

华中科技大学

软件工程大型作业

题目: 机票预订管理系统软件设计

专业班级: 物联网 1601
学 号: U201614897
姓 名: 潘越
指导教师: 朱建新
报告日期: 2018 年 11 月 12 日

计算机科学与技术学院

目录

I 摘要	1
II ABSTRACT	1
III 前言	2
III.1 课题的来源、目的、意义	2
III.2 主要解决的基本问题	2
III.3 更高的设计目标	2
III.3.1 软件后台要求	3
III.3.2 实用性要求	3
III.3.3 安全性要求	3
IV 可行性分析	4
IV.1 问题定义	4
IV.2 系统架构描述	4
IV.3 系统可行性	5
IV.4 开发计划	6
V 需求规格说明	7
V.1 引言	7
V.1.1 文档目的	7
V.1.2 范围限定	7
V.1.3 术语	7
V.1.4 综述	8
V.2 总体描述	8
V.2.1 功能需求	8
V.2.2 数据流说明	9
V.2.3 系统的数据流图	10
V.3 E-R 图设计	11
V.3.1 E-R 图综述	11
V.3.2 E-R 图模型	11
V.3.3 系统 E-R 图	11
V.4 详细功能需求	11
V.4.1 功能模块综述	11
V.4.2 功能模块流程图	12

VI 系统详细设计	16
VI.1 数据库设计	16
VI.1.1 数据库说明	16
VI.1.2 数据库详细设计	16
VI.2 系统业务流程设计	18
VI.2.1 业务流程图概述	18
VI.2.2 业务流程图设计	18
VI.3 关键技术与算法设计	19
VI.3.1 总述	19
VI.3.2 算法设计	19
VI.4 数据库存储事务设计	24
VI.4.1 数据库事务说明	24
VI.4.2 数据库事务设计	24
VI.5 安全性设计	27
VI.5.1 网站安全性	27
VI.5.2 安全设计	27
VII 界面设计	28
VII.1 界面设计概述	28
VII.2 软件界面	28
VIII 系统测试	37
VIII.1 测试目的	37
VIII.2 测试方法	37
VIII.2.1 单元测试	37
VIII.2.2 集成测试	37
VIII.3 确认测试	37
VIII.4 测试举例	39
VIII.4.1 登录和注册的数据验证测试	39
VIII.4.2 用户功能测试	40
VIII.4.3 管理员功能测试	44
IX 项目管理	47
IX.1 软件规模度量	47
IX.1.1 计算未调整功能点数	47
IX.1.2 计算技术复杂因子	47

目 录

IX.2 人员安排	48
IX.3 项目进度计划	49
附录 A 参考文献.....	51
附录 B 心得与体会.....	52

I 摘要

随着现代化城市的高速发展，管理系统的需求量越来越大，与此同时票务预订管理系统的分析与设计的问题也越来越突出。随着交通产业的发展，人们的出行也越来越依赖于各式各样的交通工具。不同的交通工具，不同的厂商，提供的票据也各不相同。正是因为这种复杂性，面对新的设计挑战，设计者必须根据不同种类的交通工具的特点，为其设计专用的票务管理系统。因此，本文选择了目前比较热门的机票预订管理系统，提供了一种基于 B/S 架构的设计方案以及具体实现，为未来的票务管理系统的设计提供了一种新的思路。

关键词：票务预订，管理系统，B/S 架构

II Abstract

With the explosive development of modern city, the demand for management system is increasing rapidly, meanwhile, the problem shown on the analysis and designation of ticket reservation management system is more and more prominent. As the transportation industry growing more and more stronger, people travel more and more dependent on a variety of vehicles. Different vehicles, different manufacturers, the tickets for them are also different. As a result of this complexity, designers must design a dedicated ticket management system based on the characteristics of different types of vehicles. Therefore, this article chooses the current popular flight ticket reservation management system, offers designation and implementation of a Browser-Server architecture based management system, which provides a new way of thinking for future ticket reservation management systems' designation.

keywords: ticket reservation, management system, B/S architecture

III 前言

III.1 课题的来源、目的、意义

数据库是数据管理的有效技术，是计算机科学的重要分支。今天，信息资源已成为各个部门的重要财富和资源，建立一个满足各级部门的信息处理要求的行之有效信息系统也是一个企业或组织生存和发展的重要条件。因此，作为信息处理系统核心和基础的数据库技术得到越来越广泛的应用。从小型单项事务处理系统到大型信息系统，从联机事务处理到联机分析处理，从一般企业管理到计算机辅助设计与制造，计算机集成制造系统、电子政务、电子商务、地理信息系统等，越来越多的应用领域采用数据库技术来存储和处理信息资源。特别是随着互联网的发展，广大用户可以之间访问并使用数据库，例如通过网上订购图书，日用品，机票，火车票。通过网上银行存款取款、检索和管理账户，等等，数据库已经成为每个人生活中不可缺少的部分。

因此，本次软件工程大型作业选题为常见的网上售票系统，以机票处理为载体，采用浏览器-服务器架构，以搭建网站框架的形式，来完成作业。

III.2 主要解决的基本问题

软件系统有两类用户来使用，对于两类用户，系统应具有以下的基本功能：

对于普通用户：

- 用户可以进行注册和登录
- 用户可以查询航班信息
- 用户可以查看自己所购买的票和退票
- 实时提醒用户持有票数和今天必须取的票数
- 用户可以在飞机起飞前一天凭取票通知取票
- 用户可以看到每个航班的预订情况，看到满座率
- 用户可以查看当前所有的航班票价
- 用户可以打印自己的账单

对于系统管理员：

- 管理员可以管理用户，修改用户信息或者删除用户
- 管理员可以查询航班信息
- 管理员可以录入航班信息以及航班的座位信息
- 管理员可以查询每个用户的订单信息，删除订单

III.3 更高的设计目标

在完成软件功能的基础上，我们应该对以下几项提出追加的要求：

III.3.1 软件后台要求

- 后台应该有一个记账表，记录所有用户执行的和金钱有关的操作信息
- 订单信息和通知信息分开存储
- 应该对用户和管理员的权限做区分

III.3.2 实用性要求

- 有一个方便用户和管理员使用的美观的界面
- 增加用户个人空间供用户管理个人信息
- 航班有两种类型的座位，供用户选择购票
- 购买操作的按钮只有在登录后才显示

III.3.3 安全性要求

- 对所有页面严格的权限控制，不满足权限时将会自动跳转到主页
- 数据库内的密码应做到加密，保证安全性
- 对表单的 POST 做 SQL 防注入处理，保证安全性

IV 可行性分析

IV.1 问题定义

目前常见的管理系统一般是采用 C/S 架构或者 B/S 架构，同时后台搭载数据库的形式实现，这个模式已经是一个成熟的体系，本次大型作业可以采用 B/S 架构，利用 Apache 服务器，以网站的形式实现管理系统，编程语言可以采用 PHP+HTML/CSS/JavaScript 来实现，数据库选用最常用的 MySQL，可以满足软件的设计目标，这部分对系统预算的贡献，以一台普通的阿里云 ECS 入门级 I/O 优化型服务器为例，一年的预算为近 500 元。若选用更高级别性能的服务器或者是付费的高性能数据库软件，则可能需要更高的预算。

同时，软件还要兼顾用户的使用体验和感受，因此应该合理使用 JavaScript 做出一些便于用户使用的功能和效果。同时利用 Photoshop 和 CSS 做出一些对网站的装饰也是很有必要的。

对于并发的安全性，MySQL 的存储事务可以安全地实现数据的存储保证不出错。而其他方面的安全性则可以由开发人员在程序编写时进行较全面的综合考虑。

目前商业中已获得成功的 12306 和携程网就是 B/S 架构的售票系统的典型范例，在开发之前可以参考一下其网站的布局和模式，吸收长处。

综合考虑到软件的开发周期，后台搭建和后期推广，以最低级别的保守设计目标为例，计算出实际的预算大致为 1000 元左右，代码总量在 6000 行左右，开发。

开发的具体问题、规模和初步目标见表 IV-1-1。

表 IV-1-1: 关于机票预订管理系统软件的规模和目标

项目名称	细节阐述
问题	实现一个叫预订管理系统
项目目标	开发一个稳定性强，安全性好且用户友好的系统
项目规模	开发总成本在 1000 元以内，代码在 6000 行左右
初步设想	使用 PHP+HTML/CSS/JavaScript 开发，Apache 作为服务器
开发周期	开发预计两周时间，先完成项目基本开发再配置环境

IV.2 系统架构描述

依据系统的功能和开发语言的选择，设计系统的总体架构，用图来表示，即为图 IV-2-1。

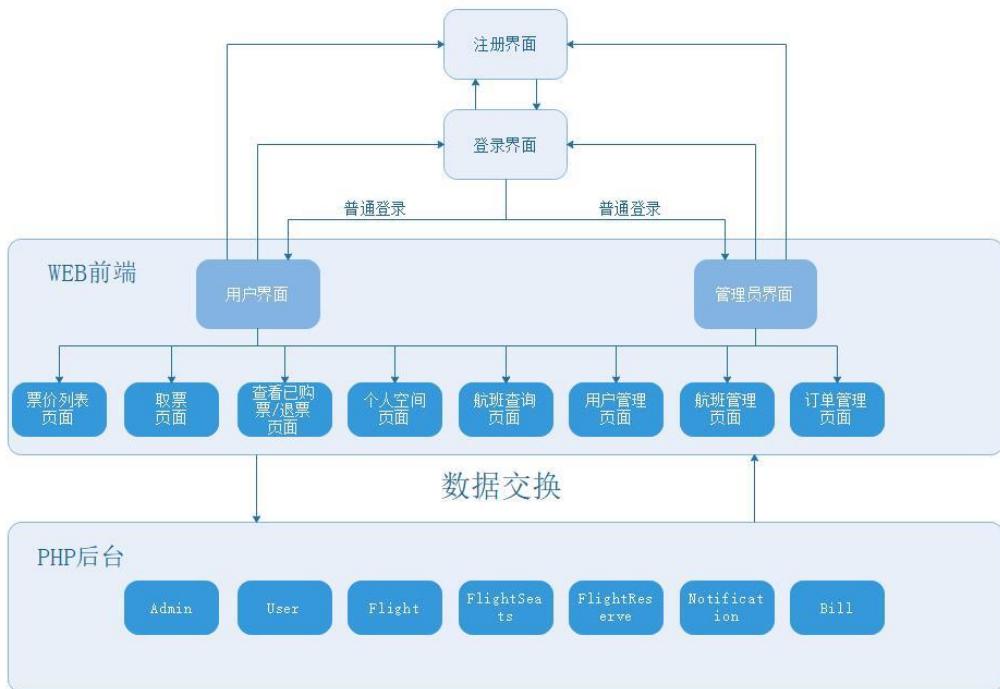


图 IV-2-1: 系统架构图

系统前端由 HTML/CSS + JS 实现，后台由 PHP + MySQL 实现。各部分的跳转指向如上图，同时对每个页面，应该区分好用户和管理员的权限，当不满足相应的权限时应该跳转离开页面。

