

课程编号：B080103303

etsys 在线考试系统 说明文档



姓 名		学 号	
班 级		指 导 教 师	
实 验 名 称			
开 设 学 期			
开 设 时 间			
报 告 日 期			
评 定 成 绩		评 定 人	
		评 定 日 期	

东北大学软件学院

摘要

1. 项目所用技术简介:

技术方面	技术内容
Java 框架	Spring+SpringMVC+Mybatis
前端界面	Html5+CSS3+JavaScript+jQuery+Bootstrap
Linux/服务器	MySQL+Redis 缓存+JDK1.8+Tomcat
项目搭建/集成/部署	Apache-Maven-3.5

2. 调试环境:

操作系统: Win10/CentOS 6.5

集成开发工具: Eclipse-Oxygen

3. 项目名称:

etsys (本系统)	角色	打包类型
etsys-root	聚合所有模块	pom
etsys-commons	通用工具	jar
etsys-orm	数据/关系映射	jar
etsys-core	实现 Web 项目逻辑	war

4. 其他补充说明:

(1) 外部交互: 本系统的部分数据来自于外部系统(教务系统), 为此, 考虑到项目调试的需要, 除了本项目的开发之外, 还额外开发了一个模拟的教务系统, 发布 Web 服务提供数据。

(2) 缓存服务器: 本系统为应对多人并发访问, 特意部署了缓存服务器(Redis), 以便在多人并发时确保数据正常回传。

目录

1 需求分析.....	1
1.1 教师端需求分析.....	1
1.2 学生端需求分析.....	1
2 系统设计.....	2
2.1 体系结构设计.....	2
2.2 详细设计.....	3
2.2.1 用例建模.....	3
2.2.2 类建模.....	4
2.2.3 动态建模.....	5
3 系统实现.....	6
3.1 教师端功能实现.....	6
3.1.1 注册/登录.....	6
3.1.2 数据刷新.....	8
3.1.3 我的课程.....	10
3.1.4 我的学生.....	11
3.1.5 我的模板.....	13
3.1.6 题库设置.....	15
3.1.7 考试管理.....	18
3.2 学生端功能实现.....	21
3.2.1 我的课程.....	21
3.2.2 我的考试信息.....	22
3.2.3 成绩查询.....	26
4 系统测试.....	27
4.1 教师端功能测试.....	27
4.1.1 注册/登录.....	27
4.1.2 我的课程.....	27
4.1.3 我的学生.....	28
4.1.4 我的模板.....	28
4.1.5 考试管理.....	29
4.1.6 题库设置.....	30
4.1.7 数据刷新.....	31
4.2 学生端功能测试.....	32
4.2.1 我的课程.....	32
4.2.2 考试信息查询.....	33
4.2.3 成绩查询.....	34
5 项目总结.....	35
5.1 技能收获.....	35
5.2 团队收获.....	35

1 需求分析

1.1 教师端需求分析

教师端的需求如下表 1.1-1 所示

需求名称	子需求名称	需求描述
教师注册/登录	/	教师在输入自己在教务系统中的用户名和密码后，该系统向教师反应教务系统中有关他本人的相关信息，确认后，即可注册入本系统。 注册完成之后，可随时登录。
数据刷新	/	教师可直接将教务系统中的课程信息和学生信息导入本系统，以便学生访问。
我的课程	/	教师可查看自己所教授的课程，但不得删除。
我的学生	/	教师可查看每门课程相对应选择修读的学生。
我的模板	查看模板	教师可查看创建的试题模板。
	新增模板	教师可为每门课的考试创建试题模板，用于在考试时生成试卷，包括试卷难度/题目数量/每题分值等信息。
题库设置	/	教师可为每门课创建题库，在考试时随机抽取，可为每道题设置试题难度/题目类型/题干内容/参考答案等信息。
考试管理	发布考试	教师可发布考试信息，包括考试时间/试题模板/试卷难度等内容。
	查看考试	教师可查询到自己发布的所有考试。
	批改试卷	教师可批改试卷中所有主观题，客观题由本系统自动批改，之后方可查看学生成绩。

表 1.1-1

1.2 学生端需求分析

学生端的需求如下表 1.2-1 所示

需求名称	需求描述
我的课程	学生可查看所选课程。
我的考试信息	学生可查看所有要参加的考试，进入考场后，方可作答结束后，可直接提交。
成绩查询	学生在交卷后可直接查看客观题成绩，待教师批改完成之后，可查看全卷成绩。

表 1.2-1

2 系统设计

2.1 体系结构设计

系统体系结构设计如下图 2.1-1 所示，体系结构设计说明如下表 2.1-1 所示

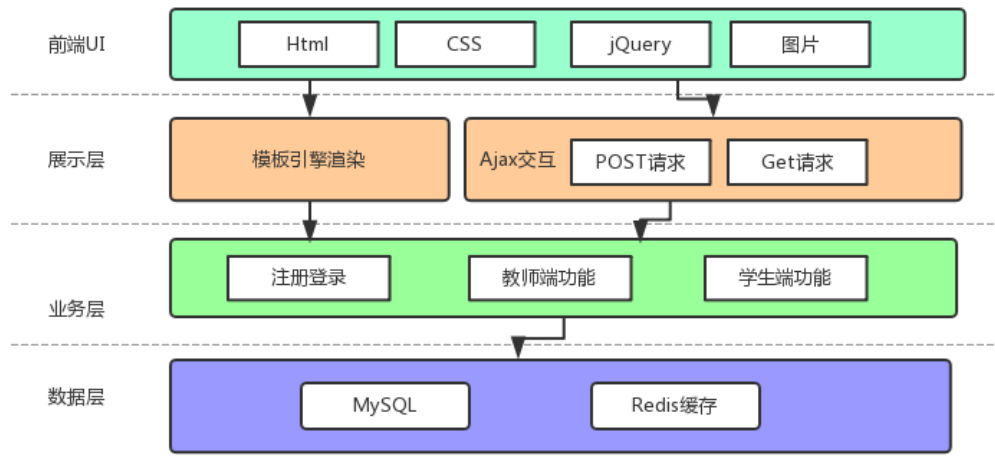


图 2.1-1

结构层次	结构内部模块	承担作用描述
前端 UI	Html	主要是 jsp 页面,负责展示动态产生数据结果。
	CSS	引用 BootStrap 样式。
	jQuery	用于表单序列化以及提交 Ajax 请求。
	图片	存入静态资源服务器,通过 nginx 访问(暂未实现)。
展示层	模板引擎渲染	处理普通请求。
	Ajax 交互	处理 ajax 请求。
业务层	注册登录	教师注册/登录,学生登录,其中教师的注册需访问教务系统(外部系统)。
	教师端功能	实现需求分析中教师端业务需求。
	学生端功能	实现需求分析中学生端业务需求。
数据层	MySQL	保存持久层数据。
	Redis 缓存	在高并发时,降低数据库负载。

表 2.1-1

2.2 详细设计

2.2.1 用例建模

系统用例建模如下图 2.2.1-1 所示，其中教师的用例可分为登录系统，注册，查看课程，试题模板管理（创建试题模板和查看试题模板），题库管理，学生信息导入，课程信息导入，批改试卷和发布考试信息等。学生的用例可分为登录系统，查看所选课程，查看成绩和进入考场。用例如下表 2.2.1-1 所示。

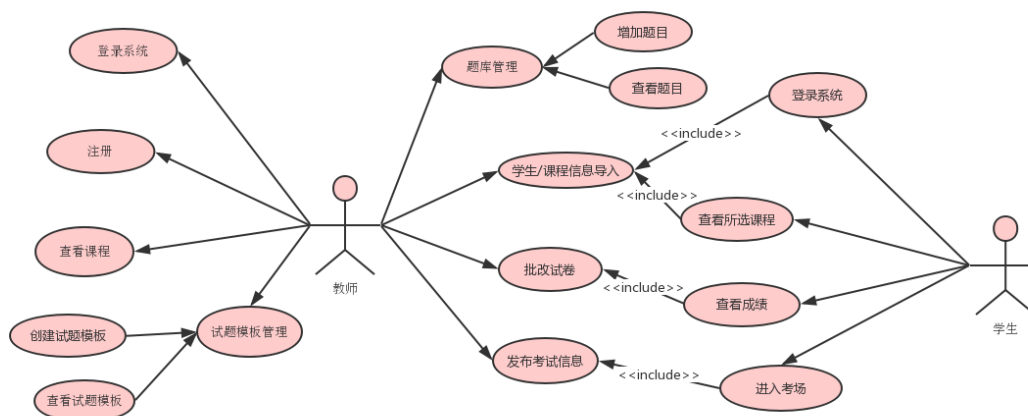


图 2.2.1-1

用例归属	用例名称
教师	登录系统
/	注册
/	查看课程
/	试题模板管理
/	题库管理
/	学生信息导入
/	课程信息导入
/	批改试卷
/	发布考试
学生	登录系统
/	查看所选课程
/	查看成绩
/	进入考场

