

基于 java web 的机票订购管理系统设计与实现

摘 要

随着社会发展的不断进步，民航事业的壮大，人们消费水平的提高，乘坐民航的消费者也越来越多，旅游也逐渐成为普通老百姓的生活组成部分，飞机票预定查询系统在各机票预定网点中的作用也越显重要。目前，我国一些旅行社和酒店的机票预定还停留在人工处理阶段，这已经严重制约了工作效率，在计算机技术高速发展的今天，有必要引入高效的计算机系统，来协助处理机票预定工作。因此，开发一套具有完整的存储，查询，核对，打印机票功能的实时机票预定系统势在必行。

机票预定系统是为机场工作人员和客户提供订票退票等与机票相关内容的管理的系统，它具有开放体系结构的、易扩充的、易维护的、具有良好人机界面的优点。它除克服了存储乘客信息少，查询效率低下等问题外，更重要的是其安全性，可靠性，实现航空公司的机票销售的自动化。它为企业的决策层提供准确、精细、迅速的机票销售信息，为乘客出行提供方便，便于机场工作人员对机票信息进行管理，提高了机场工作人员对机票管理的工作效率。

机票订购系统采用 MySQL 来设计数据库，并使用当前优秀的开发工具——Myeclipse，主流技术 jsp、ajax、easyui 以及主流 J2EE 架构—SSH 来实现系统。

关键词： 机票订购系统、JavaWeb 系统、管理系统 、jsp、ajax、easyui

Ticket ordering system

Abstract

Along with the advance of social development, civil aviation growing, people raise the level of consumption, consumers more and more, in civil aviation tourism has gradually become a part of ordinary life, air ticket reservation query system in the role of the ticket booking outlets are more important. At present, our country some travel agencies and hotels is ticket reservation in artificial processing stage, which has seriously restricted the work efficiency, in today's high-speed development of computer technology, it is necessary to introduce effective computer system, to assist processing flight reservation. Therefore, to develop a set of complete with storage, query, check and print the ticket function real-time ticket reservation system is imperative.

Ticket reservation system is for the airport staff and customers booking ticket and ticket related contents and management system, such as it has open architecture, easy expansion, easy maintenance, has a good man-machine interface. The advantages of it in addition to overcome the passenger information less storage, such problems as low query efficiency, more important is the safety, reliability, realize the automation of airline ticket sales. It for enterprise decision-making to provide accurate, precise, rapid ticket sales information, provides the convenience for passengers, facilitate the ticket information management of the airport staff, improve the efficiency of the airport staff to ticket management.

Ticket ordering system to design the database, MySQL, and use the best development tools Myeclipse, as well as the mainstream - SSH J2EE architecture to implement the system.

Keywords: Ticket ordering system, JavaWeb system, management system

TICKET ORDERING SYSTEM.....	2
ABSTRACT.....	2
第 1 章 绪 论.....	- 1 -
1.1 机票订购系统的开发背景.....	- 1 -
1.2 设计目的与意义.....	- 2 -
第 2 章 系统的需求分析和技术设计.....	- 2 -
2.1 可行性分析.....	- 2 -
2.1.1 操作可行性.....	- 2 -
2.1.2 经济可行性.....	- 2 -
2.1.3 技术可行性.....	- 2 -
2.2 方案的设计与比较.....	- 2 -
2.2.1 C/S 设计结构和 B/S 设计结构比较.....	- 2 -
2.2.2 系统模式的设计.....	- 2 -
2.2.3 系统设计的技术选择.....	- 2 -
2.2.4 系统的运行环境.....	- 3 -
2.3 系统基本功能需求.....	- 3 -
2.3.1 系统结构.....	- 3 -
2.3.2 系统业务流程图.....	- 3 -
2.4 数据库需求分析.....	- 6 -
2.5 系统目标.....	- 6 -
第 3 章 系统总体设计.....	7
3.1 总体设计.....	7
3.1.1 处理流程.....	7
3.1.2 系统的数据流程图.....	7
3.2 系统结构图.....	9
3.3 数据库设计.....	10
3.4 逻辑结构设计.....	12
3.5 数据库表的设计.....	12
第 4 章 系统详细设计与实现.....	14
4.1 前台系统设计实现.....	14
4.1.1 首页界面实现.....	14
4.1.2 用户注册功能实现.....	16
4.1.3 用户登录功能实现.....	17
4.1.4 航班信息查询.....	18
4.1.5 机票订购功能实现.....	20
4.1.6 个人信息管理功能实现.....	21
4.1.7 订单管理功能实现.....	22
4.2 后台系统设计实现.....	25
4.2.1 管理员登录功能.....	25

4.2.2 用户管理.....	27
4.2.3 客机管理.....	28
4.2.4 航班管理.....	31
4.2.5 机票管理.....	33
4.2.11 系统管理.....	35
第 5 章 系统的调试与测试.....	37
5.1 软件测试的目的.....	37
5.2 软件测试的任务.....	37
5.3 测试环境.....	37
5.4 白盒、黑盒测试.....	38
5.5 手工测试与自动测试.....	38
5.6 软件缺陷.....	38
结束语.....	39
参考文献.....	40

第 1 章 绪 论

1.1 机票订购系统的开发背景

从 20 世纪 70 年代起, 欧美及日本等发达国家就开始利用计算机开发售票业务, 如英国的 Tribute 系统、日本的 MARS 系统和德国的 TRAINS 系统等。随着互联网技术与电子商务的迅猛发展, 利用 Web 技术开发的网络售票系统也得到了蓬勃的发展并日益影响到人们的日常生活, 例如澳大利亚的 BASS 系统。20

世纪 90 年代开始, 我国的部分大型企业为了适应市场经济的条件下自身发展的需要, 陆续研究和开发了利用计算机和计算机网络运行的票务系统, 如我国的铁路售票系统, 该系统通过十年左右的开发、应用和完善, 系统已经覆盖了全国的铁路主干网络, 平均每天发售火车票达 250 万张, 计算机网络售票业务额占票务总收入的 90%以上。

目前网上订票系统及其类似应用软件的开发技术大多基于 B/S 架构, 在系统开发过程中用到的主要技术包括动态网页设计技术, 如 ASP、JSP 和 PHP 技术等、.NET 和 J2EE 技术等。在这些技术中 ASP 是一个 Web 服务器端的开发环境, 它的应用可以产生和运行动态交互的 Web 服务应用程序, 其开发语言一般为 VBScript 或者 JavaScript。具有简单易懂、无需编译、与浏览器无关、兼容性和扩展性较好等

特性。PHP 是一种跨平台的服务器端的嵌入式脚本语言, 具有便于与数据库连接和面向对象等特点。JSP 是由 SUN 公司推出的站点开发语言, 具有代码生成和显示分离、可重用性能好页面开发简单等特点。 .NET 技术是为了适应基于 Web 的应用程序和 Web 服务而产生的, 它基于 XML 和 SOAP 等互联网标准 将远程服务器所提供的计算能力和允许用户交互操作所必需的通信结合在一起, 实现了基于 Web 的分布式计算模式。.NET 采用的 CLR 技术规范允许开发者应用自己最熟悉的语言来创建应用程序。

J2EE 是一套全然不同于传统应用开发的技术架构, 包含许多组件, 主要可简化且规范应用系统的开发与部署, 进而提高可移植性、安全与再用价值。J2EE 核心是一组技术规范与指南, 其中所包含的各类组件、服务架构及技术层次, 均有共通的标准及规格, 让各种依循 J2EE 架构的不同平台之间存在良好的兼容性, 解决过去企业后端使用的信息产品彼此之间无法兼容, 企业内部或外部难以互通的窘境。J2EE 体系结构提供中间层集成框架用来满足无需太多费用而又需要高可

用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE 降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力支持，完全支持 Enterprise JavaBeans，有良好的向导支持打包和部署应用，添加目录支持，增强了安全机制，提高了性能。J2EE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制。

1.2 设计目的与意义

随着互联网的飞速发展，网络业务也得到了迅猛地发展。网上订票系统就是互联网和网络业务发展的产物。该类系统的出现和应用一方面扩展了票务销售的业务领域，丰富了票据业务营销的方式和手段，另一方面，人们不用出门就可以购买到相关的活动票据，大大方便了人们的生活。因此，网上订票系统对社会的发展和社会生活方式的变革带来了深远的影响。

本课题的任务就是开发一套网上机票订购系统，要求能通过 Internet 网络为顾客提供航班查询，购买机票以及退票的服务。该系统在 Web 中能够完成客户注册与登录、查询航班、出票等功能，在航空公司内部能够完成管理员添加或删除航班、用户管理和票务维护等功能。本系统采用 Struts+Hibernate+Spring 的 J2EE

框架，这个开源的 WEB 开发架构在市场上占主导地位。其采用的 MVC 模式使页面的显示与处理分离,使前台页面人员更专注于页面的设计,而不用关注业务逻辑。编程人员只需集中精力于业务逻辑的实现提高开发效率。

第 2 章 系统的需求分析和技术设计

2.1 可行性分析

2.1.1 操作可行性

本系统的开发是在三个月的时间内完成的。前期主要是以学习以及收集资料为主，接下来就是对系统的分析，设计数据库，界面，以及中间的连接。对于计算机专业作为毕业设计来说是可行的。

2.1.2 经济可行性

经济可行性主要是对项目的经济效益进行评价。目前，具备了运行网络平台的 MIS 的硬件基础，而且本系统是我组成员自行开发，免费帮助设计并实施的。因此开发、设计这套系统的支出费用是学校可以承担的，即经济上是可行的。

2.1.3 技术可行性

技术上的可行性要考虑将来采用的硬件和软件技术能否满足用户提出的要求。基于当前的计算机网络技术和数据技术已成熟，而且管理信息系统（MIS）的各种开发技术也已经相当成熟，并且在各个领域都不乏成熟的案例。所以开发一套网络平台的多用户共享信息的在线考试系统在技术上是可行的。

2.2 方案的设计与比较

2.2.1 C/S 设计结构和 B/S 设计结构比较

目前网络应用软件运行的模式主要有两类：Client/Server 模式，Browser/Server 模式。前者的主要的缺点是维护、升级较为麻烦。且要同时开发服务器端和客户端；后者是近几年伴随 Internet 迅速发展起来的一种技术，B/S 模式客户端是一个标准的浏览器，服务器端是 web server，而 Web server 与数据库和应用服务器的紧密结合，使得这种模式的应用范围不断扩大，它已不仅仅用于网上查询，有很多企业部门的业务系统，企业的 MIS 系统纷纷采用这种模式，它的优点是便于扩展应用、升级维护简便、不需要开发专门的客户端、不需要对用户进行特殊的设置和软件安装，降低了维护成本，客户端只要有浏览器就可以了。

2.2.2 系统模式的设计

本系统的设计是两种用户，一种是普通用户，一种是管理员用户。权限都不一样；

2.2.3 系统设计的技术选择

处于安全性，功能可扩展性。我们采用 java ee。
数据库使用开源的 Mysql；
服务器采用 Tomcat；

2.2.4 系统的运行环境

- (1). 系统:Windows XP 及其以上版本
- (2). Web 服务器:Tomcat6 及其以上版本
- (3). 数据库服务器:MySql
- (4). 浏览器:IE7, 8, 9、火狐、Chrome