用户在前端操纵网页上的按钮，同时按钮绑定 php 文件，后台执行对应的 php 文件通过 mysqli 系列函数对数据库执行操作，更新数据。

后台在一台 Linux 服务器上有 Apache 启动，然后可以通过浏览器（例如 Chrome）根据 IP 或者域名访问前端，同时设置好默认主页。

IV.3 系统可行性

现在我们对于系统需要处理的问题，系统的架构设计以及现有的商用系统都有了比较深入的了解，下面从多个方面进行可行性分析：

- **技术可行性：**到目前为止，对于开发此机票预订管理系统所需的知识所在的各个领域的研究已经比较成熟，无论是架构理论，存储算法，网页设计还是编程语言，软件所运行的平台，Linux 服务器技术等，都有较为成熟的理论成果，因此本机票预订管理系统软件在技术上是完全可行的。
- **操作可行性：**系统分了两种权限的用户，即普通用户和管理员用户，两种用户登录的方式不同，可以操纵的数据也不同，同时严格的权限控制保证了该方案的实施，因此这样的操作的可行的。

- 经济上，基于章节 II.1 对于项目可行性以及项目规模的初步研究，以满足基本的功能要求来说，项目开发、后台搭建以及后期推广的成本不会超过 1000 元，因此在经济上本系统的可行的。

IV.4 开发计划

根据选择的方案，我们制定了如表 II-4-2 所示的开发计划：

表 II-4-2: 实现开发方案的粗略开发计划

项目名称	所需时间/天
可行性研究	0.5
需求分析	1.0
概要设计	1.0
详细设计	2.5
实现	8.0
部署与测试	1.0
总计	14

这一开发并非最终的开发计划，但总体开发进度不应与此开发计划有较大的差别。

V 需求规格说明

V.1 引言

V.1.1 文档目的

本文档的目的是通过以面向对象的方法，对于将要开发的机票预订管理系统软件提供一个具有完整性和一致性的详细说明。本说明将以文字的方式来解释各个概念，使用图标的方式来阐释复杂的软件组成与交互，并使用表格来提供相关信息。

本文档的受众是所有参与机票预订管理系统的开发有关的项目人员和持股人，这包括但不限于软件开发者、项目管理者、质量检测人和消费者。

V.1.2 范围限定

本机票预订管理系统软件（下简称管理系统）需要确保用户对机票的购票、取票，管理员对航班信息的增删改查进行安全、高效的管理。管理系统将运行于 Linux 服务器上，其主要职责是根据用户在前端进行的操作，将结果从数据库中读取或写入数据库，而前端运行在各大浏览器中，由用户来操作。

处理恶意的高频用户甚至是 DDoS 攻击等不是控制管理系统的职责，管理系统将对正常情况下的低密度和高密度用户输入做出对应的处理，但对于恶意的、查过预订范围的攻击性的输入，本管理系统将采用用于处理正常情况下的输入的算法来处理。

本规格说明仅对管理系统做出要求，而并不包括对于服务器、浏览器（运行环境）部分做出要求，它们之所以被提到是因为管理系统需要基于它们运行，并且它们与管理系统的一些组成部分由交互与合作的成分，但本规格需求说明并不包含对于控制硬件与电梯的其他组成部分的要求。

V.1.3 术语

前端：前端是指 Web 浏览器上显示的页面，它由 HTML/CSS/JavaScript 实现，在开发中由于 PHP 语言对 HTML 的兼容性，所有的文件均采用.php 格式开发，因此在开发过程中的调试时必须在本地启动 Web 服务器用以运行 php 程序。

后台：后台是指用 php 实现的处理用户数据的程序，它将接收用户在表单中提交的各种数据，进行加工和处理，并对数据库进行对应的操作。

数据库：数据库指 MySQL 数据库，在数据库中创建表指使用 SQL 语句在 MySQL 数据库中创建表格，以下所有“对数据库进行操作”指使用

SQL 语句执行相应操作。

用户请求：指用户以及管理员对于一切航班数据的包括但不限于增删改查等操作，在本软件中统一以 php 文件调用 MySQL 的 API 来实现。

V.1.4 综述

本需求规格说明剩下的将以以下形式组织其内容：

章节 V.2 将提供有关本管理系统软件的粗略信息，并为其后的章节做出铺垫。章节 V.3 将提供有关本管理系统各个信息实体的 E-R 图设计，以作为后续数据库和功能设计的指导。章节 V.4 将描述系统的详细功能需求，并辅以流程图的形式，为后续算法的设计奠定基础。

V.2 总体描述

V.2.1 功能需求

管理系统需要完成的各组件的功能如下：

主页：输入网站域名后，直接显示的页面，应该设置有未登录，普通用户登录和管理员登录三种状态，对应三种状态显示不同的界面。主页中应该有所有功能的导航，并且显示今日的航班数据，但只有普通用户登录后才显示预定按钮。

新用户注册：普通用户可以在登录页面选择注册，填写个人信息和验证码完成注册，若用户名已存在则不通过注册。注册成功后跳转到登录页面。而管理员不开放注册，由后台设置管理员。

用户登录：用户分为普通用户和管理员用户，登录页面可以选择普通登录和管理员登录，在登录页面登录后进入相应的页面，不同的页面权限也不同。

航班检索：无论是否登录，点击航班查询按钮可以根据自己的设置对航班进行筛选和检索，但是只有登录后可以在航班数据旁显示预定按钮。

用户订票和退票：普通用户在航班搜索页面，可以在每个可用航班旁点击预定并打开确认页面，选择座位类型并由系统安排座位。点

击查看已购票可以查看自己持有的票，点击退票可以直接退票。

用户提醒：在用户登录的信息旁实时显示用户持有的票数和今天必须取的票数。

用户取票：用户点击取票按钮可以取票，在取票窗口可以打印账单并取票。

票价一览：用户点击票价列表，可以查看到所有航班的数据和价格列表。

用户个人中心：用户可以点击个人中心，修改个人信息和修改密码。

管理员对用户的管理：管理员可以点击用户管理，对用户的个人信息进行修改，同时可以删除用户

管理员对航班的管理：管理员可以点击航班管理，查看所有的航班信息，同时录入新的航班，在录入航班时自动按照座位个数生成座位信息。

管理员对订单的管理：管理员可以点击订单管理，按照用户来查看该用户的所有订单信息，同时可以删除订单。

V.2.2 数据流说明

数据库中应该具有以下的几张表：

- 管理员用户信息表
- 普通用户信息表
- 航班信息表
- 航班座位信息表
- 旅客订票信息表
- 取票通知表
- 账单

其中，表 1、2 用来处理用户注册和登录表，3-6 用来存储航班和订票的核心

数据，表 7 作为数据库后台的 log，存储所有和金钱有关的操作记录。对于数据库的完整性约束在章节 V.1.2 数据库详细设计中进行描述。

V.2.3 系统的数据流图

整个系统的数据流图如下，共分为三层进行描述：

顶层：

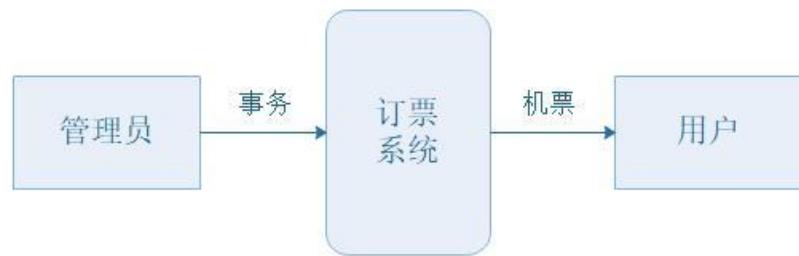


图 V-2-1：顶层数据流图

第二层：

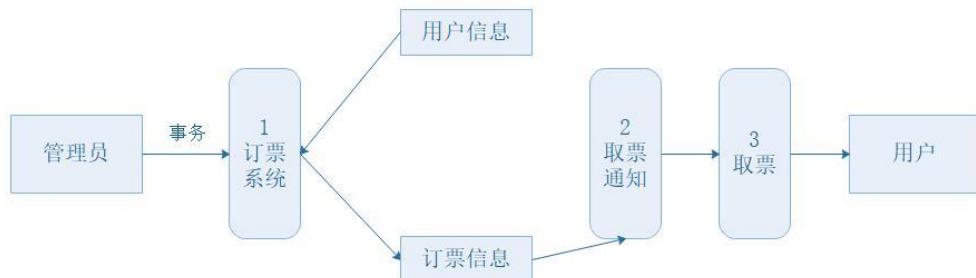


图 V-2-2：第二层数据流图

第三层：

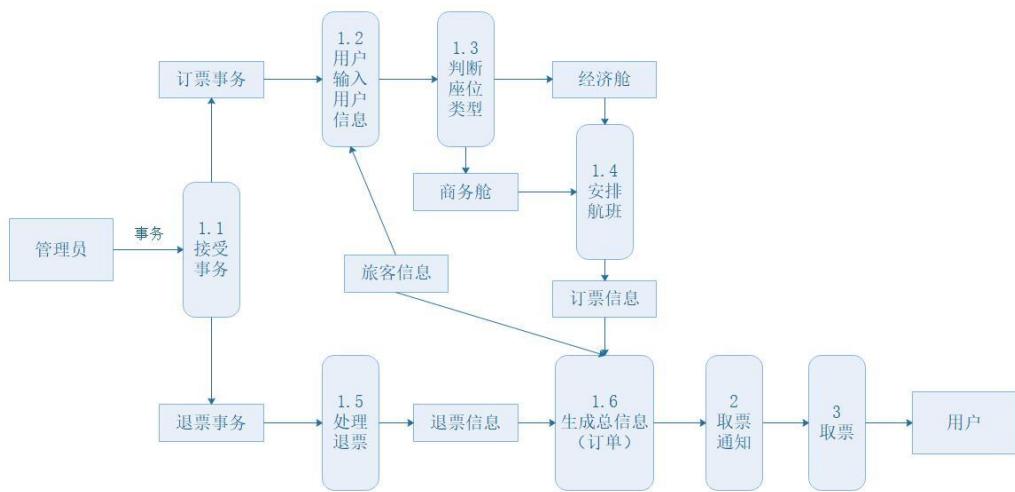


图 V-2-3: 第三层数据流图

V.3 E-R 图设计

V.3.1 E-R 图综述

E-R 图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。

它是描述现实世界关系概念模型的有效方法。是表示概念关系模型的一种方式。用“矩形框”表示实体型，矩形框内写明实体名称；用“椭圆图框”表示实体的属性，并用“实心线段”将其与相应关系的“实体型”连接起来。

