**虚拟校园系统软件设计说明书**

撰稿人： 孙彦林 213213459@seu.edu.cn

版本号： V1.0

日期： 2023-8-22

|  |  |  |  |
| --- | --- | --- | --- |
| 修改日期 | 版本号 | 修改人 | 修改日期 |
|  |  |  |  |
|  |  |  |  |

东南大学计算机科学与工程学院

目录

[1 引言 4](#_Toc143630432)

[1.1 编写目的 4](#_Toc143630433)

[1.2 背景 4](#_Toc143630434)

[1.3 定义 4](#_Toc143630435)

[1.4 参考资料 4](#_Toc143630436)

[1.5 小组成员 4](#_Toc143630437)

[2 程序系统的分析 5](#_Toc143630438)

[2.1 可行性分析 5](#_Toc143630439)

[2.2 需求分析 6](#_Toc143630440)

[2.3 开发设计环境 7](#_Toc143630441)

[2.3.1 IntelliJ IDEA Community Edition 2023.2 7](#_Toc143630442)

[2.3.2 JDK 20.0.2 7](#_Toc143630443)

[2.3.3 Access 2021 7](#_Toc143630444)

[2.3.4 第三方库/组件 7](#_Toc143630445)

[3 程序系统的结构 8](#_Toc143630446)

[4 用户管理模块设计说明 9](#_Toc143630447)

[4.1 模块背景 9](#_Toc143630448)

[4.2 需求分析 9](#_Toc143630449)

[4.3 系统设计 9](#_Toc143630450)

[4.3.1 界面设计 9](#_Toc143630451)

[4.3.2 模块流程图 10](#_Toc143630452)

[4.3.3 类分析 10](#_Toc143630453)

[5 学生学籍管理模块设计说明 11](#_Toc143630454)

[6 选课系统模块设计说明 11](#_Toc143630455)

[7 图书馆模块设计说明 11](#_Toc143630456)

[8 商店模块设计说明 11](#_Toc143630457)

[9 医院模块设计说明 11](#_Toc143630458)

[10 公共模块设计说明 11](#_Toc143630459)

[10.1 Message:header/meta,body 12](#_Toc143630460)

[10.2 User 12](#_Toc143630461)

[10.3 数据库工具类DbHelper 12](#_Toc143630462)

[10.4 数据存取类DAO 12](#_Toc143630463)

[10.5 工具类 12](#_Toc143630464)

[10.6 其他类… 13](#_Toc143630465)

[11 网络模块设计说明 14](#_Toc143630466)

[11.1 客户端IMessageClientSrv 14](#_Toc143630467)

[11.2 服务器端IMessageServerSrv 14](#_Toc143630468)

[11.3 Socket 14](#_Toc143630469)

[11.3.1 客户端 14](#_Toc143630470)

[11.3.2 服务器端 14](#_Toc143630471)

[11.4 输入输出流 15](#_Toc143630472)

[11.4.1 读取输入 15](#_Toc143630473)

[11.4.2 发送输出 15](#_Toc143630474)

[12 多线程模块设计说明 15](#_Toc143630475)

[12.1.1 客户端 15](#_Toc143630476)

[12.1.2 服务器端 16](#_Toc143630477)

[13 数据库设计说明 16](#_Toc143630478)

[13.1 元数据 16](#_Toc143630479)

[13.2 表设计 16](#_Toc143630480)

[13.2.1 tblUser表 16](#_Toc143630481)

[13.2.2 tblClass表 17](#_Toc143630482)

[13.2.3 tblXxx表 17](#_Toc143630483)

[13.2.4 tblXxx表 17](#_Toc143630484)

[14 其他 17](#_Toc143630485)

# 引言

## 编写目的

为了保证项目开发人员按时保质地完成预定目标，更好地了解项目实际情况，按照合理的顺序开展工作，现以书面的形式将项目开发生命周期中的项目任务范围、项目团队组织结构、团队成员的工作责任、团队内外沟通协作方式、开发进度、检查项目工作等内容描述出来，作为项目相关人员之间的共识和约定以及项目生命周期内的所有项目活动的行动基础。

## 背景

虚拟校园系统是专业技能实训课程提出，孙彦林、余畅、薛沛林等七人组成的开发小组面向学校开发的大型管理系统，主要功能是实现校园的信息化管理，包括用户管理、学生学籍管理、选课系统、图书馆、商店等功能。项目周期约四周。

## 定义

*列出本文件中用到专门术语的定义和外文首字母组词的原词组。*

## 参考资料

《JAVA项目开发全程实录》第一章

《Java核心技术·卷 I（原书第10版） 基础知识》

《Java核心技术·卷 II（原书第10版） 高级特性》

《Java与模式（阎宏）》

## 小组成员

09021230 孙彦林

09021334 徐家齐

09021234 余 畅

09021232 薛沛林

09021229 石倬凡

09021331 任乐然

09021235 江紫弦

# 程序系统的分析

## 可行性分析

根据《GB8567-88计算机软件产品开发文件编制指南》中可行性分析的要求，制定可行性研究报告如下。

##### 引言

###### 编写目的

以文件的形式提供项目实施的参考依据，其中包括项目存在的风险、项目需要的投资和能够收获的最大效益。

###### 背景

专业技能实训课程是一门旨在培养学生特定专业领域实际技能的课程。为了锻炼学生实践技能、培养学生团队合作能力，要求学生团队开发虚拟校园系统。

##### 可行性研究的前提

###### 要求

虚拟校园系统必须包含用户管理[注册/注销、登录/登出、授权]、学生学籍管理、选课系统、图书馆、商店模块。同时在银行、医院、宿舍、在线课堂等选做模块中选择至少一个实现。另外，该系统还必须保证数据的安全性、完整性和准确性。

###### 目标

研发虚拟校园系统的目标是创建一个综合性的在线平台，以满足学生、教职员工和其他相关方的教育和校园需求。。

###### 条件、假定和限制

为了不影响课程的正常流程，虚拟校园系统必须在四周的时间内交付。系统分析人员需要1天内到位。去除其中可能出现的问题，例如团队成员外出留学或参加校园活动，占用3天时间确认需求分析。那么程序开发人员需要在20天的时间内进行系统设计、程序编码、系统测试、程序调试和网站部署工作。其间，还包括了员工每周的休息时间。

###### 评价尺度

根据课程要求，项目主要以校园数据管理功能为主，对于学籍、藏书、商品等记录的信息应该及时、准确地保存，并提供相应的查询和统计。由于数据太多，不易维护，传统的维护方式容易出错，系统中的数据盘点功能应当支持增删改查以及冲突检测等功能，减少不必要的问题带来的损失。

##### 效益分析

项目用时约四周。项目完成后，会给学生开发团队提供资源储备，包括技术、经验的积累，其后再开发类似的项目时，可以极大地缩短项目开发周期。

##### 结论

根据上面的分析，在技术上不会存在问题，因此项目延期的可能性很小。在效益上公司投入7个人、4周的时间，效益比较可观。在学生今后的发展上，可以储备网站开发的经验和资源。因此认为该项目可以开发。

