**图书网站**

**—— 图书网站开发文档**

**1.0**

**目录**

[1 系统概述 7](#_Toc20329085)

[1.1 构建目的 7](#_Toc20329086)

[1.2 需要完成的软件和文档 7](#_Toc20329087)

[1.3 项目的开发环境： 7](#_Toc20329088)

[2 系统需求分析 9](#_Toc20329090)

[3 数据设计 9](#_Toc20329091)

[3.1 相关数据库表说明如下： 9](#_Toc20329092)

[3.2 各表详细说明 9](#_Toc20329093)

[4 系统功能设计 12](#_Toc20329094)

[4.1 用户账号管理 12](#_Toc20329095)

[4.1.1 用户注册 12](#_Toc20329096)

[4.1.2 用户登入 13](#_Toc20329097)

[4.1.3 用户退出 13](#_Toc20329098)

[4.1.4 找回密码 14](#_Toc20329099)

[4.2 用户信息管理 15](#_Toc20329100)

[4.2.1 用户修改资料界面 15](#_Toc20329101)

[4.2.2 用户的订单管理 16](#_Toc20329102)

[4.2.3 用户的购物车管理 17](#_Toc20329103)

[4.3 后台管理 17](#_Toc20329104)

[4.3.1 图书的管理 17](#_Toc20329105)

[4.3.2 用户管理 18](#_Toc20329106)

[4.3.3 类别管理 19](#_Toc20329107)

[4.4 搜索业务 19](#_Toc20329108)

[4.4.1 搜索图书 19](#_Toc20329109)

[4.4.2 搜索用户 20](#_Toc20329110)

[5 系统测试及运行 22](#_Toc20329111)

[5.1 系统测试 22](#_Toc20329112)

[5.1.1 测试过程 22](#_Toc20329113)

[5.1.2 测试目的 22](#_Toc20329119)

[5.1.3 测试方法和工具 22](#_Toc20329120)

[5.2 系统运行 23](#_Toc20329121)

# 系统概述

## 构建目的

该图书网站总体目标是满足用户购买图书的目的，为用户提供一个先进、高效、安全、可靠的能有效的解决用户购书需求的网站体系。本网站完成之后可以大大的提高用户的购书体验，让用户享受到足不出户就可以买到书的需求，且更加高效，更加符合当代计算机迅速发展的趋势。

## 需要完成的软件和文档

|  |  |
| --- | --- |
| **程序名称** | **编程语言** |
| 前台页面 | Jsp |
| 后台管理 | Jsp、Java |
| 订单、购物车管理 | Java |
| 数据存储 | SQLserver |

。

|  |  |
| --- | --- |
| **文档名称** | **内容要点** |
| 立项申请 | 介绍本组项目的大致内容和功能 |
| 数据字典 | 声明该网站的数据存储 |
| PPT | 以图片的形势介绍本组的项目 |
| 项目开发文档 | 对本组项目进行总结汇报 |

