

# 天津大学



## 网上书店管理及交易系统

### 软件测试报告

#### 5.0 版本

|         |            |
|---------|------------|
| 专    业: | 软件工程       |
| 年    级: | 2018 级     |
| 组长姓名:   | 郝煦泽        |
| 组长学号:   | 3018216039 |
| 组员姓名:   | 白文杰        |
| 组员学号:   | 3018216031 |
| 组员姓名:   | 董浩         |
| 组员学号:   | 3018216037 |
| 授课教师:   | 李罡         |

# 软件测试报告

## 目录

|                         |    |
|-------------------------|----|
| 软件测试报告.....             | 1  |
| 1. 引言.....              | 1  |
| 1.1 说明.....             | 1  |
| 1.2 术语或缩略语.....         | 1  |
| 1.3 参考资料.....           | 1  |
| 2. 软件配置.....            | 1  |
| 3. 测试准备.....            | 2  |
| 4. 测试阶段.....            | 3  |
| 4.1 单元测试.....           | 3  |
| 1) 登入模块测试.....          | 3  |
| 2) 用户账号管理模块测试.....      | 4  |
| 3) 用户订单管理模块测试.....      | 6  |
| 4) 管理员管理模块测试.....       | 9  |
| 4.2 组装测试.....           | 12 |
| 1) 增加用户登入模块.....        | 13 |
| 2) 增加注册模块和注销模块.....     | 14 |
| 3) 增加订购管理模块和撤销管理模块..... | 15 |
| 4) 增加后台供应商管理图书信息模块..... | 16 |
| 5) 增加后台管理员管理模块.....     | 17 |
| 4.3 确认测试.....           | 18 |
| 4.3.1 测试方法.....         | 18 |
| 4.3.2 测试计划.....         | 18 |
| 4.3.3 测试过程.....         | 18 |
| 4.4 性能需求测试.....         | 19 |
| 4.4.1 数据精确度.....        | 19 |
| 4.4.2 容错要求.....         | 19 |
| 4.4.3 精度灵活性.....        | 20 |
| 4.4.4 系统安全性要求.....      | 20 |
| 4.4.5 故障处理要求.....       | 20 |
| 4.4.6 时间限制.....         | 20 |
| 4.4.7 其他性能要求.....       | 20 |
| 4.5 系统测试.....           | 21 |
| 4.5.1 恢复测试.....         | 21 |
| 4.5.2 安全测试.....         | 21 |
| 4.5.3 强度与性能测试.....      | 21 |

# 1. 引言

## 1.1 说明

该系统名为网上书店交易及管理系统,面向所有用户进行网上图书的订购操作。此份软件测试报告,运用流程图、流图、使用基本测试路径等方法从各层对系统进行测试。尽可能地暴露软件中的错误,并予以修改纠正和补充,完善软件功能。

## 1.2 术语或缩略语

无

## 1.3 参考资料

【1】《软件工程-实践者的研究方法》 第八版 (美) Roger S. Pressman 著 机械工业出版社

【2】软件工程导论 第六版 张海藩 牟永敏

# 2. 软件配置

- 1) 软件需求说明书: 需求分析报告
- 2) 软件设计说明书: 软件设计报告
- 3) 源代码

### 3. 测试准备

测试计划：见测试阶段中

测试工具：自动测试工具等

测试人员：网上书店项目组小组全体成员。

测试目的：尽可能地暴露软件中的错误，并予以修改纠正和补充。

其他方面准备：

系统的运行条件准备软件测试环境，负责测试的人员需要对测试环境进行确认。

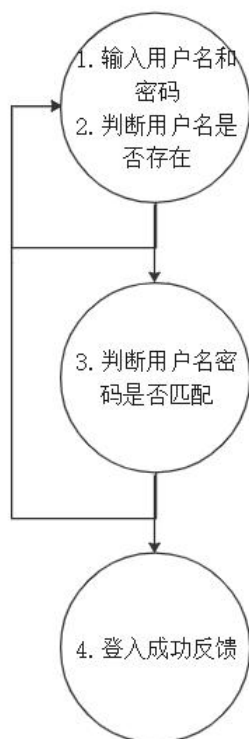
首先，确认计算机软件、硬件、网络环境正常运行，并已经满足的要求。其次，要消除病毒干扰。使用杀毒软件对测试环境进行病毒检测和杀毒，对被测软件进行检测和杀毒。在上述任何情况不能正常推进时，测试人员应对测试环境进行维护。维护工作完成后，测试人员须对环境进行再次检测和确认，才能进行测试。

## 4. 测试阶段

### 4.1 单元测试

#### 1) 登入模块测试

流图：



测试用例：

| 用户     | 用户名      | 密码          |
|--------|----------|-------------|
| 普通用户 1 | 一个存在用户名  | 能匹配存在账号的密码  |
| 普通用户 2 | 一个存在用户名  | 不能匹配存在账号的密码 |
| 普通用户 3 | 一个不存在用户名 | 自定义密码       |