用”菱形框“表示实体型之间的联系成因，在菱形框内写明联系名，并用”实心线段“分别与有关实体型连接起来，同时在”实心线段“旁标上联系的类型（1:1,1:n 或 m:n）。

本节给出管理系统各实体之间的 E-R 图模型。

V.3.2 E-R 图模型

管理系统各实体之间的 E-R 图模型如下：

管理员信息实体 ER 图：

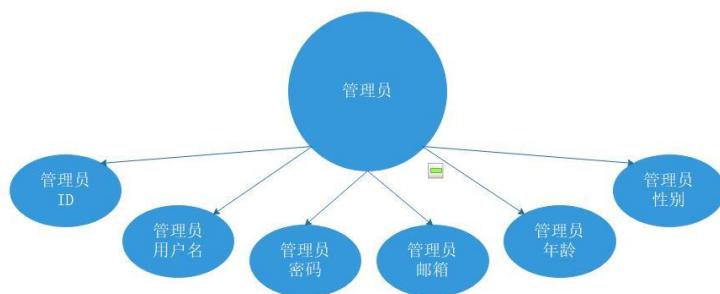


图 V-3-4: 管理员信息实体 E-R 图

普通用户信息实体 ER 图：

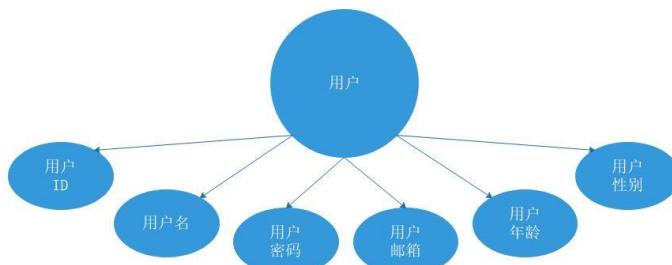


图 V-3-5: 普通用户信息实体 E-R 图

航班信息实体 ER 图：

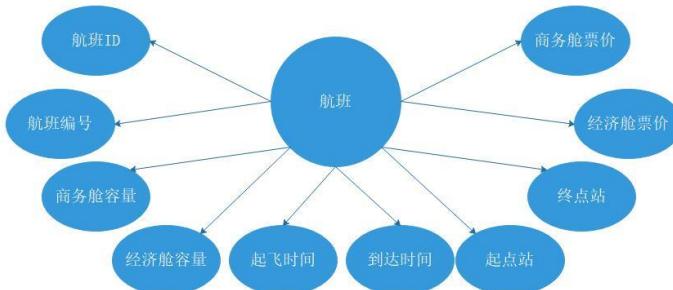


图 V-3-6: 航班信息实体 E-R 图

座位信息实体 ER 图：

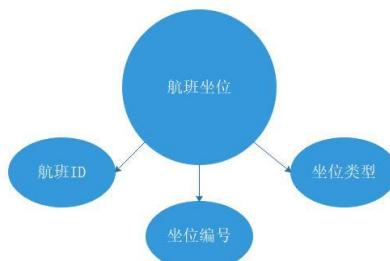


图 V-3-7: 座位信息实体 E-R 图

订票信息实体 ER 图：

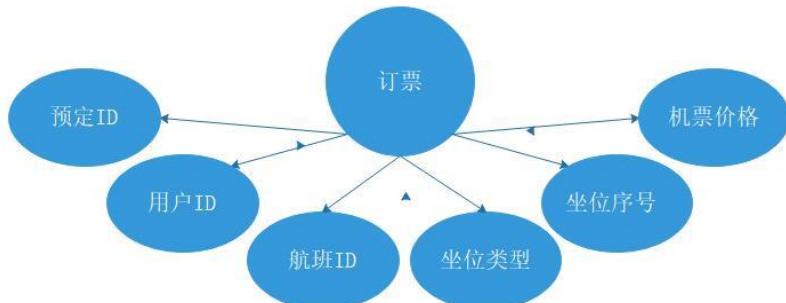


图 V-3-8: 座位信息实体 E-R 图

V.3.3 系统 E-R 图

将各个实体型的 E-R 图进行综合，可以得到以下的系统实体间 E-R 图，作为系统开发的指导概念模型。

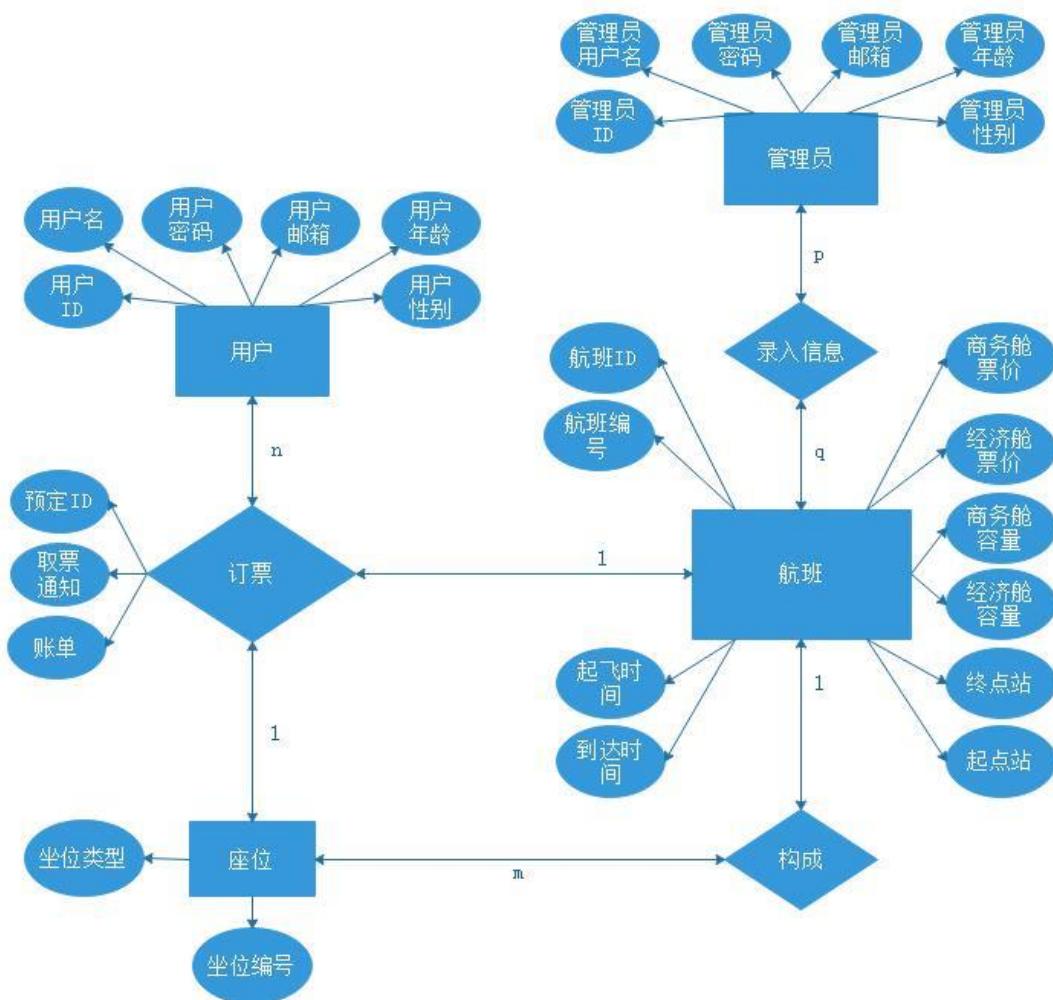


图 V-3-9: 系统实体间 E-R 图

V.4 详细功能需求

V.4.1 功能模块综述

系统需要完成的功能模块按照图 IV-2-1 所描述的系统架构，主要包括以下几个核心功能：

- 注册
- 登录
- 订票
- 退票
- 查询

系统实现的主要细节应集中与如何编程使得这部分的功能完善，性能优化，安全性有保障。

除以上五个功能以外的辅助功能，应在这五个功能完全实现并通过测试之后，再予以添加补充软件的全面性。

V.4.2 功能模块流程图

系统各功能模块的流程图如下所示：

用户注册的流程图：

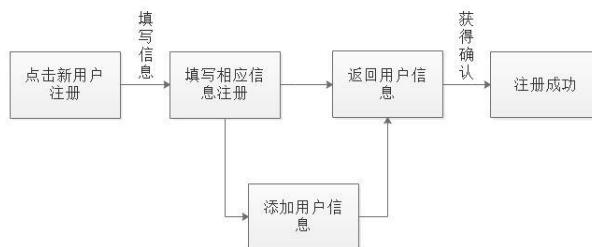


图 V-3-10: 用户注册的流程图

用户登录的流程图：

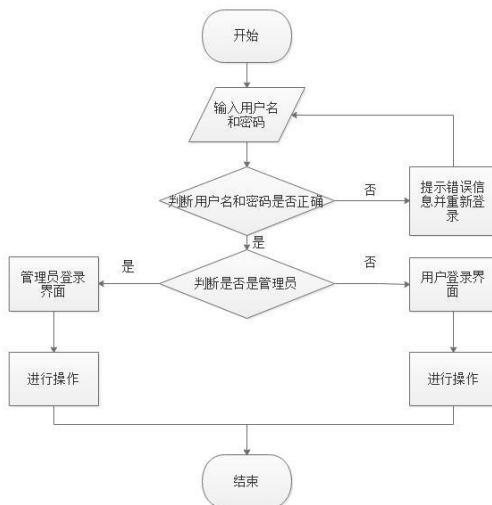


图 V-3-11: 用户登录的流程图

用户订票的流程图：

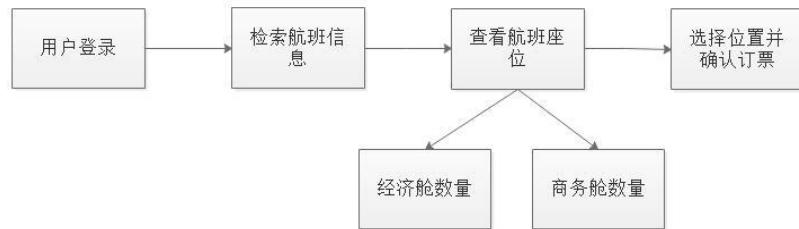


图 V-3-12: 用户订票的流程图

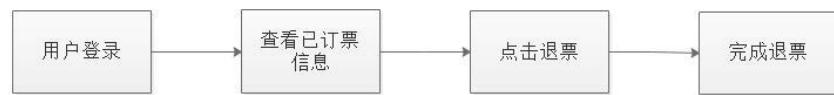
用户退票的流程图:

图 V-3-13: 用户退票的流程图

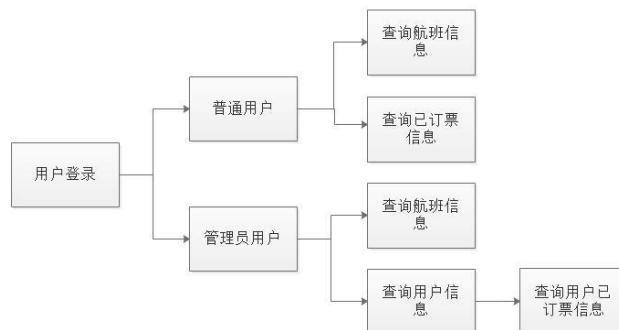
用户查询的流程图:

图 V-3-14: 用户查询的流程图

VI 系统详细设计

VI.1 数据库设计

VI.1.1 数据库说明

管理系统的数据库采用 MySQL Ver 15.1 Distrib 10.1.33-MariaDB, for Linux (x86_64), 在开发软件程序之前, 数据库中预先创建好我们设计的信息表。

VI.1.2 数据库详细设计

根据章节 V.2.2 数据流说明中的描述, 本文档给出管理系统使用的数据库表格的定义如下, 各表之间的关系在完整性约束和备注内说明。

表 VI-1-1: 管理员信息表 (Admin)

字段名	中文说明	字段类型	完整性约束	备注
Userid	用户 ID	NUMBER(10)	主码	非空唯一, 自动增长
Username	用户名	VARCHAR(20)		不重复
Password	密码	VARCHAR(32)		SHA 加密
Sex	性别	enum('男', '女', '不公开')		
Age	年龄	NUMBER(5)		整数
Email	邮箱	VARCHAR(20)		
Phone	手机号	VARCHAR(15)		
Jointime	设置时间	DATETIME		

表 VI-1-2: 普通用户信息表 (User)

字段名	中文说明	字段类型	完整性约束	备注
Userid	用户 ID	NUMBER(10)	主码	非空唯一, 自动增长
Username	用户名	VARCHAR(20)		不重复
Password	密码	VARCHAR(32)		md5 加密
Sex	性别	enum('男', '女', '不公开')		
Age	年龄	NUMBER(5)		整数
Email	邮箱	VARCHAR(20)		
Phone	手机号	VARCHAR(15)		
Jointime	注册时间	DATETIME		

表 VI-1-3: 航班信息表 (Flight)

字段名	中文说明	字段类型	完整性约束	备注
-----	------	------	-------	----

Flightid	航班 ID	NUMBER(10)	主码	非空唯一, 自动增长
Flightno	航班编号	VARCHAR(20)		
BClass	商务舱容量	NUMBER(5)		整数
NClass	经济舱容量	NUMBER(5)		整数
Leavetime	起飞时间	VARCHAR(20)		
Arrivetime	到达时间	VARCHAR(20)		
StartStation	起点站	VARCHAR(20)		
EndStation	终点站	VARCHAR(20)		
BPrice	商务舱票价	NUMBER(10)		
NPrice	经济舱票价	NUMBER(10)		

表 VI-1-4: 航班坐位信息表 (FlightSeats)

字段名	中文说明	字段类型	完整性约束	备注
Flightid	航班 ID	NUMBER(10)	外码	参考航班表
SeatType	座位类型	enum('商务舱', '经济舱')		
Seatid	座位序号	NUMBER(5)		
SeatUse	座位状态	enum('0', '1')		0:空 1:有人

表 VI-1-5: 旅客订票信息表 (FlightReserve)

字段名	中文说明	字段类型	完整性约束	备注
Reserveid	预定 ID	NUMBER(10)	主码	非空唯一
Userid	用户 ID	NUMBER(10)	外码	参考用户表
Flightid	航班 ID	NUMBER(10)	外码	参考航班表
SeatType	座位类型	enum('商务舱', '经济舱')		
Seatid	座位序号	NUMBER(5)		
Price	机票价格	NUMBER(10)		

表 VI-1-6: 取票通知表 (Notification)

字段名	中文说明	字段类型	完整性约束	备注
Userid	用户 ID	NUMBER(10)	外码	参考用户表
Flightid	航班 ID	NUMBER(10)	外码	参考航班表
SeatType	座位类型	enum('商务舱', '经济舱')		
Seatid	座位序号	NUMBER(5)		
Price	机票价格	NUMBER(10)		
Leavetime	起飞时间	VARCHAR(20)		

Arrivetime	到达时间	VARCHAR(20)		
------------	------	-------------	--	--

表 VI-2-7: 账单表 (Bill)

字段名	中文说明	字段类型	完整性约束	备注
Userid	用户 ID	NUMBER(10)	外码	参考用户表
Flightid	航班 ID	NUMBER(10)	外码	参考航班表
SeatType	座位类型	enum('商务舱', '经济舱')		
Seatid	座位序号	NUMBER(5)		
Price	机票价格	NUMBER(10)		
UserOption	操作	enum('0', '1')		0:订票, 1:退票

VI.2 系统业务流程设计

VI.2.1 业务流程图概述

业务流程图(TFD)是一种描述管理系统内各单位、人员之间的业务关系，作业顺序和管理信息流向的图表。它用一些规定的符号及连线表示某个具体业务的处理过程,帮助分析人员找出业务流程中的不合理流向。

本节根据用户种类的不同，分普通用户和管理员用户设计了业务流程图，便于指导软件的开放与设计，下面对业务流程图进行说明。

VI.2.2 业务流程图设计

根据用户权限类型，下分为普通用户的业务流程图和管理员的业务流程图：

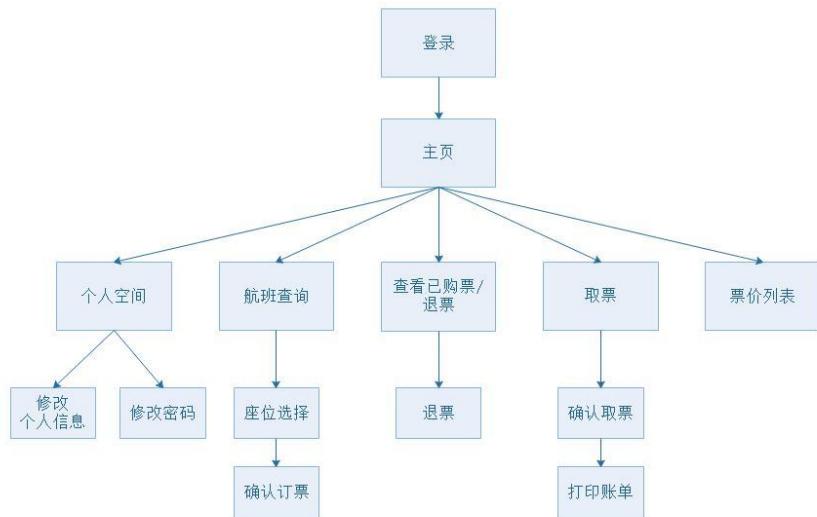


图 VI-2-1: 普通用户的业务流程图

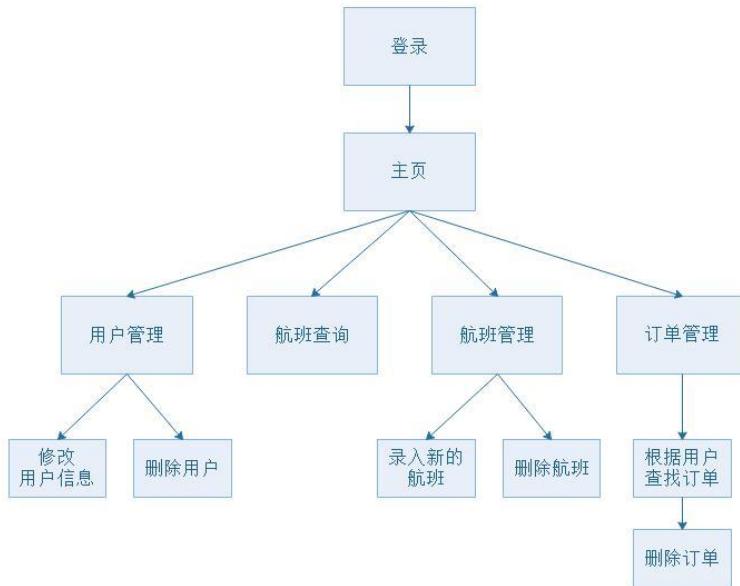


图 VI-2-2: 管理员的业务流程图

VI.3 关键技术与算法设计

VI.3.1 总述

本节针对软件开发过程中的一些核心功能的算法进行详细的说明。

算法作为程序的灵魂，在开发过程中是一个非常重要的组成部分，高效的算法可以使得程序得到极大的优化，下面本文档逐一给出说明，在必要的地方给出了 SQL 代码的代码段辅助以说明。

VI.3.2 算法设计

①登录的实现

算法设计：

- (1) 首先用户输入所有数据并提交。
- (2) 判断用户提交数据是否有空，若有空则退回 1。
- (3) 执行 SQL 查询，判断用户是否存在且密码是否输入正确，若不正确则退回 1。

```

SELECT Userid, Username
FROM User
WHERE Username = '$user_username' AND Password = SHA('$user_password');
  
```

- (4) 在超级全局变量\$_SESSION 中设置用户登录

```

$_SESSION['user_id'] = $row['Userid'];
$_SESSION['user_name'] = $row['Username'];

```

(5) 浏览器地址跳转到主页。

②注册的实现

算法设计：

- (1) 首先用户输入所有数据并提交。
- (2) 判断用户提交数据是否有空，若有空则退回 1。
- (3) 判断用户两次输入的密码是否一致，若不一致则退回 1。
- (4) 判断用户验证码是否输入正确，若不正确则退回 1。
- (5) 执行 SQL 查询，判断该用户名是否存在，若存在则退回 1。

```
SELECT * FROM User WHERE Username = '$regist_username';
```

(6) 执行 SQL 查询，在用户表里插入新用户。

```

INSERT INTO
User (Username, Password, Sex, Age, Email, Phone, Jointime)
VALUES ('$regist_username', SHA('$regist_password'), '$regist_sex', '$regist_age',
'$regist_email', '$regist_phone', NOW());

```

③管理员登录

算法设计：

管理员登录和用户登录几乎没有区别，算法的流程基本一致，不过查询是在 Admin 表中查询，而且登录成功后设置的超级全局变量不同。

④页面权限控制的实现

算法设计：

在每个页面加载之后，判断超级全局变量是否被设置，如果\$_SESSION['user_id']被设置，则当前登录的是普通用户，主页显示普通用户界面，如果\$_SESSION['admin_id']被设置，则当前登录的是管理员，主页显示管理员界面，如果两个都没有被设置，则无用户登录，主页显示游客界面。

