武汉纺织大学

Java应用开发课程设计

**基于MyBatis的考试系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11802**

**姓 名： 郭子维**

**学 号： 1804240610**

**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 3](#_Toc43569901)

[1.1用户登录 3](#_Toc43569892)

[1.2主菜单 3](#_Toc43569893)

[1.3各项功能 3](#_Toc43569894)

[1.3.1 题库管理 3](#_Toc43569895)

[1.3.2 考试管理 3](#_Toc43569896)

[1.3.3 成绩管理 4](#_Toc43569897)

[1.3.4 成绩统计 4](#_Toc43569898)

[1.3.5 退出 4](#_Toc43569899)

[1.3.6 考试 4](#_Toc43569900)

[2 系统设计 4](#_Toc43569901)

[2.1用户用例图 4](#_Toc43569902)

[2.2 ER图 4](#_Toc43569903)

[2.3 UML活动图（Activity Diagram） 5](#_Toc43569904)

[2.3.1 登陆 5](#_Toc43569905)

[2.3.2 学生活动操作图 6](#_Toc43569906)

[2.3.3 老师活动操作图 6](#_Toc43569906)

[3 系统实现 7](#_Toc43569907)

[3.1 项目结构 7](#_Toc43569908)

[3.2 配置文件 7](#_Toc43569909)

[3.2.1 ExamTest.java 7](#_Toc43569906)

[3.2.2 ExamCoutentMapper.java 12](#_Toc43569906)

[3.2.3 ExamMapper.java 14](#_Toc43569906)

[3.3 entity类 14](#_Toc43569909)

[3.3.1 Exam.java 14](#_Toc43569906)

[3.3.2 ExamCoutent.java 16](#_Toc43569906)

[3.3.3 ExamRecord.java 18](#_Toc43569906)

[3.3.4 User.java 19](#_Toc43569906)

[3.4 DAO接口类 20](#_Toc43569909)

[3.5工具包Util 21](#_Toc43569909)

[3.5.1 ExamCoutentUtil.java 21](#_Toc43569906)

[3.5.2 ExamUtil.java 23](#_Toc43569906)

[4 系统测试 25](#_Toc43570178)

[5 系统总结 27](#_Toc43570181)

1 需求分析

从数据库题库中随机抽取一定数量的题目（题型只有单选或多选）组合成试卷，学生对试卷

进行作答，答题完毕后程序自动完成阅卷，并给出成绩。教师用户可以对题库进行增加，删除、修改，设置试卷题目的数量，考试开始时间，自动生成试卷，对所有学生的成绩进行查询、统计（最高分，最低分，平均分，各分数段成绩百分比等）等功能。

## 1.1用户登录

当程序运行时，显示“欢迎进入考试系统，请登录”，提示用户依次输入用户名及其密码，当用户名及密码不正确时，显示提示信息“登录失败，请重新输入”，退出系统；

## 1.2主菜单

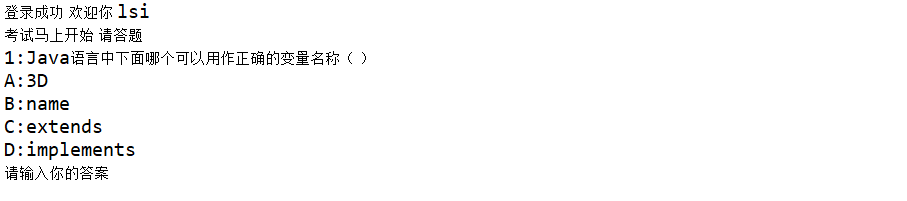
当用户名及密码正确时，进入系统主界面，如下：

当你是以老师的身份进入时：



如果输入1-5以外的选择，则显示提示信息“请输入1-5选项”后，返回主界面。

当你是以学生的身份进入时：



则会直接进行考试。

## 1.3各项功能

**用户是老师时：**

### 1.3.1 题库管理

当选择1时，显示“1：增加题目 2：删除题目 3：修改题目 4：退出”，若选择1，则会显示要添加的题目内容，选项和答案。若选择2，则会显示输入你想要删除的题目的序号。若选择3，则会显示你想要修改的题号，以及修改后题目的内容。

### 1.3.2 考试管理

当选择2时，显示“请输入考试总分:”，当输入完考试总分后，显示“请输入考试题目”，然后老师可以输入题目数量。再显示“考试日期”，若日期格式输入错误，则会显示“输入的日期格式不正确，请重新输入”。

