**信息科学技术学院**

**课程（实习）设计**

**功能及使用说明文档**

**课程设计名称：** 软件工程教学实习

**专 业：** 计算机科学与技术

(培养服务外包人才试点专业)

**学 号 姓 名：** 郑逸梅

**成 绩：**

**批 改 日 期：**

**教 师 签 名：**

**目录**

[1 ASP.NET三层架构 1](#_Toc3094)

[1.1 什么是三层架构 1](#_Toc17464)

[1.2 实现步骤 1](#_Toc26869)

[1.3 三层架构实例 3](#_Toc15743)

[1.3.1 User.cs 3](#_Toc14604)

[1.3.2 UserDAL.cs 4](#_Toc22164)

[1.3.3 UserBLL.cs 5](#_Toc6753)

[2 数据存储过程 6](#_Toc21244)

[2.1 什么是数据存储过程 6](#_Toc20883)

[2.2 数据存储过程举例 6](#_Toc32580)

[3 具体功能实现 7](#_Toc20931)

[3.1 Web层具体页面 7](#_Toc32707)

[3.2 登陆界面功能说明 8](#_Toc7350)

[3.2.1 实现注册及登录 8](#_Toc19125)

[3.2.2 注册代码 8](#_Toc1312)

[3.2.3 登录代码 9](#_Toc20439)

[3.2.4 登录使用说明 11](#_Toc27707)

[3.3 点餐界面功能说明 11](#_Toc8742)

[3.3.1 实现点餐 11](#_Toc7441)

[3.3.2 存储过程 12](#_Toc23675)

[3.3.3 添加菜品——设计思路 14](#_Toc15144)

[3.3.4 删除菜品——设计思路 14](#_Toc20)

[3.3.5 提交订单——设计思路 14](#_Toc10089)

[3.4 管理员界面功能说明 15](#_Toc17480)

[3.4.1 功能说明 15](#_Toc5509)

[3.4.2 界面设计 15](#_Toc18678)

[3.4.3 查询功能代码 17](#_Toc7448)

[3.4.4 添加新菜品功能代码 18](#_Toc27004)

[3.4.5 删除菜品/用户信息功能代码 20](#_Toc11618)

# 1 ASP.NET三层架构

## 1.1 什么是三层架构

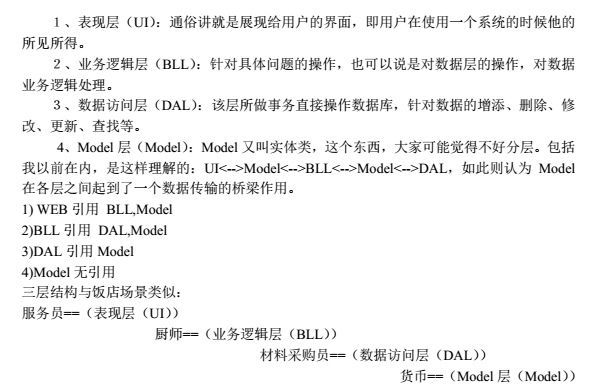


图1

1.2 实现步骤

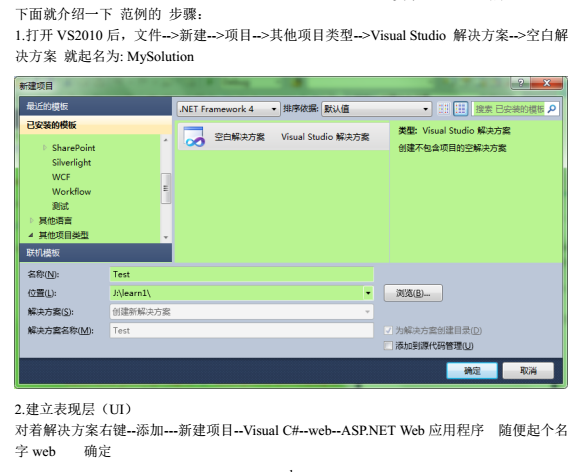


图2

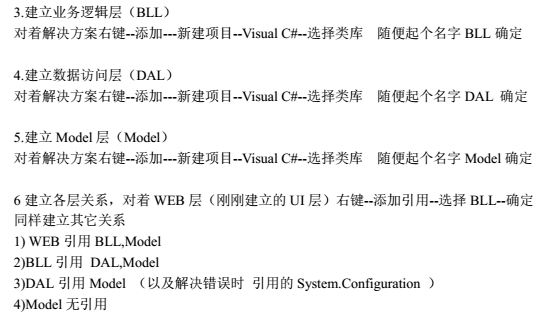


图3

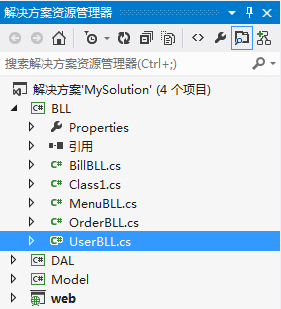
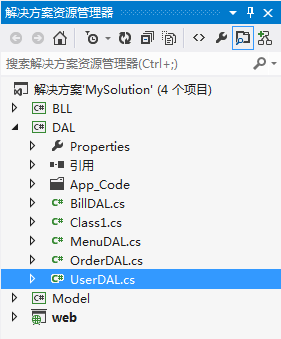
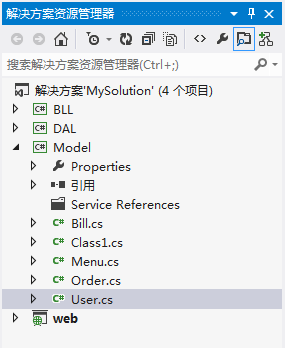


