WEB-BASED AIRLINE TICKET BOOKING SYSTEM

Yu Jianming, CSE

Problem in Lieu of Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2004

Yu, Jianming, Web-based airline ticket booking system. Master of Science (Computer Science), August 2004, 28 pp., 20 illustrations, references, 7 titles.

Online airline ticket booking system is one of the essential applications of E-commerce. With the development of Internet and security technology, more and more people begin to consume online, which is more convenient and personal than traditional way. The goal of this system is to make people purchase airline tickets easily. The system is written in JAVA^{TM}. Chapter 1 will introduce some basic conception of the technologies have been used in this system. Chapter 2 shows how the database and the system are designed. Chapter 3 shows the logic of the Web site. In Chapter 4 the interface of the system will be given. Chapter 5 tells the platform of this system.

TABLE OF CONTENTS

LIST OF TABLES

Page

TRADEMARKED NAMES USED

Product Name	Owner
BEA Weblogic™ Platform 8.1	BEA Systems, Inc., www.bea.com
Common Object Request Broker Architecture® (CORBA®)	Needham, MA, Object Management Group, Inc, www.omg.org
Java™, Java Server Pages(JSP™) and JavaScript™	Sun Microsystems, Inc., www.sun.com
Jbuilder™ 9 Enterprise Trial	Borland Software Corporation, www.borland.com
Linux®	Linus Torvalds
Oracle ® 9i.	Oracle Corporation, www.oracle.com
Tomcat™4.0	Apache Digital Corporation, Durango, CO, www.apache.com
UNIX®	The Open Group, www.opengroup.org
Windows® XP™	Microsoft Corporation, www.microsoft.com

1. INTRODUCTION

The Problem and its Purposes

Online airline ticket booking system is one of the essential applications of E-commerce. With the development of Internet and security technology, more and more people begin to consume online, which is more convenient and personal than traditional way. The goal of this system is to make people purchase airline tickets easily. The system is written in Java.

Definition of Terms

What is CORBA? The common object request broker architecture (CORBA) allows distributed applications to interoperate (application to application communication), regardless of what language they are written in or where these applications reside. The CORBA specification was adopted by the object management group to address the complexity and high cost of developing distributed object applications. CORBA uses an object-oriented approach for creating software components that can be reused and shared between applications. Each object encapsulates the details of its inner workings and presents a well-defined interface, which reduces application complexity. The cost of developing applications is reduced, because once an object is implemented and tested, it can be used over and over again. The object request broker (ORB) connects a client application with the objects it wants to use. The client program does not need to know whether the object implementation it is in communication with resides on the same computer or is located on a remote computer somewhere on the network. The client program only needs to know the object's name and understand how to use the object's interface. The ORB takes care of the details of locating the object, routing the request, and returning the result

Remote method invocation (RMI) enables you to create distributed Java-to-Java applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. A Java program can make a call on a remote object once it obtains a reference to the remote object, either by looking up the remote object in the bootstrap naming service provided by RMI or by receiving the reference as an argument or a return value. A client can call a remote object in a server, and that server can also be a client of other remote objects. RMI uses object serialization to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism.

Web Services is about interoperability, which uses Simple Object Access Protocol (SOAP) an Extensible Markup Language(XML) based protocol for exchanging information.

2. THE SYSTEM AND DATABASE DESIGN

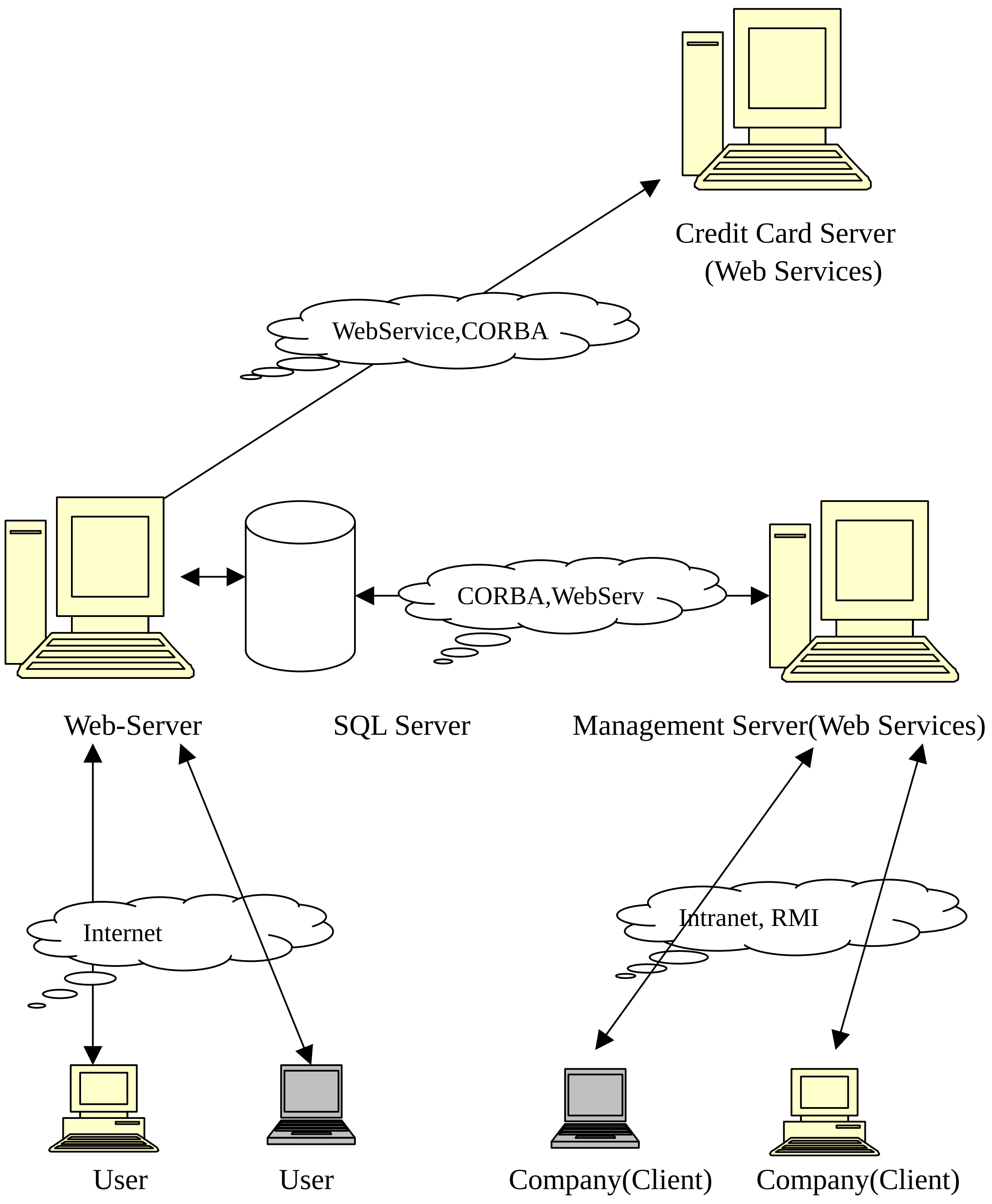


Figure 1: The structure of the web site

System Design

The Web-based airline ticket booking system uses client/server frame. The customers can use Web browser to access the system and book airline tickets. Because Internet and web browsers are widely used all over the world, there is no need to train customers how to use them. The second advantage is that there is no limit on customers' operating systems, for example, they can use almost all kinds of popular operating systems such as UNIX, Linux and Window XP. In the server part, we use Java Server Pages (JSP), which is written in Java. So the server programs can also be run on most kinds of platforms, such as UNIX, Linux, or Windows. This increases the flexibility of the system. And the companies who use the system do not need to buy new hardware devices. For the communication between the server part and the third part services such as credit card verification, we use Web Services and common object request broker architecture (CORBA). Those technologies are widely used for distributed systems. And the cost of developing applications is reduced, because the source code can be reused by other applications.

After the user send a request by JSP to the Web server, the Web server search the database according to the request and then send back the result to user. After the user input his credit card information and buy a ticket. The server will connect (by CORBA or Web Services) to the remote Airline company to confirm the ticket, and get a confirm number from the company. Then the server will connect (by CORBA or Web Services) to a remote credit card server, such as VISA or Master, to identify the user and confirm the payment. If there is a firewall between them they can use Web Services instead of CORBA. The airline companies provide all the airline information. They can input the information directly from a remote Client through their local server by remote method invocation. Different company uses different port to make sure its security.

Database Design

The database of this system is Oracle. The information in tables AIRLINE, FLIGHTCOMPANY, EMPLOYEE and FLIGHT is input by system operators. The information in Customer table is filled in by the system automatically when customers register into the system. And the Itinerary table is filled in by the system automatically also, after customers book tickets. Before they book tickets, the information about their selection, such as flight number and price, is stored in the session not in database.

The ITINERARY table will store all the information of every ticket. The FLIGHTCOMPANY and EMPLOYEE tables store the information of every company and its employees, then the system management part uses this information to identify the operators when they login in. The CUSTOMER table is filled out by the system when users register. The operators of the airline companies fill out the AIRLINE and FLIGHT tables.

