**摘要**

随着因特网的发展，航空客运业务规模的日益宏大，传统售票方式已经不能满足时代的要求。

本系统——航空售票系统，采用ASP.NET 技术作为创建应用程序的方式，以C＃作为编程语言，采用SQL Server 数据库作为后台数据库。它是基于Browser/Server结构下形成的数据驱动的Web应用程序。不仅能方便旅客，还能节省航空公司的代理费用，加速资金周转。

本系统基本遵循了软件工程方法论进行系统分析、总体设计、详细设计和软件测试。

实现了后台管理员对航空公司、航班和订票信息的维护；前台客户通过注册登陆可以查询航班信息，进行订票和查看自己的购票信息并可以删除管理员未确认的订票信息。同时可以进行购票错误以后的退票信息等功能。

关键词  ASP.NET 、 C#  、SQL Server2000 、 VS.NET 2008 、

电子商务

目录

[1系统概述 4](#_Toc23618)

[1.1系统目标 4](#_Toc16688)

[1.2系统主要功能 5](#_Toc7896)

[2可行性分析 5](#_Toc13750)

[2.1.1实用性原则 5](#_Toc8716)

[2.1.2可维护性原则 6](#_Toc18665)

[2.1.3可扩展性 6](#_Toc21667)

[3系统分析 6](#_Toc9398)

[3.1组织结构分析 6](#_Toc15910)

[3.2业务流程分析 7](#_Toc27273)

[3.3登陆界面时的流程图 8](#_Toc31289)

[3.4子系统各个实现流程图 8](#_Toc20435)

[3.3数据字典 11](#_Toc14195)

[4代码设计 12](#_Toc9206)

[4.1系统所需代码 12](#_Toc7736)

[4.2代码结构设计 14](#_Toc12866)

[5数据库设计 19](#_Toc26405)

[5.1数据库结构设计 19](#_Toc18729)

[5.2数据表关系设计 20](#_Toc26120)

[6输入设计 21](#_Toc13698)

[6.1输入界面设计 21](#_Toc24031)

[2，用户操作界面 22](#_Toc20670)

[3，管理员操作界面 24](#_Toc1134)

[6.2输入设备选择 26](#_Toc5095)

[6.3菜单设计 26](#_Toc14225)

[7输出设计 29](#_Toc8512)

[7.1系统输出内容设计 29](#_Toc17753)

[7.2输出界面设计 29](#_Toc22541)

[参考文献 32](#_Toc31655)

# 1系统概述

## 1.1系统目标

当今世界，以信息技术为主要标志的科技进步日新月异，高科技成果向现实生产力的转化越来越快。纵观全球经济发展，信息技术和信息产业已经成为经济增长的主要推动力之一，正在改变着传统的生产和经营方式以至生活方式，发达国家经过产业结构的升级和经济结构的转型已进入信息经济阶段。信息资源已经成为国民经济和社会发展的战略资源，信息化水平也已成为现代水平和综合国力的重要标志。今年是“十五”计划开局之年，中共十五届五中全会通过的国民经济和社会发展第十个五年计划建议中已明确指出：“信息化是当今世界经济和社会发展的大趋势，也是我国产业优化升级和实现工业化、现代化的关键环节。”“大力推进国民经济和社会信息化，是覆盖现代化建设全局的战略举措。”，可见，党和国家已将国民经济和社会信息化放在优先发展位置，体现了先进生产力的客观要求，是一项重要的战略决策。这是民航加快发展的机遇，更是民航信息化的难得机遇。

随着知识经济的到来，人类已经逐步进入信息化社会，信息增长的速度越来越快，人们希望利用先进的管理理论方法手段来得到并处理越来越多的信息，以提高工作效率和管理水平。由于信息资源对人们生活的重要性，不断提高信息的收集，传输，加以利用等活动，日益成为人们社会生活的重要组成部分。网上机票预订管理系统的产生和发展正好满足人们的这种需求。现在将详细介绍我的课程设计——机票预订管理系统。

本系统在前台采用了开发工具java程序语言编写，后台数据库使用了mysql数据库，建立在windows作业系统上，但适合各种安装了java的jre虚拟机的操作系统。