表 2.2.1-1

2.2.2 类建模

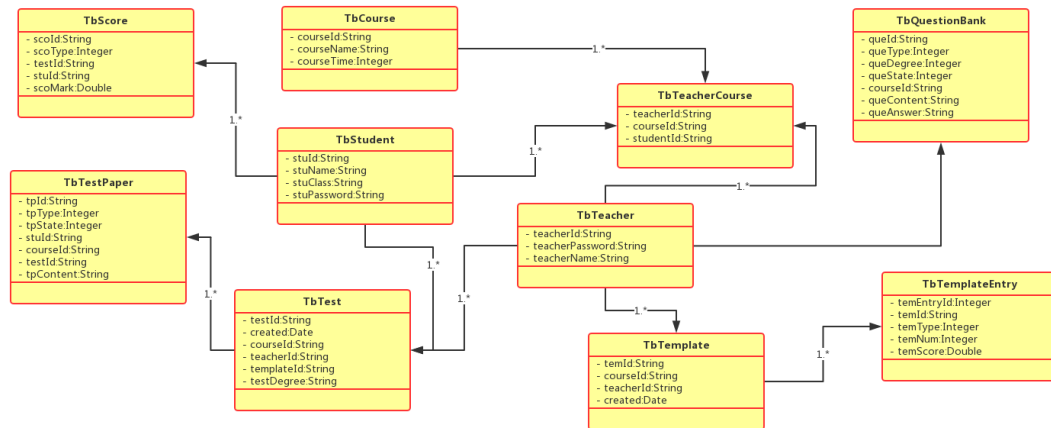


图 2.2.2-1

系统类建模如图 2.2.1-2 所示，共有 TbScore、TbCourse、TbTestPaper、TbTest、TbStudent、TbTeacher、TbTemplate、TbTemplateEntry、TbTeacherCourse 和 TbQuestionBank 等 10 个类。其详细说明如下表 2.2.1-2 所示。

类名	含义	包含属性	关联关系
TbScore	学生成绩	- scoId:String - scoType:Integer - testId:String - stuId:String - scoMark:Double	和 TbStudent 为多对一
TbCourse	课程信息	- courseId:String - courseName:String - courseTime:Integer	和 TbTeacherCourse 为一对多
TbTestPaper	考试卷	- tpId:String - tpType:Integer - tpState:Integer - stuId:String - courseId:String - testId:String - tpContent:String	和 TbTest 为多对一
TbTest	考试信息	- testId:String - created:Date - courseId:String - teacherId:String - templateId:String - testDegree:String	和 TbTestPaper 为一对多
TbStudent	学生信息	- stuId:String - stuName:String - stuClass:String - stuPassword:String	和 TbScore/ TbTest/ TbTeacherCourse 为一对多

TbTeacher	教师信息	- teacherId:String - teachPaswrd:String - teacherName:String	和 TbTest/ TbTeacherCourse/ TbTemplate/ TbQuestionBank 为 一对多
TbTemplate	试题模板	- temId:String - courseId:String - teacherId:String - created:Date	和 TbTeacher 为多对 一，和 TbTemplateEntry 为 一对多
TbTemplateEntry	试题模板条目	- temEntryId:Integer - temId:String - temType:Integer - temNum:Integer - temScore:Double	和 TbTemplate 为多 对一
TbTeacherCourse	选课记录	- teacherId:String - courseId:String - studentId:String	和 TbCourse/ TbStudent/ TbTeacher 为多对一
TbQuestionBank	试题库	- queId:String - queType:Integer - queDegree:Integer - queState:Integer - courseId:String - queContent:String - queAnswer:String	和 TbTeacher 为多对 一

表 2.2.2-2

2.2.3 动态建模

系统动态建模如图 2.2.3-1 和 2.2.3-2 所示，具体描述为如下表 2.2.3-1 所示。

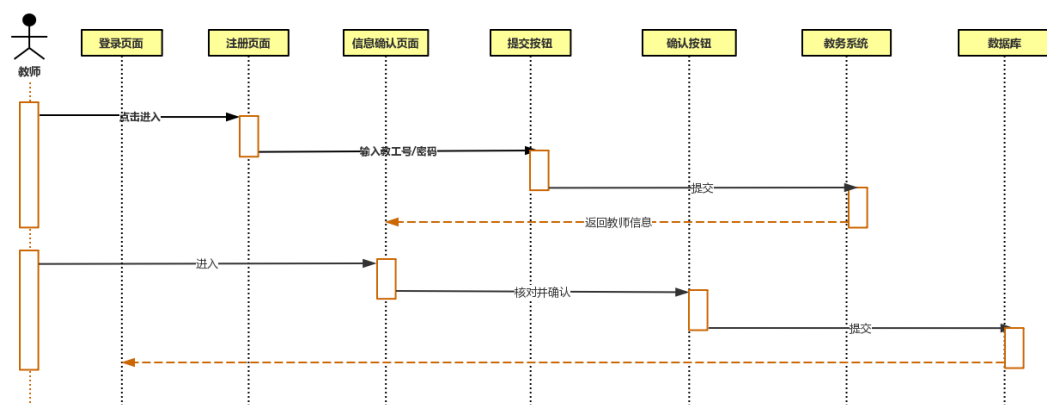


图 2.2.3-1 教师注册时本系统和外部系统交互动态模型

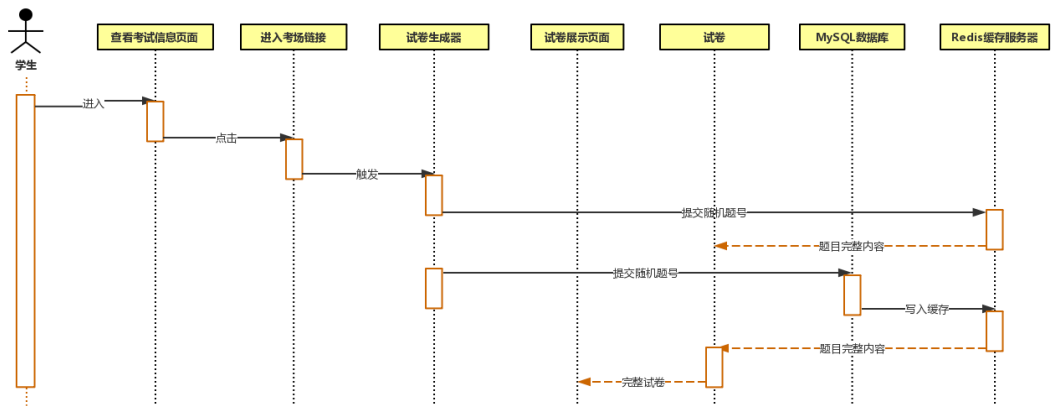


图 2.2.3-2 多人并发考试系统读写 redis 缓存动态模型

动态模型描述	动态过程描述	应用场景
教师注册时本系统和外部系统交互动态模型	1.教师进入注册页面。 2.输入教工号和密码，并提交。 3.本系统向教务系统请求验证该教师信息合法性。 4.教务系统返回该教师基本信息。 5.本系统展示教务系统返回数据。 6.教师确认信息，并注册。	该动态模型为教师注册个人信息，需要本系统与外部系统（教务系统）交互，以便验证教师信息的合法性。
多人并发考试系统读写 redis 缓存动态模型	1.学生点击“进入考试”链接。 2.本系统根据试卷生成器和试题模板生成试卷。 3.本系统进入数据库，返回试卷当中的每道题，并写入 redis 缓存。 4.此后，系统抽题时，会率先进入 redis 缓存中查询是否有该题，如果有直接返回，否则查询数据库。	该动态模型为多人并发进入考场时，为降低数据库负载，将试卷当中的每一道题从数据库抽出后，写入 redis 缓存服务器，如果后面进入考试的学生也抽到了此题，则直接从 redis 缓存中提取该题即可。

3 系统实现

3.1 教师端功能实现

3.1.1 注册/登录

为确保教师相关数据的合法性，在教师注册时，本系统会与该校教务系统交互，从而验证

教师信息的真实性。教务系统将有关数据以 web 服务的形式向外部系统提供，数据格式如下表 3.1.1-1 所示：

请求格式	请求 url	请求方式	请求参数
	http://{hostId}:{hostPort}/edas/instructor/validate/		instructorId instructorPassword
响应格式	status	msg	data
	200	OK	{ "instructorId" : "ins133373359", "instructorName" : "龚开梦", "instructorPlace" : "教授", "instructorTel" : "18238459962", "instructorEmail" : "664941743@qq.com", "instructorAge" : 44 }

