

Mobility Group Task 1:

Task 1: Mobile e-commerce system design and implementation Report

Team members:

Xing Wei

Rui Sun

08 July, 2011

Contents

1 Project Overview	3
2 Interface Design	3
Screen #1 – Main page	3
Screen #2 – Select Date	3
Screen #3 – Start Searching	4
Screen #4 – Display search result	4
Screen #5 – Show Detail	5
3 Database Design	6
4 J2ME Code	7
5 PHP Script Code	14

1 Project Overview

The objective of this project is to develop an interactive mobile application, called “Mobile e-ticket service for a Airline”. The system allows users to search Flight in the database according to the date they select. Also, the Flight detail will be sent back to the user`s mobile after they select a particular Flight. The project will use J2ME, PHP and Mysql.

2 Interface Design

This section introduce the designing of the user interface.

Screen #1 – Main page

Description

After launching the application, the user will see this page. This page will enable the user to choose a date for the flight; the default is set to current system day.

GUI Design

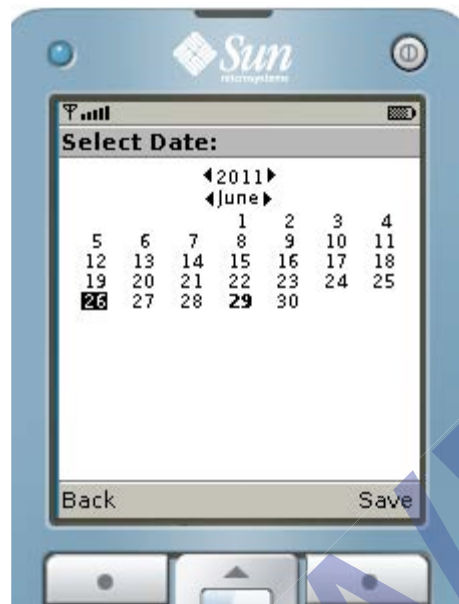


Screen #2 – Select Date

Description

This screen displays a schedule form. The date field will display a calendar, where the user can easily select a new date.

GUI Design



Screen #3 – Start Searching

Description

After users select a date for searching flights, the application will return to the main page, users can press the GetFlightList button to start searching.

GUI Design

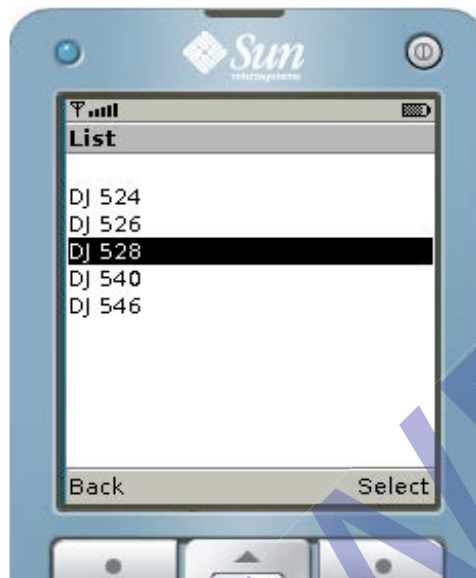


Screen #4 – Display search result

Description

This form will display search result of the flight list based on the date that the user selects before.

GUI Design



Screen #5 – Show Detail

Description

This form is displayed when the user selects the 'Select' option in the previous form. It simply displays all the flight details for the selected flight. The user can then see the detail of the flight, which includes the date, flight No., departure and arrival time, seat state, and price.

GUI Design



3 Database Design

The database used is MySQL. The database has been called airline and consists of an “ALGS” table. The structure of both tables and some sample data are showed below.

Database Structure

```
Create database airline;
use airline;
// the ALGS is a airline from Gold Coast to Sydney
create table ALGS(

    Id int    AUTO_INCREMENT NOT NULL,
    Departs_time time,
    Arrives_time time,
    Flight varchar(10),
    State tinyint(1),
    Price float(5,2)
    Date date,
    PRIMARY KEY(Id)

);
```

```
mysql> describe ALGS;
```

Field	Type	Null	Key	Default	Extra
Id	int(11)	NO	PRI	0	
Departs_time	time	YES		NULL	
Arrives_time	time	YES		NULL	
Flight	varchar(10)	YES		NULL	
State	tinyint(1)	YES		NULL	
Price	float(5,2)	YES		NULL	
Date	date	YES		NULL	

```
7 rows in set (0.00 sec)
```

