

合肥工业大学

《软件工程综合训练》课程设计报告

学 号 2017218007

姓 名 文 华

专 业 班 级 物联网工程 17-2 班

选 题 旅游信息查询系统

指 导 老 师 李 心 科

合肥工业大学

2021 年 1 月

目 录

1	概述	1
1.1	项目研究背景	1
1.2	研究目的及意义	1
1.3	开发软件介绍	1
1.3.1	技术介绍	2
1.3.2	MySQL 数据库	4
2	系统分析	6
2.1	可行性研究	6
2.1.1	经济上的可行性	6
2.1.2	技术上的可行性	6
2.1.3	操作上的可行性	6
2.2	开发结构分析	7
2.3	功能需求分析	8
2.4	数据流图	9
2.5	业务流程分析	10
3	网站总体设计	11
3.1	系统功能描述	11
3.2	系统界面设计	11
3.3	系统功能结构图	12
3.4	系统设计目标	13
3.5	系统实现的软硬件平台	13
4	数据库设计	14

4.1	数据库概念结构设计	14
4.2	数据库逻辑结构设计	17
4.3	本章小结	18
5	系统详细设计与实现	18
5.1	主界面模块设计	18
5.2	新增景点设计	20
5.3	登录模块设计	22
5.4	用户信息和密码修改	23
5.5	Web 客户端对 MySQL 的访问设计	23
6	系统测试	25
6.1	系统测试环境	25
6.2	系统测试计划	25
6.3	测试过程和结果分析	26
6.3.1	功能测试	26
6.3.2	集成测试	27
7	项目总结	28
附录	28

1 概述

1.1 项目研究背景

社会经济的发展和提高潜移默化的影响了人们对精神消费的日益看中与提高，所以越来越多的人开始选择更健康有趣的生活活动，随之而来的旅游便成了人们消费的必选。随着旅客需求的日趋丰富和个性化，这必将推动我国第三产业、旅游服务行业的迅猛发展前进。旅游行业如何能够提供多样化、个性化的服务来满足人们的多方面需求，这是一个值得深究的课题。旅游行业快速成长的同时，各个行业相互间的联系与竞争也就不断加强。互联网时代的大背景环境下，电子信息服务的将会是一个最便捷最省力最有效的服务方式。这就要求旅游行业能够提供一个综合性服务的信息平台，同时必须拥有强大的后台支撑、智能化的信息提供和服务，快速有效的信息应答，多种多样的信息功能等等。虽然目前我国电子信息科技迅猛发展，但是也有差强人意的地方，比如说：

①交互性差。多数信息系统是封闭性的，缺乏与各种其他信息系统的共享与交互，这就造成了信息资源的浪费和利用率低等问题。

②实时性差。系统在开发时存在系统漏洞，不能很好的对空间信息进行全方位的挖掘与利用。用户所需要的信息可能不能及时得到更新与回复。这是摆在多数旅游系统面前的一个现实需求。

③个性化服务缺乏。大多数旅游系统还不完善，没有按实际情况来满足用户多方面的需求，而是片面的加入一些没有必要的或者说运用率并不高的功能。

④旅游系统的单一化。不难看出，大多数旅游系统功能服务基本相同，没有自己的特色，创新程度不强。电子科技迅速发展，也需要我们不断的去追求创新。

1.2 研究目的及意义

随着旅游业的迅猛发展，21 世纪的中国将会成为世界上最大的旅游国家，对于旅游者而言获取相关的旅游信息以便于做出最优的旅游选择是十分必要的。旅游网站中以帮助客户怎么样才能更快速方便的找自己旅游公司咨询和旅游信息。因此开发旅游网站可以发布旅游信息，让客户自己在网上就可以方便快捷的了解到旅行社的所有旅行动态，大大方便了旅行者对旅游信息的查询和获取，旅行者也更愿意在电子信息平台下进行咨询和消费。

1.3 开发软件介绍

1.3.1 技术介绍

JSP 是一种能实现动态网页的技术。Sun 公司最先提倡发行，并由多家公司共同开发与完善的。JSP 技术通俗来说其实就是在普通的网页 HTML 文件中首先加上 Java 程序段来实现具体功能，然后再根据需要添加 JSP 标记(tag)进行记录，这样二者结合就可以形成 JSP 文件。其功能特别强大，基本可以实现大多数动态网页所需要的各种效果。用 JSP 技术所开发的网页具有跨平台的特点，在不同的操作系统，例如 Windows，Linux，Winxp 等操作系统上都可以运行起来。在服务器端进行大部分复杂的程序操作和功能实现操作，而这些并不需要用户直接参与或操作，最后的结果通过网络传送给用户。这样就减轻了用户工作量的同时也减少了浏览器的负担。因为有些浏览器并不支持 Java 语言，然而这不影响功能操作的实现，同样能够登录 JSP 网页，达到预想效果。

Java server page（以下简称 JSP）为 servlet 技术的简化。JSP 的功用在于它实现了 Html 语法中以<%， %>形式的 Java 扩张。但是，值得注意的是，因为 JSP 与 Servlet 均在服务器端运行，故而往往以 HTML 文本为主要形式呈现客户端面前，所以当有需求时，客户端可以便捷地通过浏览器来实现浏览。当网页服务器（web）遇到访问 JSP 提出的要求时，需要进行以下两个步骤：首先执行程序段，后将结果与 html 返回，以便系统更好运行。至于为何要进行 Java 代码段的插入操作，其主要目的在于进行数据库的连接、网页重定向等等，以满足网页动态实现所需的各类功能。

同时，我们也应该认识到 JSP 技术所使用的两种程序：即类 XML 的 tags 和 scriptlets，这些都是用 Java 语言实现的。另一方面，对于服务端也可以通过 tags 和 scriptlets 实现对资源的访问。除此之外，JSP 将网页逻辑和网页设计分离开来，大大方便了编程人员实现对应用程序的设计与开发。同时，由于 JSP 的发展运用，使得网页设计变得更加灵活便捷，也使得网页的实现更加容易。

html 语言是一种脚本语言，是一种规范和标准，在所有的开发语言中，是较为简单易学的一种，它有很多标签，运用标签就可以实现很多静态页面效果。虽然 html 语言相对其他语言来说比较简单，但是他能够实现的功能却很强大，因此受到广大用户的热爱。他可以实现静态网页设计，是学习前端设计的一门基础性语言。html 的结构包括头和主体部分，是 WWW 的描述语言。

Java 语言是一种完全面向对象的语言。具有继承，多态性，封装性等特点，还具有垃圾自动回收机构，程序员不用考虑内存使用不足的情况，因为内存存在程序员不再使用后

的一段时间内会自动释放。而且具有跨平台性。这些别的语言所没有的特点让 Java 能够快速发展强大起来，它吸收了计算机领域的各种最新成果，功能十分强大。Java 的应用也是到处可见，从计算机到移动平台，从办公软件到手机应用，从小型系统到大型企业，Java 身影无处不在。Java 能得到如此广泛的应用的另一个重要原因是因为 Java 语言是动态的，它可以适应动态变化环境。

MyEclipse 其实是在经过一系列已有软件产品的补充和更新的基础上产生出来的，是一款功能丰富的企业级开发环境。在多种程序开发、服务器效能的提高、功能实现等各个方面起到了不可估量的用处，很大程度上增强了开发效率。总之，MyEclipse 和 eclipse 有着千丝万缕的关系，它其实就相当于一个功能插件，并且它具有丰富的开发环境，使用范围广泛，能够实现多种功能，支撑目前流行的几乎所有开源工具。像 JavaServlet、JSP、JSF、Spring、AJAX 等等。而在结构上，MyEclipse 的特征有 Web 开发工具，应用程序服务器的连接器等等 7 大类（在此只举出 2 类）。这种结构布局使得模块间各自独立，各个模块各自运行，他们之间互不影响，单独完成各自的升级或者扩展。MyEclipse 是基于 Eclipse 插件的解决方案，功能强大并且支持十分广泛，支持云开发、Java Web 开发、移动开发、优化开发、RESTful Web 服务开发、保持 Maven、架构、企业交付、测试等等，此外，MyEclipse 版本更新迅速，不断完善已有功能，弥补漏洞，方便开发。MyEclipse 具有完备的编码、调试、测试和发布功能，是一款功能丰富的 JavaEE 集成开发环境。

MyEclipse 这一开发工具，每个版本都有其各自的特点，本文使用的 MyEclipse8.5。8.5 版本经过长期的投入使用，较以前旧的版本修正了许多 bug，性能比较稳定，界面比较美观，用起来也方便，操作较为简单。但是就兼容性来说，MyEclipse6.5 相对比较好。而 8.5 版本容量大，用起来可能会存在卡的情况。但是除此之外，8.5 也有很多优点，比如说它用起来十分灵活，对前台开发的支持性比较好。其实，每个版本都有自己的特点，具体选用哪一版本，还需要根据实际情况，结合所开发项目进行选择。

另外，MyEclipse 还有很好的自动查错功能、自动补全功能。程序员在进行项目开发的时候，可以方便准确的找出出错的地方。此外，自动补全功能也可以大大节省程序员的编程时间，极大的提高了工作效率。总之，在进行软件系统开发时，一个得心应手的开发环境是十分必要的，这一点是会影响到开发的效率。MyEclipse 很好地解决了这样的难题，为软件开发提供了便利条件。

Tomcat 服务器是一个免费的应用服务器，他的封面设计以一个动物的素描为灵感，

正是因为这个设计理念，所以该项目以一个英文的动物名字为项目名称。因为其免费又好用，功能强大，所以成为大多数程序员的首选。Tomcat 是一个小型的，轻量级的软件产品，他在启动运行时占用非常少的内存资源和系统开销，扩展性能比较好，支持多种应用需求，如果一个开发员对某一项功能实现特别感兴趣，他就直接可以将此功能加入到其中。Tomcat 可以独立运行，身为网站的服务器，可以加载动态网页。

1.3.2 MySQL 数据库

数据库是整个系统的核心，用来存储用户数据，合理的数据结构设计能够减少数据存储占用的空间，方便管理和查询，有利于提高系统效率。MySQL 是最流行的关系型数据库管理系统，在 Web 应用方面是最好的好的 RDBMS(Relational Database Management System，关系数据库管理系统)应用软件之一。

MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

MySQL 数据库中存在 root 用户和一般用户，不同的用户被赋予不同的操作权限 [91]。在 mysql5.5 中，对数据库用户提供 5 种不同的操作权限：全局级、数据库级、表级、列级、存储过程和函数级（指定存储过程和函数），权限依次递减。权限划分由系统中存储的六张表进行划分：

① user 表，验证数据库登录，也称为超级用户表，默认 root 用户具有所有数据库操作权限。

② db 表，指定用户对指定的数据库中的对象具有的操作权限，包含用户信息列 (Host, Db, User)，权限列（数据库级权限 19 列）。

③ host 表，作为 db 表的辅助，一般不使用。

④ tables_priv 表，指定用户对某个表格所具有的权限。包含用户信息列 (Host, Db, User, Table_name)，权限列（表级权限 Table_priv, Column_priv 用户所拥有对该表的所有列级权限，如果该列为空，那么 columns_priv 表中用户就没有对该表的列级权限记录），其他（授权时间 Timestamp，授权者 Grantor）。

