**软件工程综合作业**

**学院（系）： 计算机科学与工程学院**

**专 业： 软件工程**

**学 生： 滕依筱 马运国**

**学 号：918106840206 918106840229**

**完成日期： 2021年 1月**

**题目：南理工疫情信息系管理系统分析与设计**

**成绩：**

**总计 大作业 54页**

**表格 22表**

**插图 56**

**目 录**

[第一章 南理工疫情信息管理系统需求分析 1](#_Toc265136600)

[1.导言 1](#_Toc265136601)

[2.系统定义 2](#_Toc265136602)

[3.应用环境 3](#_Toc265136603)

[4.功能规格 5](#_Toc265136604)

[5.性能需求 17](#_Toc265136612)

[6.产品提交 18](#_Toc265136613)

[7.实现约束 19](#_Toc265136614)

[第二章 南理工疫情信息管理系统概要设计 20](#_Toc265136615)

[1.系统分析 20](#_Toc265136616)

[2. 界面设计 21](#_Toc265136617)

[3. 体系结构 23](#_Toc265136618)

[4. 数据模型 28](#_Toc265136625)

[第三章南理工疫情信息管理系统详细设计 33](#_Toc265136627)

[1.系统设计概述 33](#_Toc265136628)

[2. 详细设计概述 33](#_Toc265136629)

[3. 登录模块的详细设计（该模块可复用） 33](#_Toc265136630)

[4. 配置文件 44](#_Toc265136635)

[第四章 南理工疫情信息管理系统编码实现 49](#_Toc265136636)

[1.编码格式规范 49](#_Toc265136637)

[2.命名规范 49](#_Toc265136638)

[3.声明规范 52](#_Toc265136639)

[4.目录规范 52](#_Toc265136640)

[第五章 南理工疫情信息管理系统测试计划 53](#_Toc265136642)

[1. 测试项目 53](#_Toc265136643)

[2. 测试方法 54](#_Toc265136644)

[第六章 总结与展望 56](#_Toc265136645)

[参考文献 57](#_Toc265136649)

# 第一章 南理工疫情信息管理系统需求分析

## 1.导言

### 1.1 目的

该文档是关于用户对于南理工疫情信息管理系统的功能和性能的要求，重点描述了南理工疫情信息管理系统的功能需求，是概要设计阶段的重要输入。

本文档的预期读者是：

· 设计人员；

· 开发人员；

· 项目管理人员；

· 测试人员；

· 用户。

### 1.2 范围

该文档是借助于当前系统的逻辑模型导出目标系统的逻辑模型的，解决整个项目系统的“做什么”的问题。在这里，没有涉及开发技术，而主要是通过建立模型的方式来描述用户的需求，为客户、用户、开发方等不同参与方提供一个交流的平台。

### 1.3 编写说明

JSP,Java Server Page（Java服务器页面）的缩写，一个脚本化的语言。

UML,Unified Modeling Language（统一建模语言）的缩写，是一个标准的建模语言。

### 2.系统定义

我们分别阐述一下项目的来源、背景，项目的用户特点和项目的目标。

### 2.1 项目来源及背景

本项目是为“南京理工大学”的“管理员（辅导员及校领导）”开发的一个南理工疫情信息管理系统。疫情对校园管理提出了新的要求，准确记录每位学生和教职工的信息有利于及时发现和处理异常情况，南理工疫情信息管理平台就为了便捷疫情常态化下统计学校学生、教职工等的基础信息而建立的。校园的每一位人员都能通过该平台进行每日的信息申报、出校申请、返校反馈以及查看当前全国和学校疫情情况，校园的管理者可以通过该平台对每位用户进行信息查看，包括身体健康状况和出校运动轨迹等。

### 2.2 用户的特点

本系统的用户都是网上用户，包括两类，一类“管理员”，包括校内各学院的辅导员及学校领导。“管理员”不但负责管理所有的用户账号，也负责管理用户的健康数据及出校返校数据。

另外一类则是“用户”，包括学校的学生及教职工。他们平时需要登录系统提交自己的每日健康信息、出校申请及返校反馈，并通过系统了解到疫情实时感染数据。

### 2.3 项目目标

本项目设定的目标如下：

· 系统能够提供友好的用户界面，使操作人员的工作量最大限度的减少；

· 系统运行的效率一般，但是运算出结果所需的时间应该是用户在可以接受的范围内 （<=0.1秒）

· 平台的设计具有一定的超前性，灵活性，能够适应学校管理的变化；

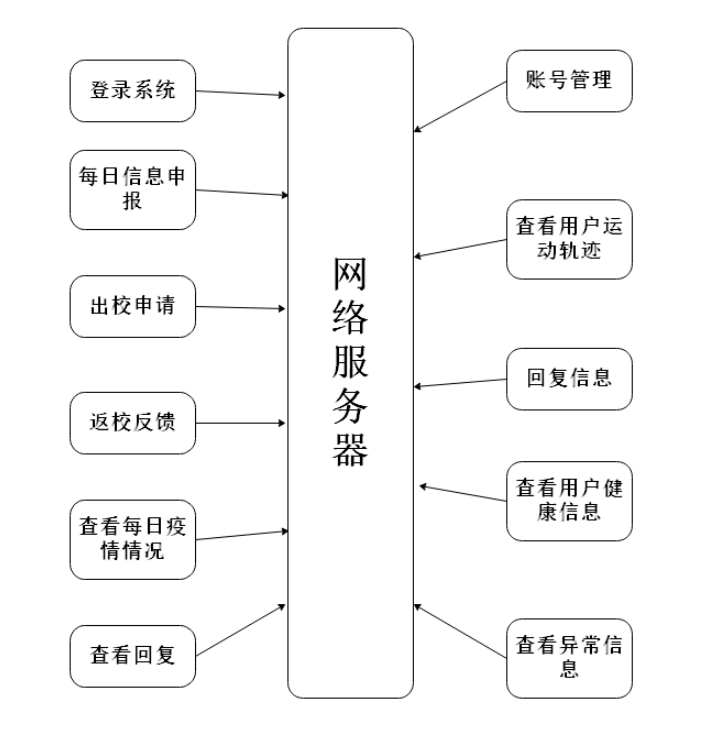
· 通过这个项目可以锻炼自己，提高开发能力和项目管理能力。

## 3.应用环境

根据用户的需求陈述，可以确定本项目分为客户端（学生及教职工）和管理端（辅导员及校领导）。

客户端主要功能是“登录系统”、“每日信息申报”、“出校申请”，“返校反馈”、“查看每日疫情情况”、“查看回复”。

管理端的功能提供“登录系统”、“账号管理”、“查看用户运动轨迹”，“查看用户健康信息”、“回复信息”、“查看异常信息”等六大功能。它们的关系如图1-1所示。



**图1-1 南理工疫情信息管理系统流程图**

### 3.1 系统运行的网络环境

本系统的网络运行图如图1-2所示，无论是客户端的导师还是管理端的管理用户都可以通过网络登录到本系统中。导师通过网络录入成绩等相关信息，教学管理部通过网络添加部门及课程信息。管理端的系统管理员需要设置导师的相关信息。

### 3.2 系统运行的硬件环境

本系统的硬件环境如下：

客户机：普通PC

· CPU：2.6 GHz以上

· 内存：2 GB以上

· 能够运行IE11.0以上版本的机器

· 分辨率：推荐使用1024×768以上像素

Web服务器

· CPU：2.6 GHz以上

· 内存：2 G以上

· 硬盘：10 GB以上

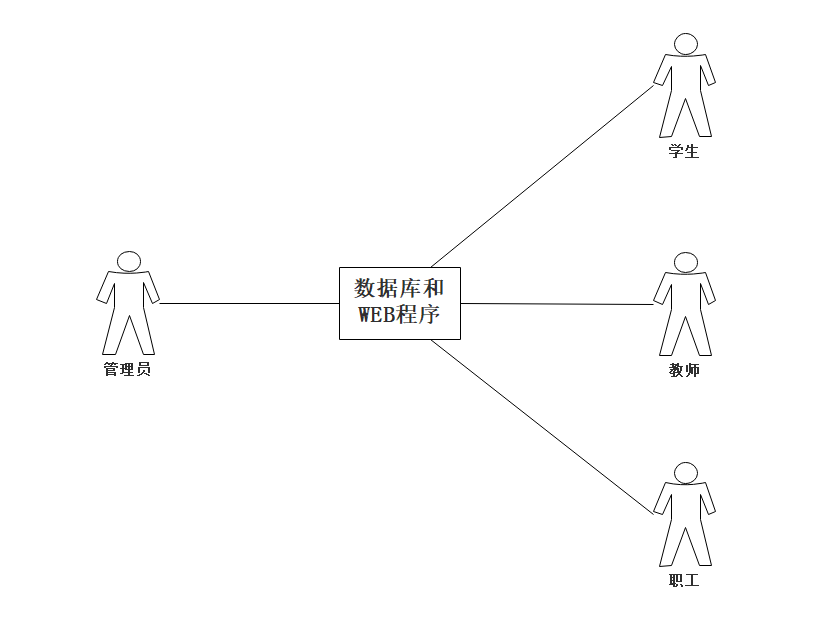
· 网卡：100Mb/s

数据库服务器

· CPU：2.6 GHz

· 内存：4 GB以上

* 硬盘：500 GB以上



**图1-2 网络拓扑结构图**

### 系统运行软件环境

本系统的软件环境如下：

· 操作系统：WinXP或以上版本

· 数据库：MySQL 5.0

· 开发工具包：JDK/IDEA

·Web服务器：Tomcat 10

·浏览器：IE11.0以上（Edge 87.0.664.66 or Chrome 87.0.4280.88）

## 4.功能规格

我们采用面向对象分析作为主要的系统建模方法，使用UML（Unified Modeling Language）作为建模语言。UML为建模活动提供了从不同角度观察和展示系统的各种特征的方法。在UML中，从任何一个角度对系统所作的抽象都可能需要几种模型来描述，而这些来自不同角度的模型图最终组成了系统的映像。

