**教室、会议预约管理系统**

**详细设计分析报告**

**软件名称**： 会议室、教室管理系统

**学生姓名**： 马瑄

**专 业**： 计算机科学与技术

**班 级**： 计1402

**学 号**： 41455030

**组 号**： 16

**目录**

[1 引言 3](#_Toc16043523)

[1.1 编写目的 3](#_Toc16043524)

[1.2 背景 3](#_Toc16043525)

[1.3 定义 3](#_Toc16043526)

[1.4 参考资料 3](#_Toc16043527)

[2 设计模式 3](#_Toc16043523)

[2.1 概述 3](#_Toc16043529)

[2.2 说明 4](#_Toc16043530)

[2.3 系统设计 4](#_Toc16043531)

[3 人机界面设计 5](#_Toc16043533)

[4 模块设计 8](#_Toc16043533)

[4.1 登陆模块 8](#_Toc16043534)

[4.2 教室查询模块 11](#_Toc16043535)

[4.3 教室预订模块 13](#_Toc16043540)

[4.4 审核 15](#_Toc16043540)

[4.5 信息管理模块 17](#_Toc16043540)

[5 类图 20](#_Toc16043533)

[6 测试计划 20](#_Toc16043533)

[6.1 测试阶段 20](#_Toc16043540)

[6.2 测试要点 21](#_Toc16043540)

[6.2.1测试内容 21](#_Toc16043540)

[6.2.2精度需求 21](#_Toc16043540)

[6.2.3时间要求 21](#_Toc16043540)

[6.2.4灵活性 21](#_Toc16043540)

[6.2.5故障处理 22](#_Toc16043540)

1. **引言**

**1.1编写目的**

详细设计的根本目标是确定应该怎样具体的实现所要求的系统，要设计出程序的“蓝图”，以后程序员将根据 这个蓝图写出实际的代码程序。因此，程序设计从根本上决定了最终的程序代码的质量。详细设计要在逻辑上正确的实现每个模块的功能。

**1.2背景**

项目名称：会议教室管理系统

项目任务开发成员：马瑄、任琳琳、宛嵇祥、张菁彧

项目任务用户：管理员、用户

项目业务范围：包括学校可租借教室会议室及其相关设备的管理、预订服务。

**1.3定义**

静态数据－－系统固化在内的描述系统实现功能的一部分数据。

动态数据－－在软件运行过程中用户输入的后系统输出给用户的一部分数据，也就是系统要处理的数据。

C/S架构——又称主从式架构或客户端-服务器结构（Client/Server），服务器被动地接收来自客户端的请求，处理请求并传回结果。

**1.4参考资料**

<<软件工程导论>>（第六版）　 ------张海藩 编著 清华大学出版社

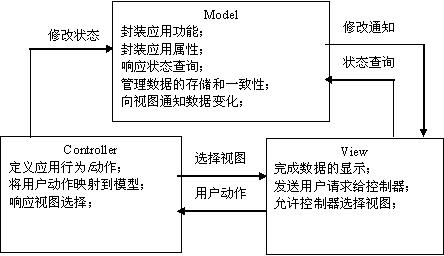
<<软件工程>>（第九版） ------张海藩编著人民邮电出版

1. **设计模式**

**2.1概述**

在本次系统的设计中，我们使用MVC架构模型。

MVC架构把数据处理，程序输入输出控制及数据显示分离开来，并且描述了不同部件的对象间的通信方式。使得软件可维护性，可扩展性，灵活性以及封装性大大提高；MVC把系统的组成分解为M（模型）、V（视图）、C（控制器）三种部件。视图表示数据在屏幕上的显示。控制器提供处理过程控制，它在模型和视图之间起连接作用。控制器本身不输出任何信息和做任何处理，它只负责把用户的请求转成针对Model的操作，和调用相应的视图来显示Model处理后的数据。三者之间关系如下：



**2.2说明**

a）View：视图用于管理信息的显示，它提供用户交互界面。使用多个包含单页面显示的用户部件。view部分大致处理流程如下：首先，定义view的布局；由页面布局策略类初始化并加载页面；用户根据自己的配置进行初始化，加载校验器并设置参数；用户提交后，发送用户请求给控制器。

b）Controller: Controller控制器是Model与View之间沟通的桥梁，它可以分派用户的请求并选择恰当的视图以用于显示，同时它也可以解释用户的输入并将它们映射为模型层可执行的操作。

