**北京工商大学**

综合项目实训三 **实验报告**

**学院：**计算机与信息工程学院

**课程名称：** 软件工程

**任课教师：** 毛明毅

**班级：** 计算机142

**学号：** 1204030825

**姓名：** 孙楷

**班级：计算142 学号：1404030825 姓名：孙楷**

**同组同学学号：1404030137、P2014005**

**同组同学姓名：周朋飞、贺然**

**实验日期：2017年 5 月 17 日 交报告日期：2017 年 6 月 21 日**

|  |
| --- |
| **实验(No.14 )题目：北工商校园点餐系统** |
| **实验目的及要求：**综合运用软件工程思想、面向对象分析和设计方法、ASP.NET、数据库等基础知识，完成一个B/S架构的Web应用程序,掌握应用ASP.NET技术开发中小型网站的相关知识和技术 |
| **实验设备：**  **windows操作系统、visual studio 2017、sql server 2016、chrome浏览器、阿里云服务器、git、BootStrap** |
| **实验内容及步骤：**  我们小组本次实训做的是“高校食堂自助配餐系统”的项目。此项目采用三层架构的实现方式，分为数据访问层（包含LINQ2SQL类的编写）、业务逻辑层（包含C#类、实体类）、页面设计和UI（包含各个页面布局、母版页、用户控件）。此项目包含买家功能模块：商品搜索模块（可以按名称和类别搜索）、买家信息模块（个人信息、注册、修改密码、找回密码、修改其他信息）、购物车模块（添加商品、移除商品）、提交结算模块、订单查询模块（查询所有订单、按订单状态查询）、订单评价模块、余额查询-充值模块；卖家功能模块：商家信息模块（同买家）、查询收益模块、商品模块（添加、修改、删除商品）、订单查询模块（同买家）；管理员功能模块：管理模块（删除卖家、买家，冻结、解冻卖家）、搜索模块（买家、卖家）。 课程设计目的 综合运用软件工程思想、面向对象分析和设计方法、ASP.NET、数据库等基础知识，完成一个B/S架构的Web应用程序,掌握应用ASP.NET技术开发中小型网站的相关知识和技术。 课程设计内容 针对一个高校食堂自助配餐系统项目需求，利用数据库管理系统（Microsoft Sql Server 2016或其他），构建项目数据库系统。利用 Visual Studio 2017 集成开发环境，采用ASP.NET技术、C#语言、WEB Forms窗体设计技术、LINQ数据访问技术等，完成系统界面设计和功能开发，并对系统进行功能测试。 需求分析（功能模块分析与数据库设计）学生到食堂用餐，在和排队上浪费很多时间，并且去晚了经常会吃不到想吃的食物；学生对食堂的满意度不高，有许多的学生会选择去学校周边的饭店用餐。因此，食堂更无法准确预测学生需求，经常会出现有些食物因为没有卖出去只好倒掉，而学生需要的一些食物却已卖完的现象。  本产品是为校园餐厅专门开发的一套订餐管理系统，旨在合理化安排餐厅的工作，提高餐厅的管理效率，同时方便学生就餐。系统功能分析：   此系统功能模块包含买家的商品搜索模块、信息模块、购物车模块、提交结算模块、订单查询模块、订单评价模块、查询余额-充值模块；卖家的信息模块、查询收益模块、商品模块、订单查询模块；管理员的管理模块、搜索模块。 系统用例图 通过对系统的需求分析，系统的基本功能已经大体确定。整个系统的用例图分为三个部分，顾客用例图，卖家用例图，管理员用例图。       系统体系结构 本网上订餐系统采用了B/S架构的设计体系，B/S结构相对于传统的C/S结构，这种结构使得数据只有结果集合在浏览器中显示，数据的处理在服务器进行，用户不用安装客户端，而且由于通过服务器端统一管理数据，很大程度简化了开发WEB应用的难度；易于保证数据的一致性。  B/S模式的三层模型：  第一层表示层。用户接口部分就是数据的表示层，把用户和系统连接起来传递信息的作用。它可以实现对输入的数据进行检查，对外输出的数据的功能，以便显示出来。当某Web浏览器想要获取资源，对一个服务器申请服务时，服务器在验证了用户信息后，把所需文件资料传送给客户端，采用的是HTTP协议传送，文件资料就可以被用户接收了并且可以显示在浏览器上。  第二层事务逻辑层。事务逻辑层是应用的主体，它也可以被称为功能层，业务处理程序基本上都在这一层中。统计、汇总、分析、打印功能等等这些都是位于功能层。申请的方式是通过SQL向数据库进行操作，根据具体的需要和用户的特别要求，链接程序使用数据库里的数据，处理之后的数据会从数据服务器转到web上的服务器，最后客户端从web服务器上得到数据。  第三层数据库层。数据库层由两部分组成，一部分是数据库管理系统，另一部分则是数据库，各个部分分别处理各自的任务，任务是对数据库进行处理，对数据库的操作例如更改、查询等，web服务器能够获取最后的结果。负责管理其数据库的事数据库管理系统，提供给应用程序各种各样的服务，数据库管理系统处在数据库服务器端 项目详细设计 系统模块详细设计  本系统总共有两大主要模块，分别是顾客模块和管理员模块。每个模块分别有不同的权限和使用功能。  登录模块的流程图  开始    用户名    密码  N  是否合法    Y  结束  顾客模块详细设计  顾客模块的主要功能是浏览，订餐。浏览与自己相关的美食信息，订单信息等。用户要下订单之前美食车要把详细食物信息列出，物品的单价，购买的数量，总共的价钱，可修改和删除购买，当符合顾客要求的时候，顾客就可以下订单。下订单时首先会让用户确定一下订单是否付合要求，确认无误后，进入填写接收信息后生成订单。然后对数据库进行操作完成。  