中国地质大学（北京）

《企业级.NET 应用开发》实践报告

专 业： 软件工程 学 号： 1004166127 姓 名： 林家辉 课程代码： DR043060 任课教师： 王雪 开课学期： 2018秋季学期 提交时间： 2018/1/1

**目 录**

1 总体任务 1

1.1 项目背景 1

1.2 项目目标 1

1.3 项目任务 1

1.4 任务分工 1

1.4.1 小组成员分工 1

1.4.2 项目进度安排 2

2 功能分析与设计 2

2.1 系统需求分析 2

2.1.1 业务流程 2

2.1.2 系统用例 5

2.2 系统架构设计 8

2.2.1 系统开发采用的技术手段 8

2.2.2 系统开发环境 8

2.2.3 系统部署实施环境 9

3 功能实现与关键代码 9

3.1 系统功能模块 9

3.1.1 前台图书浏览模块 9

3.1.2 用户注册和登录模块 10

3.1.3 购物车模块 11

3.1.4 订单结算模块 11

3.1.5 后台管理模块 12

3.2 功能代码实现 12

3.2.1 母版页 12

3.2.2 购物车 12

3.2.3 后台管理 14

3.2.4 权限管理 14

4 实训心得/错误分析 15

4.1 心得体会 15

4.2 错误分析 15

4.2.1 数据库连接问题 15

4.2.2 数据库中文乱码问题 15

4.2.3 角色授权管理问题 15

1. 总体任务

本项目的目的在于通过开发类似于当当网、卓越网或china-pub 的网上图书销售系统简易版，使开发者更加的深刻的认识与学习.NET开发的相关知识，为将来参与企业级.NET的开发、应用和研究奠定基础。

本项目来源于中国地质大学（北京）信息工程学院软件工程专业项目实践，开发周期约为15天。

本项目由中国地质大学（北京）信息工程学院MyBookShop开发团队研发，可实现用户通过系统网站进行在线图书订购，以及管理员通过后台进行对用户及图书的管理等功能。

* 1. 项目背景

随着当代社会的信息科技爆炸式的发展，线上经营的方式给线下实体经营带来愈来愈大的冲击。为了提高一个实体经营者的利润与生存空间，开发一个线上的经营网站已经成为了现如今的经营者的必要手段。

如今是互联网时代，人们衣食住行都离不开互联网。当然，人们的购物方式也发生了巨大的改变，就连买书也逐渐依赖线上图书销售网站。

* 1. 项目目标

1.实现用户注册和登陆

2.用户前台商品浏览，分类及按名称查找商品

3.用户将商品添加至购物车

4.用户订单生成和结算

5.管理员后台管理与维护系统

* 1. 项目任务

综合运用asp.net课程中所学内容，设计并实现一个具备基本功能的小型网上图书销售系统，系统实现主要包括前台商品浏览模块、用户注册登录模块、购物车模块、订单结算模块和后台管理模块。

* 1. 任务分工
     1. 小组成员分工

系统设计阶段的任务分工如表1所示。

表 1 小组成员及分工

| **组员** | **组内角色** | **组内分工** |
| --- | --- | --- |
| 林家辉 | 组长 | 安排小组分工，需求分析，完成数据库设计及连接，前台商品浏览模块，后台管理模块，报告编写 |
| 胡二雄 | 组员 | 需求分析，完成用户注册和登录模块，购物车模块，报告编写 |
| 郭浩洋 | 组员 | 需求分析，完成订单结算模块，界面设计，报告编写 |
| 焦武强 | 组员 | 需求分析，数据库数据录入，完成订单结算模块，报告编写 |

* + 1. 项目进度安排

本系统开发流程如表2所示。

表 2 系统开发流程表

|  |  |  |
| --- | --- | --- |
| **时间** | **工作安排** | **具体内容** |
| 2018/12/14-2018/12-15 | 需求分析 | 分析系统与要实现的功能 |
| 2018/12/16-2018/12-18 | 数据库设计 | 完成数据库表的创建 |
| 2018/12-19-2018/12/21 | 界面设计 | 完成网站各个界面的布局和设计 |
| 2018/12/22-2018/12/23 | 实现数据库连接 | 完成界面数据库接口与数据库的连接 |
| 2018/12/24-2018/12/28 | 项目具体开发 | 完成图书销售系统的各种功能模块 |
| 2018/12/29-2018/12/31 | 完成报告 | 绘制流程图、用例图，完成报告内容的书写 |

