订餐网站的设计与实现

摘 要

随着传统的购物方式正在向网络化发展，出现了“网购”这种越来越受欢迎的新型消费方式。为了提高菜品业务水平和效率，实现比电话订购更有效的送餐业务，网上订餐系统应运而生。通过对订餐流程和业务进行全面考察分析，结合目前最先进的软件开发技术，实现了订餐的网络化管理。目前，网上订餐系统还处于形成期，成长空间还很大。

该毕业设计采用ASP.NET的三层架构模型，基于Browser/Server结构，在Visual Studio开发工具下，把C#作为开发语言，SQL Server作为数据库，按照软件工程的方法开发订餐网站。系统主要分为前台功能模块和后台管理模块，其中前台功能模块主要包括用户注册、评论菜品、查询菜品、在线订餐、提交订单、在线留言、个人信息管理模块；后台管理模块包括系统用户管理、菜品类别管理、菜品信息管理、订餐记录管理、系统管理、评论管理模块。该系统与传统电话订餐相比，将广大消费者、菜品信息和商家紧密联系在一起，有效缩短了消费者订餐的时间，大大提高了工作效率。经过系统测试，达到了预期目标。

关键字：订餐网站；ASP.NET；三层架构；数据库；

**The design and implementation of reservation website**

**ABSTRACT**

With the traditional way of shopping is to network development, the "cyber" this new consumption patterns more and more popular. In order to improve the food business level and efficiency, to achieve more effective than the phone order room service, online reservation system arises at the historic moment. Through the analysis of the order process and business to conduct a comprehensive inspection, combining the most advanced software development technology, realized the order of network management. At present, the online reservation system is still in the formative years, the growth space is very big.

The graduation design adopts three layer architecture model of ASP.NET, based on Browser/Server structure, under the Visual Studio development tool, use c # as development language, SQL Server as the database, according to the method of software engineering development reservation website. System is mainly divided into front desk function module and the background management module, front desk function module mainly include user registration, review items, query dishes, online order, submit orders, online messages, personal information management module; Background management module includes the system user management, food category management, food information management, order management, system management, comments management module. Compared with the traditional telephone order a meal, the system will be closely linked to the broad consumer, food information and merchants together, effectively reduces the consumers to order a meal time, greatly improving the work efficiency. After the system test, the desired goal.

**Key words:** Reservation website; ASP.NET; Three layer architecture; Database;

目 录

[摘 要 1](#_Toc476641106)

[目 录 3](#_Toc476641107)

[1　　引言 1](#_Toc476641108)

[1.1　课题的背景 1](#_Toc476641109)

[1.2　开发意义 1](#_Toc476641110)

[1.3　系统设计目标 2](#_Toc476641111)

[1.4　论文构成及研究内容 2](#_Toc476641112)

[2　　系统分析 1](#_Toc476641113)

[2.1　可行性研究 1](#_Toc476641114)

[2.1.1　技术可行性 1](#_Toc476641115)

[2.1.2　经济可行性 1](#_Toc476641116)

[2.1.3　操作可行性 1](#_Toc476641117)

[2.1.4　法律可行性 1](#_Toc476641118)

[2.2　需求分析 1](#_Toc476641119)

[2.2.1 系统用户需求 2](#_Toc476641120)

[2.2.2　系统业务流程分析 2](#_Toc476641121)

[2.2.3　系统功能需求 3](#_Toc476641122)

[2.2.4　数据流分析 4](#_Toc476641123)

[3　　系统总体设计 7](#_Toc476641124)

[3.1　总体设计的原理 7](#_Toc476641125)

[3.2　系统的功能层次结构图 7](#_Toc476641126)

[3.3　数据库设计 8](#_Toc476641127)

[3.3.1　实体E-R图 9](#_Toc476641128)

[3.3.2　数据库的逻辑结构 10](#_Toc476641129)

[4　　详细设计 1](#_Toc476641130)

[4.1　数据库的连接 1](#_Toc476641131)

[4.2　系统设计流程 1](#_Toc476641132)

[5　　开发工具及相关技术介绍 1](#_Toc476641133)

[5.1　开发工具介绍 1](#_Toc476641134)

[5.1.1　Visual Studio介绍 1](#_Toc476641135)

[5.1.2　SQL Server介绍 2](#_Toc476641136)

[5.1.3　IIS介绍 2](#_Toc476641137)

[5.2　开发模式B/S介绍 2](#_Toc476641138)

[5.3　开发技术介绍 3](#_Toc476641139)

[5.3.1　ASP.NET介绍 3](#_Toc476641140)

[5.3.2　ADO.NET介绍 3](#_Toc476641141)

[6　　系统实现 4](#_Toc476641142)

[6.1　前台功能模块的实现 4](#_Toc476641143)

[6.1.1　留言主页模块 4](#_Toc476641144)

[6.1.2　用户注册模块 5](#_Toc476641145)

[6.1.3　用户登录模块 6](#_Toc476641146)

[6.1.4　个人信息管理模块 8](#_Toc476641147)

[6.1.5　在线留言模块 8](#_Toc476641148)

[6.1.6　菜品评论模块 9](#_Toc476641149)

[6.1.7　在线订餐模块 10](#_Toc476641150)

[6.2　后台管理模块的实现 11](#_Toc476641151)

[6.2.1　后台登录模块 11](#_Toc476641152)

[6.2.2　菜品类别管理模块 12](#_Toc476641153)

[6.2.3　用户管理模块 13](#_Toc476641154)

[6.2.4　评论管理模块 13](#_Toc476641155)

[6.2.5　菜品管理模块 14](#_Toc476641156)

[6.2.6 入库管理模块 15](#_Toc476641157)

[6.2.7　数据备份 16](#_Toc476641158)

[7 系统测试 1](#_Toc476641159)

[6.1 测试用例 1](#_Toc476641160)

[6.2 测试结果分析 3](#_Toc476641161)

[8　　结论 1](#_Toc476641162)

[参考文献 2](#_Toc476641163)

[致　　谢 3](#_Toc476641164)

1　　引言

1.1　课题的背景

随着时代的进步，上网的人数越来越多，网上订餐是人们的一大生活方式之一，订餐网站的完善合便捷将是现代社会必然的需求。本系统的设计目的是为了满足消费着只要通过互联网就可以足不出户的订购自己喜欢的美食。改变传统商业交易，在互联网上进行交易，实现网上购买美食。在我国，网上订餐最早出现在 20世界 90年代末期，98年代广州等大城市已出现网上订餐业务。2000年7月，北京丽华快餐公司为了深索餐料业网路经营的道路。寻求新的经济增长点，也推出了网上订餐，成为深索网上订餐的先驱。目前网上订餐已逐渐普及特别是在快餐业发达的东部地区。本系统的设计目的是为了满足消费者只要通过互联网就可以足不出户的订购自己喜欢的美食。改变传统商业交易，在互联网上进行交易，实现网上购买美食。

