基于Java的琴行课程预约系统的设计与实现

摘要：随着无纸化办公的普遍实现，信息的自动处理以及网络式的信息交互方式已经被人们广泛应用。让计算机来管理学生的信息是现在各个高校都在积极进行的工作之一，也是高校教学管理工作的重要内容之一。网上选课与传统的选课方式相比更加节约资源，增加了学生选课自主权。当今时代是飞速发展的信息时代。在各行各业中离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用他能够进行信息管理，使用计算机进行信息控制，不仅提高了工作效率，而且大大提高了其安全性。尤其对于复杂的信息管理，计算机能够充分发挥它的优越性。计算机进行信息管理和信息管理系统的开发密切相关，系统的开发是系统管理的前提。本系统就是为了管理好琴行学生选课信息而设计的。

国内的关于琴行的选课系统更多基于的是小程序，但是小程序有个缺点就是只能在微信里面操作，展示信息有限，并且不能在线学习一些免费课程。所以需要做一个易于查看且好学习的web琴行选课平台。

关键词：java； 琴行选课系统； RESTful； web

Design and implementation of a Java-based booking system for Piano Course

**Abstract:** With the general implementation of paperless office, automatic processing of information and network-based information interaction have been widely used.Letting computers manage students'information is one of the active work in various colleges and universities, and is also one of the important contents of teaching management in Colleges and universities.Online course selection is more resource-efficient than traditional course selection and increases students'autonomy in course selection.The present age is a rapidly developing information age.Information processing is indispensable in all walks of life, which is the environment in which computers are widely used in information management systems.The greatest benefit of computer is that it can manage information and control information with computer, which not only improves the efficiency but also greatly improves its security.Especially for complex information management, the computer can give full play to its superiority.Computer information management and information management system development are closely related, system development is the prerequisite of system management.This system is designed to manage the course selection information of piano students.

Domestic piano course selection system is more based on applets, but the drawback of the applets is that they can only operate in WeChat, display limited information, and can not learn some free courses online.So we need to make a web piano course selection platform which is easy to view and easy to learn.

**Keywords:** Java; Piano Course Selection System; RESTful; Web

目 录

[第1章 绪 论 1](#_Toc515023811)

[1.1 研究背景 1](#_Toc515023812)

[1.2 本课题的研究现状及发展趋势 1](#_Toc515023813)

[1.3 研究目的及意义 2](#_Toc515023814)

[1.4 论文的研究内容和结构安排 2](#_Toc515023815)

[1.5 本章小结 3](#_Toc515023816)

[第2章 系统开发相关技术 4](#_Toc515023817)

[2.1 系统开发技术 4](#_Toc515023818)

[2.1.1 开发环境 4](#_Toc515023819)

[2.1.2 开发技术 5](#_Toc515023820)

[2.2 系统结构 11](#_Toc515023821)

[2.3 系统数据库选择 11](#_Toc515023822)

[2.4 本章小结 13](#_Toc515023823)

[第3章 系统需求分析 14](#_Toc515023824)

[3.1 系统需求 14](#_Toc515023825)

[3.1.1 系统结构图 14](#_Toc515023826)

[3.1.2 功能描述 14](#_Toc515023827)

[3.1.3 用户用例 15](#_Toc515023828)

[3.1.4 数据字典 16](#_Toc515023829)

[3.1.5 数据流图 19](#_Toc515023830)

[3.2 系统性能分析 24](#_Toc515023831)

[3.2.1 安全性 24](#_Toc515023832)

[3.2.2 兼容性 25](#_Toc515023833)

[3.2.3 负载 26](#_Toc515023834)

[3.2.4 故障处理 26](#_Toc515023835)

[3.3 本章小结 27](#_Toc515023836)

[第4章 系统概要设计 28](#_Toc515023837)

[4.1 系统功能模块设计 28](#_Toc515023838)

[4.2 系统数据库设计 30](#_Toc515023839)

[4.2.1 数据库设计 30](#_Toc515023840)

[4.2.2 数据库E-R图 30](#_Toc515023841)

[4.2.3 数据库实体图 31](#_Toc515023842)

[4.2.4 数据库表设计 33](#_Toc515023843)

[4.3 本章小结 37](#_Toc515023844)

[第5章 系统详细设计 38](#_Toc515023845)

[5.1 系统功能模块详细设计 38](#_Toc515023846)

[5.1.1 登录页面 38](#_Toc515023847)

[5.1.2 管理页面首页 39](#_Toc515023848)

[5.1.3 学生信息页面 41](#_Toc515023849)

[5.1.4 寝室管理页面 42](#_Toc515023850)

[5.2 本章小结 43](#_Toc515023851)

[第6章 系统实现 44](#_Toc515023852)

[6.1 系统功能模块设计实现 44](#_Toc515023853)

[6.2 数据库实现 44](#_Toc515023854)

[6.2.1 数据表的创建 44](#_Toc515023855)

[6.2.2 数据库连接 45](#_Toc515023856)

[6.3 功能实现与展示 45](#_Toc515023857)

[6.3.1 数据请求 45](#_Toc515023858)

[6.3.2 接口返回数据结构 46](#_Toc515023859)

[6.3.3 菜单列表结构 47](#_Toc515023860)

[6.3.4 系统登录 48](#_Toc515023861)

[6.3.4 数据面板 49](#_Toc515023862)

[6.3.5 学生信息CURD实现 50](#_Toc515023863)

[6.3.2 管理页面首页 55](#_Toc515023864)

[6.3.4 寝室管理页面 55](#_Toc515023865)

[6.4 本章小结 56](#_Toc515023866)

[第7章 系统测试 57](#_Toc515023867)

[7.1 测试方法介绍 57](#_Toc515023868)

[7.1.1 黑盒测试 57](#_Toc515023869)

[7.1.2 白盒测试 57](#_Toc515023870)

[7.2 测试结果及分析 58](#_Toc515023871)

[7.2.1 系统功能测试 58](#_Toc515023872)

[7.2.2 系统兼容性测试 60](#_Toc515023873)

[7.3 本章小结 62](#_Toc515023874)

[总结与展望 63](#_Toc515023875)

[致 谢 64](#_Toc515023876)

[参考文献 65](#_Toc515023877)

第1章 绪 论

1.1 研究背景

当今时代是飞速发展的信息时代。在各行各业中离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用他能够进行信息管理，使用计算机进行信息控制，不仅提高了工作效率，而且大大提高了其安全性。

尤其对于复杂的信息管理，计算机能够充分发挥它的优越性。计算机进行信息管理和信息管理系统的开发密切相关，系统的开发是系统管理的前提。本系统就是为了管理好琴行学生选课信息而设计的。

学员选课作为一种信息资源的集散地，包含很多的信息数据的管理。由于数据繁多，容易丢失，切不易查找。总的来说，缺乏系统，规范的信息管理手段。尽管计算机已经普及，但是并未用于信息管理，没有发挥它应有的效果，资源闲置比较突出，这就是信息管理系统的开发基本环境。

数据处理手工操作，工作量大，出错率高，出错后不易更改。造成了时间上的浪费，基于这个问题，我认为有必要建立一个选课系统，是学生选课信息管理工作规范化，系统化、程序化，避免学生选课管理的随意性，提高信息处理的速度和准确性，能够及时、准确、有效的查询和修改选课情况。