1. 功能分析与设计
   1. 系统需求分析

本次图书销售系统的需求主要包括用户权限管理，图书商品的展示，购物车功能，订单及结算，后台管理等部分，各部分业务流程及系统总体用例如下：

* + 1. 业务流程

系统各部分业务流程图如下图1-5：

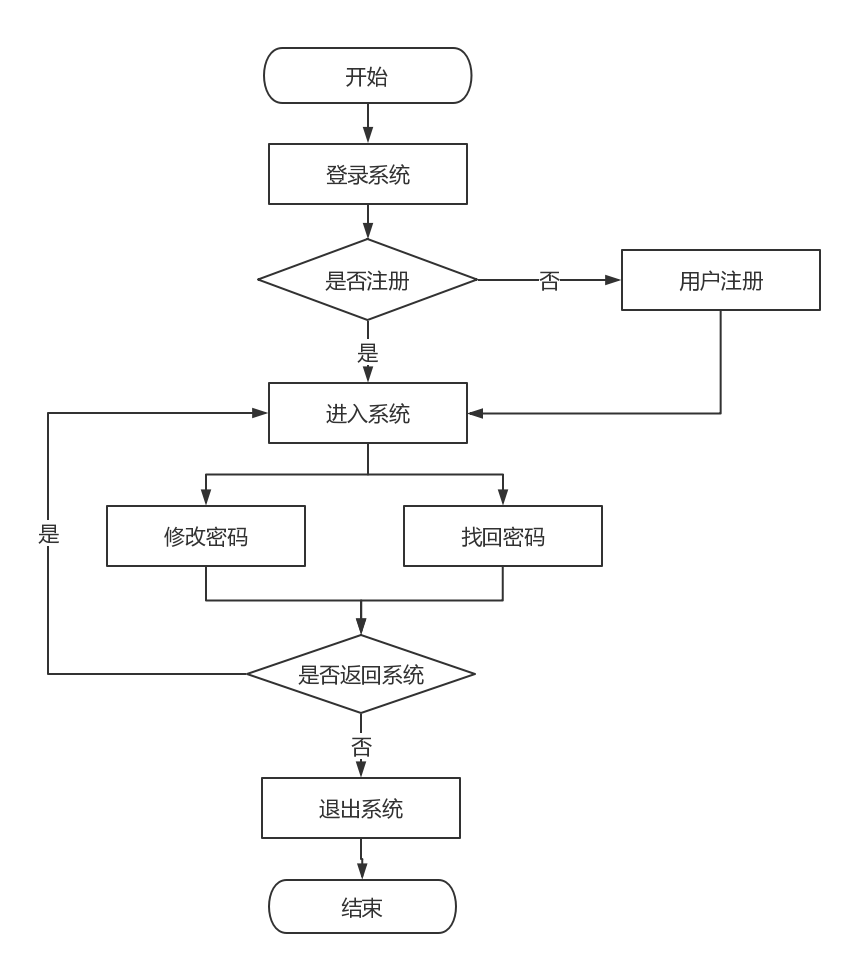


图 1 用户登录注册模块流程

用户注册和登录模块与通常的会员系统类似，用户注册以后就可以成为系统的会员。用户只有在成功登录系统后，才可以实现对所购图书的结算。注册用户还具有修改密码和找回密码的功能。

