



湖南工程學院

# 毕 业 设 计

题 目： 实验室管理平台的设计

学院： 计算机与通信学院

专业： 计算机科学与技术 班级： 1501 学号： 201503010111

学生姓名： 王元天

导师姓名： 刘学爽

完成日期： 2019 年 6 月 7 日

# 诚 信 声 明

本人声明：

1、本人所呈交的毕业设计（论文）是在老师指导下进行的研究工作及取得的研究成果；

2、据查证，除了文中特别加以标注和致谢的地方外，毕业设计（论文）中不包含其他人已经公开发表过的研究成果，也不包含为获得其他教育机构的学位而使用过的材料；

3、我承诺，本人提交的毕业设计（论文）中的所有内容均真实、可信。

作者签名：

日期：      年    月    日

# 湖南工程學院

## 毕业设计（论文）任务书



题目：\_\_\_\_\_实验室管理平台的设计\_\_\_\_\_

姓名 王元天 学院 计算机与通信学院 专业 计算机科学与技术 班级 1501 学号 201503010111

指导老师 刘学爽 职称 讲师 教研室主任 张铁楠

### 一、基本任务及要求：

1、此课题是实验室管理平台的设计，是基于 Web 的动态交互管理系统的开发，要求能够指定目标如下：

①灵活的人机交互，操作简单方便，界面简介美观。②登录管理，提供多角色登录功能。③实验室各类基本信息管理，提供各类信息基本增删改查管理功能。④实现各种查询，如多条件查询，模糊查询等。⑤提供实验室预约功能。⑥提供实验报告审核批阅，查询成绩的功能。⑦系统应当运行稳定，安全可靠。

2、主要任务：

① 实现用户登录模块：包括学生登录，教师登录，管理员登录

② 实现基本信息管理模块：包括实验室、教师、学生、课程、实验室预约、班级、实验报告等信息增删改查管理

③ 实现实验室预约模块：教师预约实验室，管理员决定是否同意预约，预约成功后，教师、学生均能查看到已预约的实验室安排。

④ 实现实验报告批阅模块：学生完成实验后，撰写实验报告，提交完成后，教师可以查看已完成的实验报告，并给予课堂成绩和报告成绩。

### 二、进度安排及完成时间：

第 1 周 老师集中指导，分析并明确课题任务与要求，学习资料收集检索方法，并搜索收集所需中英文资料。

第 2~3 周 阅读资料、书籍，学习所需知识，撰写文献综述、开题报告。

第 4~5 周 毕业实习、完成毕业实习报告。

第 6 周 建立毕业设计实验环境；初步拟订设计方案。

第 7 周 完成总体设计。

第 8~13 周 具体设计、调试、修改、实现。

---

第 14~15 周 撰写毕业论文（说明书）。

---

第 16 周 完成毕业答辩资格审查、毕业答辩准备

---

第 17 周 毕业答辩

---

# 目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 前言.....	1
1.1 研究背景.....	1
1.2 国内外发展现状.....	1
1.3 本文主体结构.....	2
第 2 章 技术概述.....	3
2.1 前端相关技术.....	3
2.1.1 BootStrap 框架.....	3
2.1.2 Thymeleaf 技术.....	3
2.2 后端相关技术.....	3
2.2.1 SpringBoot 框架.....	3
2.2.2 SpringMVC 框架.....	4
2.2.3 MyBatis 框架.....	4
2.2.4 三层架构.....	4
2.3 数据库相关技术.....	5
2.3.1 ApsaraDB.....	5
2.3.2 RDS MySQL 版.....	5
2.3.3 Druid 连接池.....	5
2.4 本章小结.....	6
第 3 章 需求分析.....	7
3.1 系统角色用例分析.....	7
3.1.1 管理员用例分析.....	7
3.1.2 教师用例分析.....	7
3.1.3 学生用例分析.....	7
3.2 系统业务流程图分析.....	8
3.2.1 实验室超级管理员业务流程图分析.....	8
3.2.2 教师业务流程图分析.....	9
3.2.3 学生业务流程图分析.....	9
3.3 系统的可行性综合分析.....	11
3.3.1 经济的可行性综合分析.....	11

3.3.2 操作的可行性综合分析.....	11
3.3.3 技术的可行性综合分析.....	11
3.4 本章小结.....	12
第4章 系统概要设计.....	13
4.1 系统整体的概要设计.....	13
4.1.1 系统整体功能结构的概要设计.....	13
4.1.2 系统整体软件架构的概要设计.....	14
4.2 系统功能模块设计.....	15
4.2.1 登录功能模块.....	15
4.2.2 信息管理功能模块.....	15
4.2.3 实验室预约功能模块.....	16
4.2.4 报告批阅功能模块.....	17
4.3 核心程序业务的概要设计.....	17
4.3.1 实验室预约程序设计.....	17
4.3.2 报告批阅程序设计.....	18
4.4 数据库的概要设计.....	19
4.4.1 数据库的概念模型的概要设计.....	19
4.4.2 数据库的逻辑结构的概要设计.....	21
4.4.3 数据库的物理结构的概要设计.....	21
4.5 本章小结.....	27
第5章 系统的总体设计与实现.....	28
5.1 实现环境.....	28
5.1.1 开发环境.....	28
5.1.2 发布环境.....	28
5.1.3 系统实现部署方式.....	28
5.2 系统项目搭建与配置.....	29
5.2.1 SpringBoot 主启动类.....	30
5.2.2 Gradle 项目依赖管理器配置.....	30
5.2.3 系统项目基础配置.....	31
5.2.4 系统项目额外配置.....	34
5.2.5 系统项目全局配置.....	37
5.3 子系统设计与实现.....	38

5.3.1 登录子系统设计与实现.....	38
5.3.2 信息管理子系统设计与实现.....	40
5.3.3 实验室预约子系统设计与实现.....	45
5.3.4 报告批阅子系统设计与实现.....	45
5.4 本章小结.....	47
第 6 章 系统的功能模块测试.....	48
6.1 系统功能模块测试的环境.....	48
6.2 系统功能模块测试.....	48
6.2.1 登录功能测试.....	48
6.2.2 实验室预约功能测试.....	49
6.2.3 报告批阅功能测试.....	50
6.3 本章小结.....	52
总结.....	53
参考文献.....	54
致谢.....	55





## 实验室管理平台的设计

**摘 要：**今天已是信息极速发展的时代，信息化的教育已经被国家看作是重点规划的工程。在信息极速发展的时代背景下，以往的实验室管理方式对比于当今信息极速发展的趋势，显然产生了鲜明的比对。显然，以往的实验室管理方式应当与时俱进，从线下逐步转变至线上。

实验室管理平台是当今信息极速发展下的线上新管理模式和计算机相关技术在实验室管理中的应用。以高等院校实验室为研究核心，采取线上新管理模式理念，以及运用第三方阿里云服务器、数据库等高可靠技术和设备，实现信息统一管理。通过对实验室信息进行全方位统一管理，以如何才能更可靠、更高效、更便捷的建立一个较完善的实验室管理平台为首要目标。严格掌控业务流程，时刻注重业务操作细节，以达到满足实验室线上管理的日常需求。

根据方案设计，采用 MVC（实体-视图-控制）的页面设计模式，以及三层架构（表示层-业务逻辑层-数据访问层）的开发模式，使用阿里云服务器 ECS，阿里云数据库 RDS MySQL 版，通过使用 Java 语言，SpringBoot 框架等快速实现实验室管理平台。实验室管理平台实现了登录子系统，信息管理子系统，实验室预约子系统以及实验报告批阅子系统。以线上的平台模式提高了线下管理员，教师与学生对于预约实验室，批阅实验报告工作效率。

在开发实验室管理平台完成后，通过对多个重要功能模块的测试，验证了这几个重要功能模块都可以完成预期需求目标。现在，此系统已发布在阿里云服务器 ECS 上，开始运行，并且时刻保持良好的运行状态。

**关键词：**实验室管理；三层架构；MVC 模式

# Design of laboratory management platform

**ABSTRACT:**Today is era of rapid development of information. Informatization education has been regarded by the state as a key project. In the context of the rapid development of information, the past laboratory management methods have clearly contrasted with the current trend of rapid development of information. Obviously, the past laboratory management methods should keep pace with the times and gradually change from offline to online.

The laboratory management platform is the application of online new management mode and computer related technology in laboratory management under the rapid development of information. Taking the laboratory of colleges and universities as the core of research, adopting the concept of online new management mode, and using high-reliability technologies and equipment such as third-party Alibaba Cloud server and database to realize unified information management. Through the unified management of laboratory information, it is the primary goal of how to build a more complete laboratory management platform more reliably, more efficiently and more conveniently. Strictly control the business process, always pay attention to the details of business operations, in order to meet the daily needs of the online management of the laboratory.

According to the scheme design, adopt MVC (solid-view-control) page design mode, and the development mode of three-tier architecture (representation layer-business logic layer-data access layer), use Alibaba Cloud server ECS, Alibaba Cloud database RDS MySQL Version, through the use of the Java language, SpringBoot framework and other rapid implementation of the laboratory management platform. The laboratory management platform implements the login subsystem, the information management subsystem, the laboratory reservation subsystem, and the experimental report review subsystem. The online platform model has improved the efficiency of offline administrators, teachers and students for booking labs and reviewing lab reports.

After the completion of the development of the laboratory management platform, through the testing of a number of important functional modules, it is verified that these important functional modules can achieve the expected demand goals. Now, this system has been released on the Alibaba Cloud Server ECS, it is running, and it is always in good running state.

**KeyWords:**Laboratory management;Three-tier architecture;MVC mode

# 第 1 章 前言

## 1.1 研究背景

随着计算机科学与技术的急速发展，关于对实验室管理的手段也有了各种程度的改进，传统的人工手动线下管理模式，在当今这个时代下已显得有些落伍。使用线上高可用、高效的实验室管理平台进行数据收集和数据处理是现代自动化线上管理模式逐步的悄然破壳而生。而对于高等院校实验室管理工作来说，迫切需要设计并实现一套能够初步实现网络化办公、符合数字化校园环境的现代综合性实验室管理平台。以便于从根本上提升整个高等院校实验室管理工作的效率。

另一方面，伴随着高等院校科研能力的不断提高，高等院校内部实验室建设的整体水平已经成为了衡量其办学水平的重要指标。现如今，高等院校实验室管理建设与其整体教学质量紧密相连，其程度可直接影响实验、实践教学环节的教学质量和教学效果。因此，不断规范、持续完善实验室管理工作也会在一定程度上对高等院校的科研水平起到推动作用。

## 1.2 国内外发展现状

国外，在对实验室管理平台的研发和应用上的技术相对比较成熟，其高等院校实验室管理平台建设普遍早于国内并且应用范围非常广泛。英国实验室管理平台的基本特点是网络化管理功能相当强大，信息化数据资源相当丰富。学生能够比较便捷地寻找到所需要的实验资源并且提供相应的在线预约功能。总体来说，对网络依赖性很强，资源开发程序相对较高。

国内，长期以来，对于数据资源的管理还并不能与数据资源的增长进行同步。我国虽然拥有的原始数据资源十分丰富，但对于这些数据资源很难再次进行使用，相当多的具有一定价值的数据资源不能进一步得到它的价值。但近年来，我国各高等院校对实验室管理平台的研究，也逐渐的得到很好的发展。国内的发展大致可以分为两个时段：第一个时段大约是从 80 年到 90 年，以单机连接和独立的功能系统为主体构建的一个简易的实验室管理平台。该时段主要处理一些简单的日常管理工作，数据一般都是以独立存在，独立用的方式，并不能很好的达成信息的交互；第二个时段是从 90 年至今，是以互联网和外部存储工具（如数据库）为基础的实验室管理平台。国内有多家大型的软件公司已经开发出较为成熟的商业化的产品，使得高等院校的实验室管理变得高效。

## 1.3 本文主体结构

本文主体结构安排如下：

### 第 1 章 前言

介绍了研究并设计实验室管理平台的必要性，同时也简单阐述了实验室管理平台在国内外研究的发展现状。

### 第 2 章 技术概述

介绍了实现设计实验室管理平台所需要的相关技术框架。

### 第 3 章 需求分析

讨论实验室管理平台实现所需要的用例，相关业务流程图，以及相关可行性分析。并讲讨论结果以一份完整的需求分析报告呈现出来。

### 第 4 章 系统概要设计

讨论实验室管理平台总体功能结构以及总体软件架构，以及各个系统的功能和系统的数据库的简要设计内容。

### 第 5 章 系统的总体设计与实现

讨论系统环境的各种详细配置，各个子系统详细业务流程实现，最后将结果以部分界面图呈现出来。

### 第 6 章 系统的功能模块测试

介绍了系统多个重要功能模块的测试内容，测试的预期结果以及测试的实际结果。

## 第 2 章 技术概述

### 2.1 前端相关技术

本系统前端，采用页面以 Html5+Thymeleaf 技术，样式以 CSS3+Bootstrap 框架，同时使用 JQuery 库设计实现并加以渲染。

#### 2.1.1 Bootstrap 框架

Bootstrap 原名 Twitter Blueprint，由 Twitter 公司使用 Html，CSS 和 JavaScript 进行开发的开源工具包，众多的已实现基本组件和通过 JQuery 原生实现的功能强大的额外插件，用来快速实现你想要页面渲染。目前，Bootstrap 是世界上使用最多的前端轻型框架。

Bootstrap 在使用上十分简易，但在内容，组件以插件上，包含了大量你能够想到的页面组件，非常适合页面渲染的快速设计与实现。Bootstrap 主要是用来为后端开发者专门定制，后端开发者无需设计各种页面渲染的设计与实现，只需面向页面本身，从 Bootstrap 组件库中找到所需要的组件，即可快速得到所需要的页面渲染效果。

#### 2.1.2 Thymeleaf 技术

Thymeleaf 是一个适用在 Web 开发应用程序基于 Java 语言的模板引擎。它可以在具有 MVC（Model-View-Control）设计模式系统的视图层提供 HTML5 页面渲染。

通过 Spring Framework 模块，与开发者自定义组件，以及高可复用，可兼容，可插入式。在 Web 应用程序中，Thymeleaf 旨在成为 JavaServer Pages（JSP）的完全替代品，并实现自然模板的概念：模板文件可以直接在浏览器中打开，并且仍然可以正确显示为网页。Thymeleaf 是现代 HTML5 JVM Web 开发的理想选择。

### 2.2 后端相关技术

本系统后端，采用 SpringMVC+SpringBoot+Mybatis 组合式的，三层架构开发模式进行功能模块以及业务逻辑上的设计与开发。

#### 2.2.1 SpringBoot 框架

SpringBoot 是一款开源的框架，简化 Spring 应用初始基础配置是这款框架的核心优势。该框架特点就是可以进行自动配置模式，以及开放了开发着自定义配置借口，使开

发人员更快，更自由的开启一个系统项目。由此可见，SpringBoot 框架主要应用在快速开发项目。

SpringBoot 只需要通过配置少量的 Spring 配置，能够快速建立一个可运行的，独立的 Spring 应用程序。同时还可以建立通过 `java -jar` 指令直接启动的 Java 应用程序。

SpringBoot 的特点包括：（1）为 Spring 应用程序开发者和 Spring 初学者更快、更好、更自由的入门体验。（2）开箱即用，SpringBoot 提供了很多自动配置来简化开发者配置 xml 过程，让开发者迅速摆脱配置的困境。（3）提供大型项目通用的众多插件与功能。例如嵌入的 Tomcat，使开发者再也无需部署 war 文件。（4）绝对没有冗余代码生成和 XML 配置的要求。

### 2.2.2 SpringMVC 框架

SpringMVC 框架是 SpringFrameWork 中的一款技术产品。SpringMVC 框架是为所有 Spring 应用程序开发者提供的一个功能全面，性价比功能高，能力更强的 MVC 模块。使用 SpringMVC 框架，能够更好的与 Spring 应用程序进行融合，较比之前的 Struts1 和 Struts2 等其他 MVC 框架显示出了更可靠，更高效，高可兼容的优势。

### 2.2.3 MyBatis 框架

MyBatis 在现阶段于市面上是一款相对优秀的用于持久层的框架，它能够支持开发者自行书写 SQL 语句，进行数据表与 Model 实体的高级映射。MyBatis 几乎不需要手写 JDBC（Java Database Connection）代码，设置数据库连接参数的代码以及获取到的结果集的代码。MyBatis 可以通过几个 XML 文件或使用注解（Annotations）的形式进行配置和执行 SQL 脚本语言，完成相应业务作业到数据库表。

MyBatis 主要特点：（1）SQL 语句与代码分离，使开发者、运维人员便于设计实现与维护管理。（2）可用标签语言实现动态 SQL 的拼接。（3）查询 SQL 语句的结果集可以进行与 Model 实体中的属性自动映射和注入。（4）编写原生 SQL，接近 JDBC，是操作变得比较灵活。

### 2.2.4 三层架构

三层架构类似于一个中间件，处于用户端与数据库之间。三层架构从上至下包含：表示层（Controller），业务逻辑层（Service），数据访问层（Mapper）。

（1）表示层：主要为 Html 的页面渲染，一般采用 MVC 模式进行表示。本系统中使用 SpringMVC。



(2) 业务逻辑层：主要是对复杂的业务流程进行相应的操作，也可以是简单的直接的对数据层进行操作。

(3) 数据访问层：主要是对数据库进行操作层，其作用主要是为业务逻辑层提供数据交互。数据访问层主要框架包括 Hibernate，MyBatis。底层原生技术则是 JDBC。本系统中使用 MyBatis 框架。

