**虚拟校园系统软件设计说明书**

撰稿人：曹邹颖

版本号： V1.0

日 期： 2021-7-8

|  |  |  |  |
| --- | --- | --- | --- |
| 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 5](#_Toc15430)

[1.1 编写目的 5](#_Toc31709)

[1.2 背景 5](#_Toc2022)

[1.3 定义 5](#_Toc13199)

[1.4 参考资料 5](#_Toc28854)

[2 程序系统的分析 6](#_Toc10893)

[2.1 可行性分析 6](#_Toc21518)

[2.2 需求分析 7](#_Toc4522)

[2.3 开发设计环境： 8](#_Toc11502)

[2.3.1 Eclipse 8](#_Toc6030)

[2.3.2 JDK 16 8](#_Toc14424)

[2.3.3 MySQL 8](#_Toc12843)

[2.3.4 第三方库 8](#_Toc30092)

[3 程序系统的结构 9](#_Toc2190)

[4 用户管理模块设计说明 10](#_Toc10498)

[4.1 模块背景 10](#_Toc10347)

[4.2 需求分析 10](#_Toc84)

[4.3 系统设计 10](#_Toc19671)

[4.3.1 界面设计 10](#_Toc9860)

[4.3.2 模块流程图 11](#_Toc26638)

[4.3.3 类分析 12](#_Toc21296)

[5 虚拟学生管理模块设计说明 13](#_Toc28730)

[5.1 模块背景 13](#_Toc29883)

[5.2 需求分析 13](#_Toc12176)

[5.3 系统设计 13](#_Toc21621)

[5.3.1 界面设计 13](#_Toc6725)

[5.3.2 模块流程图 14](#_Toc19782)

[5.3.3 类分析 15](#_Toc289)

[6 虚拟教务模块设计说明 16](#_Toc12392)

[6.1 模块背景 16](#_Toc28093)

[6.2 需求分析 17](#_Toc4401)

[6.3 系统设计 17](#_Toc7603)

[6.3.1 界面设计 17](#_Toc14337)

[6.3.2 模块流程图 18](#_Toc31816)

[6.3.3 类分析 18](#_Toc15824)

[7 虚拟图书馆模块设计说明 20](#_Toc13905)

[7.1 模块背景 20](#_Toc7407)

[7.2 需求分析 20](#_Toc8809)

[7.3 系统设计 20](#_Toc10478)

[7.3.1 界面设计 20](#_Toc18354)

[7.3.2 模块流程图 21](#_Toc11995)

[7.3.3 类分析 22](#_Toc24497)

[8 虚拟商店模块设计说明 23](#_Toc32564)

[8.1 模块背景 23](#_Toc1939)

[8.2 需求分析 24](#_Toc28464)

[8.3 系统设计 24](#_Toc1693)

[8.3.1 界面设计 24](#_Toc9275)

[8.3.2 模块流程图 25](#_Toc29605)

[8.3.3 类分析 25](#_Toc20685)

[9 虚拟银行模块设计说明 27](#_Toc18293)

[9.1 模块背景 27](#_Toc29574)

[9.2 需求分析 27](#_Toc6248)

[9.3 系统设计 28](#_Toc21042)

[9.3.1 界面设计 28](#_Toc114)

[9.3.2 模块流程图 28](#_Toc9224)

[9.3.3 类分析 29](#_Toc10937)

[10 公共模块设计说明 30](#_Toc2614)

[10.1 Message 30](#_Toc30716)

[10.2 User 31](#_Toc9988)

[10.3 数据库工具类DbHelper 31](#_Toc3022)

[10.4 数据存取类DAO 33](#_Toc14075)

[10.5 工具类 33](#_Toc23367)

[10.6 其他类… 33](#_Toc9626)

[11 网络模块设计说明 34](#_Toc26343)

[11.1 客户端IMessageClientSrv 34](#_Toc12382)

[11.2 服务器端IMessageServerSrv 34](#_Toc8525)

[11.3 Socket 35](#_Toc17972)

[11.3.1 客户端 35](#_Toc4438)

[11.3.2 服务器端 35](#_Toc24479)

[11.4 输入输出流 35](#_Toc3622)

[11.4.1 读取输入 35](#_Toc32029)

[11.4.2 发送输出 35](#_Toc11400)

[12 多线程模块设计说明 36](#_Toc9603)

[12.1.1 客户端 36](#_Toc28750)

[12.1.2 服务器端 36](#_Toc19563)

[13 数据库设计说明 37](#_Toc10637)

[13.1 元数据 37](#_Toc2284)

[13.2 表设计 38](#_Toc25839)

[13.2.1 tblUser表 38](#_Toc1841)

[13.2.2 tblChooseCourse表 38](#_Toc20344)

[13.2.3 tblCourse表 39](#_Toc13228)

[13.2.4 tblBank表 39](#_Toc12643)

[14 其他 39](#_Toc21750)

# 引言

## 编写目的

软件设计的**主要任务**是把在软件需求的基础上实现软件的具体设计方案，即设计系统程序的体系结构，确定程序由哪些模块组成以及设计模块结构的过程。软件设计说明书的**主要内容**是关于我们团队所要设计的“虚拟校园系统”来编写的，随着学校的规模不断扩大，专业、班级、学生、教师等的数量急剧增加，有关校园生活的各种公共设施也成倍增长，因此，迫切需要开发基于C/S的虚拟校园管理系统来提高校园工作与生活的效率。**设计过程**首先根据软件需求分析得到程序系统的结构，然后规划每一个模块所要实现的功能与性能等，接着进行必要的网络模块、多线程模块与数据库设计，确定测试要求并且设定测试计划。这份详细设计说明书对以后阶段的工作起指导作用，也是项目完成后系统验收的依据，**主要面向**学校教师、中间用户（系统分析人员、开发人员、测试人员、实施人员)、最终用户。

## 背景

开发软件系统的名称：基于C/S的虚拟校园系统-Vcampus

项目的任务提出者：应东南大学专业技能实训课程要求

项目的开发者：曹邹颖（组长）、许志豪、程瀚、苏哲琪、宗琦薇、汪晟宇

项目的用户：东南大学老师、学生、管理员等相关人员

运行该程序系统的计算中心：东南大学计算中心

## 定义

**实体—联系图（E-R图）**：包含实体（即数据对象）、关系和属性。作为用户与分析员之间有效交流的工具。

**数据库的逻辑结构设计**：就是把概念结构设计阶段设计好的基本 E-R 图转换为与选用的 DBMS所支持的数据模型相符合的逻辑结构。

**数据库的逻辑结构**是独立于任何一种数据模型的。在实际应用中，一般所用的数据库环境已经给定，如：MySql、SQL Server或Oracel。由于目前使用的数据库基本上都是关系数据库，因此首先需要将E-R图转换为关系模型，然后根据具体DBMS的特点和限制转换为特定的DBMS支持下的数据模型，最后进行优化。

**流程图**：由一些特定意义的图形、流程线及简要的文字说明构成，能清晰明确地表示程序的运行过程。

## 参考资料

1. Java从入门到精通， 第4版，清华大学出版社，2017年5月
2. （美）James 等著，Java编程规范，第3版 ，中国电力出版社， 2006年7月
3. （美）C.Thomas Wu 著，马素霞 等译，面向对象程序设计教程（Java版），第4版，机械工业出版社，2007年7月

# 程序系统的分析

## 可行性分析

* + 1. **技术可行性**

虚拟校园系统软件使用基于C/S架构的Windows应用程序，服务器端与客户端均采用Java作为开发语言，使用MySQL管理数据库数据，使数据的准确性与安全性得到了很大的提高，且在用户的并行操作与用户管理方面也有了极大地改善。服务器是后台支持程序，按需进行GUI界面设计；客户端程序面向实际用户，设计友好的界面；实现方面，主要采用Socket网络编程、多线程、输入输出流等技术来实现Swing界面编程。系统的性质为管理软件，因而数据库的设计与操纵是系统设计的核心。这里从两方面分析技术上的可行性。

首先，对于java编程部分，基于团队成员对C++编程语言的一定掌握，学习与使用java编程语言不是难事，基于团队成员QT、MFC等图形用户界面应用程序开发框架的一定掌握，因此在程序系统设计中，对java图形GUI编程方面的问题都能迎刃而解。

其次，对于数据库的有关知识，团队成员运用一定的时间对相关书籍、资料进行阅读以及上机学习实验，以基本掌握运用MySQL数据库技术，具备一定的系统分析与设计能力，熟悉数据库的设计与操纵；因而该系统的实现在技术上是可行的。

* + 1. **开发人员可行性**

基于团队成员对该系统软件设计有足够的技术上准备，在进行该系统软件设计中，通过进行明确的人员分工，并且已经较好的明确了该软件的设计工作工程，按部就班，以组长领导力协调组员一齐合作完成系统软件的设计。所以，在开发人员可行性方面是可行的。

* + 1. **经济可行性**

经济的可行性考虑的是系统软件的经济成本，首先，本系统使用Java、Eclipse、MySQL工具，开发工具都能轻松得到，因此开发工具的成本是可接受的。另外，对于完成开发和实施本系统的必要硬件的经济成本是不存在的，因此可以说，本系统在经济方面是可行的。

* + 1. **法律可行性**

首先，开发本系统查阅的技术与其他资料合法，技术查询是从正规渠道得到的，通过查阅图书馆的图书和浏览网络上的正规文章得到的资料信息都是真实可靠的。

另外，开发本系统过程中不存在知识产权问题，因为所查阅的知识都来自于正规地方，且传播时间已长，并且知识都是公布于大众的。不会涉及任何知识产权问题。

最后，本系统开发过程中未涉及任何法律责任，因为，未做任何破坏法律条令和违反任何明文规定的规则与法度，因此系统不存在法律的问题。

* + 1. **用户使用可行性**

本软件提高和改进管理信息服务质量、增强资源共享、减少人力和设备费用、加快信息的查询速度和准确性、管理系统建成后可以和校园网相连，提供网上查询服务。符合管理员、老师及学生的习惯，方便操作，具有可行性。

