|  |  |  |  |
| --- | --- | --- | --- |
|  | 人员 | 时间 | 备注 |
| 编写 | 贾智勇、徐可辰 | 2023.4.21 |  |
| 审核 | 贾智勇、徐可辰 | 2023.4.21 |  |
|  |  |  |  |
|  |  |  |  |

华中科技大学网上选课系统

详细设计说明书

**V1.0**

1引言 4

1.1编写目的 4

1.2背景 4

1.3定义 4

1.4参考资料 5

2系统的程序结构 5

3选课模块设计说明 6

3.1程序描述 6

3.2功能 6

3.3性能 6

3.4输入/输出项 6

3.5算法 7

3.6流程逻辑 7

3.7接口 7

3.8注释设计 8

3.9尚未解决的问题 8

4授课管理模块设计说明 8

4.1程序描述 8

4.2功能 8

4.3性能 8

4.4输入/输出项 9

4.5流程逻辑 9

4.6接口 9

4.7注释设计 10

4.8尚未解决的问题 10

5学生管理模块设计说明 10

5.1程序描述 10

5.2功能 11

5.3性能 11

5.4输入/输出项 11

5.5流程逻辑 11

5.6接口 11

5.7注释设计 12

5.8尚未解决的问题 12

6课程管理模块设计说明 12

6.1程序描述 12

6.2功能 12

6.4输入/输出项 13

6.5流程逻辑 13

6.6接口 13

6.7注释设计 14

6.8尚未解决的问题 14

7通用模块设计说明 14

7.1程序描述 14

7.2功能 14

7.4输入/输出项 15

7.5流程逻辑 15

7.6接口 15

7.7注释设计 16

7.8尚未解决的问题 17

1引言

1.1编写目的

在概要设计文档中，阐述了系统若干模块的划分方式、各模块间的接口。模块间消息的传递等设计。在本文档中，则主要根据概要设计文档，对每个模块的设计进行细化，确定每个模块的算法，内部的数据组织等，供开发人员参考，让设计可以直接应用到具体实现中，同时保证项目的技术方案和实施方案是符合架构设计文档和概要设计文档的。在后期的评审过程中，评审专家也可以参考此文档来评审系统的实现程度等。

1.2背景

待开发软件系统的名称：华中科技大学网上选课系统

项目任务提出者：贾智勇、徐可辰

用户：华中科技大学在校本科生，教师，教务人员

软件使用硬件平台：

服务器端需安装 Linux 5.16，客户端需要安装浏览器。

2） 应用程序服务器要保证8G的运行内存，30G以上的硬盘空间。数据库服务器需要保证400G以上的硬盘空间。

华中科技大学网上选课系统是面向在校本科生，教师和教务人员设计的一款Web应用，学生可使用系统完成本科四年专业课，公选课，体育课的选课，以及成绩查询等工作。教师可使用系统完成学生信息的查看，学生成绩的操作，以及申请课程的工作。教务人员主要是管理学生教师信息，安排课程，审批教师申请等工作。本系统用户类型较多，各类用户的权限差异较大，因此，本系统在基于各类用户安全操作的基础上设计了很多质量属性目标。

1.3定义

**1.3.1 MySQL**

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，在Web应用方面，MySQL是最好的RDBMS应用软件，其具有体积小，速度快，总体拥有成本低，开放源码并为多种编程语言提供了API等特点。

**1.3.2 HTML**

HyperText Mark-up Language, 超文本标记语言或超文本链接标示语言。用来表示网页的大体结构。HTML是一种文本标记语言，由不同的浏览器解释进行显示。

**1.3.3 CSS**

Cascading Style Sheet,层叠样式表。用于给修改或者增加网页的样式，丰富页面的表示形式，增强页面的交互性。

**1.3.4 JSP**

Java Server Pages,是一个动态网页技术标准，是在传统的HTML网页中插入Java程序段，从而形成JSP文件，并且用JSP开发的web应用具有跨平台的特性。

**1.3.5 JavaScript**

JavaScript 是一门跨平台、面向对象的轻量级脚本语言。

**1.3.6 Bootstrap**

Bootstrap是基于HTML,CSS和JavaScript的前端框架。

**1.3.7 JQuery**

jQuery是一个快速、简洁的JavaScript框架，它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