## 2.3 数据库相关技术

本系统数据库，采用阿里云数据库 ApsaraDB 产品服务下的 RDS（ApsaraDB For Relational Database Server）MySQL 版，以及使用阿里巴巴 Druid 连接池。

### 2.3.1 ApsaraDB

阿里云数据库 ApsaraDB 是一款具高可靠、高稳定、高效、高可兼容、高可伸缩于一体的云上数据库产品服务。它能够轻松运维如今市面上大部分流行的关系型的数据库（包括 Mysql，SQL Server）以及非关系型的数据库（包括 Oracle、Redis、MongoDB 等），同时可以将这些主流开源数据库的性能提高至 6 倍及以上，并为使用者提供百 TB 级数据实时计算能力等，更可以在重大事故或灾难发生时提供全面的数据备份、数据恢复以及数据迁移等全套的解决处理问题的方案。

### 2.3.2 RDS MySQL 版

关系型数据库（Relational Database）是通过依赖实际模型关系而组建的数据库，即是通过二维表建立数据之间的关系。关系型数据库拥有 20 多年的技术历程，技术相当成熟。而 MySQL 是目前全球在关系型的数据库中最受欢迎的一款，也是使用最多的开源的数据库之一。因此本系统选择使用 RDS MySQL 版。

### 2.3.3 Druid 连接池

Druid 不仅一个组件库，用于连接数据库的作用，Druid 还包含了 DruidDataSource 数据库连接池等其他组件。Druid 是目前市面上最常用，效果最佳，最好的连接池。

DruidDataSource 数据库连接池是阿里巴巴设计开发的一项技术。其主要功能包括数据源信息监测，SQL 语句监控，SQL 语句防火墙，URI 访问监控，WEB 应用实时监控，Session 实时监控。DruidDataSource 连接池在性能、稳定性、扩展、监控以及安全上远远超过 DBCP、C3P0 等常用连接池。

## 2.4 本章小结

本章介绍了实验室管理平台在设计和开发中用到的基础理论和关键技术。对从前端开始到后端到数据库等技术栈做了简要的阐释，为实验室管理平台的设计和实现准备好充足的技术前景。



## 第3章 需求分析

### 3.1 系统角色用例分析

#### 3.1.1 管理员用例分析

管理员主要负责实验室日常预约业务处理，具备实验室开放、授权等相关业务处理权限，具有对教师、学生信息的管理，能够进行对课程安排，以及查看所有报告和其他相关业务等的操作权限。具体用例分析如图 3.1 所示。

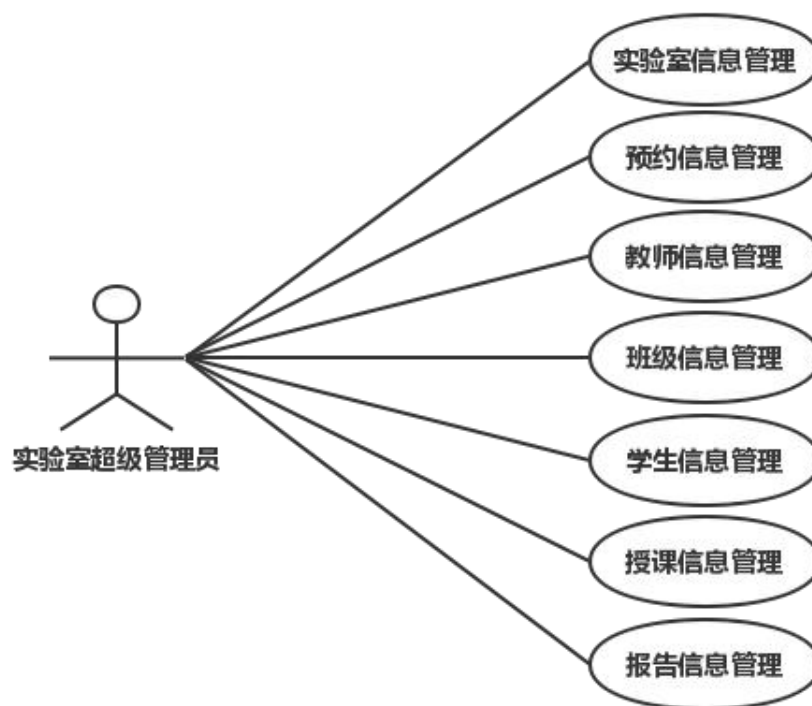


图 3.1 实验室超级管理员用例图

#### 3.1.2 教师用例分析

教师主要负责实验室预约，完成日常报告批阅等相关业务处理权限，同时可以对个人信息做相应的操作，查看管理自己由实验室超级管理员安排的课程，查看所有自己预约的实验室状态（已预约 / 预约中 / 预约失败）。具体用例分析如图 3.2 所示。

#### 3.1.3 学生用例分析

学生主要具有完成报告撰写相关业务处理权限，同时可以对个人信息做相应的操作，查看所有由教师安排的（已预约）实验记录，查看报告完成状态（报告已完成待批阅 / 已批阅 / 报告暂未完成）。具体用例分析如图 3.3 所示。

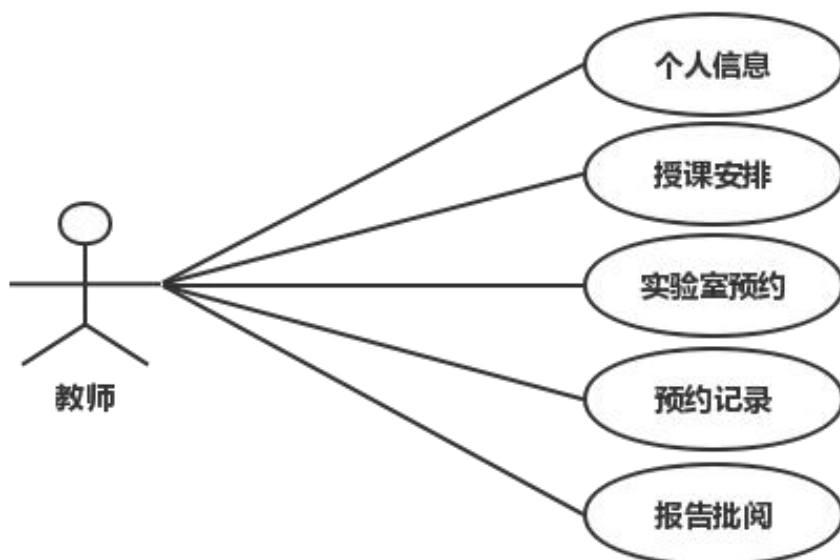


图 3.2 教师用例图

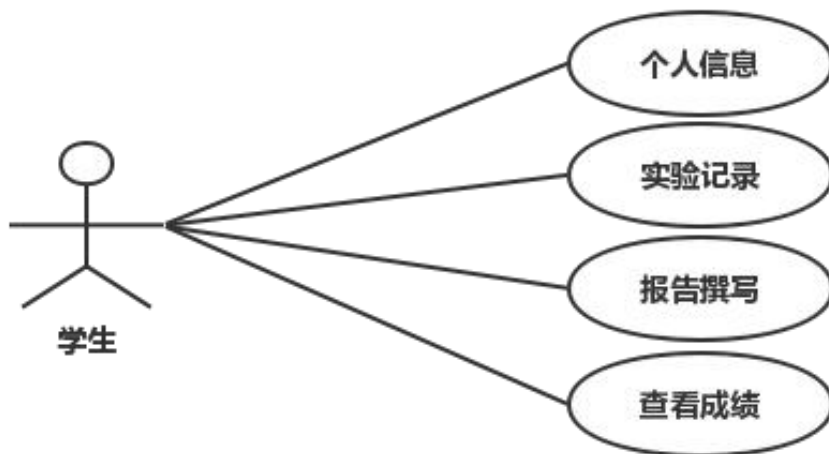


图 3.3 学生用例图

## 3.2 系统业务流程图分析

### 3.2.1 实验室超级管理员业务流程图分析

当管理员成功登录本系统后，可以实现对本系统基础信息（包括实验室信息、教师信息、学生信息、班级信息、授课信息等）的增删改查，高级模糊查询；对预约信息的查询，查看预约失败原因，添加可预约实验室记录，对暂未预约成功的实验室记录进行修改、删除；对学生撰写的报告进行查询，对报告已完成待批阅、已批阅、报告暂未完成三种状态下的报告进行查看等操作；最后可以退出系统。具体业务流程如图 3.4 所示。

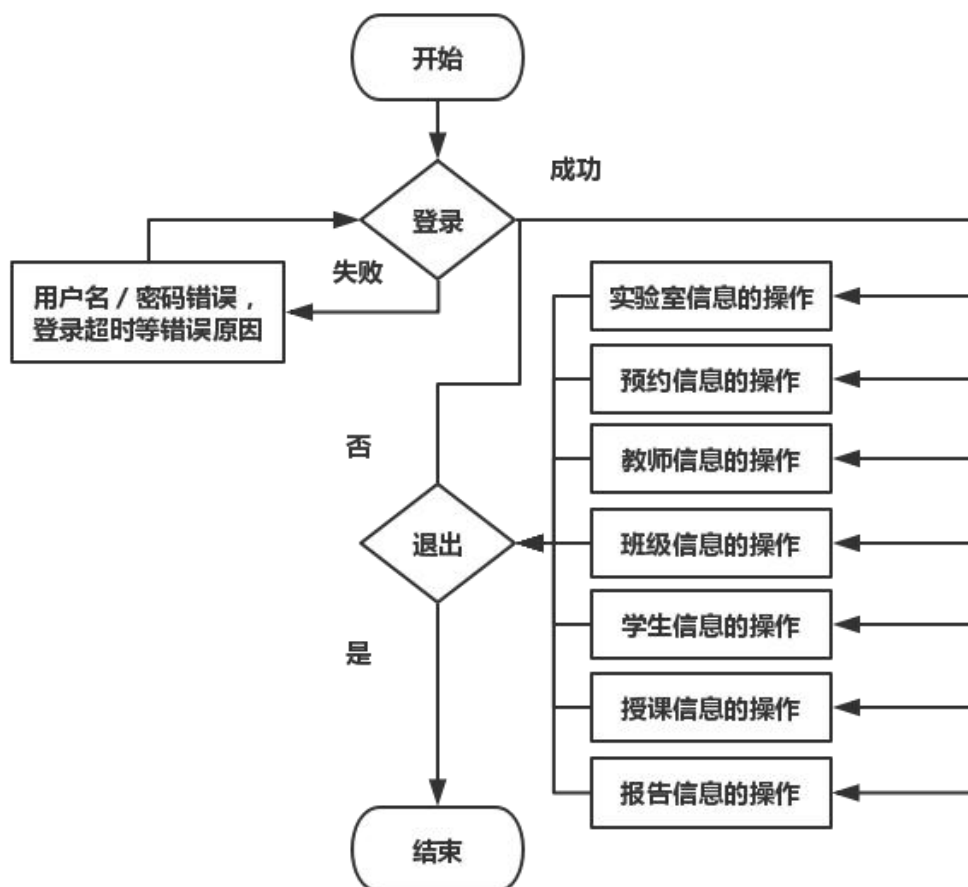


图 3.4 实验室超级管理员业务流程图

### 3.2.2 教师业务流程图分析

当教师成功登录本系统后，可以实现对个人信息（如：密码、姓名、联系方式、邮箱等）进行修改；对授课安排进行查询；对实验室超级管理员发布的可预约实验室进行查询、预约，对已预约的记录进行查看（实验室超级管理员是否同意预约），对未预约成功的实验室记录查看失败信息；对学生报告进行查看、批阅、打分；最后可以退出系统。具体业务流程如图 3.5 所示。

### 3.2.3 学生业务流程图分析

当学生成功登录本系统后，可以实现对个人信息（如：密码、姓名、联系方式、邮箱等）进行修改；可以查看所有教师已预约的实验室；可以查看所有报告完成状态（报告已完成待批阅 / 已批阅 / 报告暂未完成），可以编辑未完成的报告，可以查看已批阅的报告的成绩；最后可以退出系统。具体业务流程如图 3.6 所示。

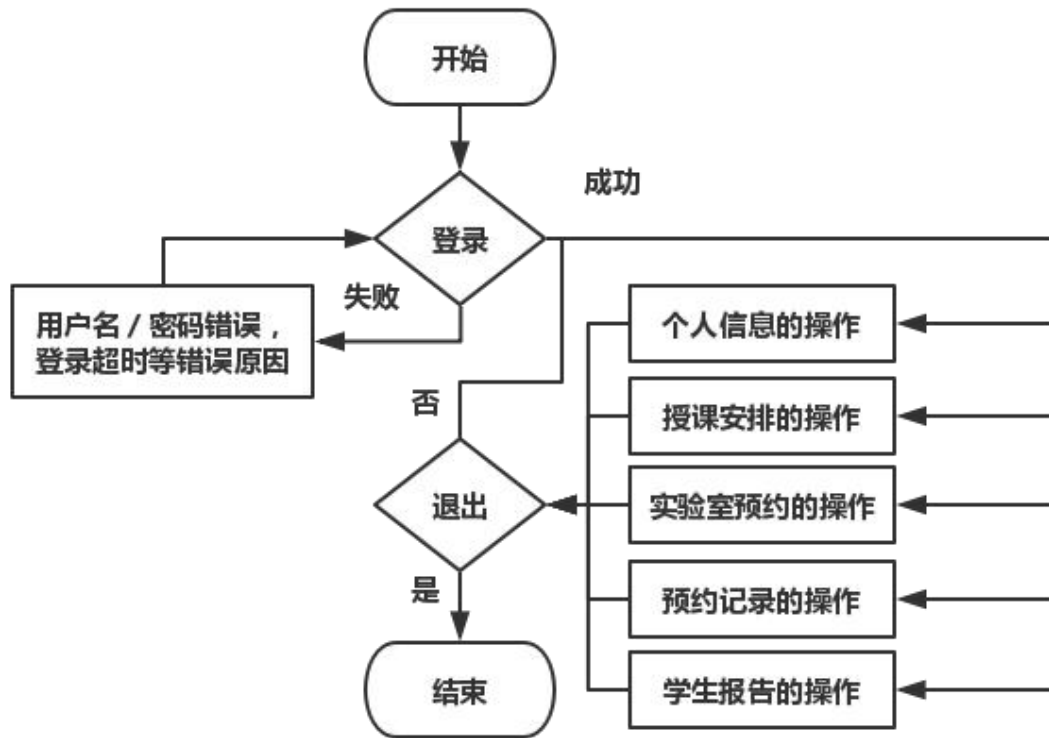


图 3.5 教师业务流程图

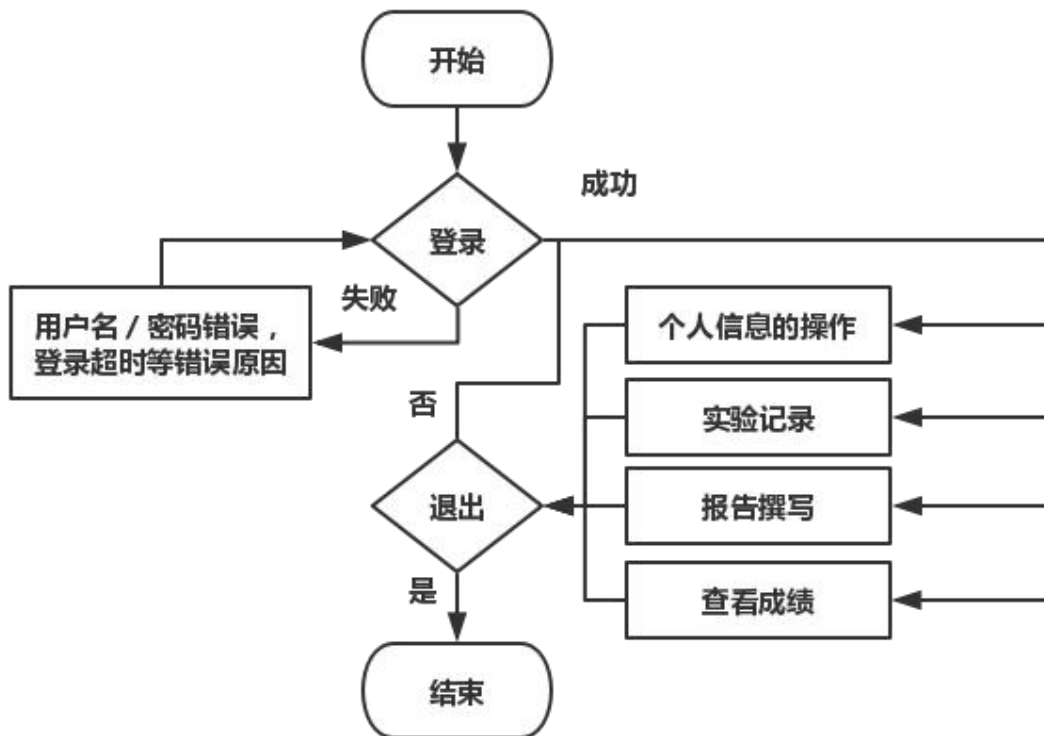


图 3.6 学生业务流程图

### 3.3 系统的可行性综合分析

系统的可行性分析目的是针对系统是否能够正常设计与实现，并在设计实现之后，能否进行正常上线所做的一项评估性分析。结合部分能够娴熟运行实验室管理平台的高等院校，将从经济、操作和技术这三方面进行简要的可行性分析。