用例描述角色（用户、外部系统以及系统处理）是如何与系统交互来完成工作的。用例模型提供了一个非常重要的方式来界定系统边界以及定义系统功能，同时，该模型将来可以派生出动态对象模型。

设计用例时，我们遵循下列步骤：

1）识别出系统的角色。角色可以是用户、外部系统，甚至是外部处理，通过某种途径与系统交互。重要的是着重从系统外部执行者的角度来描述系统需要提供哪些功能，并指明这些功能的执行者（角色）是谁。尽可能地确保所有角色都被完全识别出来。

2）描述主要的用例。可以采取不断地问自已“这个角色究竟想过系统做什么？”来准确地描述用例。

3）重新审视每个用例，为它们下个详尽的定义。

### 4.1 角色定义

角色或者执行者指与系统产生交互的外部用户或者外部系统。

**4.1.1 管理员**

管理员是指负责某个年级的辅导员或校领导。他可以使用以下功能：“登录系统”、“账号管理”、“查看用户运动轨迹”，“查看用户健康信息”、“回复信息”、“查看异常信息”。

**4.1.2 用户**

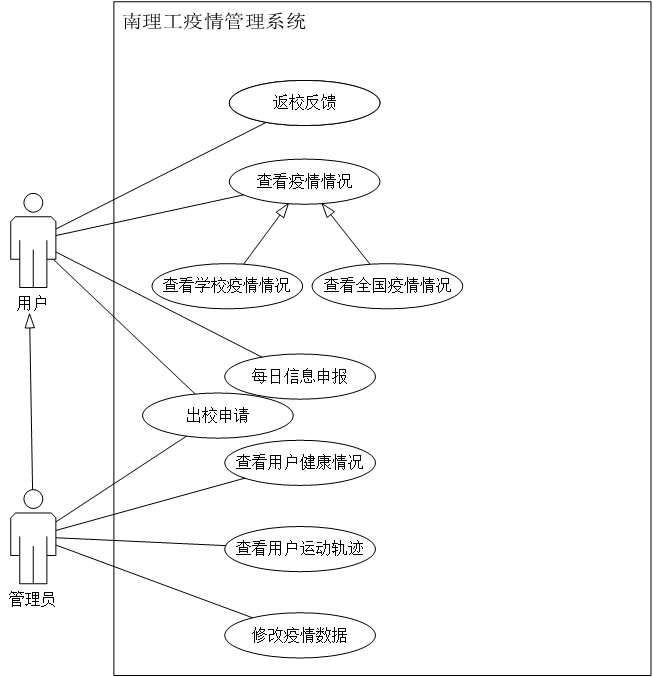
用户是学校内的学生及教职工，主要功能是“登录系统”、“每日信息申报”、“出校申请”，“返校反馈”、“查看每日疫情情况”、“查看回复”。

**4.1.3 数据库**

数据库是一个与系统产生交互的外部系统，这个角色负责系统的数据查询、增加、删除和修改等操作。

### 4.2 系统主用例图

南理工疫情信息管理系统可以分为两个主要的组成部分，一个是客户端子系统，一个是管理端子系统。管理端子系统功能为“登录系统”、“账号管理”、“查看用户运动轨迹”，“查看用户健康信息”、“回复信息”、“查看异常信息”。客户端子系统功能是 “登录系统”、“每日信息申报”、“出校申请”，“返校反馈”、“查看每日疫情情况”、“查看回复”。系统的主用例如图1-4所示。

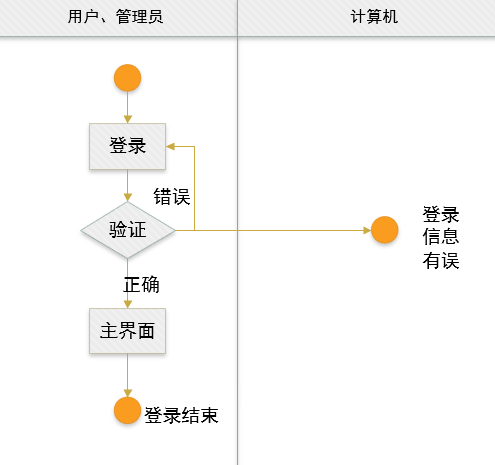


**图1-3 系统的主用例图**

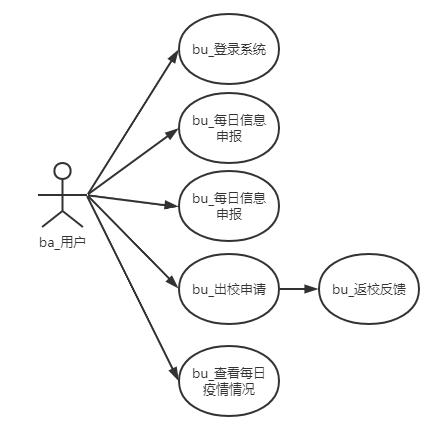
### 4.3 客户端子系统

学生及教职工通过通过系统进行管理。图1-4是它的登录活动图。

客户端的功能主要包括“登录系统”、“每日信息申报”、“出校申请”，“返校反馈”、“查看每日疫情情况”、“查看回复”。图1-5是它的用例图。



**图1-4 客户端登录的活动图**



**图1-5 客户端的功能用例图**

客户端管理的这些用例描述如下：

1.1：登录系统。

1.2：每日信息申报：填写每日健康状况及地址等信息。

1.3：出入校业务：填写出校申请和返校反馈。

1.4：查看每日疫情情况。

**4.3.1登录系统**

|  |  |
| --- | --- |
| 用例名称 | 登录系统 |
| 实现名称 | login.jsp |
| 用例描述 | 用户通过此用例进入系统 |
| 参与者 | 1.用户  2.管理员 |
| 前置条件 | 打开登录界面 |
| 后置条件 | 进入系统平台 |
| 主事件流 | 1.用户填写账号、密码，计算机显示平台界面  2.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a账号、密码错误  1.拒绝登录  2.提示信息错误，用例结束  2.a账号与身份不匹配  1.拒绝登录  2.提示身份不正确，用例结束 |
| 业务规则 | 填写个人信息 |
| 设计的业务实体 | 信息验证 |

**4.3.2每日信息申报**

|  |  |
| --- | --- |
| 用例名称 | 每日信息申报 |
| 实现名称 | everyReport.jsp |
| 用例描述 | 用户通过此用例提交每日信息 |
| 参与者 | 用户 |
| 前置条件 | 1.登录系统 |
| 后置条件 | 1.填写信息表单  2.保存信息  3.提交信息 |
| 主事件流 | 1.计算机根据用户上次保存的信息显示界面  2.用户选择保存信息，计算机更新修改后的表单  3.用户选择填写备注，计算机将其发送到管理员  4.用户选择提交信息，计算机显示提交成功发送界面  5.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a用户未完成所有必填项  1.拒绝保存或提交  2.提示信息不完整，用例结束 |
| 业务规则 | 用户根据实际情况填写信息 |
| 设计的业务实体 | 信息表单 |
| 非功能性需求 | 支持多用户同时提交信息 |

**4.3.3进出校业务**

|  |  |
| --- | --- |
| 用例名称 | 进出校 |
| 实现名称 | askOut.jsp |
| 用例描述 | 用户通过 |
| 参与者 | 用户通过此用例申请并确认出入校 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 1.申请出校  2.确认返校 |
| 主事件流 | 1.用户选择申请出校，计算机显示申请界面  2.用户选择确认返校，计算机显示确认返校信息界面  3.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a用户申请出校信息未完成所有必填项  1.拒绝申请出校  2.提示申请信息不完整，用例结束  2.a用户未成功申请出校，选择核实返校信息  1.拒绝访问  2.提示未申请出校，用例结束  3.a用户返校信息填写不完整  1.拒绝确认返校  2.提示返校信息不完整，用例结束 |
| 业务规则 | 根据实际情况填写出入校信息 |
| 设计的业务实体 | 申请出校表单、确认返校表单 |
| 非功能性需求 | 支持多用户同时填写 |

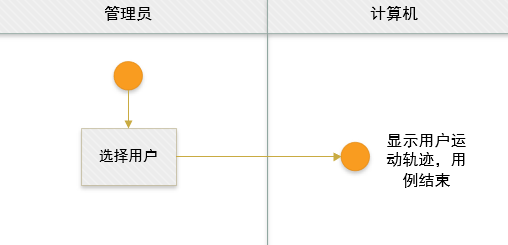
**4.3.4每日感染情况**

|  |  |
| --- | --- |
| 用例名称 | 每日感染情况 |
| 实现名称 | currentSituation.jsp |
| 用例描述 | 用户通过此用例查看全国和学校新冠感染情况  管理员通过此用例修改感染数据 |
| 参与者 | 用户、管理员 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 查看每日感染情况 |
| 主事件流 | 1.用户进入系统，计算机自动显示感染情况  2.管理选择更改数据，计算机显示更改界面  3.管理员确认更改数据，计算机更新数据 |
| 备选事件流 | 1.a抓包数据出现异常  1.拒绝更新  2.提示异常信息，用例结束 |
| 业务规则 | 管理员根据官方发布信息更新数据 |
| 设计的业务实体 | 感染数据 |