顾客  订单确认  显示购物车  修改订单  不购买此商品  结束操作  登录  删除购物车信息  提示错误  验证登陆用户id是否正确  N  Y  确认结账  N  Y  返回  订单生成  提示错误  Y  N 系统实现 买家信息模块：    上图为登录界面，简单的注册登录功能和验证码功能    使用一个随机数，只会程序会检测输入框内的文本与其是否一致    因为买家和卖的的登录界面都在一个地方，所以在登录过程中会创建SellService、buyerIntSer两个实例，通过他们的landing方法判断在数据库中是否有对应用户信息  在landing方法中调用了一个Exist的方法，将用户名、密码传过去，最终转换成对数据库的访问，得到一个布尔值    如果返回值为真，就在Session中加入一系列参数，然后将页面跳转到默认页面  之后类似的账户充值、修改密码、个人信息展示等用户信息都用buyerIntSer  中的方法实现  修改密码的程序段：  public void Updatepassword(string account, string password)  {  buyer\_Entity B = new buyer\_Entity();  B.B\_account = account;  B.B\_password = password;  Date.UpDatePassword(B);  }  充值金额的程序段  //--买家充值--  /// <summary>  /// AddMoney函数的功能：买家充值  /// </summary>  /// <param name="account">买家账户</param>  /// <param name="money">充值金额</param>  public void AddMoney(string account, double money)  {  buyer\_Entity B = new buyer\_Entity();  B.B\_account = account;  B.B\_balance = money;  Date.addmoney(B);  }  提价订单程序段  //--提交计算模块--  /// <summary>  /// SubmitOrder函数的功能：提交订单  /// </summary>  /// <param name="Oid">订单ID</param>  /// <param name="price">商品价格</param>  /// <param name="num">商品数量</param>  public void SubmitOrderItem(int Oid, int Pid, double price, int num, DateTime time)  {  b\_order\_item\_Entity I = new b\_order\_item\_Entity();  I.B\_o\_id = Oid;  I.P\_id = Pid;  I.P\_price = price;  I.P\_num = num;  Date.InsertToOrderItem(I);  }  每一个具体功能都用一个页面来实现，因为使用了母版页，所以在每一个界面中只需要改变核心区域的代码即可  这些就是母版页中引用的功能模块  用户信息的母版页：    下面是显示个人信息的web窗体  在编写显示用户信息的页面是，代码量就会少很多  <%@ Page Title="" Language="C#" MasterPageFile="~/ModifyUser.Master" AutoEventWireup="true" CodeBehind="showInformationOfBuyer.aspx.cs" Inherits="高校食堂自助配餐系统.buyer.showInformationOfBuyer" %>  <%@ Register Src="~/control\_buyer/C\_showInformationOfBuyer.ascx" TagPrefix="uc1" TagName="C\_showInformationOfBuyer" %>  <%@ Register Src="~/control\_buyer/C\_NavStatus.ascx" TagPrefix="uc1" TagName="C\_NavStatus" %>  <%@ Register Src="~/control\_buyer/C\_LoginStatus.ascx" TagPrefix="uc1" TagName="C\_LoginStatus" %>  <asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder3" runat="server">  <uc1:C\_LoginStatus runat="server" ID="C\_LoginStatus" />  </asp:Content>  <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">  <uc1:C\_NavStatus runat="server" ID="C\_NavStatus" />  </asp:Content>  <asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">  <uc1:C\_showInformationOfBuyer runat="server" ID="C\_showInformationOfBuyer" />  </asp:Content>  **使用asp:content，对特定ID号的内容进行更改就能完成页面的设计**  这是就最终的效果图，中间的内容被C\_showInformationOfBuyer.ascx代替，而其余网页中相同的部分就不用更改。  主界面商品展示的实现    程序采用datalist来显示所有的菜品，list中的每一个item包含菜品的图片、名称、价格      其中点击图片还能进入物品的详情页面  protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)  {  ImageButton b = sender as ImageButton;  string id = b.CommandArgument;  Response.Redirect("~/ProductDetail.aspx?id=" + id + "");  }  通过QueryString将id传到下一个页面    在这个页面通过对ID的选择，将相应的菜品信息显示出来，同时根据id在数据库中找到其他用户对这个商品的评价。    