## 项目的开发环境：

硬件环境：

1. CPU：英特尔 i7
2. 内存：8g
3. 硬盘：500g

软件环境：

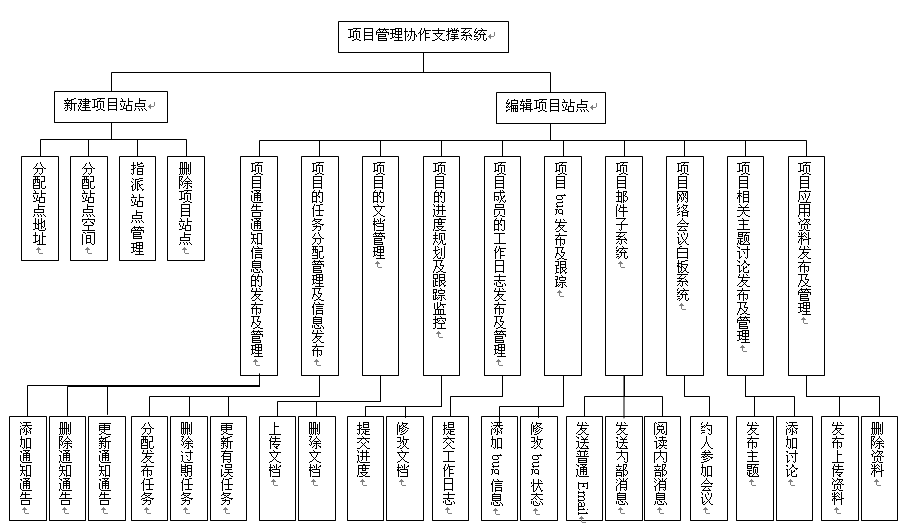
1．操作系统：Windows系统

2．开发工具：eclipse

3.数据库系统：MySQL

# 系统需求分析

产品功能需求概要分析：



# 数据设计

## 相关数据库表说明如下：

|  |  |  |
| --- | --- | --- |
| **序号** | **逻辑表名** | **说明** |
| 1 | 用户信息表 | 记录用户的基本信息 |
| 2 | 管理员表 | 记录管理员的登录信息 |
| 3 | 分类表 | 记录商品的所有分类 |
| 4 | 商品表 | 记录商品内容信息 |
| 5 | 推荐表 | 广告表，保存推荐商品信息 |
| 6 | 订单表 | 记录用户的订单信息 |
| 7 | 购物车表 | 记录用户的购物车信息 |

# 系统功能设计

要实现用户对该图书网站的正常使用，需要将用户的基本信息和本网站的信息整合起来，并为所有已注册的用户建立一张关系网。

## 用户账号管理

用户账号管理处理用户的注册、登入与退出系统功能模块

### 用户注册

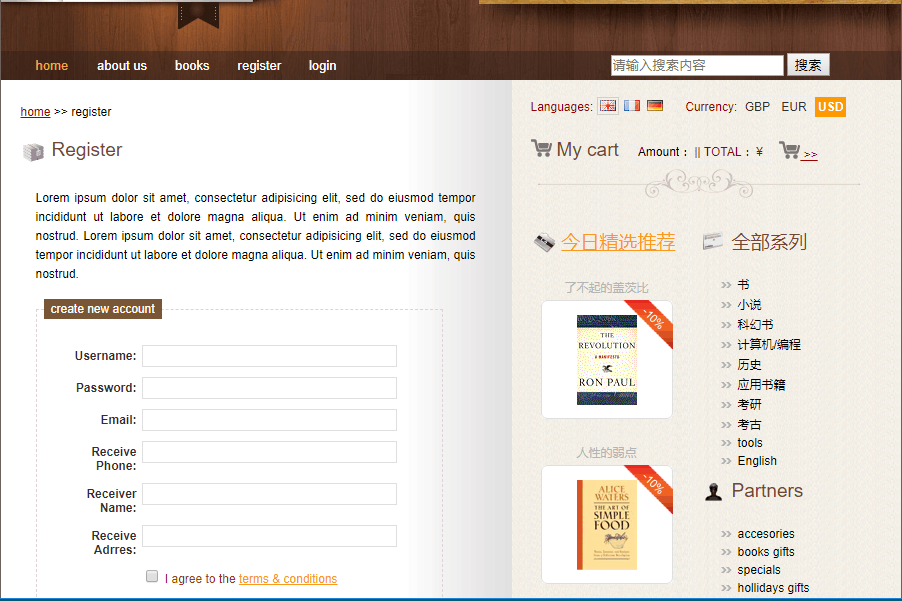
* + 1. 功能描述

用户在该网站注册页面成功注册一个用户账号，账号注册通过邮箱注册。如果注册信息有误，系统会自动提示用户输入正确的信息。

* + 1. 功能实现
  + 用户填写注册信息。
  + 信息无误，点击提交，完成注册，将数据写到用户数据库中。
  + 进入新用户首页。

具体程序实现如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **函数名称** | **功能说明** | **备注** |
| 1 | doPost | 实现用户的登陆 | HttpServletRequest request, HttpServletResponse response |