1.2　开发意义

在互联网上所有菜品的分类信息中，食品类的菜品信息无疑是最丰富的，一大批订餐类网站从中国互联网诞生初期就开始为网友提供大量一手信息。以美团外卖和饿了么外卖为例，如此庞大并仍在持续增长的信息量远非其他菜品类别能及，极大地满足了用户对菜品信息的需求。只需动动鼠标即可从资讯页面跳转到订餐页面的便捷网购模式，使得信息收集与订餐行为之间无缝对接。

随着近年来资讯网络媒体与网络订餐平台的合作加深，消费者从获取资讯到发生订餐行为的时空距离被进一步缩短，由此带来的巨大经济效益使得各大销售平台越来越重视餐饮销售。

与传统方式销售相比在线销售有很多的优点如：

1．检索便捷

在线销售提供了多种检索途径，可以从分类、新品、特价等途径进行检索，快捷准确。

2．信息量大

与传统销售方式相比，在线销售能够提供海量菜品信息。

3．成本低，风险小，无地域限制

网络销售最突出的优点是：不再限制消费者的购买时段，扩大和巩固了客户群，从而增加了商家的核心竞争力，节省实际开店时需要投入的成本和租用费用。

订餐的主要目的就是通过网站来推广互联商家的菜品和技术服务，并使客户随时可以了解商家和商家的菜品，为客户提供在线服务和订单处理功能。从长期的战略目标来说，网站不仅仅是菜品和服务的推广，而是通过Internet、商家内部网（Intranet）和商家外部网（Extranet），将买家与卖家、商家和合作伙伴紧密结合在了一起，因而消除了时间与空间带来的障碍。

1.3　系统设计目标

订餐网站为人们生活中提供了方便快捷的方式：

1.顾客去餐厅不用需要排队订餐，可以方便的打开我们的网站，选择自己喜欢的餐厅，浏览餐厅的菜单，进行网上订餐，省去了很多的时间。

2.对于餐厅，可以扩大自己的消费领域，通过我们的网站可以很好的做广告。每个餐厅有他自己的主页，管理自己的主页；餐厅不仅仅在某个商业小区经营，餐厅还可以给顾客提供送外卖服务，就这样大大的方便了那些宅男宅女们，不用再买那些快餐似的垃圾食品，还能很好地享受到可口的饭菜。通过这种方式，餐厅的营业范围将大大地提高，营业额必然会上升。

本系统针对所要用户群体，建立了客户端及服务器端。主要解决的问题包括：

（1）对于使用界面，需要人性化，能够给用户一种温馨与简易的感觉；

（2）系统应能够及时响应用户的请求，并及时提供帮助。

本系统开发的意义在提供给用户方便快速的订餐服务，满足用户的需要，解决传统订餐方式的不足。

1.4　论文构成及研究内容

本论文将按照软件工程的方法，首先做需求分析，然后是概要设计，设计出整个系统的功能模块和数据流图，最后是详细的页面设计和编码。详细来说，第一章对订餐网站进行了简要的介绍；第二章对订餐网站进行整体分析；第三章对系统做出总体设计；第四章对&订餐网站做详细设计；第五章是开发工具介绍；第六章是系统实现。

2　　系统分析

2.1　可行性研究

可行性研究是整个项目规划中的第一步，我们需要综合考虑系统实施过程中可能遇到的各种问题，确定系统是否有实施的必要。本系统是在对美团外卖网、饿了么、百度外卖网等较成熟的在线订餐网站进行调研和体验之后进行设计的，因此具有较大的针对性。下面将就系统的技术可行性、经济可行性、操作可能性和法律可行性进行具体分析。

2.1.1　技术可行性

技术可行性是对现有Web技术和硬件条件进行分析，确定在技术层面上能否实现本系统。本系统的硬件条件要求不高，现有计算机的硬件标准基本都能满足系统开发和运行。在软件层面上，本课题拟在Windows 8系统下使用微软公司的Visual Studio 2012开发平台和SQL Server 2012，按照软件工程的方法开发整个系统。Browser/Server结构可以完成用户在浏览器中的阅读；SQL Server的图形化界面和SQL语句的结合使用令数据操作变得更加便利；C#语言可以使程序员快速高效地编写应用程序；ASP.NET技术将脚本语言和高级程序设计语言以及数据库结合，使得留言可通过Web程序动态管理。因而具有技术可行性。

2.1.2　经济可行性

对系统从开发者的经济状况和用户的使用费用进行分析。本系统使用的Visual Studio、SQL Server都可以从网上免费下载，而计算机是已经具备的装置，开发者不需要在系统开发上花费相关费用。对于使用者来说，本课题在服务器端采用.NET4.0框架平台，在客户端测试IE、Google Chrome、火狐浏览器等页面CSS兼容性，这样使得系统的兼容性和版本更新以及系统维护变得更加简便，为使用者减少了相关费用。从经济角度讲可行性较高。

2.1.3　操作可行性

本订餐网站采用B/S模式，无需另外安装别的软件，使用者很容易掌握并熟练使用。因为本系统就是web应用程序，只需要安装好iis服务器和数据库，导入数据库后，既可以在iis服务器部署本系统，然后通过浏览器访问服务器地址加上端口号和系统名称既可访问本系统，系统实施成功以后也会大大减少管理用户订餐的时间，容易得到推广使用，因此具有操作可行性。

2.1.4　法律可行性

本系统作为本科生毕业设计，不用于商业目的，不侵犯专利也不会妨碍他人，所使用的开发软件也都是免费版本的，符合国家法律规定，具有法律可行性。

通过以上四方面的可行性分析，我们可以确定订餐网站是可行的。

2.2　需求分析

接下来将从系统用户需求、系统业务流程分析、系统功能需求和数据流分析几方面进行需求分析。

2.2.1 系统用户需求

本系统的用户总共有系统超级管理员、普通管理、普通注册用户，不同用户有不同的权限。

管理员需要通过登录进入后台管理界面，实现对菜品和菜品类别的管理和发布，未经审核和审核不通过的都不可以显示在前台界面；管理员可以管理系统用户和注册用户，包括添加、删除、修改、查询；并可以实现菜品类别管理、留言管理、评论管理和系统管理；普通用户可以在网站前台进行账号注册和账号登录，然后浏览网站信息和评论菜品、查询菜品信息、在线添加留言，普通用户登录个人后台进行个人资料修改、登录密码修改等功能操作。

对系统用户进行分类之后，就可以分别对各类用户进行需求分析。这样可以保证系统的需求分析更加完善，有利于之后的系统设计。

2.2.2　系统业务流程分析

对系统用户进行分类之后，可通过系统业务流程分析对需求分析进一步细化，将前台功能模块和后台管理模块的工作流程描述出来，从而使用户和开发者之间更好的沟通，而系统业务流程图就是一个很好的交流工具。

