

1 绪论

1.1 研究背景

随着生产社会化趋势的扩大、科学技术的进步、人类知识总量的增长速度不断加快、以及市场竞争的日益激烈，使人们对信息的认识产生了根本性的变化。信息被列为与物质、能源并列的人类社会发展的三大资源之一。信息化水平已成为衡量一个国家现代化水平和综合国力的重要标志。我国也正处于改革开放进一步深化的环境下，因此我们应当抓住机遇，充分利用信息，扎实的做好信息管理系统的基础工作，建设开发较为先进的应用系统，加快我国信息化建设的步伐。

邓小平同志指出“教育要面向现代化、面向世界、面向未来”。信息社会的到来已预示着电脑网络会在教育技术现代化中扮演重要角色，在实施教育技术现代化的过程中我们深刻认识到，在教学及考试的过程中应用网络化的手段解决传统的教师选课已经迫在眉睫，系统就是建立在这个基础上才得以全面、健康地发展。

目前，大多数高等院校对于教师选课仍采用人工填表的方式，教务管理部门通常根据教学计划准备课程表，并要求教师在选课的同时，填写有关课程和教材的大量信息，然而无论是教学计划还是教师所需教授的课程及其教材，和以往的学期相比往往改动甚小，这样做不但增加了很多的重复劳动，执行起来效率低下，容易出错，而且在进行整理集合之时，工作量很大，也难以进行统计分析。采用信息系统技术与数据库技术，建立合理的选课系统，不但能及时有效的解决教学管理人员安排课程和教师选课的问题，减少重复劳动，提高工作效率。同时在后续安排教学任务的时候也可以利用信息化的数据代替手工输入，以及利用自动化整合统计技术对数据进行进一步的挖掘与分析，以便于进一步建设信息化，现代化的教务系统。

1.2 研究目的及意义

本文的研究意义可以从理论意义和现实意义这两个方面来阐述。

1.2.1 从理论意义方面来说，随着互联网规模的扩大以及信息技术的成熟，计算机软硬件技术发展到了相当高的水平。利用信息技术以提高工作效率的思想正在影响着各行各业，在进行信息化建设过程中，教育行业是领头羊，教务是教育的重要保障，大多数高等院校在进行教务信息化过程中，都采用半手工、半信息化，半线上、半线下的方式。在进行课程安排及教师选课方面，目前相关的教务管理系统有很多，但是相关的功能不全或者设计有缺陷，无法满足教师选课流程的复杂需求，不能同时方便教师或者是教学管理人员使用。无法真正做到高效率的信息化，这个环节迫切需要建立信息化管理系统进行管理，以做到承上启下的作用，推动整个教务信息化的过程。

1.2.2 从现实意义来说，设计完成的教师选课系统的使用对象为教师及学院教学管理人员。教学管理人员可以利用系统对不同班级建立课程库并进行管理，根据教学计划的要求生成课程表以便于教师选课，教师在选课的过程中可以根据往年的情况，自动生成相关课程、教材信息，减轻教师的工作负担，提高教师的工作效率。

1.3 国内外研究现状

1.3.1 国外研究现状

国外高校在计算机与互联网信息技术上有着先天的历史优势,教育机构往往作为信息化建设的先锋,国外对教务信息化的研究比较多,但在教师选课及课程安排等各个方面的管理制度上面,与国内相差较大,国外开发的教务系统往往难以适应中国国情,往往只在技术上提供相关的支持与参考。

国外有以 WebCT 和 Blackboard 为代表的教学管理系统,包含选课功能,被全球超过 2800 所大学及其他教育机构使用。其中, WebCT 最初是由加拿大英属哥伦比亚大学资深教师 Murray Goldberg 于 2007 年自行开发的网络教学平台,目前其产品的市场占有率在同类产品中已经位居全球第一。WebCT 是完全基于 Web 浏览器的网络教学课件开发和管理系统。在功能方面,除了具备网络教学平台提供的基本功能外,主要有以下两个主要特征:拥有相对强大的教学跟踪功能;具有独特的选择性发布功能。而 Blackboard 是从 Course Info 平台发展而来的,目前已经发展成为两大网络教学平台之一。相对 WebCT 而言, Blackboard 对内容组织采用文件夹和子文件夹形式,层次清晰明朗。

关于教育系统, Bates 和 Holton(1995)对其作出解释,认为教务取决于多维建构,方式不同结果自然也就不尽相同。Kahn (1978)等研究学者提出了基于三维分类的教务管理方法,三维分类教务管理方法的思想是将管理的内容分为三类,第一是首先被管理内容要能够长期存在于教务组织中,第二是被管理对象要能够完成组织统一制定的教学任务,第三是在完成既定的任务和目标后,被管理目标能够完成规定之外的对组织发展有利的任务。在三维分类教务管理方法中,后两个内容对于被考核对象的整体考核影响较大,也就是说三维分类管理比较注重目标自发的管理贡献。

1.3.2 国内研究现状

自建设信息化以来,国内对于教务系统方面的相当丰富,绝大部分的高等院校都在一定程度上使用了各种各样的教务系统,这些系统往往是根据用户的大致需求开发的,功能比较多,以能较好地满足当时用户的实际需要。但很容易造成一个巨大而低效的系统,表现出投资大、效果不明显;水平低、重复建设多;硬件投入多、软件投入少;模拟手工作业增加管理负担等。这是因为开发商水平参差不齐,开发队伍缺乏相关制度及流程上的经验,同时不同的高校在制度及管理方面差别较大。导致部分软件产品质量不高,定做开发又会因用户需求变化要对系统进行不断的修改维护。到最后,使整个系统的性能和可使用性都大大降低。

当下高校教务管理系统以 B / S 结构为主辅以 C / S 架构,其核心架构是将应用 移植到系统浏览器上,浏览器载入用户交互界面用以用户操作完成管理系统需求。在国内方面清华大学一直处于国内高校教务管理系统的前列,其教务管理系统采用了 C / S 与 B / S 的混合框架进行开发。

针对教师挂牌上课制度,陈建中(2001)指出了教师挂牌上课制度的实践意义和理论意义,并针对可能会出现的问题提出相应的解决方法。

张玉玲(2013)针对新疆财经大学的教师挂牌授课制度进行分析,主张应广泛推广教师挂牌授课制度,同时也指出采用教师挂牌授课制度需要进一步完善的地方。

在选课系统的软件开发方面，刘乃嘉（2011）等详细分析了清华大学网上选课系统的性能，并通过缓存技术、物化表技术等新计算机新技术对选课系统进行了合理优化。

李淑堃（2012）开发了一种基于手机短信的选课系统，弥补了 Web 选课系统的不足，实现两条信息渠道优势互补，增加选课系统的灵活性。

1.4 研究内容与方法

1.4.1 研究内容

本课题对高等院校在课程安排及教师选课环节的现状进行需求分析并开发教师选课系统，主要针对教师选课中的‘四定’进行安排和管理。在功能上不求大而全，只求精而细、以适应用户的实际需求。

本文是按照敏捷式开发流程即：“需求分析—系统设计—开发实现—测试及留存接口”的框架模式行文。其中需求分析及开发实现是本文的重点研究内容。

本文共包括六部分，各部分内容如下：

第一部分是绪论。此部分介绍了本文的研究背景，研究目的与意义，国内外对此课题的研究现状，相关概念的界定以及本文的研究内容。

第二部分是需求分析部分，不同的课程和时期在制度上对于教师选课的规定有所不同，在这一部分详细说明需求分析中遇到的问题及解决方案。

第三部分是系统设计部分，阐明系统的工作流程，对项目进行可行性分析、主要设计方案及思路。

第四部分是开发实现部分，根据需求分析以及系统设计，分别对系统前后端开发的内容进行说明。

第五部分是测试部分，对系统进行比较完善的测试，以保证系统的高可靠性，高维护性，并说明留存的必要接口以供系统扩展及再次开发。

第六部分为结论部分。此部分对全文进行了回顾与总结，指出此文章研究的主要工作以及不足之处，并对今后的研究方向进行展望。

1.4.2 研究方法

2 需求分析

2.1 系统的设计目标

本系统设计时为了方便学院安排教学计划，对教学内容进行信息化管理，对教师的课程进行管理和统计以及教师对所教授的课程进行选课、安排课程内容、选定教材，打印四定报表。主要实现以下目标：

- 网页操作流畅，界面美观大方。
- 系统运行安全稳定。
- 充分为用户着想，注重用户体验。
- 提供必要的筛选与查找功能，方便数据展示。
- 双端支持，能够在 PC Web 端和移动端进行操作。

