书卷香系统

项目详细设计说明书

**编写人员： 张泽亚、孙振昊**

**编写时间： 20.05.22--20.05..26**

**审核人员： 孙振昊、张泽亚**

[1引言 5](#_Toc12648)

[1.1编写目的 5](#_Toc18522)

[1.2背景 5](#_Toc10614)

[1.3定义 5](#_Toc22971)

[2程序系统的结构 6](#_Toc9154)

[2.1系统功能结构 6](#_Toc10364)

[2.2概述 7](#_Toc28467)

[模块划分： 7](#_Toc29452)

[功能划分： 7](#_Toc7030)

[2.3主要模块程序名 8](#_Toc4744)

[3个人信息（Personal information）设计说明 11](#_Toc30675)

[3.1程序描述 11](#_Toc22050)

[3.1.1账号注册 11](#_Toc1939)

[3.1.2用户登录 11](#_Toc26967)

[3.1.3个人信息 11](#_Toc4144)

[3.2功能 12](#_Toc27620)

[3.2.1账号注册 12](#_Toc2553)

[3.2.2用户登陆 12](#_Toc30297)

[3.2.3个人信息 13](#_Toc3792)

[3.3性能 13](#_Toc22615)

[3.3.1精度要求 13](#_Toc32753)

[3.3.2时间特性要求 14](#_Toc1472)

[3.4输入项 14](#_Toc8937)

[3.5输出项 14](#_Toc4705)

[3.6算法 15](#_Toc14100)

[3.7流程逻辑 16](#_Toc22487)

[3.8接口 18](#_Toc20586)

[3.9存储分配 18](#_Toc32710)

[3.10注释设计 18](#_Toc26984)

[3.11限制条件 18](#_Toc15510)[[1]](#footnote-0)

[3.12测试计划 18](#_Toc27967)

[3.13尚未解决的问题 19](#_Toc3414)

[4图书信息（Book information）设计说明 19](#_Toc17611)

[4.1程序描述 19](#_Toc3443)

[4.1.1推荐栏 19](#_Toc977)

[4.1.2图书搜索 19](#_Toc5988)

[4.2功能 19](#_Toc405)

[4.2.1推荐栏 19](#_Toc31098)

[4.2.2图书搜索 20](#_Toc16077)

[4.3性能 20](#_Toc13945)

[4.3.1精度要求 20](#_Toc19577)

[4.3.2时间特性要求 20](#_Toc22432)

[4.4输入项 21](#_Toc16935)

[4.5输出项 21](#_Toc17482)

[4.6算法 21](#_Toc1964)

[4.7流程逻辑 23](#_Toc29525)

[4.8接口 23](#_Toc5470)

[4.9存储分配 23](#_Toc11496)

[4.10注释设计 23](#_Toc10057)

[4.11限制条件 24](#_Toc1880)

[4.12测试计划 24](#_Toc769)

[4.13尚未解决的问题 24](#_Toc24631)

[5消费信息（Consumer information）设计说明 24](#_Toc11509)

[5.1程序描述 24](#_Toc23247)

[5.1.1购物车 24](#_Toc4466)

[5.1.2订单信息 24](#_Toc30625)

[5.1.3支付 24](#_Toc1727)

[5.2功能 24](#_Toc13463)

[5.2.1购物车 24](#_Toc30954)

[5.2.2订单信息 25](#_Toc25611)

[5.2.3支付 26](#_Toc21833)

[5.3性能 26](#_Toc6361)

[5.3.1精度要求 26](#_Toc15037)

[5.3.2时间特性要求 26](#_Toc14344)

[5.4输入项 27](#_Toc23853)

[5.5输出项 27](#_Toc1246)

[5.6算法 27](#_Toc15840)

[5.7流程逻辑 28](#_Toc25165)

[5.8接口 29](#_Toc10138)

[5.9存储分配 29](#_Toc7460)

[5.10注释设计 29](#_Toc5288)

[5.11限制条件 30](#_Toc27750)

[5.12测试计划 30](#_Toc5694)

[5.13尚未解决的问题 30](#_Toc433)

[6用户管理（User management）设计说明 30](#_Toc20436)

[6.1程序描述 30](#_Toc4519)

[6.2功能 30](#_Toc28265)

[6.3性能 31](#_Toc15527)

[6.3.1精度要求 31](#_Toc718)

[6.3.2时间特性要求 31](#_Toc16996)

[6.4输入项 31](#_Toc8047)

[6.5输出项 32](#_Toc32537)

[6.6算法 33](#_Toc23854)

[6.7流程逻辑 33](#_Toc26469)

[6.8接口 33](#_Toc18510)

[6.9存储分配 33](#_Toc8221)

[6.10注释设计 33](#_Toc32128)

[6.11限制条件 34](#_Toc31124)

[6.12测试计划 34](#_Toc16528)

[6.13尚未解决的问题 34](#_Toc4642)

[7图书管理（Book management）设计说明 34](#_Toc24903)

[7.1程序描述 34](#_Toc16889)

[7.2功能 34](#_Toc10355)

[7.3性能 35](#_Toc17028)

[7.3.1精度要求 35](#_Toc5698)

[7.3.2时间特性要求 35](#_Toc21992)

[7.4输入项 35](#_Toc17480)

[7.5输出项 36](#_Toc5982)

[7.6算法 36](#_Toc4333)

[7.7流程逻辑 36](#_Toc9081)