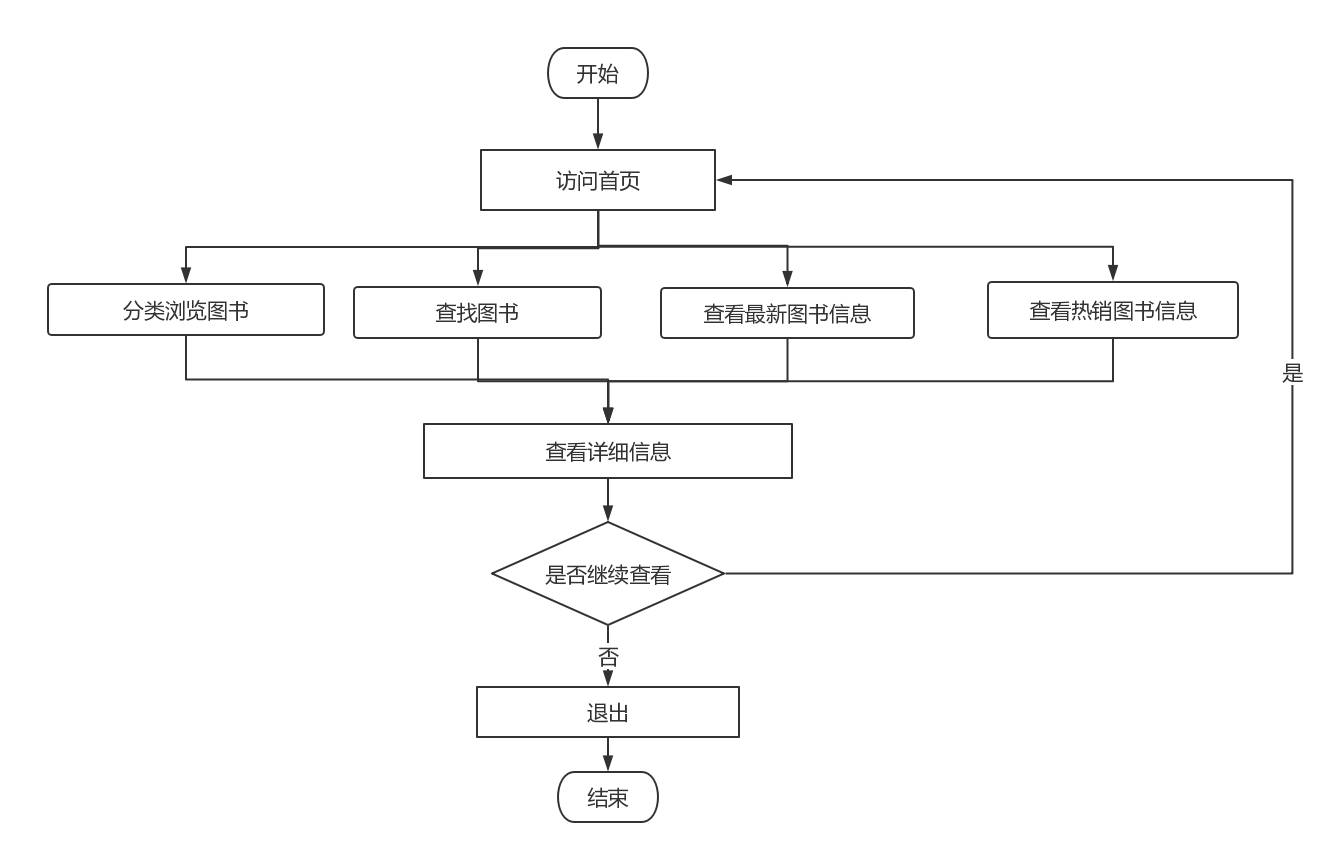


图 2 前台图书浏览模块流程

按照电子商务网站的一般规划和人们使用电子商务网站的习惯，前台图书浏览模块主要实现按照各种条件显示和查看商品的前台显示功能。用户使用前台图书浏览模块的流程图如图2所示。