随着生活节奏的加快,人们对网络的依赖程度与日俱增,传统意义上的培训班已经不能满足人们的需求，为了充分利用学员的自主空余时间,不是固定时间授课,而是让学员预约课时,这种方法显得更为入性化。伴随着智能手机的崛起,移动互联网发展迅速,人们也希望尽可能的通过网络达到某些目的。

国内的关于琴行的选课系统更多基于的是小程序，但是小程序有个缺点就是只能在微信里面操作，展示信息有限，并且不能在线学习一些免费课程。所以需要做一个易于查看且好学习的web琴行选课平台。

1.2 本课题的研究现状及发展趋势

随着无纸化办公的普遍实现，信息的自动处理以及网络式的信息交互方式已经被人们广泛应用。让计算机来管理学生的信息是现在各个高校都在积极进行的工作之一，也是高校教学管理工作的重要内容之一。网上选课与传统的选课方式相比更加节约资源，增加了学生选课自主权。

1.3 研究目的及意义

随着科技的迅速发展，各种管理系统已应用到社会的各个领域。各个大小企业无论规模如何，都充分意识到传统的手工管理模式已经逐渐不能适应时代的发展，为了更好的发展，纷纷开发适合自己的管理系统。

通过琴行教学销售管理系统这个平台，可以实现琴行教学销售管理的信息化、网络化、系统化、规范化，使工作人员从繁杂的数据查询和统计中解脱出来，减少工作量。系统的主要功能包括：课程分类管理、学生管理、选课管理，老师管理、系统管理功能。

第2章 系统开发相关技术

本系统是基于Java平台开发的一套基于Java的琴行预约选课系统，采用前后端分离技术，数据库采用Mysql，利用web浏览器来呈现所有界面。

2.1 系统开发技术

2.1.1 开发环境

Window7 专业版

Java：jdk1.8

Tomcat 8.5

Idea2020

Maven3.2

阿里云服务器Centos 7

2.1.2 开发技术

1、MVC设计模式

基于 MVC 的系统可分为三个层次分别开发。模型层（M 层）使用 Hibernate 框架和 DAO 技术；控制器（C 层）使用 Spring-Boot框架，利用AOP控制流程；视图层（V 层）为若干个 html页面。

M 层开发：新建数据库“选课系统”，创建学生信息表、课程安排表、教师数据表、学生选课情况表。向表中添加数据。Mybatis框架的使用代替了传统的 JDBC 技术，JDBC技术虽然能够满足系统的需求，但却时一种面向过程的技术，而整体的分析设计都是面向对象的，这种差异性不利于系统应用程序的统一性，而 Mybatis 将 Java 中对象与对象之间的关系映射至数据库中表与表的关系，这种转换使数据库可以以面向对象的方式被操纵，并且 Mybatis还将与数据库有关的部分封装了起来。实际操作中首先 Mybatis 框架添加至 Web 项目中，通过向导栏为数据库表生成对应的持久化 Java 类，即POJO 类，完成映射文件的配置。这个过程 Mybatis自动生成了所需的 JavaBean，取代了 JDBC 的功能。最后运用 DAO 技术，将 POJO 类封装入接口，彻底分离数据库访问操作和控制模块。DAO 层中除封装有 POJO 类的接口，还封装了不同对象针对不同数据库表处理数据的各种接口。登录接口中有登录验证的方法，学生接口中有查询，修改学生信息的方法，课程接口中有针对数据库表查询，更新，删除，插入的方法，选课接口有查询，增加，删除的方法。实现 Mybatis 的核心接口中的方法，通过 SQL 查询语句完成各接口实现类的编写。至此形成完整的 M 层数据模型。

C 层开发：控制层的开发使用 Sping-Boot 框架。加载 Sping-Boo包到 Web 项目中，修改 application.yml 文件，配置核心控制器。根据不同的功能创建多个 Action 实现类，获取对象并调用 DAO接口中的方法实现具体功能。创建 application.yml 文件配置所有的Action，当触发对应名称的 Action 时，调用 Action 类完成与数据库相关的方法实现。同时配置了业务处理完成后的跳转页面。

V 层开发：视图部分主要是面向用户的程序页面，创建多个 template 文件，设计实现系统所需要的各个界面，主要有入口界面，两种登录界面，两种功能界面，信息修改界面等等。需要注意页面中提交到的 action 名称要与其功能对应。其他无关部分的页面设计使用 HTML 标记，JavaScript 等实现即可。

2、前后端分离

在Web应用开发的过程中，业界对于前后端的分界并没有明确的概念。大部分人认为浏览器是前后端的分界线，以浏览器中为用户进行展示的部分成为前端，而将运行于服务器，为页面提供业务逻辑和数据代码称为后端。以此进行前后端开发工作的分工就是前后端分离[1]。事实上，并非如此，准确的说，前后端分离并不只是开发模式，而是Web应用的一种架构模式。在开发期，前后端工程师可以通过约定好交互接口（API），实现并行开发；在运行期，前后端分离模式需要对Web应用进行分离部署（也可以前端打包完成后，发给后端综合打包），前后端之间使用HTTP请求进行交互，通常会采用Json（或xml）作为数据交互的格式。然而作为Web应用的架构模式，前后端分离并不是通过这样一句话就能一概而谈的，可以从交互形式、代码组织方式、开发模式三个方面对前后端分离进行认识。

交互形式：

在这种前后端分离的应用架构中，前后端约定好接口（RESTful[2]规范）和数据规则，后端只需要负责按照约定的规则为前端提供API服务即可。前后端之间通过HTTP或者WebSocket请求数据交互，视具体项目要求进行选择，通常使用前者。前端获取到数据后，进行页面的装配和渲染。前后端分离的应用架构如图2-1所示。

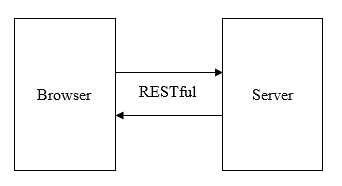


图2-1 前后端分离应用架构图

2.2 系统结构

经过研究，本文所讨论的是构建的B/S系统架构的应用。前面提到，的系统在最终打包上线，会产生web版本和客户端版本是的用户拥有更好的使用体验。本次项目的客户端构建也是一个基于web视图的客户端，只有连接到后台服务器才能使用，客户端的产生是基于web兼容性的一次探索。在后面的版本中，会对客户端加入本地数据库，存取数据提供离线使用的能力，客户端与服务器连接双向同步数据即可。由此，本文重点构建web客户终端。

2.3 系统数据库选择

开发数据库应用，选择一个好的数据库是非常重要的。目前，商品化的数据库管理系统以关系型数据库为主导产品，技术比较成熟。国际国内的主流的主导关系型数据库管理系统有MYSQL，MongoDB、ORACLE。以下是对这三款主流数据库管理系统的简单介绍。

1、MYSQL

MySQL是当今最流行的开源数据库管理系统，超过10亿的下载量足可以证明这点。MySQL以其速度、高可靠性、简单易用，广泛应用,一些大型企业也在逐渐应用，如：Facebook、维基百科等网站。MySQL最早由瑞典的MySQL AB公司开发，后于2008年被Sun收购，2009年Oracle公司收购了Sun公司。它分为社区版和商业版，其体积小、速度快、总体成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择MySQL作为网站数据库。其社区版的性能卓越，搭配Java和Apache可组成良好的开发环境。

2、Mongodb

MongoDB 是一个基于分布式文件存储的数据库。由C++ 语言编写。旨在为WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。