订餐网站的总业务流程如图2-1所示：



图2-1　　订餐网站的总业务流程图

普通用户登录成功后的业务流程如图2-2所示：



图2-2　　普通用户业务流程图

后台管理界面的业务流程如图2-3所示：



图2-3　　后台用户业务流程图

2.2.3　系统功能需求

下面进行系统的功能需求分析。

1. 系统用户管理：不管是超级管理员还是普通管理员都需要管理系统用户，包括普通管理员的添加、删除、修改、查询，修改管理员的登录密码，新添加的管理员用户可以登录系统。
2. 注册用户管理：游客前台首页注册后，经过管理员后台，不管是超级管理员还是普通管理员都需要管理注册用户，包括注册用户审核、删除、修改、查询，审核通过后，注册用户既可以通过账号和密码登录系统。
3. 修改密码：系统所有用户（管理员和注册用户）应该都要能修改自己的登录密码，修改后需要重新登录。
4. 系统简介设置：系统管理员应该可以通过系统简介设置功能设置网站前台的系统简介信息，网站前台的系统简介是随后台的变化而变化的，系统简介应该使用编辑器，实现图片，文字，列表，样式等多功能输入。
5. 系统公告设置：系统管理员应该可以通过系统公告设置功能设置网站前台的系统公告信息，网站前台的系统公告是随后台的变化而变化的，系统公告应该使用编辑器，实现图片，文字，列表，样式等多功能输入。
6. 数据备份：数据备份功能应该只有超级管理员才能使用，数据备份将系统的数据库的数据生成保存在.dat的文件中，当数据丢失时，可以通过运行.dat文件恢复数据库数据。
7. 在线留言：只有注册用户才能进行在线留言，需要登录网站前台，输入留言内容既可以完成在线留言功能，同时可以查看别的用户的留言。
8. 留言管理：由超级管理员和普通管理员使用，对系统所有的留言信息进行回复、删除、修改、查询。
9. 个人资料管理：由注册用户使用，注册用户登录个人后台，可以修改个人当初的注册信息，如修改电话号码、邮箱等，用户的用户名是无法修改的。
10. 友情链接管理：由超级管理员和普通管理员使用，对系统所有的友情链接信息进行添加、删除、修改、查询，同时网站前台的友情链接随着后台的友情链接变化而更新。
11. 批量删除：由超级管理员和普通管理员使用，对系统所有的表行清空数据，只需要选择需要清空数据表，即可一键批量清空数据
12. 在线评论：注册用户在网站前台登录个人账号后，可以选择某个菜品，进行评论，查看别人的评论信息。
13. 评论管理：在网站后台，管理员进行菜品的评论管理，每条菜品对应自己的评论，管理员可以删除、修改、查询。
14. 在线订餐：注册用户登录网站前台后，可以选择菜品进行订餐，添加订餐记录。
15. 订餐订单管理：管理员管理所有用户的订餐记录，对其进行审核、删除、修改。
16. 菜品信息管理：系统需要一个可以管理菜品的功能，包括添加、删除、修改、查询，同时网站前台菜品信息要跟着网站后台的菜品信息同步更新。
17. 菜品类别管理：在网站后台，管理员用户应该要实现对菜品类别的操作，包括添加、删除、修改、查询。

2.2.4　数据流分析

根据系统用户需求和功能需求，得知订餐网站的总体数据流图如下



图2-4　　订餐网站数据流图

用户登录数据流图如下



图2-5　　用户登录数据流图

普通用户登录成功后的数据流图如下



图2-8　　普通用户数据流图

后台管理员登录成功后的数据流图如下



图2-9　　后台管理员数据流图

3　　系统总体设计

3.1　总体设计的原理

总体设计原理包括：

（1）模块化。模块化就是把整个程序划分为多个模块，每个模块有自己的名字并可以被独立访问。所有模块的功能集合起来就可以满足用户需求。

（2）抽象。抽象是暂时忽略软件的细节，集中精力将软件的本质特征提取出来。

（3）逐步求精。总体设计需要先解决主要问题，再将问题一步步细化。

（4）信息隐藏和局部化。设计和确定模块时，使得一个模块内包含的信息对于不需要这些信息的模块来说，是不能访问的；并且应该在相近的物理地址上放置关系较密切的软件元素。

