

上海师范大学

本科毕业论文(设计)

题 目： 基于 AngularJS 和 Java Web 的排课系统的设计与实现

Title: Design and Implementation of a Course Scheduling System Based on AngularJS and Java Web

学 院 商学院

年级专业 2012 级计算机科学与技术(中法合作)

学生姓名 黄子杰

学 号 120142708

指导教师 杜文杰

完 成 日 期 2016 年 4 月

目 录

上海师范大学本科毕业论文（设计）诚信声明.....	I
上海师范大学本科毕业论文（设计）选题登记表	II
上海师范大学本科毕业论文（设计）指导记录表.....	IV
摘要.....	V
Abstract	VI
1、绪论	1
1.1 研究的背景.....	1
1.2 研究的目的和意义.....	1
1.2.1 研究目的.....	1
1.2.1 研究意义.....	2
1.3 研究的对象.....	2
1.4 研究的方法和手段.....	2
1.4.1 研究的方法.....	2
1.4.2 研究的手段.....	3
1.5 研究的基本框架.....	2
2、系统关键技术介绍	4
2.1 架构技术.....	4
2.1.1 单页面应用.....	4
2.1.2 无状态服务.....	5
2.1.3 数据传输.....	5
2.2 浏览器端技术.....	6
2.2.1 浏览器端的 MVC 模式和 AngularJS 框架.....	6
2.2.2 界面组件.....	8
2.2.3 自动化构建和模块化开发.....	9
2.2.4 构建混合型桌面应用.....	9
2.3 服务器端技术	10

2.3.1 Spring Boot.....	10
2.3.2 办公文档处理	10
2.3.3 数据持久化	10
3、需求分析与系统建模.....	12
3.1 系统核心功能需求概述	13
3.2 课程	16
3.2.1 课程功能需求	16
3.2.2 课程的模型	16
3.3 培养方案	16
3.3.1 培养方案需求	16
3.3.2 培养方案数据模型	17
3.4 执行计划	17
3.4.1 执行计划需求	17
3.4.2 执行计划数据模型	18
3.5 公共属性	19
3.6 教师和用户	20
4 系统功能实现	21
4.1 身份验证	21
4.1.1 令牌系统	21
4.1.2 权限控制	23
4.2 文档处理	24
4.3 选课数据交互和业务逻辑	25
4.3.1 简单业务逻辑举例	25
4.3.2 浏览器端复杂业务逻辑举例	28
4.4 日志系统	31
5、结论.....	33
5.1 系统达到的目标	33
5.2 设计存在的不足	33
参考文献	34

附录	35
致谢	44

上海师范大学本科毕业论文（设计）

诚信声明

本人郑重声明：所呈交的毕业论文(设计)，题目《基于AngularJS和Java Web 的排课系统的设计与实现》是本人在指导教师的指导下，进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文（设计）不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明应承担的法律责任。

作者签名：

日期：2016年3月25日

上海师范大学本科毕业论文（设计）选题登记表

学生姓名	黄子杰	学号	120142708	专业	计算机科学与技术 (中法合作)
学 院	商学院	指导教师姓名/职称		杜文杰/讲师	
题 目	基于 AngularJS 和 Java Web 的排课系统的设计与实现				
<p>本选题的意义及国内外发展简况：</p> <p>在无纸化办公和信息技术高速发展的当下，大量的教育部门的系统仍运行在疏于维护的客户机和服务器结构上。信息系统的过时和出错导致了工作效率的下降。基于现有信息系统和数据源开发互联网服务是应对这些问题的良方。笔者旨在通过查阅文献和分析本学院的实际需求，适当地选用可靠、安全的技术，设计出一个基于浏览器和服务器结构的、易于维护、响应迅速的排课系统，并为今后更多的信息管理系统建设定立数据接口和标准。</p>					
<p>研究内容：</p> <p>设计一个基于浏览器和服务器结构的系统，且将浏览器端和服务器端尽可能地解耦合，实现一个单页应用。浏览器端运用基于 MVC 的 JavaScript 库，服务器端使用成熟的 Java Web 解决方案。</p> <p>笔者将以上的架构设计结合本学院的实际需求，进行系统的整体设计，并运用 Java Web 和 AngularJS 技术实现此设计以解决系统在生产环境中可能产生的问题，同时向其他系统数据接口，以增加系统的可扩展性。</p>					
<p>研究方法、手段及步骤：</p> <p>研究方法： 观察法、调查法、实验法、文献资料法。</p> <p>研究手段： 仔细研读和分析业务需求的文本、参考其他类似系统设计出数据结构、根据数据结构和业务需求选定技术、查找相应的技术文献。</p> <p>写作步骤： 先对成熟的 Hibernate、Spring、Struts 实现的传统架构进行调研，再根据单页应用的需求，对后端架构进行重构、对前端架构重新设计实现，最后基于实际需求将设计付诸实施。</p>					

主要参考文献:

- 1、Michael S, Josh C. Single Page Web Applications[M]. Manning Publications, 2013 : 4.
- 2、Alex A. Spring Boot Cookbook[M]. Packt Publishing, 2015: 24–25.
- 3、Casimir S, 王群峰, 杜欢. 全端Web开发[M]. 人民邮电出版社, 2015 : 40.
- 4、Ari L, 赵望野, 徐飞. AngularJS权威指南[M]. 人民邮电出版社, 2014 : 3.
- 5、Travis M. Getting Started with Gulp[M]. Packt Publishing, 2015 : 7–8 .

时间进度:

1. 2015 年 9 月 1 日至 10 月 15 日: 阅读有关文献, 提交毕业论文选题登记表;
2. 2015 年 11 月 20 日之前: 收集资料, 完成系统设计;
3. 2016 年 1 月 20 日之前: 完成系统实现, 递交论文初稿;
4. 2016 年 2 月 20 日之前: 修改论文内容, 初步定稿;
5. 2016 年 2 月 28 日之前: 编辑格式, 定稿;

学生_____ (签名) 2015 年 9 月 30 日

指导教师意见:

指导教师_____ (签名) 2015 年 10 月 10 日

专业负责人意见:

专业负责人_____ (签名) 2015 年 10 月 15 日

注: 本表与毕业论文(设计)一起存档, 保存期为四年。

上海师范大学本科毕业论文（设计）指导记录表

学院： 商学院

专业	计算机科学与技术（中法合作）	指导教师	杜文杰	学生姓名	黄子杰	学号	120142708
毕业论文（设计）题目		基于 AngularJS 和 Java Web 的排课系统的设计与实现					
日期	指导内容		存在问题与进一步改进意见			教师签名	学生签名

注：本表由指导教师根据毕业论文（设计）指导工作方案和实际指导情况填写，在指导工作完成后交学院存档，保存期四年。

摘 要

在无纸化办公和信息技术高速发展的当下，大量的教育部门的系统仍运行在疏于维护的客户机和服务器结构上，无法满足新的业务要求。信息系统的过时和出错导致了工作效率的下降。基于现有信息系统和数据源开发互联网服务是应对这些问题的良方。笔者旨在通过查阅文献和分析本学院的实际需求，适当地选用可靠、安全的技术，设计出一个基于浏览器和服务器结构的、易于维护、响应迅速的排课系统，并为得以为今后更多信息的系统建设定立数据接口和标准。本文在浏览器端和服务器端所采用的框架分别为 Java 的 Spring 和 AngularJS，基于此类框架，开发人员得以快速构建需要的系统，根据变化的业务需求更新系统，具有良好的扩展性、可靠性，且得以优化现有的工作流程、减少教务工作的错误率。

关键词： 排课系统； Java Web； Spring； AngularJS ； 浏览器和服务器结构

Abstract

Despite the rapid development of information technology and the trend of paperless OA, many IT systems in educational organizations are still in C/S mode without necessary maintenance, and they cannot meet users' needs, resulting in the reduction of efficiency. Switching to Internet services built upon existing systems and data sources could cope with these problems. The author has consulted relevant articles, analyzed his institute's demands, chosen appropriate techniques, and designed a highly responsive, easy maintainable course scheduling system in B/S mode. He has also set up the standards of APIs for further extension of the system. The implementation of this system employs Spring of Java in the server side and AngularJS in the browser side. With this combination, developers could build up and update a system rapidly, and at the same time guarantee its reliability and efficiency. Users will benefit from the system by optimizing their workflow and reducing errors.

Key Words: Course Scheduling System; Java; Spring; AngularJS ; Browser /Server Architecture

1、绪论

1.1 研究的背景

在互联网服务逐渐成为不可或缺的基础设施的当下，办公无纸化的普及也成为了可能。得益于硬件成本的降低、软件技术的发展，以节省成本、降低错误率、提高效率为目的定制中小型互联网服务已经成为一种性价比较高的模式。学校信息系统因为教学制度和专业设置的差异性，一般情况下是需要定制的。

大量的国内的学校的排课系统仍运行在疏于维护的客户机和服务器结构上，系统无法满足的业务需求，经常需要导出成Excel文档手动修改，导致工作效率的下降和错误率的上升。同时，系统使用的门槛较高。由于系统基于客户机和服务器结构，所以用户必须安装客户端、安装各类运行环境（如 .NET Framework）、在特定的操作系统（如 Windows XP）上才可以使用，升级客户机上的排课系统客户端的成本较高，系统本身也往往缺乏升级推送和版本控制机制。由于不提供数据接口、提供的数据格式种类繁多，基于老旧系统开发新平台的难度也较大。

不少国外的学校由于对信息系统建设有统一、长期的规划，在初始就考虑到了维护性和可扩展性的问题。他们的信息系统被分割为若干个小型系统，互相利用数据接口通信，排课系统是其中的一环。因为用户的使用习惯和教学需求，需要兼容众多Linux发行版，所以排课系统一般采用浏览器和服务器结构（B/S架构）。对于这样的结构，系统逻辑的控制权在服务器端，不存在升级推送不到的问题。即使部分系统采用的技术变得老旧，系统在整体上可以正常工作，通过对各个小系统的定期维护，可以保证不错的运行效率。

国外的先进经验给我们国内学校应用系统的开发提供了很好的借鉴，该排课系统的开发应采用这类模式，这样可以适应新形势和新需求，做到提升效率、降低开发与维护成本。

1.2 研究的目的和意义

1.2.1 研究目的本文的

研究目的是替代传统的客户机和服务器结构（C/S结构），为学校的教务排课需求设计一个易于维护、响应迅速、基于浏览器和服务器结构的排课系统。

为了达到此目的，这个排课系统的设计应具有前瞻性，应抛弃旧的结构，使用成熟、可靠的新技术。作为信息系统平台建设的一环，为更多系统的接入做好功能和性能的预估和准备。它应当提供数据接口，定立一个数据传输的标准。

1.2.1 研究意义

本文的研究意义是推动新型信息平台的建设、优化现有的工作流程、减少教务工作的错误率，并为更多的系统建设定立数据接口和标准。

由于缺乏规范和限制，学校的信息系统庞杂，各学院的子系统也数不胜数。这样的系统暴露在公网上不仅拖慢了运行效率，而且会造成严重的网络安全问题，设计良好的系统也会为同域下低质量的系统所累。在老旧信息系统逐渐难以维系、新需求涌现的当下，重新设定立数据接口和应用开发规格的标准是一件刻不容缓的、有现实意义的事情。在大数据技术日渐成熟的当下，优秀的信息平台可以为不同用户提供高度定制化的、高效率的数据服务。因此，新型信息平台内的系统应具有高内聚、低耦合的特性，这样的设计可以方便地拆分、组合、筛选数据，有效推动资源的共享与融合。