图6 BLL层

图5 DAL层

图4 Model层

在Web.config文件中添加缺少的数据链接字符串



图7 Web.config文件

## 1.3 三层架构实例

### 1.3.1 User.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Model

{

public class User

{

public User() { }

private string \_Id;

private string \_Email;

private string \_Username;

private string \_Pwd;

private string \_Role;

public string Id

{

set { \_Id = value; }

get { return \_Id; }

}

public string Email

{

set { \_Email = value; }

get { return \_Email; }

}

public string Username

{

set { \_Username = value; }

get { return \_Username; }

}

public string Pwd

{

set { \_Pwd = value; }

get { return \_Pwd; }

}

public string Role

{

set { \_Role = value; }

get { return \_Role; }

}

}

}

### 1.3.2 UserDAL.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using DAL.App\_Code;

namespace DAL

{

public class UserDAL

{

DBOperation db = new DBOperation();//作为公共的实例化

ProOperation pr = new ProOperation();

//登录

public bool Login(Model.User ur)

{

bool flag=false;

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@pwd",SqlDbType.VarChar,20)

};

parameter[0].Value = ur.Username;

parameter[1].Value = ur.Pwd;

dt = pr.GetDataTable("login",parameter);

if (dt.Rows.Count > 0)

flag = true;

return flag;

}

}

}

### 1.3.3 UserBLL.cs

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BLL

{

public class UserBLL

{

DAL.UserDAL User = new DAL.UserDAL();

public bool Login(Model.User u)

{

return User.Login(u);

}

public DataTable LoginRecord(Model.User u)

{

return User.LoginRecord(u);

}

public bool AddUser(Model.User u)

{

return User.AddUser(u);

}

public DataTable searchUser()

{

return User.searchUser();

}

public DataTable point\_searchUser(Model.User u)

{

return User.point\_searchUser(u);

}

public void DeleteUser(Model.User u)

{

User.DeleteUser(u);

}

}

}

# 2 数据存储过程

## 2.1 什么是数据存储过程

存储过程（Stored Procedure）是在大型数据库系统中，一组为了完成特定功能的SQL 语句集，存储在数据库中，经过第一次编译后再次调用不需要再次编译，用户通过指定存储过程的名字并给出参数（如果该存储过程带有参数）来执行它。存储过程是数据库中的一个重要对象，任何一个设计良好的数据库应用程序都应该用到存储过程。

## 2.2 数据存储过程举例

**数据库中：**

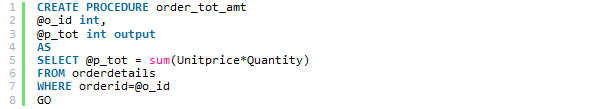


图7 Web.config文件

**ASP.NET中：**

//登录

public bool Login(Model.User ur)

{

bool flag=false;

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@pwd",SqlDbType.VarChar,20)

};

parameter[0].Value = ur.Username;

parameter[1].Value = ur.Pwd;

dt = pr.GetDataTable("login",parameter);

if (dt.Rows.Count > 0)

flag = true;

return flag;

}

# 3 具体功能实现

## 3.1 Web层具体页面

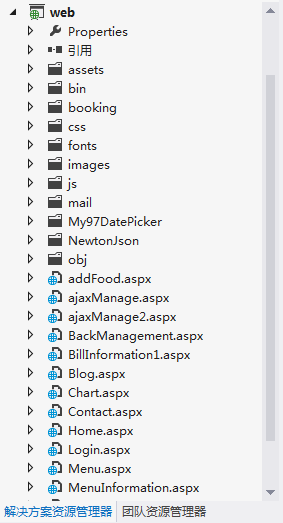


图9 具体web层程序流程图

图8 具体web层

addFood.aspx

Chart.aspx

MenuInformation.aspx

BillInformation1.aspx

OrderInformation.aspx

UserInformation1.aspx

Blog.aspx

BackManagement.aspx

Login.aspx

Contact.aspx

Menu.aspx

Home.aspx

## 3.2 登陆界面功能说明

### 3.2.1 实现注册及登录

图11 登录界面

图10 注册界面

### 3.2.2 注册代码

**DAL：**

public bool AddUser(Model.User ur)

{

bool flag = false;

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10)

};

parameter[0].Value = ur.Username;

dt = pr.GetDataTable("AddUser", parameter);

if (dt.Rows.Count > 0)

{

return flag;

}

else

{

IDataParameter[] parameter1 = {

new SqlParameter("@email",SqlDbType.VarChar,50),

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@pwd",SqlDbType.VarChar,20),

new SqlParameter("@role",SqlDbType.VarChar,6)

};

parameter1[0].Value = ur.Email;

parameter1[1].Value = ur.Username;

parameter1[2].Value = ur.Pwd;

parameter1[3].Value = "用户";

pr.ExcutePro("AddUser1", parameter1);

flag = true;

return flag;

}

**WEB:**

protected void btn\_ZhuceClick(object sender, EventArgs e)