## 1.2系统主要功能

本机票预订管理系统：

用户：可以查询所有航班，查询指定航班的信息，订票，退票，

取票，用户指南，关于我们。

管理员：查询航班信息，修改航班信息，删除用户信息，查询订

票信息，用户指南，关于我们。

# 2可行性分析

## 2.1.1实用性原则

首先符合实际需要，机票预订管理系统的业务是繁多而复杂的，一般情况是程序员对这些业务的认知程度较低，为此，要做到实用性并不简单。因此，机票预订管理系统的实用性不仅仅是一个设计的问题，还要靠软件的不断升级来实现。

## 2.1.2可维护性原则

系统运行后，维护工作是一个长期的工作。因此本系统充分考虑维护工作的需求：设计采用面向对象、模块化技术，提高软件的复用性、界面的一致性、代码的重用性，尽量降低维护工作量及困难

## 2.1.3可扩展性

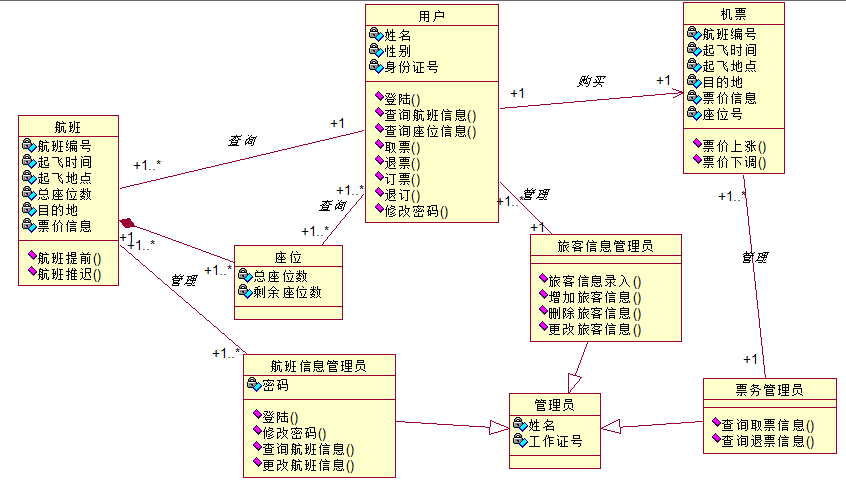
系统应具有较好的可扩展性。系统的可扩展性应包括能接纳已有的系统和在今后系统软硬件扩展时，能有效地保护已有的资源。特别是在应用需求变化时，有一个较好的平台，能容易的加以调整。