## 需求分析

* + 1. **功能需求分析**

本系统根据虚拟校园管理综合分析，便出于方便管理考虑，将虚拟校园管理系统的功能总结起来，共需要实现以下几个方面功能：

**①统一登录门户**

统一登录门户功能包括：用户注册、用户登录、用户登出、用户信息更改、用户注销五个部分。

**②虚拟学生管理**

管理所有学生的基本信息，对学生信息的操作包括添加、修改、删除等；可以根据各种条件查询出需要的信息，比如修改，可以通过学生学号查询出学生的基本信息，然后通过对需要修改项进行修改并保存修改后的结果存入数据库的学生表中。

**③虚拟教务**

**排课管理模块：**管理所有课程的基本信息，包括对课程信息的添加、修改、删除等操作；可以根据各种条件查询出需要的信息，并对相应的信息进行操作，比如修改和删除，可以通过学生的学号查询出学生的相应课程信息，对相应的课程进行修改、删除操作；

**选课管理模块：**学生可对本人的课表信息进行查询，也可以根据培养计划进行选课；

**考务管理模块：**考试安排，管理员可以安排考场和考试时间；成绩录入，任课教师可以在规定的时间段内录入学生成绩；查询成绩，学生可以根据学号查询出课程成绩。

**④虚拟图书馆**

**图书管理模块：管理员**完成图书基本信息的管理（包括：添加、修改、查询、删除）；

**借阅管理模块：**学生、老师可线上进行图书的操作（包括：图书借阅、归还、查询、续借等）。

**⑤虚拟商店**

**购物模块：**创建订单，填写订单号(ID)、商品信息、订单状态、折扣等基本信息后创建订单；向订单中添加商品，进入已创建的订单中，向已创建订单中添加新的商品；删除订单，删除一个已创建的订单。

**账户模块：**在购物订单交易时，首先要确定用户余额足够支持交易，在交易成功后，用户账户余额会扣除相应的钱额；可通过虚拟校园的另一功能模块——虚拟银行，对账户余额进行一定相关操作。

**⑥虚拟银行**

老师、学生可管理自身的基本账户信息，对个人账户余额的操作包括存款、转账、查询、取款等，可以根据各种条件查询出需要的账户余额信息，甚至对于账户进行挂失操作。

管理员可管理所有师生的基本账户信息，对学生账户余额的操作包括查询、扣钱等，对于余额不足扣钱不成功的情况再将对应的学生名单存入数据库的一张表中，甚至对于账户进行封户操作。

通过以上功能的设计与实现，并实现虚拟校园管理系统的基本功能。当然在实际的虚拟校园管理系统中，其功能要比本系统软件设计的多的多，也复杂的多，本软件设计说明书仅以比较简单易懂的方式呈现—一个虚拟校园管理系统的设计与实现的流程。

## 开发设计环境：

## Eclipse

Eclipse 是一个开放[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "https://baike.baidu.com/item/eclipse/_blank)的、基于[Java](https://baike.baidu.com/item/Java/85979" \t "https://baike.baidu.com/item/eclipse/_blank)的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java[开发工具](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7" \t "https://baike.baidu.com/item/eclipse/_blank)（Java Development Kit，[JDK](https://baike.baidu.com/item/JDK/1011" \t "https://baike.baidu.com/item/eclipse/_blank)）。

## JDK 16

JDK（Java Development Kit）是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的java应用程序。JDK是整个java开发的核心，它包含了JAVA的运行环境（JVM+Java系统类库）和JAVA工具。

SE（JavaSE），standard edition，标准版，是我们通常用的一个版本，从JDK 5.0开始，改名为Java SE。

## MySQL

MySQL是一个[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "https://baike.baidu.com/item/mySQL/_blank)，由瑞典MySQL AB 公司开发，属于 [Oracle](https://baike.baidu.com/item/Oracle" \t "https://baike.baidu.com/item/mySQL/_blank)旗下产品。MySQL 是最流行的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "https://baike.baidu.com/item/mySQL/_blank)之一，在 WEB 应用方面，MySQL是最好的 [RDBMS](https://baike.baidu.com/item/RDBMS/1048260" \t "https://baike.baidu.com/item/mySQL/_blank)（Relational Database Management System，关系数据库管理系统）应用软件之一。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "https://baike.baidu.com/item/mySQL/_blank)的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81/7176422" \t "https://baike.baidu.com/item/mySQL/_blank)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

## 第三方库

# 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**。

用户管理

用户登出、更新信息、注册、注销

用户登录

客户端

功能选择

虚拟银行

模块

虚拟商店

模块

虚拟图书馆

模块

虚拟教务

模块

虚拟学生

管理模块

接收服务器回复信息，更新显示

数据处理，并发送请求给服务器

Socket

回复客户端请求

接收客户端请求

服务器端

数据库操作

图1.程序系统的层次结构关系图

# 用户管理模块设计说明

## 模块背景

在进入虚拟校园系统前，呈现统一用户登录门户模块即用户管理模块，对于三种类型的用户（学生、老师、管理员）根据输入的用户名查询数据库中是否有该用户，若存在该用户，对应类型的登录按钮亮起，可供用户登录、登出、信息更新、用户注册/注销，若不存在该用户，即存在两种情况：1.非本虚拟校园的人员，则无法登录系统，起到信息保护的作用；2.本虚拟校园的新人员（系统信息未录入），从而可通过管理员用户注册操作后登录系统，以进行虚拟校园系统的使用。