关系型数据库：原子性、一致性、独立性、持久性。

分布式计算：可靠性（容错）、可扩展性、资源共享、灵活性、更快的速度、开放系统、更高的性能。

3、Oracle数据库

 SQL是structured Query Language(结构化查询语言)的缩写。可以使用sql语句建立或删除数据库的对象,插入,修改和更新数据库中的数据，并且可以对数据库执行各种日常管理的操作。它是所有关系数据库管理系统的标准语言.换句话说使用sql可以对所有的关系数据库进行操作。

Oracle优点如下：

（1）Oracle能所有主流平台上运行（包括Windows）完全支持所有工业标准采用完全开放策略使客户选择适合解决方案对开发商全力支持；Oracle的安全机制比Sql server好。

（2）Oracle并行服务器通过使组结点共享同簇工作来扩展windownt能力提供高用性和高伸缩性簇解决方案windowsNT能满足需要用户把数据库移UNIXOracle并行服务器对各种UNIX平台集群机制都有着相当高集成度。

（3）获得最高认证级别的ISO标准认证，安全性更高。

Oracle缺点：价格昂贵。

综合以上分析和本项目的实际相结合，最终决定使用MySQL作为的数据库。本项目中一些数据的属性并不是很固定的，例如系统中的楼层管理员等，各个地方的楼层数可能不同，没办法固定属性。或许你会想到对设计数据库直将其存储为字符串，当然，这样是可行的。经过考察，MySQL5.7能够支持json对象存储。与市面上的众多数据库相比，SQL Server略显笨重，本项目还不足以使用Oracle数据库，MySQL开源且轻量就略显优势，最终选定数据库为系统的主要数据库。