## 需求分析

通过沟通和需求分析，要求系统具有以下功能。

1. 系统操作简单，界面友好。
2. 规范、完善的基础信息设置。
3. 当外界环境(停电、网络病毒)干扰本系统时，系统可以自动保护原始数据的安全。

用户管理

1. 登录/登出。
2. 更新信息

学生学籍管理

1. 存贮学生学籍信息，支持增删改查功能。
2. 为方便用户，支持多条件查询。
3. 登记学生成绩

选课系统

1. 存贮课程设置信息，支持增删改查功能。
2. 存贮学生选课信息。
3. 成绩查询
4. 系统管理维护

图书馆

1. 存贮藏书信息，支持增删改查功能。
2. 学生可对藏书进行检索和借阅。
3. 存贮学生借阅信息。

商店

1. 存贮商品信息，支持增删改查功能。
2. 存储进货记录、退货记录以及学生购买记录。

医院

1. 病人挂号
2. 科室及挂号记录存储

## 开发设计环境

## IntelliJ IDEA Community Edition 2023.2

## JDK 20.0.2

## Access 2021

## 第三方库/组件

# 程序系统的结构

用一系列图表列出本程序系统内的每个程序（包括每个模块和子程序）的名称、标识符和它们之间 的层次结构关系。

用户登录

用户注册

用户管理

客户端

功能选择

医院

模块

商店

模块

银行

模块

选课系统

模块

图书管

模块

学生学籍管理模块

数据处理，并发送请求给服务器

接收服务器回复信息，更新显示

回复客户端请求

接收客户端请求

服务器端

数据库操作

# 用户管理模块设计说明

各个子模块的设计说明：应包含模块功能图、对应的各个用户交互界面、所涉及的实体类与对应的数据库表、字段、业务服务**接口**等。

## 模块背景

用于管理用户信息已经用户登录、登出等。

## 需求分析

系统用户管理

更新信息

注册

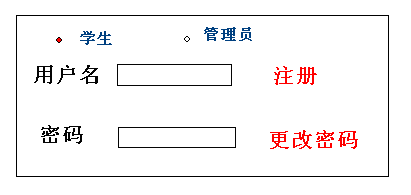
登出

登录

## 系统设计

## 界面设计

Client端：





## 模块流程图

注册

用户登录

用户登出

更新信息

## 类分析

**实体类**：登录账户(学生/教师/管理员)

User：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String |  | 登录名 |
| 2 | pwd | String | 6~16个字符 | 密码 |
| 3 | age | Integer | 非0 | 年龄 |
| 4 | role | String | 字符,[ST|TC|AD] | 角色 |

**服务类(class)/接口(interface)**：

Client端：IUserClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | User login(User) |  |
| 2 | 注册 | User register(User) |  |
| 3 | 登出 | User logout(User) |  |

Server端：IUserServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | User login(User) |  |
| 2 | 注册 | User register(User) |  |
| 3 | 登出 | User logout(User) |  |
| 4 | … | … | … |

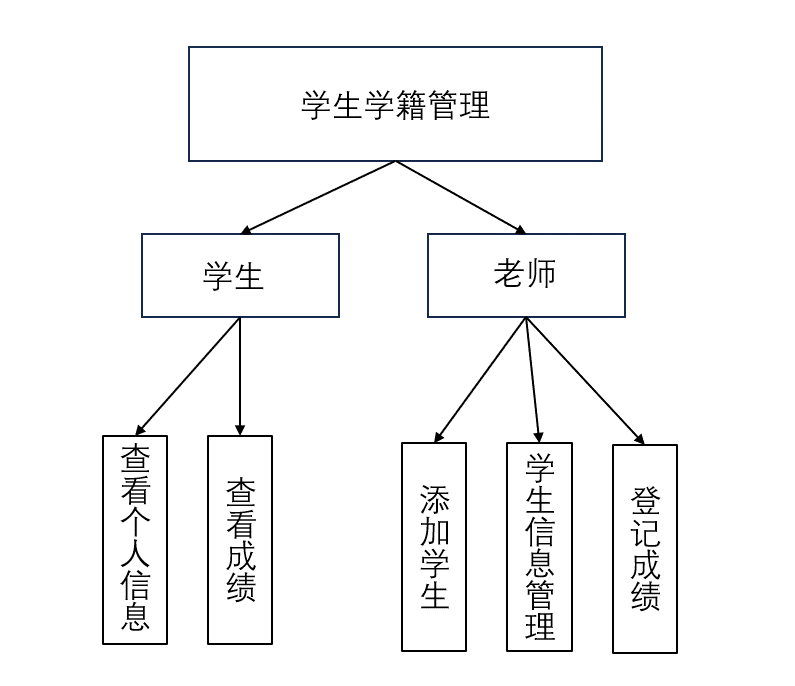
# 学生学籍管理模块设计说明

各个子模块的设计说明：应包含模块功能图、对应的各个用户交互界面、所涉及的实体类与对应的数据库表、字段、业务服务**接口**等。

## 模块背景

学生学籍管理是一个用于管理学生的基本信息、课程成绩、选课情况等的系统模块。它可以方便学校、教师和学生进行学籍管理、课程管理和成绩管理，提供实时的学生信息和成绩查询功能，帮助学校有效管理学生学习情况。