## 需求分析

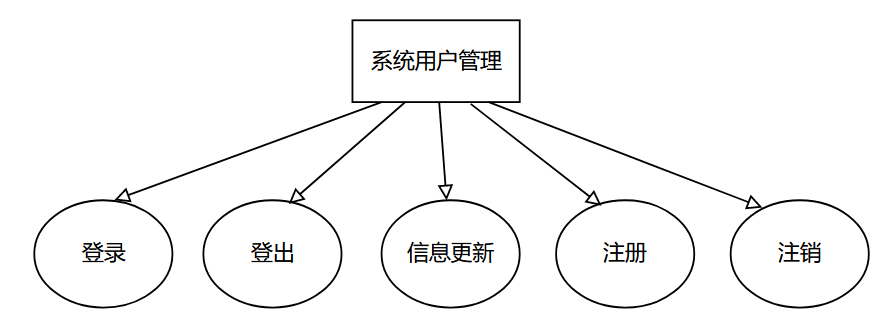


图2.用户管理模块E-R图

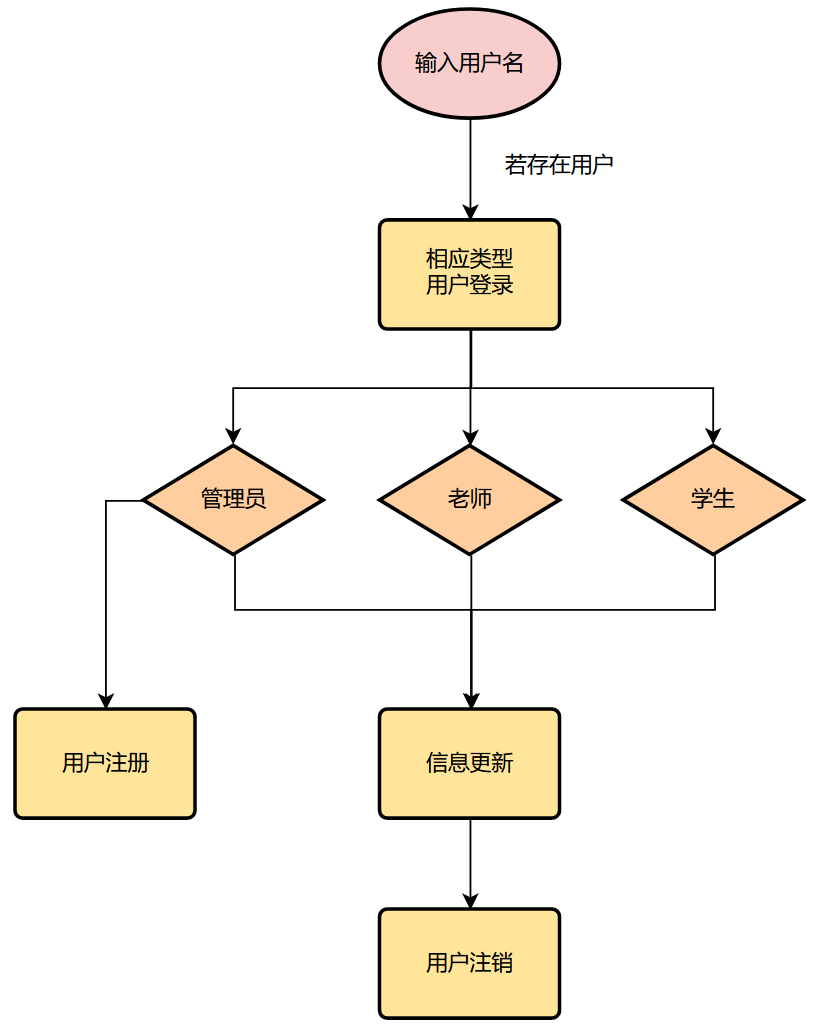
## 系统设计

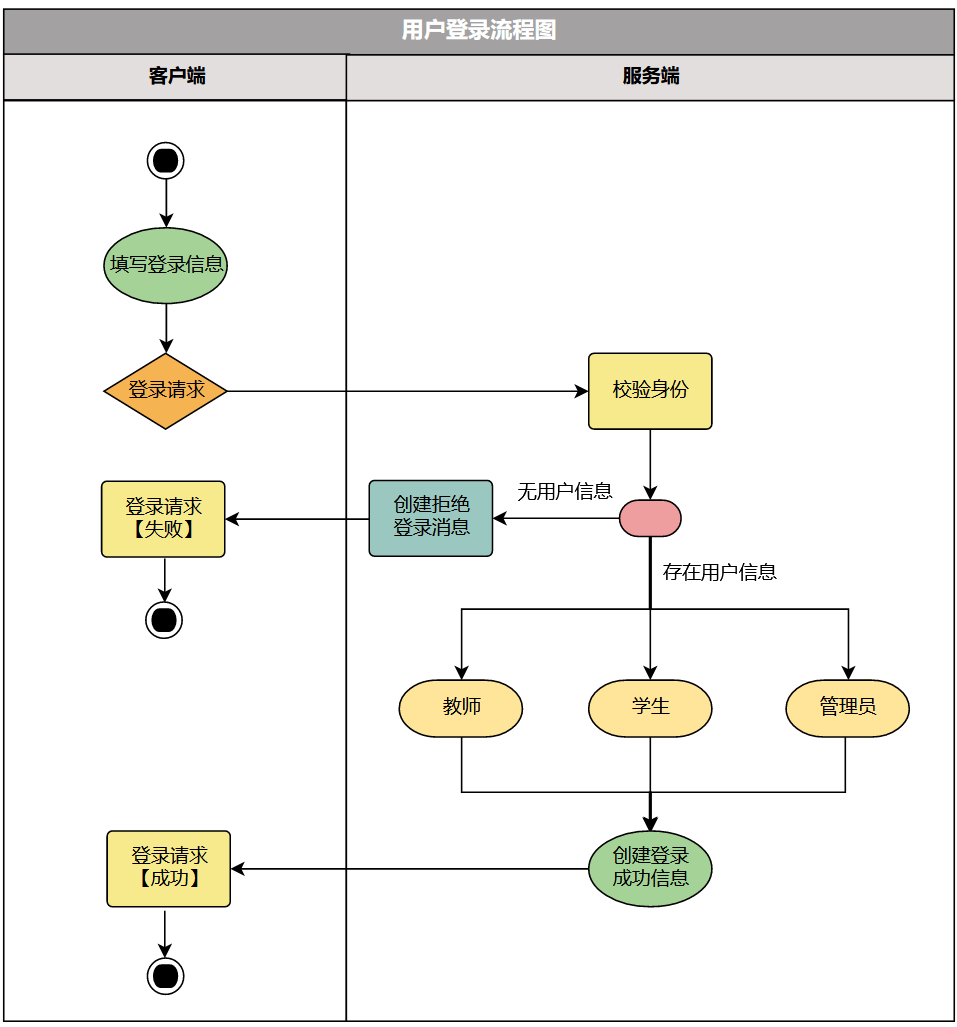
## 界面设计

Client端：



## 模块流程图





## 类分析

**实体类**：

登录账户(学生/教师/管理员)时，User：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String | 9个字符 | 登录名即一卡通号，例：  213190998 学生账号  103009076 教师账号 |
| 2 | password | String | 6~16个字符 | 密码，例：czy010612 |

个人信息更新时，UserInformation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String | 9个字符 | 登录名即一卡通号，例：  213190998 学生账号  103009076 教师账号 |
| 2 | password | String | 6~16个字符 | 当前密码，例：czy010612 |
| 3 | nickname | String | 6~16个字符 | 用户昵称，例：heyefight |
| 4 | birthday | String | 10个字符 | 生日，例：2001-06-12 |
| 5 | email | String | 8~20个字符 | 邮箱，例：1799771645@qq.com |
| 6 | mobilePhone | String | 11个字符 | 手机号，例：18362155818 |

**服务类**：

Client端：UserManagementClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | userLogin(User) |  |
| 2 | 登出 | userLogout(User) |  |
| 3 | 信息更新 | userUpdate(UserInformation) |  |
| 4 | 注册 | userRegister(User:Admin) | 管理员功能 |
| 5 | 注销 | userRegister(User:Admin) | 管理员功能 |

Server端：UserManagementServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 登录 | userLogin(User) |  |
| 2 | 登出 | userLogout(User) |  |
| 3 | 信息更新 | userUpdate(UserInformation) |  |
| 4 | 注册 | userRegister(User:Admin) | 管理员功能 |
| 5 | 注销 | userRegister(User:Admin) | 管理员功能 |

# 虚拟学生管理模块设计说明

## 模块背景

学生信息管理模块是一个教育单位不可缺少的部分。一个功能齐全、简单易用的信息管理系统不但能有效地减轻学校相关工作人员的工作负担,它的内容对于学校的决策者和管理者来说都至关重要。所以学生信息管理系统应该能够为用户提供充足的信息和快捷的查询手段。

随着科学技术的不断提高,计算机科学日渐成熟，使用计算机对学校的各类信息进行管理，具有检索迅速、查询方便、效率高、可靠性好、存储量大、保密性好、寿命长、成本低等优点，能够极大地提高学校信息管理的效率，也是一个单位科学化、正规化管理，与世界接轨的重要条件。本模块是将现代化的计算机技术和传统的校园学生管理工作相结合，按照学校的工作流程设计完成的，通过一个简化的学生信息管理模块，使学生信息管理工作系统化、规范化、自动化，从而达到提高学生信息管理效率的目的。学生信息管理系统，主要的功能有增，删，查，改，管理的学生信息有学号、姓名、性别、出生日期、政治面貌、家庭住址、电话、宿舍号等，并且使用数据库来储存学生信息。

## 需求分析

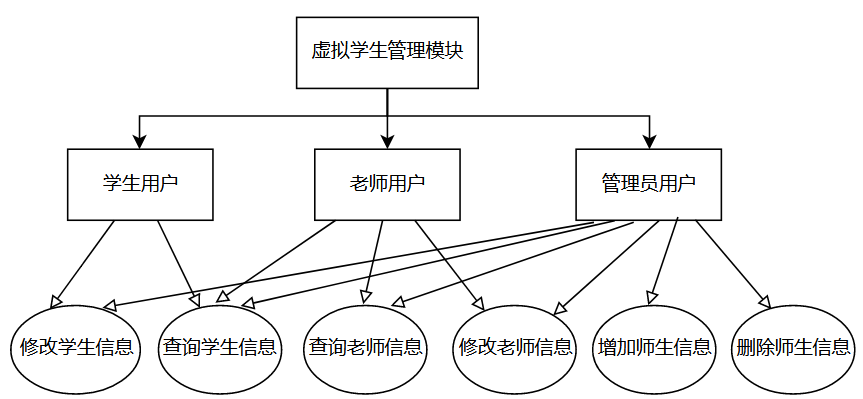


图3.虚拟学生管理模块E-R图

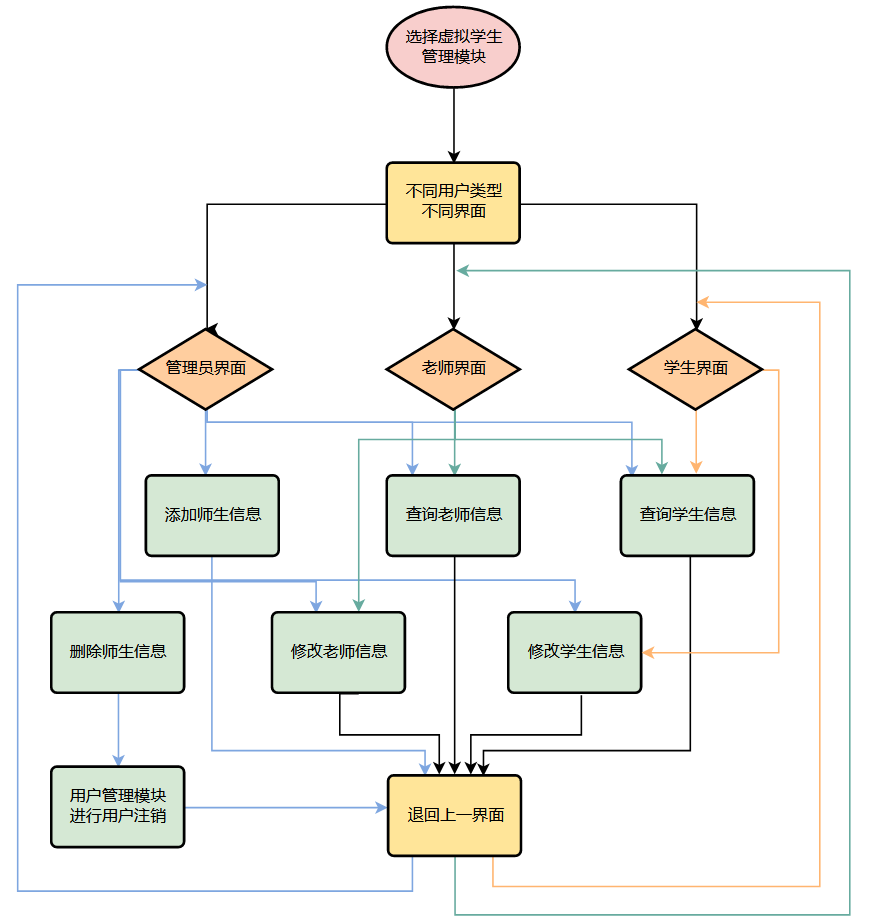
## 系统设计

## 界面设计

Client端：



## 模块流程图



## 类分析

**实体类**：师生基本信息，BasicInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | id | String | 9个字符 | 登录名即一卡通号，例：  213190998 学生账号  103009076 教师账号 |
| 2 | name | String | 2~3个字符 | 姓名，例：曹邹颖 |
| 3 | studentId | String | 8个字符 | 学号，例：09019204 |
| 4 | age | Integer | 非0 | 年龄，例：20 |
| 5 | nationality | String | 字符串 | 民族，例：han |
| 6 | birthday | String | 10个字符 | 生日，例：2001-06-12 |
| 7 | nativePlace | String | 字符串 | 籍贯，例：nantong |
| 8 | college | String | 字符串 | 学院，例：CS |
| 9 | politicalStatus | String | 三个  固定字符 | 政治面貌，分别为：  群众masses，  共青团员communist YouthLeagueMember,  中共党员communist PartyOfChinaPolitician |
| 10 | grade | String | 一些  固定字符 | 年级，例：大一freshman，大二sophomore，  大三junior，大四senior |
| 11 | major | String | 字符串 | 专业，例：math,art |
| 12 | dormitoryAdd | String | 字符串 | 宿舍地址，例：plumGarden8c428 |
| 13 | email | String | 8~20个字符 | 邮箱，例：1799771645@qq.com |
| 14 | mobilePhone | String | 11个字符 | 手机号，例：18362155818 |