2.3 系统基本功能需求

系统中涉及系统普通用户和管理员两种权限；接下来针对这两种权限分别说明下功能结构；

2.3.1 系统结构

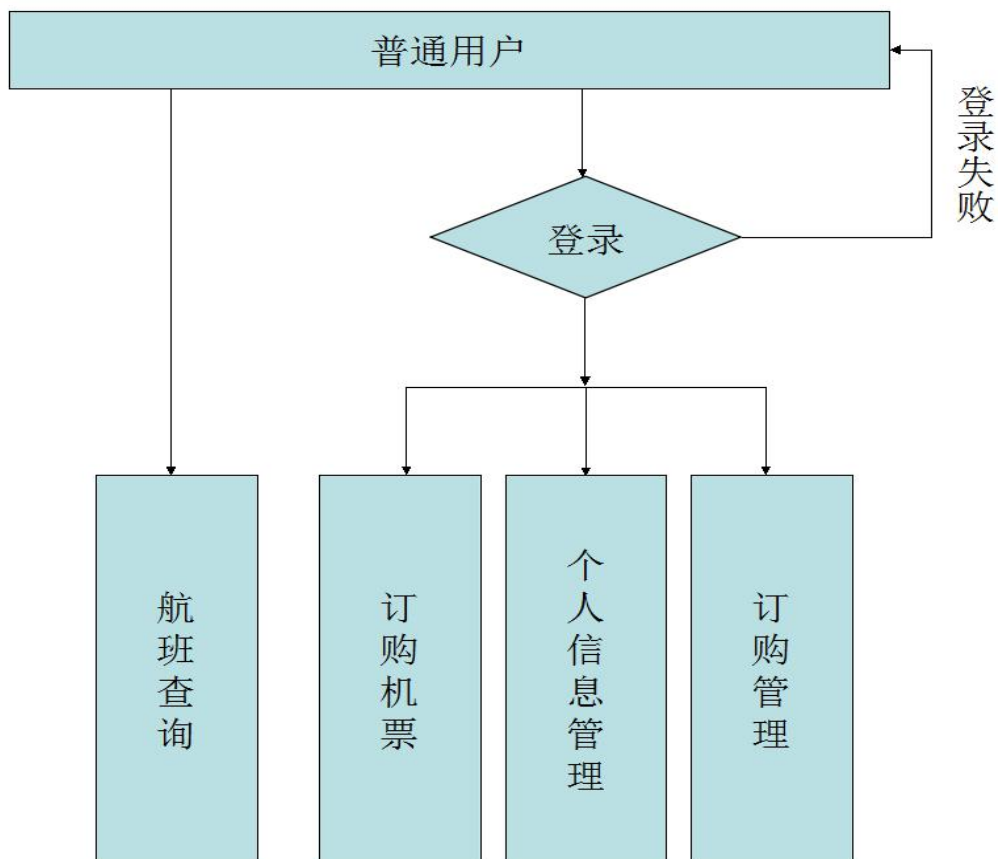
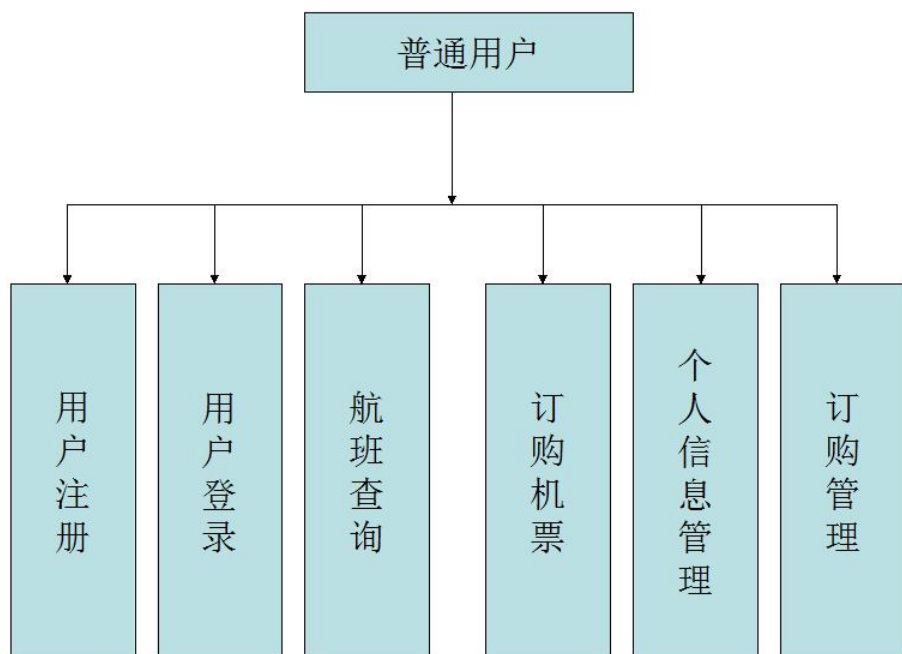
主要功能有：

普通用户：用户注册，用户登录；查询机票信息，购买机票；个人信息管理，订购管理（退票，改签）；

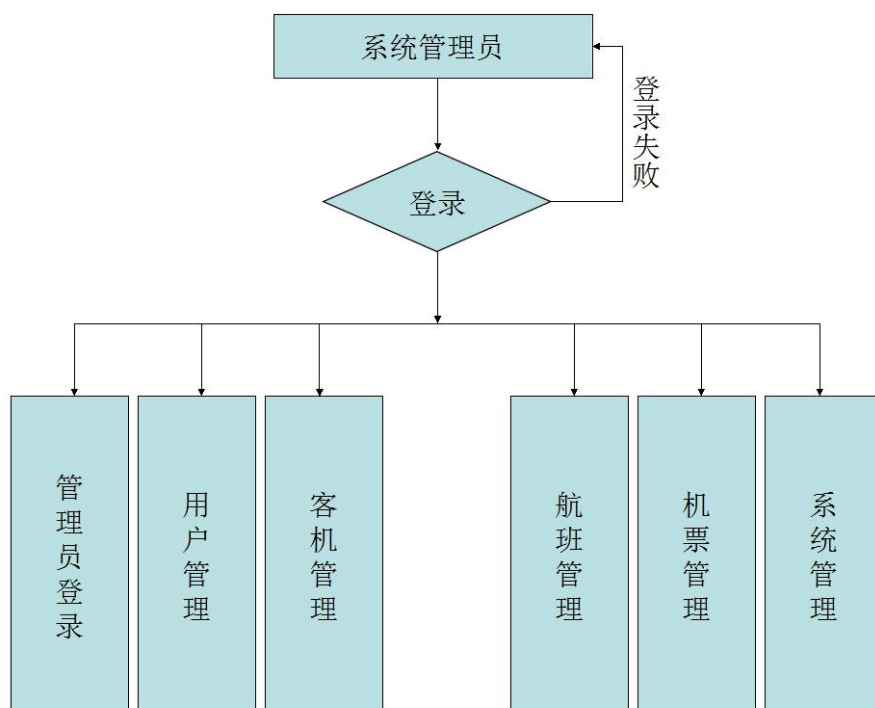
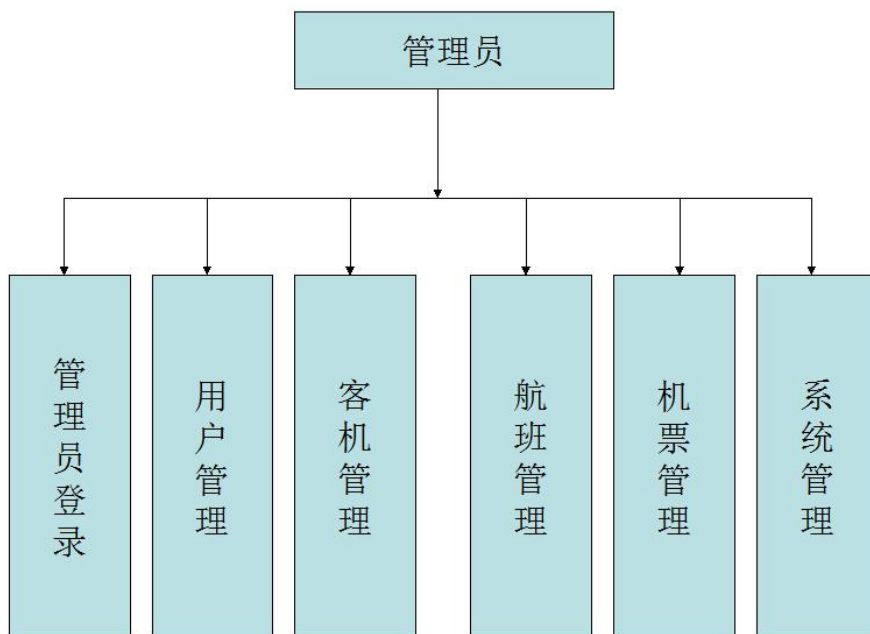
系统管理员：管理员登录；用户管理；客机管理；航班管理；机票管理；系统管理（修改密码，安全推出）；

2.3.2 系统业务流程图

普通用户：



管理员:



2.4 数据库需求分析

根据上面业务流程图所涉及到的内容，数据库要设计的表有:用户表，管理员表，~~客机表~~，航班表，机票订购表；

2.5 系统目标

本选题系统的设计主要将实现以下目标：

- (1). 人性化设计:系统界面友好,操作简单;
- (2). 前台实现用户登录，用户注册，航班查询，机票订购，个人信息管理，机票订购管理；
- (3)后台实现管理员登录，用户管理，~~客机管理~~，航班管理，订购管理，系统管理；

第 3 章 系统总体设计

3.1 总体设计

3.1.1 处理流程

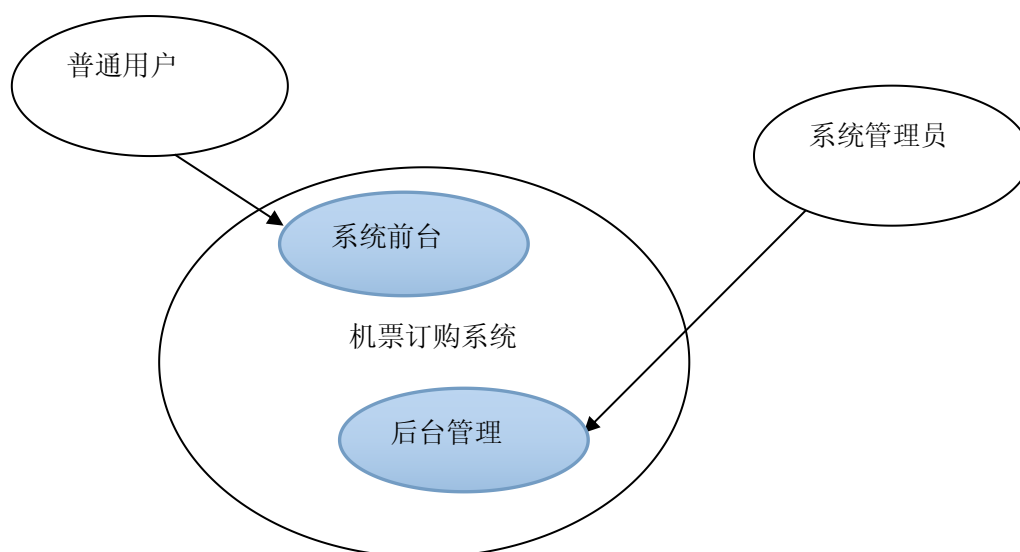
一，普通用户

- (1). 打开系统，进入系统前台首页界面, 用户可以浏览商品，查看新闻和公告；
- (2). 登录后，可以购买商品；
- (3). 用户进入个人中心，可以修改个人信息以及管理订单
- (4). 退出系统。