⑤用户个人信息显示实现

算法设计：

(1) 根据用户 id 执行 SQL 查询，获得用户信息。

```
SELECT Sex, Age FROM User WHERE Userid = '$userid';
```

(2) 根据用户 id 执行 SQL 查询，查找用户持有订单。

```
SELECT * FROM FlightReserve WHERE Userid = '$userid';
```

(3) 查询取票通知表中明天出发的航班，并进行计数。

```
SELECT COUNT(*) AS Num
FROM Notification, Flight
WHERE Notification.Flightid = Flight.Flightid AND DATE(Leavetime) = DATE(NOW()) + 1;
```

⑥主页航班展示实现

算法设计：

(1) 查询今天出发的航班，并且限制数量 15 个以内，按照时间降序排列。

```
SELECT * FROM Flight
WHERE DATE(Leavetime) = DATE(NOW()) order by Leavetime ASC LIMIT 15;
```

(2) 对每个航班，在座位表里查询座位状态为空的数量并显示。

```
SELECT SeatType, count(*) as Num
FROM FlightSeats WHERE Flightid = '$flightid' AND SeatUse = '0' AND
SeatType = '商务舱' group by SeatType";
SELECT SeatType, count(*) as Num
FROM FlightSeats WHERE Flightid = '$flightid' AND SeatUse = '0' AND
SeatType = '经济舱' group by SeatType;
```

⑦航班查询实现

算法设计：

(1) 用户输入航班信息

(2) 判断用户提交数据是否有空，若有空则退回 1。

(3) 执行 SQL 查询，获取航班信息

```
SELECT * FROM Flight
WHERE StartStation = '$leave_station' AND EndStation = '$arrive_station'
AND DATE(Leavetime) = '$leave_day' AND DATE(Arrivetime) = '$arrive_day';
```

- (4) 对每个航班，在座位表里查询座位状态为空的数量并显示（SQL 语句同⑥）

⑧订票的实现

算法设计：

- (1) 用户点击订票，弹出订票确认窗口。
- (2) 用户选择座位类型，根据座位类型的不同显示价格。
- (3) 更新座位表，将该用户的位置状态置为 1（有人）。
- (4) 在订单表，通知表，账单表中插入相应的信息。

其中，3 和 4 可以使用存储事务实现，保证了数据库内容的正确性。

```
boot_insert('$userid', '$boot_flight_id', '$boot_type', '$seatid', '$price')
```

⑨退票的实现

算法设计：

- (1) 根据订单 id 查询订单信息。

```
SELECT * FROM FlightReserve WHERE Reserveid = '$reserveid';
```

- (2) 更新座位表，将该用户的位置状态置为 1（无人）。
- (3) 在订单表和通知表中删除相应的信息。
- (4) 在账单表中插入相应的信息。

```
boot_delete('$reserveid', '$userid', '$flightid', '$seattype', '$seatid', '$price')
```

⑩取票的实现

算法设计：

- (1) 用户点击取票，弹出取票确认按钮。
- (2) 用户点击打印，打印账单。（利用 JavaScript 的 window.print() 实现）
- (3) 在订单表和通知表中删除相应信息。（SQL 语句和⑨相同）

⑪管理员用户管理的实现

算法设计：

- (1) 在用户表中查询所有的用户。

```
SELECT * FROM User;
```

- (2) 对每个用户，在订单表中查询其拥有的订单个数并显示。

```
SELECT * FROM FlightReserve WHERE Userid = '$userid';
```

⑫管理员航班管理的实现

算法设计：

展示航班：

- (1) 在航班表中查询所有航班的信息，按起飞时间降序排列。

```
SELECT * FROM Flight ORDER BY Leavetime ASC;
```

- (2) 对每个航班，在座位表里查询座位状态为空的数量并显示。（SQL 语句和⑥一致）

- (3) 在每个航班旁生成一个“删除”按钮，供管理员删除航班。

插入航班：

- (1) 管理员输入航班信息。
- (2) 判断输入航班信息是否为空，若为空则退回 1.
- (3) 对航班表执行 SQL 插入。

```
INSERT INTO
Flight (Flightno, BClass, NClass, Leavetime, Arrivetime, StartStation, EndStation, BPrice, NPrice)
VALUES ('$flightno', '$bclass', '$nclass', '$leave_time', '$arrive_time', '$start_station', '$end_station', '$bprice', '$nprice');
```

- (4) 根据 1 中插入的航班的航班次查询新生成的航班 id。

```
SELECT Flightid FROM Flight WHERE Flightno = '$flightno';
```

- (5) 执行存储事务函数，在座位表中生成座位信息。

```
CALL add_new_seats('$flightid', '$bclass', '$nclass');
```

删除航班：

- (1) 分别执行 SQL 查询，删除座位信息和航班信息即可。

⑬管理员订单管理的实现

算法设计：

- (1) 管理员输入用户名。
- (2) 判断输入用户名是否为空，若为空则退回 1。
- (3) 根据用户名，在用户表中查询用户 id。

```
SELECT Userid FROM User WHERE Username = '$username';
```

- (4) 根据用户 id，在订单表中查询该用户的订单。

```
SELECT * FROM FlightReserve, Flight
WHERE FlightReserve.Flightid = Flight.Flightid AND Userid = '$userid';
```

VI.4 数据库存储事务设计

VI.4.1 数据库事务说明

数据库事务(Database Transaction)，是一系列作为一个逻辑单元来执行的操作集合。它是数据库维护数据一致性的单位，它将数据库从一致状态转变为新的一致状态，指作为单个逻辑工作单元执行的一系列操作，要么完全地执行，要么完全地不执行。事务处理可以确保除非事务性单元内的所有操作都成功完成，否则不会永久更新面向数据的资源。事务是数据库运行中的逻辑工作单位，由 DBMS 中的事务管理子系统负责事务的处理。

本文档给出了软件开发过程中需要使用到的几个数据库存储事务的定义说明，以指导开发者的程序设计

VI.4.2 数据库事务设计

在管理系统的开发中，共需要用到三个数据库存储事务。目的是保证一系列操作的原子性，防止在发生意外情况时导致的数据库数据错误。

①生成航班的座位信息

存储事务函数：add_new_seats(flightid, bclass, nclass)

输入参数：

flightid：航班 id，

bclass：商务舱容量，

nclass：经济舱容量。

作用：向座位表中插入该航班的所有座位信息，用以插入新的航班时初始化

座位。

具体实现：

```
delimiter //
CREATE PROCEDURE add_new_seats(IN fightid INT(10), IN bclass INT(5),
IN nclass INT(5))
BEGIN
DECLARE num INT;
SET num = 1;
WHILE num < bclass + 1 DO
INSERT INTO FlightSeats VALUES(fightid, '商务舱', num, '0');
SET num = num + 1;
END WHILE;
WHILE num < bclass + nclass + 1 DO
INSERT INTO FlightSeats VALUES(fightid, '经济舱', num, '0');
SET num = num + 1;
END WHILE;
END //
```

在使用时执行 CALL add_new_seats()即可。

②新增订单

存储事务函数：boot_insert(Userid, Flightid, SeatType, Seatid, Price)

输入参数：

- Userid：订票用户 id
- Flightid,：订票航班 id
- SeatType：订票座位类型
- Seatid：座位 id
- Price：订单价格

作用：向订单表，通知表和账单中插入本次订票的信息。

具体实现：

```
DELIMITER //
CREATE PROCEDURE boot_insert(IN Userid INT(10), IN Flightid INT(10),
IN SeatType ENUM('商务舱', '经济舱'), IN Seatid INT(5), IN Price INT(10))
BEGIN
UPDATE FlightSeats SET SeatUse = '1'
WHERE Flightid = Flightid and Seatid = Seatid;
INSERT INTO FlightReserve (Userid, Flightid, SeatType, Seatid, Price)
```

```

VALUES (Userid, Flightid, SeatType, Seatid, Price);
INSERT INTO Notification
VALUES (Userid, Flightid, SeatType, Seatid, Price);
INSERT INTO Bill
VALUES (Userid, Flightid, SeatType, Seatid, Price, '0');
END //

```

在使用时执行 CALL boot_insert()即可

③删除订单

存储事务函数: boot_delete(Reserveid, Userid, Flightid , SeatType, Seatid, Price)

输入参数:

- Reserveid: 订单 id
- Userid: 订票用户 id
- Flightid,: 订票航班 id
- SeatType: 订票座位类型
- Seatid: 座位 id
- Price: 订单价格

作用: 用户退票时, 在订单表和通知表中删除消息, 在账单中插入本次操作的内容。

具体实现:

```

DELIMITER //
CREATE PROCEDURE boot_delete(IN Reserveid INT(10),IN Userid INT(10),
IN Flightid INT(10), IN SeatType ENUM('商务舱', '经济舱'), IN Seatid INT(5),
IN Price INT(10))
BEGIN
UPDATE FlightSeats SET SeatUse = '0'
WHERE Flightid = Flightid and Seatid = Seatid;
DELETE FROM FlightReserve WHERE Reserveid = Reserveid;
DELETE FROM Notification WHERE Flightid = Flightid and Seatid = Seatid;
INSERT INTO Bill VALUES (Userid, Flightid, SeatType, Seatid, Price, '1');
END //

```

在使用时执行 CALL boot_delete()即可。

VI.5 安全性设计

VI.5.1 网站安全性

由于是网站框架，因此安全性对于系统来说也是一个十分重要的成分，没有做防护的系统很容易遭到各方面的网络渗透攻击等等，因此，本文档在安全性上给系统做出以下设计。

VI.5.2 安全设计

权限控制：由于 WEB 服务器的特性，所有的页面都可以被用户得到，但对于不同权限的用户，并不是可以访问到服务器上所有的页面，因此对所有的页面都进行了权限控制。

在超级全局变量\$_SESSION 中设置了 userid 和 adminid 两个值，当用户登录时，userid 被设置，当管理员登录时，adminid 被设置，当相应的用户或者是管理员退出时，清除被设置的值。所有的页面在加载时，检测相应的值是否存在，若不存在则直接跳转到主页，不允许访问。

密码加密：后台使用 SHA 加密存储密码，一旦数据库被入侵，入侵者也不可能获取真正的密码，从而保证了用户的安全性。

SQL 防注入：在所有的使用超级全局变量\$_POST 获取表单时，全部统一使用类似如下的模式：

```
mysqli_real_escape_string($conn, trim($_POST['username']))
```

而不是使用简单的赋值操作。该操作可以替换掉例如 “,” “=” “-” 等等敏感的可以利用来进行 SQL 注入的字符串，从而保证了系统的安全性。

VII 界面设计

VII.1 界面设计概述

由于本软件使用 B/S 架构来设计，因此界面设计实际上就是 Web 前端的设计，本软件计划采用先进行静态页面开发，再移植为 PHP 页面的方法进行开发。以下本文档给出软件各个界面的设计图。