（5）模块独立。模块之间应该遵循高内聚，低耦合的原则。

3.2　系统的功能层次结构图



图3-1　　订餐网站的功能结构层次图

前台功能模块：可以在前台Web浏览器中查看这些功能。

用户注册模块：将用户信息记录到用户表

用户登录模块：用来区分三种用户，普通用户、普通管理员和超级管理员。

个人信息管理：用户登录后可以修改用户表中的个人信息

菜品主页模块：在菜品表中读取菜品信息并按照一定模板显示在首页。

菜品搜索模块：将菜品表中所有菜品的名称或图片关键字与搜索框中输入的字段进行比对，找到所有包含输入字段的菜品纪录，从而将菜品、图片和发布时间等信息展示给普通用户。

菜品浏览模块：系统根据菜品ID从菜品表中取出菜品的主体图片。

分类浏览模块：系统根据类别ID在菜品表中取出该类别下的所有菜品。

菜品评论模块：将想要评论的菜品的ID号和评论的图片以及评论人的信息和评论时间记录到评论表中。也可以根据想要查看的菜品的ID号，在评论表中调出所有他人评论。

在线留言模块：注册用户在网站前台添加留言，包括留言ID、名称、图片、头像等信息。

后台管理模块：针对普通管理员和超级管理员

添加类别：将类别信息添加到类别表，此类别可被选择。

修改类别：根据类别ID，将类别表中相应的类别信息进行更新。

删除类别：根据该类别ID找到菜品表中该类别下的所有菜品，并通过菜品ID从评论表中找到所有相关评论，先删除所有评论再删除菜品，最后删除类别本身。

添加菜品：将菜品的类别ID和菜品名称以及菜品图片等记录到到菜品表中，经管理员审核通过后，此菜品可以被浏览者查看。

修改菜品：根据菜品ID，在菜品表中找出相应记录并进行更新。

删除菜品：根据该菜品ID从评论表中找到所有相关评论，先删除所有评论再将菜品 本身从菜品表中进行删除。

审核菜品：将主编ID、菜品ID、审阅时间等记录到审阅表，经管理员审核通过后的菜品才可以在前台界面显示。

删除评论：将该评论记录从评论表中移除。

删除用户：将该用户从用户表中移除。

系统管理模块：包括系统简介设置、系统公告设置、友情链接添加、删除、修改，数据备份、数据表批量删除。

留言管理模块：管理员管理系统留言信息，可以进行回复、删除，将留言ID、名称、图片、头像等信息记录在留言表中。

添加留言：将留言的ID和留言标题以及留言图片等记录到到留言表中，经过添加后后，此留言可以被浏览者查看。

修改留言：根据留言ID，在留言表中找出相应记录并进行更新。

删除留言：根据该留言ID从留言表中找到ID等于该留言ID的留言信息，将留言本身从留言表中进行删除。

3.3　数据库设计

在系统的开发过程中，数据库的设计是非常关键的一步。良好的数据结构设计不仅可以减少数据冗余、提高查询效率，而且还可以降低应用程序的编写难度，提高工作效率。

3.3.1　实体E-R图

本系统在数据库中存放了多张表，分别是用户表、权限表、留言表、订单表、菜品类别表、菜品表、订餐记录表。

订餐网站的主要实体间关系E-R图如图3-2所示：



图3-2　　订餐网站主要实体间关系E-R图

菜品信息的实体图如图3-5。



图3-5菜品信息实体图

留言信息实体图如图3-7。



图3-7留言信息实体图

用户信息实体图如图3-8。



图3-7用户信息实体图

评论信息实体图如图3-7。



图3-7评论信息实体图

3.3.2　数据库的逻辑结构

根据上面的分析，可以把系统E-R图转换成关系模式，依据转换规则以及范式理论，可以得到本系统的关系模式如下（其中有下划线表示主键）：

Allusers表是系统用户的信息表，存储系统用户真实姓名、性别、年龄、电话、地址、备注、编号、姓名、用户密码以及其相对应的密码等相关信息。

canpinxinxi表是网站菜品信息表，存储网站的菜品景点的相关信息，其中包括编号、菜品名称、菜品类别、价格、描述、添加时间等相关信息。

diancanjilu表是用户订餐记录信息表，存储用户的订餐的内容信息，其中包括订餐记录 编号、菜品名称、价格、数量、金额、订餐人、附加要求、是否审核等相关信息。

liuyanban表是用户留言信息表，存储用户的留言的内容和回复信息，其中包括编号、头像、性别、QQ、邮箱、电话、内容、添加时间、回复内容等相关信息。

pinglun表是用户评论菜品的信息表，存储用户评论菜品的内容信息，其中包括编号、信息ID、评论内容、评论人、评分、添加时间等相关信息。

yonghuzhuce表是用户注册信息表，存储用户的注册信息，其中包括编号、用户名、密码、姓名、性别、出生年月、QQ、邮箱、电话、身份证、头像、地址、备至、折扣等相关信息。

xinwentongzhi表是系统公告、系统简介的内容信息，其中包括编号、标题、类别、内容、添加人、首页图片、点击率、添加时间、摘要等相关信息。

4　　详细设计

4.1　数据库的连接

本节在SQL Server2012中利用SQL语句和图形界面相结合的方法建立各个表并设置其主外键联系，并在项目中对数据库的连接进行详细设计。

如果将数据库的连接字符串写在cs文件中，当数据库的用户名或密码发生改变时，项目需要重新生成，这样会耗费大量资源。比较好的解决方法是将其写在Web.config中。 Web.config中数据库连接字符串如下：

<connectionStrings>

<add name="connStr" connectionString="Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=newssystem;Data Source=.;User Id=sa;Password=wangcongli"/>

</connectionStrings>

另外需要建立一个数据库助手类来执行基本的数据库增删改查工作，在cs文件建立数据库连接的主要代码是：

string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;

SqlConnection conn = new SqlConnection(connStr);

每次执行数据库的基本操作时询问数据库的连接状态，若处于关闭状态则连接数据库：conn.Open()；执行完数据库的基本增删改查工作后需要断开连接：conn.Close()。

4.2　系统设计流程

以下将分别介绍各个模块的设计流程。

1. 用户注册模块



图4-15　　用户注册流程

注册开始后，用户发送注册请求信息，经过系统验证用户名是否已经存在，如果用户名已经存在，重新注册用户，跳转至注册页面，如果不存在，则注册成功，也返回注册页面。

1. 用户登录模块



图4-16　　用户登录流程图

用户登录流程：用户输入用户名和密码，系统自动验证是否存在该用户名，如果存在用户名和密码相匹配，则登录成功，跳转至系统主界面，如果用户名与密码不匹配或者用户名不存在，则返回主界面。留言主页模块



图4-17　　主页载入流程图

主页载入流程：系统先连接数据库，显示界面参数初始化，读取数据库的数据表，读取html显示模板，读取数据库的数据标题，读取数据库的数据内容，对数据按添加时间排序，按照模板设计位置显示数据，刷新显示界面，断开数据库连接。

1. 信息搜索模块



图4-18　　信息搜索流程

系统的查询信息的流程：查询开始，用户输入需要查询的信息标题或名称等关键字，系统自动判断是否存在查询条件匹配的信息，如果存在则显示数据列表，用户可以根据需要进入相关信息的详细界面，如果不存在，则返回查询界面。

1. 信息浏览模块



图4-19　　信息浏览流程

信息浏览的流程：先连接数据库，然后显示界面参数初始化，读取数据库中的信息编号，读取显示模板，读取数据库的信息标题，读取数据库中该信息的存储路径，读取显示网页的文件，按照模板设计位置显示信息，刷新显示界面，断开数据库连接。

1. 分类浏览模块



图4-20　　分类浏览流程

分类查询浏览流程：首先系统连接数据库，获取类别名称，读取数据库中该类别的所有信息，读取显示模板，按类别显示查询结果。

1. 信息评论模块



图4-21　　信息评论流程

评论流程：用户登录成功后，浏览信息页面，输入评论语，发表该评论，修改评论，返回评论相关界面。

1. 菜品管理模块



图4-22　　类别管理流程

首先进入菜品管理，可以选择添加菜品、修改菜品、删除菜品，添加菜品要输入菜品信息，提交后，进行保存，修改菜品要编辑菜品信息，提交后保存，如果失败则抛出业务逻辑异常，删除菜品则直接在数据库删除即可。

5　　开发工具及相关技术介绍

5.1　开发工具介绍

开发工具介绍主要介绍使用到的软件，包括Visual Studio、SQL Server和IIS服务器。

5.1.1　Visual Studio介绍

Visio Studio是一个可视化编译环境，它为开发者提供了很大的便利，从编写源代码到编译链接，从调试到最终发布，可以通过预设的菜单及选项很方便的完成。

Visual Studio .NET是为建立.NET Framework应用而设置的集成开发环境（IDE），它在.NET Framework和公共语言规范CLS（Common Language SPecification）基础上可运行VB、C++、C#、Jscript、J#等多种语言。它的体系结构如下：



图5-1　　Visio Studio .NET的结构

其中CLR公共语言运行时的结构如下：



图5-2　　CLR的结构

5.1.2　SQL Server介绍

现代计算机中存储的数据呈现暴涨的趋势，这样就需要一款高效的数据库管理工具来定义数据和操纵数据，使得数据在计算机中的存储更加有秩序，降低数据的冗余度，提高数据的独立性，也让数据的操纵更加方便。SQL Server采用图形界面，使用方法一目了然。