#### 3.3.1 经济的可行性综合分析

实验室管理平台的设计与实现基本是各高等院校由教师和实验室管理员，通过个人的移动笔记本或台式电脑和相关服务发布器来设计实现的，系统从设计到开发所需的经费仅仅只是一些参考书籍，以及后期系统于阿里云服务器上运行的相应成本。尽管实现出的系统并不足以推动社会经济发展，但是在培养强能力，高素质，全面发展的人才这一方面起到不可忽视的作用。

因此，结合经济可行性的综合分析，本系统的设计与实现是可行的。

#### 3.3.2 操作的可行性综合分析

各高等院校对于实验室的建设管理高度重视，同时，在软硬件高度发展的时代背景中，网络已覆盖各高等院校各个角落，全校师生可以在校园的任意时间地点中，进行实验室预约、信息管理等相关功能。

因此，结合操作可行性的综合分析，本系统的设计与实现是可行的。

#### 3.3.3 技术的可行性综合分析

针对实验室管理平台在设计、开发、测试、上线等过程中，从运用到的相关技术和环境做简要分析。

本系统所使用的语言为 Java 语言，Java 语言具有跨平台性，高可用性，其在各大语言排行榜上长期占有第一的优势。

本系统属于经典的管理系统开发，主要针对的是实验室预约信息以及学生报告和成绩等信息的操作管理，包含实验室预约，报告撰写。前端采用目前最为广泛使用、最轻盈的 Bootstrap 框架，使用现代服务器端 Java 模板引擎 Thymeleaf 技术；后端使用目前最流行的高速启动、高可扩展、高可用容器 SpringBoot 框架作为全局管控器，以浏览器作为视图层，SpringMVC 作为控制层，MyBatis 作为 Dao 层；数据库上采用阿里巴巴 Druid 连接池，以及阿里云数据库 ApsaraDB 服务产品下的 RDS MySQL 云端数据库。

通过这些目前相当流行的技术，可使得系统变得更加高可用，高可靠，高可支持，能够处理并面对大部分突发问题。同时本实验室管理平台开发所需要的上述技术目前都已相当成熟，可供参考的成功案例及资料也很多。在硬件方面也符合系统开发的要求。

因此，结合技术可行性的综合分析，本系统的设计与实现是可行的。

### 3.4 本章小结

本章主要进行了角色用例分析和业务流程图分析，在第一节中介绍了本系统的用例，简要分析了实验室超级管理员、教师以及学生在实验室管理平台中与各个用例之间的关系；在第二节中介绍了本系统的业务流程，分析了各个角色在与相对应的用例中需要完成的业务；在第三节中介绍了本系统的可行性分析情况，通过上述可行性分析表明，实验室管理平台在经济、操作和技术等方面均为可行的。

## 第 4 章 系统概要设计

### 4.1 系统整体的概要设计

#### 4.1.1 系统整体功能结构的概要设计

本系统的整体功能结构图如图 4.1 所示。

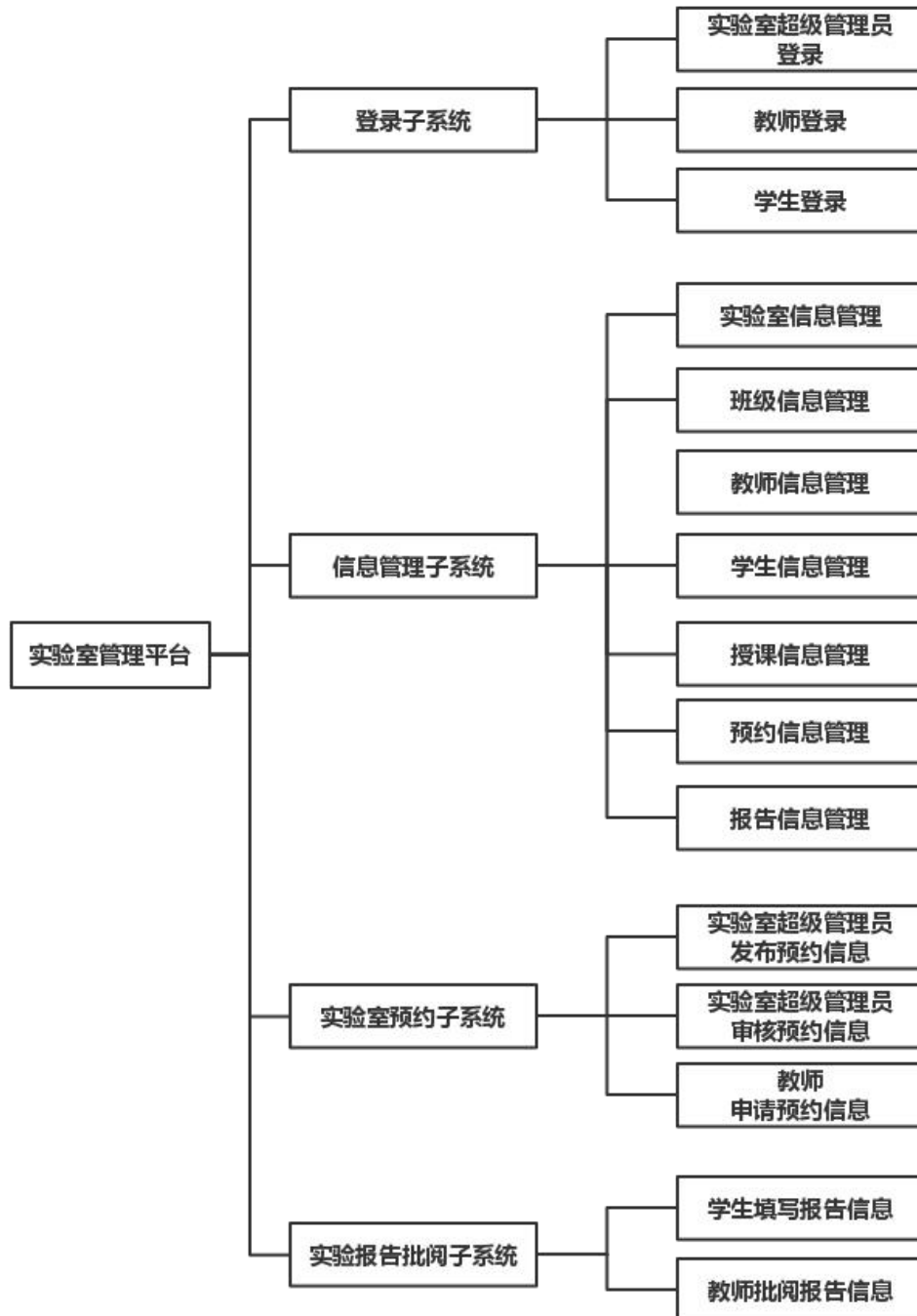


图 4.1 系统整体功能结构图

根据各高等院校实验室实际业务要求，本系统主要包括如下几个基本功能子系统：

- (1) 角色登录的子系统；
- (2) 信息管理的子系统；
- (3) 实验室预约的子系统；
- (4) 实验报告批阅的子系统。

其中，在各个功能子系统下，又被划分为若干个功能子模块，系统各个功能子模块是将实验室管理业务流程作为主线紧密的联系在了一起，从而达到实现各自不同的业务处理要求。

#### 4.1.2 系统整体软件架构的概要设计

本系统的总体软件架构图如图 4.2 所示。

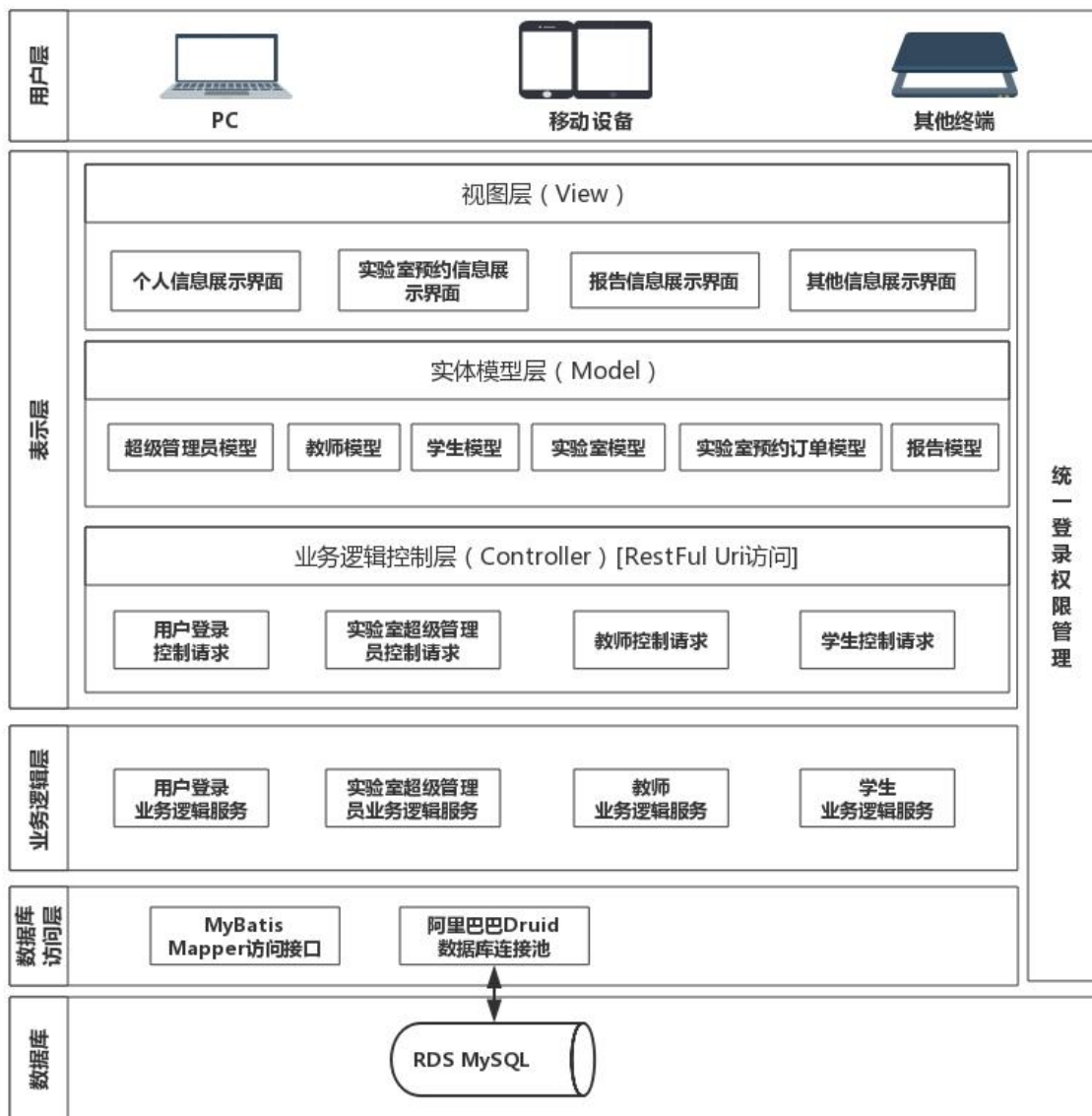


图 4.2 系统整体软件架构图



根据前面所做的系统用例以及业务流程需求分析可知，本系统应用软件是一个基于 B/S 模式下进行开发设计并加以实现的。系统应用软件借助于三层架构的开发模式实现分层技术。整个系统分为数据库访问层、业务逻辑层以及表示层。这三层互相协作、彼此融合，共同构成整个应用系统程序。另一方面，数据库访问层、业务逻辑层以及表示层均是相互独立的，都有自己独立的实现程序，为以后系统应用高可扩展，高可维护做好充足的准备。

## 4.2 系统功能模块设计

### 4.2.1 登录功能模块

为保证系统安全可靠，本系统需要登录才能访问其他功能。若在未登录的情况下，访问其他功能，会受到权限限制，跳转至登录功能，进行登录。根据上文系统用例分析中，本系统可登录角色分为实验室超级管理员、教师和学生。实验室管超级管理员通过管理账号及密码登录，教师和学生分别通过工号或学号以及密码登录。登录功能模块如图 4.3 所示。



图 4.3 登录功能模块图

### 4.2.2 信息管理功能模块

信息管理模块满足本系统大部分信息管理需求，其中有基本信息（包含实验室信息，教师信息，学生信息，班级信息，授课安排信息）的增加、删除、修改、查看。同时满足查看、修改、删除未预约的实验室，查看预约失败的实验室的失败信息，新增可预约实验室。还有查看、批阅学生报告。这些信息还可以通过各自的属性，进行相对应的模糊查询。信息管理功能模块如图 4.4 所示。

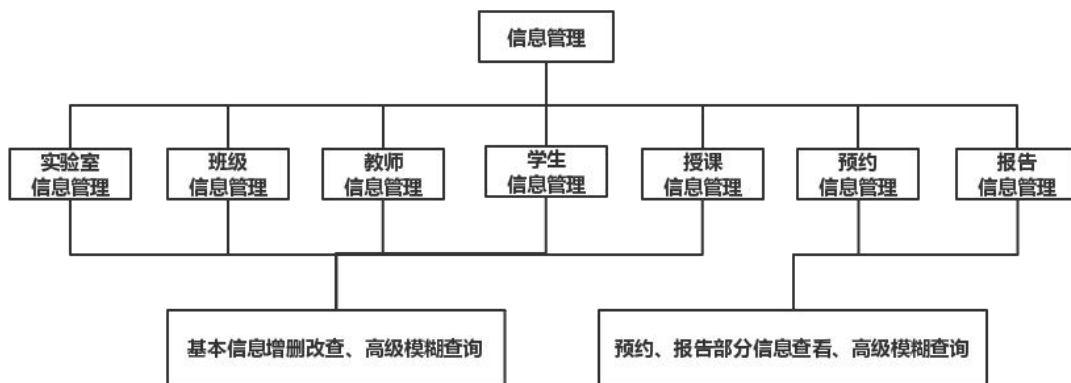


图 4.4 信息管理功能模块

### 4.2.3 实验室预约功能模块

实验室预约功能模块用于，对于教师可以查看所有可预约或未预约的实验室，发起预约实验室的请求，对于实验室超级管理员可以查看所有实验室预约申请，同意或拒绝该实验室预约申请，同时还可以发布新的可预约的实验室，并修改或删除发布的预约。实验室预约功能模块如图 4.5 所示。

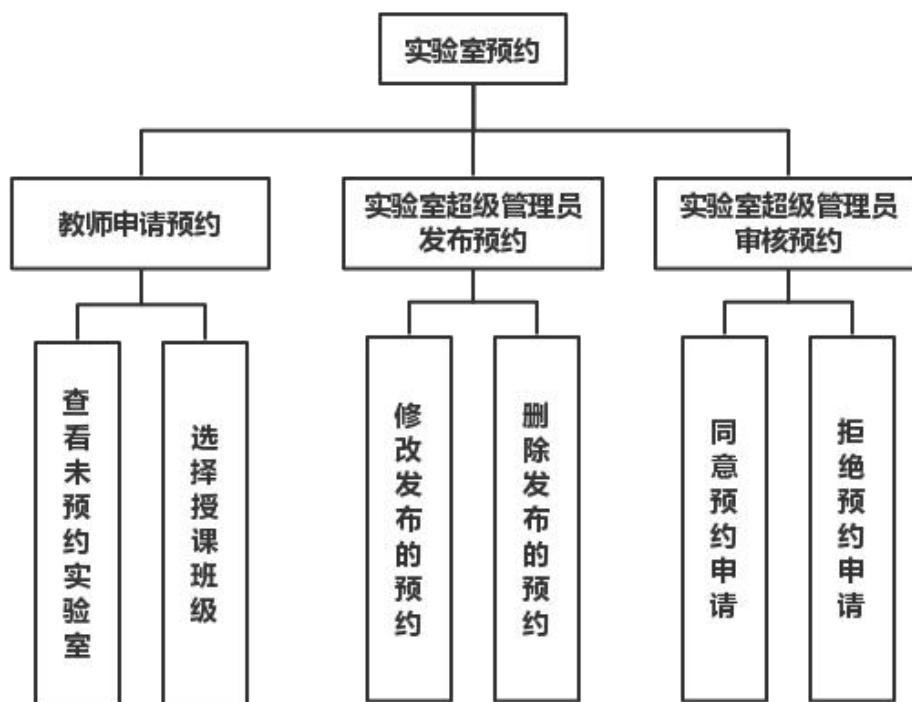


图 4.5 实验室预约功能模块

#### 4.2.4 报告批阅功能模块

报告批阅功能模块主要运用于，在教师成功预约到实验后，学生会收到相应的课程报告。在学生完成实验后，可以自己填写实验报告。而教师可以时刻查看学生完成报告进度，以及了解自己是否已经批阅同学的报告。教师可以通知未完成报告的同学，同时能够批阅已完成的报告，并给予成绩。当教师给予成绩后，该报告成绩就可以在被学生查看。实验室报告批阅功能模块如图 4.6 所示。