技术是为需求服务的，优化工作流程和错误率是提高生产率的需求。真正的无纸化、自动化办公无疑将降低时间和人力成本，工作人员可以从繁复的重复作业中被解放出来，可以将精力投入在有建设性的事务中。

1.3 研究的对象

研究的对象是排课系统的业务需求和新技术、标准的设计与选用。

1.4 研究的方法和手段

1.4.1 研究的方法

研究的方法有观察法、调查法、实验法、文献资料法。

观察法：通过使用搜索引擎提供的相关排课系统，结合本校的系统体验，发现和寻找问题、总结优点与缺陷。

调查法：通过走访询问和技术手段推测出系统使用的架构技术，并理解对应的需求。

实验法：从数据驱动和技术驱动等多种角度对需求进行整理、总结、分类，并建立小规模的原型程序验证想法，例如利用传统 Java EE 的三层结构建立系统，利用托管在第三方的 API 服务建立系统等。

文献资料法：通过查阅 PACKT 和 MANNING 出版社的在线技术书籍来获取新技术的资料、通过仔细翻阅本校和他校的学生培养方案进行数据建模。

1.4.2 研究的手段

研究的手段是仔细研读和分析业务需求的文本、参考其他类似系统设计出数据结构、根据数据结构和业务需求选定技术、查找技术的文献、做出预览版系统以验证想法。

1.5 研究的基本框架

本文研究的基本框架如下：

第一章，绪论。本章主要介绍了研究的背景、目的、意义、对象及方法手段，对研究进行了纲要性概述。

第二章，系统关键技术介绍。本章主要介绍了研究的主要技术选型，并概括了技术的简单原理和选用理由。

第三章，需求分析与系统建模。本章主要介绍了研究所实现系统的需求及基于其设计的数据结构。

第四章，系统功能实现。本章主要以源程序和图例的方式介绍了运用相关技术实现需求的方式。

第五章，结论。本章主要总结了研究的不足及成果。

2、系统关键技术介绍

2.1 架构技术

2.1.1 单页面应用

单页面应用 (Single-page Application) 是一种在使用过程中不进行浏览器重载的应用，也是一种从服务器加载的富客户端^[1]。传统的富客户端是基于客户端和服务器结构的，单页面应用的设计则致力于在Web浏览器上提供跨平台的、和桌面应用类似的用户体验。单页面应用会接管浏览器的页面刷新功能，通过应用内的脚本来实现，且在初始化时就完成整个应用所需的脚本和页面元素的载入。

在 JavaScript 和 HTML5 技术成熟之前，单页面应用通常以 Flash 作为整站的前台技术实现。JavaScript 在性能和功能性上快速发展后，开发者得以用原生 DOM 和 JavaScript 脚本的方式，将页面上的数据通过 AJAX 和应用内路由等技术动态更新，因此它同样对移动设备友好。

除了用户体验的优势外，使用 JavaScript 的单页面应用有诸多优点。

对比传统多页面 Web 应用，它可以使浏览器端和服务器端的代码解耦合，使得在相关技术领域有经验的开发者得以专注于一端的代码。得益于解耦，服务器程序不必再渲染模板、输出页面，因为无需重载每个页面，用户也无需发出很多无用的请求，同样减轻了服务器压力。

对比桌面程序，得益于 Web 应用面向浏览器的特点，单页面应用可以兼容主流操作系统，同时不用担心客户端版本不一致的问题。

使用 JavaScript 的单页面应用也有缺点，首先是运行效率的问题，首次载入较慢、JavaScript 引擎的运行效率对比桌面程序还有差距；其次是兼容性问题，由于采用了新的脚本技术，对于旧式浏览器的兼容性也较差、对搜索引擎支持不好。

综上，在有统一办公环境的内部系统的建设上，单页面应用是一个可靠的选择。

2.1.2 无状态服务

前后端解耦的 Web 应用的一个重要特性就是服务无状态（Stateless）。无状态是指，对于某个服务来说，接收的每个 HTTP 请求之间都是相互隔离的。当客户端发起 HTTP 请求，它就包含了所有服务器端需要的信息。服务器端不依赖于之前请求的任何信息^[2]。

由于实现事务的需求和便利性，目前大多数 Web 服务都是有状态的。例如使用 Session 的服务就是有状态的，Session 经常被用于存储用户登录的验证信息。使用无状态 Web 服务的用户每发送一次请求，其身份信息等数据就包含在请求里，服务端只接受身份信息、不存储信息。

在大数据和云计算兴起的时代，服务程序的可扩展性成为程序质量的一大标准。有状态的服务在这方面受到限制，需要保证同一个服务在多个服务器中的状态一致，否则就会出现数据不一致的问题，有状态服务一旦被扩展，对服务器和网络性能都会有较高要求。

实现服务无状态的最大挑战之一就是身份验证。在初次验证后，用户的验证信息就必须存储在本地。因为无法控制存储介质，所以要保证信息的不可逆性和可控的时效性。

本文实现不可逆性的方案是使用 JSON 网页令牌（JSON Web Token）和 HS256 加密算法加密数据，对于每个用户，服务都会在每次登陆生成非唯一加盐令牌，并利用服务器端脚本将令牌信息写入每次请求的头信息（Header）中，服务接收到请求后，每次都要验证头信息，再判断是否向用户提供涉及业务逻辑的服务。

若需要进一步保证安全性，每个令牌需要可控的时效性以免被盗后被恶意重复使用，可以使用数据库存储每个令牌的过期时间，再对令牌的有效性进行判断。

2.1.3 数据传输

继 XML 之后，得益于 JavaScript 的大规模应用，JSON（JavaScript Object Notation）成为了又一个对面向对象程序友好的数据传输格式，本文中所采用的 JAVA 亦对 JSON 的读取和生成有良好的支持。

如图 1、图 2 所示相较 XML，JSON 的响应体更小、和 Web 的互通性更强，带来了性能的提升^[3]。但相对地，它的可读性不比 XML。在实际应用中，XML 更多

作为应用程序的配置文件存在。考虑到对 Javascript 的原生兼容，将 JSON 作为服务器端和浏览器端数据传输的介质是首选。

```
{  
    "name": "中国",  
    "province": [  
        {"name": "黑龙江",  
         "cities": {  
             "city": ["哈尔滨", "大庆"]  
         }  
     }, {  
        "name": "广东",  
        "cities": {  
            "city": ["广州", "深圳", "珠海"]  
        }  
     }, {  
        "name": "台湾",  
        "cities": {  
            "city": ["台北", "高雄"]  
        }  
     }, {  
        "name": "新疆",  
        "cities": {  
            "city": ["乌鲁木齐"]  
        }  
    }]  
}
```

```
<?xml version="1.0" encoding="utf-8"?>  
<country>  
    <name>中国</name>  
    <province>  
        <name>黑龙江</name>  
        <cities>  
            <city>哈尔滨</city>  
            <city>大庆</city>  
        </cities>  
    </province>  
    <province>  
        <name>广东</name>  
        <cities>  
            <city>广州</city>  
            <city>深圳</city>  
            <city>珠海</city>  
        </cities>  
    </province>  
    <province>  
        <name>台湾</name>  
        <cities>  
            <city>台北</city>  
            <city>高雄</city>  
        </cities>  
    </province>
```

图 1 JSON 数据格式

图 2 XML 数据格式

2.2 浏览器端技术

2.2.1 浏览器端的 MVC 模式和 AngularJS 框架

随着 Web 应用的复杂度日渐提高，为了降低请求次数和提高服务器的响应性，更多的次要的业务逻辑被写进了浏览器端，使用用户的运算资源来实现对数据有效性的检查和其他的功能。

为了迎合类似需求，可以引入 MVC 设计模式，这样可以提高开发效率。

MVC 是 Model-View-Controller 的缩写，其中 Model 是数据模型，View 是视图，Controller 是控制器。在实际应用中，数据模型是从服务获取的数据，视图是 HTML 中的 DOM 元素，控制器就是填入业务逻辑的脚本文件。

视图和控制器的数据一致性问题在成熟的前端框架出现之前一直是影响开发效率的一大问题，每次控制器或视图的数据更新，都需要手动维护另一端的数据，造成了很多数据一致性的问题。

AngularJS 的出现解决了这个问题，它通过脏检查（Dirty-Checking）机制自动维护了两端数据，即“双向数据绑定（Two-way Data Binding）”。

AngularJS 是 Google 旗下的 JavaScript 浏览器端应用开发框架。AngularJS 则通过原生的 MVC 功能增强了 HTML，可以快捷和愉悦地构建出令人印象深刻并且极富表现力的客户端应用。^[4]它不同于传统的 jQuery 等工具库，提供了一种全新的、用设计模式模块化构建浏览器端应用的方式，且用双向数据绑定等便利实用的技术特性加速了开发周期。

双向数据绑定的基本原理如图 3 所示：

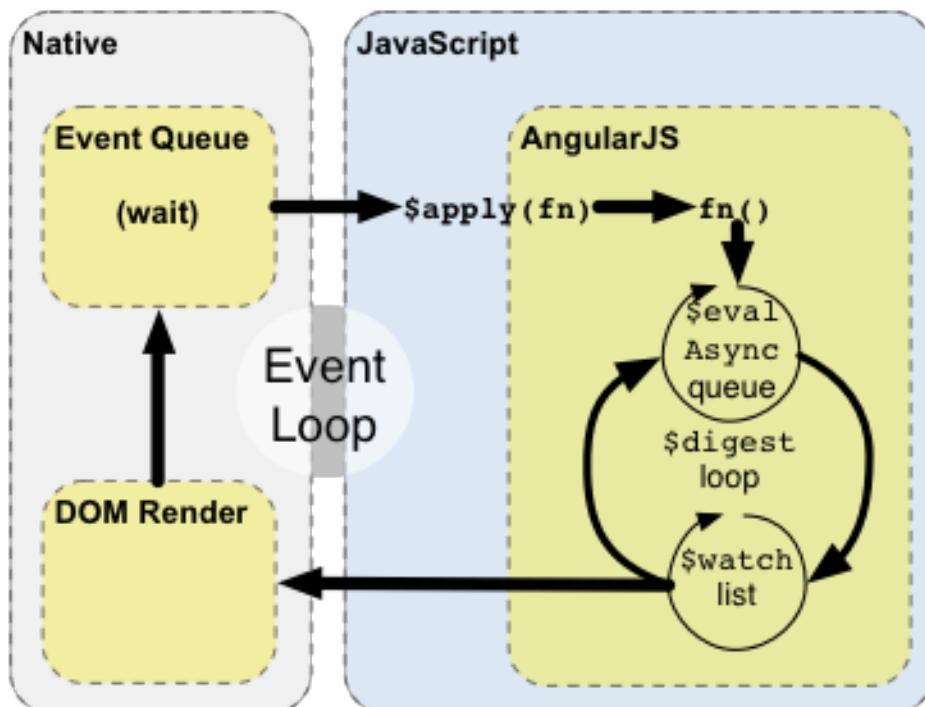


图 3 双向数据绑定

如果采用有固定时间间隔的遍历值检查，程序的效率将大大降低 AngularJS 采用了脏检查机制，每当用户触发了一个事件（例如点击、发送请求等），

AngularJS 就会进行一次数据轮询，在新的数据模型生成完毕后（\$digest 循环），更新视图。

指令系统（Directive）和视图组件化得益于 MVC 设计模式，开发者可以抽出一部分的视图和控制器将其封装成组件，用语义化标签的形式复用在多个页面中，并在页面相应的控制器中为组件填入不一样的数据模型。