### 1.3.3 成绩管理

当选择3时，则会把所有的学生的考试成绩全部展示出来，以“姓名+成绩”的形式显示出来。

### 1.3.4 成绩统计

当选择4时，显示“目前已经考试的所有学生的成绩的最高分、最低分、平均分、优良率、及格率、以及不及格率”。

### 1.3.5 退出

当选择5时，则退出该系统。

**当用户是学生时：**

### 1.3.6 考试

则直接进行考试，当考试结束后，会显示所得成绩。

# 2 系统设计

## 2.1用户用例图

系统分为老师，学生两个角色，各角色的用户用例如下图2-1所示：

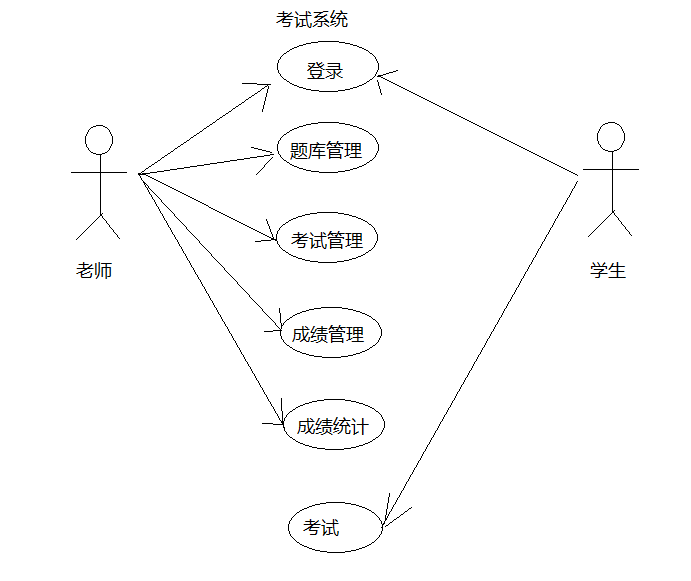


图2-1 用户用例图

## 2.2 ER图

数据库设计4张表，用户表，考试信息表，考试内容表，考试记录表。对应的ER图如图2.2.1所示。

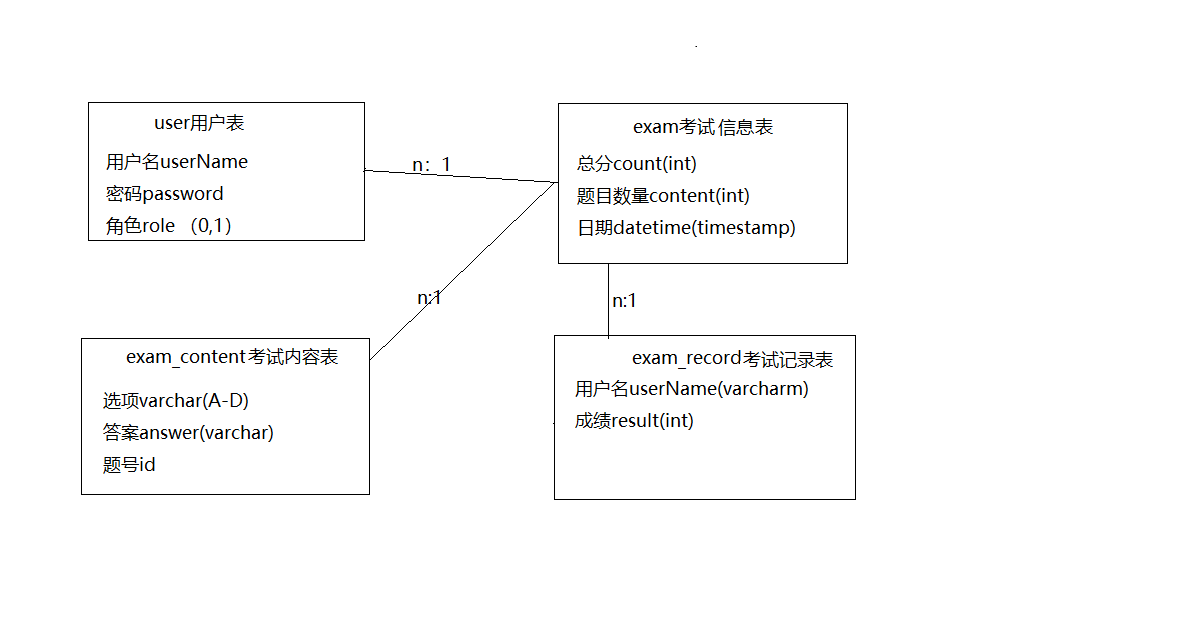


图2.2.1 数据库ER图

3

## 2.3 UML活动图（Activity Diagram）

### 2.3.1 登陆

登录操作需要验证用户名，密码，只有当用户名和密码同时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2.3.1所示。