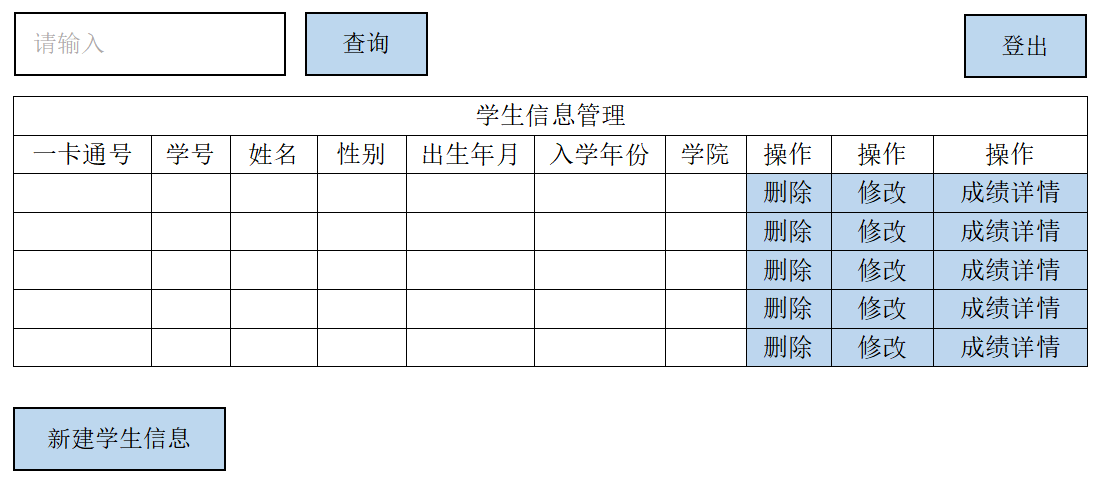
## 需求分析

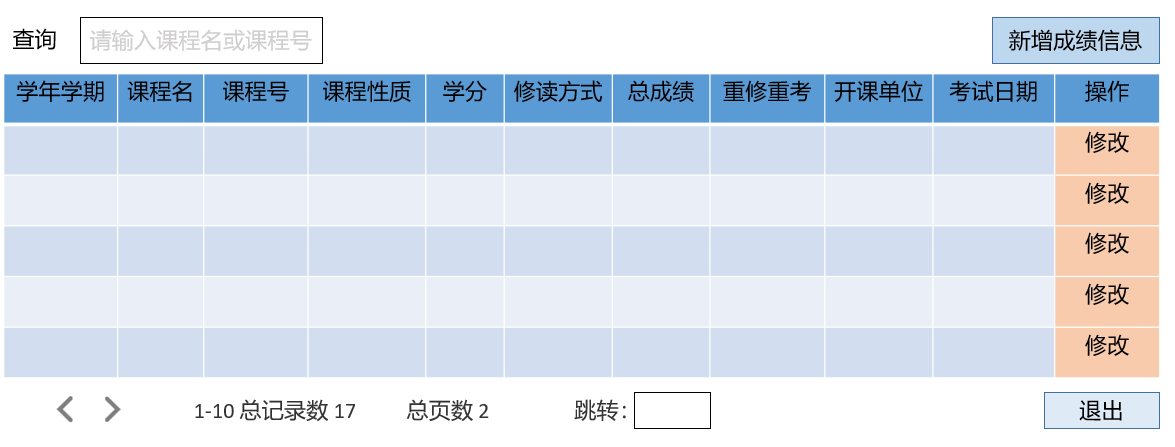


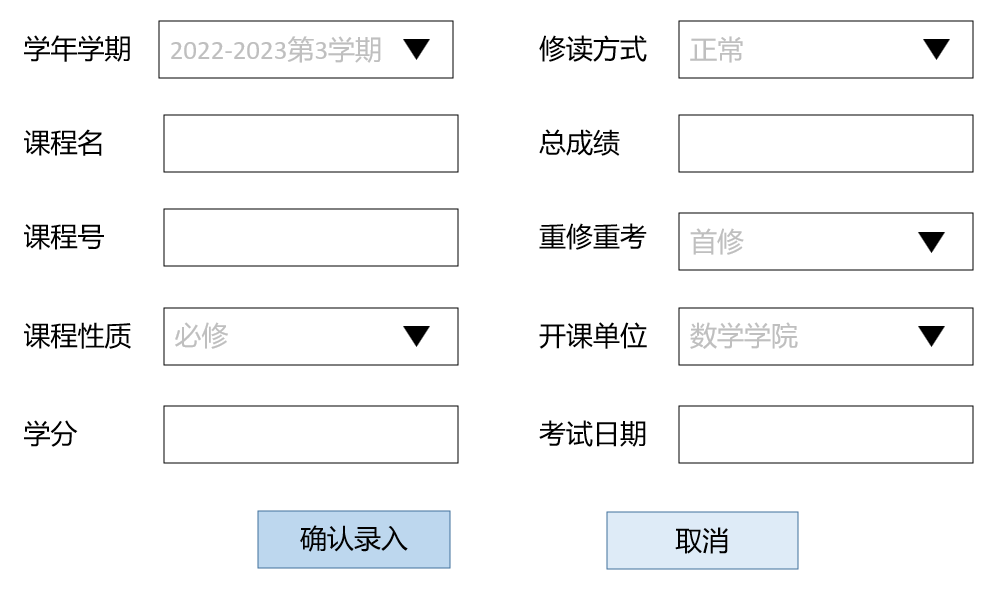
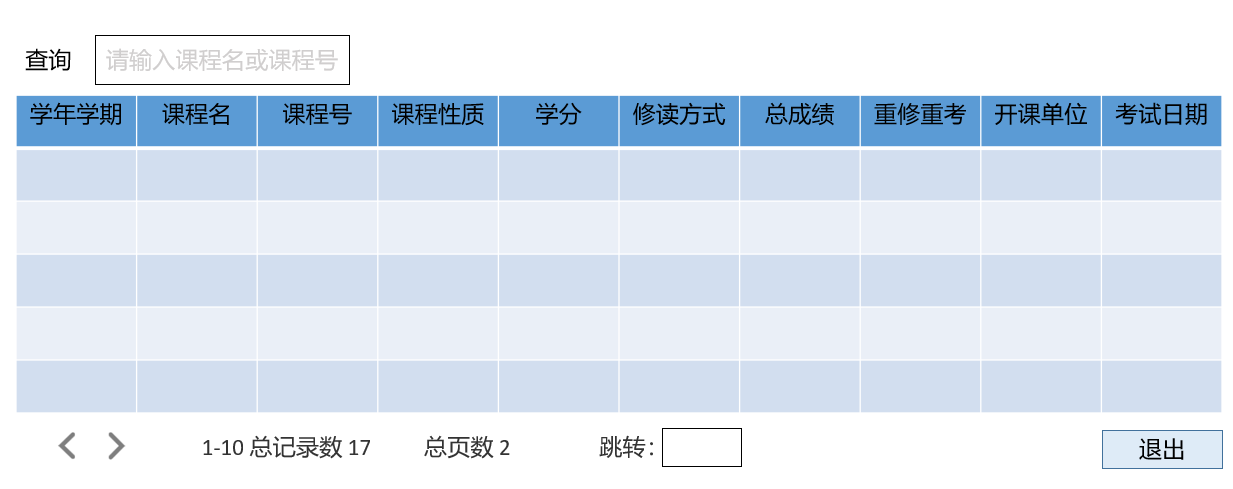
## 系统设计

## 界面设计

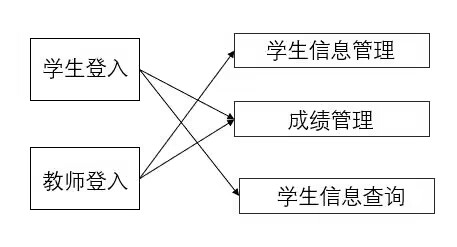
Client端：







## 模块流程图



## 类分析

**实体类**：学生信息类

StudentInfo：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | card\_id | String | 9个字符，仅数字 | 一卡通号 |
| 2 | id | String | 8个字符 | 学号 |
| 3 | name | String | 字符 | 姓名 |
| 4 | sex | bool | 男|女 | 性别，男0女1 |
| 5 | birth | LocalDate | 2000年及以后 | 出生年月 |
| 6 | grade | String | 两位字符，如21 | 入学年份 |
| 7 | college | String | 字符 | 学院 |

成绩类Grade：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | card\_id | String | 9个字符，仅数字 | 一卡通号 |
| 2 | course\_id | String | 10个字符 | 课程号 |
| 3 | grade | double | 0~100之间 | 总成绩 |

**服务类(class)/接口(interface)**：Client端：IInfoClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 学生信息查询 | StudentInfo search\_info(String) | 老师界面“查询”按键 |
| 2 | 删除 | void delete\_stu(String) | 老师界面“删除”按键，输入一卡通号 |
| 3 | 新增成绩信息 | bool add\_grade(Grade) | 老师界面“新增成绩信息”按键 |
| 4 | 修改成绩信息 | bool Modify\_grade(Grade) | 老师界面修改成绩信息 |
| 5 | 查看成绩信息 | bool StudentScore View(int) | 老师查看学生信息界面“成绩详情”按键 |
| 6 | 新建学生信息 | StudentInfo add\_info(String) | 老师界面“新建学生信息”按键 |
| 7 | 保存成绩信息 | bool save\_grade(Grade) | 老师成绩界面“保存”按键 |
| 8 | 取消 | bool cancel() | 退回上一级页面 |

Server端：IInfoServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 学生信息查询 | StudentInfo search\_info(String) | 老师界面“查询”按键 |
| 2 | 删除 | void delete\_stu(String) | 老师界面“删除”按键，输入一卡通号 |
| 3 | 新增成绩信息 | bool add\_grade(Grade) | 老师界面“新增成绩信息”按键 |
| 4 | 修改成绩信息 | bool Modify\_grade(Grade) | 老师界面修改成绩信息 |
| 5 | 查看成绩信息 | bool StudentScore View(int) | 老师查看学生信息界面“成绩详情”按键 |
| 6 | 新建学生信息 | StudentInfo add\_info(String) | 老师界面“新建学生信息”按键 |
| 7 | 保存成绩信息 | bool save\_grade(Grade) | 老师成绩界面“保存”按键 |