第3章 系统需求分析

3.1 系统需求

分为三种用户：管理员、学生、教师

3.1.1 管理员

1、登录功能

需求描述：通过数据已知的管理员账号登录，登录成功进入管理员界面，方可执行相应功能。

2、添加教师信息

需求描述：添加一个新的教师信息，教师工号作为教师身份唯一标识，添加时会对输入框中教师工号是否存在进行判断并提示。

3、修改教师信息

需求描述：可以选择教师并修改其信息，教师编号不可修改。。

4、删除教师信息

需求描述：删除某一条选中教师的所有信息。

5、查询教师信息

需求描述：选择查询条件，可查询所有教师、通过教师编号查询教师信息，并以表格的形式显示出来。

6、添加课程信息

需求描述：添加一个新的课程信息，课程编号作为该课程身份唯一标识，添加时会对输入框中课程编号是否存在进行判断并提示。其中一些属性通过下拉框方便选择。

7、修改课程信息

需求描述：可对选中的课程进行修改，其中课程编号不可修改。

8、删除课程信息

需求描述：删除选定课程的所有信息。

9、查询课程信息

需求描述：根据相应条件查询课程，所有课程、根据编号、课程类型、所属学院等进行查询。

10、添加学生信息

需求描述：添加一个新的学生信息，学号作为学生身份唯一标识，添加时会对输入框中学号是否存在进行判断并提示。其中课程、性别等信息通过下拉框方便选择。

11、修改学生信息

需求描述：修改选中学生的信息，其中学号不可修改。

12、删除学生信息

需求描述：删除选中学生的所有信息。

13、查询学生信息

需求描述：通过相应条件查询学生，例如查询所有信息，通过手机号、课程、订单等条件查询学生信息。

14、查询订单信息

需求描述：通过查相应的条件查询订单，例如查询所有订单，通过学生手机号、时间段、订单金额等条件查询订单。

15、添加公告信息

需求描述：添加一条新的公告，通过创建时间排序，有标题和内容，输入时会固定排版。

16、修改公告信息

需求描述：修改选中的公告信息。

17、查询公告信息

需求描述：通过时间段来查询公告信息。

18、删除公告信息

需求描述：删除选中公告的所有内容。

3.1.3 学生

1、登录

需求描述：通过手机号和默认密码登录。

2、选课

需求描述：选择一次课程，并进入确认，提示是否确认选课。

3、查询选课安排

需求描述：通过自己报名时选择的课程，查询该课的排课信息。

4、查询选课人数

需求描述：查询课程已选人数。

5、退选或者请假。

需求描述：退选自己已选课程或者请假。

6、会员时间

需求描述：查看自己的订单，查询自己的会员到期时间。

7、管理个人信息

需求描述：只能修改自己的密码，其他信息需联系管理员修改。

8、查询公告信息

需求描述：查看由管理员发布的公告，了解各种信息。

3.1.4 琴行老师

1、登录

需求描述：通过工号(手机号)和默认密码登录。

2、查询个人信息

需求描述：可以查询自己的所有信息。

3、查询个人信息

需求描述：可以查询自己的所有信息。

4、查询描述课程安排

需求描述：通过自己的工号查询自己的课程安排。

5、添加课程安排

需求描述：添加一节自己所教的课程，对课程进行安排包括课程开始时间，选课截止时间，地点，人数，课程时长等信息。

6、修改课程安排

需求描述：修改自己已经安排的课程信息。

7、删除课程安排

需求描述：删除自己已经安排的课程的信息。

8、查询选课信息

需求描述：选定自己教学课程，查看该课程的学生学科情况，包括一些学生信息。

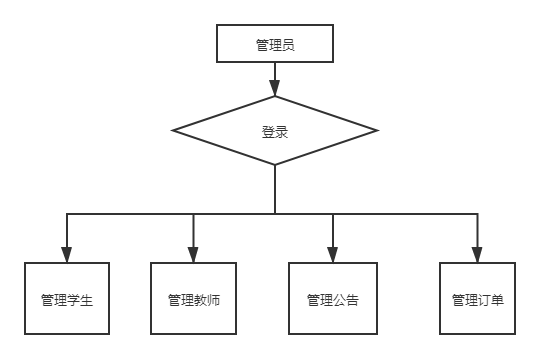
9、打印选课信息

需求描述：选择自己教学的一门课程，并打印出该课程的学生信息。

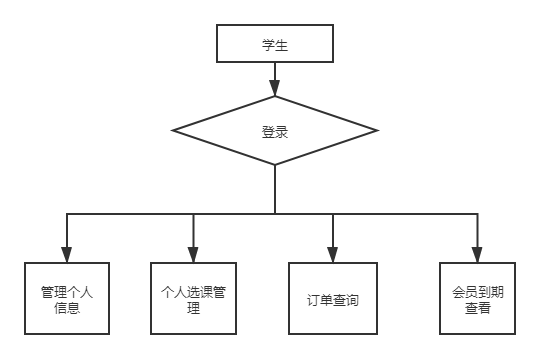
3.2 系统功能用例图

系统分为三个用例图，分别为管理员、学生、教师。

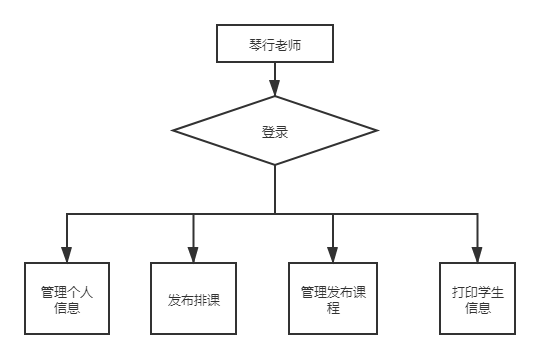
3.2.1 管理员



3.2.2 学生



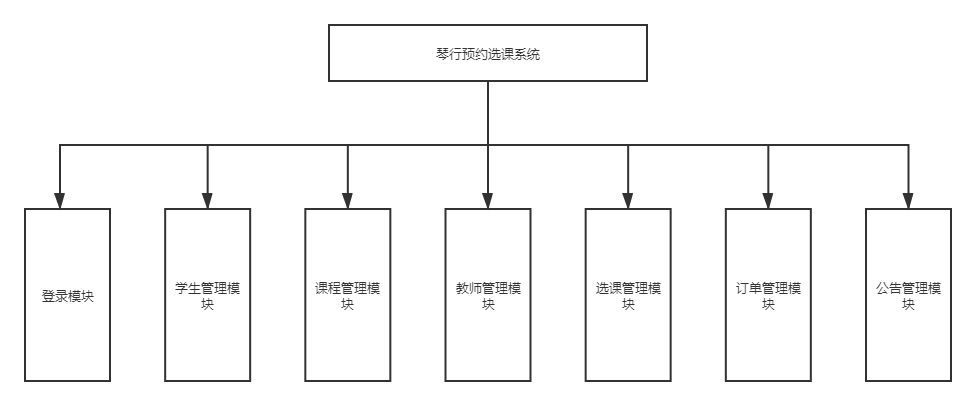
3.2.3 琴行教师



第4章 系统总体设计

4.1 系统功能模块设计

本系统采用模块化的设计方法，将系统按功能分成几个模块，本系统主要包括以下七个部分：



4.1.1 登录模块

用户在使用本系统时，需要登录时，通过web输入账号密码，可以进行登录。

4.1.2 学生管理模块

在这个模块中，管理员可以进行学生信息的录入，包括姓名、性别、年龄、课程名、套餐等信息，以及对学生信息进行修改与删除。

4.1.3 课程管理模块

这个模块中，管理员可以进行课程信息的录入，包括课程名、授课老师、学时等信息，以及对课程信息进行修改与删除。

4.1.4 教师管理模块

在这个模块中，管理员可以进行教师信息的录入，包括姓名、课程等信息，以及对教师信息进行修改与删除。

4.1.5 选课管理模块

在这个模块中，管理员可以进行课程信息的录入，包括课程名、授课老师、学时等信息，以及对课程信息进行修改与删除。

4.1.6 订单管理模块

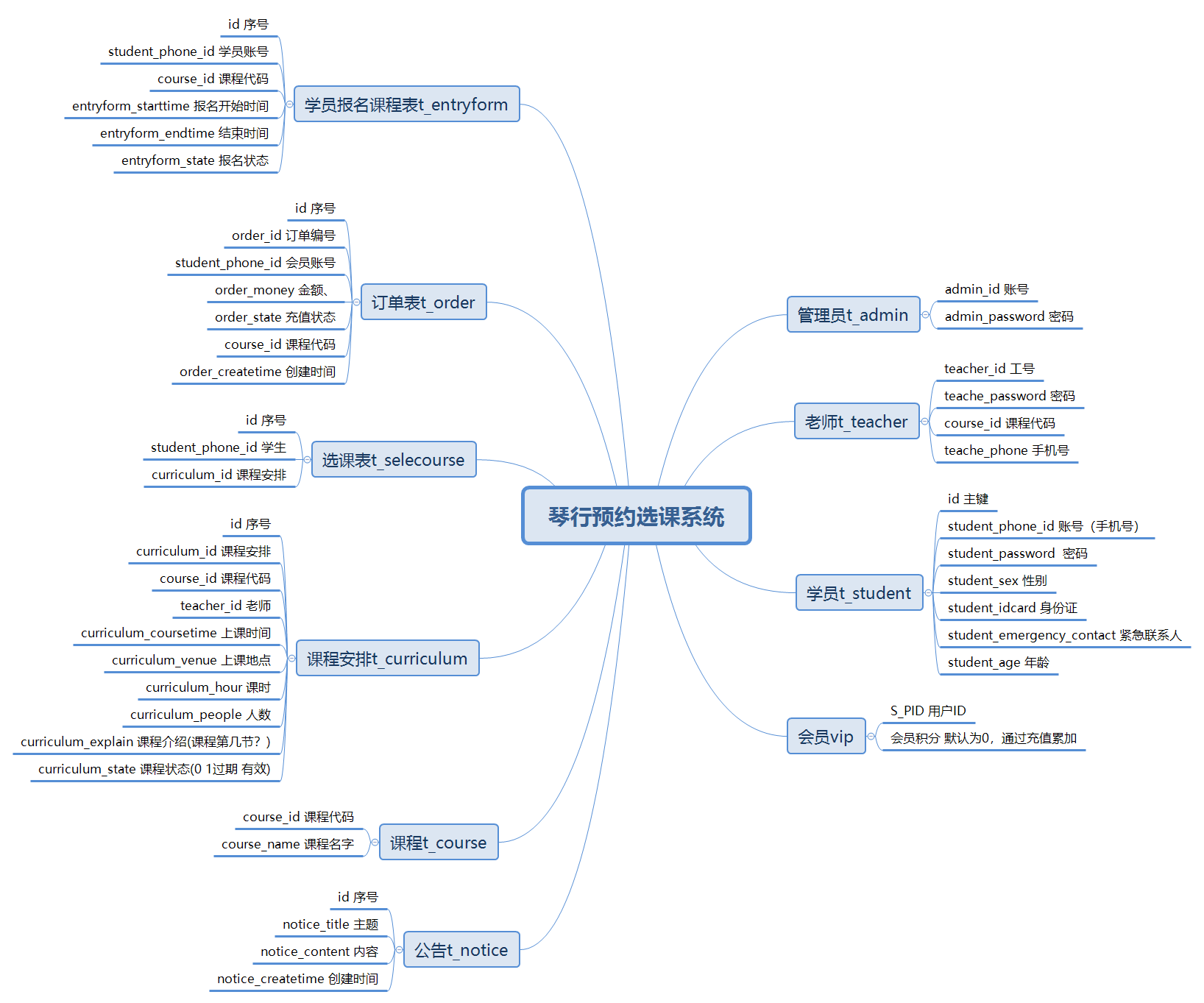
在这个模块中，管理员可以进行订单信息的录入，包括购买人、课程名、金额信息，以及对订单进行修改与删除。

4.1.7 公告管理模块

在这个模块中，管理员可以进行公告信息的录入，包括标题、内容等信息，以及对公告信息进行修改与删除。

4.2 系统数据库设计

由于本系统模块功能之间的独立性，不同模块用到不同的数据库。由理财小助手系统的功能需求入手进行详细分析，基于概念模型的设计，确定本系统所需要的数据库及数据库中的表的设计。



4.2.1 数据库设计

数据库设计(Database Design)是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求。在数据库领域内，常常把使用数据库的各类系统统称为数据库应用系统。