⑤ columns_priv 表，指定用户对某个字段所拥有的权限。包含用户信息列 (Host, Db, User, Table_name)，权限列（column_priv 列级权限），其他列（授权时间

timestamp)。

⑥procs_pri 表，指定用户对某一个单独的存储过程或函数。包含用户信息列 (Host, Db, User, Routine_name, Routine_type)，权限列 (Proc_priv 存储过程权限)，其他列 (授权时间 Timestamp，授权者 Grantor)。

结构化查询语言 SQL (Structured Query Language)，由 IBM 圣约翰研究实验室为其关系型数据库管理系统 SYSTEM R 开发。SQL 简单易学，结构简单，功能强大，经推出后得到广泛的应用，像 SQL Server、Oracle、Sybase 这类大型数据库管理系统还是像 PowerBuilder、Visual Foxpro 等数据库开发系统都支持 SQL 作为查询语言。

SQL 完成的功能主要有：查询数据；

在表中插入、修改和删除数据；建立、修改和删除数据对象；控制对数据和数据对象的存取；保证数据库的一致性和完整性。SQL 语言主要包含四部分：

SQL 完成的功能主要有：查询数据；

在表中插入、修改和删除数据；建立、修改和删除数据对象；控制对数据和数据对象的存取；保证数据库的一致性和完整性。SQL 语言主要包含四部分：

① 数据定义语言：creat, alter, drop；

② 数据查询语言：select；

③ 数据操纵语言：insert, update, delete；

④ 数据控制语言：commit work, rollback work；

总之，MySQL 数据库为实现系统后台与前台连接起到了十分重要的地位。一个没有连接数据库的系统是无法实现任何有价值的功能的。数据库可以帮助开发人员整合管理数据，有效实现系统所需功能，同时还可以进行权限的设置和数据的添加。总之，SQL 可以满足人们基本的几乎所有要求，为其提供一套经济，有效的开发策略，使得用户付出最少的代价获得最多的回报。SQL 数据库设计简单易懂，功能相对来说比较全面，为软件产品的开发应用提供了良好的环境和数据帮助。关系型数据和结构化数据为数据库提供了新的功能，使得信息能够更好的存储和组织，完善了 SQL 数据库，这样一来，程序员可以运用这一优势，在软件产品开发过程中，能够为用户开发出高质量高性能的数据应用软件。此外，Microsoft SQL Server 2005 还整合了各种功能：例如分析、集成等。用户凭借此完成了对方案的构建。

2 系统分析

2.1 可行性研究

2.1.1 经济上的可行性

实现旅游网站所需要花费的时间，人力，金钱成本较该网站投入使用时能够获得的收益来讲，差距是比较大的。所以，从经济方面讨论时，开发该网站显然是比较合理的。这样可以对当地村镇旅游信息进行有效的整合和集中，及时的发布与更新，可以使旅客及时获得各种景点、住宿、攻略等信息，方便得使消费者获得想要的信息，同时使企业盈利。这样可以为当地经济带来更好的工作效益和社会效益，开发出本网站可以减少旅游人力成本，而且管理便捷，各功能实现迅速，以对数据实施有效的管理。管理上的有效便捷，信息得以及时获取，选择上的方便等所能达到的利益是不可估量的，这样帮助村镇实现信息化的同时帮助他们宣传家乡文化、提高收入。所以，从经济这个角度来说，可以进行该网站的建设。

2.1.2 技术上的可行性

技术是否成熟关乎该网站能否可以顺利进行，是不可忽视的一个重要方面。技术可行性的研究是指的目前而言已知的各项技术中哪些可以高质量的完成对该软件系统的开发。同时要兼顾已有的软硬件配置的要求。就本系统而言，需要软硬件配置较高的 PC 机，Windows 操作系统，另外，需要安装 MySQL 数据库服务器，JDK、MyEclipse、Tomcat，任意浏览器，配置好基本的所需环境以后，采用合 Java、JSP 语言进行旅游网站的设计、开发与运行。从技术上来说，这些基本的软件系统开发所需技术是不成问题的。

2.1.3 操作上的可行性

首先，进行网站设计的几种软件都具有较高的普及性和较高的可操作性，设计人员可以较为轻松地使用软件对旅游网站进行设计和完善。第二，管理员可对客户进行验证，对信息平台的信息进行发布、修改、管理，还可以实现酒店预订、修改、删除等各种功能。第三，客户可以在提示下进行一系列的留言、预订、查看等各项功能。而这些功能都是比较简单可行的，这些功能也比较普遍化，一般人员胜任是没有问题的。这样说来，从操作上来说说是可行的。

结合以上分析与介绍，从经济、技术、操作等方面来说，现有的有效环境和条件、技术是可以开发出一个具有较高可操作性和功能强大的个性化旅游网站，实现各项功能，

完成测试。

2.2 开发结构分析

随着网络技术和浏览器功能的完善，上世纪 90 年代末出现了 B/S(Browser/Server)结构，并成为当今 web 开发的主流 B/S 结构是指客户端(不需要专门软件，主要由 Web 浏览器组成)通过 TCP/IP 协议和 HTTP 协议的支持，通过浏览器访问 Web 服务器和后台数据库的体系结构。

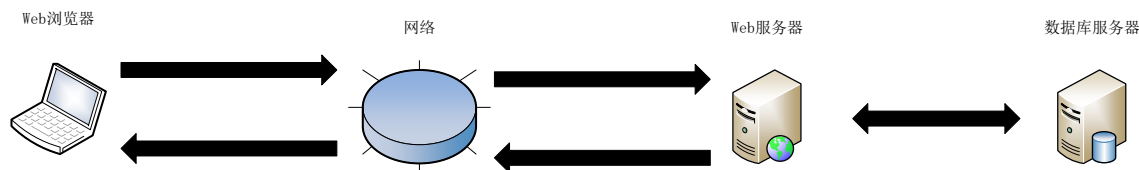


图 2.1 B/S 结构示意图

图 2.1 所示为 BS 结构的三层示意图，Web 浏览器作为客户端，可以认为是应用服务器，是表示层，Web 服务器作为应用层，数据库服务器为数据层。客户端可以在不使用任何用户程序的情况下访问系统，完成对数据的查询、添加、删除和修改。系统发现有涉及到数据库的相关指令时，首先将指令发送到 Web 服务器，数据库执行 Web 服务器发送过来的指令，执行完毕后将结果返回 Web 服务器，用户就可以看到结果。

B/S 体系结构的优点：不需要专门的软件，对操作系统也没有要求，只要有浏览器就可以，提高了用户可选择性，降低了客户机的存储空间和内存应用；在客户端实现了零维护，容易进行系统扩展；维护简单，只需要在服务器端进行就可以，不会因为用户规模影响工作量；具有更好的网络支持，扩展程度高。

当然，B/S 体系结构也有其自身的劣势：服务器负担较重，所有的逻辑业务和数据业务都在服务器端进行，服务器性能成为系统瓶颈；建立在广域网上，面对不可知的用户，安全风险控制能力相对较低。

本系统设计采用“浏览器+Web 服务器+中间层+数据库”多级分布体系结构，来源于对 B/S 体系结构的优化，继承其优点，同时对完善了不足。本项目的系统结构，如图 2.2 所示。

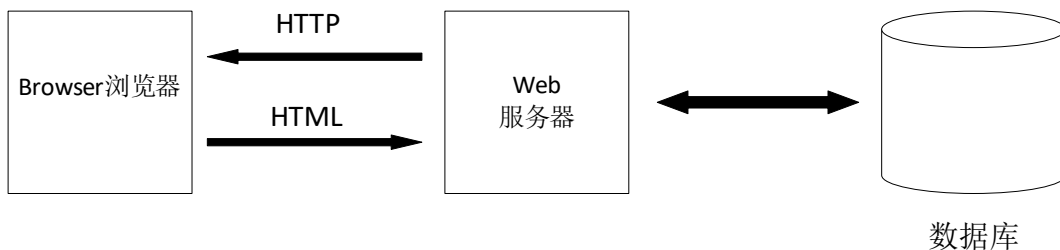


图 2.2 系统结构示意图

2.3 功能需求分析

需求分析在软件开发周期中起着十分重要的作用也可以说起着决定性作用，它是每一个软件产品在设计与实现之前必须进行的。当然了，该项工作如此重要，一定也有它的难度。在进行需求时，了解用户的需求是最关键的。旅游网站面向广大网络在线用户，它的特点就在于电子科技的广泛应用。根据该特点，我们可以通过网上调查、问卷调查、走访等方式了解网站需求。如何满足广大用户需要，尽可能多的细化网站功能就是摆在眼前的一个问题。同样的，不光用户需求需要注意，了解管理者的需求也同样很重要，一般来说，网站后台的主要对象主要是系统管理者，为了满足管理者的需求，开发出合适产品，在实现网站之前，需要提前与管理者进行沟通，需要系统实现什么功能，整合统计什么数据，完成哪些便捷操作。最后，还应该使网站具有较好的系统伸缩性和容错性。因为不管是用户还是系统管理者，他们的需求会根据实际情况，时间推移发生不断的变换，所以，系统应该尽可能提高容错性和可维护性，避免后期修改消耗人力物力。下面对该网站需求功能进行研究探讨：

旅游网站的功能比较全面丰富，尽量满足大部分旅游客户的要求，在网站的首页显示了该网站的具体实现模块部分。包含旅游人文、旅游故事、酒店信息、旅游景点、旅游地图查询、在线评价、用户注册、后台管理等功能：后台管理包括：旅游人文、旅游故事、酒店信息、特色小吃、旅游景点、路线选择、出行搭配（包括火车、高铁、飞机、自驾游等）、酒店预订，在线留言、用户增加删除修改、系统等的管理。

上边所述的功能是基本的旅游网站所需功能，它涵盖了一般客户的需求，也是此系统的主体功能。此外，在开发过程中还要不断的完善和修改，增加一些个性化和细节功能，使得系统具有很好的完整性和可操作性。

系统管理：管理员登录系统后台，可以对系统页面，功能，数据信息和数据库进行

操作。

用户管理：由管理员在后台进行管理，主要包括新用户的添加，用户权限的更改。

旅游景点：所有用户可以使用，可以浏览景点，点评景点等。

旅游故事：所有用户可以使用，可以浏览与景点有关的典故、传说，进行点评等。

酒店信息：所有用户可以使用，可以浏览酒店，订约酒店，点评酒店等。

数据查询：数据查询支持生成报表进行打印。不同用户的数据查询功能权限不同，普通用户只能查询景点与酒店信息，管理员用户可以查询所有进入景点与入住酒店的人员以及景点和酒店的信息，可以依据景点，酒店，浏览景点、入住酒店等关键词进行查询。