# 3系统分析

# 

## 3.1组织结构分析

各个类的类图



## 3.2业务流程分析

注册

用户

管理员

用户/管理员

查询订票信息

订票

所有航班

删除用户信息

修改航班信息

查询航班信息

关于我们

系统指南

退票

取票

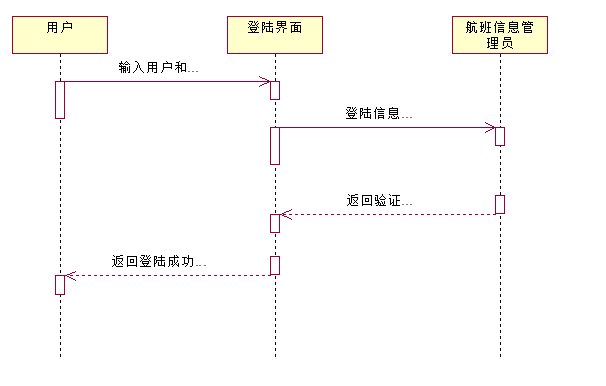
查询航班

## 3.3登陆界面时的流程图

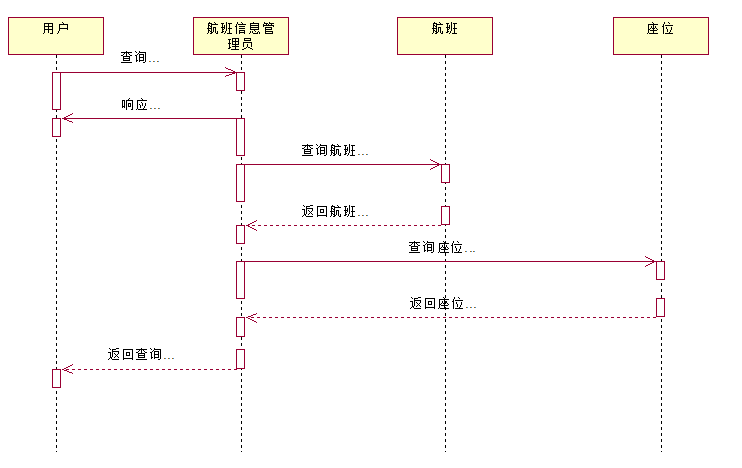


## 3.4子系统各个实现流程图

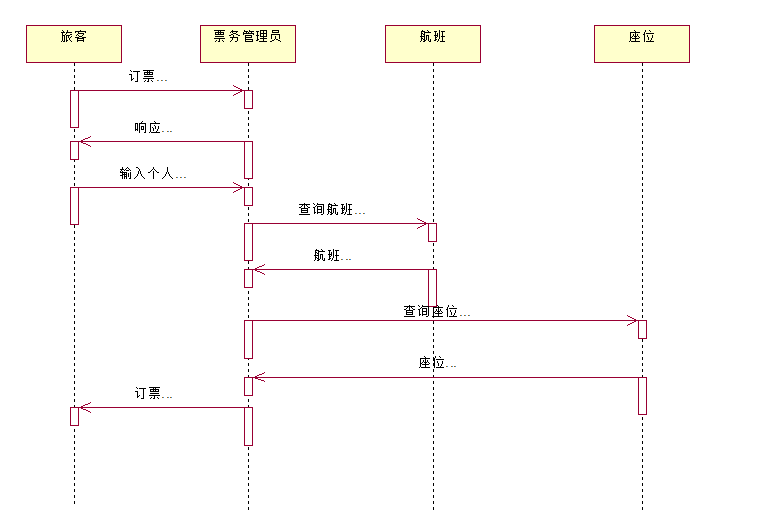
1、登陆流程图



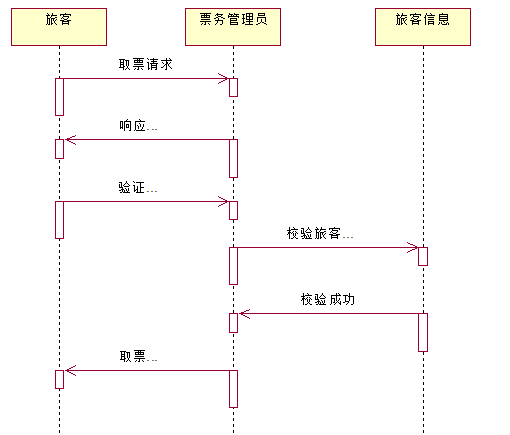
2、查询航班信息流程图



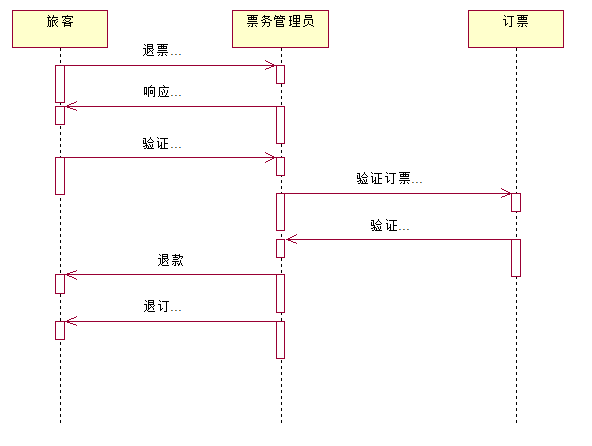
3，订票顺序图



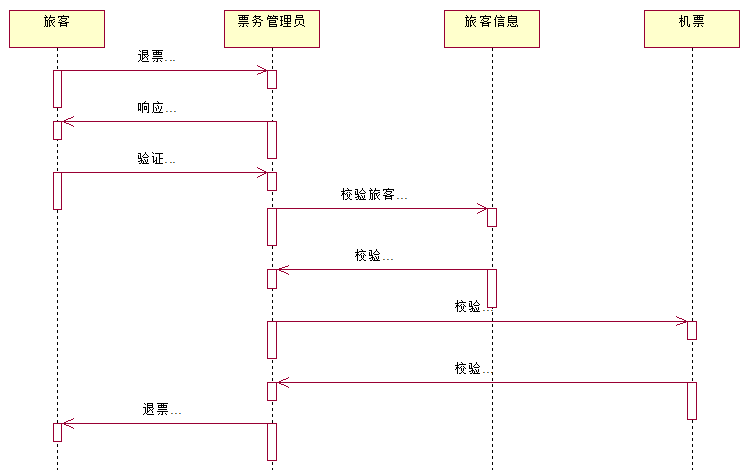
3、取票顺序图



1. 退订顺序图



1. 退票顺序图



## 3.3数据字典

用户信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | UserNsme | varchar | 20 |  |  |  | 否 |  |  |
| 2 | UserId | varchar | 20 |  | 是 | 是 | 否 |  |  |
| 3 | UserPass | int | 10 |  |  |  | 否 |  |  |
| 4 | UserIdentify | varchar | 20 |  |  |  | 否 |  |  |

管理员信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | MangerNsme | varchar | 20 |  |  |  | 否 |  |  |
| 2 | MangerId | varchar | 20 |  | 是 | 是 | 否 |  |  |
| 3 | MangerPass | int | 10 |  |  |  | 否 |  |  |
| 4 | MangerIdentify | varchar | 20 |  |  |  | 否 |  |  |

订票信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | UserId | varchar | 20 |  | 是 | 是 | 否 |  |  |
| 2 | FlightId | varchar | 20 |  | 是 | 是 | 否 |  |  |
| 3 | State | varchar | 10 |  |  |  | 否 |  |  |

航班信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | FlightId | varchar | 20 |  |  |  | 否 |  |  |
| 2 | StartTime | varchar | 20 |  | 是 | 是 | 否 |  |  |
| 3 | StatePoint | varchar | 10 |  |  |  | 否 |  |  |
| 4 | Dedtination | varchar | 20 |  |  |  | 否 |  |  |
| 5 | Remainseat | int | 10 |  |  |  | 否 |  |  |
| 6 | Price | int | 20 |  |  |  | 否 |  |  |

# 4代码设计

## 4.1系统所需代码

管理员数据库连接：