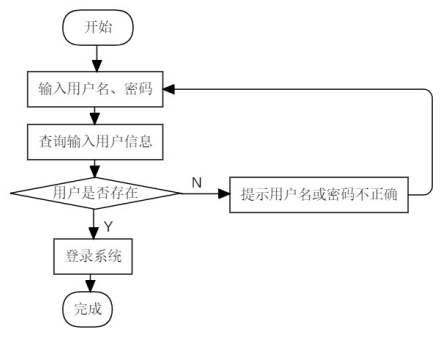
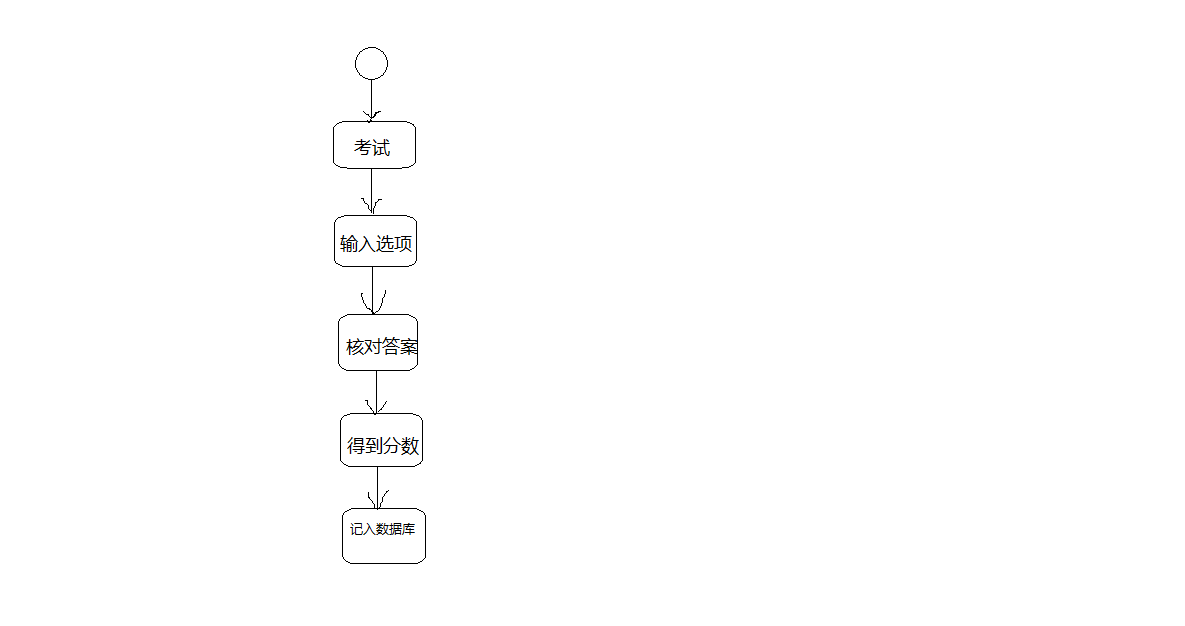


图2.3.1 登录流程图

### 2.3.2 当角色为学生时，登录后直接进入考试

图2.3.2 学生活动操作图

**2.3.3 当角色为老师时，登录后可进行选择**

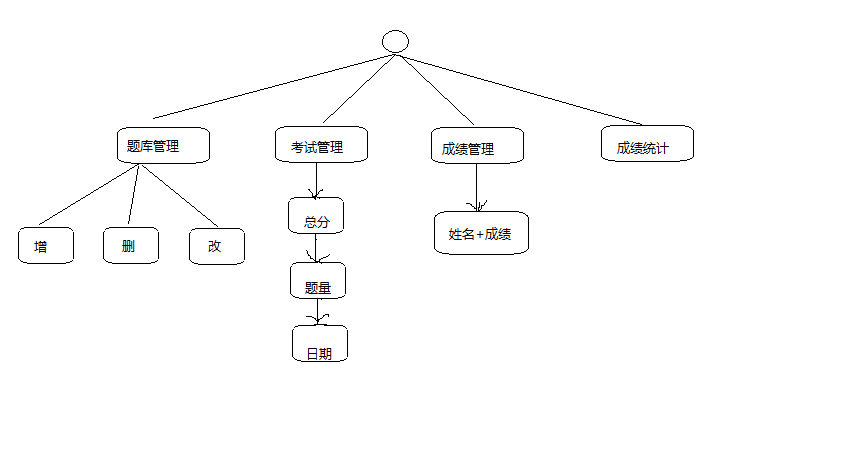
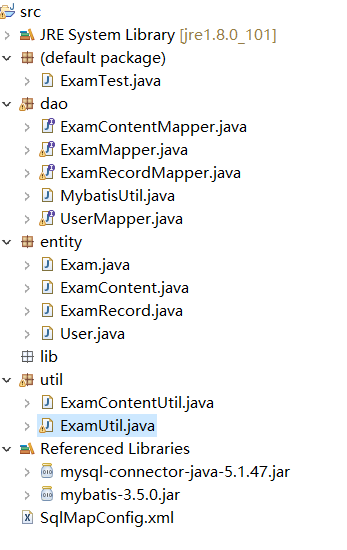
****