2.4 数据流图

进行业务流程分析也是一个重要环节。他的分析应该参照信息流动的整个过程，对每一个环节进行业务处理、数据流图的调查分析。数据流图可以清除的表明计算机软件系统的工作情况，他是一种图形表示。这样，可以简化工作难度，便于开发人员准确表达交流。同时便于开发端，维护端，测试端，应用端的相互交流合作。也就是说，数据流图不仅适合专业人员读取了解方便，也同样适用于非专业人员。在需求分析中是一种便于修改和交流的表达工具。

数据流程分析，如图 2.3 所示：

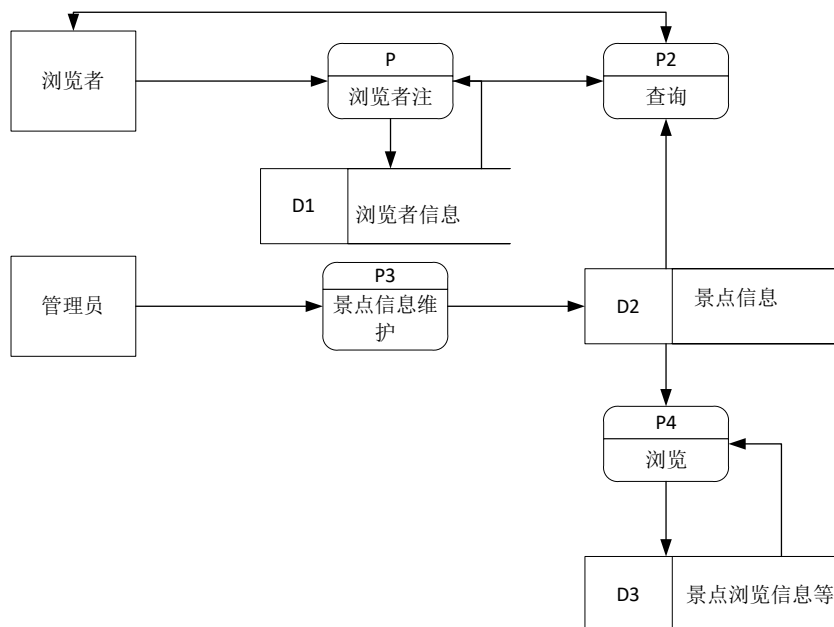


图 2.3 用户流程图

2.5 业务流程分析

业务流程分析在需求分析中的占有十分重要的分量。业务流程分析其实就是对产品业务处理过程的详细细节进行反应，也就是信息的流动过程的具体描述。业务流程分析需要根据具体业务画出流程中的具体活动，还有各个活动相互间的联系与逻辑，流程的输入输出以及流程的参与者等等。流程分析方法有顺向法、查询法、具体分析法、逻辑推理法。

业务流程分析可以帮助系统开发人员理清工作流程顺序，同时更好的分析业务相互间联系的因素关系。了解网站具体要求和业务需要，从而形成科学的、合理的工作管理和运作方式。而具体流程图的实现过程还具有相应的规范和标准，一般具有所规定的特定的一些符号、方框、箭头和线段来表示。具体如图 2.4 所示。

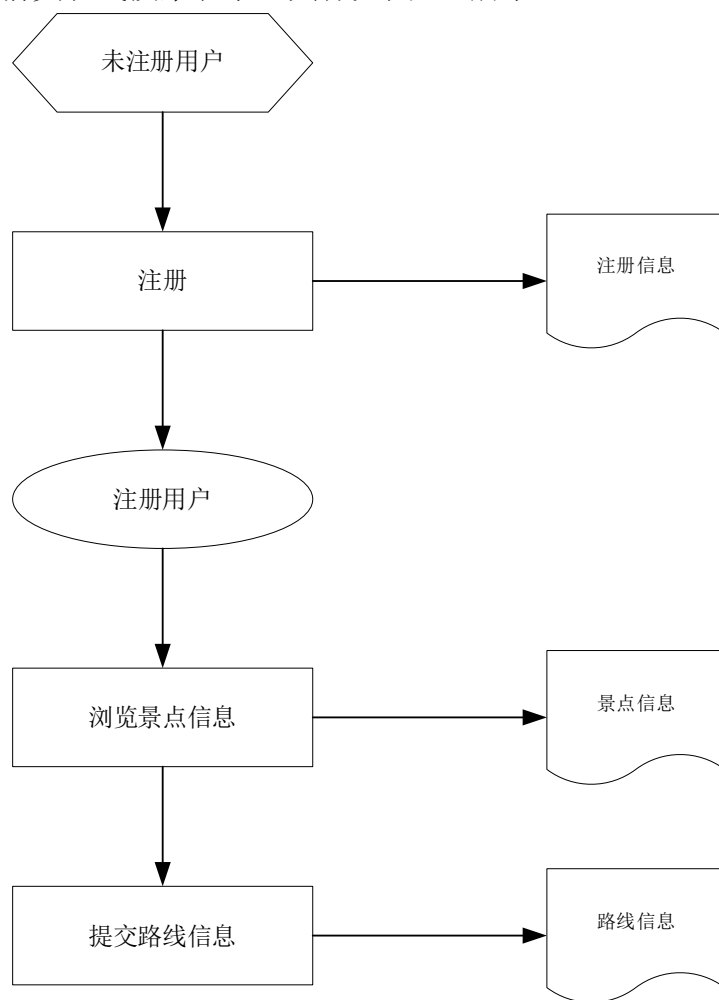


图 2.4 业务流程分析图

3 网站总体设计

3.1 系统功能描述

旅游网站的功能主要分为两部分：前台界面管理和后台数据库管理。它的主要功能包括：对旅游人文的增加删除修改查看，对旅游故事的增加删除修改查看，对酒店信息进行增加删除修改预订，对旅游景点的信息进行增加删除修改查看，还有路线查询，天气预报、用户评价、管理员后台登录管理。这些具体的板块涵盖了网站的设计的总体布局。

首页详情板块、旅游故事板块、酒店信息板块、旅游景点板块、自驾游线路板块、用户评论板块、在线留言板块、天气预报板块都有自己具体实现的操作，供客户进行选择 and 查看、管理。

3.2 系统界面设计

一个美观、清晰、操作便捷的界面可以增加用户量，吸引更多的人进行访问。通过对界面的访问，用户可以在最短时间内了解该网站的功能、属性、特点等等，同样也可以满足自己的需求。所以，在进行界面设计的时候一定要多花时间使得界面具有吸引力并且以最少的语言和图形传达最多的信息。该网站在设计界面的时候是根据以下几个方面进行的：

① 图文搭配。在进行界面设计的时候，太多的文字会使得用户觉得晦涩难懂，不愿意多花时间去进行研究。所以很有必要用图片进行展示，再加以必要的简洁的文字说明，这样用户会节省大量的时间去浏览更多网页并且可以获得自己想要的信息。旅游旅游网站界面设计采用文字加图片的形式，文字可以直观的向用户传递具体模块构造与功能，而图片可以更加形象的让用户感受到旅游地方的具体实景。

② 界面设计应当美观大方。太多的文字会使人们感觉到无聊不愿意阅读，然而太多的图片、动画也同样会使得界面太过花哨，凡事应该讲究适度原则。所以，在进行界面设计的时候切记要做到美观大方。旅游网站在设计的时候秉承这个原则，整个界面大方又不失美感。既不是一味的进行文字说明，也不是夸张的色泽艳丽。

③ 字体选用应该注意突出重点。文字说明的时候也有讲究，对于重点的内容，就应该加以特别说明，字体应该适当加粗或者换种颜色。

④ 色彩搭配应该柔和。在进行界面美化设计方面，色彩搭配也是一个很重要的环节。界面应该具有自己的主色调，而这个主色调最好采用暖色，这样可以给人一种柔和亲

切的感觉。另外，界面颜色不宜太多。与网站所宣传的内容相符合、有自己的特色就好。

3.3 系统功能结构图

旅游网站是基于 B/S 体系结构的旅游网站。通过全方位综合分析，该系统功能设计相对比较全面，能够满足大部分用户的需求。但是，做到满足每一个客户要求也是不可能的，只能是争取尽可能完善网站功能，到达更好的效果。旅游网站主要是通过各功能模块的介绍，对旅游进行宣传，包括首页，旅游人文，旅游故事等部分。本项目的系统整体框架，如图 3.1 所示。