****

### 用户登入

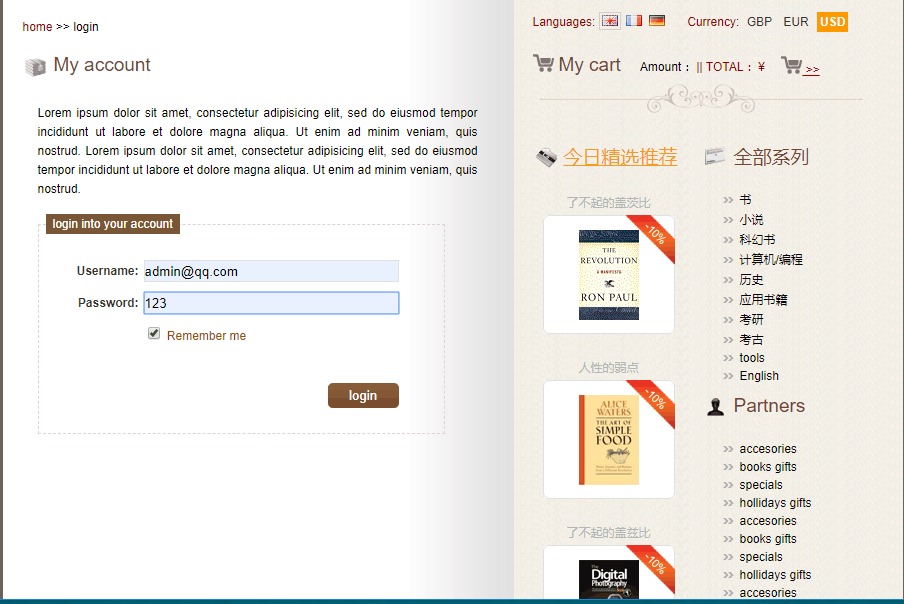
* + 1. 功能描述

已注册的用户可以根据注册的账号和密码登入系统，如果信息有误，系统提示用户输入有误。

* + 1. 功能实现
  + 用户填写登入信息。
  + 信息无误，点击提交，完成登入。
  + 进入用户首页。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **序号** | **类型** | **备注** |
| 1 | userLoginServlt | 封装函数 |
|  | doPost | 完成用户的登陆功能 |



### 用户退出

* + 1. 功能描述

登入成功的用户可在使用完系统后安全退出，浏览器会将保存的cookie删除，保证用户信息的安全。

* + 1. 功能实现
  + 用户点击退出按钮。
  + 退出系统，删除cookie。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| C | com.codecore.servlet.RemoveServlet | 用户退出的servlet，方法setCookie用于置空cookie。 |



### 找回密码

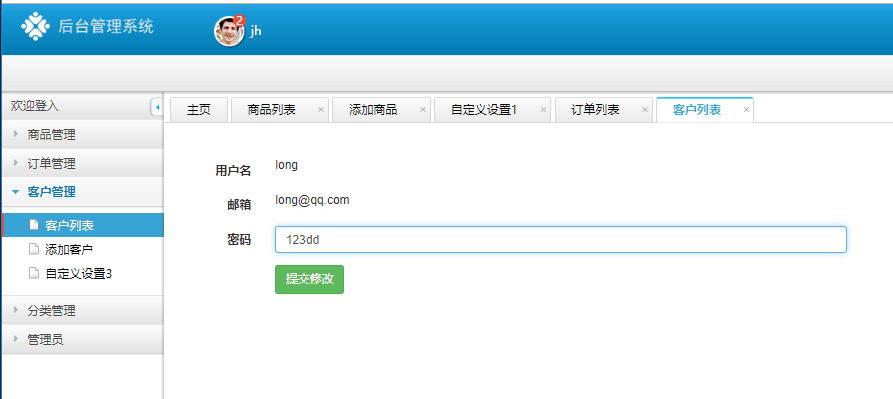
* + 1. 功能描述

用户在登入时，如果忘记密码，可以在找回密码页面输入信息，若更改成功则在数据库中更新密码。

* + 1. 功能实现
  + 用户填写信息。
  + 信息无误，点击提交，将密码发送到数据库。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| M | com.codecore.dao.UserDao | 方法findPassword用于查找信息匹配的密码 |
| C | com.codecore.servlet.FindPassword | 用户找回密码的servlet，给用户发邮件 |
| V | findpassword.jsp | 找回密码界面 |