（1）SQL Server是一款Relational Database Management System

（2）SQL Server与Windows系统完美结合

（3）SQL Server对计算机的硬件条件要求不高

（4）SQL Server具有良好的可伸缩性

（5）SQL Server具有良好的灵活性，它可以适应快速变化的环境

5.1.3　IIS介绍

IIS(Internet Information Server)是微软公司推出的Web服务器，具有安全、高效、易于管理的特点，是目前主流的Web服务器之一，可支持HTML、ASP、ASP.NET等。IIS在管理与配置上继承了Windows的特点，全部是图形界面的方式，操作起来十分简单。

5.2　开发模式B/S介绍

B/S模式也就是浏览器/服务器模式，它的界面部分是在浏览器端展示，而主要工作是由服务器端进行实现的，用户的请求由浏览器端提交给服务器端进行处理，而服务器将处理结果反馈给浏览器端，在浏览器端界面描画给用户查看。采用B/S模式不仅可以避免用户必须安装专业软件才能开发系统或者访问系统的局限性，而且更加便利。另外，软件系统的更新换代速度越来越快，采用B/S模式开发的系统的升级工作主要在服务器端进行，这样不管系统的用户群是多么的庞大，都不会影响系统维护工作的工作量，降低了系统的维护成本，这也是B/S模式的一大优点。

B/S模式的结构模型如下：



图5-3　　B/S模式及其工作过程

5.3　开发技术介绍

开发技术介绍主要介绍系统中使用到的关键技术，包括ASP.NET和ADO.NET。

5.3.1　ASP.NET介绍

ASP.NET是一种动态网页技术，它开发的是Web应用程序，可以和数据库以及其他程序进行交互，是一种快捷的动态服务器页面开发工具。另外，ASP.NET的代码和页面分离，大大提高了代码的可读性，这样使得功能的扩展更加容易。

本课题将采用ASP.NET的三层架构，三层架构模型如下：



图5-4　　三层架构

表示层用来和用户交互，接受用户的请求，并且将服务器的处理结果展示给用户。

业务逻辑层位于表示层和数据访问层之间，它在两者之间传递数据，按功能需求调用数据访问层中的方法，并向客户展示数据和信息。

数据访问层使用了一个强类型的DataSet，通过ADO.NET操纵数据，不涉及具体业务，它为业务逻辑层提供数据服务，如存储数据操作结果、返回数据检索结果等。

5.3.2　ADO.NET介绍

数据库是ASP.NET动态网站的基石，离开了数据库，ASP.NET的魅力将大大减少，所以ASP.NET支持ADO.NET以实现对数据库的访问。通过ADO.NET能够建立起用户与数据库之间的桥梁。用户只需在网页中执行SQL指令，即可对Web服务器上的数据库进行添加、修改和删除操作。

ADO.NET的结构模型如图5-5所示：



图5-5　　ADO.NET的结构

6　　系统实现

6.1　前台功能模块的实现

本系统是用作毕业设计，IIS在本机上开启一个服务器，系统实现的功能局限在本机使用。本章将对前台功能模块的实现界面及核心代码进行阐述。

6.1.1　留言主页模块

在浏览器的地址栏里输入http://localhost:1675/default.aspx地址，就可以跳转到订餐网站的首页，前台首页是由4个页面组成的，包括top、left、down和center等页面，这里也是浏览者访问系统的入口，通过@ Register引入，如图6-1所示。



图6-1　　主页模块

其中载入前台页面的主要代码如下：

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<%@ Register src="qttop.ascx" tagname="qttop" tagprefix="uc1" %>

<%@ Register src="qtleft.ascx" tagname="qtleft" tagprefix="uc2" %>

<%@ Register src="qtdown.ascx" tagname="qtdown" tagprefix="uc3" %>

前台数据准备，通过按ID降序排，主要代码如下：

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

{

if (!IsPostBack)

{

sql = "select top 5 id,title,shouyetupian from allgonggao where shouyetupian<>'' and shouyetupian like '%.jpg' order by id desc";

getsyt(sql);

sql = "select top 9 id,title,addtime,leibie from allgonggao where leibie=' ' order by id desc";

getdata(sql);

sql = "select top 4 \* from yaoshanpeifang where tupian<>'' order by id desc";

getdata2(sql);

sql = "select top 8 id,title,addtime,leibie from allgonggao where leibie=' ' order by id desc";

getdata3(sql);

sql = "select top 8 id,title,addtime,leibie from allgonggao where leibie=' ' order by id desc";

getdata4(sql);

}

}

6.1.2　用户注册模块

点击主页上方的注册会弹出如下注册框：



图6-2　　用户注册界面

王在前台注册页面的html代码如下：

<td nowrap="nowrap" style="width: 130px">

<font face="宋体">用户名:</font></td>

<td align="left" width="79%">

<asp:TextBox ID="yonghuming" runat="server"></asp:TextBox>\*<asp:RequiredFieldValidator

ID="RequiredFieldValidatoryonghuming" runat="server" ControlToValidate="yonghuming"

ErrorMessage="必填"></asp:RequiredFieldValidator></td>

</tr>

<tr style="color: #000000; font-family: 宋体">

<td nowrap="nowrap" style="width: 130px">

<font face="宋体"><span>密码</span>:</font></td>

<td align="left" style="font-family: Times New Roman" width="79%">

<asp:TextBox ID="mima" runat="server" TextMode="Password" Width="147px"></asp:TextBox>\*<asp:RequiredFieldValidator

ID="RequiredFieldValidatormima" runat="server" ControlToValidate="mima" ErrorMessage="必填"></asp:RequiredFieldValidator>确认密码：<asp:TextBox

ID="mima2" runat="server" TextMode="Password" Width="147px"></asp:TextBox>

<asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="mima"

ControlToValidate="mima2" ErrorMessage="两次密码不一致"></asp:CompareValidator></td>

6.1.3　用户登录模块

在网站前台的left页面里，包含着登录页面，通过,table画出登录页面，登录后台提交表单到login.aspx，点击主页上方的登录会弹出如下登录框：



图6-3　　前台用户登录界面

前台用户登录的html代码如下：

<table align="center" border="0" cellpadding="0" cellspacing="0" height="103" style="width: 80%">

<tr>

<td align="left">

用户名:</td>

<td align="left" colspan="2">

<asp:TextBox ID="TextBox1" runat="server" Style="border-right: #cadcb2 1px solid;

border-top: #cadcb2 1px solid; font-size: 12px; border-left: #cadcb2 1px solid;

width: 100px; color: #81b432; border-bottom: #cadcb2 1px solid; height: 18px"></asp:TextBox></td>

</tr>

<tr>

<td align="left">

密码:</td>

<td align="left" colspan="2">

<asp:TextBox ID="TextBox2" runat="server" Style="border-right: #cadcb2 1px solid;

border-top: #cadcb2 1px solid; font-size: 12px; border-left: #cadcb2 1px solid;

width: 100px; color: #81b432; border-bottom: #cadcb2 1px solid; height: 18px"