VII.2 软件界面

主页（无用户登录）



图 VII-2-1: 主页（无用户登录）

主页（普通用户登录）



图 VII-2-2: 主页 (普通用户登录)

主页 (管理员登录)



图 VII-2-3: 主页 (管理员登录)

登录界面



图 VII-2-4: 登录界面

注册界面



图 VII-2-5: 注册界面

航班查询界面



图 VII-2-6: 航班查询界面

查看已购票/退票界面



图 VII-2-7: 查看已购票/退票界面

取票界面



图 VII-2-8: 取票界面

票价列表界面

航班次	商务舱 容量	经济舱 容量	起飞时间	到达时间	起点站	终点站	商务舱 价格	经济舱 价格
CZ6606	40	40	2018-06-20 06:40:00	2018-06-20 09:05:00	北京	武汉	570	330
CZ9210	40	40	2018-06-20 07:15:00	2018-06-20 08:45:00	武汉	上海	600	400
HO1212	40	40	2018-06-20 15:00:00	2018-06-20 17:20:00	西安	上海	618	410
CZ6172	40	40	2018-06-19 06:45:00	2018-06-19 08:50:00	上海	武汉	620	480
MU2514	40	40	2018-06-18 21:20:00	2018-06-18 23:20:00	上海	武汉	630	480
MU5010	50	50	2018-06-20 08:50:00	2018-06-20 11:25:00	西安	上海	650	440
MU2253	50	50	2018-06-20 07:10:00	2018-06-20 11:20:00	西安	上海	650	330
MU2456	50	50	2018-06-20 21:05:00	2018-06-20 23:25:00	北京	武汉	720	480
FM9328	50	50	2018-06-19 07:30:00	2018-06-19 09:25:00	郑州	上海	730	510
MU9364	50	50	2018-06-20 21:30:00	2018-06-20 23:05:00	武汉	上海	750	580
CZ6590	50	50	2018-06-19 20:00:00	2018-06-19 21:55:00	武汉	广州	780	550
CZ6197	50	50	2018-06-21 08:20:00	2018-06-21 10:00:00	武汉	上海	780	480
CZ3670	40	40	2018-06-18 21:20:00	2018-06-18 23:40:00	青岛	武汉	860	580
9C8930	50	50	2018-06-20 06:45:00	2018-06-20 09:10:00	广州	上海	880	590
MU2542	50	50	2018-06-19 07:30:00	2018-06-19 09:20:00	武汉	广州	920	680
MU2691	40	40	2018-06-21 15:05:00	2018-06-21 16:40:00	武汉	上海	940	680
MU3185	40	40	2018-06-18 19:00:00	2018-06-18 21:40:00	上海	广州	960	760
CZ3586	40	40	2018-06-20 22:05:00	2018-06-20 12:30:00	广州	上海	960	680
HU7810	80	80	2018-06-18 22:05:00	2018-06-19 01:15:00	广州	北京	1020	770
MF7714	40	40	2018-06-19 20:00:00	2018-06-19 21:40:00	南京	上海	1020	770

图 VII-2-9: 票价列表界面

用户个人中心界面



图 VII-2-10: 个人中心界面

修改密码界面



图 VII-2-11: 修改密码界面

订单确认界面



图 VII-2-12: 订单确认界面

账单界面



图 VII-2-13: 账单界面

打印账单界面（调用浏览器接口生成）

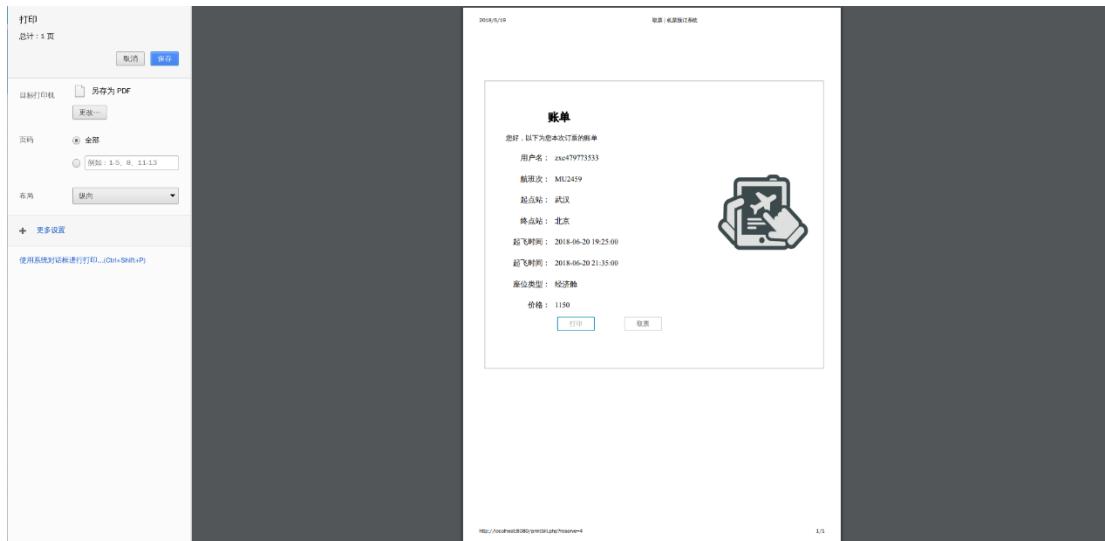


图 VII-2-14: 打印账单界面

用户管理界面（管理员）

用户名	性别	年龄	邮箱	电话	注册时间	订票数	修改	删除
zxc479773533	男	20	479773533@qq.com	13995718033	2018-06-15 16:21:26	3	<button>修改</button>	<button>删除</button>
test1	男	15	123@gmail.com	123456789	2018-06-19 19:20:47	0	<button>修改</button>	<button>删除</button>
test2	男	20	456@gmail.com	123456789	2018-06-19 19:21:16	0	<button>修改</button>	<button>删除</button>

图 VII-2-15: 用户管理界面

航班管理界面（管理员）



图 VII-2-16: 航班管理界面

订单管理界面（管理员）



图 VII-2-17: 订单管理界

VIII 系统测试

VIII.1 测试目的

这一测试的目的是为了找出应用程序中的缺陷，从而提高管理系统的质量。对于本管理系统不可能对于每一个细节均匀测试，但需要在尽可能广泛的范围内测试，其中，测试并对程序作出修改后至少需要保证程序可用，且具有较高的可移植性。

VIII.2 测试方法

由于本管理系统包含有多个模块，因此系统的测试应该先从小模块开始，在代码编写过程中对于代码中每一个模块进行集成测试。在开发完成之后，依据需求说明中的功能模块流程图和详细设计中的业务流程图，对于每一个模块进行集成测试，并采用自底向上的测试方法。最后，使用黑盒测试方法，让非开发人员以及用户进行确认测试。只有在这三种测试完成之后，才能够确定管理系统能够投放使用。对于测试用例的设计需要考虑边界值的选取以及等价类的划分。

VIII.2.1 单元测试

单元测试主要是测试代码是否存在语法错误、逻辑错误等。其主要目的是验证进行代码编写时是否能够正确执行而产出符合预期值的结果，在本控制软件的代码编写的过中对于管理系统实现中的所有类方法进行的单元测试应在软件编写的过程中进行。

VIII.2.2 集成测试

集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照章节 IV-2-1 设计的架构图组装成为子系统或系统，进行集成测试。

VIII.3 确认测试

对于确认测试而言，其主要目的是测试管理系统每一个功能的完整性。因此，根据表 VIII-3-1 所示的管理系统的功能对其进行测试。对于每一个功能，根据其功能描述在限制条件下进行测试，测试结果应与“功能描述/预期结果”中描述的结果相同。

表 VIII-3-1: 功能表

功能名称	功能描述/预期结果	细节与限制
用户登录	用户正常登录，进入用户主界面	当用户输入错误的用户名或密码时，给出错误提示
用户注册	提示用户注册成功，页	当出现以下类型错误时，

	面跳转至登录界面	给用户相应的提示： 1. 两次输入的密码不同 2. 注册的用户名已存在 3. 验证码输入错误 4. 输入信息不完全
用户查询航班	用户输入查询信息后，可以在界面上看到正确的查询结果	当用户输入的信息不完整时，系统应提示用户
用户订票	用户对查询的结果点击订票按钮，页面跳转到用户的个人票仓，可以看到用户正确预订了票	对于没有位置的航班，订票按钮应为不可点状态
用户退票	用户在个人票仓点击退票按钮，可以看到票仓中去掉了这张票	无
用户取票	用户点击取票按钮，可以看到账单，再确认之后，可以看到票仓中去掉了这张票	无
用户打印账单	用户在取票时，点击打印按钮，可以进入浏览器的打印账单页面，生成账单 pdf 文件	确保主流浏览器可以调用打印接口
用户修改个人信息	用户可以进入个人空间，修改自己的信息。	注意用户无法修改自己的用户名，其中，修改密码还要求同时输入原密码和新密码，当原密码出错时，应给出对应的提示
管理员管理用户	管理员点击用户管理，可以看到所有的用户信息，点击用户旁边的修改按钮，可以进入用户个人中心修改其信息。管理员还可以删除用户	注意仍然拥有订单的用户无法直接删除，需要先清理对应的订单信息

管理员录入航班	管理员点击航班管理，可以看到所有的航班信息，同时可以在上方的新航班录入栏里录入新的航班	无
管理员删除航班	管理员点击航班管理，可以看到所有的航班信息，同时可以点击航班信息旁边的删除按钮删除航班	注意只有空航班可以被删除，有用户预订的航班无法被直接删除
管理员管理订单	管理员进入订单管理界面，输入用户名，可以查看该用户持有的所有订单，点击删除按钮可以删除订单	无

VIII.4 测试举例

我们基于章节 VIII.3 给出的功能表，对每个功能进行了一定范围的测试，下面对每个功能以一张图的形式，举例给出程序功能的演示。

VIII.4.1 登录和注册的数据验证测试

①用户名或密码错误

当输入错误的用户名和密码时，看到如下的提示：



图 VIII-3-1: 用户名或密码错误

②注册时，两次输入的密码不同

在注册页面输入两个不同的密码，点击注册会看到如下的提示：



图 VIII-3-2: 两次输入的密码不同

③注册的用户名已存在

尝试注册一个用户名已存在的用户，会得到如下提示：

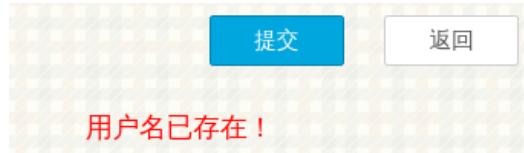


图 VIII-3-3: 注册的用户名已存在

④验证码输入错误

当注册输入了错误的验证码，会得到如下提示：



图 VIII-3-4: 验证码输入错误

⑤输入信息不完全

当漏输入某一项信息时，会得到如下提示：



图 VIII-3-5: 输入信息不完全

通过以上测试，验证系统在登录和注册时的数据验证是完备的。

VIII.4.2 用户功能测试

①用户查询功能

例如，查询武汉到上海的 6 月 20 日出发，6 月 20 日到达的航班，查询前后结果如下：

A screenshot of a flight search form. It has four input fields: '起点站' (Departure Station) with '武汉' (Wuhan) entered, '终点站' (Arrival Station) with '上海' (Shanghai) entered, '起飞日期' (Departure Date) with '2018/06/20' entered, and '到达日期' (Arrival Date) with '2018/06/20' entered. To the right of these fields is an orange '查询' (Search) button.