[7.8接口 36](#_Toc10009)

[7.9存储分配 36](#_Toc32672)

[7.10注释设计 37](#_Toc19531)

[7.11限制条件 37](#_Toc8761)

[7.12测试计划 37](#_Toc28814)

[7.13尚未解决的问题 37](#_Toc18785)

[8订单管理（Order management）设计说明 37](#_Toc5859)

[8.1程序描述 37](#_Toc31775)

[8.2功能 37](#_Toc8721)

[8.3性能 38](#_Toc12918)

[8.3.1精度要求 38](#_Toc12552)

[8.3.2时间特性要求 38](#_Toc17090)

[8.4输入项 38](#_Toc20273)

[8.5输出项 39](#_Toc26435)

[8.6算法 39](#_Toc8286)

[8.7流程逻辑 40](#_Toc1224)

[8.8接口 40](#_Toc11588)

[8.9存储分配 40](#_Toc25553)

[8.10注释设计 40](#_Toc23706)

[8.11限制条件 40](#_Toc22434)

[8.12测试计划 41](#_Toc18001)

[8.13尚未解决的问题 41](#_Toc27425)

**详细设计说明书**

# 1引言

## 1.1编写目的

本文档是建立在概要设计文档之上，也参考了需求文档及数据库设计文档，为明确书卷香网上书店的的结构与功能的详细划分，对各模块的功能调用细节，安排项目规划与进度、制定详细测试计划、组织软件开发与测试，特撰写此文档。

在上一阶段概要设计中，已经明确系统的模块功能的划分，大致划分了各个模块的接口、接口的设计，同时也对数据结构作出定义，以下为本设计文档的报告内容。

本文档的预期读者为书卷香项目的开发人员，项目经理，测试员，架构师等相关人员。

## 1.2背景

软件名称：《书卷香网上图书交易系统》

项目提出者：张泽亚

开发者：张泽亚、孙振昊

用户：网上书店管理者和购书者

## 1.3定义

MVC： M(Model)：JavaBean

V(View):JSP

C(Controller):Servelet

## 1.4参考资料

《软件需求说明书》（GB856T---88）

《软件需求工程 第二版》 毋国庆等编著 机械工业出版社

《软件工程导论 第六版》 张海藩著 清华大学出版社

# 2程序系统的结构

## 2.1系统功能结构

前台：

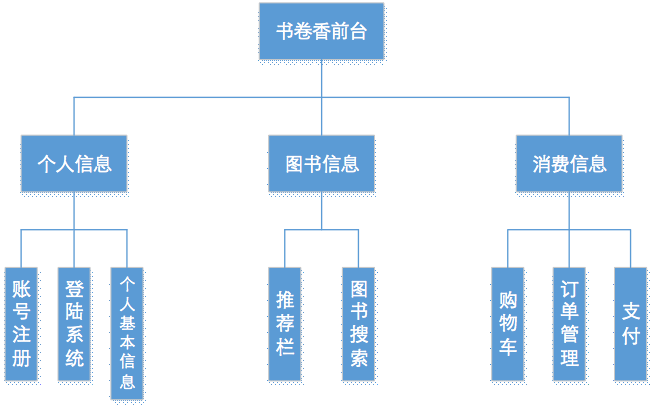


图1 前台功能结构图

后台：

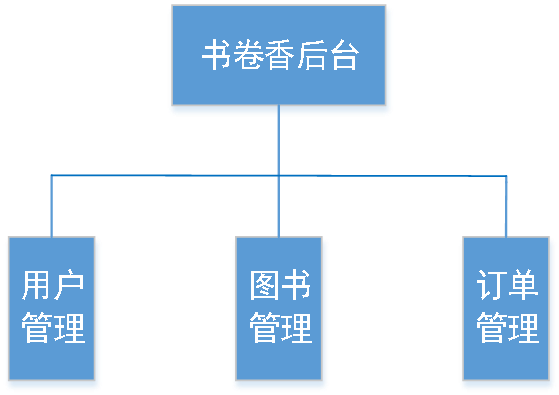


图2 后台功能结构图

## 2.2概述

### 模块划分：

|  |  |  |
| --- | --- | --- |
| 系统名称 | 模块 | 功能 |
| 书卷香 | 个人信息 | 登录、注册、个人信息管理 |
| 图书信息 | 推荐栏、图书搜素 |
| 消费信息 | 购物车、订单管理、支付 |
| 用户管理 | 对用户进行管理 |
| 图书管理 | 对图书信息的增删改 |
| 订单管理 | 查看、删除订单和确认发货 |

表 1 模块划分表

### 功能划分：

|  |  |  |
| --- | --- | --- |
| 系统名称 | 功能 |  |
| 书卷香 | 账号注册 | 游客输入个人信息创建账户 |
| 用户登录 | 输入邮箱和密码登录系统 |
| 个人信息 | 用户对个人信息的管理 |
| 推荐栏 | 推荐用户图书，用户可以直接点击对应图书进行购买 |
| 图书搜索 | 用户搜索自己想要购买的图书 |
| 购物车 | 存放用户想要购买的图书 |
| 订单 | 用户订单管理，提交、查看、修改、取消订单和确认收货 |
| 支付 | 用户对订单进行结账 |
| 用户管理 | 管理员对用户进行管理 |
| 图书管理 | 管理员对书城图书信息进行增删改 |

