## M&M商城说明文档

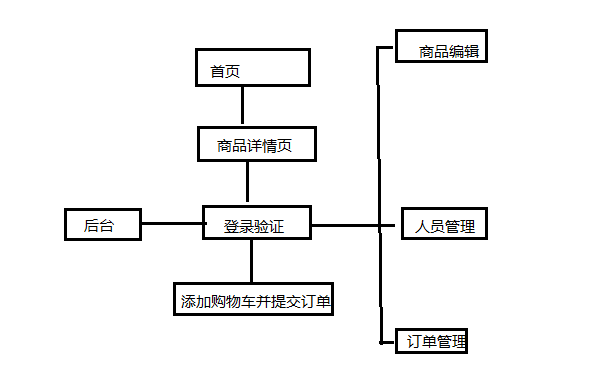
### 作品概述

1. 作品介绍：

M&M商城采用微软IIS体系，基于c#与.net开发，使用多种技术栈结合开发，前端技术采用基本的html5+css3+JavaScript，借助目前主流的Vue.js、Jquery、bookstrap、echarts等一些个人平常封装的js工具类。其中Vue主要用在数据的渲染和根据适合情况进行dom模块的渲染，主要用于购物车提交订单并付款页面以及后台人员的添加和删除。Jquery用于对dom对象的操作，bookstarp用于页面的布局，本项目主要用在登录界面上，响应式布局。Echarts用于数据的可视化，主要用于后台管理的数据可视化。后台采用c#+.net，数据库采用sqlserver，开发工具为visual studio。

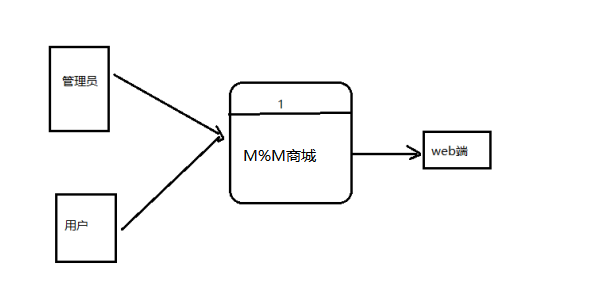
1. 开发设计：

系统的界面操作方便。层次图为：



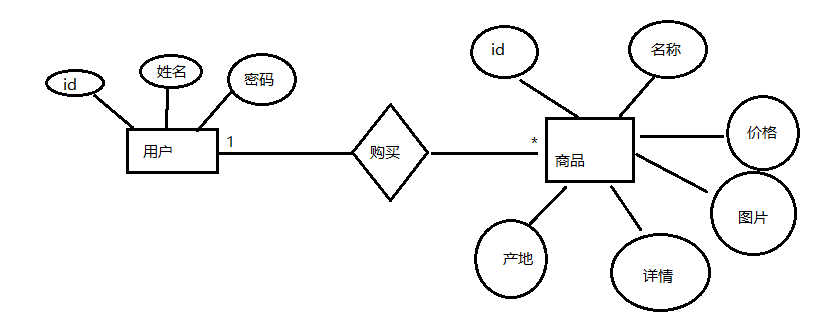
系统层次图

上层数据流图为：



上层数据流图

系统有两个实体，分别为用户和商品，其数据模型为：



数据模型：

数据库设计略。

1. 功能介绍

主要分为登录功能其中有图片验证功能，购物车功能其中有订单提交并付款功能，商品编辑功能其中包括商品的检索功能以及人员管理和订单管理功能。逻辑功能记录用户的登录状态。详细说明如下：

登录功能：此功能模块基于通过验证码进行验证前提下进行改进，其思路都是差不多，验证码主要是先定义一个验证码组成库存在数组里，然后根据数组长度以及结合Math.random()函数获取对用的数组小标生成一组随机的验证码，再进行渲染在页面上。把用户输入的验证码统一转化为大写，并验证。那图片验证也是同样的道理，在一个图片数组里存放一些图片的路径，然后设置图片旋转的误差范围，根据旋转判断是否成功。这里的成功和失败信息图标我采用阿里云矢量图标库，并生成base64文件。部分代码在附录中。