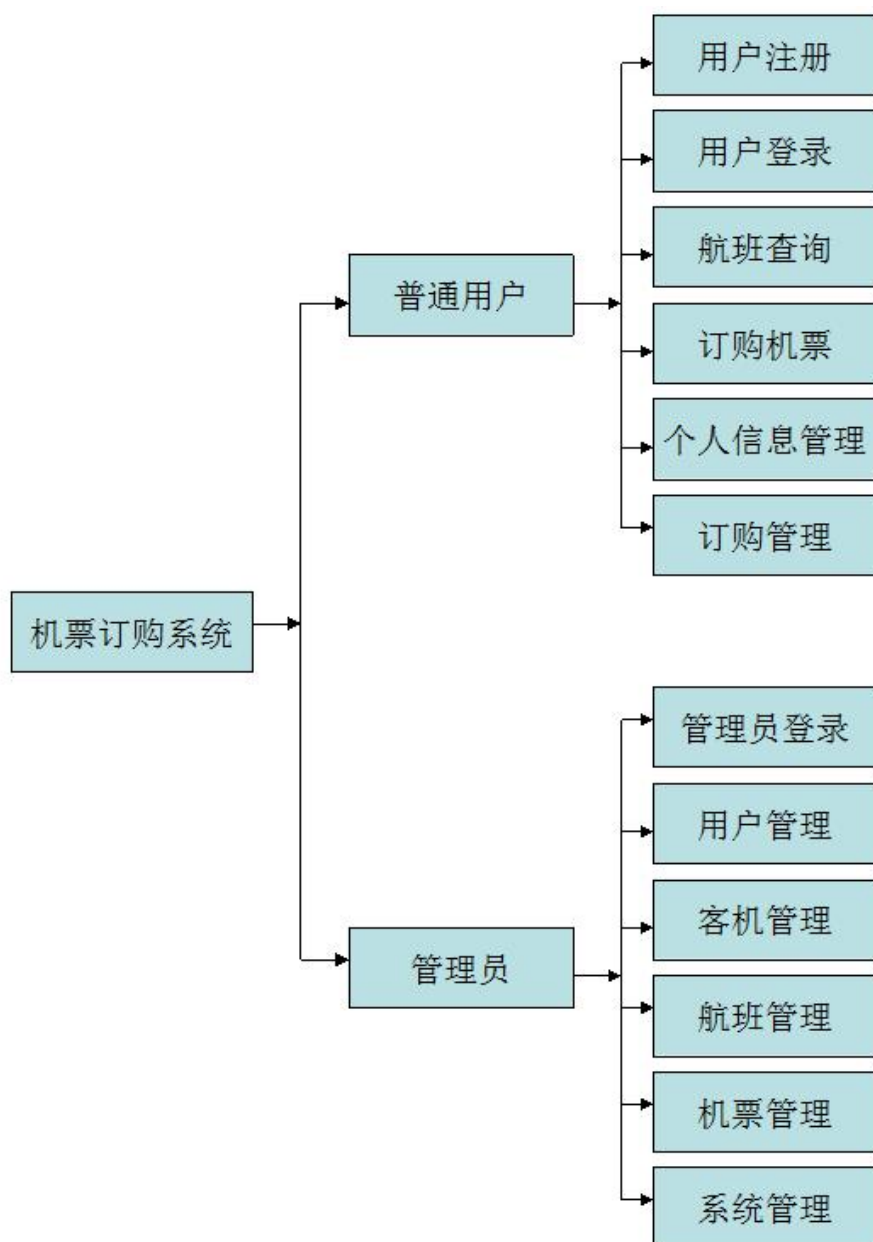
二，管理员

- (1). 打开系统，进入系统登录界面，
- (2). 通过验证后，进入系统管理主界面
- (3). 用户操作，执行相应的功能，
- (4). 退出系统。

3.1.2 系统的数据流图



3.2 系统结构图



针对本系统的需求，设计出如下面所示的数据项和数据结构：

用户表:编号，用户名，密码，真实姓名，性别，身份证，联系电话；

管理员表:编号，用户名，密码，真实姓名，联系电话；

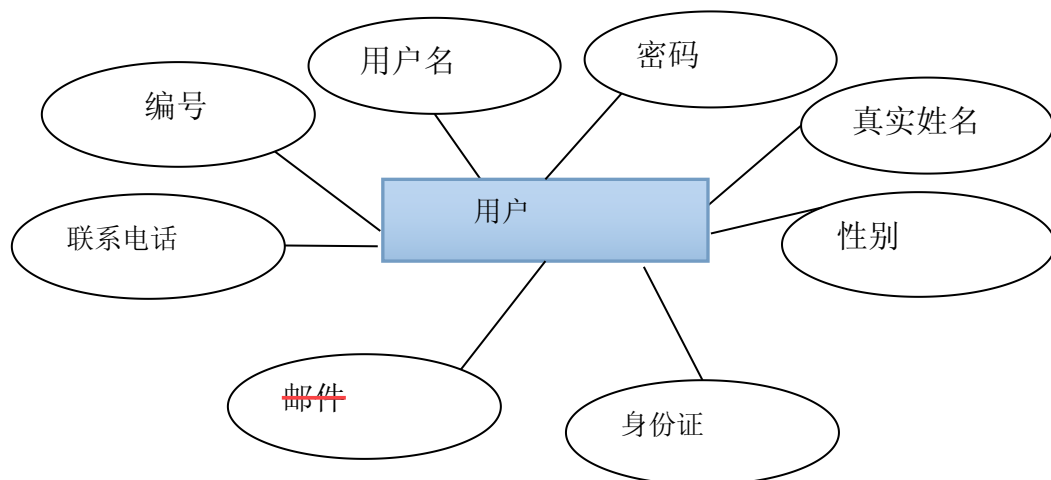
客机表:编号，客机名称，购买日期，服役日期，备注；

航班表: 编号，航班名称，航班类型，出发城市，目的城市，出发时间，到点时间，使用客机，经济舱票价以及座位数，头等舱票价以及座位数；

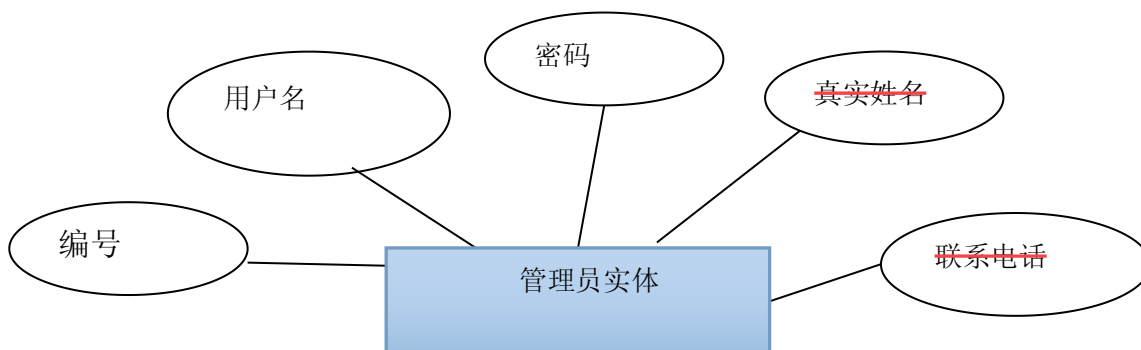
订单表: 编号，订单号，订购时间，票价，座舱类别，航班，用户，订购数量，总价；

3.3 数据库设计

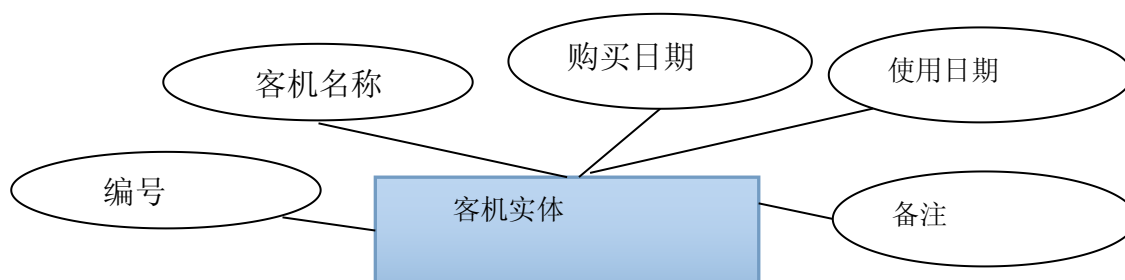
(1).用户信息 E-R 图



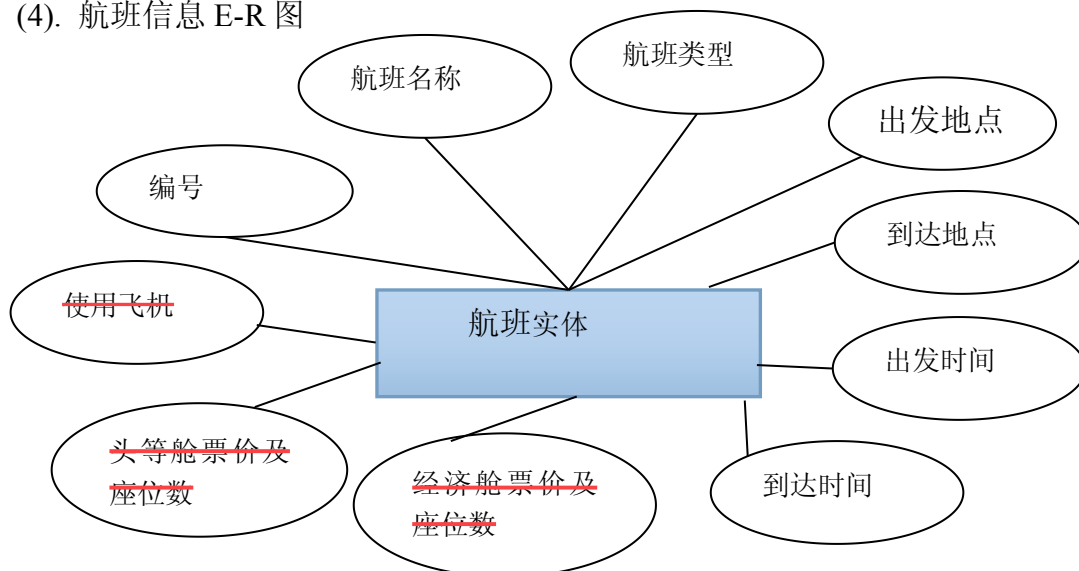
(2). 管理员信息 E-R 图



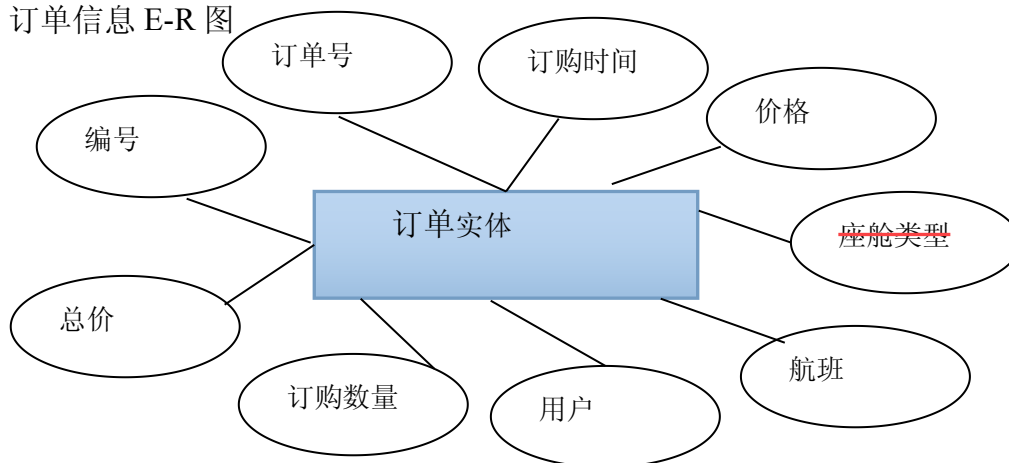
(3). 客机信息 E-R 图



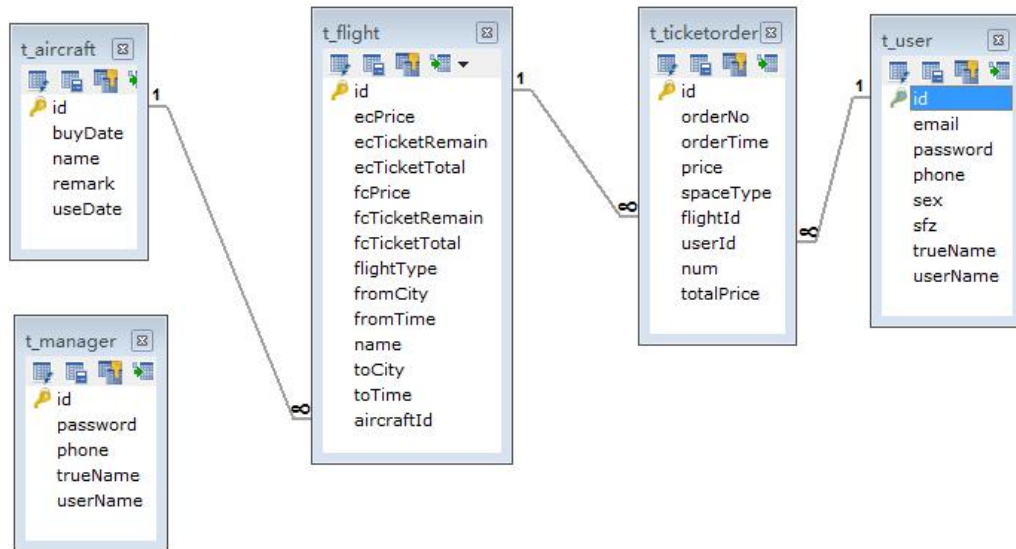
(4). 航班信息 E-R 图



(5). 订单信息 E-R 图



(8) 实体之间的关系图



3.4 逻辑结构设计

- 1, 航班和客机实体的关系是多对一;
- 2, 订单实体和航班实体的关系是多对一;
- 3, 用户和订单是一对多;

3.5 数据库表的设计

(1). 用户信息表

字段名	数据类型	长度	是否主键	描述
id	Int	11	是	自动编号
Username	varchar	20		用户名
Password	varchar	20		密码
trueName	Varchar	20		用户姓名

sex	Varchar	5		性别
sfz	Varchar	20		身份证
email	Varchar	20		邮件
phone	Varchar	20		联系电话

(2). 客机表

字段名	数据类型	长度	是否主键	描述
id	varchar	40	是	
name	varchar	20		客机名称
buyDate	datetime			购买日期
useDate	datetime			使用日期
remark	varchar	1000		备注

(3). 航班表

字段名	数据类型	长度	是否主键	描述
id	Int	11	是	自动编号
ecPrice	int			经济舱票价
ecTicketRemain	int			经济舱剩余票
ecTicketTotal	int			经济舱总票数
fcPrice	int			头等舱票价
fcTicketRemain	int			头等舱剩余票
fcTicketTotal	int			头等舱总票数
flightType	varchar	20		航班类型
fromCity	varchar	20		出发地点
fromTime	datetime			出发时间
Name	varchar			航班名称
toCity	varchar			到达城市
toTime	datetime			到达时间
aircraftId	int			使用飞机

(4). 管理员表

字段名	数据类型	长度	是否主键	描述
id	Int	11	是	自动编号
userName	Varchar	20		用户名
password	Varchar	20		密码
trueName	Varchar	20		真实姓名
phone	Varchar	20		联系电话

(5). 订单表

字段名	数据类型	长度	是否主键	描述
id	Int	11	是	自动编号
orderNo	varchar	30		订单号
orderTime	Datetime			订购时间
price	int			价格
spaceType	varchar	20		座位类型
flightId	int			航班 ID
userId	int			用户 ID
num	int			购买数量
totalPrice	int			总价

第 4 章 系统详细设计与实现

4.1 前台系统设计实现

采用 struts2+spring+hibernate 三大框架，自主设计，分配各个功能模块，让其实现各自的功能，共同完成程序的运行。

4.1.1 首页界面实现

效果图：



视频观赏



用户信息

用户名:

密码:

国内航班

航班	航线	出发日期	起落时间	票价	操作
大连航空003	北京 - 太原	11-11	03:11 - 05:11	398	订票
大连航空001	大连 - 广州	11-10	05:11 - 05:11	500	订票
大连航空003	北京 - 合肥	11-10	03:11 - 03:11	600	订票
上海航空002	上海 - 广州	11-10	03:11 - 03:11	400	订票