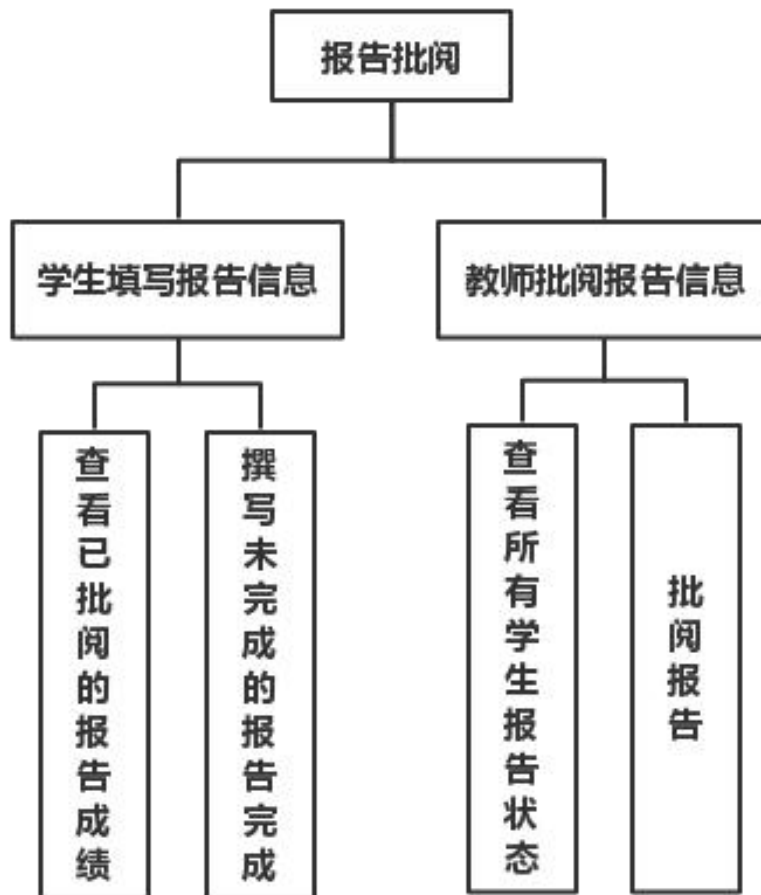


图 4.6 实验报告批阅功能模块

### 4.3 核心程序业务的概要设计

#### 4.3.1 实验室预约程序设计

实验室预约程序的业务流程处理大致如图 4.7 所示。

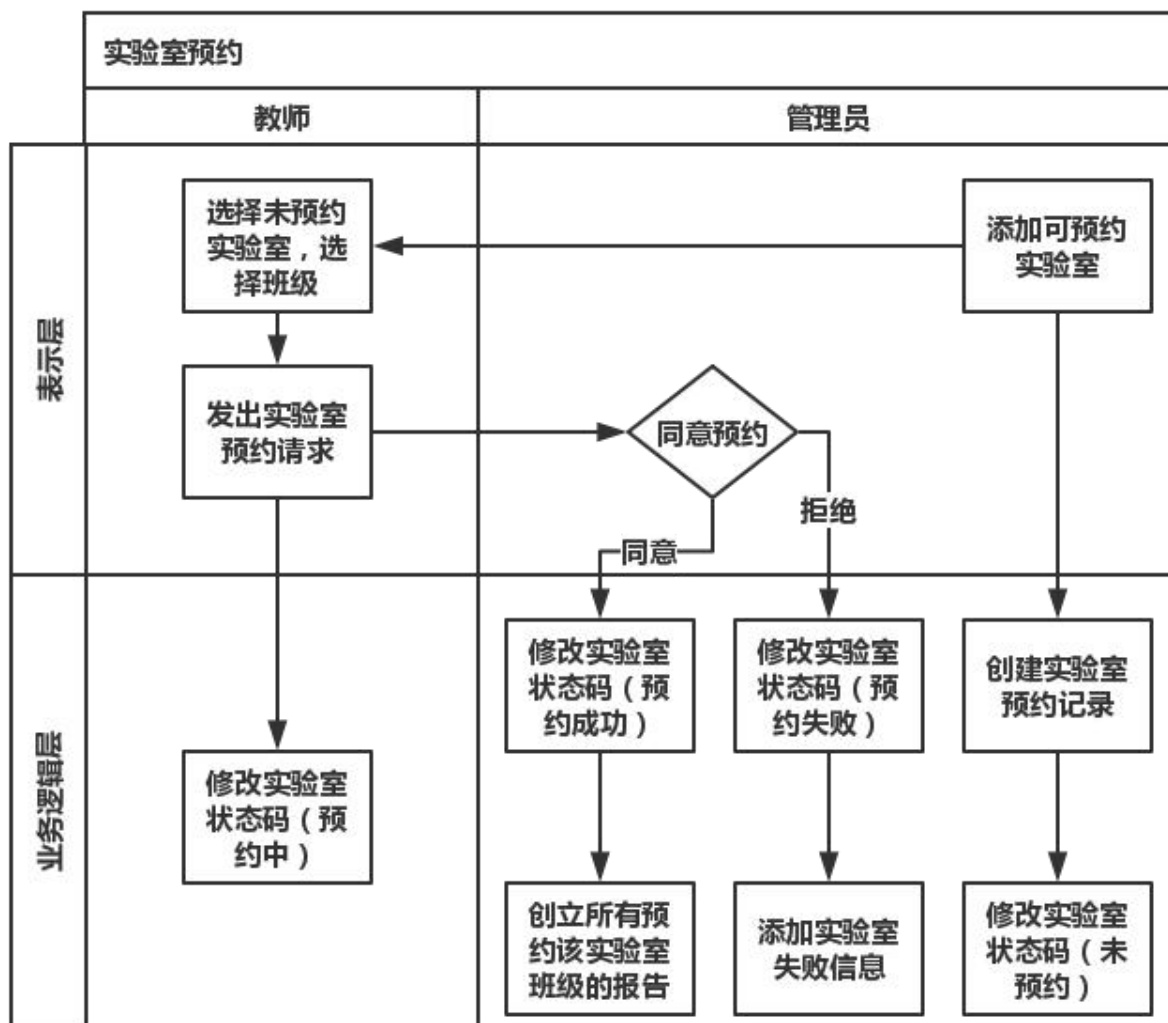


图 4.7 实验室预约泳道图

在实验室预约的过程中。在管理员预约信息管理页，点击新增预约记录，就创建了一条可预约的实验室记录，在教师实验室预约页，就可以查看到这些未预约的实验室，并选择实验室预约，选择实验室后，需要选择授课班级，在页面点击确认后，会向管理员发出一条预约中的实验室记录，由管理员确认是否同意预约。若管理员点击同意预约，则将该条记录状态转变会预约成功，并为预约该实验室的班级中所有同学创建实验报告记录。若管理员点击拒绝预约，则将该条记录状态转变为预约失败，并添加失败缘由。

#### 4.3.2 报告批阅程序设计

报告批阅程序的业务流程处理大致如图 4.8 所示。

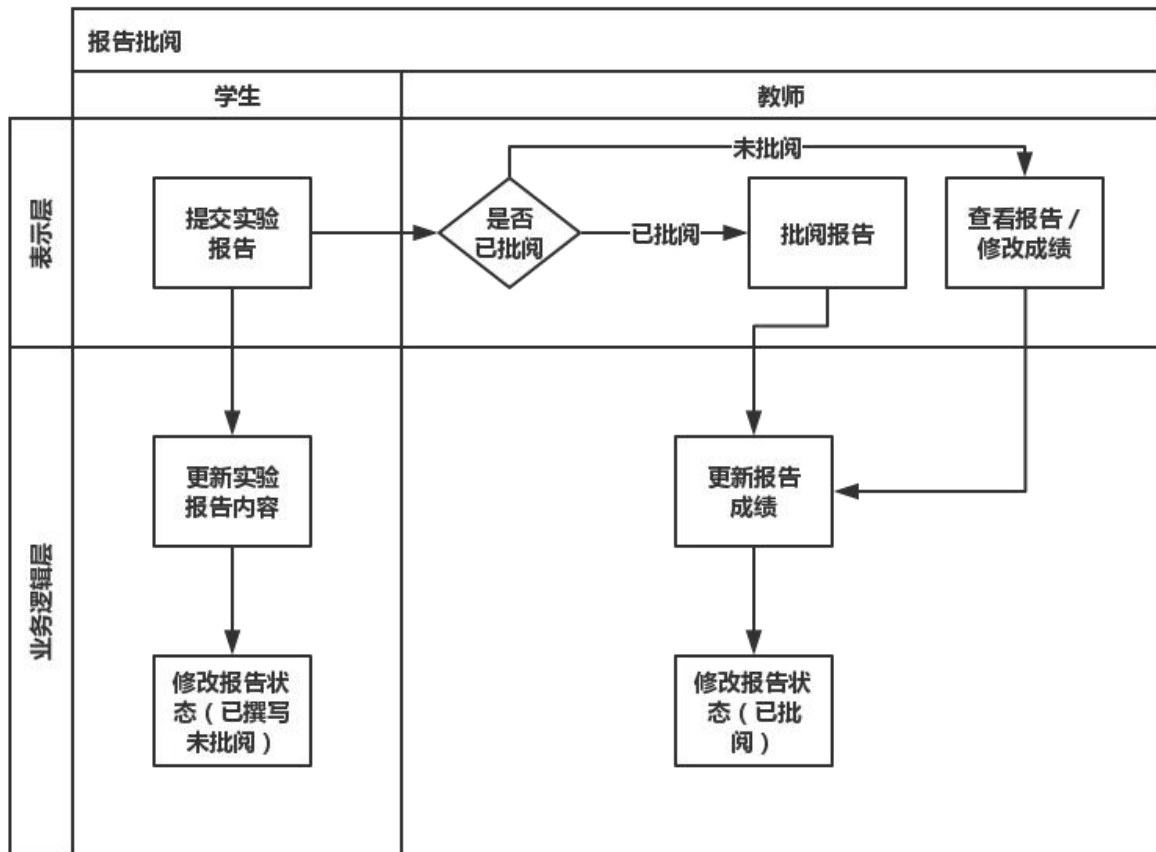


图 4.8 报告批阅泳道图

在报告批阅的过程中。学生完成实验后，在学生报告撰写页，自行主动填写实验报告内容，在提交实验报告之后，实验报告就会被标记为，已撰写未批阅。在此期间，学生也可以进行实验报告内容的修改。在教师报告批阅页，可以查看到所有的学生实验报告状态，若实验报告状态为未批阅，则教师可以批阅，并给予课堂成绩和报告成绩。若为已批阅，可以查看成绩，或是修改成绩。

## 4.4 数据库的概要设计

### 4.4.1 数据库的概念模型的概要设计

通过上述角色用例需求分析，以及功能模块的概要设计，得知在本系统中需要对 8 类实体进行管理，分别是实验室超级管理员、实验室、实验室预约记录、实验报告、学生、班级、课程以及教师。分析各个功能模块实体关系，进行数据建模。系统数据库整体 E-R 图（Entity-Relationship Diagram）如图 4.9 所示。

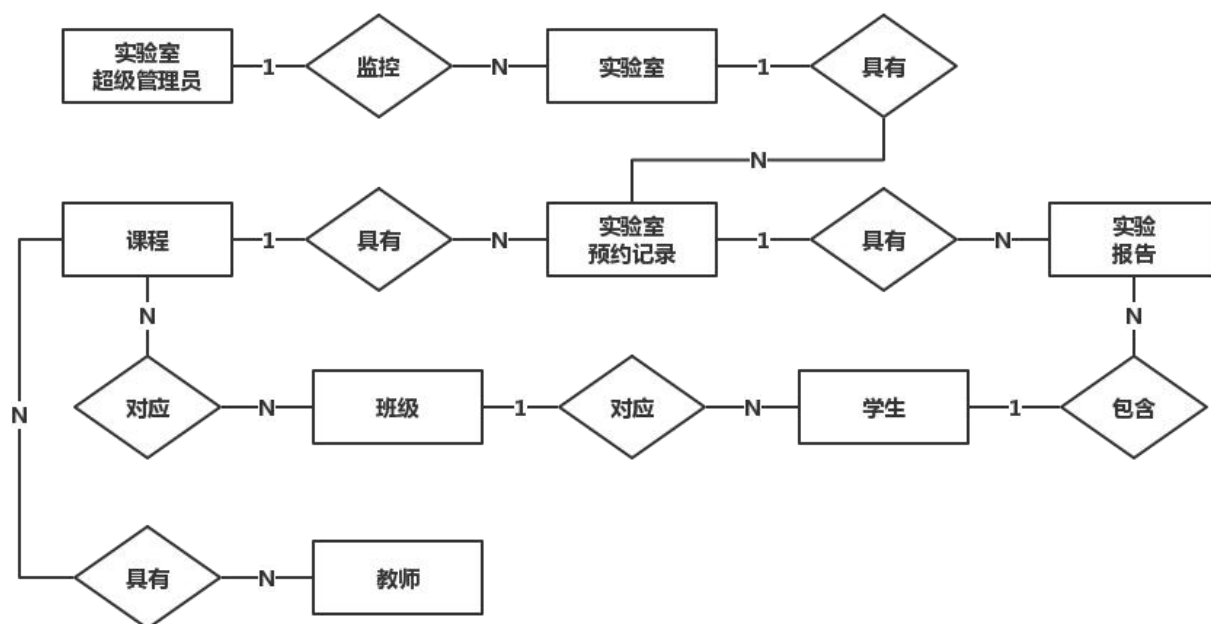


图 4.9 系统数据库整体 E-R 图

如图 4.9 所示，系统各个实体之间关系较为复杂。尤其是在实验室预约记录与实验室、课程以及实验报告之间的关系。从图 4.9 中可基本得知：

- (1) 一名实验室超级管理员监控多间实验室；
- (2) 一间实验室有且仅由一名管理员监控；
- (3) 一间实验室具有多条实验室预约记录；
- (4) 一条实验室预约记录有且仅具有一间实验室；
- (5) 一条实验室预约记录具有多份实验报告；
- (6) 一条实验室预约记录有且仅具有一门课程；
- (7) 一份实验报告有且仅具有一条实验预约记录；
- (8) 一份实验报告有且仅包含一名学生；
- (9) 一名学生包含多份实验报告；
- (10) 一名学生有且仅在一个班级中；
- (11) 一个班级中包含有许多名学生；
- (12) 一个班级可能需要上多门课程；
- (13) 一门课程可能会有许多个班级在上；
- (14) 一门课程具有多条实验室预约记录；
- (15) 一门课程对应多名教师；
- (16) 一名教师对应多门课程。

从上述基本实体对应关系中，可以看出部分实体的复杂关系，如：

通过（1）、（2）、（3）、（4）可知，一名实验室超级管理员管理多条实验室预约记录，一条实验室预约记录有且仅由一名实验室超级管理员管理。

通过（6）、（14）、（15）、（16）可知，一条实验室预约记录有且仅由一名教师预约，一名教师预约有多条实验室预约记录。

通过（5）、（6）、（7）、（14）、（15）、（16）可知，一份实验报告有且仅由一名教师批阅，一名教师批阅多份实验报告。

通过对实体的具体分析，可以得出各个实体的各自属性。以下是各个实体属性的实体图：

- （1）管理员的属性实体图如图 4.10 所示；
- （2）教师的属性实体图如图 4.11 所示；
- （3）学生的属性实体图如图 4.12 所示；
- （4）班级的属性实体图如图 4.13 所示；
- （5）课程的属性实体图如图 4.14 所示；
- （6）实验室的属性实体图如图 4.15 所示；
- （7）实验室预约记录的属性实体图如图 4.16 所示；
- （8）实验报告的属性实体图如图 4.17 所示。

#### 4.4.2 数据库的逻辑结构的概要设计

在逻辑结构的概要设计中，其主要任务就是把在概念模型的概要设计基础上，将设计完成的 E-R 图转变为与本系统所选用的数据库的管理系统（Database Management System）的概念模型相同的逻辑结构。