## 用户信息管理

用户信息用于标示用户的特征，提供用户修改基本信息的接口。

### 用户修改资料界面

* + 1. 功能描述

资料修改界面方便用户对自己的个人信息进行改动，包括基本的用户信息、密码等的修改。

* + 1. 功能实现

修改资料需要将当前用户的信息显示出来，在用户对信息进行修改提交后，更新数据库信息。

用户信息通过请求servlet来获取，servlet调用DAO方法，返回一个UserInfo实体对象，包括所有的用户信息。

* + 将UserInfo转发到JSP页面，并相应的显示出来。
  + 根据用户的提交申请，及时更新数据库。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.UserInfo | 封装用户信息的实体 |
| M | com.codecore.dao.UserDao | 获取用户信息的DAO类，通过getUserInfoById获取用户信息 |
| C | com.codecore.servlet.UpdateimgServlet  com.codecore.servlet.UpdatepasswordServlet  com.codecore.servlet.UpdateuserInfoServlet | 获取用户信息并进行相应更新的的三个servlet |
| V | mypassword.jsp  userinfo.jsp  myface.jsp | 显示用户信息 |



### 用户的订单管理

* + 1. 功能描述

用户可以在订单页面查看自己的订单信息，并对订单进行修改、删除等操作。

* + 1. 功能实现
  + 用户查看订单
  + 若没有订单则显示为空，若有订单则可以对订单进行修改和删除，当数量为0时商品从订单中消失。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.AttentionDao | 添加关注的DAO，方法isAttention判断是否已添加订单 |



### 用户的购物车管理

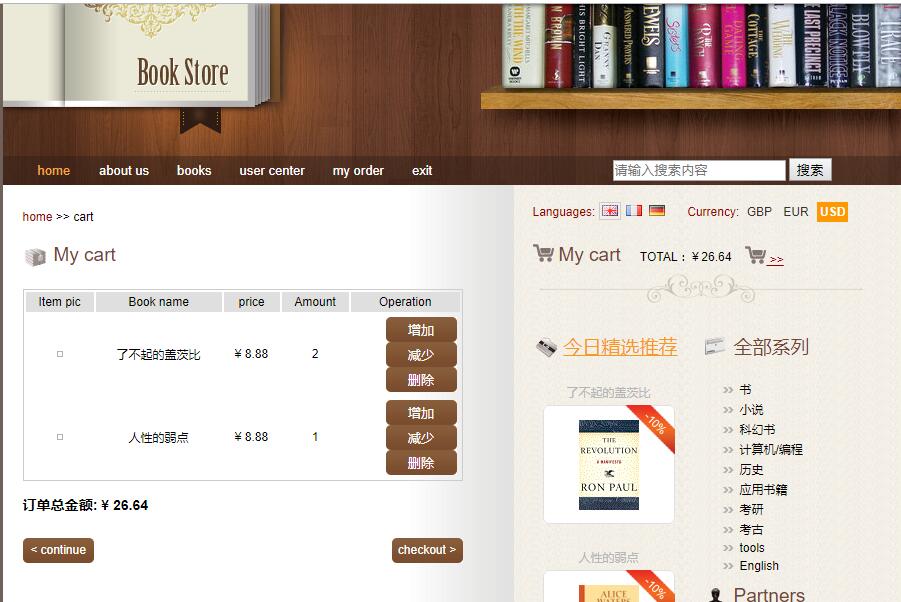
* + 1. 功能描述

用户可以将商品添加到购物车，并可以对购物车里的商品进行删除、添加等的操作。

* + 1. 功能实现
  + 可以通过数量的改变来改变价格
  + 当购物车商品数量为0时，商品从购物车中消失。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.entity.Friends | 封装用户关系的实体 |
| M | com.codecore.dao.FansDao | 获取图书信息的DAO，方法getFansByUid查找购物车信息 |
| V | friend.jsp | 显示购物车信息 |



## 后台管理

后台管理模块主要实现管理员对于图书和用户的管理，比如图书的上架与下架、图书信息的修改、类别的修改以及用户的增删改查等。

### 图书的管理

* + 1. 功能描述

管理员在登陆后台管理系统后，可以对图书进行上架和下架的操作，可以实现图书的增删改查以及图书的分类。

* + 1. 功能实现
  + 管理员在查询过图书后可对图书进行增删改查的操作。
  + 将修改过的数据插入到数据库中，并通过servlet将最新的修改信息显示出来

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.BlogPublish | 用来实现新图书的发布，blogpublish向数据库写入数据 |
| M | com.codecore.entity.Blog | 表示图书信息的实体类 |
| C | com.codecore.servlet.BlogPublishServlet | 发布图书的servlet，并将最新信息转发 |
| V | home.jsp | 显示发布图书的界面 |