TextMode="Password"></asp:TextBox></td>

</tr>

<tr style="display: none">

<td align="left">

6.1.4　个人信息管理模块

用户登录成功后，点击右上方的管理个人信息即可进入个人信息管理模块。

****

图6-5　　个人信息管理界面

6.1.5　在线留言模块

在网站前台，未登录的游客用户只能查看留言，不能在线添加留言，登录后的用户可以查看留言并添加留言，和管理员交流，留言主要的页面是liuyanban.aspx。如图6-7所示：



图6-7　　在线留言界面

留言板的添加留言的主要代码：

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

{

string sql;

//sfdigdh ischongfu("zhujianquchong");

sql="insert into liuyanban(cheng,biaoqing,biaoti,neirong,huifu) values('"+cheng.Text.ToString().Trim()+"','"+biaoqing.Text.ToString().Trim()+"','"+biaoti.Text.ToString().Trim()+"','"+neirong.Text.ToString().Trim()+"','"+huifu.Text.ToString().Trim()+"') ";

int result;

result = new Class1().hsgexucute(sql);

if (result == 1)

{

Response.Write("<script>javascript:alert('添加成功');</script>");

}

else

{

Response.Write("<script>javascript:alert('系统错误，请检查数据库设置问题');</script>");

}

}

6.1.6　菜品评论模块

本系统可以对菜品进行评论和查看他人对该条菜品的评论，如图6-9所示：



图6-9　　菜品评论界面

添加评论的主要代码:

public bool Insert(Comment c)//c是评论实体类

{

bool flag = false;

string sql = "insert into comment([content],userIp, newsId) values(@content,@userIp,@newsId)";

SqlParameter[] paras = new SqlParameter[] {

new SqlParameter("@content", c.Content),

new SqlParameter("@userIp", c.UserIp),

new SqlParameter("@newsId", c.NewsId)

};

int res = sqlhelper.ExecuteNonQuery(sql, paras, CommandType.Text);

if (res > 0){flag = true;}

return flag;

}

6.1.7　在线订餐模块

注册用户登录系统前台后，可以选择喜欢的菜品，进行在线订餐，添加订餐记录，在个人后台可以查看个人的订餐记录，对其进行管理，如图6-9所示：



图6-9　　在线订餐界面

6.2　后台管理模块的实现

后台管理模块主要包括系统用户管理、留言动态管理、菜品类别管理、菜品信息管理、订餐记录管理、评论管理、系统管理。以下将分别介绍各个模块的实现。

6.2.1　后台登录模块

后台登录界面如下：



图6-10　　后台登录界面

输入用户名、密码和验证码之后点击登陆按钮会进行以下判断：判断验证码是否输入正确，验证码不正确会提示重新输入；判断用户名和密码是否正确，只有当用户名和密码都正确时才会登录成功，否则就会提示登录失败。

6.2.2　菜品类别管理模块

菜品类别管理的界面，如图6-13所示：



图6-13　　菜品类别管理界面

增加菜品类别的核心代码：

public bool Insert(string caName)//caName是菜品类别名称

{

bool flag = false;

private SQLHelper sqlhelper =new SQLHelper();

string sql = "insert into category(name) values(@caName)";

SqlParameter[] paras = new SqlParameter[] {

new SqlParameter("@caName", caName)

};

int res = sqlhelper.ExecuteNonQuery(sql, paras, CommandType.Text);

if (res > 0){flag = true;}//插入成功

return flag;

}

6.2.3　用户管理模块

用户管理的界面，如图6-17所示：



图6-17　　用户管理界面

6.2.4　评论管理模块

删除评论的界面，如图6-18所示：

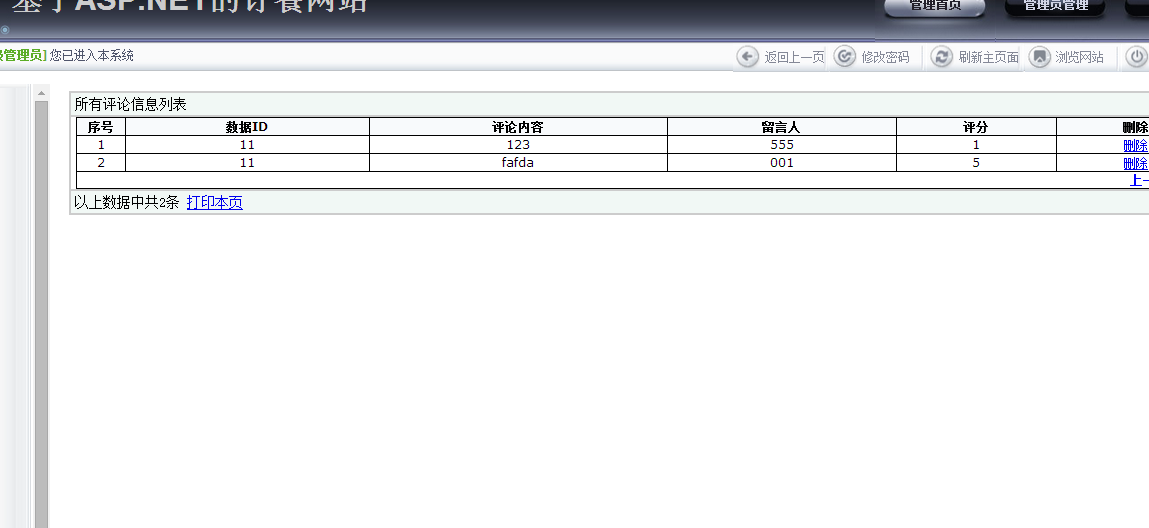


图6-18　　删除评论界面

删除评论的主要代码为：

public bool Delete(string id)

{

bool flag = false;

string sql = "delete comment where id=@id";

SqlParameter[] paras = new SqlParameter[] { new SqlParameter("@id",id)}; int res = sqlhelper.ExecuteNonQuery(sql, paras, CommandType.Text);

if (res > 0){flag = true;}

return flag;

}

6.2.5　菜品管理模块



图6-14　　菜品管理界面

添加菜品的界面，如图6-16所示：

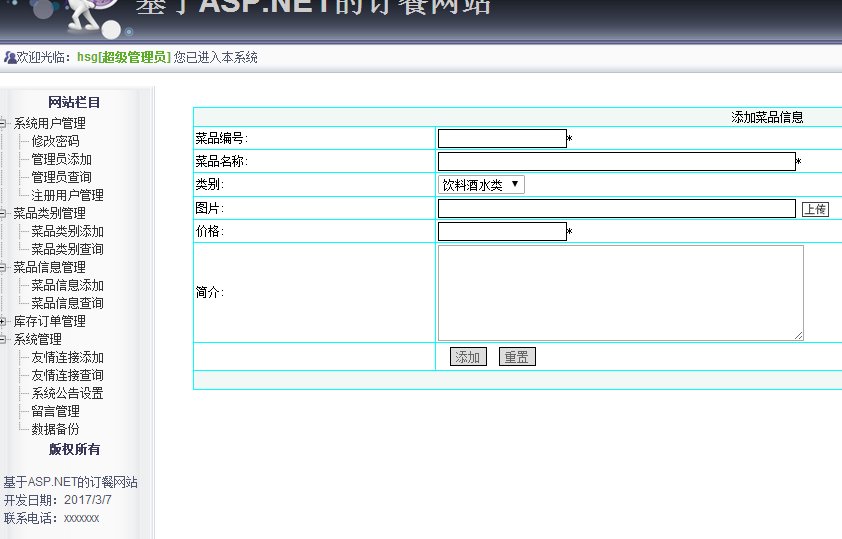


图6-16　　添加菜品界面

添加菜品的核心代码：

public bool Insert(News n)