表 3.1.1-1

首先，教师进入注册界面，输入教务系统的用户名和密码，如下图 3.1.1-1 所示；点击验证后，本系统会调用教务系统接口来验证，调用方式如下：

```
public Instructor getInstructor(String instructorId, String instructorPassword) {
    //构造 post 请求参数
    Map<String, String> param = new Hashtable<String, String>();
    param.put("instructorId", instructorId);
    param.put("instructorPassword", instructorPassword);

    //调用教务系统接口
    String json = HttpClientUtil.doPost(EDUCATION_ADMIN_BASE_URL +
    EDUCATION_ADMIN_INSTRUCTOR_URL, param);

    if (!StringUtils.isBlank(json)) {

        JsonResult result = JsonResult.format(json);
        String jsonData = (String) result.getData();
        //将教务系统返回教师的 json 数据取出
        Instructor instructor = JsonUtils.jsonToPojo(jsonData, Instructor.class);

        return instructor;
    }

    return null;
}
```

本系统会将该教师的详细信息展示，如图 3.1.1-1 所示；确认无误后，即可成功注册入本系统。

注册

用户名

ins133373359

密码

●●●●●●

验证

教师身份验证信息确认

教工编号

ins133373359

教师姓名

龚开梦

职称

教授

年龄

44

电话号码

18238459962

邮箱

664941743@qq.com

确认无误？点我注册

图 3.1.1-1

之后，教师可选择登录，登录后主页如下图 3.1.1-3 所示：

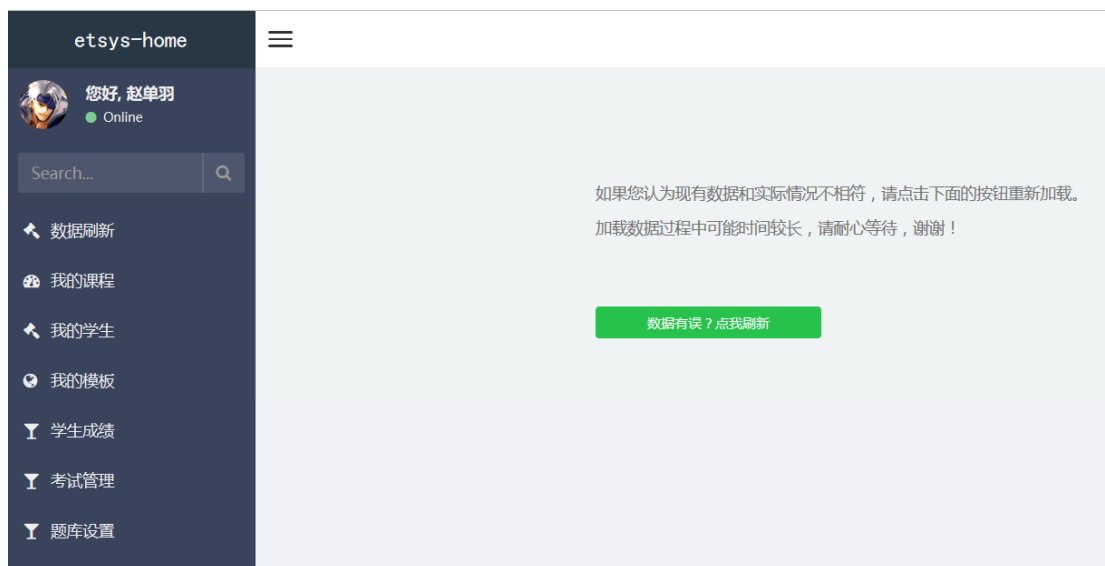


图 3.1.1-3

3.1.2 数据刷新

当教师注册入本系统后，可将他教授的课程和所有学生信息从教务系统导入本系统，点击“数据有误？点我刷新按钮”即可完成数据同步，如下图 3.1.2-1 所示。

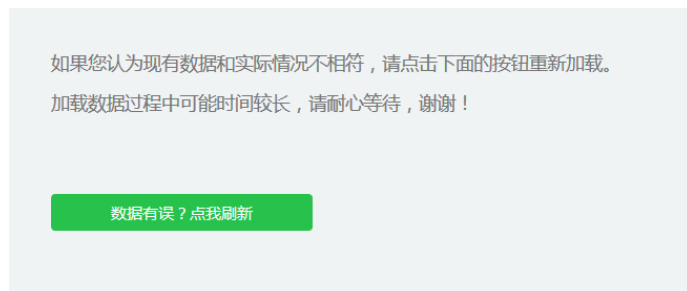


图 3.1.2-1

访问教务系统并写入本系统数据库的实现如下：

```
//根据教师 id 查询其所教授课程
String courseJson = HttpClientUtil
    .doGet(EDUCATION_ADMIN_BASE_URL
    EDUCATION_ADMIN_COURSE_RECORD_URL + teacherId);
String courseJsonData = (String) JsonResult.format(courseJson).getData();
List<Course> courseList = JsonUtils.jsonToList(courseJsonData, Course.class);
for (Course course : courseList) {
    TbCourse tbCourse = new TbCourse();
    .....
    //将课程信息写入本系统
    try {
        courseMapper.insert(tbCourse);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
for (Course course : courseList) {
    //根据教师 id 和课程 id 向教务系统请求选修该门课的学生
    Map<String, String> params = new Hashtable<String, String>();
    params.put("instructorId", teacherId);
    params.put("courseId", course.getCourseId());
    //向教务系统发送请求
    String stuJson = HttpClientUtil.doPost(EDUCATION_ADMIN_BASE_URL
    EDUCATION_ADMIN_STUDENT_URL, params);
    String stuJsonData = (String) JsonResult.format(stuJson).getData();
    List<Student> stuList = JsonUtils.jsonToList(stuJsonData, Student.class);
    for (Student student : stuList) {
        TbStudent tbStudent = new TbStudent();
        .....
        //将学生学号作为默认初始密码，同时通过 md5 哈希散列值存储。
        tbStudent.setStuPassword(DigestUtils.md5DigestAsHex(student.getStudentId().getBytes()));
        try {
            //将学生信息写入本系统
            studentMapper.insert(tbStudent);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
//通过已经从教务系统获取的教师/学生/课程数据构造选课记录
for (Student student : stuList) {
    TbTeacherCourse teacherCourse = new TbTeacherCourse();
    teacherCourses.add(teacherCourse);
}
}
```

```

}
//将构造的选课记录写入本系统
for (TbTeacherCourse tbTeacherCourse : teacherCourses) {
    try {
        teacherCourseMapper.insert(tbTeacherCourse);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

3.1.3 我的课程

当教师选择“我的课程”时，会从本系统数据库中抽取该老师所教授的课程并展示，如图 3.1.3-1 所示。如果教师认为数据未和实际情况同步，可以点击“数据有误？点我刷新”的按钮，直接跳转至“数据刷新”的页面，从而重新同步数据。

课程编号	课程名称	课程时长
ed611951469	软件测试	48
ed888855008	人机交互设计	16

[数据有误？点我刷新](#)

图 3.1.3-1

课程数据的列表展示的实现如下所示：

页面请求：

```

<a href="/course/getByTeacherId?teacherId=${sessionScope.teacher.teacherId }
&returnPage=teach-courses"><i class="fa fa-dashboard"></i> <span>我的课程</span></a>

```

Controller 层：

```

@RequestMapping("/getByTeacherId")
public String getCoursesByTeacherId(@RequestParam("teacherId") String teacherId,
    @RequestParam("returnPage") String returnPage, ModelMap modelMap) {
    //调用 Service 业务层
    List<TbCourse> list = courseService.getCoursesByTeacherId(teacherId);
    //将数据写入视图
    modelMap.put("courseList", list);
    return returnPage;
}

```

Service 层：

```

@Override
public List<TbCourse> getCoursesByTeacherId(String teacherId) {

    TbTeacherCourseExample example = new TbTeacherCourseExample();
    //拼接 sql 语句

```

```

example.setDistinct(true);
Criteria criteria = example.createCriteria();
criteria.andTeacherIdEqualTo(teacherId);

