

扬州市职业大学

毕业设计说明书

题目：基于微信公众平台的校外实习管理分析系统的设计与开发--后台管理模块

学 院： 信息工程学院

专 业： 计算机网络技术

班 级： 网络1403

姓 名： 倪志兵

学 号： 140402318

指导教师： 周 欢

完成时间： 2017年5月

摘要

高职院校以培养高素质技能型人才为主要目标，因此实习实训环节尤为重要。随着校企合作的不断深入，越来越多的学生走向实习岗位，对实习生的资料以及实习考勤情况的信息管理给班主任增加了许多工作量。因此，在校外实习期间，学院如何有效管理学生，了解统计各班学生出勤实习情况，减少班主任工作量，提高管理人员工作效率，是一项迫切需要解决的问题。本课题来自我校纺织学院，旨为学院的学生管理工作人员设计开发一个基于微信公众平台的学生校外实习管理分析信息化管理系统，利用信息化手段，推动学院校外实习管理水平的提高。

系统采用MyEclipse2017和MySQL关系型数据库及Navicat开发工具来实现的，其界面简洁直观、便于操作。整个项目采用Maven进行管理，便于部署和管理；整体框架为springboot，结构清晰，代码管理方便；项目中采用mybatis连接数据库，代码独立，方便转换不同数据库类型。主界面采用HTML中的Iframe框架，这种框架使得页面的布局操作更加便捷。

系统前台基于微信公众平台进行二次开发，设计完成有学生客户端和教师客户端两部分，可实现身份认证、学生签到、查阅签到历史记录、师生互动交流等功能。

后台使用Java开发技术，设计完成基于Web端管理分析平台，实现了管理员对教师、班级、学生、实习单位管理及分配的功能；同时班主任可管理本班的班级、学生信息以及进行实习单位的管理和分配，并实现对学生签到的有效分析。本人承担了Web端后台管理设计任务，完成了登录管理、教师管理、班级管理、学生管理、实习单位管理五个子模块的开发。

关键字： SpringB00t MySQL MVC 校外实习

目 录

摘 要.....	I
1 引言.....	1
1.1 开发背景.....	1
1.2 研究目的与意义.....	1
2 相关技术简介.....	3
2.1 SpringMVC.....	3
2.2 SpringBoot.....	4
2.3 MyBatis 及 JDBC 技术.....	4
2.4 Maven 技术.....	4
2.5 MySQL 数据库.....	5
3 系统分析.....	6
3.1 需求分析.....	6
3.1.1 功能需求分析.....	6
3.1.2 性能需求分析.....	6
3.2 可行性分析.....	6
4 系统设计.....	8
4.1 系统总体结构设计.....	8
4.2 数据库设计与连接.....	10
4.2.1 数据库与表的设计.....	10
4.2.2 数据库连接.....	15
4.3 系统技术开发流程.....	15
5 后台管理模块的实现.....	17
5.1 后台登录.....	17
5.2 教师管理.....	21
5.2.1 添加教师帐号.....	21
5.2.2 修改教师信息.....	25
5.2.3 删除教师信息.....	26
5.3 班级管理.....	28
5.4 学生管理.....	28
5.5 实习单位管理.....	29
5.5.1 管理实习单位信息.....	30
5.5.2 学生实习单位分配.....	32
6 系统测试.....	36
6.1 系统测试方法.....	36
6.2 测试内容.....	36
6.3 测试结果.....	36
7 总结.....	38
参考文献.....	38
致 谢.....	39

1 引言

1.1 开发背景

高职院校以培养高素质技能型人才为主要目标，因此实习实训环节尤为重要。随着校企合作的不断深入，越来越多的学生走向实习岗位，实习指导过程中产生的信息量也在急剧膨胀，对实习生的资料以及实习考勤情况的信息管理给班主任增加了许多工作量。对于如此大规模的实习管理工作，人工的手工处理已经不能满足这种大数据处理的需要，这种管理方法太过片面，不能及时得到实习生的实习情况和企业的实习意见，更无法对大数据进行有效的管理。一旦实习时间过长，实习生与学校的联系就会逐渐减少，所以这种管理方式已然不能适应当下不断变化的实习需求。

随着互联网的快速普及以及网络通信技术的快速发展，如何利用这些技术加强实习单位、实习生和学校的联系已然成为各高校关注的焦点。因此，以互联网技术为基础，全国各高校围绕着如何更为便捷的管理实习生的实习问题而展开了许多研究，开发了一些单机版的信息管理系统。

虽然单机版本的部分管理系统软件已经在一定的程度上弱化了学生管理压力，可是在当前信息变化迅速、信息流量巨大的现实面前，其不足之处也就逐渐显现出来了。首先，单机版的信息管理系统只能为学校管理人员使用；其次，当学生个人信息及实习单位信息发生变化，而当这些信息被多处使用时，需要向很多管理人员联系，在及时性、有效性、全面性等方面存在一定的不足；最后，单机版的软件版本更新不能保证，有的单位经常不更新软件或是更新软件操作困难，造成数据迁移太过麻烦。

伴随智能手机的普遍，互联网和手机媒体的高速发展，大学生使用智能手机软件，尤其是手机微信已经越来越普遍。采用当前流行的移动互联网技术开发一套基于服务器端和手机微信客户端的校外实习管理分析系统是一个很好的解决方案。

1.2 研究目的与意义

我们团队受扬州市职业大学纺织服装学院委托开发了这套“基于微信公众平台的高校校外实习管理分析系统”。本系统以实习学社和管理老师这两个经常进行信息查看与维护的群体为中心，以 J2EE 技术作为开发的基础，借助微信公众平台实现学生在外实习期间的地理位置定位，通过网络记录并跟踪每一个毕业生的实习信息和实习情况，不仅便于学校顺利实行实习计划，还方便了实习生与老师之间的及时交流。做到教师及时了解每个学生的在岗实习情况，对实习生的

工作情况进行准确指导、远程监督，提高了对实习生的管理质量。

本系统以现代网络技术作为基础，构建学生校外实习管理分析系统，完善了学生与实习单位之间的衔接功能，满足高校、实习单位和学生多方的需求。学生通过系统将自己的实习经历上传到系统，教师只要通过网络就能第一时间在这个系统上查询到实习生的相关信息，并且管理人员能直接通过系统对该实习生进行管理，减少了很多不必要的麻烦。系统为学院、实习单位、学生搭建一个方便快捷的交流平台对于学生与教师间的信息共享，促进当前大学生实习信息管理建设有一定的积极意义。

2 相关技术简介

2.1 SpringMVC

Spring 是有一个普及率非常高的 MVC 框架，MVC 即 Model - View - Controller，该模式能很好地将数据、业务与展现进行分离。使用 Spring 带来的可插入的 MVC 架构，可以选择使用其本身具有的 SpringWeb 框架还是其他类似 Struts 这样的 Web 框架。通过 MVC 提供的策略接口，Spring 框架可以非常便捷的进行配置，而且 Spring 还提供了多种视图技术，比如常见的 JSP 技术、Velocity、Tiles、iText 和 POI 等。而 SpringMVC 框架并不清楚其使用的视图类型，所以不会强迫开发者只使用 JSP 技术或者其他某一技术。SpringMVC 这一结构将控制器、模型对象、分派器以及处理程序对象的这些不同的角色进行分离，这种分离让这些对象更容易配置与调试。

Spring MVC 的请求处理过程如图 2.1:

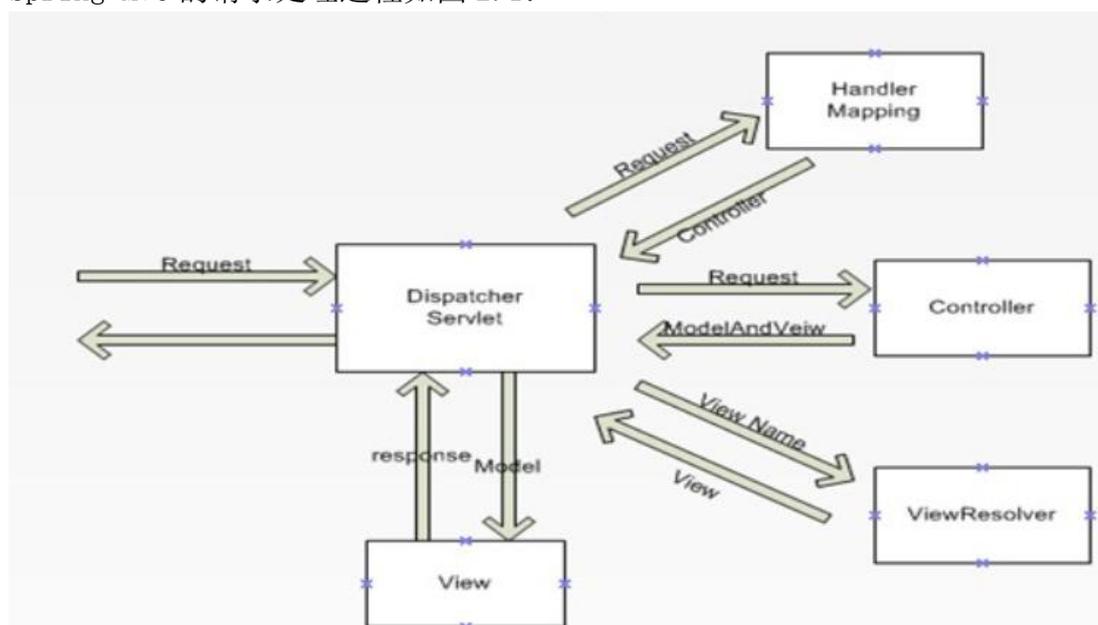


图 2.1 MVC 处理流程

- 1、客户端请求提交到 DispatcherServlet;
- 2、DispatcherServlet 这一核心控制器查询一个或者查询多个 HandlerMapping，找到可以处理这个请求的 Controller;
- 3、核心容器 DispatcherServlet 将收到的请求提交到可以处理这个请求的 Controller;
- 4、Controller 接收到请求后调用业务逻辑处理功能，并将这些请求处理后的结果返回 ModelAndView;
- 5、控制器 DispatcherServlet 查询一个或者多个 ViewResoler 视图解析器，