4.2.2 数据库表设计

数据表一般为产品或资料提供一个详细具体的数据资料，方便人们使用和工作时能够清楚方便的获得相应的数据信息。数据表(或称表)是数据库最重要的组成部分之一。

对系统功能的了解和需求分析，设计出学员表、管理员管、琴行导师表、课程信息表、选课表、公告表、选课安排表、订单表、报名表共9张表。4-1所示。

表4-1 学员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| S\_ID | int(10) | 是 | 自增长主键 |
| S\_PID | bigint(20) | 否 | 手机号 |
| S\_PASSWORD | varchar(255) | 否 | 密码 |
| S\_SEX | int(2) | 否 | 性别 |
| S\_IDCARD | bigint(20) | 是 | 身份证 |
| S\_EMERGENCY\_CONTACT | bigint(20) | 是 | 紧急联系人 |
| S\_AGE | int(10) | 否 | 年龄 |
| C\_ID | int(5) | 否 | 课程id |
| S\_NAME | Varchar(25) | 是 | 姓名 |

表4-2 管理员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| A\_ID | bigint(20) | 是 | 管理员账号 |
| A\_PASSWORD | varchar(255) | 否 |  |

续表4-3 课程安排表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| M\_ID | int(10) | 是 | 课程安排id |
| M\_CURRICULUM\_ID | bigint(20) | 否 | 课程安排编号 |
| C\_ID | int(5) | 否 | 课程id |
| T\_ID | bigint(20) | 否 | 授课老师id |
| M\_CURRICULUM\_CTIME | varchar(255) | 否 | 上课时间 |
| M\_CORRICULUM\_VANUE | varchar(255) | 否 | 上课地点 |
| M\_CORRICULUM\_HOUR | varchar(255) | 否 | 课时 |
| M\_CORRICULUM\_PEOPLE | int(10) | 否 | 人数 |
| M\_CORRICULUM\_EXPLAIN | varchar(255) | 否 | 介绍 |
| M\_CORRICULUM\_STATE | int(5) | 否 | 状态 |

表4-4 订单信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| ID | int(10) | 是 | 订单表序号 |
| O\_ID | varchar(255) | 否 | 订单编号 |
| S\_ID | bigint(20) | 否 | 学员id |
| O\_MONEY | double(20,0) | 否 | 订单金额 |
| O\_STATE | int(2) | 否 | 订单状态 |
| C\_ID | int(5) | 否 | 课程id |
| O\_CREATETIME | datetime | 否 | 创建时间 |

表4-5 报名表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| ID | int(10) | 是 | 报名课程表 |
| S\_ID | bigint(20) | 否 | 学员id |
| C\_ID | int(5) | 否 | 课程id |
| E\_STARTTIME | datetime | 否 | 报名时间 |
| E\_ENDTIME | varchar(255) | 否 | 到期时间 |
| E\_STATE | int(5) | 否 | 报名状态 |

表4-6 琴行导师表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| T\_ID | bigint(25) | 是 | 老师id |
| T\_PASSWORD | varchar(255) | 否 | 登录密码 |
| C\_ID | int(8) | 否 | 课程id |
| AGE | int(2) | 否 | 年龄 |

表4-7 公告表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| N\_ID | int(10) | 是 | 主键id |
| N\_TITILE | varchar(255) | 否 | 公告标题 |
| N\_CONTENT | varchar(255) | 否 | 内容 |
| N\_CREATETIME | datetime | 否 | 创建时间 |

表4-8 课程信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| C\_ID | int(5) | 是 | 课程id |
| C\_COURESE\_NAME | varchar(255) | 否 | 课程名 |

表4-9 选课表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型及长度 | 允许空 | 字段说明 |
| ID | int(10) | 是 | 选课表序号 |
| S\_ID | bigint(20) | 否 | 学员id |
| M\_CORRICULUM\_ID | bigint(20) | 否 | 选课id |

第5章 系统详细设计

5.1 系统功能模块详细设计

本小节根据系统的分析，设计出各个系统功能模块并以模块形式体现在具体的用户界面中，然后根据具体的用户界面分析各个功能的细节要求，包括页面中可以操作的元素：按钮，输入框等的操作要求等。下面将分析系统部分功能的设计。

5.1.1 登录页面

系统登录页面是系统的统一入口，提供给用户登入系统的唯一接口，用户在此页面能进行的操作有输入用户信息登录系统。系统登录页面如图5-1所示。

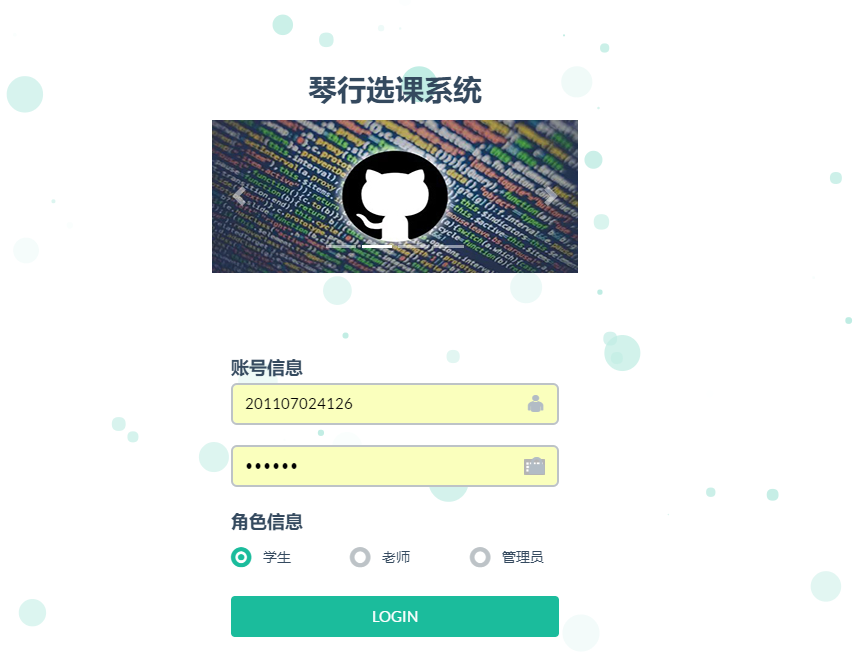


图5-1 系统登录页面

（1）用户登录

用户角色信息：选择自己的角色然后登录。

用户名输入框：数字输入，长度限制为11个字符，同为手机号。

密码输入框：此输入框为密码输入框，密码不能以明文方式回显，长度限制为20个字符。

登录按钮：用户输入信息后，验证成功后正常跳转到界面首页。用户信息验证失败，有相应错误提示，页面停留在当前页面。

5.1.2 管理页面首页

管理员页面首页即登录成功后面的主页面，进入系统就可以看到本系统的主要界面。根据界面能够很直观的看到系统的一些功能，但是还是大致的介绍一下页面功能分区，见图中标识区域。管理页面首页图5-3所示。