图2.3.3老师活动操作图

# 3 系统实现

## 3.1 项目结构



## 3.2 配置文件

### 3.2.1 ExamTest.java文件

用户界面文件，学生或老师登录时和登录后的界面在此文件形成。

import dao.\*;

import entity.Exam;

import entity.ExamContent;

import entity.ExamRecord;

import entity.User;

import util.ExamContentUtil;

import util.ExamUtil;

import java.math.BigDecimal;

import java.util.List;

import java.util.Scanner;

public class ExamTest {

static Scanner scanner=new Scanner(System.in);

UserMapper userDto= MybatisUtil.getMapper(UserMapper.class);

ExamRecordMapper examRecorddto=MybatisUtil.getMapper(ExamRecordMapper.class); ExamContentMapper examContentDto=MybatisUtil.getMapper(ExamContentMapper.class); ExamMapper examDto=MybatisUtil.getMapper(ExamMapper.class);

public static void main(String[] args) {

ExamTest exam=new ExamTest();

exam.exam();

}

/\*\*

\* 开始考试

\*/

public void exam(){

welcome();

User loginUser=login();

if (loginUser==null){

System.out.println("登录失败 退出");

return;

}

System.out.println("登录成功 欢迎你 "+loginUser.getUsername());

if (loginUser.getRole()==1){

//教师登录 展示管理

admin();

}else{

//学生登录 考试

studentExam(loginUser.getId());

}

}

/\*\*

\* 欢迎语

\*/

public void welcome(){

System.out.println("欢迎进入考试系统,");

}

//登录

public User login(){

System.out.println("请输入用户名");

String username=scanner.next();

System.out.println("请输入密码");

String password=scanner.next();

return userDto.login(username,password);