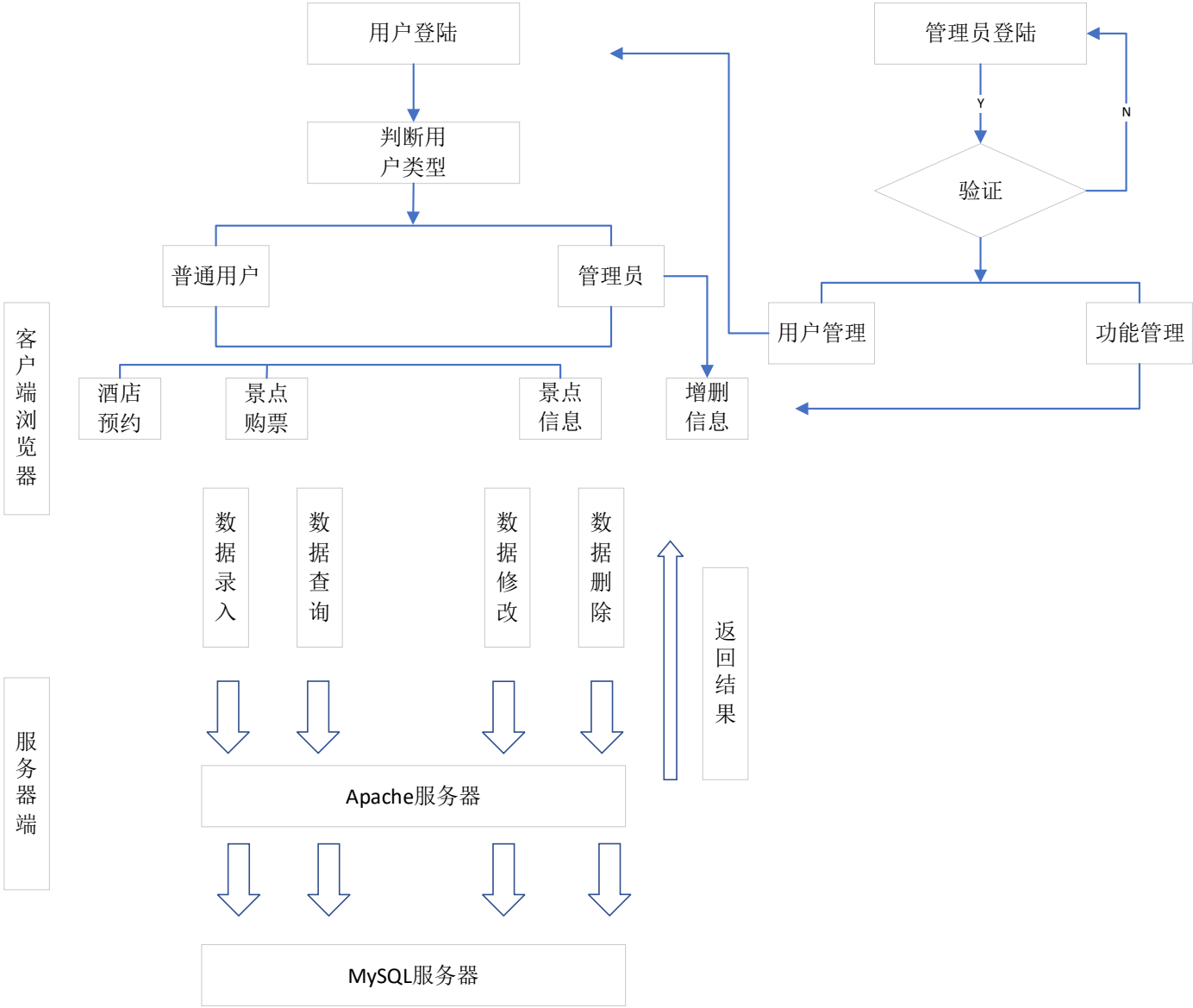


图 3.1 系统整体框图

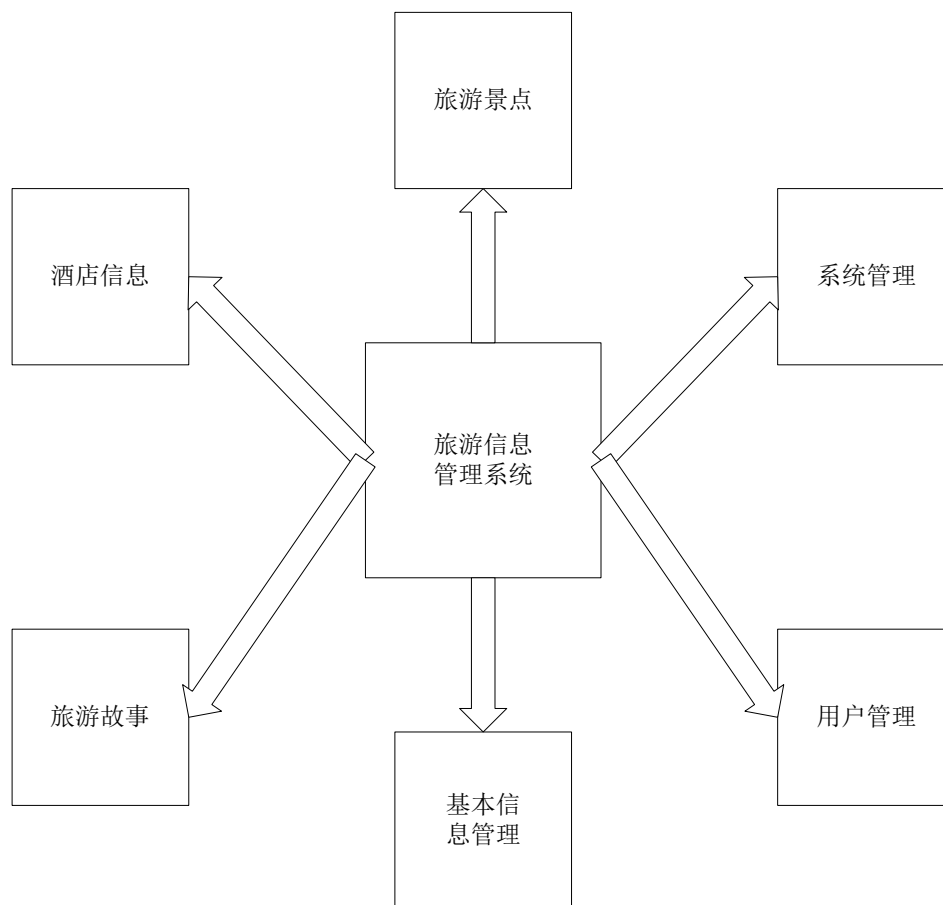


图 3.2 系统功能框图

3.4 系统设计目标

用户特点、系统属性特点和实际需求是系统目标重要的考虑因素，该网络管理系统应该尽量满足操作灵活，界面友好，使用方便等设计要求。本网站应实现下面提到的几个任务目标：

- ① 人机交互。用户可以方便快速的获取所需要的信息，同时系统必须可以给予客户及时的应答。界面设计灵活友好，数据存取安全。
- ② 对旅游网站的质量进行评估，达到预期效果，各功能完善。
- ③ 实现以上所述各部分模块的功能操作。
- ④ 设置数据检查，尽量避免数据的冗杂和出错。

3.5 系统实现的软硬件平台

① 硬件系统：硬件平台实际上就是支撑系统进行开发运行的物理设备，具有计算，输入输出、存储等功能，即计算机或 PC 机。本网站由 Windows 操作系统作为硬件支撑平

台得以实现。

② 软件服务：MyEclipse、Tomcat、SQL.

4 数据库设计

4.1 数据库概念结构设计

数据库是每一个系统所必需具备的。数据库的主要用途是实现对各种资源数据的存储，提供给该网站多种服务。通俗来讲，数据库其实就是一组具有相互关联的信息资源的有序集合，这些信息以一定的方式存放在一起，供系统使用。而对于程序来说，其与数据库又保持高度的独立性。