购物车功能：用户根据首页的商品点击后进入对应商品的详情页，有加入购物车按钮，点击后，会跳转到购物车页面，并携带对应的商品id，然后后台根据id值进行与数据库查询并将结果渲染在页面上。用户确认信息无误后，点击确认，然后提交订单，这时系统会根据vue设置的数据渲染用户信息的模块，用户输入信息后点击确认订单，系统又会再渲染输入密码模块。用户输入密码后系统会把这些模块进行消除或者是隐藏。然后在vue对象里面调用localtion.reload()。不过此方法对系统的性能消耗比较大。可以的话建议采用组件传值的方式。但是这里由于我的组件不多，所以我就用了这个方法。

商品编辑：商品编辑这里除了基本的CURD（增删改查）外，我增加了较简单的商品检索功能。通过模糊查询语句然后与数据库进行匹配，匹配到到进行渲染。核心代码为：select \* from list where name like ‘%TextBox1.text.toString()%’;

人员管理:在后台登录中，又记录用户登录的人员表，此功能与商品编辑的基本功能与实现思路也差不多。增加的是一些用vue渲染以及一些动画的显示。

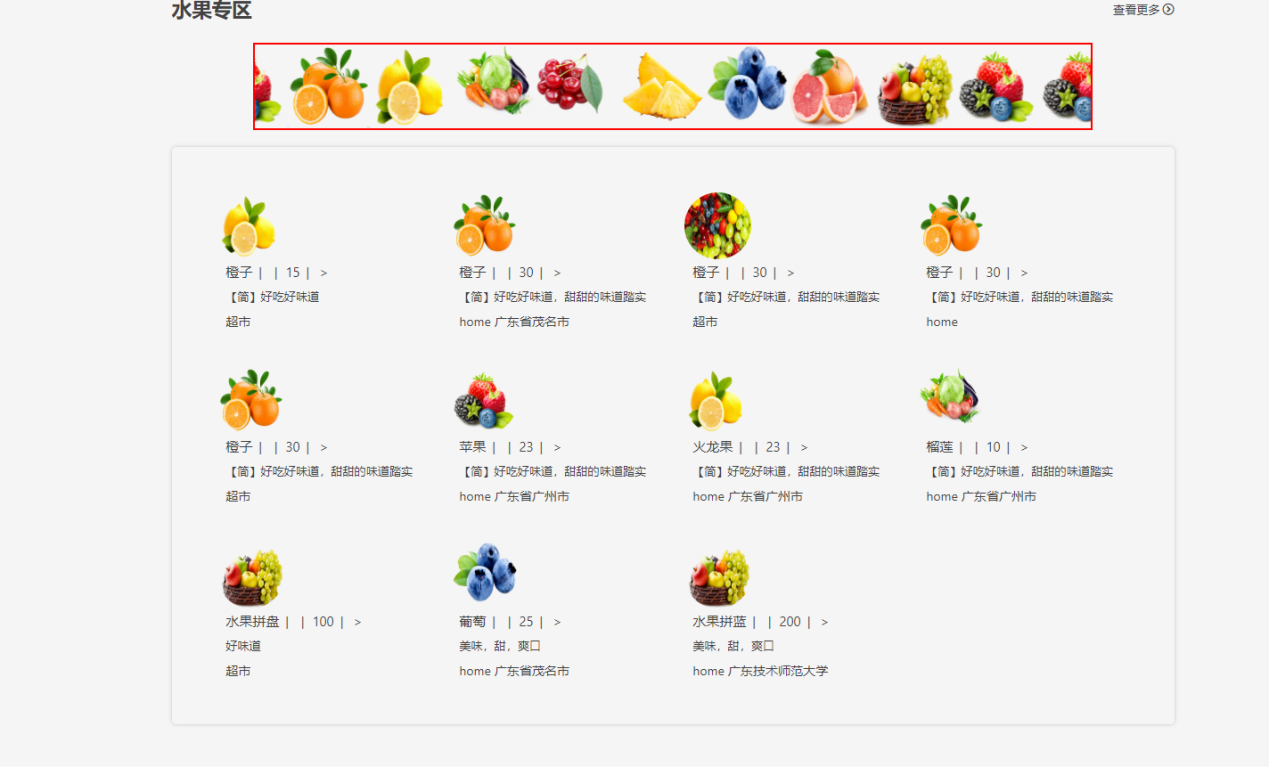
订单管理：用户提交订单并付款后，会将相关信息存入到card数据表中，然后在订单管理页面在根据gridView控件进行显示。此功能后期可以扩展记录订单的状态，如发货，以及配送信息。