# 选课系统模块设计说明

## 模块背景

随着信息技术的日益发展和计算机网络的普遍应用，学校的学生人数不断增加，现有的学生信息管理的手工管理模式已日显吃力。随着社会的发展，培养计划也在不断的修改。学分制的实行给学生以更大的选择空间，但也增加了日常教务的管理工作量。因此，利用现代计算机和数据库开发技术，在网络环境下建立学生信息管理系统对于减轻学生信息管理人员的劳动强度、提高工作质量和效率、方便管理人员对信息的查询、提高信息资源的利用率和管理水平都具有重要意义。

## 需求分析

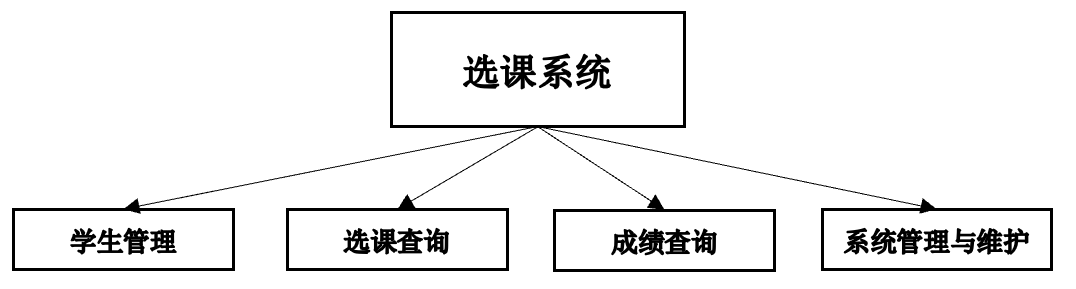
选课工作要完成学生管理、选课查询、成绩查询、系统管理与维护等功能

1. 学生管理：学生信息查询、插入、删除、修改等

2. 选课管理：选课信息查询、插入、删除、修改等

3.成绩管理：成绩的查询。

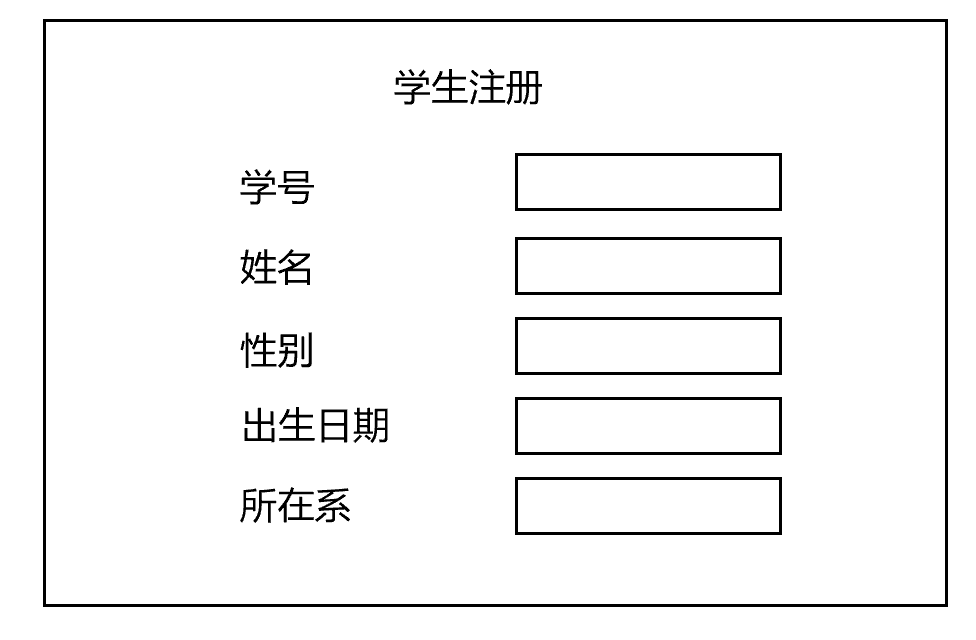
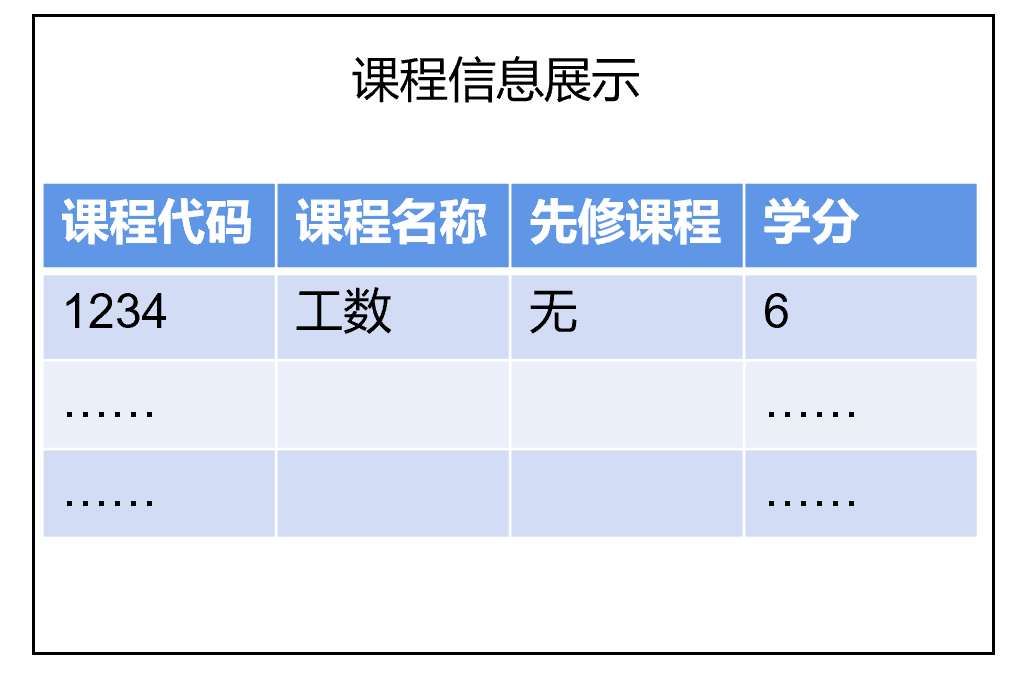
4. 系统管理与维护：操作员信息的管理，数据安全管理（含备份与恢复）、操作员管理、权限设置等

