《测试文档》

1. 编写目的

编写测试文档作为指导此测试项目循序渐进的基础，帮助我们安排合理资源和进度，避免可能的分险。本文档有助于实现以下目标：

确定现有项目的信息和应测试的软件结构。

列出推荐的测试需求

推荐可用的测试策略，并对这些策略加以详细说明

确定所需的资源，并对测试的工作量进行估计

列出测试项目的可交付元素，包括用例以及测试报告等。

1. 背景

随着人们知识层次的提高，旅游成为日常生活中不可缺少的一部分。而旅游数

和业务量庞大，仅仅靠传统的人工管理是不可行的。旅游管理系统应运而生，逐渐成为信息化建设的重要组成部分。旅游管理系统为各游客提供了详细信息，以及旅游社的详细情况，提供旅客对网上订票功能进行合理操纵。

项目名称：旅游后台管理系统。设计的目的：为游客和旅行社之间提供一个友好的、直接的交流平台。

开发者人员：组长：杨天超；组员：孙锦喆、王小倩、杜娣

1. 定义

主键：每一笔资料中的主键都是表格中的唯一值。换言之，它是用来独一无二的确认一个表格中的每一行资料。

外键：设表t1，t2中都有一个name字段，而且是t1的主键，那么如果设t2中的name为外键的话，向t2中添加数据的时候，如果name值不在t1之中就会报错。

1. 计划

4.1、软件说明

|  |  |  |
| --- | --- | --- |
| 测试功能 | 输入 | 输出 |
| 登录注册身份验证 | 用户名、密码、昵称 | 进入用户界面或管理员界面 |
| 旅游景点管理模块 | 选择页面跳转 | 进入旅游信息界面 |
| 旅游留言管理模块 | 选择页面跳转 | 进入旅游留言界面 |
| 旅游优惠管理模块 | 选择跳转页面 | 进入旅游优惠界面 |
| 旅游游记管理模块 | 选择页面跳转 | 进入旅游游记管理界面 |
| 管理员管理模块 | 选择页面跳转 | 进入管理员界面 |
| 退出系统模块 | 选择页面跳转 | 退出系统 |

4.2、测试内容

测试1：名称：系统身份验证测试

目的：测试系统登录界面

内容：用户名、密码、昵称输入、合理性检查，系统操作界面显示控制。

测试2：名称:浏览测试

目的：测试页面跳转页面

内容：页面跳转，内容更新

测试3：名称：旅游景点管理测试

内容：景点管理界面显示图片，景点浏览，选择景点记录

测试4：名称：修改密码测试

目的：测试修改密码功能

内容：用户名、密码、新密码提交，合理性检查、更改密码

测试5：名称：注册管理员测试

目的；测试注册管理员功能

内容：用户名、密码提交、合理性检查。增加新的管理员

4.3、分以下四步进行：单元测试、集成测试、系统测试、验收测试；

单元测试

单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。因为单元测试需要知道内部程序设计和编码的细节知识，一般应由程序员而非测试员来完成，往往需要开发测试驱动模块和桩模块来辅助完成单元测试。因此应用系统有一个设计很好的体系结构就显得尤为重要。

一个软件单元的正确性是相对于该单元的规约而言的。因此，单元测试以被测试单位的规约为基准。单元测试的主要方法有控制流测试、数据流测试、排错测试、分域测试等等。

集成测试

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。它根据集成测试计划，一边将模块或其他软件单位组合成越来越大的系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。集成测试的策略主要有自顶向下和自底向上两种。

系统测试

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。因此，系统测试应该按照测试计划进行，其输入、输出和其他动态运行行为应该与软件规约进行对比。软件系统测试方法很多，主要有功能测试、性能测试、随机测试等等。

验收测试

验收测试旨在向软件的购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集。所不同的是，验收测试常常有软件系统的购买者代表在现场，甚至是在软件安装使用的现场。这是软件在投入使用之前的最后测试。

5、测试数据的整理方法

白盒测试

白盒测试也称结构测试或逻辑驱动测试，是指基于一个应用代码的内部逻辑知识，即基于覆盖全部代码、分支、路径、条件的测试，它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

“白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。“白盒”法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是天文数字。但即使每条路径都测试了仍然可能有错误。第一，穷举路径测试决不能查出程序违反了设计规范，即程序本身是个错误的程序。第二，穷举路径测试不可能查出程序中因遗漏路径而出错。第三，穷举路径测试可能发现不了一些与数据相关的错误。

白盒测试可以借助一些工具来完成如 Junit Framework ， Jtest 等。

黑盒测试

黑盒测试是指不基于内部设计和代码的任何知识，而基于需求和功能性的测试，黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用，在测试时，把程序看作一个不能打开的黑盆子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。黑盒测试方法主要有等价类划分、边值分析、因 — 果图、错误推测等，主要用于软件确认测试。

“黑盒”法着眼于程序外部结构、不考虑内部逻辑结构、针对软件界面和软件功能进行测试。“黑盒”法是穷举输入测试，只有把所有可能的输入都作为测试情况使用，才能以这种方法查出程序中所有的错误。实际上测试情况有无穷多个，人们不仅要测试所有合法的输入，而且还要对那些不合法但是可能的输入进行测试。