**1.3.8 Servlet**

Java Servlet，称为小服务程序或服务连接器，用Java编写的服务器端程序，主要功能在于交互式地浏览和修改数据，生成动态Web内容。

**1.3.9 JavaBean**

JavaBean 是一种JAVA语言写成的可重用组件。

**1.3.10 IPO图**

IPO图：在计算机领域IPO是指结构化设计中变换型结构的输入（Input）、加工（Processing）、输出（Output）。IPO图是对每个模块进行详细设计的工具，它是输入加工输出(INPUT PROCESS OUTPUT)图的简称，它是由美国IBM公司发起并完善起来的一种工具。

1.4参考资料

华中科技大学网上选课系统概要设计说明书V1.0

华中科技大学网上选课系统软件需求说明书V1.0

华中科技大学网上选课系统数据要求说明书V1.0

2系统的程序结构

本系统采用分层设计的思想，并使用浏览器/服务器（B/S）结构实现。系统建立在Linux操作系统平台之上，总体分为三层：由web浏览器所表示的表示层，由应用服务器实现的业务逻辑层，以及由MySQL数据服务器所实现的数据访问层。表示层：由HTML，CSS，JSP，JavaScript共同实现。使用Bootstrap，JQuery的框架。接受页面的请求和信息输入，接受并显示业务层的信息反馈。业务层：由Servlet控制器和JavaBean共同构成。数据访问层：主要由MySQL服务器和数据库组成。

B/S分层结构将系统的核心功能集中到了服务器上，客户通过浏览器便可以轻松的实现对于应用程序的访问，而且系统的维护和升级方式相对简单，维护成本较低。并且采用分层结构可以实现对于表示层和业务逻辑层的分离，能够保证应用服务逻辑的一致性和稳定性、结构的开放性、功能的可扩展性和可维护性、开发的可并行性。

本系统的主要目的是对以高等教育学校为服务对象的选课管理环境中，为选修课程、开设选修课、选修课成绩评定、管理选修课等提供数字化服务支持。实现学生的选修课程、查看课程相关信息，教师开设选修课、录入成绩，教务处人员管理师生信息、安排选修课时间等功能。

本系统主要分为三个大的子系统：学生系统、教师系统和教务系统，分别学生、教师和教务人员三类用户。学生系统可以分为选课模块和查询模块，教师系统可以分为授课管理模块和查询模块，教务系统可以分为学生管理模块、教师管理模块和课程管理模块，还有通用模块。其中学生与教师的查询模块流程简单，在文档中不进行单独的详细说明。

通过对系统模块和实体的分析，得到如下的类图：

2.1.1 通用信息类图

该部分为三个子系统的通用模块，主要负责处理三类用户共有的基础功能，包括：登录、找回密码、修改密码、修改个人资料等。

2.1.2 选课信息类图

这个部分主要负责学生选课相关的流程，包括：选修课程、退选课程、补选课程、查询课程的情况、查询学分等。

2.1.3 授课管理模块类图

此部分主要是负责授课管理模块相关的功能，包括申请开设新课程、查看教授的课程、查看教授班级学生名单等

2.1.4 教务子系统整体类图

此部分主要是教务子系统中涉及的功能，包括审核新课程、安排和管理课程清单、审核学生成绩、管理教师和学生的基本信息等。具体如下图2-5所示：

3选课模块设计说明

3.1程序描述

选课模块主要是处理学生在选课时间段内对课程的选修操作，和在退选时间段对课程的退选操作。学生在登录后，系统判断该学生所在学院是否处于可选修状态，然后将学生选择的课程记录在系统中，选修时间结束后，按照规则从中挑出中签的学生，并将结果通知学生。