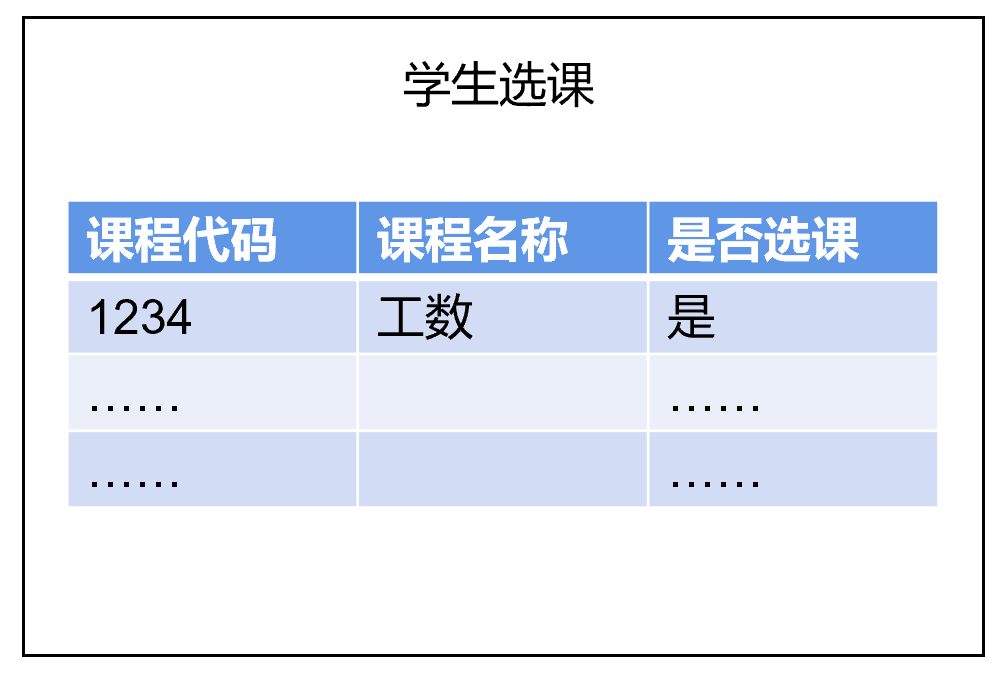
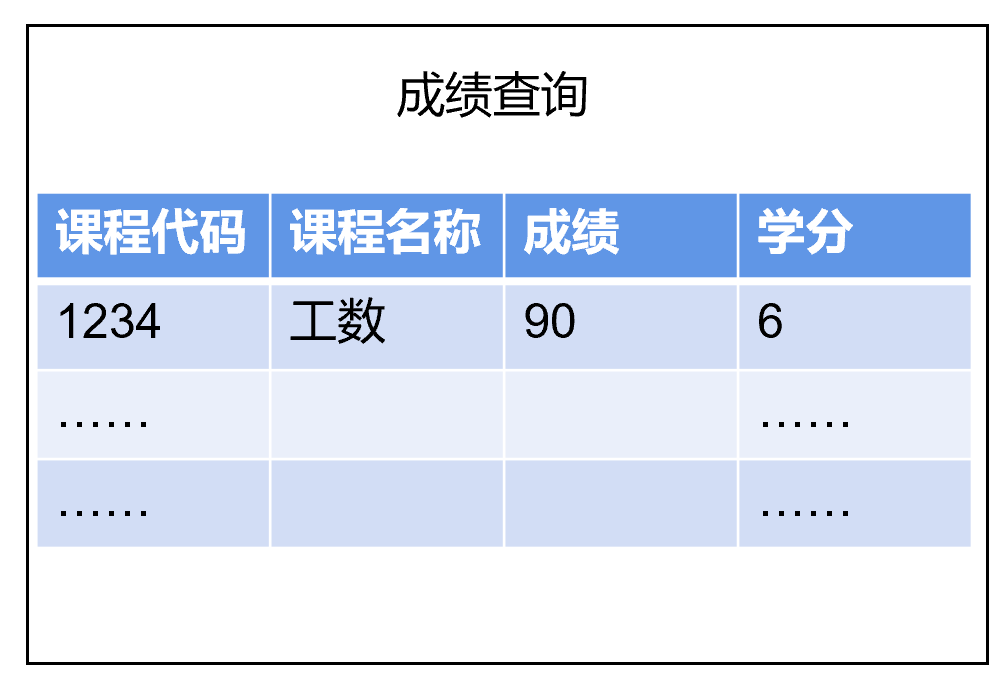
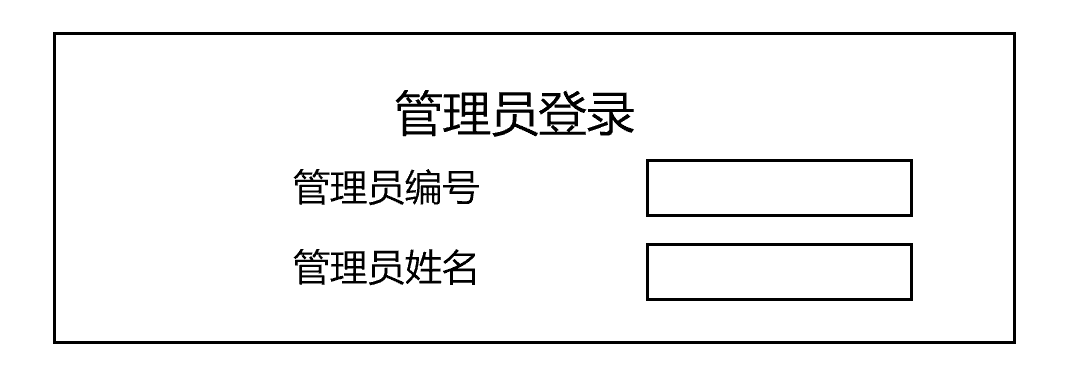


## 系统设计

## 界面设计

Client端：





## 模块流程图

管理员登录并管理

学生注册

课程信息展示

学生选课

查询成绩

## 类分析

**实体类**：学生信息表Student

Student：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Sno | Char | 8字符 | 学号 |
| 2 | Sname | Char | 20字符 | 姓名 |
| 3 | Ssex | Char | 2字符 | 性别 |
| 4 | Ssage | SmallDateTime |  | 出生日期 |
| 5 | Sdept | Char | 20字符 | 所在系 |

**实体类**：课程信息表Course

Course：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Cno | Char | 4字符 | 课程代码 |
| 2 | Cname | Char | 40字符 | 课程名称 |
| 3 | Cpno | Char | 4字符 | 先修课程 |
| 4 | Credit | Decimal | 4字符 | 学分 |

**实体类**：成绩信息表SC

SC：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Sno | Char | 5字符 | 学号 |
| 2 | Cno | Char | 3字符 | 课程代码 |
| 3 | Grade | Decimal | 3字符 | 成绩 |

**实体类**：管理员信息表Admin

Admin：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Adminid | Char | 4字符 | 操作员编号 |
| 2 | Adminname | Char | 12字符 | 操作员姓名 |
| 3 | Permission | Char | 20字符 | 权限 |

**服务类(class)/接口(interface)**：

Client端：IStudentClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | Student login(Student) |  |
| 2 | 注册 | Student register(Student) |  |
| 3 | 课程信息展示 | Student show(Student) |  |
| 4 | 选课 | Student choose(Student) |  |
| 5 | 查分 | Student inquire(Student) |  |

Server端：IUserServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | Student login(Student) |  |
| 2 | 注册 | Student register(Student) |  |
| 3 | 课程信息展示 | Student show(Student) |  |
| 4 | 选课 | Student choose(Student) |  |
| 5 | 查分 | Student inquire(Student) |  |

