**资源共享管理系统设计文档**

****

**学 院 智能与计算学部**

**专 业 软件工程**

**姓 名 张赫**

**学 号 3018216246**

**姓 名 马骁**

**学 号 3018216235**

**姓 名 邢思洋**

**学 号 3018216242**

**姓 名 郑启俊**

**学 号 3018216250**

**班 级 软工5班**

**版 本 4.0**

目录

[1. 概要设计 3](#_Toc18622)

[1.1编写目的 3](#_Toc6670)

[1.2背景 3](#_Toc19075)

[1.3需求规定 3](#_Toc23585)

[1.3.1运行环境 3](#_Toc19904)

[1.3.2软件体系架构图 4](#_Toc5930)

[1.3.3人工处理过程 4](#_Toc27401)

[1.4接口设计 4](#_Toc32323)

[1.4.1用户接口 4](#_Toc31364)

[1.4.2外部接口 5](#_Toc31911)

[1.4.3内部接口 5](#_Toc1920)

[1.5功能模块的设计与实现 5](#_Toc4447)

[1.5.1客户端： 5](#_Toc6641)

[1.5.2服务器端： 5](#_Toc23900)

[2.框架设计 6](#_Toc19628)

[2.1数据流复审 6](#_Toc7992)

[2.2 数据流图设计 6](#_Toc7826)

[2.3由程序流图导出的程序结构 10](#_Toc29242)

[2.3.1 总体架构 10](#_Toc7695)

[3.3 运行设计 14](#_Toc4694)

[3.4 错误处理设计 14](#_Toc18841)

[3.4.1出错信息 14](#_Toc13625)

[3.4.2 应对措施 14](#_Toc5390)

[4.数据库设计文档 15](#_Toc18593)

[4.1 数据库结构详细设计 15](#_Toc17188)

[4.1.1数据库类型 15](#_Toc10387)

[4.1.2 概念设计 15](#_Toc31568)

[4.2数据库表设计 16](#_Toc13359)

[3.5 E-R图设计 16](#_Toc1984)

[3.6 数据表分布结构 17](#_Toc7729)

[3.7外部设计 17](#_Toc17511)

[3.7.1需要遵守的外部约定 17](#_Toc32347)

[3.8安全性设计 17](#_Toc20267)

[3.8.1防止用户直接操作数据库的方法 17](#_Toc4962)

[3.8.2 用户帐号密码的加密方法 17](#_Toc19122)

[4. 小组分工 18](#_Toc24855)

# 1. 概要设计

## 1.1编写目的

编写该文档的目的在于对于 资源共享管理系统整体框架的明确，以便对于概要设计，详细设计，数据库设计的进一步完成。该管理系统主要根据大学校园用户中进行闲置资源共享的需求进行设计，并充分了解了传统网上交易系统的特点及不足。进而对该系统进行功能定位和模块划分。以用来在 资源共享管理系统开发过程中起到指导作用，并给使用者及开发者提供简要的说明。

## 1.2背景

这两年来“共享”二字成了高频率出现的社会热词，共享单车、共享汽车、共享充电宝甚至共享宿舍一一出现在社会公众的视野里毫无疑问，共享经济正在不知不觉得颠覆着人们传统的消费理念与生活习惯。

共享经济的出现，对于生产者来说，降低了市场交易成本，对传统的经济模式带来不小的冲击，减少了中间商的作用；对于消费者来说，由原来的“买”变为了现在的“租”，大大减少交易成本，给消费者带来了便捷，节省了开支。这种相对于传统经济更加自由的交易模式在一定程度上提高了经济的运行效率与资源的利用效率。

随着计算机技术，网络通讯技术的快速发展和普及，各种基于互联网建立的服务平台正在影响着人们日常生活的方方面面，对于高校教育来说，为加强广大学生教师之间的联系，最大化合理利用教学资源，构建一个面对全部师生的校园资源共享管理系统是十分有必要的。

总而言之，在当前互联网将人与人之间的距离无限拉近之后，随时随地的租借变的更为方便，我们可以便捷地把身边的东西租借出去，一方面提高物品的利用率，另一方面也可以获取部分额外收入。

所以，本项目在五大发展理念的基础上和"互联网+"的背景下,提出优化革新传统高校校园服务平台模式,探索传统高校校园服务平台在共享时代的转型发展方向,并着力打造出一个精准把握学生需求的，自由的，灵活的，开放式的，安全的新型优质校园资源共享平台。