逻辑功能：若直接打开后台管理系统页面会重定向到登录页面，此处是使用Session来实现。记录用户的登录状态。

### 作品页面截图

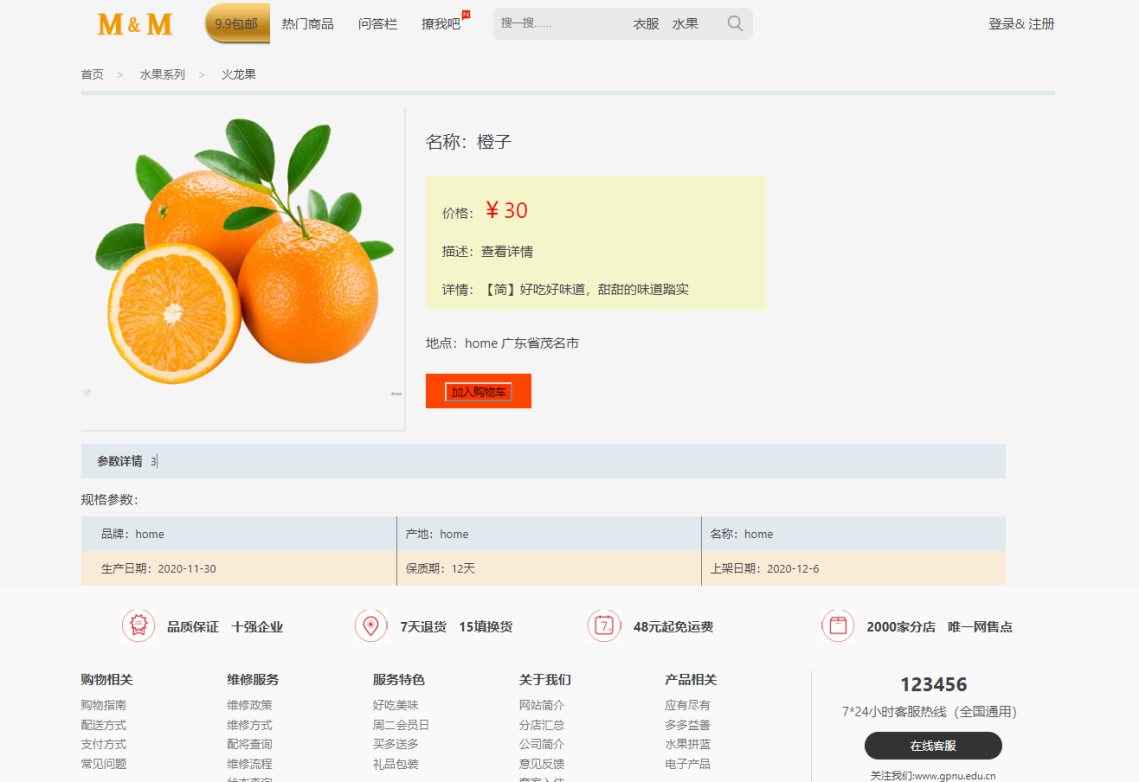
1. 首页：



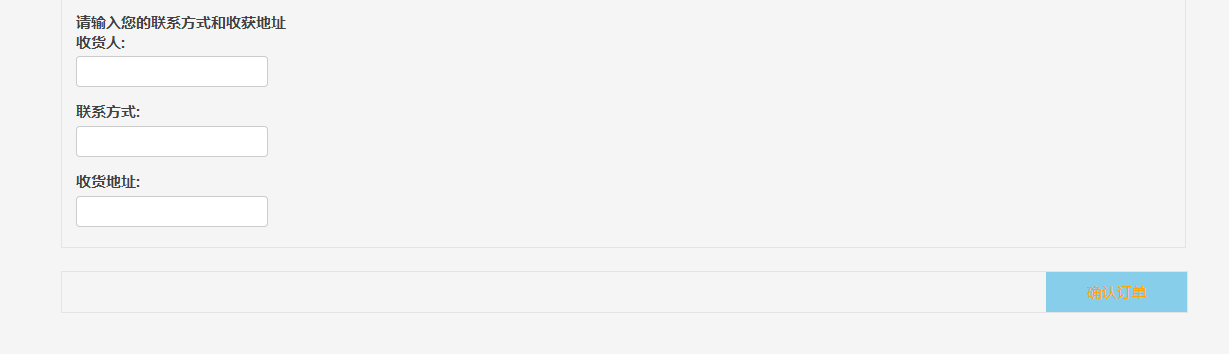




1. 商品详情页与购物车订单提交：

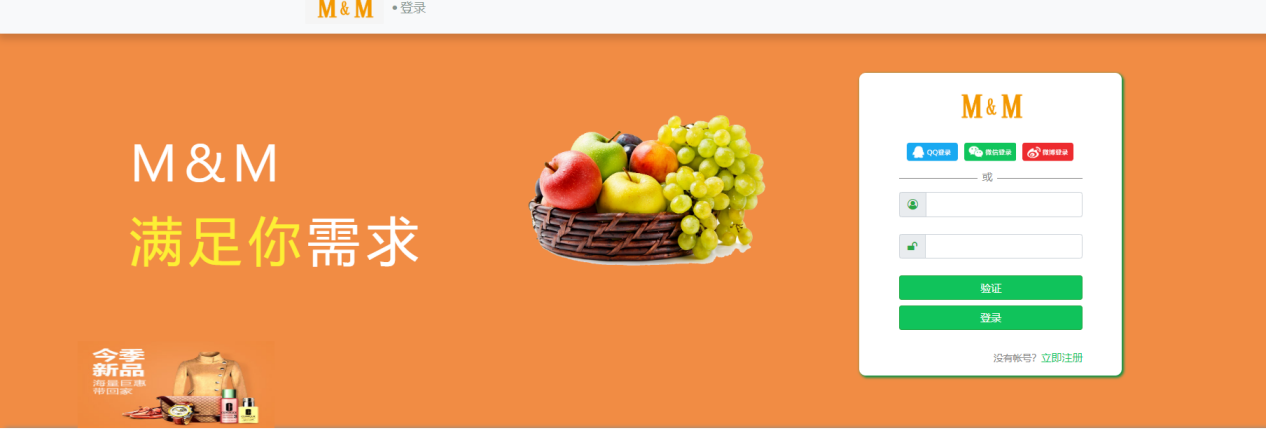


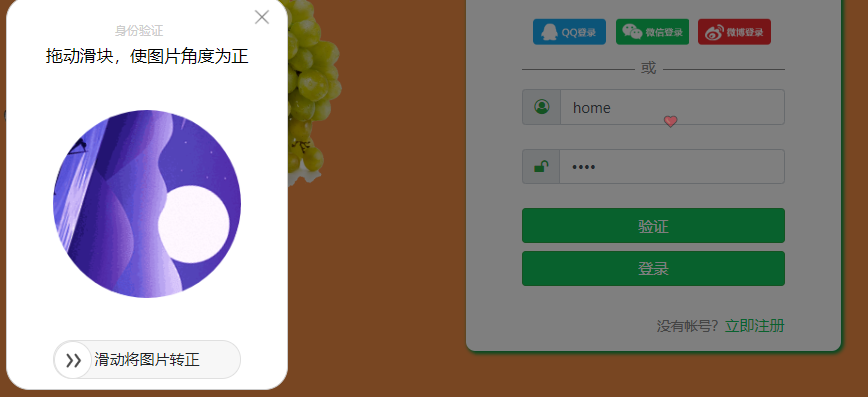


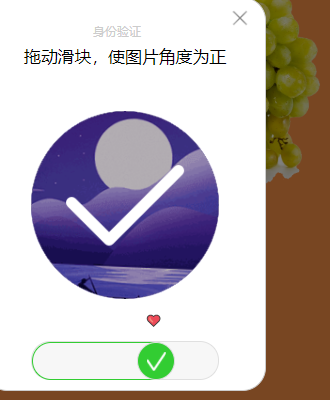




1. 登录







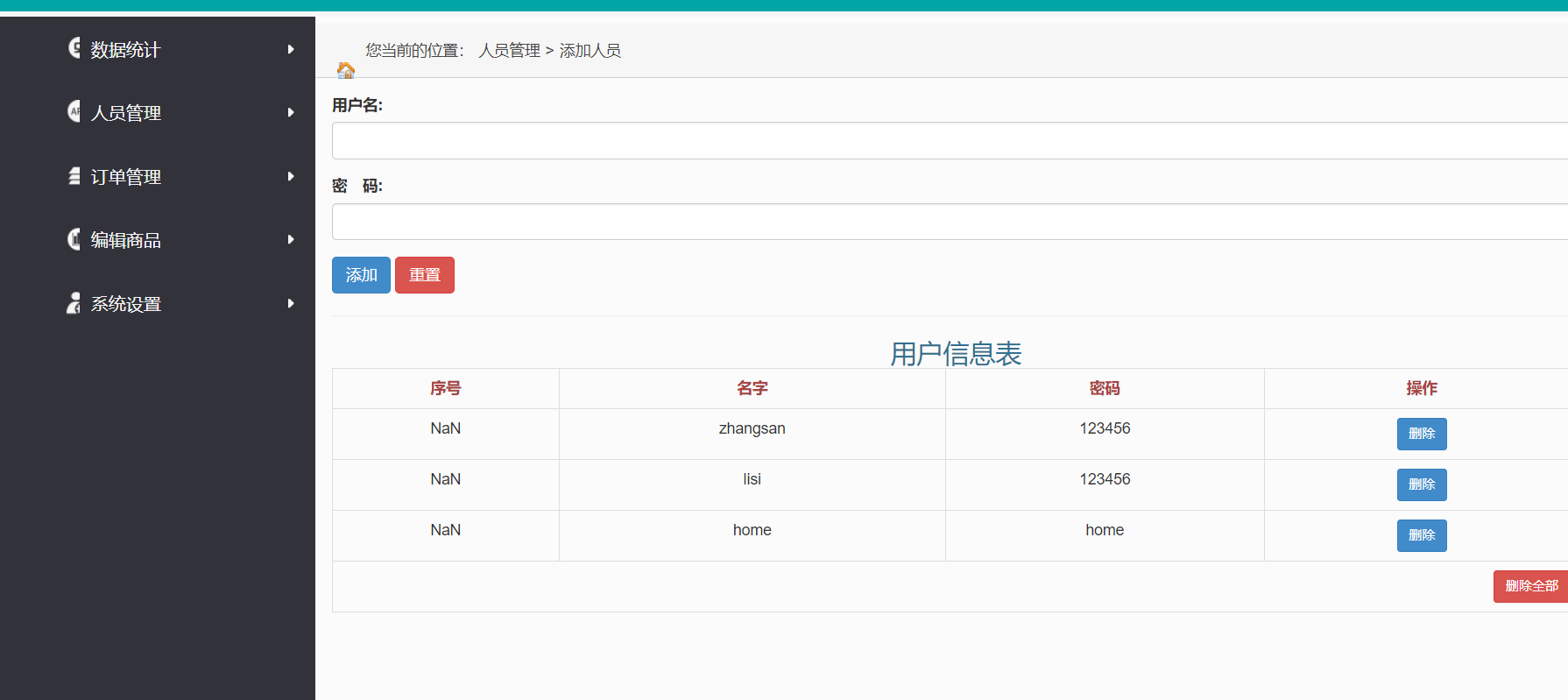
1. 后台管理页：



登录记录：



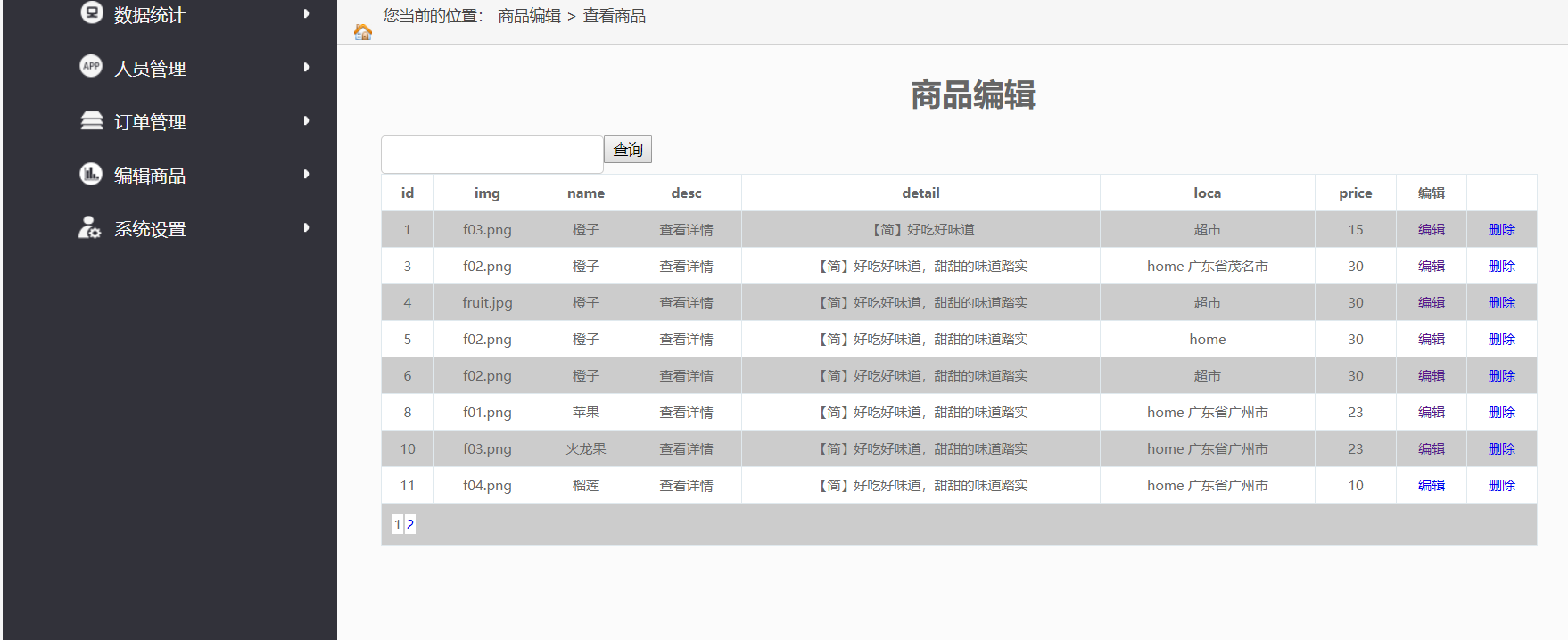
人员管理：



订单管理：