找到 ModelAndView 制定的视图；

6、视图负责将结果显示到浏览器客户端。

2.2 SpringBoot

Spring Boot 是由 Pivotal 团队提出的全新框架，其设计目的是简化新 Spring 应用的初始搭建以及开发过程。通常的 Spring 网站随着功能及逻辑业务的逐渐复杂，随之带有大量的 XML 配置文件和复杂的 Bean 依赖关系。而 Springboot 正是在这样一个环境下抽象而出的开发框架，其本身并不提供 Spring 框架的核心特性和拓展功能，仅用于快捷地开发新一代以 Spring 框架为基础的应用程序。因为使用不同的方式进行配置，使得开发人员不再需要进行样板化的配置。虽然如此，它却集成了许多常用的第三方库配置（如 JDBC、Jackson 等），在 Springboot 应用中这些库几乎可以开箱即用，大部分的应用都只需要少量的配置代码，通过这种方式，Spring Boot 致力于使快速应用开发领域（rapid application development）蓬勃发展。其特点如下：

- 1、创建独立的 Spring 应用程序
- 2、自动配置 Spring
- 3、简化 Maven 配置
- 4、嵌入的 Tomcat，无需部署 WAR 文件
- 5、提供生产就绪型功能
- 6、绝对没有代码生成和对 XML 没有要求配置

2.3 MyBatis 及 JDBC 技术

MyBatis 能够支持定制化 SQL 语句、存储过程以及高级映射功能的持久层框架。MyBatis 技术几乎涵盖了连接所有数据库类型的 JDBC 代码及常用设置参数。MyBatis 配置的 XML 或是注解能够将接口和 Plain Old Java Objects 映射成为几乎所有类型数据库中的记录。

JavaDataBaseConnectivity 是一套能够让 Java 与 SQL 数据库建立对话的程序设计接口。作为执行 SQL 语句的 JavaAPI，其本身是一种 Java 应用程序，它是连接数据库、存取数据的一种处理机制，当访问多种不同类型的数据库时，它可以为其提供一个统一的访问。如今，JDBC 已经成为一套标准 API 给开发人员使用。利用这一技术，用户就不再需要为每一个关系数据库单独编写一个程序。

2.4 Maven 技术

Maven 用一小段简单的描述信息，就能够管理项目的构建。以 POM 作为基础（POM 全称为 ProjectObjectModel，项目对象模型），来管理软件开发过程中的各项文件。项目中 pom.xml 较为重要，作为 Maven 项目的描述文件，pom.xml 的

功能类似于 antx 的 project.xml 文件。Pom.xml 文件描述了项目的众多详细信息，这些信息包括项目的名称、版本、项目 ID 等一些列描述。

Maven 是 Apache 的一个顶级项目，更是一个强大的构建工具，从清理、编译、测试直至生成报告，都能够帮我们自动化构建过程。其通过一个坐标系统准确地定位构建 (artifact)，它给类库世界引入了坐标，我们可以借助它来管理依赖关系，解决繁杂的依赖问题。

2.5 MySQL 数据库

MySQL 作为一种关系型数据库管理系统 (RDBMS)，其源代码完全开放，并且使用常用的数据库语句进行数据库管理。

MySQL 数据库的第一个版本在 1988 年初发布。它的多线程运行模式源于其核心模块的多线程机制，同时还提供了针对多种编程语言的编程接口 (APIs)，不仅支持多种字段类型，还提供完整的操作符，支持查询中的 select 和 where 操作。随着时间推进和技术的进步，MySQL 将对 ANSI98/ANSI99 标准完全兼容。

常见的可视化数据库管理软件：MySQLWorkbench、SQLyog、phpMyAdmin、AquaDataStudio、MySQLFront、Navicat 等。

3 系统分析

3.1 需求分析

3.1.1 功能需求分析

本系统旨在及时获取学生校外实习考勤信息，并对考勤数据进行统计、汇总和分析。主要针对三类用户，其对应的功能如下：

1、学生用户：微信注册功能、登录功能、信息完善功能、修改密码、退出系统、微信帐号绑定功能；

2、教师用户：微信登录功能、学生信息管理功能、修改密码、统计功能、班级管理功能、退出系统；

3、管理员用户：WEB 端登录功能、帐号管理功能、修改密码、退出系统、学生管理、班级管理、教师管理、实习单位管理、学生实习出勤情况分析汇总等。

3.1.2 性能需求分析

本系统的性能需求原则如下：

- 1、准确性：要有足够高的准确率以保证系统的正确运行；
- 2、灵活性：软件系统的结构设计应灵活高效，能够快速地实现用户的新需求；
- 3、可维护性：软件运行中发现错误时，能够迅速、准确定位、诊断并解决；
- 4、可测试性：设计时尽可能减少软件各功能所需的工作量；
- 5、可理解性：本软件提供的各种命令及信息提示，应该易于用户理解。

3.2 可行性分析

1、系统适用性

为了更好的解决工作人员管理学生实习信息这些大数据处理问题，其中不仅涉及到实习管理工作的方方面面，还需要为管理工作设计出合理的管理措施，使大数据处理问题得到全面解决。

2、系统信息强大性

信息管理系统的核心就是信息的记录和处理。本系统采用 Web 技术与 MySQL 数据库技术相结合的方式，保证后台数据记录的准确与安全。

3、系统的先进性

本系统不仅在数据库的结构、数据库格式以及分类方法等开发内容上做到精简便捷，还在操作系统、使用平台、使用人群上具有广泛适用性。

4、系统运行环境稳定性

本系统采用合理的设计技术，Web 访问的方式能够适应多种操作系统环境，而且在后期升级维护上十分方便，只需要升级服务器端，即做到了实时升级，又能够实现数据的无缝迁移。

5、系统操作简便性

本系统采用 Web 前端技术，将数据库操作过程和数据处理过程可视化，极大的减少了管理人员的学习难度。

3.3 系统运行环境

数据服务器系统配置：数据库服务器安装于 Windows 10 系统，系统内存 8G 或更高，采用 MySQL 数据库类型，可视化管理软件采用 Navicat。

应用服务器系统配置：应用服务器与数据服务器共用同一台服务器，服务器安装 Windows 10 系统，系统内存 8G 或更高。应用服务器基本配置如下：

- 1、JVM/JDK：JDK 1.8.0
- 2、开发语言：HTML+Java
- 3、开发平台：MyEclipse 2017、Dreamware 2017
- 4、数据库：MySQL 5.7
- 5、应用程序服务：Springboot 集成 Jetty 9.2.14
- 6、开发技术：JSP、SpringMVC

4 系统设计

4.1 系统总体结构设计

系统总体功能结构如图 4.1 所示,系统分为微信客户端和 Web 管理分析平台两部分。微信客户端又分为教师客户端和学生客户端,教师用户通过访问微信公众号,完成身份认证之后,可以查看自己本班学生的签到出勤情况,也可通过师生交流模块与学生进行个别交流,并可及时回复学生的留言。学生用户通过访问微信公众号,首先进行身份认证,进而能提交或修改所在实习单位信息,此后能通过学生签到模块实现每日签到;通过签到历史模块,查看自己既往的签到统计信息和每日签到详细信息;通过师生交流模块,可向教师留言提问。

Web 管理分析平台为管理员和教师用户所使用。管理员可以通过该平台管理全院所有学生的全部信息,分析统计学生实习情况;教师用户仅可管理自己本班级的学生信息。该平台基于 Java Web 开发技术实现,分为后台管理、统计分析两个模块。

本人负责完成后台管理模块,如图 4.2 所示。具体实现有五个子模块:登录管理、教师管理、班级管理、学生管理、实习单位管理功能。

(1) 登录管理子模块验证用户登录的账户及密码是否正确并判断登录账户的权限。

(2) 教师管理子模块是便于管理员管理教师信息,实现教师信息的新增、修改、删除和查询功能。

(3) 班级管理子模块是便于管理员管理班级信息,实现班级信息与班主任相绑定,以及班级信息的新增、修改、删除和查询功能。

(4) 学生管理子模块便于管理员和教师管理学生信息,将学生信息与班级相绑定,以及学生信息的新增、修改、删除和查询功能。

(5) 实习单位管理子模块便于管理员和教师管理实习单位信息和分配学生功能。单位信息管理包括新增、修改、删除和查询信息。分配学生功能是指将学生分配到某实习单位或者解除学生与单位间的实习关系。在分配学生时,既可以按照班级分配学生至单位,也可以逐一分配,同时,在批量分配时,系统能识别出已有实习单位的同学,提示是否需要重新分配到新的单位。