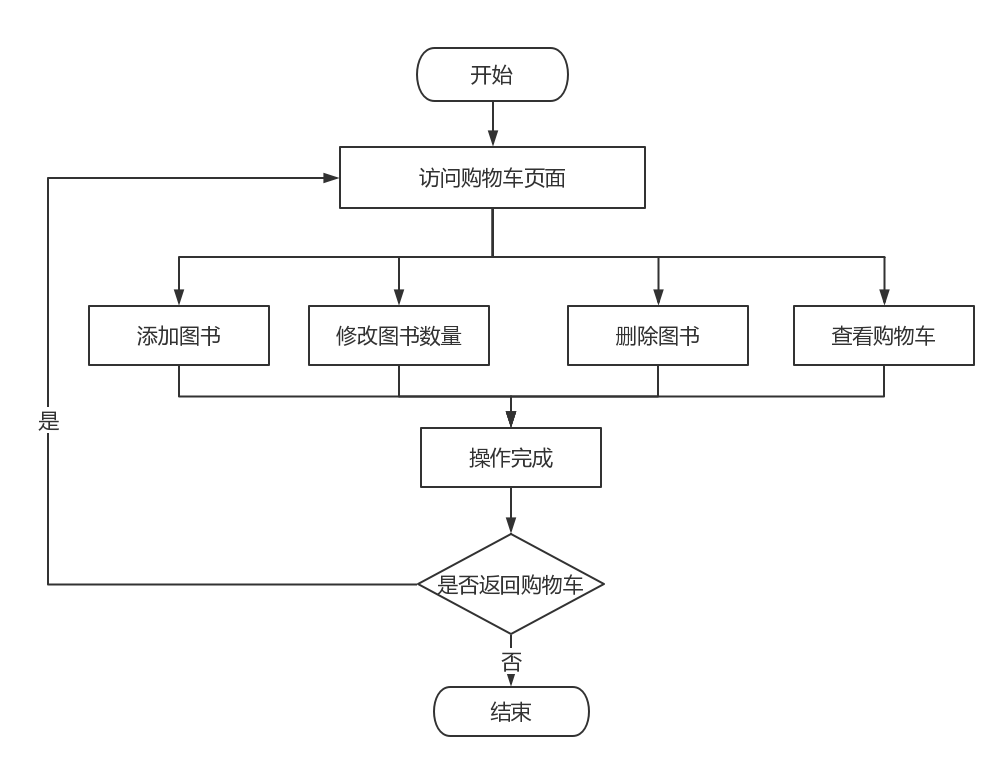


图 3 购物车模块

购物车是每个电子商务网站的基本元。本系统购物车中包含了用户准备购买的所有图书的信息，包括书号、图书名称、图书价格、购买数量以及用户应付总价等。用户在查看图书详细信息时，如果决定购买即可将图书加入购物车，之后可以继续浏览其它图书。