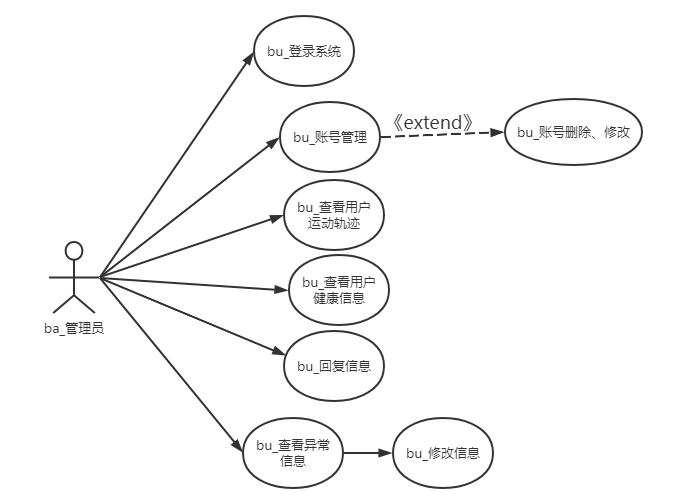
### 4.4 管理端子系统

导师通过通过成绩系统对成绩进行管理。图1-6是它的部分活动图。

管理端的功能提供“登录系统”、“账号管理”、“查看用户运动轨迹”，“查看用户健康信息”、“回复信息”、“查看异常信息”等功能。图1-7是它的用例图。



**图1-6管理员的功能活动图**



**图1-7 管理端用例图**

2.1登录系统

2.2账号管理

2.3查看用户运动轨迹

2.4查看用户健康信息

2.5回复信息

**4.4.1登录系统**

|  |  |
| --- | --- |
| 用例名称 | 登录系统 |
| 实现名称 | login.jsp |
| 用例描述 | 用户通过此用例进入系统 |
| 参与者 | 1.用户  2.管理员 |
| 前置条件 | 打开登录界面 |
| 后置条件 | 进入系统平台 |
| 主事件流 | 1.用户填写账号、密码，计算机显示平台界面  2.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a账号、密码错误  1.拒绝登录  2.提示信息错误，用例结束  2.a账号与身份不匹配  1.拒绝登录  2.提示身份不正确，用例结束 |
| 业务规则 | 填写个人信息 |
| 设计的业务实体 | 信息验证 |

**4.4.2账号管理**

|  |  |
| --- | --- |
| 用例名称 | 修改用户信息 |
| 实现名称 | change.jsp |
| 用例描述 | 管理员用过此用例修改用户信息 |
| 参与者 | 管理员 |
| 前置条件 | 登录界面 |
| 后置条件 | 1.修改用户账号、密码、身份  2.修改用户填写信息 |
| 主事件流 | 1.管理员选择修改用户账号、密码，计算机显示修改界面  2.管理选择修改用户信息，计算机显示修改用户信息界面  3.计算机执行后置条件，用例结束 |
| 业务规则 | 管理员可以根据需求修改用户信息 |
| 设计的业务实体 | 修改指定用户的信息 |

**4.4.3查看用户运动轨迹**

|  |  |
| --- | --- |
| 用例名称 | 查看用户运动轨迹 |
| 实现名称 | track.jsp |
| 用例描述 | 管理员通过此用例查看用户运动轨迹 |
| 参与者 | 管理员 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 查看用户运动轨迹 |
| 主事件流 | 1.管理员选择特定用户，计算机显示该用户的运动轨迹  2.管理员选择轨迹显示范围，计算机显示不同范围的轨迹  3.计算机执行后置条件，用例结束 |
| 业务规则 | 管理员可以根据需求选择特定用户的运动轨迹 |
| 设计的业务实体 | 运动轨迹 |

**4.4.4查看用户健康信息**

|  |  |
| --- | --- |
| 用例名称 | 查看用户健康信息 |
| 实现名称 | health.jsp |
| 用例描述 | 管理员通过此用例查看用户健康信息 |
| 参与者 | 管理员 |
| 前置条件 | 用户登录界面 |
| 后置条件 | 1.查看用户健康信息 |
| 主事件流 | 1.管理员选择查看用户信息，计算机显示该用户信息表  2.计算机执行后置条件，用例结束 |
| 业务规则 | 管理员可以根据需求查看特定用户的健康信息 |
| 设计的业务实体 | 健康信息表 |

**4.4.5回复用户信息**

|  |  |
| --- | --- |
| 用例名称 | 回复用户信息 |
| 实现名称 | adminMessage.jsp |
| 用例描述 | 管理员通过此用例回复用户信息 |
| 参与者 | 管理员 |
| 前置条件 | 登录界面 |
| 后置条件 | 1.回复用户消息  2.发送通知 |
| 主事件流 | 1.用户选择回复用户消息，计算机显示用户消息  2.用户选择发送通知，计算机显示通知发送框 |
| 备选事件流 | 1.a回复消息发送失败  1.拒绝发送  2.提示发送失败，用例结束 |
| 业务规则 | 管理员可以根据需求回复用户消息 |
| 设计的业务实体 | 管理员回复用户信息 |

## 5.性能需求

本系统对于用户的管理权限有很好的限定，用户不能随便操作不属于他的功能。

由于时间限制，本系统并未解决并发性问题。

### 5.1 界面需求

系统的界面要求如下。

1）页面内容：主题突出，站点定义、术语和行文格式统一、规范、明确、栏目、菜单设置和布局合理，传递的信息准确、及时。内容丰富，文字准确，语句通顺，专用术语规范，行文格式统一规范。

2）导航结构：页面具有明确的导航指示，且便于理解，方便用户使用。

3）技术环境：页面大小适当，能用各种常用浏览器以不同分辨率浏览，无错误链接和空链接；采用CSS处理，控制字体大小和版面布局。

4）艺术风格：界面、版面形象清晰悦目、布局合理，字号大小适宜、字体选择合理，前后一致，美观大方，动与静搭配恰当，动静效果好；色彩和谐自然，与主题内容相协调。

### 5.2 响应时间需求

无论是客户端还是管理端，当用户登录，进行任何操作的时候，系统应该及时地进行反应，反应的时间在10秒以内。系统应能监测出各种非正常情况，如与设备的通信中断，无法连接数据库服务器等，以避免出现长时间等待甚至无响应。