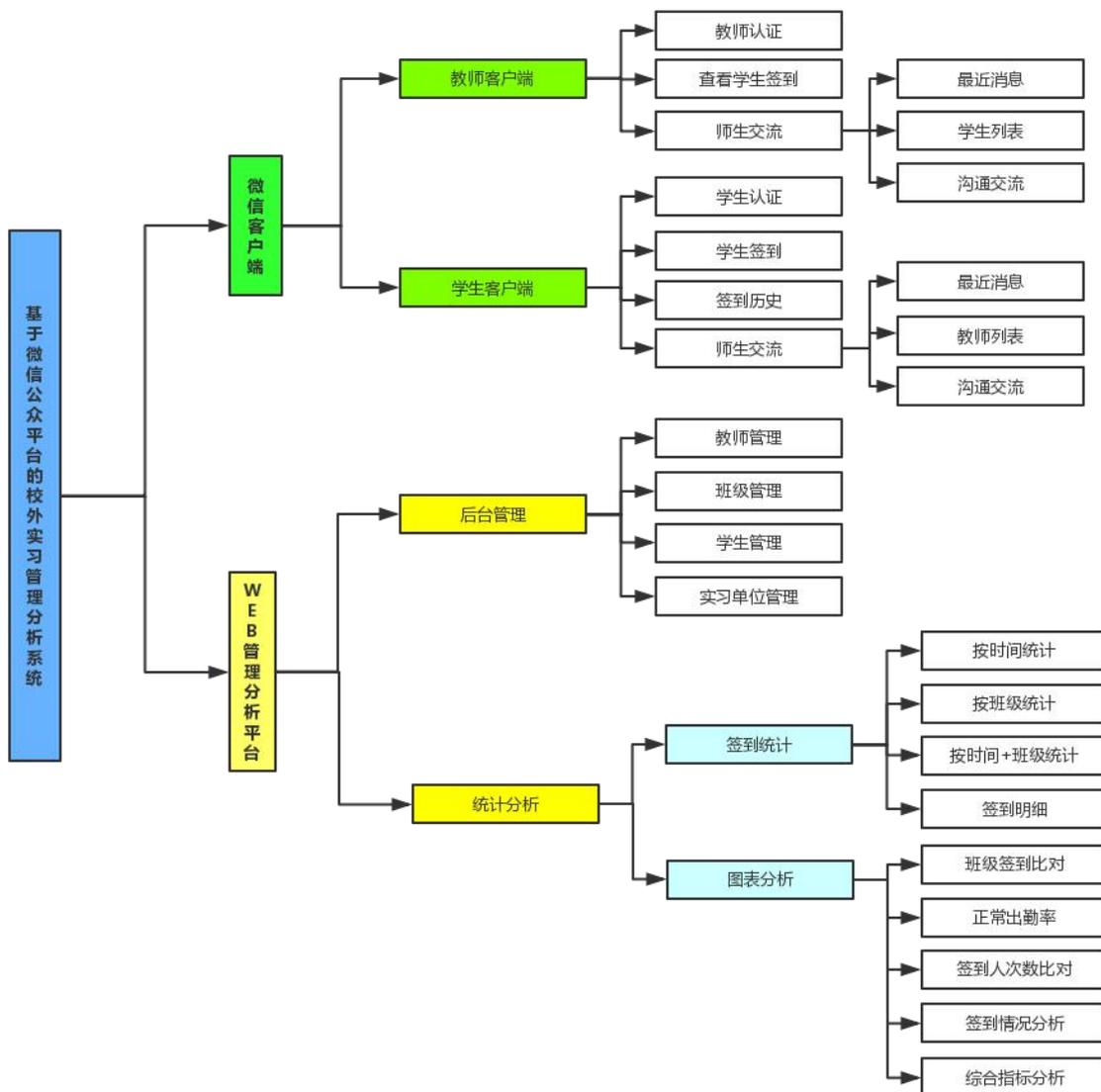


图 4.1 系统总体结构图

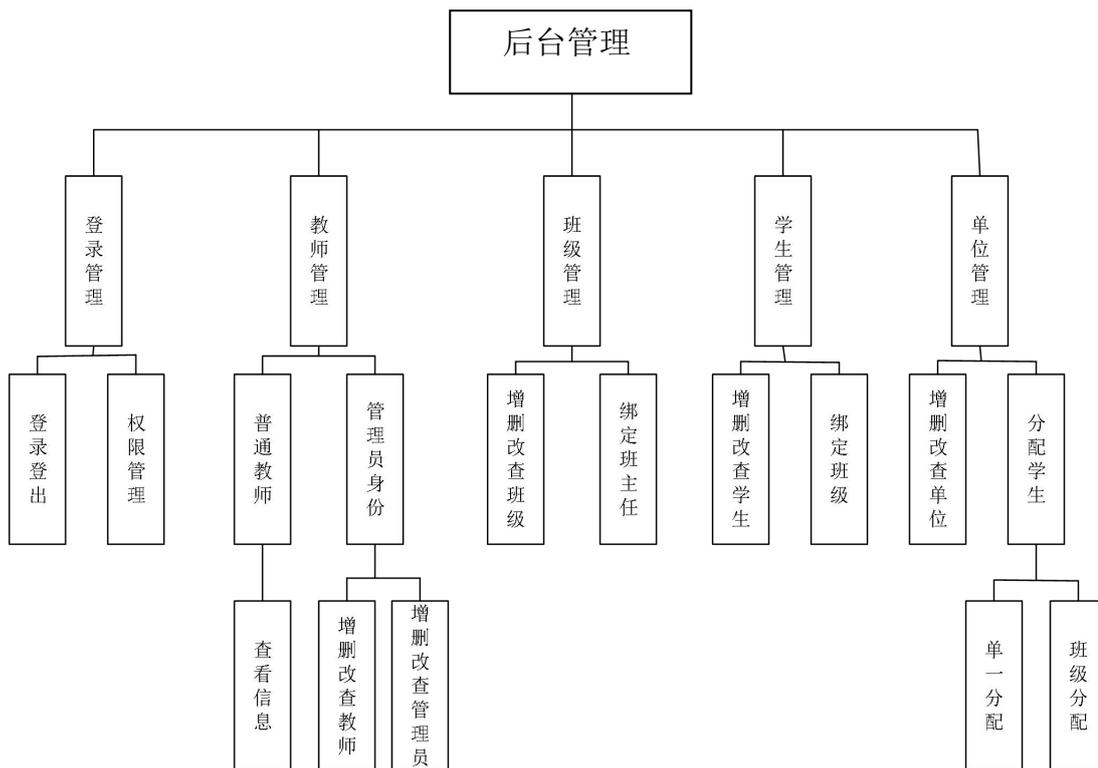


图 4.2 后台模块结构图

4.2 数据库设计与连接

该系统可满足学校对毕业生校外实习情况的远程管理，及时了解学生的出勤情况、进行师生交流，进而实现高效率、规范化的管理。通过对系统需求的分析，得知本系统中有学生、班级、教师、签到地点、单位实体，各实体间的关系为：班级与学生之间为一对多的关系，学生与签到地点之间是一对多的关系，教师对班级之间为一对一的关系，实习单位与学生为一对多的关系等。

4.2.1 数据库与表的设计

数据库的设计一共包含九个表，主要包括学生表、实习单位表、教师表、消息表、教师班级表、用户与学生关系表、用户与教师关系表、用户签到表和用户表。各数据的清单，如表4.1所示：

表 4.1 数据库中数据表清单

序号	表名称	注释
1	t_student	学生表
2	t_work	实习单位表
3	t_teacher	教师表
4	t_message	消息表
5	t_class	班级表
6	t_user_student	用户与学生关系表
7	t_user_teacher	用户与教师关系表
8	t_user_sign	用户签到表

9	t_user	用户表
---	--------	-----

各个数据表的说明如下所示：

1. 学生表(t_student)

主要用于存储学生信息。各项相关数据包括每个学生的学号、学生编号、姓名、班级编号、班级名称、联系方式、教师姓名、单位编号、状态等。其中用学号为主键。数据库各个字段数据类型及说明如表4.2所示：

表 4.2 学生表 (t_student)

字段名	数据类型	主/外键	注释
XH	varchar(50)	PK	主键
STUDENT_NUM	varchar(50)		学生编号
STUDENT_NAME	varchar(20)		学生姓名
CLASS_NUM	varchar(50)	FK	班级编号
CLASS_NAME	varchar(20)		班级名称
PHONE_NUM	varchar(15)		手机号码
TEACHER_NAME	varchar(20)		教师姓名
WORK_NUM	varchar(50)		实习工作地点编号
STATE	char(1)		状态 1 正常, 0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间

2. 实习单位表(t_work)

用于存储实习单位信息，表中各个字段数据类型及说明如表4.3所示：

表 4.3 实习单位表(t_work)

字段名	数据类型	主/外键	注释
WORK_NUM	varchar(50)	PK	实习工作地点编号
WORK_NAME	varchar(500)		工作地点名称
WORK_TEACHER	varchar(20)		带队老师
CHECKIN_TIME	datetime		规定考勤时间
CHECKIN_LONGITUDE	double(20,6)		规定考勤经度
CHECKIN_LATITUDE	double(20,6)		规定考勤纬度
CHECKIN_ADDRESS	varchar(2000)		规定考勤地址
STATE	char(1)		状态 1 正常, 0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间
AFTER_HOUR	int(11)		几点后开始考勤
BEFORE_HOUR	int(11)		几点前需完成考勤

3. 教师表(t_teacher)

用于存储教师信息，表中各个字段数据类型及说明如表4.4所示：

表 4.4 教师表 (t_teacher)

字段名	数据类型	主/外键	注释
TEACHER_NUM	varchar(50)	PK	教师编号
USER_NAME	varchar(20)		登录账号
USER_PASSWORD	varchar(100)		密码
TEACHER_PHONE	varchar(20)		教师手机号码
IS_ADMIN	char(1)		管理员标记，1 是 0 否
TEACHER_NAME	varchar(20)		教师姓名
STATE	char(1)		状态 1 正常，0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间

4. 消息表(t_message)

用于存储师生间的交流信息，表中各个字段数据类型及说明如表4.5所示：

表 4.5 消息表 (t_message)

字段名称	数据类型	主/外键	注释
PK_ID	varchar(50)	PK	消息编号
FROM_USER	varchar(50)		消息来源 OPENID
TO_USER	varchar(50)		发送给 OPENID
TEXT	varchar(4000)		内容
CREATE_TIME	datetime		新增时间
IS_READ	char(1)		1 为未读；0 为已读

5. 班级表(t_class)

用于存储班级信息，表中各个字段数据类型及说明如表4.6所示：

表 4.6 班级表 (t_class)

字段名称	数据类型	主/外键	注释
CLASS_NUM	varchar(50)	PK	班级编号
CLASS_NAME	varchar(20)		班级名称
PHONE_NUM	varchar(15)		手机号码
TEACHER_NUM	varchar(50)	FK	教师编号
TEACHER_NAME	varchar(20)		教师姓名
STATE	char(1)		状态 1 正常，0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间

6. 用户与学生关系表(user_student)

用于存储学生学号与个人微信账号间的对应关系，表中各个字段数据类型及说明如表4.7所示：

表 4.7 用户与学生关系表 (user_student)

字段名称	数据类型	主/外键	注释
PK_ID	varchar(50)	PK	主键
XH	varchar(50)	FK	人员编号
USER_NUM	varchar(50)		用户编号，OPENID
STUDENT_NUM	varchar(50)	FK	学生编号

STUDENT_NAME	varchar(20)		学生姓名
STATE	char(1)		状态 1 正常, 0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间

