



闽江学院

本科毕业论文(设计)

题 目 基于 MVVM 的学生网上自主
选课系统的设计与实现

学生姓名 欧鹏程

学 号 5177101314

系 别 软件学院

年 级 2017 级

专 业 软件工程

指导教师 朱婧

职 称 副教授

完成日期 2019 年 4 月 22 日

闽江学院毕业论文（设计）诚信声明书

本人郑重声明：

兹提交的毕业论文（设计）《基于 MVVM 的学生网上自主选课系统的设计与实现》，是本人在指导老师 朱婧 的指导下独立研究、撰写的成果；论文（设计）未剽窃、抄袭他人的学术观点、思想和成果，未篡改研究数据，论文（设计）中所引用的文字、研究成果均已在论文（设计）中以明确的方式标明；在毕业论文（设计）工作过程中，本人恪守学术规范，遵守高校有关规定，依法享有和承担由此论文（设计）产生的权利和责任。

声明人（签名）：

年 月 日

摘 要

近年来，各个高校的学生数量在不断的增加。针对各个方面的人才，各高校开设出不同的课程体系并且为其配备了教学资源。为了解决了高校学生的选课问题，于是诞生了学生网上自主选课系统。对比传统的选修方案，该系统解决了以往课程选修的效率低，保密性差等问题。同时以推崇公选课为主，学生可以在该系统上可以选修到自己喜欢的课程，同时也能看到授课的教师，以及相关的课程简介。旨在提升高校学生的素质教养或者个人的能力。

该系统采用了前后端分离的技术，通过 API 请求方式将内容显示在网页上。该系统的前端采用 MVVM 的设计模式，后端的各个 API 的设计皆有权限限制，不允许非法权限进行数据操作。更加安全、稳定、可靠是该系统在部署至服务器之后主要体现的特点。

关键词： 公选课；前后端分离；MVVM；自主选课；

Abstract

In recent years, the number of students in various universities has been increasing. For all aspects of talent, each university has a different curriculum system and is equipped with teaching resources. In order to solve the problem of college students choosing courses, the students online self-selection course system was born. Compared with the traditional elective program, the system solves the problems of low efficiency and poor confidentiality of previous course electives. At the same time, the main course is to promote the public elective course, students can choose to take the course they like, and also can see the teachers who teach, and the related course introduction. It aims to improve the quality of college students education or personal ability.

The system uses a separation the front-end and back-end to display content on a web page via API request. The front end of the system adopts the design mode of MVVM, and the design of each API of the back end has permission restrictions, and does not allow illegal permissions for data operations. More secure, stable, and reliable is the main feature of the system after it is deployed to the server.

Key words: public elective course; separation the front-end and back-end; MVVM; self-selected courses;

目 录

1. 引言	1
1.1. 选题背景	1
1.2. 研究目的和意义	1
1.3. 国内外现有状况分析	1
2. 开发环境以及系统框架概述.....	3
2.1. 系统环境概述	3
2.2. 系统开发框架概述	3
2.3. 前后端分离技术概述	3
2.4. 什么是 MVVM 设计模式.....	4
2.5. MVVM 设计模式的特点.....	5
3. 需求分析和功能模块分析	6
3.1. 可行性分析	6
3.2. 系统需求分析	6
3.3. 系统全景图以及用例图	8
3.4. 系统功能模块分析	10
4. 系统概要设计	11
4.1. 系统概要设计	11
4.2. 系统前端功能模块设计	11
4.3. 系统后端 API 设计	12
4.4. 系统数据库设计	13
5. 系统详细设计与实现	17
5.1. 系统整体原型设计	17
5.2. 系统后端数据库连接配置	18
5.3. 权限验证模块	19
5.4. 个人信息管理模块	22
5.5. 信息录入管理模块	24
5.6. 学生选课模块	26
6. 单元测试和功能测试	30
6.1. 测试概述	30
6.2. 单元测试	30

6.3. 功能测试	31
6.4. 测试报告分析	32
6.5. 测试结果总结	33
7. 总结以及展望	34
7.1. 总结	34
7.2. 展望	34
参考文献.....	35
致 谢.....	36

1. 引言

1.1. 选题背景

随着无纸化办公的逐步实现，高校的公选课也逐步走向在线的自主选课方式。以往的公选课都是通过纸质的形式进行选修，该选修的方式存在着泄密性、安全性低，效率低，不能适应教学管理的需求等问题，而学生自主选课打破了原有的行政班级管理体制，扩大了学生的自由时间和空间，教学管理更加人性化、科学化和规范化^[1]。运用学生网上自主选课系统进行辅助选课，既能对大量的选课数据进行浏览校验并且进行统计学生选课情况又能直观的看到学生的选课和成绩情况，这也能促使高校与教师开设出更加适合学生进行学术研究的公选课，还避免了在人工处理期间可能发生的错误。使用学生网上自主选课系统也是为了完善学生自主选课机制，更能激发学生的兴趣爱好与学习的热情，不仅促使学生内在潜能的开发，还促使高校更能了解到学生的兴趣爱好，这样就能高质量的教育出优秀的学生。

1.2. 研究目的和意义

高校学生在进行公选课选修时，总会想着选择自己喜欢方向的课程或者想着选择自己喜欢的教师开设的课程，这样不仅对于学生还是对于教师都是有利的，学生方面可以选到自己感兴趣的课程这样就能更加上心的学习，教师方面也能将自己的授课知识全部传授给与学生。

在此期间，将通过对网站开发和前端分离技术的研究与学习来掌握 Web 应用程序网站的开发，并习惯于框架的各种方法调用。掌握 Model-View-ViewModel (MVVM) 与 Model-View-Controller (MVC) 的设计模式^{[2][3]}。在结合基于 MVVM 设计模式的前端框架 Vue.js，基于 MVC 设计模式的后端框架 Egg.js，针对高校学生公选课的体系，进行系统的设计与实现，通过该系统能够了解到前后端分离的技术，同时也能了解掌握 MVVM 和 MVC 的设计模式，熟练的掌握基于 node.js 的框架 Vue.js 和 Egg.js。

1.3. 国内外现有状况分析

目前，高校公选课存在着缺乏科学合理公选课课程体系，缺乏稳定优质师资队伍，学生选课存在娱乐性、盲目性，缺乏完善质量管理体系等问题^[4]。学分制的制度是我国高校普遍存在的一个教学制度，在这样的制度下，高校开设的公选课，学生皆可以进行自由选择，但是由于受一些校方的政策或者一些制约因素，

选课的效果不是非常好，这也映射了学生选课存在娱乐性、盲目性这一问题。

国外教育研究软件比国内教育研究软件更早开发和成熟。20 世纪 70 年代末，美国建立了国家科学基金会网络（NSFNET），在线进展或主题进展报告在线进行。外国学生的在线教学管理系统也很完善。来自世界各地的学生可以坐在家里完成一系列活动，如学习，课程，考试，毕业论文和通过互联网获得学位。在国外的网上选课系统中，会有一份课程清单，里面详细地展示了自己需要学习的专业知识和一些其他科目的要求，为学生提供了一份极为重要的学习路线图，学生可以根据自己的职业发展来选择自己要进行选修的课程，可以说是一个人性化的选择。

2. 开发环境以及系统框架概述

2.1. 系统环境概述

学生网上自主选课系统前端采用 MVVM 设计模式的 Vue.js 框架进行开发，后端则是采用 Egg.js 框架进行打造一个符合 RESTful API 的设计规范接口，是一个前后端分离的 WEB 应用网站。

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效^[5]。Node.js 在 Adobe、阿里巴巴、腾讯、Facebook、Google、百度等公司都得到了应用。本设计采用的 Vue.js 和 Egg.js 两大框架都是基于 Node.js 的。

数据存储使用到的是 MySQL 关系型数据库管理系统。MySQL 具有功能强、使用简便、管理方便、运行速度快、安全可靠性强等优点，用户可利用许多语言编写访问 MySQL 数据库的程序^[6]。

2.2. 系统开发框架概述

Egg.js 是“阿里旗下产品”，它基于 Node.js 和 Koa，是基于 Es6，Es7 和 Typescript，Koa2 的 Node.js 企业应用程序开发框架。Egg.js 由 Koa.js 扩展而来，添加了多进程的支持，并且参考了 Ruby On Rails 的设计哲学，奉行“约定优于配置”原则，可以帮助团队和开发人员降低开发和维护成本。

Vue（读音/vju:/，类似于 view）是一套用于构建用户界面的渐进式框架，与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用^[7]。易用、灵活、高效是 Vue.js 具备的特点，易用上，可以通过阅读官方提供的文档即可开始构建单页面应用；灵活上，Vue.js 可以嵌套在任何一个前端库或者一个已经完整的框架上而不影响之前的功能；高效上，Vue.js 源代码经过压缩仅有 23KB 大小，拥有超快的虚拟 DOM、最省心的优化。

2.3. 前后端分离技术概述

在以往的 WEB 应用开发中，用于展示用户可以浏览到页面的浏览器部分被称作前端，而用于为前端应用提供业务逻辑和数据准备称作后端，所以久而久之浏览器就变成了程序员分离前后端的分界线。事实上，前后端分离不仅仅是一种开发模型，而是一种 WEB 应用程序的体系结构模型。在开发阶段，前后端工程师在约定好的接口规范下进行并行开发和测试。使用前后端分离的架构，可以为我们

带来一下四个改进：

1. 为优质产品创建精益团队。前后端工程师在他们的开发团队中使用前分离技术和后分离技术使他们更加专注于前端或后端开发。
2. 提升开发效率。在分离前端和后端之后，可以实现前端和后端代码的分离，应用所需接口以及接口参数在前后端开发人员的沟通约定下完成，之后就可以开始并行开发，无需等待对方的开发工作结束。
3. 完美地响应前端复杂多样的需求。在一个团队中，完成前后端分离的转型，开发人员可以专注于专业化，开发能力和效率将不可避免地提高，同时也能应对复杂多变的前端需求。
4. 增强代码可维护性。在分离前端和后端之后，应用程序的代码不再是前后混合，并且调用依赖项仅在运行时可用。