### }

### /\*\*

### \* 老师登录 管理平台

### \*/

### public void admin(){

### System.out.println("1:题库管理");

### System.out.println("2:考试管理");

### System.out.println("3:成绩管理");

### System.out.println("4:成绩统计");

### System.out.println("5:退出");

### int choose=0;

### while (choose!=5){

### System.out.println("请选择（1-5）");

### choose=scanner.nextInt();

### switch (choose){

### case 1:

### examContentManage() ;

### break;

### case 2:

### examManage();

### break;

### case 3:

### queryRecord();

### break;

### case 4:

### recordStatistics();

### break;

### default:

### System.out.println("请输入1-5的选项");

### break;

### }

### }

### }

### /\*\*

### \* 题库管理

### \*/

### public void examContentManage(){

### System.out.println("1:增加题目");

### System.out.println("2:删除题目");

### System.out.println("3:修改题目");

### System.out.println("4:退出");

### int choose=0;

### while (choose!=4){

### System.out.println("请选择（1-4）");

### choose=scanner.nextInt();

### switch (choose){

### case 1:

### ExamContentUtil.add();

### break;

### case 2:

### ExamContentUtil.delete();

### break;

### case 3:

### ExamContentUtil.update();

### break;

### default:

### System.out.println("请输入1-4的选项");

### break;

### }

### }

### }

### /\*\*

### \* 考试管理

### \*/

### public void examManage(){

### System.out.println("1:修改考试信息");

### System.out.println("2:退出");

### int choose=0;

### while (choose!=2){

### System.out.println("请选择（1-2）");

### choose=scanner.nextInt();

### switch (choose){

### case 1:

### ExamUtil.update();

### break;

### default:

### System.out.println("请输入1-2的选项");

### break;

### }

### }

### }

### /\*\*

### \* 查询考试成绩

### \*/

### public void queryRecord(){

### List<ExamRecord> list=examRecorddto.list();

### for (ExamRecord exam:list){

### System.out.println("姓名+"+exam.getUsername()+"成绩:"+exam.getResult());

### }

### }

### /\*\*

### \* 统计成绩

### \*/

### public void recordStatistics(){

### List<ExamRecord> list=examRecorddto.list();

### //总人数

### int count=list.size();

### //最高分

### int max=list.get(0).getResult();

### //最低分

### int min=list.get(0).getResult();

### //总分

### int sum=0;

### //通过人数

### int pass=0;

### //未及格人数

### int nopass=0;

### //优秀人数

### int excellent=0;

### for (ExamRecord exam:list){

### if (exam.getResult()>max){

### max=exam.getResult();

### }

### if (exam.getResult()<min){

### min=exam.getResult();

### }

### sum+=exam.getResult();

### if (exam.getResult()>=80){

### excellent++;

### }else if (exam.getResult()>=60){

### pass++;

### }else {

### nopass++;

### }

### }

### System.out.printf("最高分 %d %n",max);

### System.out.printf("最低分 %d %n",min);

### System.out.println("平均分"+except(sum,count));

### System.out.println("优良率"+except(excellent,count)+"及格率"+except(pass,count)+" 不及格率 "+except(nopass,count));

### }

### /\*\*

### \* 除运算 (保留两位小数)

### \* @param a

### \* @param b

### \* @return

### \*/

### public double except(double a,double b){

### BigDecimal bigDecimal=new BigDecimal(a/b);

### bigDecimal.setScale(2,BigDecimal.ROUND\_DOWN);

### return bigDecimal.doubleValue();

### }

### //学生参加考试

### public void studentExam(int uid){

### Exam exam= examDto.queryone();

### //每题分数

### int avg=exam.getCount()/exam.getContent\_count();

### //总分

### int sum=0;

### System.out.println("考试马上开始 请答题");

### List<ExamContent> list=examContentDto.list(exam.getCount());

### for (int i=0;i<list.size(); i++){

### System.out.println(i+1+":"+list.get(i).getDesc());

### System.out.println("A:"+list.get(i).getA());

### System.out.println("B:"+list.get(i).getB());

### System.out.println("C:"+list.get(i).getC());

### System.out.println("D:"+list.get(i).getD());

### System.out.println("请输入你的答案");

### String answer= scanner.next();

### if(list.get(i).getAnswer().equals(answer)){

### sum+=avg;

### }

### }

### //结束后 将本次考试记录数据库

### System.out.println("答题结束："+sum+"分");

### ExamRecord record=new ExamRecord();

### record.setUid(uid);

### record.setResult(sum);

### examRecorddto.save(record);

### }

### }

### 3.2.2 ExamContentMapper.java文件

考试题目数据库操作

package dao;

import entity.ExamContent;

import org.apache.ibatis.annotations.\*;

import java.util.List;

### /\*\*

### \* 考试题目

### \*/

### public interface ExamContentMapper {

### /\*\*

### \* 保存

### \* @param content

### \*/

### @Insert("insert into exam\_content values(null,#{content.desc},#{content.A},#{content.B},#{content.C},#{content.D},#{content.answer})")

### public void save(@Param("content") ExamContent content);

### /\*\*

### \* 删除

### \* @param id

### \*/

### @Delete("delete from exam\_content where id=#{id}")

### public void delete(@Param("id") Integer id);

### /\*\*

### \* 修改

### \* @param content

### \*/

### @Update("update exam\_content set `desc`=#{content.desc}, A=#{content.A} ,B=#{content.B},C=#{content.C},D=#{content.D},answer=#{content.answer} where id=#{content.id}")

### public void update(@Param("content") ExamContent content);

### /\*\*

### \* 获取考试题目

### \* @param count

### \* @return

### \*/

### @Select("select \* from exam\_content limit #{count}")

### public List<ExamContent> list(@Param("count") int count);

### }

### 3.2.3 ExamMapper.java文件

考试数据库操作

package dao;

import entity.Exam;

## import org.apache.ibatis.annotations.Param;

## import org.apache.ibatis.annotations.Select;

## import org.apache.ibatis.annotations.Update;

## import java.sql.Connection;

## import java.sql.PreparedStatement;

## import java.sql.ResultSet;

## import java.sql.SQLException;

## public interface ExamMapper {

## /\*\*

## \* 修改

## \* @param exam

## \*/

## @Update("update exam set count=#{exam.count},content\_count=#{exam.contentCount},date\_time=#{exam.dateTime}")

## public void update(@Param("exam") Exam exam);

## /\*\*

## \* 查询本次考试信息

## \* @return

## \*/

## @Select("select \* from exam limit 1")

## public Exam queryone();

## }

## 3.3 entity类

**3.3.1：Exam.java**

考试信息：总分，题量，开考日期

package entity;

import java.util.Date;

/\*\*

\* 考试信息

\*/

public class Exam {

private int id;

/\*\*

\* 总分

\*/

private int count;

/\*\*

\* 题目数量

\*/

private int content\_count;