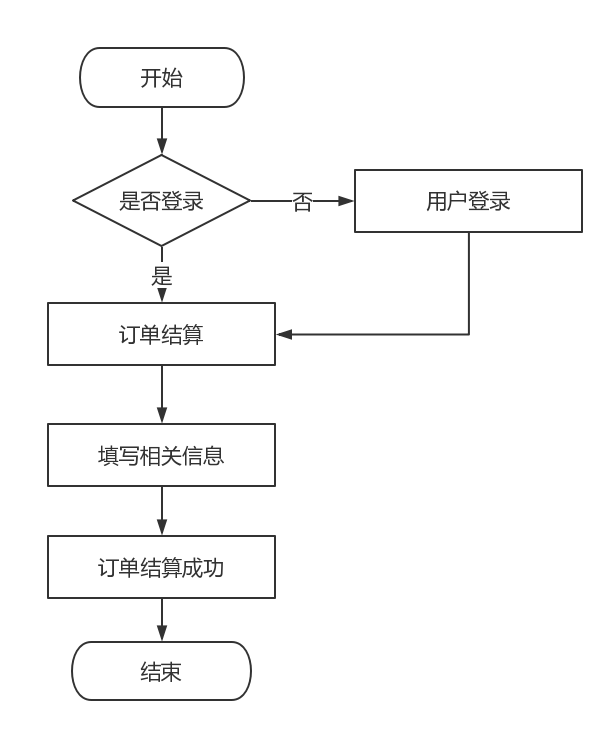


图 4 订单结算模块流程

用户完成购物后即可进入结算中心，系统对购买的图书及数量进行价格计算， 最后生成用户应付金额。然后用户向系统下达订单并提供送货地址等信息。

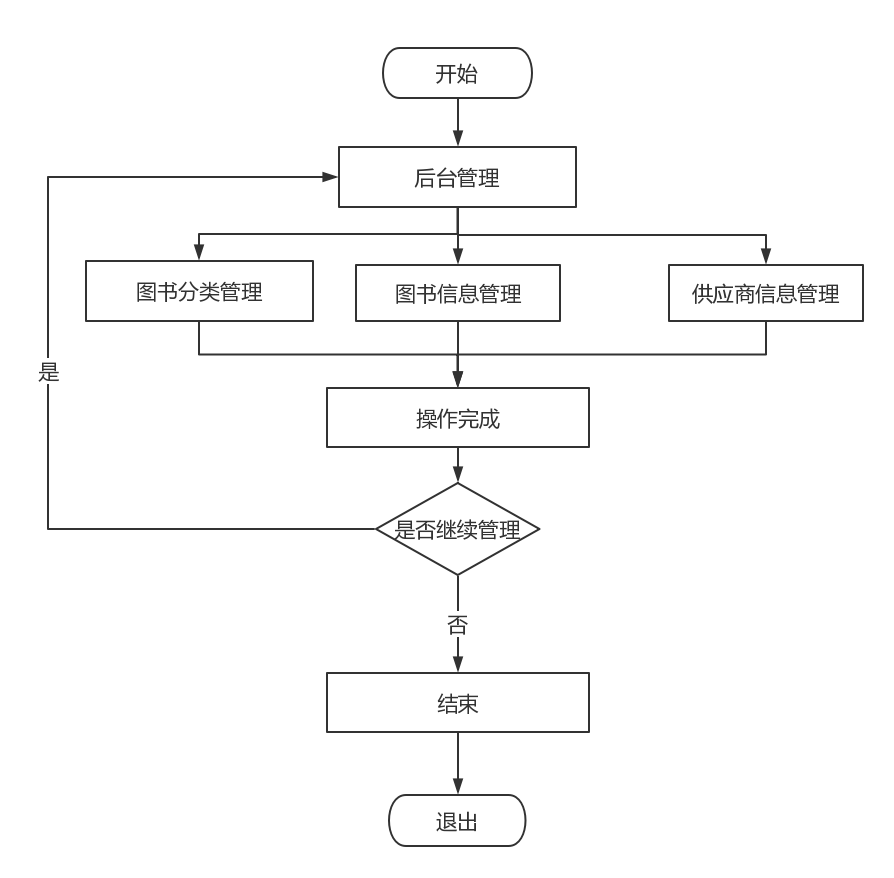


图 5 后台管理模块流程

后台管理模块是根据系统数据维护要求而设计的后台管理平台，只有拥有管理员角色的用户才可以进入后台功能模块实现系统的维护和管理。

* + 1. 系统用例

系统参与者及用例：

用户：注册、登录

顾客（用户泛化）：浏览在售图书、（查看图书详情）、添加到购物车、管理购物车、结算、（支付订单）

管理员（顾客泛化）：管理出版社信息、管理订单、管理图书分类、管理图书信息

系统用例图如下图6：



图 6 系统用例图

用例描述如下表3-9：

表 3 login用例描述

|  |  |
| --- | --- |
| 用例名称 | login |
| 参与者 | 用户 |
| 前置条件 | 用户进入网上图书销售系统 |
| 用例描述 | 用户登录系统 |
| 基本操作流程 | 用户输入正确的账号密码后可以进入系统 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

表 4 register用例描述

|  |  |
| --- | --- |
| 用例名称 | register |
| 参与者 | 顾客 |
| 前置条件 | 顾客进入网上图书销售系统 |
| 用例描述 | 新顾客注册账号 |
| 基本操作流程 | 进入系统后顾客进入注册界面可注册账号 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

表 5 browse books用例描述

|  |  |
| --- | --- |
| 用例名称 | browse books |
| 用例描述 | 顾客浏览销售中的图书的信息 |
| 参与者 | 顾客 |
| 前置条件 | 顾客登陆成功 |
| 后置条件 | 无 |
| 基本操作流程 | 顾客登录后，在前台浏览在售的图书信息，图书信息分页展示，可调整每页显示图书数量，点击页面右下角的换页按钮可以换页 |
| 被泛化的用例 | 无 |
| 被扩展的用例 | add to shopping cart、browse details |

