranks:[]  
 },  
 methods:{  
 getRank:function () {  
 let that =this;  
 axios.get("http://www.liurui13.cn:8080/rank").then  
 (function (response) {  
 var result =response.data;  
 ***console***.log(response.data)  
 for (var i=0;i<result.length;i++){  
 that.ranks.push(result[i].rank+"\xa0\xa0\xa0\xa0"+  
 result[i].userName+"\xa0\xa0\xa0\xa0"+  
 result[i].id+"\xa0\xa0\xa0\xa0"+result[i].grade+"\xa0\xa0\xa0\xa0"+  
 result[i].testTime);  
 }  
 }, function (err) {  
 ***console***.log(err);  
 })  
 }  
 },  
  
 })  
</script>  
</body>  
</html>

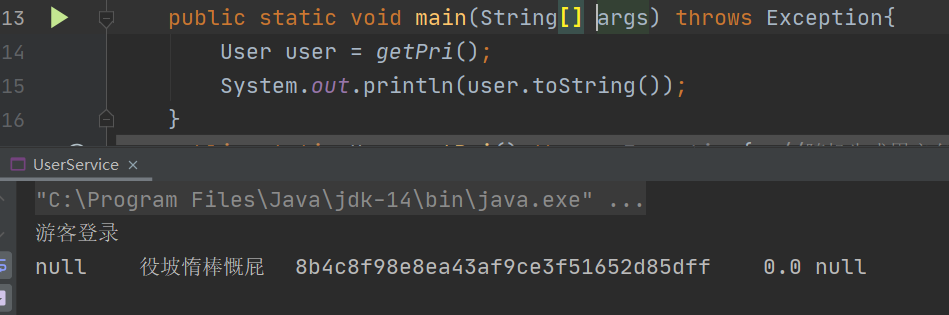
## 3.7 demo层DemoApplication.java

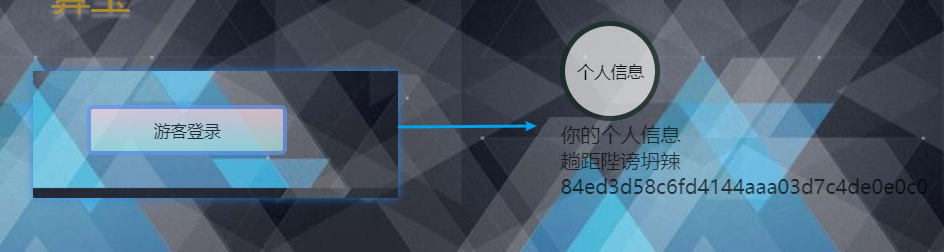
用于整个spring-boot项目的启动

package com.example.demo;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
  
@SpringBootApplication(scanBasePackages = "com.example")  
public class DemoApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(DemoApplication.class, args);  
 }  
}

