



本科毕业设计(论文)

题目：基于 Java 的日程管理系统

院 系： 计算机科学与工程学院
专 业： 网络工程
学 生： 柯贤睿
学 号： 130611106
班 级： 130611
指导教师： 王 辉

2017 年 06 月



本科毕业设计(论文)

题目：基于 Java 的日程管理系统

院 系： 计算机科学与工程学院
专 业： 网络工程
学 生： 柯贤睿
学 号： 130611106
班 级： 130611
指导教师： 王 辉

2017 年 06 月

西安工业大学毕业设计（论文）任务书

院（系）计算机学院 专业 网络工程 班 130611 姓名 柯贤睿 学号 130611106

1. 毕业设计（论文）题目：基于 Java 的日程管理系统设计与实现

2. 题目背景和意义：在日常繁杂的工作中，总是有很多重要日程需要安排，特别是大学生的课程越来越多，而且每个班级同一门课程每天的时间和教室都不固定，给学生和教师的上课带来了一定的麻烦，都需要在特定时间给予醒目标识或者提示，手机日程管理系统就是基于这个原因而做出，使之更好的规划生活，让生活有条不紊。可以随时随地拿出手机查看日程安排，而且手机日常管理系统，还提供到点提醒功能，能够准时完成日程安排。会在行程前通知使用者，而且还方便于用户随时随地的查阅上课的地点与课程名，该系统的使用对象主要是老师和同学。

3. 设计(论文)的主要内容（理工科含技术指标）：以 Java 开发平台为基础实现，主要为用户提供日程的基本管理功能，完成个人一般日程和重要日程安排，该软件需完成如下基本功能：新建日程、删除日程、修改日程、查找日程、删除过期日程、日程类维护。系统要求使用图形化表格，操作简单方便、设置一目了然，可以选择提醒方式。

4. 设计的基本要求及进度安排（含起始时间、设计地点）：大四第二学期 1-18 周于未央校内进行，具体安排如下：

第 1--3 周：需求分析，查阅相关资料，概要设计，完成开题答辩和报告提交

第 4--8 周：分析系统设计任务，并且设计各模块的基本功能框架

第 9--12 周：进行通用模块设计，数据库设计实现，并进行系统调试

第 13--15 周：系统的模块设计，联调，修复

第 16--18 周：实现部分专用功能，撰写毕业设计论文

5. 毕业设计（论文）的工作量要求

① 实验（时数）*或实习（天数）：上机 300 学时

② 图纸（幅面和张数）*：

③ 其他要求：外文翻译字数要求为 3000 左右

指导教师签名： 年 月 日

学生签名: 柯贤睿 2016 年 12 月 28 日
系(教研室)主任审批: _____ 年 月 日

说明: 1 本表一式二份, 一份由学生装订入论文, 一份教师自留。

2 带*项可根据学科特点选填。

基于 Java 的日程管理系统开发

摘 要

日程管理在日常生活中是十分普通的一件事情，人们无论在生活中还是工作中都会有大小、各种各样的事情安排，如果仅仅靠纸张或者自己记录这些事情，往往会遗忘。针对这样的痛点，本文提供了日程管理系统开发的一整套流程，从需求分析到项目架构，再到功能的具体实现，最后测试和使用。希望通过本文能够对日程管理系统的开发作以较全面的总结。

本文章主要以 `thymeleaf` 作为动态模板引擎，替代以前的 JSP。利用 `JQuery` 等脚本语言完成前端页面的设计，综合运用 `mybatis` 框架作为后台数据库链接和 `mysql` 作为数据库提供数据服务，后台采用目前比较流行的 `spring-boot` 和 `spring-MVC` 等开源框架为项目架构，开发出一套 B/S 架构的日程管理系统软件。

根据实际使用情况看，使用本日程管理管理系统，能够有效提高用户的工作效率，加强工作、学习、生活管理，提升了工作质量和人们的日常活动安排效率，促进日常安排顺利进行，为用户提供更加有效的日程管理平台。系统经过了相关测试能够非常方便地运行。

关键词： 日程管理；JavaScript； Node.JS； Mysql

Development of Java - based Scheduling System

ABSTRACT

Scheduling is a very common thing in everyday life. People will have a lot of things in their lives and in work. They are often forgotten if they are only recorded on paper or by themselves. In view of this pain point, this paper provides a set of process development process management, from the needs analysis to the project architecture, to the specific implementation of the function, the final test and use. It is hoped that this article can make a more comprehensive summary of the development of the schedule management system.

This article mainly uses thymeleaf as a dynamic template engine to replace previous JSPs. Using JQuery and other scripting language to complete the front page of the design, the use of mybatis framework as a backstage database link and mysql as a database to provide data services, the background of the more popular spring-boot and spring-MVC open source framework for the project structure, Set B / S architecture of the schedule management system software.

According to the actual use of the situation, the use of this management management system, can effectively improve the efficiency of users, strengthen the work, study, life management, improve the quality of work and people's daily activities to arrange the efficiency of daily arrangements for the smooth, Provide a more effective schedule management platform. The system has been tested for easy operation.

Key Words: Schedule management; JavaScript; Node.JS; Mysql

目录

基于 Java 的日程管理系统开发.....	I
摘 要.....	I
ABSTRACT.....	II
1 绪论.....	1
1.1 前言.....	1
1.2 研究意义.....	1
1.3 国内外研究情况.....	1
1.4 本文的主要研究内容.....	2
2 技术简介.....	3
2.1 系统使用相关技术介绍.....	3
2.2.1 Java 语言简介.....	3
2.2.2 Java 的简单工作原理.....	3
2.2.3 Java 的基本语法.....	4
2.2.4 Java 访问数据库的方式.....	4
2.2 B/S 结构模式的相关介绍.....	6
3 系统需求分析.....	8
3.1 系统运行环境.....	8
3.2 系统模块工作流程.....	8
3.2.1 系统模块设计.....	8
3.2.2 系统主程序流程分析.....	9
3.2.3 用户管理模块流程分析.....	9
3.2.4 日程管理模块流程分析.....	10
3.2.5 日志管理模块流程分析.....	11
3.3 数据库的表关系.....	12
3.3.1 系统 E—R 模型.....	12
3.3.2 关系模型.....	14
3.3.3 数据库物理模型.....	15
3.3.4 表之间关系.....	17
3.4 功能需求.....	17
4 系统设计与实现.....	18
4.1 系统总体设计.....	18
4.2 系统详细设计.....	18
4.2.1 登陆注册界面设计.....	18
4.2.2 系统默认首页面.....	20
4.2.3 展示当前用户所有的日程安排.....	21
4.2.4 日志管理界面设计.....	22
4.2.5 用户信息管理界面设计.....	23
4.2.6 提醒客户界面设计.....	23
4.2.7 查询管理界面设计.....	23
5 系统运行测试.....	25
5.1 用户管理模块系统测试.....	25

5.1.1 登录界面运行测试.....	25
5.1.2 用户信息管理界面运行测试.....	26
5.2 日程管理模块系统测试.....	27
5.2.1 用户日程操作界面运行测试.....	28
5.2.2 用户日志管理界面运行测试.....	28
5.2.3 提醒功能界面测试.....	29
5.3 查询模块系统测试.....	29
6 结束语.....	32
6.1 总结.....	32
6.2 展望.....	错误！未定义书签。
参考文献.....	33
致 谢.....	错误！未定义书签。
毕业设计（论文）知识产权声明.....	34
毕业设计（论文）独创性声明.....	1

1 绪论

1.1 前言

计算机技术在我国飞速发展，计算机应用的推广和普及越来越迅速。而且日程管理是非常重要的，人们不希望每天把大部分时间放在整理和记录琐碎的日常工作事项上，因此用一个日程管理系统来规划每天的工作很有必要。课题基于 J2EE 技术设计并实现了一个日程管理系统，本系统将要实现对个人的日程信息管理，并能按照个人需求，实现系统对事项的管理，从而达到个人能够高效合理的安排自己的工作时间。本文按软件工程的过程记述日程管理系统的设计与开发。当下的社会环境企业对于个人的时间安排和工作效率都是有较高的要求，但是日常的工作中，总是会充斥的各种各样的很繁杂并且比较重要的事情，日程管理系统可以很好地将每一件事情记录并保存，提供给使用者随时查看，这样一来，不仅能够减轻工作的压力，还能保质保量完成工作。基于 WEB 形式的系统展现，在工作的时候，随时能够打开，做记录或者查看任务，也是目前软件行业的提供服务的主流形式。

1.2 研究意义

在计算机刚开始使用的时候，个人的日程安排已经有了明显的效率提升，我们可以通过计算机来记录事件和添加备忘，并且可以通过提醒的方式来让用户及时完成任务。但是由于网络的原因，用户们的工作日程还不能够共享等功能，但是在 20 世纪 90 年代的时候，微软推出了 Office 工具对人们的办公操作提供了很大的帮助，而且现在有很多的日程管理软件都能提供比较好的备忘功能，类似于 Apple 公司的提醒事项，并且在日历等软件中兼并了日程管理的功能。目前在中国很多传统企业对这种日程管理系统的需求很大，很多企业会寻求技术开发公司定制一套属于公司的办公自动化系统。比起以前的人工管理方式，其优越性是显而易见的，所以说，日程管理系统的研究开发是很有必要也很有价值。

1.3 国内外研究情况

目前的日程管理系统有很多很多，前文中提到的微软的 office 工具，outlook 社区给使用者提供共享的功能，Mac 中很好用的 Creative Schedule Mod，功能十分齐全，画面感观也很好，对于日程管理，国外很多公司和个人都很重视，所以在初期的时候就进行开发研究，目前国外受欢迎的 APP 和系统功能上都很全。虽然在中国软件开发行业是近些年才开始迸发，但是发展十分迅速，出现很多的软件供人们使用，类似天天桌面便签、Sticker 桌面便签，这种桌面应用很大程度上为人们提供方便的记录方式，如同便利贴。现在移动设备也很普及，日程管理

软件也是繁多，类似说说日历、calenMob 都是很小的安装软件，让我们随时随地记录，能够合理更改日程安排，从而高效工作生活。

从这么多的软件应用来看，国内外的开发人员能够发现人们的痛点：如何合理高效的安排日程。利用计算机的高性能来帮助人们安排日程，记录重要或者容易被忘记的事项，用这种方式解决日程安排的问题。