## 测试路径及其测试期望结果：

路径一：1-2-3-4

测试用例：普通用户 1

期望结果：用户成功登入进系统

路径二：1-2

测试用例：普通用户 2

期望结果：用户登入系统失败，提示账号密码不匹配，然后可以重新进行输入

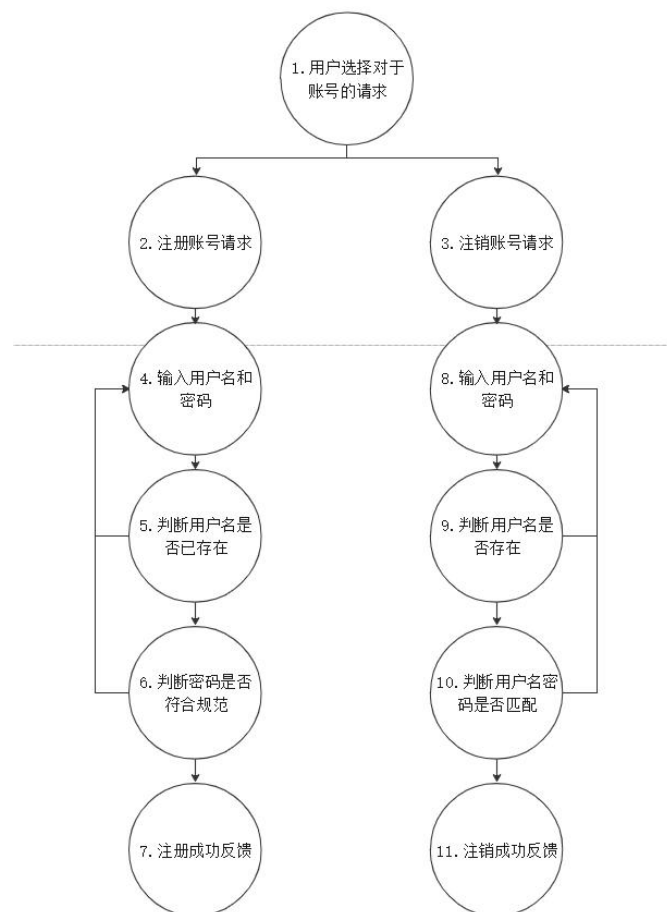
路径三：1-2-3

测试用例：普通用户 3

期望结果：用户登入系统失败，提示用户名不存在，然后可以重新进行输入

## 2) 用户账号管理模块测试

流图：



**测试用例：**

| 用户     | 请求类型 | 账号       | 密码          |
|--------|------|----------|-------------|
| 普通用户 1 | 注册账号 | 一个未使用用户名 | 一个符合规定的密码   |
| 普通用户 2 | 注册账号 | 一个未使用用户名 | 一个不符合规定的密码  |
| 普通用户 3 | 注册账号 | 一个已使用用户名 | 自定义密码       |
| 普通用户 4 | 注销账号 | 一个存在用户名  | 能匹配存在账号的密码  |
| 普通用户 5 | 注销账号 | 一个存在用户名  | 不能匹配存在账号的密码 |
| 普通用户 6 | 注销账号 | 一个不存在用户名 | 自定义密码       |