# 图书馆模块设计说明

## 模块背景

用于管理图书馆信息，包括藏书信息以及借阅信息。

## 需求分析

学生借阅

增加藏书

图书馆管理模块

学生续借

删除藏书

学生还书

修改藏书信息

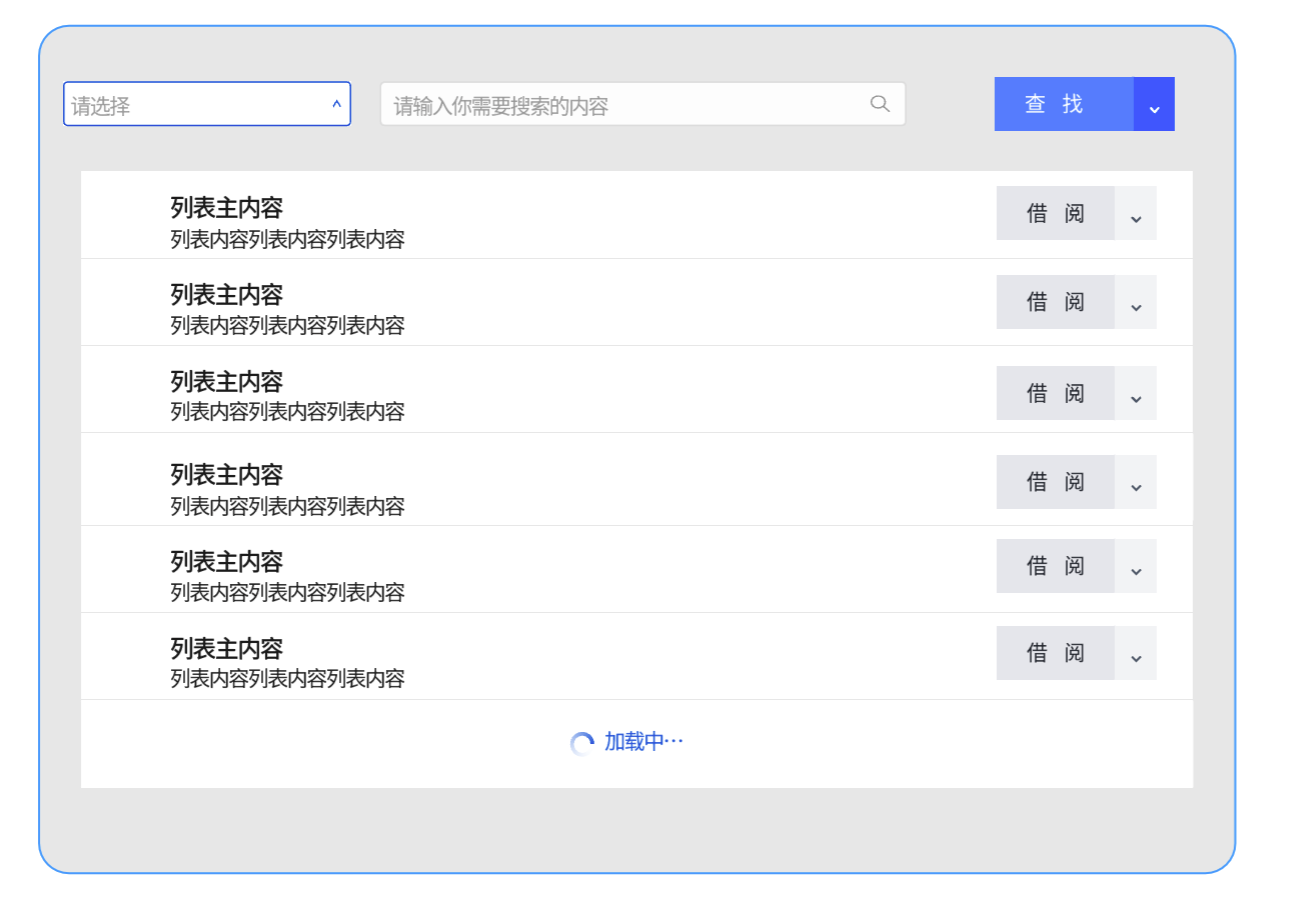
借阅信息查找

查找藏书

## 系统设计

## 界面设计

Client端：





## 模块流程图

查找藏书

用户借阅

填写信息

增加/修改藏书

删除藏书

查找信息

## 类分析

**实体类**：藏书、操作记录、用户

Book：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | bookId | String |  | 书名 |
| 2 | bookISBN | String |  | 书号 |
| 3 | author | String |  | 作者 |
| 4 | bookType | String |  | 类型 |
| 5 | bookPrice | Float |  | 价格 |
| 6 | publisher | String |  | 出版社 |
| 7 | summary | String |  | 摘要 |
| 8 | totalNum | Integer |  | 馆藏册数 |
| 9 | freeNum | Integer |  | 在馆册数 |
| 10 | bookPos | String |  | 位置 |
| 11 | borrowNum | Integer |  | 被借次数 |

BookOperationRecord：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | oprId | String | YYYYMMDD+三位顺序编码 | 操作序列号 |
| 2 | id | String |  | 学号 |
| 3 | ISBN | String |  | 书号 |
| 4 | oprTime | Integer |  | 操作时间 |
| 5 | oprType | String | 字符,[BOR|REN|RET] | 操作类型 |
| 6 | oprMark | String |  | 备注 |

BookBorrower：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String |  | 学号 |
| 2 | name | String |  | 姓名 |
| 3 | holdNum | Integer | 小于等于4 | 持有数量 |
| 4 | holdISBN | String[4] |  | 借阅数目 |
| 5 | outdateTime | Int[4] |  | 过期时间 |
| 6 | userMark | String |  | 备注 |

**服务类(class)/接口(interface)**：

Client端：IUserClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 增加藏书 | Book create(Book) |  |
| 2 | 删除藏书 | Book delete(Book) |  |
| 3 | 修改藏书 | Book update(Book) |  |
| 4 | 查找藏书 | Book retrieve(Book) |  |
| 5 | 学生借阅 | BookOperationRecord borrow(Book, BookBorrower) |  |
| 6 | 学生续借 | BookOperationRecord renew(Book, BookBorrower) |  |
| 7 | 学生还书 | BookOperationRecord return(Book, BookBorrower) |  |
| 8 | 信息查询 | List<Map<Book,Int>> find(BookBorrower) |  |

Server端：IUserServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 增加藏书 | Book create(Book) |  |
| 2 | 删除藏书 | Book delete(Book) |  |
| 3 | 修改藏书 | Book update(Book) |  |
| 4 | 查找藏书 | Book retrieve(Book) |  |
| 5 | 学生借阅 | BookOperationRecord borrow(Book, BookBorrower) |  |
| 6 | 学生续借 | BookOperationRecord renew(Book, BookBorrower) |  |
| 7 | 学生还书 | BookOperationRecord return(Book, BookBorrower) |  |
| 8 | 信息查询 | List<Map<Book,Int>> find(BookBorrower) |  |