1.4 本文的主要研究内容

日程管理系统主要分为两大模块：

日程管理：用户需要先登录系统，成功之后会进入首页面，展示给用户的是当前的日程表，当天如果有日程安排会有提醒用户的标识，用户可以在日程表中选择日期添加日程，或者对已经添加的日程做修改，或者计划改变，删除当前日程；

日志管理：用户每天完成任务之后可以根据需求添加日志，记录每天的工作，对这些工作日志经查询修改，或者删除，并且会有日志列表，根据时间的顺序显示。



图 1.1 功能模块展示图

用户进入系统功能页面之前需要先进行用户的身份确认，登录本系统之后才能进行操作，如果没有身份，可以进行身份注册，创建新用户，进行系统功能的体验。

访问系统，会先进入登录界面，用户输入账号密码进入系统，如果当前用户存在，进行密码验证，通过后进入主界面，主界面展用户的当前的日程信息，如果有最近的日程安排，会有提示标识；用户在主界面进入需要规划日程的日期中进行日程编辑，完毕之后进行保存，之后也可以对已经做好的日程修改或者删除；用户每天完成任务之后也可以进行记录。

2 技术简介

2.1 系统使用相关技术介绍

2.2.1 Java 语言简介

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 面向对象程序设计语言和 Java 平台的总称。由 James Gosling 和同事们共同研发，并在 1995 年正式推出。

Java 分为三个体系：

- JavaSE (J2SE) (Java2 Platform Standard Edition, java 平台标准版)
- JavaEE(J2EE)(Java 2 Platform,Enterprise Edition,java 平台企业版)
- JavaME(J2ME)(Java 2 Platform Micro Edition, java 平台微型版)。

2005 年 6 月，JavaOne 大会召开，SUN 公司公开 Java SE 6。此时，Java 的各种版本已经更名以取消其中的数字"2"：J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2.2.2 Java 的简单工作原理

在 Java 中引入了虚拟机的概念，即在机器和编译程序之间加入了一层抽象的虚拟的机器。这台虚拟的机器在任何平台上都提供给编译程序一个的共同的接口。编译程序只需要面向虚拟机，生成虚拟机能够理解的代码，然后由解释器来将虚拟机代码转换为特定系统的机器码执行。在 Java 中，这种供虚拟机理解的代码叫做字节码(ByteCode) (class 文件的内容)，它不面向任何特定的处理器，只面向虚拟机。每一种平台的解释器是不同的，但是实现的虚拟机是相同的。Java 源程序经过编译器编译后变成字节码，字节码由虚拟机解释执行，虚拟机将每一条要执行的字节码送给解释器，解释器将其翻译成特定机器上的机器码，然后在特定的机器上运行。

Java 虚拟机(JVM)是 Java Virtual Machine 的缩写,它是一个虚构出来的计算机,是通过在实际的计算机上仿真模拟各种计算机功能模拟来实现的。

Java 中，类加载器把一个类装入 JAVA 虚拟机需要经过三个步骤来完成：装载、链接、初始化，其中链接又分来校验、准备、解析过程

装载：查找和导入.class 文件

链接：检查装入.class 文件的正确性，然后，java 虚拟机为变量分配内存，设置默认值

初始化：把符号引用变成直接引用。

2.2.3 Java 的基本语法

一个 Java 程序可以认为是一系列对象的集合，而这些对象通过调用彼此的方法来协同工作。下面简要介绍下类、对象、方法和实例变量的概念。

- 对象：对象是类的一个实例，有状态和行为。例如，一条狗是一个对象，它的状态有：颜色、名字、品种；行为有：摇尾巴、叫、吃等。
- 类：类是一个模板，它描述一类对象的行为和状态。
- 方法：方法就是行为，一个类可以有很多方法。逻辑运算、数据修改以及所有动作都是在方法中完成的。
- 实例变量：每个对象都有独特的实例变量，对象的状态由这些实例变量的值决定。

一个最简单的 Java 程序：

```
package com.tuqi.mappertest;

/**
 * Created by Xianrui Ke on 2017/5/13.
 */
public class HelloWorld { //类名
    public static void main(String[] args) { //主方法
        System.out.println("Hello World"); //方法体
    }
}
```

图 2.1 最基础的 Java 程序展示图

2.2.4 Java 访问数据库的方式

Java 连接各类数据库的方法是类似的，本系统开发采用 MySQL，叙述链接 mysql 的方法流程。

a. 首先导入需要的工具

连接 MySQL 所需的驱动包，可以选用 mysql-connector-java-5.1.26-bin
Java 编辑器，常用的是 Eclipse、IDEAJ、MyEclipse

引入 java.sql 包（作用：提供使用 Java 编程语言访问并处理存储在数据库中的数据 API）

```
import java.sql.*;
```

加载 JDBC 驱动程序

将下载好的 jar 包 mysql-connector-java-5.1.26-bin 导入至项目

```
加载驱动 Class.forName("com.mysql.jdbc.Driver")
```

b. 产生 Connection

如已成功加载 JDBC 驱动程序，就可以利用加载的驱动程序连接数据库

```
Connection con=DriverManager.getConnection(URL,UserName>Password);
```

URL: JDBC:(subprotocol):(subname)

subprotocol:子协议指定连接何种数据库或用什么方式连接数据库;

subname:确立一个连接,可以是一个数据源名,也可是指向一个网上数据库.

经过上述三个步骤,就已完成了 MySQL 数据库的连接工作,通过相应的 sql 语句可以对数据进行相应的操作了~

下面放出一个简单的 Demo, 进一步了解 JDBC 连接 MySQL 的操作。

DBManager 类, 用于连接访问数据库, 并写好了对数据进行增删改查的接口:

附注: 一个标准的 JDBC 程序

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
public class TestJDBC {
```

```
    public static void main(String[] args) {
```

```
        Connection conn = null;
```

```
        PreparedStatement pStmt = null;
```

```
        ResultSet rs = null;
```

```
        try {
```

```
            // 加载 MySQL 驱动类
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            // 创建数据库连接,取得数据库连接对象
```

```
            conn
```

```
=
```

```
            DriverManager.getConnection("jdbc:mysql://hostname/databaseName", "root", "root");
```

// 取得 pStmt 对象,PreparedStatement 比 Statement 效率高, 安全性好 (预编译的 SQL 语句, 防止 SQL 注入)