表 2 功能划分表

## 

## 2.3主要模块程序名

|  |  |
| --- | --- |
| **模块名** | **子程序名** |
| **个人信息** | // 用户注册  private void userRegiste(HttpServletRequest req, HttpServletResponse resp) {  // 用户登录  private void userLogin(HttpServletRequest req, HttpServletResponse resp) {  /验证用户是否存在，登录用  public User checkUser(String email, String password) throws Exception;  //验证邮箱是否已存在，注册用  public boolean checkNickname(String email) throws Exception;    //修改个人信息  public boolean reviseUser(String email, String nickname, String password, String address, String phone)throws Exception; |
| **图书信息** | //根据楼层查Book  public List<Book> queryBooksByFloor(String floor) throws Exception;  //根据一级目录id和二级目录id查图书  public Map<String,Object> queryBooksByCidAndSid(Integer cid, Integer sid,Integer currPage,Integer pageSize)throws Exception;  //根据图书id查图书  public Book queryBookById(Integer bookId)throws Exception;    //根据出版社查图书  public Map<String,Object> queryByPublish(String publish,Integer currPage,Integer pageSize)throws Exception;    //根据价格查图书  public Map<String,Object> queryByPrice(String price,Integer currPage,Integer pageSize)throws Exception;    //根据折扣查图书  public Map<String,Object> queryByDiscount(String discount,Integer currPage,Integer pageSize)throws Exception;  //查询全部书籍信息  public Map<String, Object> queryAll(Integer currPage, Integer pageSize)throws Exception;  //图书模糊查询  public Map<String, Object> bookSearch(String search,Integer currPage,Integer pageSize)throws Exception;  //get图书URL  public String getPictureUrl(String picture) throws Exception; |
| **消费信息** | //将对应图书的数量加一。ajax  private void addOne(HttpServletRequest req, HttpServletResponse resp) {  //将对应图书的数量减一。ajax  private void subOne(HttpServletRequest req, HttpServletResponse resp) {  //添加图书到购物车中  private void addCart(HttpServletRequest req, HttpServletResponse resp) {  //从购物车中删除图书  private void deleteCart(HttpServletRequest req, HttpServletResponse resp) {  //得到购物车中图书的总价  private double getCartSumPrice(List<Cart> cartList){  //修改订单状态为已完成  private void sureOrder(HttpServletRequest req, HttpServletResponse resp) {  //查询当前用户所有订单  private void queryAllOrder(HttpServletRequest req, HttpServletResponse resp) {  //添加订单  private void addOrder(HttpServletRequest req, HttpServletResponse resp)  等 |
| **用户管理** | //删除用户  Private void deleteUSER(HttpServletRequest req,HttpServletResponse resp )  //修改用户信息  Private void change USER(HttpServletRequest req,HttpServletResponse resp ) |
| **图书管理** | //增加图书  Private void addbook()  //修改图书  Private void changebook()  //删除图书  Private void deletebook() |
| **订单管理** | public List<Map<String,Object>> getOrders(int start, int end) throws Exception;  //订单计数  public int queryOrderCount() throws Exception;  //删除订单  public boolean deleteOrderByOrderid(Integer orderid) throws Exception;  //确认发货  public Order getOrderByOrderSeq(String orderSeq) throws Exception; |

表 3 模块程序名表

# 3个人信息（Personal information）设计说明

## 3.1程序描述

### 3.1.1账号注册

新用户打开网站后，点击新用户注册，获得一个新账户。

### 3.1.2用户登录

此功能用于用户登录，用户输入用户账号，密码登录系统，登录成功后跳转至主界面。登陆后方可购买图书。

### 3.1.3个人信息

在用户的个人中心，点击编辑个人资料，进入编辑个人信息界面。用户可以修改个人相应的信息。

## 3.2功能

### 

### 3.2.1账号注册

|  |  |
| --- | --- |
|  | 账号注册 |
| 被调用 | 用户登录 |
| 调用 | 无 |
| 输入 | 用户信息 |
| 处理 | 上传并保存用户信息，创建账户 |
| 输出 | 注册结果（成功或失败） |
| 局部数据元素 | 邮箱、手机号、密码、昵称 |
| 注释 | 新用户注册，所以输入信息均为局部数据元素 |

表 4 账号注册ipo表

### 3.2.2用户登陆

|  |  |
| --- | --- |
| 程序 | 用户登录 |
| 被调用 | 其他注册用户操作 |
| 调用 | 无 |
| 输入 | 用户邮箱、手机号、密码、验证码，若忘记密码则输入找回密码请求，并输入验证码与新密码 |
| 处理 | 核对用户信息，返回登录结果 |
| 输出 | 登录结果（成功或失败） |
| 局部数据元素 | 验证信息 |
| 注释 | 无 |

表 5 用户登录ipo表

### 3.2.3个人信息

|  |  |
| --- | --- |
| 程序 | 个人信息管理 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 用户修改项的信息 |
| 处理 | 系统保存更新后的信息 |
| 输出 | 显示更新后的个人资料 |
| 局部数据元素 | 验证信息 |
| 注释 | 密码以圆点方式显示 |