图5-3 管理员首页

（1）该页面顶部为管理员页面菜单栏，分别为学生管理、教师管理、课程管理、公告管理、订单管理、个人信息。

（2）顶部右边为管理员状态。

（3）登录首页是一个当前时钟页面，并且会一直刷新。

5.1.3 学生信息页面

学生信息页面是对琴行的学生信息进行管理的模块，管理员能够在此页面对学生信息进行查询，修改和添加。学生信息页面图5-5所示。



图5-5 学生信息页面

（1）查询信息输入框，用户能输入姓名、学号、寝室号，输入框支持回车提交事件。

（2）添加学生，点击该图标添加学生，并刷新列表。

（3）修改、删除：可以操作学员信息。

（4）首页按钮，点击按钮跳转学生信息第一页。

（5）新增按钮，点击按钮弹窗提示填写学生信息，信息填写无误，并插入数据库，并在页面进行相应提示，若失败，给出相应提示。

（6）点击导入按钮，弹窗提示选择要导入的用户信息表格。导入成功，有相应提示，导入失败，提示相应错误。

5.1.4 琴行学生管理页面

琴行学生管理页面（如图5-7所示）是学生功能管理的页面，此功能模块中，学生能够选课、查看选课等的功能。



图5-7 学生登录首页

（1）登录首先进入首页。

（2）页面顶部为学生功能模块菜单栏。

5.1.5 选课页面

琴行学生管理页面（如图5-7所示）是选课结果的页面，此功能模块中，学生能够查看选课结果。



5.1.6 琴行老师管理页面

琴行老师管理页面（如图5-7所示）是老师功能管理的页面，此功能模块中，老师能够排课、管理自己安排课程、查看选课结果，管理选课学生名单等的功能。



图5-7 学生登录首页

（1）登录首先进入首页。

（2）页面顶部为学生功能模块菜单栏。

5.1.6 教学课程管理页面

琴行老师管理教学课程页面（如图5-7所示）是老师管理已经发布的课程的页面，此功能模块中，老师能够修改自己安排课程、删除排课。



图5-7 学生登录首页

（1）修改：修改选课信息。

（2）删除：删除该排课信息。

第6章 系统实现

6.1 系统功能模块设计实现

本系统采用前后端分离技术编写，在本章节中介绍相应模块的使用功能中，会进行前后端实现功能的思维，必要时提供相应的代码。

6.2 数据库实现

6.2.1 数据表的创建

数据库表的创建是使用SQL语句创建数据库表，根据之前设计的数据库表字段名称，长度，说明等创建SQL语句。在语句中还应设置表格的主键、字段是否允许为空等属性，在本项目中数据库字符集统一采用utf-8编码格式。

1、学员信息表是用户存储学员信息的数据库表，表名称为cmb\_student\_mst 。辅导员信息表创建语句如下：

1. CREATE TABLE `cmb\_student\_mst` (
2. `S\_ID` int(10) NOT NULL AUTO\_INCREMENT,
3. `S\_PID` bigint(20) DEFAULT NULL,
4. `S\_PASSWORD` varchar(255) DEFAULT NULL,
5. `S\_SEX` int(2) DEFAULT NULL,
6. `S\_IDCARD` bigint(20) NOT NULL,
7. `S\_EMERGENCY\_CONTACT` bigint(20) NOT NULL COMMENT '紧急联系人',
8. `S\_AGE` int(10) DEFAULT NULL,
9. `C\_ID` int(5) DEFAULT NULL,
10. PRIMARY KEY (`S\_ID`)
11. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

2、管理员信息表是用于存储管理员信息的数据表，表名称为cmb\_administrators\_mst 。

1. CREATE TABLE `cmb\_administrators\_mst` (
2. `A\_ID` bigint(20) NOT NULL COMMENT '管理员账号',
3. `A\_PASSWORD` varchar(255) DEFAULT NULL,
4. PRIMARY KEY (`A\_ID`)
5. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

3、教师信息表是用于存储教师的数据库表，表名称为cmb\_teacher\_mst。

1. CREATE TABLE `cmb\_teacher\_mst` (
2. `T\_ID` bigint(25) NOT NULL,
3. `T\_PASSWORD` varchar(255) DEFAULT NULL,
4. `C\_ID` int(8) DEFAULT NULL,
5. `AGE` int(2) DEFAULT NULL,
6. PRIMARY KEY (`T\_ID`)
7. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

4、课程安排表是用来存储课程安排记录的数据库表，表名称为cmb\_corriculum\_mst 。

1. CREATE TABLE `cmb\_corriculum\_mst` (
2. `M\_ID` int(10) NOT NULL AUTO\_INCREMENT COMMENT '课程安排id',
3. `M\_CURRICULUM\_ID` bigint(20) DEFAULT NULL COMMENT '课程安排编号',
4. `C\_ID` int(5) DEFAULT NULL,
5. `T\_ID` bigint(20) DEFAULT NULL,
6. `M\_CURRICULUM\_CTIME` varchar(255) DEFAULT NULL,
7. `M\_CORRICULUM\_VANUE` varchar(255) DEFAULT NULL,
8. `M\_CORRICULUM\_HOUR` varchar(255) DEFAULT NULL,
9. `M\_CORRICULUM\_PEOPLE` int(10) DEFAULT NULL,
10. `M\_CORRICULUM\_EXPLAIN` varchar(255) DEFAULT NULL,
11. `M\_CORRICULUM\_STATE` int(5) DEFAULT NULL,
12. PRIMARY KEY (`M\_ID`)
13. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

5、课程信息表是用来存储课程信息记录的数据库表，表名称为cmb\_course\_mst。

1. CREATE TABLE `cmb\_course\_mst` (
2. `C\_ID` int(5) NOT NULL,
3. `C\_COURESE\_NAME` varchar(255) DEFAULT NULL,
4. PRIMARY KEY (`C\_ID`)
5. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

6、学员报名表是用来存储报名信息记录的数据库表，表名称为cmb\_entryform\_mst。

1. CREATE TABLE `cmb\_entryform\_mst` (
2. `ID` int(10) NOT NULL AUTO\_INCREMENT COMMENT '报名课程表',
3. `S\_ID` bigint(20) DEFAULT NULL,
4. `C\_ID` int(5) DEFAULT NULL,
5. `E\_STARTTIME` datetime DEFAULT NULL,
6. `E\_ENDTIME` varchar(255) DEFAULT NULL,
7. `E\_STATE` int(5) DEFAULT NULL,
8. PRIMARY KEY (`ID`)
9. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

7、公告表是用来存储公告信息的数据库表，表名称为cmb\_notice\_mst。

1. CREATE TABLE `cmb\_notice\_mst` (
2. `N\_ID` int(10) NOT NULL AUTO\_INCREMENT,
3. `N\_TITILE` varchar(255) DEFAULT NULL,
4. `N\_CONTENT` varchar(255) DEFAULT NULL,
5. `N\_CREATETIME` datetime DEFAULT NULL,
6. PRIMARY KEY (`N\_ID`)
7. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

8、订单表是用来存储订单记录的数据库表，表名称为cmb\_order\_mst。

1. CREATE TABLE `cmb\_order\_mst` (
2. `ID` int(10) NOT NULL AUTO\_INCREMENT COMMENT '订单表序号',
3. `O\_ID` varchar(255) DEFAULT NULL,
4. `S\_ID` bigint(20) DEFAULT NULL,
5. `O\_MONEY` double(20,0) DEFAULT NULL,
6. `O\_STATE` int(2) DEFAULT NULL,
7. `C\_ID` int(5) DEFAULT NULL,
8. `O\_CREATETIME` datetime DEFAULT NULL,
9. PRIMARY KEY (`ID`)
10. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

9、选课表是用来存储选课记录的数据库表，表名称为cmb\_selecourse\_mst。

1. CREATE TABLE `cmb\_selecourse\_mst` (
2. `ID` int(10) NOT NULL DEFAULT '202010001' COMMENT '选课表序号',
3. `S\_ID` bigint(20) DEFAULT NULL,
4. `M\_CORRICULUM\_ID` bigint(20) DEFAULT NULL,
5. PRIMARY KEY (`ID`)
6. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