```
        pStmt = conn.prepareStatement("SELECT * FROM user");
```

```
        // 取得 ResultSet 对象, 暂时存放查询取得的结果
```

```
        rs = pStmt.executeQuery();
```

```
        while (rs.next()) {
```

```
            System.out.println(rs.getInt("id"));
```

```
        }
```

```
    } catch (ClassNotFoundException e) {
```

```
        e.printStackTrace();
```

```
        System.out.println("找不到对应的驱动类");
```

```
    } catch (SQLException e) {
```

```

        e.printStackTrace();
        System.out.println("数据库连接失败");
    } finally {
        try {
            //关闭结果集
            if(rs != null) {
                rs.close();
                rs = null;
            }
            //关闭声明
            if(pStmt != null) {
                pStmt.close();
                pStmt = null;
            }
            //关闭连接对象
            if(conn != null) {
                conn.close();
                conn = null;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}

```

2.2 B/S 结构模式的相关介绍

B/S 结构是 Browser/Server 的缩写，即浏览器和服务器的结构。随着互联网技术的发展和用户需求的提高，C/S 结构的弊端渐渐显现出来。客户端需要安装固定程序进入局域网中才可进行数据交换，无法面向广大用户，使得 C/S 结构适用范围变小且用户群体较为固定。另外，维护成本高，需专门的维护人员定期维护，每进行一次升级，客户端所有程序都需要改变。针对 C/S 结构的这些问题 B/S 结构应运而生，在 B/S 结构中，用户无需专门安装特定程序，只需要通过 web 浏览器即可完成简单事务处理，减轻了客户端电脑复核，减轻了系统定期维护的成本和工作量。又由于 B/S 结构直接放在广域网中，仅需通过一定的权限管理就可实现多用户访问，有较强的交互性和及时性。

目前，软件系统及时升级改进的需求越来越频繁，B/S 结构的产品的更为方便优势逐渐显现出来。因浏览器客户端的便捷性，B/S 结构的系统维护只需进行

服务器的管理即可，用户数量的扩大和信息的增多，都不会增加升级维护的一点工作量。B/S 结构的广泛运用是信息化发展的主流方向，对于提升工作效率和人力物力的节省有很大帮助。更便捷的一点是，B/S 结构软件的开发对服务器操作系统的选择是多元化的，无论选择哪种操作系统都不会影响到用户使用。这样，在选择服务器操作系统就可以选择成本较低的，来降低整体开发费用。

3 系统需求分析

3.1 系统运行环境

客户端:CPU 133MHz 以上,内存 32MB 以上,操作系统 WIN95 以上,Google chrome。

数据库服务器端: CPU 1.5GHz 以上, 内存 1GB 以上, 操作系统 Linux, VMware workstation 10, 数据库 MySQL。

3.2 系统模块工作流程

3.2.1 系统模块设计

结合对相类似的日程管理软件的分析, 按各业务将系统划分成用户管理、日程管理、日志管理等三个模块, 该系统软件的模块

日程管理管理系统分为三大模块

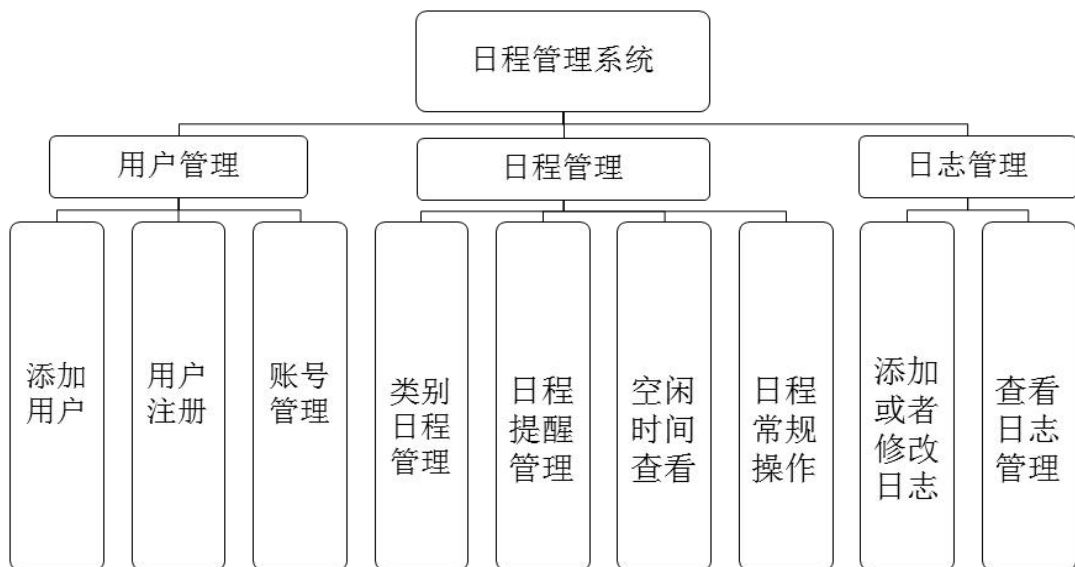


图 3.1 系统管理模块图

a.用户管理模块

进入本系统需要用户进行用户信息验证, 如果本系统中没有此用户信息, 需要先进行用户注册, 注册成功之后就可以登录本系统, 已经存在用户如果忘记密码, 可以请求管理员帮助进行账号管理, 对密码进行修改。本系统对用户权限没有做强制要求, 使用的用户可以自主创建账号, 进入本系统的操作页面, 对自己的日程安排进行管理。

3.2.2 系统主程序流程分析

用户在登录界面输入用户名和密码成功登陆之后,即可进入系统主页面自行选择相应模块进行操作,系统的主程序流程图如图 3.2 所示:

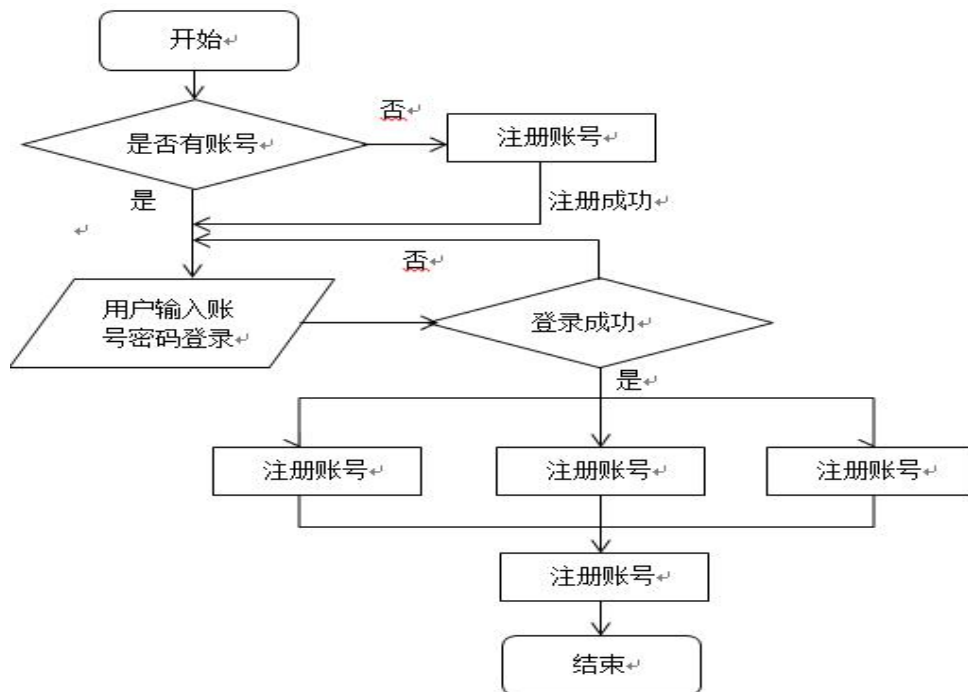


图 3.2 系统主程序流程图

3.2.3 用户管理模块流程分析

用户管理模块的添加用户和修改密码功能在系统登陆页面实现,进入系统之后的用户管理模块需要使用管理员权限操作,可对存在用户的信息进行查看和修改。用户管理模块的具体工作流程图如图 3.3 所示

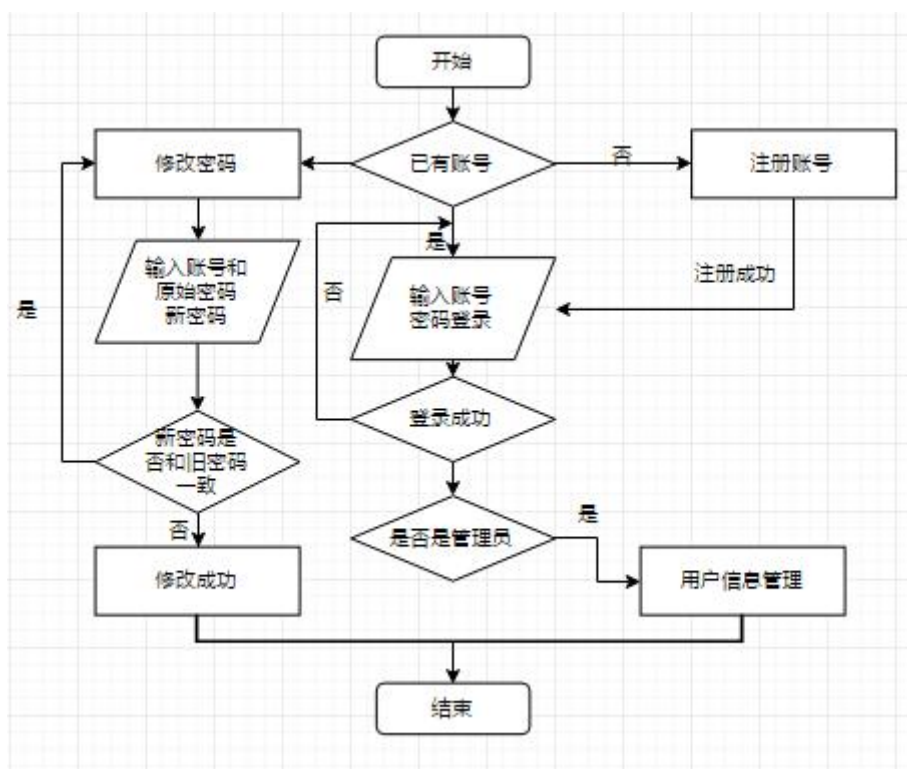


图 3.3 用户管理流程图

由图 3.2 可以看出用户管理主要流程如下

用户进入登录页面判断是否有已有账号，已注册用户直接输入用户名和密码进行登陆，新用户进入注册界面输入新用户名和密码后确认注册，返回登录页面进行登陆。

若已有账号可进入修改密码界面，输入用户名、原密码、新密码。若输入的两次原密码和新密码相同，则需重新输入，不同则系统提示修改成功。

管理员用户输入用户名密码登录成功后，普通用户的账号信息可以由管理员进行修改操作。

3.2.4 日程管理模块流程分析

流程模块是本系统的核心模块，成功登录本系统的用户，能够进入到系统的主界面，页面会默认展示当前月份的所有日期，每一个日期就相当于一个容器，可以由用户自主安排时间，添加日程，并且同一日期内可以添加多条日程安排，系统会检测当前用户的日程安排，在事件发生的前 30 分钟前会提醒用户将要进行的日程安排。

对日程管理模块具体工作具体分析后，得出如图 3.4 所示的日程管理模块流程图。

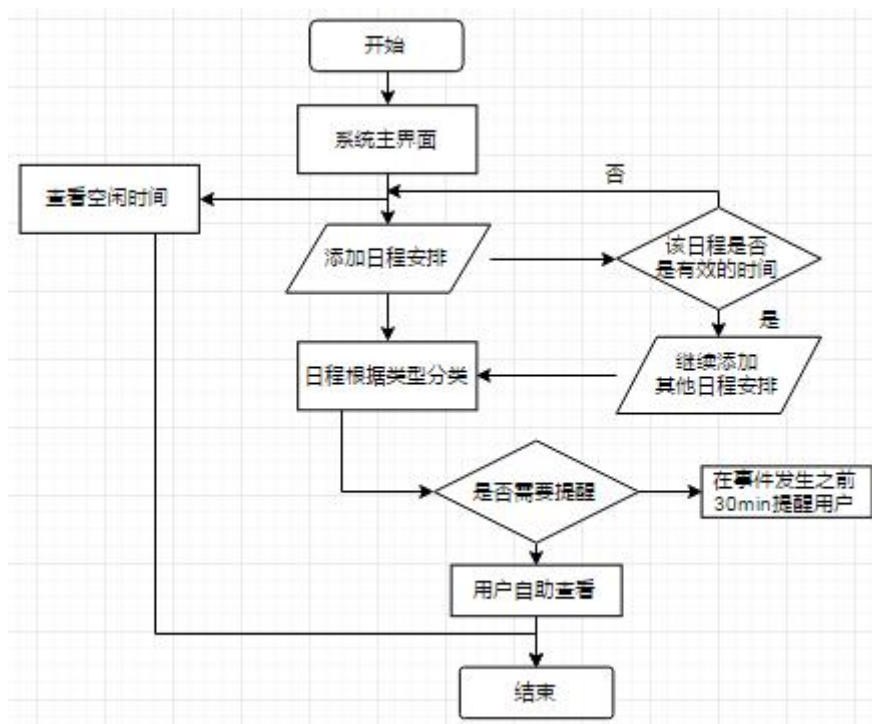


图 3.4 日程管理模块流程图

由图 3.4 可以看出停电管理模块流程如下

a.添加日程安排

在某一天的框内选择添加日程安排，进入日程添加安排的界面，按照要求输入相关信息，并且选择日程安排类型（工作、会议、聚会等等），根据不同的类型将日程归类划分。

b.添加日程可能存在的问题

可能存在同时间点会有多项事件存在，这种情况是比较不合常理的，所以在添加日程的时候会对操作检测，如果是这种存在冲突的事件，需要重新操作。

c.日程提醒

存在某些事件需要提醒的，尤其是会议或者是重要的聚会，很可能被用户遗忘，所以事件提醒功能是比较要的，所以在本系统中添加了日程提醒功能，系统会在事件发生的前 30 分钟给用户发出提醒，目前是在站内提醒，另外比较好的提醒方式是手机短信提醒，或者比较流行的通讯软件。

d.用户查看自己的空闲时间

除开了添加了日程安排的其他空余时间，这个功能的目的在于能够更加合理的使用时间，用户可以将某个时间点冲突的事件分开安排，大大提高了时间的利用率。

e.用户查看安排

用户也能够自主查看所有已经添加的日程安排。

3.2.5 日志管理模块流程分析

本模块是为了给用户记录提供方便，例如今天完成了一项比较重要的事