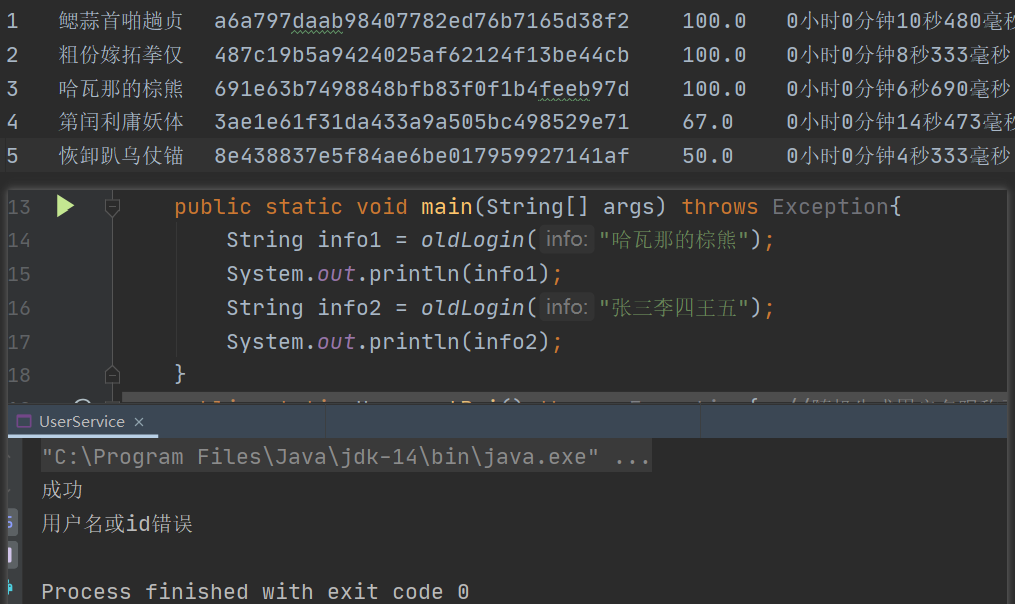
# 4 系统测试

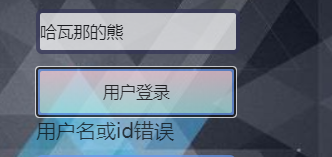
### 4.1.1游客登录功能

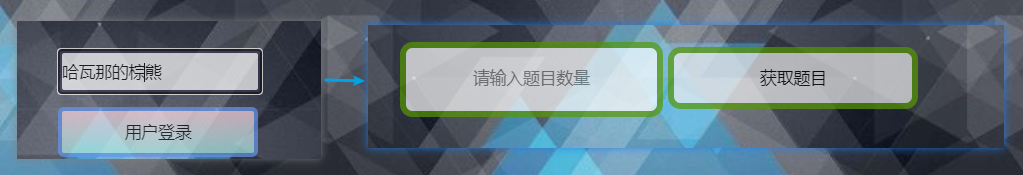