# 商店模块设计说明

## 模块背景

用于学生查询商品信息与购买商品记录、购买商品；管理员查询商品信息与学生购买记录、进退商品。

## 需求分析

商店管理（学生）

查询商品

查询购买商品记录

购买商品

商店管理（管理员）

查询商品

查询学生购买商品记录

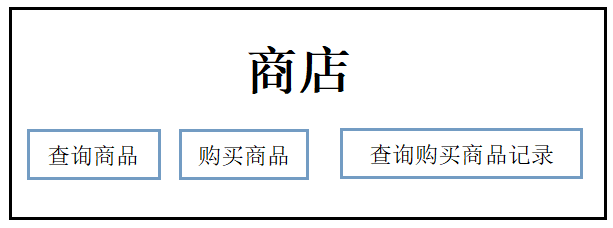
进退商品

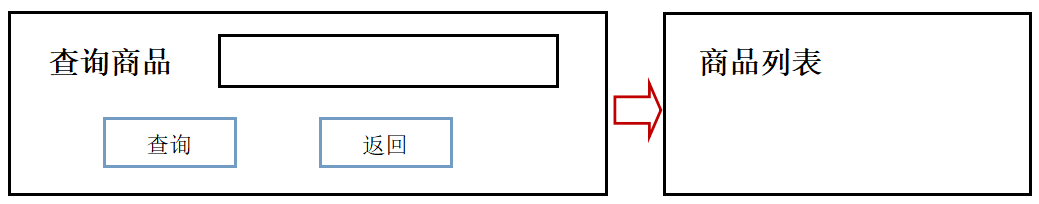
## 系统设计

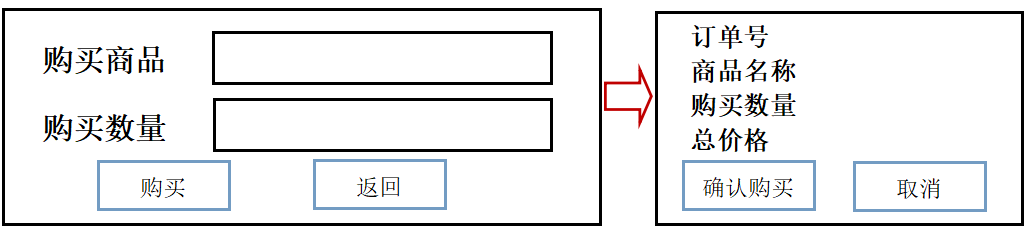
## 界面设计

Client端：

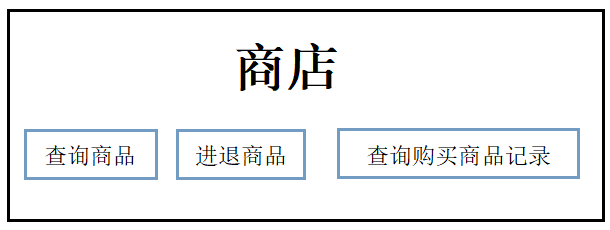
学生界面：

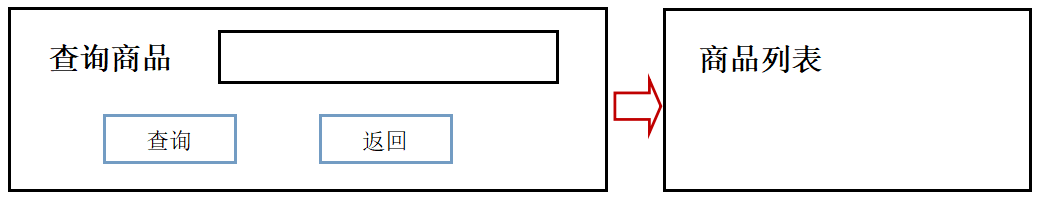


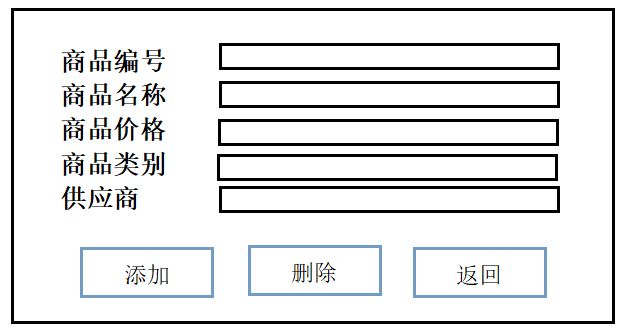




管理员界面：







## 模块流程图

学生界面：

查询商品

确认购买/取消

购买商品

购买

查询购买商品记录

管理员界面：

查询商品

进退商品

添加/删除

查询购买商品记录

## 类分析

**实体类**：

Good商品(学生/管理员)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | good\_id | String |  | 商品号 |
| 2 | good\_name | String |  | 商品名称 |
| 3 | good\_price | double | 小数点后两位 | 商品价格 |
| 4 | category | String |  | 商品类别 |
| 5 | provider | String |  | 供应商 |

Purchase（学生）:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | order\_id | String |  | 购买订单号 |
| 2 | good\_name | double | 小数点后两位 | 商品名称 |
| 3 | nums | Integer | 大于等于1 | 购买数量 |

**服务类(class)/接口(interface)**：

Client端：IShopClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查询商品 | Good search(Good) |  |
| 2 | 购买商品 | Buy(Good,User) |  |
| 3 | 查询记录 | Purchase record(User) |  |
| 4 | 添加商品 | Add good(Good) |  |
| 5 | 删除商品 | Delete good(Good) |  |

Server端：IShopServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查询商品 | Good search(Good) |  |
| 2 | 购买商品 | Buy(Good,User) |  |
| 3 | 查询记录 | Purchase record(User) |  |
| 4 | 添加商品 | Add good(Good) |  |
| 5 | 删除商品 | Delete good(Good) |  |

# 医院模块设计说明

## 模块背景

学生用户用于操作挂号预约、空闲科室查询、挂号记录查询等功能。

管理员用户用于操作添加删除科室、全体学生挂号总记录查询等功能

## 需求分析

医院模块（学生用户登录）

个人挂号记录查询

空闲科室查询

挂号预约

医院模块（管理员用户登录）

全体学生挂号总记录查询

删除科室

添加科室

## 系统设计

## 界面设计

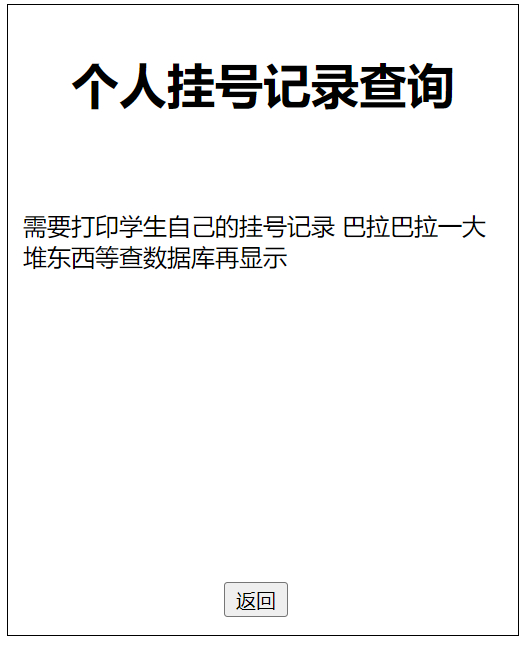
Client端：

学生界面









管理员界面：









## 模块流程图

学生用户：

挂号预约

选择需要查询的科室类型

空闲科室查询

个人挂号记录查询

更新信息

管理员用户：

填写添加科室的信息

添加科室

删除科室

选择需要删除的科室号

全体挂号记录查询

更新信息

## 类分析

**实体类**：

科室(学生/教师/管理员)

Department：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Department\_type | String | 限制在8种科室类型之中 | 科室类型 |
| 2 | Department\_ID | String | 形如“ks1001” | 科室ID（唯一编号） |
| 3 | Department\_doctor | String | 无 | 科室主任 |
| 4 | Department\_state | Bool型数组 | 8位数组，代表每天八个时间段该科室的预约挂号状态 | 该科室在当天各个时间段是否已被预约 |

挂号信息类

HospitalRegister：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | Register\_time | String | 日期格式 | 挂号日期 |
| 2 | Register\_person | String | 无 | 挂号人 |
| 3 | Register\_type | String | 限制在8种科室类型之中 | 挂号科室类型 |
| 4 | Register\_ID | String | 形如“gh1001” | 挂号单ID |