2.2 详细需求分析

2.2.1 组织机构

系统的组织机构根据其学校的院系机构进行设计，分为学院、系部、专业三级，不同的学院之间是相互独立的。在从属关系上，教师从属于系部；课程从属于学院；班级、教学计划从属于专业；而教材库则是系统统一设定，不在学院之间独立开来。

2.2.2 课程与教学计划

教务管理人员可以对其学院的课程库进行管理，设置课程的参数，对于课程的一些参数如学时可以在教学模板中再次设定，已适应相同课程在不同专业不同的情况。

在每学期的教师选课时段开始前，教务管理人员根据各个专业的教学计划模板生成本学期的教学计划，在教学计划模板中指定相应的课程和上课的年级，以及其他参数，在设定教学计划模板时，可以给模板设置多个专业，以适应一门课程多个专业上课的情况。在生成教学计划时，如果一个专业同年级有多个班级，则会将这多个班级分配到同一个教学计划。生成教学计划后，教务管理人员也能够对教学计划进行新增、修改和删除，教务管理人员还可以根据班级查看本学期及以往各学期的教学计划。

对于性质不同的课程对选课教师的要求不一致，如有些专业课只允许本系教师选课，教务管理人员可以在教学计划模板中指定选课教师的范围。

课程人数默认按照所有上课班级的总人数来定，教务管理人员可以对课程的上课人数进行修改。

2.2.3 教师选课

在选择教授课程时，教师需填写所需教材以及相关信息。所需教材的信息可以在个人中选取对应的教材信息，也可以自己添加教材内容并保存到教材库中，在添加教材时，教师可以添加多个教材，但需指定其中一个教材作为主教材，在打印四定报表时，只会打印主教材的详细情况，而其他教材只会打印教材名称、ISBN 号及出版信息。

教师在选课时，默认选择所有课时，对于需要多个教师一起上课的课程，教师在选课的时候，只需指定自己上课的课时，当一门课的课时没有被选择完时，这门课可以继续被其他教师选择，在打印四定报表时，只会输出第一个选课教师所填写的选课信息。

2.2.4 教材管理

教师可以添加教材，所添加的教材会被新增到教材库中以及个人教材中，对于教材库中已有的教材，可以直接添加到个人教材中，在移除教材时，只会将教材从个人教材中移除。教师在选课过程中也可以直接填写教材信息，填写的教材信息也会如同新增的教材一般添加到教材库及个人教材中。

2.2.5 账户与身份

在系统中，权限与身份绑定，不同的身份进行不同的操作，主要身份有以下三种：

- 系统管理员
- 教务管理人员
- 教师

系统管理员的主要功能是维护系统的基础数据以及配置系统基本设置。在系统中班级信息以及教师信息由系统管理员进行初始化和维护，系统管理员拥有其增、删、改、查的所有权限。系统管理员也能够对教材库进行编辑和删除。在系统中，教务管理人员也由系统管理员进行制定。

教务管理人员主要在系统中对课程以及教学计划的安排和管理，教务管理人员可以对课程进修增、删、改、查的基本操作，除此之外，教务管理人员可以修改针对于专业的教学计划模板，并根据其模板自动生成相应的针对于班级教学计划，对于自动生成的教学计划，在选课时间段内，教务管理人员也能对其进行增、删、改、查的基本操作。对于选课之后的生成的四定表，教务管理人员可以进行查看和下载。

教师是系统中的主要用户，教师可以进行选课操作以及对个人教材管理，教师在选课后也能在选课时间段内取消选课。

在系统中还需要一个超级管理员用户用来分配管理员账户，这个超级管理员用户可以由系统生成，保证系统的完整性。

登录系统的账户可以绑定多个身份，账户可以绑定一个教师信息，在添加教师时会自动生成一个账户，并且默认绑定教师身份。

3 系统设计

3.1 系统的设计模式

3.1.1 MVC 模式

MVC 模式是一种规范，即模型(model)、视图(view)、控制器(controller)，将业务逻辑、数据、界面显示分离的方式组织代码，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

3.1.2 数据与页面分离

在系统设计过程中，后端利用 **AJAX** 发送数据给前端，前端处理数据展现页面及页面渲染，使页面能够快速响应，提升系统流畅性。

3.1.3 响应式布局

响应式布局是一个前端设计模式的概念，为了解决移动互联网浏览的问题，开发人员只需要一次编码就能为不同终端的用户提供舒适的界面和的用户体验。本系统采用响应式布局以解决移动端及 **PC** 端双端适配问题。

3.1.4 单页面模式

单页面模式是网站只开放单个 **url** 连接并加载单个 **HTML** 页面并在用户与应用程序交互时动态更新该页面的 **Web** 应用程序，操作过程中无需进行 **url** 跳转，窗口的打开与关闭都使用弹出层实现，不仅能给用户带来良好的体验，还能减少网络延时及设备因素对系统流畅性的影响。

3.2 相关技术和工具

3.2.1 开发技术介绍

本系统采用了 **python** 语言作为系统的开发语言，基于 **Web** 的三层体系结构的应用，利用数据与页面分离技术对数据库的动态查询、统计、更新等功能。将计算机网络、数据库、**Internet** 技术结合为一体，实现跨平台、界面统一美观、易于维护和扩展的教师选课系统。

3.2.2 框架介绍

➤ Django 框架介绍

Django 是一个开放源代码的 **Web** 应用框架，由 **Python** 写成。采用了 **MVC** 的框架模式，**Django** 能在极短的时间内构建全面动态的网站。**Django** 的主旨是让你集中精力在有趣的工作上，减轻重复劳作的痛苦。为此，它为常用的 **Web** 开发模式提供了高层抽象，为常见的编程任务提供了捷径，还为解决问题提供了清晰的约定，特别适合“内容型”网站开发。

➤ layui 框架介绍

layui 发布于 2016 年的经典前端框架，是一款采用自身模块规范编写的前端 **UI** 框架，遵循原生 **HTML/CSS/JS** 的书写与组织形式，区别于那些基于 **MVVM** 底层的 **UI** 框架，是为服务端程序员量身定做，框架自身对于浏览器兼容性极佳，无需涉及复杂的前端配置，可作为 **PC** 网页端后台系统与前台界面的速成开发方案。

3.2.3 中间件介绍

➤ uWSGI 介绍

uWSGI 是一个 Web 服务器，它实现了 WSGI 协议、uwsgi、http 等协议。WSGI 是一种 Web 服务器网关接口。它是 Web 服务器与 web 应用通信的一种规范，uWSGI 在处理动态网页上有很高的性能，因此特别适合作为 Django 等遵循 WSGI 接口框架的动态页面中间件。

➤ Nginx 介绍

Nginx 是一个很强大的高性能 Web 和反向代理服务器，在一个 BSD-like 协议下发行。采用事件驱动，占有内存少，并发能力强，具有很多非常优越的特性，在连接高并发的情况下，能够支持高达 50,000 个并发连接数的响应，并且几乎可以做到 7*24 不间断运行，即使运行数个月也不需要重新启动。