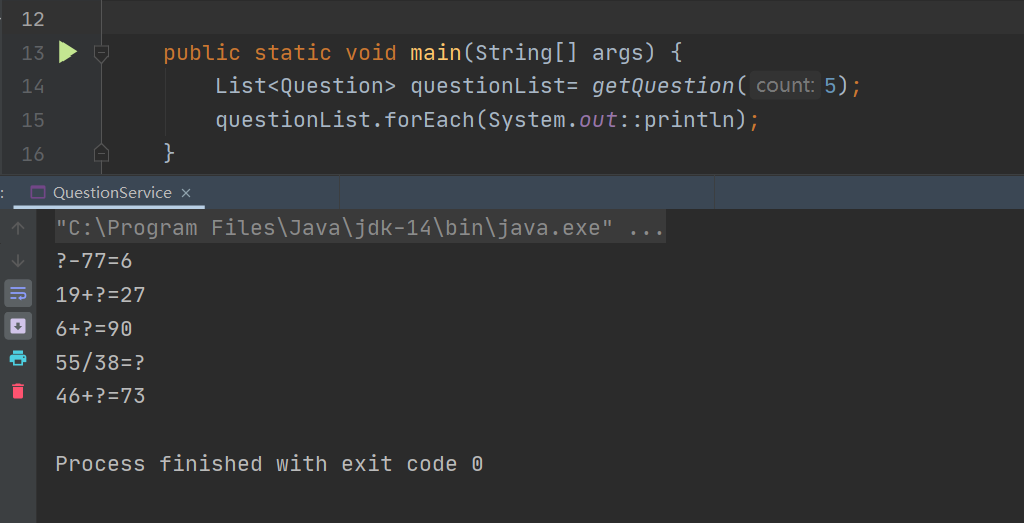
### 4.1.2用户登录功能

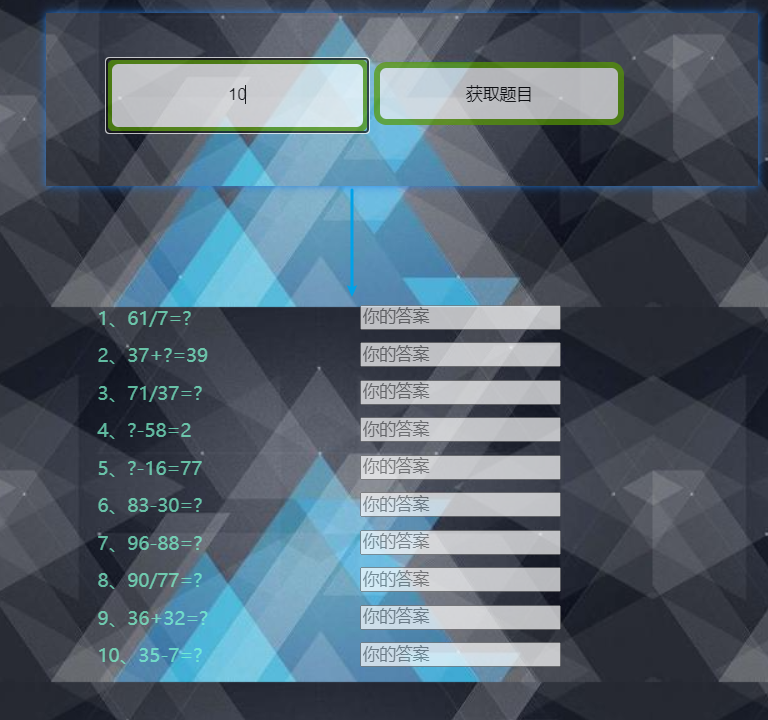
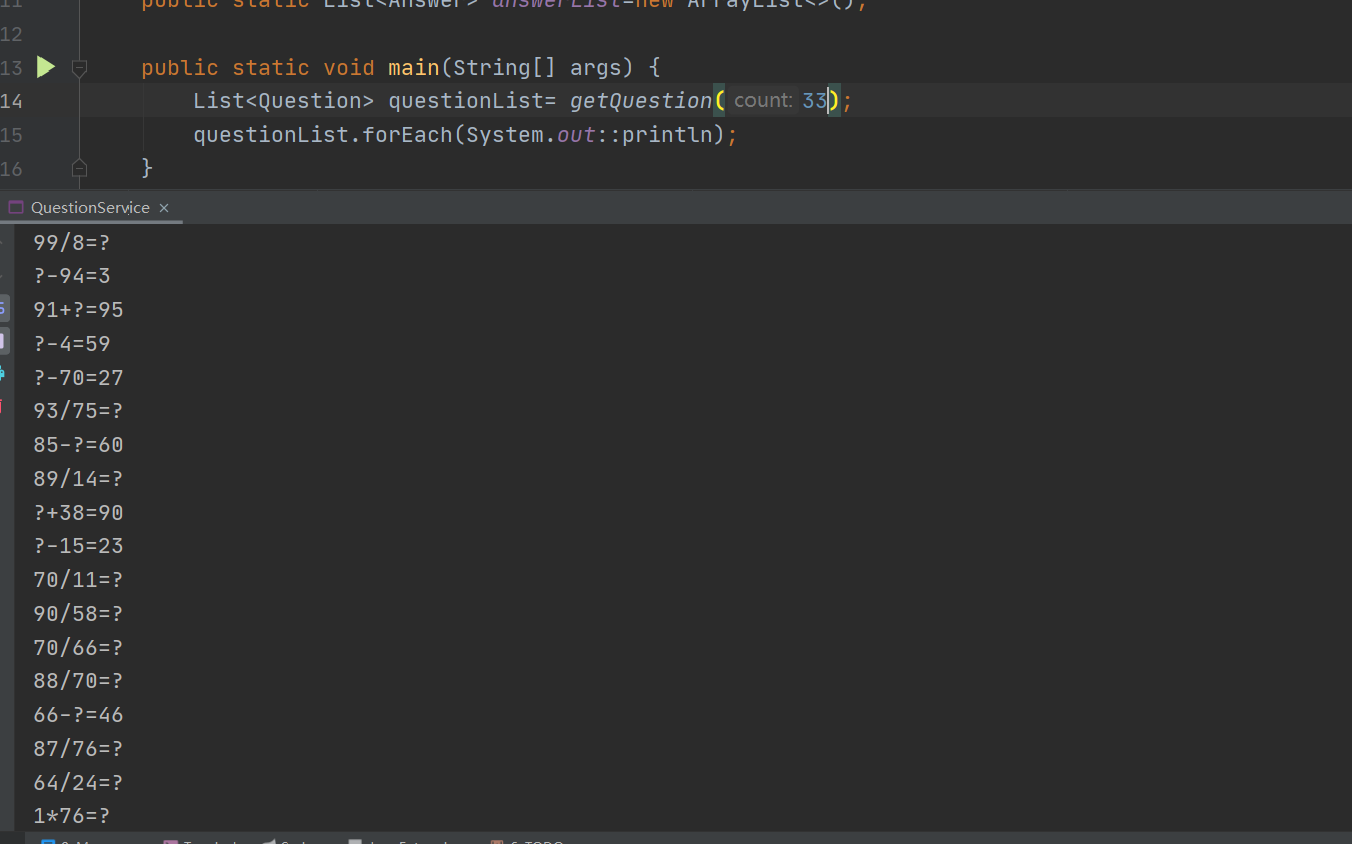