**服务类(class)/接口(interface)**：

Client端：IUserClientSrv

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 方法 | | 备注 |
| 1 | 挂号 | Department HospitalRegister(User ，string[]) | 根据表单收集到的科室类型、预约时间段等信息，在数据库中查询一个可预约的科室，并将该科室信息返回；同时更新数据库中的挂号记录信息 | |
| 2 | 科室查询 | Department[] DepartmentSearch(string[]) | 通过表单收集的科室类型信息，在数据库中查询所有有空闲时段的该类型科室，并将这些科室信息返回 | |
| 3 | 查询个人挂号记录 | HospitalRegister[] RegisterInfo(User) | 根据登录的用户，返回数据库中记录的该用户的挂号记录信息 | |
| 4 | 添加科室 | Bool AddDepartment(Department) |  | |
| 5 | 删除科室 | Bool DeleteDepartment(Department) | 通过表单收集的科室号，在数据库找到该科室，并将其删除 | |

Server端：IUserServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 挂号 | Department HospitalRegister(User ，string[]) |  |
| 2 | 科室查询 | Department[] DepartmentSearch(string[]) |  |
| 3 | 查询个人挂号记录 | HospitalRegister[] RegisterInfo(User) |  |
| 4 | 添加科室 | Bool AddDepartment(Department) |  |
| 5 | 删除科室 | Bool DeleteDepartment(Department) |  |

# 公共模块设计说明

列出同时用于服务器端和客户端的接口、类或其他资源。

用于网络传输的类必须实现序列化接口：**java.io.Serializable。**

## Message:header/meta,body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | uid/token | Long | 唯一性 | 标识符 |
| 2 | name | String | 6~16个字符 | 名称 |
| 3 | type | Enumeration | 命令|数据 | 类型 |
| 4 | statusCode | String |  | 状态码 |
| 5 | data | Object | 可序列化 | 传输数据 |
| 6 | sender | Object/String |  | 发送者/用户名 |
| … | … | … | … | … |
|  |  |  |  |  |

## User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String |  | 登录名 |
| 2 | pwd | String | 6~16个字符 | 密码 |
| 3 | age | Integer | 非0 | 年龄 |
| 4 | … | … | … | … |

## 数据库工具类DbHelper

DB config,

Jdbc driver,

## 数据存取类DAO

ORM: Hibernate, Mybatis, JPA, ...

JAVA-DB

## 工具类

## 其他类…

# 网络模块设计说明

## 客户端IMessageClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 发送 | Boolean send(Message) |  |
| 2 | 接收 | Boolean recieve(Message) |  |
| 3 | 校验 | Boolean validate(Message) |  |
| 4 | … | … | … |

## 服务器端IMessageServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 发送 | Boolean send(Message) |  |
| 2 | 接收 | Boolean recieve(Message) |  |
| 3 | 校验 | Boolean validate(Message) |  |
| 4 | … | … | … |

## Socket

## 客户端

// Code snippet::连接服务器Socket

clientSocket **= new** Socket(*SERVER\_ADDRESS*, *SERVER\_PORT);*

## 服务器端

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 说明 | 备注 |
| 1 | IServer | 服务器管理接口 | 提供启动、关闭和重启服务 |
| 2 | start() |  |  |
| 3 | stop() |  |  |
| 4 | restart() |  |  |
| 5 | isClosed() |  |  |
| 6 | … | … | … |

// Code snippet::创建服务器Socket

serverSocket = **new** ServerSocket(IConstant.*SERVER\_PORT*);

## 输入输出流

## 读取输入

// Code snippet

ObjectInputStream ois =

**new** ObjectInputStream (cts.getInputStream());

Message msg = (Message) ois.readObject();

## 发送输出

// Code snippet

ObjectOutputStream oos =

**new** ObjectOutputStream(cts.getOutputStream());

Message msg = (Message) ois.writeObject();

oos.flush();

# 多线程模块设计说明

## 客户端

主程序调用：

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

SwingUtilities.*invokeLater*(**new** Runnable() {

@Override

**public** **void** run() {

**new** MainFrame(*title*).setVisible(**true**);

}

});

}

## 服务器端

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 说明 | 备注 |
| 1 | ClientThread | 客户端线程管理 | 为请求的每个客户端建立一个线程 |
| 2 | ClientThreadMan | 线程管理类 |  |
| 3 | … | … | … |

# 数据库设计说明

## 元数据

以表格说明或E-R图直接体现。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 说明 | 备注 |
| 1 | vCampus.accdb | 数据库 | ACCESS 2021 |
| 2 | tblUser | 用户表 |  |
| 3 | tblBook | 藏书表 |  |
| 4 | tblBookOperationRecord | 操作记录表 |  |
| 5 | tblBookBorrower | 借阅人信息表 |  |

## 表设计

## tblUser表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 是否主键 | 默认值 | 约束 | 备注 |
| 1 | uId | nvarchar（8） | PRI | NULL | 定长—09010210 | 登录ID |
| 2 | uName | varchar（10） |  | NULL |  | 姓名 |
| 3 | uAge | int（4） |  | NULL | 0-100 | 年龄 |
| 4 | uSex | ENUM |  | NULL | 男|女 | 性别 |
| 5 | uPwd | nvarchar（8） |  | NULL | 定长8位 | 密码 |
| 6 | uRole | ENUM |  | 学生 | 学生|教师|管理员 | 用户的角色 |
| 7 | … | … | … | … | … | … |

## tblXxx表

## tblBook表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 是否主键 | 默认值 | 约束 | 备注 |
| 1 | bookISBN | varchar(16) | PRI | NULL |  | 书号 |
| 2 | bookname | varchar(64) |  | NULL |  | 书名 |
| 3 | author | varchar(25) |  | NULL |  | 作者 |
| 4 | bookType | text |  | NULL |  | 类型 |
| 5 | bookPrice | float |  | NULL |  | 价格 |
| 6 | publisher | varchar(64) |  | NULL |  | 出版社 |
| 7 | summary | mediumtext |  | NULL |  | 摘要 |
| 8 | totalNum | int |  | NULL |  | 馆藏册数 |
| 9 | freeNum | int |  | NULL |  | 在馆册数 |
| 10 | bookPos | text |  | NULL |  | 位置 |
| 11 | borrowNum | int |  | NULL |  | 被借次数 |

## tblBookOperationRecord表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 是否主键 | 默认值 | 约束 | 备注 |
| 1 | oprId | char(11) | PRI | NULL | YYYYMMDD+三位顺序编码 | 操作序列号 |
| 2 | id | nvarchar(8) |  | NULL | 定长—09010210 | 学号 |
| 3 | bookISBN | varchar(16) |  | NULL |  | 书号 |
| 4 | oprTime | date |  | NULL |  | 操作时间 |
| 5 | oprType | enum |  | NULL | [BOR|REN|RET] | 操作类型 |
| 6 | oprMark | text |  | NULL |  | 备注 |

## tblBookBorrower表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 是否主键 | 默认值 | 约束 | 备注 |
| 1 | id | nvarchar(8) | PRI | NULL | 定长—09010210 | 学号 |
| 2 | holdNum | int |  | NULL | 0-4 | 持有数量 |
| 3 | holdISBN | text |  | NULL |  | 借阅数目  数组用@分隔 |
| 4 | outdateTime | text |  | NULL |  | 过期时间  数组用@分隔 |
| 5 | userMark | text |  | NULL |  | 备注 |

# 其他