**测试路径及其测试期望结果：**

路经一：1-2-4-5-6-7

测试用例：普通用户 1

期望结果：用户成功注册账号

路经二：1-2-4-5-6

测试用例：普通用户 2

期望结果：用户注册失败，提示账号密码不符合规定，然后可以重新进行输入

路经三：1-2-4-5

测试用例：普通用户 3

期望结果：用户注册失败，提示用户名不可用，然后可以重新进行输入

路经四：1-3-8-9-10-11

测试用例：普通用户 4

期望结果：用户成功注销账号

路经五：1-3-8-9-10

测试用例：普通用户 5

期望结果：用户注销失败，提示账号密码不匹配，然后可以重新进行输入

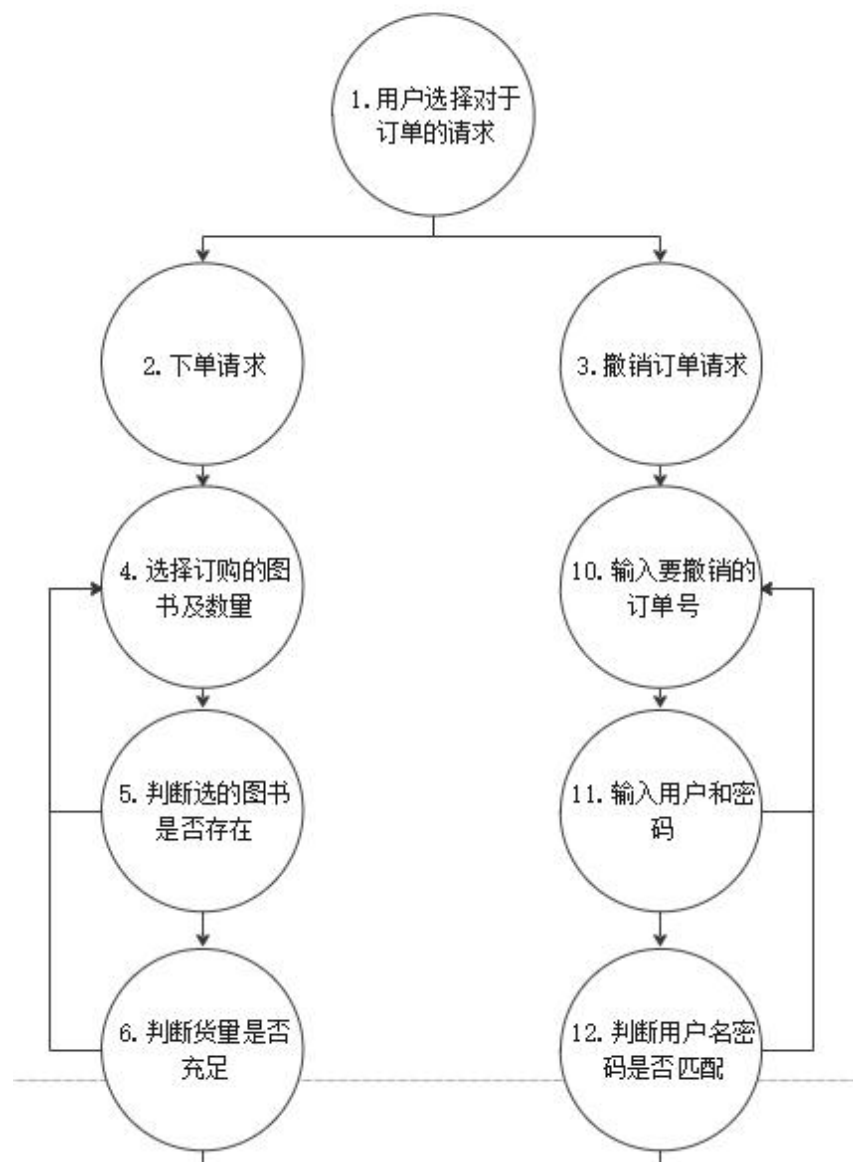
路经六：1-3-8-9

测试用例：普通用户 6

期望结果：用户注销失败，提示用户名不存在，然后可以重新进行输入

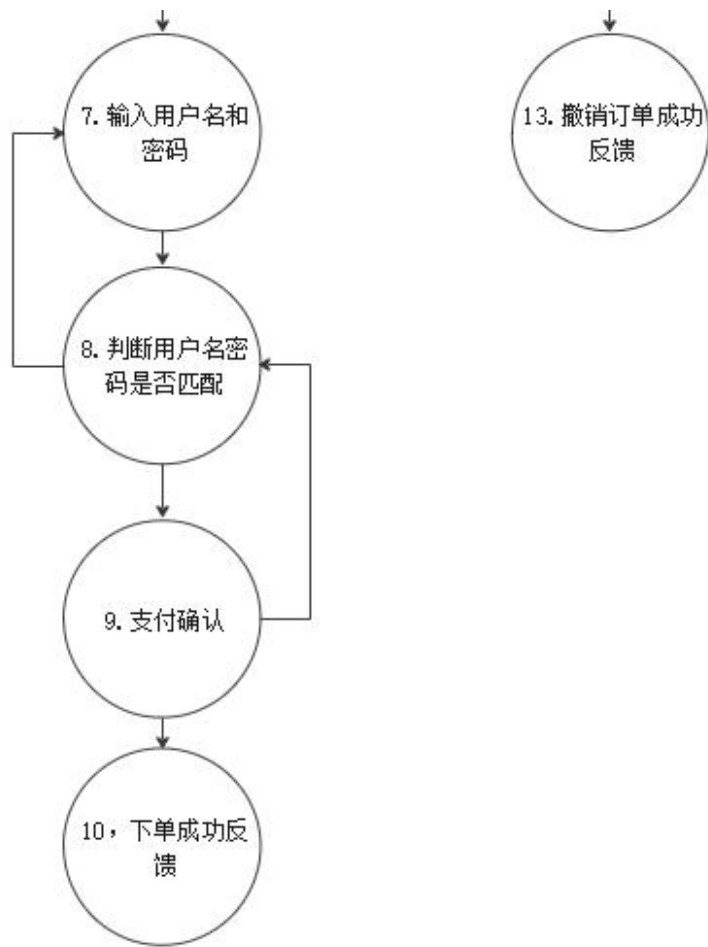
### 3) 用户订单管理模块测试

流图：



流图接下页





**测试用例：**

| 用户     | 请求类型 | 订购图书  | 订购图书个数 | 订单号 | 账号密码匹配情况 | 支付情况 |
|--------|------|-------|--------|-----|----------|------|
| 普通用户 1 | 下单   | 图书存在  | 数量充足   | 无   | 匹配       | 成功   |
| 普通用户 2 | 下单   | 图书不存在 | 自定义    | 无   | 自定义      | 无    |
| 普通用户 3 | 下单   | 图书存在  | 数量不足   | 无   | 自定义      | 无    |
| 普通用户 4 | 下单   | 图书存在  | 数量充足   | 无   | 不匹配      | 无    |
| 普通用户 5 | 下单   | 图书存在  | 数量充足   | 无   | 匹配       | 失败   |
| 普通用户 6 | 撤销订单 | 无     | 无      | 存在  | 匹配       | 无    |
| 普通用户 7 | 撤销订单 | 无     | 无      | 不存在 | 自定义      | 无    |
| 普通用户 8 | 撤销订单 | 无     | 无      | 存在  | 不匹配      | 无    |