**服务类**：

Client端：StudentManagementClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改学生信息 | updateStudent(BasicInformation) | 管理员学生功能 |
| 2 | 修改老师信息 | updateTeacher(BasicInformation) | 管理员老师功能 |
| 3 | 查询学生信息 | searchStudent(BasicInformation) | 共有功能 |
| 4 | 查询老师信息 | searchTeacher(BasicInformation) | 管理员老师功能 |
| 5 | 增加师生信息 | addInfo(BasicInformation) | 管理员功能 |
| 6 | 删除师生信息 | deleteInfo(BasicInformation) | 管理员功能 |

Server端：StudentManagementServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改学生信息 | updateStudent(BasicInformation) | 管理员学生功能 |
| 2 | 修改老师信息 | updateTeacher(BasicInformation) | 管理员老师功能 |
| 3 | 查询学生信息 | searchStudent(BasicInformation) | 共有功能 |
| 4 | 查询老师信息 | searchTeacher(BasicInformation) | 管理员老师功能 |
| 5 | 增加师生信息 | addInfo(BasicInformation) | 管理员功能 |
| 6 | 删除师生信息 | deleteInfo(BasicInformation) | 管理员功能 |

# 虚拟教务模块设计说明

## 模块背景

高校教务管理工作是高等教育中的一个极为重要的环节，是整个院校管理的核心和基础。面对种类繁多的数据和报表，手工处理方式已经很难跟上现代化管理的步伐，随着计算机及通讯技术的飞速发展，高等教育对教务管理工作提出了更高的要求。本系统的虚拟教务模块主要是针对课程管理与成绩处理的一系列相关工作，本模块的建立使课程管理与成绩处理更加规范化、系统化，查询手段更加便捷化。同时，教务管理这一模块也是实现高校数据处管理现代化和信息化的重要内容。

本系统虚拟教务模块面向管理员、教师和全校学生，实现课程安排、成绩处理，从而细分几个小模块：排课管理模块：管理所有课程的基本信息，包括对课程信息的添加、修改、删除等操作；可以根据各种条件查询出需要的信息，并对相应的信息进行操作，比如修改和删除，可以通过学生的学号查询出学生的相应课程信息，对相应的课程进行修改、删除操作；选课管理模块：学生可对本人的课表信息进行查询，也可以根据培养计划进行选课；考务管理模块：考试安排，管理员可以安排考场和考试时间；成绩录入，任课教师可以在规定的时间段内录入学生成绩；查询成绩，学生可以根据学号查询出课程成绩。

## 需求分析

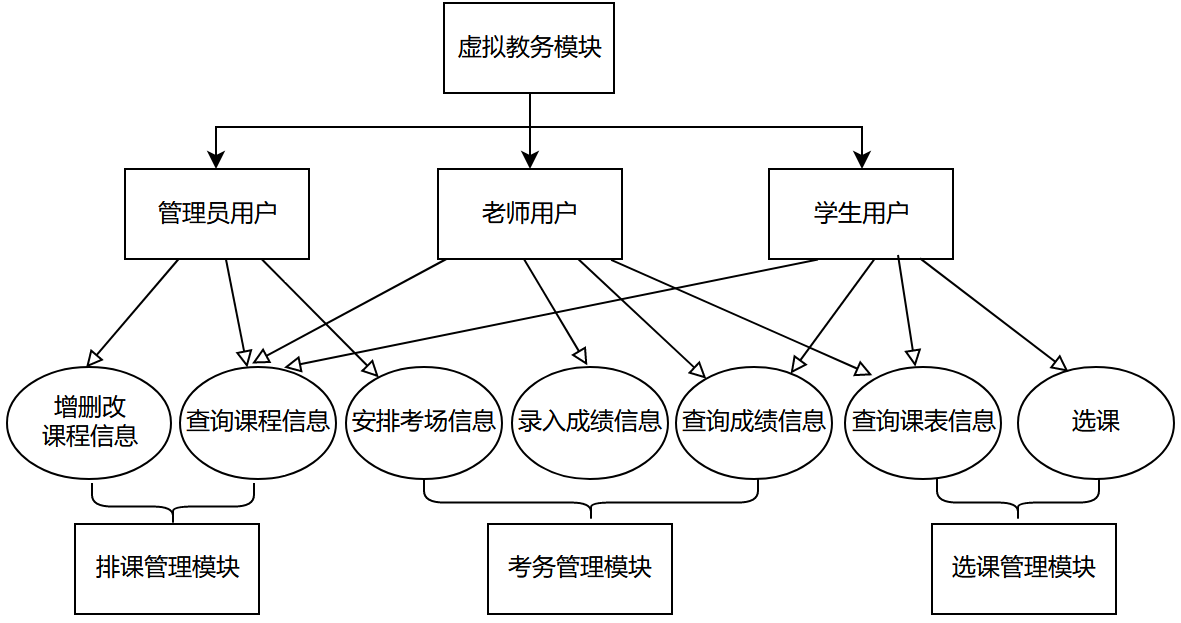


图4.虚拟教务模块E-R图

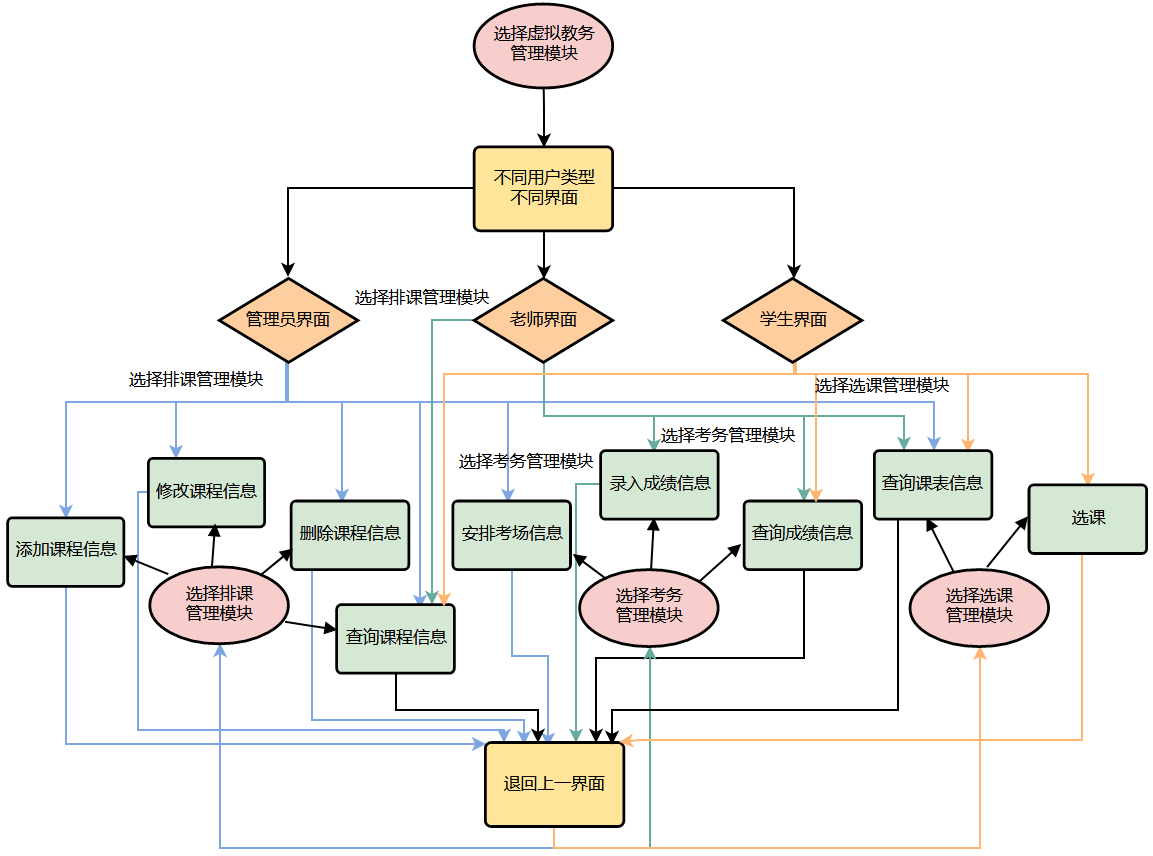
## 系统设计

## 界面设计

Client端：



## 模块流程图



## 类分析

**实体类**：

课程信息+考务信息，CourseInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | courseId | String | 8个字符 | 课程号，例：B18M0040 |
| 2 | classification | String | 字符串 | 类别，例：compulsory |
| 3 | courseName | String | 字符串 | 课程名，例：PEIV |
| 4 | credit | integer | 非0 | 学分，例：4 |
| 5 | roomId | String | 字符串 | 教室，例：J2-212 |
| 6 | courseDate | String | 字符串 | 上课时间，例：Mon5-6 |
| 7 | testDate | datetime |  | 考试时间，例：2021-7-10 |
| 8 | testRoomID | String | 字符串 | 考场，例：J2-212 |
| 9 | grade | integer | >=0，<=100 | 成绩，例：100 |
| 10 | teacher | String | 字符串 | 老师，例：Rechard |