/\*\*

\* 开考试卷

\*/

private Date dateTime;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getCount() {

return count;

}

public void setCount(int count) {

this.count = count;

}

public int getContent\_count() {

return content\_count;

}

public void setContent\_count(int content\_count) {

this.content\_count = content\_count;

}

public Date getDateTime() {

return dateTime;

}

public void setDateTime(Date dateTime) {

this.dateTime = dateTime;

}

}

**3.3.2:ExamContent.java**

考试内容：题目内容，选项显示，答案等

package entity;

/\*\*

\* 考试内容（题目）

\*/

public class ExamContent {

private int id;

/\*\*

\* 题目内容

\*/

private String desc;

/\*\*

\* A选项

\*/

private String A;

/\*\*

\* B选项

\*/

private String B;

/\*\*

\* C选项

\*/

private String C;

/\*\*

\* D选项

\*/

private String D;

/\*\*

\* 答案

\*/

private String answer;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

public String getA() {

return A;

}

public void setA(String a) {

A = a;

}

public String getB() {

return B;

}

public void setB(String b) {

B = b;

}

public String getC() {

return C;

}

public void setC(String c) {

C = c;

}

public String getD() {

return D;

}

public void setD(String d) {

D = d;

}

public String getAnswer() {

return answer;

}

public void setAnswer(String answer) {

this.answer = answer;

}

}

3.3.3 ExamRecord.java

考试记录：记录考试的同学的成绩以及用户名

package entity;

/\*\*

\* 考试记录

\*/

public class ExamRecord {

private int id;

/\*\*

\* 用户编号

\*/

private int uid;

/\*\*

\* 考试成绩

\*/

private int result;

/\*\*

\* 用户名称

\*/

private String username;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getUid() {

return uid;

}

public void setUid(int uid) {

this.uid = uid;

}

public int getResult() {

return result;

}

public void setResult(int result) {

this.result = result;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

}

3.3.4 User.java

用户信息：用户名，密码，角色等。

package entity;

/\*\*

\* 用户

\*/

public class User {

private int id;

/\*\*

\* 用户名

\*/

private String username;

/\*\*

\* 密码

\*/

private String password;

/\*\*

\* 角色 1老师 0学生

\*/

private int role;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public int getRole() {

return role;

}

public void setRole(int role) {

this.role = role;

}

}

## 3.4 DAO接口类MybatisUtil.java

获取mapper工具类

Package dao;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import java.io.IOException;

import java.io.InputStream;

public class MybatisUtil {

static SqlSession sqlSession=null;

static {

//1. 读取配置文件

InputStream is = null;

try {

is = Resources.getResourceAsStream("SqlMapConfig.xml");

} catch (IOException e) {

e.printStackTrace();

}

//2. 创建SqlSessionFactory工厂

SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();

SqlSessionFactory factory = builder.build(is);

//3. 使用工厂生产SqlSession对象

sqlSession = factory.openSession(true);

}

/\*\*

\* 获取 mapper

\* @param clazz

\* @param <T>

\* @return

\*/

public static<T> T getMapper(Class<T> clazz){

return sqlSession.getMapper(clazz);

}

}

## 3.5 工具包Util

### 3.5.1 ExamContentUtil.java

考试题目管理操作类，对题目的增删改的功能。

package util;

import dao.ExamContentMapper;

import dao.MybatisUtil;

import entity.ExamContent;

import java.util.Scanner;

/\*\*

\* 考试题目管理操作类

\*/

public class ExamContentUtil {

static Scanner scanner=new Scanner(System.in);

static ExamContentMapper dto= MybatisUtil.getMapper(ExamContentMapper.class);

/\*\*

\* 添加

\*/

public static void add(){

System.out.println("请输入题目内容");

String desc=scanner.next();

System.out.println("a:");

String A=scanner.next();

System.out.println("b:");

String B=scanner.next();

System.out.println("c:");

String C=scanner.next();

System.out.println("d:");

String D=scanner.next();

System.out.println("答案:");

String answer=scanner.next();

ExamContent content=new ExamContent();

content.setDesc(desc);

content.setA(A);

content.setB(B);

content.setC(C);

content.setD(D);

content.setAnswer(answer);

dto.save(content);

}

/\*\*

\* 删除

\*/

public static void delete(){

System.out.println("请输入你想要删除的题目ID");

Integer id=scanner.nextInt();

dto.delete(id);

}

/\*\*

\* 修改

\*/

public static void update(){

System.out.println("请输入你想要修改的题目ID");

Integer id=scanner.nextInt();

System.out.println("请输入修改后题目内容");

String desc=scanner.next();

System.out.println("a:");

String A=scanner.next();

System.out.println("b:");