{

string email = this.Email.Value.ToString(); if(Regex.IsMatch(email,"[A-Za-z0-9][@][A-Za-z0-9]+[.][A-Za-z0-9]")==true)

{

string username = this.Zhuce\_Username.Value.ToString();

string password = this.Zhuce\_Password.Value.ToString();

user.Email = email;

user.Username = username;

user.Pwd = password;

bool flag = ur.AddUser(user);

if (flag)

Response.Write("<script>alert('注册成功!');</script>");

else

Response.Write("<script>alert('注册失败!');</script>");

}

else

{

Response.Write("<script>alert('您的电子邮箱格式不正确!');</script>");

}

}

### 3.2.3 登录代码

**DAL：**

**//登录**

public bool Login(Model.User ur)

{

bool flag=false;

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@pwd",SqlDbType.VarChar,20)

};

parameter[0].Value = ur.Username;

parameter[1].Value = ur.Pwd;

dt = pr.GetDataTable("login",parameter);

if (dt.Rows.Count > 0)

flag = true;

return flag;

}

**//登录时数据的返回**

public DataTable LoginRecord(Model.User ur)

{

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@pwd",SqlDbType.VarChar,20)

};

parameter[0].Value = ur.Username;

parameter[1].Value = ur.Pwd;

dt = pr.GetDataTable("Login", parameter);

return dt;

}

**WEB:**

protected void btn\_LoginClick(object sender, EventArgs e)

{

string username = this.Username1.Value.ToString();

string password=this.Password1.Value.ToString();

user.Username = username;

user.Pwd = password;

bool flag = ur.Login(user);

if (flag)

{

DataTable dt = new DataTable();

dt = ur.LoginRecord(user);

if (dt.Rows.Count > 0)

{

Session["username"] = dt.Rows[0]["username"].ToString();

if (dt.Rows[0]["role"].ToString() == "管理员")

{

Response.Redirect("BackManagement.aspx");

}

else

{

Response.Redirect("Home.aspx");

}

}

}

else

Response.Write("<script>alert('用户和密码不匹配!');</script>");