本系统数据库的逻辑结构的设计主要包括管理员表，教师表，学生表，班级表，课程表，实验室表，预约表和报告表等二维表以及表和表之间的关联关系。如图 4.18 所示。

#### 4.4.3 数据库的物理结构的概要设计

（1）管理员表（admin\_tbl）中的字段如表 4.1 所示，主要是管理员相关的一些信息，包括 Id，姓名和密码等信息。



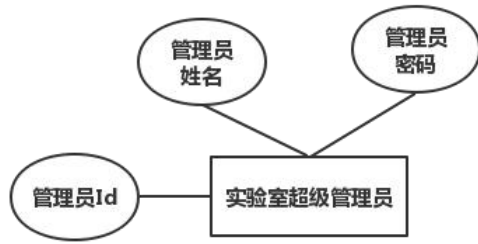


图 4.10 管理员的属性实体图

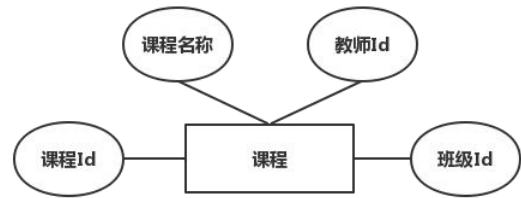


图 4.14 课程的属性实体图

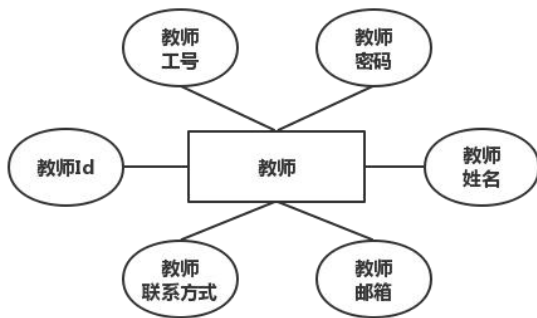


图 4.11 教师的属性实体图

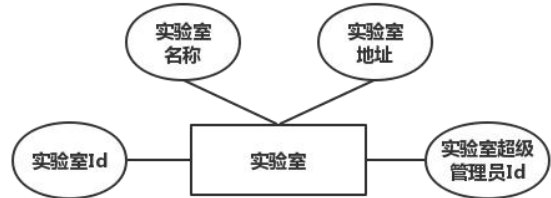


图 4.15 实验室的属性实体图

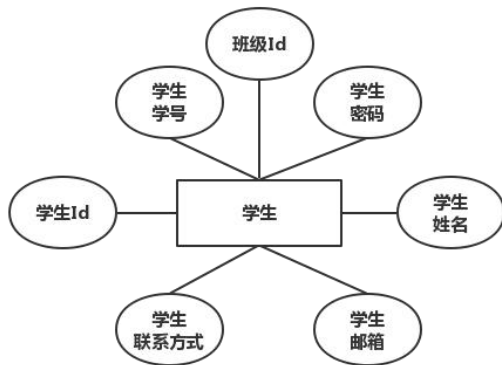


图 4.12 学生的属性实体图

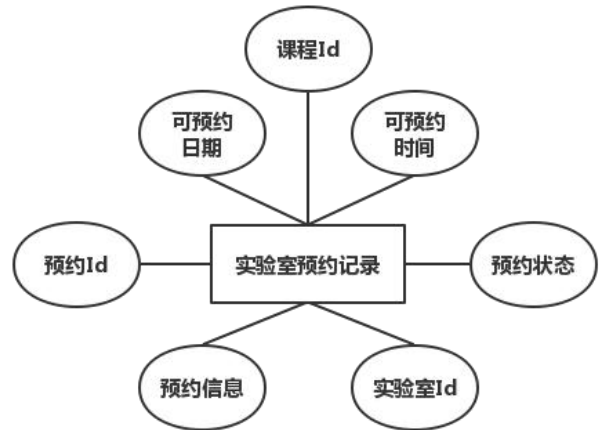


图 4.16 实验室预约记录的属性实体图

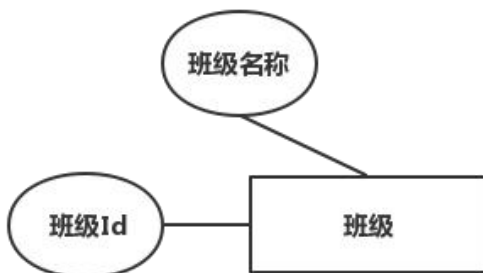


图 4.13 班级的属性实体图

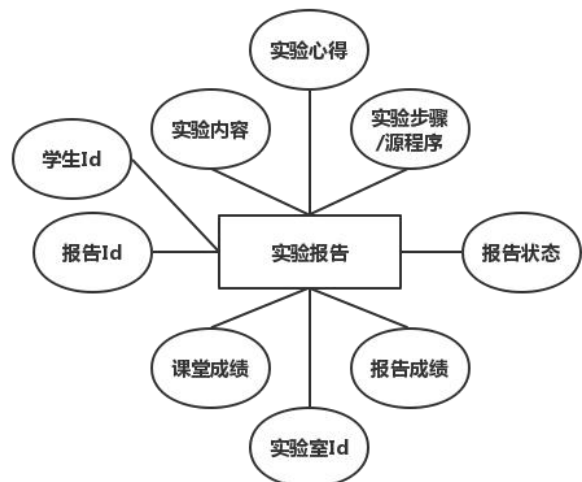


图 4.17 实验报告的属性实体图



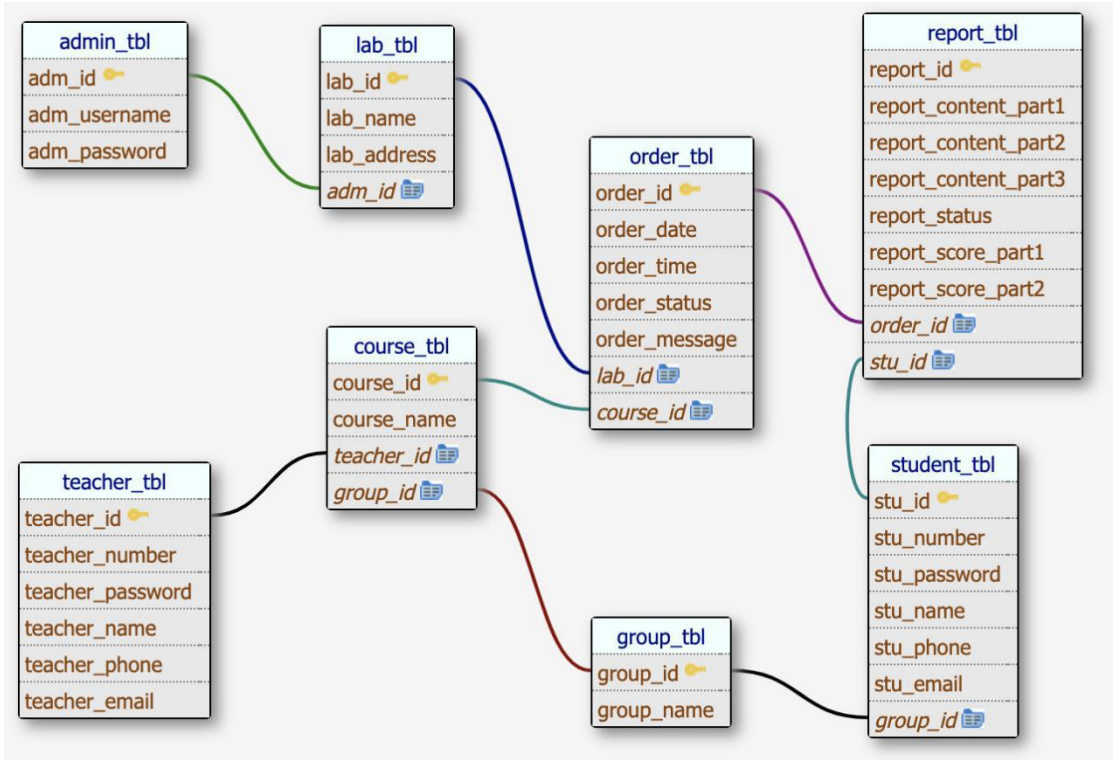


图 4.18 系统数据库整体逻辑结构图

表 4.1 实验室超级管理员表

名称	代码	数据类型	空值	备注
管理员 Id	adm_id	int(11) auto_increment	NOT NULL	PRIMARY KEY
管理员姓名	adm_username	varchar(255)	NULL	
管理员密码	adm_password	varchar(255)	NULL	

- (2) 教师表（teacher\_tbl）中的字段如表 4.2 所示，主要是教师相关的一些信息，包括 Id，工号，密码，姓名，手机号以及邮箱等信息。
- (3) 学生表（student\_tbl）中的字段如表 4.3 所示，主要是学生相关的一些信息，包括 Id，学号，密码，姓名，手机号，邮箱，以及班级等信息。
- (4) 班级表（group\_tbl）中的字段如表 4.4 所示，主要是班级相关的一些信息，包括 Id 和名称等信息。

表 4.2 教师表

名称	代码	数据类型	空值	备注
教师 Id	teacher_id	int(11)	NOT	PRIMARY
		auto_increment	NULL	KEY
教师工号	teacher_number	varchar(255)	NULL	
教师密码	teacher_password	varchar(255)	NULL	
教师姓名	teacher_name	varchar(255)	NULL	
教师联系方式	teacher_phone	varchar(255)	NULL	
教师邮箱	teacher_email	varchar(255)	NULL	

表 4.3 学生表

名称	代码	数据类型	空值	备注
学生 Id	stu_id	int(11)	NOT NULL	PRIMARY
		auto_increment		KEY
学生学号	stu_number	varchar(255)	NULL	
学生密码	stu_password	varchar(255)	NULL	
学生姓名	stu_name	varchar(255)	NULL	
学生联系方式	stu_phone	varchar(255)	NULL	
学生邮箱	stu_email	varchar(255)	NULL	
所在班级	group_id	int(11)	NULL	FOREIGNER KEY

表 4.4 班级表

名称	代码	数据类型	空值	备注
班级 Id	group_id	int(11)	NOT NULL	PRIMARY
		auto_increment		KEY
班级名称	group_name	varchar(255)	NULL	

（5）课程表（course\_tbl）字段如表 4.5 所示，主要是课程相关的一些信息，包括 Id，名称，老师以及班级等信息。

表 4.5 课程表

名称	代码	数据类型	空值	备注
课程 Id	course_id	int(11) auto_increment	NOT NULL	PRIMARY KEY
课程名称	course_name	varchar(255)	NULL	
授课老师	teacher_id	int(11)	NULL	FOREIGNER KEY
授课班级	group_	int(11)	NULL	FOREIGNER KEY

（6）实验室表（lab\_tbl）字段如表 4.6 所示，主要是实验室的相关信息，包括 Id，名称，地点以及管理员等信息。

表 4.6 实验室表

名称	代码	数据类型	空值	备注
实验室 Id	lab_id	int(11) auto_increment	NOT NULL	PRIMARY KEY
实验室名称	lab_name	varchar(255)	NULL	
实验室地点	lab_address	varchar(255)	NULL	
实验室管理员	adm_id	int(11)	NULL	FOREIGNER KEY

（7）实验室预约记录表（order\_tbl）字段如表 4.7 所示，主要是实验室预约的相关信息，包括预约 Id，预约日期，预约时间，预约状态，预约信息，预约的实验室以及预约的课程等信息。

（8）报告表（report\_tbl）字段如图 4.8 所示，主要是报告的相关信息，包括报告 Id，报告内容，报告步骤 / 源程序，报告心得，报告状态，课堂成绩，报告成绩，预约的实验以及填写报告的学生等信息

表 4.7 实验室预约表

名称	代码	数据类型	空值	备注
预约 Id	order_id	int(11)	NOT	PRIMARY
		auto_increment	NULL	KEY
预约日期	order_date	varchar(255)	NULL	
预约时间	order_time	int(11)	NULL	
预约状态	order_status	int(11)	NULL	
预约信息	order_message	varchar(255)	NULL	
预约的实验室	lab_id	int(11)	NULL	FOREIGNER KEY
预约的课程	course_id	int(11)	NULL	FOREIGNER KEY

表 4.8 报告表

名称	代码	数据类型	空值	备注
报告 Id	report_id	int(11)	NOT	PRIMARY
		auto_increment	NULL	KEY
报告内容	report_content_part1	varchar(255)	NULL	
报告步骤 / 源程序	report_content_part2	varchar(255)	NULL	
报告心得	report_content_part3	varchar(255)	NULL	
报告状态	report_status	int(11)	NULL	
课堂成绩	report_score_part1	double	NULL	
报告成绩	report_score_part2	double	NULL	
预约的实验	order_id	int(11)	NULL	FOREIGNER KEY
填写报告的学生	stu_id	int(11)	NULL	FOREIGNER KEY

## 4.5 本章小结

本章主要对实验室管理平台的设计原则与设计目标做了明确的说明，从对系统总体功能结构，总体软件架构设计出发，对系统各功能模块做了概要的设计。并在最后从数据库概念结构转到逻辑结构，最后到物理结构的层次性的设计，完成了对系统的数据库的简要设计。

## 第 5 章 系统的总体设计与实现

### 5.1 实现环境

#### 5.1.1 开发环境

(1) 系统实现开发硬件环境如表 5.1 所示。

表 5.1 开发硬件环境表

操作系统	处理器	内存	硬盘
MacOS Mojave	2.9 GHz Inter Core i9	32 GB 2400 MHz DDR4	1TB 固态

(2) 系统实现开发软件环境如表 5.2 所示。

表 5.2 开发软件环境表

设计平台	设计语言	数据库
IntelliJ IDEA 2018.3.6	Java 1.8.0_211	MySQL 5.7.26

#### 5.1.2 发布环境

(1) 系统实现发布硬件环境如表 5.3 所示。

表 5.3 发布硬件环境表

操作系统	处理器	内存	硬盘
云服务器 ECS Centos_7_06_64	2.5 GHz Intel Xeon	1GiB	40GiB 高效云盘

(2) 系统实现发布软件环境如表 5.4 所示。

表 5.4 发布软件环境表

服务器	处理语言	数据库
云服务器 ECS	Java 1.8.0_211	云数据库 ApsaraDB RDS MySQL 5.7 版

#### 5.1.3 系统实现部署方式

本系统为集中式开发系统，所有模块都将同时上线，同时测试，同时下线，缺一不可。所有业务信息流都统一通过服务器集中处理。

## 5.2 系统项目搭建与配置

本系统使用 SpringBoot+Gradle 的快速启动+项目依赖管理的模式构建。其目录结构如图 5.1 所示。

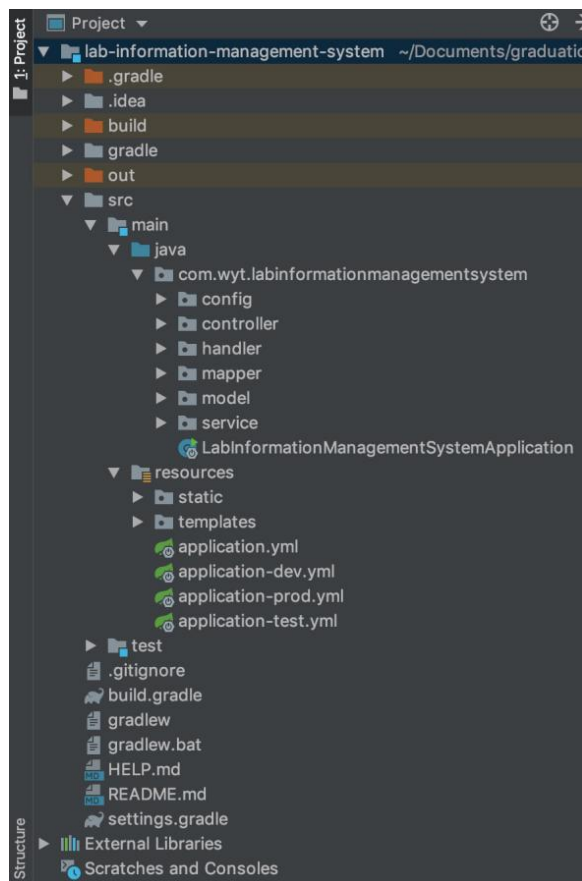


图 5.1 项目目录结构图

从图 5.1 中可以看到 LabInformationManagementSystemApplication.java 就是 SpringBoot 快速启动类，运行该类就可以快速启动本系统。在目录结构中 build.gradle 就是本系统项目的项目依赖管理器配置。application.yml， application-dev.yml， application-prod.yml， application-test.yml 则是本系统项目的基础配置，开发环境配置，发布环境配置，测试环境配置。在 static 文件夹中存放本系统项目所有需要使用到的 JS， CSS， Image 等静态文件。在 templates 文件夹中则是存放本系统项目所有的 Html 页面。 templates+controller+model 则象征着本系统项目使用 MVC 的设计模式。 controller+service+mapper 则象征着本系统项目使用三层架构开发模式。 config 文件夹中存放开发者自行额外配置的相关配置文件，在本系统项目中则需要额外配置 Druid 连接池配置， SpringMVC 配置和 MyBatis 配置。在 handler 文件夹中存放全局配置文件，在本系统项目中则需要配置全局登录拦截器（即登录权限管理器）。

### 5.2.1 SpringBoot 主启动类

通过该类可以快速启动 SpringBoot 的系统项目。其具体代码如下。

```
@SpringBootApplication
public class LabInformationManagementSystemApplication {
    public static void main(String[] args) {
        SpringApplication.run(LabInformationManagementSystemApplication.class,
args);
    }
}
```

### 5.2.2 Gradle 项目依赖管理器配置

通过该配置文件可以进行对 SpringBoot 版本，相关插件（如：Spring 的版本依赖控制器），依赖仓库，以及依赖进行管理。其具体配置代码如下。

```
plugins {
    id 'org.springframework.boot' version '2.1.5.RELEASE'
    id 'java'
}
apply plugin: 'io.spring.dependency-management'
group = 'com.wyt'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '1.8'
configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}
repositories {
    maven { url 'http://maven.aliyun.com/nexus/content/groups/public/' }
    mavenCentral()
}
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-jdbc'
```



```

implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
implementation 'org.springframework.boot:spring-boot-starter-web'
implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.0.1'
compileOnly 'org.projectlombok:lombok'
runtimeOnly 'mysql:mysql-connector-java'
annotationProcessor 'org.projectlombok:lombok'
testImplementation 'org.springframework.boot:spring-boot-starter-test'
compile 'org.webjars:jquery:3.4.1'
compile 'org.webjars:bootstrap:4.3.1'
compile 'com.alibaba:druid:1.1.10'
compile 'org.webjars.npm:feather-icons:4.10.0'
}

```

### 5.2.3 系统项目基础配置

系统项目基础配置主要为 Spring 的部分自定义属性配置，如数据库连接池，数据库连接，Thymeleaf 模板缓存，SpringMVC 日期格式等相关属性配置。同时还可以额外添加开发配置，测试配置以及发布配置，来合并并修改相关属性配置，比如数据库连接属性配置，Tomcat 服务端口、路径属性配置来满足各种环境下的不同应用。