表 6 settle用例描述

|  |  |
| --- | --- |
| 用例名称 | settle |
| 用例描述 | 顾客选好图书后结算 |
| 参与者 | 顾客 |
| 前置条件 | 顾客登陆成功 |
| 后置条件 | 支付成功 |
| 基本操作流程 | 顾客选好图书后，添加至购物车，在购物车中选择结算 |
| 被包含的用例 | 无 |
| 被扩展的用例 | pay for order |

表 7 modified books’ quantity用例描述

|  |  |
| --- | --- |
| 用例名称 | modified books’quantity |
| 用例描述 | 管理图书信息 |
| 参与者 | 管理员 |
| 前置条件 | 管理员账号登陆成功 |
| 后置条件 | 无 |
| 基本操作流程 | 管理员在后台可以修改图书信息 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

表 8 manage suppliers’ information用例描述

|  |  |
| --- | --- |
| 用例名称 | manage suppliers’information |
| 用例描述 | 管理出版社信息 |
| 参与者 | 管理员 |
| 前置条件 | 管理员账号登陆成功 |
| 后置条件 | 无 |
| 基本操作流程 | 管理员在后台对出版社信息进行管理 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

表 9 manage orders用例描述

|  |  |
| --- | --- |
| 用例名称 | manage orders |
| 用例描述 | 管理订单信息 |
| 参与者 | 管理员 |
| 前置条件 | 管理员账号登陆成功 |
| 后置条件 | 无 |
| 基本操作流程 | 管理员在后台对订单信息进行管理 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

* 1. 系统开发环境及技术
     1. 系统开发采用的技术手段
* C#、XHTML5、CSS3、JavaScript
* ASP.NET web程序开发技术
* SQL Server数据库存储技术
* Microsoft Ajax技术
  + 1. 系统开发环境
* ASP.NET 4.5
* IIS 10.0.17763.1
* SQL Server 2016 LocalDB数据库
* SQL Server 2017数据库
* Microsoft Visual Studio 2017
* Windows 10操作系统
  + 1. 系统部署实施环境
* ASP.NET 4.5
* IIS 10.0.17763.1
* SQL Server 2017

1. 功能实现与关键代码
   1. 系统功能模块

系统共包括5个模块，分别为前台商品浏览模块，用户注册和登录模块，购物车模块，订单结算模块，后台管理模块。

* + 1. 前台图书浏览模块

前台图书浏览模块主要包括首页，商品详情页，搜索页面等部分。首页通过自定义用户控件提供最新图书，热销图书，图书分类以及在售所有图书的信息显示，同时各部分提供超链接导向图书详情页。在图书详情页通过GridView显示图书的详细信息，并提供分页显示功能和购买连接。商品搜索页面提供模糊查询功能，根据用户输入的关键字在数据库Book表中实现对书名的模糊查询。搜索页面布局及显示与详情页类似。

图 7 首页



图 8 图书详情

* + 1. 用户注册和登录模块

用户注册和登录模块主要包括注册新用户，用户登录，修改密码，找回密码，以及退出系统等部分。注册新用户页面使用了CreateUserWizard控件，同时在Web.config中配置了<system.net>配置节并注册了126邮箱实现发送注册邮件功能，将随机生成的密码发送到用户指定邮箱中。用户登录通过Login控件实现登录功能，同时页面提供链接导向注册新用户和忘记密码页面。修改密码通过ChangePassword控件实现，若用户未登录需要提供用户名，原密码，新密码并确认，用户已登录则不需要提供用户名。找回用户密码通过PasswordRecovery控件实现，用户正确回答注册时的安全问题后将会通过邮件把随机生成的新密码发送到用户注册的邮箱中。退出系统由LoginStatus控件实现。所有用户信息保存在aspnetdb数据库中。



图 9 用户登录

* + 1. 购物车模块

购物车模块主要包括购物车页面。首先在Web.config中配置了<profile>元素，定义了一个用户个性化配置属性组Cart，包括书号，书名，购买数量，单价，总价五个属性。购物车页面提供购物车的全部功能，包括删除图书，修改数量，清空购物车等，通过GridView显示购物车内图书。