7. 用户与教师关系表(t_user_teacher)

用于存储教师工号与个人微信账号间的对应关系, 表中各个字段数据类型及说明如表4.8所示:

表 4.8 用户与教师关系表 (t_user_teacher)

字段名称	数据类型	主/外键	注释
PK_ID	varchar(50)	PK	主键
USER_NUM	varchar(50)	FK	用户编号, OPENID
USER_NAME	varchar(20)		登录名
STATE	char(1)		状态 1 正常, 0 无效
U_TIME	datetime		更新时间
C_TIME	datetime		新增时间

8. 用户签到表(t_user_sign)

用于记录每位学生的签到时间和签到位置等信息, 表中各个字段数据类型及说明如表4.9所示:

表 4.9 用户签到表 (t_user_sign)

字段名称	数据类型	主/外键	注释
SIGN_NUM	varchar(50)	PK	签到主键, UUID
USER_NUM	varchar(50)		用户编号, OPENID
SIGN_DATE	date		签到日期 YYYY-MM-DD
SIGN_TIME	datetime		签到时间, 精确到毫秒
SIGN_ADDRESS	varchar(500)		签到点地址
SIGN_LONGITUDE	double(20, 6)		纬度
SIGN_LATITUDE	double(20, 6)		经度
LATE	char(1)		是否迟到, 1 是, 0 否, 2 早到
OFFSET	double(20, 2)		与设定点偏差距离, 单位为米
STUDENT_NUM	varchar(20)	FK	学号
STUDENT_NAME	varchar(20)		学生姓名
CHECKIN_TIME	datetime		规定考勤时间
CHECKIN_LONGITUDE	double(20, 6)		规定考勤经度
CHECKIN_LATITUDE	double(20, 6)		规定考勤纬度
CHECKIN_ADDRESS	varchar(500)		规定考勤地址
CLASS_NAME	varchar(50)		班级名称
PHONE_NUM	varchar(15)		手机号码
TEACHER_NAME	varchar(50)		教师
XH	varchar(50)	FK	用户序号

AFTER_HOUR	int		几点后开始考勤
BEFORE_HOUR	int		几点前需完成考勤
WORK_NUM	varchar(50)	FK	实习工作地点编号
WORK_NAME	varchar(500)		工作地点名称
WORK_TEACHER	varchar(20)		实习老师
CLASS_NUM	varchar(60)		班级编号
TEACHER_NUM	varchar(60)		教师编号

9. 用户表 (t_user)

用于记录每位访问微信公众号的用户微信账号信息和访问记录, 表中各个字段数据类型及说明如表4.10所示:

表 4.10 用户表 (t_user)

字段名称	数据类型	主/外键	注释
USER_NUM	varchar(50)	PK	用户编号, OPENID
NAME	varchar(200)		姓名
PHONE	varchar(20)		联系电话
SEX	char(1)		性别. 1 是男性, 值为 2 是女性, 值为 0 是未知
AGE	decimal		年龄
NICKNAME	varchar(200)		昵称
PROVINCE	varchar(200)		省份
CITY	varchar(20)		城市
COUNTRY	varchar(20)		国家
HEADIMGURL	varchar(300)		头像
PRIVILEGE	varchar(2000)		特权信息. 用户特权信息, json 数组, 如微信沃卡用户为 (chinaunicom)
UNIONID	varchar(200)		unionid, 只有在用户将公众号绑定到微信开放平台帐号后, 才会出现该字段
USER_LV	decimal		用户等级
STATUS	char(1)		状态, 0 未关注, 1 已关注, 2 取消关注
USERINFO	char(1)		是否 snsapi_userinfo 认证 0 未认证, 1 已认证
ATI_NUM	decimal		关注次数, 第一次为 1, 如果取消后再关注, 累加
ATI_DT	timestamp		关注时间
IN_DT	timestamp		加入时间, 只扫码不关注
OUT_DT	timestamp		取消关注时间
ATI_SOURCE	char(1)		关注来源, 1 自行加入, 2 推广加入
ATI_USER	varchar(50)		推广用户, 二维码广播用户编号

USERINFO_DT	timestamp		更新用户信息时间
-------------	-----------	--	----------

4.2.2 数据库连接

Springboot 中配置 JDBC 驱动连接 MySQL 数据库，需要 2 个步骤：

1、pom.xml 配置 maven 依赖，加载 MySQL 的 JDBC 驱动

```

<!-- MYSQL -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
</dependency>
<!-- Spring Boot JDBC -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>

```

2、在属性文件 application.properties 中配置链接信息

#数据库的 URL 地址及编码格式

```

spring.datasource.url=jdbc:mysql://127.0.0.1:3306/bsuseUnicode=true&characterEncoding=UTF8&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&useSSL=true&serverTimezone=UTC

```

#用户名

```
spring.datasource.username=root
```

#密码

```
spring.datasource.password=1234
```

#数据库驱动

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

4.3 系统技术开发流程

本系统采用 MVC 框架技术进行开发，它将程序的输入、处理和输出部分相分离，分为 Model（数据模型）、View（视图）和 Controller（控制器）三层，有助于管理复杂的应用程序。其具体执行过程为：用户使用 WebBrowser（浏览器）访问 Web 页面，根据页面提示及个人需求点击页面中的功能按钮；页面中的按钮根据其自身功能向 Controller 控制层（由 Servlet 容器负责）提交 Request（请求），Controller 根据接收到的请求类型来判断可以处理这个请求的方法，并通过 Model 层（由 JavaBean 实体类实现）将请求的数据与数据库（DB）中的表结构映射并通过数据持久层（DAO）进行 DML 操作；Controller 判断请求需返回

的页面视图 (View)，由 Model 将返回数据交由选择的 View 进行显示；View 将数据及页面整合为一个完整的页面后响应 (response) 给用户的 WebBrowser；最终用户看到所需的信息或结果，如图 4.3 所示。

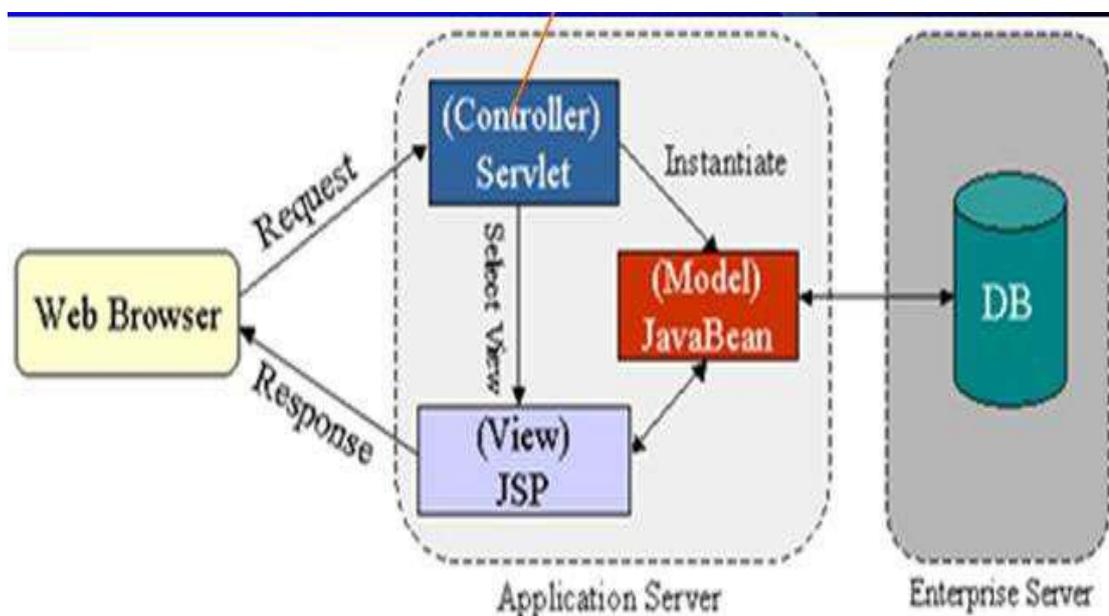


图 4.3 控制流程

各层次在开发中所使用的页面及文件不同，下面以添加班级功能为例，说明本系统的开发结构，如表 4.11 所示：

表 4.11 文件结构说明

序号	层	文件路径	技术简介
1	View 视图	Src/main/resources/templates/admin/manage/class-add.html	使用 html、css 样式表。通过 ajax 技术实现无刷新与后台 Controller 进行数据交互
2	Controller 控制层	Src/main/java/com/yzpc/weixin/controller/AdminController.java	控制器有返回新页面和返回 Json 数据两种类型，此处采用返回 Json 数据，其方法返回信息为 Map
3	Service 业务处理层	Src/main/java/com/yzpc/weixin/service/AdminService.java	此处进行数据处理，将数据按照特定类型进行传输控制
4	DAO 数据持久层	Src/main/resources/mybatis/TClassMapper.xml	对数据库进行操作，定义不同的接口，对数据表进行 DML 操作
5	Javabean 实体类	Src/main/java/com/yzpc/weixin/vo/TClass.java	对表结构进行映射，便于在 DAO 层实现 DML 操作