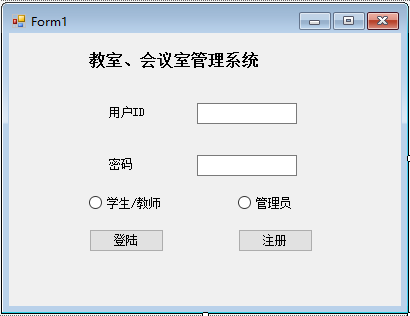
c)Model:model对象代表了数据容器。model提供访问显示数据的操作，提供控制内部行为的操作以及其他必要的操作接口。和View之间创建双向绑定使得View与Model同步变化。

**2.3系统设计**

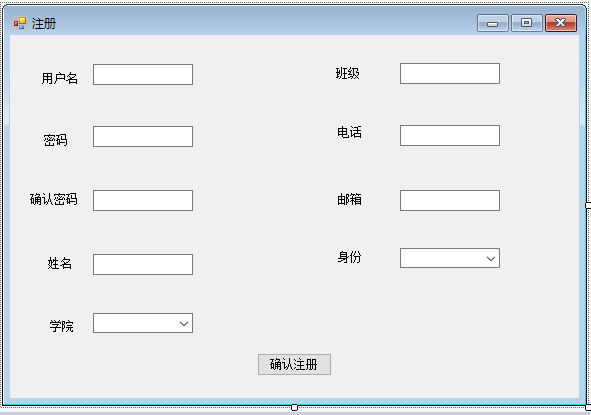


1. **人机界面设计**

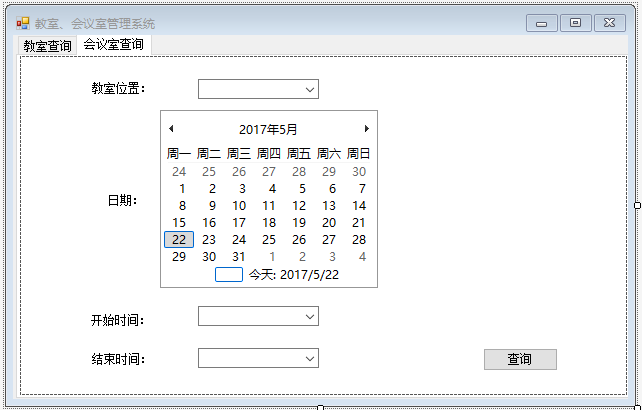
3.1登陆界面设计



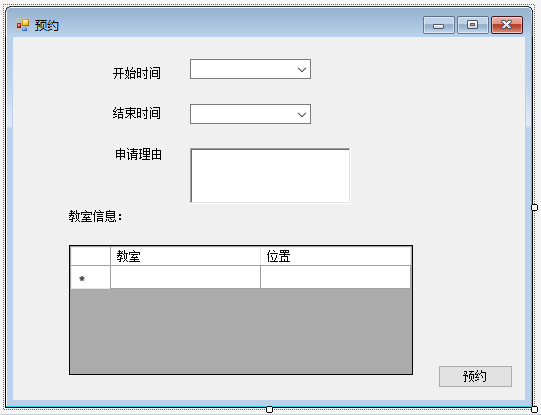
注册界面：



查询界面：



预约界面：



管理员界面





1. **模块设计**

**4.1登陆模块：**

a）模块描述

用户需要借用或预约教室时，登入该系统。

b）功能

①用户ID、密码验证，检查用户身份是否正确，登入主页面。

②若用户无账号，注册用户，填写相应信息成为系统成员。

c）输入项

普通用户登录：

|  |  |
| --- | --- |
| 学号/工号 | varchar（10） |
| 密码 | Varchar（10） |

注册用户：

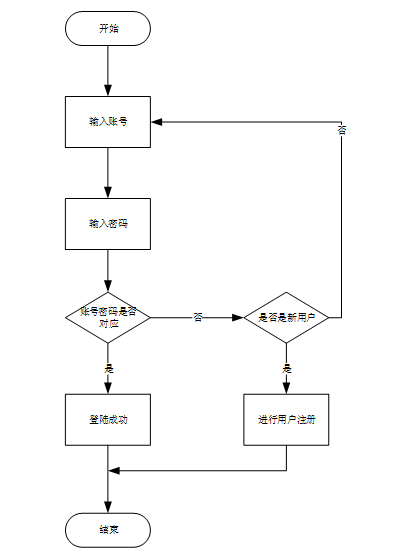
|  |  |
| --- | --- |
| 名称 | 类型 |
| 学号/工号 | varchar（10） |
| 密码 | Varchar（10） |
| 姓名 | varchar（10） |
| 学院 | varchar（20） |
| 班级 | varchar（10） |
| 电话 | varchar（20） |
| 邮箱 | varchar（20） |
| 是否为学生 | Bool |