在 AngularJS 中，指令系统可以实现这样的功能。它可以使用指定的数据模型动态编译组件的 HTML，注入当前的视图中，并且可以在组件的视图中使用和当前控制器隔离的子控制器，他们使用不同的作用域（\$scope），亦可以有选择地互相通信。

AngularJS 在实际开发中也有缺点。

首先是效率问题，由于数据绑定的实现复杂，当页面有过多的双向绑定的数据模型，界面会产生卡顿，具体的情况取决于客户机的性能。

其次是操作粒度和类库兼容性的问题。AngularJS 更适合复用界面组件，无法像 jQuery 这样的工作类精确地操控每个 DOM 元素，不适合动画效果复杂的网页。AngularJS 在引入常用的第三方组件库（例如 jQuery 实现的 Datepicker）时，常常需要将组件封装在指令系统中不然可能会引发数据绑定失效的问题。

最后，由于双向绑定的特性，在逻辑复杂的页面中，难以确认问题产生在视图还是控制器，增加了调试的难度。

2.2.2 界面组件

在内部系统的开发中，视觉设计人员和浏览器端技术人员往往需要协同工作，因为界面效果在内部系统的重要程度不如功能性，很可能没有专职设计人员。为了方便浏览器端的技术人员在实现业务逻辑后可以保证视图界面的美观统一，往往需要将重复使用的界面元素组件化。

本文采用的界面组件为 Angular Material，是 AngularJS 的官方团队负责维护的项目。它用 Angular 的指令系统重写了最基本的网页元素，例如单选框、复选框、按钮、弹窗等。

根据系统的实际需求，往往需要更复杂的组件，本文在其基础上二次开发，实现了表格、向导等多个自定义的界面组件。

2.2.3 自动化构建和模块化开发

模块化开发是当应用的复杂度变高，需要让每个模块、每个脚本专注于单一职责而形成的一个开发模式。模块化开发不仅需要确定代码的组织形式，还要确定零散的代码最终如何合并、编译、压缩。

本文中，代码的组织形式为按功能需求分模块，每个模块为一个文件夹，文件夹中的每个脚本文件仅包括一个组件，命名方式为“功能.组件类型.文件后缀”，组件分为 Controller、Service、tmpl（模板）等，Service 获取数据模型，Controller 填写业务逻辑，tmpl 是 HTML 视图。

将零散的代码和库合并、编译、压缩是必要的，而且过程要尽量自动化。不然因为请求多、文件体积大，加载的速度会大幅劣化，每次使用都进行声明和注入无疑会增加很多冗余代码、减缓开发效率^[5]。

本文采用了 ng-annotate、gulp 和 browserify 来管理组件的自动依赖注入和自动化构建，gulp 是基于 node.js 的开源自动化构建程序，它可以通过运行特定的脚本来执行一系列任务，gulp 管理了 ng-annotate 和 browserify 的执行，其中 ng-annotate 通过注解式标注会添加对应 angular 类库的依赖注入信息，browserify 监视目录内 js 后缀和 html 后缀文件的改动并自动编译脚本、刷新页面。

在开发周期完结后可以执行 gulp 的 build 命令将 JavaScript 的业务代码和所需类库压缩混淆为一个文件，减少了脚本文件体积的同时也增加了逆向工程的难度。

2.2.4 构建混合型桌面应用

在一些特定的应用场景下，会出现用户希望继续使用过时计算机软件的情况。适配以 IE6 为代表的过时软件是成本高昂的，也会大大降低开发效率和系统的安全性。作为替代方案，本文使用了 NW.js 构建混合型桌面应用。

NW.js 不仅可以在一个窗口里加载本地网页，而且可以利用 JavaScript 的接口同操作系统交互^[6]。

仅需一个重定向页面，就可以实现类似桌面应用的效果，可以将所需文件打包为单文件 EXE，这对不熟悉计算机系统的用户是一个相对便利的方式。

2.3 服务器端技术

2.3.1 Spring Boot

Spring 框架是 JAVA EE 应用程序的主流解决方案之一，它提供了众多的框架功能，例如依赖注入、AOP 等。AOP 可以将业务逻辑中的横切逻辑分离，例如权限控制、日志记录等^[7]。

随着框架变得越来越复杂，XML 配置文件变得繁多和杂乱，构建起一个能够运行的应用也越发困难。

Spring Boot 的出现解决了一部分这样的问题，用户可以向导式地生成一个默认的应用，可以自定义的项目包括采用 Maven 或 Gradle 来管理包、Hibernate 或 Spring Data JPA 来接管持久层等。每个必须的参数都有常用的默认配置，开发者也可以针对某些参数进行自定义。配置文件也采用更贴近面向对象方式的 JAVA 类。

2.3.2 办公文档处理

Microsoft Office 的格式是办公系统中最为流行的系列，一个常见的需求是把这些数据格式化并填入数据库。

通常的做法有将文件保存成 XML 格式通过模板引擎读取数据，或采用能够读取和操作 Office 文档格式的 Apache POI。本文采用了后者，使用 I/O 流进行浏览器端和服务器端交互。I/O 流抽象是用于处理硬盘或其他源发出的连续字节流，它的实现足够低层，可以保证数据的兼容性^[8]。

Apache POI 可以将 Excel 和 Word 以面向对象的形式构造出来，采用了单例模式后，对单元格的操作也是面向对象的。尽管抽象地操作一个文档并非易事，但编写强大的工具类后，可以实现对有固定格式的表格化数据的按需导入导出。

2.3.3 数据持久化

数据持久化是将内存中的瞬时数据转化为数据库中的持久数据的过程，在面向对象的数据持久化的语境下，它就指利用 SQL 在关系数据库中存储数据^[9]。一般需要一个和表结构有映射关系的实体类，由数据持久化工具维护两者的关系。

本文选用了成熟可靠的 MySQL 作为数据库，它是使用最广泛的关系型开源数据库。它包括了常用的关系型数据库的大部分功能，且逐步支持了存储过程和

触发器等高级功能^[10]。数据持久化工具为 Spring Data JPA，并将 Hibernate 作为 JPA 的实现框架。Spring Data JPA 提供了一系列强大的内建自动化查询逻辑，在需求不复杂的情况下，只需定义一个接口，根据方法名字和传入参数，它就会自动生成查询的 HQL 语句，交与 Hibernate 翻译为数据库能够识别的 SQL 语句。