### 用户管理

* + 1. 功能描述

管理员登陆后台管理系统后可以对用户进行增删改查的操作。

* + 1. 功能实现
  + 通过id查询所要修改的用户
  + 将修改过的用户信息更新到数据库，并可以在用户界面显示最新修改信息。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.BlogPublish | 查看图书的方法dispatchBlog |
| M | com.codecore.entity.Blog | 封装图书的实体类 |
| C | com.codecore.servlet.DispatchBlogServlet | 转发图书的servlet |
| V | home.jsp | 显示图书界面 |

### 类别管理

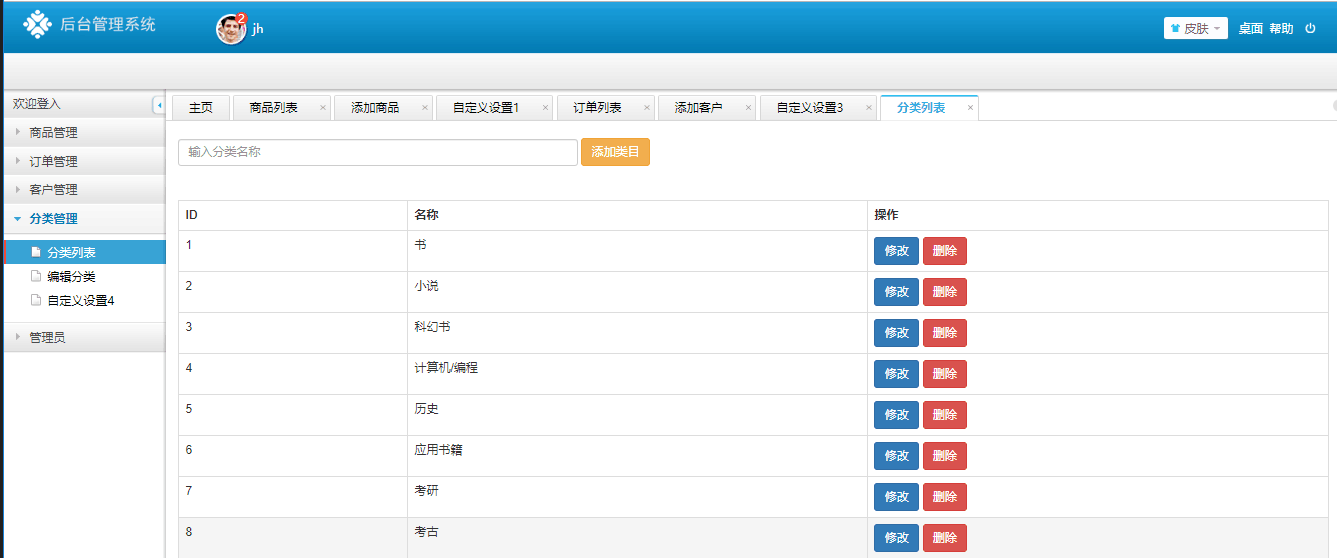
* + 1. 功能描述

管理员可以对类别进行增删改查等操作。

* + 1. 功能实现
  + 选择要修改的图书类别
  + 将修改后的图书类别更新到数据库，并且可以在图书页面显示出更新信息

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.CollectDao | 添加类别的方法addcollect |
| M | com.codecore.entity.Blog | 封装图书信息的实体类 |
| C | com.codecore.servlet.AddCollectServlet | 添加图书的servlet |
| C | com.codecore.servlet.ShowCollectServlet | 显示收藏的servlet |
| V | collect.jsp | 显示收藏的界面 |