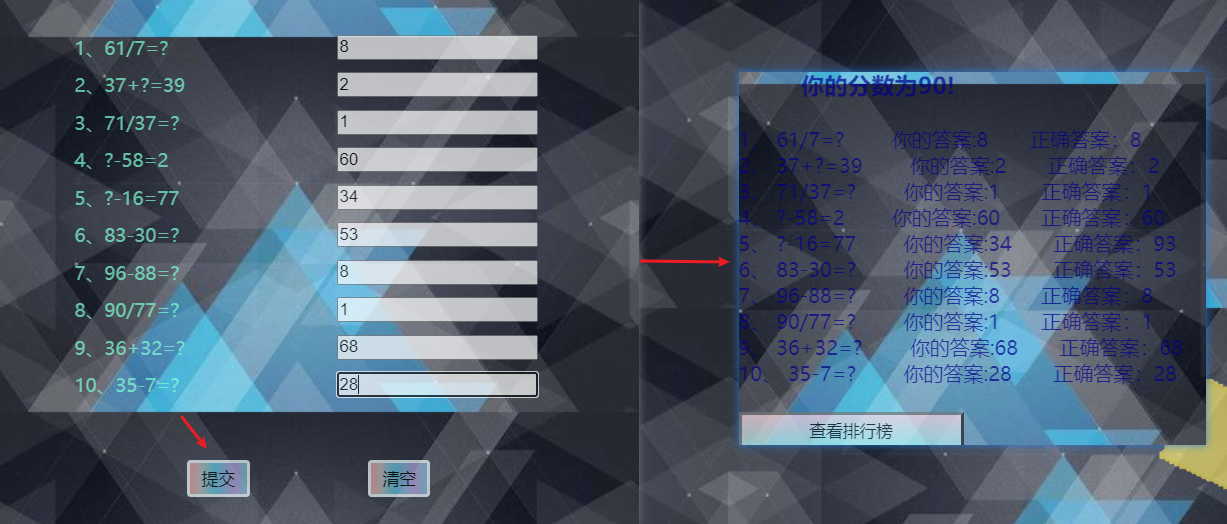
### 4.2出题功能



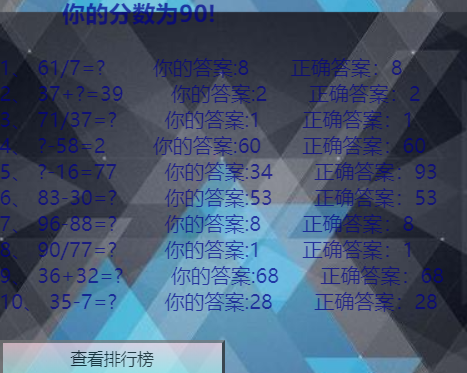


### 4.3答题功能

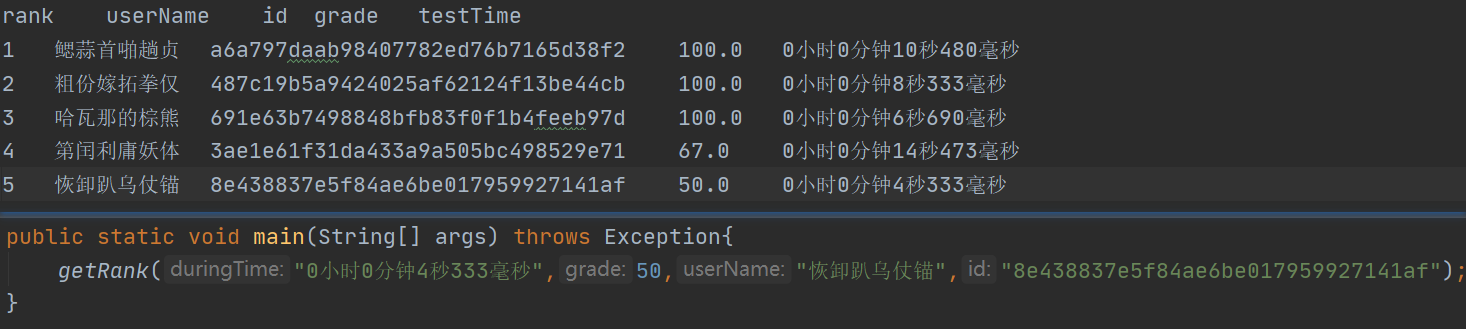


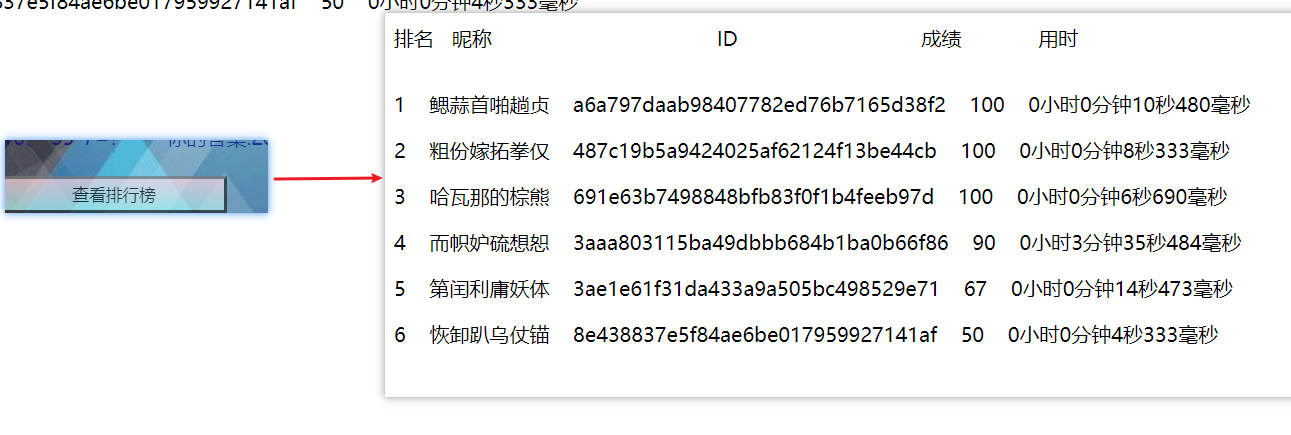


### 4.4显示题目详情功能



### 4.5显示排行榜功能





# 5 系统总结

### 5.1项目不足

在QuestionService里用到了全局变量，造成了线程安全问题，在写程序时未考虑到多线程问题，对cookie缓存机制没有做较好地处理，在读文件时可以考虑使用多线程模式，提高CPU利用率。为了后台可以控制项目的关闭在本项目中服务器端没有部署至tomcat，所以采用mvn spring-boot:run开启进程，导致当在xshell ssh 连接服务器时如果xshell进程关闭，整个maven项目也会关闭。Answer传向服务器采用的是明文传输等

### 5.2项目收获

经过了一学期java课程的学习，使自己在编程水平上有了很大的提高，在面对新知识时找到了较为适合自己的学习方法，这个项目是我独立完成的第一个完整java web项目，在完成过程中遇到了一些挫折和困难，但是正是在困难与挫折中让我有了更深的收获，

寒假时学习的前端知识和之前学习的服务器知识也真正有了用武之地，也让我意识到一个完整的项目需要的技术栈是很多的，所以需要不断健全自己，同时做到能通某一门，在完成项目的过程中同时需要自己的理论知识的辅助，如网络原理操作系统等，只有在基础知识深厚的情况下，遇到困难才不会忙手忙脚不知道从何做起。