（1）基础配置文件 application.yml 具体配置代码如下：

```

# spring Configuration
spring:
  profiles:
    active: dev
# DB Configuration
datasource:
  # DB Pool Configuration
  type: com.alibaba.druid.pool.DruidDataSource
  # 连接池配置信息
  initialSize: 5
  minIdle: 5
  maxActive: 20
  maxWait: 60000

```

```
timeBetweenEvictionRunsMillis: 60000
minEvictableIdleTimeMillis: 300000
validationQuery: SELECT 1 FROM DUAL
testWhileIdle: true
testOnBorrow: false
testOnReturn: false
poolPreparedStatements: true
# 连接池其他配置信息
filters: stat,wall
maxPoolPreparedStatementPerConnectionSize: 20
useGlobalDataSourceStat: true
connectionProperties: druid.stat.mergeSql=true;druid.stat.slowSqlMillis=500
# Thymeleaf Configuration
thymeleaf:
  cache: false
# MVC date format
# default configuration:
mvc:
  date-format: yyyy-MM-dd
```

(2) 开发配置文件 application-dev.yml 具体配置代码如下:

```
# server Configuration
server:
  servlet:
    context-path: /lims
  port: 8888
# spring Configuration
spring:
  # DB Configuration
  datasource:
    username: root
    driver-class-name: com.mysql.jdbc.Driver
```

```
password: root
url: jdbc:mysql://localhost/lims
# show sql in console
logging:
  level:
    com.wyt.labinformationmanagementsystem.mapper.*: debug
```

（3）测试配置文件 application-test.yml 具体配置代码如下：

```
# server Configuration
server:
  servlet:
    context-path: /lims
  port: 80
# spring Configuration
spring:
  # DB Configuration
  datasource:
    username: root
    driver-class-name: com.mysql.jdbc.Driver
    password: 199612034532Wyt
    url: jdbc:mysql://rm-bp12yhr4bu4nn8dslco.mysql.rds.aliyuncs.com:3306/lims
# show sql in console
logging:
  level:
    com.wyt.labinformationmanagementsystem.mapper.*: debug
```

（4）发布配置文件 application-prod.yml 具体代码如下：

```
# server Configuration
server:
  servlet:
    context-path: /lims
  port: 80
# spring Configuration
```

spring:

```
# DB Configuration
```

```
datasource:
```

```
username: root
```

```
driver-class-name: com.mysql.jdbc.Driver
```

```
password: 199612034532Wyt
```

```
url: jdbc:mysql://rm-bp12yhr4bu4nn8dsl.mysql.rds.aliyuncs.com:3306/lims
```

## 5.2.4 系统项目额外配置

### (1) Druid 连接池相关配置

将阿里巴巴开源的 Druid 连接池注册到 Spring 容器中，就可以使用“项目访问路径+/druid”监控全部交互的数据操作。Druid 界面如图 5.2 所示，Druid 具体配置代码如下：

```
@Configuration
```

```
public class DruidConfig {
```

```
    @Bean
```

```
    @ConfigurationProperties(prefix = "spring.datasource")
```

```
    public DataSource druid(){
```

```
        return new DruidDataSource();
```

```
    }
```

```
    //配置 Druid 的监控
```

```
    //1.配置一个管理后台的 Servlet
```

```
    @Bean
```

```
    public ServletRegistrationBean statViewServlet(){
```

```
        ServletRegistrationBean bean = new ServletRegistrationBean(new  
StatViewServlet(), "/druid/*");
```

```
        Map<String, String> initParam = new HashMap<>();
```

```
        initParam.put("loginUsername", "admin");
```

```
        initParam.put("loginPassword", "admin");
```

```
        bean.setInitParameters(initParam);
```

```
        return bean;
```

```
    }
```

//2.配置一个监控的 filter

@Bean

```
public FilterRegistrationBean webStatFilter(){
    FilterRegistrationBean bean = new FilterRegistrationBean<>();
    bean.setFilter(new WebStatFilter());
    Map<String, String> initParam = new HashMap<>();
    initParam.put("exclusions", "*.*.js,*.css,/druid/*");
    bean.setInitParameters(initParam);
    bean.setUrlPatterns(Arrays.asList("/*"));
    return bean;
}
```

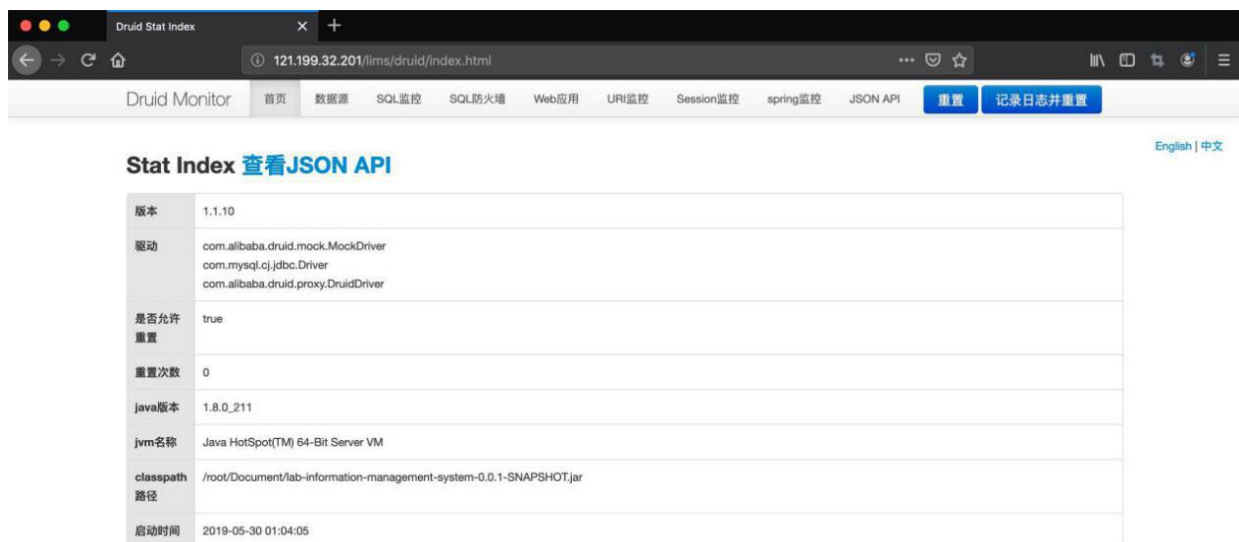


图 5.2 Druid 界面图

## (2) SpringMVC 配置

由于 SpringMVC 不支持直接访问页面，只能通过控制器返回才能访问。若想直接访问主页则需要在 SpringMVC 配置中额外添加主页的映射关系。其具体代码如下：

@Configuration

```
public class MvcConfig implements WebMvcConfigurer {
```

    @Bean

```
    public WebMvcConfigurer webMvcConfigurer(){
```

```
        WebMvcConfigurer configurator = new WebMvcConfigurer() {
```

```
            @Override
```

```
            public void addViewControllers(ViewControllerRegistry registry) {
```

```
                registry.addViewController("/").setViewName("login");
```

```
                registry.addViewController("index.html").setViewName("login");
```

```
                registry.addViewController("stuMain.html").setViewName("student/dashboard");
```

```
                registry.addViewController("adminMain.html").setViewName("admin/dashboard");
```

```
                registry.addViewController("teacherMain.html").setViewName("teacher/dashboard");
```

```
            }
```

```
            @Override
```

```
            public void addInterceptors(InterceptorRegistry registry) {
```

```
                registry.addInterceptor(new LoginHandlerInterceptor())
```

```
                    .addPathPatterns("/**")
```

```
                    .excludePathPatterns("/index.html","/","/user/login","/webj
```

```
ars/**","/css/**","/js/**","/img/**","/laydate/**");
```

```
            }
```

```
        };
```

```
        return configurator;
```

```
    }
```

```
}
```

### （3）MyBatis 配置

由于在 Java 程序设计中默认的命名规范为驼峰命名法，而在数据库设计中大多使用的命名规则是以下划线分割单词，所以若要使用 MyBatis 自动映射表中字段与 Model 中的实体，则需要配置驼峰转换。其具体配置代码如下：

@Configuration

```
public class MyBatisConfig {
    @Bean
    public ConfigurationCustomizer configurationCustomizer(){
        return configuration -> configuration.setMapUnderscoreToCamelCase(true);
    }
}
```

#### 5.2.5 系统项目全局配置

通过之前的系统总体软件架构设计可知在访问本系统的三层架构的同时，需要有一个统一的登录权限管理，即当你没有登录的时候就不能访问本系统的表示层，业务逻辑层，数据库访问层，并返回错误信息到登录界面。其具体配置代码如下：

```
public class LoginHandlerInterceptor implements HandlerInterceptor {
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {
        Object user = request.getSession().getAttribute("loginUser");
        if(user == null){
            //未登录,返回登录页面
            request.setAttribute("msg","没有权限,请先登录");
            request.getRequestDispatcher("/index.html").forward(request,response);
            return false;
        } else {
            return true;
        }
    }
}
```

## 5.3 子系统设计与实现

根据角色用例图以及业务流程图的需求分析，我们按角色设计每个子系统，并加以实现。按照 MVC 设计模式和三层架构的设计思想，我们在表示层控制器（controller）中建立 AdminController.java，StudentController.java，TeacherController.java 和 UserController.java。在业务逻辑层（service）中建立 AdminService.java，StudentService，TeacherService，UserService。根据系统数据库设计，按照数据库表与 Model 实体互相映射，表中字段与 Model 中实体属性互相映射的原则，我们在表示层模型（model）中建立 Admin.java，Course.java，Group.java，Lab.java，Order.java，Report.java，Student.java，Teacher.java 八大实体类。根据业务流程需要，我们额外建立 OrderRecordCondition.java，PageBean.java，ReprotCondition.java 三个额外封装类。同时在数据库访问层（mapper）中建立 AdminMapper.java，CourseMapper.java，GroupMapper.java，LabMapper.java，OrderMapper.java，ReportMapper.java，StudentMapper.java，TeacherMapper.java 八大实体数据库访问接口。

### 5.3.1 登录子系统设计与实现

为了保证实验室管理平台信息的安全性，在访问系统登录之外的功能之前需要通过角色验证。在登录的界面需要输入正确用户名和密码登录。在角色下拉选择器中可供选择教师、学生、管理员三种角色身份。登录信息采用了角色身份是否为空校验，数据库用户校验，后端密码校验，如果任意一个不匹配，则换回登录界面并提示错误信息。

用户登录界面图如图 5.3 所示。

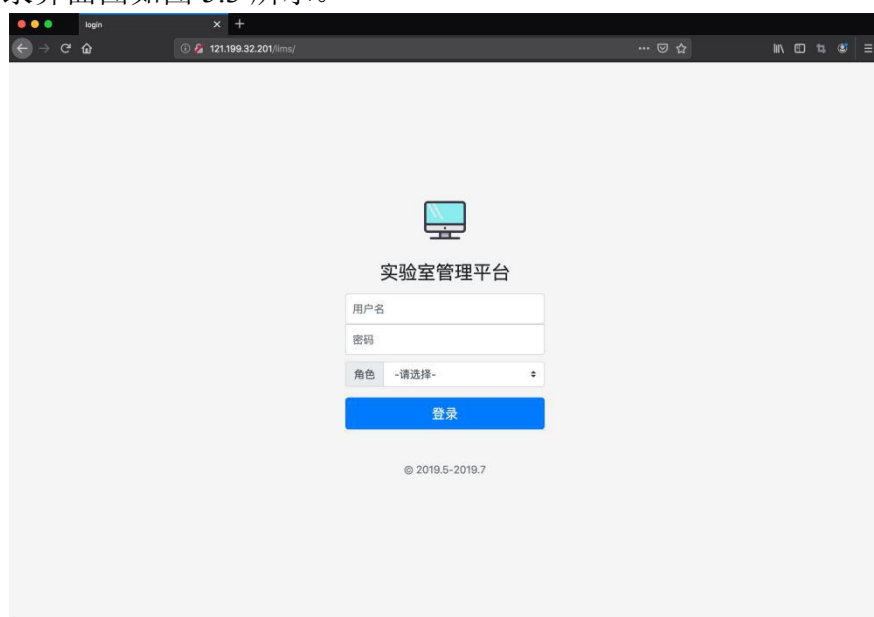


图 5.3 用户登录界面图



登录核心代码如下：

```
@PostMapping("/login")

public String login(@RequestParam("number") String number,
                    @RequestParam("password") String password,
                    @RequestParam("role") String role,
                    Map<String, Object> map, HttpSession session){
    if("").equals(role)){
        map.put("msg","请选择角色");
        return "login";
    }
    if("admin".equals(role)){
        Admin admin = userService.adminLogin(number);
        if(admin!=null && admin.getAdmPassword().equals(password)){
            session.setAttribute("loginUser",admin);
            return "redirect:/adminMain.html";
        }
    } else if("student".equals(role)){
        Student student = userService.stuentLogin(number);
        if(student!=null && student.getStuPassword().equals(password)){
            session.setAttribute("loginUser",student);
            return "redirect:/stuMain.html";
        }
    } else if("teacher".equals(role)){
        Teacher teacher = userService.teacherLogin(number);
        if(teacher!=null && teacher.getTeacherPassword().equals(password)){
            session.setAttribute("loginUser",teacher);
            return "redirect:/teacherMain.html";
        }
    }
    map.put("msg","登录失败");
    return "login";
}
```

### 5.3.2 信息管理子系统设计与实现

(1) 以管理员身份登录后,可以对实验室信息、教师信息、班级信息、学生信息、授课信息进行增删改查以及高级模糊查询。也可以查看或模糊查询所有预约信息以及报告信息。界面图如图 5.4,图 5.5,图 5.6,图 5.7,图 5.8,图 5.9,图 5.10 所示。

(2) 以教师身份登录后,可以对个人信息进行查看、修改。授课信息、可预约实验室信息查看、预约记录、学生实验报告查看,高级模糊查询。界面图如图 5.11,图 5.12,图 5.13,图 5.14,图 5.15 所示。

(3) 以学生身份登录后,可以对个人信息作查看、修改等操作。查看教师已成功预约的实验记录,以及查看自己做的所有的实验报告。界面图如图 5.16,图 5.17 所示。