情，可以做一个总结，写一份博客或者是总结性文章，存储在数据库中，利于用户记录自己的工作历程；或者是用户在解决了一个比较重要的问题之后，可以用日志的形式记录下来，当再次遇到类似的问题之后，也能够很快找到解决方案，利于提高工作，学习的效率。

对日志管理模块的具体工作进行用户端逻辑分析，得出如图 3.5 所示的查询管理流程图。

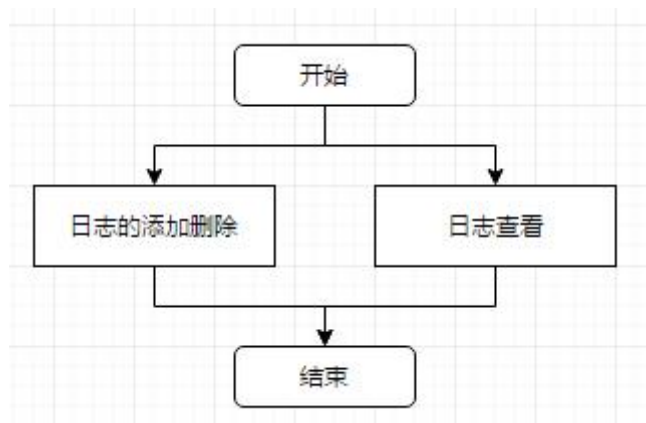


图 3.5 日志管理模块流程图

由图 3.5 可以看出查询管理模块流程如下

(1) 日志管理类似于博客系统，但是不需要评论的功能展示，仅提供给用户添加记录就可以，所以最基础的添加日志功能、修改已经存在的日志、删除现有的日志，本系统采取逻辑删除，不会彻底删除掉记录，这样可能方便找回用户误删或者丢失的资料。

(3) 用户在可以自己查看想要查看的日志。

3.3 数据库的表关系

3.3.1 系统 E—R 模型

本系统操作人员依次在网页填写计划停电申请表，最终生成报表，审核人员可以对申请表内容进行审批，不通过需要重新填写申请表。

系统全局 E-R 设计如图 3.6 所示

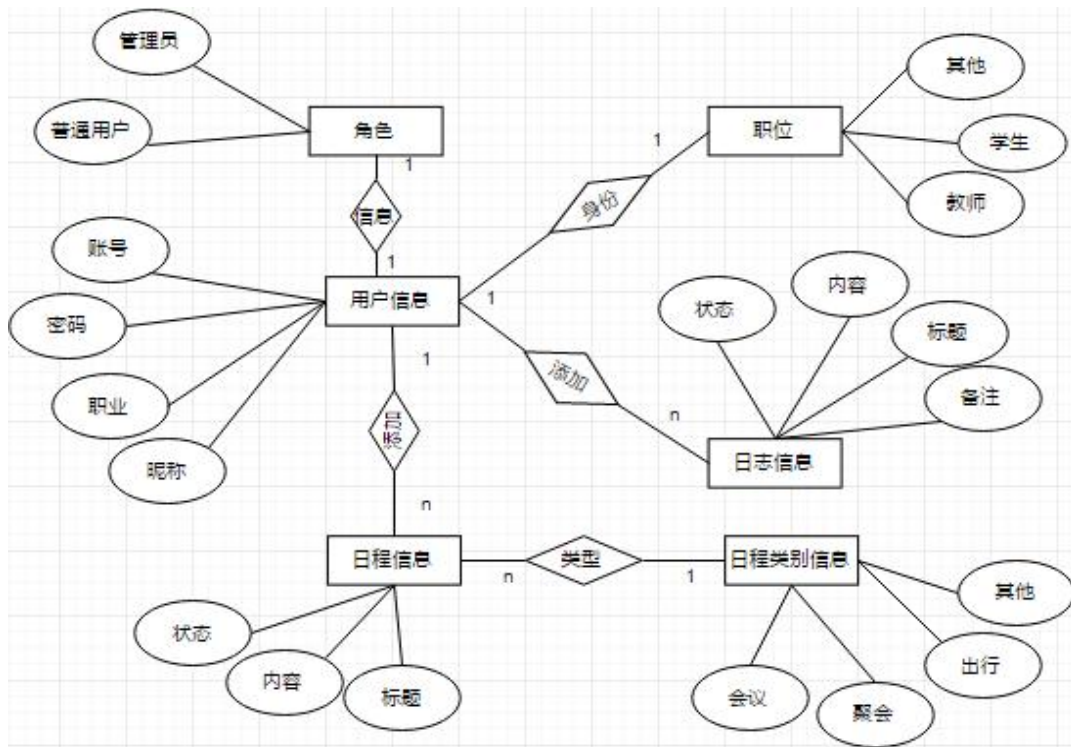


图 3.6 管理系统整体 E-R 图

结合数据库的创建标准和常规建表操作，构造的系统用户信息实体应包含以下几个属性：用户名，登录密码，用户昵称。用户信息实体图如下图 3.7 所示。

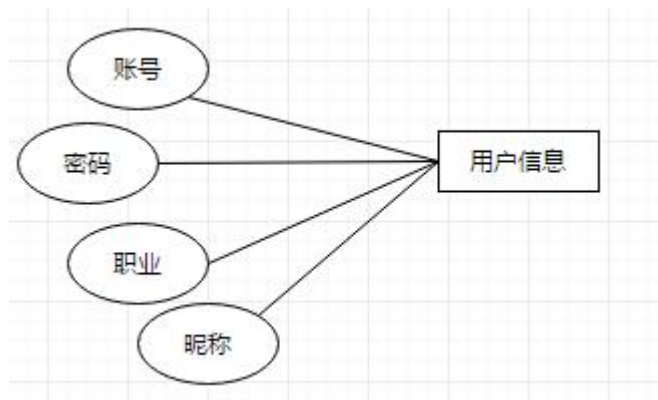


图 3.7 用户信息实体图

日程信息包含属性：日程标题、日程内容、状态信息。日程信息实体图如下图 3.8 所示。



图 3.8 日程信息实体图

因为存在日程的分类，所以有日程类别信息，包含属性如下：编号、类别名称。完整的日程类别信息表实体图如下图 3.9 所示。

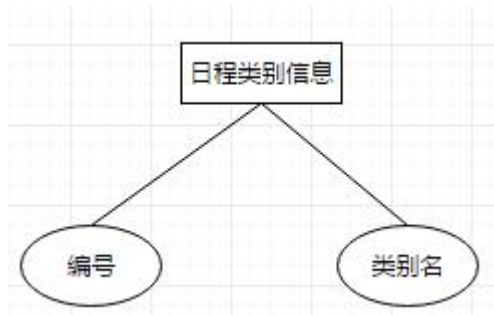


图 3.9 日程类别信息实体图

日志信息表属性：日志标题、日志内容、日志备注、日志状态。日志信息实体图如下图 3.10 所示。

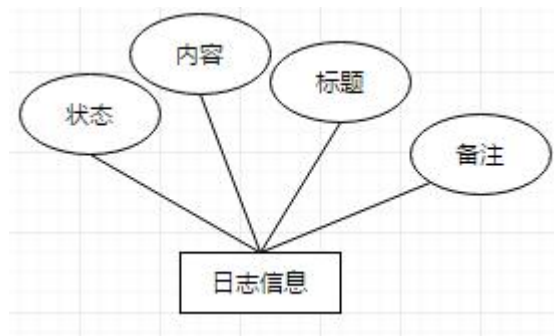


图 3.10 日志信息实体图

职位信息表属性：编号、职业名。日志信息实体图如下图 3.11 所示。

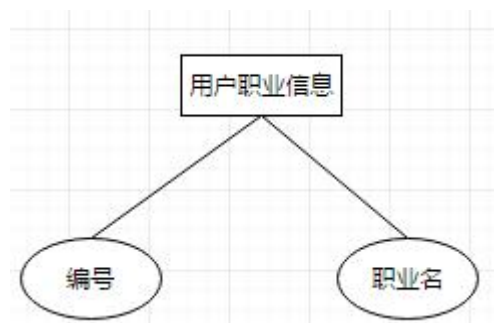


图 3.11 用户职业信息实体图

3.3.2 关系模型

将系统 E-R 图转换成关系模型如下：

系统用户（用户名，登录密码，用户昵称，用户职业，登录时候用昵称，用户注册的时候如果没有添加昵称，则默认用用户名作为昵称，编号是自增，职业是用来区别不同的职业对日程安排不同）；

日程信息（日程标题、日程内容、状态信息，一条日程必定对应了一个用户，

所以添加日程信息必须校验该用户是否存在，该表中有用户 id 作为外键）；

日程类别信息（编号、类别名称）；

用户职业信息（编号、职业名）；

日志信息（日志标题、日志内容、日志备注、日志状态，标题能够让用户快速检索，备注信息是要求在一定的字数之内，可以对日志做一个简介概括，状态值记录该条日程是否还有效，如果执行删除操作采取逻辑删除）。

3.3.3 数据库物理模型

经过系统需求分析并结合系统的数据结构，严格遵循数据库设计的原则，本日程管理系统共设计五张数据表：

a.user 表为用户信息表

用来存储登录用户的用户名和密码信息，用户身份，设有以下字段：user_id（用户编号，自增长），user_name（用户名），password（密码），user_nick_name（用户昵称），gmt_create（数据插入时间），其中 user_id 为主键，采用自增的方式自动生成，递增量为 1。详见表 3.1

表 3.1 用户信息表（User）

字段	类型	NULL	主键	备注
user_id	bigint	否	是	用户编号、自增
user_name	varchar（45）	否		用户名称
user_nick_name	varchar（45）	是		用户昵称
password	varchar（100）	是		用户密码
gmt_create	datetime	是		创建时间

b.user_position 表为用户职业信息表

用于存储大致的职位信息，设有以下字段：position_id（执业编号），position_name（职业名称），gmt_create（创建时间），gmt_modified（修改时间），其中 position_id 为主键，采用自增 1 的方式自动生成。详见表 3.2

表 3.2 用户职业信息表（user_position）

字段	类型	Null	主键	备注
position_id	int(11)	否	是	职业编号、自增
position_name	varchar（45）	否		职业名称
gmt_create	datetime	是		创建时间
gmt_modified	datetime	是		修改时间

c.programme 表为日程信息表

存储了添加的日程详细信息，设有字段 programmeId（日程编号），content（日程内容）status（日程状态信息：0-该日程已经被删除，无效；1-该日程正常记录，有效）programme_time（日程发生时间）program_user_id（该日程所属用户，即该条记录的用户编号）program_type_id（日程类型编号，对应的是 program_type 表中的编号，gmt_create（该条日程的添加时间）gmt_modified（本条记录最后一次的修改时间），其中 programmeId 为主键，采用自增 1 的方式自

动生成。详见表 3.3

表 3.3 日程信息表 (programme)

字段	类型	Null	主键	备注
programmeId	bigint (20)	否	是	自增
content	varchar(300)	是		日程内容
status	varchar(5)	否		日程状态 (删除/存在)
programme_time	datetime	是		日程进行时间
program_user_id	Bigint(20)	否		所属用户
program_type_id	Int(20)	否		日程类型
gmt_create	datetime	是		创建时间
gmt_modified	datetime	是		修改时间

d.program_type 表为日程类别信息表

用来存储日程的类别信息，设有字段 prog_type_id (日程类别编号)，prog_type_name (日程类别名称)，gmt_create (该条日程的添加时间) gmt_modified (本条记录最后一次的修改时间)，其中 prog_type_id 为主键，采用自增 1 的方式自动生成。详见表 3.4

表 3.4 日程类别信息表 (program_type)

字段	类型	Null	主键	备注
prog_type_id	int(11)	否	是	自增
prog_type_name	varchar (45)	否		类型名称
gmt_create	datetime	是		创建时间
gmt_modified	datetime	是		修改时间

e.daily_record 表为日志信息信息表

用来存储所有的日志信息，设有字段 recordId (日志编号)，title (日志标题)，content (日志内容)，status (日志状态，逻辑值和日程表中相同)，remarks (备注信息，可以给文章作以简介)，gmt_create (该条日程的添加时间) gmt_modified (本条记录最后一次的修改时间)，其中 recordId 为主键，采用自增 1 的方式自动生成。详见表 3.5

表 3.5 日志信息表 (daily_record)

字段	类型	Null	主键	备注
recordId	bigint (20)	否	是	自增
title	varchar (45)	否		日志标题