6.2.2 数据库连接

在SpringBoot框架中，配置文件存放在src/main/resources/application.yml中，数据库连接配置文件如下：

1. datasource:
2. name: test
3. url: jdbc:mysql://127.0.0.1:3306/bs\_bcssoph?characterEncoding=utf8
4. username: root
5. password: 201314
6. # 使用druid数据源
7. type: com.alibaba.druid.pool.DruidDataSource
8. driver-class-name: com.mysql.jdbc.Driver

以上代码中，url为数据库主机地址，username为数据库用户名，password为数据库密码。

6.3 功能实现与展示

6.3.1 数据请求

数据请求主要是前端页面向后端接口取数据的过程，在前后端分离开发的项目中必须要有数据请求的工具，本系统前台页面使用JavaScript构建的页面服务代码，数据请求使用ajax技术。

关键代码：

1. $.ajax({
2. contentType: "application/json;charset=utf-8",
3. type: "POST",
4. url: 'http://47.102.156.110:8080/SPP/user/userCheck',
5. data: JSON.stringify({
6. "userid": userid,
7. "userpsd": password,
8. }),
9. dataType: "JSON",
10. success: function (data) {
11. console.log(data);
12. if(data.message=="success"){
13. alert("登录成功!");
14. window.location.href = "./admin/admindefault.html";
15. }else{
16. alert("用户名或者密码错误!");
17. }
18. }
19. })

6.3.2 接口返回数据结构

前后端分离开的项目，需要前后端协商好接口返回数据的格式，即json的格式。约定好格式并且所有接口返回数据遵循格式返回数据，前端根据预先约定的格式解析json数据。本项目约定的返回数据结构如下：

1. {
2. "code": 200,
3. "msg": "请求成功",
4. "data": {},
5. "pageCount": null
6. }

上述代码是后台返回的json示例，其中包含四个元素。code是后台返回的数据处理状态码，200代表成功，0代表数据异常。msg是后台状态码附带的提示信息，data是请求的数据结构集，通常请求的后台数据都包装在里面。pageCount是查询结果页数，主要用于分页使用。

6.3.3 菜单列表结构

菜单列表结构是以json格式配置的菜单列表结构，菜单列表以json数据格式返回给前端，前端进行解析，使得菜单能够进行动态管理和配置。菜单列表示例格式如下代码片段：

1. {
2. "pageCount": 1,
3. "data": [
4. {
5. "id": 1234,
6. "password": "1234",
7. "headPortrait": "http://b-ssl.duitang.com/uploads/item/201811/04/20181104074412\_wcelx.jpg",
8. "nickname": "张雨",
9. "age": "5",
10. "sex": "男",
11. "permissions": "1",
12. "time": 1565530821000,
13. "score": 50
14. }
15. ],
16. "code": 200,
17. "message": null
18. }

上述代码为一个学生详细信息的请求结果格式。

6.3.4 系统登录

系统登录页面是提供给用户登录系统的界面，是用户使用系统前验证身份的重要一个步骤。用户登录页面用户可以进行输入用户信息登录操作。系统登录页面如图6-2所示。

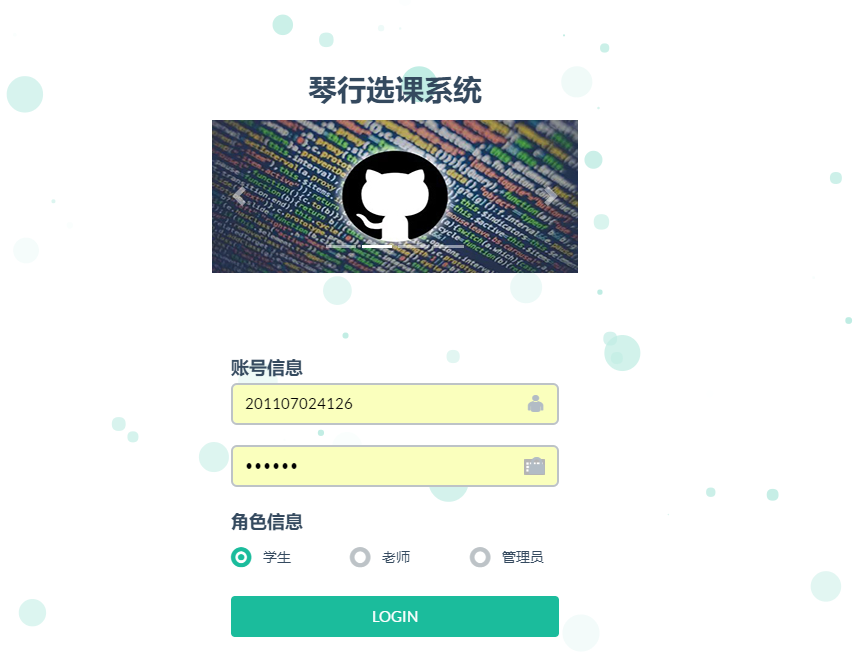


图6-2 登录页面

关键代码：

1. @RequestMapping(value = "/login",method = RequestMethod.POST)
2. public Object login(HttpServletRequest request,@RequestBody Admin admin){
3. Result res = new Result();
4. HttpSession session = request.getSession();
5. List<Object> list = new ArrayList<>(adminService.login(admin));
6. System.out.println(admin.toString());
7. if (list.size() == 0){
8. res.setCode(0);
9. res.setMessage("登录失败！");
10. }else {
11. res.setCode(200);
12. res.setMessage("登录成功！");
13. res.setData(adminService.login(admin));
14. JSONObject js = new JSONObject();
15. js=listkey(list);
16. session.setAttribute("id",js.getLong("id"));
17. session.setAttribute("name",js.getString("name"));
18. session.setAttribute("permission",js.getString("permission"));
19. }
20. return res;
21. }

以上代码是更新对象信息的方法，更改信息之前，先根据id查询出对应的信息，与传入的信息进行匹配，并经行相应提示。

6.3.4 数据面板

数据面板是管理员登录系统管理学生的页面，主要用于展示学生相关数据，例如：姓名、手机、紧急联系人。根据这个面板，琴行管理员就能直观的琴行学生情况。

接口名称：/user/getAll?pms={参数}

关键代码：

1. @RequestMapping(value = "/getAll",method = RequestMethod.GET)
2. Public Object getAll(@RequestParam("pms")String pms,@RequestParam("pageNum")int pageNum){
3. Result res =new Result();
4. if(userService.getAll(pms).size()!=0){
5. res.setCode(200);
6. double num=userService.getAll(pms).size();
7. double count = num/10;
8. int pageCount = ((int)(Math.ceil(count)));
9. res.setPageCount(pageCount);
10. PageHelper.startPage(pageNum,10);
11. res.setData(userService.getAll(pms));
12. }else{
13. res.setCode(200);
14. res.setMessage("暂时还没有数据哟");
15. }
16. return res;
17. }

上述代码根据前端传过来的参数去数据查询符合条件的学员。把学员数据装如list集合，编码成json格式返回给前端。如图所示：



6.3.5 CURD实现

系统中处理信息大部分都是基于数据库表的CURD即，增删改查（Create, Update, Retrieve, Delete）实现。本小节使用学生信息CURD实现介绍相关的操作，其他模块与之类似，将不再介绍。

1、后台服务代码

（1）Service服务层结构

1. @Service("TeacherService")
2. public class TeacherServiceImpl implements TeacherService {
3. @Autowired
4. private TeacherMapper teacherMapper;
5. public List<Object> getAll(){
6. return teacherMapper.getAll();
7. }
8. public int inTea(Teacher teacher){
9. return teacherMapper.inTea(teacher);
10. }
11. public int updateId(Teacher teacher){
12. return teacherMapper.updateById(teacher);
13. }
14. public int deleteById(Teacher id){
15. return teacherMapper.deleteById(id);
16. }
17. }
18. }