表 6 个人信息ipo表

## 3.3性能

### 3.3.1精度要求

1. 用户邮箱：不超过50字符
2. 昵称：不超过20字符
3. 用户密码：不超过20字符
4. 手机号：11字符
5. 地址：不超过100字符

### 3.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于2s
3. 数据的转换和传送时间；小于4s

## 3.4输入项

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 数据类型 | 有效范围 | 输入方式 | 输入来源 |
| 1 | 用户邮箱 | Email | 字符串 | 不超过50字符 | 用户键入 | 用户 |
| 2 | 昵称 | Nickname | 字符串 | 不超过20字符 | 用户键入 | 用户 |
| 3 | 用户密码 | UserPassword | 字符串 | 不超过20字符 | 用户键入 | 用户 |
| 4 | 地址 | Address | 字符串 | 不超过100字符 | 用户键入 | 用户 |
| 5 | 手机号 | Phone | 字符串 | 11字符 | 用户键入 | 用户 |

表 7 个人信息输入项表

## 3.5输出项

用户注册成功或失败、登陆成功或失败、修改信息成功或失败。

## 3.6算法

public class UserDaoImp implements UserDao{

@Override

public User checkUser(String email, String password) throws Exception {

String sql = "SELECT \* from t\_user where email=? AND password=?";

return JDBCUtils.queryForObject(sql,User.class,email,password);

}

@Override

public boolean checkNickname(String email) throws Exception {

String sql = "SELECT \* from t\_user where email=?";

User user = JDBCUtils.queryForObject(sql,User.class,email);

if(user==null) {

return false;

}

else return true;

}

@Override

public boolean registerUser(String email, String nickname, String password,String address,String date)

throws Exception {

String sql = "INSERT INTO t\_user(email,nickname,password,address,registerdate) VALUES (?,?,?,?,?)";

return JDBCUtils.update(sql,email,nickname,password,address,date);

}

@Override

public boolean reviseUser(String email, String nickname, String password, String address, String phone)

throws Exception {

String sql = "update t\_user set nickname=?,password=?,address=?,phone=? where email=?";

return JDBCUtils.update(sql,nickname,password,address,phone,email);

}

## 3.7流程逻辑

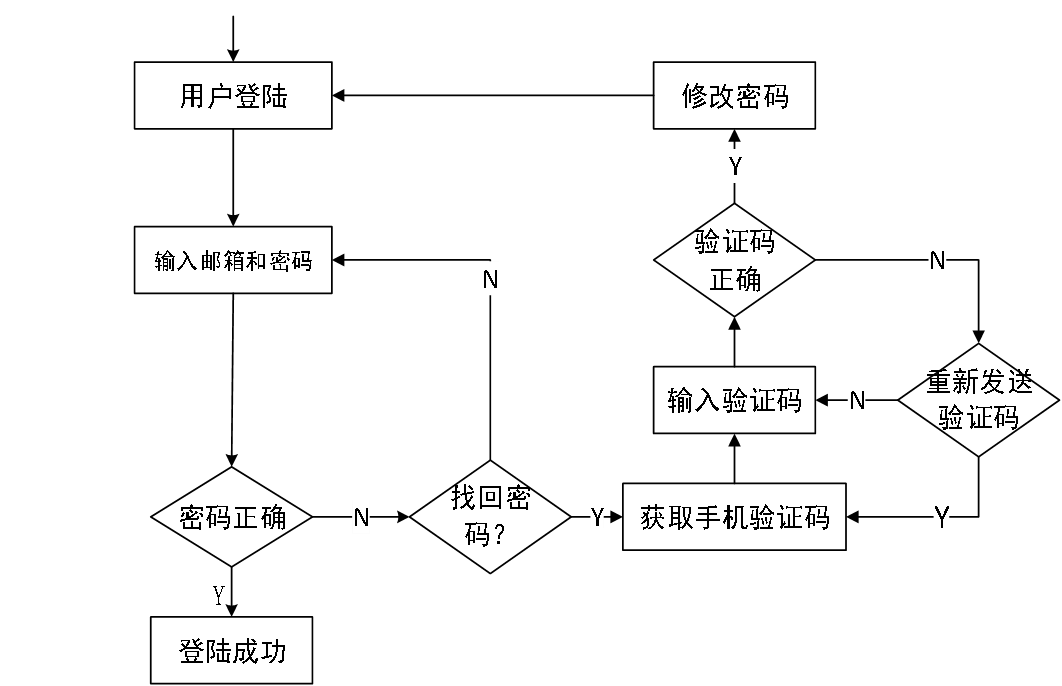
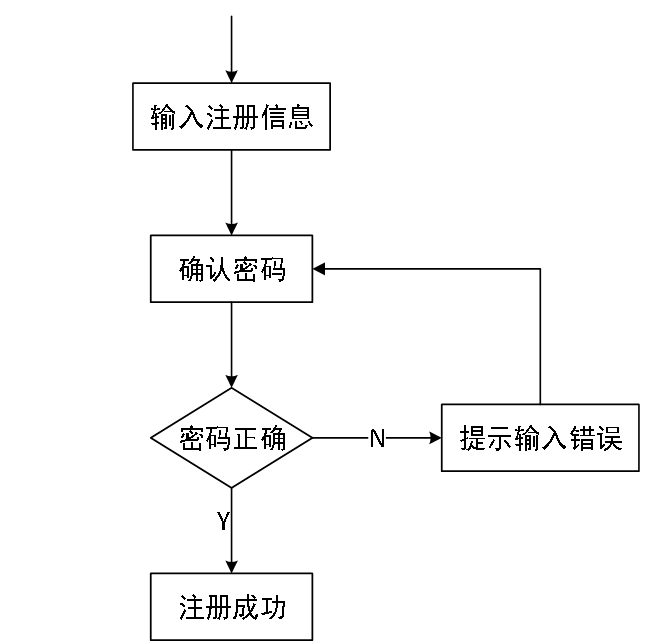