5 后台管理模块的实现

后台管理模块有教师管理、班级管理、学生管理和实习单位管理组成。管理人员利用该平台，实现对全院教师、班级、学生、实习单位信息的管理。

5.1 后台登录

用户在PC端通过输入以下网址访问：<http://localhost:8080/yzpc/>。打开如图所示的5.1所示的登录界面。

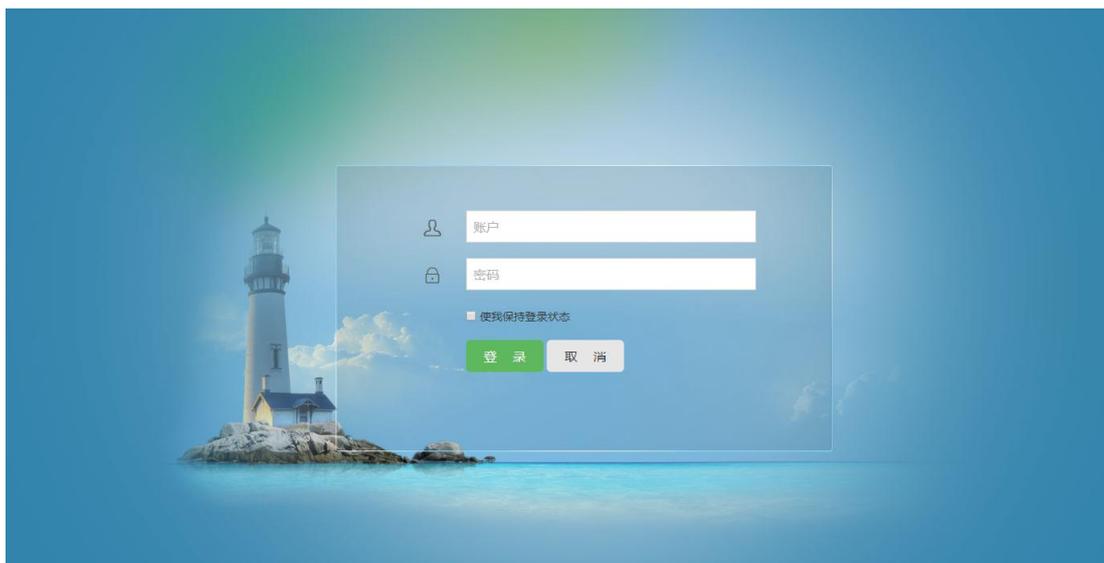


图 5.1 登录界面

用户填写登录信息，选择是否保持登录状态，若帐号密码正确，则进入操作主界面，若错误则给出提示；选择登出后，页面释放当前占用资源，关闭会话。

本功能需要开发静态页面、Controller、Service、Mapper、MyBatis各层模块，需要对不同模块进行不同包分类，各层的业务逻辑关系如图5.2所示：

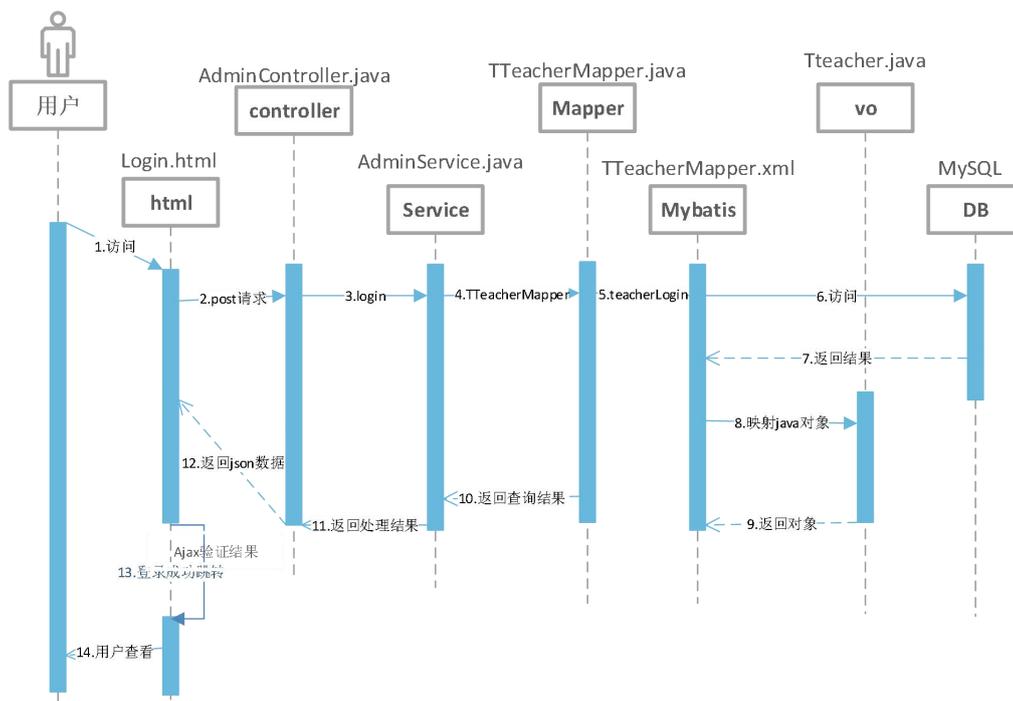


图 5.2 登录流程设计

1、视图 (view) 层

View层主要为控制页面显示的代码，包含主体的HTML代码、CSS样式表、JS文件以及图片等。CSS样式表中代码用以修饰页面的布局、背景颜色、字体颜色以及控制文本的格式设置等，代码如下：

```

body {
    -webkit-transition: -webkit-transform0.3sease;
    transition: transform0.3sease;
}
.login-box,.dorm-book,.contact-box{
    background: #fff;
}
img {
    cursor: pointer;
    vertical-align: middle;
    border: 0none;
}
input[type="radio"],input[type="checkbox"]{
    margin-top: 0;
}
    
```


件中的login功能进行判断帐号的正确性，并将结果及帐号权限返回。具体代码如下：

```
@ResponseBody
    public Map<String,String>
login(  HttpServletRequesthttpServletRequest,HttpServletResponsehttpServletRe
sponse,@RequestParam("userName") String
userName,@RequestParam("passWord") String passWord,Model model){
    TTeacher teacher = tTeacherMapper.check(userName,passWord);
    Map<String,String> result = newHashMap<String,String>();
    if(null ==teacher){
        result.put("code","0");
    }else {
        result.put("code","1");
        httpServletRequest.getSession().setAttribute("teacher", teacher);
    }
}
```

通过ajax技术实现不刷新页面而动态获取登录结果并根据结果返回对应信息。具体代码如下：

```
$(function(){
    $("#loginform").validate({
        rules: {userName: {required:true,
            minlength:2,
            maxlength:50},
            passWord: {
                required:true}},},
    },
```

```
        required:true}},},
    onkeyup:false,
    focusCleanup:true,
    success:"valid",
    submitHandler:function(form){
        layer.load(2);
        $(form).ajaxSubmit({
dataType:"json",
success:function( jsondata ){
    layer.closeAll('loading');
    console.log(jsondata);
    if( jsondata.code =='1' ){
        window.location.href="index"
    }else {layer.msg("登录失败, 用户名或密码错误", {icon:
5, time: 2000});
```

5.2 教师管理

教师管理子模块实现管理员对教师信息的新增、修改、删除和查询等功能。各功能需要分为不同页面处理，相应的要进行不同的业务逻辑流程。

5.2.1 添加教师帐号

完成登录后，点击左侧目录树中的“教师管理”如图5.3所示，进入教师管理页面完成教师的管理任务。

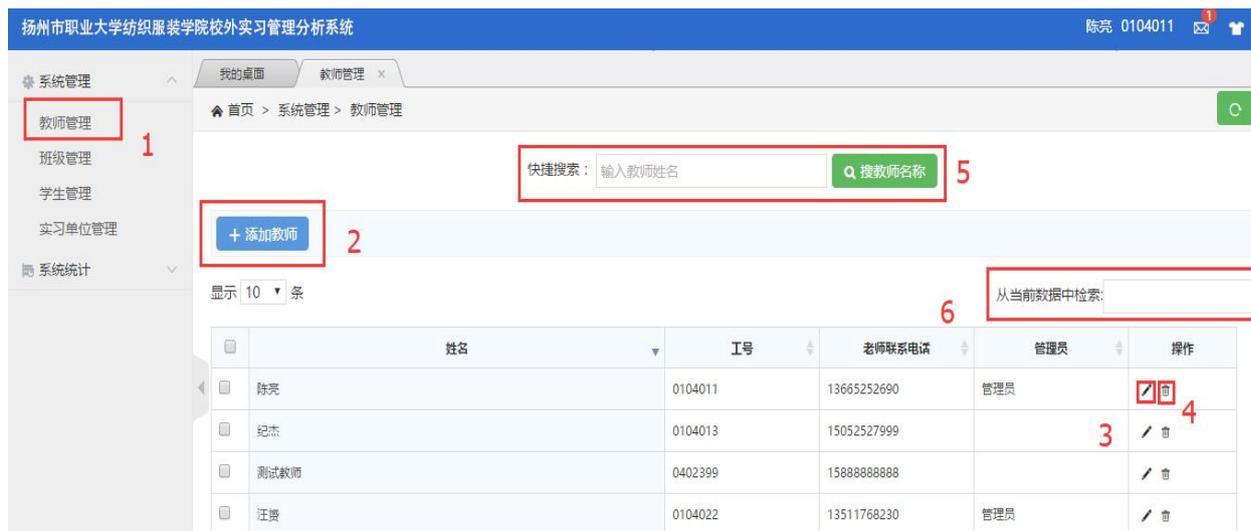


图 5.3 教师管理界面

单击图 5.3 位置 2 中添加教师按钮，进入教师添加界面如图 5.4 所示。添加教师帐号信息时，需要填写教师的登录工号、教师姓名、联系电话、是否为管理员等信息，帐号默认密码为 123456，登录后可修改密码。



图 5.4 添加教师界面

本功能入口为教师管理（默认为 teacher-list 页面）中 teacher_add 功能按钮，后台数据处理及逻辑分析如图 5.5 所示：

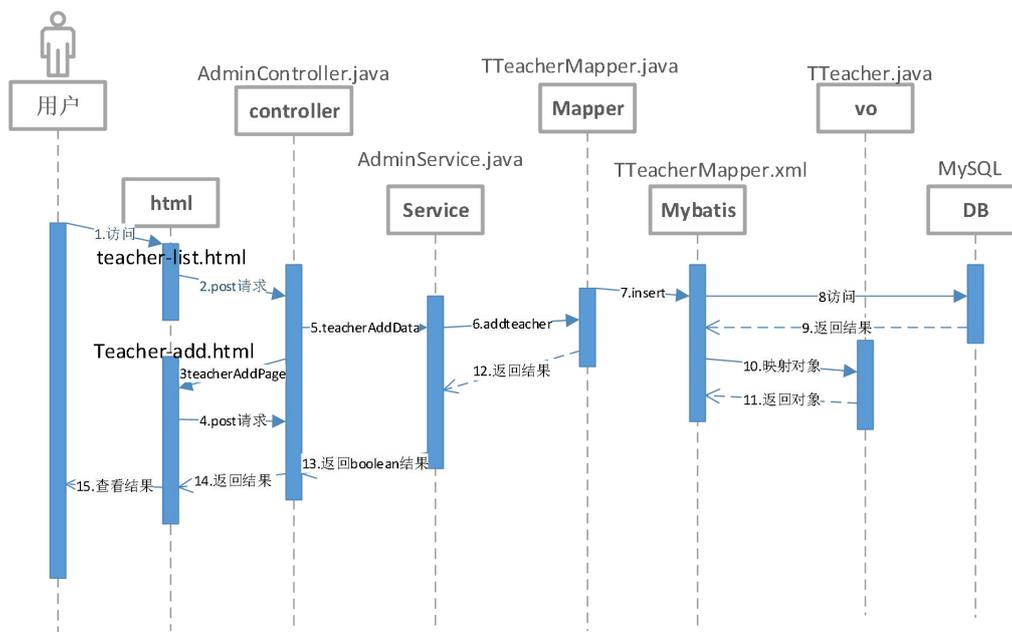


图 5.5 添加教师流程设计

核心代码如下：

通过页面功能按钮提交添加教师的请求：

```
function teacher_add(){
    layer_show('添加教师','/yzpc/teacherAddPage',"510");
}
```