### 5.3 可靠性需求

系统应保证7×24小时内不宕机，保证15人可以同时在客户端登录，此时系统能正常运行，正确提示相关内容。

### 5.4 开放性需求

系统应具有较强的灵活性，以适应将来功能扩展的需求。

### 5.5 可扩展性需求

还有很多功能可以添加，如：

1使用简便方法提交信息。

2如果已提交了一次每日信息申报，那么再次填写时可以快速添加信息。

3即使是第一次使用，也有简便方法添加信息。

4统计、查看功能。

5与基本信息系统链接获取学期的学生。

6完善CURD操作。

以上功能均可实现，时间有限，尚未完成。

### 5.6 系统安全性需求

1应该使用过滤器（Filter）或拦截器，对非法进入页面进行拦截

2许多session对象使用后没有释放，很占资源，如果多人使用容易使系统崩溃，真正使用时，必须释放。

## 6.产品提交

提交产品为：

a）应用系统软件包；

b）数据库中有一些测试是的数据

c）系统开发过程文档；

d）系统使用、维护说明文档，提交方式为CD介质。

## 7.实现约束

系统的实现约束如下：

a）操作系统为Windows 10

b）开发平台为：IDEA 2020.2

c）数据库为MySQL 5.0

# 第二章 南理工疫情信息管理系统概要设计

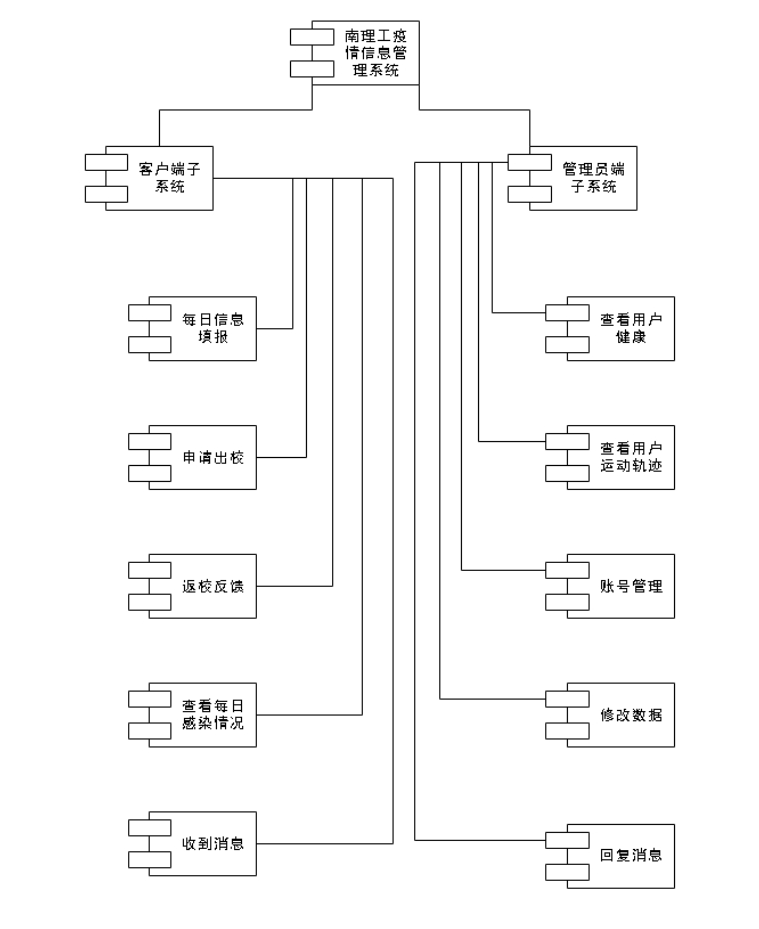
## 1.系统分析

本系统主要可以实现疫情期间在校所有人员的信息统计，管理员可以通过系统查看汇总信息以及后台管理。

客户端主要功能是“每日信息申报”、“申请出校”、“返校反馈”、“每日感染情况”、“收到消息”等五大功能。

管理端的主要功能是“查看用户健康”、“查看用户运动轨迹”、“账号管理”、“修改疫情数据”、“回复消息”等五大功能。

图2-1为客户端和管理端的组成构图。



**图2-1 客户端子系统 管理员端子系统**

## 2. 界面设计

本系统的用户界面按功能分为客户端界面和管理端界面。



**图2-2 登录页面**

### 2.1 管理端界面设计

管理端的主要提供“查看用户健康”、“查看用户运动轨迹”、“账号管理”、“修改疫情数据”、“回复消息”等五大功能。

主要界面设计如下：

·登录界面：通过输入用户名和密码，实现用户登录，后台根据登录用户身份自动跳转到管理员首页；

·管理首页：链接到五大功能；

·查看用户健康情况：包括“查看所有用户健康数据”、“查看特定用户健康数据”和“查看不健康用户数据”等界面。

·查看用户运动轨迹：包括“查看特定用户途径地点及时间”和“可视化显示用户轨迹”等界面。

·账号管理：包括“查询特定用户基本账号信息”、“删除用户”、“修改用户信息”、“修改用户权限”和“批量重置用户密码”等界面。

·修改疫情数据：包括“修改校园相关疫情数据”等界面。

·回复留言：包括“查看用户留言”和“回复用户留言”等界面。



**图2-3 管理员登录后页面**

### 2.2 客户端界面设计

客户端主要提供“每日信息申报”、“申请出校”、“返校反馈”、“每日感染情况”、“收到消息”等五大功能。

主要界面设计如下：

·登录界面：通过输入用户名和密码，后台根据登录用户的身份权限自动跳转到客户端首页；

·客户端首页：链接到五大大功能

·每日信息申报：包括“填写填报信息”、“保存填报信息”、“留言”和“提交填报信息”等界面。

·申请出校：“填写申请出校信息”、“保存申请信息”和“提交出校申请信息”等界面。

·返校反馈：包括“填写实际返校时间”、“填写途径地点和时间信息”和“提交信息”等界面。

·查看每日感染情况：包括“查看实时全国疫情情况”和“查看每日校园疫情情况”等界面。

·收到消息：包括“查看收到回复消息”等界面。



**图2-4 用户登录后页面**

## 3. 体系结构

系统的总体结构设计遵循如下原则。

1）系统应具有良好的适应性：能适应用户对系统的软件环境、管理内容、模式和界面的要求；

2）系统应具有可靠性：采用成熟的技术方法和软件开发平台，以保证系统在以后的实际应用中安全、可靠；

3）系统应具有较好的安全性：应提高安全机制和用户权限限制机制的完善程度，确保数据的受限访问；

4）系统应具有良好的可维护性：系统应易于维护、安装；

5）系统应具有良好的可扩展性：系统应适应未来信息化建设的要求，能方便地进行功能扩展，以建立完善的信息集成管理体系。

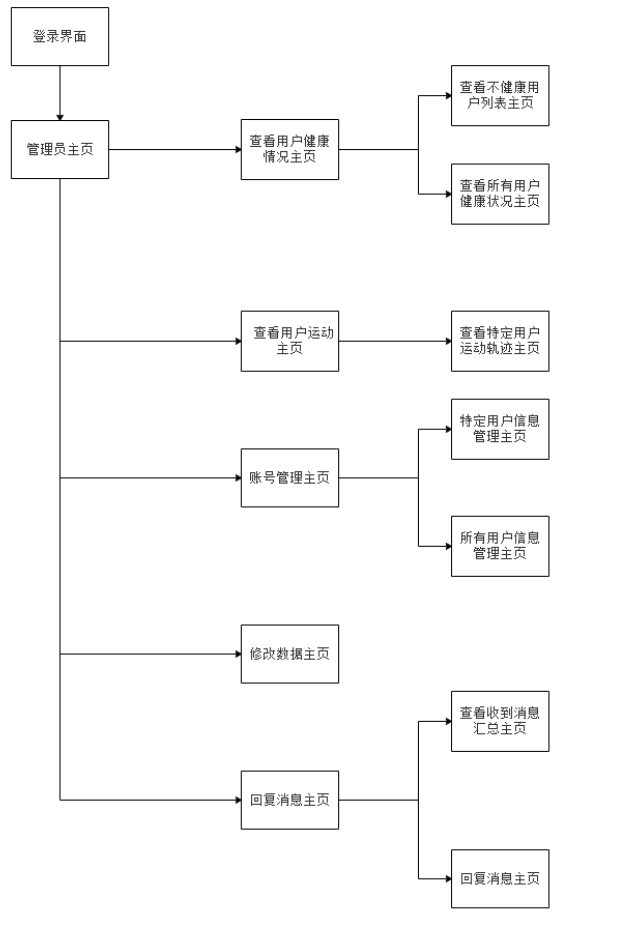
本系统采用体系结构，SSM是一个基于模型（Model）一视图（View）一控制器（Controller），即MVC模式的应用架构的开源框架。