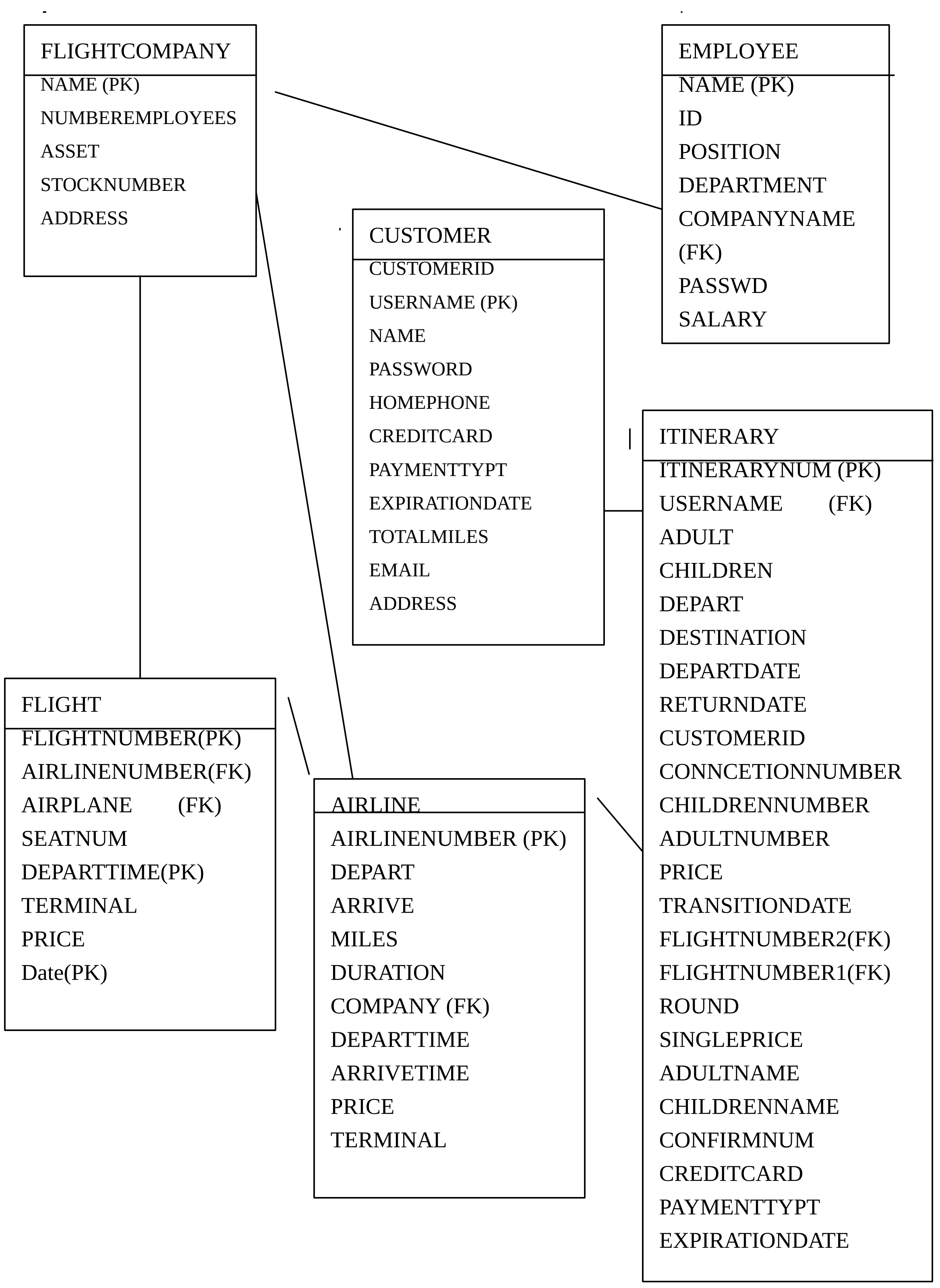


Figure 2. The tables of database

3. THE LOGIC OF THE WEB SITE

Through the Web server, users can booking airline tickets and pay bills online. To use this system the user should register first through the registration page by inputting the user ID, password, user name, address and so on. Then the user can login to the system and trace their purchase by the user ID and password. The user information will be stored in the entire session until he or her logouts. After the user logins the system, he can search for flights by the search bar or the main search page. The search based on the destination, departure airport, date and Airline companies. When the use press the search button at the bottom, the system will search for him from the database and return the result to the user sorted by price. The result includes the price, airline company, airline number and so on. To select the flight, he user just need to press the "select" button corresponding to that flight result. If it is a round trip, the user will be asked to choose the return flight continually. Finally the system will show the user what he has chosen. Then the use can click "Buy it" or just search again. If the user wants to buy it, he will go to the next step to pay for it. In the payment page, he is required to input his credit card information by the instruction of the page. Then the system will check the credit card information with the credit card company by using common object request broker architecture (CORBA) or Web Services. After the information is proved, the user will be asked to confirm their purchase. Then the system connects the airline company to confirm the purchase and returns the user a Confirm number. Then the system will send the purchase information to the Credit Card company to finish the transaction. All the server parts of the credit card companies and the airline companies are simulated by this system. They just waiting for the invocation from the client and serve them by just returning a simple result without dealing with the data really. They are also written in Java. After the purchase is finished, the user can view the Shopping Cart to see what he has bought. And the system will send a confirm email to the user according to the information they input when they registered. There is a Personal Assistant Bar at the left side of all pages, through which the user can change their personal information, such as, password and address.

The above is the main purchase logic. Next I will talk about the system management part at the airline company. It is written in Java using remote method invocation (RMI). The server registers to a RMI registration and waits for clients to connect to it. The user part can be run either on the airline company or the website company. After they login to the system, the system will check the user name and password in the database. Then the verified user can use the system to add/delete/modify/search airline information, user information and the ticket information.

Now let me introduce the data stream in this system. For the booking system, all the user information, such as the user name and the depart airport, are stored in the whole session until the user logs out or the session is expired. After the user buy the ticket, the ticket information will be stored in the local Oracle database and sent to the airline company also to book seats.

Figure 3. Main pages of Airline Ticket Booking System.

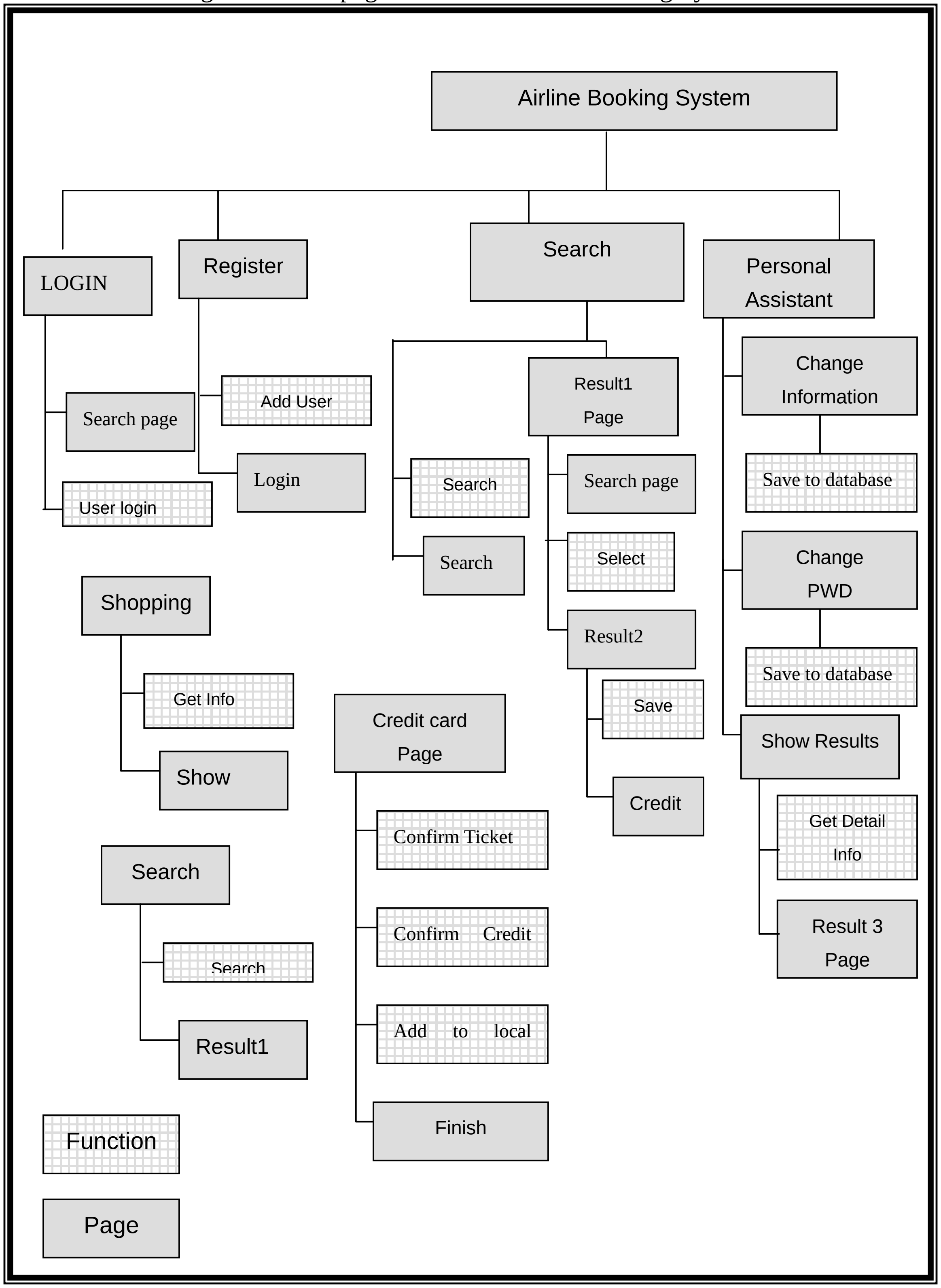
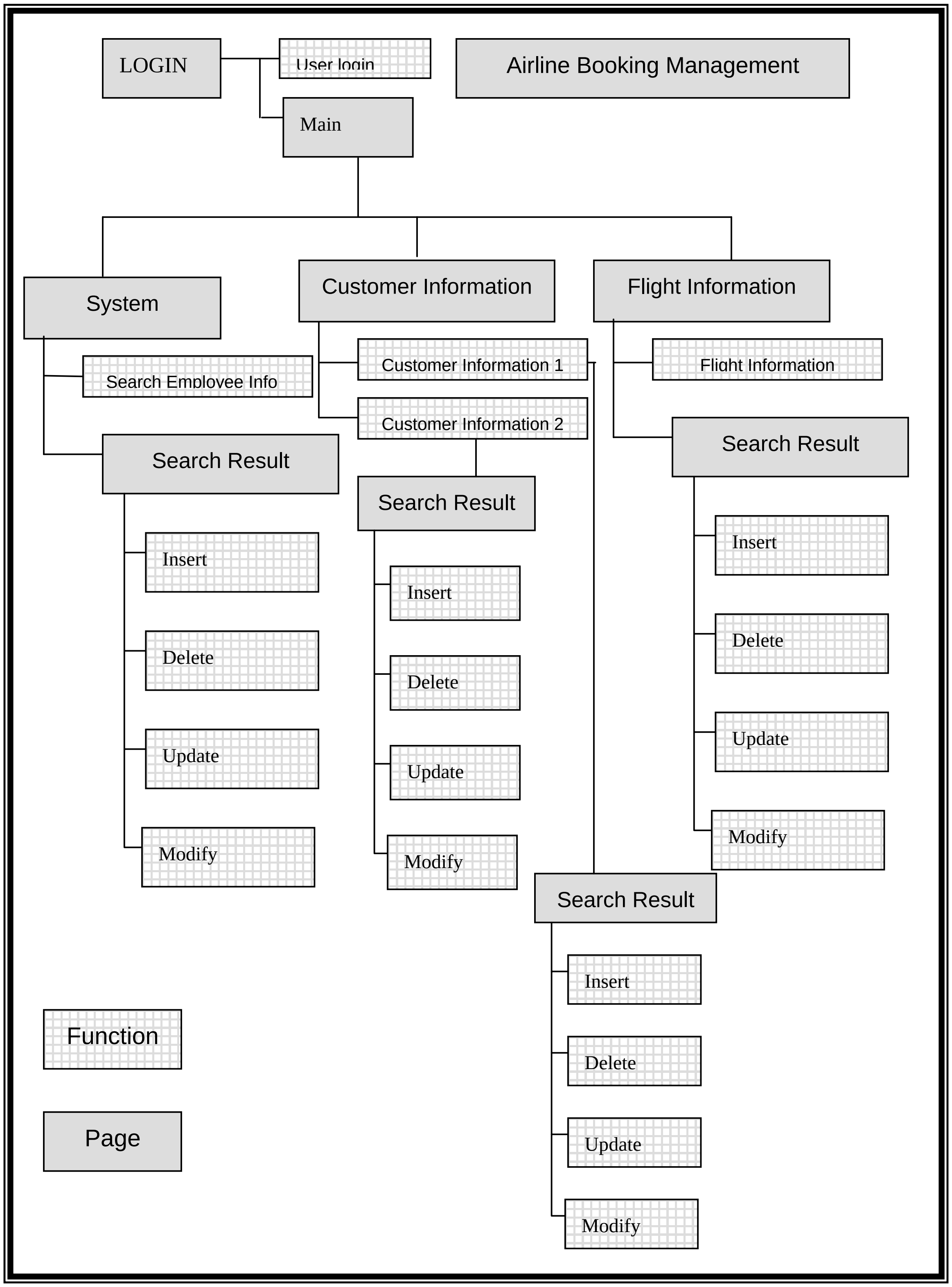