通过 Controller 找到处理这个请求的 teacher-add 页面：

```
@RequestMapping("/teacherAddPage")
public String teacherAddPage(HttpServletRequest httpServletRequest,Model model){
    return "admin/manage/teacher-add";
}
```

Teacher-add 页面再次通过 action 标签调用 controller 找到处理请求的方法：

```

@RequestMapping("/teacherAddData")
@ResponseBody
public
Map<String,Object>teacherAddData(TTeacher tTeacher){tTeacher.setcTime(new
Date());
tTeacher.setState("1");
adminService.addteacher(tTeacher);
Map<String,Object> result = newHashMap<String,Object>();
result.put("errorCode","1");
return result;}

```

找到处理方法后调用 Service 业务处理层中相应代码进行数据操作，在代码前面加上@Transactional，一旦处理过程中有失败操作则回滚该处理：

```

@Transactional
public void addteacher(TTeacher tTeacher) {
tTeacherMapper.insert(tTeacher);
}
public booleancheckTeacherNum(String stNum) {
intnum = tTeacherMapper.checkTeacherNum(stNum);
return num>0?false:true;
}

```

通过 javabean 实体类获取映射关系，实现映射功能的代码文件所在位置为：src/main/java/com/yzpc/weixin/vo/TTeacher.java，获取关系后调用 DAO 模块进行数据库操作：用 SQL 语句用来插入对应数据。

DAO 数据持久层：src/main/resources/mybatis/TTeacherMapper.xml
以下为数据库操作代码：

```

insert into t_teacher (TEACHER_NUM, TEACHER_NAME, USER_NAME,
USER_PASSWORD, TEACHER_PHONE, STATE,
U_TIME, C_TIME, IS_ADMIN
)
values (#{teacherNum,jdbcType=VARCHAR},
#{teacherName,jdbcType=VARCHAR}, #{userName,jdbcType=VARCHAR},
#{userPassword,jdbcType=VARCHAR},
#{teacherPhone,jdbcType=VARCHAR}, #{state,jdbcType=CHAR},
now(), now(), #{isAdmin,jdbcType=CHAR})

```

5.2.2 修改教师信息

点击图 5.3 中教师列表中某个教师后面编辑按钮(位置 3)，打开显示当前教师信息并进入可编辑状态，如图 5.6 所示：



图 5.6 教师信息编辑界面

本功能入口为教师管理（默认为 teacher-list 页面）中 teacher_edit 功能按钮，后台数据处理及逻辑分析如图 5.7 所示：

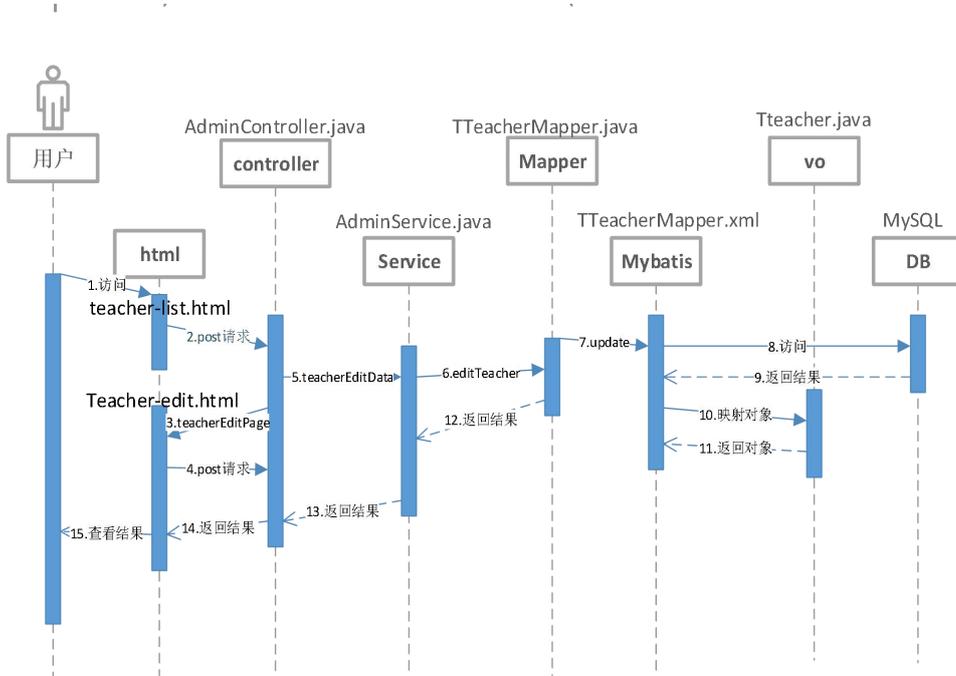


图 5.7 编辑教师流程设计

修改教师信息与新增教师实现方式相似，不同的是入口的功能按钮跳转到 teacher-edit 页面，而 controller 找到的处理方法为 teacherEditDate：

```
@RequestMapping("/teacherEditData")
@ResponseBody
public Map<String,Object>teacherEditData(TTeacher tTeacher ){
    tTeacher.setuTime(newDate());
    adminService.editTeacher(tTeacher);
    Map<String,Object> result = newHashMap<String,Object>();
    result.put("errorCode","1");
    return result;}

```

在操作中将表单数据更新到数据库中，而不是新建一条记录，数据库更新代码如下：

```
update t_teacher
set TEACHER_NAME = #{teacherName,jdbcType=VARCHAR},
    USER_NAME = #{userName,jdbcType=VARCHAR},
    TEACHER_PHONE = #{teacherPhone,jdbcType=VARCHAR},
    U_TIME = now(),
    IS_ADMIN = #{isAdmin,jdbcType=CHAR}
where TEACHER_NUM = #{teacherNum,jdbcType=VARCHAR}
</update>

```

5.2.3 删除教师信息

点击教师列表某教师后的删除按钮，弹出确认提示，效果如图 5.8 所示：



图 5.8 确认删除教师界面

本功能入口为教师管理（默认为 teacher-list 页面）中 teacher_del 功能按钮，后台数据处理及逻辑分析如图 5.9 所示：

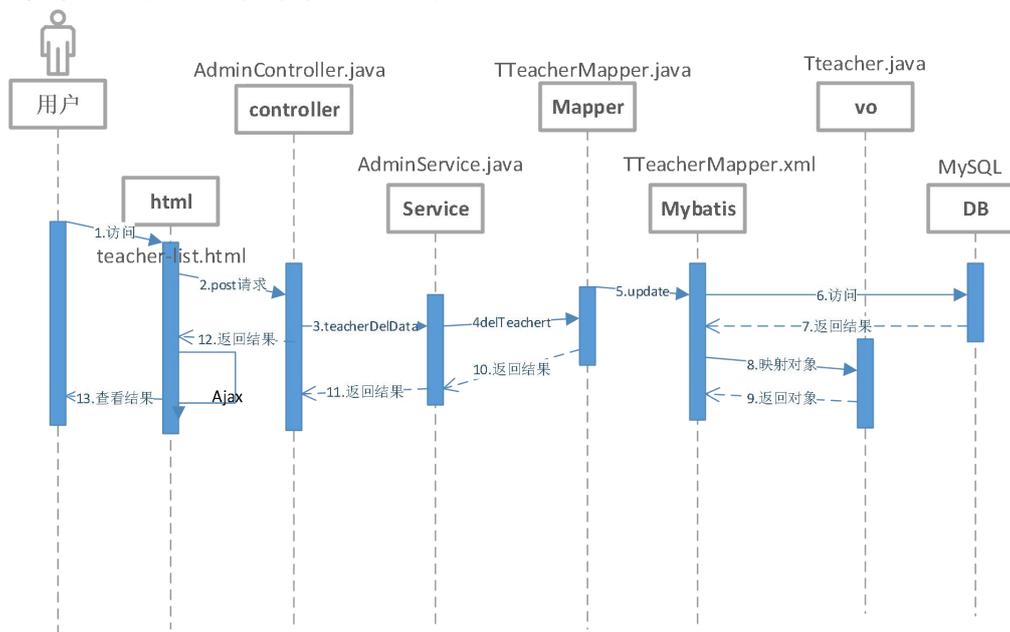


图 5.9 删除教师功能流程设计

该功能与上述其他功能实现方法类似，不同之处为通过功能按钮，直接调用 controller 中处理本请求的方法：

```

@RequestMapping("/teacherDelData")@ResponseBody
public Map<String,Object>teacherDelData(String teacherNum){
    intnum = adminService.delTeachert(teacherNum);
    Map<String,Object> result = newHashMap<String,Object>();
    result.put("errorCode",num);
    return result;}
    
```

由数据表结构设计可知删除教师功能的实现为：将该教师的状态设置为 0，即操作数据库时不使用 delete 语句，而是采用 update 将教师状态更新，核心代码如下：

```

update t_teacher set state='0' ,U_TIME = now()
where TEACHER_NUM = #{teacherNum, jdbcType=VARCHAR}
    
```