图3 用户登录流程逻辑图



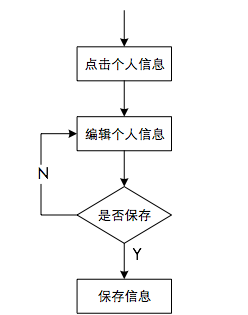


图4 用户注册流程逻辑图

## 3.8接口

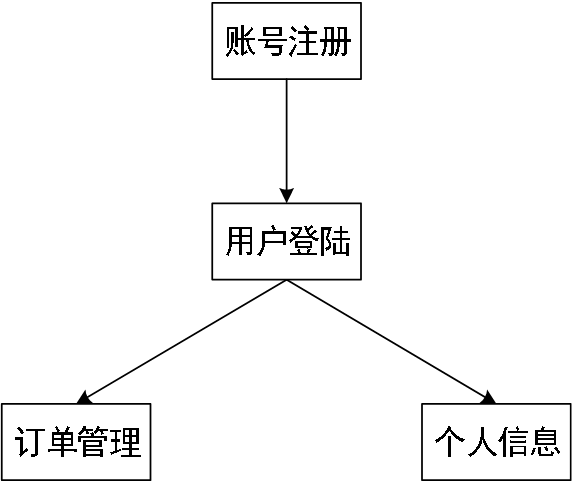


图5 个人信息模块相关接口图

## 3.9存储分配

无特殊存储分配

## 3.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 3.11限制条件

1. 输入项中的邮箱不能超过50位字符，且不能与数据库中已有邮箱重复。
2. 输入项中的密码不能超过20位字符。
3. 输入项中的手机号必须是11位字符，且不能与数据库中已有的手机号重复。
4. 输入项中的验证码必须是6位字符。
5. 输入项中的地址必须是100字符以内。

## 3.12测试计划

见测试计划说明书。

## 3.13尚未解决的问题

无。

# 4图书信息（Book information）设计说明

## 4.1程序描述

### 4.1.1推荐栏

用户打开网站会有一系列图书推荐给他们，供他们选择查看购买，有利于用户读书兴趣的扩展和书城书籍的畅销。

### 4.1.2图书搜索

用于用户对想要购买的图书进行搜索购买，可以搜索图书的名字、出版社、价格等。

## 4.2功能

### 4.2.1推荐栏

|  |  |
| --- | --- |
| 程序 | 推荐栏 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 无 |
| 处理 | 从数据库得到推荐图书信息 |
| 输出 | 推荐的图书信息 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 8 推荐栏ipo表

### 4.2.2图书搜索

|  |  |
| --- | --- |
| 程序 | 搜索图书 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 搜索信息（书名、作者、出版社、折扣、价格） |
| 处理 | 从数据库搜索具有相关信息的图书 |
| 输出 | 搜索到的图书 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 9 图书搜索ipo表

### 

## 4.3性能

### 4.3.1精度要求

搜索图书输入的信息不能超过100字符

### 4.3.2时间特性要求

（1）操作响应时间：小于0.5s

（2）更新处理时间：小于2s

（3）数据转换和传送时间：小于4s

## 4.4输入项

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 数据类型 | 有效范围 | 输入方式 | 输入来源 |
| 1 | 搜索信息 | search | 字符串 | 不超过50字符 | 用户键入 | 用户 |

表 10 图书信息输入项ipo表

## 4.5输出项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 类型及长度 | 输出方式 | 输出来源 |
| 1 | 图书名称 | BookName | varchar(50) | 页面显示 | 从数据库读取 |
| 2 | 作者 | Author | varchar(50) | 页面显示 | 从数据库读取 |
| 3 | 价格 | Price | double | 页面显示 | 从数据库读取 |
| 4 | 折扣 | Discount | int(11) | 页面显示 | 从数据库读取 |
| 5 | 出版社 | Publishing | varchar(100) | 页面显示 | 从数据库读取 |

表 11 图书信息输出项ipo表

## 4.6算法

public class BookDaoImp implements BookDao{

@Override

public List<Book> queryBooksByFloor(String floor) throws Exception{

String sql = "select \* from t\_book where floor=? order by bid desc limit 0,8";

return JDBCUtils.queryForList(sql, Book.class, floor);

}

@Override

public Map<String,Object> queryBooksByCidAndSid(Integer cid, Integer sid,Integer currPage,Integer pageSize) throws Exception {

String bookListSql = "SELECT \* from t\_book where categoryid=? and secondcategoryid=? order by bid desc limit ?,?";

String countSql = "SELECT count(\*) from t\_book where categoryid=? and secondcategoryid=?";

return SplitPage.split(bookListSql,countSql,currPage,pageSize,Book.class,cid,sid);

//JDBCUtils.queryForList(sql, Book.class, cid,sid);