上海航空002	贵州 - 广州	11-10	03:11 - 03:11	500	订票

国际航班

航班	航线	出发日期	起落时间	票价	操作
大连航空003国际航班	上海 - 加拿大	11-11	14:31 - 14:31	1300	订票
南方航空002	北京 - 鄂尔多斯	11-10	15:30 - 17:00	400	订票

相关说明:

用户进入系统首页, 上方从左是系统 Logo 最左侧是一个登录, 注册, 航班查询功能; 右侧的话, 是航班信息, 上方是最新的国内航班信息, 下方是国际航班信息; 视频欣赏等等

关键代码实现:

主界面布局:

```

<table width="960px" align="center" cellspacing="20px;" cellpadding="20px;">
  <tr>
    <td colspan="2">
      <jsp:include page="common/head.jsp"></jsp:include>
    </td>
  </tr>
  <tr>
    <td align="center" style="padding-bottom: 10px;">
      <table width="100%">
        <tr>
          <td align="center" width="10%"><a href="${pageContext.request.contextPath}/flight_indexList.action"></a></td>
          <td align="center" width="90%"><iframe width="420" scrolling="no" height="60" frameborder="0" allowtransparency="true" src="http://i.tianqi.com/index.php?c=code&id=12&icon=1&num=5" style="float:right;"></iframe></td>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

Foot.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<div align="center" style="padding-top: 200px" >
Copyright © 2013-2015 XXX 版权所有
</div>
```

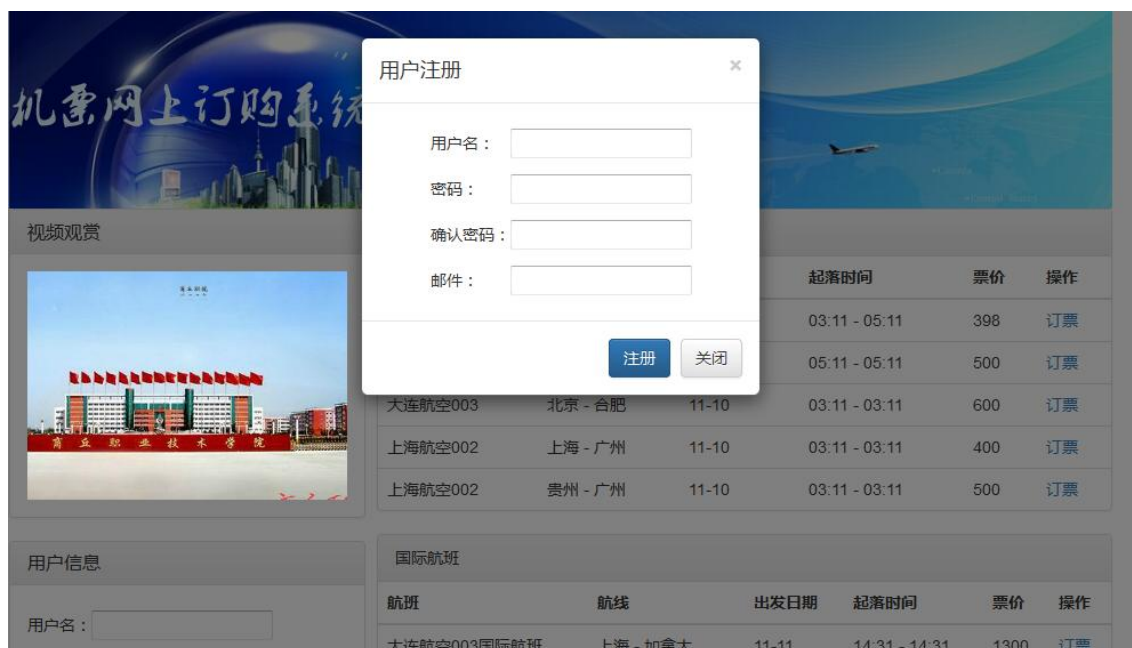
查询航班信息:

```
public List<Flight> findFlightList(Flight s_flight,PageBean pageBean){
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from Flight");
    if(s_flight!=null){
        if(StringUtil.isEmpty(s_flight.getName())){
            hql.append(" and name like '%" +s_flight.getName()+"%'");
        }
        if(StringUtil.isEmpty(s_flight.getFlightType())){
            hql.append(" and flightType='"+s_flight.getFlightType()+"'");
        }
        if(StringUtil.isEmpty(s_flight.getFromCity())){
            hql.append(" and fromCity='"+s_flight.getFromCity()+"'");
        }
        if(StringUtil.isEmpty(s_flight.getToCity())){
            hql.append(" and toCity='"+s_flight.getToCity()+"'");
        }
        if(s_flight.getFromTime()!=null){
            hql.append("                and                fromTime
<" +DateUtil.formatDate(s_flight.getFromTime(), "yyyy-MM-dd")+""");
        }

        hql.append(" order by fromTime desc");
        if(pageBean!=null)                return  baseDAO.find(hql.toString().replaceFirst("and",
"where"), param, pageBean);
    }else{
```