课表信息，CurriculumInfo：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | college | String | 字符串 | 学院，例：CS |
| 2 | grade | String | 字符串 | 年级，例：grade2019 |
| 3 | major | String | 字符串 | 专业，例：math |
| 4 | courseName | String | 字符串 | 课程名，例：PEIV |
| 5 | roomId | String | 字符串 | 教室，例：J2-212 |
| 6 | courseDate | String | 字符串 | 上课时间，例：Mon5-6 |
| 7 | teacher | String | 字符串 | 老师，例：Rechard |

**服务类**：

Client端：AducationalAdminiClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改课程信息 | updateCourse(CourseInformation) | 管理员功能 |
| 2 | 增加课程信息 | addCourse(CourseInformation) | 管理员功能 |
| 3 | 删除课程信息 | deleteCourse(CourseInformation) | 管理员功能 |
| 4 | 查询课程信息 | searchCourse(CourseInformation) | 共有功能 |
| 5 | 安排考场信息 | arrangeTest(CourseInformation) | 管理员功能 |
| 6 | 录入成绩信息 | entryResults(CourseInformation) | 老师功能 |
| 7 | 查询成绩信息 | queryResults(CourseInformation) | 共有功能 |
| 8 | 查询课表信息 | queryTimetable(CurriculumInfo) | 共有功能 |
| 9 | 选课 | ChooseCourse(CurriculumInfo) | 学生功能 |

Server端：AducationalAdminiServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改课程信息 | updateCourse(CourseInformation) | 管理员功能 |
| 2 | 增加课程信息 | addCourse(CourseInformation) | 管理员功能 |
| 3 | 删除课程信息 | deleteCourse(CourseInformation) | 管理员功能 |
| 4 | 查询课程信息 | searchCourse(CourseInformation) | 共有功能 |
| 5 | 安排考场信息 | arrangeTest(CourseInformation) | 管理员功能 |
| 6 | 录入成绩信息 | entryResults(CourseInformation) | 老师功能 |
| 7 | 查询成绩信息 | queryResults(CourseInformation) | 共有功能 |
| 8 | 查询课表信息 | queryTimetable(CurriculumInfo) | 共有功能 |
| 9 | 选课 | ChooseCourse(CurriculumInfo) | 学生功能 |

# 虚拟图书馆模块设计说明

## 模块背景

随着智能手机的普及，虚拟图书馆这种模式能更加贴近师生的生活习惯，并且适合师生年龄段的习性，这对师生更多的接触到图书馆的信息有一定得促进作用。一套功能齐全的校内的虚拟图书馆模块，使其具有一般图书管理系统所具有的功能，可以实现管理员对图书信息的录入、增删改查功能，也帮助师生读者书籍查询、书籍借阅与归还、续借等功能，方便师生搜索书籍、借阅图书。

## 需求分析

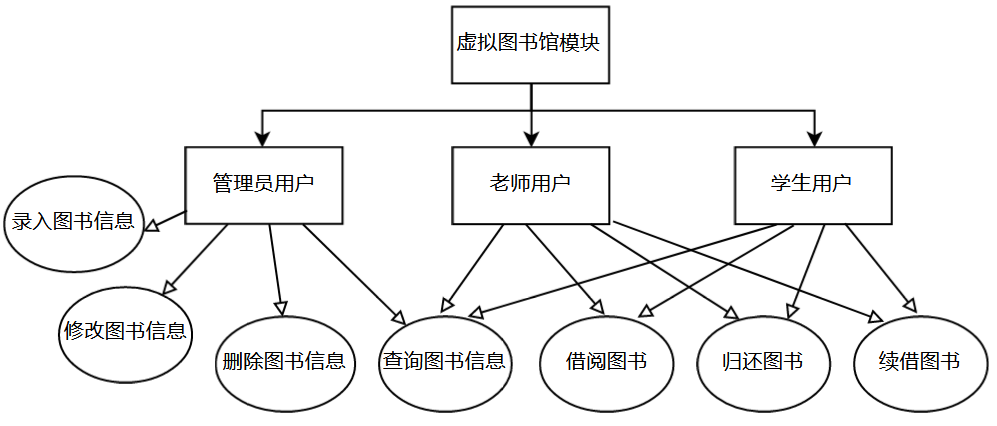


图5.虚拟图书馆模块E-R图

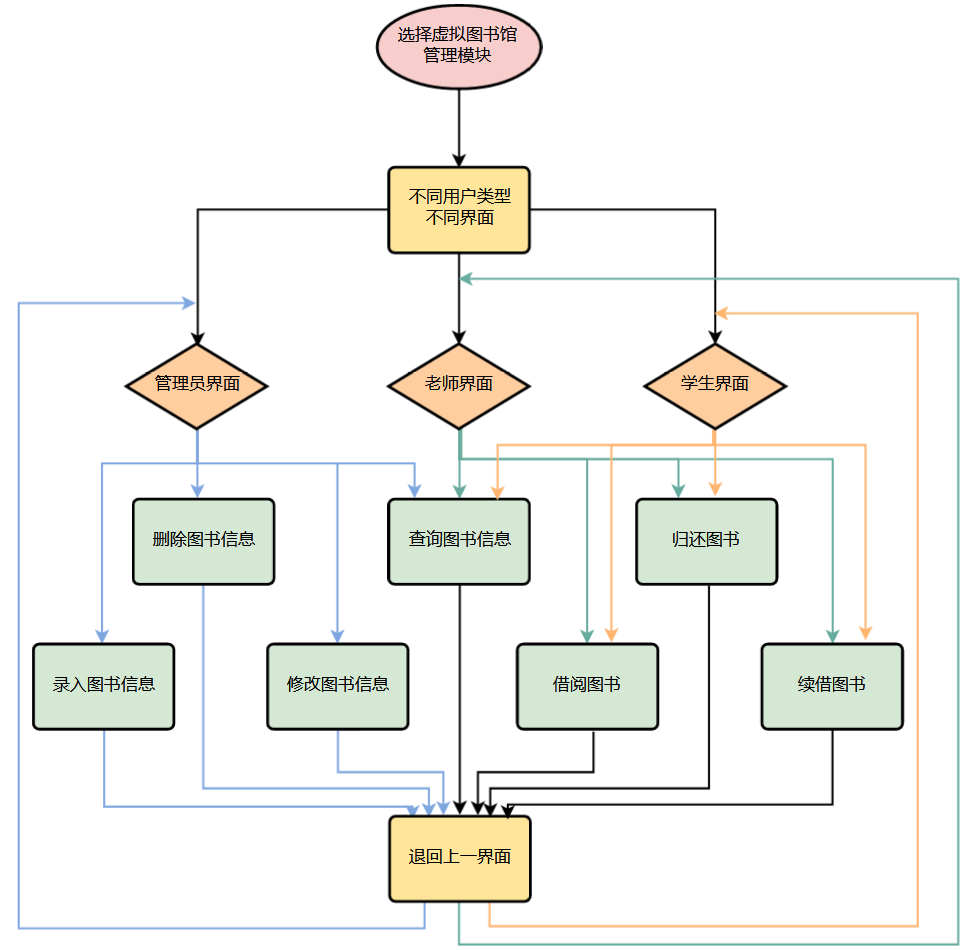
## 系统设计

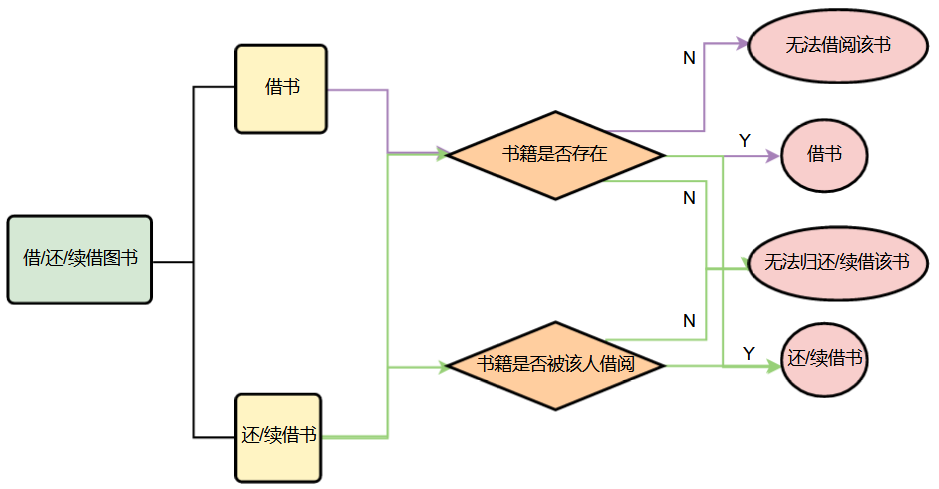
## 界面设计

Client端：



## 模块流程图





## 类分析

**实体类**：

图书信息，BookInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | bookId | String | 字符串 | 索书号，例：TP312JA/2579 |
| 2 | bookName | String | 字符串 | 书籍名称，例：面向对象程序设计:Java |
| 3 | state | String | 字符串 | 书籍状态(是否被借阅),例：已借 |
| 4 | price | Double | 非负 | 书籍价格，例：88.00 |
| 5 | writer | String | 字符串 | 书籍作者，例：李建勋,郭建华,佟瑞 |
| 6 | address | String | 字符串 | 馆藏地，例：基础书库 |
| 7 | borrower | String | 字符串 | 借阅学生编号，例：213190998,-(未借) |
| 8 | borrowDate | String | 字符串 | 借阅时间，例：2021-07-10，-(未借) |
| 9 | returnDate | String | 字符串 | 归还时间，例：2021-08-10，-(未借) |