{

bool flag = false;

private SQLHelper sqlhelper =new SQLHelper();

string cmdText = "news\_insert";

SqlParameter[] paras = new SqlParameter[] {

new SqlParameter("@title", n.Title),

new SqlParameter("@content", n.Content),

new SqlParameter("@caid", n.CaId)

};

int res = sqlhelper.ExecuteNonQuery(cmdText, paras,CommandType.StoredProcedure);

if (res > 0){flag = true;}

return flag;

}

6.2.6 入库管理模块

过去传统的手工登记入库信息很不方便，也很容易造成数据出错，所以本系统需要一个可以管理登记入库信息的功能，管理员登入系统，可以添加新的入库信息，修改现有的入库信息，添加入库信息需要指定菜品编号、菜品名称、类别、入库价格、入库数量等重要信息，管理员可以查看入库的时间和修改的时间，管理员还可以通过菜品名称、入库编号等关键字进行入库信息的搜索。

入库操作的界面如图5-9。



图5-9入库操作界面

入库记录的界面如图5-9。



图5-9入库记录界面

6.2.7　数据备份

系统管理只包含数据备份、友情链接添加、友情链接查询、系统简介设置、系统公告设置，为了防止系统的数据丢失，网站管理员可以经常使用数据备份功能进行系统数据备份。

系统管理的界面如图5-12。

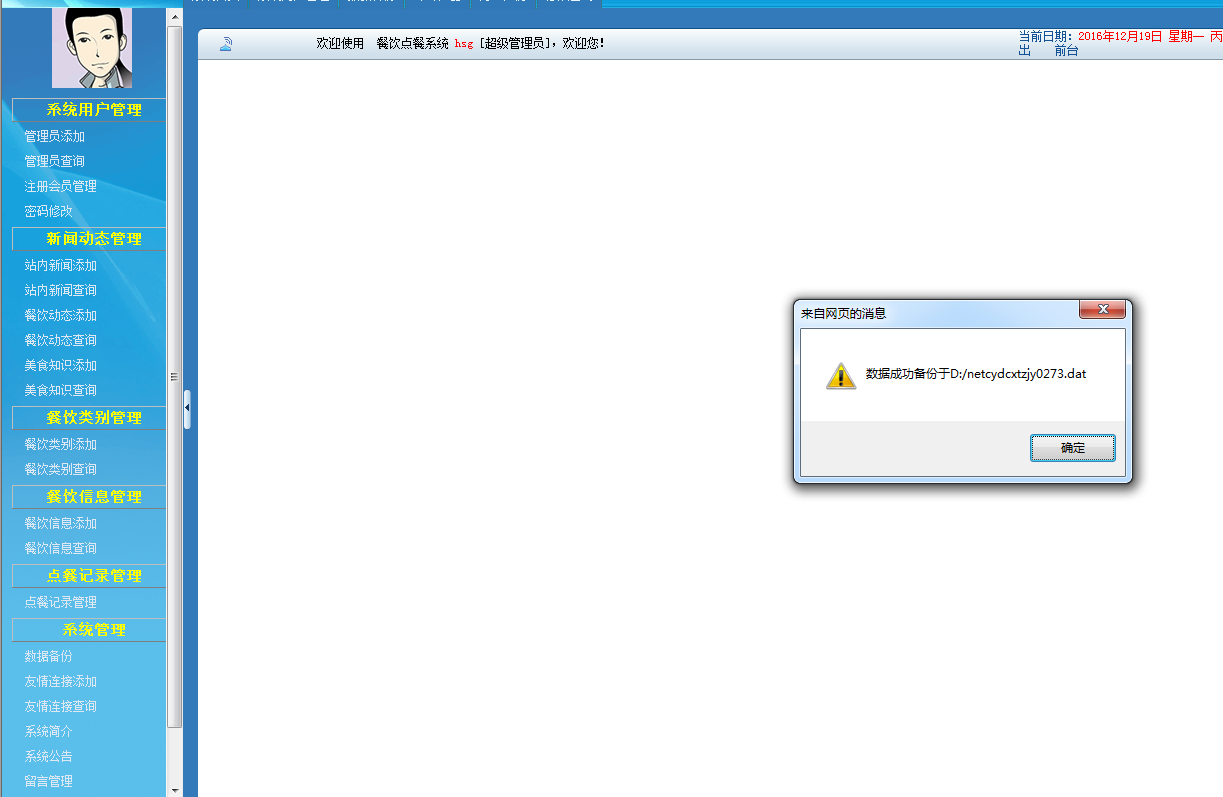


图5-12系统管理界面

数据备份关键代码：

public partial class databack : System.Web.UI.Page

{

SqlConnection myConnection = new SqlConnection(ConfigurationSettings.AppSettings["cn"]);

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

{

if (Session["cx"].ToString().Trim() != "超级管理员")

{

Response.Write("<script>javascript:alert('对不起,您没有这个权限');history.back();</script>");

Response.End();

}

SqlCommand mycmd = new SqlCommand();

string myExecuteQuery = "backup database nethclqlwkf99A4 to disk='D:/nethclqlwkf99A4.dat'";

SqlCommand myCommand = new SqlCommand(myExecuteQuery, myConnection);

myCommand.Connection.Open();

myCommand.ExecuteNonQuery();

myConnection.Close();

Response.Write("<script>javascript:alert('数据成功备份于D:/nethclqlwkf99A4.dat');location.href='sy.aspx';</script>");

Response.End();

}

7 系统测试

软件需要按照测试方案和测试流程对软件的性能和功能进行测试。系统测试是通过执行程序来发现系统中的错误。测试和开发需要同步进行。

6.1 测试用例

用户登录功能测试，如表6.1所示。

表6.1 用户登录功能测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| LO\_1 | 用户登录模块测试 | 正确输入用户信息，用户成功登陆。 | 用户的信息 | 1.在电子邮箱输入框输入用户的电子邮箱。2.在密码框输入用户的密码。3.点击登陆按钮。 | 跳转到首页界面 | 正确 |