package Link;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import Person.Manger;

public class MangerLink {

static Connection conn = null;

static PreparedStatement ps = null;

static ResultSet rs = null;

public static boolean add(Manger manger) {

try {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

conn = DriverManager.getConnection(

"jdbc:sqlserver://localhost:1433;DatabaseName=TicketBook",

"sa", "123456");

ps = conn

.prepareStatement("insert into MangerTable(MangerName,MangerId,MangerPass,MangerIdentify) values(?,?,?,?)");

ps.setString(1, manger.getMangerName());

ps.setString(2, manger.getMangerId());

ps.setString(3, manger.getMangerPass());

ps.setString(4, manger.getMangerIdentify());

ps.execute();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

ps.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return false;

}

public static boolean find(String name, String pass, String identify) {

try {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

conn = DriverManager.getConnection(

"jdbc:sqlserver://localhost:1433;DatabaseName=TicketBook",

"sa", "123456");

ps = conn.prepareStatement("select \* from MangerTable where MangerName=?");

ps.setString(1, name);

rs = ps.executeQuery();

while (rs.next()) {

String pass1 = rs.getString("MangerPass");

System.out.println(pass1);

String identify2 = rs.getString("MangerIdentify");

if (pass1.equals(pass) && identify.equals(identify2.trim())) {

return true;

}

}

} catch (Exception e) {

e.printStackTrace();

} finally {

if (rs != null) {

try {

rs.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

try {

ps.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return false;