在菜品的上面有一些分类信息和搜索功能  两种功能的实现都是通过QueryString 将传递的关键字，然后刷新页面实现的    这是通过点击不同的按钮，将菜品通过category传递到当前页面  在页面载入的时候  通过select语句筛选数据库中的菜品，将得到的数据集绑定在datalist中，这样就能达到筛选菜品的功能了  同时 程序通过一个ajax控件PopupControlExtender在点击的时候会在界面上弹出一个悬浮框      将购物车里的布局信息写到一个panel控件中，该控件在平时的时候是不会显示出来的，当通过PopupControlExtender 控件，设置其响应事件后，在指定位置就能将购物车里的内容展示出来  **买家的业务代码：**    1.login 登录\*  2.registered 注册\*  3.Modify 修改\*  4.Retrieve\_pwd 密码找回\*  5.Add\_goods 添加商品\*  6.Modify\_goods 修改商品\*  7.Delete\_goods 删除商品\*  8.Number\_query 订单编号查询\*  10.Accept\_orders 接受订单\*  程序演示：    在首页点击加盟按钮，就能申请一个卖家账户    对应的业务代码  protected void Button1\_Click(object sender, EventArgs e)  {  SellService db = new SellService();  seller\_Entity a = new seller\_Entity();  a.S\_account = txtSell\_Account.Text;  a.S\_password = txtSell\_password.Text;  a.S\_name = txtSell\_Name.Text;  a.S\_storeName = txtSell\_storename.Text;  a.S\_tel = txtSell\_tel.Text;  a.S\_question = ddlQuestion.Text;  a.S\_answer = txtSell\_answer.Text;  if (!db.exist(txtSell\_Account.Text))  {  Session["UserAccount"] = a.S\_account.Trim();  Session["UserPassword"] = a.S\_password.Trim();  Session["Logged"] = true;  Session["UserType"] = "1";  db.create(a);  Response.Write(MessagesBox.showMessages("注册成功"));  System.Web.Security.FormsAuthentication.RedirectFromLoginPage(a.S\_account, true);  }  else  {  Response.Write(MessagesBox.showMessages("账号已存在，请重新输入"));  }  用户在点击注册按钮后，程序会创建一个seller\_Entity类，将用户填的信息都传进去，之后如果账号不存在，就在数据库中写入该用户，并在session中填入对应信息，之后将页面跳转到首页    在点击用户名的时候，就会依次出现个人信息、修改密码、添加商品、在售商品、编号查询、个人收益等信息  protected void Page\_Load(object sender, EventArgs e)  {  string account = Session["UserAccount"].ToString();  seller\_T seller = new seller\_T();  SellService sell = new SellService();  seller = (seller\_T)sell.SearchSellinformation(account);  lbl\_s\_account.Text = seller.s\_account;  lbl\_s\_name.Text = seller.s\_name;  lbl\_s\_profit.Text = seller.s\_profit.ToString();  lbl\_s\_state.Text = seller.s\_state?"账号有效":"账号无效";  lbl\_s\_storename.Text = seller.s\_storeName;  lbl\_s\_tel.Text = seller.s\_tel;  }  在卖家页面，用户点击个人信息系统会从session中取出UserAccount  在根据SellService中的SearchSellinformation方法从数据库中取出该用户的所有信息，并将其绑定在相应控件上      protected void Button1\_Click(object sender, EventArgs e)  {  SellService a = new SellService();  product\_Entity b = new product\_Entity();  b.S\_account = Session["UserAccount"].ToString().Trim();  b.P\_name = txtProduct\_name.Text;  b.P\_category = txtProduct\_category.Text;  b.P\_price = double.Parse(txtProduct\_price.Text);  b.P\_cNum = int.Parse(txtProduct\_number.Text);  b.P\_photo = FileUpload1.FileName;  a.add(b, FileUpload1);  Response.Write(MessagesBox.showMessages("添加成功"));  }  利用SellService product\_Entity 两个对象，能很容易地完成数据的添加，先使用product\_Entity对象，将用户上传的信息保存，之后使用SellService中的add方法就能将菜品写到表中    编号查询使用一个gridview控件完成  protected void btn\_Search\_Click(object sender, EventArgs e)  {  SellService a = new SellService();  GridView1.DataSource = a.searchAllOrder(txtSell\_Id.Text);  GridView1.DataBind();  }  管理员界面的实现      管理员登录后能查看所有的买家、卖家信息    应为管理员界面比较少，没有使用母版页      在设计过程中陷入了一些比较繁琐的界面设计流程，实现起来效率比较低。    在查询信息的时候，显示判断搜索框内是否有文本，之后调用adminInterSevice的searchUserReport方法，最后将得到的数据绑定在gridview中 |
|  |