**测试路径及其测试期望结果：**

路经一：1-2-4-5-6-7-8-9-10

测试用例：普通用户 1

期望结果：用户下单成功

路经二：1-2-4-5

测试用例：普通用户 2

期望结果：用户下单失败，提示图书不存在，然后可以重新进行输入

路经三：1-2-3-4-5-6

测试用例：普通用户 3

期望结果：用户下单失败，提示库存不足，然后可以重新进行输入

路经四：1-2-4-5-6-7-8

测试用例：普通用户 4

期望结果：用户下单失败，提示账号密码不匹配，然后可以重新进行输入

路经五：1-2-4-5-6-7-8-9

测试用例：普通用户 5

期望结果：用户下单失败，提示支付失败，然后可以重新进行输入

路经六：1-3-10-11-12-13

测试用例：普通用户 6

期望结果：用户撤销订单成功

路经七：1-3-10-11

测试用例：普通用户 7

期望结果：用户撤销订单失败，提示订单号不存在，然后可以重新进行输入

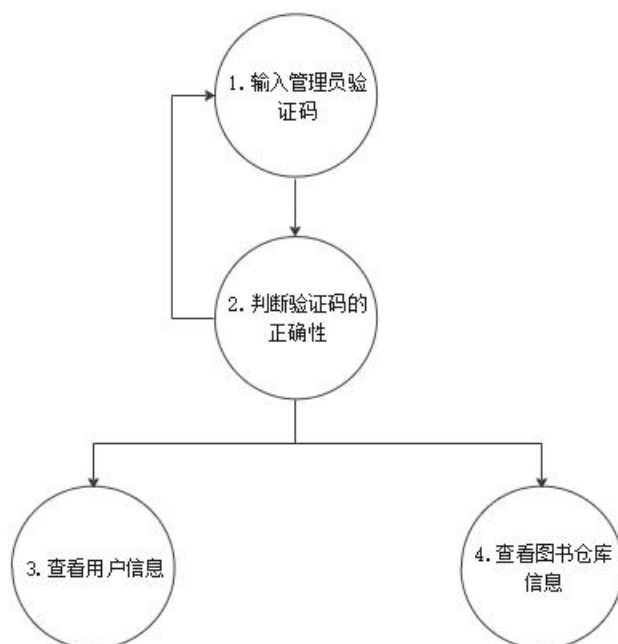
路经八：1-3-10-11-12

测试用例：普通用户 8

期望结果：用户下单失败，提示用户名密码不匹配，然后可以重新进行输入

#### 4) 管理员管理模块测试

流图：



数据用例：

| 用户    | 验证码  |
|-------|------|
| 管理员 1 | 可识别  |
| 管理员 2 | 不可识别 |

测试路径及其测试期望结果：

路径一：1-2-3

测试用例：管理员 1

期望结果：管理员登入成功并可查看用户信息

路径二：1-2-4

测试用例：管理员 1

期望结果：管理员登入成功并可查看图书仓库信息

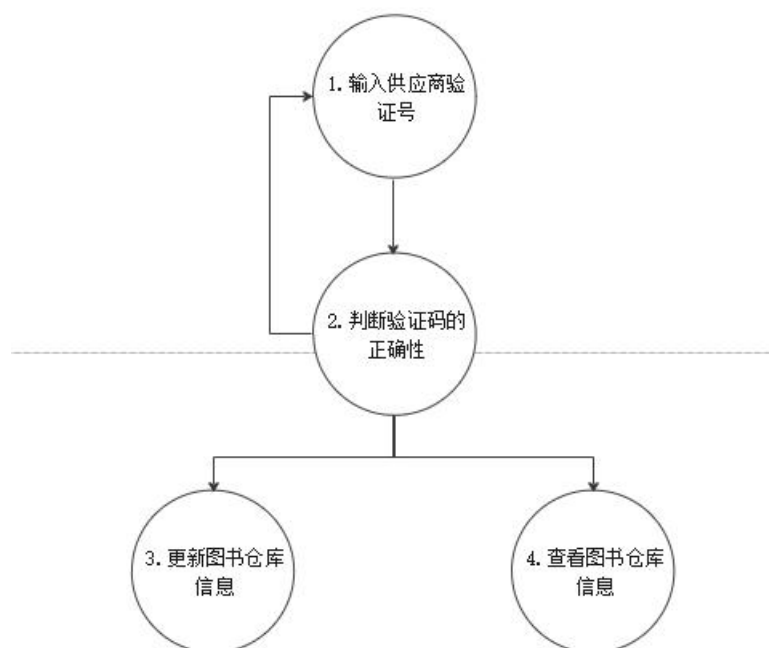
路径二：1-2

测试用例：管理员 2

期望结果：管理员身份验证失败，可重新输入验证码

## 5) 供应商供应信息模块测试

流图：



**数据用例：**

| 用户    | 验证码  |
|-------|------|
| 供应商 1 | 可识别  |