**服务类**：

Client端：LibraryClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改图书信息 | updateBook(BookInformation) | 管理员功能 |
| 2 | 录入图书信息 | addBook(BookInformation) | 管理员功能 |
| 3 | 删除图书信息 | deleteBook(BookInformation) | 管理员功能 |
| 4 | 查询图书信息 | searchBook(BookInformation) | 共有功能 |
| 5 | 借阅图书 | borrowBook(BookInformation) | 师生功能 |
| 6 | 归还图书 | returnBook(BookInformation) | 师生功能 |
| 7 | 续借图书 | renewBook(BookInformation) | 师生功能 |

Server端：LibraryServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 修改图书信息 | updateBook(BookInformation) | 管理员功能 |
| 2 | 录入图书信息 | addBook(BookInformation) | 管理员功能 |
| 3 | 删除图书信息 | deleteBook(BookInformation) | 管理员功能 |
| 4 | 查询图书信息 | searchBook(BookInformation) | 共有功能 |
| 5 | 借阅图书 | borrowBook(BookInformation) | 师生功能 |
| 6 | 归还图书 | returnBook(BookInformation) | 师生功能 |
| 7 | 续借图书 | renewBook(BookInformation) | 师生功能 |

# 虚拟商店模块设计说明

## 模块背景

校园的虚拟商店模块通过将网络技术与商店运营相结合，商铺通过网上商店的方式作为商品展售的平台，师生通过智能终端完成所需商品的订购，使得师生们的在校学习和生活愈加快捷和方便，以更加高效的方式满足生活需求。虚拟商店模块通过集校内所有商铺到该模块中，师生更加方便的对各个商铺间的商品进行对比，同时也有利于商铺之间合理分配资源，从而大大提升商铺的收益。

虚拟商店模块能打造一个更贴近在校大学生的日常生活所需的购物软件，服务于师生，能给全校师生带来极大方便，师生只需要通过虚拟商店对需要的商品下单，当天下单，基本能确保当天或者第二天就能收到商品，甚至可以在校园商铺上进行二手交易，将自己闲置的物品销售出去等，还可以给学校同学提供兼职的机会。管理员负责商店运营模块，即对商品的操作，师生用户使用购物模块与账户模块，购物模块：创建订单，填写订单号(ID)、商品信息、订单状态、折扣等基本信息后创建订单；向订单中添加商品，进入已创建的订单中，向已创建订单中添加新的商品；删除订单，删除一个已创建的订单；账户模块：在购物订单交易时，首先要确定用户余额足够支持交易，在交易成功后，用户账户余额会扣除相应的钱额；可通过虚拟校园的另一功能模块——虚拟银行，对账户余额进行一定相关操作，享受线上购物功能。

## 需求分析

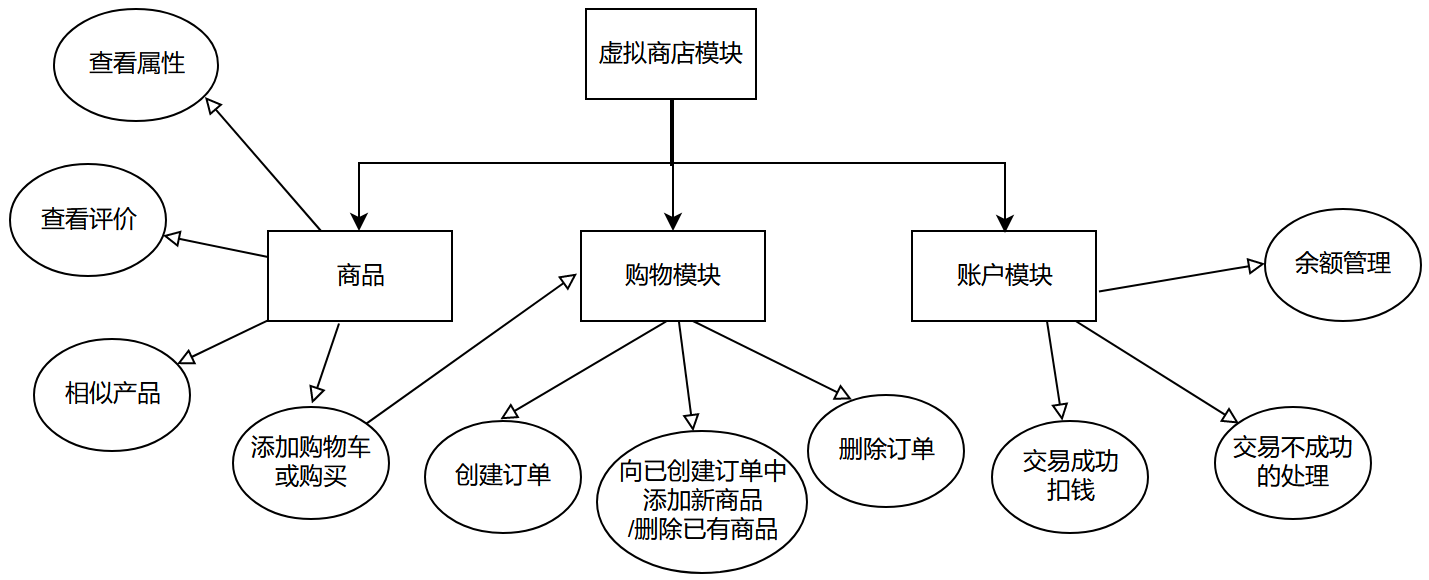


图6.虚拟商店模块E-R图

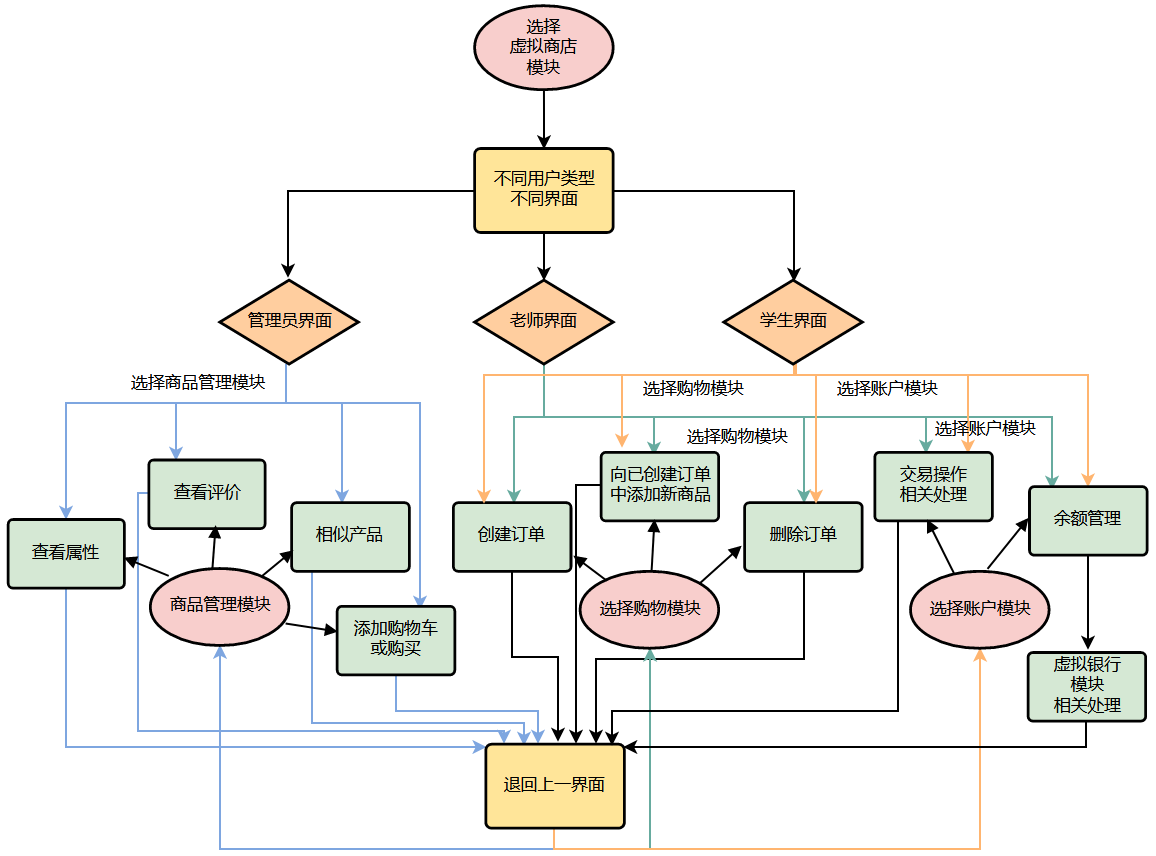
## 系统设计

## 界面设计

Client端：



## 模块流程图



## 类分析

**实体类**：商品信息，CommodityInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | commodityId | String | 字符串 | 商品号，例：SC124350612 |
| 2 | commodityName | String | 字符串 | 商品名称，例：南瓜吐司 |
| 3 | priceTag | Double | 非负 | 商品标价，例：29.90 |
| 4 | brand | String | 字符串 | 商品品牌，例：来伊份 |
| 5 | commodityType | String | 字符串 | 商品类型，例：食品 |
| 6 | display | 图片 |  | 商品图片，例： |
| 7 | quantity | Integer | 非负 | 商品库存，例：62 |
| 8 | commands | String | 字符串 | 商品评价，例：第一次购买，尝了下酥松绵软，味道非常好，是旅途中充饥好伴侣！ |
| 9 | classifier | String | 字符串 | 商品量词，例：500g/箱、50g/包 |
| 10 | discount | Double | 非负 | 商品折扣，例：0.90 |