前端应用与后端应分离之后，此时，我们需要一个统一的机制来促进不同前端和后端之间进行通信。RESTful API 是目前比较成熟的一套互联网应用程序的 API 设计理论，前后端分离规范最典型的的就是 RESTful API 规范^[8]。在 RESTful 系统中，服务器使用 URI 公开资源，客户端使用多个 HTTP 动词（GET/POST/PUT/PATCH/DELETE）来访问资源，后端应用将数据返回给前端，前端对其进行对应的渲染即可。

2.4. 什么是 MVVM 设计模式

MVVM 模式是自 2005 年微软 WPF 和 Silverlight 架构师 John Gossman 在其博客中首次提出的，它利用了 WPF 的 XAML 和 DataBinding，使软件的架构可以分离成为 View、Model 和 ViewModel 3 层，使数据和界面得到很好的分离，达到软件设计中的“高内聚，低耦合”的目标^[9]。MVVM 设计模式的核心是，当 View（视图层）进行数据改变时，会自动将相关数据更新到 ViewModel（视图模型），反之也是如此。View（视图层）和 ViewModel（视图模型层）之间的联系是通过双向数据绑定（data-binding）进行建立的，View（视图层）的变化会自动更新到 ViewModel（视图模型层），ViewModel（视图模型层）的变化也会同步到 View（视图层）上显示，MVVM 的设计模式在 Vue.js 的设计上也得到了验证，如图 2-1 所示。

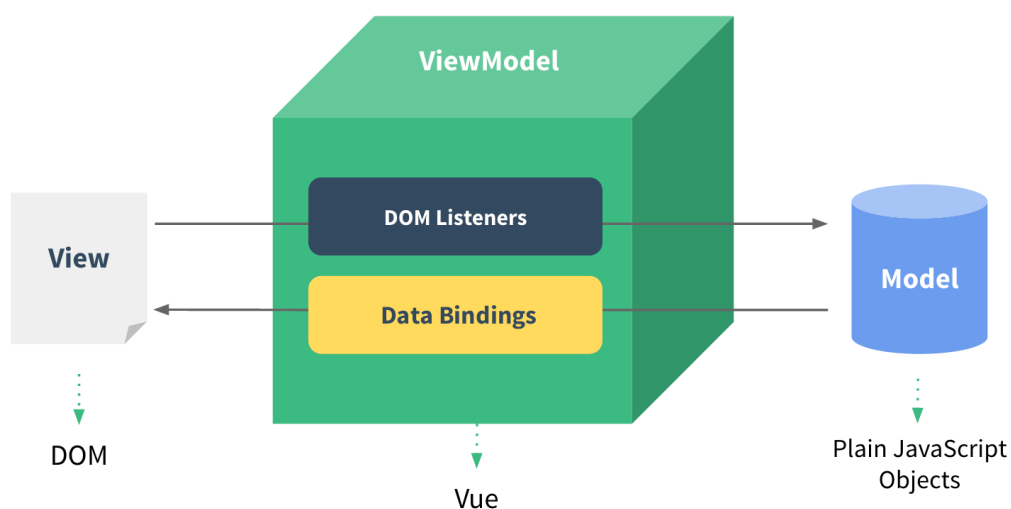


图 2-1 MVVM 在 Vue.js 中的应用

MVVM 的代表框架有：知名度相对比较低的 Knockout、早期的 Ember.js、目前比较火热的是来自 Google 的 AngularJS，还有现在的 Vue.js。目前国内比较知名的，基于 Vue.js 开发的网站有简书网、掘金网、哔哩哔哩视频弹幕网、手机搜狐网、奇趣网等。基于 Vue.js 开发的前端 UI 框架有 Element UI、Vant、Antd、Vux 等。所以 MVVM 设计模式的应用还算比较广的。

2.5. MVVM 设计模式的特点

MVVM 自推出以后，便得到了大多数开发人员的认可，下面列出一些 MVVM 设计模式的特点：

1. 低耦合：视图可以独立于模型进行更改和修改。ViewModel 可以链接到不同的视图上。可以在视图层更改时更改模型图层。可以在模型层更改时更改视图层。
2. 可重用性：一些视图的逻辑可以放入 ViewModel，其他视图可以通过调用该 ViewModel 来重用视图逻辑。
3. 独立开发：开发人员可以使用 MVVM 设计模式专注于业务逻辑和数据开发，界面的设计工作则是 UI 设计人员的主要关注方向。
4. 可测试：通常，很难运行界面的测试，但是在 MVVM 的设计模式下，可以使用 ViewModel 对界面进行接口性测试。

3. 需求分析和功能模块分析

3.1. 可行性分析

随着高校学生人数的迅速增加以及学生信息的爆炸式增长，在高校对学生信息管理的自动化和准确性的需求日益严格的背景下，无纸化办公的需求日益增长。高校的规模愈来愈大，学生的选课信息也愈来愈重要，学生网上自主选课系统就由此诞生。简单分析可得，学生网上自主选课系统的可行性有以下几点：

1. 技术可行性。学生网上自主选课系统的前后端开发都是基于 Node.js，前端框架 Vue.js 和后端框架 Egg.js 总体来说上手快，开发周期短，移植性快简单。
2. 操作可行性。学生网上自主选课系统功能简单易用，用户可以通过界面直接了解所需功能的操作方式。
3. 性能、效率可行性。该系统解决了传统的纸质选课方式的繁琐，用户通过该系统能够快速的录入数据、浏览数据等操作，该系统要求可以长时间运行而不会出现问题，并且可以快速响应学生选课所属的数据。
4. 数据安全性、保密性。系统为具有不同权限的用户，同时为其提供不同的功能模块。拥有特定权限的用户可以运营，普通用户只能执行查询操作。
5. 经济可行性。为了节约开发成本，更好地完成网站开发建设，该系统的开发都是使用开源的框架进行开发的，也能快速的部署到服务器上工用户来进行使用。

综上分析，学生网上自主选课系统可以进行开发，并且也是拥有足够的信心将该系统变成人人易用的一个应用。

3.2. 系统需求分析

经过对学生网上自主选课系统的简单需求分析可得，该系统需要设定了三种权限的角色：系统管理员角色（ADMIN）、学生角色（STUDENT）、教师角色（TEACHER）。每一种角色拥有不同的权限，当然登录系统可进行的操作也是不一致的。

- 1) 系统管理员角色（ADMIN）：维护整个系统，管理各种的信息（包含个人信息、院系信息、专业信息、课程信息、学生信息、教师信息）等。在每一个阶段，系统管理员进行的任务皆是不同的，阶段任务如表 3-1 所示。

表 3-1 系统管理员的阶段任务

阶段	任务
前期准备	1. 院系信息的录入 2. 专业信息的录入 3. 学生信息的录入 4. 教师信息的录入
开课准备	1. 校验学生的信息 2. 检验教师的信息 3. 通知教师进行开课或者由系统管理员向教师分配教程
课程选修	1. 检验课程信息 2. 通知学生进行选课或者由学生自主选课
课程考核	1. 校验学生的选课信息 2. 通知教师进行课程成绩发布或者由系统管理员进行成绩发布
课程结束	1. 通知学生可以进行成绩查询或者学生自主进行课程成绩的查询

2) 教师角色 (TEACHER)：维护课程信息，进行课程的开设和关闭，能够进行课程成绩的发布等操作。在每一个阶段，教师进行的任务皆是不同的，阶段任务如表 3-2 所示。

表 3-2 教师的阶段任务

阶段	任务
前期准备	1. 课程信息的录入
开课准备	1. 校验课程的信息 2. 发布课程信息 3. 通知学生进行选课或者由系统管理员向教师分配教程
课程选修	1. 检验学生选课信息
课程考核	1. 校验学生的选课信息 2. 通知学生进行课程考核
课程结束	1. 通知学生可以进行成绩查询或者学生自主进行课程成绩的查询

3) 学生角色 (STUDENT)：能够进行课程信息的浏览，进行课程的选修或退选。在每一个阶段，学生进行的任务皆是不同的，阶段任务如表 3-3 所示。

表 3-3 学生的阶段任务

阶段	任务
前期准备	1. 县教师了解课程信息
开课准备	1. 季羨林课程信息的浏览
课程选修	1. 选择自己喜欢的课程 2. 允许学生在课程成绩没有发布前进行退选
课程考核	1. 进行线上或者线下课程考核
课程结束	1. 学生可以进行成绩查询

3.3. 系统全景图以及用例图

根据系统的定义，系统管理员权限的用户主要任务是维护整个系统的，而教师权限的用户在该系统中主要提供课程的信息，学生则是进行课程的选修和退选，学生网上自主选课系统全景图如图 3-1 所示。