}

}

### 3.2.4 登录使用说明

* 管理员登录账号：

用户名：zym

密码：12345

* 普通用户登录：

1. 先注册自己的账号
2. 再用注册的账号进行登录

## 3.3 点餐界面功能说明

### 3.3.1 实现点餐

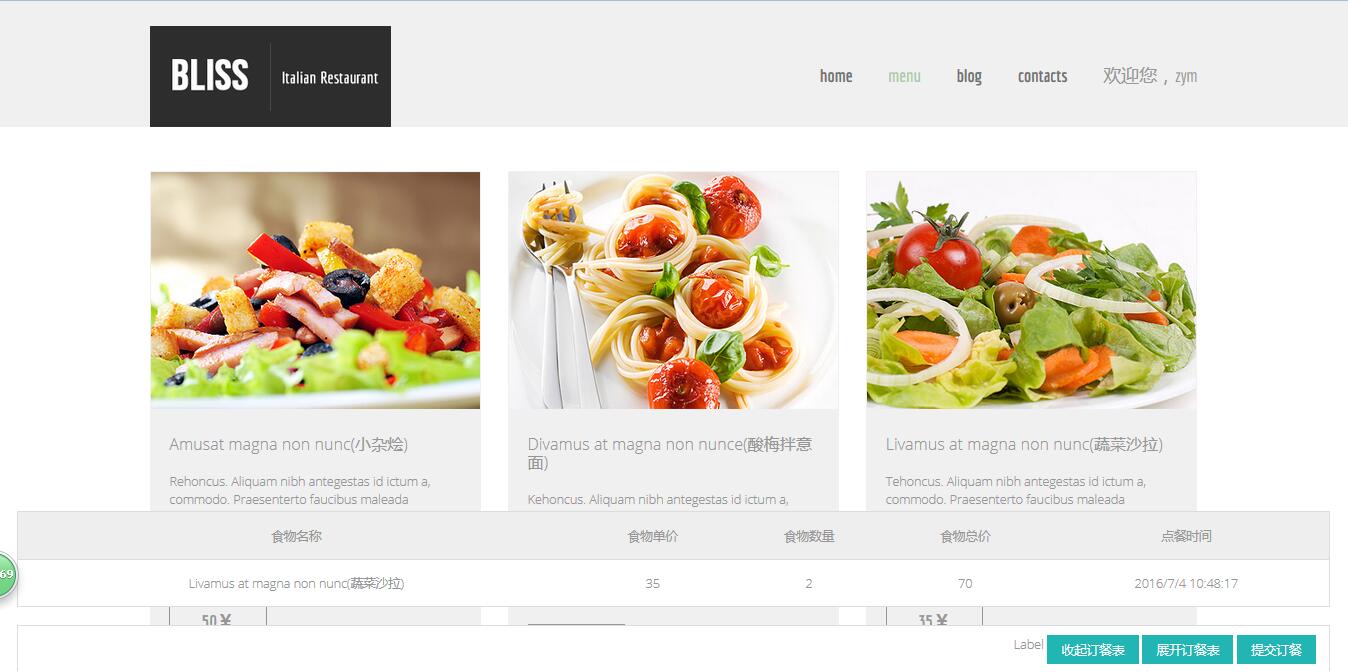


图12 点餐界面

### 3.3.2 存储过程

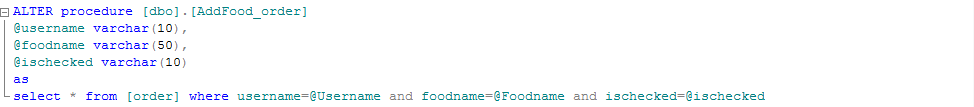


图13 AddFood\_order的存储过程

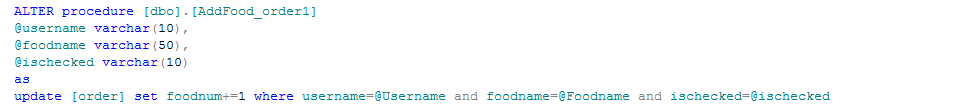


图14 AddFood\_order1的存储过程

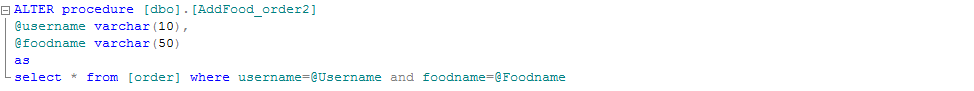


图15 AddFood\_order2的存储过程

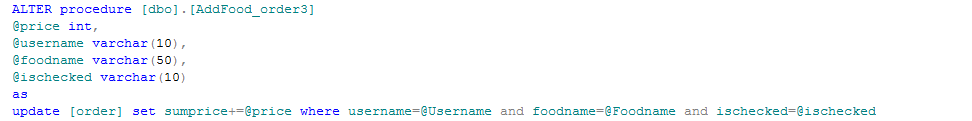


图16 AddFood\_order3的存储过程

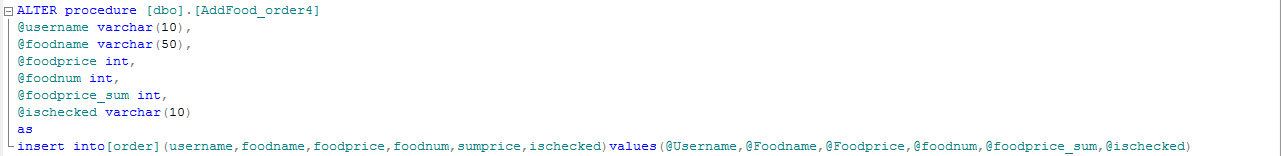


图17 AddFood\_order4的存储过程

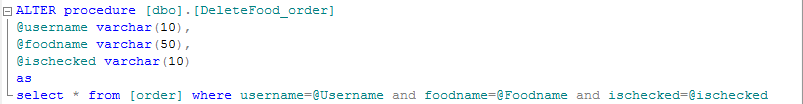


图18 DeleteFood\_order的存储过程

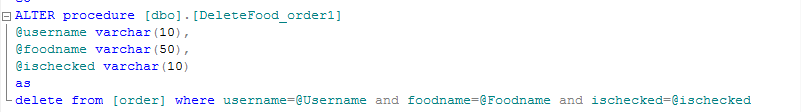


图19 DeleteFood\_order1的存储过程

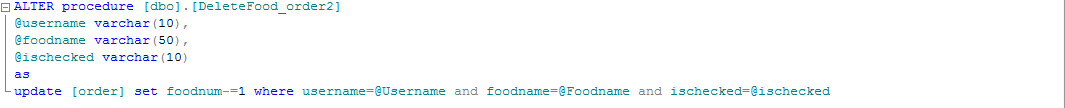


图20 DeleteFood\_order2的存储过程

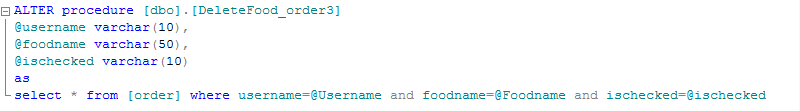


图21 DeleteFood\_order3的存储过程

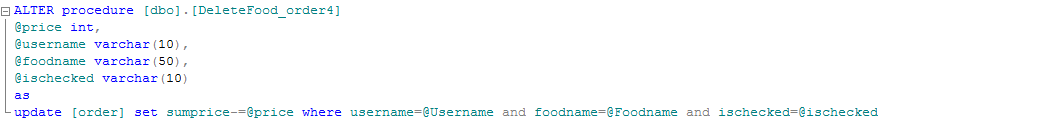


图22 DeleteFood\_order4的存储过程

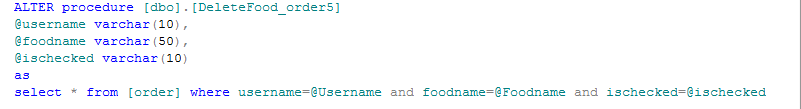


图23 DeleteFood\_order5的存储过程

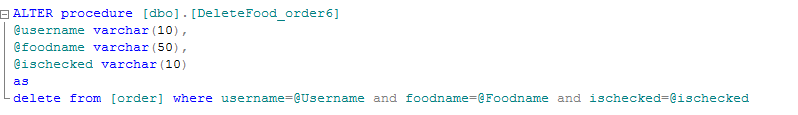