}

……

## 4.7流程逻辑

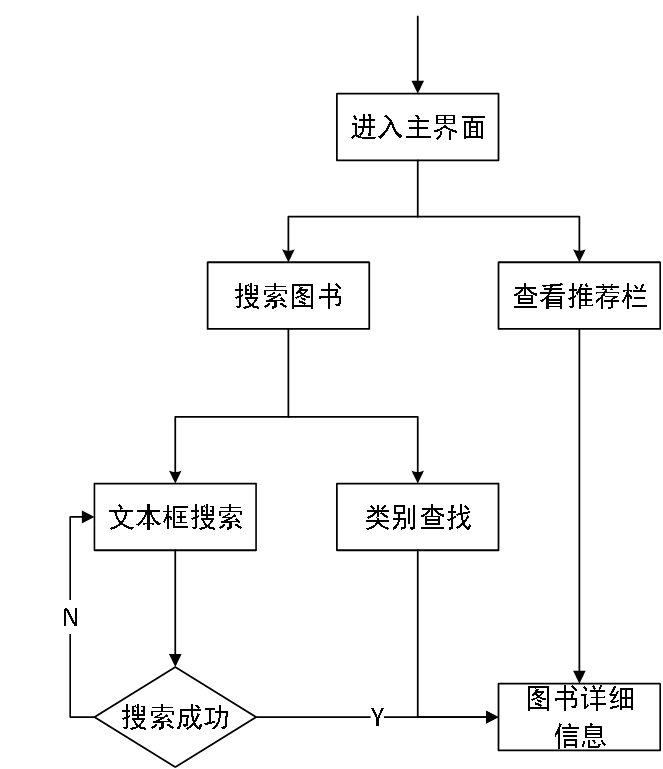


图6 图书信息流程逻辑图

## 4.8接口

该模块程序只与数据库进行交互，与其他模块程序无接口

## 4.9存储分配

无特殊存储分配。

## 4.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 4.11限制条件

搜索图书输入的信息不能超过100字符。

## 4.12测试计划

见测试计划说明书。

## 4.13尚未解决的问题

无

# 5消费信息（Consumer information）设计说明

## 5.1程序描述

### 5.1.1购物车

用于存放用户要购买的书籍，可以选择提交订单来对购物车中所有书籍结账购买。

### 5.1.2订单信息

用户查看历史订单信息，修改、取消订单和确认收货。

### 5.1.3支付

用户通过此程序向网站进行支付，为购买的图书付款。

## 5.2功能

### 5.2.1购物车

|  |  |
| --- | --- |
| 程序 | 购物车 |
| 被调用 | 订单信息 |
| 调用 | 无 |
| 输入 | 用户选择查看购物车，增加、删除商品 |
| 处理 | 显示购物车信息，添加商品到购物车、删除购物车里的商品 |
| 输出 | 操作成功或失败 |
| 局部数据元素 | 无 |
| 注释 | 购物车自动计算价格总和 |

表 12 购物车ipo表

### 5.2.2订单信息

|  |  |
| --- | --- |
| 程序 | 订单信息 |
| 被调用 | 购物车 |
| 调用 | 支付功能 |
| 输入 | 用户选择提交订单、查看订单、修改订单、取消订单、确认收货 |
| 处理 | 根据用户的请求对订单产生相应处理 |
| 输出 | 操作成功或失败 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 13 订单信息ipo表

### 5.2.3支付

|  |  |
| --- | --- |
| 程序 | 支付功能 |
| 被调用 | 订单管理 |
| 调用 | 无 |
| 输入 | 用户选择支付并确认付款 |
| 处理 | 申请银行进行转账服务 |
| 输出 | 支付成功或失败 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 14 支付ipo表

## 5.3性能

### 5.3.1精度要求

支付金额精确到小数点后两位

### 5.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于1s

## 5.4输入项

该模块输入都是通过用户点击按钮进行操作。

## 5.5输出项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 类型及长度 | 输出方式 | 输出来源 |
| 1 | 图书名称 | BookName | varchar(50) | 页面显示 | 从数据库读取 |
| 2 | 图书数量 | booknum | varchar(50) | 页面显示 | 计算得出 |
| 3 | 价格 | Price | double | 页面显示 | 计算得出 |

表 15 消费信息输出项表

还有订单处理成功或失败的信息，支付成功或失败的信息。

## 5.6算法

public boolean addOrder(Order order) throws Exception {

String sql = "insert into t\_order(orderseq,uid,num,date,total,status,address) VALUES(?,?,?,?,?,?,?)";

return JDBCUtils.update(sql, order.getOrderSeq(),order.getUid(),order.getNum(),order.getDate(),order.getTotal(),order.getStatus(),order.getAddress());

}

@Override

public List<Order> queryOrdersByUid(Integer uid) throws Exception {

String sql = "select \* from t\_order where uid=?";

return JDBCUtils.queryForList(sql, Order.class, uid);

…….