String B=scanner.next();

System.out.println("c:");

String C=scanner.next();

System.out.println("d:");

String D=scanner.next();

System.out.println("答案:");

String answer=scanner.next();

ExamContent content=new ExamContent();

content.setId(id);

content.setDesc(desc);

content.setA(A);

content.setB(B);

content.setC(C);

content.setD(D);

content.setAnswer(answer);

dto.update(content);

}

}

### 3.5.2 ExamUtil.java

考试管理工具类，修改考试信息如题目，总分，题量等

package util;

## import dao.ExamMapper;

## import dao.MybatisUtil;

## import entity.Exam;

## import java.text.DateFormat;

## import java.text.ParseException;

## import java.text.SimpleDateFormat;

## import java.util.Date;

## import java.util.Scanner;

## /\*\*

## \* 考试工具类

## \*/

## public class ExamUtil {

## static Scanner scanner=new Scanner(System.in);

## static ExamMapper dto= MybatisUtil.getMapper(ExamMapper.class);

## /\*\*

## \* 修改考试信息

## \*/

## public static void update(){

## System.out.println("请输入考试总分");

## Integer count=scanner.nextInt();

## System.out.println("请输入考试题目");

## Integer contentCount=scanner.nextInt();

## System.out.println("请输入考试开始时间");

## String time=scanner.next();

## DateFormat dtf=new SimpleDateFormat("yyyy-MM-dd");