### 3.1 体系结构

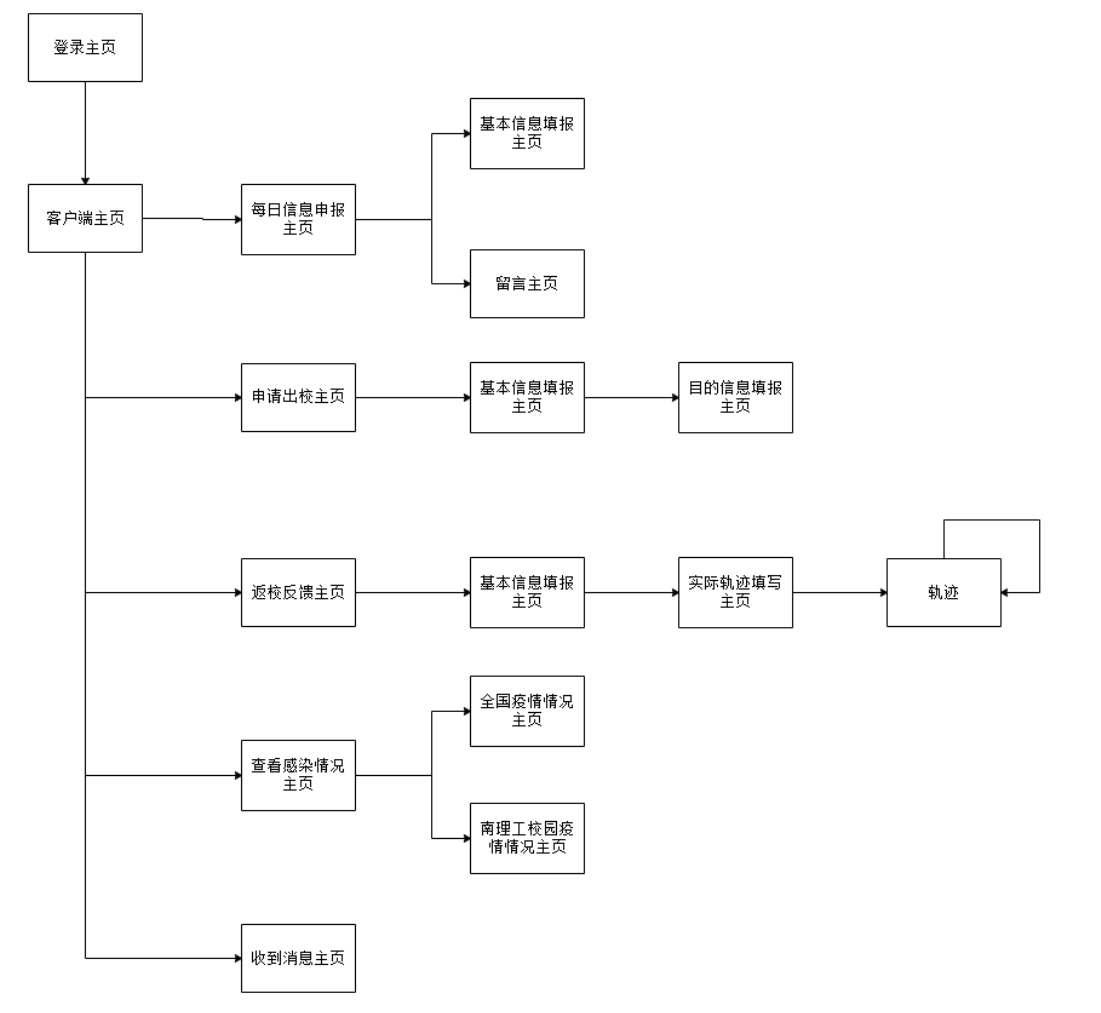
目前软件项目中有很多体系结构，其中SSM是比较流行的一种。

### 3.1.1 SSM体系结构

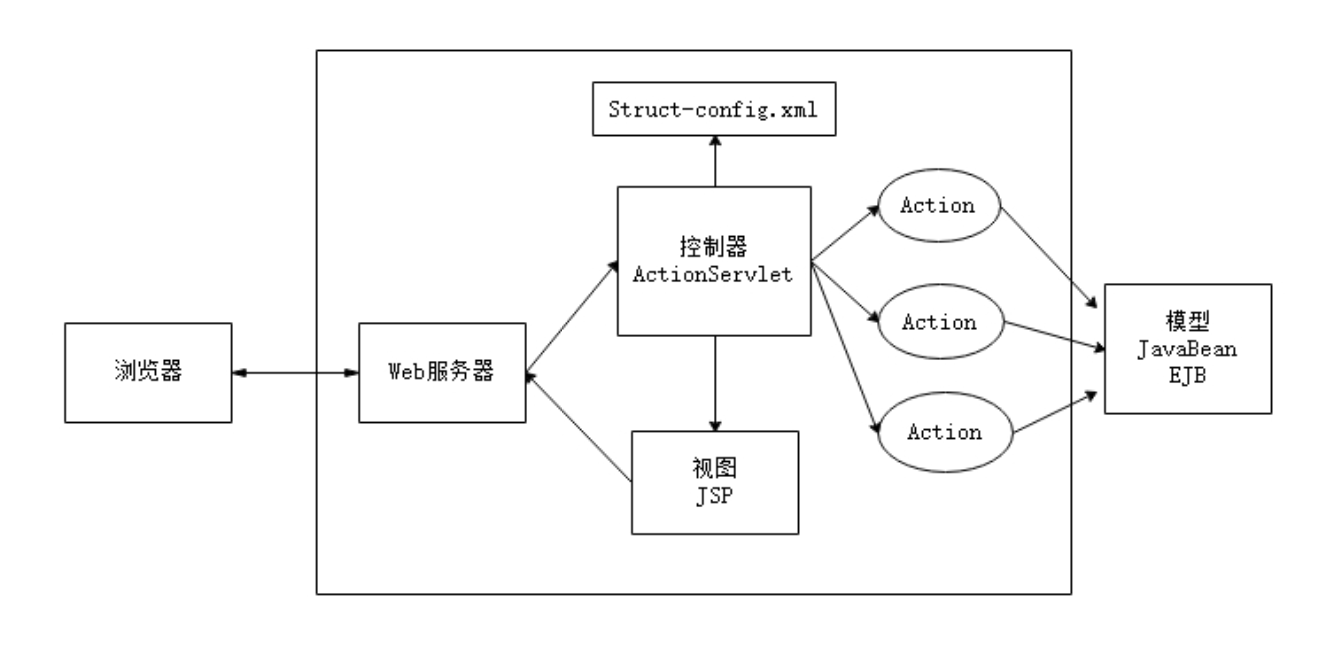
对于开发Web应用，要从头设计并开发出一个可靠、稳定的框架不是一件容易的事情。随着Web开发技术的日趋成熟，在Web开发领域出现了一些现成的优秀的框架、开发者可以直接使用它们，SSM就是一个很好的框架结构，它是在JSP Model2基础上实现的一个MVC框架，在SSM框架在模型由实现业务逻辑的JavaBean或者EJB组件构成，控制器由ActionServlet和Action来实现，视图由一组JSP文件组成，图2-5显示了SSM实现的MVC框架。



**图2-5 管理端的页面流程**



**图2-6 客户端的页面流程**

**·视图，**就是一组JSP文件，这些JSP文件没有业务逻辑，也没有模型信息，只有标签，这些标签可以是标准的JSP标签或者是客户化标签，如SSM标签库的标签。

**图2-7 SSM实现的MVC框架**

**·控制器，**控制器由Servlet类实现， Servlet类是SSM框架中的核心组件，是这个MVC的中央控制器的角色。Servlet主要负责接收HTTP请求的信息，根据配置文件SSM-config.xml的配置信息，将请求转发给适当的Servlet对象，如果该Servlet对象不存在，Servlet会先创建这个Servlet对象.

**·模型，**模型表示应用程序的状态和业务逻辑，业务逻辑常常由JavaBean或者EJB组件实现。

如果在Web应用开发中套用现成的SSM框架，就可以简化每个开发阶段的工作，开发人员可以更加有针对性地分析应用需求，不必重新设计框架，只需在SSM框架的基础上，设计MVC各个模块包含的具体组件，在编码过程中，可以充分利用SSM提供的各种实用类和标签库，简化编码工作。

SSM框架可以方便迅速地将一个复杂的应用划分成模型、视图和控制器组件，简化开发过程。

### 3.1.2 系统体系结构

根据系统分析结果，该系统从结构上应满足：

·基于游览器进行显示以方便用户使用；

·采用MVC的三层体系结构，分化各个功能组件；

·采用JDBC技术与数据库通信以便于数据库的转换；

·采用标签技术完成动态页面的简单逻辑。

本系统的体系结构如图2-6所示，它基本遵循了SSM体系的MVC框架规范。



**图2-8 系统的体系结构**

其中：

**·表示层，**用于与用户进行交互并显示结果。包括所有的JSP，提供用户界面，接受用户输入，用来存放表单数据，并进行表单数据验证；

**·控制层，**包括所有的Servlet类，它完成三项任务，一是进行业务逻辑验证，二是调用模型组件，三是决定将合适的视图组件返回给用户；

**·模型，**包括进行逻辑处理的JavaBean等，数据库采用JDBC技术以提供数据库的可移植性。

**1）客户层：**用于与企业信息系统的用户进行交互以及显示根据特定业务规则进行计算后的结果。本系统将完全采用基于Web的（B/S架构）客户端，即用户可以直接通过浏览器来访问和使用本系统。

**2）中间层：**这相当于三层标准架构中的Web应用服务层，支持诸如响应客户请求以及查询等功能。并且由中间层进行逻辑处理，再将处理的结果反馈给客户或者发送到数据库中。

**3）服务层：**主要是数据库系统，这里的数据库系统主要是关系数据库系统（RDMS）。

### 3.2 系统进行环境

下面讲述系统运行的网络结构，硬件、软件环境。

**3.2.1 网络结构**

主要是希望在局域网内使用。

**3.2.2 硬件环境**

本系统的硬件环境如下。

1）客户机：普通PC

·CPU：2.60 GHz以上

·内存：2 GB以上

·能够运行IE11.0以上

·分辨率：推荐使用1024×768以上像素

2）Web服务器

·CPU：2.60 GHz

·内存：2 GB以上

·硬盘：10 GB以上

·网卡：100 Mb/s速度网卡

3）数据库服务器

·CPU：2.60 GHz

·内存：4 GB以上

·硬盘：500 GB以上

**3.2.3 软件环境**

本系统的软件环境如下：

·操作系统： Windows XP/Windows 7/Windows 10

·数据库：MySQL

·开发工具包：JDK/IDEA

·开发环境：Intellij IDEA

·Web服务器：Tomcat

·浏览器： Google Chrome、IE、Edge、火狐

## 4. 数据模型

本系统的数据模型设计内容主要是进行数据库的设计。

### 4.1 数据库的概念结构模型设计

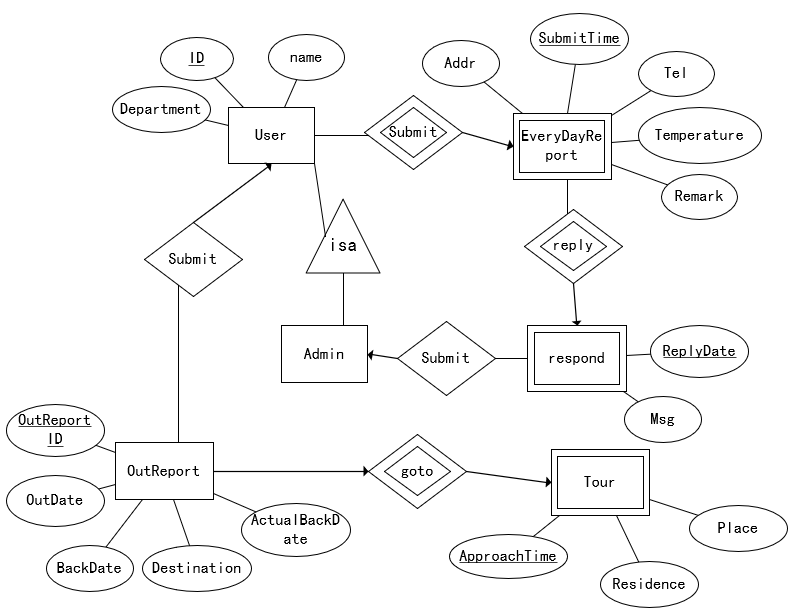
概念设计用来反映现实世界中的实体、属性和它们之间的关系等的原始数据形式，建立数据库的每一幅用户视图。图2-8是系统E-R图。

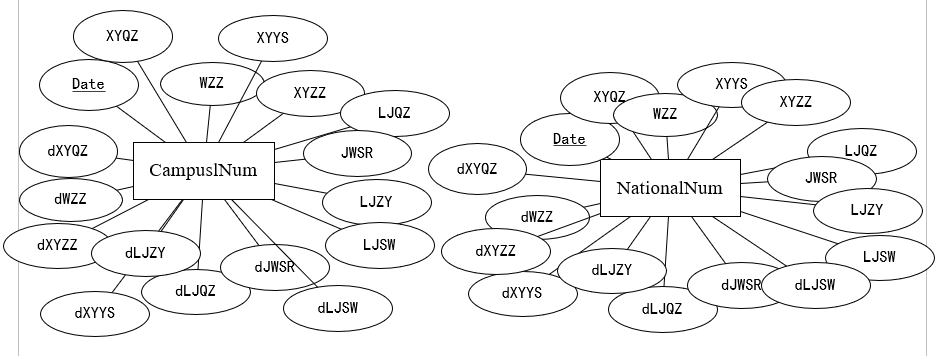
客户端主要功能是“每日信息申报”、“申请出校”、“返校反馈”、“每日感染情况”、“收到消息”等五大功能。

管理端的主要功能是“查看用户健康”、“查看用户运动轨迹”、“账号管理”、“修改疫情数据”、“回复消息”等五大功能。

4.2 数据库的逻辑结构模型设计

数据库的逻辑设计是将各局部的E-R图进行分解、合并后重新组织起来形成数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构、所建立的各个数据之间的相互关系。根据本系统需求分析，数据库有用户、每日信息申报、申请出校、返校反馈、全国疫情、南理工校园疫情。