5.3 班级管理

班级管理子模块主要包含以下功能：添加班级、修改班级信息、删除班级以及修改班主任信息。该功能需要管理员权限访问，班级信息添加后，学生才可以选择该班级为自己班级，而教师只能从已存在的教师列表中选择，不可任意填写。

管理员完成登录后，点击主界面左侧目录树中的“班级管理”，进入班级管理页面完成班级管理任务。添加班级界面如图 5.10 所示

当选择某一个教师时，网页调用函数通过 ajax 方式提交并获取该教师的联系电话。当管理员选择删除该班级时，只将该班级状态设置为不可用，并非将其从数据库中抹去。

图 5.10 添加班级信息界面

本功能实现与添加教师功能类似，在此不再给出实现代码。在班级添加成功后，通过 ajax 技术重新获取班级信息。核心代码如下：

```

$("#teacherNum").change(function(){
    varteacherName = $(this).find("option:selected").text();
    varteacherPhone = $(this).find("option:selected").attr("id");
    $("#phoneNum").val(teacherPhone);
    $("#teacherName").val(teacherName);
});

```

班级编辑、删除等功能和教师管理子模块操作相似，这里不再赘述。

5.4 学生管理

学生管理功能与教师管理功能类似，该功能操作权限为管理员或教师，学生本人仅能修改个人信息。点击主界面左侧目录树中的“学生管理”菜单，进入学生管理界面，点击目录树中“学生管理”，进入学生管理首页，如图 5.11 所示。在该窗口中完成学生的管理任务。具体可以实现添加学生，点击位置 2 进入；编辑学生信息，点击位置 3 进入；删除学生信息，点击位置 4 进入；查询学生信息，点击位置 5，实现按姓名查找；点击位置 6，实现从既有学生列表中按任意字段进行模糊查找。

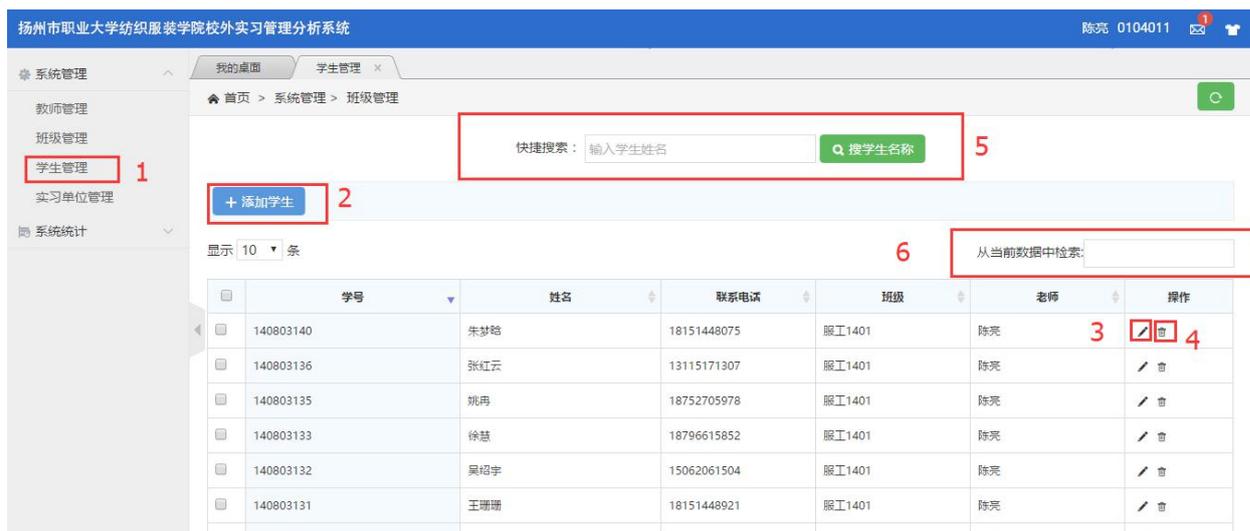


图 5.11 学生管理界面

其中，班级选择只能从数据库中已存在的班级列表中选择，实习单位必须在实习单位表中存在，实习单位可以为空。

选择班级代码如下：

```
<select th:value="{student.classNum}" class="select" name="classNum"
size="1">
<option th:each="c:{cls}" th:value="{c.classNum}"
th:attr="selected={student.classNum == c.classNum}"
th:text="{c.className}+'-'+{c.teacherName}"></option></select>
```

选择实习单位代码如下：

```
<select th:value="{student.classNum}" class="select" name="workNum"
size="1">
<option th:each="c:{work}" th:value="{c.workNum}"
th:selected="{student.workNum == c.workNum}"
th:text="{c.workName}"></option></select>
```

5.5 实习单位管理

实习单位管理包含管理实习单位信息（增、删、改查）和分配学生实习这两个主要功能。

5.5.1 管理实习单位信息

点击主页面目录树中“实习单位管理”，进入实习单位管理首页，如图 5.12 所示。再点击“添加实习单位”（位置 2），打开“添加实习单位”页面，如图 5.13，在其中输入单位名称后，本系统能从地图中查找到该单位的地址，将其自动填入“单位地址”栏，并将其位置在地图中标识出来，用户核实信息后，可点击“提交”按钮，将单位信息提交至服务器中。



图 5.12 实习单位管理界面

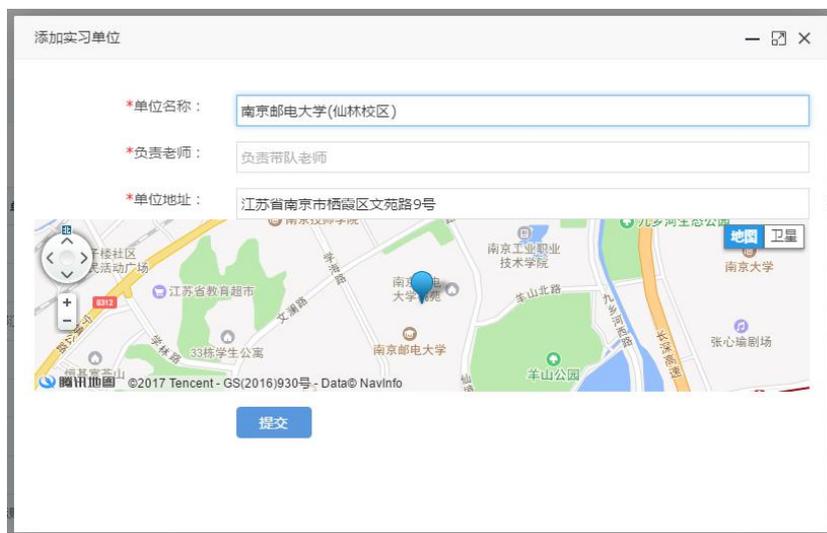


图 5.13 实习单位信息添加界面

这里以添加单位功能进行分析，业务流程如图 5.14 所示：

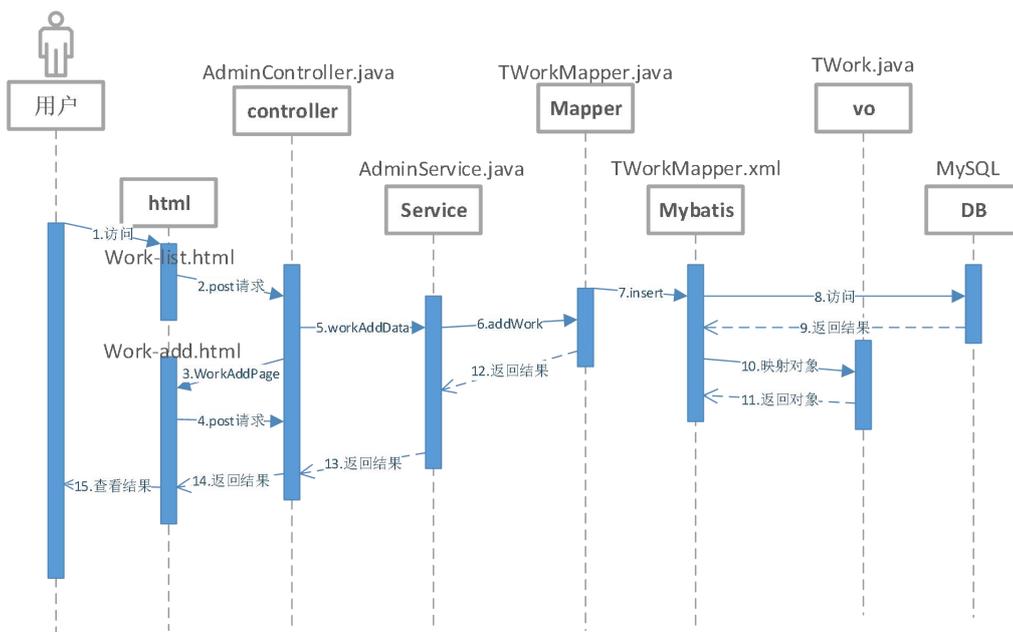


图 5.14 添加单位功能流程设计

添加单位时，系统实时通过用户输入的地址来获取位置信息，此处调用腾讯 LBS 服务支持，调用服务代码见 LBS 地图简介。获取位置信息核心代码如下：

```

citylocation = new qq.maps.CityService({
complete : function(results){
map.setCenter(results.detail.latLng);
    if (marker != null) {
marker.setMap(null);
    }
    //设置 marker 标记
    marker = new qq.maps.Marker({
        map: map,
        position: results.detail.latLng
    });
    }
});//调用城市服务信息
    
```

```
function geolocation_latlng() {
    //获取经纬度信息
    varlat = $("#checkinLatitude").val();
    varlng = $("#checkinLognitude").val();
    //设置经纬度信息
    varlatLng = new qq.maps.LatLng(lat, lng);
    //调用城市经纬度查询接口实现经纬查询
    citylocation.searchCityByLatLng(latLng);
}
```

5.5.2 学生实习单位分配

教师或管理员可以为某单位分配实习生，教师只可操作本班学生，管理员可操作所有学生，可选择整班批量分配单位、为某班未实习学生分配单位或为某位学生单独分配单位。单击某个单位后面的添加实习生按钮进入对该单位分配实习生的页面，分配界面如图 5.15 所示：