图24 DeleteFood\_order6的存储过程



图25 SearchNo的存储过程

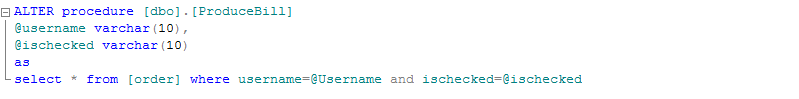


图26 ProduceBill的存储过程

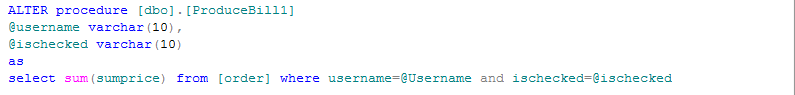


图27 ProduceBill1的存储过程

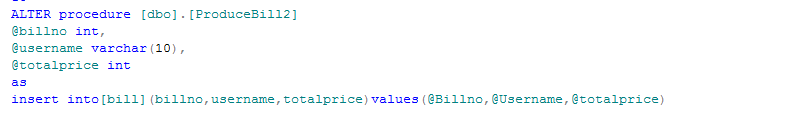


图28 ProduceBill2的存储过程

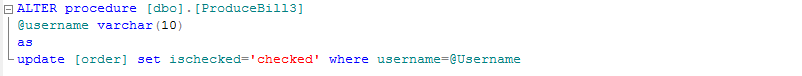


图29 ProduceBill3的存储过程

### 3.3.3 添加菜品——设计思路

* 设计思路：

点击Add按钮，激活事件protected void Add\_FoodClick(object sender, EventArgs e)，先从数据库的order表中查看有没有这道菜的记录，如果有，说明之前已经点过，那么跟新数据，foodnum+1,foodprice价格变动，而如果没有这道菜的记录，那么就在数据库中插入这道菜的记录。

### 3.3.4 删除菜品——设计思路

* 设计思路：

点击Delete按钮，激活事件protected void Delete\_FoodClick1(object sender, EventArgs e)，同样先从数据库的order表中查看有没有这道菜的记录，如果没有，那么弹出窗口，提示用户没有点这道菜，如果有，那么要先判断菜品数量（foodnum）是否为0，如果为0，删除这条记录，不为0，那么跟新数据，foodnum-1,foodprice价格变动，然后继续再order表里搜索这条记录，判断foodnum是否为0，如果为0，删除这条记录。

### 3.3.5 提交订单——设计思路

设计思路：

点击提交按钮，激活事件protected void saveOrder\_ServerClick(object sender, EventArgs e)，在web层生成一个6位的随机数作为账单号，然后在数据库的bill表里搜索这条记录，如果搜到了，那么这个账单号便不能使用，继续重新生成，直到bill表里搜索不到这条记录，接着插入账单信息。同时在order表里的菜品属性ischecked变为TRUE，表示这道菜已经结单。