3.2功能

选课模块主要包括选修功能、退选功能和通知功能等。具体功能如下图3-1所示：

3.3性能

**3.3.1** 时间特性要求

系统的速度要在用户可接受的范围内，在学生选修课程的时候，因为对学院进行了选修时间段的分流，而且选课时间先后不影响中签的概率，所以速度性能的要求不会太高。

**3.3.2** 可靠性

系统要有较高的可靠性，至少要保证在选课高峰期间的时段内，保持可用。

**3.3.3** 灵活性

系统要有良好的接口，以对接校方的通用平台。

3.4输入/输出项

根据上面的模块功能结构图，表示出该模块各个功能的输入/出项。具体情况如下：

1、选修课程：系统通过用户提交的课程ID号和用户ID号，完成选修操作，并返回是否选修成功。

1. 退选课程：系统通过用户提交的课程ID号和用户ID号，完成退选操作，并返回是否退选成功。
2. 补选课程：和选修课程相同

4 、通知功能：以邮件的形式通知学生最终的抽签结果。

5、查询学分：输入用户ID，查询该专业的总所需学分，计算已修课程的总分，进而计算剩余所需学分

3.5算法

大四有单独的提前批次中进行选修，其他年级选修时抽签按照以下算法抽取：

将所有符合选修条件的学生分为四个优先级（优先级从1-4依次降低）：

1. 大三未修满学分者
2. 大一/大二未修满学分者
3. 大三已修满学分者
4. 大二已修满学分者

1-4级所对应的权重分别为4-1，根据权重向数据池中添加相应数量的学号，如：等级为1的学生A的权重为4，那么A同学的学号将在数据池中有四个副本。因此，优先级越高，权重越大，在数据池中被随机抽取到的几率越大。

补选课程时，不再使用上述规则，选一门课直接通过，直到选满为止

3.6流程逻辑

3.7接口

用图的形式说明本程序所隶属的上一层模块及隶属于本程序的下一层模块、子程序，说明参数赋值和调用方式，说明与本程序相直接关联的数据结构（数据库、数据文卷）。

**1**．**ChooseServlet**

主要用于处理选修操作，上层为Choose.jsp，下层为学生的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| courseID | String | 选修课程的ID |
| Student | Student | 发起选课的学生用户的实例 |
| type | int | 被选课所属的类别 |

**2**．**WithdrawServlet**

主要用于处理退选操作，上层为Withdraw.jsp，下层为学生的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| courseID | String | 退选课程的ID |
| Student | Student | 发起退选申请的学生用户的实例 |

**3**．**void SendNotice(String[])**

参数列表：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| coursesIDList | String[] | 全部课程的ID列表 |

3.8注释设计

**3.8.1** 加在模块首部的注释

选课模块,主要包括:选修功能、退选功能和通知功能

**3.8.2** 对各变量的功能、范围、缺省条件等所加的注释

**ChooseServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**courseID**：获取的所选课程ID

**stuID**：获取的学生学号

**classification**：获取的课程类型

**WithdrawServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**stu**：用于记录用户的实例

**courseID**：获取的所选课程ID

**stuID**：获取的学生学号

**classification**：获取的课程类型

3.9尚未解决的问题

当前系统无法适用于一些浏览器。受限于系统和技术水平，无法让系统适用于所有种类的浏览器。但可以在无法用于浏览器时，显示一个提示页面，告知用户应该使用什么浏览器。

4授课管理模块设计说明

4.1程序描述

授课管理模块主要是为教师提供与所教授课程相关的管理服务，包括开设新的课程、对已开设课程的学生成绩的管理等功能。

4.2功能

4.3性能

**4.3.1** 易用性

此模块供教师使用，使用频率较高，要求设计时操作简单方便，易于学习和掌握。

**4.3.2** 安全性

此模块中涉及到学生成绩和学生信息等敏感信息，要保证安全性，在数据传送过程和存储中，都要做好安全性防护，防止被不法分子窃取、破坏或篡改。

**4.3.3** 时间特性要求

使用此模块的用户较少，且时间不集中，所以对时间特性的要求不高。

4.4输入/输出项

根据上面的模块功能结构图，表示出该模块各个功能的输入/出项。具体情况如下：

1、申请开设新课程：当请求通过财务部门审核后，需要根据财务部门操作人员设定的请求所需的审批级别，由相应级别的领导依次审批，最终记录审批结果。

2、管理学生课程成绩：查询所有开始方式相关信息。

4.5流程逻辑

图4-5 管理学生课程成绩流程图

4.6接口

**1**．**ApplyServlet**

主要用于处理对新课程的申请操作，上层为Apply.jsp，下层为教师的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| teachID | String | 申请发起教师的ID |
| courseName | String | 新课程的名字 |
| detail | String | 课程的详细描述 |
| credit | float | 课程学分 |
| hours | int | 课程学时 |

**2**．**AddScoreServlet**

主要用于处理成绩录入操作，上层为AddScore.jsp，下层为教师的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| courseID | String | 退选课程的ID |
| teachID | String | 录入成绩的教师ID |
| stuID | String[] | 学生学号的数组 |
| scores | float[] | 学生成绩的数组 |
| inputMethod | int | 导入方式，1-手动导入，2-Excel导入 |
| scoreFile | FILE | 成绩文件 |