List<TbCourse> courses = new ArrayList<TbCourse>();
//调用 Mapper 层读取数据库
List<TbTeacherCourse> list = teacherCourseMapper.selectByExample(example);
for (TbTeacherCourse tbTeacherCourse : list) {
    TbCourse course = courseMapper.
        selectByPrimaryKey(tbTeacherCourse.getCourseId());
    courses.add(course);
}
return courses;
}

```

Mapper 层和 Pojo 由 Mybatis generator 自动生成，不需要实现。

3.1.4 我的学生

当教师选择“我的学生”时，系统会首先展示教师所教授的课程列表，当点击课程编号的链接时，系统会展示出所有选修这门课的学生，并分页展示，如下图 3.1.4-1 所示：

数据有误？点我刷新		
学生编号	学生姓名	所在班级
stu18250900	李开心	软件工程410
stu22056972	李子跃	软件工程410
stu23774156	李海必	软件工程410
stu25215934	李兆峰	软件工程410
stu27747715	李远	软件工程410
stu29391564	李浔	软件工程410
stu31203869	李蛹	软件工程410
stu33102240	李伟	软件工程410
stu36026759	李川	软件工程410
stu38714856	李海	软件工程410
<div>« 1 2 3 4 5 »</div>		

图 3.1.4-1

学生信息列表的实现如下所示：

页面请求：

```

<a href="/student/getByCourseAndTeacher?teacherId=${sessionScope.teacher.teacherId }
&courseId=${course.courseId }&returnPage=teach-stus-check">${course.courseId }</a>

```

Controller 层：

```

@RequestMapping(value = "/getByCourseAndTeacher", method = RequestMethod.GET)
public String getStuByCourseAndTeacher(@RequestParam("teacherId") String teacherId,

```

```

@RequestParam("courseId") String courseId, @RequestParam("returnPage") String
returnPage,
//默认取第一页
@RequestParam(defaultValue="1") Integer pageNum,
//每页数量默认为 10 个
@RequestParam(defaultValue = "10") Integer pageSize, ModelMap modelMap) {
    TbCourse course = courseService.getCourseById(courseId);
    //Service 层读取该页数据，封装成 PageHelperResult 对象返回
    PageHelperResult<TbStudent> result = studentService.
    getByCourseAndTeacher(teacherId, courseId, pageNum, pageSize);
    modelMap.put("students", result.getResultList());
    modelMap.put("total", result.getTotal());
    modelMap.put("course", course);

    return returnPage;
}

```

Service 层:

```

@Override
public PageHelperResult<TbStudent> getByCourseAndTeacher(String teacherId, String
courseId, int pageNum, int pageSize) {

    TbTeacherCourseExample example = new TbTeacherCourseExample();
    //拼接 sql 语句
    Criteria criteria = example.createCriteria();
    criteria.andTeacherIdEqualTo(teacherId);
    criteria.andCourseIdEqualTo(courseId);

    List<TbStudent> students = new ArrayList<TbStudent>();
    //分页插件实现 sql 拼接 limit 语句
    PageHelper.startPage(pageNum, pageSize);
    List<TbTeacherCourse> list = teacherCourseMapper.selectByExample(example);

    PageInfo<TbTeacherCourse> pageInfo = new PageInfo<>(list);
    .....
    //构造 PageHelperResult 对象，包括数据列表，和所有分页数据数量总和。
    PageHelperResult<TbStudent> result = new PageHelperResult<>();
    result.setResultList(students);
    result.setTotal(pageInfo.getTotal());

    return result;
}

```

Pojo 层:

```

PageHelperResult 对象定义:
public class PageHelperResult<T> {

```

```
private List<T> resultList;//分页数据
private long total;//所有分页数据量总和
}
```

3.1.5 我的模板

当教师选择“我的模板”时，系统同样会列出该教师所教授的所有课程，点击某一门课程的编号，教师即可为该门课创建考试的试题模板，以便在考试时生成试卷，如下图 3.1.5-1 所示：

每个模板分为 5 部分，对应考试当中的 5 种题型，分别是选择题/填空题/判断题/简答题/论述与分析题，教师可为每一种题型定义题目的数量和每题的分值。

在创建完成之后，直接点击“提交”，即可查看，如下图 3.1.5-2 所示。

添加考试模板，该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

模板编号：

试题类型：

试题数量：

每题分值：

模板编号：

试题类型：

试题数量：

每题分值：

模板编号：

图 3.1.5-1

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

该门课程的试题模板详细信息如下：

题目类型	题目数量	题目分值
选择题	10	3.0
填空题	5	2.0
判断题	10	1.0
简答题	4	5.0
论述与分析题	3	10.0

图 3.1.5-2

创建模板的实现如下所示：

页面请求:

```
$(document).ready(function() {
    //点击“提交按钮”，触发事件
    $("#commit").click(function() {
        //用于保存序列化后的表单数据
        var json = "[";
        //设置请求数据格式为 json
        $.ajaxSetup({
            contentType : 'application/json;charset=utf-8'
        });
        //遍历每个模板项的表单
        $("form").each(function() {
            //序列化表单数据
            var jsonUnit = SerializeToJson.formToJson($(this).serialize());
            json += jsonUnit + ",";
        });
        json = json.substr(0,json.length - 1);
        json += "]";
        //去除无效表单数据
        json = json.replace("{\\\"\",\", \"\");
        //发送 ajax 的 post 请求，包含序列化的表单数据
        $.post('/template/insertTemplateEntry', json, function(data) {
            if (data.status == 200) {
                alert("提交成功")
                .....
            } else {
                alert("不允许为空")
            }
        });
    });
});
```

Controller 层:

```
@RequestMapping("/insertTemplateEntry")
@ResponseBody
//Spring 框架会将 ajax 请求提交的 json 数组转换为 Java 的 List
public JsonResult insertTemplateEntry(@RequestBody List<TbTemplateEntry> templateEntry)
{
    try {
        //将模板写入数据库
        for (TbTemplateEntry tbTemplateEntry : templateEntry) {
            templateService.insertTemplateEntry(tbTemplateEntry);
        }
        return JsonResult.ok();
    } catch (Exception e) {
```

```

        e.printStackTrace();
        //如有异常，返回异常信息
        return JsonResult.build(500, ExceptionUtil.getStackTrace(e));
    }
}

```

Service 层：

```

@Override
public void insertTemplateEntry(TbTemplateEntry templateEntry) {
    templateEntryMapper.insert(templateEntry);
}

```

3.1.6 题库设置

当教师选择“题库设置”，系统首先展示课程列表。

教师可选择一门课，点击该门课的课程编号链接，即可看到该门课的包括 5 种题型的所有题库，点击“查看”，可看到每个题目的详细信息，如下图 3.1.6-1 所示。

点击增加题目，即可扩充该门课题库，包括题目类型/题目难度/题干描述/答案描述等信息，如下图 3.1.6-2 所示。

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

[增加题目](#)

该门课程的题库信息如下：

题目编号	题目类型	题目难度	题目状态	课程编号	题目内容
que150787787747448	1002	3	1	ed611951469	查看
que150787787853670	1002	3	1	ed611951469	查看
que150787787879269	1002	3	1	ed611951469	查看

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

该题目信息如下：

题目编号	题目类型	题目难度	题目状态	课程编号
que150791130542459	1004	3	1	ed611951469

详细信息如下：

问题描述	答案描述
<p>请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。</p>	<p>黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。</p> <p>白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。</p> <p>软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。</p>

图 3.1.6-1

扩充题库的实现如下所示：

页面请求：

```
$(document).ready(function() {
    //点击“提交”按钮，触发事件
    $(".button").click(function() {
        //将表单各项序列化为映射列表
        var serialized = $("form").serializeArray();
        $.ajaxSetup({
            contentType: 'application/json;charset=utf-8'
        });
        var json = "";
        //遍历序列化映射列表，构造 json 数据，同时将回车换行符换为<br>
        for(var i = 0; i < serialized.length; i++) {
            json+="\""+serialized[i].name+"\": \""+ serialized[i].
                value.replace(/\r\n/g, "<br>") + "\", ";
        }
        json = json.substr(0,json.length - 1);
        json = "{" + json + "}";
        .....
        //发送 ajax 的 post 请求
        $.post('/questionBank/insertBankEntry', json, function(data) {
            if (data.status == 200) {
                .....
            } else {
            }
        });
    });
});
```

});

添加考试模板，该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

题目类型: 选择题

题目难度: 简单

课程编号: ed611951469

题干描述:

答案描述:

提交

图 3.1.6-2

Controller 层:

```

@RequestMapping("/insertBankEntry")
@ResponseBody
//接收 ajax 请求的 json 数据，以对象的引用为参数
public JsonResult insertBankEntry(@RequestBody TbQuestionBankWithBLOBs questionBank)
{
    questionBank.setQueId("que" + IdUtils.genItemId());
    questionBank.setQueState(1);
    //将题目写入数据库
    questionBankService.insertBankEntry(questionBank);

    return JsonResult.ok();
}

```

Service 层

```

@Override
public Integer insertBankEntry(TbQuestionBankWithBLOBs questionBank) {
    int result = questionBankMapper.insert(questionBank);
    return result;
}

```

3.1.7 考试管理

当教师选择“考试管理”，系统会弹出下拉列表。

教师可选择发布考试，系统首先展示课程列表，点击一门课的课程编号链接，可为该门课创建一场考试，可设置考试时间和试卷难度，如下图 3.1.7-1 所示。

教师可选择批改试卷，系统首先展示课程列表，点击一门课的课程编号链接，可为已交卷的学生批改试卷的主观题部分，客观题由系统自动批阅，如下图 3.1.7-2 所示。

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

发布考试信息

选择模板：

 考试时间：

 课程编号：

 教师编号：

 试卷难度：

图 3.1.7-1

发布考试的实现如下所示：

页面请求：

```
$(document).ready(function() {
    $("#create").click(function() {
        //表单序列化过程
        var json = SerializeToJson.formToJson($("#form").serialize());
        json = json.replace("+", "T");
        $.ajaxSetup({
            contentType : 'application/json;charset=utf-8'
        });
        //提交 ajax 请求
        $.post('/test/createTest', json, function(data) {
            if (data.status == 200) {

            } else {

            }
        });
    });
});
```

Controller 层：

```
@RequestMapping("/createTest")
@ResponseBody
```

```
//把考试信息对象的引用作为参数接收
public JsonResult createTest(@RequestBody TbTest test) {

    try {
        test.setTestId("test" + IdUtils.genItemId());
        testService.insertTest(test);
    } catch (Exception e) {
        e.printStackTrace();
        return JsonResult.build(500, ExceptionUtil.getStackTrace(e));
    }

    return JsonResult.ok();
}
```

Service 层

```
@Override
public Integer insertTest(TbTest test) {
    int result = testMapper.insert(test);
    return result;
}
```

试卷主观题部分批改的实现如下所示：

页面请求：

```
$(document).ready(function() {
    $("#finish").click(function() {
        $.ajaxSetup({
            contentType : 'application/json;charset=utf-8'
        });

        var flag = true;

        var stuId = "${studentId}";
        var courseId = "${courseId}";
        var testId = "${testId}";
        //将填空题得分的表单序列化，并计算分数
        var jsonBlank = SerializeToJson.serializeScore(
            testId, stuId, 1001, "#blank form");
        //提交填空题分数
        $.post('/score/insertScore', jsonBlank, function(data) {
            if (data.status != 200) {
                flag = false;
            }
        });
        //其余题目的表单序列化和分数提交类似
        if (flag == true) {
            alert("批改成功！")
        }
    });
});
```

```

.....
});
});
});

```

论述与分析题

得分：
 (本题满分：10.0)

题目描述：为什么要在一个团队中开展软件测试工作？

参考答案：因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

学生答案：

得分：
 (本题满分：10.0)

题目描述：请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

参考答案：黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。
 白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。
 软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。

学生答案：

确认提交批改

图 3.1.7-2

Controller 层:

```

@RequestMapping("/insertScore")
@ResponseBody
//将分数对象引用作为参数接收
public JsonResult insertScore(@RequestBody TbScore score) {

    try {
        //将分数写入数据库
        score.setScoId("sco" + IdUtils.genItemId());
        scoreService.insertScore(score);
        .....
        //将试卷状态改为“已批改”
        testPaperService.updateTestPaper(testPaper);

        return JsonResult.ok();
    } catch (Exception e) {
        e.printStackTrace();
        return JsonResult.build(500, ExceptionUtil.getStackTrace(e));
    }
}

```

Service 层

```

@Override
public Integer insertScore(TbScore score) {
    int result = scoreMapper.insert(score);
}

```

```

        return result;
    }
    @Override
    public Integer updateTestPaper(TbTestPaper testPaper) {
        int result = testPaperMapper.updateByPrimaryKey(testPaper);
        return result;
    }

```

3.2 学生端功能实现

3.2.1 我的课程

当学生选择左侧“我的课程”一栏，系统列出该学生所选课程，如下图 3.2.1-1 所示。

课程编号	课程名称	课程时长
ed611951469	软件测试	48

图 3.2.1-1

课程列表的实现如下所示：

页面请求：

```

<a href="/course/getByStudentId?studentId=${sessionScope.student.stuId }
&returnPage=stu-courses"> <i class="fa fa-dashboard"></i> <span>我的课程</span></a>

```

Controller 层：

```

@RequestMapping("/getByStudentId")
public String getCoursesByStudentId(@RequestParam("studentId") String studentId,
    @RequestParam("returnPage") String returnPage, ModelMap modelMap) {
    //将课表数据写入视图
    List<TbCourse> list = courseService.getCoursesByStudentId(studentId);
    modelMap.put("courseList", list);
    //跳转至返回页面
    return returnPage;
}

```

Service 层：

```

@Override
public List<TbCourse> getCoursesByStudentId(String studentId) {
    //首先在选课记录当中查找
    TbTeacherCourseExample example = new TbTeacherCourseExample();
    //拼接 sql 语句
    example.setDistinct(true);
    Criteria criteria = example.createCriteria();
    criteria.andStudentIdEqualTo(studentId);

    List<TbCourse> courses = new ArrayList<TbCourse>();
}

```



```

//在选课记录中查到课程编号后，到课程列表中查找课程详细信息
List<TbTeacherCourse> list = teacherCourseMapper.selectByExample(example);
for (TbTeacherCourse tbTeacherCourse : list) {
    TbCourse course= courseMapper.
        selectByPrimaryKey(tbTeacherCourse.getCourseId());
    courses.add(course);
}
//去除重复元素
Set<TbCourse> courseSet = new HashSet<TbCourse>(courses);
courses = new ArrayList<>(courseSet);

return courses;
}

```

3.2.2 我的考试信息

当学生点击左侧“我的考试信息”时，系统将展示该学生需要参加的考试和已经完成的考试，如图 3.2.2-1 所示。

点击“进入考试”，即可开考，作答完成之后，可提交试卷，如图 3.2.2-2 所示。

考试时间	课程名称	任课教师	操作
2017-10-13 18:00:00	软件测试	赵单羽	考试已完成
2017-10-13 18:00:00	软件测试	赵单羽	进入考试
2017-10-13 18:00:00	软件测试	赵单羽	进入考试

图 3.2.2-1

选择题

如果某测试用例集实现了某软件的路径覆盖，那么它一定同时实现了该软件的
 (A) 判定覆盖 (B) 条件覆盖
 (C) 判定/条件覆盖 (D) 组合覆盖

单元测试中用来模拟被测模块调用者的模块是
 (A) 父模块 (B) 子模块
 (C) 驱动模块 (D) 桩模块

填空题

渐增式测试有两种不同的组装模块的方法：____和____。

论述与分析题

您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？

为什么要在一个团队中开展软件测试工作？

提交试卷

图 3.2.2-2

根据试题模板和考试难度生成试卷的实现如下所示：

页面请求：

```
<a href="/testPaper/generate?testId=${test.testId}">进入考试</a>
```

Controller 层：

```
@RequestMapping("/generate")
public String generateTestPaper(@RequestParam("testId") String testId, ModelMap modelMap)
{

    TbTest test = testService.getTestById(testId);
    //获取试题模板
    List<TbTemplateEntry> templateEntries = templateService.
        getEntries(test.getTemplateId());
    .....