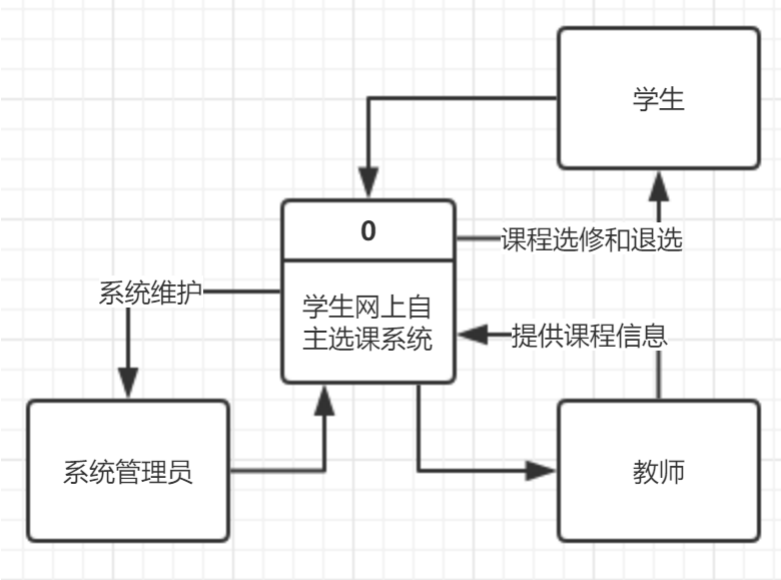


图 3-1 学生网上自主选课系统全景图

系统管理员权限用户可以登录系统，修改个人信息和登录密码，能够对院系、专业、课程等信息进行管理操作，对应的用例图如图 3-2 所示。

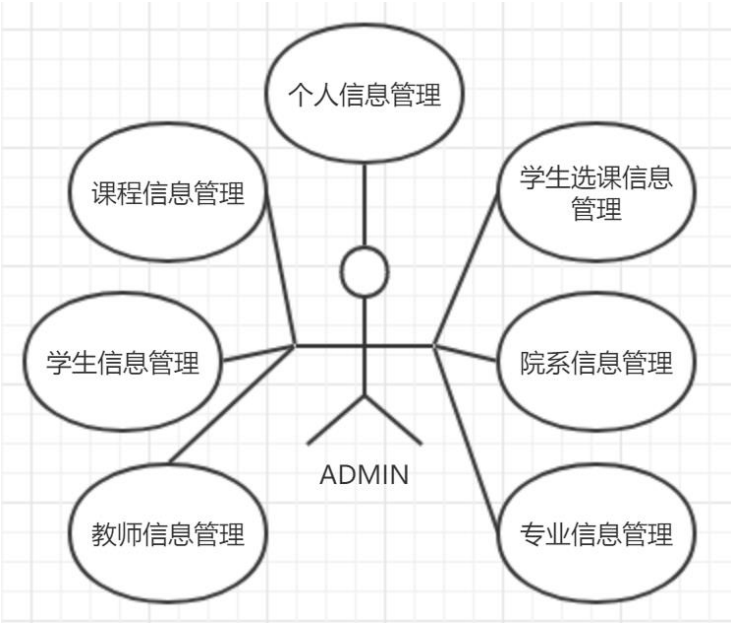


图 3-2 系统管理员权限用户的用例图

学生权限用户登录系统后能够对个人信息、登录密码的修改，学生能够进行的操作只有课程的浏览、选修、退选、成绩查询，对应的用例图如图 3-3 所示。

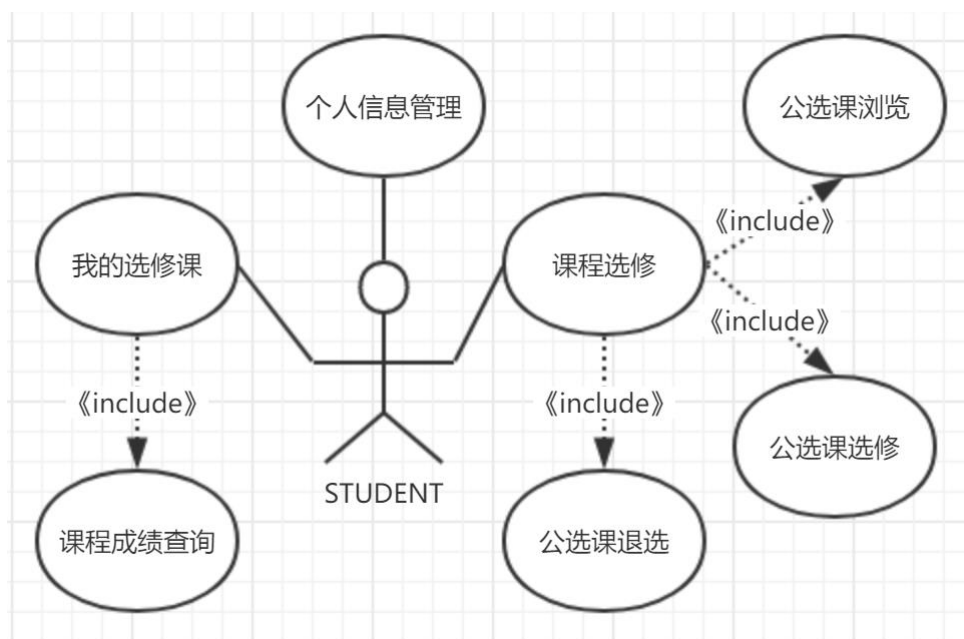


图 3-3 学生权限用户的用例图

教师权限用户登录系统后能够对个人信息的管理，包含个人信息的修改和登录密码的修改，还能够进行课程的开设或者关闭、成绩信息的发布等操作，对应的用例图如图 3-4 所示。

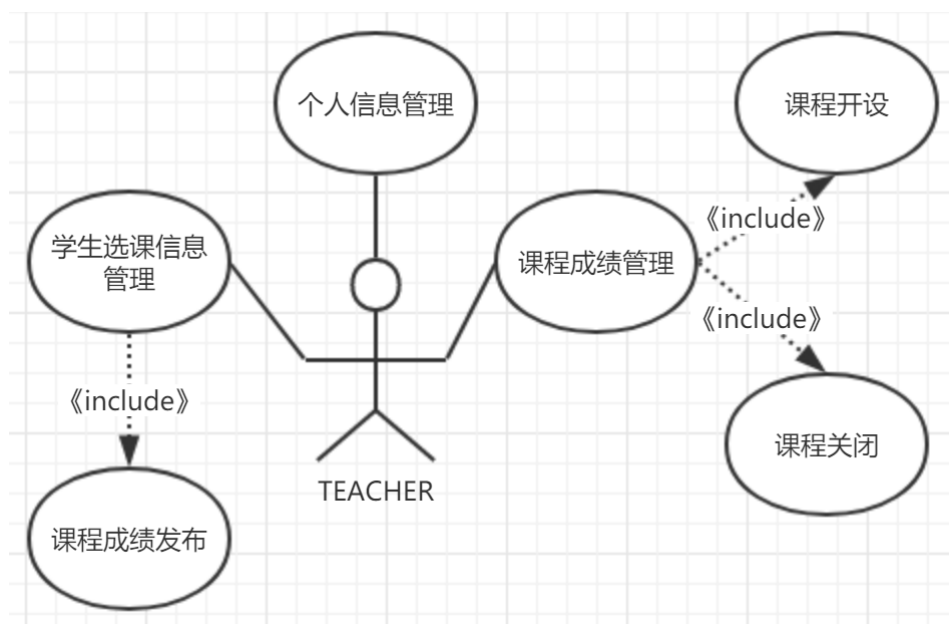


图 3-4 教师权限用户的用例图

3.4. 系统功能模块分析

学生网上自主选课系统设计了三种权限的用户，根据这三种权限的用户登录该系统能够进行不同的操作，将系统大致划分为权限验证模块（三种权限用户皆可以从此进行登录操作）、个人信息管理模块（提供了统一的界面，三种权限用户皆可修改自己的个人信息、登录密码等）、信息录入管理模块（包含院系信息、专业信息、课程信息、教师信息、学生信息的管理等）、学生选课模块（包含学生的选课中心、课程成绩发布等）。系统管理员权限用户登录系统可以对院系信息、专业信息、课程信息、学生信息、教师信息等进行录入、浏览、删除等操作。学生权限用户登录系统可以进行公选课的浏览、选修、退选等操作，可以进行课程成绩查询。教师权限用户登录系统以进行课程开设和关闭，也可以进行课程成绩的发布等操作。

4. 系统概要设计

4.1. 系统概要设计

由以上系统功能模块分析可得，该学生网上选课系统中的权限验证模块提供了三种权限用户的统一登录验证，在权限验证模块中提供了前端路由的生成和前端菜单的生成子功能。个人信息管理模块中提供了个人信息浏览和修改、登录密码修改的子功能。在信息录入模块中只有系统管理员和教师具有录入权限，其中教师权限只提供课程开设和关闭、课程成绩的发布子功能；系统管理员具有最高权限，提供院系信息、专业信息、教师信息、课程信息、学生信息的录入管理子功能。在学生选课模块中实际有权限的是教师和学生用户，其中学生用户提供了课程选修和退选、课程成绩查询子功能；系统管理提供的功能与信息录入管理模块中的一致。该系统的功能模块结构图如图 4-1 所示。

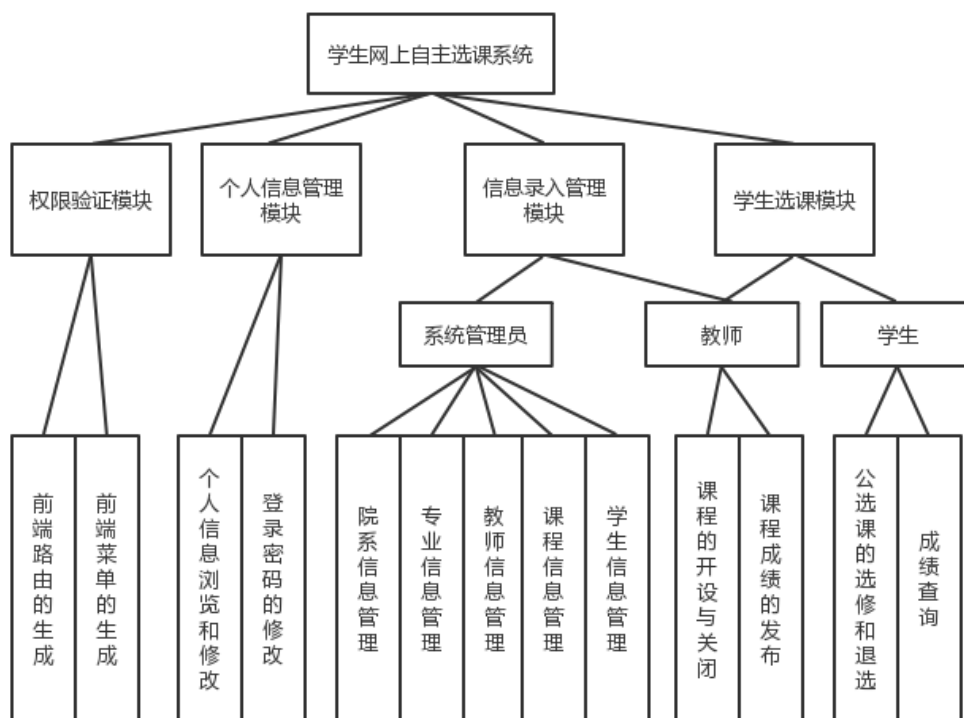


图 4-1 系统功能模块结构图

4.2. 系统前端功能模块设计

从以上的需求分析可得学生网上自主选课系统的功能模块的详细设计如下：

a) 权限验证模块。该模块提供能够支持三种权限用户都能进行登录的功能，

无论是哪一种权限用户登录系统皆生成不一致的前端路由，如果有越权的行为的话应该进行相关的报错或者将路由拦截至 404 Not Found 页面，如果没有进行登录的话应该将路由拦截到。用户登录成功之后，前端将生成对应权限能够操作的前端菜单项。