## 3.4 管理员界面功能说明

### 3.4.1 功能说明

管理员登陆后的界面，左侧有一个导航栏，分为五个部分，基础信息，菜单信息，用餐信息，结单信息和每日营业额的情况。

### 3.4.2 界面设计

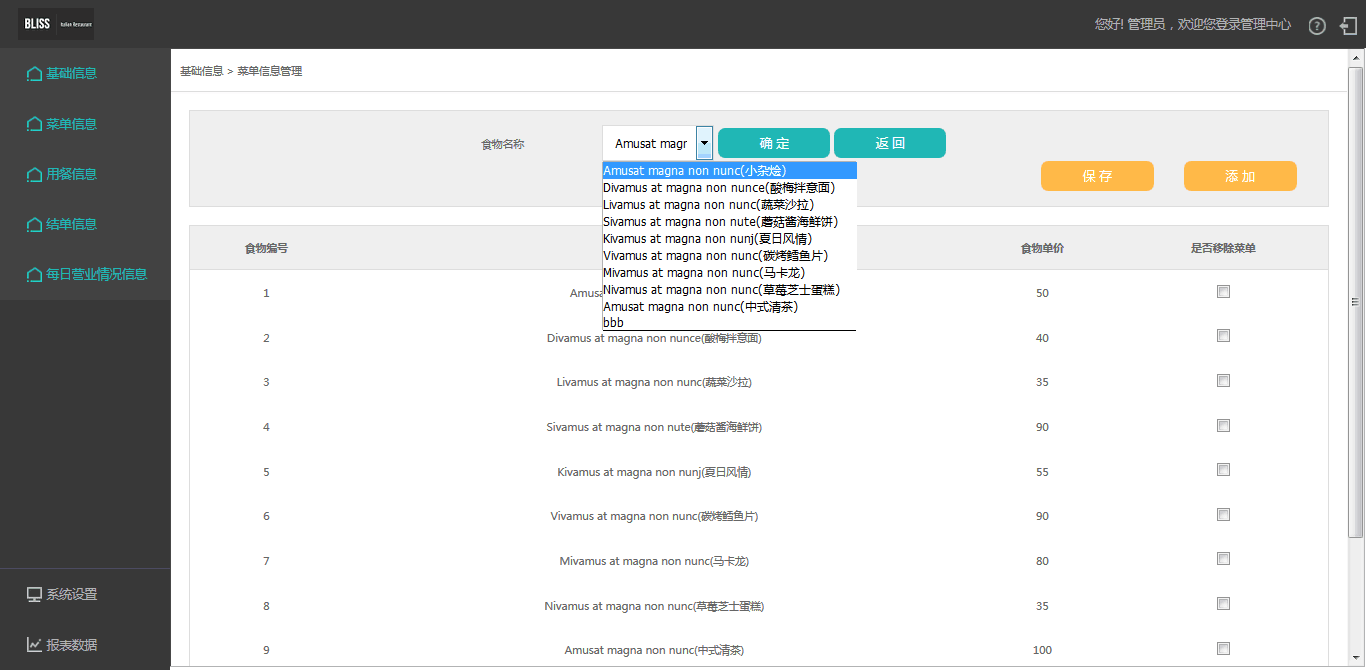


图30 管理员管理菜单

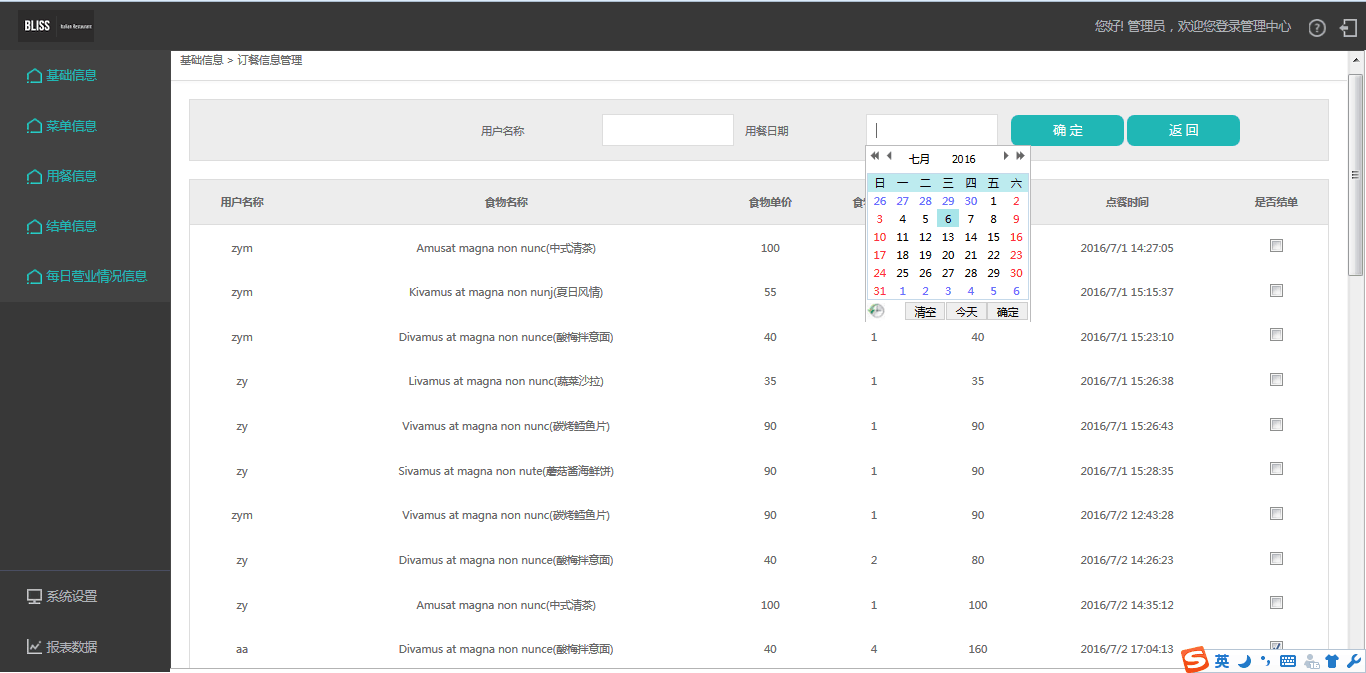


图31 管理员管理用餐信息

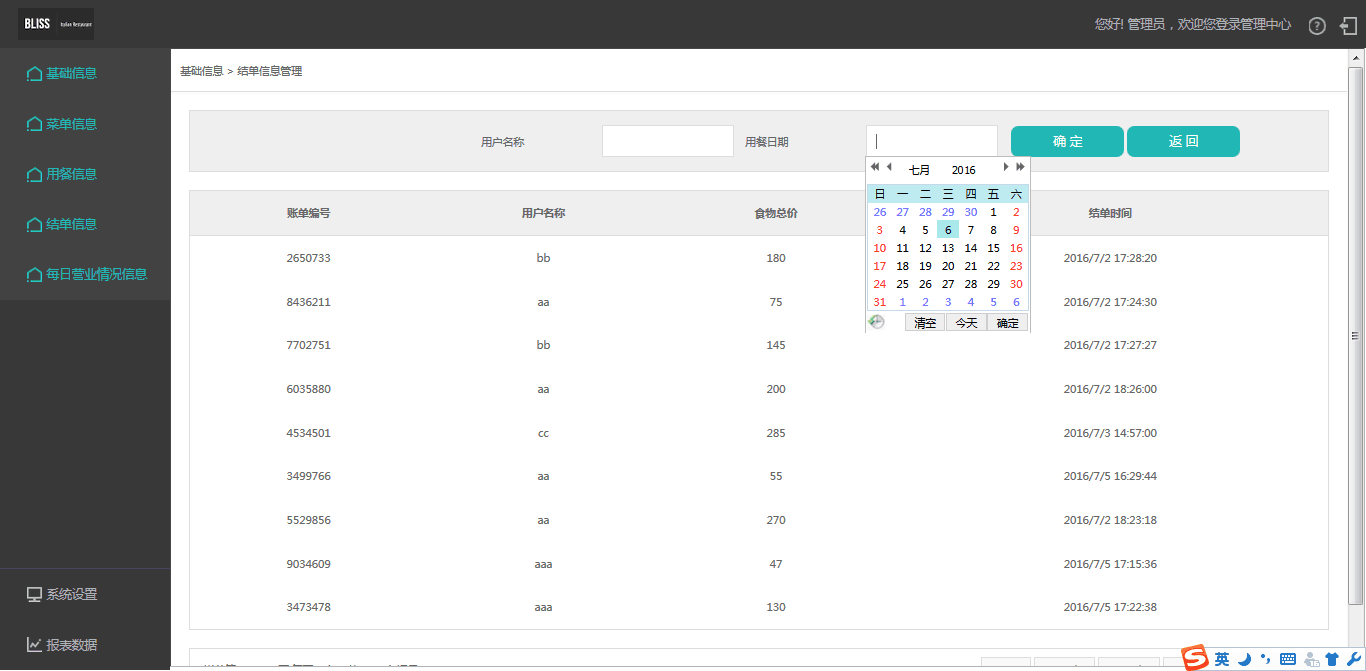


图32 管理员管理结单信息

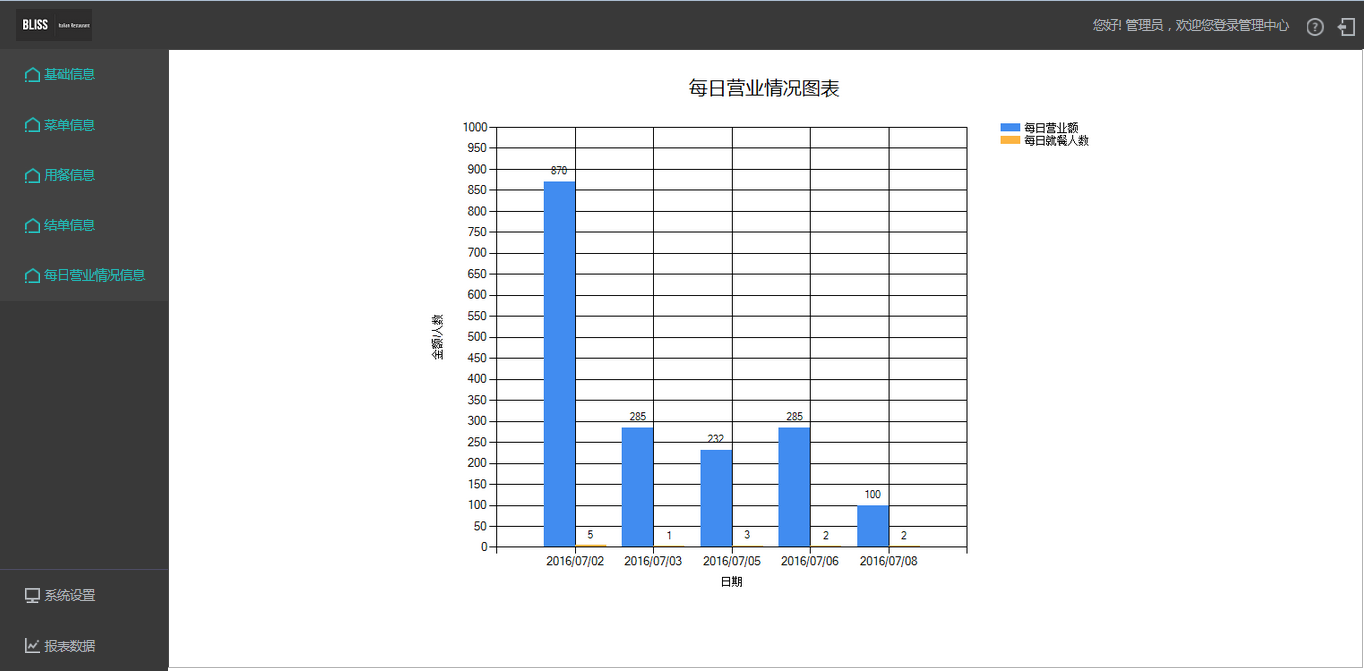


图33 每日营业情况一览



图34管理员添加食物

### 3.4.3 查询功能代码

**DAL：**

**//管理员查询相应的用户名和日期**

public DataTable searchPointedOrderInformation(Model.Order o,string date)

{

DataTable dt = new DataTable();

IDataParameter[] parameter = {

new SqlParameter("@username",SqlDbType.VarChar,10),

new SqlParameter("@date",SqlDbType.VarChar,200)

};

parameter[0].Value = o.Username;

parameter[1].Value = date;

dt = pr.GetDataTable("searchPointedOrderInformation", parameter);

//string SQL = "select \* from [order] where username='" + o.Username + "' and Convert(varchar(200),datetime,111)='" + date + "'";

//dt = db.SQLGetDataTable(SQL);

return dt;

}

**WEB：**

**//管理员按照条件搜索相关信息**

protected void search\_ServerClick(object sender, EventArgs e)

{

if (this.txtUserName.Text != "" && this.txtDateTime.Text != "")

{

string d = this.txtDateTime.Text;

string date = Convert.ToDateTime(d).ToString("yyyy/MM/dd");

o.Username = this.txtUserName.Text;

dt = order.searchPointedOrderInformation(o, date);