}

}

管理员界面代码：

package Frame;

import javax.swing.\*;

import javax.swing.event.\*;

import java.awt.\*;

import java.awt.event.\*;

public class MangerJieMian extends JFrame implements ActionListener{

JMenuItem jtm1=null;

JMenuItem jtm2=null;

// JMenuItem jtm3=null;

JMenuItem jtm4=null;

// JMenuItem jtm5=null;

JMenuItem jtm6=null;

// JMenuItem jtm7=null;

// JMenuItem jtm8=null;

JMenuItem jtm9=null;

JMenuItem jtm10=null;

JMenuItem jtm11=null;

JMenuItem jtm12=null;

public MangerJieMian() {

this.setTitle("管理员菜单");

JMenuBar menubar1 = new JMenuBar(); //创建菜单栏

this.setJMenuBar(menubar1); //设置此窗体的菜单栏 参数：放置于该窗体的菜单栏

JMenu menu1 = new JMenu("航班管理"); //创建3个菜单

JMenu menu2 = new JMenu("旅客管理");

JMenu menu3 = new JMenu("票务管理");

JMenu menu4 = new JMenu("系统帮助");

menubar1.add(menu1); //将菜单添加进菜单栏

menubar1.add(menu2);

menubar1.add(menu3);

menubar1.add(menu4);

jtm1 = new JMenuItem("查询航班信息");

jtm2 = new JMenuItem("更改航班信息");

// jtm3 = new JMenuItem("校验登陆信息");

jtm4 = new JMenuItem("退出");

// jtm5 = new JMenuItem("旅客信息录入");

jtm6 = new JMenuItem("删除用户信息");

// jtm7 = new JMenuItem("取票信息");

// jtm8 = new JMenuItem("退票信息");

jtm9 = new JMenuItem("订票信息");

// jtm10 = new JMenuItem("退订信息");

jtm10 = new JMenuItem("查询订票信息");

jtm11 = new JMenuItem("系统指南");

jtm12 = new JMenuItem("关于我们");

menu1.add(jtm1);

menu1.addSeparator();

menu1.add(jtm2);

menu1.addSeparator(); //将新分隔符追加到菜单的末尾

// menu1.add(item3);

// menu1.addSeparator();

menu1.add(jtm4);

// menu2.add(jtm5);

menu2.addSeparator();

menu2.add(jtm6);

// menu3.add(jtm7);

// menu3.addSeparator();

// menu3.add(jtm8);

// menu3.addSeparator();

menu3.add(jtm9);

menu3.addSeparator();

menu3.add(jtm10);

menu4.add(jtm11);

menu4.addSeparator();

menu4.add(jtm12);

jtm1.addActionListener(this);

jtm2.addActionListener(this);

// jtm3.addActionListener(this);

jtm4.addActionListener(this);

// jtm5.addActionListener(this);

jtm6.addActionListener(this);

// jtm7.addActionListener(this);

// jtm8.addActionListener(this);

jtm9.addActionListener(this);

jtm10.addActionListener(this);

jtm11.addActionListener(this);

jtm12.addActionListener(this);

this.show();

this.setBounds(100, 100, 500, 300);

this.setVisible(true);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

// public static void main(String[] args) {

// MangerJieMian mj = new MangerJieMian();

// }

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if(e.getSource()==jtm1) //查询航班信息

{

new MangerSearchFlight();

}

if(e.getSource()==jtm2) //查询航班信息

{

new UpdateFlight();

}

if(e.getSource()==jtm4)

{

System.exit(0);

}

if(e.getSource()==jtm6)

{

new DeleteUser();

}

if(e.getSource()==jtm9)

{

new MangerDingPiaoInfor();

}

if(e.getSource()==jtm10)

{

new SearchBook();

}

if(e.getSource()==jtm11)

{

new ZhiNan();

}

if(e.getSource()==jtm12)