本项目的用户资料实体，如图 4.1 所示。

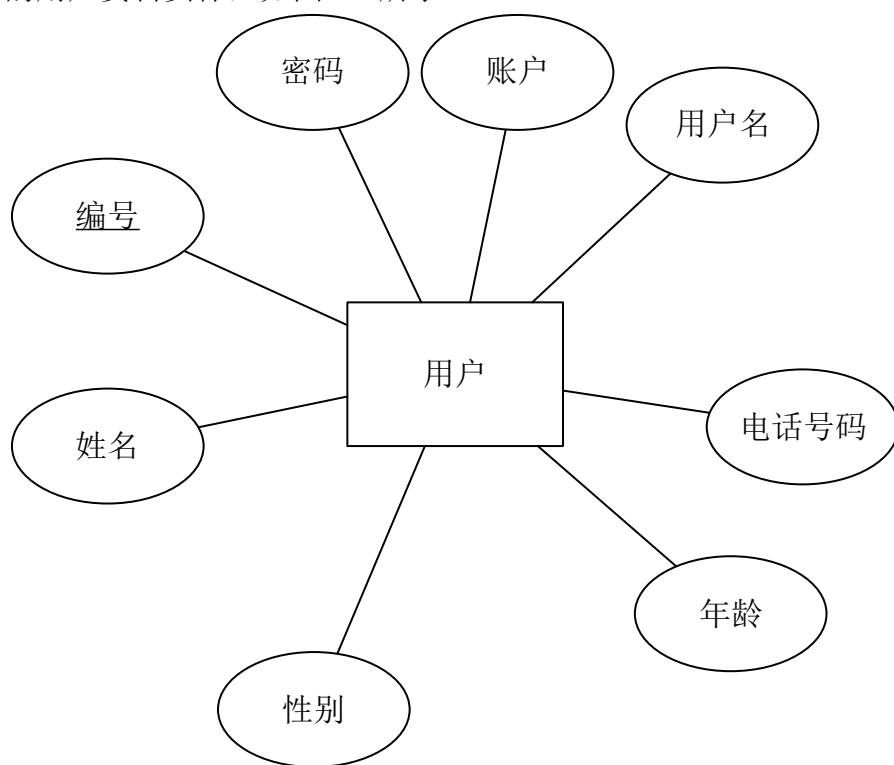


图 4.1 用户信息实体属性图

本项目的景点资料实体，如图 4.2 所示。

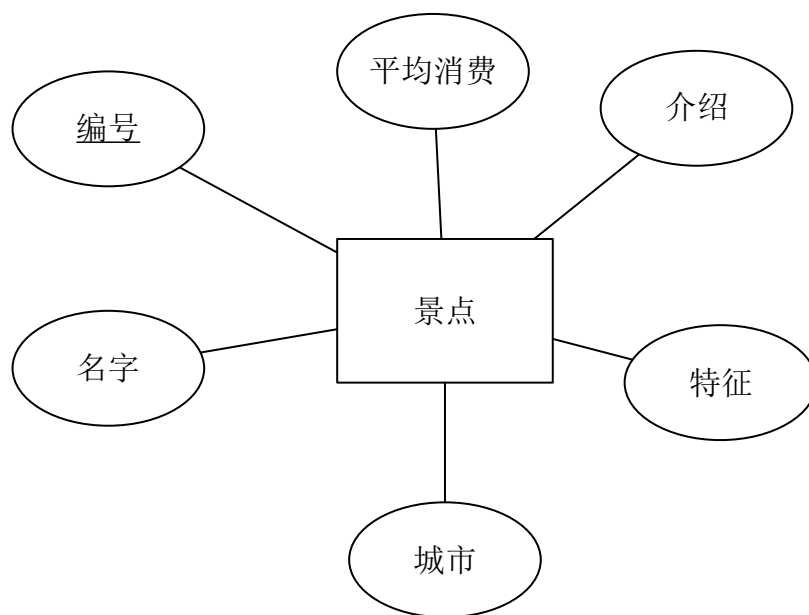


图 4.2 景点实体属性图

本项目的餐厅实体属性图，如图 4.3 所示。

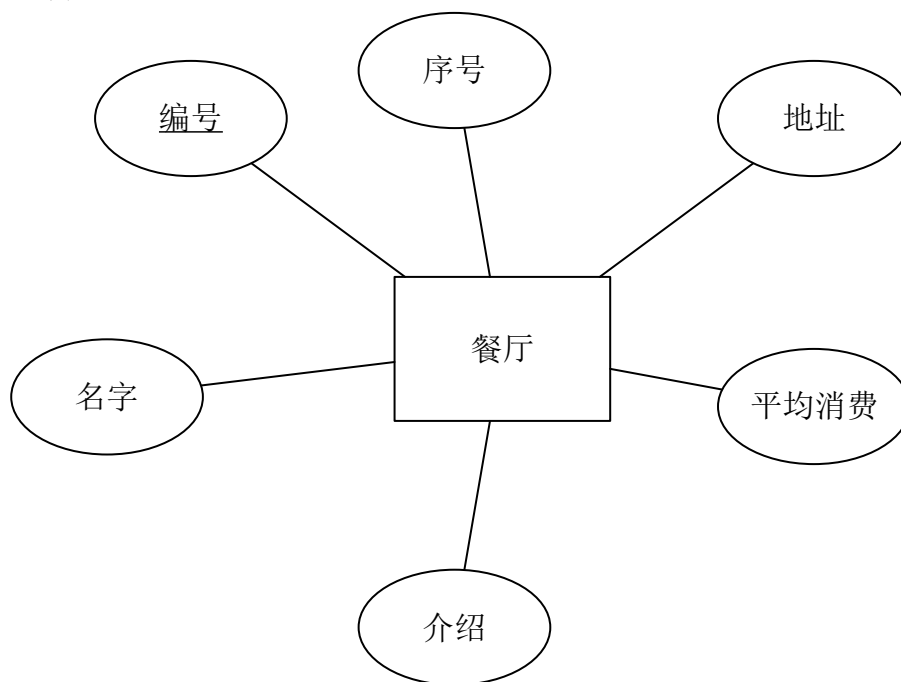


图 4.3 餐厅实体属性图

本项目的酒店实体属性图，如图 4.4 所示。

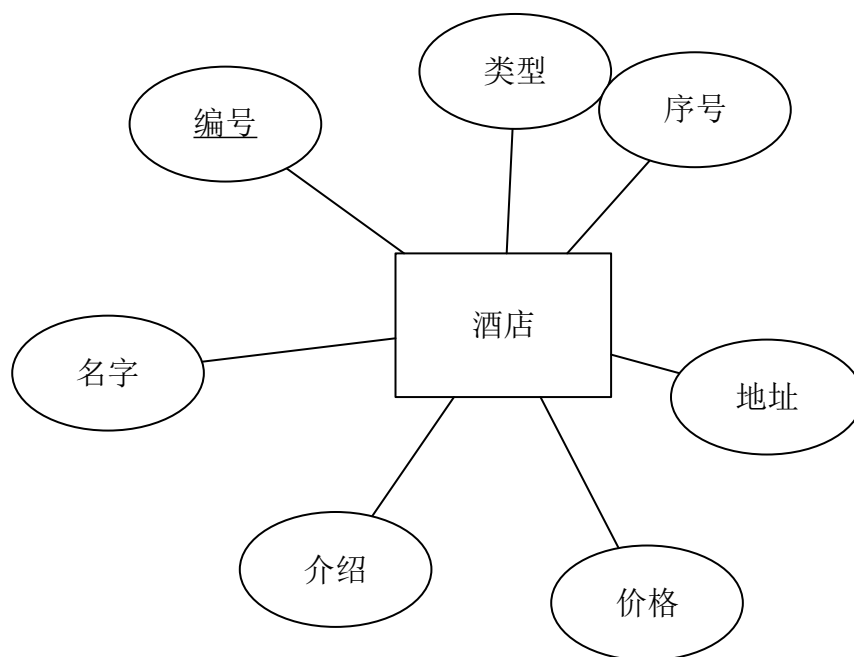


图 4.4 酒店实体属性图

本项目的数据库 E-R 图，如图 4.5 所示。

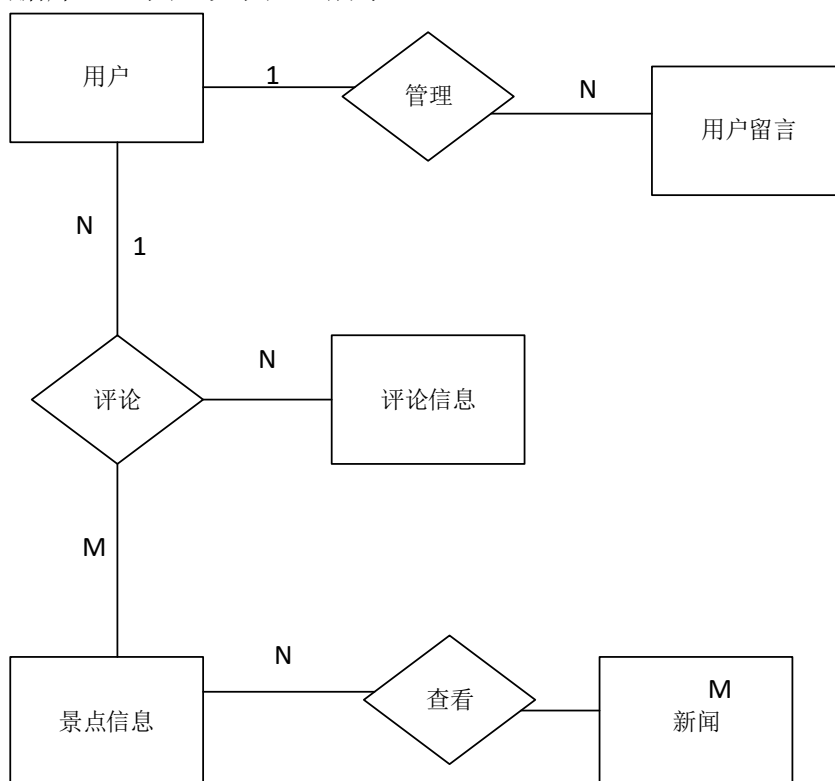


图 4.5 数据库 E-R 图

4.2 数据库逻辑结构设计

数据库其实可以理解为一个仓库，而这个仓库不像一般的仓库存放物体，这个仓库是用来存储数据信息的。而这些数据并不是杂乱无章的进行存储，它们是按照一定的结构、组织方式、方法规定将这些数据以最少的冗余度最好的方式存储在一起的数据的集合。数据库的逻辑结构就是根据 E-R 图转换而来的模型结构。数据库可以帮助系统整合数据信息，可以将数据库比作人的大脑系统，它帮助系统获取、记忆、识别信息。一个没有数据库的系统是无法完成特定功能要求的，更不能进行后台连接应用。旅游网站对数据库的逻辑结构进行个性化设计，保证数据存储的合理有效性。帮助网站进行客户数据信息的记录，整理从而使得该网站具有实践意义。

表 4.1 是用户的信息表，记录数据库中所设定的信息。

表 4.1 用户实体信息表

1	ID	Int		自增编号	
2	Id	int	11	是	
3	Name	varchar	255	否	
4	sex	varchar	255	否	
5	age	int	11	否	
6	phone	varchar	255	否	
7	username	varchar	255	否	
8	password	varchar	255	否	
9	admin	int	11	否	

表 4.2 是酒店的信息表，记录数据库中所设定的酒店信息。

表 4.2 酒店实体信息表

1	ID	Int		自增编号	
2	Id	int	11	是	
3	Name	varchar	255	否	
4	sid	int	11	否	
5	introduce	varchar	255	否	
6	type	varchar	255	否	
7	price	int	11	否	
8	address	varchar	255	否	

表 4.3 是餐厅的信息表，记录数据库中所记录的餐厅信息。

表 4.3 餐厅实体信息表

1	ID	Int		自增编号	
2	Id	int	11	是	
3	Name	varchar	255	否	
4	sid	int	11	否	
5	introduce	varchar	255	否	
6	consumption	int	11	否	
7	address	varchar	255	否	

表 4.4 是景点的信息表，记录数据库中所设定的景点信息。

表 4.4 景点实体信息表

1	ID	Int		自增编号	
2	Id	int	11	是	
3	Name	varchar	255	否	
4	city	varchar	255	否	
5	introduce	varchar	255	否	
6	consumption	int	11	否	
7	address	varchar	255	否	
8	characteristic	varchar	255	否	

4.3 本章小结

在旅游网站的设计和实现过程中，对于后台管理的设计，首先是对数据库进行数据的概念结构进行设计，之后又对数据库的逻辑进行设计。这是因为一个适合本系统的数据库对于软件开发的十分重要的，所以本网站在数据库设计过程中花费了很长的时间，争取将数据存放在一个结构清晰，安排合理的数据库中，这样就可以有效的存储，满足用户要求。

5 系统详细设计与实现

5.1 主界面模块设计

旅游网站的首页界面设计以简单、便捷、清晰为主。色彩搭配柔和，最上边包含各个模块的链接，另外还有系统公告、站内新闻，村镇旅游景点图片，友情链接等部分组成。旅游主界面整体结构比较单调大方，网页最上边以不同色彩的字体显示网站的名称，并配有网站特征标志，而在界面的名称下边包括系统公告和用户登录、站内搜索、友情链接。另外，界面上还显示了站内新闻、酒店推荐和系统简介。除此之外，界面还配置相应

的图片，显示部分景点风景。还有推荐酒店的具体详细图文。系统用户将光标移动到对应模块并进行单击时候，都会跳转到相应的另一个网页。例如，当单击旅游人文时，页面会进入到对应的旅游人文介绍网页界面。类似的，单击旅游故事时，页面也会进入到旅游故事介绍网页界面。点击酒店信息时，界面下端会出现旅游网站所有酒店信息供用户选择，包括酒店星级，编号，名称，电话，照片，操作等。而同样在旅游景点界面中，包括该旅游景点星级，名字，具体方位，简介，图片还有旅游价格。点击天气预报，出现天气预报界面，将鼠标移动到指定省或者地方之后会显示该地具体天气预报，点击旅游地图查询，可以查看各个地方旅游路线。点击在线评价模块，用户就能够对该网站服务进行评价或者给出意见。在用户申请模块中，用户可以根据自己的实际情况进行基本信息的设置与提供，等待管理员审核通过之后方能登录。登录后，可以对该网站的各个功能进行管理，进行增加删除修改预订等功能。下图中显示的是旅游网站的首页，点击各个板块之后运行效果也如图 5.1 首页效果图、图 5.2 首页详情、图 5.3 首页酒店信息、图 5.4 首页景点信息等。