注：《实验内容及步骤》项目的内容如果较多，可以加附页。

|  |
| --- |
| **实验结果及心得:**  我们实训小组做的高校食堂自助配餐系统实现了买家注册、登录、修改、改密、找回密码、查询订单、订单评价、充值、查询余额、搜索商品、加入购物车、下单；卖家注册、登录、修改、改密、找回密码、查询订单、添加商品、删除商品、修改商品、查询某个时期的净收益；管理员的查询买家、卖家，删除买家、卖家，冻结、解冻卖家等一系列操作。其中商品展示、订单提交页面的逻辑代码稍微复杂一点，也算得上是两大亮点吧。商品展示使用DataList控件，每页显示20个数据，采用分页的技术方便用户查看商品；购物车的代码逻辑比较复杂，要判断各种复杂的情况防止出错，同时还要向买家订单表数据表插入数据，插入的过程中用到了触发器向卖家订单表插入数据以及修改卖家收益、买家余额。页面的设计结合了BootStrap框架的技术，使页面变得绚丽多彩。不足的地方有很多，比如说代码书写不规范，注释不详细，代码不健壮有很多漏洞，异常处理机制不完善。  本次实训体会颇深，我们三个人一起完成中小型的项目。从中也发现：小组人数越多，项目分工难度就越大，协调好整个项目组的难度也就越大。实训意味着你要和别人一起进行项目开发，这个时候就要学会怎么和别人沟通、交流。程序不是一个人写的，而是整个项目小组智慧的结晶，代码的质量将决定程序修改和扩展的难易程度，一份高质量的代码，对于项目维护来说无疑变得更加轻松。其实实训也就是为你工作做准备，通过实训检查你的编程能力，锻炼你的编程思维，提高你的代码熟练度。在实训过程中，经常遇到各种奇怪的错误，有时候一个问题要搞一上午，这时候就需要足够的耐心和毅力了。有时候由于中途的功能的变更，导致之前写好的数据访问的函数无法使用，这个时候就得重构代码了。每当这个时候，组员就崩溃了，做了好久的东西一下就报废了，从中我也意识到一份好的设计文档非常重要，模块的功能划分要非常清晰。数据库的设计十分重要，在详细功能确定以后才进行，数据库一旦确定就不能修改了，各数据表的字段及其类型都得确定。数据库是整个项目的底层，也是整个项目的核心。一旦数据库发生改变，那么逻辑代码要进行大的变动。本次实训我们采用了三层架构的设计思想，实现数据的分层处理。在实训过程中发现自己有许多知识没有搞透彻，经常需要百度查资料。总的来说本次实训意义非凡、收获颇丰，实训锻炼了我们的各方面的能力，希望在以后的实训中能够表现的更加优秀。 |
| **教师评语：**  **成绩:**  **教师签字： 年 月 日** |