4.7注释设计

**4.7.1** 加在模块首部的注释

授课管理模块，主要包括开设新的课程、对已开设课程的学生成绩的管理等功能。

**4.7.2** 对各变量的功能、范围、缺省条件等所加的注释

**ApplyServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**teacherID**：用于记录提交申请的教师ID

**detail**：用于记录课程的详细描述

**AddScoreServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**scores**：用于记录本次提交的成绩

**stuIDList**：用于记录本次提交了成绩的对应学生名单

4.8尚未解决的问题

无。

5学生管理模块设计说明

5.1程序描述

学生管理模块主要是给教务处工作人员提供对学生基本信息的管理、对学生成绩的审核管理等功能。

5.2功能

5.3性能

**5.3.1** 安全性

此模块的功能涉及到大量敏感信息，对安全性的要求较高。要保证安全性，在数据传送过程和存储中，都要做好安全性防护，防止被不法分子窃取、破坏或篡改。

**5.3.2** 可用性

此模块使用频率较高，而且一旦不能正常工作，对教务工作影响较大，所以对模块可用性的要求高，要保证较长时间的可用状态。

5.4输入/输出项

根据上面的模块功能结构图，表示出该模块各个功能的输入/出项。具体情况如下：

1、基本信息管理：管理学生、教师、教务人员三类用户的基本信息。

2、成绩审核管理：审核来自教师录入的成绩单或者通过教师修改成绩的申请。

5.5流程逻辑

5.6接口

**1**．**InfoManageServlet**

主要用于处理用户基本信息管理，上层为InfoManage.jsp，下层为管理员的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| userID | String | 需要修改信息的用户的ID |
| adminID | String | 修改操作执行者ID |

**2**．**ScoreManageServlet**

主要用于处理成绩录入审核操作，上层为ScoreManage.jsp，下层为管理员的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| teachID | String | 提交成绩的教师的ID |
| scores | float[] | 成绩列表数组 |
| stuID | String[] | 成绩列表对应的学号列表数组 |
| adminID | String | 成绩审核者ID |

5.7注释设计

**5.7.1** 加在模块首部的注释

学生管理模块，包括学生基本信息的管理、对学生成绩的审核管理等功能。

**5.7.2** 对各变量的功能、范围、缺省条件等所加的注释

**InfoManageServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**userID**：用于记录需要修改信息的用户的ID

**adminID**：用于记录修改操作执行者ID

**ScoreManageServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**scores**：用于记录成绩列表的数组

**stuID**：用于记录成绩列表对应的学号列表的数组

**adminID**：用于记录成绩审核者ID

**time**：用于记录审核申请的时间

**status**：用于记录申请审核状态

5.8尚未解决的问题

无。

6课程管理模块设计说明

6.1程序描述

课程管理模块主要为教务人员安排课程、审理教师开课申请和安排各学院的选课时间提供数字化的管理功能。

6.2功能

课程管理模块功能主要包括新课程申请审批功能、排课功能、选课时间安排功能等。

6.3性能

**6.3.1** 安全性

此模块的功能涉及到大量敏感信息，对安全性的要求较高。要保证安全性，在数据传送过程和存储中，都要做好安全性防护，防止被不法分子窃取、破坏或篡改。

**6.3.2** 可用性

此模块使用频率较高，而且一旦不能正常工作，对教务工作影响较大，所以对模块可用性的要求高，要保证较长时间的可用状态。

6.4输入/输出项

根据上面的模块功能结构图，表示出该模块各个功能的输入/出项。具体情况如下：

1、新课程申请审批：审批教师开设新选修课的申请。

2、安排课程：调整课程的上课时间和地点。

3、选课时间安排：对选课的学院进行分流，安排不同时段内可以选课的学院。

6.5流程逻辑

6.6接口

**1**．**ApplyManageServlet**

主要用于处理对开课申请的审核操作，上层为ApplyManage.jsp，下层为管理员的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| teachID | String | 申请发起教师的ID |
| applySheet | Aplication | 封装申请具体内容的类 |
| adminID | String | 申请审核者ID |

**2**．**ArrangeServlet**

主要用于处理安排课程的时间和地点，安排学员选课批次，上层为Arrange.jsp，下层为管理员的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| adminID | String | 操作人员的ID |
| semester | String | 要排课的学期 |
| function | int | 功能选择，1-安排课程时间地点，2-安排选课时间 |