3.3 总体设计

3.3.1 业务流程

➤ 输入输出流：

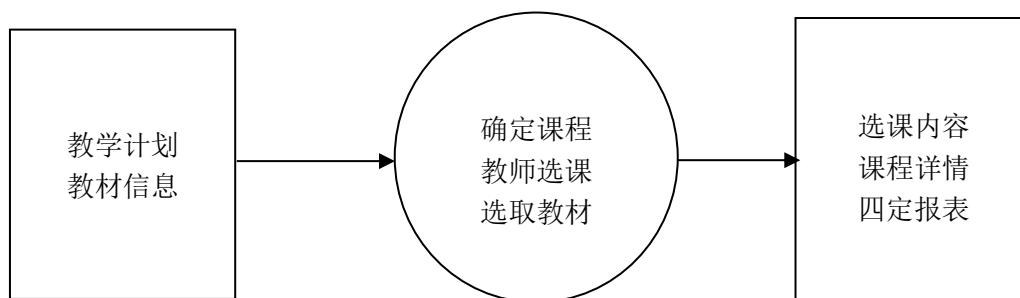


图 3.1 要输入输出流图

➤ 操作流程：

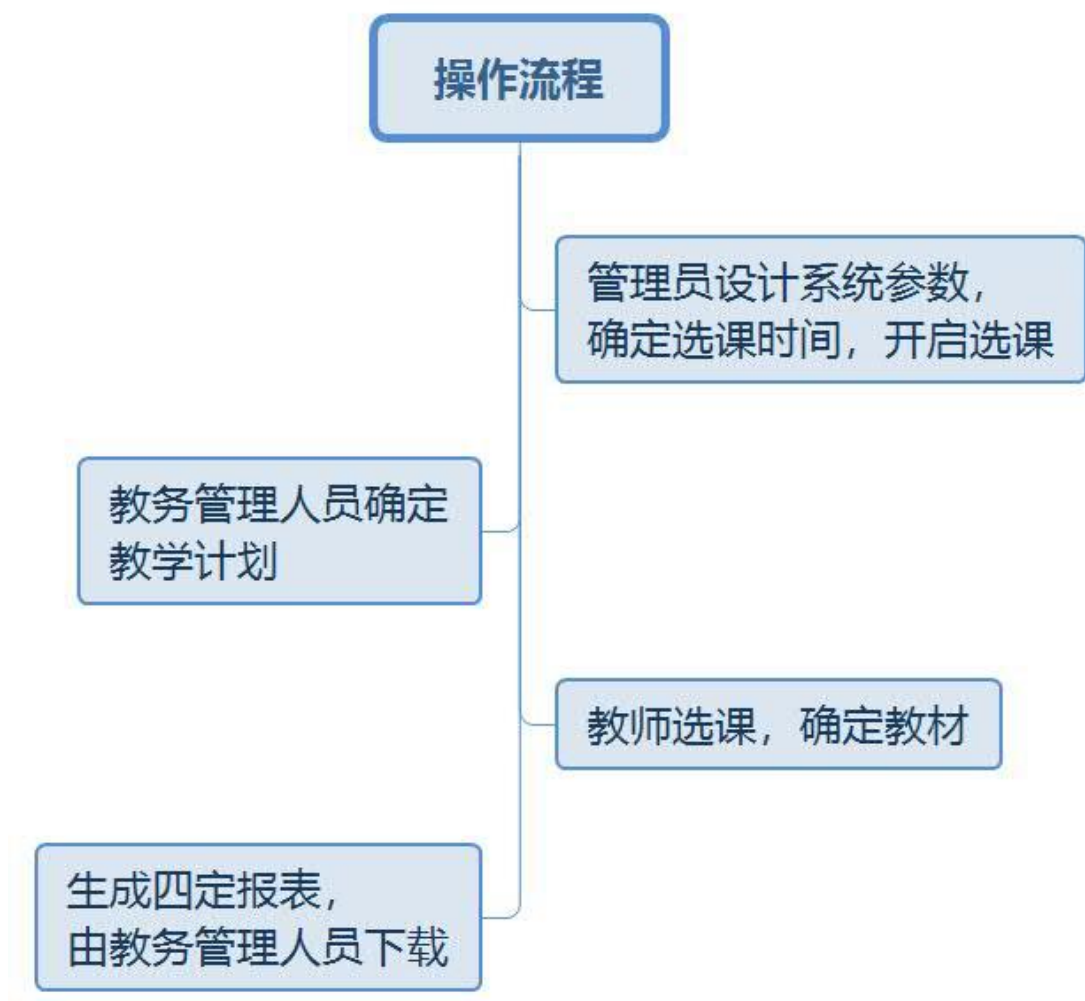


图 3.2 操作流程图

3.4 模块设计



图 3.3 模块设计图

3.5 数据库设计

本系统使用 MySQL 数据库，对于数据库的设计，采用了基于 Django 框架模型层的数据结构，利用强大的 ORM 进行数据库的连接及操作。除了对数据库基本范式的要求之外，密切配合业务逻辑及查询优化，提高性能。

对于数据表中的字典字段，本系统有针对性地采用了动态字典的配置，以便于能够在系

统使用过程中动态配置选项，以适应复杂的业务需求。

对于数据库中的基础数据，系统将会记录其创建时间和最后一次修改时间，在删除基础数据时，只做逻辑删除，以便于日后进行数据维护及痕迹查询。

在 MySQL5.7 版本以上的数据库中对于 Varchar 等可变长字段进行了优化，一般而言对于长度小于 1000 字节的字段在进行存储和检索时会根据其性能只与其实际长度有关系。考虑到系统的扩展性及优化，在数据库设计时不进行严格的考虑，而在前端及后端逻辑中进行限制。

3.5.1 数据表清单

本系统的数据表分为三种：业务逻辑表、系统表和多对多关系表，其中业务逻辑表存储着本系统业务逻辑中产生的数据或者所必须的基础数据，这一部分数据时系统业务的基石；系统表是系统运行及构架中不可缺少的数据表格，这一部分数据表支撑着系统的运行；多对多关系表则是表与表之前的多对多关系而产生的关系表。

➤ 主要业务逻辑表

数据库表名	解释	所属模块	关联外键表
organization	组织机构表	system	无
classes	班级表	classes	organization
course_lib	课程库	course	organization
teacher	教师表	instructor	organization
materaial_lib	教材库	teachmaterial	无
plan_model	教学计划模板表	course	course_lib、organization
plan	教学计划表	course	course_lib
account	账户表	system	organization、teacher

➤ 系统表

数据库表名	解释	所属模块	关联外键表
system_token	登录 token 表	system	无
operationlog	操作日志表	system	account
system_accountidentity	身份表	system	account
system_dictionary	动态字典表	system	无
system_systemconfig	系统配置表	system	无

➤ 多对多关系表

数据库表名	解释	所属模块	对应关系
plan_class	教学计划-班级对应表	classes	plan- classes
course_modelmajor	计划模板-专业对应表	course	plan_model- organization
plan_teacher	教学计划-教师对应表	instructor	plan- teacher
plan_materaial	教学计划-教材对应表	teachmaterial	plan- materaial_lib

materail_teacher	教师-教材对应表	teachmaterial	teacher- materiaial_lib
------------------	----------	---------------	-------------------------

3.5.2 数据字典

➤ 组织机构表

字段名	字段名解释	字段类型	其他说明
name	名称	varchar	主键
type	类型	varchar	字典项
state	是否可用	Boolean	
super_id	上一级组织机构	外键	外键表:organization

➤ 班级表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
code	班级编号	Varchar	唯一约束
name	班级名称	Varchar	
number	班级人数	Int	
session	级	Int	字典项
state	班级状态	Varchar	字典项
major_id	专业	外键	外键表:organization

➤ 课程库

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
code	课程编号	Varchar	唯一约束
name	课程名称	Varchar	
describe	课程描述	Varchar	
type	课程类型	Varchar	字典项
remark	备注	Varchar	
academe_id	学院	外键	外键表:organization

➤ 教师表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	

updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
code	教师工号	Varchar	唯一约束
name	教师姓名	Varchar	
professional_rank	职称	Varchar	字典项
education	学历	Varchar	字典项
state	在职状态	Varchar	字典项
remark	备注	Varchar	
faculty_id	所属系部	外键	外键表:organization

➤ 教材库

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
name	教材名称	Varchar	
isbn	ISBN 号	Varchar	唯一约束
price	价格	Decimal	
language	教材语言	Varchar	字典项
author	编者	Varchar	
publisher	出版社	Varchar	
pubdate	出版时间	Varchar	
editions	版次	Varchar	
is21st	是否为 21 世纪教材	Boolean	
recommend	是否为推荐教材	Boolean	
spec	教材规格	Varchar	
remark	备注	Varchar	

➤ 教学计划模板表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
grade	学年	Varchar	
period	总学时	Decimal	
credit	学分	Decimal	
range	仅本系可选	Boolean	
course_id	课程编号	外键	外键表: course_lib
academe_id	学院	外键	外键表:organization
remark	备注	Varchar	

➤ 教学计划表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
term	学期	Varchar	
period	总学时	Decimal	
surperiod	剩余可选学时	Decimal	
number	上课人数	Int	
credit	学分	Decimal	
range	仅本系可选	Boolean	
course_id	课程编号	外键	外键表: course_lib
remark	备注	Varchar	

➤ 账户表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
accname	用户名	Varchar	唯一约束
password	密码	Varchar	
name	姓名	Varchar	
number	上课人数	Int	
email	邮箱	Varchar	
birth	出生年月日	Date	
lastlogin	上一次登录时间	Datetime	
login	本次登录时间	Datetime	
sex	性别	Boolean	
first	是否首次登录	Boolean	
teacher_id	教师信息	外键	外键表: teacher
academe_id	所属学院	外键	外键表: organization

➤ 登录 token 表

字段名	字段名解释	字段类型	其他说明
token	token 字符串	varchar	主键
value	存储值	blob	
expiration	过期时间	Datetime	

➤ 操作日志表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键

createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
ip	登录 IP	Varchar	
user	操作者	外键	外键表：account
url	访问 url	Varchar	
level	操作等级	Varchar	
describe	描述	Varchar	
remark	备注	Varchar	