Figure 4. Structure of Client part of Management of Airline Ticket Booking System.



4. THE PLATFORM OF THIS SYSTEM AND CONFIGURATION

WEB-Server : BEA Weblogic^{TM} Platform 8.1, Tomcat^{TM} 4.0

Operating System: Windows XP

Programming languages: Java, JavaScript, HTML.

Database : Oracle 9i.

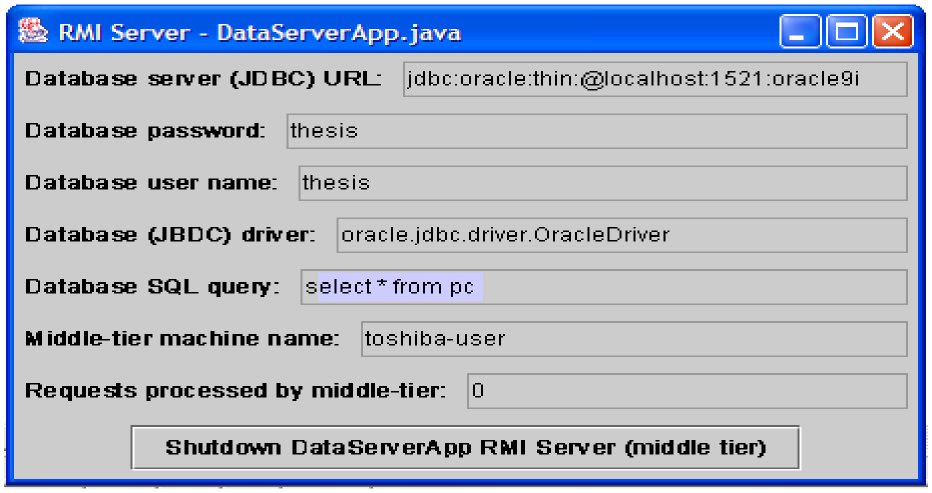
JDBC: oracle.jdbc.driver.OracleDriver.

Development tools: Jbuilder^{TM} 9 Enterprise Trial

5. INTERFACE OF THE SYSTEM

System Management Part

After starting the Remote Method Invocation (RMI) registry, we can start the server. And the server will listen on the port to accept the connection from clients.


RMI Server - DataServerApp.java

Database server (JDBC) URL: jdbc:oracle:thin:@localhost:1521:oracle9i

Database password: thesis

Database user name: thesis

Database (JBDC) driver: oracle.jdbc.driver.OracleDriver

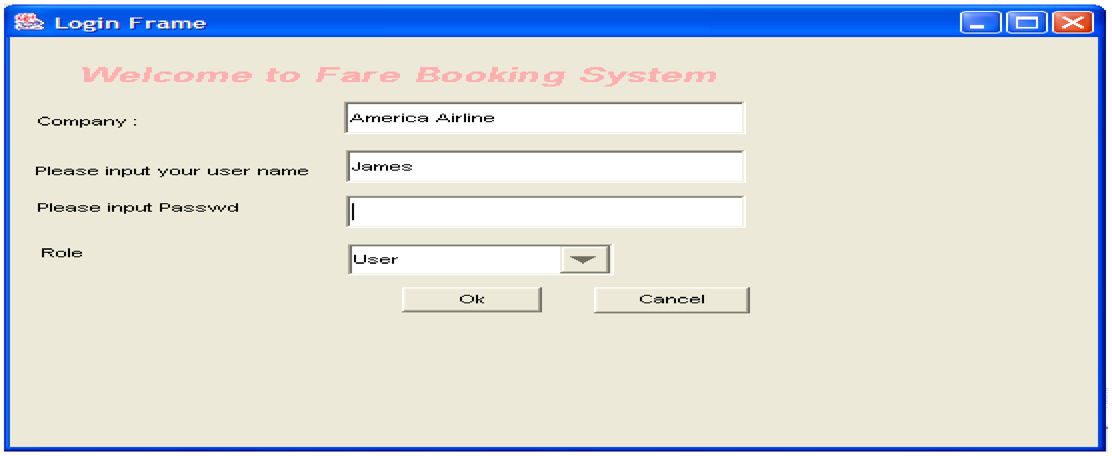
Database SQL query: select * from pc

Middle-tier machine name: toshiba-user

Requests processed by middle-tier: 0

Shutdown DataServerApp RMI Server (middle tier)

Figure 5. The Server panel of RMI



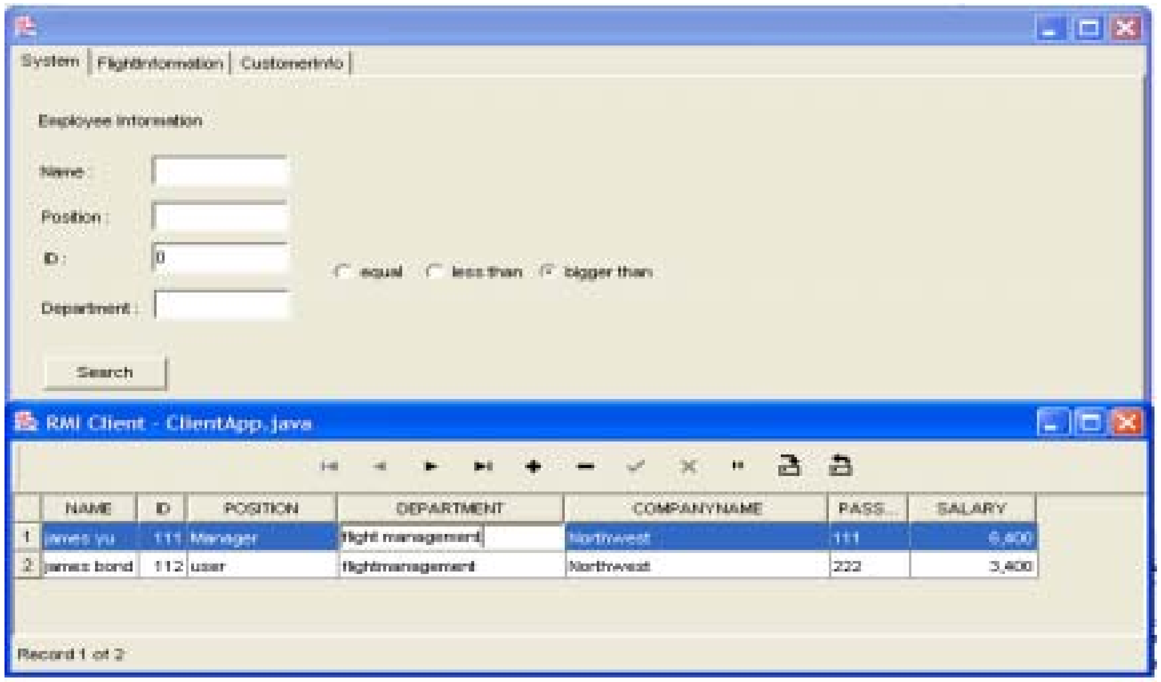
Company : America Airline

Please input your user name James

Please input Passwd

Role User

Figure 6. Login form of the Client part of RMI



Employee Information

Name :