6.7注释设计

**6.7.1** 加在模块首部的注释

课程管理模块，包括安排课程、审理教师开课申请和安排各学院的选课时间的功能。

**6.7.2** 对各变量的功能、范围、缺省条件等所加的注释

**ApplyManageServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**teachID**：用于记录申请发起教师的ID

**applySheet**：用于记录封装申请具体内容的类

**adminID**：用于记录申请审核者ID

**time**：用于记录审核申请的时间

**status**：用于记录申请审核状态

**ArrangeServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**adminID**：用于记录操作人员的ID

**function**：用于记录用户选择的功能

**courseList**：用于记录要安排的课程

**place**：用于记录课程地点

**period**：选课时段

**colleges**：学院列表

6.8尚未解决的问题

无。

7通用模块设计说明

7.1程序描述

通用模块主要为三类用户提供公共服务，包括登录、注册、修改密码、找回密码等基础服务。

7.2功能

通用模块功能主要包括登录、注册、修改密码和找回密码等功能。

7.3性能

**7.3.1** 安全性

此模块的功能涉及到对系统的访问权限的问题，对安全性的要求较高。要保证安全性，在用户名和密码等数据传送过程和存储中，都要做好安全性防护，防止被不法分子窃取、破坏或篡改。

**7.3.2** 可用性

此模块使用频率较高，而且一旦不能正常工作，会造成系统无法访问，影响很大，所以对模块可用性的要求高，要保证较长时间的可用状态。

**7.3.3** 可维护性

作为公用的底层模块，要有良好的可维护性，能在出现故障时快速修复，把损失控制在较小范围内。

7.4输入/输出项

根据上面的模块功能结构图，表示出该模块各个功能的输入/出项。具体情况如下：

1、登录：根据用户提供的密码和账号登录本系统。

2、注册：根据用户填写的注册信息与系统的预录信息比对，激活信息符合的账户，并将密码更新为用户重新设定的密码。

3、修改密码：用户在登录状态下，验证过原密码后可以修改密码。

4、找回密码：提供用户通过绑定的邮箱修改原密码的功能。

7.5流程逻辑

7.6接口

**1**．**PsdServlet**

主要用于处理用户登录，上层为index.jsp，下层为三类用户的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| userID | String | 登录者的ID，即账户名 |
| password | String | 密码 |
| role | int | 登录者的身份标识 |

**2**．**RegisterServlet**

主要用于处理新用户注册，上层为index.jsp，下层为三类用户的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| userID | String | 注册者的ID，即账户名 |
| password | String | 注册后重置的密码 |
| role | int | 注册者的身份标识 |
| IDNumber | String | 身份证号码 |
| AdmissionNum | String | 准考证号（学生注册选用） |

**3**．**ChangePsdServlet**

主要用于处理修改密码操作，上层为PsdChange.jsp，下层为三类用户的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| userID | String | 用户的ID，即账户名 |
| oldPsd | String | 原密码 |
| newPsd | String | 新密码 |

**4**．**FindPsdServlet**

主要用于处理找回密码操作，上层为PsdFind.jsp，下层为三类用户的service类。

提交表单内容包括：

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| userID | String | 登录者的ID，即账户名 |
| emailAddr | String | 用户绑定的邮箱地址 |
| question | String | 用户设置的密保问题 |
| answer | String | 用户对密保问题的回答 |

7.7注释设计

**7.7.1** 加在模块首部的注释

通用模块，主要包括登录、注册、修改密码、找回密码等基础服务。

**7.7.2** 对各变量的功能、范围、缺省条件等所加的注释

**PsdServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**userID**：用于记录登录者的ID

**password**：用于记录登录者填写的密码

**role**：用于记录登录者的身份

**time**：用于记录登录时间

**RegisterServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**userID**：用于记录注册者的ID

**password**：用于记录注册者重置的密码

**IDNumber**：用于记录注册人的身份证号码

**AdmissionNum**：用于记录注册人的准考证号（学生注册选用）

**ChangePsdServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**userID**：用于记录登录者的ID

**oldPsd**：用于记录原密码

**newPsd**：用于记录新密码

**FindPsdServlet**中**doPost()**函数局部变量**:**

**url**：用于记录servlet要跳转的下一个地址。

**session**：用于记录用户本次会话

**userID**：用于记录登录者的ID

**emailAddr**：用于记录用户绑定的邮箱地址

**question**：用于记录用户设置的密保问题

**answer**：用于记录用户对密保问题的回答

7.8尚未解决的问题

无。