| 供应商 2 | 不可识别 |

**测试路径及其测试期望结果：**

路经一：1-2-3

测试用例：供应商 1

期望结果：供应商登入成功并可更新图书仓库信息

路经二：1-2-4

测试用例：供应商 1

期望结果：供应商登入成功并可查看图书仓库信息

路经二：1-2

测试用例：供应商 2

期望结果：供应商身份验证失败，可重新输入验证码

## 4.2 组装测试

采取自顶向下的集成方式，从主控模块开始，按照软件的控制层次结构，以深度优先策略，逐步把各个模块集中起来。

按设计要求把通过单元测试的各个模块组装在一起，测试各个模块之间的连接组合功能是否按照需求完成，以便发现与接口有关的各种错误。

步骤：

（1）以主控模块作为测试驱动模块，把对主控模块进行单元测试时引入的所有桩模块用实际模块替代。

（2）依据深度优先的集成策略,每次只替代一个桩模块。

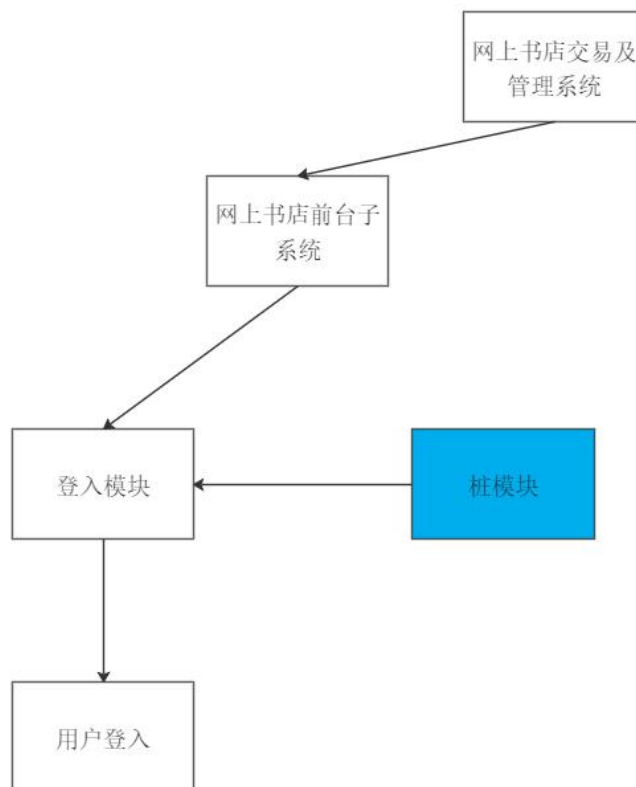
（3）每集成一个模块立即测试一遍。

（4）每组测试完成后，着手替换下一个桩模块。

（5）为避免引入新错误,须不断进行回归测试（即全部或部分地重复已做过的测试）。

（6）从第二步开始，循环执行上述步骤，直至整个程序结构构造完毕。

## 1) 增加用户登入模块

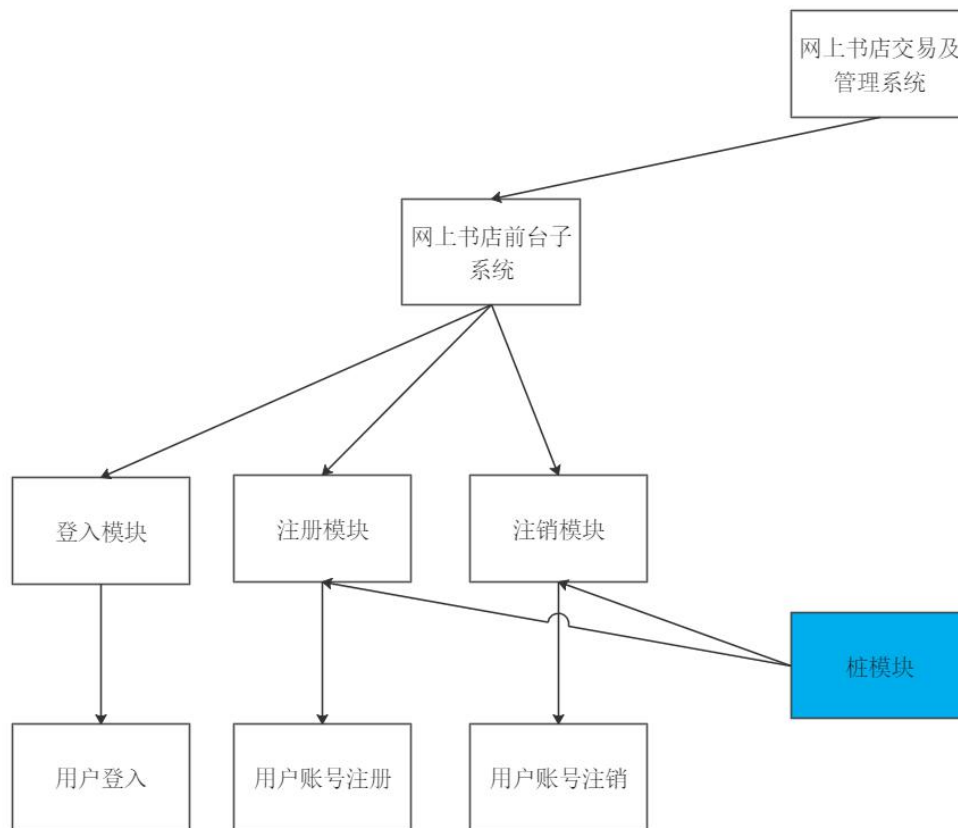
**桩模块功能:**

满足网上书店前台子系统的普通用户登入需求。

**数据用例及期望结果:**

| 用户     | 用户名      | 密码          | 期望结果              |
|--------|----------|-------------|-------------------|
| 普通用户 1 | 一个存在用户名  | 能匹配存在账号的密码  | 登入成功              |
| 普通用户 2 | 一个存在用户名  | 不能匹配存在账号的密码 | 登入失败并给出账号密码不匹配的提示 |
| 普通用户 3 | 一个不存在用户名 | 自定义密码       | 登入失败并给出用户不存在提示    |