**图2-9数据库ER图**

\*\*\*系统数据库设计物理模型\*\*\*

\*\*\*\*用户表 (User)----------------1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 |
| 用户ID | ID | char(20) | Y | N |
| 姓名 | Name | char(50) | N | N |
| 密码 | Pswd | char(50) | N | N |
| 学院 | Department | char(50) | N | N |
| 是否管理员 | IsAdmin | Boolean | N | N |

\*\*\*\*每日信息填报表 (EverydayReport)-----------1.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 |
| ID | ID | VARCHAR(20) | Y | N |
| 提交时间 | SubmitTime | DATEYIME | Y | N |
| 电话 | Tel | VARCHAR(20) | N | N |
| 地址 | Addr | VARCHAR(100) | N | N |
| 备注 | Remark | VARCHAR(200) | N | Y |

\*\*\*\*回复消息表(EverydayRespond)--------------------1.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主外键 | 是否为空 |
| ID | ID | VARCHAR(20) | Y | N |
| 发送时间 | SubmitTime | DATETIME | Y | N |
| 回复时间 | ReplyTime | DATETIME | N | N |
| 管理员ID | AdminID | VARCHAR(20) | N | N |
| 回复消息 | Msg | VARCHAR(100) | N | N |

\*\*\*\*申请出校表(OutReport)--------------------1.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注 |
| 申请序号 | OutReportID | Integer | Y | N | 自动增加 |
| ID | ID | VARCHAR(20) | Y | N |  |
| 出校日期 | OutDate | DATE | N | N |  |
| 预计返校时间 | PlannedBackDate | DATE | N | N |  |
| 目的地 | Destination | VARCHAR(100) | N | N |  |
| 实际返校日期 | ActualBackDate | DATE | N | N |  |

\*\*\*\*返校反馈表(Tour)---------------1.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 |
| 出校序号 | OutReportID | Integer | Y | N |
| 到达时间 | ApproachTime | DATETIME | N | N |
| 省 | Province | VARCHAR(50) | N | N |
| 市 | City | VARCHAR(50) | N | N |
| 区/县 | Area | VARCHAR(50) | N | N |
| 详细地址 | Address | VARCHAR(100) | N | N |

\*\*\*\*全国疫情表(NationalNum)---------------------1.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 |
| 日期 | Date | DATE | Y | N |
| 现有确诊 | XYQZ | Integer | N | N |
| 无症状 | WZZ | Integer | N | N |
| 现有疑似 | XYYS | Integer | N | N |
| 现有重症 | XYZZ | Integer | N | N |
| 累计确诊 | LJQZ | Integer | N | N |
| 境外输入 | JWSR | Integer | N | N |
| 累计治愈 | LJZY | Integer | N | N |
| 累计死亡 | LJSW | Integer | N | N |

\*\*\*\*南理工校园疫情表(CampusNum)---------------------1.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 |
| 日期 | Date | DATE | Y | N |
| 现有确诊 | XYQZ | Integer | N | N |
| 无症状 | WZZ | Integer | N | N |
| 现有疑似 | XYYS | Integer | N | N |
| 现有重症 | XYZZ | Integer | N | N |
| 累计确诊 | LJQZ | Integer | N | N |
| 境外输入 | JWSR | Integer | N | N |
| 累计治愈 | LJZY | Integer | N | N |
| 累计死亡 | LJSW | Integer | N | N |

# 第三章 南理工疫情信息管理系统详细设计

## 1.系统设计概述

根据《南理工疫情信息管理系统》的概要设计，本系统按照功能分解，如下图。



图3-1 模块设计

## 2. 详细设计概述

由于本系统采用了基于SSM体系结构的设计，即MVC的三层设计模式，并采用面向对象的Java语言以及JSP的脚本语言实现。在整个开发过程中，尽可能采用复用的原则，例如采用标签库，统一数据库的基本操作，统一结果显示等。

本文档的详细设计主要地按照SSM的MVC的三个层次分别编制视图层、控制层和模型层模块代码

**此项目由于代码量太多，页面太多，下面只给出登录模块的设计。**

## 3. 登录模块的详细设计（该模块可复用）

登录模块主要实现登录验证功能，登录模块的表示层、控制层和模型层三个层次的模块如表3-2所示。

表3-2 登陆模块的三层模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 视图 | | | 控制器 | 业务 |
| Bean | User | 客户标签 |
| Login | LoginForm | Html标签 | LoginServlet | LoginService |

### 3.1 视图层

视图层主要实现表示层的功能，视图层包括JSP组件、Form组件以及标签库等。

**3.1.1 JSP组件**

登录模块1个jsp页面

<%@ page import="edu.njust.entity.User" %>

<%@ page import="edu.njust.service.UserService" %>

<%@ page import="edu.njust.entity.Admin" %>

<%@ page import="java.io.PrintWriter" %><%--

Created by IntelliJ IDEA.

User:

To change this template use File | Settings | File Templates.

--%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<link href="css/login2.css" rel="stylesheet">

<html>

<head>

<title>Title</title>

</head>

<body>

<%

if(session.getAttribute("user")!=null){

session.setAttribute("user",null);

}

if(session.getAttribute("admin")!=null){

session.setAttribute("admin",null);

}

%>

<div align="center" class="title\_top">南京理工大学疫情信息管理平台</div>

<div class="container">

<div class="shadow len\_top">

<div class="len\_inside">

<div class="text-center">

<h1>登录</h1>

</div>

<form class="user" action="/CEIMS/servlet/LoginServlet" method="post" id="loginForm">

<!--

action="/CEIMS/servlet/LoginServlet"

-->

<div class="form-group">

<input type="text" class="control control-user" name="login\_account" placeholder="请输入账号" id="uname">

</div>

<div class="form-group">

<input type="password" class="control control-user" name="login\_pwd" placeholder="请输入密码" id="upwd">

</div>

<div class="form-group">

<span id="msg" style="font-size: 15px; color: red"></span> <br>

</div>

<button type="button" class="btn btn-primary btn-user btn-block" id="loginBtn">登录</button>

<hr>

</form>

<hr>

<div class="text-center">

<a class="small" href="register.jsp">注册账号!</a>

</div>

</div>

</div>

</div>

</body>

<script type="text/javascript" src="js/jquery-3.4.1.js"></script>

<script type="text/javascript">

$("#loginBtn").click(function (){

//获取用户ID、密码

var uname = $("#uname").val();

var upwd = $("#upwd").val();

if(isEmpty(uname)){

$("#msg").html("信息不能为空！");

return ;

}

if(isEmpty(upwd)){

$("#msg").html("信息不能为空！");

return ;

}

$("#loginForm").submit();

});

//判断字符串是否为空

function isEmpty(str){

if(str==null ||str.trim()==""){

return true;

}

else return false;

}

</script>

</html>

**3.1.2 JavaBean组件**

package edu.njust.entity;

public class User {

private String ID;

private String password;

private String name;

private String department;

public User() {

}

public User(String ID, String password, String name, String department) {

this.ID = ID;

this.password = password;

this.name = name;

this.department = department;

}

//留作修改密码用的数据结构

public User(String ID, String password) {

this.ID = ID;

this.password = password;

}

public String show(){

return "User id:"+this.ID+",password:"+this.password+",name:"+this.name+",department:"+this.department;

}

public String getName() {

return name;

}

public String getID() {

return ID;

}

public void setID(String ID) {

this.ID = ID;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public void setName(String name) {

this.name = name;

}

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

}

### 3.2 控制层

**下面是LoginServlet的伪代码描述。**

package edu.njust.servlet;

import edu.njust.entity.Admin;

import edu.njust.entity.User;

import edu.njust.service.UserService;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.io.PrintWriter;

public class LoginServlet extends HttpServlet {

public LoginServlet() {

}

@Override

public void destroy() {

super.destroy();

}

@Override

protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

super.service(req, resp);

}

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

//super.doGet(req, resp);

resp.setContentType("text/html;charset=utf-8");

PrintWriter out=resp.getWriter();

HttpSession session=req.getSession();

String ID=req.getParameter("regist\_user");

String name=req.getParameter("regist\_name");

String department=req.getParameter("regist\_department");

String password=req.getParameter("regist\_pwd");

User user=new User(ID,password,name,department);

UserService userService=new UserService();

if(userService.existUser(user.getID())){

out.print("<script type=\"text/javascript\" charset=\"UTF-8\">\n" +

" alert(\"用户已存在！\");\n" +

" document.location.href=\"../login.jsp\";\n" +

" </script>");

}

else{

if(userService.addUser(user)){

out.print("<script type=\"text/javascript\" charset=\"UTF-8\">\n" +

" alert(\"注册成功！\");\n" +

" document.location.href=\"../login.jsp\";\n" +

" </script>");

}else{

out.print("<script type=\"text/javascript\" charset=\"UTF-8\">\n" +

" alert(\"注册失败！\");\n" +

" document.location.href=\"../register.jsp\";\n" +

" </script>");

}

}

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

//super.doPost(req, resp);

resp.setContentType("text/html;charset=utf-8");

PrintWriter out=resp.getWriter();

HttpSession session=req.getSession();

String ID=req.getParameter("login\_account");

String password=req.getParameter("login\_pwd");

User user=new User(ID,password);

UserService userService=new UserService();

int failureCode=0;

if(!userService.existUser(user.getID())){

out.print("<script type=\"text/javascript\" charset=\"UTF-8\">\n" +

" alert(\"用户不存在！\");\n" +

" document.location.href=\"../login.jsp\";\n" +

" </script>");

}

else{

if(userService.login(user)){

user=userService.findUser(ID);

if(userService.isAdmin(user)) {//该用户为管理员

Admin admin=new Admin(user.getID(),user.getPassword(),user.getName(),user.getDepartment());

session.setAttribute("user",null);

session.setAttribute("admin",admin);

resp.sendRedirect("../adminIndex.jsp");

}

else{

session.setAttribute("user",user);

session.setAttribute("admin",null);

resp.sendRedirect("../userIndex.jsp");