d）输出项

用户信息如下，传送信息到预订模块。

|  |  |
| --- | --- |
| 学号/工号 | varchar（10） |
| 姓名 | varchar（10） |
| 学院 | varchar（20） |
| 班级 | varchar（10） |

e）模块程序流程图



f）类设计



说明：登录模块要与数据库相连

Verify：验证用户名与密码是否正确，正确则登入系统，错误选择注册或退出系统。

Register：注册页面，注册信息通过C/S模式传到数据库服务器，进行数据库增加成员的操作。

Login：登入系统，与verify配合使用。

Logout：注销用户，退出系统。以便下次用户登入。

View：预约人机交互页面。

**4.2教室查询模块**

a）模块描述

用户根据查询得到教室的使用情况。

b）功能

①学生通过时间、位置进行教室查找；

②系统根据学生信息寻找符合要求的教室和最佳教室（待定），得出信息详细列表。

c）输入项

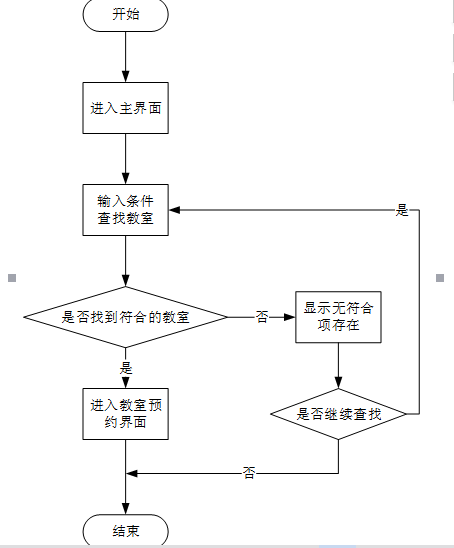
|  |  |
| --- | --- |
| 名称 | 类型 |
| 编号 | varchar（10） |
| 申请人学号、工号 | varchar（10） |
| 教室编号 | varchar（10） |
| 开始时间 | Date |
| 结束时间 | Date |

d）输出项：教室状态信息并进入教室预订模块

|  |  |
| --- | --- |
| 名称 | 类型 |
| 状态 | Int |

状态：1（审核通过）；2（等待审核）；3（审核未通过）。

e）模块程序流程图



f）类设计



说明：预约模块同样与数据库DB相连，实现教室的查询与匹配。

Select：通过属性中的查询，选择符合条件的教室；

Subscribe：选定合适的教室进行预订，进入预订模块；

View：预约查询人机界面（符合条件的教室列表）。

**4.3教室预订模块**

a）模块描述

用户对权限内、空闲且符合条件的教室进行预订。

b）功能

用户对所选的教室信息进行预订，预订教室，选定预定时间。询问是否预订教室，用户确认后传送到审核模块。

c）输入项

|  |  |
| --- | --- |
| 名称 | 类型 |
| 编号 | varchar（10） |
| 申请人学号、工号 | varchar（10） |
| 教室编号 | varchar（10） |
| 开始时间 | Date |
| 结束时间 | Date |
| 申请理由 | varchar（30） |

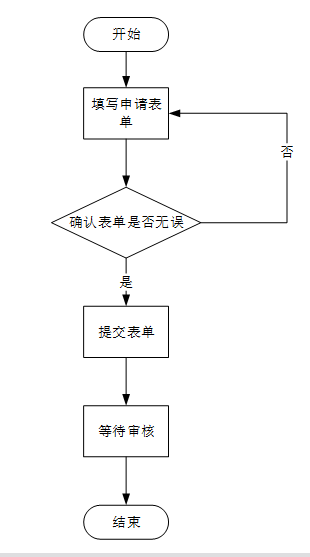
再次输入预订信息并进行确认。

d）输出项

申请表信息传送到审核模块

|  |  |
| --- | --- |
| 名称 | 类型 |
| 编号 | varchar（10） |
| 申请人学号、工号 | varchar（10） |
| 教室编号 | varchar（10） |
| 开始时间 | Date |
| 结束时间 | Date |
| 申请理由 | varchar（30） |
| 状态 | Int |