### 1.3需求规定

### 1.3.1运行环境

系统开发平台：java

数据库管理系统软件：MySQL

操作系统：客户端：windows10

服务器端：windows10

### 1.3.2软件体系架构图

### 1.3.3人工处理过程

1.需要用户进行登录/注册。

2.需要用户主动发布/修改商品。

3.需要用户主动发起订购/处理订单状态。

## 1.4接口设计

### 1.4.1用户接口

本软件不需要用户知道语法结构，只需要用户做到：

1. 在注册登录时，提供用户名和密码，以便使用利用第三方库完成登录访问本软件功能。
2. 在查找商品时，必须输入相关关键字。
3. 在需要发布商品/修改商品信息时，必须键入商品相应的正确信息。
4. 在订单正在进行时，必须用户主动完成对订单状态的修改。

### 1.4.2外部接口

通过restful接口访问数据库，用Mysql数据库访问技术对数据库访问并进行封装。

### 1.4.3内部接口

窗口化，耦合化。

## 1.5功能模块的设计与实现

### 1.5.1客户端：

1.用户注册：在主界面点击登录按钮，跳转到第三方登陆页面。

2.用户登录：登录界面输入ID与个人密码，由第三方服务器处理登录请求。

3.资源管理：此功能包括我的商品，创建商品、所有商品等界面功能。

4.订单管理：此功能包括我的订单，发起订购，修改订单状态等功能。

### 1.5.2服务器端：

1.消息响应：服务器监听用户的请求，当客户端发来请求时，对数据库进行相应的处理。

# 2.框架设计

## 2.1数据流复审

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 简述 | 定义 | 来源 | 去向 |
| 系统登录请求 | 个人用户名及密码登入系统 | 系统登录请求=用户名+密码 | 用户 | 服务器 |
| 登陆信息 | 系统对登陆结果的响应 | 个人信息=用户名+个人信息+系统提示 | 用户 | 服务器、用户 |
| 资源信息查询 | 浏览资源的请求 | 资源查询请求指令+目标资源名称 | 用户 | 服务器、用户 |
| 订单信息查询 | 用户对于订单信息的查询请求 | 订单信息查询=信息查询请求指令+目标订单名称 | 用户 | 服务器、用户 |
| 信息查询成功反馈 | 系统对于查询信息请求的反馈 | 员工信息反馈=目标信息+系统提示 | 服务器 | 用户 |
| 信息查询失败反馈 | 系统对于查询信息请求的反馈 | 员工信息反馈=系统提示 | 服务器 | 用户 |
| 资源发布请求 | 用户对于发布个人资源的请求 | 资源发布申请信息=资源发布指令+资源信息+时间长度 | 用户 | 服务器、用户 |
| 资源订购请求 | 用户对于发起资源订单申请的请求 | 资源订购信息=资源订购申请指令+目标资源信息+订单信息 | 用户 | 服务器、用户 |
| 订单审批请求 | 用户对于通过订单审批的请求 | 订单审批信息=订单审批指令+订单信息修改指令 | 用户 | 服务器、用户 |
| 资源信息修改请求 | 用户对于资源修改信息的请求 | 资源信息修改信息=资源信息修改指令+资源信息指令 | 用户 | 服务器 |