- b) 个人信息管理模块。该管理模块提供能够支持三种权限用户都能进行浏览或修改个人信息、对登录密码进行修改。
- c) 院系信息、专业信息、学生信息、教师信息、课程信息管理模块，统称信息录入管理模块。这些模块都是只有系统管理员权限登录才能看到的，其中课程信息管理模块教师权限登录也可以看到的，与之的唯一区别是教师开设课程是不需要填写教师工号而系统管理员开设课程是需要填写教师工号，提供对院系、专业、学生、教师、课程、学生选课信息进行新增、删除、修改、浏览等操作，提供能够根据院系名称、专业名称、学生姓名、教师姓名、课程姓名进行模糊匹配的查询功能，能够根据院系编号、专业编号、学生学号、教师工号、课程编号进行精确匹配的查询功能。
- d) 学生选课模块。该模块是三种权限登录都可以看到的，系统管理员权限和教师权限登录可以对学生选课信息进行浏览、可以对课程成绩及发布状态进行修改操作，教师只能浏览自己开设课程的选课信息而系统管理员可以浏览所有的选课信息；学生权限登录可以浏览所有教师开设的公选课并且能够进行选修或者退选的操作，可以进行已选修课程成绩的查询操作。

4.3. 系统后端 API 设计

学生网上自主选课系统是一个前后端分离的系统，当然后端的 API 设计也是非常重要的，下面列出该系统后端部分 API 接口的设计，具体的 API swagger 文档详见附件：

- a) POST: /api/login, 用于验证各个权限用户的登录，需要提供当前用户的账号、密码、角色等信息，登录成功返回 Access Token。
- b) GET/PUT: /api/user/current, 用于用户信息的返回或者修改，修改用户信息时必须提供完整的用户信息。
- c) PATCH: /api/user/restpwd, 用户在提交自己的旧密码和新密码信息之后，即可修改当前登录密码。
- d) GET/POST/PUT/DELETE: /api/dept、/api/major、/api/course、/api/student、/api/teacher、/api/student-course, 用于院系信息、

专业信息、课程信息、学生信息、教师信息、学生选课信息的获取或者新增或者修改或者删除。

- e) GET: /api/dept/:id、/api/major/:id、/api/course/:id、
/api/student/:id、/api/teacher/:id, 用于根据院系编号、专业编号、
课程编号、学生学号、教师工号来精确匹配相关信息。

4.4. 系统数据库设计

数据库是一种存储数据并对数据进行操作的工具，数据库的作用在于组织和表达信息，简而言之，数据库就是信息的集合^[10]。为确保数据完整性，将根据要求完成数据库概念结构的设计，然后将数据库概念的结构转换为实际的数据模型。下方的 E-R 图（实体-联系图(Entity Relationship Diagram)）就是描述了学生网上自主选课系统各个主体之间的联系，如图 4-2 所示。

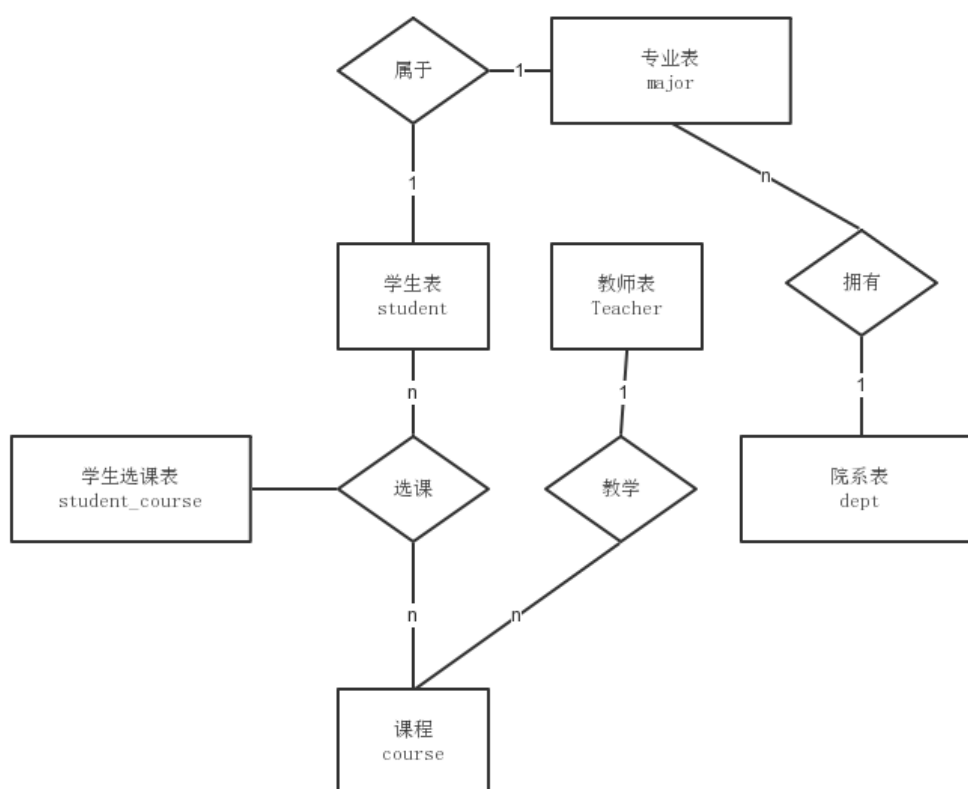


图 4-2 学生网上自主选课系统 E-R 图

本系统后端应用框架是 Egg.js，使用到的是 MySQL 数据库，为了能够做到更快更好的数据库连接和更快的数据库迁移，将会采用 node.js 中的插件 egg-sequelize 和 mysql2。根据该系统的功能以及相关的模块需求在数据库添加系统管理员表 (admin)、课程表 (course)、院系表 (dept)、专业表 (major)、

学生表（student）、教师表（teacher）、学生选课信息表（student-course）。

系统管理员表（admin）是用来存放系统管理员的登录账号、登录信息的。系统管理员表（admin）的部分字段设计及相关的说明如表 4-1。

表 4-1 系统管理员表（admin）

列名	类型	说明	备注
id	int	自增长 ID	主键
account	varchar(20)	管理员账号	唯一值，长度 5~20
user_name	varchar(10)	管理员用户名	长度 5~10
pwd	varchar(255)	登录密码	不允许空
role	varchar(10)	角色	默认值' ADMIN'，有效值:' ADMIN'，' STUDENT'，' TEACHER

课程表（course）是用来存放授课教师开设的课程或者系统管理员向授课教师进行课程委派信息的，该表与教师表（teacher）。课程表（course）的部分字段设计及相关的说明如表 4-2。

表 4-2 课程表（course）

列名	类型	说明	备注
id	int	自增长 ID	主键
c_id	varchar(20)	课程编号	唯一值，长度 1~20
t_id	varchar(20)	授课教师工号	外键，参考教师表的 t_id
c_name	varchar(30)	课程名称	长度 2~30
c_info	varchar(255)	课程介绍	允许空
credit	double	课程学分	默认值 10, 有效值:0.5 ~ 4.5
count	int	可供选课人数	默认 10, 最小 0, 最大 100
current	int	当前被选人数	默认 0, 最小 0, 最大 100

院系表（dept）是用来存放学生所在院系信息的。院系表（dept）的部分字段设计及相关的说明如表 4-3。

表 4-3 院系表 (dept)

列名	类型	说明	备注
id	int	自增长 ID	主键
d_id	varchar(20)	院系编号	唯一值, 长度 1~20
d_name	varchar(30)	院系名称	长度 2~30
d_info	varchar(255)	院系介绍	允许空

专业表 (major) 是用来存放学生所属专业信息的, 该表与院系表 (dept) 进行关联。专业表 (major) 的部分字段设计及相关的说明如表 4-4。

表 4-4 专业表 (major)

列名	类型	说明	备注
id	int	自增长 ID	主键
m_id	varchar(20)	专业编号	唯一值, 长度 1~20
m_name	varchar(30)	专业名称	长度 2~30
m_info	varchar(255)	专业介绍	允许空

学生表 (student) 用来存放学生的登录账号、登录信息的, 该表与专业表 (major) 进行关联。学生表 (student) 的部分字段设计及相关的说明如表 4-5。

表 4-5 学生表 (student)

列名	类型	说明	备注
id	int	自增长 ID	主键
s_id	varchar(20)	学号	唯一值, 长度 5~20
m_id	varchar(20)	专业编号	外键, 参照专业表的 m_id
s_name	varchar(5)	学生姓名	长度 2~5
sex	varchar(5)	性别	默认 '女', '男' 或者 '女'
pwd	varchar(255)	登录密码	不允许空
role	varchar(10)	角色	默认值 'ADMIN', 有效值: 'ADMIN', 'STUDENT', 'TEACHER'

学生选课表 (student_course) 是用来存放学生的选课信息以及学生的课程成绩信息的, 该表将与学生表 (student) 和课程表 (course) 进行关联。学生选

课表 (student_course) 的部分字段设计及相关的说明如表 4-6。

表 4-6 学生选课表 (student_course)