黎明村-实习生维护

服工1401 服艺1402 服艺1403 服艺1401 服展1401 输入学生姓名直接添加

显示 10 条 从当前数据中检索:

学号	姓名	联系电话	班级	老师	操作
140801208	龚丽	13013713591	服艺1402	纪杰	🗑️
140801211	黄玉霞	18752705145	服艺1402	纪杰	🗑️
140801109	高鑫玥	15720760761	服艺1401	丁苏宁	🗑️
140801106	陈晓霞	17352460013	服艺1401	丁苏宁	🗑️
140801203	陈思彤	18752717410	服艺1402	纪杰	🗑️
140801104	陈嘉欣	15052519516	服艺1401	丁苏宁	🗑️
140801204	陈叶	15052526937	服艺1402	纪杰	🗑️
140801245	邹宇	15952725634	服艺1402	纪杰	🗑️

图 5.15 分配学生实习

通过各实习单位后的添加学生按钮进入分配学生实习功能,同时将单位的编号和名字记录传递,按钮代码如下:

```
function worker_edit(id,name)
{
layer_show(name+"-实习生维护",
'/yzpc/workersPage'+"?workNum="+id+"&workName="+name,",510);
}
```

根据班级学生实习情况给出提示,若该班级均未参加或均以参加实习,可以将整个班级添加到实习单位,界面如图 5.16 所示:



图 5.16 整班级调整实习单位界面

若该班级部分学生为参加实习,则给出不同选择,详细界面如图 5.17 所示:



图 5.17 修改班级部分学生实习单位界面

按班级分配学生实习直接获取该班级的编号:

```
@RequestMapping("/classjoinWork")
@ResponseBody
public
Map<String,Object>classjoinWork(@RequestParam("classNum")String
classNum,@RequestParam("workNum")String
workNum,@RequestParam("type")String type){
adminService.joinWork(classNum,workNum,type);
Map<String,Object> result = newHashMap<String,Object>();
result.put("errorCode","1");
return result;
}
```

Service 层对应方法分别实现两种分配方式:

```
@Transactional
public void joinWork(String classNum, String workNum, String type) {
    if(StringUtils.equals(type, "1"))
        tStudentMapper.joinWork1(classNum,workNum);
    else
        tStudentMapper.joinWork2(classNum,workNum); }
public void delWorker(String xh) {
    tStudentMapper.delWorker(xh);}
public void joinWork(String xh, String workNum) {
    tStudentMapper.studentJoinWork(xh,workNum);}
```

两种方法均是将学生表中的实习单位进行更新，所有都是使用 update 语句，joinWork1 和 joinWork2 为按班级分配，studentJoinWork 为单个学生分配，分配学生实习单位的数据库实现语句如下：

```
<update id="joinWork1">
    update t_student set work_num=#{workNum,jdbcType=VARCHAR},
    u_time=now() where state='1'and class_num=
    #{classNum,jdbcType=VARCHAR}
    and ISNULL(WORK_NUM)
</update>
<update id="joinWork2">
    update t_student set work_num=#{workNum,jdbcType=VARCHAR},
    u_time=now() where state='1'and class_num=
    #{classNum,jdbcType=VARCHAR}
</update>
<update id="studentJoinWork">
    update t_student set work_num=#{workNum,jdbcType=VARCHAR} ,
    u_time=now() where xh= #{xh,jdbcType=VARCHAR}
</update>
```

也可以根据学生姓名单独对某一个学生进行实习单位的分配，通过搜索该学生姓名直接添加到实习单位，操作界面如图 5.18 所示：



图 5.18 修改某个学生实习单位界面

以个人形式分配学生实习通过学生姓名获取该学生的学号并将其与实习单位的编号进行绑定，以此来达到功能需求，实现代码如下：

```

@RequestMapping("/studentjoinWork")
@ResponseBody
public
Map<String,Object>studentjoinWork(@RequestParam("xh")String
xh,@RequestParam("workNum")String workNum){
    adminService.joinWork(xh,workNum);
    Map<String,Object> result = new HashMap<String,Object>();
    result.put("errorCode","1");
    return result;
}

```

6 系统测试

系统测试是开发过程中一个重要的阶段，是在系统正式投入使用前，对系统的需求分析、设计和编码等阶段产品的最终检查，能够尽可能的检测系统运行过程中的问题，避免运行过程中出现错误的机率，对运行中可能出现的问题早发现早解决，所以系统测试是开发过程中不可或缺的一部分。

6.1 系统测试方法

此系统因按照设计人员对需求和操作的理解进行设计，在运行和操作上可能存在许多问题，这些局限性期望通过一些测试进行弥补。

测试的主要内容：

正确性：操作结果及提示的正确性；

可用性：是否可以满足设计需求；

安全性：输入边界值、特殊值等进行测试；

性能测试：正常使用时完成某任务的时间，多人同时访问的响应时间等是否在可接受范围内。

本实习管理分析系统针对以上内容测试，每次测试对运行中存在的问题进行了修缮，最终使其性能达到了令人较为满意的程度。

6.2 测试内容

此次系统测试主要包括Web应用系统的功能完善问题、用户登录的安全问题等。具体测试包含一下几点：用户登录窗口测试、信息添加测试、用户权限判断测试。

1、用户登录窗口测试

为了保证登录的正确性与容错、纠错性，首先进行登录测试。

具体实现过程为分别使用正确的用户名密码、错误的用户名密码、含有特殊字符的用户名密码。要求系统在遇到不同情况给出正确的提示。

2、信息添加测试

为了保证信息的准确性，页面处理数据信息的正确性及数据库接收保存信息的完整性，我们进行信息添加测试。

具体实现为在不同的功能页面进行信息添加，查看添加提示信息，并通过数据库直接查看和页面调取方式查看测试的数据。

3. 用户权限判断测试

为了保证信息的安全性，系统对用户权限的判断十分重要。不同用户的操作权限不同，只有正确识别用户权限，才能判断用户可进行的操作。

具体测试方法为使用两个用户名密码相似但是权限不同的账户登录，测试所能操作的权限。

6.3 测试结果

经过测试，系统功能符合初步设想，操作较为灵活简洁。数据库与前台交互正常，系统响应速度较为理想，支持一定的并发量。能够满足普通的管理工作需要，并且在安全性、稳定性上基本满足要求。测试结果表 6.1 所示。

表 6.1 测试结果

测试项目名称	输入	提示	结果
登录	正确信息	登录正确	跳转用户页面
	错误信息	登录错误	保持当前页面
	常用 SQL 注入代码	登录错误	保持当前页面
信息添加	添加学生信息	添加成功	后台保存正常
	查询学生信息		返回查询信息
用户权限	管理员账户	登录成功	跳转管理员页面
	教师账户	登录成功	跳转教师页面

7 总结

在毕业生实习期间，总是存在无法对学生校外实习情况及时、有效的管理和分析。得益于移动互联的发展，智能手机集成的GPS定位功能可以帮助我们准确知晓学生的位置信息，同时，日益进步的数据分析和展示功能为我们统计分析数据提供了帮助。

大半年来，整个团队在指导教师的带领下，从整理需求、技术选型、完成设计，到最后撰写论文，克服了种种困难与难题，在大家的共同的努力下《基于微信公众平台的高校校外实习管理分析系统》开发完成并上线投入使用。

通过该系统的设计开发，提升了团队成员课堂知识的灵活运用能力，同时自学掌握了微信公众号接口开发知识，提升了专业技术能力，积累了项目开发实战经验。

当然该系统还有不足之处，有许多功能有待完善。期待在系统逐步完善的基础上，能在全校甚至其他院校推广使用。

参考文献

- [1]罗军舟.《TCP/IP协议及网络编程技术》[M].北京:清华大学出版社,2008:101~131.
- [2]廖勇、陈亚辉、李新锋、高立辉等.《JSP网络开发》[M].北京:清华大学出版社,2010:109~129.
- [3]张银鹤、刘治国、张豪等.《JSP动态网络开发》[M].北京:清华大学出版社,2007:29~56.
- [4]李刚编著.《Java 疯狂讲义》电子工业出版社[M] 2014.7
- [5]林信良. Spring2.0技术手册[M].北京:电子工业出版社,2015
- [6]张孝祥.深入Java Web开发内幕——核心基础[M].北京:电子工业出版社.2010
- [7]田珂,谢世波,方马. J2EE数据持久层的解决方案[J].计算机工程,2013,29(22):93-95
- [8]福达.《SQL必知必会》[M].北京:人民邮电出版社.2013
- [9]AndyLizh. SSM框架——详细整合教程(Spring+SpringMVC+MyBatis)[EB/OL].
<http://blog.csdn.net/gebitan505/article/details/44455235/>, 2015-03-19
- [10]唐小新.spring MVC 原理 [EB/OL].
http://blog.csdn.net/xtu_xiaoxin/article/details/8796499, 2013-04-13
- [11]李洋. SSM框架在Web应用开发中的设计与实现[J].计算机技术与发展,2016,26(12):190-194
- [12]shu_lin. SSM框架——详细整合教程(Spring+SpringMVC+MyBatis)[EB/OL].
<http://blog.csdn.net/zhshulin/article/details/37956105/>, 2014-07-19
- [13]茶饮月. Spring Boot学习记录(二)--thymeleaf模板[EB/OL].

致 谢

首先要感谢的是我们的指导老师对我们的耐心指导和循循善诱,她认真的教学态度,严谨的治学精神以及精益求精的工作作风,都深深地感染和激励着我们。从课题的选择确定到项目开发的最终完成,周老师都始终给予我们细心的指导和不懈的支持,在此谨向周欢老师致以诚挚的感谢和崇高的敬意。

同时还要感谢跟我同一个团队的两位同学。遇到问题时共同探讨、共同进步,存在疑问时耐心解答,提出想法时共同思考……这些都让我体会到了团队的强大力量,正是因为有这种力量支持,我才能克服一个个困难与疑惑,直至本文的顺利完成。

最后感谢所有关心和帮助过我的老师和同学,如果没有他们的帮助,这个系统的开发也就不会那么顺利。