3、需求分析与系统建模

本系统的核心功能包括导入培养方案、制定执行计划、并班并课和输出报表。为了实现这些功能，必须建立模型和与其对应的实体类。

模型（Model）是开发人员与用户之间的沟通工具，它只关注核心业务^[10]，它可以用可视化的图表来表示，如图 4 所示。

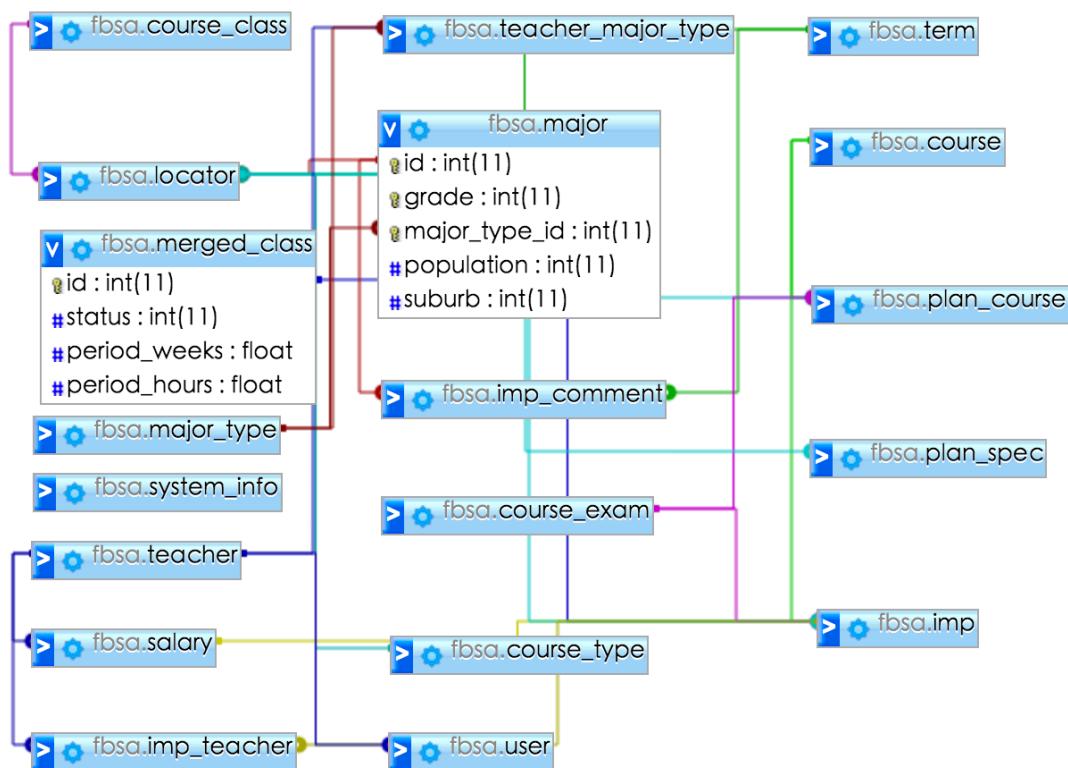


图 4 本系统的全部模型

本例中实体类和模型是相互映射关系，数据结构基本一致。以下仅对核心业务逻辑（排课部分）做介绍。

3.1 系统核心功能需求概述

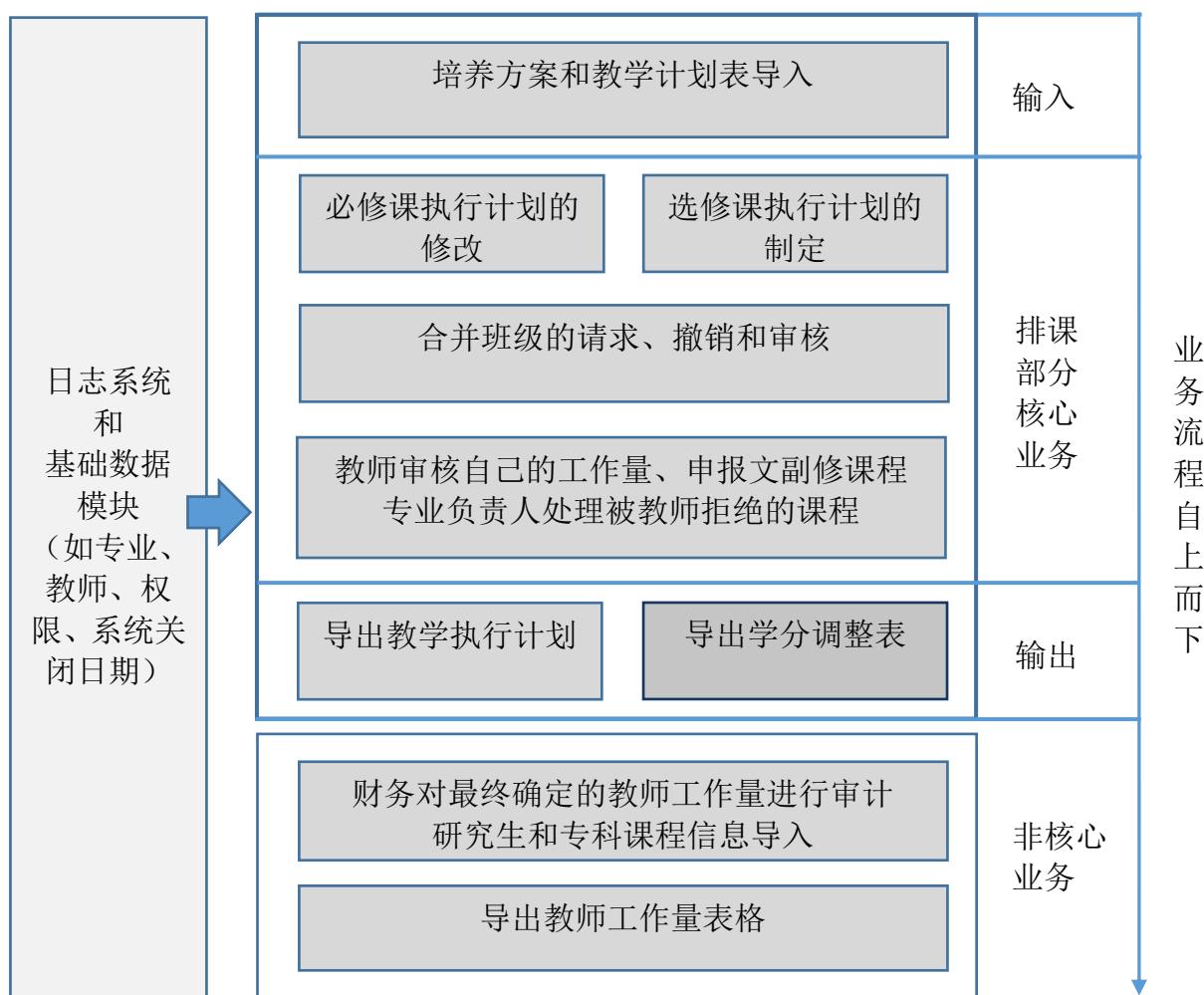


图 5 本系统的功能模块图

核心功能的操作流程是导入数据、制定计划、导出数据，完整的功能模块图如如图5所示。对于不同的课程类型，这三个步骤的业务逻辑有些许出入。

必修课的导入仅有教学计划表一个数据源，教学计划表如图6所示。课程会在此指定。选修课的导入有两个数据源：教学计划表和培养方案，如图7所示。教学计划表并不指定详细的课程明目，仅给出对应类目的学分规则。此类目最多分两级（例如限定选修课下的综合素质类课程）。培养方案会给出可选修的课程列表。

制定计划时，不能删减或增加必修课，仅可以更改开设学期。选修课可以从列表中选择，允许不符合教学计划表的情况，在系统内仅作提示。

表二：四年制本科经济学专业（中美合作）教学计划表

类别	课程代码	学年 学期	一		二		三		四		总计		考核方式		
			学时A 学分B		A	B	A	B	A	B	A	B	A	B	
			A	B	A	B	A	B	A	B	A	B	A	B	
专业必修课	4720034	概率论与数理统计					3	3							48 3 ✓
	4720028	*综合英语	6	4	6	4	3	3	3	3					288 14 ✓
	4720041	英语口语	2	2	2	2									64 4 ✓
	4720043	英语写作					2	2	2	2					64 4 ✓
	4720042	英语听力	2	1	2	1	2	2							96 4 ✓
	4720006	*微观经济学			3	3									48 3 ✓
	4720007	*宏观经济学					3	3							48 3 ✓
	4720019	*商务统计					3	3							48 3 ✓
	4720008	*会计学					4	4							64 4 ✓
	4720035	管理会计									3	3			48 3 ✓
专业课	4720018	*商法			3	3									48 3 ✓
	4720001	#商务基础	2	2											32 2 ✓
	4720104	*金融学					3	3							48 3 ✓
	4720066	*产业经济学							3	3					48 3 ✓
	4720089	*国际金融							3	3					48 3 ✓
	4720091	*国际贸易							3	3					48 3 ✓
限定选修课	4720105	*经济学说史									2	2			32 2 ✓
	4720064	*财政学							2	2					32 2 ✓
	4720003	*管理学					3	3							48 3 ✓
		综合素质类课程					2	2	2	2					2 64 6 ✓
		学院素质教育课程			2	2									32 2
任意选修课		创新创业教育课程	2	2	2	2									64 4
		专业限定选修课	3	3			9	9	10	10	8	8	2	2	512 32
		任意选修课					2	2							32 2

图 6 教学计划表

表四：任意选修课教学计划表				
系列	课程代码	课程名称	总学时	学分
学院任选课	4780008	高级英语听力	32	2
	4780005	TOEFL 英语	32	2
	4780002	#商务基础	32	2
	4780010	金融英语	32	2
	4780004	EXCEL 在经济管理中的高级应用	32	2
	4780009	管理学综合	32	2
	4780150	中级经济学	48	3
	4780006	大学数学综合	32	2
	4780151	综合素质类课程	20	2

图 7 培养方案

导出的数据格式均为Excel文档，内容包括教学执行计划表和学分调整表，如图8和图9所示。表格必须遵循由教务处指定的格式。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O													
1	经济学（中美合作）专业教学执行计划																											
2	学院（盖章）：								专业代码：0701																			
3	第 1 学期								学制：		四年																	
4	本学期原培养方案规定的学分	公必计划学分	15	专必计划学分	9	限选计划学分	0	任选计划学分	0	实践计划学分	0.5	本学期计划合计学分数	24.5															
5	本学期学院实际要求完成的学分	公必要求学分	5	专必要求学分	0	限选要求学分	13	任选要求学分	5	实践要求学分	0	本学期要求合计学分数	23															
6	说明：	本学期学院实际要求完成的各课程类型学分，应是依据各专业培养方案并在实际开课基础上要求该专业学生应完成的学分数。 一般选修课环节中开课学分原则上应大于或等于计划学分。 除选修课环节，实际要求完成的公共必修课、专业必修课和实践性环节的学分数应与下表中实际开课的学分一致。																										
7	关于申请对本学期培养方案实际执行进行调整的说明：																											
8	测试导出																											
9	本学期实际开课情况汇总表																											
10	课程类型	课程代码	课程名称	考核方式	学分	教学周数	周学时	是否学位课程	教师工号	任课教师	教师职称	备注																
11	公必	0000014	大学英语		4	8	4	否	02515,03 222	李凤, 士 研	中级, 高级																	
12	公必	0000013	体育	考查	1	8	2	否																				
13	本学期公必修课开课学分合计：																											
14	本学期专业必修课开课学分合计：																											
15	限选	4740374	创新战略与创业实务	考试	2	8	4	否																				
16	限选	4740003	管理学	考试	2	8	4	否																				
17	限选	4740004	经济伦理学	考查	1	8	2	否																				
18	限选	4740008	统计分析与SPSS应用	考查	2	8	6	否	02515	李凤	中级																	
19	限选	4740281	中级微观经济学	考试	2	16	2	否	02515	李凤	中级																	
20	限选	4740002	管理信息系统	考查	2	8	4	否																				
21	限选	4740007	企业战略管理	考查	2	8	4	否																				
22	本学期限选修课开课学分合计：																											
23	任选	4740374	创新战略与创业实务	考试	2	8	4	否	03651,03 222	黄静, 士 研	副高, 高级																	
24	任选	4740003	管理学	考试	2	8	4	否																				
25	任选	4740004	经济伦理学	考查	1	8	2	否																				
26	本学期任意选修课开课学分合计：																											
27	本学期实践性环节学分合计：																											

图 8 教学执行计划

上海师范大学专业教学计划学分调整表(学分制)

年级： 2015		学院： 商学院		专业代号： 0701		专业名称： 经济学（中美合作）（全称）													
课程类别	学分	学年		一		二		三		四		总计							
		总学分	学期	1	2	3	4	5	6	7	8	A	B	A	B	A	B		
公共必修课	5			21		6		7	4	5		2					46	4	
专业必修课				17	11	9	6	12	12	8	8	5	5				51	42	
限定选修课	13			14		2				2		6					37	0	
文修核心				2		2							2				6	0	
讲座													2				2	0	
任意选修课	5			2				4									11	0	
实践性环节				1.5				1		2		3.5					13	21	0
毕业论文													6				6	0	
总计		23	0	57.5	11	19	6	20	16	21	8	16.5	5	2	0	21	0	180	46

图 9 学分调整表

3.2 课程

3.2.1 课程功能需求

课程自身的属性有课程代码和课程标题。

一门课可分为不同阶段，例如基础法语（一）、基础法语（二），但它们可以拥有同一个课程代码。对于系统，这样语义上不同阶段的一门课应该划分成多门课来对待。

同一门课程对于不同专业可以分为不同类型，这样的分类至多有两级，至少有一级。课程可以由多个老师同时或分别教授给多个专业。课程是否学位课因专业而异。

3.2.2 课程的模型

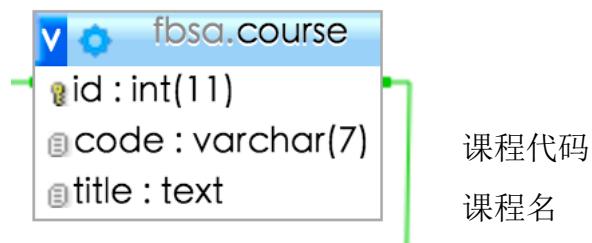


图 10 课程的模型

如图 10 所示，课程的模型仅有一个课程实体类 course。和课程概念相关的更多属性（例如是否学位课）由于因专业而异，所以更应由其他的实体类接管，或者划为公共属性。根据需求特性，课程代码不能作为确定唯一课程的主键，所以用数据库分配的自增 id 来确定唯一的课程。