## 2.2 数据流图设计

零层数据流图：



一层数据流图：



流程细化：

登录：



资源发布：



资源信息查询：



资源信息修改：



订单审批：



资源订购：



订单信息查询：

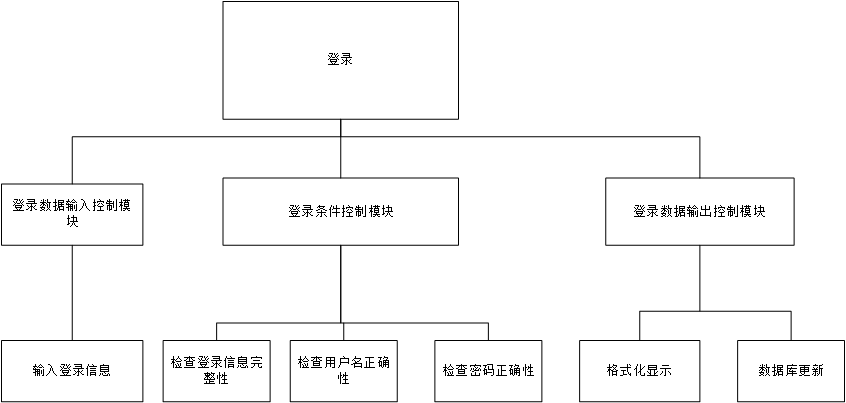


2.3由程序流图导出的程序结构

### **2.3**.1 总体架构



### 2.3.2 注册登录模块



### 2.3.3 资源发布模块



### 2.3.4 资源信息查询模块



### 2.3.5 资源订购模块



### 2.3.6 订单审批模块



### 2.3.7 基础信息管理模块

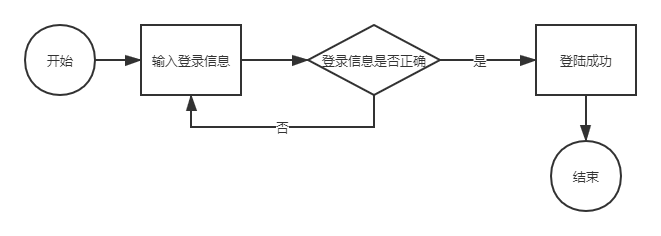


### 2.3.8 信息查询请求管理模块

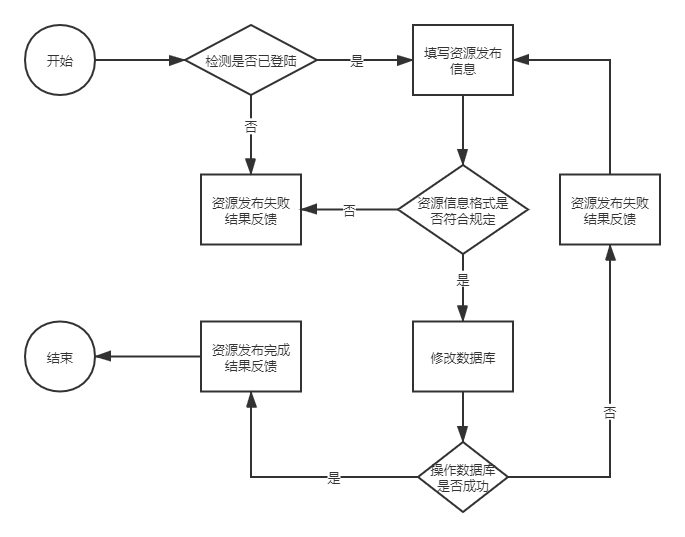


1. **详细设计**
   1. **模块设计**

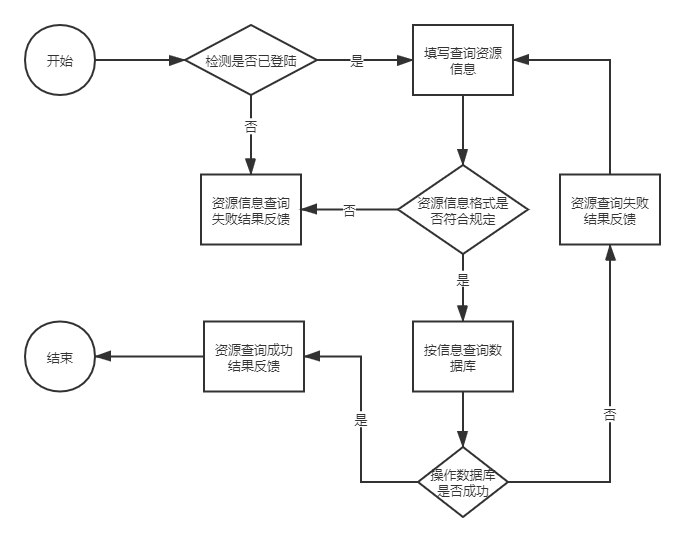
3.1.1 注册/登录模块



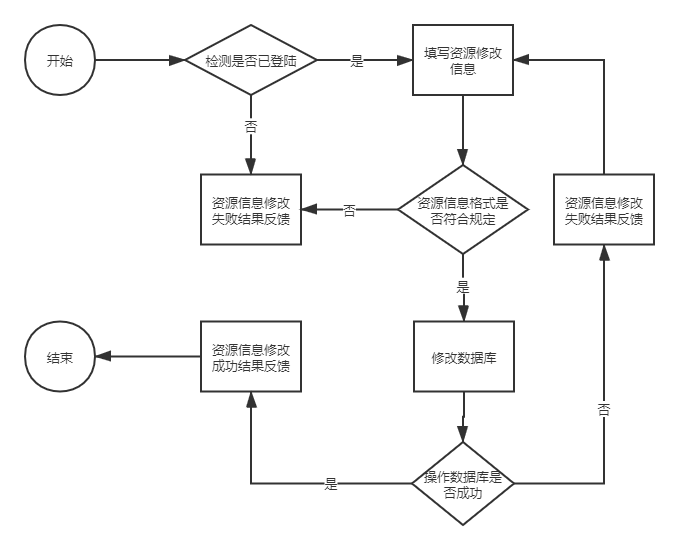
3.1.2 资源发布模块



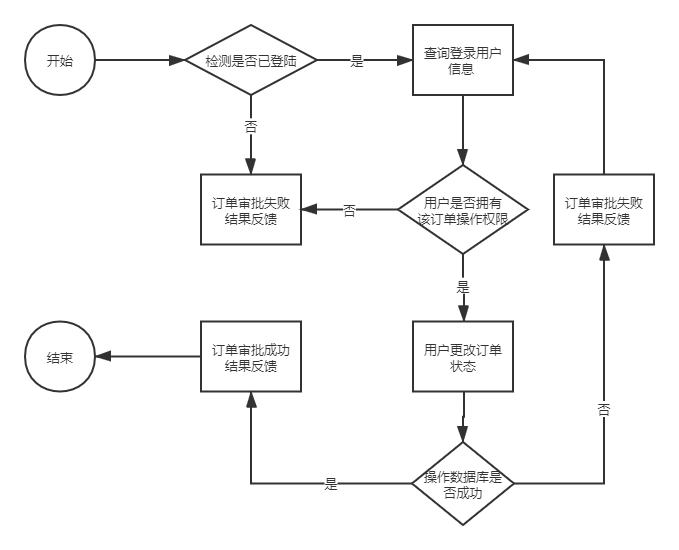