}

}else{

//账号密码不符合

out.print("<script type=\"text/javascript\" charset=\"UTF-8\">\n" +

" alert(\"密码错误！\");\n" +

" document.location.href=\"../login.jsp\";\n" +

" </script>");

}

}

}

}

### 3.3 业务层

package edu.njust.service;

import edu.njust.dao.UserDAO;

import edu.njust.entity.User;

import java.io.IOException;

import java.util.List;

public class UserService {

/\*\*

\* 登录

\* @param user User 用户信息封装

\* @return boolean 操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean login(User user) throws IOException//根据用户账号密码判断是否登录成功

{

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

String password=userDAO.findPasswordByID(user.getID());

if(password.equals(user.getPassword())){

res=true;

}

System.out.println("UserService.login:User:"+user.getID()+" login "+(res?"successfully!":"failed!"));

}

return res;

}

/\*\*

\* 根据ID查找用户所有信息

\* @param id String 用户ID

\* @return User 用户信息封装

\* @throws IOException 数据库读写错误

\*/

public User findUser(String id) throws IOException {

User user=null;

if(id!=null){

UserDAO userDAO=new UserDAO();

user=userDAO.findUserByID(id);

System.out.println("UserService.findUser:Have found :"+user.show());

}

return user;

}

/\*\*

\* 某用户是否为管理员

\* @param user User 用户信息封装

\* @return boolean 该用户是否为管理员

\* @throws IOException 数据库读写错误

\*/

public boolean isAdmin(User user) throws IOException {

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.isAdmin(user.getID());

System.out.println("UserService.isAdmin:User:"+user.getID()+" is"+(res?" ":" not ")+"an Admin!");

}

return res;

}

/\*\*

\* 数据库中是否存在用户（该用户是否被注册）

\* @param id String 用户ID

\* @return boolean 是否存在

\* @throws IOException 数据库读写错误

\*/

public boolean existUser(String id) throws IOException {

boolean res=false;

if(id!=null){

UserDAO userDAO=new UserDAO();

User user=userDAO.findUserByID(id);

if(user!=null){

res=true;

}

System.out.println("UserService.existUser:"+(res?"":"Not ")+"exist user id:"+id);

}

return res;

}

/\*\*

\* 增加用户记录

\* @param user User 封装用户信息

\* @return boolean 操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean addUser(User user) throws IOException//根据账号密码注册用户

{

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.addUser(user);

System.out.println("UserService.addUser:Have"+(res?" ":" not ")+"inserted user whose id is:"+user.getID());

}

return res;

}

/\*\*

\* 增加管理员权限

\* @param user 被赋予管理员权限的用户信息封装

\* @return boolean 操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean addAdmin(User user) throws IOException {

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.addAdmin(user);

System.out.println("UserService.addAdmin:Have"+(res?" ":" not ")+"inserted admin whose id is:"+user.getID());

}

return res;

}

/\*\*

\* 修改用户信息

\* @param user User 用户信息封装

\* @return boolean 修改操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean modifyUser(User user) throws IOException {

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.modifyUserByID(user);

}

System.out.println("UserService.modifyUser:Have"+(res?" ":" not ")+"modified user!");

return res;

}

/\*\*

\* 取消管理员权限

\* @param user User 被取消权限的用户

\* @return 取消操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean cancelAdmin(User user) throws IOException {

boolean res=false;

if(user!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.cancelAdmin(user);

}

System.out.println("UserService.cancelAdmin:Have"+(res?" ":" not ")+"canceled admin!");

return res;

}

/\*\*

\* 根据学院批量重置密码

\* @param department String 学院名

\* @param afterModifyPswd String 重置后密码

\* @return boolean 操作成败

\* @throws IOException 数据库读写错误

\*/

public boolean batchModifyUser(String department, String afterModifyPswd) throws IOException//根据学院批量修改密码

{

boolean res=false;

if(department!=null && afterModifyPswd!=null){

UserDAO userDAO=new UserDAO();

res=userDAO.batchModifyUser(department, afterModifyPswd);

}

System.out.println("UserService.batchModifyUser:Have"+(res?" ":" not ")+"modified users' password to "+afterModifyPswd

+",whose department is:"+department);

return res;

}

/\*\*

\* 某学院的用户

\* @param dept String 学院

\* @return List<User> 某学院用户集合

\* @throws IOException 数据库读写错误

\*/

public List<User> findUserByDept(String dept) throws IOException {

List<User> res=new UserDAO().findUserByDept(dept);

System.out.println("UserService.findUserByDept:Have found "+res.size()+" users!");

return res;

}

/\*\*

\* 查找所有用户

\* @return List<User> 用户集合

\* @throws IOException 数据库读写错误

\*/

public List<User> findAllUser() throws IOException{

List<User> res=new UserDAO().findAllUsers();

System.out.println("UserService.findAllUsers:Have found "+res.size()+" users!");

return res;

}

}

## 4. 配置文件

1.使用的是J2EE 所以必有web.xml

2.使用MyBatis技术，所以有文件mybatis-config.xml

3.配置文件

3.1 web.xml

下面讲述web.xml配置文件的配置步骤和范围。

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<welcome-file-list>

<welcome-file>login.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>edu.njust.servlet.LoginServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>EveryServlet</servlet-name>

<servlet-class>edu.njust.servlet.EveryServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>OutInServlet</servlet-name>

<servlet-class>edu.njust.servlet.OutInServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>UserAddServlet</servlet-name>

<servlet-class>edu.njust.servlet.UserAddServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>UserModifyServlet</servlet-name>

<servlet-class>edu.njust.servlet.UserModifyServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>NumberServlet</servlet-name>

<servlet-class>edu.njust.servlet.NumberServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>TrackServlet</servlet-name>

<servlet-class>edu.njust.servlet.TrackServlet</servlet-class>

</servlet>

<servlet>

<servlet-name>RespondServlet</servlet-name>

<servlet-class>edu.njust.servlet.RespondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/servlet/LoginServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EveryServlet</servlet-name>

<url-pattern>/servlet/EveryServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>OutInServlet</servlet-name>

<url-pattern>/servlet/OutInServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>UserAddServlet</servlet-name>

<url-pattern>/servlet/UserAddServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>UserModifyServlet</servlet-name>

<url-pattern>/servlet/UserModifyServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>NumberServlet</servlet-name>

<url-pattern>/servlet/NumberServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>TrackServlet</servlet-name>

<url-pattern>/servlet/TrackServlet</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>RespondServlet</servlet-name>

<url-pattern>/servlet/RespondServlet</url-pattern>

</servlet-mapping>

</web-app>

3.2 mybatis-config.xml

MyBatis框架在启动的时候会读入其配置文件，根据它来创建和配置各种组件。MyBatis配置文件使得开发者可以灵活地组装和配置各个组件，提高了应用软件的可扩展性和灵活性。

下面给出mybatis-config.xml配置文件：

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE configuration

PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<environments default="development">

<environment id="development">

<transactionManager type="JDBC"/>

<dataSource type="POOLED">

<property name="driver" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/ceims"/>

<property name="username" value="root"/>

<property name="password" value="991011"/>

</dataSource>

</environment>

</environments>

<mappers>

<mapper resource="edu/njust/mapper/UserMapper.xml"/>

<mapper resource="edu/njust/mapper/EverydayReportMapper.xml"/>

<mapper resource="edu/njust/mapper/EverydayRespondMapper.xml"/>

<mapper resource="edu/njust/mapper/OutReportMapper.xml"/>

<mapper resource="edu/njust/mapper/TourMapper.xml"/>

<mapper resource="edu/njust/mapper/NationalMapper.xml"/>

<mapper resource="edu/njust/mapper/CampusMapper.xml"/>

</mappers>

</configuration>

# 第四章 南理工疫情信息管理系统编码实现

## 1.编码格式规范

### 1.1缩进排版

4个空格作为缩进排版的一个单位。

### 1.2行工度

1.2.1程序块要采用缩进风格编写，缩进的空格数为4个。说明：对于由开发工具自动生成的代码可以有不一致。

1.2.2相对独立的程序块之间、变量说明之后必须加空行。

1.2.3较长的语句（>80字符）要分成多行书写，长表达式要在低优先级操作符处划分新行，操作符放在新行之首，划分出的新行要进行适当的缩进，使排版整齐，语句可读。

1.2.4循环、判断等语句中若有较长的表达式或语句，则要进行适应的划分，长表达式要在低优先级操作符处划分新行，操作符放在新行之首。

1.2.5若函数或过程中的参数较长，则要进行适当的划分。

1.2.6不允许把多个短语句写在一行中，即一行只写一条语句。

1.2.7if、for、do、while、case、switch、default等语句自占一行，且if、for、 do、while等语句的执行语句部分无论多少都要加括号{}。

1.2.8对齐只使用空格键，不使用TAB键。

说明：以免用不同的编辑器阅读程序时，因TAB键所设置的空格数目不同而造成程序布局不整齐，不要使用BC作为编辑器合版本，因为BC会自动将8个空格变为一个TAB键，因此使用BC合入的版本大多会将缩进变乱。

1.2.9函数或过程的开始、结构的定义及循环、判断等语句中的代码都要采用缩进风格，case 语句下的情况处理语句也要遵从语句缩进要求。

1.2.10程序块的分界符（如C/C++语言的大括号‘{’和‘}’）应各独占一行并且位于同一列，同时与引用它们的语句左对齐。在函数体的开始、类的定义、结构的定义、枚举的定义以及if、for、do、while、switch、case语句中的程序都要采用如上的缩进方式。

1.2.11在两个以上的关键字、变量、常量进行对等操作时，它们之间的操作符之前、之后或者前后要加空格；进行非对等操作时，如果是关系密切的立即操作符（如->），后不应加空格。