## Date date=null;

## try {

## date =dtf.parse(time);

## } catch (ParseException e) {

## System.out.println("输入的时间格式不正确");

## e.printStackTrace();

## }

## Exam exam=new Exam();

## // exam.setContentCount(contentCount);

## exam.setCount(count);

## exam.setDateTime(date);

## dto.update(exam);

## }

## }

# 4 系统测试

设计测试用例，给出程序每个功能模块的运行结果截图。

**4.1**当用户为学生时，系统运行结果如下图：

 图4.1.1学生登录图

当作答完成后，会直接显示成绩，如下图：

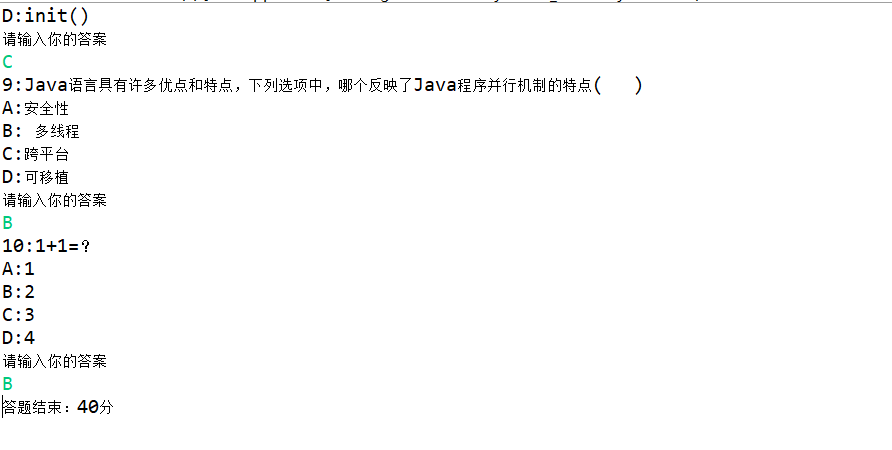


图4.1.2学生成绩图

# 4.2当用户是老师时，登录后显示如下图：

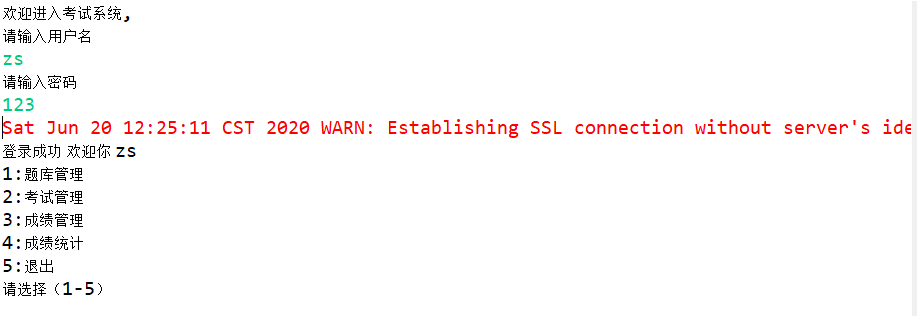


图4.2.1老师登录图

当老师选择1题库管理时(题目的增删改)，会显示如下图：

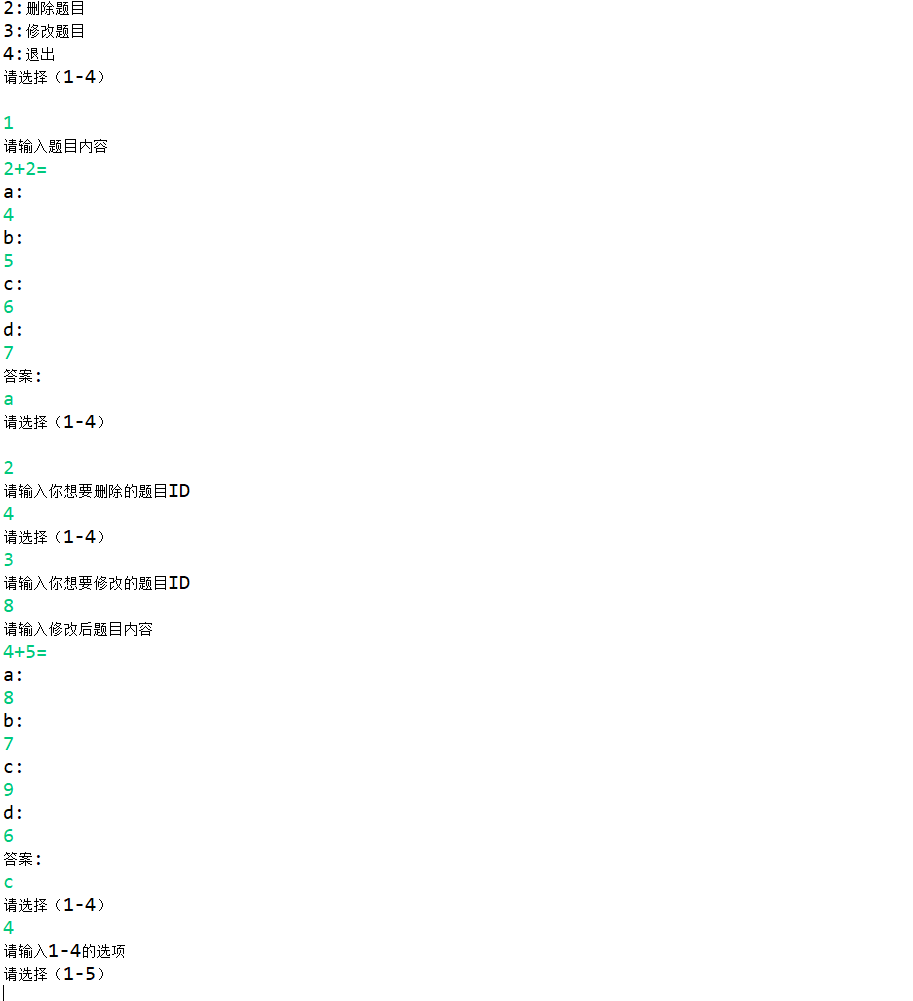


图4.2.2老师题库管理图

# 当老师选择2考试管理时，会显示以下功能，如图：



图4.2.3老师考试管理图

当老师选择3成绩管理时，会显示出所有考试后学生的名字和得分，如下图：



图4.2.4老师成绩管理图

当老师选择4成绩统计时，系统会自动将成绩统计好，如图：

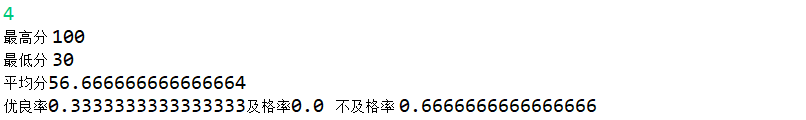


图4.2.5老师成绩统计图

# 5 系统总结

主要问题的解决过程:主要问题就是对Eclipse的使用不够熟练，还有对MyBatis的框架不够熟悉。解决的方法就是百度，然后问问成绩优秀的同学。还有运行过程中跟数据库连接不上，然后通过看上课保存的录屏加上百度进行解决。

系统存在的不足：没有将学生的学号也录入，所以当有学生重名时，系统无法识别。另外，老师登录后的考试管理这个选项功能有点少。

本门课程的收获：作为一门编程类学科，想要学好，无他，唯熟耳。所以，要想学好编程类课程，就必须自己多动手。另外，对Java这门语言也有了新的认识。

课程的建议：上课适当可以讲慢一点点，我就经常因为上课跟不上，所以下课只能看录屏回放。当然，这只是我个人建议，也有可能只是我一个人跟不上。

以后的发展方向：发展方向，明年的考研争取上岸。

自评成绩：功能还算比较完善，老师需要的功能也基本都有，给自己80分吧。