列名	类型	说明	备注
id	int	自增长 ID	主键
s_id	varchar(20)	学生学号	主键, 外键, 参照学生表的 s_id
c_id	varchar(20)	课程编号	主键, 外键, 参照课程表的 c_id
score	double	成绩	默认 0, 最小值 0, 最大值 100
is_publish	varchar(3)	是否已公布成绩	默认 '未公布', 有效值: '已公布'、 '未公布'

教师表 (teacher) 用来存放教师的登录账号、登录信息的。教师表 (teacher) 的部分字段设计及相关的说明如表 4-7。

表 4-7 教师表 (teacher)

列名	类型	说明	备注
id	int	自增长 ID	主键
t_id	varchar(20)	教师工号	唯一值, 长度 5~20
t_name	varchar(5)	教师姓名	长度 2~5
sex	varchar(5)	性别	默认 '女', '男' 或者 '女'
job	varchar(5)	职称	默认 '讲师', 有效值: '讲师', '教授', '副教授'
count	int	可供开课数	默认 10, 最小值 0, 最大值 100
current	int	当前已开课数	默认 0, 最小值 0, 最大值 100
pwd	varchar(255)	登录密码	不允许空
role	varchar(10)	角色	默认值 'ADMIN', 有效 值: 'ADMIN', 'STUDENT', 'TEAC HER'

5. 系统详细设计与实现

5.1. 系统整体原型设计

用户登录系统之后，可以看到系统界面，在系统界面的左方提供当前用户登录之后生成的权限菜单，权限菜单中可包含主菜单和子菜单。在系统管理员或者教师管理操作数据页面中，需要提供查询、新增、删除功能，在以表格的形式将当前的操作信息展示出来，表格中提供选择框进行选择，用户可以根据自己的需求对不需要的数据项进行删除。点击编辑数据时，相应行对应的数据转变成输入框的形式，操作栏的按钮转变成保存、取消。查询中根据当前页面的功能需求，分别提供编号、关键词查询等。由于数据量会比较多，所以在表格展示的下方需要提供分页功能。信息管理原型设计图如图 5-1 所示。

<input checked="" type="checkbox"/>	列1	列2	列3	列 4	操作
<input type="checkbox"/>					编辑 更多
<input type="checkbox"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	保存 取消
<input type="checkbox"/>					编辑 更多
<input type="checkbox"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	保存 取消

图 5-1 信息管理原型设计图

学生选课不同于信息管理。学生登录系统之后可以浏览到所有教师开设的课程信息，在每一门课程中都含有该门课程的详细信息（包含课程名称、课程可供选课人数、课程简介等）以及该门课程的授课教师的详细信息（包含教师名字、教师职称、教师的联系电话等），在学生的个人选修课中提供课程成绩信息。系统中提供课程名称、教师名称的快速筛选查询功能，学生在浏览课程的相关信息之后允许其进行选课操作，若是已出成绩状态，不允许学生进行退选操作，相应的按钮置灰。学生选课原型设计图如图 5-2 所示。

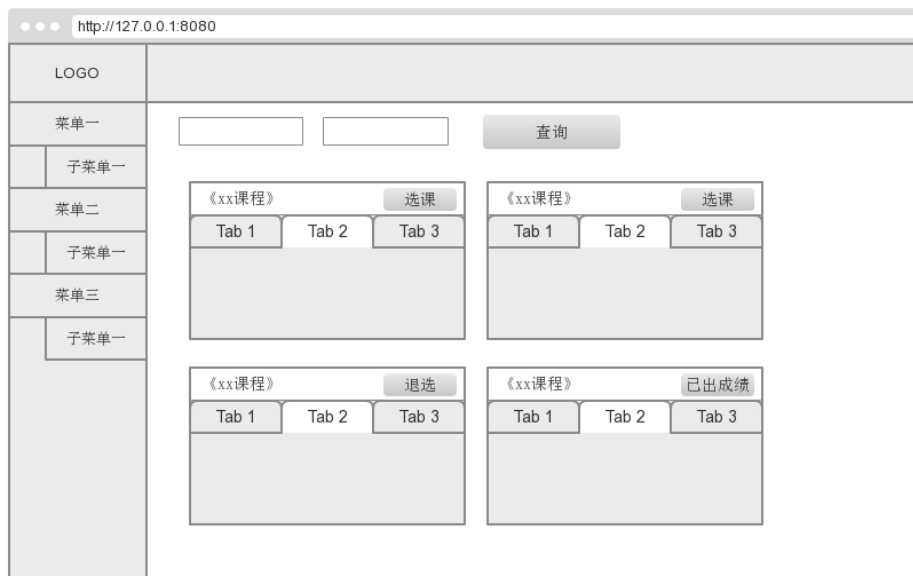


图 5-2 学生选课原型设计图

5.2. 系统后端数据库连接配置

Node.js 中的插件 egg-sequalize 和 mysql2 是 Egg.js 官方推荐的数据库连接工具和数据库迁移工具，下方展示的是插件 egg-sequalize 连接 MySQL 数据库的部分配置代码。

```
'use strict';

module.exports = appInfo => {
  const config = exports = {};
  // mysql
  config.mysql = {
    // 单数据库信息配置
    client: {
      // host
      host: '127.0.0.1',
      // 端口号
      port: 3306,
      // 用户名
      user: 'root',
      // 密码
      password: 'root',
```



```

        // 数据库名
        database: 'student_elective_system',
    },
    // 是否加载到 app 上, 默认开启
    app: true,
    // 是否加载到 agent 上, 默认关闭
    agent: false,
};

config.sequelize = {
    "dialect": "mysql",
    "host": "127.0.0.1",
    "port": 3306,
    "database": "student_elective_system",
    "username": "root",
    "password": "root"
}

return config;
};

```

5.3. 权限验证模块

权限验证模块中设置了三种权限，系统管理员、学生、教师，用户在权限验证界面中输入相应的账号密码，在前端的逻辑中将会对用户输入的内容进行简单的验证，验证通过之后将会将数据提交给后端 API 进行二次验证，二次验证中包含了账号是否有效的验证，验证通过之后，前端将会生成对应权限的菜单路由。的流程图如图 5-3 所示。

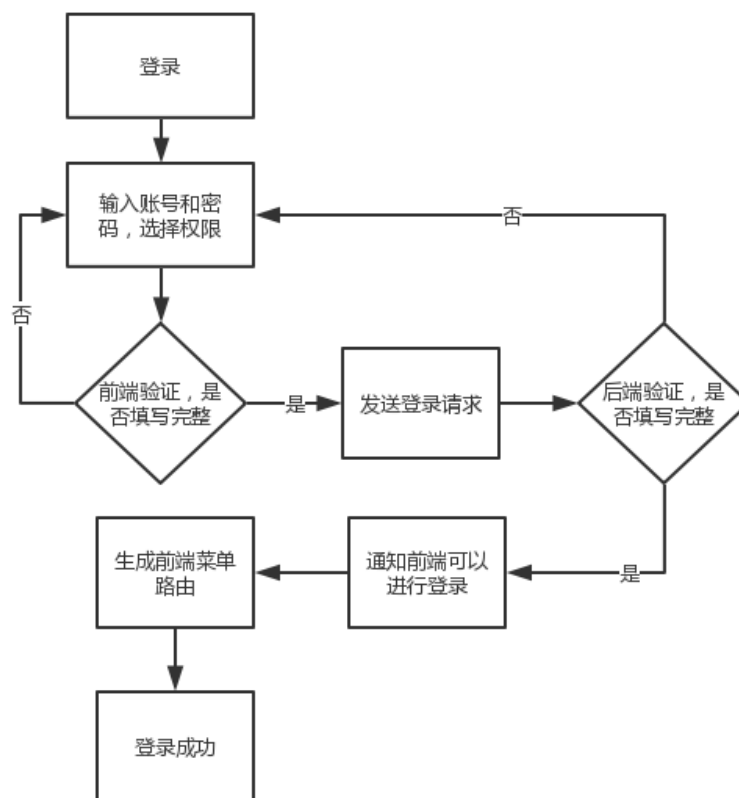


图 5-3 权限验证流程图

在权限验证界面上设有通用的权限验证方式，无论是哪种权限用户都通过账号密码的方式登录系统，系统使用的默认权限是学生，其中系统管理员账号是其邮箱，教师的账号是教师工号，学生则是学生学号，权限验证界面如图 5-4 所示。

图 5-4 系统权限验证界面

系统中定义了所有页面的路由，用户登录系统之后通过对比前端路由文件中的路由权限在进行对前端路由的过滤，在过滤出当前用户所需的路由生成对应的菜单页面，此时用户即可进行相应的操作，该流程图如图 5-5 所示。

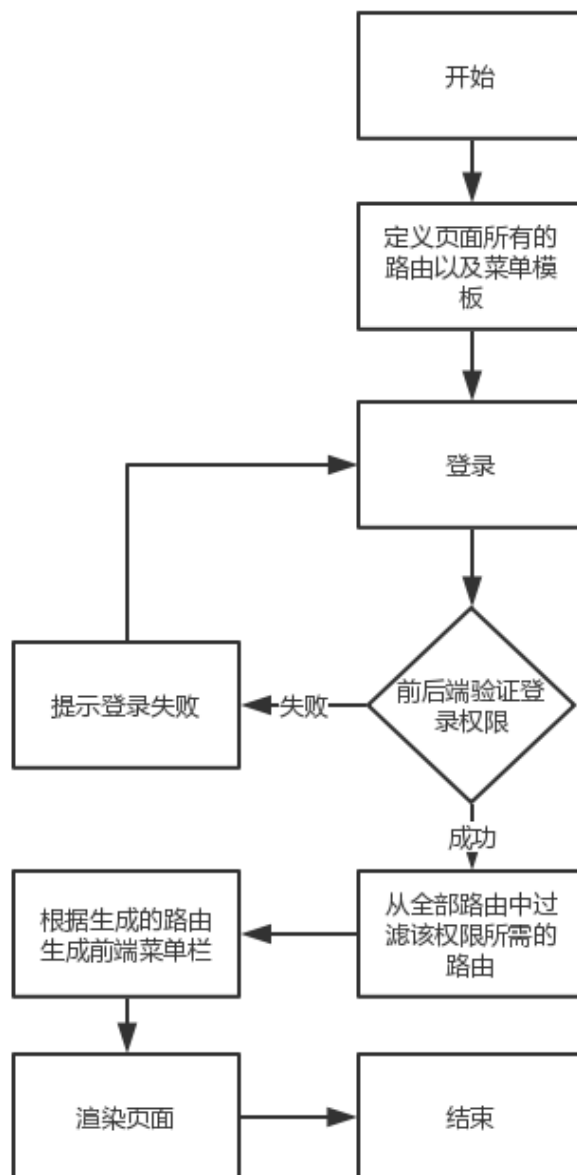


图 5-5 生成前端路由的流程图

系统管理员权限用户、教师权限用户、学生权限用户登录系统之后生成相应的访问菜单，每一个菜单都对应着不同的信息管理和不同的路由。系统管理员权限生成的菜单包含了个人中心、各种信息管理，教师权限生成的信息管理只有课程信息管理，学生权限生成的只有课程选修相关的菜单。其中学生权限用户生成的前端菜单如图 5-6 所示，系统管理员权限用户生成的前端菜单如图 5-7 所示。