content	text	是		日志内容
status	varchar(5)	否		日志状态
remarks	varchar(20)	是		备注
gmt_create	datetime	是		创建时间
gmt_modified	datetime	是		修改时间

3.3.4 表之间关系

日程管理系统使用的是灵活性高速度快捷的 MySQL 数据库，作为一种关联性数据库管理系统拥有开放源码的功能使其拥有庞大的用户全体，它的核心线程完全是多线程，支持多处理器，所以说效率很高。数据表类型很丰富，多种类型字段，操作简便，存取迅速，在开发团队的不断探索下，使其灵活性和简便性达到最佳平衡。

对日程管理系统的功能进行分析，该系统用户只用分为两个角色：普通用户和管理员，不涉及权限管理，建立 user 表作为用户信息表，用来存储所有的用户信息，主要是用户账号信息，表 user_position 是用户角色表，创建 programme 表记录日程安排信息，另外创建 program_type 表作为事件类型的信息、创建 daily_record 表存储所有的日志数据。

结合本系统的功能需求及数据结构，建立出的系统数据表如下表 3.6 所示：

表 3.6 数据库结构

数据表名称	数据表备注	数据表用途
user	用户信息表	存储用户信息
user_position	用户职业表	用户职业信息
programme	日程表	存储日程信息
program_type	日程类型表	日程类型信息
daily_record	日志表	存储所有日志

3.4 功能需求

在日常繁杂的工作中，总是有很多重要日程需要安排，举例来说大学生的课程越来越多，而且每个班级同一门课程每天的时间和教室都不固定，给学生和教师的上课带来了一定的麻烦，都需要在特定时间给予醒目标识或者提示，日程管理系统就是基于这个原因而做出，使之更好的规划生活，让生活有条不紊。可以登录日程管理系统查看日程安排，或者说某段时间工作上会很忙，往往这种时候，越容易忘记事情，所以日程管理系统能够帮助人们合理安排时间，该系统的使用对象可以是学生，在职工作人员，较为宽泛。并且除了日程管理之外，系统还会提供日志的记录功能，用户在工作完之后做一个小的工作总结，存储在数据中。系统中的各种信息都由数据库进行管理，管理员管理用户信息，帮助用户账户管理，从而使系统能及时更新得到维护，人机操作更加友好。

日程管理系统主要的功能如下：

(1) 完成系统用户的登录。日程管理系统在设计中通过密码校验来控制用户的登录状况，因为不涉及敏感内容，所以系统的权限控制暂且不需要太严格的控制。

(2) 成功登录的所有用户都可以使用本系统的功能，按照需求添加日程安排。

(3) 用户添加日程之后，系统会对用户进行日程安排提醒，帮用户记住时间。

(4) 日志管理，用户记载本日的工作记录，添加日志，删除和修改。

4 系统设计与实现

4.1 系统总体设计

日程管理系统侧重于用户操作方便，使用起来高效便捷，不需要复杂权限管理或者业务需求管理，本系统的核心功能就是日程的管理、分类，最终的目的是时间上的合理安排；用户管理主要是对用户信息的查看或者修改，权限只去别区普通用户和管理员。对于本系统，对用户日常时间管理需求认真分析，利用 J2EE 技术和 node.JS 技术结合开发出 web 形式的管理系统。

4.2 系统详细设计

系统在数据库方面使用数据传输与数据存储量属于中等水平，小用户量的情况下，使用 MySQL 来进行数据存储，在开发方面使用 spring-boot 框架，开发效率较高，且可以避免很多配置过程，较快进入开发阶段。

同时本系统在硬件方面，只需要一台连接着 Internet 的 web 服务器即可。

利用 Java 开发的过程中，考虑到本系统针对不同用户群可能有不同的使用习惯，因此在 user 表中加上了 user_position_id 属性，目前设置有 6 个具体的用户身份（医生、律师、学生、教师等）和一个模糊群体：其他用户。这种分别可以根据用户群体的不同培养不同的使用习惯，让系统更加人性化。

日程管理方面，也考虑到会有不同类型的日程安排，所以在 programme 表中添加了 program_type_id 字段，关联了 program_type 表，添加日程的时候会有类型选择的选项，现有的分类有：1-普通 2-出行 3-会议 4-约会 5-纪念日 6-其他，序号对应的不同的类型，根据不同类型的日程信息，用户可以更加明确的安排和管理自己的日程，日程的有效管理使用的是 status 字段管理，如果用户想要删除某一条记录，使用逻辑删除，status 属性是整形数字 1，如果被删除了，字段信息被修改成 2，数据库中本记录不会被删除。

本系统中还有日志管理模块，日志的功能操作是为了配合日程管理，让用户的试用体验更佳，用户可以在本日工作完成、或者是和他人约会之后添加一份日志记录，方便对之前发生的事件查看。

4.2.1 登陆注册界面设计

用户输入本系统的访问链接（<http://10.148.17.130:8080>），进入到系统登录页面，在登录界面用户按照操作输入个人的用户昵称和密码。点击登录之后，后台服务器会根据页面传递的数据对用户账号和密码进行判断。登录会返回 3 种结果，分别为：登录成功，用户不存在，密码错误。

登录页面如图 4.1 所示



The login interface features a dark gray background. At the top, the title '登录到你的账户' (Log in to your account) is displayed in white. Below the title are two white input fields: the first is labeled '用户名' (Username) with a user icon, and the second is labeled '密码' (Password) with a lock icon. At the bottom left, there is a checkbox labeled '记住我' (Remember me). At the bottom right, there is a blue button labeled '登录' (Login) with a right-pointing arrow icon.

图 4.1 登录界面

如果当前用户在本系统的没有账号，可以跳转到注册页面注册本系统的账号，注册时需要用户提供用户名，设置初始密码，昵称和选择职位。如果用户昵称已经被他人使用，那么用户需要重新输入一个昵称。

注册页面如图 4.2 所示：



The registration interface has a dark gray background. The title '注册你的账户' (Register your account) is at the top in white. Below the title are four white input fields: '用户名' (Username) with a user icon, '昵称' (Nickname) with a user icon, '密码' (Password) with a lock icon, and '确认密码' (Confirm password) with a lock icon. Below these fields is a dropdown menu currently showing '学生' (Student). At the bottom left is a blue button labeled '重置' (Reset) with a right-pointing arrow icon, and at the bottom right is a blue button labeled '注册' (Register) with a right-pointing arrow icon.