（2）mapper接口定义

1. public interface TeacherMapper {
2. List<Object> getAll(); //查
3. int inTea(Teacher teacher); //增
4. int updateById(Teacher teacher); //改
5. int deleteById(Teacher teacherid);//删
7. }

2、mapping的实现：

（1）新增信息。

inTea方法接收到对象信息后，先查询信息表中的id再生成自增id，最后插入数据库。新增学生失败，抛出异常信息“新增对象失败！”

关键代码：

1. <insert id="inTea" parameterType="com.webmaster.model.Teacher" >
2. insert into t\_teacher
3. <trim prefix="(" suffix=")" suffixOverrides="," >
4. <if test="teacherid != null" >
5. teacherID,
6. </if>
7. <if test="teacherName != null" >
8. teacher\_name,
9. </if>
10. <if test="teacherClassid != null" >
11. teacher\_classID,
12. </if>
13. <if test="tImg != null" >
14. T\_img,
15. </if>
16. <if test="tInfo != null" >
17. T\_info,
18. </if>
19. <if test="tStatus != null" >
20. T\_status,
21. </if>
22. </trim>
23. <trim prefix="values (" suffix=")" suffixOverrides="," >
24. <if test="teacherid != null" >
25. #{teacherid,jdbcType=BIGINT},
26. </if>
27. <if test="teacherName != null" >
28. #{teacherName,jdbcType=VARCHAR},
29. </if>
30. <if test="teacherClassid != null" >
31. #{teacherClassid,jdbcType=INTEGER},
32. </if>
33. <if test="tImg != null" >
34. #{tImg,jdbcType=VARCHAR},
35. </if>
36. <if test="tInfo != null" >
37. #{tInfo,jdbcType=VARCHAR},
38. </if>
39. <if test="tStatus != null" >
40. #{tStatus,jdbcType=INTEGER},
41. </if>
42. </trim>
43. </insert>

（2）修改对象信息。

update方法接收到学生信息对象后，根据学生信息表主键id，找到与之对应的记录，执行更新操作。更新失败，抛出异常信息“修改失败！”。

关键代码：

1. <update id="updateById" parameterType="com.webmaster.model.Teacher" >
2. update t\_teacher
3. <set >
4. <if test="teacherName != null" >
5. teacher\_name = #{teacherName,jdbcType=VARCHAR},
6. </if>
7. <if test="teacherClassid != null" >
8. teacher\_classID = #{teacherClassid,jdbcType=INTEGER},
9. </if>
10. <if test="tImg != null" >
11. T\_img = #{tImg,jdbcType=VARCHAR},
12. </if>
13. <if test="tInfo != null" >
14. T\_info = #{tInfo,jdbcType=VARCHAR},
15. </if>
16. <if test="tStatus != null" >
17. T\_status = #{tStatus,jdbcType=INTEGER},
18. </if>
19. </set>
20. where teacherID = #{teacherid,jdbcType=BIGINT}
21. </update>

（3）删除学生信息。delete方法接收到学生id后，根据id删除与id对应的学生信息。删除失败，抛出异常信息“删除失败！”。操作删除学生信息时，弹窗提示用户操作。

关键代码：

1. <delete id="deleteById" parameterType="com.webmaster.model.Teacher" >
2. delete from t\_teacher
3. where teacherID = #{teacherid,jdbcType=BIGINT}
4. </delete>

（4）查询对象信息。select方法根据传入的对象属性的值，为方法内的属性进行赋值，为空的属性将不被插入到查询语句中。把查询结果封装为json对象，返回给前端。

关键代码：

1. <select id="getAll" resultMap="BaseResultMap" >
2. select
3. <include refid="Base\_Column\_List" />
4. from t\_teacher
5. </select>

查询界面如图6-4所示。



图6-4查询界面如图

总结与展望

本论文主要设计了琴行信息化管理系统，主要针对现在已有的信息管理功能过于落后和不够精确，不利于琴行人员管理，学生和琴行两边还停留在纸质登记报名等诸多弊端。将纸质转到电脑，将多种信息整合到一个系统中，进行统一的管理，在学习web应用开发技术的基础上，基于主流前端和后端多种框架结合，研发了基于Java的琴行预约选课系统。

在结构设计上，我采用的是角色模块化设计，首先确定了三中角色，学员、老师和管理员。然后又细分功能模块：

学生：登录模块，选课模块，订单管理模块。

教师：登录模块，课程安排模块。

管理员：学员管理模块，教师管理模块，订单管理模块，公告模块等

基于java的琴行预约选课系统的开发基本大致完成，已经根据详细设计初步完成了系统的大致业务模块功能，但是仍有许多不足：

(1) 计划的在线支付、请假和会员积分功能暂时由于某些不能实现。

(2) 在系统开发初期过程中，在数据库设计方面经验的有所欠缺，不能根据现在公司标注设计出合理的数据库。

(3) 未能真正的运用到实际当中去，在功能合理性是有一定程度的偏差。

以上存在的不足，在以后的时间里，会需要进一步的改进和发展。通过本次项目设计与实现，让我对系统架构设计和系统开发有了进一步的理解，明白了自己的不足，让我学到了许多宝贵的经验，也让我对于以后的开发工作有了更大的动力。

参考文献

[1] 刘东飞, 郑焕. 基于JAVA开发WEB应用中MVC模式的研究[J]. 软件导刊(22):10-12.

[2] 杨家炜. 基于Spring Boot的web设计与实现[J]. 轻工科技, 2016(7):86-89.

[3] 张峰. 应用SpringBoot改变web应用开发模式[J]. 科技创新与应用, 000(023):193-194.

[4] 徐雯, 高建华. 基于Spring MVC及MyBatis的Web应用框架研究[J]. 微型电脑应用, 2012(07):5-8+14.

[5] 刘玲霞, 王凌, 周斌, et al. MVC模式在Tomcat动态管理中的应用%Application of MVC Pattern to Dynamic Management of Tomcat[J]. 计算机工程, 031(6):216-217.

[6] 胡敏. Web系统下提高MySQL数据库安全性的研究与实现[D]. 北京邮电大学, 2015.

[7] 赵光亮, 舒小松. Navicat for MySQL平台中的SQL语言分析与应用[J]. 无线互联科技(19).

[8] 莫倩, 李炽明. 基于Java技术的Web环境下分布式数据库互操作性的实现[J]. 微型机与应用, 1999(3):49-53.

[9] 周萍, 赵娜, 李慕. Bootstrap框架在响应式Web设计中的应用[J]. 软件导刊(6):138-140.

[10] 陈员义, 李艺志. 基于Bootstrap响应式Web前端研究[J]. 福建电脑, v.31(12):76-77.

[11] 李光师. AJAX技术在动态Web开发中的应用探讨[J]. 科学技术创新, 2010(17):69-69.

[12] 王艳清, 陈红. Research and design of intelligent web system based on SSM frame%基于SSM框架的智能web系统研发设计[J]. 计算机工程与设计, 2012, 033(012):4751-4757.

[13] 范会联, 李献礼. 基于J2EE的网上选课系统设计[J]. 计算机应用研究, 2006, 23(9).