图 5-6 学生权限用户生成的前端菜单



图 5-7 系统管理员权限用户生成的前端菜单

5.4. 个人信息管理模块

个人信息管理模块是每种权限用户登录之后即可看到的模块，在该模块中，将根据当前登录的用户信息生成对应的个人信息列表。在该界面中，当前的用户可以对自己的信息浏览和修改等操作，其中的修改密码将验证用户的初始密码、新密码等，密码修改成功之后，前端页面会将当前的路由重置到权限验证界面，此时用户使用新修改的密码即可登录。个人信息管理模块的流程图如图 5-8 所示。

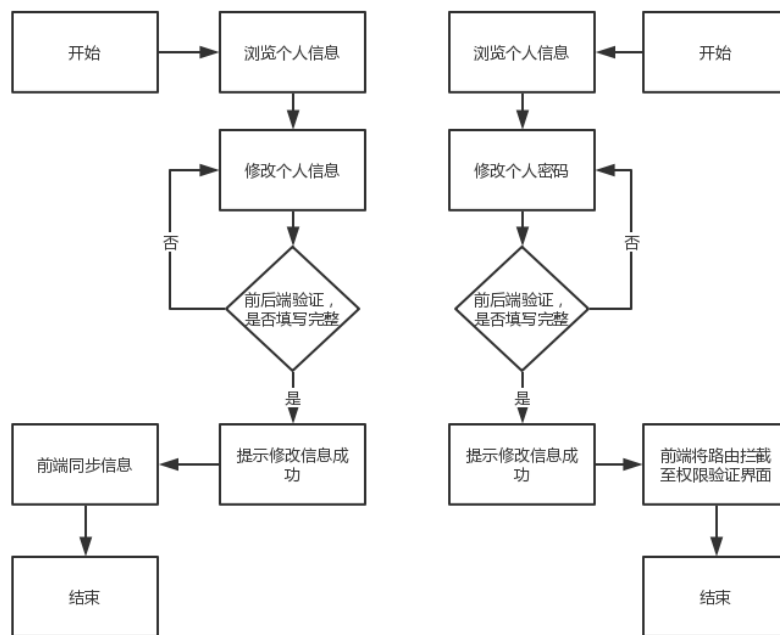


图 5-8 个人信息管理模块的流程图

用于提交个人信息修改的部分前端 JavaScript 代码如下所示。

```

this.$PUT(CURRENT, payload)
  .then(res => {
    this.isSaveInfo = false
    if (res.code === 200) {
      this.$notification.success({
        message: res.message,
        description: 'Successfully'
      })
      this.$store.dispatch('GetCurrent')
      this.isInputDisabled = true
    }
  })
  .catch(_ => {
    this.isSaveInfo = false
    this.$store.dispatch('GetCurrent')
    this.isInputDisabled = true
  })

```

5.5. 信息录入管理模块

在信息录入模块中只有系统管理员和教师拥有录入信息的权限。其中系统管理员可以对院系信息、专业信息、课程信息、教师信息、学生信息的数据录入进行操作，教师登录系统可以进行课程开设。在录入专业信息时，需要验证其所属院系编号是否存在，在录入课程信息时，需要验证其所属授课教师工号是否存在，在录入学生信息时，需要验证其所属专业编号是否存在。教师在开设课程时或者系统管理员给教师进行委派课程时，需要验证课程的学分和上限的选课人数。信息录入模块的流程图如图 5-9 所示。

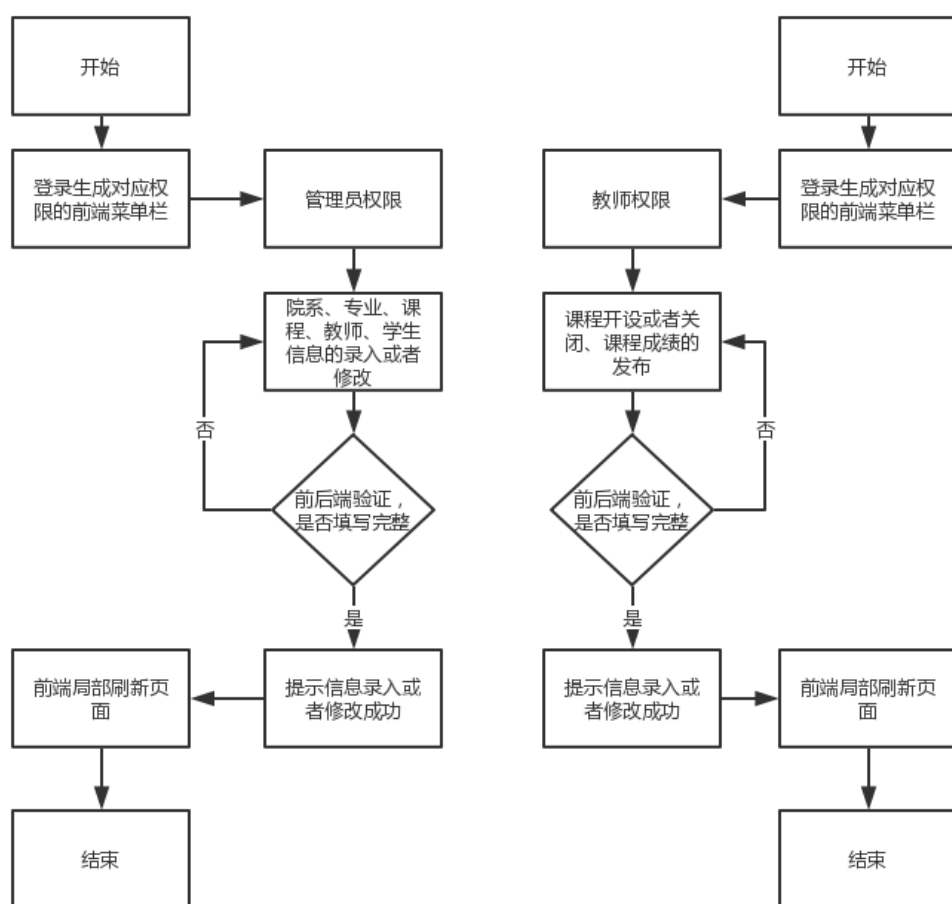


图 5-9 信息录入管理模块的流程图

在信息录入时需要完整的填写相关信息，在信息管理中用户在筛选信息之后，对其进行数据修改，修改之后允许其保存数据，当然还未点击保存之前允许用户取消当前操作的。如果想要删除部分信息，只需对相应的选择框进行勾选，点击删除选中项按钮，删除时将进行二次确认，防止误删除。如图 5-10 展示了系统管理员录入课程信息界面，图 5-11 展示了系统管理员操作课程信息界面。

添加一个课程

* 课程编号:

* 教师工号:

* 课程名称:

课程简介:

请输入课程简介

* 课程学分:

* 可供选课人数:

图 5-10 系统管理员添加课程界面

<input type="checkbox"/>	课程编号	教师工号	课程名称	课程简介	课程学分	可供选课人数	当前被选人数	操作
<input type="checkbox"/>	9876	600036	习近平的政治思想		3.5	15	0	编辑 更多

图 5-11 系统管理员对课程进行管理界面

用于查询课程信息的 Vue 前端部分 template 模板代码如下所示。

```

<operation @add="add" @submit="handleSubmit" @del="del('c_id', COURSE)"
    :form="form" :disabledAdd="disabledAdd"
    :disabledDel="disabledDel">
  <a-form-item>
    <a-input type="number" size="large" placeholder="输入课程编号"
      v-decorator="['c_id', { initialValue: '' }]" >
    </a-input>
  </a-form-item>
  <a-form-item>
    <a-input size="large" placeholder="输入课程名称关键词"
      v-decorator="['c_name', { initialValue: '' }]" >
    </a-input>
  </a-form-item>
  <a-form-item>

```

```

<a-tooltip placement="top">
    <template slot="title">注意！同时查询编号和名字，结果以编号的返回
    结果为准</template>
    <a-button size="large" htmlType="submit" type="primary"
    icon="search" @click.stop.prevent="handleSubmit">
        查询
    </a-button>
</a-tooltip>
</a-form-item>
</operation>

```

5.6. 学生选课模块

在学生选课模块中，只有学生登录该系统才能看到的，登录系统之后，学生通过简单的课程搜索查询到自己喜欢的课程，然后进行选课操作。教师在课程结束之后对学生的选课的课进行成绩评定，并且发布成绩，在没有发布成绩之前学生都可以进行课程退选的操作，如果是已发布成绩状态，学生是不可以进行退选。学生选课管理模块的流程图如图 5-12 所示。