➤ 身份表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
name	身份名	Varchar	
account_id	对应账户	外键	外键表：account

➤ 动态字典表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
name	字典名	Varchar	
value	字典值	Varchar	
index	序号	int	用作排序

➤ 系统配置表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
name	配置字段名	Varchar	
value	配置字段值	Varchar	

➤ 教学计划-班级关系表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
plan_id	教学计划	外键	外键表：plan
classes_id	班级	外键	外键表：classes
number	上课人数	Number	

➤ 计划模板-专业对应表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	

model_id	计划模板	外键	外键表: plan_model
major_id	专业	外键	外键表: organization

➤ 教学计划-教师对应表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
df	逻辑删除字段	Boolean	
plan_id	教学计划	外键	外键表: plan
teacher_id	教师	外键	外键表: teacher
bilingual	是否双语教学	Boolean	
starttime	计划开课时间	Datetime	
multimedia	是否多媒体教学	Boolean	
thisterm	是否本学期开课	Boolean	
computer	上机人数	Int	
period	上课课时	Decimal	
remark	排课要求	Varchar	

➤ 教学计划-教材对应表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
plan_id	教学计划	外键	外键表: plan
material_id	教材	外键	外键表: materiaal_lib
type	教材类型	Varchar	

➤ 教师-教材对应表

字段名	字段名解释	字段类型	其他说明
id	自增 id	Int	主键
createdate	创建时间	Datetime	
updatedate	最后一次修改时间	Datetime	
teacher_id	教师	外键	外键表: teacher
material_id	教材	外键	外键表: materiaal_lib

3.5.3 E-R 图

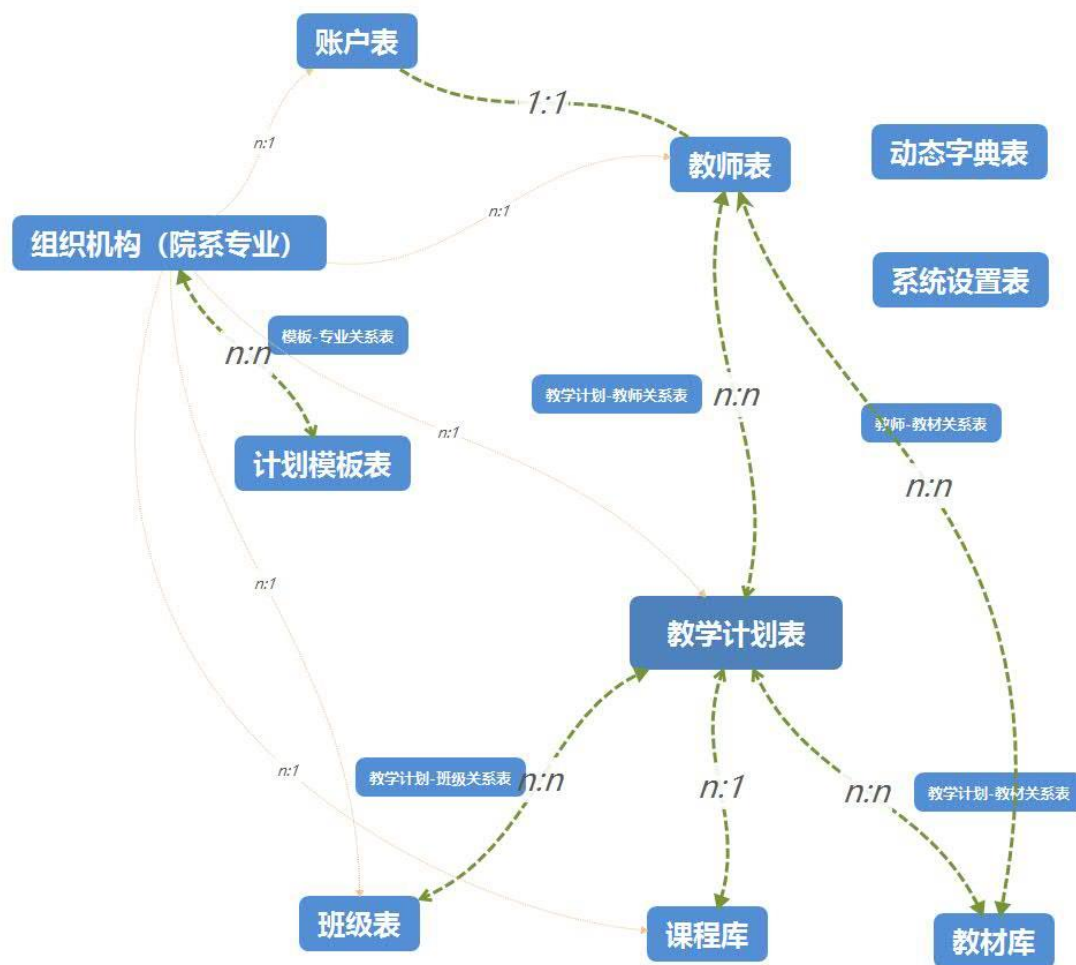


图 3.4 ER 图

4 系统的开发与实现

4.1 开发及运行环境

硬件平台：

- CPU：P41.8GHz；
- 内存：256MB 以上。

软件平台：

- 操作系统：CentOS 7；
- 数据库：Mysql；
- 开发环境：Python3.6；
- 中间件：Nginx+uwsgi；
- 浏览器：chrome；
- 分辨率：最佳效果 1366*768 像素。

4.2 系统逻辑实现

4.2.1 BaseView 基类

面对复杂多样的业务逻辑，本系统采用了模块化的设计思路，设计了 BaseView 作为其他视图的基类，在 BaseView 中，实现了对数据的新增，修改，查询，删除等大部分的操作，并且能自动生成相应的控制器 url。在使用 BaseView 作为基类实现后端主要的逻辑时，有一些参数是必须指定：

BaseView 的子类必需指定给一个相应的模型与之绑定，作为操作的模型，在 Django 框架中模型所对应着数据库表，指定模型后，BaseView 会自定生成该模型对应数据表的增、删、改、查的 API 方法。

BaseView 在实现新增和修改时，可以指定允许新增或者修改的字段，以保证业务逻辑需求以及安全性，这个功能由 create_fields、update_fields 两个变量进行控制。

对于一个完整的模块来说，除了 API 之外，还需要有页面，在 BaseView 中可以利用 templaters_path 变量来指定页面模板所在路径，BaseView 能自动映射相应的模板文件，以及 js 文件。

在 MVC 设计模式中，除了需要处理模型和视图之外，还需指定 url 控制器，在 BaseView 中设计了.router()作为 url 控制器的实现方法，实现了一下几个方法的 url 配置：

- url(r'^view/(.+?)/\$', cls.to_retrieve), 跳转到查询单个实例界面；
- url(r'^view/\$', cls.to_view), 跳转到实例列表界面
- url(r'^createframe/(?.*?)/\$', cls.to_create_frame), 跳转到新增界面
- url(r'^updateframe/(.+?)/\$', cls.to_update_frame), 跳转到修改界面
- url(r'^screen.json/\$', cls.get_screen_data), 筛选信息，用于列表界面数据筛选
- url(r'^field/(.+?)/\$', cls.get_fields), 用于获取单个字段的数据集
- url(r'^api/view/\$', cls.api_view), 列表 api，获取实例列表数据，
- url(r'^api/create/\$', cls.api_create), 创建 api
- url(r'^api/update/(\d+)/', cls.api_update), 修改 api
- url(r'^api/retrieve/(\d+)/', cls.api_retrieve), 查询实例详情 api
- url(r'^api/delete/(\d+)/', cls.api_delete) 删除 api

上述所注册的 url 主要分为两类，一类是页面的 url，一类是数据的 api

对于从数据库中取出的数据，BaseView 会自动处理部分所需要传达给前端或者从前端传递过来的数据，而需要根据业务来确定的一些数据，便预留了许多接口对数据进行操作，如 get_view_context() 用于列表页的数据处理，于此相似的还有 get_create_context(), get_update_context(), get_retrieve_context(), get_screen_context() 等，这些接口在子类中重写之后便能够对数据进行操作与管理。

除了对于页面相关的数据管理之外，对于 web api 接口的数据也有相关相关的方法进行操作，对于新增，修改，查询列表等 api 接口，一般预留了两个方法，对数据进行整理，这两个方法分别在业务操作之前、以及之后对数据进行整理，比如在查询列表的 api 中，预留了一下两个方法：