商品编辑：







### 学习与开发体会

本学期课程知识点让我印象深刻的是数据图，数据模型，数据加工，数据存储以及设计一个系统所具备的基本思路和实现步骤等，在这次的课程作业的设计中，我也运用了一点点这部分的知识体系。这次开发我接触到了visual studio这个工具，在之前我不怎么使用过它，项目开始的适合也不怎么习惯这个工具，编写前端页面适合过程出现经常出现一些不是自己想象出来的原型。随着后来我也慢慢习惯了，项目的什么代码基本都在这里直接写。也发现之前很多原因是因为浏览器的缓存做导致的。在本次开发中我加深了对整体页面布局的思路，更加灵活的使用flex布局和css3动画，以及通过Jquery来操作dom对象，进一步熟悉对原生JavaScript代码的理解。加深了对Vue.js的基本使用，特别是在数据渲染这方面。作为目前最主流的前端框架之一，vue的作用是非常大。在整个项目中我也植入了一个我用vue结合animate.js来实现一个todilist列表，这个是一个待做事项列表。在首页导航页面点击可以看到这个。本次开发使用的是微软体系，c#+.net，再在里面写前端代码。根据这个开发过程我觉得这个与javaweb的原理其实是差不多的。只是所依赖和运行的体系不一样。但是思维是差不多的。Javaweb是以java为后台代码编写，接口是在servlet这里编写，在里面编写前端代码，基于tomcat运行web项目。两者只是语言以及运行的体系不同，但是实现原理差不多。更让我觉得这些语言和框架是有互相参考与借鉴的。就比如es6的装饰器其实跟java的注解差不多一样。就比如本次中我做了一次模糊查询，这个在当时验收时候我还没做出来，主要是因为我进行模糊查询后，后端给我返回的是一个集合。我当时在页面的cs页面上新建一个List集合，也导入了相关命名空间（在java中说是包），但是是在泛型这里报错。因为我是根据之前javaweb一样，写了List<shop> list = new List<shop>，之所以报错是因为我的工程中没有shop这个类或者说是命名空间。