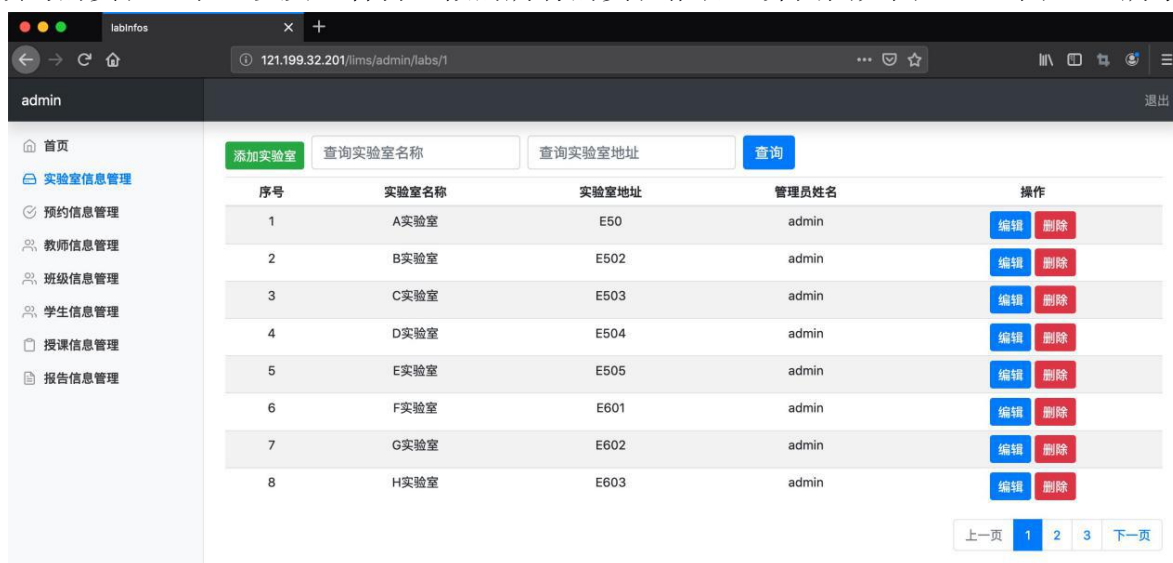


图 5.4 管理员实验室信息管理图

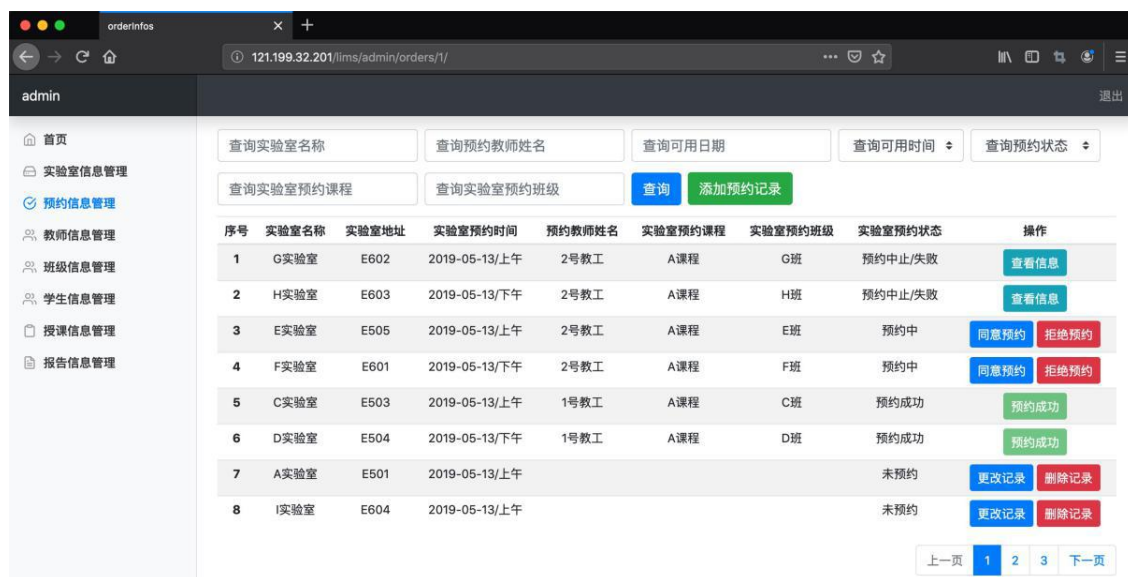


图 5.5 管理员预约信息管理图

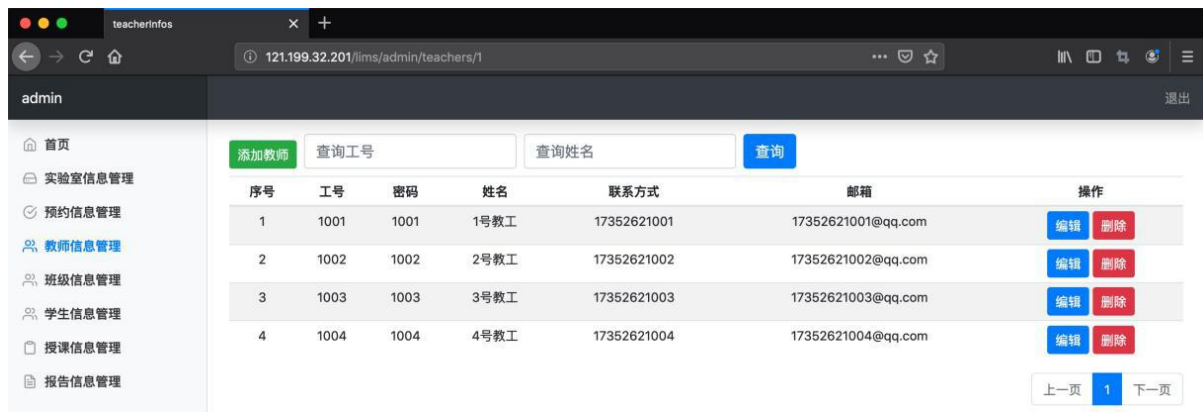


图 5.6 管理员教师信息图

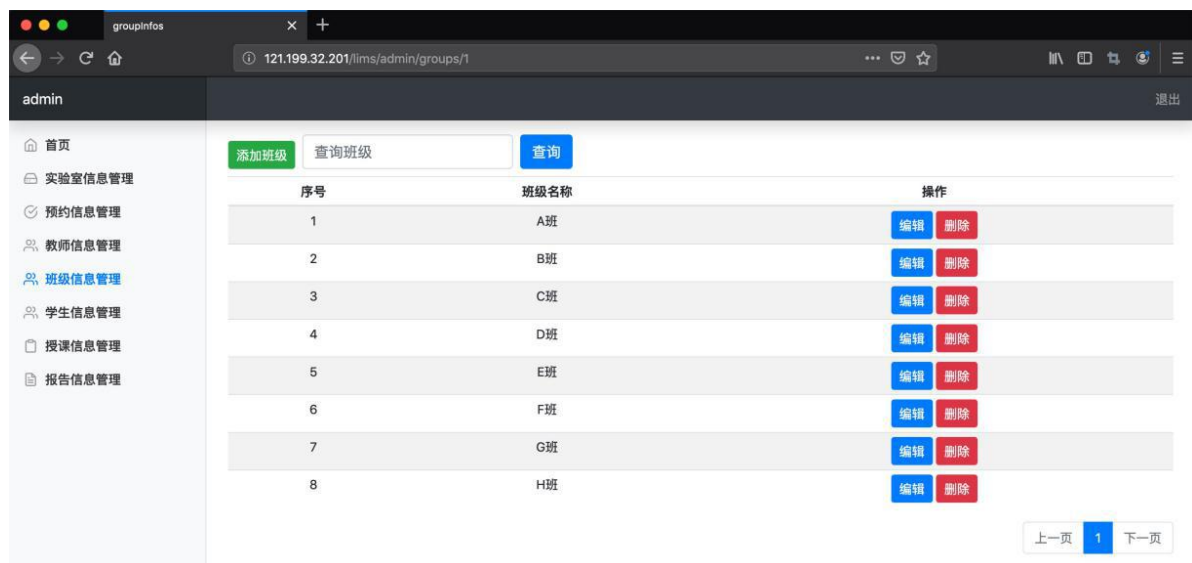


图 5.7 管理员班级管理图

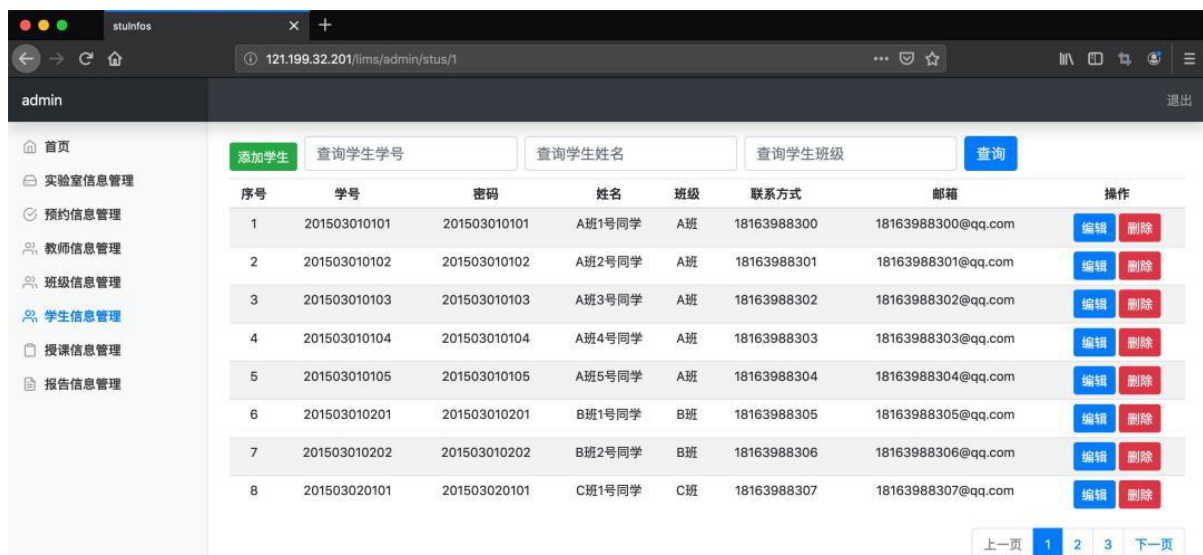


图 5.8 管理员学生信息管理图



图 5.9 管理员授课信息图

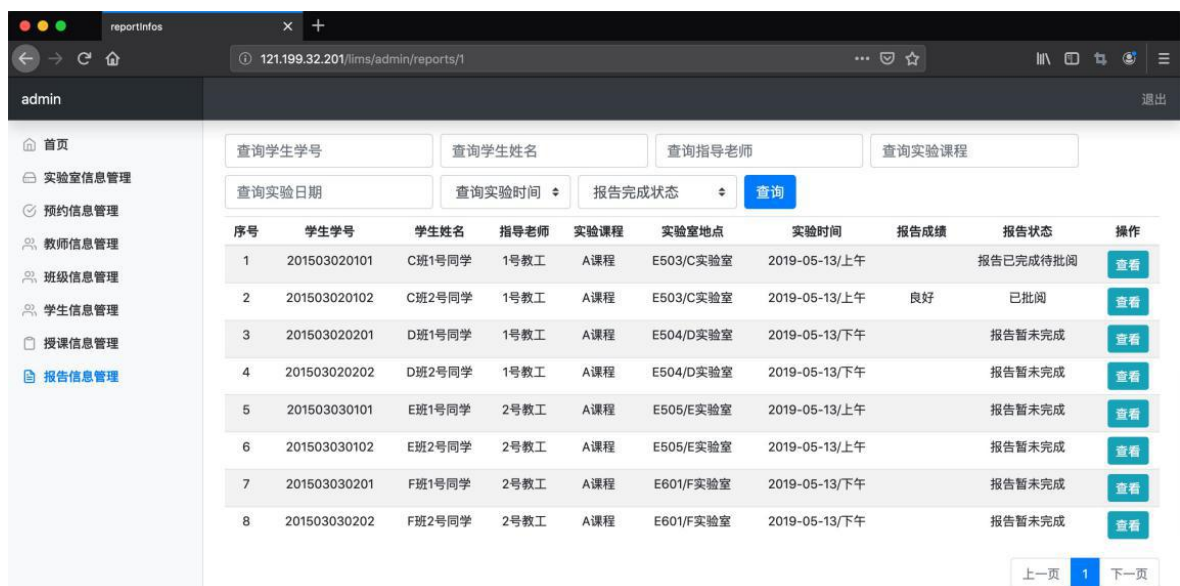


图 5.10 管理员报告信息图

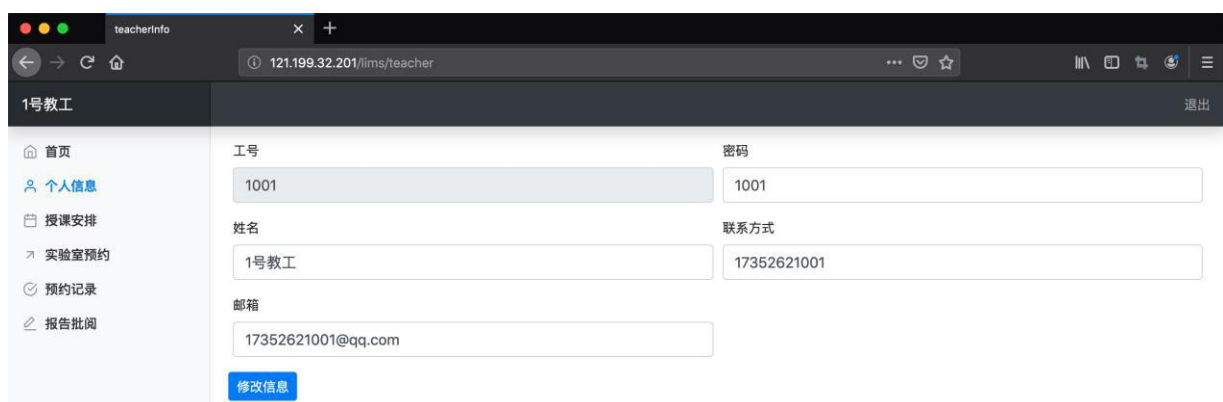


图 5.11 教师个人信息图

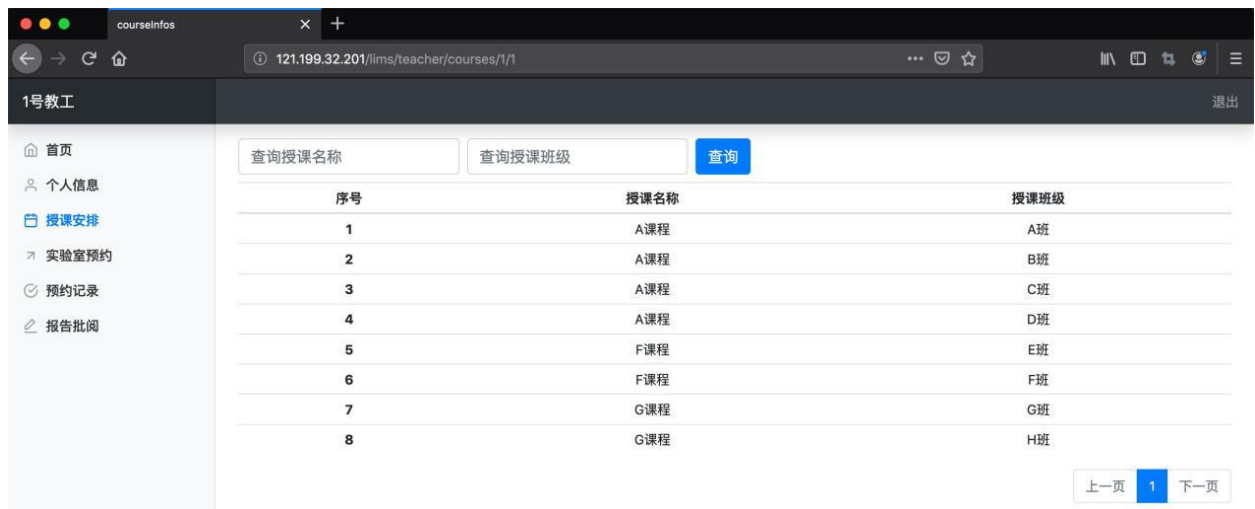


图 5.12 教师授课安排信息图

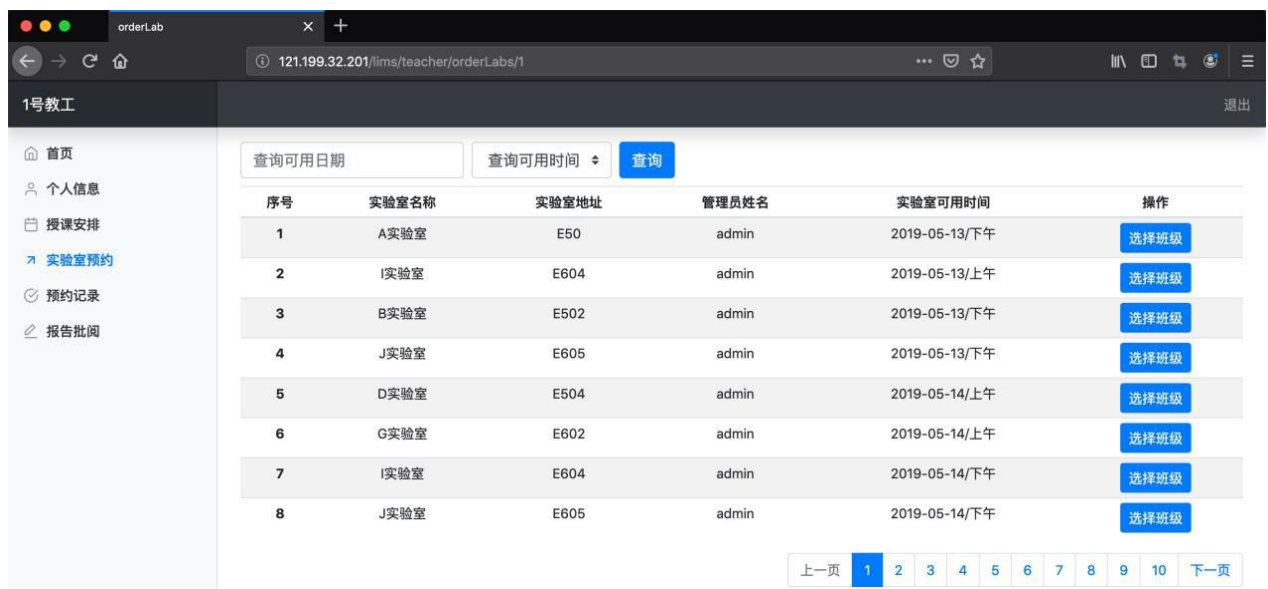


图 5.13 教师实验室预约信息图

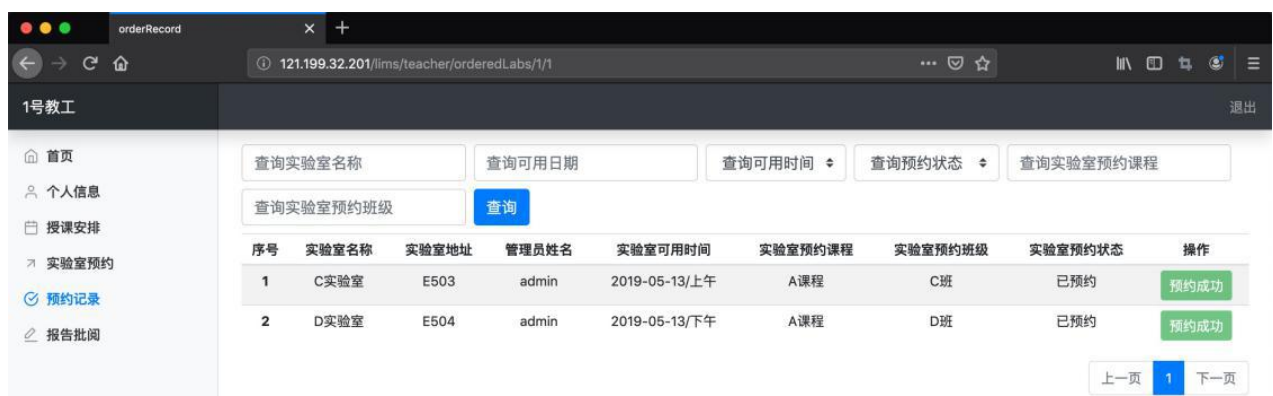


图 5.14 教师预约记录信息图

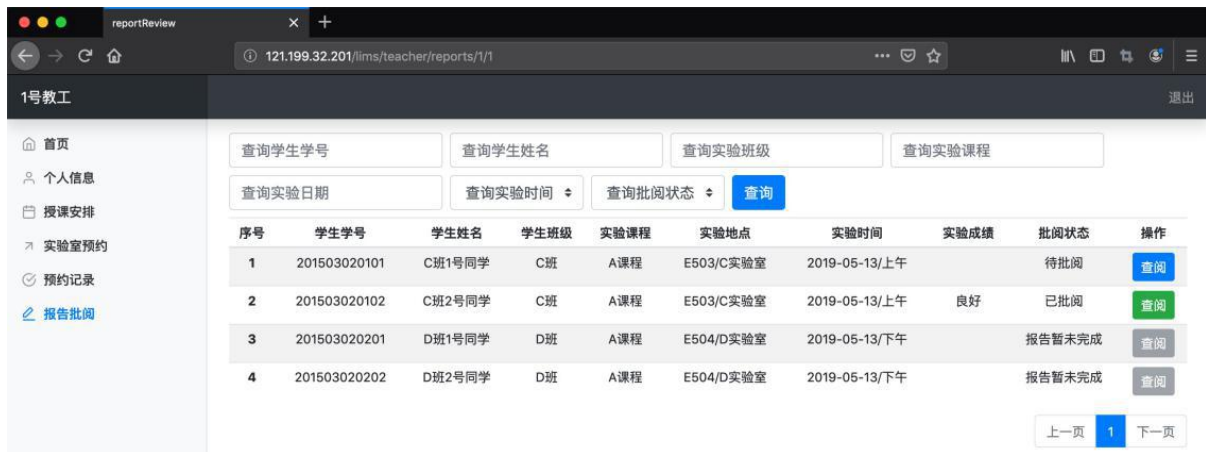


图 5.15 教师实验室报告信息图



图 5.16 学生个人信息图

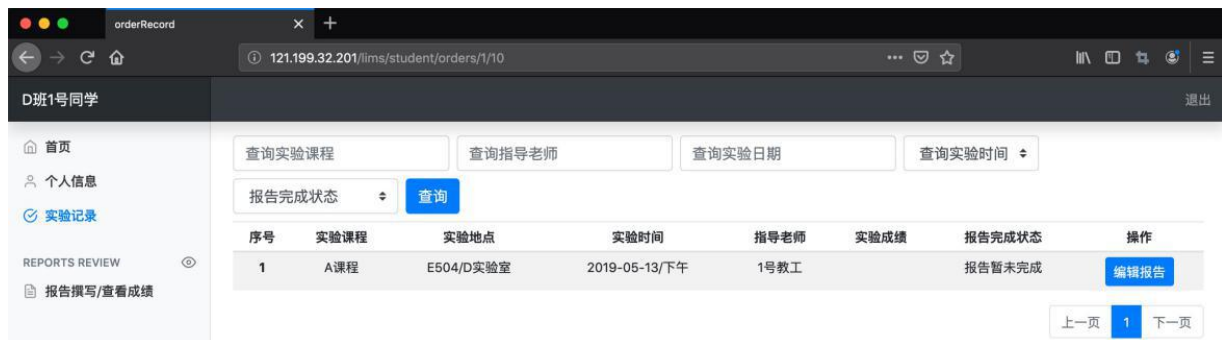


图 5.17 学生报告记录图



### 5.3.3 实验室预约子系统设计与实现

(1) 以管理员身份登录后，在预约信息管理界面中，可以发布新的预约记录，可以修改或删除未预约记录，同意或拒绝预约中的记录，查看预约成功以及失败的记录信息。界面图如图 5.5 所示。

(2) 以教师身份登录后，在实验室预约界面中，可以预约管理员发布的预约记录。在预约记录中，可以查看自己预约中的记录，查看预约成功以及失败记录的信息。界面如图 5.13，图 5.14 所示。

(3) 以学生身份登录后，在实验记录界面中，可以查看教师预约成功的记录。界面如图 5.17 所示。

### 5.3.4 报告批阅子系统设计与实现

(1) 以管理员的角色身份登录后，在报告信息界面中，可以查看所有的报告。该界面如图 5.10，图 5.18 所示。

(2) 以教师的角色身份登录后，在报告批阅界面中，可以查阅所有待批阅的报告，并打分。也可以查阅所以已批阅的报告，修改成绩。界面如图 5.15，图 5.19 所示。

(3) 以学生身份登录后，在实验记录界面中，可以编辑所有未完成的报告。查看所有已批阅的报告成绩。界面如图 5.17，图 5.20 所示。

实验课程	实验地点	实验时间
A课程	E503/C实验室	2019-05-13/上午

班级	姓名	学号
C班	C班2号同学	201503020102

指导老师	课堂成绩	报告成绩
1号教工	85.0	75.0

实验内容

实验步骤/源程序

实验心得

图 5.18 管理员报告查看图

1号教工

实验课程: A课程

实验地点: E503/C实验室

实验时间: 2019-05-13/上午

班级: C班

姓名: C班1号同学

学号: 201503020101

指导老师: 1号教工

课堂成绩: 请录入课堂成绩

报告成绩: 请录入报告成绩

实验内容

实验步骤/源程序

实验心得

录入成绩 取消

图 5.19 教师报告批阅图

D班1号同学

实验课程: A课程

实验地点: E504/D实验室

实验时间: 2019-05-13/下午

班级: D班

姓名: D班1号同学

学号: 201503020201

指导老师: 1号教工

课堂成绩: 请录入课堂成绩

报告成绩: 请录入报告成绩

实验内容

实验步骤/源程序

实验心得

提交报告 取消

图 5.20 学生报告撰写或查看成绩图



## 5.4 本章小结

本章在概要设计的基础上做出了详细设计，将每一个功能细化，具体诠释了每个角色在每一个功能中具体可以做哪些业务流程。并列举了主要功能模块的实现代码及界面。

## 第 6 章 系统的功能模块测试

### 6.1 系统功能模块测试的环境

系统功能模块测试的环境与系统发布的环境相同，系统测试环境配置文件使用 application-test.yml。

### 6.2 系统功能模块测试

#### 6.2.1 登录功能测试

测试内容：（1）角色校验失败；（2）用户名密码校验失败；（3）无权限访问；（4）登录成功。

预期结果：（1）、（2）、（3）返回登录失败信息。（4）进入登录界面。

测试结果：符合预期结果。测试结果如图 6.1，图 6.2，图 6.3 所示。



图 6.1 登录测试（1）



图 6.2 登录测试（2）



图 6.3 登录测试（3）

## 6.2.2 实验室预约功能测试

测试内容：

- (1) 管理员新建 2 条 2020 年 06 月 01 日上午可预约实验室。
- (2) 1 号教师预约 A 课程 A 班和 B 班，取消预约 B 班，中止信息为测试，再次预约 B 班。
- (3) 管理员同意 A 班预约，拒绝 B 班预约。

预期结果：

- (1) 管理员登录后，在预约信息管理界面中有该两条可预约实验室记录。
- (2) 管理员登陆后，在预约信息管理界面中有 2 条预约中记录，1 条预约中止记录；教师登录后，在预约记录界面中有 2 条预约中记录，1 条预约中止记录。
- (3) 管理员预约信息管理界面有 1 条预约中止，1 条预约失败，1 条预约成功；教授预约记录中有 1 条预约中止，1 条预约失败，1 条预约成功；管理员报告信息管理界面新增 5 条报告记录；学生登录后可以看到实验室预约记录。

测试结果：符合预期结果。测试结果如图 6.4，图 6.5，图 6.6，图 6.7 所示。



图 6.4 预约测试（1）



图 6.5 预约测试（2）



图 6.6 预约测试 (3)