订单信息，OrderInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | orderId | String | 字符串 | 订单号，例：431579744612 |
| 2 | orderState | String | 字符串 | 订单状态，例：已支付 |
| 3 | commoditySum | Double | 非负 | 交易金额，例：0.01 |
| 4 | brand | String | 字符串 | 买家购物号，例：213190998 |
| 5 | payTime | String | 字符串 | 支付时间，例：2021-07-10 16:08 |
| 6 | bulidTime | String | 字符串 | 创建时间，例：2021-07-10 15:58 |
| 7 | number | Integer | 非负 | 商品数量，例：2 |
| 8 | deliveryTime | String | 字符串 | 发货时间，例：2021-07-11 08:08 |
| 9 | balance | Double | 非负 | 账户余额，例：3848.00 |

**服务类**：

Client端：StoreClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查看商品属性 | checkGoods(CommodityInformation) |  |
| 2 | 查看商品评价 | checkCommands(CommodityInformation) |  |
| 3 | 查看相似商品 | searchSimilar(CommodityInformation) |  |
| 4 | 添加购物车或购买  <=>创建订单 | buildOrder(CommodityInformation,  OrderInformation) |  |
| 5 | 订单删除 | orderDelete(OrderInformation) |  |
| 6 | 向已创建订单中添加新商品/删除已有商品 | orderChange(CommodityInformation,  OrderInformation) |  |
| 7 | 订单交易 | orderTransaction(OrderInformation) |  |
| 8 | 余额管理 | balanceManagement(OrderInformation) |  |

Server端：StoreServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查看商品属性 | checkGoods(CommodityInformation) |  |
| 2 | 查看商品评价 | checkCommands(CommodityInformation) |  |
| 3 | 查看相似商品 | searchSimilar(CommodityInformation) |  |
| 4 | 添加购物车  或购买  <=>创建订单 | buildOrder(CommodityInformation,  OrderInformation) |  |
| 5 | 订单删除 | orderDelete(OrderInformation) |  |
| 6 | 向已创建订单中添加新商品/删除已有商品 | orderChange(CommodityInformation,  OrderInformation) |  |
| 7 | 订单交易 | orderTransaction(OrderInformation) |  |
| 8 | 余额管理 | balanceManagement(OrderInformation) |  |

# 虚拟银行模块设计说明

## 模块背景

作为智慧校园系统，虚拟银行模块可用于校园银行，模拟银行的存取款业务，高校机构内部记账支付平台，可以让师生用户使用在本系统中的账号能像银行卡一样使用，也可登录自助客户端自助转账抑或挂失已丢失的一卡通账户，管理员也可登录管理端对师生用户进行最高权限的管理，例如管理账户信息、账户扣钱相关操作乃至封户操作。

老师、学生可管理自身的基本账户信息，对个人账户余额的操作包括存款、转账、查询、取款等，可以根据各种条件查询出需要的账户余额信息，甚至对于账户进行挂失操作。

管理员可管理所有师生的基本账户信息，对学生账户余额的操作包括查询、扣钱等，对于余额不足扣钱不成功的情况再将对应的学生名单存入数据库的一张表中，甚至对于账户进行封户操作。

## 需求分析

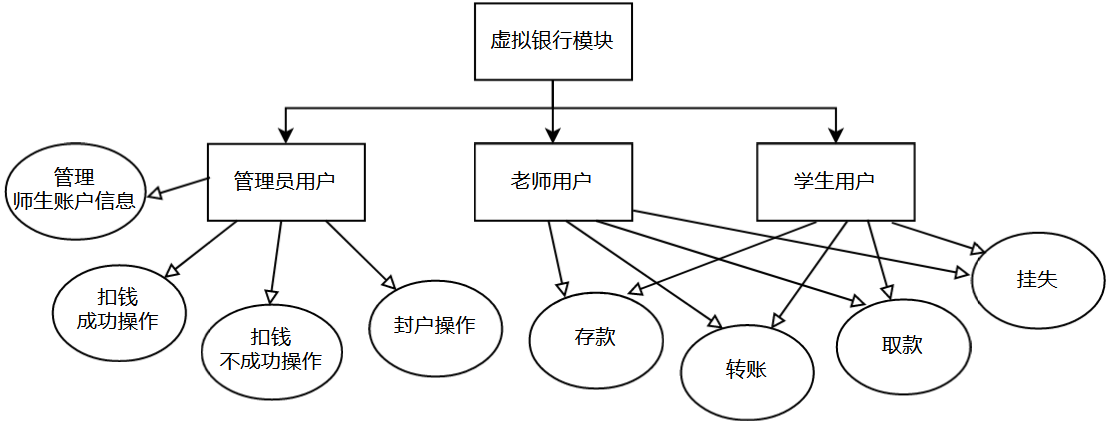


图7.虚拟银行模块E-R图

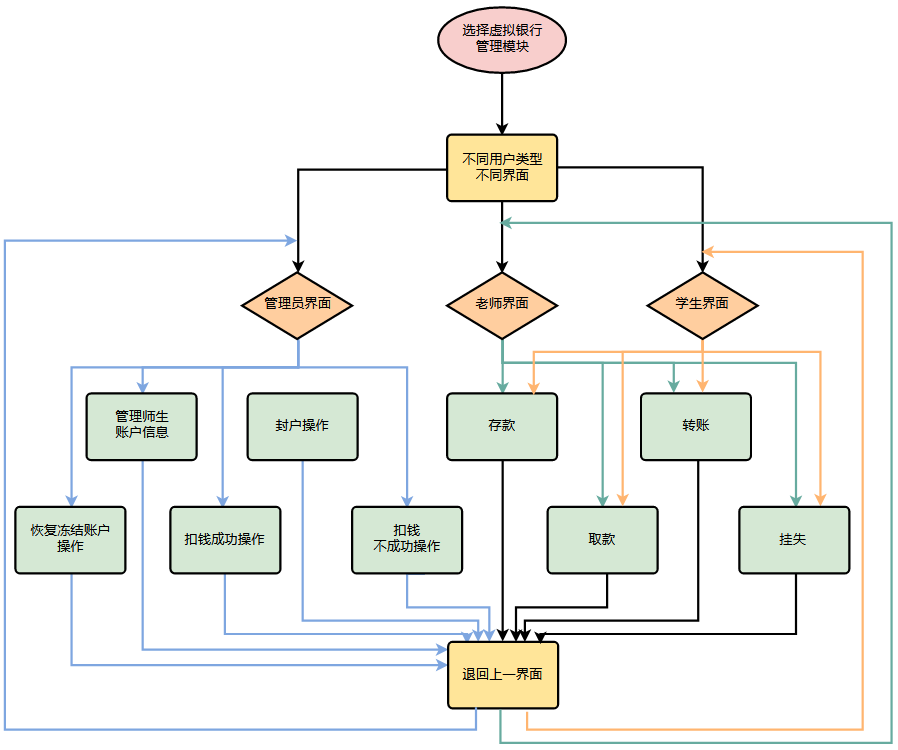
## 系统设计

## 界面设计

Client端：



## 模块流程图



## 类分析

**实体类**：账户信息，AccountInformation：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | accountId | String | 字符串 | 银行账号ID，例：213190998 |
| 2 | accountbalance | Double | 非负 | 账户余额，例：126.00 |
| 3 | accountStatus | String | 字符串 | 账户状态，例：激活状态 |
| 4 | accountOwner | String | 字符串 | 账户持有人，例：曹邹颖 |
| 5 | accountPassword | String | 字符串 | 账户密码，例：czy010612 |

**服务类**：

Client端：BankClientSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查看账户信息 | checkAccount(AccountInformation) | 共有功能 |
| 2 | 账户扣钱操作 | checkCommands(AccountInformation) | 管理员功能 |
| 3 | 封户操作 | freezeAccount(AccountInformation) | 管理员功能 |
| 4 | 恢复冻结账户 | rebackAccount(AccountInformation) | 管理员功能 |
| 5 | 存款 | deposit(AccountInformation) | 师生功能 |
| 6 | 取款 | withdraw(AccountInformation) | 师生功能 |
| 7 | 转账 | transfer(AccountInformation) | 师生功能 |
| 8 | 修改密码 | modifyPassword(AccountInformation) | 师生功能 |
| 9 | 挂失 | deleteAccount(AccountInformation) | 师生功能 |

Server端：BankServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 查看账户信息 | checkAccount(AccountInformation) | 共有功能 |
| 2 | 账户扣钱操作 | checkCommands(AccountInformation) | 管理员功能 |
| 3 | 封户操作 | freezeAccount(AccountInformation) | 管理员功能 |
| 4 | 恢复冻结账户 | rebackAccount(AccountInformation) | 管理员功能 |
| 5 | 存款 | deposit(AccountInformation) | 师生功能 |
| 6 | 取款 | withdraw(AccountInformation) | 师生功能 |
| 7 | 转账 | transfer(AccountInformation) | 师生功能 |
| 8 | 修改密码 | modifyPassword(AccountInformation) | 师生功能 |
| 9 | 挂失 | deleteAccount(AccountInformation) | 师生功能 |