图 4.2 用户注册界面

用户如果忘记密码可以选择让用户管理员修改，也就是让管理员协助修改密码，用户不能自行修改密码，因为系统中使用了 MD5 对密码加密，所以无法

查看到明文的密码。

这里是对密码的加密过程，使用户密码不会直接暴露给其他人，如果用户忘记了初始密码就需要进行修改。

修改密码页面如图 4.3 所示：

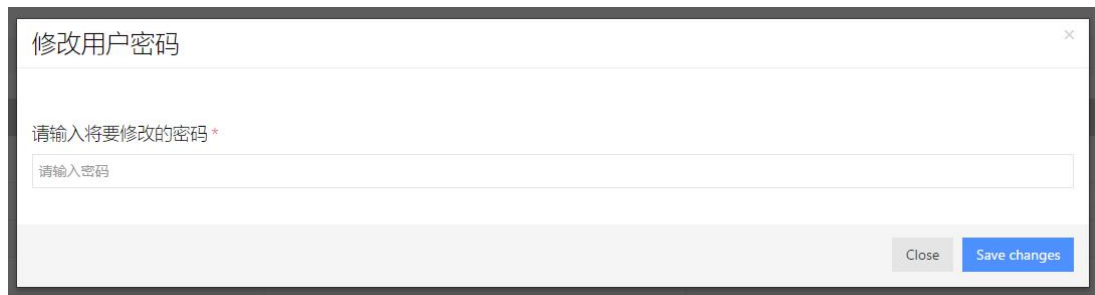


图 4.3 用户修改密码页面

4.2.2 系统默认首页面

用户成功登录之后，会跳转到首页面，当前页面显示的是当前月份的日期，相当于一个日历，每一个日期当中会有添加日程的选项，为了美观，默认情况下是没有显示这个按钮，当鼠标覆盖到某一个日期的格子中后，就会有添加按钮，点击添加按钮，用户按照要求添加日程信息，访问 `ProgrammeController` 控制器的 `addProgramm.do` 方法，执行添加日程操作逻辑。

首页显示如下图 4.4 所示：

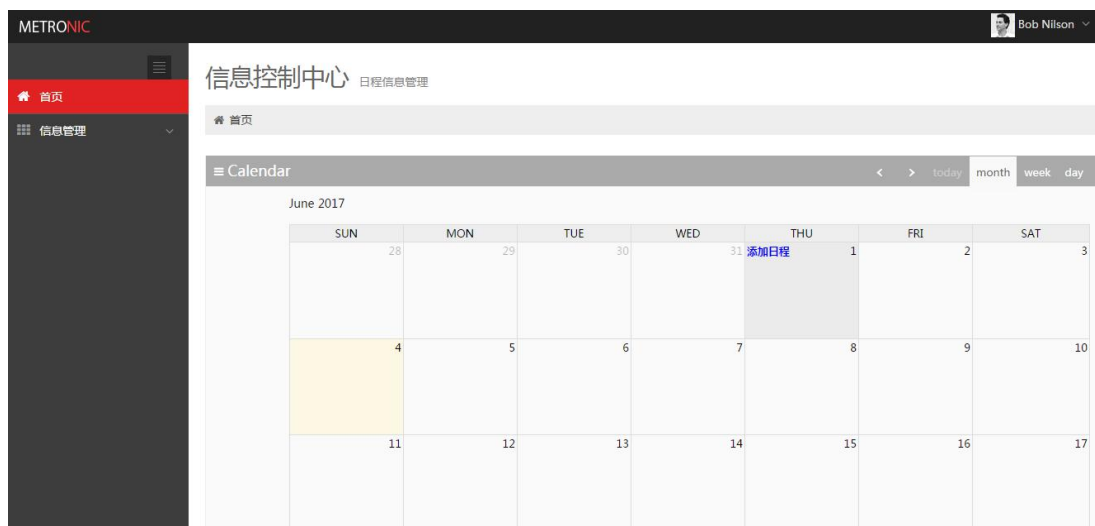


图 4.4 “默认首页”窗口

日程安排信息的添加过程实现逻辑：点击了添加日程安排的链接之后，会弹出添加日程安排的窗口，用户根据页面中的显示窗口，填入相关的日程信息，日程内容是比较简短的信息，所以字数需要控制在 200 个字数以内，需要用户添加的信息就是日程内容和日程类型，日程的发生日期会根据添加的日期自动生成，不需要用户自己去判断添加。

新增日程记录调用的是 `programmeManager` 中的 `insert` 方法，传入的参数是一个新创建的 `programmeDO` 对象，插入新的数据时，需要对数据的完整性做校验，所以添加日程过程中，前端需要用 `JavaScript` 初步做校验，等数据传送到服务端的时候，再次做校验，才能避免错误。

修改功能的实现与添加相类似，不过也有一些区别，修改的过程中，首先要获取需要修改的对象的编号，要在数据库中查找此对象，如果没有查找到此对象，那么本次的操作是非法的；如果存在此对象，那么接着根据新传入的数据，重新给需要修改的对象赋值，并且将数据传入到数据库中，完成数据更新。

4.2.3 展示当前用户所有的日程安排

在首页面的导航菜单中选择“我的日程”选项，后台会执行查找操作，对应的是 `programmeController` 中的 `queryProgramme()` 方法，获取到当前用户所有的日程安排，如图4.4所示。本页面显示所有的用户日程安排，其主要目的就是为了让用户更加集中查看日程，也方便用户自由更改日程信息。当然用户可以在首页直接看到已经添加的日程信息。

用户查看所有的日程信息，可以第一时间查看接下来的日程安排，对于生活工作安排有很好的帮助。

展示当前用户所有日程安排页面图如下图4.5所示

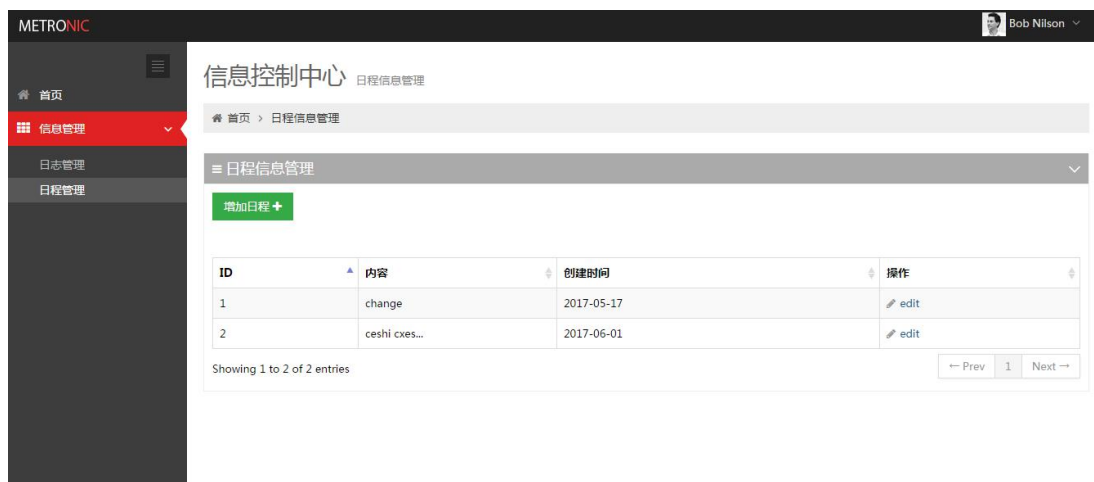


图 4.5 “我的日程”操作窗口

通过图片展示可以看到在操作栏有“修改”“删除”操作，用户在查看日程的时候，可以直接对日程进行操作。点击“修改”，前台获取到当前日程信息的编号，将需要操作的日程信息获取到，可以展示原内容并且作以修改，更新操作就如前面所提到的更新操作。如果是“删除”某条日程信息，仅仅获取日程信息的编号即可，将编号传入后端，执行 `programmeManager.deleteByPrimaryKey()` 方法，根据日程的编号直接删除本条记录。而且本系统采取的删除方法是逻辑删除。

而且系统会根据不同的日程类型进行日程筛选，可以根据类型不同来查看

日程安排。

4.2.4 日志管理界面设计

在系统首页的左侧导航菜单中，有日志管理导航栏，点击之后主界面会跳转到日志管理的页面，如下图 4.6 所示。日志列表展示页面，主要是表格形式，每一列分别是标题、时间和操作，这种日志的形式和发帖很相似，但是区别不同的是，发帖能够和他人共享，而本系统的日志是共享的，也不会有评论等。因为日志的作用和目的是在于能够让用户记录一天的工作，总结和积累，这样可以在之后的工作安排中更加合理和高效。

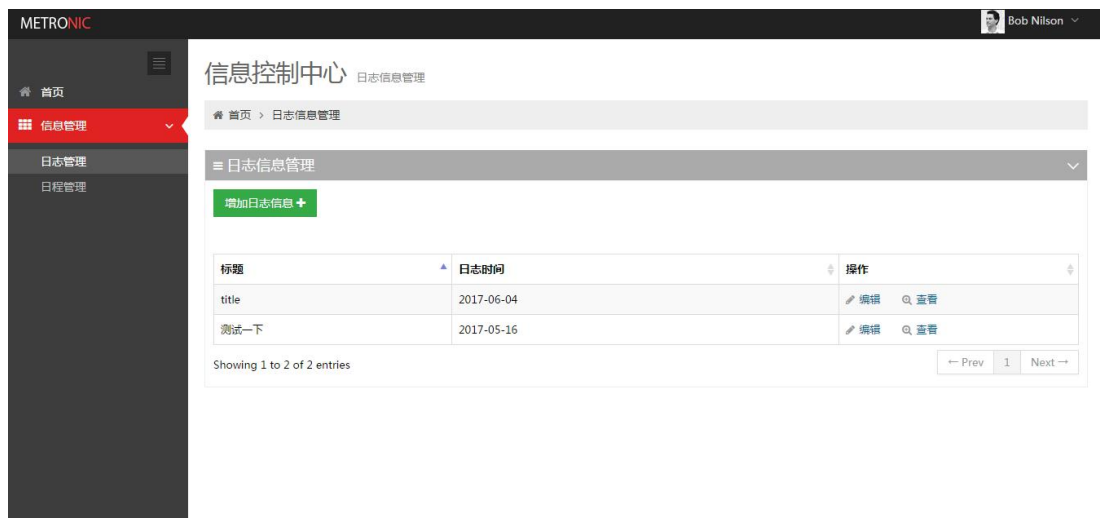


图 4.6 “日志管理”操作窗口

在添加日志操作的时候实现过程是传入用户信息，因为在日志表中，有用户编号的字段和将用户，日志关联，在 `addDailyRecord()` 方法中必须的参数是 `userId`、`title`、`content`（默认不允许添加内容为空的日志记录）、还有 `remarks` 备注信息。

首先检测传入的参数不能为空，使用 `StringUtils` 的 `isNotBlank()` 方法检测不为空，然后将对应的属性参数赋到新的对象中，最终利用 `dailyRecordManager.insertSelective()` 方法将新数据插入到数据库中；。

更新数据操作时仍需要判断 `userId` 的值是否为空，否则不能被修改，而且修改时也要注意日志和用户之间的匹配关系，不能被不互相匹配的日志信息被修改，如果所有的数据都是能够正常操作的，那么最终就会更新数据，将原始数据替换成我们想要修改的样子。

删除操作同修改操作都需进行用户匹配关系的判断，获取选中信息的 `id` 之后，执行 `delete` 操作即可完成记录的删除。

4.2.5 用户信息管理界面设计

此界面是管理员身份的用户登录进来才会看到的页面，普通用户是无法看到展示的信息，如果普通用户忘记了自己的密码，管理员可以通过在此页面进行操作，将密码进行修改。



ID	用户名	用户昵称	用户职位	操作
1	森森	Karry	学生	修改密码
2	林林	Duke	学生	修改密码
3	伊泽瑞尔	EZ	设计师	修改密码
4	内瑟斯	抛头	自由职业者	修改密码
5	内瑟斯	抛头	自由职业者	修改密码
6	艾希	黑冰	自由职业者	修改密码

图 4.7 “用户管理”操作窗口

进入本页面需对登陆用户进行身份权限判断如果 `user.getUserType != 1`，则会提示当前用户没有管理权限。因为本系统不会有比较分明的权限规范，所以管理员的操作只需要管理一下用户的个人信息就可以了。

4.2.6 提醒客户界面设计

进入默认主页面中，在上方的右侧地方有一个“提醒用户”的图标，设计为日程在提前半小时会有日程安排提醒，如果有提醒，这个图标会显示有消息提醒，如下图 4.8 所示。如果用户登录后在半小时之内会有日程安排，系统会给用户提醒，有任务需要等待执行。



图 4.8 “提醒客户”展示页面

4.2.7 查询管理界面设计

查询管理可以从时间和关键字进行查询，输入日程发生的时间即可看到设定时间段内的所有日程信息记录，此外，用户可以在输入框中输入一些关键字，根据这些关键字在数据库中匹配相应的日程信息。

“搜索查询”页面图如下图 4.9 所示

图 4.9 “日程信息查询”操作窗口

该模块的实现使用的是数据库模糊查询，首先客户端表单提交关键词，然后服务器端获取用户表单提交的关键词，由服务器端获取到关键字之后，拼接到 sql 语句当中去，执行该语句，访问数据库，查询含有该关键字的日程信息。最后将查询到的日程信息返回给客户端界面显示。系统中，会调用 programmeQuery 的 createCriteria() 方法，添加筛选条件 andContentLike()，匹配数据。实质上是利用 MySQL where 子句的基础上结合 like 运算符实现。

搜索匹配字段的指定内容，LIKE 通常与通配符 % 一起使用，% 表示通配 param 框中输入的内容。而不加通配符 % 的 LIKE 语法，表示精确匹配，其实际效果等同于 = 等于运算符，本模块使用的是通配符%。相同的，在日志信息管理页面也能够有相似的操作。利用 like 模糊查询，详细代码如下：