3.3 培养方案

3.3.1 培养方案需求

培养方案的数据源是一张有固定格式的 Excel 表格，属性依据表格而建，包含了四年所有的课程数据。导入培养方案的同时，系统就必须生成对应的课程实体。

单个培养方案条目包含的数据有课程的所有属性、专业、开设学期、学分、学时、是否学位课。

必修课在培养方案中就指定了开课学期，选修课在培养方案中只给出待选列表，不指定开设学期。

3.3.2 培养方案数据模型

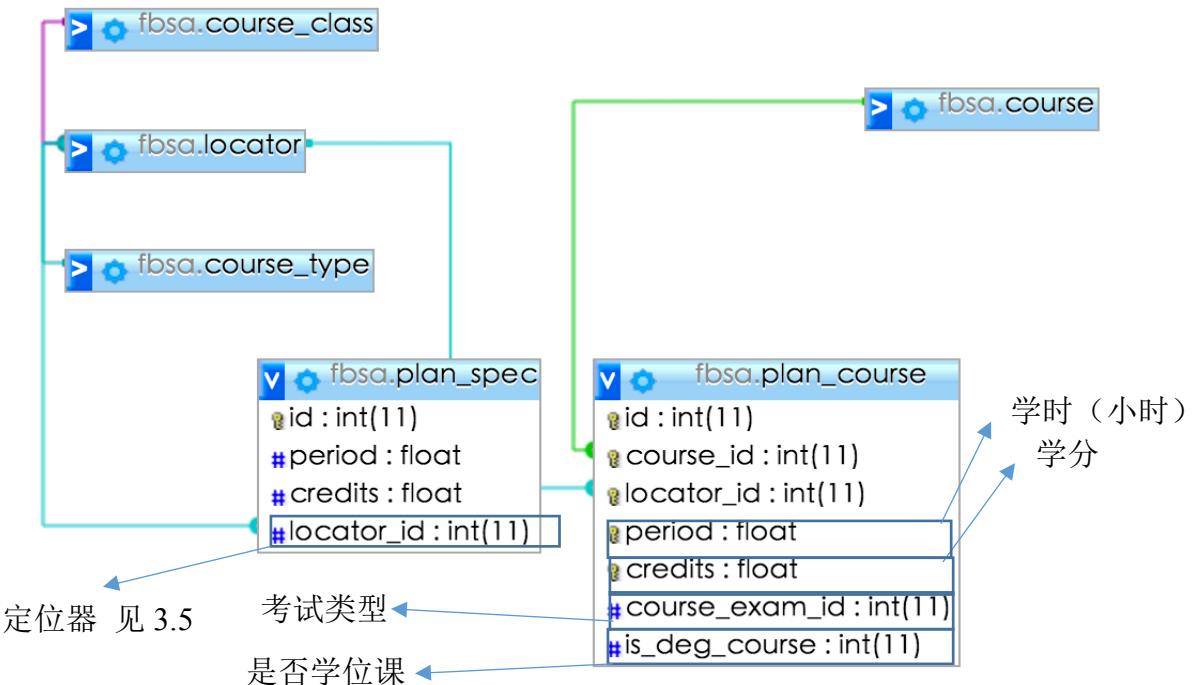


图 11 培养方案的模型

如图 11 所示，培养方案的模型有两个实体类 `plan_course`、`plan_spec`。

`plan_course` 存储必修和选修课程的培养方案条目，`plan_spec` 存储课程类型的培养方案条目。

3.4 执行计划

3.4.1 执行计划需求

一个执行计划以一个专业的一个学期为单位，同一门课的执行计划条目和培养方案条目可能有出入，每个执行计划条目都可以有备注。

必修课的执行计划条目在初始化时和培养方案完全一致，且不能在执行计划中移除，可以更改开设的学期、考核方式和教师。

选修课可以在执行计划中被增删，可以更改的属性为开设的学期、考核方式和教师。

对于每个专业的每个学期的执行计划，教务员都可以单独输入备注，这个备注会在导出的教学执行计划表中体现。

3.4.2 执行计划数据模型

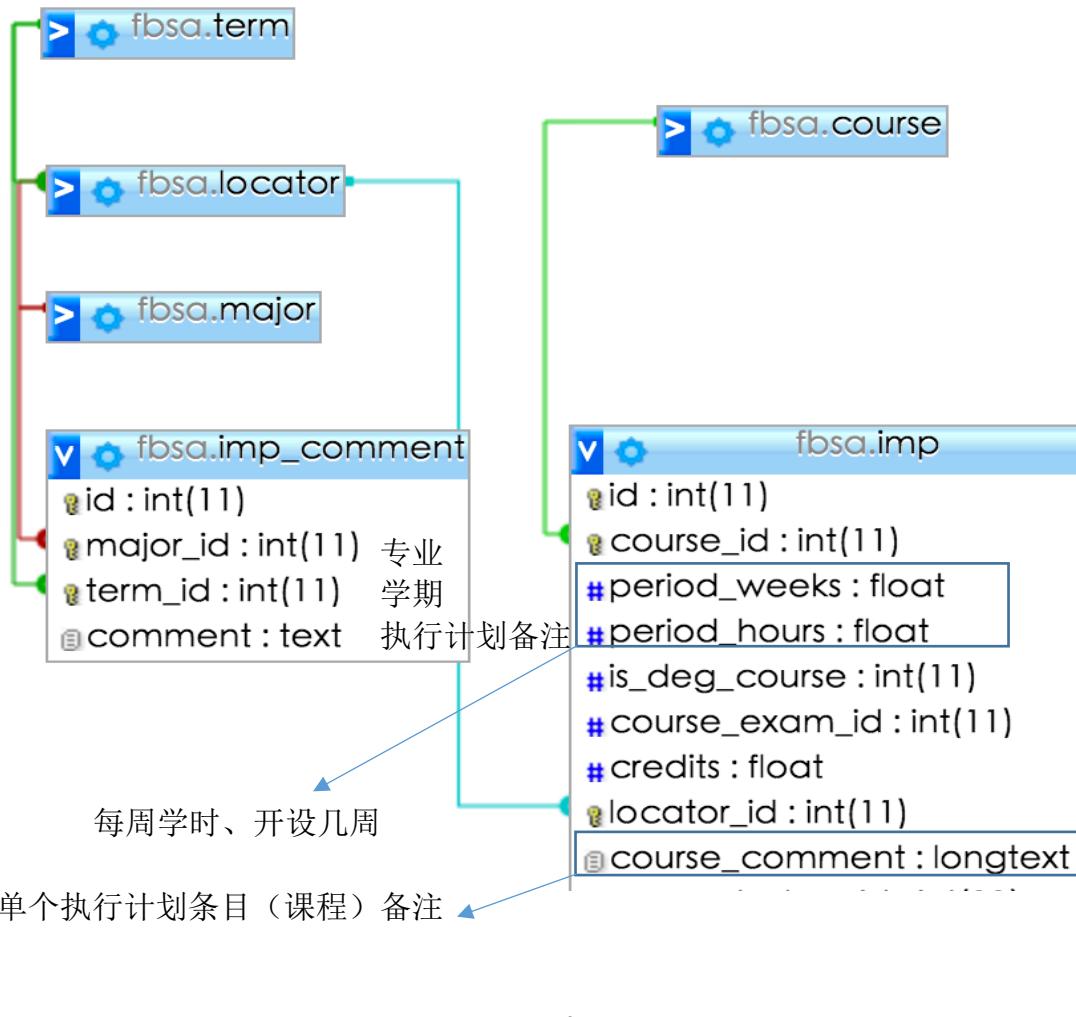


图 12 执行计划的模型

如图 12 所示，执行计划的模型有 **imp**（执行计划条目）和 **imp_comment**（执行计划备注）。**imp** 存储执行计划条目，**imp_comment** 存储对于整个执行计划的备注信息。

3.5 公共属性

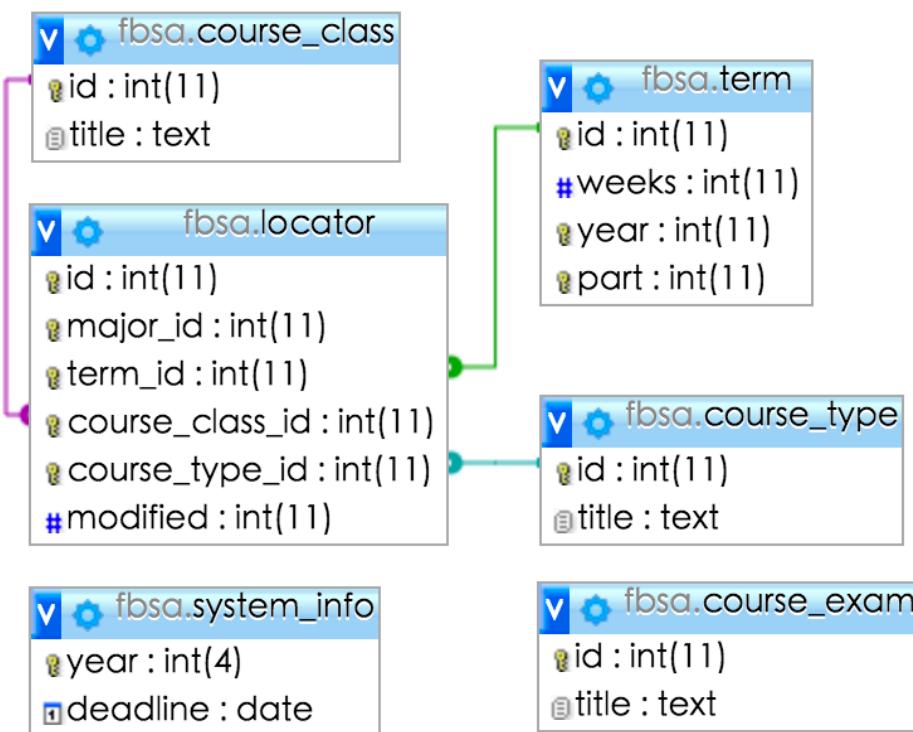


图 13 公共属性的模型

如图 13 所示，课程分类的两个属性，实体类分别为 `course_class`（第一级，例如限定选修课）和 `course_type`（第二级，例如综合素质类课程）。

课程的考核方式属性，实体类为 `course_exam`。

学期的实体类为 `term`，包含的信息有学年（`year`），上半/下半学期（`part`），周数（`weeks`）。

定位器（locator）是公共属性最重要的实体类，它包含四个可以为空的属性：课程第一级分类、课程第二级分类、学期、专业，以及用于确定是否编辑过的 `modified` 属性。它的功能是确定唯一的培养方案条目和执行计划条目。由于数据持久化技术对于多主键的处理是生成主键临时实体类，所以本系统中就直接将它定义为一个实体类，同时也便于浏览器端交互的业务逻辑。

根据需求，还需要一个系统信息实体类 `system_info`，用来存储系统关闭时间。

3.6 教师和用户

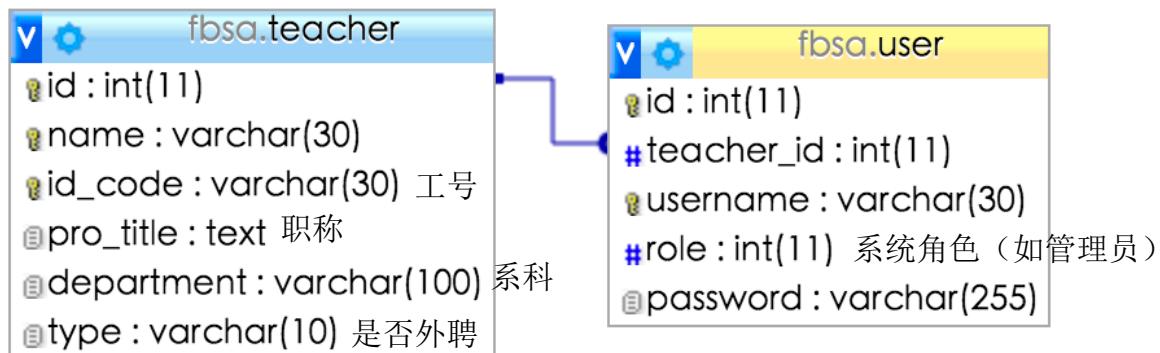


图 14 教师和用户的模型

如图 14 所示，教师的实体类为 teacher，用户的实体类为 user。

一个用户对应一个教师，一个用户仅有一种角色属性。

4 系统功能实现

4.1 身份验证

用户身份验证系统是管理类 Web 服务必须的功能。对于浏览器端和服务端分离的无状态 Web 服务来说，身份验证系统的设计不能使用 Session 的同时，要保证本地信息的不可逆性和时效性，要保证服务端的安全性和准确性。