图 10 购物车

* + 1. 订单结算模块

订单结算模块包括创建订单，查看订单。用户点击购物车内结算按钮将会跳转到创建订单页面，采集到的页面主要包括文本框和数据验证控件，用于用户填写信息，并存入数据库。查看订单页面使用GirdView控件显示用户所有订单信息。

* + 1. 后台管理模块

后台管理模块位于Admin文件夹下并通过aspnetdb数据库赋予权限，包括图书分类管理，出版社信息管理，图书管理和订单管理四大部分。

图书分类管理通过GridView及数据库连接实现对图书分类的增加、删除、修改等功能；出版社信息管理，图书管理及订单管理实现原理类似。

* 1. 功能代码实现

系统重要功能包括母版页，购物车，以及权限管理和后台管理等功能，具体实现及说明如下：

* + 1. 母版页

母版页的设计包括网站整体结构，使用了css样式表以及html标签布局，同时使用了4个ContentPlaceHolder供内容页布局。

由于母版页代码过多故不在此展示。

* + 1. 购物车

购物车页面使用了GridView控件展示购物车信息，同时在Web.config中配置了<profile>元素，定义了一个用户个性化配置属性组Cart，包括书号，书名，购买数量，单价，总价五个属性。

GridView部分代码如下：

<asp:GridView ID="gvCart" runat="server" Width="100%" AutoGenerateColumns="false">

<Columns>

<asp:TemplateField>

<ItemTemplate>

<asp:CheckBox ID="chkBook" runat="server" />

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="BookId" HeaderText="商品ID" />

<asp:BoundField DataField="Name" HeaderText="商品名称" />

<asp:BoundField DataField="ListPrice" HeaderText="商品价格" DataFormatString="{0:c}" />

<asp:TemplateField HeaderText="购买数量">

<ItemTemplate>

<asp:TextBox ID="txtQty" runat="server" Text='<%# Bind("Qty") %>' Width="30"></asp:TextBox>

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

添加图书到购物车代码如下：

/// <summary>

/// 将指定图书号的图书添加到购物车，购物车以Profile形式存储

/// </summary>

/// <param name="bookId">指定的图书号</param>

protected void AddBook(string bookId)

{

//isExit表示商品是否已在购物车中，值1表示“是”，值0表示“否”

int isExit = 0;

for (int j = 0; j < Profile.Cart.Name.Count; j++)

{

//若同类图书已经存在于购物车中，则将该图书的购买数量加1

if (bookId == Profile.Cart.BookId[j].ToString())

{

int qty = (int)Profile.Cart.Qty[j];

qty++;

Profile.Cart.Qty[j] = qty;

Profile.Save();

isExit = 1;

}

}

if (isExit == 0)

{

//若购物车中无指定图书编号的图书，则添加一个新图书到Profile.Cart的各个属性中

var books = from p in db.Book

where p.BookId == bookId.ToString()

select p;

foreach (var book in books)

{

Profile.Cart.ListPrice.Add(book.ListPrice);

Profile.Cart.Qty.Add(1);

Profile.Cart.BookId.Add(book.BookId);

Profile.Cart.Name.Add(book.Name);

}

Profile.Save();

}

}

Web.config中的Profile设置如下：

<!--用于存储购物车信息的Profile设置-->

<profile enabled="true" defaultProvider="ProfileProvider" automaticSaveEnabled="true">

<providers>

<clear />

<add name="ProfileProvider" type="System.Web.Profile.SqlProfileProvider" connectionStringName="ASPNETDBConnectionString" passwordFormat="Clear" />

</providers>

<properties>

<group name="Cart">

<add name="BookId" type="System.Collections.ArrayList" allowAnonymous="true" />

<add name="Name" type="System.Collections.ArrayList" allowAnonymous="true" />

<add name="Qty" type="System.Collections.ArrayList" allowAnonymous="true" />

<add name="ListPrice" type="System.Collections.ArrayList" allowAnonymous="true" />

<add name="TotalPrice" allowAnonymous="true" />

</group>

</properties>

</profile>

* + 1. 后台管理

后台管理包括分类管理，图书管理，出版社管理，订单管理等，由Admin文件夹下的各个aspx页面分别实现。

图书管理中添加图书的代码如下：

protected void btnAdd\_Click(object sender, EventArgs e)