## 5.7流程逻辑

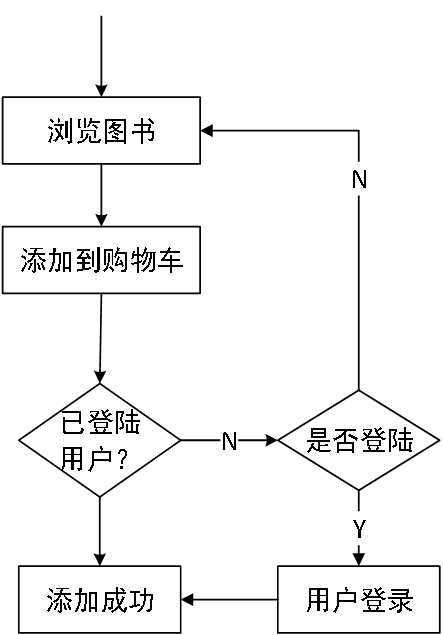


图7 购物车流程逻辑图1

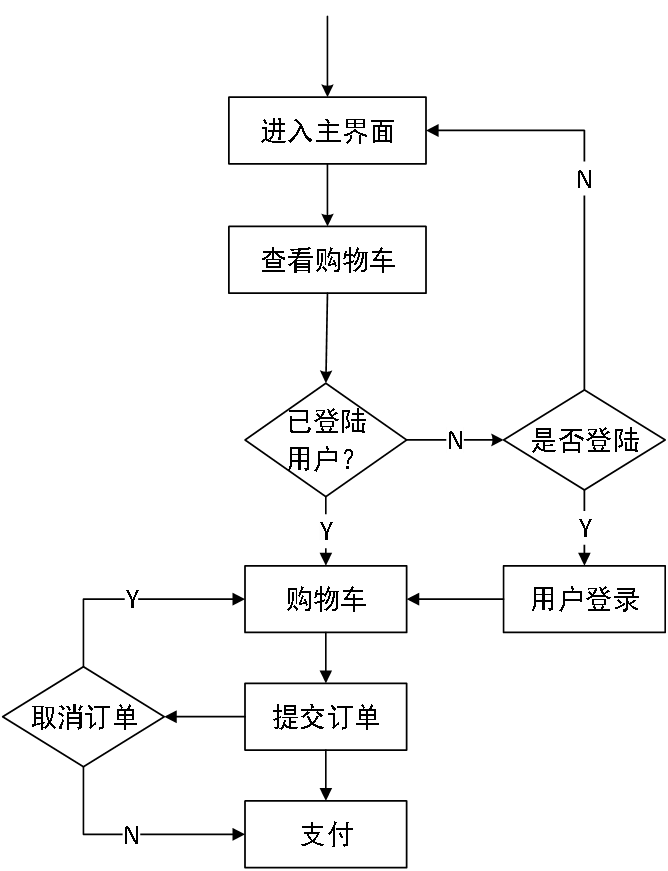


图8 购物车流程逻辑图2

## 5.8接口

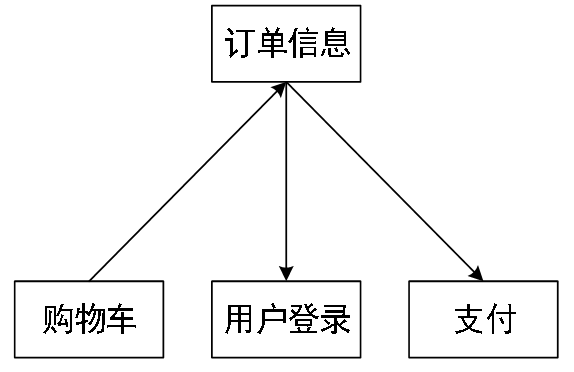


图9 消费信息接口图

## 5.9存储分配

本模块无特殊要求。

## 5.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 5.11限制条件

无

## 5.12测试计划

见测试计划说明书。

## 5.13尚未解决的问题

无。

# 6用户管理（User management）设计说明

## 6.1程序描述

管理员对用户进行管理，进行修改用户信息或删除用户。

## 6.2功能

|  |  |
| --- | --- |
| 模块 | 用户管理 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 管理员选择查看用户信息，修改用户信息并输入对应信息，删除用户 |
| 处理 | 查询用户信息，修改用户信息，删除用户 |
| 输出 | 显示更新后的用户信息 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 16 用户管理ipo表

## 6.3性能

### 6.3.1精度要求

精度无特殊要求

### 6.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于1s

## 6.4输入项

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 数据类型 | 有效范围 | 输入方式 | 输入来源 |
| 1 | 用户邮箱 | Email | 字符串 | 不超过50字符 | 管理员键入 | 管理员 |
| 2 | 昵称 | Nickname | 字符串 | 不超过20字符 | 管理员键入 | 管理员 |
| 3 | 用户密码 | UserPassword | 字符串 | 不超过20字符 | 管理员键入 | 管理员 |
| 4 | 地址 | Address | 字符串 | 不超过100字符 | 管理员键入 | 管理员 |
| 5 | 手机号 | Phone | 字符串 | 11字符 | 管理员键入 | 管理员 |

表 17 用户管理输入项表

## 6.5输出项

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 数据类型 | 有效范围 | 输入方式 | 输入来源 |
| 1 | 用户邮箱 | Email | 字符串 | 不超过50字符 | 管理员键入 | 管理员 |
| 2 | 昵称 | Nickname | 字符串 | 不超过20字符 | 管理员键入 | 管理员 |
| 3 | 用户密码 | UserPassword | 字符串 | 不超过20字符 | 管理员键入 | 管理员 |
| 4 | 地址 | Address | 字符串 | 不超过100字符 | 管理员键入 | 管理员 |
| 5 | 手机号 | Phone | 字符串 | 11字符 | 管理员键入 | 管理员 |