```
programmeQuery.createCriteria().andProgramUserIdEqualTo(Long.valueOf(userId));
```

```
//如果传入的参数存在日程类型，加入筛选条件
if(StringUtils.isNotBlank(programmeType)){
```

```
programmeQuery.createCriteria().andProgramTypeIdEqualTo(Integer.valueOf(programmeType));
```

```
}
//如果传入的参数存在关键字，加入筛选条件
if(StringUtils.isNotBlank(key)){
    programmeQuery.createCriteria().andContentLike(key);
}
```

上面的是在对 query 拼装的过程，这里是对传入参数作判断。这种做法能够减少代码量，避免代码冗余。

```
List<ProgrammeDO> programmeList =
    programmeManager.selectByQuery(programmeQuery);
```

本系统中使用的是 mybatis 作为数据库连接的工具，简单的来说实现步骤是：经过 controller 再经过 service 然后执行 service 中的相关方法并关联到 mapper 再执行 mapper.xml 中的 sql 语句。

5 系统运行测试

通过前期的系统功能需求分析，完成了数据库的设计和数据库表的创建，同时针对不同模块完善了模块功能，基本完成了计划停电管理系统的软件开发，在产品正式投入使用之前，还需进行软件运行测试，应尽可能多的找到系统中存在的错误和缺陷，为最终用户提供一个高质量，稳定性好的产品。在实际操作过程中，让其在不同硬件的客户端及网络中进行测试，结合续需求分析检验系统的功能是否正常，在测试中不断完善，提高系统可靠性。

5.1 用户管理模块系统测试

5.1.1 登录界面运行测试

登录测试是各类用户登录到本系统，软件提供了登录和注册、修改登录密码功能。用户登陆测试的操作过程如表 5.1 所示。

表 5.1 登录测试用例表

实例名称	用户登录
描述	通过登录页面，系统中不同用户的管理权限设置不同，进入界面不同。在登录前可以注册和修改密码。
测试方法	查看系统登录是否有错、以及出错后是否显示相关提示，使用不同的登录方法进行登录，从而进行验证测试。
	用正确用户名密码登录；用错误用户名登录；用错误密码登录；修改用户名登录；注册用户登录。
输入	使用用户名Karry 密码 111111 登录； 使用用户名karry 密码 111111 登录； 使用用户名Karry 密码 11s1 登录； 更改用户Karry 的密码为 111 进行登录； 注册用户Candy 密码为 111111 进行登录；
结果预想	测试身份验证成功，进入系统界面； 测试显示“用户不存在”，确认后返回登录界面； 测试显示“用户密码错误”，确认后返回登录界面； 测试显示密码修改成功； 显示注册用户成功，并跳转至登录界面；
结论	测试结果符合预期内容，测试成功。

(1) 当用户名输入错误时，无法登录系统，显示“用户不存在”。如图 5.1 所示：



图 5.1 “用户不存在” 登陆界面

(2) 当密码输入错误时，无法登录系统，显示 “用户密码错误”。如图 5.2 所示：



图 5.2 “用户密码错误” 登陆界面

5.1.2 用户信息管理界面运行测试

信息管理主要针对管理员用户，是管理员维护系统安全和更改用户权限最直接有效的途径，因为用户信息管理涉及到整个系统中的用户能否正确运行本系统，所以在对管理员登陆时需要严格判断。

信息管理测试的操作过程如表 5.2 所示

表 5.2 信息管理测试用例表

实例名称	管理员对用户信息进行管理
描述	管理员能成功更改用户信息，非管理员用户不能显示用户管理页面。
测试方法	分别使用不同权限用户登录，观察能否查看到所有用户信息。
输入	使用用户名admin 密码admin登录； 使用用户名Karry 密码 1111111 登录；
结果预想	进入系统选择用户管理模块，可对选择用户信息进行更改； 当非管理员用户无法进入用户管理界面，更不可操作；
结论	测试结果符合预期内容，测试成功。

获取管理员用户权限后，即可更改用户信息如下图 5.3 所示：

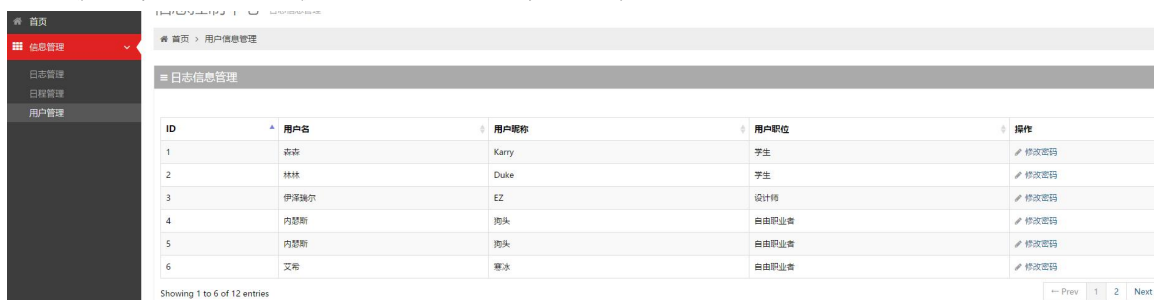


图 5.3 “用户信息管理”界面

当非管理员用户进入首页，导航菜单界面图 5.4 所示提示：



图 5.4 “管理员用户登录查看导航”界面

5.2 日程管理模块系统测试

系统测试之前，主要是在软件的集成中，对程序内部结构中着重在于结构中各模块之间的连接是否符合要求来进行测试。测试时注意程序各模块单元、在模块的

接口连接中能否协调配合工作。测试系统主要的一些业务功能和模块间的连接，从测试结果看系统各接口连接正常，模块间能进行工作，测试成功。

5.2.1 用户日程操作界面运行测试

本功能是系统中核心的功能点，其中的操作包括：用户如果有日程安排，需要根据需求添加日程，如果日程有改动，用户可以对已经添加的日程进行修改。已经添加的日程如果取消可以进行主动删除。

日程管理操作测试的操作过程如表 5.3 所示。

表 5.3 日程信息管理测试用例表

实例名称	日程操作功能
描述	当前用户根据需求对日程进行操作。
测试目的	添加日程，对日程信息管理。
输入	根据前端页面现实的信息，用户按照要求添加日程信息。
结果预想	确认添加保存之后，信息成功录入。
其他操作	删除或者修改，点击操作按钮，系统做出响应。
结论	测试结果符合预期内容，测试成功。

测试步骤：

- (1) 新添加日程，对已添加的日程金心信息管理；
- (2) 查看所有日程信息；
- (3) 对日程信息管理。

5.2.2 用户日志管理界面运行测试

为了帮助用户更好的提升工作效率和合理安排时间，系统扩展了日志的功能，日志管理的功能是为了记录用户的工作，类似于工作总结，或者是用户其他的活动之后的内容记录，和日程的信息管理相辅相成。

日志管理操作测试的操作过程如表 5.4 所示

表 5.4 日志信息管理测试用例表

实例名称	日志管理操作
描述	用户在每一天完成工作或者某项活动之后，可以做一个工作总结、活动感受等等
测试目的	日志信息管理
输入	添加的时候需要添加日志的信息；其他功能不需要输入信息
结果预想	点击操作按钮，系统做出响应。
结论	测试结果符合预期内容，测试成功。

测试步骤：

- (1) 添加日志信息、或者是修改日志信息，增加信息。

- (2) 进入管理界面。
- (3) 日志信息新增或者是被修改、删除。

日志信息管理界面如图 5.5 所示：

≡ 日志信息管理

添加日志 +

标题：内容：

创建时间：

起止时间：

搜索 Q

标题	日志时间	操作
测试01	2017-06-09	<div>编辑</div> <div>查看</div> <div>删除</div>
测试02	2017-06-09	<div>编辑</div> <div>查看</div> <div>删除</div>
测试一下	2017-05-16	<div>编辑</div> <div>查看</div> <div>删除</div>

图 5.5 “日志信息” 管理界面

5.2.3 提醒功能界面测试

在用户成功登录本系统之后，系统监测当前用户的所有日程，根据已经添加日程的时间，对日程监控。设置计时器，当日程在发生时间的前 30 分钟会有一个消息提醒，提醒用户将要准备的的事件。

日程提醒测试的操作过程如表 5.5 所示：

表 5.5 日程提醒测试用例表

实例名称	日程提醒功能
描述	监控当前用户的所有日程信息，在日程将要发生的前 30 分钟会有消息提醒
测试目的	系统对用户的添加日程的提醒
输入	无需输入，用户成功登陆并且添加了一条日程记录
结果预想	成功提醒
结论	测试结果符合预期内容，测试成功

测试步骤：

- (1) 用户成功登录本系统；
- (2) 后台查询到当前用户的所有日程信息；
- (3) 等待日程时间。
- (4) 成功提醒。

5.3 查询模块系统测试

对于本系统，用户可以在日程信息管理界面或者日志信息管理界面看到多项筛选条件。管理页面默认展示给用户的是该用户添加过的所有日程信息，如果信息众多，想要翻页浏览查询就很痛苦，所以查询模块可以让用户添加一些筛选条件进行

筛选：例如在搜索框中输入关键字，查询就会在内容中匹配；或者选择日程类型，筛选出所有该类型的日程；也可以根据时间搜索查询。给用户提供更加强的信息搜索体验。

查询管理模块测试操作过程如表 5.6 所示：

表 5.6 查询管理模块测试用例表

实例名称	信息查询管理
描述	输入关键字、匹配类型、根据时间筛选
测试目的	多项筛选条件拼接，更精确地搜索出目的信息