说明：采用这种松散方式编写代码的目的是使代码更加清晰。

由于留空格所产生的清晰性是相对的，所以，在已经非常清晰的语句中没有必要再留空格，如果语句已足够清晰则括号内侧(即左括号后面和右括号前面)不需要加空格，多重括号间不必加空格，因为在C/C++语言中括号已经是 清晰的标志了。

在长语句中，如果需要加的空格非常多，那么应该保持整体清晰，而在局部不加空格。给操作符留空格时不要连续留两个以上空格。

1.2.12一般情况下，源程序有效注释量必须在20％以上。说明：注释的原则是有助于对程序的阅读理解，在该加的地方都加了，注释不宜太多也不能太少，注释语言必须准确、易懂、简洁。

2.命名规范

命名规范使程序更易读，从而更易于理解。它们也可以提供一些有关标识符功能的信息，以助于理解代码。

### 2.1包（Packages）

结构的功能要单一，是针对一种事务的抽象。说明：设计结构时应力争使结构代表一种现实事务的抽象，而不是同时代表多种。结构中的各元素应代表同一事务的不同侧面，而不应把描述没有关系或关系很弱的不同事务的元素放到同一结构中。

com.sxzlc.score.模块名.功能 本次写的内容

com.city.sxzlc.···使用之前的模块

### 2.2类（Classes）

类名是一个名词，采用大小写混合的方式，每个单词的首字母大写。尽量使你的类名简洁而富于描述。使用完整单词，避免缩写词（除非该缩写词被更广泛使用，像URL，HTML）。

### 2.3接口（Interfaces）

大小写规则与类名相似。

在同一软件产品内，应规划好接口部分标识符（变量、结构、函数及常量）的命名，防止编译、链接时产生冲突。说明：对接口部分的标识符应该有更严格限制，防止冲突。如可规定接口部分的变量与常量之前加上“模块”标识等。

### 2.4方法（Methods）

方法名是一个动词，采用大小写混合的方式，第一个单词的首字母大写，其后单词的首字母大写。

不要使用难懂的技巧性很高的语句，除非很有必要时。说明：高技巧语句不等于高效率的程序，实际上程序的效率关键在于算法。

防止将函数的参数作为工作变量。

说明：将函数的参数作为工作变量，有可能错误地改变参数内容，所以很危险。对必须改变的参数， 好先用局部变量代之， 后再将该局部变量的内容赋给该参数。示例：下函数的实现不太好。

### 2.5变量（Variabies）

采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写。变量名不应以下划线或美元符号开头，尽管这在语法上是允许的。变量名应简短且富于描述。变量名的选用应该易于记忆，即，能够指出其用途。尽量避免单个字符的变量名，除非是一次性的临时变量。临时变量通常被取名为i, j, k, m和n,它们一般用于整形；c, d, e,它们一般用于字符型。

标识符的命名要清晰、明了，有明确含义，同时使用完整的单词或大家基本可以理解的缩写，避免使人产生误解。

说明：较短的单词可通过去掉“元音”形成缩写；较长的单词可取单词的头几个字母形成缩写；一些单词有大家公认的缩写。

## 3.声明规范

### 3.1每行声明变量的数量

推荐一行一个声明，因为这样以利于写注释。亦即：

Int leve1;//indentation leve1

Int size;//size of table

要优于：

不要将不同类型变量的声明放在同一行，例如：

Int foo,fooarray[];//WRONG!

注意：上面的例子中，在类型和标识符之间放了一个空格。空格可使用制表符替代。

## 4.目录规范

开发环境是IDEA，开发之后的代码需要部署到Tomcat服务器环境上。所以开发环境的目录结构与运行环境的目录结构是一致的，只是在部署的运行环境中，可以不设置源代码的目录。

## 代码实例参见“登录操作的模块”

# 第五章 南理工疫情信息管理系统测试计划

## 1. 测试项目

南理工疫情信息管理系统。

### 1.1 测试项目的背景

为了系统能正常使用。

### 1.2 测试要点

1.能否正常登录，并跳转到相应的页面。

2.数据能否正常插入数据库。

3.一些插入的数据能否正常在下拉列表中显示。

4.每日健康申报模块的选择功能是否正常。

5.账号管理的权限限制功能是否正常。

6.管理员录入每日感染数据后，导出的结果是否正确。

### 1.3 测试内容

本测试分为单元测试、集成测试和系统测试。

定义：

单元测试（unit testing）：按照设定好的最小测试单元进行按单元测试，主要是测试程序代码，为的是确保各单元模块被正确的编译，单元的具体划分按不同的单位与不同的软件有不同，比如有具体到模块的测试，也有具体到类，函数的测试等。

集成测试（integration testing）：经过了单元测试后，将各单元组合成完整的体系，主要测试各模块间组合后的功能实现情况，以及模块接口连接的成功与否，数据传递的正确性等，其主要目的是检查软件单位之间的接口是否正确。根据集成测试计划，一边将模块或其他软件单位组合成系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。

系统测试（System Testing）：经过了单元测试和集成测试以后，我们要把软件系统搭建起来，按照软件规格说明书中所要求，测试软件其性能功能等是否和用户需求相符合，在系统中运行是否存在漏洞等。

## 2. 测试方法

单元测试为白盒测试，同时黑盒测试为辅。且在项目开发过程中已经融合了单元测试，故详细测试结果不再赘述。

集成测试：

对项目中的用户模块测试：

|  |  |  |
| --- | --- | --- |
| 测试内容 | 测试预期 | 测试结果 |
| 输入用户有效ID和正确密码并单击登录按钮 | 成功登录 | 成功 |
| 单击注册按钮 | 将出现注册的页面 | 成功 |
| 输入正确的每日健康信息数据并提交 | 提交成功 | 成功 |
| 输入正确的出校申请数据并提交 | 提交成功 | 成功 |
| 单击回到首页选项 | 将回到首页 | 成功 |
| 退出登录 | 退出成功 | 成功 |
| 未提交出校申请则单击提交返校申请 | 弹出“未提交出校申请”弹窗 | 成功 |
| 点击查看疫情数据 | 出现疫情数据界面 | 成功 |
| 单击查看回复 | 出现管理员已回复的信息 | 成功 |

对项目中管理员模块测试：

|  |  |  |
| --- | --- | --- |
| 测试内容 | 测试预期 | 测试结果 |
| 单击查看用户健康的按钮 | 将出现用户提交的每日健康数据 | 成功 |
| 点击账户管理按钮 | 出现账户管理窗口 | 成功 |
| 退出登录 | 退出成功 | 成功 |
| 点击查看运动轨迹按钮 | 出现查看用户运动轨迹窗口 | 成功 |
| 单击修改数据按钮 | 出现修改数据窗口 | 成功 |

系统测试：

对项目的兼容性进行测试

|  |  |  |
| --- | --- | --- |
| 测试内容 | 测试预期 | 测试结果 |
| 用Chrome打开该项目 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开登录页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开注册页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开每日信息申报页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开出校申请页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开返校反馈页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开查看疫情数据页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开账号管理页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开查看健康数据页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开查看用户运动轨迹页面 | 页面比例正常，与开发环境相似 | 成功 |
| 用Chrome打开回复信息页面 | 页面比例正常，与开发环境相似 | 成功 |

# 第六章 总结与展望

## 1.总结和展望

本系统从2020年9月课设期间和同学一起编写《南理工疫情信息管理系统》时开始，2016年时根据软件工程课程综合作业要求，重新设计和编写了《南理工疫情信息管理系统》，也就是现在的程序。

在技术方面，当时只是简单的使用了jsoup抓包和mybatis技术，将程序划分为用户模块，管理员模块。

依赖注入（控制翻转），只是了解了这个技术，但是感觉配置相当费事，如果以后自己开发web，应该不会使用spring。只是使用mybatis就行了。同时在页面布局方面使用了tiles标签，这个标签给我带来的最大好处就是解决了由于使用switchAction而带来的找不到form Action的问题。以后如果使用switchAction，一定要配合使用tiles标签。

这个系统的层次结构很清晰。这个系统只是一个学习的产物，使用了很多新的技术，包的命名，接口的使用也都符合规范。

## 2.感想

对于IDEA工具的使用加强了，更重要的是了解到了工具软件的重要性，在做大作业之前，我对工具软件是极为漠视的。但是在做大作业的过程中，我的思想发生了巨大的改变。意识到了工具软件的重要作用。

# 参考文献

[1].Struts 网路编程例学与实践 清华大学出版社。 编著 温涛

[2].J2EE 项目实训----Spring框架技术。清华大学出版社，

编著 杨少波 顾益军等

[3].java学习笔记 清华大学出版社，编著 良葛格

[4].javaEE编程技术 清华大学出版社，北京交通大学出版社

编著 郝玉龙 姜华

[4] .javaEE编程技术 清华大学出版社，北京交通大学出版社

编著 郝玉龙 姜华

[5] .javaWeb开发详解 电子工业出版社 编著 孙鑫

[6] .软件工程 大连理工大学城市学院 编著 张应博

[7] .Frand M.Carrano,Janet J.Prichard .著 韩志宏 译.数据抽象和问题求解——JAVA语言描述.清华大学出版社2005年4月

[8] .Ian Sommerville著.程成等译.软件工程.2003年1月第1版.机械工业出版社.

[9] . (美)D. 阿卢尔: 《J2EE核心模式》，机械工业出版社，2002

[10] .Ken Arnold, Jams Gosling, David Holmes: Java programming language , Post & Telecompress, 2006

[11] .Mary Campione, Kathy Walrath, Alison Huml : Java Tutorlal , China Machine Press, 2004

[12] .John Lewis: Java Softwate Solutions, Publishing House Of Electronics Industry, 2007

**代码光盘粘贴处**

****