4.1.2 用户注册功能实现

点击“注册”功能，弹出用户注册弹窗；如图：



相关说明：

用户填写 用户名，密码，确认密码，邮件。然后点击“注册”，即可完成用户注册；

关键代码：

```
public String register()throws Exception{
    JSONObject result=new JSONObject();
    if(userService.existUserWithUserName(user.getUserName())){
        result.put("error", "用户名已存在，请更换！");
    }else{
        userService.saveUser(user);
        result.put("success", true);
    }
    ResponseUtil.write(ServletActionContext.getResponse(), result);
    return null;
}
```

4.1.3 用户登录功能实现

效果图：

机票网上订购系统

视频观赏



国内航班

航班	航线	出发日期	起落时间	票价	操作
大连航空003	北京 - 太原	11-11	03:11 - 05:11	398	订票
大连航空001	大连 - 广州	11-10	05:11 - 05:11	500	订票
大连航空003	北京 - 合肥	11-10	03:11 - 03:11	600	订票
上海航空002	上海 - 广州	11-10	03:11 - 03:11	400	订票
上海航空002	贵州 - 广州	11-10	03:11 - 03:11	500	订票

用户信息

欢迎旅客: 111 [【安全退出】](#)

[进入个人中心](#)

国际航班

航班	航线	出发日期	起落时间	票价	操作
大连航空003国际航班	上海 - 加拿大	11-11	14:31 - 14:31	1300	订票
南方航空002	北京 - 鄂尔多斯	11-10	15:30 - 17:00	400	订票

相关说明:

输入用户名, 密码, 然后点击登录, 假如用户名, 密码正确, 则登录成功, 否则提示错误信息;

关键代码:

用户登录:

```
public String login()throws Exception{
    HttpSession session=request.getSession();
    User currentUser=userService.login(user);
    boolean flag=false;
    if(currentUser==null){
        flag=false;
    }else{
        session.setAttribute("currentUser", currentUser);
        flag=true;
    }
    JSONObject result=new JSONObject();
    result.put("success", flag);
    ResponseUtil.write(ServletActionContext.getResponse(), result);
    return null;
}
```

4.1.4 航班信息查询

效果图:

视频观赏



航班查询结果

航班	航线	出发日期	起落时间	票价	操作
大连航空001	大连 - 广州	11-10	17:58 - 17:58	500	订票
上海航空002	上海 - 广州	11-10	15:30 - 15:30	400	订票
上海航空002	贵州 - 广州	11-10	15:30 - 15:30	500	订票

[首页](#)
[1](#)
[尾页](#)

用户信息

欢迎旅客：111 [【安全退出】](#)
[进入个人中心](#)

航班查询

出发地点：
 到达地点：

相关说明：

在航班查询框内，输入查询条件信息，然后点击搜索，即可查询出相应结果；

关键代码实现：

条件查询航班信息

```

public List<Flight> findFlightList(Flight s_flight,PageBean pageBean){
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from Flight");
    if(s_flight!=null){
        if(StringUtil.isNotEmpty(s_flight.getName())){
            hql.append(" and name like '%"+s_flight.getName()+"%'");
        }
        if(StringUtil.isNotEmpty(s_flight.getFlightType())){
            hql.append(" and flightType='"+s_flight.getFlightType()+"'");
        }
        if(StringUtil.isNotEmpty(s_flight.getFromCity())){
            hql.append(" and fromCity='"+s_flight.getFromCity()+"'");
        }
        if(StringUtil.isNotEmpty(s_flight.getToCity())){
            hql.append(" and toCity='"+s_flight.getToCity()+"'");
        }
        if(s_flight.getFromTime()!=null){
            hql.append(" and fromTime <='"+DateUtil.formatDate(s_flight.getFromTime(), "yyyy-MM-dd")+"'");
        }
    }
    hql.append(" order by fromTime desc");
    if(pageBean!=null){

```


4.1.5 机票订购功能实现

击“订票”，显示：

机票网上订购系统

视频观赏

订购机票

航班： 大连航空001

航线： 大连 -> 广州

出发到点时间：2015-11-10 17:58 ~ 2015-11-19 17:58

票价：
☒ 500元(经济舱)
☐ 800元(头等舱)

订购数量：

提交

用户信息

欢迎旅客：111 【安全退出】

[进入个人中心](#)

说明：

点击“订票”跳转到机票订购界面，用户可以选择座舱类别（经济舱，头等舱），可以设置要订购的机票数量。最后点击提交。完成机票订购；

关键代码实现：

通过 ID 获取航班信息；

```
/**
 * 机票订购预操作
 * @return
 * @throws Exception
 */
public String preTicketOrder()throws Exception{
    flight=flyService.getFlightById(flightId);
    mainPage="ticket/ticketOrder.jsp";
    return SUCCESS;
}
```

保存订单信息：

```
/**
 * 保存机票订单
 * @return
 * @throws Exception
 */
public String saveTicketOrder()throws Exception{
    String orderNo="NO"+DateUtil.getCurrentDateStr(); // 生成订单号
    ticketOrder.setOrderNo(orderNo);
}
```

4.1.6 个人信息管理功能实现

进入个人中心，如图：



点击“修改”按钮，如图：



用户对信息进行编辑，修改完，点击“保存”，完成用户信息修改；

关键代码实现：

用户信息保存

```
/**
 * 用户修改
 * @return
 * @throws Exception
 */
public String save()throws Exception{
    HttpSession session=request.getSession();
    JSONObject result=new JSONObject();
    userService.saveUser(user);
    result.put("success", true);
    session.setAttribute("currentUser", user);
    ResponseUtil.write(ServletActionContext.getResponse(), result);
    return null;
}
```

4.1.7 订单管理功能实现

点击订单管理效果图：



订单	航班	订购日期	订购数量	票价	总价	操作
NO20151110024626	大连航空003国际航班 上海 - 东京 2015-11-10 15:30 - 18:00	2015-11-10	1	1860	1860	退票 改签
NO20151110024439	大连航空003国际航班 上海 - 东京 2015-11-10 15:30 - 18:00	2015-11-10	1	3000	3000	退票 改签
NO20151110024337	大连航空001 大连 - 广州 2015-11-10 17:58 - 17:58	2015-11-10	1	800	800	退票 改签
NO20151110023520	南方航空001 连云港 - 新加坡 2015-11-10	2015-11-10	8	2000	16000	退票 改签

用户可以进行“退票”，“改签操作”；

点击“确定”，则完成“退票”，点击“取消”，不执行；

点击“改签”，如图：



弹出航班信息查询界面，用户可以搜索自己想要改签的航班，然后点击确定，完成“改签”关键代码实现：

获取当前用户的机票订单信息：

```
public List<TicketOrder> findTicketOrderList(TicketOrder s_ticketOrder,
        PageBean pageBean) {
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from TicketOrder");
    if(s_ticketOrder!=null){
        if(s_ticketOrder.getUser().getId()!=null){
            hql.append(" and user.id="+s_ticketOrder.getUser().getId());
        }
        if(s_ticketOrder.getUser()!=null &&
StringUtil.isNotEmpty(s_ticketOrder.getUser().getUserName())){
            hql.append("          and          user.userName          like
"+" "+s_ticketOrder.getUser().getUserName()+"%");
        }
        if(StringUtil.isNotEmpty(s_ticketOrder.getOrderNo())){
```

```
public void deleteTicketOrder(Integer id) {
    TicketOrder ticketOrder=baseDAO.get(TicketOrder.class, id);
    baseDAO.delete(ticketOrder);
    Flight flight=baseDAO.get(Flight.class, ticketOrder.getFlight().getId());
    if("经济舱".equals(ticketOrder.getSpaceType())){
        flight.setEcTicketRemain(flight.getEcTicketRemain()+ticketOrder.getNum());
    }else{
        flight.setFcTicketRemain(flight.getFcTicketRemain()+ticketOrder.getNum());
    }
}
```

改签功能实现：

```
public void saveMeal(Integer id, Integer newFlightId, User currentUser) throws Exception {
    Flight newFlight=baseDAO2.get(Flight.class, newFlightId); // 新的航班
    TicketOrder ticketOrder=baseDAO.get(TicketOrder.class, id); // 原先的机票订购
    实体
    Flight oldFlight=ticketOrder.getFlight(); // 原先的航班
    TicketOrder newTicketOrder=new TicketOrder();
    newTicketOrder.setOrderNo("NO"+DateUtil.getCurrentDateStr());
    newTicketOrder.setOrderTime(new Date());
    newTicketOrder.setNum(ticketOrder.getNum());
    newTicketOrder.setSpaceType(ticketOrder.getSpaceType());
    if("经济舱".equals(ticketOrder.getSpaceType())){
        newTicketOrder.setPrice(newFlight.getEcPrice());

        newTicketOrder.setTotalPrice(newFlight.getEcPrice()*newTicketOrder.getNum());

        oldFlight.setEcTicketRemain(oldFlight.getEcTicketRemain()+newTicketOrder.getNum())
        ; // 原先的航班要加上机票

        newFlight.setEcTicketRemain(newFlight.getEcTicketRemain()-newTicketOrder.getNum(
        )); // 新的航班要剪掉机票
    }else{
        newTicketOrder.setPrice(newFlight.getFcPrice());

        newTicketOrder.setTotalPrice(newFlight.getFcPrice()*newTicketOrder.getNum());

        oldFlight.setFcTicketRemain(oldFlight.getFcTicketRemain()+newTicketOrder.getNum())
        ; // 原先的航班要加上机票

        newFlight.setFcTicketRemain(newFlight.getFcTicketRemain()-newTicketOrder.getNum(
        )); // 新的航班要剪掉机票
    }
    newTicketOrder.setFlight(newFlight);
    newTicketOrder.setUser(currentUser);

    baseDAO.save(newTicketOrder); // 保存新的订单
    baseDAO2.save(oldFlight); // 保存原先的航班信息
    baseDAO2.save(newFlight); // 保存新的航班信息

    baseDAO.delete(ticketOrder); // 删除原先的订单
```