## 2) 增加注册模块和注销模块



### 桩模块功能:

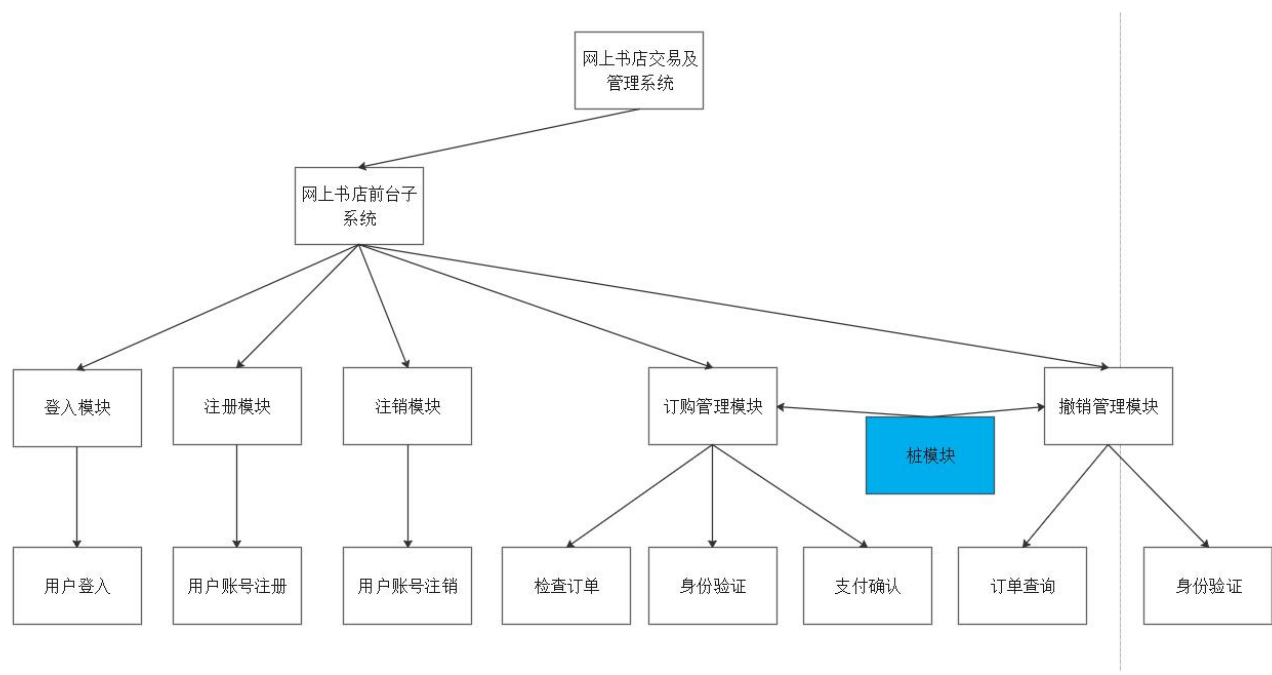
满足网上书店前台子系统的普通用户的账号管理（注销和注册）需求。

### 数据用例及期望结果:

| 用户     | 请求类型 | 账号       | 密码          | 期望结果 |
|--------|------|----------|-------------|------|
| 普通用户 1 | 注册账号 | 一个未使用用户名 | 一个符合规定的密码   | 注册成功 |
| 普通用户 2 | 注册账号 | 一个未使用用户名 | 一个不符合规定的密码  | 注册失败 |
| 普通用户 3 | 注册账号 | 一个已使用用户名 | 自定义密码       | 注册失败 |
| 普通用户 4 | 注销账号 | 一个存在用户名  | 能匹配存在账号的密码  | 注销成功 |
| 普通用户 5 | 注销账号 | 一个存在用户名  | 不能匹配存在账号的密码 | 注销失败 |
| 普通用户 6 | 注销账号 | 一个不存在用户名 | 自定义密码       | 注销失败 |



3) 增加订购管理模块和撤销管理模块



桩模块功能:

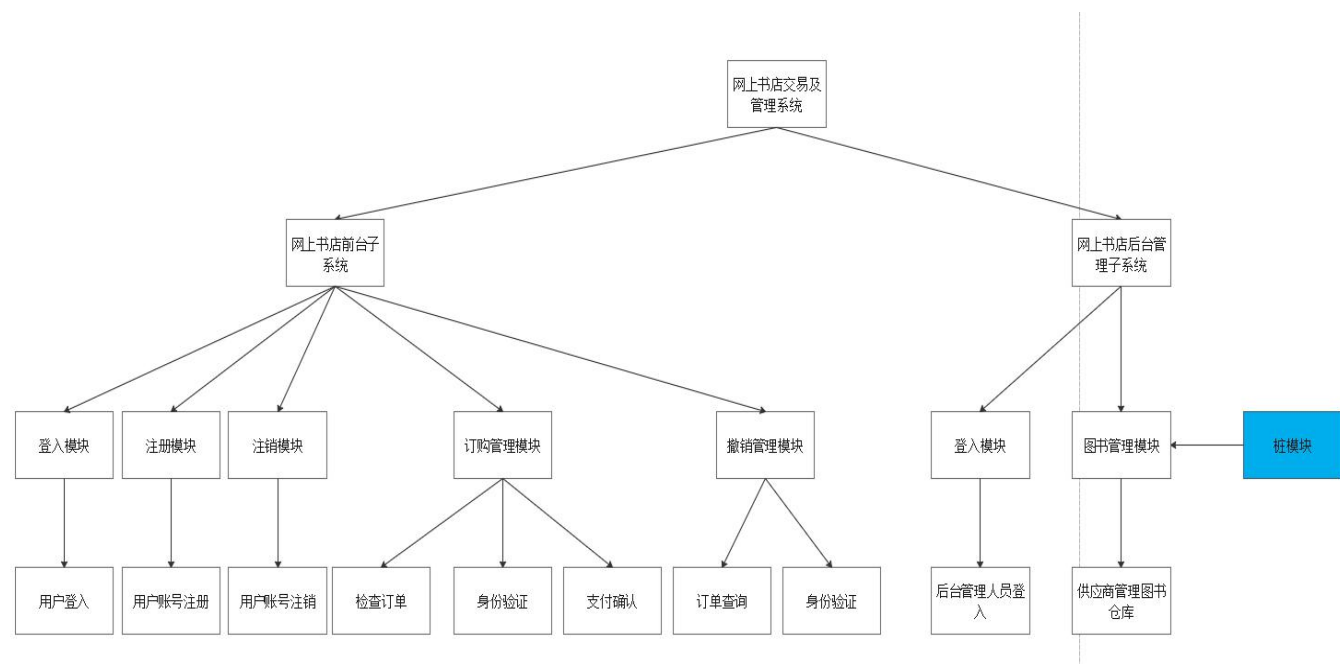
满足网上书店前台子系统的普通用户的订单管理（下单和撤销订单）需求。

数据用例及期望结果:

| 用户    | 请求类型 | 订购图书  | 订购图书个数 | 订单号 | 账号密码匹配情况 | 支付情况 | 期望结果           |
|-------|------|-------|--------|-----|----------|------|----------------|
| 普通用户1 | 下单   | 图书存在  | 数量充足   | 无   | 匹配       | 成功   | 订购成功           |
| 普通用户2 | 下单   | 图书不存在 | 自定义    | 无   | 自定义      | 无    | 订购失败并提示图书名错误   |
| 普通用户3 | 下单   | 图书存在  | 数量不足   | 无   | 自定义      | 无    | 订购失败并提示数量不符合要求 |
| 普通用户4 | 下单   | 图书存在  | 数量充足   | 无   | 不匹配      | 无    | 订购失败并提示用户匹配错误  |

| 用户    | 请求类型 | 订购图书 | 订购图书个数 | 订单号 | 账号密码匹配情况 | 支付情况 | 期望结果            |
|-------|------|------|--------|-----|----------|------|-----------------|
| 普通用户5 | 下单   | 图书存在 | 数量充足   | 无   | 匹配       | 失败   | 订购失败并提示支付错误     |
| 普通用户6 | 撤销订单 | 无    | 无      | 存在  | 匹配       | 无    | 撤销订单成功          |
| 普通用户7 | 撤销订单 | 无    | 无      | 不存在 | 自定义      | 无    | 撤销订单失败并提示订单号不存在 |
| 普通用户8 | 撤销订单 | 无    | 无      | 存在  | 不匹配      | 无    | 撤销订单失败并提示用户匹配错误 |

4) 增加后台供应商管理图书信息模块



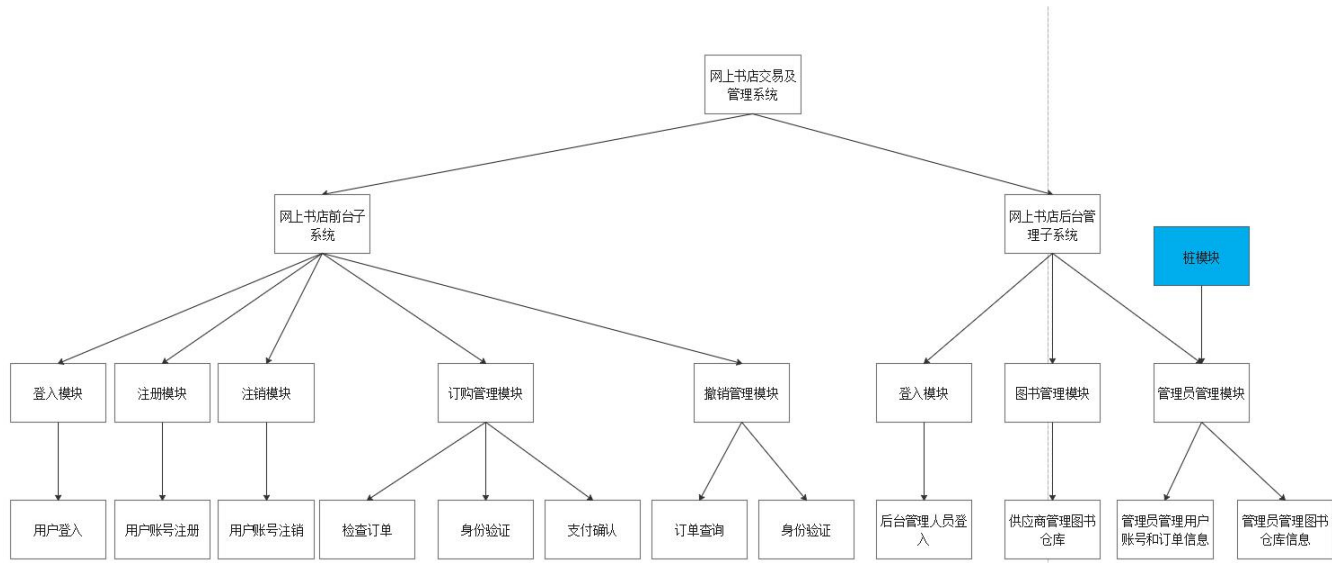
桩模块功能:

满足网上书店后台子系统的供应商的验证登入管理图书仓库信息的需求。

数据用例及期望结果:

| 用户    | 验证码  | 期望结果 |
|-------|------|------|
| 供应商 1 | 可识别  | 测试成功 |
| 供应商 2 | 不可识别 | 测试失败 |

5) 增加后台管理员管理模块



桩模块功能:

满足网上书店后台子系统的供应商的验证登入管理图书仓库信息的需求。

数据用例及期望结果:

| 用户    | 验证码  | 期望结果 |
|-------|------|------|
| 管理员 1 | 可识别  | 测试成功 |
| 管理员 2 | 不可识别 | 测试失败 |