输入	内容的关键字（日程信息）； 标题的关键字（日志信息）； 选择类型； 选择时间； 不同搜索条件之间的搭配组合搜索；
结果预想	反馈符合要求的相关记录。
结论	测试结果符合预期内容，测试成功

测试步骤如下：

- （1）检查是否选择正确的查询方式。
- （2）若查询方式正确，在检索框输入需要查询的关键字。
- （3）显示查询记录
- （4）没有对应的记录返回为空。

日程、日志信息记录查询成功界面，如下图 5.7 所示：

日志信息管理

增加日志 +

标题： 1 内容：

创建时间：

起止时间： - 搜索 Q

标题	日志时间	操作
11	2017-06-04	编辑 查看 删除

Showing 1 to 1 of 1 entries

← Prev 1 Next →

图 5.7 “日程、日志信息查询成功”界面

信息记录查询数据为空时界面，如下图 5.8 所示

日程信息管理

增加日程 +

关键词：

测试

发生时间：

日程类型：

会议

起止时间： -

搜索 Q

ID	内容	创建时间	操作
No data available in table			

图 5.8 “信息记录数据为空”界面

6 结论

6.1 总结

基于 Java 的日程管理系统将我大学前三年的所学到的软件开发知识独立运用到了实际的开发中去，对于理论知识有了更深层次的理解，也锻炼了自己的动手能力。为了做好本次的毕业设计，动手之前，查阅了很多关于日程管理的相关流程和思想，确定了开发语言之后，将之前看过的相关 Java 书籍知识点简单回顾，并且新翻阅了一些关于 Java Web 开发的书籍。结合自己之前的开发经验，设计开发出这套日程管理系统。在此之前的几个学期，期末也有相关的软件开发任务，但是由于自己不够认真的原因，并没有积累很多经验，在本次的开过程中，全程都是需要自己动手开发，不会的问题需要查阅相关资料，说明平时的基本功很重要，往后的工作中一定吸取教训，认真对待每一个任务，只有好好积累经验才能更好完成工作。

本次的开发过程中，使用的是 spring-boot，相比较其他的框架来说，这是属于比较新的后端框架，相当于一些库的合集。过去使用的 spring+springMVC+Mybatis 框架，也就是常说的 SSM。Spring-boot 直接整合了这些常用的框架，需要使用的东西，直接在配置文件中添加配置就可以，而 spring-boot 开发团队的初衷就是避免繁杂的配置项，让开发更加简单。这也是我选择该框架的目的。

对于本次的开发，收获还是蛮大的。从接到毕设任务之后开始，就对题目开始查询资料，对题目进行分析，需求分析，功能设计。在进行数据库设计，经常用的 Mysql 数据库，简单免费并且功能强大。再到编码阶段，不断完善。个人来说，锻炼了自己的动手能力，锻炼了独立思考和解决问题的能力，对之前所学到的理论知识在实践中得到了巩固。

日程管理对于个人的日程管理效率提升上有很大的帮助。由于本次的开发时间和本人的技术限制，还是存在很多的不足之处，很多的功能还是需要改进的，需要做更进一步的需求分析和调查，更加深入的完善。目前日程管理系统是一个很常见的软件，大致公司的管理系统，小致我们的智能手机，都有这些功能。总之，好的软件总是需要慢慢迭代，不断进步才能够满足用户的需求。

参考文献

- [1] Cay S.Horstmann 著 . JAVA 核心技术卷 I : 基础知识 . 周立新 陈波译 . 第 9 版 . 北京 : 机械工业出版社 , 第 1 版 (2014 年 1 月 1 日)
- [2] 雷富强 , 王志勇著 . Effective Java Programming Language Guide . 北京 : 机械工业出版社,2013
- [3] (美)Joshua Bloch 著 . Java SE 6.0 编程指南 . 第一版 . 北京 : 人民邮电出版社 , 2010
- [4] Bruce Eckel 著 . Java 编程思想 . 陈昊鹏译 . 第四版 . 北京 : 机械工业出版社 , 2010
- [5] 郭克华主编 . JavaEE 程序设计与应用开发 . 北京 : 清华大学出版社 , 2011
- [6] 王晓华 , 于瑞云 , 蔡国民编著 . JavaEE 主流开源框架 . 北京 : 电子工业出版社 , 2011
- [7] (美)David Harms 著 黄习福等译 . JSP、Servlets 与 MySQL 开发指南 . 北京 : 电子工业出版社 , 2011
- [8] MySQL AB . MySQL . Indianapolis, Ind. :MySQL Press , 2010
- [9] Horstmann.C.S and Cornell.G . Core Java 2,Volume I-Fundamentals . Seventh Edition,2015
- [10] Atkinson C., Bostan P.,Butter T. Context-Sensitive Service Discovery for Mobile Commerce Applications[C]. Wireless and Mobile Communications, 2013. ICWMC ' 08.The Fourth International Conference on, 2008: 352-357.
- [11] 田珂, 谢世波, 方马. J2EE 数据持久层的解决方案[J]. 计算机工程, 2010
- [12] 冯燕奎 . JSP 实用案例教程 . 北京 : 清华大学出版社 , 2011
- [13] 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. 2014
- [14] 王福强著 . Spring 揭秘 . 北京 : 人民邮电出版社 , 2013
- [15] 孙卫琴, 李洪成. Tomcat 与 Java Web 开发技术详解[M]. 电子工业出版社 , 2011
- [16] 林信良. Spring2.0 技术手册[M]. 北京 : 电子工业出版社 , 2012
- [17] Jason Brittain, Ian F.Darwin 著 吴豪等译 . Tomcat 权威指南 . 北京 : 中国电力出版社 , 2010
- [18] (美)Thomas A. Powell 著 王德才, 傅吉尧, 胡强译 . JavaScript 完全参考手册 . 北京 : 清华大学出版社 , 2014
- [19] Michael V. Mannino . 数据库设计、应用开发与管理[M] . 第二版 . 北京 : 电子工业出版社 , 2010
- [20] 鲁强, 屈红刚等编著 . 程序设计 . 北京 : 清华大学出版社 , 2009
- [21] 王意洁著 . 面向对象数据库的并行查询处理与事务管理 . 长沙 : 国防科技大学出版社 , 2011

致 谢

感谢导师王辉老师的关心、指导和教诲。王辉老师追求真理、献身科学、严以律己、宽以待人的崇高品质对学生将是永远的鞭策。

作者在攻读学士学位期间的工作自始至终都是在王辉老师全面、具体的指导下进行的。王老师渊博的学识、敏锐的思维、民主而严谨的作风，使学生受益匪浅，终生难忘。

衷心感谢在此次项目开发过程中，精心指导过我的各位老师。老师们对学术的认真使我们备受鼓舞，从而促使我们全身心的投入到学习和工作当中去。同时还要感谢所有学院的老师在学习和工作各方面给过我指导，给我提供了许多宝贵建议的各位同学们。

感谢同学们，是他们在工作和学习的过程中不断的帮助我，为我解决了各种困难，使我至始至终都能有一个快乐的心情。

毕业设计（论文）知识产权声明

本人完全了解西安工业大学有关保护知识产权的规定，即：本科学生在校攻读学士学位期间毕业设计（论文）工作的知识产权属于西安工业大学。本人保证毕业离校后，使用毕业设计（论文）工作成果或用毕业设计（论文）工作成果发表论文时署名单位仍然为西安工业大学。学校有权保留送交的毕业设计（论文）的原文或复印件，允许毕业设计（论文）被查阅和借阅；学校可以公布毕业设计（论文）的全部或部分内容，可以采用影印、缩印或其他复制手段保存毕业设计（论文）。

（保密的毕业设计（论文）在解密后应遵守此规定）

毕业设计（论文）作者签名：

指导教师签名：

日期：

毕业设计（论文）独创性声明

秉承学校严谨的学风与优良的科学道德，本人声明所呈交的毕业设计（论文）是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，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发表或撰写过的成果，不包含他人已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了致谢。

毕业设计（论文）与资料若有不实之处，本人承担一切相关责任。

毕业设计（论文）作者签名：

指导教师签名：

日期：