一时间我不知道怎么在.net工程上建一个类型java中一个的JavaBean类和vo层。后来我逐渐了解到c#工程中的.net core这个控制台应用，它封装了一些库。其实与javaweb是差不多的，包括实现的原理也基本一样。这个可以说是在整个开发过程中体会比较深刻的。在本次开发中我也认识到了c#中控件的使用，特别是在其怎么添加样式以及连接数据库后，对布局的影响这些。慢慢的也是熟悉怎么在对它进行一些美化，如在控件属性上添加cssClass，然后在外部实现这个css类就行了，与html不同，一般html上我们直接设置class就行。然后通过数据源连接数据库并进行渲染时候，它会受内部原有的布局的限制，因为控件有自己的属性，这是我们想到的第一个方法应该是根据它的属性来对它进行修改。

本次开发让我体会了以visual studio为基础的C#项目，扩展了本身的知识层面。也熟悉结合多种框架技术来进行项目的开发。

### 附录：

验证码验证部分代码：

function change(){

// 验证码组成库

var arrays=new Array(

'1','2','3','4','5','6','7','8','9','0',

'a','b','c','d','e','f','g','h','i','j',

'k','l','m','n','o','p','q','r','s','t',

'u','v','w','x','y','z',

'A','B','C','D','E','F','G','H','I','J',

'K','L','M','N','O','P','Q','R','S','T',

'U','V','W','X','Y','Z'

);

// 重新初始化验证码

codes ='';

// 随机从数组中获取四个元素组成验证码

for(var i = 0; i<4; i++){

// 随机获取一个数组的下标

var r = parseInt(Math.random()\*arrays.length);

codes += arrays[r];