图 5.1 首页效果图

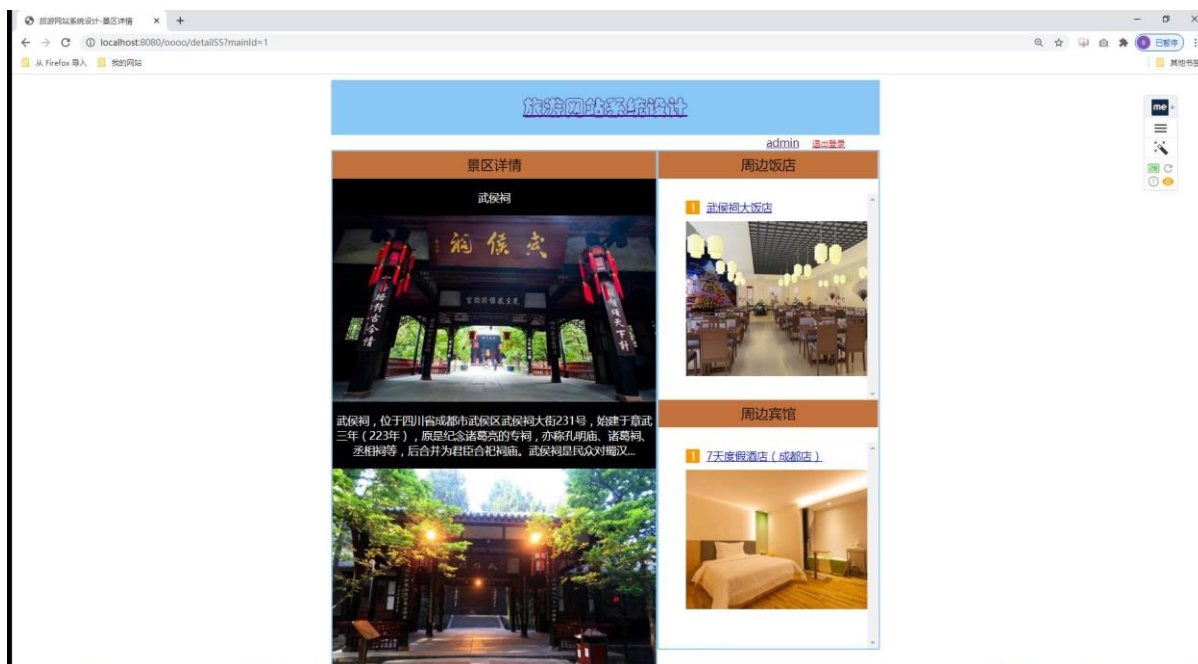


图 5.2 首页详情

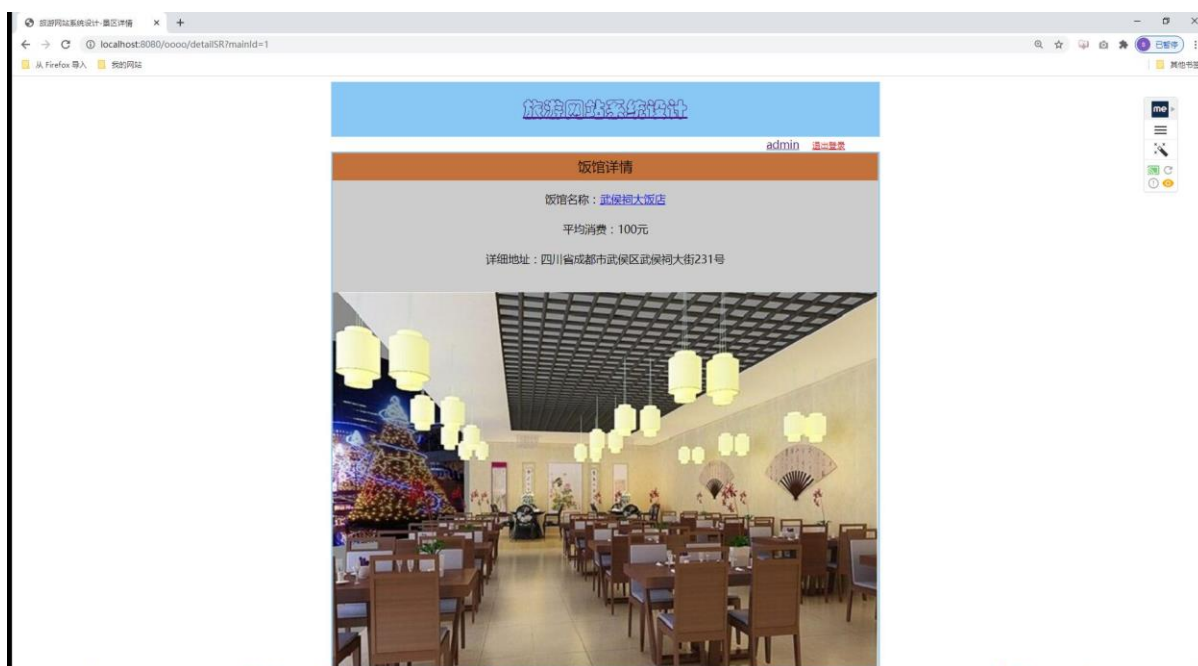


图 5.3 首页酒店信息

5.2 新增景点设计

点击“新增按钮”可以进行新用户添加，页面显示如图 5.4 所示，其中带红色星号的数据为必填数据，页面提交之前会对该项数据进行判断，是否为空，数据类型是否正

确，如果为空或数据类型不正确，无法完成页面提交。正确填写用户信息后，提交于服务器，储存在数据库中的用户信息表内。主要代码如下：

```
protected void Submit1_Click1(object sender, EventArgs e)
{
    HttpFileCollection files = HttpContext.Current.Request.Files; //提供对上传文件的访问
    string filename; //文件名
    for (int iFile = 0; iFile < files.Count; iFile++) //对上传文件的遍历
    {
        HttpPostedFile postFile = files[iFile]; //读取上传文件的访问
        filename = System.IO.Path.GetFileName(postFile.FileName); //获取到文件名
        if (filename != "")
        {
            postFile.SaveAs(Request.MapPath("~/upload_files/" + filename)); //保存上传来的文件
        }
        SqlConnection Myconnection = new SqlConnection(Connction);
        Myconnection.Open();
        string strCmd = "insert into New values("
            + TextBox3.Text.Trim() + ","
            + TextBox2.Text + ","
            + TextBox1.Text + ","
            + content1.Value + ","
            + TextBox4.Text + ","
            + TextBox5.Text + ","
            + filename + ","
            + 0 + ","
            + "getdate()" + ","
            + DropDownList1.SelectedItem.Text + ","
            + 0
            + ")"
            ;
        SqlCommand Mycommand = new SqlCommand(strCmd, Myconnection);
        int nResult;
        nResult = Mycommand.ExecuteNonQuery();
        ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert(' " + "填加信息成功，请妥善保管好你的信息！ " + "');</script>");
        Myconnection.Close();
    }
    TextBox1.Text = "";
    TextBox2.Text = "";
    TextBox3.Text = "";
    content1.Value = "";
}
```




图 5.4 添加新景点

5.3 登录模块设计

系统登录界面是进入系统的入口，使用人员和管理人员都要通过入口登录后才能进行使用和管理。登录页面采用 JavaScript 进行账号和密码的验证，当账号和密码同时不为零时，登录按钮才能够起作用，向系统提交数据。用户登陆流程图如图 5.5 所示，登陆界面如图 5.6 所示。

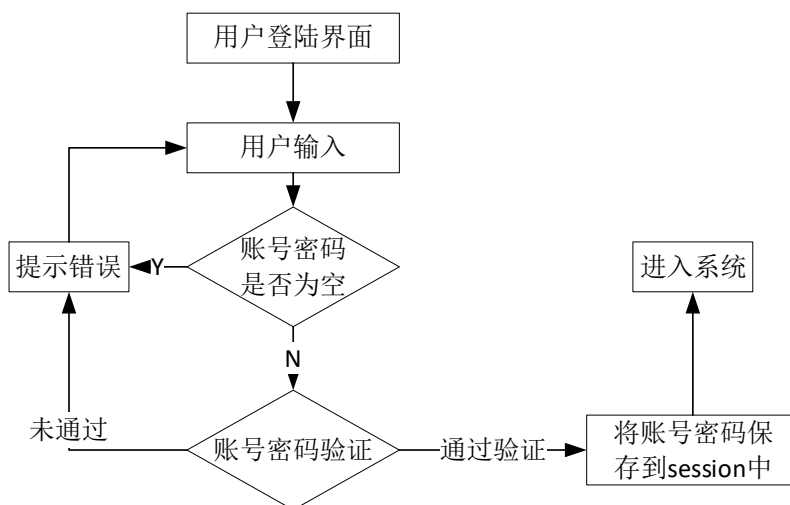


图 5.5 用户登录流程图



图 5.6 系统入口设计示意图

5.4 用户信息和密码修改

新用户获得用户名密码后，可以在登录系统后的个人中心里完成用户资料和密码的修改，用户修改信息流程图如图 5.7 所示。

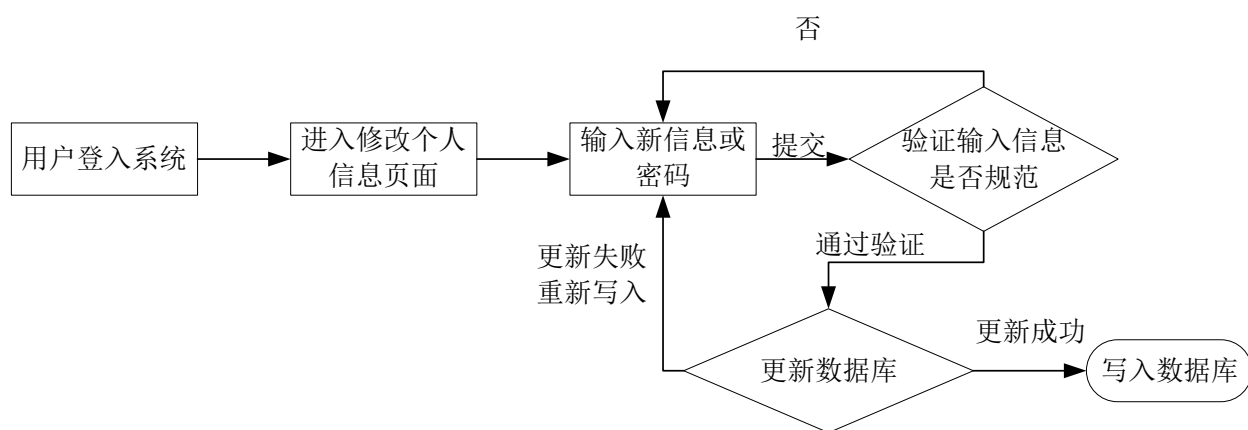


图 5.7 用户信息修改流程图

5.5 Web 客户端对 MySQL 的访问设计

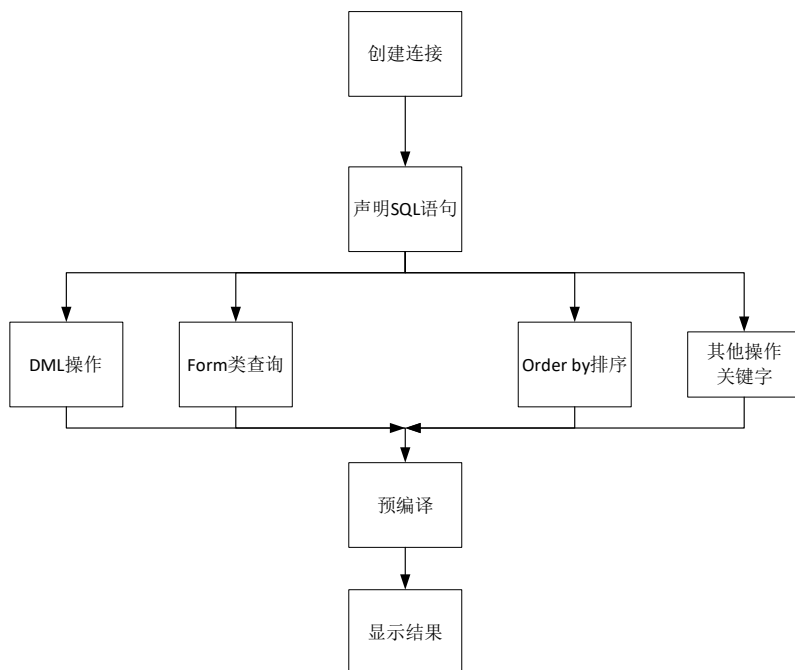


图 5.8 数据库操作流程

系统用户使用 Apache 服务器提供的 WWW 服务，可以在 Web 页面上对基于用户输入数据库的信息进行添加，更新和删除处理。从 Web 上访问数据库主要分为 6 个步骤，具体如图 5.9 所示。