3.1.3 资源信息查询模块



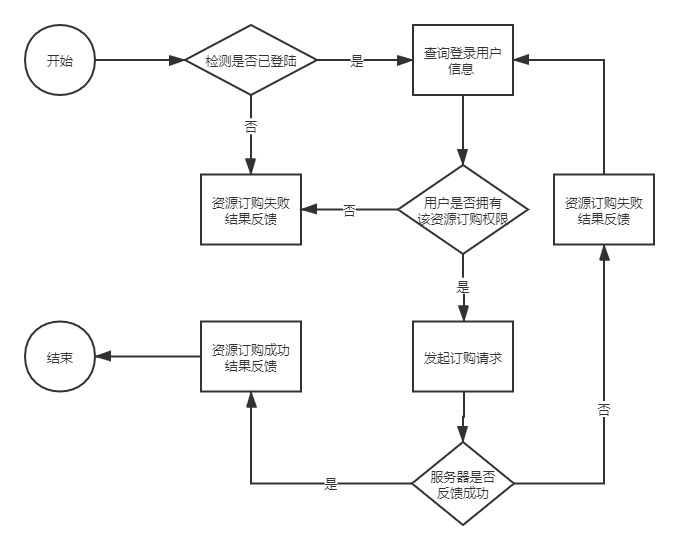
3.1.4 资源信息修改模块



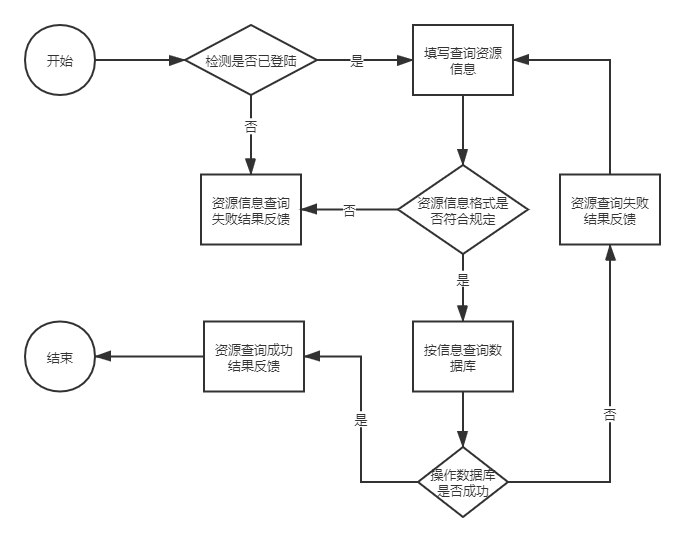
3.1.5 订单审批模块



3.1.6 资源订购模块



3.1.7 订单信息查询模块

****

* 1. **软件需求实现：**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [...nextauth].js | me.js | Myorder.js | mysell | form.js | oid.js | pid.js | Sid |
| 登陆 | √ |  |  |  |  |  |  |  |
| 资源信息查询 |  | √ |  | √ |  |  |  |  |
| 订单信息查询 |  |  | √ |  |  | √ |  |  |
| 订单审批请求 |  |  | √ |  |  |  |  |  |
| 资源信息修改 |  | √ |  | √ |  |  | √ |  |
| 资源发布 |  |  |  |  | √ |  |  |  |
| 资源信息查询 |  |  |  |  |  |  |  | √ |