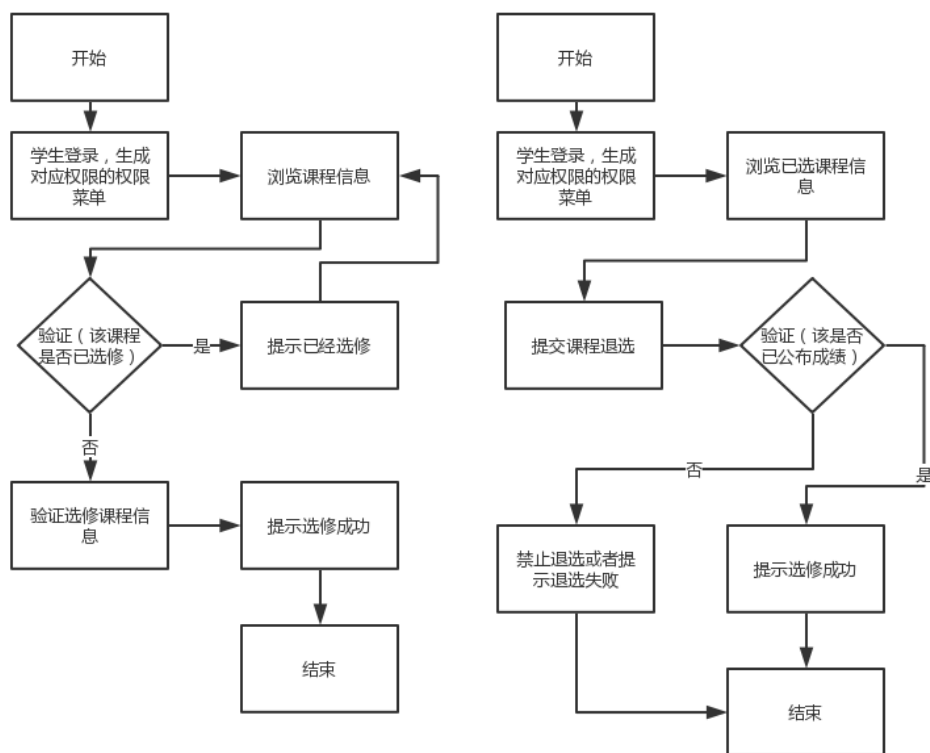


图 5-12 学生选课管理模块的流程图

学生网上自主选课系统中核心的功能就是，学生的选课操作，学生在登录系统之后能够浏览所有教师开设课程，并通过课程名、教师名来搜索匹配自己感兴趣的课程，列表中还展现了课程、开设教师的详细信息，学生可以直观的看到当前的课程的学分、剩余的选课人数、授课教师的联系电话等信息，页面展示如图 5-13 所示。



图 5-13 学生的选课操作

学生课程选修之后，授课教师即可在系统的学生选课信息管理中查看到学生的选课信息，课程结束之后，授课教师对学生选课的课程进行课程成绩评定。在还没有发布课程成绩之前，学生皆可以退选课程，如果是已发布课程成绩那么就不允许学生进行退选操作，退选时验证学生的选课信息，并给出验证结果，验证成功之后，学生可以进行重新选课等操作。课程退选流程图如图 5-14 所示。

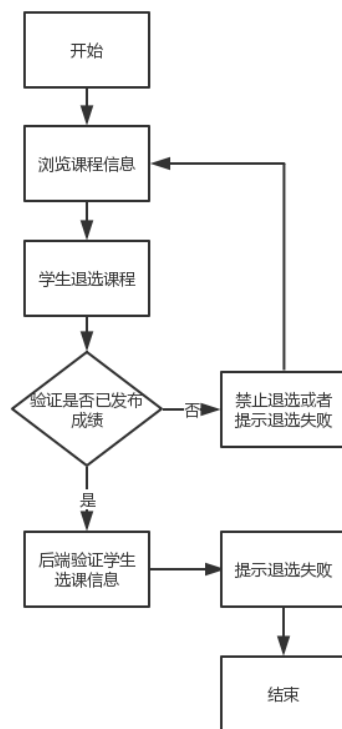


图 5-14 学生退选课程流程图

课程结束之后，教师登录系统可以浏览到自己开设的课程以及选修该课程的学生信息列表，在学生选课管理中教师可以进行成绩的修改与发布、通过操作栏中的更多可以看到学生的信息和课程信息。学生的课程成绩查询界面如图 5-15 所示，教师发布课程成绩的页面如图 5-16 所示。

图 5-15 学生的课程成绩查询

学号	课程编号	成绩	补考成绩	是否已公布成绩	操作
50100	10202	80	0	未公布	保存 取消

图 5-16 教师进行成绩发布

用于课程退选前端部分前端 JavaScript 代码如下所示。

```
onChangeElective (val) {  
  this.loading = true  
  this.$DELETE(STUDENT_COURSE, val)  
    .then(res => {  
    if (res.code) {  
      this.$notification.success({  
        message: res.message,  
        description: 'Successfully'  
      })  
      this.getInfo({  
        url: STUDENT_COURSE  
      })  
    } else {  
      this.loading = false  
    }  
  })  
  .catch(_ => {  
    this.loading = false  
    this.$notification.error({  
      message: '退选失败，请重试 !!!',  
      description: 'Error'  
    })  
    this.getInfo({  
      url: STUDENT_COURSE  
    })  
  })  
}
```

6. 单元测试和功能测试

6.1. 测试概述

软件测试是保证软件质量的重要手段，无论怎样强调软件测试的重要性以及它对软件质量的影响都不过分^[12]。从软件开发过程的角度来看，软件测试可分为需求测试，单元测试，集成测试，验证测试和系统测试。单元测试（也称为模块测试）是测试程序模块以测试最小单元的软件设计，即在完成代码之后执行的测试。利用单元测试可以快速识别系统中的一些漏洞和缺陷，并指导程序员修改他们的响应。

单元测试的过程一般认为单元测试应紧接在编码之后，当源程序编制完成并通过复审和编译检查，便可开始单元测试^[11]。进行动态的单元测试前先要对程序进行静态分析和代码审查，这样会发现一些代码的逻辑表达错误，而且一旦发现错误，就会同时对错误的性质和其位置定位，从而降低调试的代价^[12]。接下来将从单元测试和功能性测试来检测这个系统的完整性。

6.2. 单元测试

这里的单元测试主要针对的是后端 API 的简单的单元测试，通过编写一定的单元测试用例来验证接口返回数据的正确性。由于每一个接口都有进行权限验证，所以在单元测试数据构建中会将权限加上，权限验证成功之后，将会把构造好的测试数据注入到测试用例中，然后验证其返回的 HTTP 状态码并且验证其返回数据的完整性等，验证成功之后将会输出测试结果。

下面列出部分的测试用例以及其耗时的返回情况，具体的测试用例以及返回情况详见附件。

- 1) 在 `/api/user/current` 接口中创建一个测试用例，要求返回状态码为 200，如图 6-1 展示该测试用例的返回结果，该接口的测试用例总耗时 7ms。

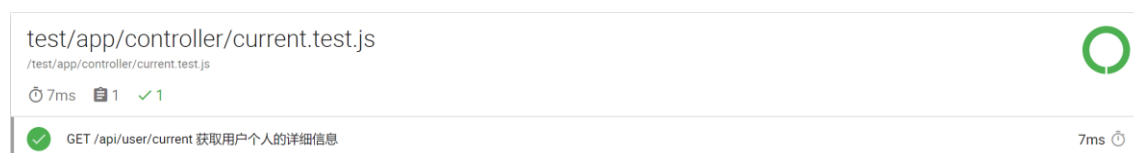


图 6-1 个人信息接口的测试用例

- 2) 在 `/api/student-course` 接口中创建两个测试用例，要求返回状态码为 200，如图 6-2 展示该测试用例的返回结果，该接口的两个测试用例总耗

时 49ms。

test/app/controller/studentCourse.test.js <small>/test/app/controller/studentCourse.test.js</small>		
🕒 49ms 📄 2 ✓ 2		
✓	GET /api/student-course 获取所有的选课信息	16ms 🕒
✓	POST /api/student-course 添加选课信息	33ms 🕒

图 6-2 学生选课信息接口的测试用例

3) 在 /api/course 接口中创建三个测试用例，要求返回状态码为 200，如图 6-3 展示该测试用例的返回结果，该接口的两个测试用例总耗时 70ms。

test/app/controller/course.test.js <small>/test/app/controller/course.test.js</small>		
🕒 70ms 📄 3 ✓ 3		
✓	GET /api/course 获取所有的课程信息	48ms 🕒
✓	POST /api/course 添加课程信息	14ms 🕒
✓	GET /api/course/id 根据课程编号来获取课程信息	8ms 🕒

图 6-3 课程信息管理接口的测试用例

4) 在 /api/student 接口中创建三个测试用例，要求返回状态码为 200，如图 6-4 展示该测试用例的返回结果，该接口的两个测试用例总耗时 116ms。

test/app/controller/student.test.js <small>/test/app/controller/student.test.js</small>		
🕒 116ms 📄 3 ✓ 3		
✓	GET /api/student 获取所有的学生信息	15ms 🕒
✓	POST /api/student 添加学生信息	94ms 🕒
✓	GET /api/student/id 根据学生学号来获取学生信息	7ms 🕒

图 6-4 学生信息管理接口的测试用例

6.3. 功能测试

当然除了单元测试，还需进行部分功能模块的可用性测试即功能测试，这也是为了保证系统功能的完整性和易用性以及前端页面功能能够完美匹配后端设计的 API，使得页面的功能模块是完整的没有出现较大错误。在可用性测试中，主要测试学生的选课和退选操作，在学生选课之前，需要系统管理员录入相关的院系信息、专业信息、教师信息等，教师在此基础上进行课程开设并设定课程上限选课人数。开课之后，使用学生账号登录进行选课操作，无论选课成功还是失败都需要对当前学生发起一个信息通知。学生选课测试流程图如图 6-4 所示。