Position :

ID : 0

Department :

( ) equal ( ) less than (•) bigger than

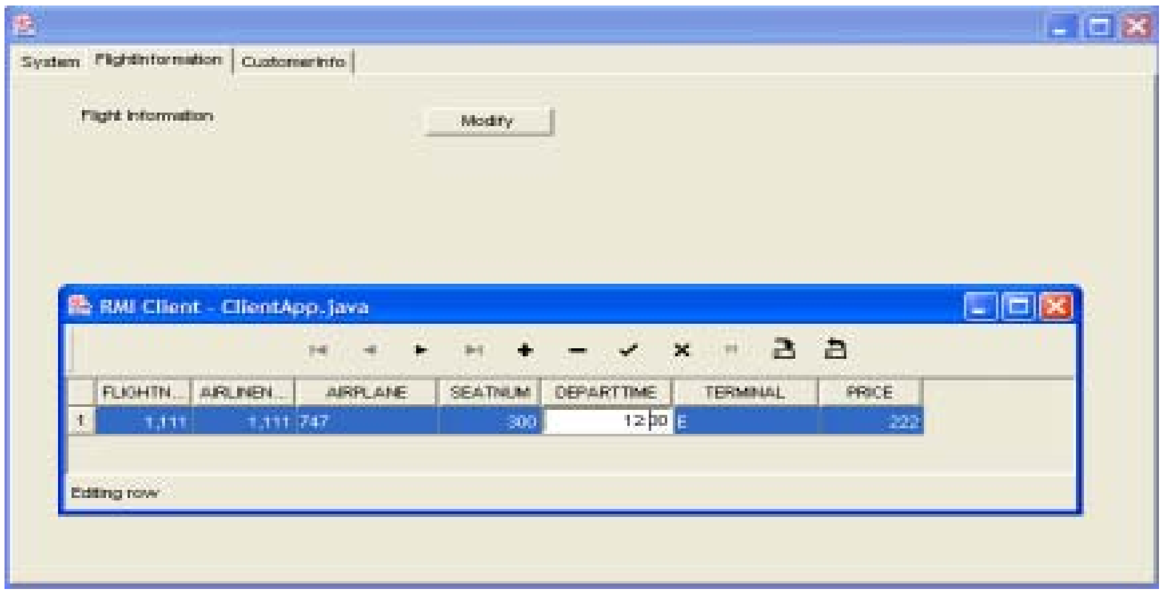
RMI Client - ClientApp.java

	NAME	ID	POSITION	DEPARTMENT	COMPANYNAME	PASS...	SALARY
1	james yu	111	Manager	flight management	Northwest	111	6,400
2	james bond	112	user	flightmanagement	Northwest	222	3,400

Record 1 of 2

Figure 7. The Client part of RMI for Employee Information

Users can search Employees information according to name, position, ID or Department.



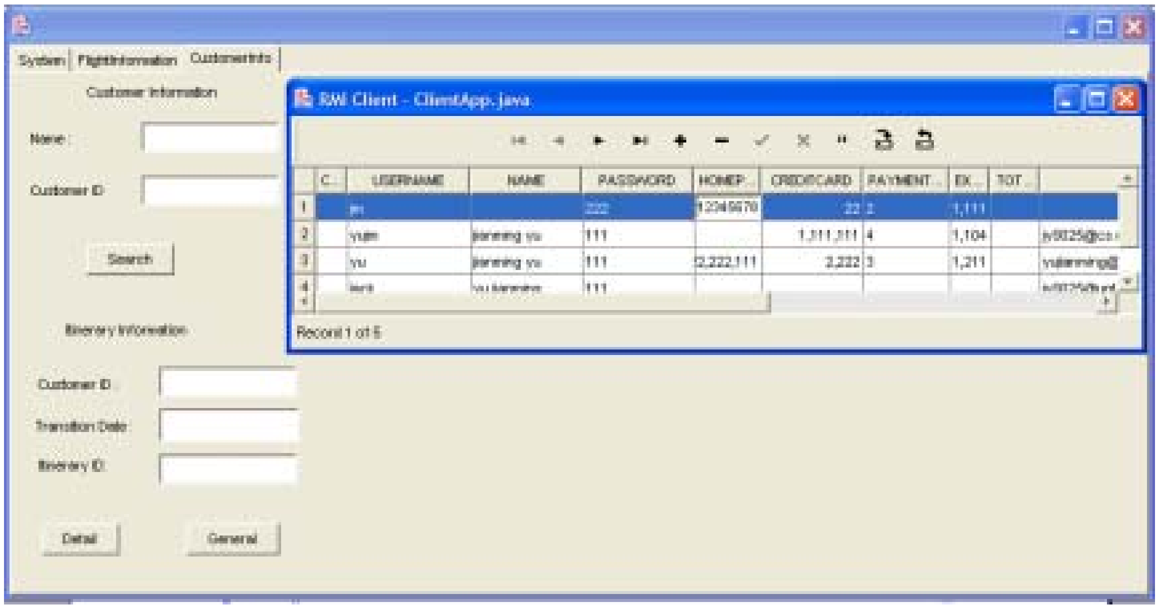
Flight Information

RMI Client - ClientApp.java

	FLIGHTN...	AIRLINEN...	AIRPLANE	SEATNUM	DEPARTTIME	TERMINAL	PRICE
1	1,111	1,111	747	300	12:00	E	222

Editing row

Figure 8. The Client part of RMI for Flight Information

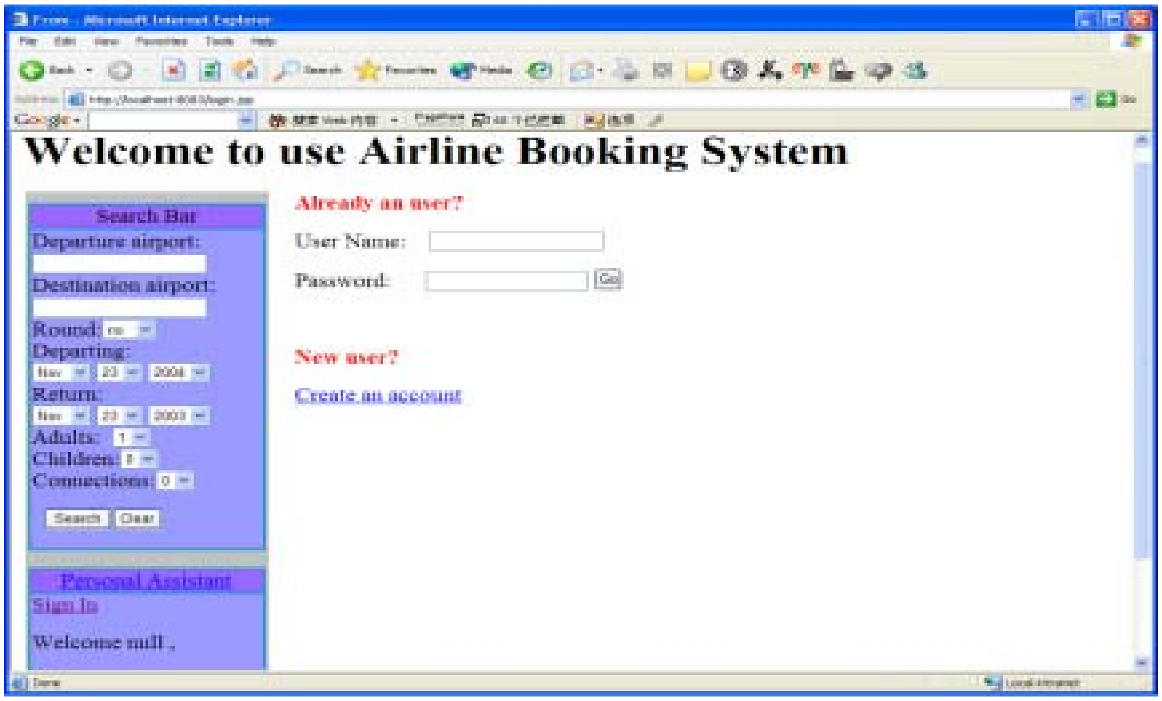


	C...	USERNAME	NAME	PASSWORD	HOMEP...	CREDITCARD	PAYMENT...	EX...	TOT...	
1		jin		222	12345678	22	2	1,111		
2		yujin	janning yu	111		1,111,111	4	1,104		jy5025@cs
3		yu	janning yu	111	2,222,111	2,222	3	1,211		yujanning@
4		[illegible]	yu janning	111						[illegible]

Record 1 of 5

Figure 9.The Client part of RMI for Customer Information

Ticket Booking System on Web End


Form - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Google
Done
Local intranet