4.1.1 令牌系统

令牌系统是一整套用户身份验证的解决方案，它提供了基本用户名/密码验证的方法。有三种用户名令牌：仅用户名、明文密码、密码摘要，一般后两种使用较多^[12]。本文采用的是第三种。

浏览器端：登录验证服务

```
function authService($q, $log, $http, $cookieStore){
    var vm = this; vm.role = [];
    function logout() {
        vm.user = {};vm.role = [];// 清空用户数据
        $cookieStore.remove('token'); $http.defaults.headers.common['Authorization'] = '';
        // 清空令牌和 header 的 Authorization 数据
    }
    function doLogin(user) {
        return $http.post(AUTH_CONFIG.url + 'login', user).then(function(response) {
            return response.data.token;// 向服务器端发送请求
        });
    }
    function login(user) {
        return doLogin(user).then(function (token) {
            vm.user = user;vm.token = token; $cookieStore.put('token', token);
            // 设定所有的请求 Header 附带令牌信息
            $http.defaults.headers.common['Authorization'] = 'FBSA ' + token;
        },
        function (error) {
            $log.error(error.status);
            vm.user = {};
            return $q.reject(error.status); // 指定 Promise 的状态为 Reject
        });
    }
}
```

服务器端对登录请求的处理:

```
// 登录时验证和获取令牌的代码:  
public class AuthFacade {  
    // 注入用户相关的业务逻辑类  
    @Autowired  
    UserRepository userRepository;  
  
    @RequestMapping(value = "/login", method = RequestMethod.POST)  
    public LoginResponse login(@RequestBody final UserLogin login)  
        throws ServletException {  
        if (login.username == null) {  
            throw new ServletException("Invalid login");  
        }  
        User user = userRepository.authOK(login.username, login.password);  
        return new LoginResponse(Jwts.builder()  
            .setSubject(login.username)  
            .claim("userId", user.getId()) // 令牌附带信息  
            .setIssuedAt(new Date()) // 签发时间  
            .signWith(SignatureAlgorithm.HS256, "FBSASECRET!").compact());  
        // 加密算法和盐  
    }  
}
```

服务器端对一般请求的处理:

如图15所示，发送的一般请求中的Headers会带有验证信息，需要用过滤器拦截所有对”/api/*” URL的请求。

过滤器(Filter)常被用于过滤有害字符和进行验证，它可以拦截对特定URL的请求，并对请求携带的内容进行处理和判断。^[13]



图 15 发送的一般请求

```
public class JwtFilter extends GenericFilterBean {  
    @Override  
    public void doFilter(final ServletRequest req,  
        final ServletResponse res,  
        final FilterChain chain) throws IOException, ServletException {
```

```
final HttpServletRequest request = (HttpServletRequest) req;

final String authHeader = request.getHeader("Authorization");
if (authHeader == null || !authHeader.startsWith("FBSA ")) {
    // 非法 Authorization Header
    throw new ServletException("Missing or invalid Authorization header.");
}

final String token = authHeader.substring(5); // 令牌的主体
try {
    // FBSASECRET! 是令牌加密盐
    final Claims claims = Jwts.parser().setSigningKey("FBSASECRET!")
        .parseClaimsJws(token).getBody();
    request.setAttribute("claims", claims); // 令牌附带的信息
}
catch (final Exception e) {
    throw new ServletException("Invalid token."); // 非法令牌
}
chain.doFilter(req, res);
}
```

4.1.2 权限控制

浏览器端：登录验证服务（上文 authService 内）

```
function loadAuthInfo(){
    if($cookieStore.get("token") == null){
        return $q.when($state.go('authentication.login'));
    }else{
        return getToken()
            .then(checkRoles) //权限检查，见下
            .then($rootScope.$broadcast("rolesLoadComplete")); //加载完毕信号
    }
}

function checkRoles() {
    return hasRole('1').then(function(admin) {
        return $q.when(vm.role['admin'] = admin);});
}

function hasRole(role) {
    return $http.get(AUTH_CONFIG.url + 'role/' + role).then(function(response){
        return response.data;
    });
}
```

服务器端：

在登录验证的同时判断用户的权限，通过 claim 写入令牌（代码略）。
接收到从浏览器端的请求后，解密令牌的附加信息，获得权限信息。

```

@RequestBody
@RequestMapping(value = "/role/{role}", method = RequestMethod.GET)
public Boolean login(@PathVariable final String role,
    final HttpServletRequest request) throws ServletException {
    final Claims claims = (Claims) request.getAttribute("claims");
    return ((Integer) claims.get("roles")).equals(Integer.valueOf(role));
}

```

4.2 文档处理

对于信息系统来说，最理想的数据源莫过于API式的数据接口。但在大部分情况下，内部办公系统都需要对文档进行处理，对Microsoft Office格式的Doc和Xls文档处理类是不可缺少的。此处以输出一个Excel为例。

文件输出请求的入口定义：

```

@RequestMapping(value = "/o/i/m/{majorId}/t/{termCount}", method =
RequestMethod.GET)
public ModelAndView downloadImpExcel(@PathVariable Integer majorId, @PathVariable
Integer termCount) {
    ImpExcelDTO res = excelService.generateImpExcelDTO(majorId, termCount);
    return new ModelAndView("excelView", "impExcelDTOs", res);
}

```

在此入口会生成一个ImpExcelDTO对象，此对象将会被转换成一个用于表示Excel文件的ModelAndView对象发送给浏览器端。

上文ImpExcelDTO 数据结构完全对应Excel模板文件，如图16所示：

#majorTitle专业教学执行计划									
学院(盖章):							专业代码:	#majorCode	
第 #termCount 学期							学制:		四年
本学期培养方案规定 的学分	公必计划学分	#planPOC	专必计划学分	#planMOC	限选计划学分	#planREC	任选计划学分	#planFEC	实践计划学分
本学期学院实际要求完 成的学分	公必要求学分	#impPOC	专必要求学分	#impMOC	限选要求学分	#impREC	任选要求学分	#impFEC	实践要求学分
								#impTC	本学期要求合计学分
								#impSumC	
#impComment <pre> public class ImpExcelDTO { MajAndTmAndImpComDTO mtdto; ImpExcelHeaderDTO header; List<ImpExcelGridDTO> podtos; List<ImpExcelGridDTO> modtos; List<ImpExcelGridDTO> redtos; List<ImpExcelGridDTO> fedtos; List<ImpExcelGridDTO> tdtos; }</pre>									
本学期实际开课情况汇报									
课程类型	课程代码	课程名称	考核方式	学分	教学周数	周学时	是否 计 入 学 分		
公必	#POcode	#POtitle	#POexam	#POcre dits	#POweeks	#POhou rs	#P O I n c l u d e		
本学期公共必修课开课学分合计:							# P O T O T O T O		
专必	#MOcode	#MOtitle	#MOexam	#Mocre dits	#MOWeeks	#MOhou rs	#M O I n c l u d e		
本学期专业必修课开课学分合计:							# M O T O T O T O		
限选	#REcode	#REtitle	#REexam	#REcre dits	#REWEEKS	#REhou rs	#R E I n c l u d e		
本学期限选修课开课学分合计:							# R E T O T O T O		
任选	#FEcode	#FEtitle	#FEexam	#FEcre dits	#FEWEEKS	#FEhou rs	#F E I n c l u d e		

图 16 Excel 模板文件

ImpExcelDTO对象生成完后，会被传入一个AbstractXlsView 的视图对象。视图会根据ImpExcelDTO的数据生成表示Excel文件的数据流，最后再被转换为 ModelAndView对象返回到浏览器端。代码如下：

```

public class ExcelView extends AbstractXlsView {
    @Autowired
    ImpDao impDao;
    @Override
    protected void buildExcelDocument(Map<String, Object> model,
        Workbook workbook, HttpServletRequest request, HttpServletResponse
        response) throws Exception {
        InputStream is = this.getClass().getResourceAsStream(request.getContextPath() +
        "/impTemplate.xls");           //读入模板文件
        ExcelTemplate template = ExcelTemplate.newInstance(is); //生成单例对象
        ImpExcelDTO imp = (ImpExcelDTO)model.get("impExcelDTOs"); //获取 DTO
        //替换 Header
        ImpExcelHeaderDTO headerDTO = imp.getHeader();
        Map<String, Object> map = new HashMap<>();
        IOAssembler.flushParams(map, headerDTO); //将 DTO 对象转换为 Map
        //根据占位符替换关键字（需要遍历 Excel）
        template.replaceParametersByKeyword(map, "#");
        //替换模板中的执行计划下的各类课程
        List<ImpExcelGridDTO> iePodtos = imp.getPodtos();
        ArrayList<Map> pomapList = new ArrayList<>();
        //生成 Map 对象
        for(ImpExcelGridDTO dto : iePodtos) {
            map = new HashMap<>();IOAssembler.flushParams(map, dto);
            pomapList.add(map);
        }
        //按行输出
        template.createRowByHashMap(pomapList, "#PO");
        //获取 Apache POI 对象
        workbook = template.getWorkbook();
        //输出数据流
        OutputStream ouputStream = response.getOutputStream();
        workbook.write(ouputStream);ouputStream.flush();ouputStream.close();
    }
}

```

4. 3 选课数据交互和业务逻辑

4. 3. 1 简单业务逻辑举例

以课程修改通用模块为例。通用模块如图 17 所示。

执行计划									
课程种类	课程代码 ↓	课程名	教学周数	周学时	学分	教师信息	是否学位课	考试类型	调整
实践性环节	4790037	实践实习（教学实践周）	8周	0小时	1		否		Q
专业必修课	4720040	线性代数	8周	3小时	3		否	考试	Q
专业必修课	4720027	政治经济学（社）	8周	2小时	2		是	考试	Q
专业必修课	4720026	政治经济学（资）	8周	3小时	3	杜文杰、李丹、黄丽筠	是	考试	Q
专业必修课	4720011	基础会计学	8周	3小时	3		是	考试	Q

Rows per page: 5 ▾ 1 - 5 of 16 |< < > >|

图 17 课程修改通用模块

浏览器代码：

点击放大镜按钮后会呼出 `vm.modifyImpCourse(course)`:

```
function modifyImpCourse(course) { //传入
    $mdDialog.show({ // $mdDialog 是组件库 Angular Material 的对话框组件
        controller: 'ImpCourseModifyDialogController',
        templateUrl: 'app/ui/imp/imp-course-modify-dialog.tmpl.html',//模板页面
        locals: {impCourse: course, grade: vm.major.grade}, // 传入课程和专业
    }).finally(function() {reinit(); // 页面刷新});}
```



图 18 排课信息编辑对话框

浏览器端对话框的控制器如图 18 所示。以下是部分实现代码（点击确认后触发的部分）：

```
function okClick() {
    impDataService.updateOneImp(vm.impCourse,termCount)//更新数据
    .then(function(data){ // 服务器端返回更新的信息
        vm.impCourse = data; $mdDialog.hide(vm.impCourse);
```

```
//将更新过的课程信息带回});}
```