Sample Data

```
INSERT INTO ALGS VALUES (1,'06:00:00','07:30:00','DJ 500',0,75.00,'2011-06-25'),
(2,'09:05:00','10:35:00','DJ 512',1,119.00,'2011-06-25'),
(3,'11:30:00','13:00:00','JQ 409',1,127.00,'2011-06-25'),
(4,'17:35:00','19:00:00','TT8763',1,123.35,'2011-06-25'),
(5,'12:30:00','14:00:00','DJ 524',1,315.00,'2011-06-26'),
(6,'13:55:00','15:25:00','DJ 526',0,175.00,'2011-06-26'),
(7,'15:00:00','16:30:00','DJ 528',0,75.00,'2011-06-26'),
(8,'18:00:00','19:30:00','DJ 540',1,175.00,'2011-06-26'),
(9,'20:00:00','21:30:00','DJ 546',0,75.00,'2011-06-26'),
(12,'07:00:00','08:30:00','DJ 504',1,75.00,'2011-06-29'),
(13,'10:20:00','11:50:00','DJ 516',1,175.00,'2011-06-29'),
(14,'13:00:00','14:30:00','DJ 524',1,315.00,'2011-06-29'),
(15,'14:00:00','15:30:00','DJ 526',0,75.00,'2011-06-29'),
(16,'15:00:00','14:30:00','DJ 528',1,75.00,'2011-06-29'),
(17,'17:00:00','18:30:00','DJ 528',0,175.00,'2011-06-29'),
(18,'22:00:00','23:30:00','DJ 546',1,75.00,'2011-06-29');
```

```
mysql> select * from ALGS;
```

Id	Departs_time	Arrives_time	Flight	State	Price	Date
1	06:00:00	07:30:00	DJ 500	0	75.00	2011-06-25
2	09:05:00	10:35:00	DJ 512	1	119.00	2011-06-25
3	11:30:00	13:00:00	JQ 409	1	127.00	2011-06-25
4	17:35:00	19:00:00	TT8763	1	123.35	2011-06-25
5	12:30:00	14:00:00	DJ 524	1	315.00	2011-06-26
6	13:55:00	15:25:00	DJ 526	0	175.00	2011-06-26
7	15:00:00	16:30:00	DJ 528	0	75.00	2011-06-26
8	18:00:00	19:30:00	DJ 540	1	175.00	2011-06-26
9	20:00:00	21:30:00	DJ 546	0	75.00	2011-06-26
12	07:00:00	08:30:00	DJ 504	1	75.00	2011-06-29
13	10:20:00	11:50:00	DJ 516	1	175.00	2011-06-29
14	13:00:00	14:30:00	DJ 524	1	315.00	2011-06-29
15	14:00:00	15:30:00	DJ 526	0	75.00	2011-06-29
16	15:00:00	14:30:00	DJ 528	1	75.00	2011-06-29
17	17:00:00	18:30:00	DJ 528	0	175.00	2011-06-29
18	22:00:00	23:30:00	DJ 546	1	75.00	2011-06-29

```
16 rows in set (0.00 sec)
```

```
//airline.sql is a database file of MySQL
```

```
//import database
```

```
//go to airline.sql folder
```

```
sudo cp airline.sql /var/lib/mysql
```

```
sudo chmod 660 /var/lib/mysql/airline.sql
```

```
mysql -u root -p
```

```
mysql>create database airline;
```

```
mysql>use airline;
```

```
mysql>mysql -u root -p airline < airline.sql;
```

```
//export database R is export all value from table not just scheme of database
```

```
cd /usr/bin
```

```
mysqldump -u root -p airline -R > /path/airline.sql
```

4 J2ME Code

```
//AirLine Class
```

```
//this class is use Web Server, HTTPconnect to PHP script test and return data from database airline;
```

```
//First list all Flight of Date which use select, then show the detail of Flight No. which is particular choice.
```

```
import javax.microedition.midlet.*;
```

```
import javax.microedition.lcdui.*;
```

```
import java.io.*;
```

```
import javax.microedition.io.*;
```

```
import java.util.*;
```

```

// SearchResult control class to show the result
// to create an SearchResult object, it take two arguments
// a AirLine, and a string to show
class SearchResult extends TextBox implements CommandListener
{
    static final Command okCommand = new Command ("OK", Command.BACK, 1);

    AirLine al;

    // passing two arguments
    // the AirLine object and string to show
    public SearchResult (AirLine al, String s)
    {
        super ("Enquired Results", s, 1024, 0);
        this.al = al;

        addCommand (okCommand);
        setCommandListener (this);
    }

    public void commandAction (Command c, Displayable d){
        if(c == okCommand){
            al.back();
        }
    }
}

public class AirLine extends MIDlet implements CommandListener
{
    //Display
    Display display;

    // Departures
    TextField From;
    // Arrives
    TextField To;
    // exit button
    static final Command exitCommand = new Command("Exit",Command.EXIT, 1);
    // going back button
    static final Command backCommand = new Command("Back",Command.BACK, 0);
    // next button, searching Flight list basing on the date given.
    static final Command nextCommand = new Command("GetFlightList",
Command.SCREEN,2);
    // select button, click to show all the detail for the Flight No.
    static final Command selectCommand = new Command( "Select", Command.OK, 1 );

    Form screen;

    // search date field
    public DateField date = new DateField("Select Date:", DateField.DATE);

    // url address for the server script
    String url;
    // type to show the different steps of searching.