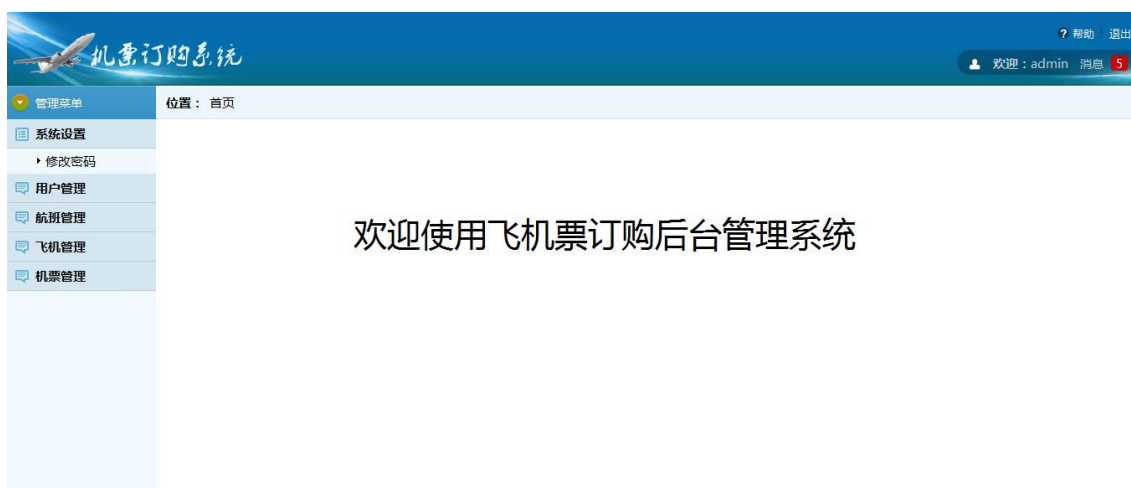
4.2 后台系统设计实现

4.2.1 管理员登录功能

管理员在浏览器地址栏输入：<http://localhost:8080/TicketOrderSys/admin/login.jsp>
则进入后台管理登录页面，如图：



管理员输入正确的用户名和密码，则进去系统主界面；如图：



关键代码:

用户登录:

```
/**
 * 用户登录
 * @return
 * @throws Exception
 */
public String login()throws Exception{
    HttpSession session=request.getSession();
    User currentUser=userService.login(user);
    boolean flag=false;
    if(currentUser==null){
        flag=false;
    }else{
        session.setAttribute("currentUser", currentUser);
        flag=true;
    }
    JSONObject result=new JSONObject();
    result.put("success", flag);
    ResponseUtil.write(ServletActionContext.getResponse(), result);
    return null;
}
```


4.2.2 用户管理

点击“管理用户”菜单按钮，显示：

说明：

用户管理模块包括用户信息的查询，用户添加，用户修改，用户删除；



说明：

进入添加用户界面，输入用户信息，点击“保存”按钮。即可完成保存功能；

关键代码实现：

分页查询获取所有用户信息；

```
public List<User> findUserList(User s_user, PageBean pageBean) {
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from User");
    if(s_user!=null){
        if(StringUtil.isEmpty(s_user.getUserName())){
            hql.append(" and userName like ? ");
            param.add("%"+s_user.getUserName()+"%");
        }
    }
    if(pageBean!=null){
        return baseDAO.find(hql.toString().replaceFirst("and", "where"),
        param, pageBean);
    }else{
        return null;
    }
}
```


添加或者修改用户信息：

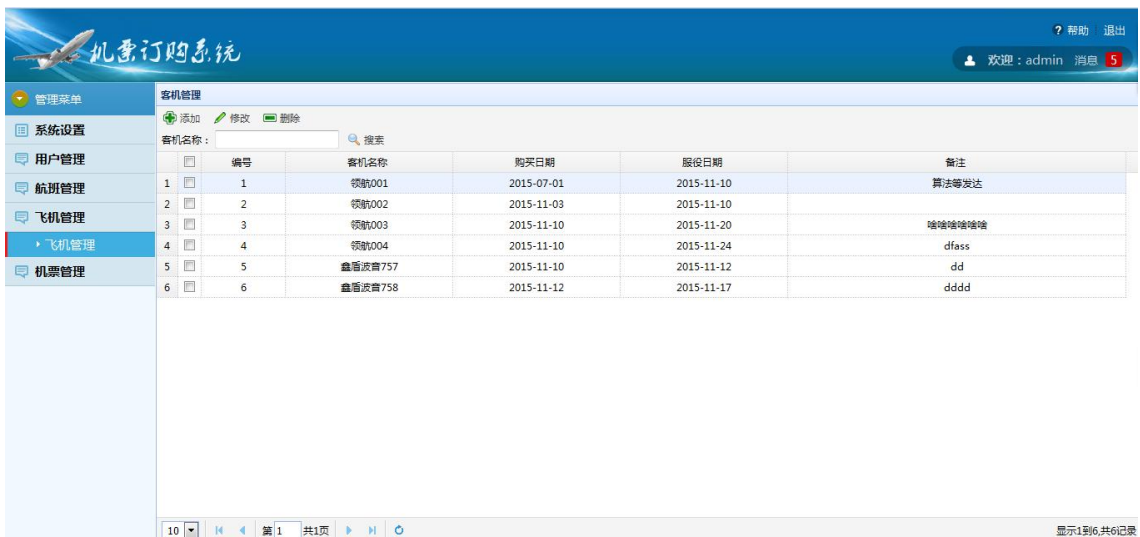
```
public void saveUser(User user) {
    baseDAO.merge(user);
}
```

删除用户信息：

```
public void delete(User user) {
    baseDAO.delete(user);
}
```

4.2.3 客机管理

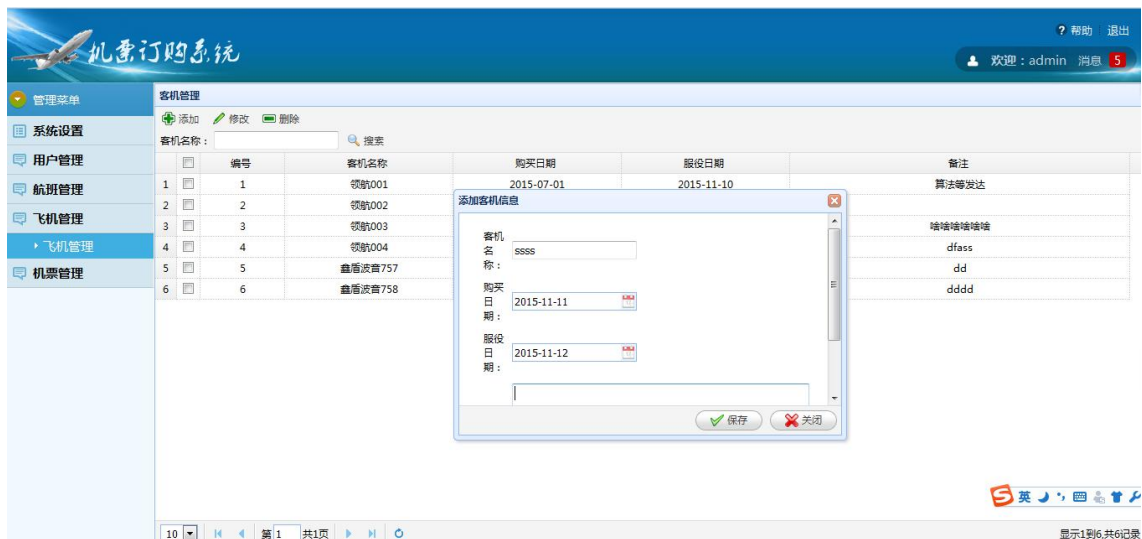
点击“客机管理”菜单按钮，显示：



编号	客机名称	购买日期	服役日期	备注
1	领航001	2015-07-01	2015-11-10	算法等发达
2	领航002	2015-11-03	2015-11-10	
3	领航003	2015-11-10	2015-11-20	嗡嗡嗡嗡嗡嗡
4	领航004	2015-11-10	2015-11-24	dfass
5	鑫盾波音757	2015-11-10	2015-11-12	ddd
6	鑫盾波音758	2015-11-12	2015-11-17	dddd

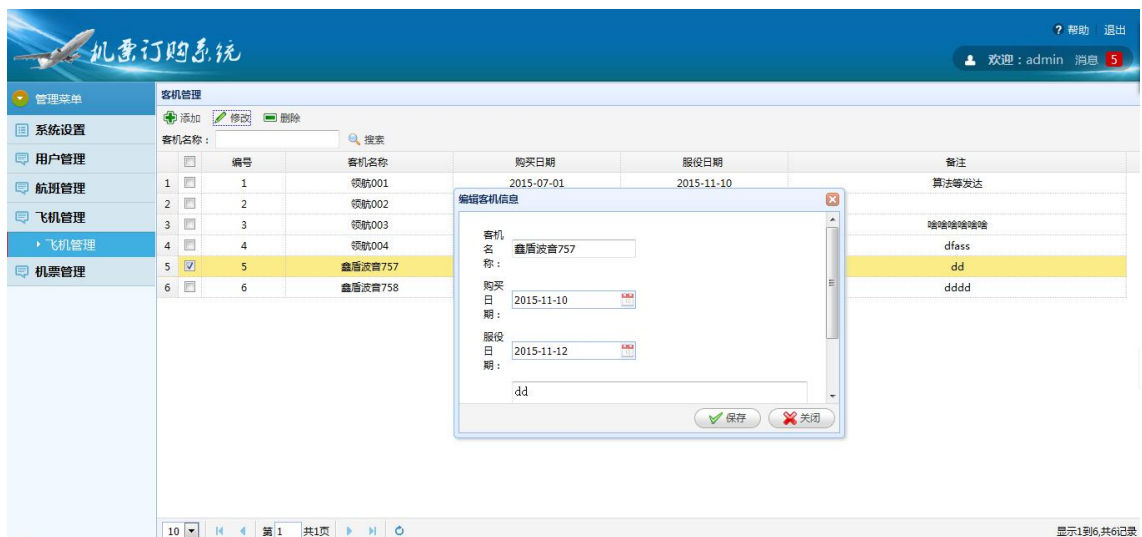
说明：客机管理模块里，有客机分页显示功能，按照客机名称查询功能，以及客机添加，客机修改，客机删除；

点击“添加按钮”，效果如图：



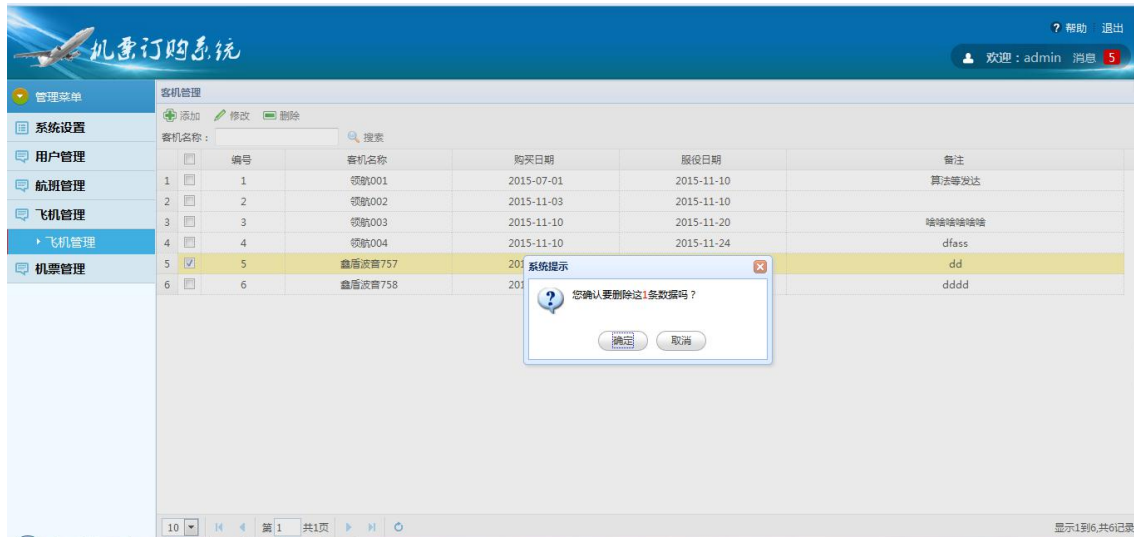
说明：进入客机添加页面，输入客机信息，点击“保存”，即可完成客机添加功能；

点击“修改”按钮，效果如图：



说明：进入客机修改界面，根据需要，修改指定字段，点击“保存”按钮，即可完成保存功能；

选中需要删除的客机，点击“删除”按钮，效果如图：



说明：点击“确定”按钮，即可删除客机；

关键代码实现：

分页查询客机信息功能：

```
public List<Aircraft> findAircraftList(Aircraft s_aircraft,
    PageBean pageBean) {
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from Aircraft");
    if(s_aircraft!=null){
        if(StringUtil.isEmpty(s_aircraft.getName())){
            hql.append(" and name like ? ");
            param.add("%"+s_aircraft.getName()+"%");
        }
    }
    if(pageBean!=null){
        return baseDAO.find(hql.toString().replaceFirst("and", "where"),
            param, pageBean);
    }else{
        return baseDAO.find(hql.toString().replaceFirst("and", "where"),
            param);
    }
}
```