e）模块程序流程图



f）类设计



说明：

Subscribe：提交申请表，这个操作中可以再次修改所选教室时间，根据所选信息完成申请表。

Confirm：确认提交申请表，提交到审核模块。

View：预订模块的人机交互页面，尽量让用户选定界面内的信息就可以完成预订。

**4.4审核模块**

a）模块描述

管理员对教室借用信息进行审核，批准或驳回

b）功能

①管理员对用户的申请表进行审核，根据实际情况进行批准；

②管理员对驳回的申请表进行理由阐述，表明拒绝原因。

c）输入项

无

d）输出项

|  |  |
| --- | --- |
| 名称 | 类型 |
| 审核状态 | Int |
| 驳回理由 | varchar（30） |

e）模块程序流程图



f）类设计



说明：

Pass：通过申请，返回信息给数据库服务器，教室占用；

Reject：驳回申请并说明理由；

Result：将上述两个操作的结果返回给用户；

View：管理员审核界面模块。

**4.5信息管理模块**

a）模块描述

管理员对教室、会议室和用户的信息进行管理

b）功能

①对教室、会议室的属性进行修改，包括设备信息和强制借用情况。

②对用户的注册信息进行审核，添加用户。

c）输入项

注册信息管理：

|  |  |
| --- | --- |
| 名称 | 类型 |
| 学号/工号 | varchar（10） |
| 密码 | Varchar（10） |
| 姓名 | varchar（10） |
| 学院 | varchar（20） |
| 班级 | varchar（10） |
| 电话 | varchar（20） |
| 邮箱 | varchar（20） |
| 是否为学生 | Bool |

教室、会议室信息管理

|  |  |
| --- | --- |
| 名称 | 类型 |
| 教室编号 | varchar（10） |
| 学院 | varchar（20） |
| 位置 | varchar（10） |
| 权限 | Int |
| 设备 | varchar（20） |
| 开放时间 | Date |
| 结束时间 | Date |

d）输出项

审核状态，教室、用户信息更改。

e）模块程序流程图



f）类设计



1. **类图**

类图表示了各个模块间的关系，通过主页面连接到数据库，从而进行相应的操作与数据更新。



1. **测试计划**

**6.1测试阶段**

a）单元测试阶段，此阶段主要进行各个模块的内部测试工作。从数据接口、数据结构和边界条件等方面进行测试。主要采用白盒测试的技术，可以对多个模块进行并行测试。

b）集成测试阶段，本阶段进行模块之间配合工作的测试，验证各个模块配合工作的稳定性是否符合预期。把模块按照要求组装起来进行测试，主要检查与接口有关的问题。

c)验收测试/平行运行阶段：这一阶段将本系统投入模拟使用的环境，比较实际运行的结果和预期是否一致，是否符合用户需求。

**6.2测试要点**

**6.2.1测试内容：**

a）对教室、会议室管理系统的登陆模块进行测试，是否能正确验证用户身份，进行用户注册，并进行相应的安全性检查。

b）测试查询模块，是否能对符合条件的教室进行筛选。

c）测试预约和审核模块，是否能对教室进行预约，将数据正确输出；并对数据进行审核，对相应的数据库模块进行修改。

d）测试信息管理模块，是否能对教室和用户的信息进行正确的修改，进行功能性测试。

e）对各模块进行集成测试，主要检查查询-预约-审核等几个模块间的接口，看数据是否能够正确传送。

f）将整个系统运行，测试是否能够在正确运行兵满足用户的需求。

**6.2.2精度需求**

在精度需求上，根据使用需要，在各项数据的输入，输出及传送过程中进行测试，满足各种精度的需求。

**6.2.3时间要求**

在软件方面，相应时间，更新处理时间快且迅速，完全满足用户的需求。

**6.2.4灵活性**

当用户需求，如操作方式、运行时间、结果精度、数据结果于其他软件接口发生变化时，设计的软件应做适当调整，灵活性较大。

**6.2.5故障处理**

a）内部故障处理

在开发阶段可以随即修改数据库里面的内容

b）外部故障处理

对因外部操作引起的错误，进行错误提示并修改，对编辑的程序进行重新装载。