```



```

// value 0: the first step to search Flight list. 1: searching detail Flight
No.
int type;
// list of result Flight
List ls;

public AirLine()
{
    //UI display
    Ticker TickerTitle = new Ticker("Welcome to Airline Centre");

    // Setup Display
    display = Display.getDisplay(this);

    //airline information

    From = new TextField("AirLine \n From:", "Gold Coast", 256,
TextField.UNEDITABLE);
    To = new TextField("To:", "Sydeny", 256, TextField.UNEDITABLE);
    screen = new Form("Mobile e-ticket service");
    // date
    java.util.Date now = new java.util.Date();
    // set default date to current
    date.setDate(now);

    //show the components
    screen.setTicker(TickerTitle);
    screen.append(From);
    screen.append(To);
    screen.append(date);
    screen.addCommand(exitCommand);
    screen.addCommand(nextCommand);
    screen.setCommandListener(this);
}

public void exitMIDlet(){
    destroyApp(false);
    notifyDestroyed();
}

public void startApp()
{
    //Show the Window Form on the Screen
    display.setCurrent(screen);
}

public void pauseApp(){}

public void destroyApp(boolean unconditional){}

public void back () {display.setCurrent(screen);}

// getFlightList display the Flight lists,
// giving a string return from the server
// it contains the Flight list connecting with a special string, like '-'.

```

```

public void getFlightList(String str){
    // show Flight list if it's not empty
    if (!str.equals("")) {

        // string from the server.
        String toSplit = str;
        // delim has be same as server script
        char delim = '-';
        // string list to store flight list returing from the server
        String[] items = split(toSplit, delim, false);
        // list component to keep Flight list
        ls = new List("List", List.IMPLICIT, items, null);
        ls.addCommand( backCommand );
        ls.addCommand( selectCommand );
        ls.setCommandListener( this );
        // dispay the Flight lists
        display.setCurrent(ls);
    }else
    {
        showDetail("There is not Result for your searching condition!");
    }
}

// show the detail of selecting Flight No.
// giving a string result from the server
public void showDetail(String str){
    SearchResult sr = new SearchResult(this, str);
    display.setCurrent(sr);
}

// method to call the server script and return the result from the server
public void find_action()
{
    // Create a thread to actually connect to the webserver
    Thread t = new Thread()
    {
        public void run()
        {
            //open connection
            try
            {
                // return the string from the server script
                String str = search(url);
                System.out.print("\n" + url + "\n");
                //searching for the flight list
                if(type==0) {
                    getFlightList(str);
                }else //searching for the detail of Flight No.
                {
                    showDetail(str);
                }
            }
            catch(IOException e)
            {
            }
        }
    }
}

```

```

        }
    }
};

t.start();
}

// search method to call the server script
// it takes an argument which is the script address
// and return the result from the server
String search(String url) throws IOException {
    HttpURLConnection connection = null;
    InputStream is = null;
    OutputStream os = null;
    StringBuffer stringBuffer = new StringBuffer();

    try {
        // create the connect
        connection = (HttpURLConnection)Connector.open(url);
        connection.setRequestMethod(HttpURLConnection.GET);
        connection.setRequestProperty("IF-Modified-Since","20 Jan 2001
16:19:14 GMT");
        connection.setRequestProperty("User-Agent","Profile/MIDP-2.0
Confirguration/CLDC-1.0");
        connection.setRequestProperty("Content-Language", "en-CA");
        connection.setRequestProperty("Content-Type", "application/x-www-
form-urlencoded");
        os = connection.openOutputStream();
        is = connection.openDataInputStream();

        int ch;
        while ((ch = is.read()) != -1) {
            stringBuffer.append((char) ch);
        }
        // return the result
        return stringBuffer.toString();
    } finally {
        if(is!= null) {
            is.close();
        }

        if(os != null) {
            os.close();
        }
        // close the connection
        if(connection != null) {
            connection.close();
        }
    }
}

public void commandAction(Command c, Displayable s)
{
    // server script address
    url = "http://127.0.0.1/test.php";
    // searching for the Flight list