用户注册功能测试，如表6.2所示。

表6.2 用户注册功能测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| RE\_1 | 用户注册模块测试 | 正确输入用户信息，完成用户的注册。 | 用户的基本信息 | 用户在网站的注册页面根据提示填写相应的信息，然后点击“注册”按钮。 | 提示成功并跳转到登陆界面。 | 正确 |
| RE\_2 | 用户注册模块测试 | 用户注册失败的情况 | 用户不填写用户名 | 用户在网站的注册页面根据提示填写除了用户名以外相应的信息，然后点击“注册”按钮。 | 注册失败，提示“\*用户名不符合要求”。 | 正确 |
| RE\_3 | 用户注册模块测试 | 用户注册失败的情况 | 用户两次输入密码不一致 | 用户在网站的注册页面根据提示填写相应的信息，填写密码时两次输入的密码不一致，然后点击“注册”按钮。 | 注册失败，提示“\*两次输入密码不一致”。 | 正确 |

菜品信息界面测试，如表6.3所示。

表6.3 菜品信息界面测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| GINFO\_1 | 菜品信息显示功能模块测试 | 菜品信息显示的正常情况 |  | 在网站的首页选择一个菜品分类进入菜品分类列表页，然后选择一件菜品。 | 进入菜品信息界面 | 正确 |

加入购物车功能测试，如表6.4所示。

表6.4 加入购物车功能测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| ADDCART\_1 | 加入购物车功能测试 | 菜品加入购物车成功场景。 |  | 在网站的菜品信息页选择菜品点击“加入购物车”按钮。 | 界面提示“有一件菜品加入购物车！” | 正确 |

用户评论功能测试，如表6.5所示。

表6.5 用户评论功能测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| CO\_1 | 用户评论功能测试 | 用户评论成功场景。 | 用户评论的信息。 | 用户在网站的菜品信息界面输入评论内容，然后点击“添加评论”按钮。 | 界面显示用户的评论。 | 正确 |
| CO\_2 | 用户评论功能测试 | 用户评论失败场景。 | 用户不输入评论内容。 | 用户在网站的菜品信息界面不输入评论内容，然后点击“添加评论”按钮。 | 评论失败，提示“\*评论内容不能为空！”。 | 正确 |

管理员添加菜品功能测试，如表6.6所示。

表6.6管理员添加菜品功能测试用例表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 测试功能 | 输入 | 操作过程 | 预期输出 | 测试结果 |
| ADDG\_1 | 管理员添加菜品功能测试 | 添加菜品成功的情况。 | 新菜品的基本信息。 | 管理员在后台首页选择“新菜品”选项，然后填写新菜品信息，点击“添加”按钮。 | 新菜品添加成功。 | 正确 |
| ADDG \_2 | 管理员添加菜品功能测试 | 添加菜品失败的情况。 | 管理员不填写菜品名称。 | 管理员在后台首页选择“新菜品”选项，然后填写除菜品名称以外的新菜品信息，点击“添加”按钮。 | 添加失败，提示“\*请填写菜品名称！”。 | 正确 |
| ADDG \_3 | 管理员添加菜品功能测试 | 添加菜品失败的情况。 | 管理员在价格框输入非数字字符。 | 管理员在后台首页选择“新菜品”选项，然后填在价格框输入非数字字符，点击“添加”按钮。 | 添加失败，提示“\*价格必须是货币格式！”。 | 正确 |

6.2 测试结果分析

在网站完成后，及时的对网站进行一系列的测试，通过一次次的测试来去发现程序中的问题。首先对各个功能模块进行了独立的测试。然后再对整个网站进行测试。测试用户的登陆和注册功能，菜品的显示、购买、评论，以及管理员在网站后台对网站的各部分进行管理，根据测试结果得知各个功能模块都达到了设计的要求。各个模块在整个系统中都能完成各自的功能，相互之间的连接也很顺利。对数据库的增加、删除、更新、查看操作也都能够顺利的完成。经过测试整个网站的功能都达到了预期的要求。在测试过程中发现的程序错误等也得到了及时的解决，从而保证了后期系统的稳定运行。

8　　结论

在订餐网站的设计过程中，由于对ASP.NET并不熟悉，系统的开发也属于边学习边应用的过程，因此对ASP.NET的理解不免出现偏差和错误，程序中肯定存在许多需要改正的地方。另外，由于毕业设计的时间比较紧张，而且自己技术有限，系统不免有些简陋，这也是今后系统需要改进和完善的地方。

在开发过程中遇到了很多疑问，在查阅了大量的文献和教学视频以及老师同学的指导之后，终于解决了问题。在学习过程中学到了很多新知识，对ASP.NET、ADO.NET、HTML、CSS、Javascript等都有了更深层次的理解，对三层架构模型也更加的熟悉。这次毕业设计使我不但学习到了新技术，而且也积累了开发经验，从中也得到了许多乐趣。

# 参考文献

[1]Eric L.Brown Quilogy.SQL Server 2005中文版精粹.机械工业出版社华章公司，2012

[2]杨晓光.ASP.net程序设计（修订本）[M].清华大学出版社，2011

[3]萨师煊，王珊.数据库系统概论.北京：高等教育出版社，2010

[4]刘志成.Windows 程序设计（ASP2.0）实例教程[M].清华大学出版社，2010

[5]张德静.深入浅出ASP.net程序设计与开发.中国青年出版社，2015年5月

[6]宋昆，李严.SQL Server数据库开发实例解析[M].北京：机械工业出版社，2012

[7]张海藩.软件工程导论.北京：清华大学出版社，2013

[8]汪小平，吴永强．Asp.net网络开发技术.北京：北京邮电出版社，2012

[9]百度百科．B/S结构.<http://baike.baidu.com/view/268862.htm>

[10]刘晓华.SQL Server 2000数据库应用开发.北京：电子工业出版社，2011

[11]张林.软件工程.北京：中国铁道出版社，2009

[12]姜力，高群.SQL Server数据库设计与管理.中国林业出版社，2006

[13]张蓓.ASP.NET通用模块及典型系统开发实例导航.北京：人民邮电出版社，2006

[14]（美）Fritz Onion著，施诺译.ASP.NET基础教程ASP案例版.北京：清华大学出版社，2003

[15]Jesse Liberty，Dan Hurwitz.Programmimg ASP.NET 3rd Edition.New York：O'Reilly

致　　谢

光阴如梭，毕业在即。本次系统的设计过程中，有许多人给了我很大的帮助，没有他们的支持，毕业设计也不会进行得这么顺利。在此，要衷心感谢给予我帮助的老师和同学们。

首先要衷心感谢我的指导老师，感谢老师对我精心指导，在寒假的时候老师就提前给了我们学习ASP.NET的电子书和论文相关的资料，从论文的选题、开题报告、系统的设计到论文的撰写，都倾注了大量的心血和提出了宝贵的意见，帮我调整思路和方向，督促毕业设计的进度，使我能够顺利完成本次毕业设计。再次向老师表达我衷心的感谢！

同时，感谢这四年来教过我的老师和我的导员们，他们不仅教授了我知识，也在我这四年的成长中给了我许多宝贵的意见和鼓励。也非常感谢我的同学和舍友们，在学习的过程中他们不厌其烦的帮我解答编写代码中遇到的难题，鼓励我继续努力。有了他们的支持，我才顺利的完成了毕业设计！