## 3.3 运行设计

用户启动客户端后，弹出客户端登录界面。根据客户端上的按键，选择所需要的服务即可，例如对于需要注册的用户，点击注册按钮转到注册界面，注册成功后自动转至登录界面。在登录界面输入相应账号密码等信息，即可登录成功。

由用户自由控制所需要的服务内容。

## 3.4 错误处理设计

## 3.4.1出错信息

|  |  |  |
| --- | --- | --- |
| 错误/警告显示 | | 原因 |
| 网络连接失败 | 网络连接超时 | |
| 无用户名 | 用户在第三方数据库中查询不到 | |
| 密码输入错误 | 密码与第三方数据库中的不匹配 | |
| 输入为空 | 搜索商品界面/创建商品界面 | |
| 未登录 | 未在登陆状态使用相应功能 | |
| 查找失败 | 数据库中并未存贮该查找信息 | |
| …… | …… | |

### 3.4.2 应对措施

对于上述警告或者错误的发生，预备了错误应对机制，面对发生的错误或警告，将会给用户跳出警告弹窗，让用户理解是什么错误正在发生，以期提升人机交互感和用户体验感

## 4.数据库设计文档

4.1 数据库结构详细设计

4.1.1数据库类型

关系数据库：使用MySQL构建关系型数据库

4.1.2 概念设计

数据库负责存储商品信息和订单信息，每个用户的数据独立保存。

第三方数据库负责存储每个用户的当前状态、密码。并返回给软件数据库一个带有用户标签的session。

1.用户个人资料包含以下字段username、password、userimg、useremail，分别记录用户名，密码，用户头像，用户邮箱，都是字符串型

2.商品信息包含以下字段：p\_name，p\_text，p\_img,starttime,endtime指的是商品名和该商品对应的商品介绍，商品图片，开始时间，结束时间。字符串型。

3.订单信息包含以下字段：OrderID，fromID，toID, p\_name ,orderstate分别是订单ID,来源ID，目的ID，订单商品名称，订单状态。都是字符串型。

用户通过网页生成请求并将请求发至服务器，服务器连接数据库并处理请求信息，数据库根据服务器处理结果返回数据库信息

## 4.2数据库表设计

1)资源信息表：

|  |  |  |
| --- | --- | --- |
| ProductData | | |
| 字段类型 | 属性 | 数据类型 |
| PK | PID | integer |
|  | ProductImage | Char(100) |
|  | ProductName | Char(100) |
| FK | Uid\_id | Char(100) |
|  | Price | Char(100) |
|  | Starttime | DateTime(100) |
|  | Endtime | DateTime(100) |
|  | ProductState | Char(100) |

2)用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
| 字段类型 | 属性 | | | 数据类型 |
| PK | | Uid | Char(100) | |
|  | | Ukey | Char(100) | |

3）订单信息表

|  |  |  |
| --- | --- | --- |
| OrderData | | |
| 字段类型 | 属性 | 数据类型 |
| PK | FromID | integer |
|  | ToID | Char(100) |
|  | Pid | Char(100) |
| FK | OrderID | Char(100) |
|  | State | Char(100) |
|  | Ordertime | DateTime(100) |

## 未命名文件3.5 E-R图设计

3.6 数据表分布结构

客户端数据库中为每个用户保存一张商品数据表、一张订单数据表，分别命名为“用户ID\_Product”和“用户ID\_Order”。服务器与客户端的数据通过消息接收与传送模块进行同步于更新.

3.7外部设计

3.7.1需要遵守的外部约定

（1）商品图片链接大小上限为255字节。

（2）商品信息需符合各信息格式。

（3）时间格式为yyyy/MM/dd hh:mm:ss，小时为24小时制。

（4）订单关系为双向关系，仅当双方都成为某订单的关系用户时才显示在双方订单列表里

（5）当前用户仅能访问该用户对应的商品信息界面。

## 3.8安全性设计

3.8.1防止用户直接操作数据库的方法

用户只能用帐号登陆到网页，通过网页访问数据库，而没有其他途径操作数据库。

3.8.2 用户帐号密码的加密方法

依靠第三方对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。

3.8.3 角色与权限

确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。

## 4. 小组分工

|  |  |
| --- | --- |
| 马骁 | 详细设计服务器端部分和框架设计服务器端部分 |
| 张赫 | 概要设计和框架设计客户端部分 |
| 郑启俊 | 框架设计客户端部分和数据库设计 |
| 邢思洋 | 总体框架设计和及数据流复审 |