{

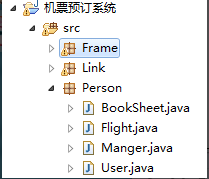
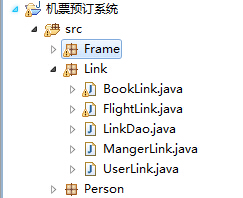
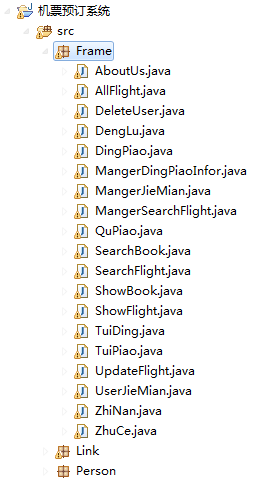
new AboutUs();

}

}

}

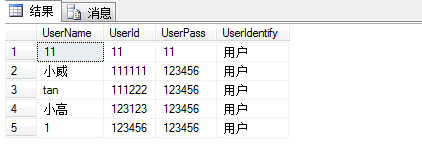
## 4.2代码结构设计



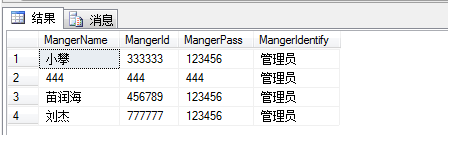
# 5数据库设计

## 5.1数据库结构设计

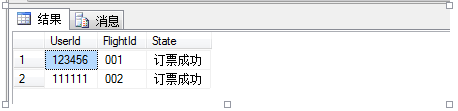
1 用户信息表



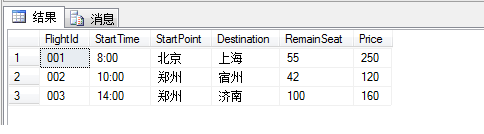
2.管理员信息表



1. 订票信息表



1. 航班信息表



## 5.2数据表关系设计

管理员删除用户

管理员

用户表

# 

增加删除航班

生成订单

更新航班票数

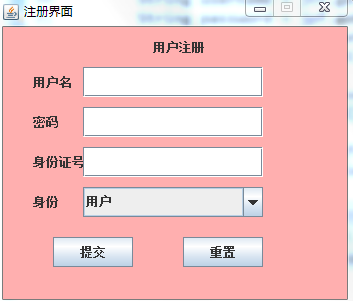
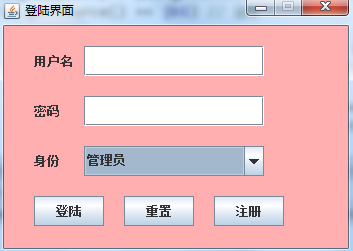
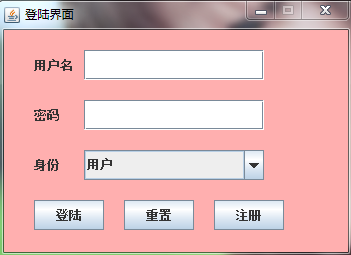
航班表

订票表

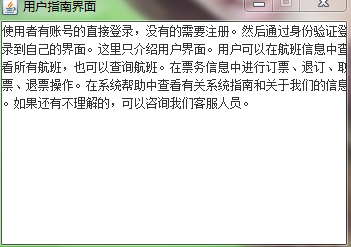
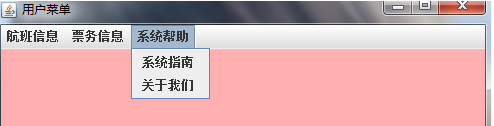
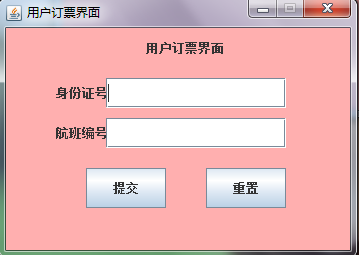
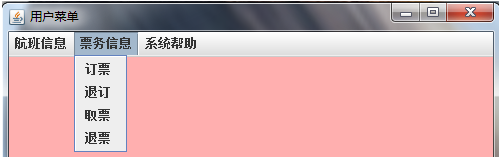
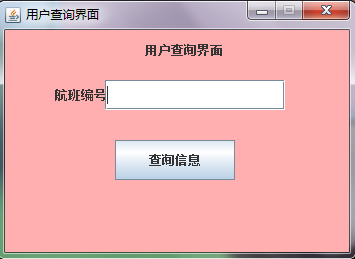
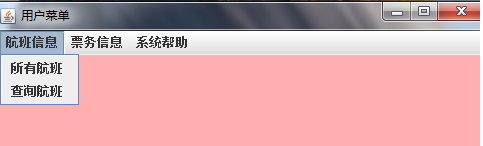
# 6输入设计