黑盒测试也可以借助一些工具，如 WinRunner ， QuickTestPro ， Rational Robot 等。

各功能模块测试结果及图面效果：

系统首页面：





景点信息介绍界面：





登录界面：



游记信息界面：



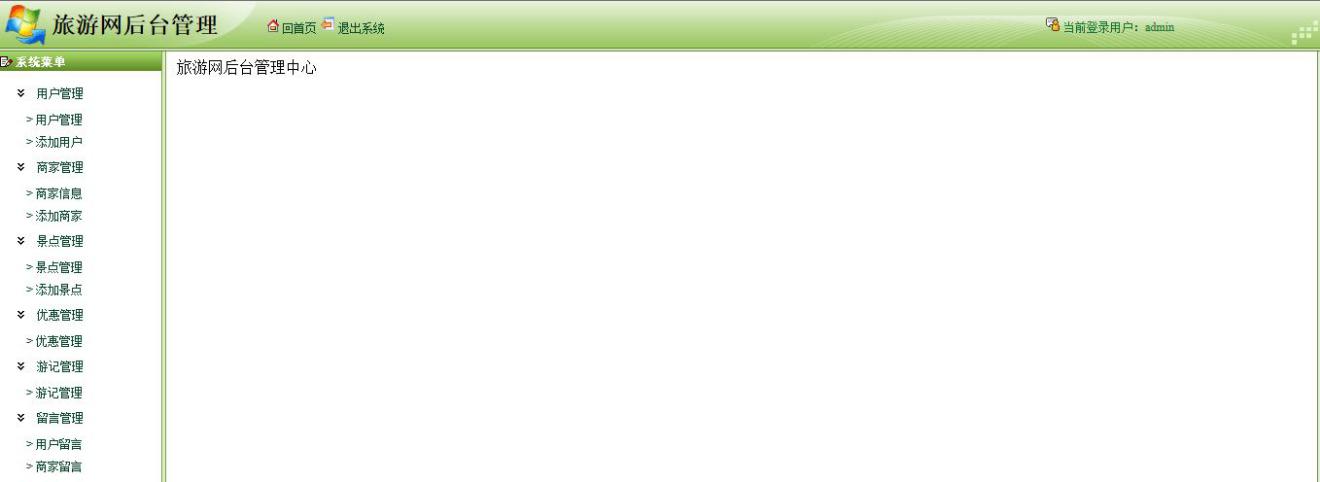


优惠信息界面：





后台首页：



后台登录界面：

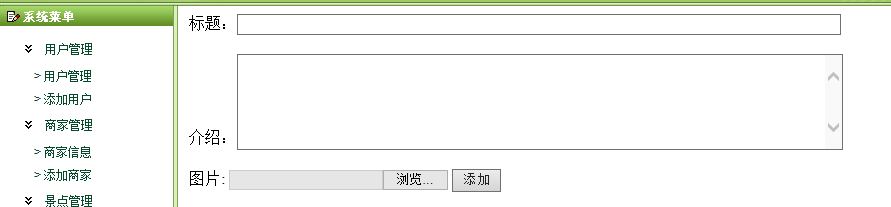


后台功能模块界面：

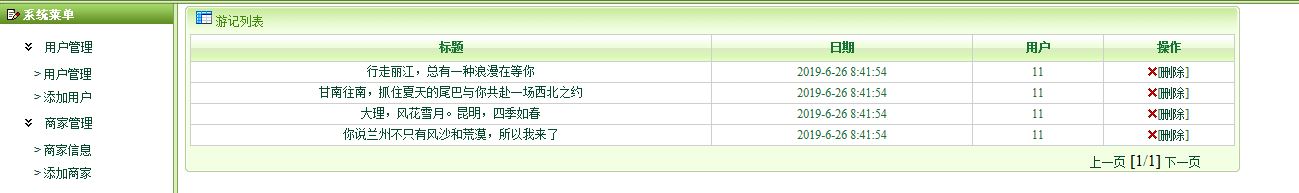


后台景点列表界面：

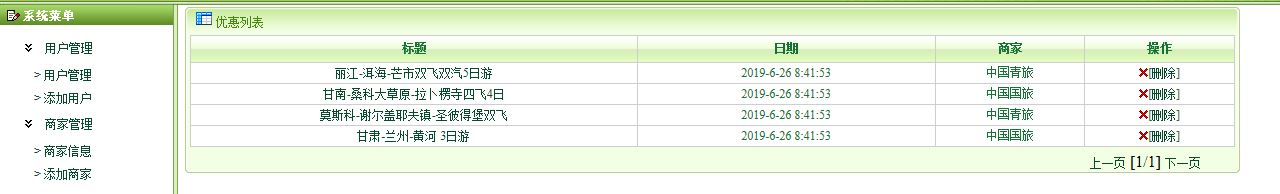




后台游记列表界面：



后台优惠列表界面：



后台用户列表界面：



以上就是各个各功能模块的运行截图。