    List<TbQuestionBankWithBLOBs> questions = null;

    for (TbTemplateEntry tbTemplateEntry : templateEntries) {

        questions = new ArrayList<TbQuestionBankWithBLOBs>();
        //获取题库中每种类型试题某一难度题目的总数
        int count = questionBankService.
            countByTypeDegreeAndCourse(tbTemplateEntry.getTemType(),
                test.getTestDegree(), test.getCourseId());
        //获取随机抽题编号
```

```

        List<Integer> list = PaperStrategyUtil.
generateRandomStrategy(tbTemplateEntry.getTemNum(), count);
        if (list != null && list.size() > 0) {
            for (Integer integer : list) {
                //根据随机生成的编号从数据库中抽题
                TbQuestionBankWithBLOBs question = questionBankService.
getByCourseDegreeAndType(test.getCourseId(),
test.getTestDegree(), integer, tbTemplateEntry.getTemType());
                questions.add(question);
            }
            modelMap.put("quesList" + tbTemplateEntry.getTemType(), questions);
        }

    }

    return "stu-test-pojo";
}

```

Service 层:

```

@Override
public TbQuestionBankWithBLOBs getByCourseDegreeAndType(String courseId, Integer
degree, Integer seqNum, Integer type) {

    TbQuestionBankExample example = new TbQuestionBankExample();

    Criteria criteria = example.createCriteria();
    criteria.andCourseIdEqualTo(courseId);
    criteria.andQueDegreeEqualTo(degree);
    criteria.andQueTypeEqualTo(type);
    //通过分页插件设置页大小为 1，再输入题目序号，即可从数据库抽出第某一个题
    PageHelper.startPage(seqNum, 1);
    List<TbQuestionBankWithBLOBs> list = questionBankMapper.
selectByExampleWithBLOBs(example);
    if (list != null && list.size() > 0) {
        list.get(0).setQueAnswer(null);
        return list.get(0);
    }

    return null;
}

```

Aspectj:

```

/**为确保多人并发时站点性能，在抽取题目时，将从数据库抽取出来的题目写入 redis 缓存，作为 nosql 的内存数据库，后续考生若抽取到同样的题目，直接从该缓存中取出，提高效率。*/

/*通过 Spring AOP 实现，通知函数中的切点函数为 Service 层实现的函数*/

```

```
@Around("execution(*com.etsys.core.service.QuestionBankService.getByCourseDegreeAndType(..))")
public TbQuestionBankWithBLOBs around(ProceedingJoinPoint joinPoint) {

    TbQuestionBankWithBLOBs question = null;
    .....
    try {
        //从 redis 缓存中取出题目
        String json = jedisClient.get(strbuf.toString());

        if (!StringUtils.isBlank(json)) {
            question = JsonUtils.jsonToPojo(json, TbQuestionBankWithBLOBs.class);
            if (question != null) {
                //如果从缓存中找到，直接返回
                return question;
            }
        }

    } catch (Exception e1) {
        //如果发生异常，则不能影响正常业务
        e1.printStackTrace();
    }

    try {
        //如果未从 redis 找到，则执行被切函数
        question = (TbQuestionBankWithBLOBs) joinPoint.proceed();

    } catch (Throwable e) {
        e.printStackTrace();
    }

    try {
        //直接将抽取到的题目写入 redis 缓存
        jedisClient.set(strbuf.toString(), JsonUtils.objectToJson(question));
        jedisClient.expire(strbuf.toString(), 600);

    } catch (Exception e) {
        e.printStackTrace();
    }

    return question;
}
```

3.2.3 成绩查询

当学生点击左侧“成绩查询时”，即可查询已批改试卷的每种题型的成绩，如下图 3.2.3-1 所示：

题型	得分
选择题	0.0
填空题	6.0
判断题	0.0
简答题	12.0
分析与论述题	20.0

图 3.2.3-1

成绩查询的实现如下所示：

```
<a href="/score/getByTestAndStudent?testId=${score.testId }&studentId=
${sessionScope.student.stuId }&returnPage=stu-score-detail">${score.testId }</a>
```

Controller 层：

```
@RequestMapping("/getByTestAndStudent")
public String getByTestAndStudent(@RequestParam String testId, @RequestParam String
studentId, @RequestParam String returnPage, ModelMap modelMap) {

    List<TbScore> scores = scoreService.getByTestAndStudent(testId, studentId);

    modelMap.put("scores", scores);
    return returnPage;
}
```

Service 层：

```
@Override
public List<TbScore> getByTestAndStudent(String testId, String studentId) {

    TbScoreExample example = new TbScoreExample();

    com.etsys.orm.pojo.TbScoreExample.Criteria criteria = example.createCriteria();
    criteria.andTestIdEqualTo(testId);
    criteria.andStuIdEqualTo(studentId);

    List<TbScore> list = scoreMapper.selectByExample(example);
    if (list != null && list.size() > 0) {
        return list;
    }