## 6.1输入界面设计

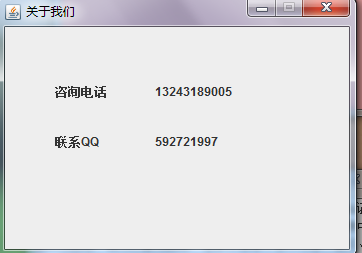
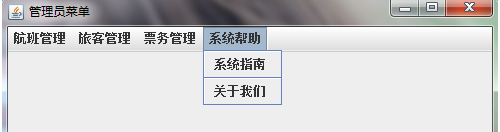
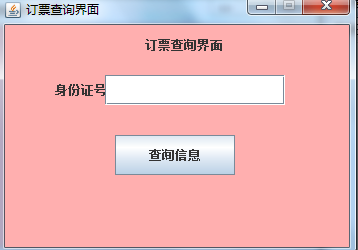
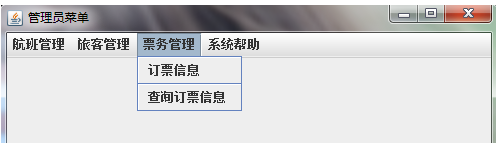
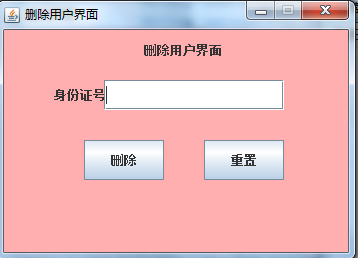
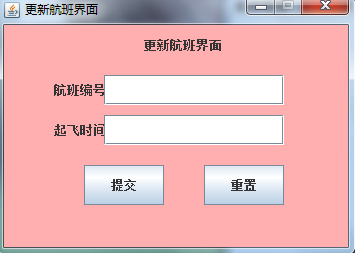
1，用户，管理员登陆注册界面



## 2，用户操作界面



## 3，管理员操作界面



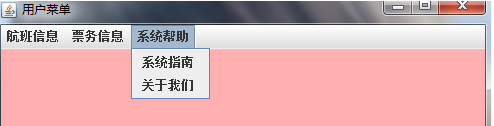
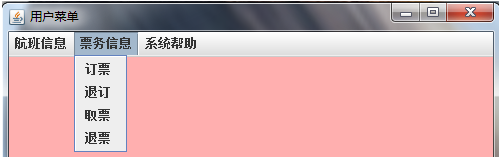
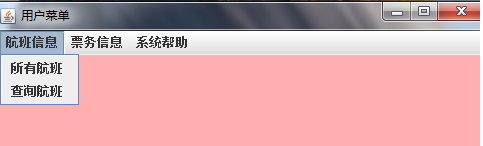
## 6.2输入设备选择

1.从键盘中输入

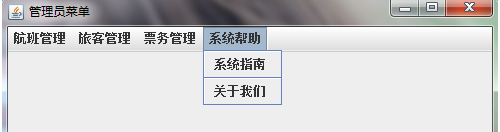
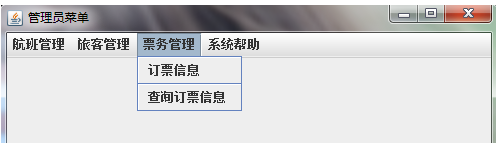
2.从数据库中读取