浏览器端 impDataService 的更新课程函数：

```
function updateOneImp(impCourse,termCount) {
    var impURI = API_CONFIG.url + 'i/update/t/' + termCount;
    return $http({method: 'POST',url: impURI,data: impCourse}) }
```

服务器端的课程更新入口：

```
@ResponseBody
@RequestMapping(value="/i/update/t/{termCount}",method=RequestMethod.POST,
consumes = "application/json")
public GridEntityDTO updateImp(@RequestBody GridEntityDTO grid , @PathVariable
Integer termCount){
    return impRepository.updateImpByGridEntity(grid,termCount);
}
```

服务器端的课程更新业务逻辑：

```
public GridEntityDTO updateImpByGridEntity(GridEntityDTO entity , Integer
termCount) {
    Imp res ;
    if(termCount>0 && termCount<9) {
        if (entity != null) {
            Imp imp = impDao.findOne(entity.getId());
            Locator locator;
            if(imp==null || imp.getCourse().getId()!=entity.getCourseId()) {
                imp=new Imp();
                PlanCourse pc = planCourseDao.findOne(entity.getId());
                locator = pc.getLocator();imp.setCourse(pc.getCourse());
            }else{locator = imp.getLocator();}
            imp.setCredits(entity.getCredits()[0]);
            imp.setPeriodHours(entity.getPeriod()[0]);
            imp.setPeriodWeeks(entity.getPeriodWeeks());
            imp.setIsDegCourse(entity.getIsDegCourse());
            Integer ctId = 0;
            if (locator.getCourseType() != null) {
                ctId = locator.getCourseType().getId();
            }
            Locator newLocator =
                locatorRepository.getLocator (
                    locator.getMajor().getId(), termCount,
                    locator.getCourseClass().getId(), ctId);
            newLocator.setModified(1);
            imp.setLocator(newLocator);
            CourseExam courseExam =
```

```
courseExamDao.findOne(entity.getCourseExamId());
imp.setCourseExam(courseExam);
int[] teacherIds = entity.getTeacherIds();
List<Teacher> teacherList = new ArrayList<>();
for (Integer tId : teacherIds) {
    if(tId!=0) {Teacher teacher = teacherDao.findOne(tId);
        if (teacher != null) {teacherList.add(teacher);}}
    imp.setTeachers(teacherList);
    if(entity.getComment()!=null && !entity.getComment().isEmpty()) {
        imp.setCourseComment(entity.getComment());}
    res = impDao.save(imp);
    return new GridEntityDTO(res);}
return entity;
}
```

4.3.2 浏览器端复杂业务逻辑举例

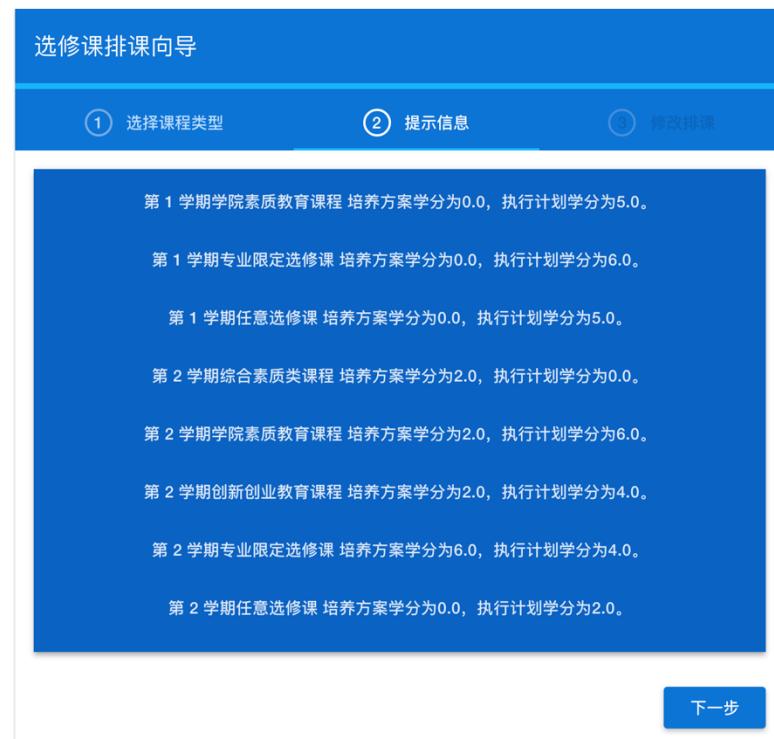
如图19、图20、图21所示，以下是制定选修课执行计划的三步向导式页面：



图 19 向导第一页 - 选择课程类型

执行计划 > 制定选修课执行计划

2015级经济学（中美合作）第3学期 管理



下一步

图 20 向导第二页 - 提示信息

执行计划 > 制定选修课执行计划

2015级经济学（中美合作）第3学期 管理

待选课程列表						限定选修课 - 创新创业教育课程	
课程代码	课程名	考核方式	学分	总学时	添加		
4740372	创新思维与管理	考试	2	32	+		
4740374	创新战略与创业财务规划	考试	2	32	+		
4740375	现代创新专题	考试	2	32	+		
4740376	职业能力提升	考试	2	32	+		

Rows per page: 5 ▾ 1 - 4 of 4 |< < > >|

已选课程列表						限定选修课-创新创业教育课程		
课程代码	课程名	考核方式	学分	周学时	教学周	教师	修改	删除
4740373	创新创业企业战略决策与经营模拟	考查	2	2	16			-

Rows per page: 5 ▾ 1 - 4 of 4 |< < > >|

确认

取消

图 21 向导第三页 - 排课页面

页面控制器：

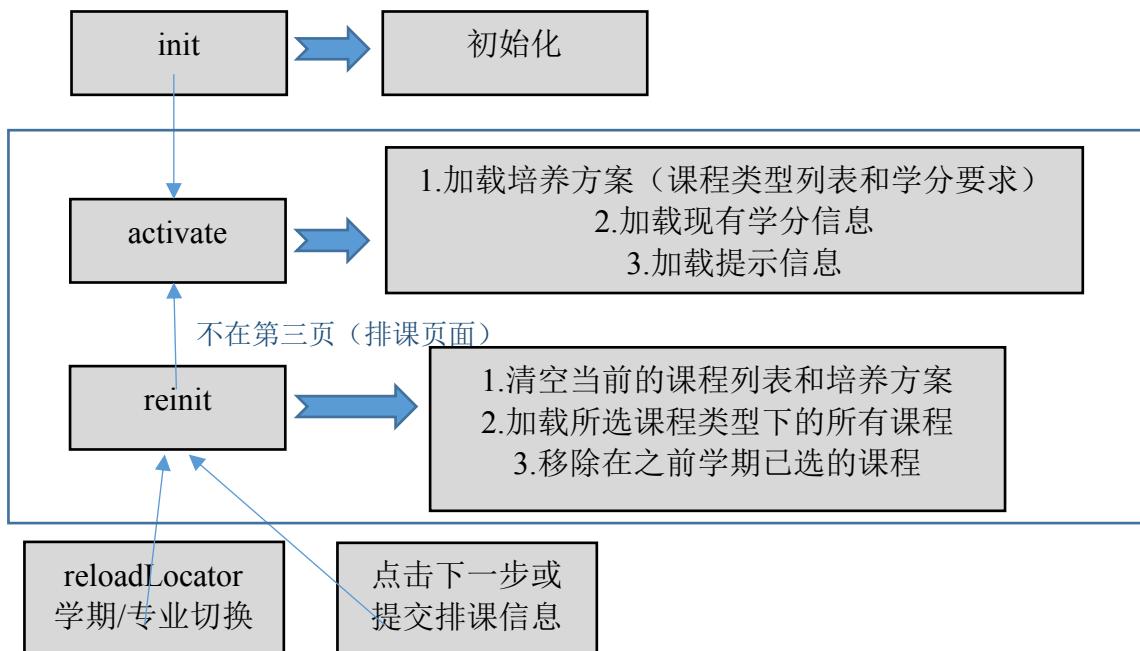


图 22 控制器工作流程

控制器采用了Promise机制来严格控制每个状态中任务执行的步骤。Promise是一种对象或函数，后接一个then方法，在then方法中可以标明之前一个异步操作执行的状态。Promise有三类状态：Pending(初始态)、Fulfilled(表示成功)和Rejected(表示失败)^[14]。利用Promise的三类状态，便可以使异步代码顺序执行。执行逻辑如图22所示。

控制器代码举例：

```

function activate() {
    vm.term=impDataService.getTerm(); // 从框架服务获取学期
    vm.major=impDataService.getMajor(); // 从框架服务获取专业
    vm.termCount= (vm.term.year - vm.major.grade) * 2 + vm.term.part;
    vm.selectedCourse=[]; // 已选课程清空
    vm.locators = []; // 课程类型和学分要求的列表清空
    return getGridImp().then(getGridImpCredits).then(function() { //课程类型和学分要求
        return getLocator().then(function () {
            vm.selectedLocator = null; // 选择的课程类型和学分要求清空
            for (var i = 0 ; i < vm.locators.length ; i++) {
                if (vm.locators[i].modified == 0 && vm.locators[i].creditsNeeded !=0){
                    // 如果有未排课的且学分要求不符合的
                    vm.selectedLocator = vm.locators[i]; // 设置一个默认的课程类型
                }
            }
        })
    })
}
    
```

```

        }
        if (vm.selectedLocator == null) { // 查找失败
            vm.selectedLocator = vm.locators[0]; // 默认选第一个
        }
        return getCaution(); // 获取第二页的提示信息
    });
}
}

function reinit(noReload){
    vm.selectedCourse=[];
    vm.availableTermYears=
    [vm.major.grade,vm.major.grade+1,vm.major.grade+2,vm.major.grade+3];
    if(vm.step <= 1 && !noReload){ // 是否在排课页面
        activate();
    }
    return getPlanCourses().then(removeElectedCourseFromEarlyTermsFromPlan);
}

```

单个任务(例如getPlanCourses、getCaution等)的实现方式可参考4.3.1的更新课程函数。

4.4 日志系统

日志系统的需求是将日志写入一个文本文件，以供浏览器端直接读取，如图23所示。日志需要包含操作人的信息和操作时间。

在Java的日志框架中，log4J和logback是定制自由度最高和功能最强大的。其中logback的性能更好、开销更小。本文采用了logback。

Date	Message
2016-01-08 14:58:54	test 调整执行计划
2016-01-10 01:13:07	test 申报文修副修
2016-01-10 01:13:07	test 申报文修副修
2016-02-14 14:15:14	test 调整执行计划
2016-02-14 14:15:17	test 调整执行计划
2016-02-14 14:15:45	test 调整执行计划
2016-02-14 14:16:44	test 调整执行计划
2016-02-14 14:16:53	test 调整执行计划
2016-02-14 14:23:44	test 调整执行计划
2016-02-14 14:26:23	test 调整执行计划
2016-02-14 14:26:37	test 调整执行计划

图 23 日志系统浏览器端页面

实现代码：

在Filter中写入记录每次请求发起人信息：

```
import org.apache.log4j.MDC;  
MDC.put("name", claims.get("name")); //
```