## 4.3 确认测试

### 4.3.1 测试方法

黑盒测试

### 4.3.2 测试计划

有效性测试是在模拟的环境（可能是就是开发的环境）下，运用黑盒测试的方法：验证所测证性是不满足需求规格说明书列出的需求、首先制定测试计划，规定要做测试的种类，还需要制定一组测试步骤，描述具体的测试用例。通过实施预定的测试计划和测试步骤，确定软件的特性是否与需求相符，确保所有的软件功能需求都能得到满足，所有的软件性能需求能达到，所有的文档都是正确且便于使用。

### 4.3.3 测试过程

此处挑选系统重要流程进行确认测试

#### 1) 针对用户订单管理的流程进行确认测试

| 步骤 | 测试人员 | 执行动作         | 预期结果       |
|----|------|--------------|------------|
| 1  | 用户   | 输入匹配的用户名密码登入 | 用户登入成功     |
| 2  | 用户   | 选购图书并完成订单    | 用户订购成功     |
| 3  | 管理员  | 管理顾客资料       | 能够看到订单信息   |
| 4  | 管理员  | 管理图书仓库       | 更改图书的数目    |
| 5  | 供应商  | 管理订单信息       | 进行图书发货     |
| 6  | 用户   | 查看用户订单       | 订单变为已经受理状态 |

用户撤销订单操作同理

#### 2) 针对用户账号管理的流程进行确认测试

| 步骤 | 测试人员 | 执行动作        | 预期结果 |
|----|------|-------------|------|
| 1  | 用户   | 输入自定义的用户名密码 | 等待验证 |

| 步骤 | 测试人员 | 执行动作            | 预期结果             |
|----|------|-----------------|------------------|
| 2  | 管理员  | 查看用户的注册信息是否符合规定 | 验证反馈             |
| 3  | 管理员  | 管理顾客资料          | 将正确注册的账号添加到顾客资料中 |
| 4  | 用户   | 等待验证            | 注册结果反馈           |

用户注销账号操作同理

### 3) 针对供应商管理图书信息进行确认测试

| 步骤 | 测试人员 | 执行动作       | 预期结果             |
|----|------|------------|------------------|
| 1  | 用户   | 查看网上书店图书信息 | 页面反馈出所有可购书籍      |
| 2  | 供应商  | 管理图书仓库     | 提交上下架图书信息更新的请求   |
| 3  | 管理员  | 管理图书仓库     | 将正确注册的账号添加到顾客资料中 |
| 4  | 用户   | 刷新页面       | 页面反馈出更新后的可购书籍    |

## 4.4 性能需求测试

### 4.4.1 数据精确度

对于查询为了保证查全率，尽量让所有相应域包含查询关键字的记录都应能查到。由于通常有文件的记录会很多，所以本系统采用两种方法进行查询：直接查询和模糊查询

### 4.4.2 容错要求

为了保障系统站能正常运行，尽量使得服务器数据库的数据合法化，以及前台能有效操作和浏览，该系统应做到：

① 对数据输入或条件输入进行有效检验，禁止非合法数据阻碍客户端与服务器的正常运行，使得数据库也得到进一步安全性的保障。

② 对连接数据库的错误操作或用户错误操作进行一定化的提示和警告，控制非正常浏览网站。

#### 4.4.3 精度灵活性

- ① 当执行查询时，查询完后提供返回按钮可继续查找；
- ② 用户处于一定时间的“不操作”，服务器可不再响应其对数据库的请求操作，除非再次登陆。

#### 4.4.4 系统安全性要求

- ① 数据库连接采用 MSSQL 方法进行连接，从而提高数据库的可操作性和安全性；
- ② 对某些页面进行强制用户登陆后方能进行相关操作，避免用户直接在地址栏输入找到。

#### 4.4.5 故障处理要求

软、硬件故障：

- a) 用户所使用电脑的系统有问题，或者浏览器有问题。
- b) 数据库有问题也无法正常使用
- c) 内存太小会影响运行速度
- d) 对于各类突发事件的要注意避免
- e) 对于电脑的存储量要随时关注，以免出现存储量不足的情况

#### 4.4.6 时间限制

网页响应时间应保持在 1 秒以内，更新处理时间应保持在 3 秒内，网站服务器运行应尽量保持永不宕机。

#### 4.4.7 其他性能要求

保密性：

系统数据要做好保密设施，以免重要数据的丢失或者盗取。

简洁性：

用户的操作界面必须做到简单明了，不能有太复杂的设置或者操作。

可更新性：

对于系统可以根据客户需要进行更新，维护等操作，扩展其功能，并使系统稳定。

可转换性：

只要系统能保证环境部署成功，并且网络正常，就能保证系统的正常运行。

## 4.5 系统测试

### 4.5.1 恢复测试

采用各种办法使系统发生以下故障：

- （1）软件在使用过程中所使用的计算机突然断电或者计算机系统突然崩溃
- （2）软件出现不可预测的 BUG
- （3）软件使用过程网络连接突然断开

基于前两点的故障，检测软件是否能够实时保护数据并存储，做到数据不因故障而丢失

对于第三点故障，检测软件是否能检测到异常的连接断开并能尝试重连或给出错误信息。

### 4.5.2 安全测试

检查系统对非法侵入的防范能力。测试人员假扮非法入侵者，采用各种办法试图突破防线。验证是否对账户信息进行经过加密后传输与存储，保证安全性的设计，使非法侵入的代价超过信息价值。

### 4.5.3 强度与性能测试

迫使系统在异常的资源配置下运行，检查系统对异常情况的抵抗能力。对高并发等系统压力较大的情况，保证系统能够正常运行。