Welcome to use Airline Booking System

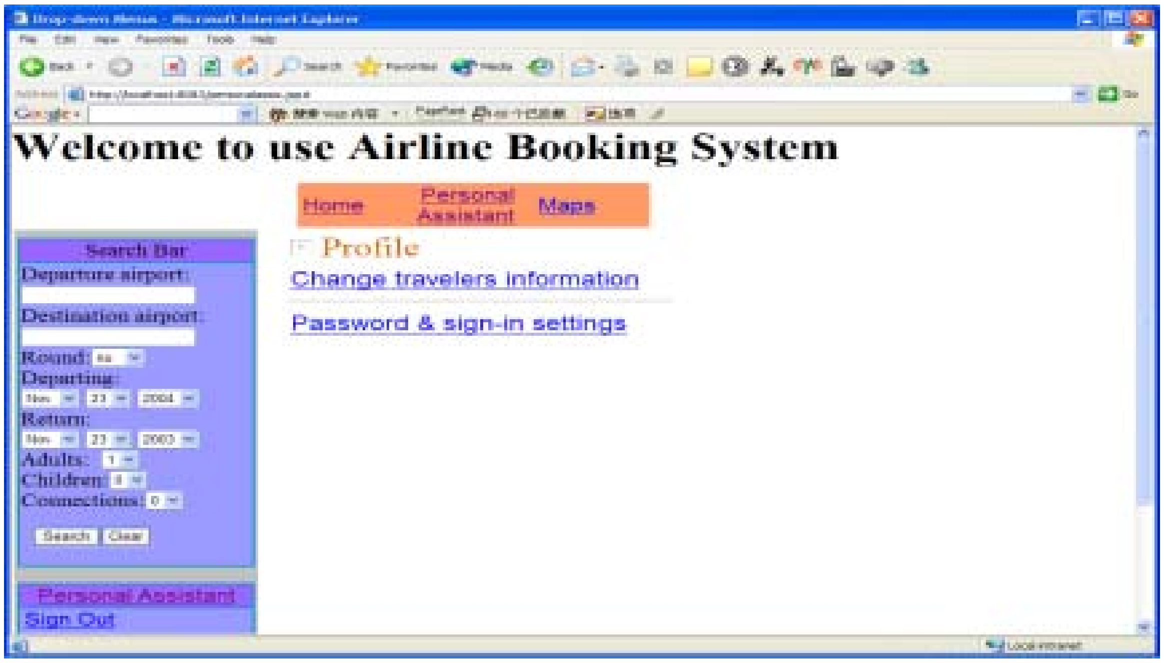
Search Bar
Departure airport:
Destination airport:
Round: no
Departing: Nov 23 2004
Return: Nov 23 2003
Adults: 1
Children: 0
Connections: 0
Search Clear

Personal Assistant
Sign In
Welcome null ,

Already an user?
User Name:
Password: Go

New user?
Create an account

Figure 10. Login form of the Java Server Pages (JSP) Part



Welcome to use Airline Booking System

Home | Personal Assistant | Maps

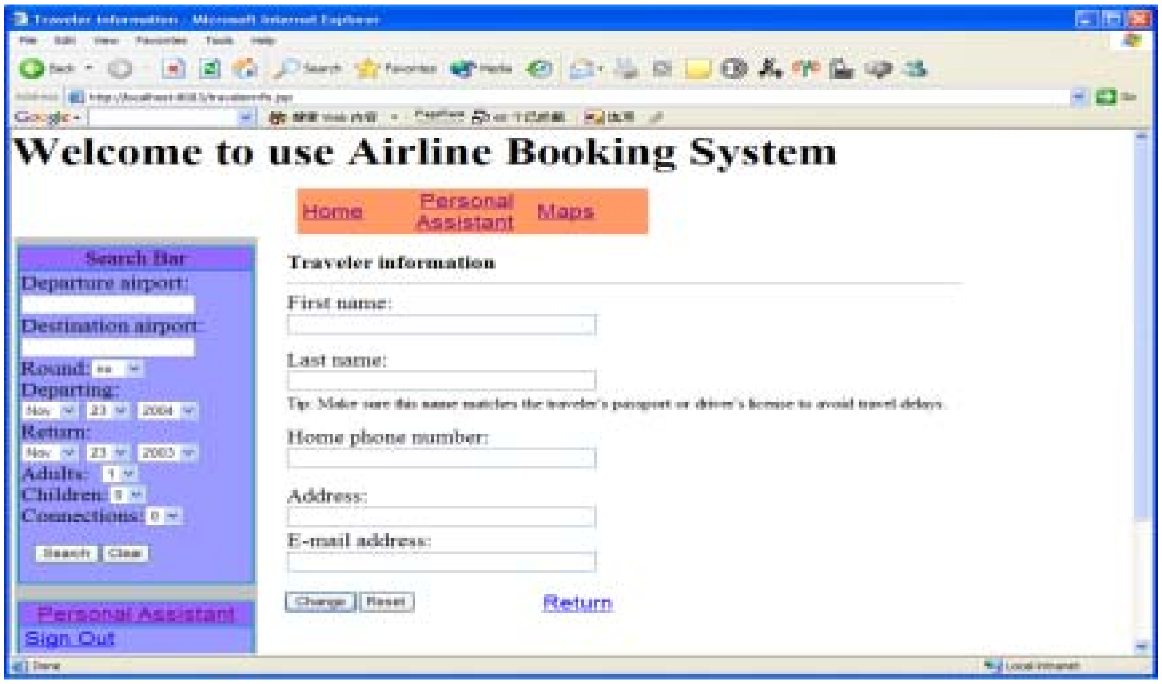
Search Bar
Departure airport:
Destination airport:
Round: no
Departing: Nov 23 2004
Return: Nov 23 2003
Adults: 1
Children: 0
Connections: 0
Search | Clear

Personal Assistant
Sign Out

Profile
Change travelers information
Password & sign-in settings

Figure 11. Personal Assistant of JSP Part

Users can modify their own information by the Personal Assistant



Welcome to use Airline Booking System

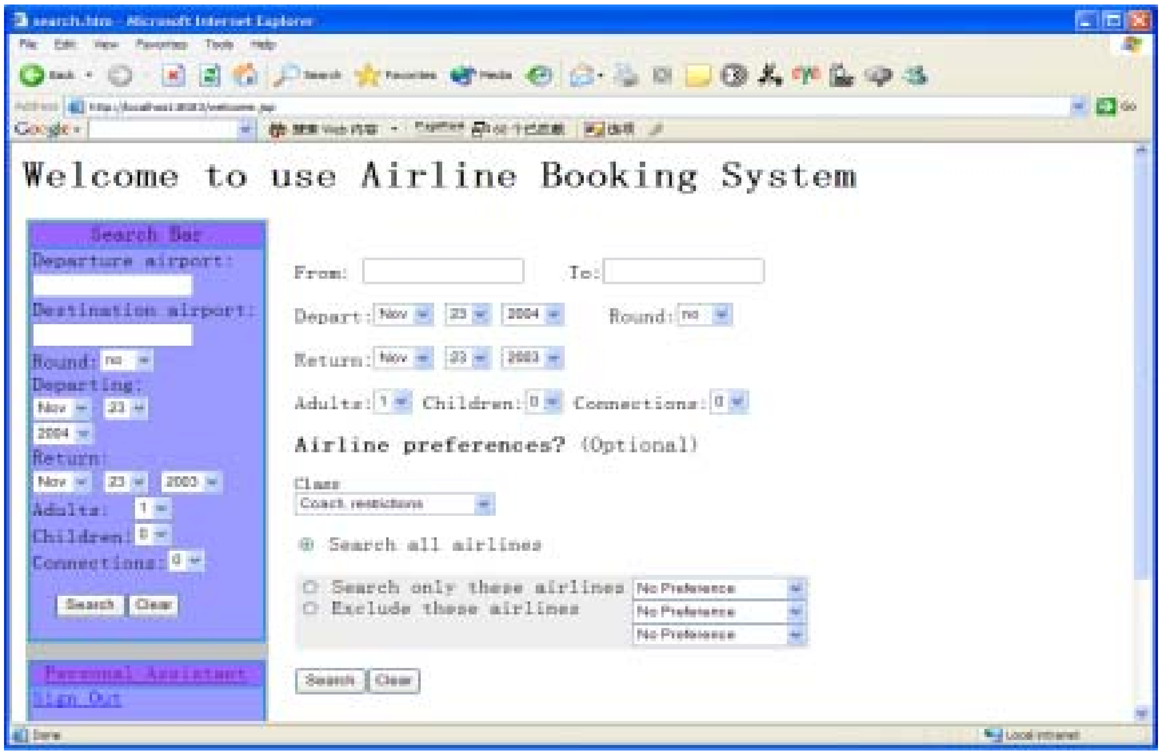
Home | Personal Assistant | Maps

Search Bar
Departure airport:
Destination airport:
Round: no
Departing: Nov 23 2004
Return: Nov 23 2003
Adults: 1
Children: 0
Connections: 0
Search | Clear

Personal Assistant
Sign Out

Traveler information
First name:
Last name:
Tip: Make sure this name matches the traveler's passport or driver's license to avoid travel delays.
Home phone number:
Address:
E-mail address:
Change | Reset   Return

Figure 12. Traveler Information Modification form of JSP Part


Welcome to use Airline Booking System

Search Bar
Departure airport:
Destination airport:
Round: no
Departing:
Nov 23
2004
Return:
Nov 23 2003
Adults: 1
Children: 0
Connections: 0
Search Clear

Personal Assistant
Sign Out

From: To:
Depart: Nov 23 2004 Round: no
Return: Nov 23 2003
Adults: 1 Children: 0 Connections: 0

Airline preferences? (Optional)