在 Spring 中的配置文件：Logback-spring.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<configuration>  
    <include resource="org/springframework/boot/logging/logback/base.xml"/>  
    <appender name="FILEOUT" class="ch.qos.logback.core.FileAppender">  
        <file>src/main/webapp/log/log.txt</file>  
        <append>true</append>  
        <encoder>  
            <Pattern>  
                %d{yyyy-MM-dd HH:mm:ss} %X{name} %msg%on  
            </Pattern>  
        </encoder>  
    </appender>  
    <logger name="cn.edu.shnu.fb.application" level="TRACE">  
        <appender-ref ref="FILEOUT" />  
    </logger>  
</configuration>
```

可供其他类调用的日志服务：

```
@Service  
public class LogService {  
    private final Logger log = LoggerFactory.getLogger(this.getClass());  
    public void event(String property , String operation , String origin , String result){  
        log.trace("对象属性 {} , 行为 {} , 原值为 {} , 现值为 {}" , property,operation,origin,result);  
    }  
    public void action(String property , String operation){  
        log.trace("{}{}" , operation,property);  
    }  
}
```

调用代码举例：

```
@Autowired  
LogService logService ;  
logService.action("工作量","更新");
```

5、结论

5.1 系统达到的目标

本系统在功能上成功实现了用户的业务需求，提供了在不同系统环境下的
一致体验。

本系统在技术上实现了设计之初制定的目标，提供了数据接口，得以被重用
和二次开发。

本系统在目的上得以减少办公的失误，加快工作效率。

5.2 设计存在的不足

本系统的身份认证是定制实现的，并未接入学校统一身份认证的LDAP服务
器。

为了提供良好的用户体验、保证数据一致性，本系统调用了本地资源进行部
分界面运算，在业务逻辑复杂的页面造成了一定的卡顿现象。

本系统在文档导入时暂未考虑高并发的情况，若需部署到大量用户的环境，
进一步改进文档导入，增加队列系统。

本系统没有对小屏终端进行适配。

参考文献

- 1、Michael S,Josh C.Single Page Web Applications[M].Manning Publications,2013 : 4 .
- 2、Sam R,Leonard R . RESTful Web Services[M] . O' Reilly Media, Inc., 2007 : 86 .
- 3、Casimir S , 王群峰 , 杜欢 . 全端Web开发[M] . 人民邮电出版社 , 2015 : 40 .
- 4、Ari L , 赵望野 , 徐飞 . AngularJS权威指南[M] . 人民邮电出版社 , 2014 : 3 .
- 5、Travis M . Getting Started with Gulp[M] . Packt Publishing, 2015 : 7-8 .
- 6、Paul B. NW.js in Action[M]. Manning Publications, 2016 : 2 .
- 7、黄勇. 架构探险-从零开始写Java Web框架[M] 电子工业出版社 , 2015 : 118-136 .
- 8、Benjamin J,David F,安道 Java技术手册[M].人民邮电出版社 , 2015 : 254 .
- 9、Christian B, Gavin K, 杨春花. Hibernate实战(第2版) [M]. 人民邮电出版社 , 2008 : 2-3 .
- 10、Michael K, 杨晓云, 王建桥. MySQL 5权威指南[M]. 人民邮电出版社 , 2006 : 1-4
- 11、Jimmy N, 赵俐, 马燕新. 领域驱动设计与模式实战[M]. 人民邮电出版社 , 2009 : 3-6
- 12、汪红波, 袁利永 , 汪红刚. 基于用户令牌实现 Web 服务身份验证[J]. 计算机与数字工程 , 2006 : 1 .
- 13、Alex A. Spring Boot Cookbook[M]. Packt Publishing , 2015: 24-25 .
- 14、Muzzamil H. Mastering JavaScript Promises[M]. Packt Publishing , 2015: 2-3 .

15、Gerald G. Enterprise Application Development with Ext JS and Spring [M].

Packt Publishing , 2013: 153-160 .

附 录

商学院排课系统用户手册

附录. 1 推荐运行环境

本系统可在主流操作系统（Windows、Mac OSX和大部分Linux发行版）上运行。用户需经Web浏览器访问本系统，推荐的浏览器为Google Chrome 45、猎豹5.3、搜狗6.1和360安全浏览器8.1。

附录. 2 系统账户相关功能



附图. 1 登陆界面

进入系统后，如附图. 1所示，首先会出现登录界面。

教职员必须登录系统，以获取相应权限。教职员可以用工号登陆，默认密码和工号相同。

财务等特殊角色的帐号密码须咨询教学办。



附图. 2 修改密码界面

完成首次登陆后，用户应修改默认密码。

修改密码的入口在左边栏最底部，修改密码的界面如附图. 2所示。

依据提示输入新旧密码并点击更改密码即可。



附图. 3 注销按钮

如附图. 3所示，点击左侧的注销可安全退出系统。

2013级经济学（中美合作）第5学期 管理员

附图. 4 专业切换对话框入口

如附图. 4所示，点击系统界面右上方的专业名，可以切换负责的专业和学期。



附图.5 专业切换对话框

如附图.5所示，可根据界面提示切换专业和学期，此切换操作在应用全局即时生效。

附录.3 导入课程数据



附图.6 导入方案界面

如附图.6所示，在培养方案 > 导入方案中可以导入课程数据。

课程数据由两个文件组成，培养方案的xls文件和选修课的doc文件。

系统不支持xlsx和docx文档，如现有资料为以上格式，请将文件另存为xls

和doc格式后进行导入操作。点击 选择需要上传的文件。

课程种类	课程代码	课程名	学时	学分	总学时	总学分	考核方式											
公共必修课	0000031	马克思主义基本原理	3	3												3	3	考查
公共必修课	0000013	体育	2	1	2	1	2	1	2	1						8	4	考查
公共必修课	0000004	计算机	4	3	4	3										8	6	考查
公共必修课		高等数学	4	4	4	4										8	8	考试
公共必修课	0000020	思想道德修养与法律基础			3	3										3	3	考查
公共必修课	0000032	中国近现代史纲要			2	2										2	2	考查
公共必修课	0000005	军事			1	1										1	1	考查
公共必修课	0000035	形势与政策				2										2		考查
公共必修课	0000034	毛泽东思想和中国特色社会主义理论体系概论（一）					4	4								4	4	考试
公共必修课	0000084	就业指导							1	1						1	1	考查

每页显示列数: 10 1 - 10 共 39 条 < >

附图. 7 显示方案界面

如附图. 7所示，导入完成后，可以在显示方案中查看数据是否正确。

若数据有误，可重做一遍课程导入，当前所有的课程数据将会被清空。

附录. 4 制定必修课执行计划

课程种类	课程代码	课程名	周学时	学分	是否学位课	考试类型
公共必修课	0000084	就业指导	1	1	否	考查
专业必修课	4720091	国际贸易	3	3	是	考试
专业必修课	4720064	财政学	2	2	是	考试

每页显示列数: 10 1 - 3 共 3 条 < >

课程种类	课程代码	课程名	教学周数	周学时	学分	教师信息	是否学位课	考试类型	调整
专业必修课	4720091	国际贸易	16周	3	3		是	考试	调整
专业必修课	4720064	财政学	16周	2	2		是	考试	调整
公共必修课	0000084	就业指导	16周	1	1		否	考查	调整

Rows per page: 5 1 - 3 of 3 |< < > >|

附图. 8 必修课执行计划界面

如附图. 8所示，专业负责人在此页面可以制定必修课执行计划。

顶端为选课改动的提示信息，显示当前专业在所有学期的必修课程变动信息和备注。

中段的培养方案细则框则是计划中的方案。

底部的框是现有的执行计划，默认和计划的数据一致。若要修改，则需点击底部列表最右侧的“调整”按钮呼出排课信息编辑对话框。

第 4 学期执行计划中新增了《毛泽东思想和中国特色社会主义理论体系概论（一）》(备注：公必课《毛泽东思想和

国际贸易 排课信息编辑

×

课程代码	4720091
课程类型	专业必修课
学分	3
是否学位课:	是
周学时	教学周数
3	16
开设学期:	考核方式:
<input type="radio"/> 2013-2014	<input checked="" type="radio"/> 第一学期
<input type="radio"/> 2014-2015	<input type="radio"/> 第二学期
<input checked="" type="radio"/> 2015-2016	<input type="radio"/> 考查
<input type="radio"/> 2016-2017	
教师:	
姓名	搜索
备注信息	0/150

确认修改 放弃修改

附图. 9 排课信息编辑对话框

教师:

王虹 03327 中级 × 李凤 02515 中级 ×

姓名
李

附图. 10 添加课程教师

如附图. 9所示，在此处可以调整一门课的开设信息及录入课程开设备注。

如附图. 10所示，目前，仅提供姓名匹配添加教师，支持模糊查询，支持添加多个教师。

附录. 5 制定选修课执行计划



附图. 11 制定选修课计划向导 第一页



附图. 12 为不在计划中的选修课程类型排课

如附图. 11所示，此页选修课执行计划制定页面的第一页，整个流程采用三步向导式。在第一页可以选择需要制定计划的课程类型。

本页默认只显示有学分计划的课程类型，若需显示没有计划的课程类型，需点击“选修课程调整”展开名，效果如附图. 12所示。



附图. 13 制定选修课计划向导 第二页

如附图. 13所示，点击某一类型课程后，系统会显示本专业第一学期至当前学期的选修课计划调整的提示信息。若无调整，此信息将被略过。

待选课程列表 限定选修课 - 专业限定选修课					
课程代码	课程名	考核方式	学分	总学时	添加
4740185	商务交流	考查	3	48	<button>添加</button>
4740072	风险管理	考查	3	48	<button>添加</button>
4740208	投资管理	考查	3	48	<button>添加</button>
4740281	中级微观经济学	考查	2	32	<button>添加</button>
4740259	中级宏观经济学	考查	2	32	<button>添加</button>

已选课程列表 限定选修课-专业限定选修课							
课程代码	课程名	考核方式	学分	周学时	教学周	教师	修改 移除
4740019	GATEWAY	考查	3	3	16		<button>修改</button> <button>移除</button>
4740102	国际金融市场	考查	3	3	16		<button>修改</button> <button>移除</button>

附图. 14 制定选修课计划向导 第三页（排课页面）

如附图. 14所示，此页为选修课排课页面，上方为可选的课程列表，下方为已选的课程列表。专业负责人可以通过点击“添加”和“移除”调整排课计划，可以点击修改呼出排课信息编辑对话框（见上），并修改课程的属性。

若需额外加入列表外的课程，可点击“添加课程”。添加的课程会直接进入已选列表，且除后不会被存储在待选列表里。

附录. 6 数据导出



附图. 15 数据导出页面

如附图. 15所示，此界面仅对专业负责人和管理员可见。

此处可以点击下载按钮（）导出相关专业对应学期的教学执行计划和学分调整表，并可输入执行计划的备注。

致 谢

本论文是在导师杜文杰的悉心指导下完成的。恩师高超的学术能力、辩证的思维方式、严谨的研究态度和独特的思想见解在技术上为本人提供了更高的视野。她在为人等方面所给予的教导和意见，亦使本人深受启发。

同时，要衷心地感谢卢长智老师和上海师范大学商学院教学办对我论文的技术和业务指导。他们不仅在工作安排上给予本人足够的学习机会和包容态度，而且也十分关心本人的学习状态，提出建议、及时指导，使得本人便于将在课堂上学到的知识恰当运用和深化。

在此，本人再一次向所有给予本人支持和关心的老师、同学、亲人和朋友表示最诚挚的感谢！

作者签名：

日期：2016 年 3 月 25 日