- view_queryset(), 在处理搜索，筛选，排序之前对数据进行整理

- `view_return_queryset()`, 在返回给前端之前, 对数据进行整理

4.2.2 主页布局

本系统使用单页面模式, 用户登录系统进入主页后, 不再进行页面跳转, 而是通过 `tab` 层或者弹出层浏览内容, 主页分为上侧导航栏, 左侧导航栏, 选项卡区域, 内容主体区域, 下侧区域。

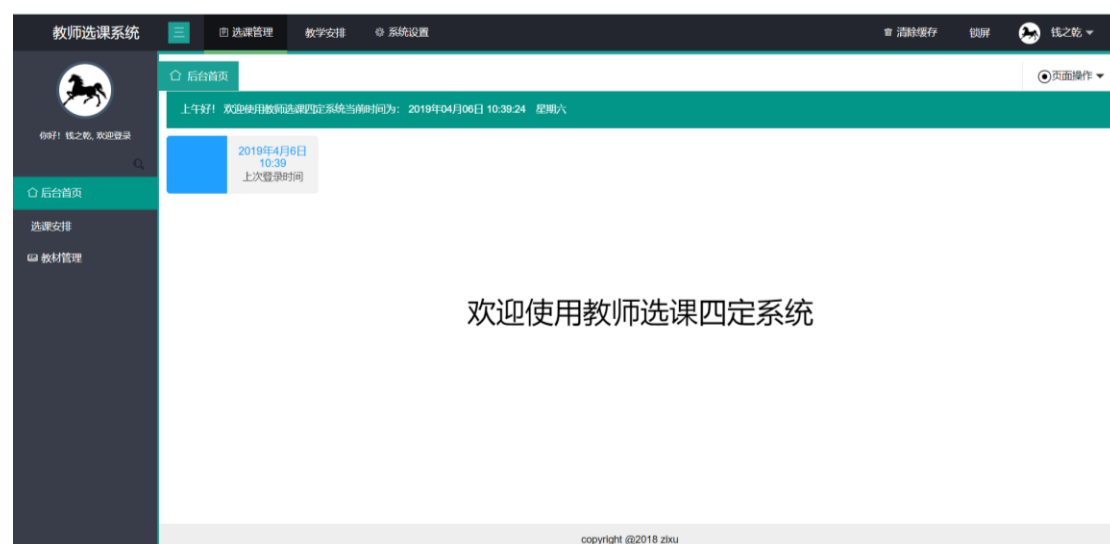


图 4.1 主页

上侧导航栏分为左右两个区域, 左侧区域是系统主体业务导航栏——‘选课管理’、‘教学安排’和‘系统设置’, 分别对应三个身份的操作功能, 教师可以对‘选课管理’进行操作, 教务管理人员能操作‘教学安排’, 系统管理员则可以操作‘系统设置’; 右侧区域主要是用户个人信息管理区域, 以及用户对系统的个人设置。

左侧导航栏是上侧导航栏中主体业务导航栏的二级导航栏, 内容如下图所示。

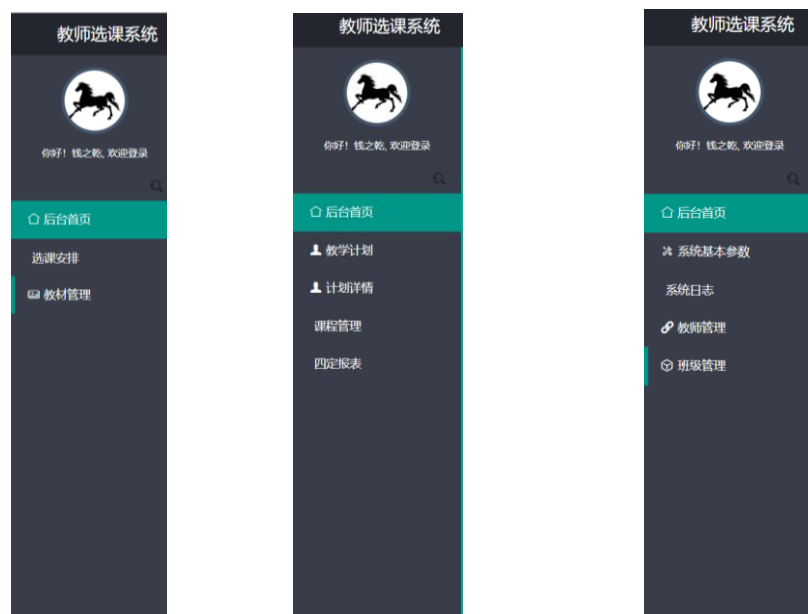


图 4.2 选课管理、教学安排、系统设置对应左侧导航栏

选项卡区域的内容当前用户打开的选项卡，用户打一个页面后会将其保留在在选项卡区域，用户可以点击选项卡随时查看所打开页面的内容，而无需再次打开页面，减少对系统内容的重复访问，提高系统的运行效率。



图 4.3 选项卡区域

4.2.3 前端页面实现

前端页面实现主要分为两类，一类是数据展示页面，一类是表单页面。

➤ 数据展示界面

数据展示页面使用了 layui 框架的数据表格模块实现，如下图所示，分为三个区：工具栏区，数据表格区，分页区。

工具栏区内容主要是各种操作的控件，如：筛选、添加、批量删除、搜索等，以及右边的数据表格模块自带的控件。

数据表格区作为数据的展示区域以及表头，在表头上有排序操作的控件。在数据表格区的最右边有对于单行表格操作的控件，如编辑。

分页区则是分页的控件展示，提供了异步分页、跳页、设置每页显示数据条数等功能。

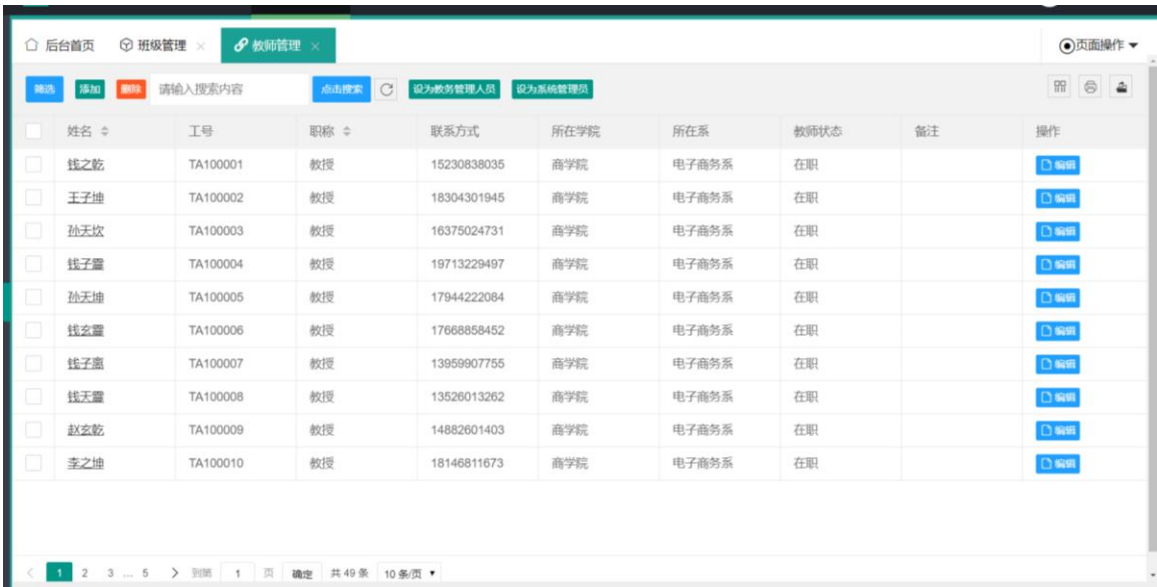


图 4.4 数据展示页面（教师管理）

➤ 表单页面

表单界面主要是对表单的填写，如下图所示，表单页面一般分布着各类表单控件和功能按钮。

上课课时45.0

计划开课时间请输入计划开课时间

双语教学

多媒体教学

本学期开课

是否上机

排课要求请输入备注

新增教材

教材计算机原理(唐玉燕(新华))类型主教材

教材选择或者搜索类型主教材

删除

继续添加

立即提交

重置

图 4.5 表单页面（选课表单）

在系统中，表单的提交不使用 html 的 from 的 action 方法，而使用 ajax 动态请求进行提交，表单提交行为本身不会进行页面的跳转，所以在提交失败后，只会有相应的提示（如下图所示），用户可以直接对表单进行修改，再次提交。提交成功后，才会进行页面跳转。