Class
Coach, restrictions

Search all airlines

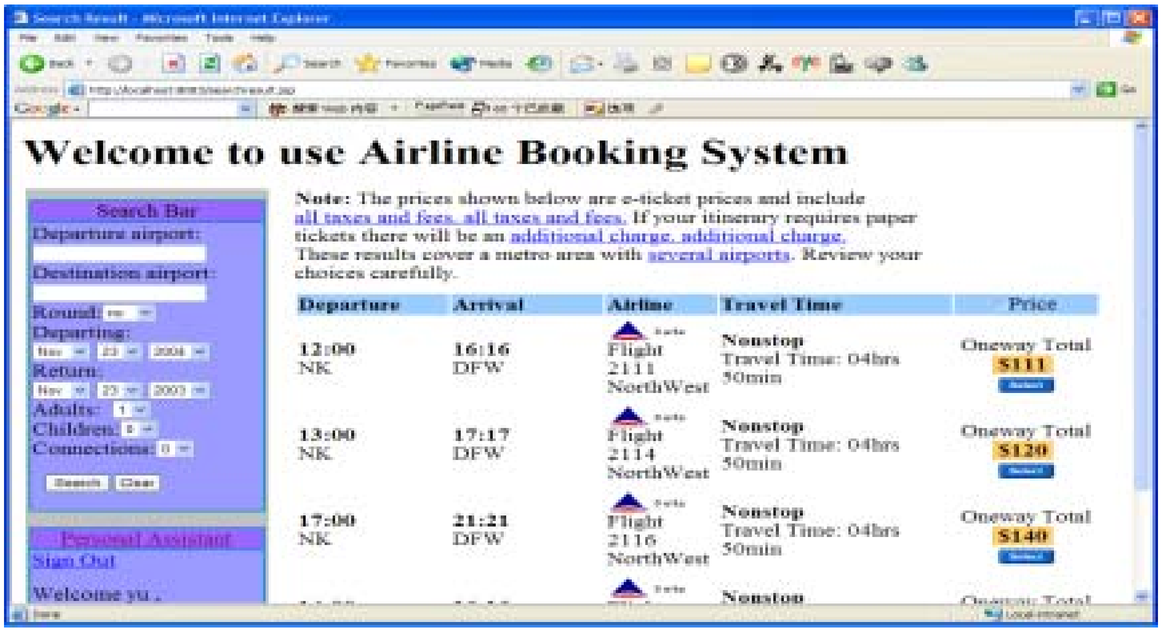
Search only these airlines
Exclude these airlines
No Preference
No Preference
No Preference

Search Clear

Figure 13. Search form of JSP Part

After the user login the system, they can access this form to search flights or use the Search Bar on the left of the page. After they buy a ticket, they can see the message “There is a new ticket in your shopping cart”.

After the user fills out the form and hit “Search” button, he will go to the result page (figure 12). If the user chooses to buy round trip tickets, he will be asked to choose the return flights also. The price will be added automatically. And the information of the flights is stored in this session, so the next page can use them.



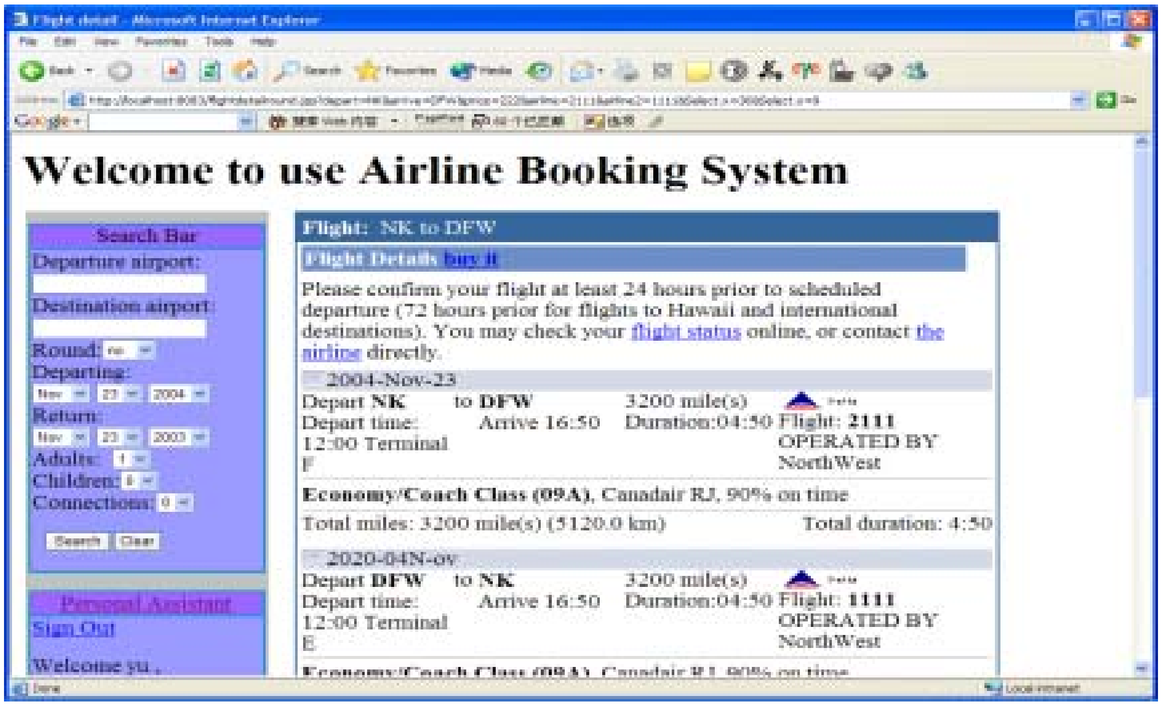
Welcome to use Airline Booking System

Search Bar
Departure airport:
Destination airport:
Round:
Departing:
Return:
Adults:
Children:
Connections:
Personal Assistant
Sign Out
Welcome yu .

Note: The prices shown below are e-ticket prices and include all taxes and fees. all taxes and fees. If your itinerary requires paper tickets there will be an additional charge. additional charge.
These results cover a metro area with several airports. Review your choices carefully.

Departure	Arrival	Airline	Travel Time	Price
12:00 NK	16:16 DFW	Flight 2111 NorthWest	Nonstop Travel Time: 04hrs 50min	Oneway Total $111
13:00 NK	17:17 DFW	Flight 2114 NorthWest	Nonstop Travel Time: 04hrs 50min	Oneway Total $120
17:00 NK	21:21 DFW	Flight 2116 NorthWest	Nonstop Travel Time: 04hrs 50min	Oneway Total $140

Figure 14. the search result of JSP Part



Welcome to use Airline Booking System

Search Bar
Departure airport:
Destination airport:
Round:
Departing:
Return:
Adults:
Children:
Connections:
Personal Assistant
Sign Out
Welcome yu .

Flight: NK to DFW

Flight Details buy it

Please confirm your flight at least 24 hours prior to scheduled departure (72 hours prior for flights to Hawaii and international destinations). You may check your flight status online, or contact the airline directly.

2004-Nov-23

Depart **NK** to **DFW** 3200 mile(s)
Depart time: 12:00 Terminal F Arrive 16:50 Duration:04:50 Flight: **2111** OPERATED BY NorthWest

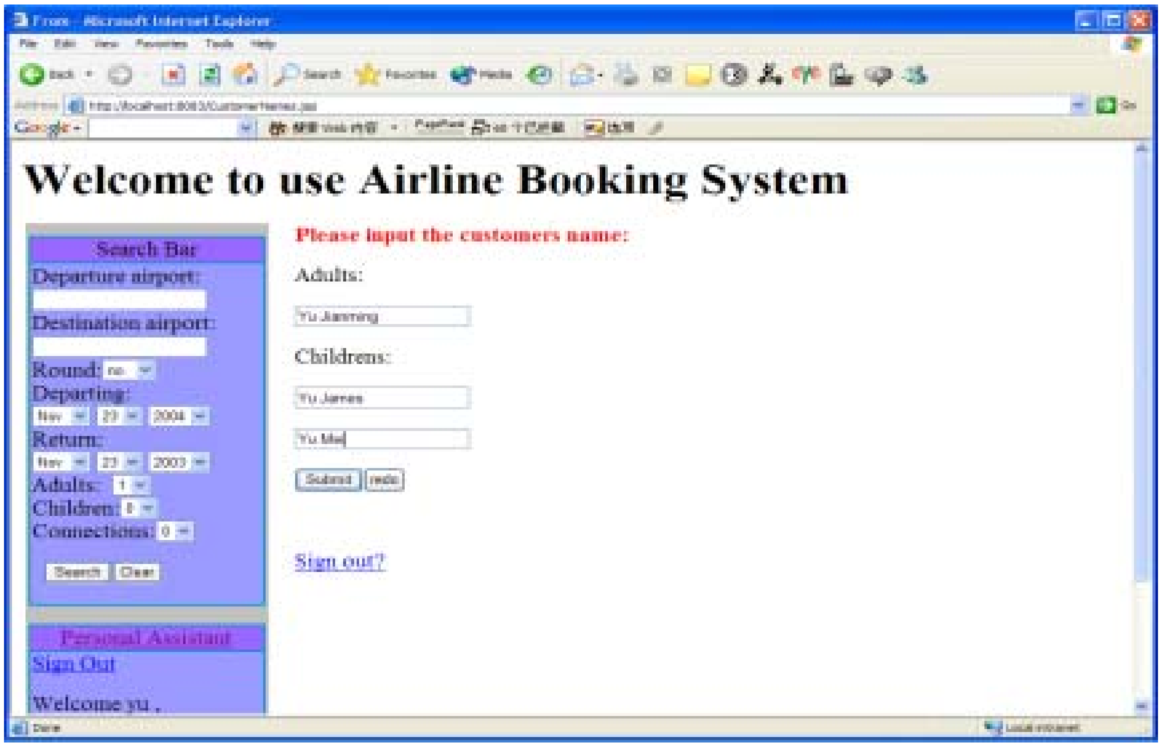
Economy/Coach Class (09A), Canadair RJ, 90% on time

Total miles: 3200 mile(s) (5120.0 km) Total duration: 4:50

2020-04N-ov

Depart **DFW** to **NK** 3200 mile(s)
Depart time: 12:00 Terminal E Arrive 16:50 Duration:04:50 Flight: **1111** OPERATED BY NorthWest

Figure 15. Search summary of the JSP Part



Welcome to use Airline Booking System

Search Bar

Departure airport:

Destination airport:

Round: no

Departing: Nov 23 2004

Return: Nov 23 2003

Adults: 1

Children: 0

Connections: 0

Search Clear

Personal Assistant

Sign Out

Welcome yu ,

Please input the customers name:

Adults:

Yu Jianming

Childrens:

Yu James

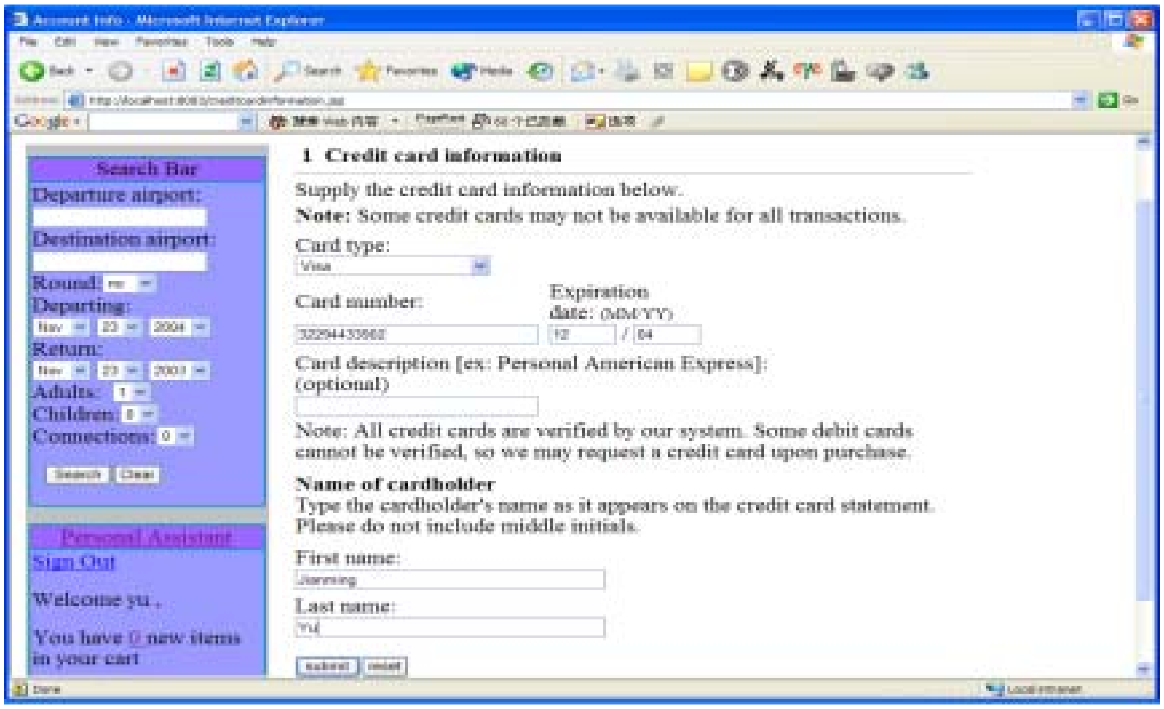
Yu Mei

Submit redo

Sign out?

Figure 16. Customer input form of the JSP Part

If the user will be asked to input the adults and children names, which will appear in the tickets, according to the adults and children number the user choose when searching.


Account Info - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Search Bar
Search
Clear
Personal Assistant
Sign Out
submit
reset
Done
Local intranet

Departure airport:

Destination airport:

Round: no

Departing: Nov 23 2004

Return: Nov 23 2003

Adults: 1

Children: 0

Connections: 0

Welcome yu ,

You have 0 new items in your cart

1 Credit card information

Supply the credit card information below.

Note: Some credit cards may not be available for all transactions.

Card type: Visa

Card number: 32294433602

Expiration date: (MM/YY) 12 / 04

Card description [ex: Personal American Express]: (optional)

Note: All credit cards are verified by our system. Some debit cards cannot be verified, so we may request a credit card upon purchase.

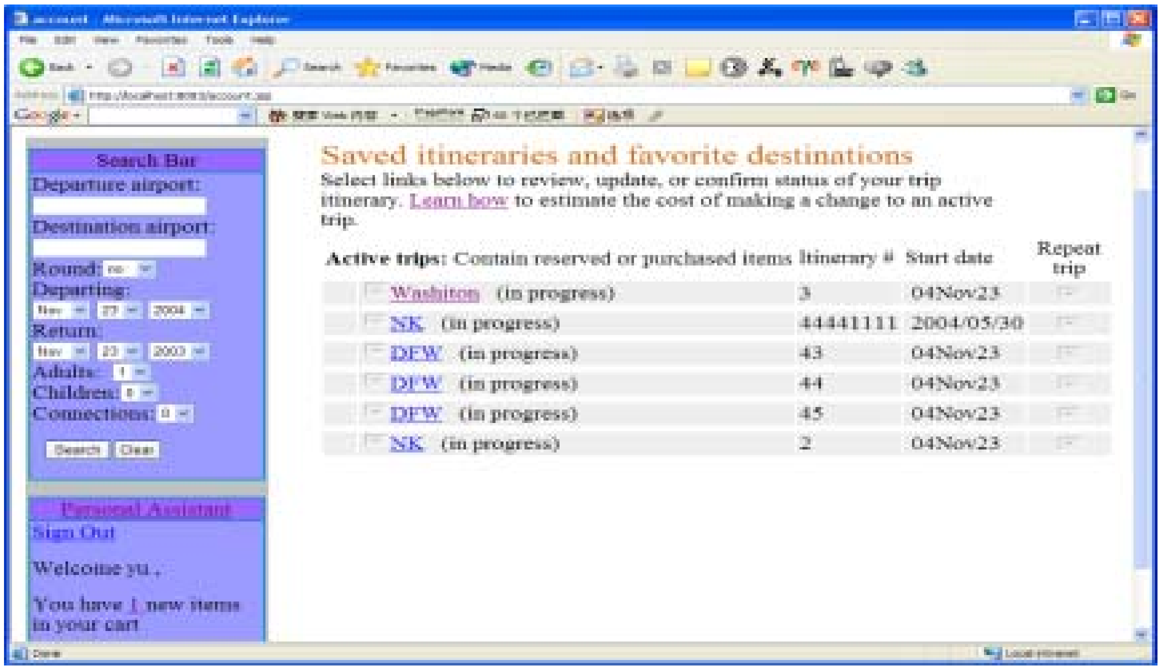
Name of cardholder

Type the cardholder's name as it appears on the credit card statement. Please do not include middle initials.

First name: Jianming

Last name: Yu

Figure 17. Credit card information input form of the JSP Part


account - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Search Bar
Search
Clear
Personal Assistant
Sign Out
Done
Local intranet

Departure airport:

Destination airport:

Round: no

Departing: Nov 23 2004

Return: Nov 23 2003

Adults: 1

Children: 0

Connections: 0

Welcome yu ,

You have 1 new items in your cart

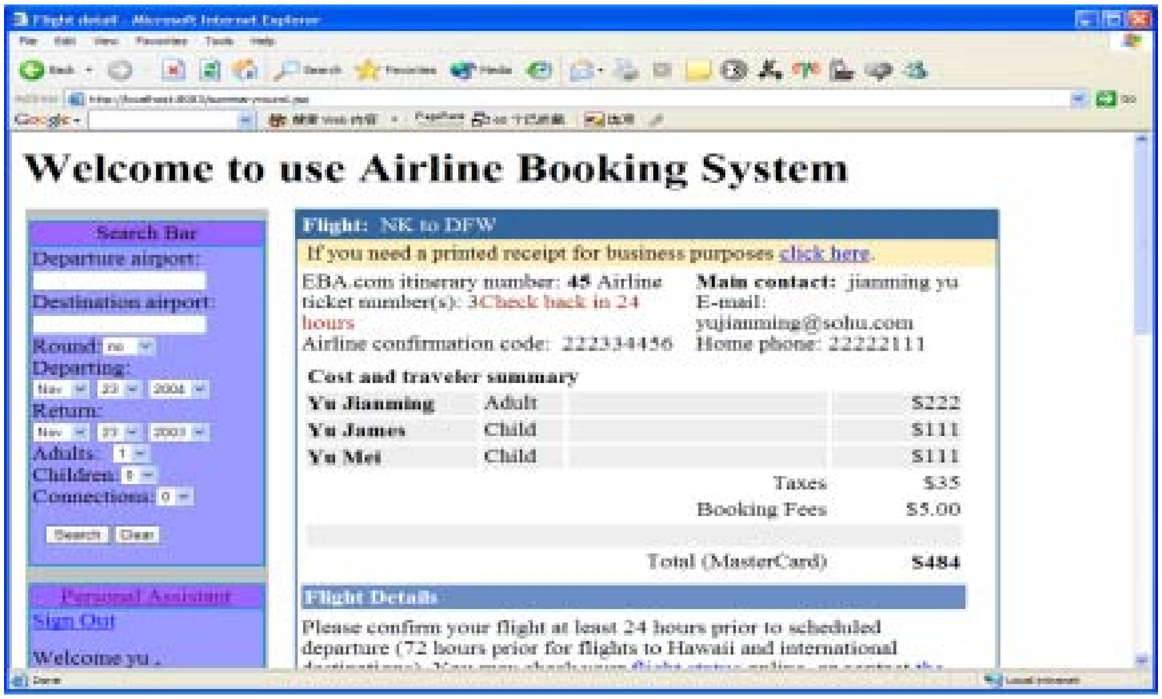
Saved itineraries and favorite destinations

Select links below to review, update, or confirm status of your trip itinerary. Learn how to estimate the cost of making a change to an active trip.

Active trips: Contain reserved or purchased items	Itinerary #	Start date	Repeat trip
Washiton (in progress)	3	04Nov23	
NK (in progress)	44441111	2004/05/30	
DFW (in progress)	43	04Nov23	
DFW (in progress)	44	04Nov23	
DFW (in progress)	45	04Nov23	
NK (in progress)	2	04Nov23	

Figure 18. Saved itineraries of the JSP part

The user can see all the itineraries that he has bought.