    return null;
}
```

4 系统测试

4.1 教师端功能测试

4.1.1 注册/登录

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010101	注册测试	教师用户名 /密码	数据库写入 该教师信息	数据库存在 该教师信息	是
test_v1_010102	登录测试	教师用户名 /密码	跳转至教师 首页	跳转至教师 首页	是

teacher_id	teacher_password	teacher_name
ins133373359	e5256036ab2c2bddc7ca2a26c706bf36	龚开梦
ins158045163	43fec21056f933d30202ed654bda4df5	王好隐
ins315927474	0c285df2f3ec0e9240cbaff401399454	赵单羽

图 4.1.1-1 注册测试结果

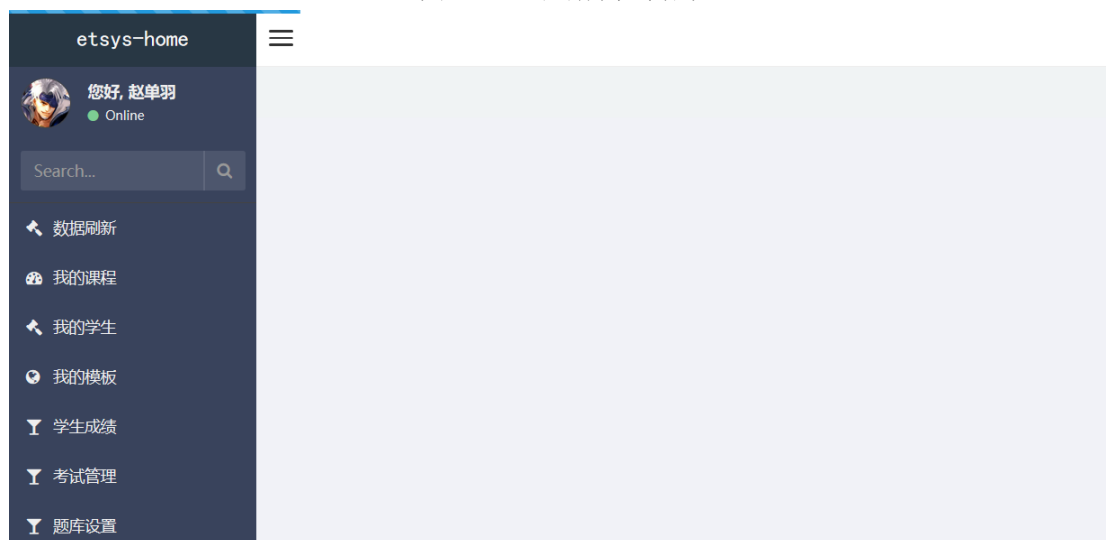


图 4.1.1-2 登录测试结果

4.1.2 我的课程

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010201	课程列表 测试	教师点击 “我的课 程”链接	页面展示该 名教师的课 程	页面展示该 名教师的课 程	是

课程编号	课程名称	课程时长
ed888855008	人机交互设计	16
ed611951469	软件测试	48
数据有误？点我刷新		

图 4.1.2-1 课程列表测试结果

4.1.3 我的学生

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010301	学生列表测试	教师点击“我的学生”链接	页面展示该名教师的学生	页面展示该名教师的学生	是

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48
数据有误？点我刷新		
学生编号	学生姓名	所在班级
stu76961901	李钢	软件工程410
stu83828117	李为敏	软件工程410
stu85567247	李辛华	软件工程410
stu86723381	李事	软件工程410
stu86799281	李一钰	软件工程410
stu89745219	李雨充	软件工程410
stu92391992	李壁遥	软件工程410
stu94266884	李奕杨	软件工程410
stu94874685	李长宝	软件工程410
stu95943117	李诗晴	软件工程410

图 4.1.3-1 学生列表测试结果

4.1.4 我的模板

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010401	查看模板测试	教师选择某一门课程查看	该门课的所有模板和任一模板详情	该门课的所有模板和任一模板详情	是
test_v1_010402	新增模板测试	教师选择“新增模板”	数据库中写入该模板	数据库中出現该模板	是

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

该门课程的试题模板信息如下：

模板编号	课程编号	教师编号	创建时间
tem150782493782212	ed611951469	ins315927474	2017-10-13
tem150782591206602	ed611951469	ins315927474	2017-10-13
tem150842054006467	ed611951469	ins315927474	2017-10-19

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

该门课程的试题模板详细信息如下：

题目类型	题目数量	题目分值
选择题	2	3.0
填空题	3	3.0
判断题	2	1.0
简答题	3	5.0
论述与分析题	2	10.0

图 4.1.4-1 查看模板测试结果

🔑 tem_id	🔑 course_id	🔑 teacher_id	created
tem150782493782212	ed611951469	ins315927474	2017-10-13 00:15:38
tem150782591206602	ed611951469	ins315927474	2017-10-13 00:31:52
tem150813248263278	ed888855008	ins315927474	2017-10-16 13:41:23
tem150842054006467	ed611951469	ins315927474	2017-10-19 21:42:20

🔑 tem_entry_id	🔑 tem_id	🔑 tem_type	tem_num	tem_score
1	tem150782493782212	1,000	2	3
2	tem150782493782212	1,001	3	3
3	tem150782493782212	1,002	2	1
4	tem150782493782212	1,003	3	5
5	tem150782493782212	1,004	2	10

图 4.1.4-2 新增模板测试结果

4.1.5 考试管理

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010501	发布考试测试	教师输入一次考试信息	数据库写入本次考试信息	数据库出现本次考试信息	是
test_v1_010502	查看考	教师选择“查	页面展示考	页面展示考	是

	试测试	看考试”	试信息	试信息	
test_v1_010503	批改试 卷测试	教师批改试卷 后提交	数据库写入 该学生成绩	数据库出现 该学生成绩	是
 test_id	created	 course_id	 teacher_id	 template_id	test_degree
test150787359809008	2017-10-13 18:00:00	ed611951469	ins315927474	tem150782493782212	3
test150813471087460	2017-10-13 18:00:00	ed611951469	ins315927474	tem150813248263278	3
test150821079468250	2017-10-13 18:00:00	ed611951469	ins315927474	tem150782493782212	3

图 4.1.5-1 发布考试测试结果

课程编号	课程名称	课程时长			
ed611951469	软件测试	48			
考试编号	考试时间	课程编号	教师编号	模板引用	试卷难度
test150787359809008	2017-10-13 18:00:00	ed611951469	ins315927474	tem150782493782212	3
test150813471087460	2017-10-13 18:00:00	ed611951469	ins315927474	tem150813248263278	3
test150821079468250	2017-10-13 18:00:00	ed611951469	ins315927474	tem150782493782212	3

图 4.1.5-2 查看考试测试结果





 sco_id	 test_id	 stu_id	sco_mark	 sco_type
sco150805494446830	test150787359809008	stu18250900	0	1,000
sco150805494448396	test150787359809008	stu18250900	0	1,002
sco150805657554548	test150787359809008	stu22056972	0	1,000
sco150805657657158	test150787359809008	stu22056972	1	1,002
sco150807631488915	test150787359809008	stu18250900	12	1,003
sco150807631488927	test150787359809008	stu18250900	20	1,004
sco150807631488933	test150787359809008	stu18250900	6	1,001
sco150813480213926	test150787359809008	stu22056972	12	1,004
sco150813480214047	test150787359809008	stu22056972	2	1,003
sco150813480214803	test150787359809008	stu22056972	6	1,001
sco150813480214806	test150787359809008	stu22056972	0	1,002

图 4.1.5-3 批改试卷测试结果

4.1.6 题库设置

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_010601	查看题库测试	教师选择某一门课的题库	页面展示题库题目	页面展示题库题目	是
test_v1_010602	增加题目测试	教师点击“增加题目”	数据库写入该题目信息	数据库出现该题目信息	是

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

增加题目

该门课程的题库信息如下：

题目编号	题目类型	题目难度	题目状态	课程编号	题目内容
que150787787747448	1002	3	1	ed611951469	查看
que150787787853670	1002	3	1	ed611951469	查看
que150787787879269	1002	3	1	ed611951469	查看

该门课程信息如下：

课程编号	课程名称	课程时长
ed611951469	软件测试	48

该题目信息如下：

题目编号	题目类型	题目难度	题目状态	课程编号
que150791123962179	1004	3	1	ed611951469

详细信息如下：

问题描述	答案描述
您在以往的测试工作中都曾经具体从事过哪些工作？其中最擅长哪部分工作？	我曾经做过web测试，后台测试，客户端软件，其中包括功能测试，性能测试，用户体验测试。最擅长的是功能测试

图 4.1.6-1 查看题库测试结果

que_id	que_type	que_degree	que_content	que_answer	que_state	course_id
que150787961824685	1,001	3	运行被测程序的方法称为_____测试。	动态	1	ed611951469
que150787961855953	1,001	3	在单元测试中，测试一个模块时，需要设计_____和_____。	驱动模块、桩模块	1	ed611951469
que150787984916325	1,003	3	请简述软件测试活动的生命周期？	软件从进入测试到退出测试的过程中，所要经历的...	1	ed611951469
que150787991942329	1,003	3	软件的缺陷等应如何划分？	致命错误，可能导致本模块以及其他相关模块异常...	1	ed611951469
que150787997623307	1,003	3	什么是V模型？简述V模型在软件测试过程中的作用...	V模型是软件开发瀑布模型的变种，它反映了测试活...	1	ed611951469
que150788014926264	1,003	3	软件测试应该划分几个阶段？简述各个阶段应重点测试...	大体上未说可分为单元测试、集成测试、系统测试、验...	1	ed611951469
que150788023796229	1,003	3	你认为一个优秀的测试工程师应具备哪些素质？	1、具有良好的计算机编程基础+2、具有创新精神 and ...	1	ed611951469
que150788027243247	1,003	3	什么是软件缺陷？请简述软件缺陷出现的原因。	存在于软件之中的那些不希望，或不可接受的偏差...	1	ed611951469
que150788563714811	1,000	3	在软件生命周期中，哪一个阶段，软件缺陷修复费用...	A	1	ed611951469
que150788569249417	1,000	3	单元测试中用来模拟被测模块调用者的模块是 (...	C	1	ed611951469
que150788574131112	1,000	3	为了提高测试的效率，应该 (A) 随机地选...	D	1	ed611951469
que150788576850240	1,000	3	侧重于观察资源耗尽情况下的软件表现的系统测试...	B	1	ed611951469
que150788578287506	1,000	3	必须要求用户参与的测试阶段是 (A) 单元测...	D	1	ed611951469
que150788599535215	1,000	3	软件测试员究竟做些什么？ (A) 软件测试员的...	C	1	ed611951469
que150788601947163	1,000	3	下面四种说法中正确的是 (A) 因果图法是建...	C	1	ed611951469
que150788604524409	1,000	3	不属于单元测试内容的是 (A) 模块接口测试...	A	1	ed611951469
que150788606885743	1,000	3	划分软件测试属于白盒测试还是黑盒测试的依据是 ...	C	1	ed611951469
que150788608888821	1,000	3	下列项目中不属于测试文档的是 (A) 测试计...	C	1	ed611951469
que150788612224584	1,000	3	几乎没有产品计划、进度安排和正规的开发过程的...	A	1	ed611951469
que150788613612759	1,000	3	如果某测试用例集实现了某软件的路径覆盖，那么...	A	1	ed611951469
que150788615934506	1,000	3	下列说法不正确的是 (A) 测试不能证明软件...	D	1	ed611951469