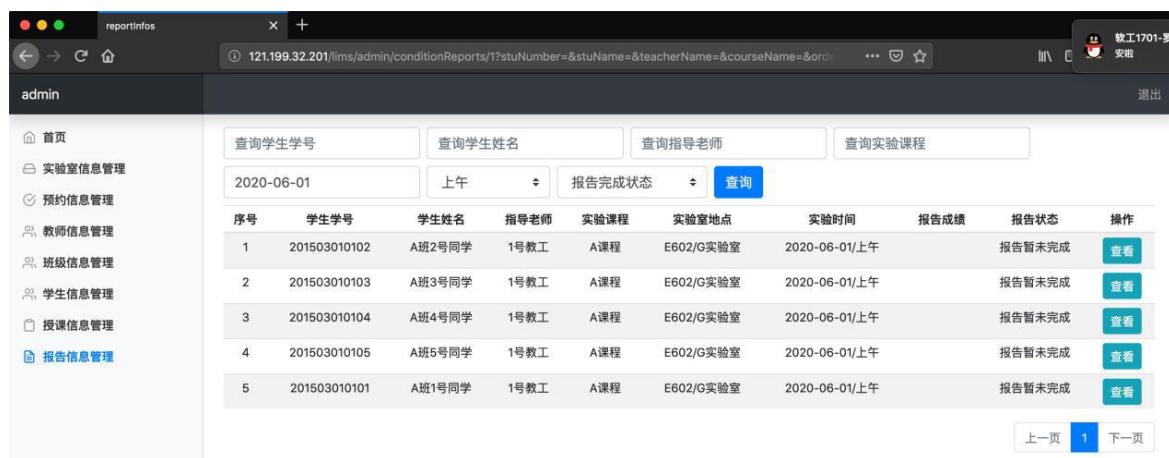


图 6.7 预约测试 (4)

### 6.2.3 报告批阅功能测试

测试内容:

- (1) A 班 1 号学生填写实验报告。
- (2) 1 号教师批阅 1 号学生报告，并给予 80 分课堂成绩，90 分报告成绩。
- (3) A 班 1 号学生查阅成绩。

预期结果:

- (1) 教师登录后，在批阅报告界面显示 A 班 1 号学生已完成待批阅。
- (2) 学生登录后，在实验记录中可以看到报告已批阅，成绩等第为良好。 点击查看成绩，可以看到 80 分课堂成绩，90 分报告成绩。

测试结果: 符合预期结果。测试结果如图 6.8，图 6.9，图 6.10 所示。

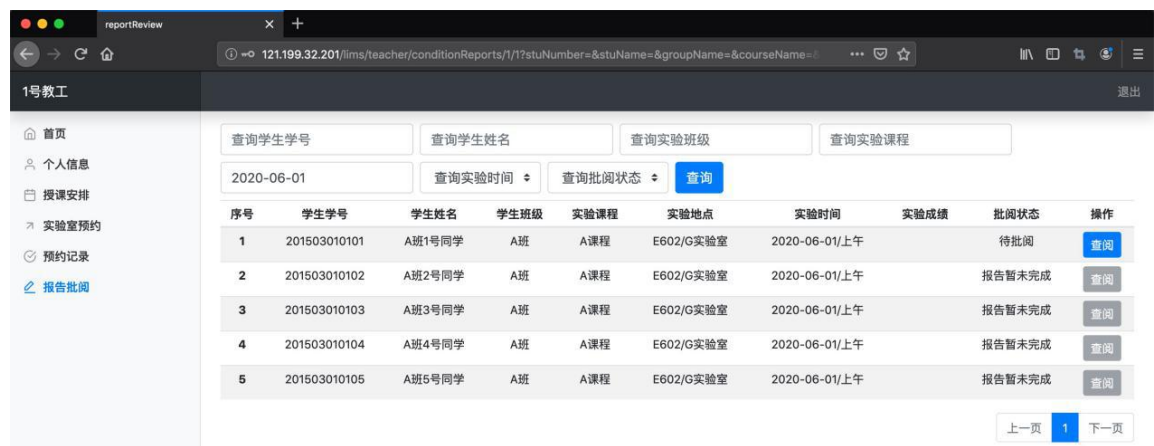


图 6.8 报告测试（1）



图 6.9 报告测试（2）



图 6.10 报告测试（3）

## 6.3 本章小结

本章主要对实验室管理平台的一些主要功能业务逻辑进行测试。第一节介绍了测试的环境和配置。第二节对功能业务逻辑测试进行了详细的描述，包括给出了测试内容，预期结果，以及测试结果图。综上所述可以看出整个系统的程序代码在测试配置环境下均能正常运行，能够基本完成实验室管理平台日常的业务，较好的达到了预期的目标。

## 总结

通过最开始的调查分析、业务需求分析、系统可行性分析、概要设计、详细设计和系统测试等各个环节探讨和分析，终于实现了实验室管理平台这个系统项目。在需求分析阶段明确了系统的用例关系、系统所要实现的业务流程并完成项目的可行性的分析。在概要设计和详细设计阶段，明确了系统在技术上的 MVC 设计模式和三层架构开发模式，进一步的细化功能模块以及完成数据库的设计。通过 Bootstrap 框架以及 Html 和 Thymeleaf 技术完成表示层的界面渲染，最终完成了实验室管理平台系统项目。本次开发的系统项目最主要完成了系统登录子系统，实验室预约子系统以及实验报告批阅子系统。

通过此次系统的开发和研究，本文认为在日益发展的互联网的趋势下，在“互联”这个大时代的带领下，信息化技术将会慢慢将越来越多的实体化经济转换为数字经济，社会的经济效益会越来越高。

高等院校实验室建设是强综合、高复杂的工程，本文的研究仅仅是对高等院校实验室特有的线上登录子系统，实验室信息管理子系统、实验室预约子系统、实验报告批阅子系统等做了详细的设计与实现。目前，高等院校实验室的建设更是具有指纹打卡，动态二维码签到，短信提醒，移动位置定位等高端技术于一体高综合，高复杂工程。在国内外少数高等院校已经实现了指纹打卡，动态二维码签到，短信提醒，移动位置定位等功能。初次之外，更有对已预约的实验采用指纹门禁系统管理模式，只有已预约该实验的同学，才能顺利进入实验室。

## 参考文献

- [1]钟仙. 实验室管理信息系统的设计与实现[D].电子科技大学,2018.
- [2]樊冬梅. 基于 web 的高等学校实验室管理系统的设计与实现[D].青岛大学,2017.
- [3]刘文红. 开放式实验室预约管理系统的开发[D].河北科技大学,2014.
- [4]钱永刚. 艺术类高校实验室管理系统的设计与实现[D].电子科技大学,2013.
- [5]刘定军. 开放性实验室管理系统的设计与实现[D].南昌大学,2010.
- [6]杨海峰,周威佳,贾彬彬,赵洪运.工科高校教学实验室信息化建设和管理研究[J].中国现代教育装备,2018(01):29-31.
- [7]段丙跃. 基于微服务的物联网开放体系架构的设计与实现[D].北京邮电大学,2018.
- [8]辛建平.智慧校园之数字化办公平台设计[J].数字技术与应用,2016(12):144-145.
- [9]潘宁. 培训管理信息系统设计与实现[D].吉林大学,2014.
- [10]卓攀. 独立院校实验室教学辅助系统开发与设计[D].电子科技大学,2011.
- [11]李丹. 基于 B/S 的实验教学平台的设计与实现[D].电子科技大学,2010.
- [12]宋羽韬. 基于 MVC 的学生管理信息系统的设计与实现[D].电子科技大学,2009.
- [13]许雪松,王峥,郭丹丹,王雨辰,康燕燕.动态数据服务发布引擎设计与实现[J].计算机应用与软件,2018,35(07):126-130+177.
- [14]吕鹏. 阿里云上基于规则的数据质量管理系统的设计与实现[D].山东大学,2017.
- [15]杨家炜.基于 Spring Boot 的 web 设计与实现[J].轻工科技,2016,32(07):86-89.
- [16]Joshua Bloch.Effective Java[M].Piscataway,N.J: IEEE Press, 2009
- [17] Roger S. Pressman, Software Engineering: A Practitioner's Approach, Seventh Edition, US: McGraw Hill Higher Education, 2009
- [18] Y.Daniel Liang, Introduction to Java Programming, Eighth Edition, US: Pearson Education, 2010



## 致谢

一转眼的时间，在湖南工程学院的四年的时光悄然离去，自己也到了毕业的时候，四年的时间自己在学校不仅学到了很多专业性的知识，而且学到了很多生活中、工作中处理问题的方法。在这里给四年含辛茹苦的老师 and 亲爱的同学们表示最真诚的感谢。

在论文的结尾，我要特别感谢我的导师，刘学爽老师。我在论文的构思、调研、撰写与完善过程中遇到了很多的困难和障碍，刘学爽老师对我进行了无私的指导和帮助，并且不厌其烦地帮助我对论文进行改进与完善，在刘学爽老师的帮助下这些困难和障碍才得以逐一解决。

感谢所有在我做设计和撰写论文过程中提出宝贵建议，并热情帮助我查找程序缺陷漏洞和订正排版的同学们。

还要感谢湖南工程学院对我的培养之恩，为我提供了良好的学习环境和学习机会。让我在学到很多专业知识的同时，又增长了见识，学会很多做人的道理。这些都将成为我人生的宝贵财富。

最后，我的论文还存在很多不足之处，尤其是理论层次的深入，还望各位专家指正。