Welcome to use Airline Booking System

Search Bar
Departure airport:
Destination airport:
Round: no
Departing: Nov 23 2004
Return: Nov 23 2003
Adults: 1
Children: 0
Connections: 0

Personal Assistant
Sign Out
Welcome yu .

Flight: NK to DFW

If you need a printed receipt for business purposes click here.

EBA.com itinerary number: **45** Airline ticket number(s): 3Check back in 24 hours
Airline confirmation code: 222334456

Main contact: jianming yu
E-mail: yujianming@sohu.com
Home phone: 22222111

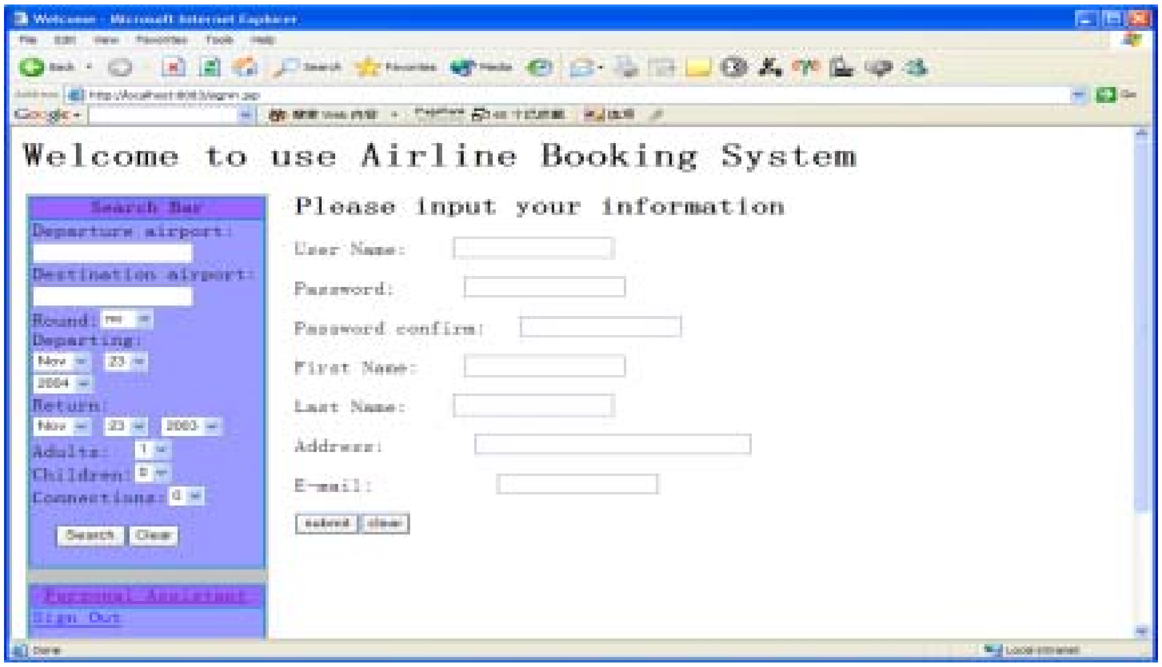
Cost and traveler summary

Yu Jianming	Adult	$222
Yu James	Child	$111
Yu Mei	Child	$111
	Taxes	$35
	Booking Fees	$5.00
	Total (MasterCard)	**$484**

Flight Details

Please confirm your flight at least 24 hours prior to scheduled departure (72 hours prior for flights to Hawaii and international

Figure 19. Detail of itinerary of the JSP part



Welcome to use Airline Booking System

Search Bar
Departure airport:
Destination airport:
Round: no
Departing: Nov 23 2004
Return: Nov 23 2003
Adults: 1
Children: 0
Connections: 0

Personal Assistant
Sign Out

Please input your information

User Name:
Password:
Password confirm:
First Name:
Last Name:
Address:
E-mail:

Figure 20. Register form of the JSP part

The user can register to the system through this page.

6. PROGRAMS

Common Object Request Broker Architecture (CORBA)

IDL(Interface Define Language):

```
module RemoteServer {
   interface CreditCard {
      string Transaction(in string cardnum, in string expiredate,in string name,in
string extend);
   };
   interface AirlineCompony {
      string confirmTicket(in string itinerationNum,in string CompanyName,in string
      AirlineNum,in string date,in string num,in string extend);
   };
};
```

Remote Method Invocation (RMI)

The policy file is :

```
grant {
     // allows anyone to listen on un-privileged ports
     permission java.net.SocketPermission "*:1024-65535", "listen,accept,connect";
};
```

```
//Client part, show the GUI to the user and get dataset from remote server.

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

import com.borland.dbswing.*;
import com.borland.dx.sql.dataset.*;
import com.borland.dx.dataset.*;
public class ClientFrame1 extends JFrame {
    String serverobj=null;
    String SqlText=null;
    JPanel contentPane;
    BorderLayout borderLayout1 = new BorderLayout();
    JPanel jPanel1 = new JPanel();
    com.borland.dx.dataset.TableDataSet tableDataSet1 =
              new com.borland.dx.dataset.TableDataSet();
    ClientProvider clientProvider1;
    ClientResolver clientResolver1;

    JdbNavToolBar jdbNavToolBar1 = new JdbNavToolBar();
    TableScrollPane tableScrollPane1 = new TableScrollPane();
    JdbStatusLabel jdbStatusLabel1 = new JdbStatusLabel();
    JdbTable jdbTable1 = new JdbTable();
    ResourceBundle res = Res.getBundle("com.borland.samples.dx.datasetdata.Res");

    //Construct the frame
    public ClientFrame1(String obje,String Sqltext ) {
        serverobj=obje;
        SqlText=Sqltext;
        System.out.println(obje);
          clientProvider1=    new ClientProvider(serverobj,SqlText);
          clientResolver1=    new ClientResolver(serverobj);
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try {
            jbInit();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    //Component initialization
    private void jbInit() throws Exception{
        contentPane = (JPanel) this.getContentPane();
        this.setSize(new Dimension(500, 300));
        this.setTitle(res.getString("RS_ClientTitle"));
        jPanel1.setLayout(borderLayout1);
        contentPane.add(jPanel1, BorderLayout.CENTER);
        contentPane.add(jdbStatusLabel1, BorderLayout.SOUTH);
        jPanel1.add(jdbNavToolBar1, BorderLayout.NORTH);
        jdbTable1.setDataSet(tableDataSet1);
        jPanel1.add(tableScrollPane1, BorderLayout.CENTER);
        tableScrollPane1.getViewport().add(jdbTable1);
        tableDataSet1.setProvider(clientProvider1);
        tableDataSet1.setResolver(clientResolver1);
    }

    //Overridden so we can exit when window is closed
    protected void processWindowEvent(WindowEvent e) {
        super.processWindowEvent(e);
        if (e.getID() == WindowEvent.WINDOW_CLOSING) {
            //System.exit(0);
            this.dispose();
        }
    }
}
```

Database

There are many functions. This sample program gets flight information from database based on the flight number.

```
   public ResultSet getFlightinfo(String flightnum){
       Connection conn = null;
       try {
              //conn = ConnectionFactory.getConnection();
              Class.forName("oracle.jdbc.driver.OracleDriver");
              String url="jdbc:oracle:thin:@localhost:1521:oracle9i";
              conn=DriverManager.getConnection(url,"thesis","thesis");
              Statement stmt= conn.createStatement(
                        java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE,
                              java.sql.ResultSet.CONCUR_READ_ONLY );
              String query1 = "SELECT * FROM Airline WHERE AirlineNumber ="
```

```
                                    +Integer.valueOf(flightnum) ;
                  ResultSet rs1 = stmt.executeQuery(query1);
                  if (rs1==null){
                     rs1.close();
                     return null;
                  }
                  else{
                    return rs1;
                  }
            }
            catch (Exception e) {
               System.out.println("Exception found::"+e.getMessage());
               e.printStackTrace();
            }
            System.out.println("DBmain::getFlightinfo: end");
            return null;
    }
```

Java Server Pages (JSP)

This is the example of a JSP page, which is used to process data uploaded by user and invoke a function to get information from database and transfer to the next page.

```
<%@ page session="true" %>

<%@ page contentType="text/html;charset=gb2312" %>

<%@ page import="java.util.*" %>

<jsp:useBean id="DBMainID" scope="session" class="jsps.DBMain" />

<%      session = request.getSession(true); %>

<%
```

```
  String username=null;

  username = (String)session.getAttribute("username");

  if(username==null){

            response.sendRedirect("login.jsp");

            System.out.println("session userid is null!");

  }

  String firstname=request.getParameter("firstname");

  String lastname=request.getParameter("lastname");

  String phone=request.getParameter("phone");

  String address=request.getParameter("address");

  String email=request.getParameter("email");

if(DBMainID.changeUserinfo(username,firstname,lastname,phone,address,email)!=1

){

             System.out.println("change traveler information failed");

             response.sendRedirect("travelerinfo.jsp");

             return;

  }

   String url = response.encodeURL("personalassis.jsp");

   response.sendRedirect(url);
```

```
%>
```

REFERENCES

Bergsten, Hans, *Java Server Pages*, Sebastopol, CA, O'Reilly & Associates, Inc., 2001.

Heinle, Nick and Bill Pena, *Designing with JavaScript*, Second Edition, Sebastopol, CA, O'Reilly & Associates, Inc., 2002.

Johnson, Rod, *Expert one-on-one J2EE Design and Development*, Birmingham, UK, Wrox Press Ltd, 2002.

McLanghlin, Brett, *Java & XML, 2nd Edition*, Sebastopol, CA, O'Reilly & Associates, Inc., 2001.

O'Donahue. John, *Java Database Programming Bible*, New York, Wiley Publishing, Inc., 2002.

Walsh, Aaron, Justin Couch and Daniel H. Steinberg, *Java 2 Bible*, Foster City, CA, IDG Books Worldwide, Inc., 2000.