添加商品和修改客机功能：

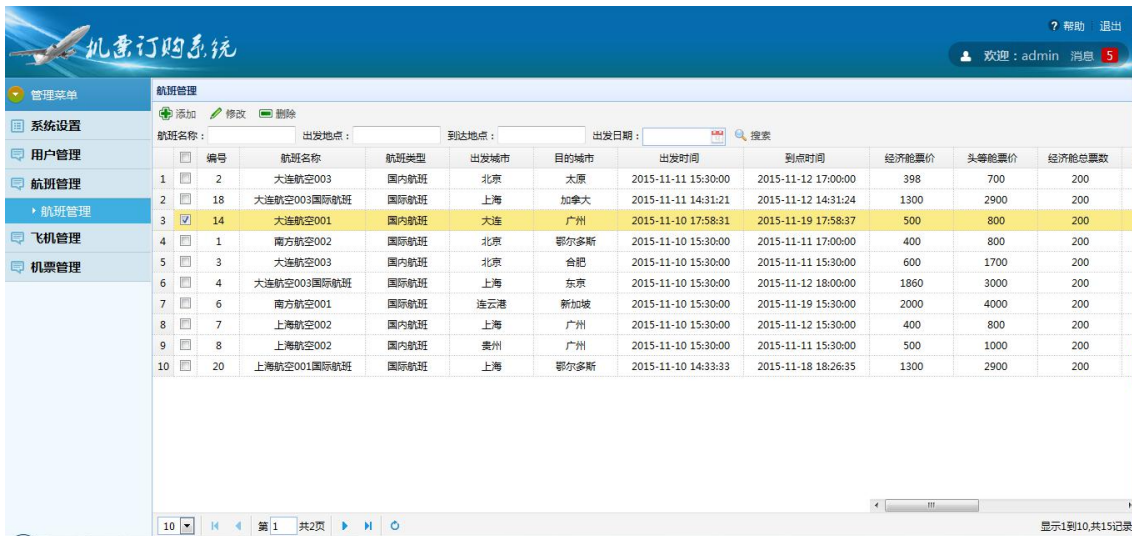
```
public void saveAircraft(Aircraft aircraft) {
    baseDAO.merge(aircraft);
}
```

删除客机功能:

```
public void delete(Aircraft aircraft) {
    baseDAO.delete(aircraft);
}
```

4.2.4 航班管理

点击菜单栏中的“航班管理”，显示：

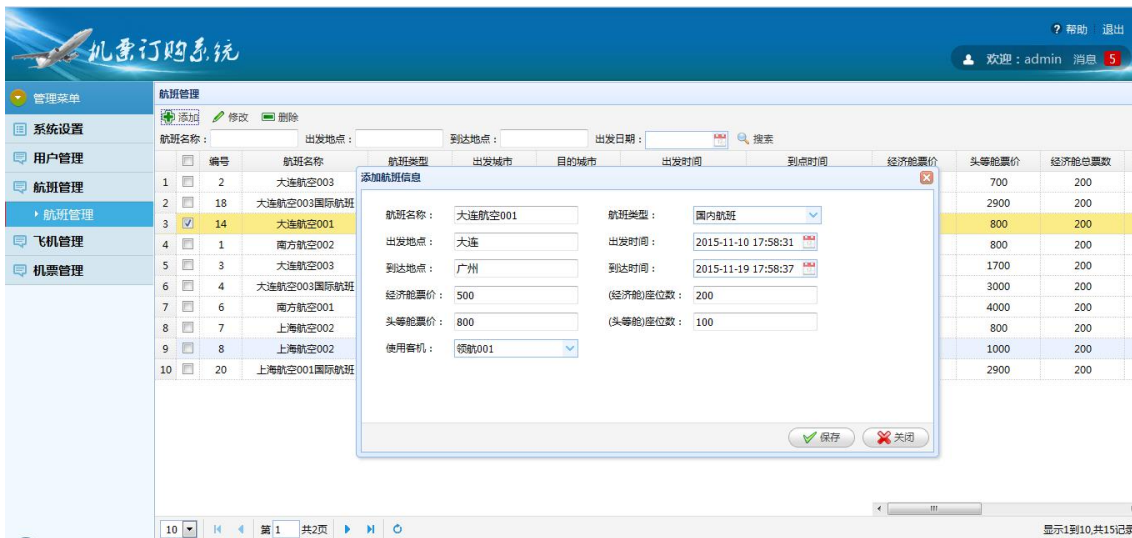


The screenshot shows the '航班管理' (Flight Management) page. It includes a sidebar with navigation options like '管理菜单', '系统设置', '用户管理', '航班管理', '飞机管理', and '机票管理'. The main area displays a table of flight information with columns for flight number, name, type, origin, destination, departure time, arrival time, and pricing. A search bar is at the top right, and a pagination bar is at the bottom.

航班名称	编号	航班名称	航班类型	出发城市	目的城市	出发时间	到达时间	经济舱票价	头等舱票价	经济舱总票数
大连航空003	2	大连航空003	国内航班	北京	太原	2015-11-11 15:30:00	2015-11-12 17:00:00	398	700	200
大连航空003国际航班	18	大连航空003国际航班	国际航班	上海	加拿大	2015-11-11 14:31:21	2015-11-12 14:31:24	1300	2900	200
大连航空001	14	大连航空001	国内航班	大连	广州	2015-11-10 17:58:31	2015-11-19 17:58:37	500	800	200
南方航空002	1	南方航空002	国际航班	北京	鄂尔多斯	2015-11-10 15:30:00	2015-11-11 17:00:00	400	800	200
大连航空003	3	大连航空003	国内航班	北京	合肥	2015-11-10 15:30:00	2015-11-11 15:30:00	600	1700	200
大连航空003国际航班	4	大连航空003国际航班	国际航班	上海	东京	2015-11-10 15:30:00	2015-11-12 18:00:00	1860	3000	200
南方航空001	6	南方航空001	国际航班	连云港	新加坡	2015-11-10 15:30:00	2015-11-19 15:30:00	2000	4000	200
上海航空002	7	上海航空002	国内航班	上海	广州	2015-11-10 15:30:00	2015-11-12 15:30:00	400	800	200
上海航空002	8	上海航空002	国内航班	贵州	广州	2015-11-10 15:30:00	2015-11-11 15:30:00	500	1000	200
上海航空001国际航班	20	上海航空001国际航班	国际航班	上海	鄂尔多斯	2015-11-10 14:33:33	2015-11-18 18:26:35	1300	2900	200

说明：进入航班管理页面，管理员可以查询航班，添加航班，修改航班，删除航班。

点击“添加”按钮，显示：



The screenshot shows the '航班管理' (Flight Management) page with the '添加航班信息' (Add Flight Information) dialog box open. The dialog box contains fields for flight name, type, origin, destination, departure time, arrival time, and pricing. The '保存' (Save) button is highlighted.

航班名称	编号	航班名称	航班类型	出发城市	目的城市	出发时间	到达时间	经济舱票价	头等舱票价	经济舱总票数
大连航空003	2	大连航空003	国内航班	北京	太原	2015-11-11 15:30:00	2015-11-12 17:00:00	398	700	200
大连航空003国际航班	18	大连航空003国际航班	国际航班	上海	加拿大	2015-11-11 14:31:21	2015-11-12 14:31:24	1300	2900	200
大连航空001	14	大连航空001	国内航班	大连	广州	2015-11-10 17:58:31	2015-11-19 17:58:37	500	800	200
南方航空002	1	南方航空002	国际航班	北京	鄂尔多斯	2015-11-10 15:30:00	2015-11-11 17:00:00	400	800	200
大连航空003	3	大连航空003	国内航班	北京	合肥	2015-11-10 15:30:00	2015-11-11 15:30:00	600	1700	200
大连航空003国际航班	4	大连航空003国际航班	国际航班	上海	东京	2015-11-10 15:30:00	2015-11-12 18:00:00	1860	3000	200
南方航空001	6	南方航空001	国际航班	连云港	新加坡	2015-11-10 15:30:00	2015-11-19 15:30:00	2000	4000	200
上海航空002	7	上海航空002	国内航班	上海	广州	2015-11-10 15:30:00	2015-11-12 15:30:00	400	800	200
上海航空002	8	上海航空002	国内航班	贵州	广州	2015-11-10 15:30:00	2015-11-11 15:30:00	500	1000	200
上海航空001国际航班	20	上海航空001国际航班	国际航班	上海	鄂尔多斯	2015-11-10 14:33:33	2015-11-18 18:26:35	1300	2900	200

说明：进入航班添加页面，输入信息，点击“保存”按钮，即可完成航班的添加功能；

说明：进入航班修改界面，修改信息，点击“保存”按钮，即可完成航班修改功能；
选择“确定”按钮，即可删除数据，完成商品大类删除功能；
关键代码实现：

分页查询所有航班信息：

```
public List<Flight> findFlightList(Flight s_flight,PageBean pageBean){
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from Flight");
    if(s_flight!=null){
        if(StringUtil.isNotEmpty(s_flight.getName())){
            hql.append(" and name like '%" +s_flight.getName()+"%'");
        }
        if(StringUtil.isNotEmpty(s_flight.getFlightType())){
            hql.append(" and flightType='"+s_flight.getFlightType()+"'");
        }
        if(StringUtil.isNotEmpty(s_flight.getFromCity())){
            hql.append(" and fromCity='"+s_flight.getFromCity()+"'");
        }
        if(StringUtil.isNotEmpty(s_flight.getToCity())){
            hql.append(" and toCity='"+s_flight.getToCity()+"'");
        }
        if(s_flight.getFromTime()!=null){
            hql.append("                and                fromTime
<" +DateUtil.formatDate(s_flight.getFromTime(), "yyyy-MM-dd")+"'");
        }
    }
    hql.append(" order by fromTime desc");
    if(pageBean!=null){
        return baseDAO.find(hql.toString().replaceFirst("and", "where"),
param, pageBean);
    }else{
        return baseDAO.find(hql.toString().replaceFirst("and", "where"),
param);
    }
}
```

添加和修改航班功能：

```
public void saveFlight(Flight flight) {
    baseDAO.merge(flight);
}
```

删除指定航班:

```
public void delete(Flight flight) {
    baseDAO.delete(flight);
}
```

4.2.5 机票管理

点击菜单栏“机票管理”，显示:

机票订购系统

?