图 VIII-3-6: 用户查询

The screenshot shows a flight search form with fields for '起点站' (Start Station) and '终点站' (End Station), and date ranges '起飞日期' (Departure Date) and '到达日期' (Arrival Date). A '查询' (Search) button is also present. Below the form is a title '航班查询' (Flight Query). The main area displays a table of flight results:

航班次	商务舱	经济舱	起飞时间	到达时间	起点站	终点站	商务舱 价格	经济舱 价格	备注
CZ9210	39/40	40/40	2018-06-20 07:15:00	2018-06-20 08:45:00	武汉	上海	600	400	<button>预定</button>
MU2505	40/40	39/40	2018-06-20 16:40:00	2018-06-20 18:15:00	武汉	上海	1650	1230	<button>预定</button>
MU9364	50/50	50/50	2018-06-20 21:30:00	2018-06-20 23:05:00	武汉	上海	750	580	<button>预定</button>

图 VIII-3-7: 查询结果

②用户订票功能

接着上面，我们尝试将这三个航班的经济舱都预定一下，使用新注册的用户 test1 来执行，结果如下：

The screenshot shows a user profile summary on the left and a '我的票' (My Tickets) list on the right.

已登录用户

- 用户名 : test1 普通用户
- 性别 : 男
- 年龄 : 15
- 已预定票数 : 3
- 待取票数 : 3

我的票

航班次	起点站	终点站	起飞时间	到达时间	坐位号	座位类型	价格	操作
CZ9210	武汉	上海	2018-06-20 07:15:00	2018-06-20 08:45:00	41	经济舱	400	<button>退票</button>
MU2505	武汉	上海	2018-06-20 16:40:00	2018-06-20 18:15:00	42	经济舱	1230	<button>退票</button>
MU9364	武汉	上海	2018-06-20 21:30:00	2018-06-20 23:05:00	51	经济舱	580	<button>退票</button>

图 VIII-3-8: 用户订票

③用户退票功能

接着，我们尝试退掉 CZ9210 航班的票，点击退票，结果如下：

The screenshot shows a user profile summary on the left and a '我的票' (My Tickets) list on the right.

已登录用户

- 用户名 : test1 普通用户
- 性别 : 男
- 年龄 : 15
- 已预定票数 : 2
- 待取票数 : 2

我的票

航班次	起点站	终点站	起飞时间	到达时间	坐位号	座位类型	价格	操作
MU2505	武汉	上海	2018-06-20 16:40:00	2018-06-20 18:15:00	42	经济舱	1230	<button>退票</button>
MU9364	武汉	上海	2018-06-20 21:30:00	2018-06-20 23:05:00	51	经济舱	580	<button>退票</button>

图 VIII-3-9: 用户退票

④用户取票

接着，我们尝试取走 MU2505 的票，可以看到如下的账单：



图 VIII-3-10: 用户取票

⑤用户打印账单

接着点击打印按钮，生成账单 pdf 如下：



图 VIII-3-11: 打印订单

在取票之后，结果如下图：



图 VIII-3-12: 用户取票之后

⑥用户进入个人空间修改信息

修改前，用户个人信息如下图：

The screenshot shows the 'Personal Center' interface. At the top, it says '欢迎来到个人中心，您可以在这里修改自己的个人信息或者是密码'. Below that, there are input fields for '用户名' (test1), '性别' (男), '年龄' (15), '邮箱' (123@gmail.com), and '手机号' (123456789). To the right of the form is a blue silhouette of a person holding a pencil. At the bottom, it shows the '注册时间' (2018-06-19 19:20:47) and three buttons: '提交' (Submit), '修改密码' (Change Password), and '返回' (Back).

图 VIII-3-13: 用户修改信息前

尝试将性别改为女，年龄改为 20，手机改为 13995718033，点击提交，可以看到提交成功界面：

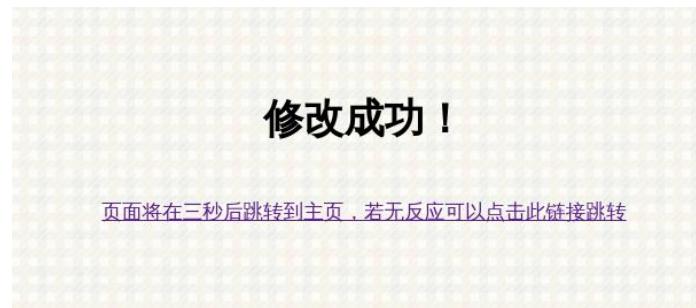


图 VIII-3-14: 用户修改信息成功

三秒后跳转到主页，可以看到用户的信息已经修改：

The screenshot shows the main homepage. At the top, a red banner says '您好，欢迎来到机票预订系统，您可以根据您的需求搜索相应航班并预订机票。 我们衷心地为您' (Hello, welcome to the flight booking system, you can search for flights according to your needs and book them. We sincerely serve you). On the left, a sidebar shows '已登录用户' (Logged-in User) with details: '用户名 : test1 普通用户' (Username: test1 Ordinary User), '性别 : 女' (Gender: Female), '年龄 : 20' (Age: 20), '已预定票数 : 1' (Number of booked tickets: 1), and '待取票数 : 1' (Number of tickets to be picked up: 1). It has '个人中心' (Personal Center) and '退出' (Logout) buttons. Below it are two large buttons: '主页' (Home) and '购票用户注册' (Buyer User Registration). On the right, there's a '联系作者' (Contact Author) section featuring an illustration of a girl named Yue Pan with her GitHub info: 'github: zxc479773533'. Below that is a '今日航班' (Today's Flights) table:

航班次	商务舱	经济舱	起飞时间	到达时间	起点站	终点站	商务舱 价格	经济舱 价格	备注
CA8205	40/40	40/40	2018-06-19 00:40:00	2018-06-19 14:40:00	武汉	北京	1520	1130	预定
CZ6172	39/40	40/40	2018-06-19 06:45:00	2018-06-19 08:50:00	上海	武汉	620	480	预定
MU2519	40/40	40/40	2018-06-19 07:15:00	2018-06-19 07:15:00	武汉	上海	1570	1120	预定

图 VIII-3-15: 用户修改信息后

综上，通过以上的测试，验证系统用户功能完善，达到了预期的目标。

VIII.4.3 管理员功能测试

①管理员管理用户

点击用户管理，可以看到如下的用户信息：

用户管理								
用户名	性别	年龄	邮箱	电话	注册时间	订票数	修改	删除
zxc479773533	男	20	479773533@qq.com	13995718033	2018-06-15 16:21:26	3	修改	删除
test1	女	20	123@gmail.com	13995718033	2018-06-19 19:20:47	1	修改	删除
test2	男	20	456@gmail.com	123456789	2018-06-19 19:21:16	0	修改	删除

图 VIII-3-16: 管理员修改信息前

接着我们尝试将 test1 的年龄邮箱改为 479773533@qq.com，并删除 test2 用户，分别点击修改进入 test1 的个人中心，点击删除删除 test2，得到。

用户管理								
用户名	性别	年龄	邮箱	电话	注册时间	订票数	修改	删除
zxc479773533	男	20	479773533@qq.com	13995718033	2018-06-15 16:21:26	3	修改	删除
test1	女	20	479773533@qq.com	13995718033	2018-06-19 19:20:47	1	修改	删除

图 VIII-3-17: 管理员修改信息后

②管理员管理航班

进入航班管理界面，看到如下的结果：

The screenshot shows the 'New Flight Entry' (新航班录入) interface. It includes fields for flight number (航班次), business class capacity (商务舱容量), economy class capacity (经济舱容量), departure time (起飞时间), arrival time (到达时间), start station (起点站), end station (终点站), business class price (商务舱价格), and economy class price (经济舱价格). A blue 'Add' (添加) button is located to the right of the input fields. Below the form is a red tip message: 'Tips: 1. Input information incomplete entry will be discarded. 2. You can only delete empty flights.' To the right is a 'Flight Management' (航班管理) table listing four existing flights with columns for flight number, business class capacity, economy class capacity, departure time, arrival time, start station, end station, business class price, economy class price, and an 'Delete' (删除) button.

航班次	商务舱容 量	经济舱容 量	起飞时间	到达时间	起点站	终点站	商务舱 价格	经济舱 价格	操作
CA1522	80/80	80/80	2018-06-18 17:55:00	2018-06-18 20:10:00	上海	北京	4710	3890	<button>删除</button>
ZH1865	80/80	80/80	2018-06-18 18:10:00	2018-06-18 20:45:00	上海	广州	1350	950	<button>删除</button>
MU9363	40/40	40/40	2018-06-18 18:30:00	2018-06-18 20:30:00	上海	武汉	1210	880	<button>删除</button>
CA1516	80/80	80/80	2018-06-18 18:55:00	2018-06-18 21:25:00	北京	上海	4590	3610	<button>删除</button>

图 VIII-3-18: 录入航班前

接下来尝试录入一个测试航班，并删除第 CA1522 次航班：

The screenshot shows the 'New Flight Entry' (新航班录入) interface with the following data entered: flight number TEST001, business class capacity 50, economy class capacity 50, departure time 2018/06/20 上午 01:00, arrival time 2018/06/20 上午 02:00, start station 'Test One' (测试一), end station 'Test Two' (测试二), business class price 999, and economy class price 888. A blue 'Add' (添加) button is visible to the right.