图 4.6 提交失败后提示内容

对于表单的数据验证，在点击提交之后会对数据进行验证校验，对于不合要求的数据，会给出相应的提示。

班级编号

请输入班级编号

班级名称

请输入班级名称

届

选择或者搜索

班级状态

选择或者搜索

班级人数

请输入单价

学院

选择或者搜索

系

请先选择学院

专业

请先选择系

备注

请输入备注

立即提交

重置

图 4.7 表单验证信息

4.2.4 班级模块实现

班级模块主要是对班级的增、删、改、查操作的实现，只需要继承 BaseView 类即可，在新增班级时，可以选择一个已存在的班级作为新增的模板，并能够自动填充模板的信息，如下图所示：

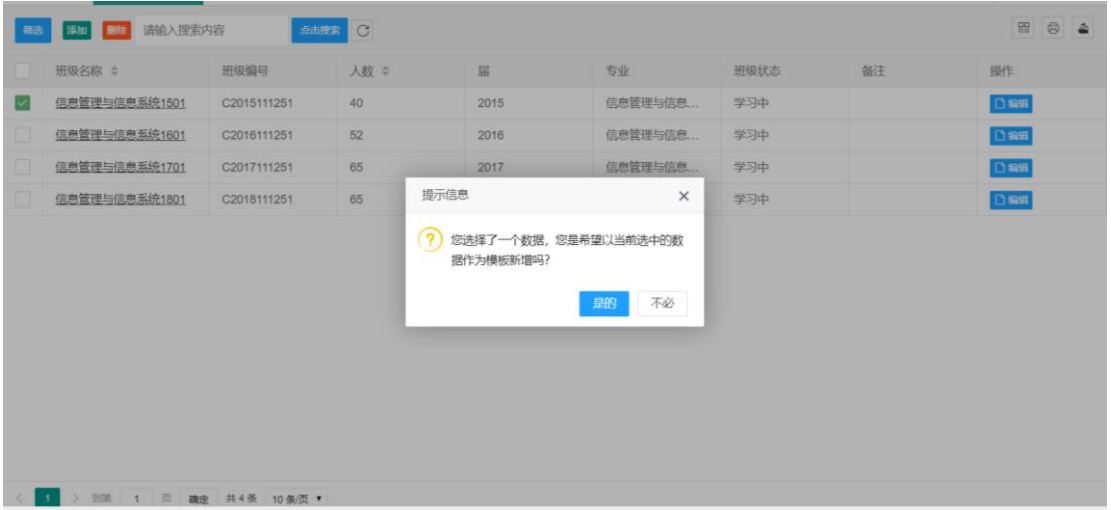


图 4.8 模板新增方式提示

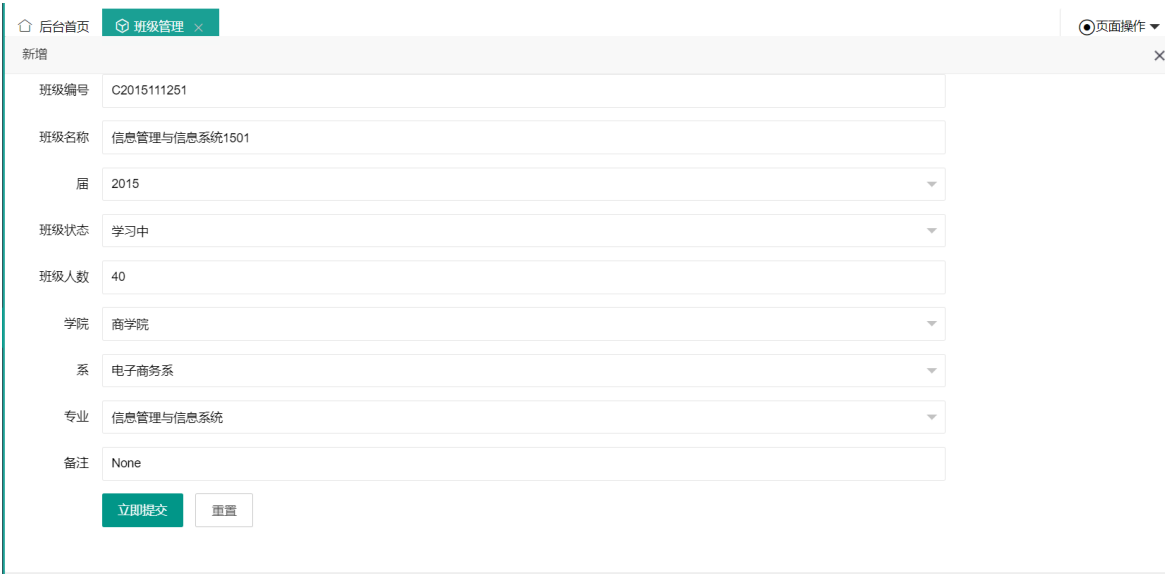


图 4.9 自动填充模板的信息

在添加班级时，需要对学院、系、专业进行三级联动查询，如下图所示，在没有选择学院时，系部选项区会提示，只有选择了学院之后才能选择系，并且可以选择的系只能是该学院的系部。



图 4.10 未选择学院时，系部提示

图 4.11 选学院后，系部选项

班级模块是本系统中功能全面的基础数据模块，实现了对数据表的增、删、改、查基本操作以及筛选、排序、搜索等数据查看功能。在整个系统中的基础数据模块，如教师信息、课程信息均实现了相关的功能。

4.2.5 教师模块实现

教师模块除了对教师信息增、删、改、查功能的实现之外，选课部分的逻辑也由教师模块实现。

教师信息的增删改查的基本操作与班级模块类似。但在新增教师时，除了添加教师信息外，也需要给教师添加一个账号并绑定一个身份，由重写 `perform_create` 方法即可实现：

```
def perform_create(self, data):
    sex = data.pop('sex')
    password = data.pop('password')
    teacher = self.Model.objects.create(**data)
    if sex=='0':
        sex=0
    else:
        sex=1
    user = Account.objects.create (accname=teacher.code,
                                  lastlogin=timezone.now(),academe=teacher.faculty.super,
                                  password=make_password(password),name=teacher.name,sex=sex,first=True,teacher=teacher)
    AccountIdentity.objects.create(name='teacher',account=user)
    return teacher
```

在账号中由于也绑定了学院信息，而教师信息中的院系内容是可以更改的，所以在更改教师信息后，需要重新指定教师对应账号的学院信息。

```
instance.account.academe = instance.faculty.super
instance.account.save()
```

选课安排部分是系统的核心功能，在浏览选课信息的界面仍然是一个数据表格，系统默认展示可选课程，可选在上述查看类型中选择查看数据的范围。



图 4.12 选课内容查看类型



图 4.13 选择全部选课时内容

教师点击选课后，会进入选课信息填写界面，其中上课课时会默认填入课程计划对应的总学时，每当教师提交选课信息后，该计划的可选课时将会减去教师所填入的上课课时：

$$\text{this_plan.surperiod} = \text{Decimal}(\text{teacher_plan.period})$$

若计划的可选课时大于零，则该计划可选再次被选课，以实现一门课程多个教师授课。

在填写选课信息时，若选择上机，将出现上机人数的输入框。

上课课时
48.0

计划开课时间
请输入计划开课时间

☐ 双选教学
☐ 多段体教学
☒ 本学期开课
☐ 是否上机

排课要求
请输入备注

新增教材

教材
选择或者搜索

教材
选择或者搜索

类型

主教材

主教材

教材

立即提交

重置

图 4.14 选课信息填入界面

图 4.15 选择是否上机后，出现上机人数输入框

在选课教材过程中，可以选择多个教材，但只能选择一个主教材，在选择教材过程中，有可以点击新增教材按钮跳转到新增教材界面。

在选课时间段内，教师对于已选的课程可以取消选课，取消选课后，填写的选课信息将会被删除，同时会将改课程对应计划的可选课时，实现代码如下：

```
token = request.META.get('TOKEN')
teacher_id = token.get('user').teacher_id
plan_teacher = PlanTeacher.objects.filter(plan=instance,teacher_id=teacher_id)
if plan_teacher:
    plan_teacher=plan_teacher.first()
instance.surperiod = plan_teacher.period
plan_teacher.delete()
instance.save()
```

4.2.6 教材模块实现

教材模块的功能模块主要分为两部分，教材库及教师教材。教师在教材管理界面可以查看个人教材及教材库所有教材，教师在添加教材至个人教材时，只对该教材建立一个引用，移除时，同样也只移除该引用。