}

else if (this.txtUserName.Text == "" && this.txtDateTime.Text != "")

{

string d = this.txtDateTime.Text;

string date = Convert.ToDateTime(d).ToString("yyyy/MM/dd");

dt = order.search\_Date\_OrderInformation(date);

}

else if (this.txtUserName.Text != "" && this.txtDateTime.Text == "")

{

o.Username = this.txtUserName.Text;

dt = order.search\_User\_OrderInformation(o);

}

else

{

Response.Write("<script>alert('您未输入搜索条件!');</script>");

}

}

### 3.4.4 添加新菜品功能代码

**DAL：**

**//管理员添加新菜**

public bool addFood(Model.Menu m)

{

bool flag = false;

IDataParameter[] parameter = {

new SqlParameter("@foodno",SqlDbType.Int),

new SqlParameter("@foodname",SqlDbType.VarChar,50),

new SqlParameter("@foodprice",SqlDbType.Int),

new SqlParameter("@foodimage",SqlDbType.VarChar,50)

};

parameter[0].Value = m.Foodno;

parameter[1].Value = m.Foodname;

parameter[2].Value = m.Foodprice;

parameter[3].Value = m.FoodImage;

pr.ExcutePro("addFood", parameter);

//string SQL = "insert into [menu](foodno,foodname,foodprice,foodimage)values('"+m.Foodno+"','"+m.Foodname+"','"+m.Foodprice+"','"+m.FoodImage+"')";

//db.SQLExecute(SQL);

flag = true;

return flag;

}

**WEB:**

**//管理员添加新菜**

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

{

bool flag = false;

dt\_search\_menu = menu.searchMenu(m);

if (this.txtFoodname == null || this.txtFoodno == null || this.txtFoodprice == null)//信息不能为空

{

Response.Write("<script language='javascript'>alert('请把菜的信息填写完整！');</script>");

}

else

{

if (dt\_search\_menu.Rows.Count > 8)//即只能在菜单上显示9道菜

{

Response.Write("<script language='javascript'>alert('您的菜单数量已达上限，不可添加新菜！');</script>");

}

else

{

int foodno = int.Parse(this.txtFoodno.Text);

m.Foodno = foodno;

dt\_search\_no = menu.AddInfo(m);

if (dt\_search\_no.Rows.Count > 0)//编号不可以重复

{

this.Label6.Visible = true;

}

else

{

string fooname = this.txtFoodname.Text;

int foodprice = int.Parse(this.txtFoodprice.Text);

m.Foodname = fooname;

m.Foodprice = foodprice;

if (FileUpload1.HasFile)//是否上传文件

{

string savePath = Server.MapPath("~/images/");//指定上传文件在服务器上的保存路径

//检查服务器上是否存在这个物理路径，如果不存在则创建

if (!System.IO.Directory.Exists(savePath))

{

System.IO.Directory.CreateDirectory(savePath);

}

savePath = savePath + "\\" + FileUpload1.FileName;

FileUpload1.SaveAs(savePath);

m.FoodImage = "images" + "\\" + FileUpload1.FileName;

flag = menu.addFood(m);

if (flag)

{

Response.Write("<script language='javascript'>alert('" + m.Foodname + " 已加入到菜单中！');</script>");

Response.Write("<script language=javascript>javascript:location.href='MenuInformation.aspx'</script>");

}

else

Response.Write("<script language='javascript'>alert('添加失败！');</script>");

}

else

{

Response.Write("<script language='javascript'>alert('您还没有上传文件！');</script>");

}

}

}

}

}

### 3.4.5 删除菜品/用户信息功能代码

**DAL:**

**//管理员删除菜**

public void DeleteFood(Model.Menu m)

{

IDataParameter[] parameter = {

new SqlParameter("@foodno",SqlDbType.Int)

};

parameter[0].Value = m.Foodno;

pr.ExcutePro("DeleteFood", parameter);

}

**WEB:**

**<前端>**

<script type="text/javascript">

function submit() {

var table = document.getElementById("table");

var deleteData = new Array();

for (var i = 1; i < table.rows.length; i++) {

if (table.rows[i].cells[3].getElementsByTagName("INPUT")[0].checked) {

var data = {};

data.id = table.rows[i].cells[0].innerText;

deleteData.push(data);

}

}

$.ajax({

type: "get",

url: "ajaxManage2.aspx",

data: { json: JSON.stringify(deleteData) },

dateType: "json",

success: function (reg) {

var arr = reg.split('|');

var returnval = arr[0].toString();

alert(returnval);

window.location.href = "MenuInformation.aspx";

},

error: function () {

alert("error");

}

});

}

</script>

**<ajaxManagement.aspx>**

protected void Page\_Load(object sender, EventArgs e)

{

string json = Request.QueryString["json"].ToString();

BLL.UserBLL user = new BLL.UserBLL();

Model.User u = new Model.User();

if (json != null)

{

JArray ja = (JArray)JsonConvert.DeserializeObject(json);

string[] datas = new string[ja.Count];

for (int i = 0; i < ja.Count; i++)

{

datas[i] = ja[i]["id"].ToString();

u.Id=datas[i];

user.DeleteUser(u);

}

Response.Write("共删除"+datas.Length+"条数据"+ "|");

}

else {

Response.Write("error");

}

}