帮助

退出

+

管理菜单

■

系统设置

✈

航班管理

✈

飞机管理

✈

机票管理

➤

机票订单管理

✕

删除订单

订单号:

用户名:

搜索

	<input type="checkbox"/>	编号	订单号	订购时间	航班名称	用户名	座舱类型	票价	订购数量	总价
1	<input type="checkbox"/>	34	NO20151110024439	2015-11-10 14:45:07	大连航空003国际航班	111	头等舱	3000	1	3000
2	<input type="checkbox"/>	33	NO20151110024337	2015-11-10 14:43:37	大连航空001	111	头等舱	800	1	800
3	<input type="checkbox"/>	32	NO20151110023934	2015-11-10 14:39:34	上海航空002	222	经济舱	400	1	400
4	<input type="checkbox"/>	31	NO20151110023905	2015-11-10 14:39:05	大连航空003	222	经济舱	600	1	600
5	<input type="checkbox"/>	30	NO20151110023857	2015-11-10 14:38:57	上海航空002	222	经济舱	400	1	400
6	<input type="checkbox"/>	29	NO20151110023849	2015-11-10 14:38:49	南方航空001	222	头等舱	4000	1	4000
7	<input type="checkbox"/>	28	NO20151110023831	2015-11-10 14:38:31	大连航空003国际航班	222	头等舱	3000	1	3000
8	<input type="checkbox"/>	27	NO20151110023520	2015-11-10 14:35:20	南方航空001	111	经济舱	2000	8	16000
9	<input type="checkbox"/>	26	NO20151110023502	2015-11-10 14:35:02	南方航空001	111	头等舱	4000	1	4000
10	<input type="checkbox"/>	25	NO20151110023450	2015-11-10 14:34:50	南方航空002	111	经济舱	400	1	400

10

1

共2页

显示1到10,共11条

说明：进入机票管理页面，可以查询机票，删除机票；
点击“确定”按钮，即可删除数据，完成删除功能；

关键代码实现:

分页查询机票信息:

```
public List<TicketOrder> findTicketOrderList(TicketOrder s_ticketOrder,
        PageBean pageBean) {
    List<Object> param=new LinkedList<Object>();
    StringBuffer hql=new StringBuffer("from TicketOrder");
    if(s_ticketOrder!=null){
        if(s_ticketOrder.getUser().getId()!=null){
            hql.append(" and user.id="+s_ticketOrder.getUser().getId());
        }
        if(s_ticketOrder.getUser()!=null
            StringUtil.isEmpty(s_ticketOrder.getUser().getUserName())){
            &&
```

删除机票信息:

```
public void deleteTicketOrder(Integer id) {
    TicketOrder ticketOrder=baseDAO.get(TicketOrder.class, id);
    baseDAO.delete(ticketOrder);
    Flight flight=baseDAO2.get(Flight.class,
ticketOrder.getFlight().getId());
    if("经济舱".equals(ticketOrder.getSpaceType())){

        flight.setEcTicketRemain(flight.getEcTicketRemain()+ticketOrder.getNum()
);
    }else{

        flight.setFcTicketRemain(flight.getFcTicketRemain()+ticketOrder.getNum()
);
    }
}
```

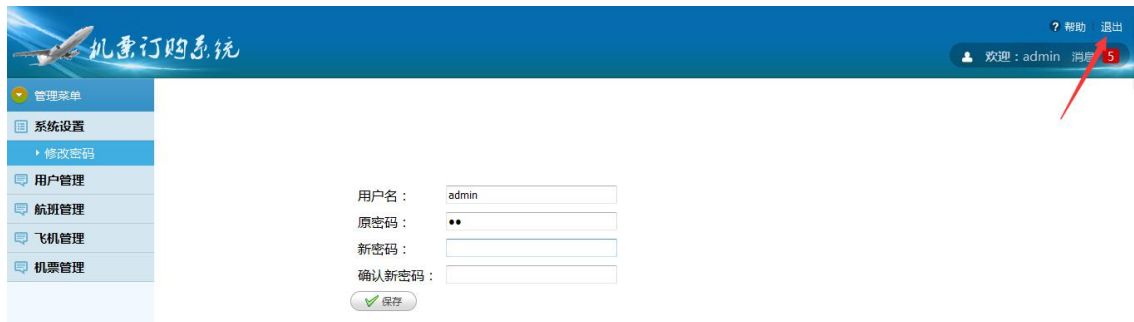
4.2.11 系统管理

切换到系统管理，点击“修改密码”，如图：



说明：在系统管理下，有“修改密码功能；

点击“安全退出”，如图：



说明：点击“确定”，即可完成系统的退出；

关键代码实现：

修改密码功能实现：

```
@Override
public void saveUser(User user) {
    // TODO Auto-generated method stub
    baseDAO.merge(user);
}
```

安全退出功能实现：

```
/**
 * 注销用户
 * @throws Exception
 */
public String logout()throws Exception{
    request.getSession().invalidate();
    return "logout";
}
```

第 5 章 系统的调试与测试

5.1 软件测试的目的

软件测试的目的：1) 看软件是否能够按照正确的方法，正确的运行实现我们想要的结果。2) 测试也是为我们自己获得一些条件信息，有利于我们对软件作出评测。3) 测试是贯穿整个的开发过程的，假如软件开发完后出现了好多问题这说明软是有缺陷的所以，测试也是为软件开发过程的质量保驾护航。

衡量软件质量的几个标准：

按着时间的规划，在相应的时间段，用正确的方法完成相应所要求的任务。

严格记住一些应用性的标准要求，像一些的操作习惯，一些维护性，测试性硬性要求。

虽然经过我们的努力软件能过做我们想要的功能，但是代码的完美也并不能说明软件完成的非常完美。

客户需要才是软件的一切，不能从客户要求出发软件也不能算完美，我们最重要的是看客户需要什么的产品，怎样使用，还有最重要的是客户在使用过程中能够遇到哪些问题。

5.2 软件测试的任务

- 1、寻找 Bug。
- 2、避免软件开发过程中的缺陷。
- 3、衡量软件的品质。
- 5、关注用户的需求。

总的目标是：确保软件的质量。

5.3 测试环境

Apache 提供的 Web 服务器 Tomcat 6.0 使得用户可以在 Windows 2007 以上操作系统下发布自己设计的动态网页。所以说系统的调试是在 Tomcat6.0 这个大环境下进行并且完成的。我们可以创建系统的一个站点。Ip 设定为 192.168.2.3 它的缺省域名值是 admin。然后将 URL 输入 IE web 能够收到 URL 之后找到网页文件。

5.4 白盒、黑盒测试

白盒测试技术(White Box Testing)测试的能够从程序内部进行,开发人员能够凭自己经验和对程序的熟知,对应该重新改编的代码进行重新改编,编译员依据自己的理解感知,以及对代码操作进行的测试叫做白盒测试。检测主要是程序编译员为主,能够以自己的想法逻辑判定来对程序检测。

黑盒测试(Black Box Testing)产品应该能够独立完成的事情,以及牵涉的方面都知道。通过检测来检验程序的各个工道是不是能够健全的运行。

5.5 手工测试与自动测试

手工测试(Manual Testing):主要是人工的进行检测,可以借鉴黑盒白盒测试,编译员根据自己能力和自己经验对程序进行的检测。

自动测试(Automation Testing)是自主的的检查根据,有名的检测经验。相对来说手工监测还是比较常用的。自测比较笨拙,没有很好的灵活性能。但是手测有非常多的工序,工作量比较大。由于测试环境操作系统,网络环境,带宽等情况都有可能对测试结果有一定的影响,这时就需要长期积累的经验来判断,在测试环境的保护方面进行一些努力。

5.6 软件缺陷

软件缺陷是在测试过程中出现的对系统有一定影响的,但是在设计中没有出现的或者对修改后的 bug 测试和开发人员存在着不同的想法等使所编写的软件未达到产品说明书标明的功能。产品说明书指明不会出现错误但是在编写的软件中出现了;软件所实现的功能超出产品说明书的范围;软件没有实现产品说明书虽未指出但应实现的功能目标;软件测试员认为软件非常难去理解、使用起来也不方便、运行速度非常慢,或者用户评价低不适用,认为不好。

结束语

经过一段时间的努力，毕业设计终于一段落。本文是关于商城系统的文稿设计，通过阅读本论文可以对本系统有一定的了解。本文介绍了系统开发所用到的技术，编程语言，卡法环境等。同时在做了分析调查后，对本系统做了详细的需求分析。然后对系统的概要设计以及详细设计也做了详细的说明，包括系统的前台设计，数据逻辑层设计以及后台数据库设计。

系统的主要特点有：人性化设计，界面友好，简洁，大方，操作简单方便，权限操作

虽然系统设计已经暂告一个段落，但由于本人的能力有限，再加上知识欠缺，以及对的业务不甚熟悉，所以在很多方面还存在着不足，如在系统需求分析方面还有很多方面没有考虑到，而且在具体实现各个功能的时候也存在一定的局限性。比如测试功能不完善，用户界面功能太简单、系统的界面不是太美观，系统安全性考虑不充分等等。所有本系统还有很多地方有待完善。

参考文献

1. 王海涛, 贾宗璞. 基于 Struts 和 Hibernate 的 Web 应用开发[J]. 计算机工程, 2011, 37(9): 113.
2. 傅鹏, 殷旻昊. 基于 Struts+Spring+Hibernate 技术的科研管理系统设计[J]. 软件导刊, 2009, 8(1): 135-136.
3. 龚瑜江, 红黄永. 基于 Struts 的 Web 开发 [J]. 计算机与数字工程, 2009, 37(232) : 58-62.
4. 李峰, 刘彦隆. 基于 SSH 框架与 jquery 技术的 Java Web 开发应用[J]. 科技情报开发与经济, 2010, 20(6): 106-109.
5. 赵洋, 张丽, 王恩东, 张素宁. 基于 Struts+Spring+Hibernate 的 J2EE 的架构
6. 梁胜彬, 乔保军. Java 程序设计实例教程. 清华大学出版社, 2012.8.
7. 李建宏, 李广振. Web 编程基础. 北京大学出版社, 2007.12.
8. 王珊, 萨师煊. 数据库系统概论(第四版). 高等教育出版社, 2006.5.
9. 杨小平. 《Java 项目案例导航》. 第 1 版. 科学出版社, 2003.
10. 唐爱国, 杭志, 王建明. 《Web 2.0 动态网站开发—JSP 技术详解与应用实践》. 第 1 版. 清华大学出版社, 2009.