教师在添加教材时，除了将教材添加到教材库之外，也会创建一个引用到个人教材。教师不能删除和修改教材库中的教材，只有管理员有此权限。

教材名称	ISBN号	价格	规格	出版时间	出版社	作者	版次	语言	推荐教材	21世纪...	备注	操作
<input type="checkbox"/> 统计学...	978-745...	123.00	11	2019-03...	湖北大...	李之坤	2	中文	false	false	11	添加至我的
<input type="checkbox"/> 计算机...	978-754...	45.00	182页	2019-03...	新华书店	唐玉高	1	中文	false	false		添加至我的
<input type="checkbox"/> 计算机...	978-784...	45.00	182页	2019-03...	新华书店	唐玉高	3	中文	false	false		添加至我的
<input type="checkbox"/> 计算机...	978-744...	45.00	182页	2019-03...	新华书店	唐玉高	2	中文	false	false		删除
<input type="checkbox"/> 数据结构	978-731...	45.00	278页	2019-04...	清华大...	唐玉丽	12	中文	false	true	111	删除
<input type="checkbox"/> 电子高...	978-844...	111.00	1222	2019-04...	清华大...	唐玉丽	12	中文	true	true		删除
<input type="checkbox"/> 电子高...	978-444...	1112.00	11	2019-04...	北京大...	唐玉丽	1	中文	false	false		添加至我的

图 4.16 教材展示页面

4.2.7 课程模块实现

课程模块是系统的核心模块，除了实现对课程的增、删、改、查基本操作之外，教学计划、教学计划模板以及四定报表的相关操作也由此模块实现。

教务管理人员在使用本系统时，无需对每个班级的没门课程进行一一安排，而根据教学计划的模板生成教学计划，对于每个专业来说，同一个学期所安排的课程大抵相似，教务管理人员只需要关注每学期变动的课程情况，修订教学计划即可。

课程名称	年级	默认课时	上课专业	备注	操作
<input type="checkbox"/> 信息安全	第一学期	45.0	市场营销信息管理与信息...		删除
<input type="checkbox"/> 数据结构	第一学期	45.0	信息管理与信息系统		删除
<input type="checkbox"/> 计算机原理	第一学期	35.0	信息管理与信息系统		删除

图 4.17 教学计划模板

在创建或修改教学计划模板时，可以选择多哥上课专业，以适应多个班级上课的情况，也可以不指定上课的班级，以适应如体育课、公选课等由学生自由选课的情况。

图 4.18 添加教学计划模板

在确定教学计划模板后，点击提交计划安排即可自动生成相应的教学计划，代码逻辑如下：

```

user = request.META.get('TOKEN').get('user')
academe = user.academe.name
term = get_term()
if SystemConfig.objects.first(name='plan',value=academe):
    return JsonResponse({'code':0,'err': '已提交过教学计划'})
model_set = PlanModel.objects.filter(df=0,academe=academe)
for model in model_set:
    plan = TeachPlan.objects.create(term=term,
                                     course=model.course,period=model.period,
                                     surperiod=model.period,number=0)

    for major in model.modelmajor_set.all():
        session = get_session_for_grade(model.grade,term)
        classes = Classes.objects.filter(major=major.major,session=str(session),df=0)
        print(session,classes,major)
        for class_ in classes:
            plan_class = PlanClass.objects.create
            (plan=plan,classes=class_,number=class_.number)
            plan.number += plan_class.number
    plan.save()
return JsonResponse({'code':200,})

```

在每一个学期中，每个院的教务管理人员只允许提交一次，提交之后，按钮将会被禁用。

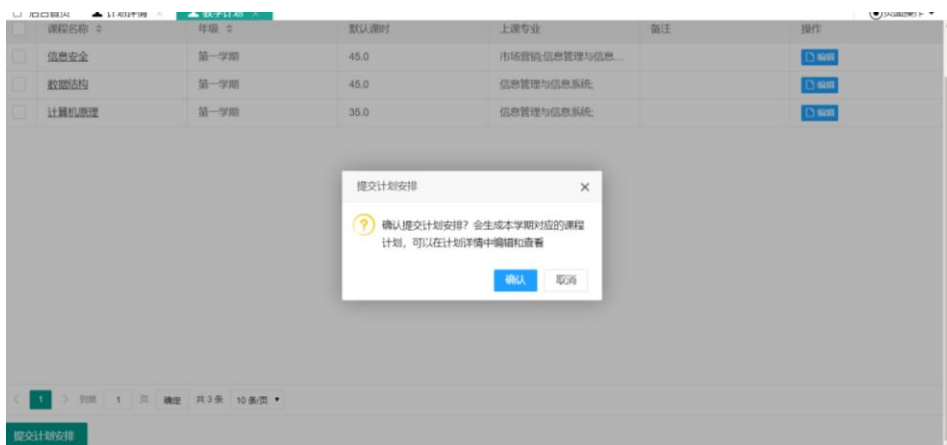


图 4.19 提交教学计划提示



图 4.20 提交计划安排后，按钮处于禁用状态

统计教学计划模板自动生成的教学计划可以在计划详情中分班级查看，选定一个班级后可以查看该班级当前的教学计划以及历史教学计划。



图 4.21 分班级查看教学计划

班级名称: 信息管理与信息系统1701 班级人数: 65 级: 2017 专业: 电子商务系-信息管理与信息系统					
当前学期课程安排!					
<div> <div>添加</div> <div>删除</div> <div>刷新</div> </div> <div> <div>课程名称</div> <div>学期</div> <div>上课课时</div> <div>上课总人数</div> <div>备注</div> <div>操作</div> </div>					
<input type="checkbox"/>	信息安全	2019上学期	45.0	65	<div>编辑</div>

图 4.22 班级当前学期教学安排

<div> <div>1</div> <div>到第</div> <div>1</div> <div>页</div> <div>确定</div> <div>共 1 条</div> <div>10 条/页</div> </div>					
历史课程安排!					
<div> <div>编辑</div> <div>请输入搜索内容</div> <div>高级搜索</div> <div>刷新</div> </div> <div> <div>课程名称</div> <div>学期</div> <div>上课课时</div> <div>上课总人数</div> <div>备注</div> </div>					
<input type="checkbox"/>	信息安全	2018上学期	45.0	65	

图 4.23 班级历史教学计划安排

对于当前学期的教学计划，教务管理人员可以进行添加、修改、和删除。在新增一个教学计划时，系统会默认添加当前班级作为课程的上课班级，并且无法删除或更改，。

新增		X	
课程	选择或者搜索		
默认课时	请输入默认课时		
备注	请输入备注		
上课班级			
班级名称	2017-信息管理与信息系统170	上课人数	65
对应内容	选择或者搜索	上课人数	请输入默认课时
<div> <div>继续添加</div> <div>立即提交</div> <div>重置</div> </div>			

图 4.24 教学计划添加界面

教师选课后，系统会根据教师选课的内容打印四定报表，以供教学安排，四定报表是系统的重要输出功能。

系统选课过程中，对于一门课程而言，其上课班级、教学教师、使用教材都可能为多个，输入的选课信息也会有多个，这样是不符合四定报表的格式，所以在打印四定报表时，上课年级、专业和班级中，若有多个信息将会以‘、’进行隔开，单选课信息如开课时间、是否双语教学等会以第一个选课教师填写信息为准，在‘教师姓名’一栏中会将所有选课教师的姓名也以‘、’隔开，而教材相关的信息只会显示主教材相关的信息，其他教材会在‘其他教材读物’一栏中显示。

下载报表

四定

报表

<input type="checkbox"/>	上课年级	上课专业	上课班级	课程名称	总学时	学分	课程属性	是双语教学	计划开课时间	是多媒体教学	是本
<input type="checkbox"/>	2018	信息管理与...	信息管理与...	计算机原理	48.0	3.0	专业课				
<input type="checkbox"/>	2017	信息管理与...	信息管理与...	信息安全	45.0	3.0	专业课				
<input type="checkbox"/>	2016	信息管理与...	信息管理与...	数据结构	45.0	3.0	专业课	true	2019-04-12	false	true
<input type="checkbox"/>	2015	信息管理与...	信息管理与...	计算机原理	35.0	3.0	专业课	false	2019-04-12	true	true

图 4.25 四定报表界面

系统还提供了 Excel 下载报表的功能，点击‘下载报表’按钮可以下载 Excel 四定报表，文件格式如下：

四定工作安排表														
序号	开课系部	上课年份	上课专业	上课班级	课程名称	总学时	周学时	学分	课程属性	是双语教学	计划开课时间	是多媒体教学	是本	备注
1	电子商务	2018	信息管理与...	信息管理与...	计算机原理	48.0	3.0	3.0	专业课					
2	电子商务	2017	信息管理与...	信息管理与...	信息安全	45.0	3.0	3.0	专业课					
3	电子商务	2016	信息管理与...	信息管理与...	数据结构	45.0	3.0	3.0	专业课	True	2019-04-12	False	True	
4	电子商务	2015	信息管理与...	信息管理与...	计算机原理	35.0	3.0	3.0	专业课	False	2019-04-12	True	True	