# 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | uid | 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 |
| … | … | … | … | … |

其中data类包括的数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 约束 | 备注 |
| 1 | pwd | String | 6~16个字符 | 密码 |
| 2 | name | String | 6~16个字符 | 名字 |
| 3 | grade | Integer |  | 年级 |
| 4 | call | String | 6~16个字符 | 联系方式 |
| 5 | courseName | String | 6~16个字符 | 课程名称 |
| 6 | courseData | Data |  | 上课时间 |
| 7 | coursePlace | String |  | 上课地点 |
| 8 | courseGrade | Integer | 0~100 | 课程成绩 |
| 9 | bookName | String |  | 书籍名称 |
| 10 | bookId | Integer | 8位 | 书籍编号 |
| 11 | bookNum | Integer | 非负 | 书籍在馆数量 |
| 12 | borrowTime | Data |  | 借阅时间 |
| 13 | returnTime | Data |  | 归还时间 |
| 14 | goodsName | String |  | 商品名称 |
| 15 | goodsNum | Integer | 非负 | 商品数量 |
| 16 | goodsPrice | Integer | 正数 | 商品价格 |
| 17 | balance | Integer | 非负 | 账户余额 |

## 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 | privilege | Integer | 取值为0,1,2 | 权限 |

## 数据库工具类DbHelper

**AppConstants.java中，**

**public** **class** AppConstants {

// JDBC

**public** **static** **final** String ***JDBC\_URL*** = "jdbc:mysql://localhost:3306/student?useUnicode=true&characterEncodeing=UTF-8";

**public** **static** **final** String ***JDBC\_USERNAME*** = "root";

**public** **static** **final** String ***JDBC\_PASSWORD*** = "heye010612";

**public** **static** **final** String ***JDBC\_DRIVER*** = "com.mysql.cj.jdbc.Driver";

}

**DbHelper.java中，**

**public** **class** DbHelper {

// 定义连接对象

**private** Connection connection;

**private** Statement statement;

/\*\*

\*

\* **@throws** Exception

\*/

**public** DbHelper() **throws** Exception {

**this**.connection = getCon();

**this**.statement = connection.createStatement();

}

/\*\*

\* 获取数据库连接

\*

\* **@return** 数据库连接Connection

\* **@throws** Exception

\*/

**public** Connection getCon() **throws** Exception {

Class.*forName*(AppConstants.***JDBC\_DRIVER***);

Connection con = DriverManager.*getConnection*(AppConstants.***JDBC\_URL***, AppConstants.***JDBC\_USERNAME***,

AppConstants.***JDBC\_PASSWORD***);

**return** con;

}

/\*\*

\* 关闭数据库连接

\*

\* **@param** con

\* **@throws** SQLException

\*/

**public** **void** closeCon(Connection con) **throws** SQLException {

**if** (con != **null**) {

con.close();

}

}

/\*\*

\* 更新

\*

\* **@param** sql

\* **@return**

\* **@throws** SQLException

\*/

**public** **boolean** update(String sql) **throws** SQLException {

**return** statement.execute(sql);

}

/\*\*

\* 添加

\*

\* **@param** sql

\* **@return**

\* **@throws** SQLException

\*/

**public** **boolean** add(String sql) **throws** SQLException {

**return** statement.execute(sql);

}

/\*\*

\* 删除

\*

\* **@param** sql

\* **@return**

\* **@throws** SQLException

\*/

**public** **boolean** delete(String sql) **throws** SQLException {

**return** statement.execute(sql);

}

/\*\*

\* 查询

\*

\* **@param** sql

\* **@return**

\* **@throws** SQLException

\*/

**public** ResultSet query(String sql) **throws** SQLException{

**return** statement.executeQuery(sql);

}

}

## 数据存取类DAO

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 增加记录 | addRecord(sheet,row) |  |
| 2 | 删除记录 | deleteRecord(sheet,index) |  |
| 3 | 修改记录 | updateRecord(sheet,index,colomn,newvalue) |  |
| 4 | 查找记录 | searchRecord(sheet, criteria) |  |
| 5 | … | … | … |

## 工具类

构建适当的工具类

## 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 receive(Message) |  |
| 3 | 校验 | Boolean validate(Message) |  |
| 4 | 解析类型 | getType(Message) |  |
| 5 | 得到名字 | String getName(Message) |  |
| 6 | 得到发送者 | String getSender(Message) |  |
| 4 | 设定类型 | setType(Message) |  |
| 5 | 设定名字 | setName(Message) |  |
| 6 | 设定发送者 | setSender(Message) |  |
| 7 | … | … | … |

## 服务器端IMessageServerSrv

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 备注 |
| 1 | 发送 | Boolean send(Message) |  |
| 2 | 接收 | Boolean receive(Message) |  |
| 3 | 校验 | Boolean validate(Message) |  |
| 4 | 解析类型 | getType(Message) |  |
| 5 | 得到名字 | String getName(Message) |  |
| 6 | 得到发送者 | String getSender(Message) |  |
| 4 | 设定类型 | setType(Message) |  |
| 5 | 设定名字 | setName(Message) |  |
| 6 | 设定发送者 | setSender(Message) |  |
| 7 | … | … |  |

## 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() | 启动服务器 |  |
| **3** | stop() | 中止服务器 |  |
| **4** | restart() | 重新启动服务器 |  |
| **5** | isClosed() | 是否关闭 |  |
| **6** | netConnect() | 开启网络连接 |  |
| **7** | … | … | … |

// 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**);

}

});

}

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **名称** | **说明** | **备注** |
| **1** | EventListenerThread | 事件监听类 | 监听可能发生的事件 |
| **2** | ThreadMan | 线程管理类 |  |

## 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** |
|  | vCampus.accdb | 数据库 | MySQL |
| **1** | tblUser | 用户表 | ID（这里是一卡通号码）  登陆密码  学号（如果有的话）[1]  姓名  性别  生日  联系方式  Email  宿舍（如果有的话）  所在年级  逻辑班级[2]（如果有的话）  访问权限[3] |
| **2** | tblChooseCourse | 选课表 | 区别于课程表 |
| **3** | tblCourse | 课程表 | 课程ID  课程名称  任课老师  学分  任课学期  学期总课时  （上课教室）[4]  （周上课时间安排） |
| **4** | tblBank | 银行表 | 一卡通号码  一卡通余额  一卡通贷款额  一卡通欠款额度 |

## 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（代表了用户的权限） |
| **7** | StuNum | nvachar（8） |  | NULL |  | 如果有的话 |
| **8** | uPhone | nvachar（8） |  | NULL | 定长—17077889653 | 电话号码 |
| **9** | E-mail | varchar（10） |  | NULL |  | Email |
| **10** | uRoom | varchar（10） |  | NULL | M4C425 | 宿舍具体位置 |
| **11** | StuGrade | nvarchar（8） |  | NULL | Ux|Px|Dx|AD | U、P、D、AD分别表示本科、硕士、博士生、博士后 |
| **12** | uClass | Nvachar（8） |  | NULL | 具体来定 | 代表学生所在班级以及老师所带班级（如果是班主任的话）、管理员此栏不填 |

## tblChooseCourse表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段** | **类型** | **是否主键** | **默认值** | **约束** | **备注** |
| **1** | cId | nvarchar（8） | PRI | NULL |  | 课程ID |
| **2** | uId | nvarchar（8） |  | NULL |  | 学生一卡通 |
| **3** | cTeacher | nvachar（8） |  | NULL |  | 老师姓名 |
| **4** | cGrade | float(4) |  | NULL |  | 学生获得的成绩 |
| **5** | cUpdate | Boolean |  | False |  | 老师是否提交了成绩 |
| **6** | cCheck | Boolead |  | False |  | 管理员是否审核了成绩 |

## tblCourse表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段** | **类型** | **是否主键** | **默认值** | **约束** | **备注** |
| **1** | cId | nvarchar（8） | PRI | NULL |  | 课程ID |
| **2** | cName | varchar（10） |  | NULL |  | 课程名称 |
| **3** | cTeacher | nvachar（8） |  | NULL |  | 老师姓名 |
| **4** | cMark | float |  | NULL |  | 学分 |
| **5** | cTerm | nvarchar（8） |  | NULL |  | 任课学期 |
| **6** | cSumTime | int（4） |  | NULL |  | 学期总课时 |
| **7** | cRoom | nvachar（8） |  | NULL |  | 课程教室 |
| **8** | cTiming | nvachar（8） |  | NULL |  | 每周上课安排 |

## tblBank表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段** | **类型** | **是否主键** | **默认值** | **约束** | **备注** |
| **1** | bId | nvarchar（8） | PRI | NULL | 定长—09010210 | 一卡通号 |
| **2** | bDeposit | double（16） |  | NULL |  | 一卡通余额 |
| **3** | bLoad | double（16） |  | NULL |  | 一卡通贷款额 |
| **4** | bDebit | double（16） |  | NULL |  | 一卡通欠款额度 |

[1]如果没有这一栏就直接不填，初始化为NULL，比如管理员就没有宿舍这一说。

[2]逻辑班级是指不同于上课班级，比如计科一班和王世杰老师的信号与系统班级，前者就为逻辑班级。

[3]访问权限是指一个专属的枚举值，在对每一个表访问之间都要先检查这个权限值，用来赋予对于这个表能给予的操作权限，如学生对于大部份表就只能查看，而老师可以修改成绩表，管理员可以修改大部份表。

[4]带括号的是可以增加的表项、你们可以考虑一下加不加，因为涉及到选课算法，可能会复杂一些。

# 其他