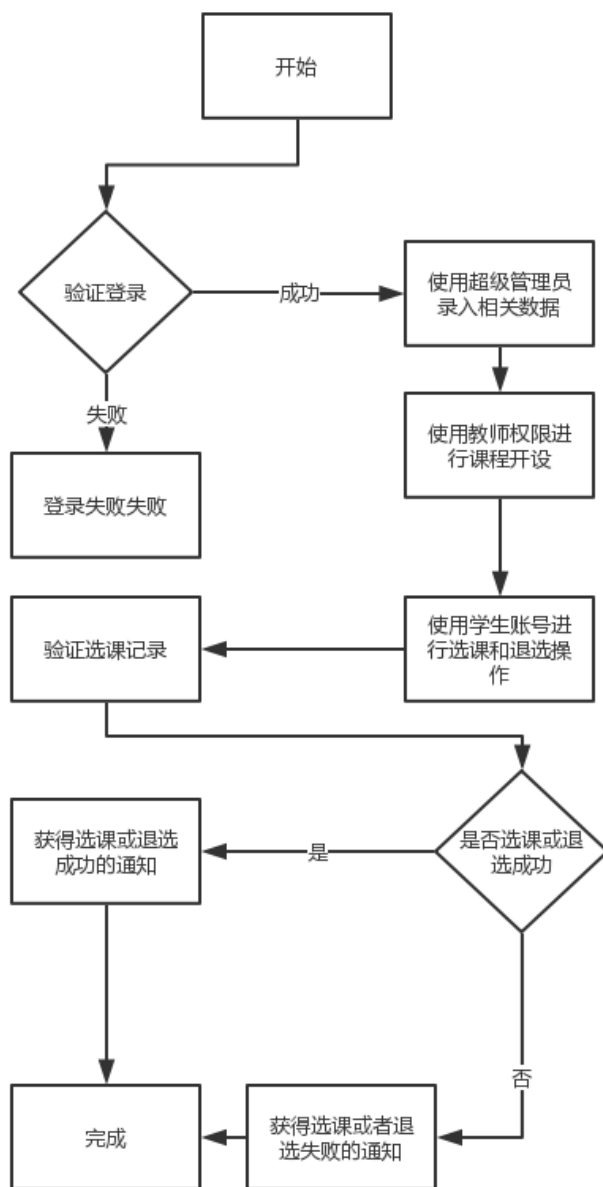


图 6-5 学生选课测试流程图

在进行简单的可用性测试之后，亦能看到学生的成功选课、退选等操作是正常的，也能对相应的错误给出相应的提示。教师在未公布成绩之前，学生是可以对某门课程进行退选，发布成绩之后，学生对某门课程进行退选是无法进行的，前端页面也有对相应的按钮进行置灰。总体上该系统的可用性较高。

6.4. 测试报告分析

此次的单元测试覆盖的后端 API 文件 14 个，总共测试了 31 个相关的函数和方法，每一个接口的测试用时均不超过 150 毫秒，符合单元测试的效率高的要求，也符合系统的可操作性、易用性、安全系数高的要求。此次单元测试覆了 45.59%

Statements, 42.32% Branches, 58.18% Functions, 45.62% Lines, 下表列出单元测试用例在后端代码中各个模块的覆盖情况, 如图 6-6 所示。

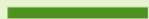

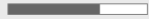


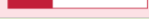


File		Statements		Branches		Functions		Lines	
app		100%	69/69	100%	0/0	100%	1/1	100%	69/69
app/controller		36.16%	162/448	31.82%	63/198	50%	18/36	36.24%	162/447
app/extend		66.13%	41/62	54.9%	28/51	60%	6/10	68.42%	39/57
app/middleware		42.86%	15/35	22.22%	4/18	100%	4/4	42.86%	15/35
app/model		100%	48/48	50%	2/4	100%	15/15	100%	48/48
app/service		32.29%	103/319	64.86%	48/74	44.19%	19/43	32.29%	103/319
app/utils		100%	2/2	100%	0/0	100%	0/0	100%	2/2
config		100%	20/20	100%	0/0	100%	1/1	100%	20/20

图 6-6 单元测试覆盖情况

覆盖率测试是白盒测试的重要手段, 在测试报告中可作为量化指标的依据, 对于软件的重点模块, 应使用多种覆盖率标准衡量代码的覆盖率^[13]。此次的单元测试虽然做不到 100% 的覆盖率, 但是由以上的这些覆盖率以及返回结果来看, 这次的单元测试能够说明该系统是可用的、没有较大错误的一个易用的学生网上自主选课系统。

6.5. 测试结果总结

经过一系列的测试, 而且从上面的测试结果也能看出来本次的单元测试以及简单的可用性测试即功能测试, 出现的测试结果基本与预期的结果是一致的, 但是也不排除有点问题或者错误是存在的。进行单元测试不仅用于确保当前代码的合法性, 还用以确保修复, 改进或重构后代码的合法性。虽然页面的各个功能以及后端的 API 在测试过程中均测试通过, 且能够进行正常使用, 但还有许多功能并不是特别完善。所以, 在今后的开发过程中会继续对该系统的更多细节进行测试, 极大的保证该系统的正确性, 使得该系统能够变成一个人人易用的应用。

7. 总结以及展望

7.1. 总结

伴随着无纸化办公的逐步实现，高校对学生的选课质量、选课效率都有一定的要求，目前高校公选课存在缺乏科学合理的课程体系，缺乏稳定的师资队伍，学生选课也存在娱乐性、盲目性，同时缺乏完善的质量监督管理体系。本系统是为了完善学生自主选课机制，激发学生的兴趣爱好而诞生的，旨在实现一个学生能够在网上自主选课（选修课）的微系统。现从论文和设计两大方面进行总结：

论文方面，首先从选题背景出发讨论为什么要做这个学生网上选课系统；接着阐述了该系统采用的相关技术要点；然后开始进行需求分析，需求分析包括了系统的功能模块分析、系统全景图和用例图等；在概要设计中体现了功能模块设计、系统后端 API 设计、系统数据库设计等；系统详细设计中详细的说明了各个模块的实现过程；最后就是通过系统的单元测试和功能测试来检查该系统是否还有瑕疵等。

设计方面，学生网上自主选课系统主要针对的是高校学生和教师使用的，学生方面可以通过这个系统来自选选课，避免了传统教学模式（学生按照学校安排好的课程上课）不能适应新型的教学手段等问题。教师方面，也能节省大量的工作量，通过该系统能够大幅度的减少错误的发生几率，更加适应无纸化办公的需求。总体 UI 界面上美观优雅，简洁明了，是一个易上手的 WEB 应用网站。

7.2. 展望

虽然已经按照需求实现了系统的基本功能，但是经过单元测试也能看出还有很多潜在的问题，这些潜在的问题往往就是导致系统崩溃的因素，经过一定的排查和导师的指导，发现系统不是很完善，很多地方的逻辑也不够严谨，以及一些待实现的功能，比如不能进行批量的数据导入导出、在高并发情况下系统容易崩溃或者会破坏数据的完整、系统权限的设置还不够完善，学生在选课时无法实时推送给授课教师等等。希望在今后的不断学习和实践过程中，能够改善自己的不足之处，也能写出更好的、易用的一个应用程序。

参考文献

- [1] 孟国荣, 司海清. 高校教学改革与学生自主选课的研究[J]. 教育与职业, 2008, 32: 047.
- [2] Sorensen E, Mikailasc M. Model-view-ViewModel (MVVM) design pattern using Windows Presentation Foundation (WPF) technology[J]. MegaByte Journal, 2010, 9(4): 1-19.
- [3] 李容. 基于 MVC 模式的 WEB 应用研究[J]. 软件导刊, 2010, 9(1): 19-21.
- [4] 吴慧华, 徐燕. 高校公选课教学存在的问题与对策[J]. 教育探索, 2011, 2011(1): 71-72.
- [5] 贾民政. 基于区块链的投票系统的设计与实现[J]. 北京工业职业技术学院学报, 2018, 17(2).
- [6] 兰旭辉, 熊家军, 邓刚. 基于 MySQL 的应用程序设计[J]. 计算机工程与设计, 2004, 25(3): 442-443.
- [7] 王胜, 张靖. 基于 Vue.js 高速路政管理系统的设计与实现[J]. 电脑知识与技术, 2017(21): 40.
- [8] Fielding R T, Taylor R N. Architectural styles and the design of network-based software architectures[M]. Irvine, USA: University of California, Irvine, 2000.
- [9] 刘立. MVVM 模式分析与应用[J]. 微型电脑应用, 2013, 28(12): 57-60.
- [10] 沈文娟. 学生智能选课系统的设计与实现[D]. 南昌大学, 2012.
- [11] 孙杨. 软件自动化测试集成系统的研究与实现[D]. 西安电子科技大学, 2012.
- [12] 陈站华. 软件单元测试[J]. 无线电通信技术, 2003, 29(005): 50-51.
- [13] 路晓波. 软件开发过程中白盒测试方法和工具的研究及应用[D]. 2013.

致 谢

大学学习生活在这儿即将划上一个句号，而对于我的人生来说却仅仅只是一个逗号，将是我面对新的征程的开始。本次的网站设计及论文是在我的导师朱婧的亲切关怀和耐心的指导下完成的也许我不是您最出色的学生，但您却是我所最尊敬的老师。您是如此的治学严谨，学识渊博，视野广阔，思想深刻，您用心为我营造一种良好的学术氛围，让我的论文更加的严谨。至此论文付梓之际，我的心情无法保持平静，从开始选择课题到论文的顺利答辩，有无数可敬的师长、朋友给了我很多的帮助，在这里请您接受我诚挚的谢意！

感谢我的父母，如果没有他们的无私的奉献、默默支持和无微不至的帮助，我不可能取得今日的成果。也很庆幸我能够遇到很多的良师益友，感谢他们在生活上、学习上给予我很多的帮助和照顾，在此我谨以最朴实的话语致以最崇高的敬意。

“人生就像一场旅行在意不是目的地，而是沿途风景以及观看风景心情”这句座右铭一直铭记于我心，愉快的生活重要的不是结果而是享受过程，今后我会继续努力，永不放弃。最后，再次对那些在论文完成过程中，关心、帮助我的同学和朋友们表示衷心地感谢！