图 4.26 Excel 报表

4.2.8 系统模块实现

系统模块主要实现了登录及身份验证的相关逻辑，其他诸如系统参数，组织机构等实现过程也在系统模块中，由于功能较为简单，在此不再细说。

系统登录采用了 token 令牌的验证方式，用户在登录系统过程中，验证用户名密码后将会生成一个随机 token ID，token ID 本身不含有任何数据信息，但会作为一个唯一的标识将用户的账号信息和身份列表保存在数据库中，并且设置过期时间，写入 cookie 中，当用户再次访问系统时，会从 cookie 中获取并验证 token ID，并将用户的账号信息和身份列表写入 request.META 中，以供后台逻辑调用。

下面是登录相关实现代码：

```

username = request.POST.get('username')
password = request.POST.get('password')
user = Account.objects.filter(df=0,accname=username)
if user:
    user = user[0]
    print(password)
    if check_password(password,user.password):
        user.lastlogin = datetime.datetime.now()
        identity = user.accountidentity_set.all().values_list('name',flat=True)
        token_id = create_token({'user':user,'identity':identity})
        token = get_token(token_id)
        print(token)
        request.META['TOKEN_ID'] = token_id

```

```
request.META['TOKEN'] = token
return JsonResponse({'code':200})
else:
    return JsonResponse({'code': 0,'err':'密码不正确'})
else:
    return JsonResponse({'code':0,'err':'账户不存在或被禁用'})
```



图 4.27 登录界面

5 系统的扩展及维护

5.1 常见的问题及解决方案

系统在开发过程中虽然进行了较为完善的测试，但也无法保证经善尽美，在系统的运行过程中，可能会出现以下问题：

在一段时间不访问系统后，打开任意选项卡时出现登录界面，如下图所示：

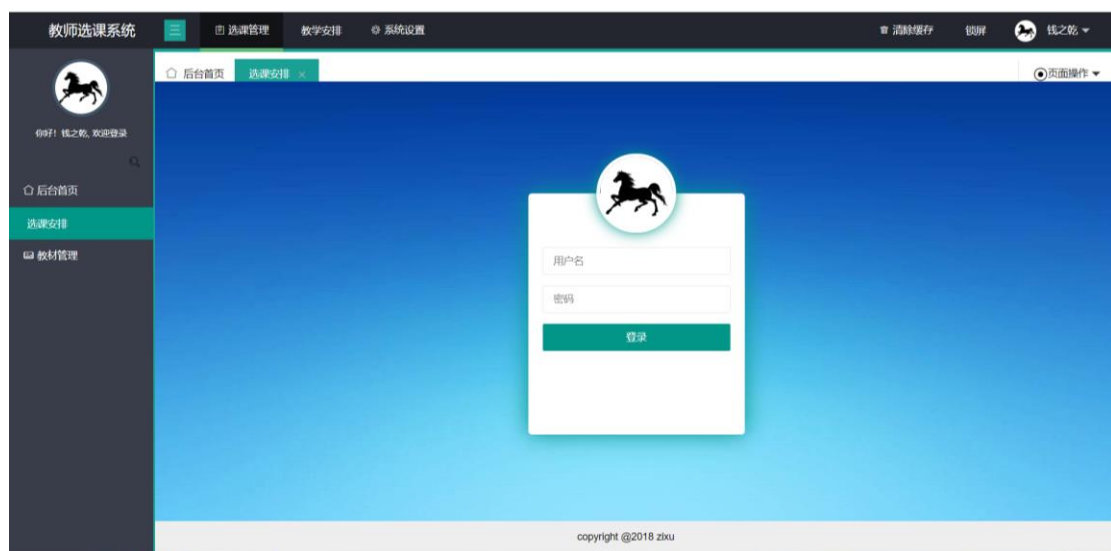


图 5.1 选项卡出现登录界面

这是因为用户登录系统时间过期所导致的，系统通过保存用户登录信息时，只保留 token 有

效时间为 20 分钟，在每次访问系统时都会刷新 token 的有效时间，如果没有操作系统超过 20 分钟，则 token 失效后需重新登录，再使用浏览器刷新页面。

为了适应在移动端使用本系统，进行适应性布局时，导航栏与在 PC 端不一致。移动端的上侧导航栏进行了折叠，而右侧导航栏需要点击下方的控件唤出。本系统在移动端进行适配时，尽量按照移动端的使用习惯，所以许多方面与 PC 有所不一致。



图 5.2 移动端界面

5.2 系统的功能扩展及预留接口

本系统在设计时切合教师选课过程中的各个业务逻辑，而对于其他方面如教师信息管理、班级管理，则有所保留，只做了基本的信息维护，其次对于教材管理方面，在选课之后教务管理人员往往要根据选课的内容统一订购教材，系统可以将选课教材进行统计和管理，生成订购清单，这部分是可以作为系统主要的功能扩展部分。

系统在设计选项字段时，对所有字典都做使用了动态字典，将字典选项存入数据库中，一变随时更新。本系统当前只实现了对部分重要字典的动态更新，但对于所有动态字典选项都做了相关的逻辑，以便于功能扩展。

在进行系统设计时，使用了模块化的设计思路，对模块进行设计时预留了一些接口，以供以后对系统进行扩展。在 BaseView 类中，有以下方法作为系统的预留接口：

- get_fields(cls, request, field)：用于获取单个字段的数据集。
- register_router(self, regular, func)：在 BaseView 中注册其他 url。
- api_retrieve(self, request, id)：查询一条记录的详细信息 API
- to_retrieve(cls, request, id):跳转到查询详情界面视图

6 结论与展望

随着互联网规模的扩大以及信息化程度的提高，各行各业对于数据标准化和透明化越来越重视，信息化系统以及线上与线下数据的交互成为提高工作效率的重要手段。本系统给予 B/S 模式，实现了较为完整的业务逻辑和业务流程。经过了细心的调试和排错解决了绝大部分的问题。

在编写系统的过程中，我也遇到了很多的问题。通过老师的指点和查阅资料等得到了解决，并圆满完成了整个系统的开发工作，同时积累了许多解决经验。这次的毕业设计达到了

预期的目的,实现了教师选课系统的整个流程。通过这次毕业设计,使我从理论到实践迈出了坚实的一步。在学习理论、分析和组织程序结构以及具体的实现等整个过程中,我体会到了开发过程的酸、甜、苦、辣。要编写出一个逻辑业务完整的程序,必须要有缜密的思维,谨慎的作风和坚忍不拔的毅力。

参考文献

- 1.Henriksson , K . , K . Nordlund and J . Wallenius , Simulating model steels : An analytical .bond. order potential for Fe—C. 2008: 中国北京. P.138
- 2.[美]Gary B.Shelly, Thomas J.Cashman, Harry J.Rosenblatt. Systems Analysis and Design(Fifth Edition)[M], 机械工业出版社, 2004
- 3.M.Tamer Ozsü&Patrick Valduriez. Principles of Distributed Database Systems(Second Edition)[M],Prentice-Hall,Inc., 2002.
- 4.黄明,梁旭编著. 信息系统设计与开发实例[MI. 北京: 机械工业出版社 2004. 4, 8—100.
- 徐立波. 音像超市管理系统设计【J】. 考试周刊. 2016(14): 77-80
- 5.董萍萍. 驻外使、领馆人员管理系统设计与体会叨. 市场与电脑. 1999(02)
- 6.陈建中, 郇良勋. 教师挂牌上课制度的意义及问题解决方略【J】. 现代大学教 育, 2001. 3: 96, 98
- 7.张玉玲. 应推广教师挂牌授课制度. 新疆财经大学的教师挂牌授课制度分析【J】. 天津市经理学院学报, 2013, 3: 47. 49
- 8.李淑堃. 基于 GSM MODEM 的高校选课辅助系统设计与开发【J】. 计算机应用 与软件, 2012, 29(4): 203. 205
- 9.刘乃嘉, 彭宇, 王鑫, 付小龙. 清华大学网上选课性能优化研究与实践【J】. 实 验技术与管理, 2011, 28(5): 247—249
- 10.周会强. 基于 Struts 框架的 JavaWeb 应用开发研究【J】. 科技通报, 2012, 28(6): 114—116
11. Nigel George. 《精通 Django》安道译,2017.4
12. 周吉波. 基于 django web 框架的车载管理系统的设计与应用, 浙江工业大学硕士论文, 2012