```

```

        if(c==nextCommand)
        {
            type = 0;
            url = generateGetUrl(url, date.getDate(), "");
            find_action();
        }

        // searching for the detail time schedule, giving a particular Flight
        if(c==selectCommand)
        {
            type = 1;
            url = generateGetUrl(url, date.getDate(),
ls.getString(ls.getSelectedIndex()));
            find_action();
        }

        // go back last step
        if(c==backCommand)
        {
            this.back();
        }

        // exit
        if(c==exitCommand)
        {
            exitMIDlet();
        }
    }

    // Function that formats the url string coding (IE %20 for spaces)
    // it takes three arguments, url, a date, and searching key value
    public String generateGetUrl(String url, Date da, String k)
    {
        String newUrl = new String();

        StringBuffer c = new StringBuffer("?d=" + da + "&k=" + k);

        for(int i=0; i<c.length(); i++)
        {
            if(c.charAt(i)==' ')
            {
                c = c.insert(i,"%20");
                c = c.deleteCharAt(i+3);
            }
        }
        newUrl = url + c.toString();

        return newUrl.toString();
    }

    // function to split a string to a string list
    // it takes 3 arguments, toSplit: string for splitting,
    // delim: speccial char, like '-' using here. ignoreEmpty: boolean argument
    // return a string list

```

```

public static String[] split(String toSplit,char delim,boolean ignoreEmpty)
{
    StringBuffer buffer = new StringBuffer();
    Stack stringStack = new Stack();
    for(int i = 0;i<toSplit.length();i++)
    {
        if(toSplit.charAt(i) != delim)
        {
            buffer.append((char)toSplit.charAt(i));
        }
        else
        {
            if(buffer.toString().trim().length() == 0 && ignoreEmpty)
            {
            }
            else
            {
                stringStack.addElement(buffer.toString());
            }
            buffer = new StringBuffer();
        }
    }

    if(buffer.length() != 0)
        stringStack.addElement(buffer.toString());

    String [] split = new String[stringStack.size()];
    for(int i = 0;i<split.length;i++)
    {
        split[split.length-1-i] = (String)stringStack.pop();
    }

    stringStack = null;
    buffer = null;
    // return the split string list
    return split;
}

```

```

}

```

5 PHP Script Code

```
<?php
//test.php
//get value from url and search by date and Flight NO. to get a line date from
database airline table ALGS

//config
$username = "root";
$password = "password";
$hostname = "127.0.0.1";
//temp for replace '%20' in $_GET["d"]
//$temp = "";

// get query from user

// date value
$q_date = "";
// key value for Flight No.
$q_FNo = "";
// query
$q = "";

if (isset($_GET)) {
    // $temp = $_GET["d"];
    // $temp = preg_replace("/%20/", " ", $temp);
    // $q_date = ($temp != "" ? date("Y-m-d", strtotime($temp)) : date("Y-m-d
H:i")); //date
    $q_date = ($_GET["d"] != "" ? date("Y-m-d", strtotime($_GET["d"])) : date("Y-m-d
H:i")); //date, if it not works please // this line and use above 3 lines
    $q_FNo = $_GET["k"]; //keyword
}

// if there is Flight No. key value, it will search the flight available time
schedule
// giving the date and the Flight No.
// table ALGS(Airline Gold Coast to Sydney) from database airline
if ($q_FNo != "") {
    $q = "SELECT Id, date_format(Departs_time, '%l:%i %p') as DTime,
date_format(Arrives_time, '%l:%i %p') as ATime,
Flight, State, Price, date_format(Date, '%W, %Y %M %d') as FDate FROM ALGS
where date_format(Date, '%Y-%m-%d') = '". $q_date.'" and Flight = '". $q_FNo.'" Order
by Flight, Date Asc ";

}

// search for the Flight list giving the date
else
{
    $q = "SELECT distinct Flight FROM ALGS where date_format(Date, '%Y-%m-%d') =
'". $q_date.'" Order by Flight Asc ";
}
```

```

//connection to the database
$dbhandle = mysql_connect($hostname, $username, $password)
or die("Unable to connect to MySQL");

//select a database airline to work with
$selected = mysql_select_db("airline",$dbhandle)
or die("Could not select airline");

//execute the SQL query and return records
$result = mysql_query($q);

//fetch the data from the database
//format
//Date: Friday, 22 July 2011
//Flight No. QL 789
//Departure from Gold Coast: 8:00 am
//Arrival at Sydney: 9:30 am
//State: Available
//Price: $95
while ($row = mysql_fetch_array($result)) {
    // return the Flight detail if no Flight NO key value
    if ($q_FNo != ""){
        echo $row{'FDate'}."\n". //display the results
        "Flight No.: ".$row{'Flight'}."\n".
        "Departure from Gold Coast: ".$row{'DTime'}."\n".
        "Arrive at Sydney: ".$row{'ATime'}."\n".
        "State:".($row{'State'} ? "Available" : "Full")."\n".
        "Price: ".$row{'Price'}.".
        "\n\n";
    }
    else // return the Flight list
    {
        echo $row{'Flight'}."-";
    }
}

//close the connection
mysql_close($dbhandle);
?>

```