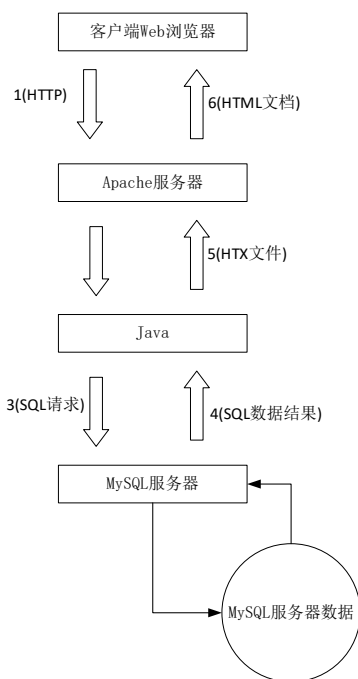


图 5.9 Web 页面与 MySQL 的交互

6 系统测试

完成系统设计和开发后，在正式投入使用前需要进行系统测试，对系统的功能进行检验，看是否满足需求，这是保证系统质量的必要步骤。系统测试分为白盒模式和黑盒模式两种，白盒模式侧重系统内部结构，黑盒模式侧重于系统外部功能。针对本系统的测试是采用黑盒模式进行，即把系统当做一个打不开盒子，不考虑内部结构，只考虑输入和输出，查看系统功能是否满足需求。

6.1 系统测试环境

系统测试环境应当尽量与使用环境一致，避免因环境差异造成的测试误差，使一些 bug 无法发现。表 6.1 和表 6.2 分别是系统对客户端和服务器的软硬件要求。

表 6.1 系统测试的硬件环境

硬件名称	详细要求
服务器	CPU 2 Ghz 以上，内存 2G 以上，硬盘 300G 以上
客户端	具有网络功能的 PC
网络	基于单位内部局域网，含路由交换等设备

表 6.2 系统测试的软件环境

软件名称	详细要求
服务器端操作系统	Ubuntu 18.04
服务器端数据库系统	MySQL 8.0
服务器端应用程序服务器	Apache 2.4
服务器端脚本解释器	Java 1.8
客户端操作系统	Windows 10
客户端浏览器	Chrome 86 及以上版本

6.2 系统测试计划

测试工作分为两个阶段，功能测试和集成测试，表 6.3 给出了各阶段需要完成的任务。

表 6.3 测试阶段任务安排

序号	测试阶段	测试方法	完成工作	测试时间
1	功能测试	黑盒	用户和管理员功能是否正常使用	5 天
2	集成测试	黑盒	测试业务流程和数据流向是否正确，完成模块间组合测试	5 天

① 功能测试阶段。对于网络应用软件系统来说，其功能测试阶段一般分为用户连接测试、界面测试、链接测试和数据库测试等方面。用户连接测试用于测试是否正确用户信息能否登入系统，错误用户信息是否被拒绝登陆。界面测试主要分为两个方面，一方面，各个窗口页面实现功能是否与功能模块划分对应，是否能准确完成页面功能；另一方面，测试页面所包含的所有元素，例如按钮、图表和文字等是否正确显示，页面提交后表格数据是否正常，与设计初衷是否保持一致等。数据库测试用于检测页面操作与数据库的同步，检查页面提交内容是否如实记录在数据库表中，数据库表之间的信息是否一致等。在这一阶段，可以请实际应用系统的人参与测试，体验系统操作界面和操作流程，提出改善意见，完善系统开发。

② 集成测试阶段。集成测试阶段主要分为三个部分，集成测试阶段需要完成功能集成测试、模块内部的集成测试和整个系统集成测试三个部分的测试任务，由于三者是相互关联甚至互相包含的，因此，在实际测试过程中，可能一个测试用例会包含三个测试部分的内容，无法严格区分单独用于某一个部分的测试用例。

功能界面集成测试指在单个界面功能测试的基础之上，增加测试互相影响的数据是否正确的测试用例进行测试。模块集成测试指模块通过了各自的集成测试，按模块组合执行测试用例，在覆盖界面操作功能的同时，加入各个界面之间跳转以及数据存储逻辑的测试用例进行测试。系统集成测试按系统组合执行测试用例，基于模块集成测试的基础，加入不同模块之间业务逻辑关联的测试用例进行测试。

在上一阶段的测试工作中，界面功能测试主要着眼于单个页面的图表、按钮等的测试，而在功能界面的集成测试中，加入了不同界面相互之间可能有影响的测试用例。模块内的集成测试主要在同一业务逻辑处理模块的内部，完成各自的集成测试内容，不涉及与其他模块之间的操作相关的测试内容。系统整体集成测试主要关注各个模块之间的相互影响，按照模块之间的松散耦合相关性，以及系统需求分析和设计中的业务操作流程，设计响应的测试用例，验证系统的正确性。

与系统功能测试阶段一样，系统集成测试也采用黑盒的方式，该方式将系统认为是密闭的盒子，不需要关系内部结构，只看是不是能正确执行系统功能。

6.3 测试过程和结果分析

6.3.1 功能测试

① 测试过程

基本的功能测试表如表 6.4 所示。

表 6.4 功能测试表

序号	测试方法和过程描述	预期结果	实际结果
1	管理员用户登录，输入正确的用户名、密码	成功	成功
2	管理员登录，输入错误的用户名、密码	失败	失败
3	普通用户登录，输入正确的用户名、密码	成功	成功
4	普通用户登录，输入错误的用户名、密码	失败	失败
5	切换左侧菜单，看右侧页面是否显示正常	正确	正确
6	修改密码，两次输入密码不一致	失败	失败
7	修改密码，两次输入密码一致	正常	正常

在测试过程中，每项操作都给出了预期结果和实际结果，当两者不一致时，系统出现错误，需要进行调试。

② 测试结果分析

测试过程中，预期结果和实际结果一致。通过测试发现系统界面友好，易于上手，提高了用户浏览景点以及进行相关操作的效率。

6.3.2 集成测试

在系统集成测试过程中，主要完成了用户信息管理模块，景点信息模块，新增景点模块，表 6.5 是系统集成测试表。

表 6.5 系统集成测试表

测试功能	测试具体描述	预期结果	正确
管理员设置某用户不能登录系统	管理员设置某用户不能登录系统，查看数据库中的表数据是否正确	正确	正确
	该用户登录系统，能否登入	失败	失败
添加景点信息	管理员用户添加景点信息、酒店信息，查看数据库中表数据是否正确	正确	正确
	系统管理员查看所有酒店信息，是否有变化	正确	正确
景点订票/酒店预订	用户提交新的申请，表单必填元素不全	失败	成功
	用户提交新的申请，表单元素齐全正确	成功	成功

从表中可以看到，系统测试有预期结果和实际结果不一致的地方，需要进行调试，确保系统的正确性。经过修改程序反复调试后解决了系统 bug。通过功能测试和集成测

试，发现系统在操作界面和数据方面不存在太大问题，可以投入使用。

7 项目总结

本文选题从景区当前管理实际存在的难点、痛点出发，结合主流的 MyEclipse 开发平台，采用 B/S 结构，完成了基于 Java + MySQL 的景点信息管理系统的设计。首先介绍了项目的背景，对比两种体系结构和三种开发平台，确定系统开发方案，并对相关的知识进行简单的介绍；之后根据用户需求确定功能逻辑，开发系统界面进行人机交互；重点进行了数据库和表的设计，合理存储数据；最后进行界面和数据库的整合，完成系统并进行测试。通过测试表明，采用本系统可以提高景点管理的科学化水平，有助于提高景区常规管理工作和信息化建设。

所做的工作主要有：

- ① 依据用户需求和景区管理实际流程，确定功能的逻辑，通过软件编程实现。
- ② 从技术层面和商业运用上对 B/S 和 C/S 体系结构进行比较，以数据安全性，实时性，系统的投入和扩展性等方面确定系统采用 B/S 结构。
- ③ 比较主流的网站系统开发平台，从开发难度，时间，扩展性，经济成本角度出发确定系统采用 Tomcat 平台。
- ④ 搭建开发环境，采用 HTML 开发前台显示界面，嵌入 JavaScript，使用控件实现数据校验功能，例如登录用户用户名密码验证，数据类型是否一致，出现错误时及时提示，确保数据的正确性。
- ⑤ 设计数据库和表，通过表中字段值的设计构建各个表的相互联系，使数据的调用和查询更加方便快捷。
- ⑥ 使用黑盒方式对系统进行功能测试，包括登录模块，界面显示，数据库存储等，测试结果表明系统运行稳定，达到预期目标。

附录

核心代码摘录：

```
using System;  
using System.Collections;  
using System.Configuration;  
using System.Data;  
using System.Linq;  
using System.Web;
```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class news_manager_SingleNewsInfo : System.Web.UI.Page
{
    int NewID;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request.Params["NewID"] != "")
        {
            NewID = Convert.ToInt32(Request.Params["NewID"]);
        }
        if (!IsPostBack)
        {
            GetSinUser();
        }
    }
    public void GetSinUser()
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        try
        {
            conn.Open();
            string strCmd = "select * from new where NewID=" + NewID.ToString();
            SqlCommand comm = new SqlCommand(strCmd, conn);
            SqlDataReader dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
            if (dr.Read())
            {
                Label20.Text = dr["NewID"].ToString();
                TextBox1.Text = dr["Title_Fu"].ToString();
                TextBox2.Text = dr["Title"].ToString();
                TextBox3.Text = dr["Author"].ToString();
                TextBox4.Text = dr["Txshe"].ToString();
                TextBox5.Text = dr["Beizhu"].ToString();
                Label21.Text = dr["AddTime"].ToString();
                Label22.Text = dr["NewFl"].ToString();
                Label23.Text = dr["Title"].ToString();
                Label24.Text = dr["News"].ToString();
                if (dr["Picture"].ToString() != string.Empty)
                {
                    Literal1.Text = "<img src=\"../upload_files/\" + dr[\"Picture\"] + \"\" width=\"300px\" height=\"200px\" alt=\"\" align=\"center\" />";
                }
            }
        }
        catch (SqlException ex)
        {
        }
    }
}

```



```

        Page.ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert('" + ex.Message +
        "')</script>");
    }
    finally
    {
        conn.Close();
    }
}

```

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

```

```

public partial class news_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        jingdian();
        ziyuan();
        TeacherNew();
        JiaoxueNew();
        JiaokeNew();
        StudentNew();
        mangguo();
        gonggao();

        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select * from NewsFl");
        //DataList1.DataSource = ds;
        //DataList1.DataBind();
    }
    private void jingdian()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='旅游常识' order by AddTime DESC");
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
    private void ziyuan()
    {
        Class1 c1 = new Class1();

```

```

        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='华坪资源' order by AddTime DESC");
        GridView2.DataSource = ds;
        GridView2.DataBind();
    }

    private void TeacherNew()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='精品线路' order by AddTime DESC");
        GridView3.DataSource = ds;
        GridView3.DataBind();
    }

    private void JiaoxueNew()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='景点介绍' order by AddTime DESC");
        GridView4.DataSource = ds;
        GridView4.DataBind();
    }

    private void mangguo()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='芒果之乡' order by AddTime DESC");
        GridView5.DataSource = ds;
        GridView5.DataBind();
    }

    private void JiaokeNew()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='民族风情' order by AddTime DESC");
        GridView6.DataSource = ds;
        GridView6.DataBind();
    }

    private void StudentNew()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='华坪特色' order by AddTime DESC");
        GridView8.DataSource = ds;
        GridView8.DataBind();
    }

    private void gonggao()
    {
        Class1 c1 = new Class1();
        DataSet ds = new DataSet();
        ds = c1.GetInfo("select top 10 * from New where NewFl='公告' order by AddTime DESC");
        GridView7.DataSource = ds;
    }