## 6.3菜单设计

1. 用户系统信息菜单



1. 管理员系统信息菜单



# 7输出设计

## 7.1系统输出内容设计

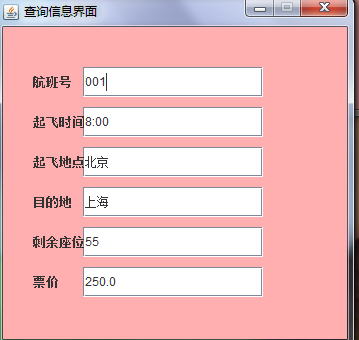
1. 输出各个数据库表中的数据
2. 输出指定查询信息

## 7.2输出界面设计

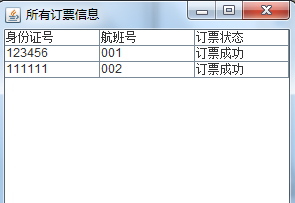
1. 用户查询所有航班界面



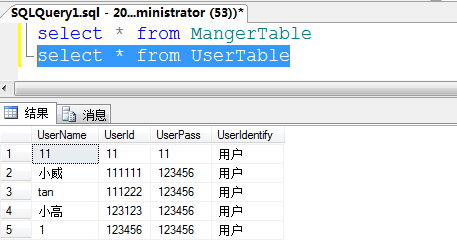
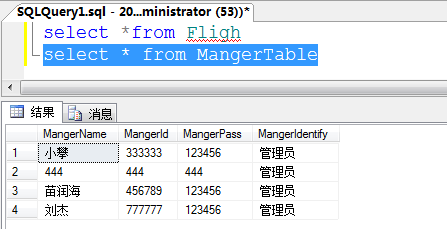
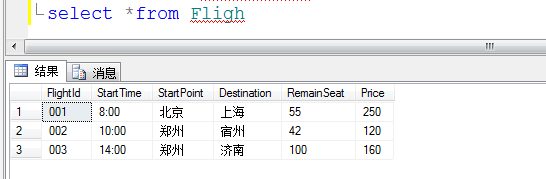
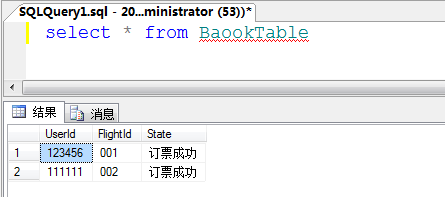
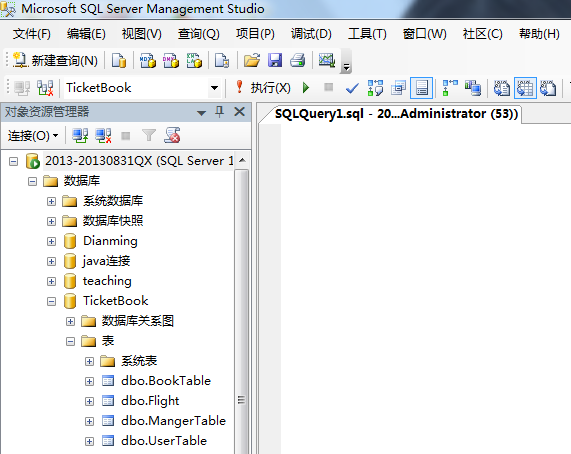
1. 用户查询指定航班信息界面



1. 订票信息界面



1. 数据库表的设计界面



# 参考文献

[1] 李 刚. 《疯狂JAVA讲义》.

[2] 麻志毅 《面向对象分析与设计》.

[3] 刘金安 《实用软件工程教程》 .

[1] <<.数据库技术基础及应用>>,刘国燊编著.北京电子工业出版社.2003年9月.

[2] <<SQLServer2000实例教程>>.杨学全主编,赵慧勤,李英杰,张永周编著北京:电子工业出版社.2004年7月

[3] <<软件工程——原理、方法与应用>>.史济民,顾春华,李昌武,苑荣编著.北京:高等教育出版社.2004年1月.

[4] <<软件工程导>>张海藩编著.论.北京:清华大学出版社.2005年10月.

[5] <<.数据库技术基础及应用>>,刘国燊编著.北京电子工业出版社.2003年9月.

[6] <<SQLServer2000实例教程>>.杨学全主编,赵慧勤,李英杰,张永周编著北京:电子工业出版社.2004年7月

[7] <<软件工程——原理、方法与应用>>.史济民,顾春华,李昌武,苑荣编著.北京:高等教育出版社.2004年1月.