## 搜索业务

### 搜索图书

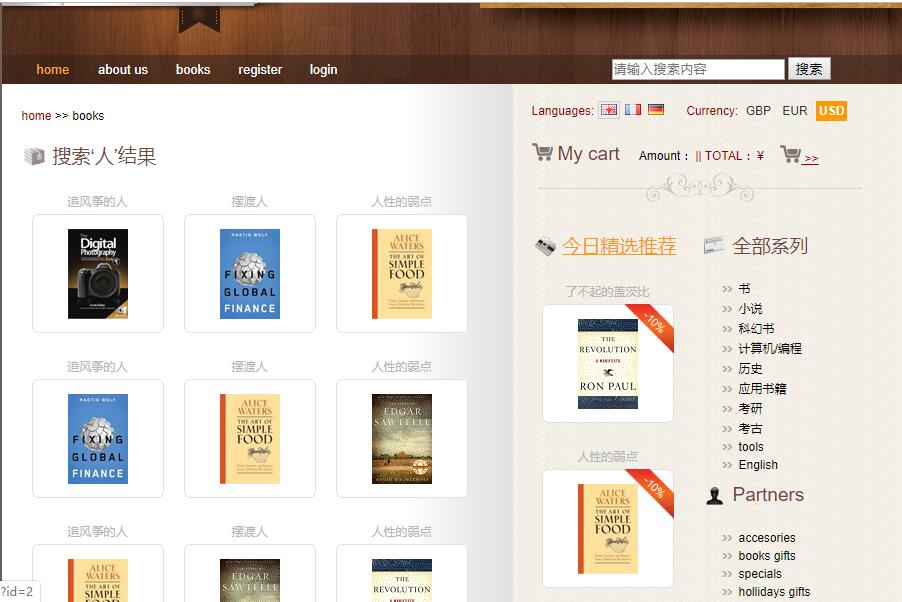
* + 1. 功能描述

用户在登入账户后，可对首页所有的图书按关键字或类别进行搜索，搜索结果返回在搜索结果集页中显示。

* + 1. 功能实现
  + 用户在搜索框中输入关键字或选择类别
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现图书的搜索，searchHome是实现搜索首页的方法。 |
| M | com.codecore.entity.Blog | 表示图书信息的实体类 |
| C | com.codecore.servlet.SearchHomeServlet | 搜索首页的servlet，将结果转发到其他页面 |
| C | com.codecore.servlet.SearchProfileServlet | 搜索我的图书页面的servlet，将结果返回到其他页面。 |
| V | home1.jsp | 显示首页搜索结果的界面 |
| V | profile1.jsp | 显示我的图书页面结果的界面 |