}

// 验证码添加到input里

code.value = codes;

}

change();//加载显示在页面上

code.onclick = change;//单击更换验证码

//单击验证

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

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

check.onclick=function(){

var inputCode = input.value.toUpperCase(); //取得输入的验证码并转化为大写

if(inputCode.length==0) { //若输入的验证码长度为0

alert("请输入验证码！"); //则弹出请输入验证码

}

else if(inputCode!=codes.toUpperCase()) { //若输入的验证码与产生的验证码不一致时

alert("验证码输入错误！请重新输入"); //则弹出验证码输入错误

change();//刷新验证码

input.value="";//清空文本框

}

else{ //输入正确时

alert("输入正确"); //弹出输入正确

}

}

图片验证部分代码：

//获取随机图片

function getRandomImg(imgArr) {

return imgArr[getRandomNumber(0,imgArr.length-1)]

}

//判断 是否处于动画状态

function ifAnimate(ele) {

if(ele.is(":animated")){

return true

}else{

return false

}

}

var myRotateVerify = new RotateVerify('#rotateWrap1', {

initText: '滑动将图片转正', //默认

slideImage: ['./images/1.png',

'./images/2.png',

'./images/4.png',

'./images/5.png'

], //arr [imgsrc1,imgsrc2] 或者str 'imgsrc1'

slideAreaNum: 10, // 误差范围角度 +- 10(默认)

getSuccessState: function (res) { //验证通过 返回 true;

if (res) {

setTimeout(function () {

$(".box-bg").css("display", "none");

}, 800);

//验证通过

imgVerify = true;

}

}

})

购物车订单提交部分代码：

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

{

string ConnStr = System.Configuration.ConfigurationManager.ConnectionStrings["listConnectionString"].ToString();

string sql = "select \* from shop\_list where id = " + Request["card\_id"].ToString();

SqlConnection conn = new SqlConnection(ConnStr);

conn.Open();

SqlCommand comm = new SqlCommand(sql, conn);

SqlDataReader dr = comm.ExecuteReader();

if (dr.Read())

{

Label1.Text = dr["name"].ToString();

Label2.Text = dr["price"].ToString();

Label3.Text = dr["price"].ToString();

Label4.Text = dr["price"].ToString();

Label5.Text = dr["price"].ToString();

Image1.ImageUrl = "imgs/" + dr["img"].ToString();

}

conn.Close();

}

protected void Button1\_Click(object sender, EventArgs e)

{

string ConnStr = System.Configuration.ConfigurationManager.ConnectionStrings["listConnectionString"].ToString();

String sql = "insert into car(cname,cprice,amount,sum,ctime) values ('" + Label1.Text + "','" + Label2.Text + "','" + Label6.Text + "','" + Label3.Text + "','" + DateTime.Now.ToString() + "')";

SqlConnection conn = new SqlConnection(ConnStr);

conn.Open();

SqlCommand sqlcmd = new SqlCommand(sql, conn);//构造一个sql命令sqlcmd.ExecuteNonQuery();

//关闭数据库

conn.Close();

//Response.Write("ok");

}

<script src="js/vue.js"></script>

<script>

let vm = new Vue({

el: '#app',

data: {

score: 90,

isShow: false,

isShow2: false,

},

methods: {

showAdress() {

console.log("点击了订单");

this.isShow = true;

},

showPaid() {

let name = $('#name').val();

let phone = $('#phone').val();

let adress = $('#adress').val();

console.log(name);

if (name == '' || phone == '' || adress == '') {

alert("请输入收获信息")

} else {

console.log("确认订单")

this.isShow2 = true;

}

},

paid() {

let pwd = $('#paidPas').val();

if (pwd == '') {

alert("请输入支付密码");

} else if (pwd != 'home') {

alert("密码错误");

} else {

alert("支付成功！")

this.isShow = false;

this.isShow2 = false;

}

}

}

})

</script>