```

```

        GridView7.DataBind();
    }
}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class news_manager_manager : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            GetNewInfo("select * from New ");
        }
    }

    private void GetNewInfo(string strCmd)
    {
        try
        {
            SqlConnection conn = new SqlConnection(Class1.connstring);
            conn.Open();
            SqlDataAdapter da = new SqlDataAdapter(strCmd, conn);
            DataSet ds = new DataSet();
            da.Fill(ds, "xcfvfxdvxc");
            GridView1.DataSource = ds.Tables[0].DefaultView;
            GridView1.DataBind();
            Label1.Text = (GridView1.PageIndex + 1).ToString() + "页";
            Label2.Text = (GridView1.PageCount).ToString() + "页";
            Label3.Text = ds.Tables[0].Rows.Count.ToString();
            if (GridView1.PageIndex == 0)
            {
                btnFirst.Visible = false;
                btnPrev.Visible = false;
                btnNext.Visible = true;
                btnLast.Visible = true;
            }
        }
    }
}

```

```

if (GridView1.PageIndex == GridView1.PageCount - 1)
{
    btnFirst.Visible = true;
    btnPrev.Visible = true;
    btnNext.Visible = false;
    btnLast.Visible = false;
}
if ((GridView1.PageIndex != 0) && (GridView1.PageIndex != GridView1.PageCount - 1))
{
    btnFirst.Visible = true;
    btnPrev.Visible = true;
    btnNext.Visible = true;
    btnLast.Visible = true;
}
if (GridView1.PageCount == 1)
{
    btnFirst.Visible = false;
    btnPrev.Visible = false;
    btnNext.Visible = false;
    btnLast.Visible = false;
}
if (!Page.IsPostBack)
{
    for (int i = 1; i <= GridView1.PageCount; i++)
    {
        DropDownList2.Items.Add(i.ToString());
    }
}
catch (SqlException ex)
{
    Page.ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert('\" + ex.Message +
    \"')</script>");
}
}

protected void GridView1_PageIndexChanging1(object sender, GridViewPageEventArgs e)
{
    GridView1.PageIndex = e.NewPageIndex;
    GetNewInfo("select * from New");
}
protected void PagerButtonClick1(object sender, EventArgs e)
{
    GridView1.PageIndex = 0;
    GetNewInfo("select * from New ");
}
protected void PagerButtonClick2(object sender, EventArgs e)
{
    GridView1.PageIndex = Convert.ToInt32(GridView1.PageIndex) - 1;
    GetNewInfo("select * from New ");
}
protected void PagerButtonClick3(object sender, EventArgs e)
{

```

```

        GridView1.PageIndex = GridView1.PageIndex + 1;
        GetNewInfo("select * from New ");
    }
    protected void PagerButtonClick4(object sender, EventArgs e)
    {
        GridView1.PageIndex = GridView1.PageCount - 1;
        GetNewInfo("select * from New ");
    }
    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        string commandName = e.CommandName;
        int nUserID = -1;
        nUserID = Convert.ToInt32(e.CommandArgument);
        if (e.CommandName == "delete")
        {
            Class1 c1 = new Class1();
            c1.DeleteUser("delete from New where NewID = " + nUserID.ToString());
            GetNewInfo("select * from New ");
        }
    }
    protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        e.Row.Attributes["onmouseover"] = "ItemOver(this)";
    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
    }
    protected void LinkButton1_Click(object sender, EventArgs e)
    {
        try
        {
            Response.Clear();
            Response.AddHeader("content-disposition", "attachment;filename=FileName.xls");
            Response.Charset = "gb2312";
            Response.ContentType = "application/vnd.xls";
            System.IO.StringWriter stringWrite = new System.IO.StringWriter();
            System.Web.UI.HtmlTextWriter htmlWrite = new HtmlTextWriter(stringWrite);
            GridView1.AllowPaging = false;
            GetNewInfo("select * from New ");
            GridView1.RenderControl(htmlWrite);
            Response.Write(stringWrite.ToString());
            Response.End();
            GridView1.AllowPaging = true;
            GetNewInfo("select * from New ");
        }
        catch (Exception ex)
        {
            Page.ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert('" + ex.Message +
            "')</script>");
        }
    }
    protected void LinkButton2_Click(object sender, EventArgs e)

```

```

    {
        GridView1.PageIndex = Convert.ToInt32(DropDownList2.SelectedItem.Text) - 1;
        GetNewInfo("select * from New ");
    }

}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class user_denglu1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    //private SqlDataReader GetUserLogin(string username, string password)
    //{
    //    string strMd5 =
    System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(Password.Text.ToString(),
    "md5");
    //    SqlConnection conn = new SqlConnection(Class1.connstring);
    //    conn.Open();
    //    SqlCommand comm = new SqlCommand("Pr_GetUser", conn);
    //    comm.CommandType = CommandType.StoredProcedure;
    //    SqlParameter Yhm = new SqlParameter("@UserName", SqlDbType.VarChar, 32);
    //    Yhm.Value = username;
    //    comm.Parameters.Add(Yhm);
    //    SqlParameter mima = new SqlParameter("@strMd5", SqlDbType.VarChar, 255);
    //    mima.Value = password;
    //    comm.Parameters.Add(mima);
    //    SqlDataReader dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
    //    return dr;
    //}

    protected void LoginBtn_Click(object sender, EventArgs e)
    {
        string userid = "";
        string strMd5 =
    System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(Password.Text.ToString(),
    "md5");
        string strcmd = "select UserID from Users where UserName='" + UserName.Text.Trim() + "' and
    password='" + strMd5 + "'";

```

```

Class1 c1 = new Class1();
SqlDataReader dr = c1.GetNewFl(strcmd);
if (dr.Read())
{
    userid = dr["UserID"].ToString();
}
string stemyzhm = Session["checkcode"].ToString();
if (userid != "" && Validator.Text == stemyzhm)
{
    Response.Redirect("~/user/SelectUser.aspx");
}
else
{
    ClientScript.RegisterStartupScript(GetType(), "", "<script>alert('请检查你的输入，用户名、密码和验证码输入是否正确!!!')</script>");
}
}
protected void CancelBtn_Click(object sender, EventArgs e)
{
}
}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class user_UpaterUser : System.Web.UI.Page
{
    int UserIDs;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request.Params["UserID"] != null)
        {
            UserIDs = Convert.ToInt32(Request.Params["UserID"]);
        }
        if (!Page.IsPostBack)
        {
            GetSinUser(UserIDs);
        }
    }
    public void GetSinUser(int Userid)

```

```

{
    SqlConnection conn = new SqlConnection(Class1.connstring);
    try
    {
        conn.Open();
        string strCmd = "select * from UserName where UserID=" + Userid.ToString();
        SqlCommand comm = new SqlCommand(strCmd, conn);
        SqlDataReader dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
        if (dr.Read())
        {
            Label1.Text = dr["Username"].ToString();
            Label20.Text = dr["UserID"].ToString();
            TextBox2.Text = dr["Username"].ToString();
            TextBox3.Text = dr["Email"].ToString();
            TextBox4.Text = dr["sex"].ToString();
            TextBox5.Text = dr["Birthday"].ToString();
            TextBox6.Text = dr["TelPhone"].ToString();
            TextBox7.Text = dr["Arear"].ToString();
            Label21.Text = dr["Zhuceriqi"].ToString();
            content1.Value = dr["prontery"].ToString();
        }
    }
    catch (SqlException ex)
    {
        ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert('" + ex.Message + "')</script>");
    }
    finally
    {
        conn.Close();
    }
}

protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(Class1.connstring);
    conn.Open();
    string strCmd = "update Username set UserName =" + TextBox2.Text + ","
        + "Email =" + TextBox3.Text + ","
        + "sex =" + TextBox4.Text + ","
        + "Birthday=" + TextBox5.Text + ","
        + "TelPhone=" + TextBox6.Text + ","
        + "Arear=" + TextBox7.Text + ","
        + "prontery=" + content1.Value + ""
        + "where UserID=" + UserIDs
        ;
    SqlCommand comm = new SqlCommand(strCmd, conn);
    int nResult;
    nResult = comm.ExecuteNonQuery();
    ClientScript.RegisterStartupScript(Page.GetType(), "", "<script>alert('更新成功！！')</script>");
}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("~/user/SelectUser.aspx");
}

```



```

}

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Data.SqlClient;

/// <summary>
/// Class1 的摘要说明
/// </summary>
public class Class1
{
    public static string connstring = "Data Source=.;Initial Catalog=yy;User ID=sa;pwd=sa;";//字段
    /// <summary>
    /// 使用DataAdapter查询数据
    /// </summary>
    /// <param name="strcmd">SQL语句</param>
    /// <returns></returns>
    public DataSet GetInfo(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlDataAdapter da = new SqlDataAdapter(strcmd, conn);
        DataSet ds = new DataSet();
        da.Fill(ds, "chaxun");
        return ds;
    }

    public int AddDatabase(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand comm = new SqlCommand(strcmd, conn);
        int nResult = comm.ExecuteNonQuery();
        return nResult;
    }
    /// <summary>
    /// 删除数据
    /// </summary>
    /// <param name="nUserID"></param>
    /// <returns></returns>

    public int DeleteUser(string strcmd)
    {
        int nResult;
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
    }

```

```

        SqlCommand MyCommDel = new SqlCommand(strcmd, conn);
        nResult = MyCommDel.ExecuteNonQuery();
        return nResult;
        conn.Close();
    }
    /// <summary>
    /// 使用ExecuteReader()方法查询数据
    /// </summary>
    /// <param name="strcmd"></param>
    /// <returns></returns>
    public SqlDataReader Getdr(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand comm = new SqlCommand(strcmd, conn);
        SqlDataReader dr;
        dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
        return dr;
    }
    /// <summary>
    /// 功能:使用DataReader查询数据
    /// </summary>
    /// <param name="strcmd">Sql语句</param>
    /// <returns></returns>
    public SqlDataReader GetNewFl(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand comm = new SqlCommand(strcmd, conn);
        SqlDataReader dr;
        dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
        return dr;
    }
    public DataSet GetChacun(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlDataAdapter da = new SqlDataAdapter(strcmd, conn);
        DataSet ds = new DataSet();
        da.Fill(ds, "yyy");
        conn.Close();
        return ds;
    }
    public int Caozuo(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand com = new SqlCommand(strcmd, conn);
        int nResult = -1;
        nResult = com.ExecuteNonQuery();
        conn.Close();
        return nResult;
    }

```

```

    }
    public SqlDataReader GetDd(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand comm = new SqlCommand(strcmd, conn);
        SqlDataReader dr;
        dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
        return dr;
    }
    public SqlDataReader GetNewffl(string strcmd)
    {
        SqlConnection conn = new SqlConnection(Class1.connstring);
        conn.Open();
        SqlCommand comm = new SqlCommand(strcmd, conn);
        SqlDataReader dr = comm.ExecuteReader(CommandBehavior.CloseConnection);
        return dr;
    }
}

```