### 搜索用户

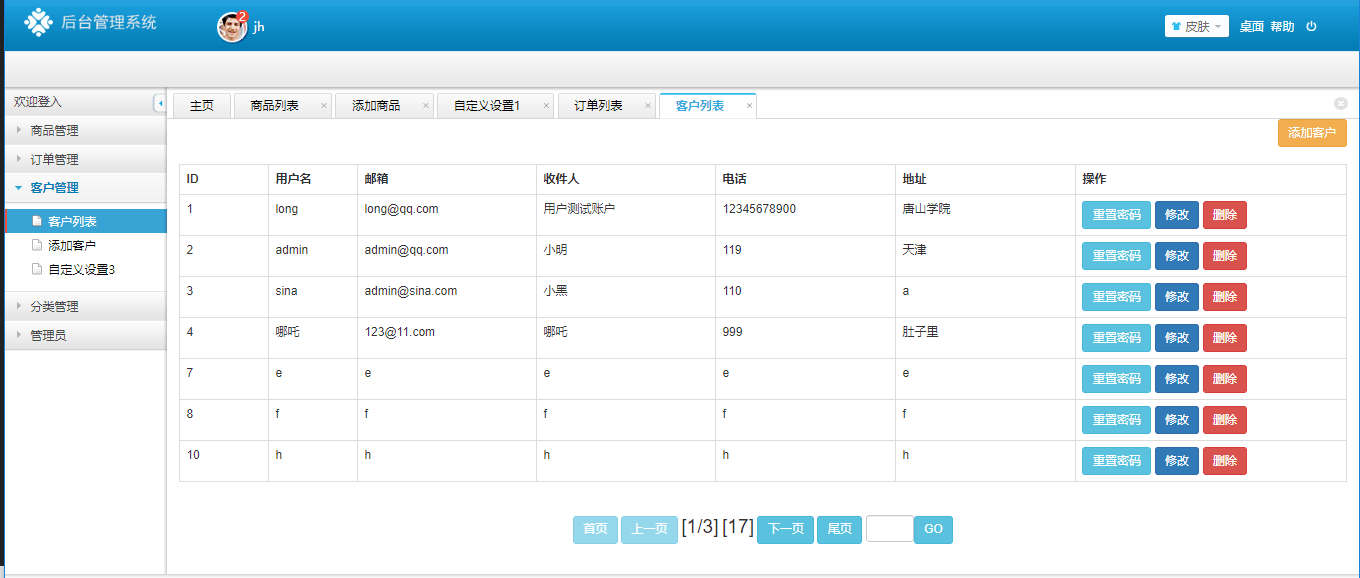
* + 1. 功能描述

管理员在登陆后台管理系统后，可通过用户id查看该用户的基本信息。

* + 1. 功能实现
  + 用户在搜索框中输入用户id。
  + 搜索请求servlet，servlet将搜索结果返回在搜索页中。

具体程序实现如下：

|  |  |  |
| --- | --- | --- |
| **层次** | **类型** | **备注** |
| M | com.codecore.dao.SearchDao | 用来实现搜索，searchFollowing是实现搜索关注人的方法。 |
| M | com.codecore.entity.UserInfo | 表示用户信息的实体类 |
| C | com.codecore.servlet.SearchFriendServlet | 搜索关注的servlet，将结果转发到其他页面 |
| V | friend1.jsp | 显示用户搜索结果的界面 |



# 系统测试及运行

## 系统测试

测试就是利用测试工具按照测试方案和[流程](http://baike.baidu.com/view/158951.htm)对产品进行功能和[性能测试](http://baike.baidu.com/view/106187.htm)，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

### 测试过程

测试过程按4个[步骤](http://baike.baidu.com/view/1121831.htm" \t "_blank)进行，即单元测试、集成测试、确认测试和系统测试及发版测试。

开始是单元测试，集中对用源代码实现的每一个[程序单元](http://baike.baidu.com/view/2610106.htm" \t "_blank)进行测试，检查各个[程序模块](http://baike.baidu.com/view/421086.htm)是否正确地实现了规定的功能。

集成测试把已测试过的模块组装起来，主要对与设计相关的[软件体系结构](http://baike.baidu.com/view/1317046.htm" \t "_blank)的构造进行测试。

确认测试则是要检查已实现的软件是否满足了需求规格说明中确定了的各种需求，以及[软件配置](http://baike.baidu.com/view/551674.htm" \t "_blank)是否完全、正确。

系统测试把已经经过确认的软件纳入实际运行环境中，与其它系统成份组合在一起进行测试。

### 测试目的

(1)测试并不仅仅是为了找出错误.通过分析错误产生的原因和错误的发生趋势,可以帮助[项目管理](http://baike.baidu.com/view/65955.htm" \t "_blank)者发现当前[软件开发](http://baike.baidu.com/view/190611.htm)过程中的[缺陷](http://baike.baidu.com/view/1122313.htm),以便及时改进。

(2)这种分析也能帮助测试人员设计出有针对性的测试方法,改善测试的[效率](http://baike.baidu.com/view/47610.htm" \t "_blank)和有效性。

(3)没有发现错误的测试也是有价值的,完整的测试是评定该网站质量的一种方法。

### 测试方法和工具

本系统采用如下方法进行测试：

（1）后台打印相关数据：在系统测试出错误的时候，可以把相关重要的数据和方法用后台打印的方法打印出来，通过打印出的数据进行分析从而找出漏洞所在。后台打印的好处是简单易行，工作量比较小。

（2）借助编译工具的辅助功能：可以在测试中使用MyEclipse的DeBug功能设置程序断点，跟踪程序找出错误所在，这样测试的好处是可以准确的跟踪程序走向，从整个程序运行的流程上一步一步找出问题所在。

测试工具：MyEclipse，通过编译工具MyEclipse的debug功能有效的跟踪代码，找出出错原因。

## 系统运行

本系统为JAVA编写的B/S结构的WEB项目，运行步骤如下。

1.运行windows操作系统。

2.由于本系统是由JAVA编写所以需要安装JDK1.6JAVA虚拟机器，安装方法为：运行JDK1.6安装文件，选择路径进行安装。

3.运行tomcat安装文件，选择安装路径，选择第2步已经安装好的JRE路径，进行tomcat的安装。

4.部署程序至tomcat。

5.安装MySQL数据库，导入表结构的SQL脚本，启动数据库实例服务。

6.启动tomcat。

7.打开浏览器并且输入如下网址。

<http://localhost:8080/>mail/welcome.jsp

系统需也要一些日常管理：包括日志的查看，服务器的维护等。