{

if (IsExitBook(txtName.Text.Trim())) //输入的书名已存在

{

lblNameErr.Text = "图书已经存在";

}

else //添加图书到Book表

{

Book book = new Book();

string fileName;

string fileFolder;

fileName = Path.GetFileName(fupImage.PostedFile.FileName);

fileFolder = Server.MapPath("~/") + "Images\\bookimg\\";

fileFolder = fileFolder + fileName;

fupImage.PostedFile.SaveAs(fileFolder);

book.BookId = txtId.Text.Trim();

book.Image = "~\\Images\\bookimg\\" + ddlCategoryId.SelectedItem.Text + "\\" + fileName;

book.Name = txtName.Text.Trim();

book.CategoryId = int.Parse(ddlCategoryId.SelectedValue);

book.PressId = int.Parse(ddlPressId.SelectedValue);

book.ListPrice = decimal.Parse(txtListPrice.Text.Trim());

book.Author = txtAuthor.Text.Trim();

book.Descn = txtDescn.Text.Trim();

book.Qty = int.Parse(txtQty.Text.Trim());

db.Book.InsertOnSubmit(book);

db.SubmitChanges();

Response.Redirect("BookMaster.aspx");

}

}

* + 1. 权限管理

权限管理通过aspnetdb以及Admin文件夹中的Web.config配置实现，Web.config中内容如下：

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<system.web>

<authorization>

<allow roles="Admin" />

<deny roles="Member" />

<deny users="?" />

</authorization>

</system.web>

</configuration>

1. 实训心得/错误分析
   1. 心得体会

这次课程实践，虽然进行时间较紧，由于有了之前Javaee的项目经验，小组还是顺利的完成了项目任务。

本次实践我们在课堂教学以及书后实例的基础上，对图书销售系统的需求进行了分析，进行了数据库的设计，实现了用户通过系统网站进行在线图书订购，以及管理员通过后台进行对用户及图书的管理等功能。顺利完成了项目目标。

通过本次实践，我们了解了使用ASP.NET开发项目的大体流程，熟悉了各个控件的使用方式，学会了如何进行用户身份验证以及如何使用aspnetdb进行用户权限管理。对将来使用ASP.NET开发打下了基础。为课程学习画上了圆满的句号。

* 1. 错误分析

这次实践中遇到的问题及错误主要包括数据库连接问题，数据库中文乱码问题，角色授权管理问题等，对错误的分析如下：

* + 1. 数据库连接问题

由于使用到了aspnetdb数据库进行用户权限管理，所以需要aspnetdb数据库，但是因为未知原因VS2017并没有自动生成aspnetdb，最终使用了SQL Server通过aspnet\_regsql.exe工具在SQL Server中生成aspnetdb并配置连接字符串。此外WebPart使用个性化服务时需要在web.config文件里面配置一个连接字符串，否则运行时会报错，所以使用网上的方法覆盖了默认的连接字符串，而使用自定义的。

* + 1. 数据库中文乱码问题

图书数据库使用的是SQL Server 2016(LocalDB)，默认中文字符乱码，在查询网上相关信息后，执行以下查询语句并将需要输入中文字符的字段类型改为nvarchar即可解决：

alter database "E:\.Net Core\Database\hm.mdf" set single\_user with rollback immediate ;

go

alter database "E:\.Net Core\Database\hm.mdf" collate Chinese\_PRC\_CI\_AS ;

go

alter database "E:\.Net Core\Database\hm.mdf" set multi\_user;

其中引号部分改为自己数据库文件路径。

* + 1. 角色授权管理问题

由于VS2017没有提供ASP.NET网站管理工具，事实上该工具于VS2013中就从IDE中去除，但可以通过命令行方式启动，而VS2017通过命令行方式仍然不能启动，所以最后管理员账号的添加是通过对aspnetdb数据库的表进行分析和修改实现的。具体操作大致为正常注册新账号，在aspnetdb中aspnet\_UserInRoles表中添加相关条目即可。