图 VIII-3-19: 录入航班

点击添加后，可以找到我们添加的航班：

The screenshot shows the 'Flight Management' (航班管理) table with the following data:

CZ6590	50/50	50/50	2018-06-19 20:00:00	2018-06-19 21:55:00	武汉	广州	780	550	<button>删除</button>
TEST001	50/50	50/50	2018-06-20 01:00:00	2018-06-20 02:00:00	测试一	测试二	999	888	<button>删除</button>
CZ6606	40/40	40/40	2018-06-20 06:40:00	2018-06-20 09:05:00	北京	武汉	570	330	<button>删除</button>

图 VIII-3-20: 录入航班后

同时，也找不到第 CA1522 次航班：

Tips: 1.输入信息不完整的录入将会被舍弃。2.您只能删除空航班哦。

航班管理										
航班次	商务舱容量	经济舱容量	起飞时间	到达时间	起点站	终点站	商务舱价格	经济舱价格	操作	
ZH1865	80/80	80/80	2018-06-18 18:10:00	2018-06-18 20:45:00	上海	广州	1350	950	删除	
MU9363	40/40	40/40	2018-06-18 18:30:00	2018-06-18 20:30:00	上海	武汉	1210	880	删除	
CA1516	80/80	80/80	2018-06-18 18:55:00	2018-06-18 21:25:00	北京	上海	4590	3610	删除	

图 VIII-3-21: 删除航班后

③管理员管理订单

进入管理订单界面，输入用户名 zxc479773533，得到：

根据用户名查找订单

订单管理										
用户名	航班次	座位类型	座位号	起飞时间	到达时间	起点站	终点站	票价	操作	
zxc479773533	MU2459	经济舱	51	2018-06-20 19:25:00	2018-06-20 21:35:00	武汉	北京	1150	删除	
zxc479773533	CA8203	经济舱	51	2018-06-21 00:40:00	2018-06-21 14:40:00	武汉	北京	1120	删除	
zxc479773533	CZ6172	商务舱	1	2018-06-19 06:45:00	2018-06-19 08:50:00	上海	武汉	620	删除	

图 VIII-3-22: 删除订单前

接着尝试删除第一个 MU2459 次航班的订单，得到：

根据用户名查找订单

订单管理										
用户名	航班次	座位类型	座位号	起飞时间	到达时间	起点站	终点站	票价	操作	
zxc479773533	CA8203	经济舱	51	2018-06-21 00:40:00	2018-06-21 14:40:00	武汉	北京	1120	删除	
zxc479773533	CZ6172	商务舱	1	2018-06-19 06:45:00	2018-06-19 08:50:00	上海	武汉	620	删除	

图 VIII-3-23: 删除订单后

综合以上几点，可以验证系统的管理员功能是完备的，达到了预期的要求。

IX 项目管理

IX.1 软件规模度量

为了对本机票预订管理系统软件进行信息与特性和软件复杂性评估，从而估算软件规模，本节将采用功能点技术对软件规模进行度量。

IX.1.1 计算未调整功能点数

对于本软件而言，其信息域特性如表 IX-1-1 所示：

表 IX-1-1：机票预订管理系统信息域特性

信息域特性	功能点数	复杂级别	特性系数
输入项数(<i>Inp</i>)	4	简单	3
输出项数(<i>Out</i>)	5	中等	5
查询数(<i>Inq</i>)	0	简单	3
主文件数(<i>Maf</i>)	1	简单	7
外部接口数(<i>Inf</i>)	10	中等	7

根据：

$$UFP = a_1 \times Inp + a_2 \times Out + a_3 \times Inq + a_4 \times Maf + a_5 \times Inf$$

可得：

$$UFP = 4 \times 3 + 5 \times 5 + 0 \times 3 + 1 \times 7 + 10 \times 10 = 144$$

IX.1.2 计算技术复杂因子

表 IX-1-2 展示了软件的技术因素及其影响值。影响值分布在 0~5 之间，0 表示无影响，5 表示有很大影响，中间的值影响力均匀分布。

表 IX-1-2：技术因素

序号	F_i	技术因素
1	5	数据通信
2	0	分布式数据处理
3	3	性能标准
4	2	高负荷的硬件
5	4	高处理率
6	0	联机数据输入

7	1	终端用户效率
8	0	联机更新
9	1	复杂的计算
10	4	可重用性
11	5	安装方便
12	4	操作方便
13	4	可移植性
14	5	可维护性

根据：

$$\left\{ \begin{array}{l} DI = \sum_{i=1}^{14} F_i \\ TCF = 0.65 + 0.01 \times DI \end{array} \right.$$

可得：

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i = 1.03$$

又结合 UFP 和 TCF，可得：

$$FP = UFP \times TCP = 148$$

又根据 Kemerer 模型，计算得：

$$E = 60.62 \times 7.728 \times 10^{-8} \times 148^3 = 15.1$$

根据 COCOMO2 模型 $T = 2.5 \times 148^{0.35} = 14.37$ 计算，正常情况下完成这一项目需要大约 15 天。项目拟聘用 4 个人开发。

IX.2 人员安排

对于本机票预订管理系统的开发，使用 4 个人进行开发。一方面，人数太多会增加开发时的沟通成本，反而可能增加开发时间，另一方面，管理系统使用 B/S 架构，浏览器端和服务器端各自为独立的模块，互相之间进行通信，开发的人员过多容易出现代码接口混乱，代码难以管理的情况，开发人员较少则便于管理和交流。

在分组方面，由于没有很多的开发人员，因此不需要细分小组，直接采用图 IX-2-3 所示的组织结构即可。其中产品负责保证整个程序功能的正确性以及开发时间，负责与甲方进行交流以及考虑产品的法律问题；设计负责绘制产品前端的详细图纸与功能效果，并转交给前端程序员进行开发；程序主要由一个前端程序员和一个后台程序员开发，各自独立完成自己的模块，并及时向产品经理汇报自

己的工作进展状况。

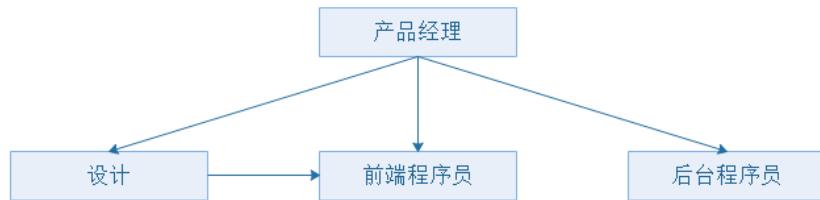


图 IX-2-1: 小组组织结构

IX.3 项目进度计划

鉴于管理系统软件中数据库的核心作用，它必须被最先配置，且首先将数据库接口设计完成，供其他模块调用。在此后的过程中，由于前端页面和后台之间要进行数据交互，因此他们需要被同时开发。每一步开发完成后，都可以及时进行调试。当前端和后台都完工后，可以在真实的环境上进行部署调试，同时考虑性能优化问题。总体的开发工程网络图见图 IX-3-2。

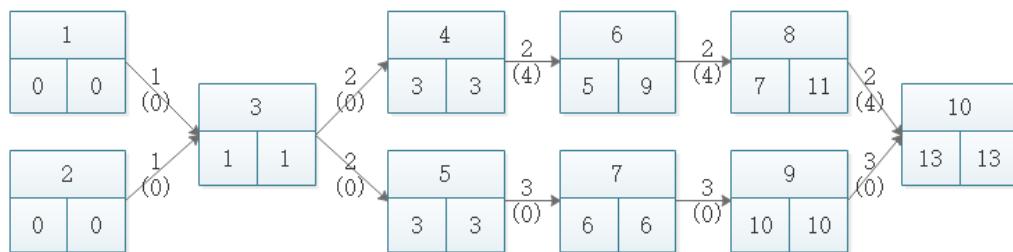


图 IX-3-2: 工程网络图

对于图 IX-3-2 中的编号做出如下解释：

表 IX-3-3: 工程网络图过程

过程	开发部分
1->3	设计师出前端设计图
2->3	数据库创建与配置
3->4	登录与注册界面前端开发
3->5	登录与注册后台开发
4->6	用户界面前端开发
5->7	用户功能后台开发
6->8	管理员界面前端开发
7->9	管理员界面后台开发
8->10	前端调试

9->10

后台与前端连接，并在服务器部署

由于整个工程中基本没有交叉依赖关系，大部分时间都在并行开发，并且前端的开发要比后台快，因此工程网络图中的关键路径为下方的后台开发路径，要减少开发时间，就需要从对结束时间影响最大的路径减少开发时间。

对于本软件的预想开发过程，使用图 IX-3-3 所示 Gantt 图表示。

#	Task Name	Start	Finish	Duration	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	设计师出前端设计图	0	1	0.1 周														
2	数据库创建与配置	0	1	0.1 周														
3	登录与注册界面前端开发	1	3	0.3 周														
4	登录与注册界面后端开发	1	3	0.3 周														
5	用户界面前端开发	3	5	0.3 周														
6	用户界面后台开发	3	6	0.4 周														
7	管理员界面前端开发	5	7	0.3 周														
8	管理员界面后台开发	6	10	0.6 周														
9	前端调试	7	9	0.3 周														
10	后台与前端连接，并在服务器部署	10	13	0.4 周														

图 IX-3-3: 机票预订管理系统 Gantt 图

附录 A 参考文献

- 1 张海藩, 倪宁著. 软件工程 (第 3 版). 北京: 人民邮电出版社, 2010.
- 2 [美] Roger S. Pressman, Bruce R. Maxim 著, 郑人杰, 马素霞等译. 软件工程 实践者的研究方法 (原书第 8 版). 北京: 机械工业出版社, 2016.
- 3 王珊, 萨师煊著. 数据库系统概论 (第 5 版). 北京: 高等教育出版社, 2014.
- 4 [美] Abraham Silberschatz, Henry F. Korth, S.Sudarshan 著, 杨冬青, 李红燕, 唐世渭译. 数据库系统概念 (原书第 6 版). 北京: 机械工业出版社, 2013.
- 5 [美] Lynn Beighley, Michael Morrison 著, 苏金国, 徐阳译. Head First PHP & MySQL. 北京: 中国电力出版社, 2011.
- 6 [美] Nicholas C. Zakas 著. 李松峰, 曹力译. JavaScript 高级程序设计(第 3 版) . 人民邮电出版社, 2016.

附录 B 心得与体会

通过这次软件工程试验大型作业，我体会到了一个软件开发的方方面面，深刻理解了软件工程的各项理论。明白了在一个项目开始之前，需要问题定义和可行性分析，需要谨慎的去分析项目需要什么，目的是什么，是否值得做，成本是多少等等问题，真正把一个项目上升到工程的水准。对于将要开发的项目，详细的工程文档能够快速解决开发者之间的沟通问题，能够事先合理地构建出工程的结构，方便开发者后续的快速开发，可以统一项目工程的数据结构和接口，节省开的时间精力和资源。

同时，在完成此项目的设计过程中，我的综合能力也得到了锻炼，对于问题的解决考虑得也更加周全。同时，借由这个项目，我也更加熟练了网站的搭建与开发，MySQL 数据库的使用等等技术能力。

在编写此文档的过程中，我发现很多方面与书本上的描述不完全相同，虽然大体内涵相同，但是细节的不同也给此报告的编写带来了不小困扰，因此希望以后的课程中可以对报告编写的细节方面做一些描述。