que150788617716752	1,000	3	对Web网站进行的测试中，属于功能测试的是 ...	B	1	ed611951469
que150788619451906	1,000	3	在进行单元测试时，常用的方法是 (A) 采用...	B	1	ed611951469
que150788624720558	1,000	3	使用白盒测试方法时，确定测试数据的依据是指定...	B	1	ed611951469
que150788626158228	1,000	3	下列_____不是软件自动化测试的优点 (A)速度快...	D	1	ed611951469
que150788629512738	1,000	3	配置测试 (A) 是指检查软件之间是否正确交互和...	C	1	ed611951469
que150788630967959	1,000	3	下列各选项中_____不是一个测试计划所应包含的内容 ...	D	1	ed611951469
que150788632946523	1,000	3	下列不属于正式审查的方式是 (A)同事审查 (...	B	1	ed611951469
que150791120288254	1,004	3	为什么要在一个团队中开展软件测试工作？	因为没有经过测试的软件很难在发布之前知道该款...	1	ed611951469
que150791123962179	1,004	3	您在以往的测试工作中都曾经具体从事过哪些工作...	我曾经做过web测试，后台测试，客户端软件，其中...	1	ed611951469

图 4.1.6-1 增加题目测试结果

4.1.7 数据刷新

用例编号	用例名	输入值	实际输出值	期望输出值	是否符合预期
------	-----	-----	-------	-------	--------

	称				
test_v1_010701	数据刷新测试	教师点击“数据有误？点我刷新”按钮	数据库写入课程/学生/选课记录	数据库出现课程/学生/选课记录	是

🔑 course_id	course_name	course_time
ed611951469	软件测试	48
ed888855008	人机交互设计	16
ed968502086	概率论与数理统计	64

🔑 stu_id	stu_name	stu_class	stu_password
stu16900887	杨舟察	软件工程402	9f4ff7853245294a1867ddbba3372d3b
stu17335875	陈宏子	软件工程412	396aa894615412af9599756614387f7c
stu18250900	李开心	软件工程410	12a9dfae5c43d5056b20fc289c68147b
stu18484285	范志坚	软件工程412	ed2fd2d9dad3361de3ba33750722421a
stu21360849	李均华	软件工程412	9ff41008abd9df9ebc46211bc03ca209
stu21987790	李锐汉	软件工程402	fc477ee8af3eaf04ffaafec459ebc6eb
stu22056972	李子跃	软件工程410	d5fcd5b13fb0aa4fedad52e76917c310
stu23774156	李海必	软件工程410	9d8e16f53528e6d7227008aba78a5c72
stu24090196	彭玉林	软件工程412	4ccb5b9a27e9fc22c52209525774ca0b
stu24786390	呈安占	软件工程402	2875d6802151f9e62aa15511f1c2c81d

🔑 teacher_id	🔑 course_id	🔑 student_id
ins158045163	ed968502086	stu17335875
ins158045163	ed968502086	stu18484285
ins158045163	ed968502086	stu21360849
ins158045163	ed968502086	stu24090196
ins158045163	ed968502086	stu25110770
ins158045163	ed968502086	stu28401942
ins158045163	ed968502086	stu28758083
ins158045163	ed968502086	stu34911683
ins158045163	ed968502086	stu35698661
ins158045163	ed968502086	stu37391868
ins158045163	ed968502086	stu39502321
ins158045163	ed968502086	stu41157812

图 4.1.7-1 数据刷新测试结果

4.2 学生端功能测试

4.2.1 我的课程

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_020101	课程列表测试	学生点击“我的课程”	页面展示课程信息	页面展示课程信息	是

课程编号	课程名称	课程时长
ed611951469	软件测试	48

图 4.2.1-1 课程列表测试结果

4.2.2 考试信息查询

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_020201	查看考试测试	学生点击“查看考试信息”	考试信息列表	考试信息列表	是
test_v1_020202	进入考试测试	学生点击“进入考试”	考试试卷	考试试卷	是
test_v1_020203	提交试卷测试	学生提交试卷	数据库写入该试卷	数据库出现该试卷	是

考试时间	课程名称	任课教师	操作
2017-10-13 18:00:00	软件测试	赵单羽	考试已完成
2017-10-13 18:00:00	软件测试	赵单羽	进入考试
2017-10-13 18:00:00	软件测试	赵单羽	进入考试

图 4.2.2-1 查看考试测试结果

选择题

下列__不是软件自动化测试的优点
(A)速度快、效率高 (B)准确度和精确度高
(C)能提高测试的质量 (D)能充分测试软件

划分软件测试属于白盒测试还是黑盒测试的依据是
(A)是否执行程序代码
(B)是否能看到软件设计文档
(C)是否能看到被测源程序
(D)运行结果是否确定

填空题

验证测试在模拟环境下运用__测试方法，由__和__参加的测试。

软件缺陷产生的原因包括__、__、__以及其他原因。

软件测试过程中需要3类信息：__、__、__。

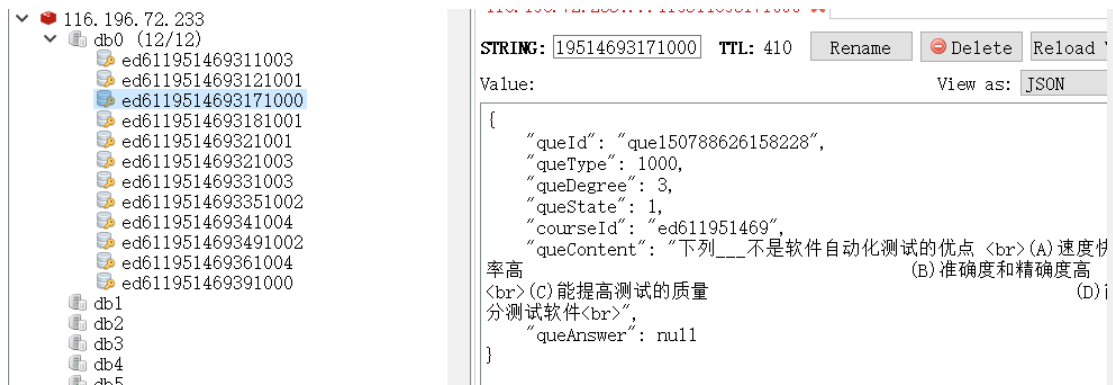


图 4.2.2-2 进入考试（redis 缓存写入题目）测试结果

tp_id	tp_content	tp_type	tp_state	stu_id	course_id	test_id
tp150805494418608	[{"questionId": "que150787788957333", "content": "...", "q...	1,002	4	stu18250900	ed611951469	test150787359809008
tp150805494418636	[{"questionId": "que150787961644707", "content": "...123"}...	1,001	4	stu18250900	ed611951469	test150787359809008
tp150805494418660	[{"questionId": "que150788632946523", "content": "...", "q...	1,000	4	stu18250900	ed611951469	test150787359809008
tp150805494418676	[{"questionId": "que150788023796229", "content": "...", "q...	1,003	4	stu18250900	ed611951469	test150787359809008
tp150805494421828	[{"questionId": "que150791134839873", "content": "...", "q...	1,004	4	stu18250900	ed611951469	test150787359809008
tp150805657517040	[{"questionId": "que150787961674886", "content": "...12", "q...	1,001	4	stu22056972	ed611951469	test150787359809008
tp150805657517078	[{"questionId": "que150788599535215", "content": "...", "q...	1,000	4	stu22056972	ed611951469	test150787359809008
tp150805657517090	[{"questionId": "que150787788188494", "content": "...1", "q...	1,002	4	stu22056972	ed611951469	test150787359809008
tp150805657518646	[{"questionId": "que150791134839873", "content": "...23", "q...	1,004	4	stu22056972	ed611951469	test150787359809008
tp150805657518654	[{"questionId": "que150787991942329", "content": "...12312...", "q...	1,003	4	stu22056972	ed611951469	test150787359809008
tp150813497397594	[{"questionId": "que150787788344466", "content": "...", "q...	1,002	4	stu23774156	ed611951469	test150787359809008
tp150813497397647	[{"questionId": "que150788569249417", "content": "...", "q...	1,000	4	stu23774156	ed611951469	test150787359809008
tp150813497400846	[{"questionId": "que150791127559358", "content": "...", "q...	1,004	2	stu23774156	ed611951469	test150787359809008
tp150813497401074	[{"questionId": "que150787961254842", "content": "...", "q...	1,001	2	stu23774156	ed611951469	test150787359809008
tp150813497401234	[{"questionId": "que150788014926264", "content": "...", "q...	1,003	2	stu23774156	ed611951469	test150787359809008
tp150815257381513	[{"questionId": "que150788632946523", "content": "...", "q...	1,000	4	stu31203869	ed611951469	test150787359809008
tp150815257381953	[{"questionId": "que150787788265632", "content": "...", "q...	1,002	4	stu31203869	ed611951469	test150787359809008
tp150815257382300	[{"questionId": "que150787961499085", "content": "...", "q...	1,001	2	stu31203869	ed611951469	test150787359809008
tp150815257382774	[{"questionId": "que150788014926264", "content": "...", "q...	1,003	2	stu31203869	ed611951469	test150787359809008
tp150815257382843	[{"questionId": "que150791120288254", "content": "...", "q...	1,004	2	stu31203869	ed611951469	test150787359809008

图 4.2.2-3 提交试卷测试结果

4.2.3 成绩查询

用例编号	用例名称	输入值	实际输出值	期望输出值	是否符合预期
test_v1_020301	成绩查询测试	学生点击“成绩查询”按钮	成绩列表	成绩列表	是

考试编号	学生编号	课程名称
test150787359809008	stu18250900	软件测试

题型	得分
选择题	0.0
填空题	6.0
判断题	0.0
简答题	12.0
分析与论述题	20.0

图 4.2.3-1 成绩查询测试结果

5 项目总结

5.1 技能收获

本次项目通过对 etsys 在线考试系统的开发，在 Java Web 的开发技能上，主要有以下几点收获：

- (1) Java 框架运用：Spring+SpringMVC+Mybatis。
- (2) 前端手段运用：Html5+CSS3+Javascript+JQuery+Ajax+Bootstrap。
- (3) Linux/服务器：CentOS 6.5 搭建 MySQL+Redis 缓存+JDK1.8+Tomcat 运行环境。
- (4) 项目搭建/集成/部署：Apache-Maven-3.5。
- (5) Web 服务：包括发布和调用教务系统数据接口。

5.2 团队收获

本次项目开发的团队合作采取“一带一”的合作方式，让小组成员“在教中学，在学中教”，每个人都贡献自己的力量，共同进步，真正实现“不坐车”的团队合作。