表 18 用户管理输出项ipo表

删除用户会输出删除成功或失败提示。

## 6.6算法

对数据库中的用户的相关信息做出对应的增删改操作。

## 6.7流程逻辑

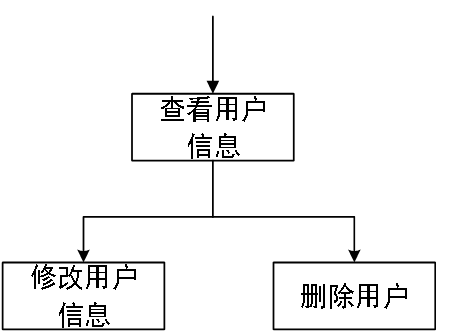


图10 用户管理流程逻辑图

## 6.8接口

该模块程序只与数据库进行交互，与其他模块程序无接口。

## 6.9存储分配

本模块无特殊要求。

## 6.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 6.11限制条件

无

## 6.12测试计划

见测试计划说明书。

## 6.13尚未解决的问题

无。

# 7图书管理（Book management）设计说明

## 7.1程序描述

管理员对图书进行管理，对图书信息进行增删改。

## 7.2功能

|  |  |
| --- | --- |
| 模块 | 图书管理-后台 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 管理员选择增加图书并录入图书信息(书名、作者、简介、价格等)，选择修改图书并输入信息，或选择删除图书 |
| 处理 | 增加对应图书，更改图书的信息，从数据库删除图书信息 |
| 输出 | 更新操作后的图书信息 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 19图书管理ipo表

## 7.3性能

### 7.3.1精度要求

图书价格保留到小数点后两位。

### 7.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于1s

## 7.4输入项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 类型及长度 | 输入方式 | 输入来源 |
| 1 | 图书名称 | BookName | varchar(50) | 键入 | 管理员 |
| 2 | 作者 | Author | varchar(50) | 键入 | 管理员 |
| 3 | 价格 | Price | double | 键入 | 管理员 |
| 4 | 折扣 | Discount | int(11) | 键入 | 管理员 |
| 5 | 出版社 | Publishing | varchar(100) | 键入 | 管理员 |

表 20 图书管理输入项表

## 7.5输出项

输出对应操作成功或失败。

## 7.6算法

对数据库中图书的相关信息做出对应的增删改操作

## 7.7流程逻辑

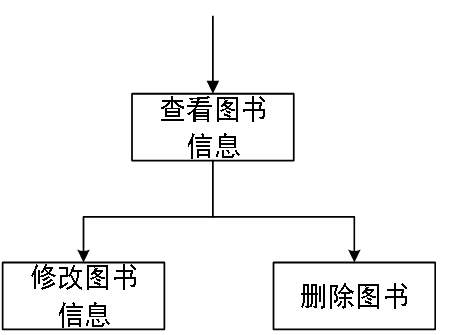


图11 图书管理流程逻辑图

## 

## 7.8接口

该模块程序只与数据库进行交互，与其他模块程序无接口。

## 7.9存储分配

本模块无特殊要求。

## 7.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 7.11限制条件

无

## 7.12测试计划

见测试计划说明书。

## 7.13尚未解决的问题

无。

# 8订单管理（Order management）设计说明

## 8.1程序描述

管理员对用户提交的订单进行查看，确认发货，删除订单等

**8.2功能**

|  |  |
| --- | --- |
| 模块 | 订单管理 |
| 被调用 | 无 |
| 调用 | 无 |
| 输入 | 管理员选择查看订单、确认发货、取消订单 |
| 处理 | 显示订单信息、将订单状态修改为已发货、将订单取消 |
| 输出 | 操作成功或失败 |
| 局部数据元素 | 无 |
| 注释 | 无 |

表 21订单管理ipo表

## 8.3性能

### 8.3.1精度要求

精度无特殊要求

### 8.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于1s

## 8.4输入项

该模块输入都是通过用户点击按钮进行操作。

## 8.5输出项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据名称 | 标识符 | 类型及长度 | 输出方式 | 输出来源 |
| 1 | 图书名称 | BookName | varchar(50) | 页面显示 | 从数据库读取 |
| 2 | 图书数量 | booknum | varchar(50) | 页面显示 | 计算得出 |
| 3 | 价格 | Price | double | 页面显示 | 计算得出 |

表 22 订单管理输出项表

## 8.6算法

public class OrderServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private OrderService orderService = BeanFactory.getBean(OrderServiceImp.class);

private OrderItemService orderItemService = BeanFactory.getBean(OrderItemServiceImp.class);

private BookService bookService = BeanFactory.getBean(BookServiceImp.class);

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

doPost(req,resp);

}

//修改订单状态为已完成

private void sureOrder(HttpServletRequest req, HttpServletResponse resp) {

try {

## 8.7流程逻辑

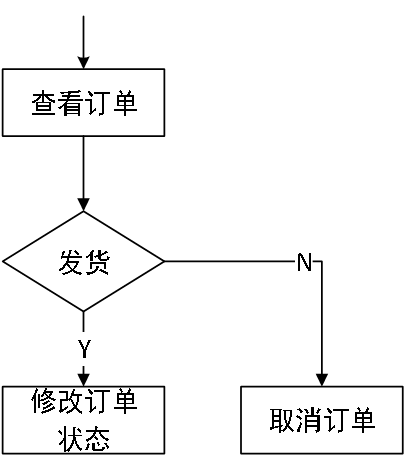


图12 订单管理流程逻辑图

## 8.8接口

该模块程序只与数据库进行交互，与其他模块程序无接口。

## 8.9存储分配

无特殊存储分配。

## 8.10注释设计

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等进行注释；
4. 对使用的逻辑进行注释。

## 8.11限制条件

搜索图书输入的信息不能超过100字符。

## 8.12测试计划

见测试计划说明书。

## 8.13尚未解决的问题

无

1. [↑](#footnote-ref-0)