



**SCHOOL OF ENGINEERING AND BUILT ENVIRONMENT**

**Introduction to Mobile Device Programming**

M3G621212-14-B: 14/15 B

**Coursework Assignment**

Report

Zoltan Tompa

*S1112414*

## Table of Contents

Notes for testing the application.....	3
Design Report .....	4
Testing Report .....	5
Ambient Mode Report.....	6
Program Code.....	7
MySimpleArrayAdapter.java .....	9
UpdaterThread.java.....	11
CarParkListingTestActivity.java .....	19
testCarpark.java .....	28
row.xml.....	29
Main.xml (port) .....	29
Main.xml (land) .....	33
Strings.xml .....	37
Appendix.....	38
Screenshots .....	38
Class Diagram .....	40
References.....	41

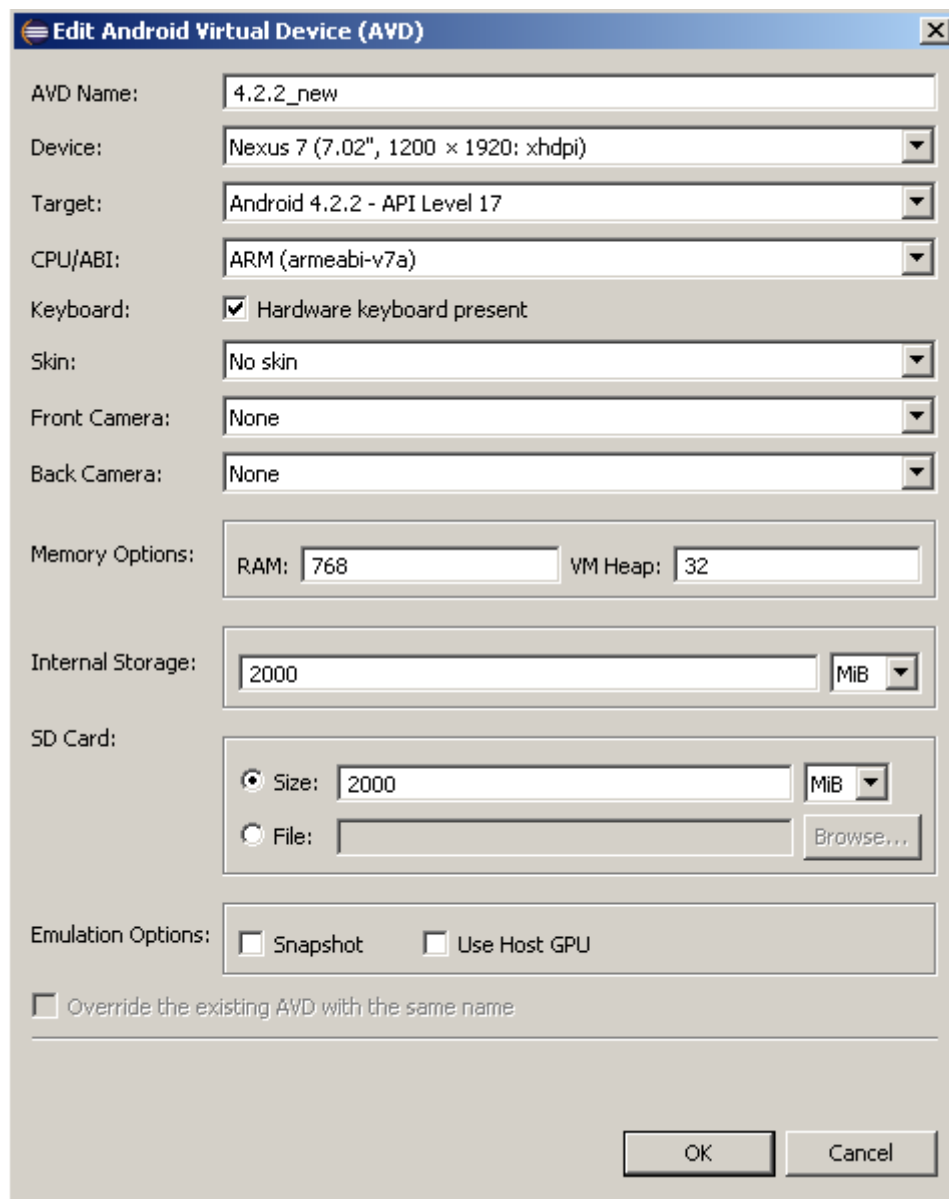
## Notes for testing the application

When developing the application I've used an emulator with the settings below. Using a different screen/resolution can cause the app to get displayed differently, so please use these settings for optimal experience.

Also I've encountered a small error, where the IDE doesn't allow to run the solution as my landscape layout requires API14 features, but the minimum API level in the Manifest file had to be locked to 6, as anything above that would cause the supplied string reader to malfunction.

If this error comes up, simply clean the build and run afterwards.

This is a known bug and will be addressed in later development prior release.



/ fig. 0 – emulator settings /

## Design Report

This report is a quick overview, outlining the main features of the application.

The app when first started displays a loading text (*fig1*), till the initialisation and the reading of the first set of data finishes.

Once the app loads, the notification disappears and the data gets displayed on the user's screen, containing the city-wide statistical figures and a list of available car parks in the city.

Next to the car parks' name a coloured dot is displayed, indicating the number of available spaces. (*fig2*) Green dot appears when more than 60% of the spaces in that car park are available, an orange if it's between 60 and 80% and red if the taken spaces are above 80%.

When the user selects a car park from the list, the screen switches, and the details screen appears, displaying relevant data about the selected car park. The occupancy rate is expressed by a number here (a percentage, rounded to two decimal places) and also visualised with an appropriate pie-chart. (*fig3*)

The application is designed to use two independent lists to store objects (an idea borrowed from computer graphics rendering practices, where components are rendered to a temporary buffer before displaying), so the user can use the application even in the middle of an update process, even though the updater process was designed to effectively reconstruct nearly all the displayed data on the screen. This decision was made to ensure that the app can cope with changes like car park gets taken off the list, or a new one gets added. Normally this process would make the application vulnerable to Null Point Exceptions during the update, but with the implementation of the double buffer mentioned above, this issue was successfully addressed.

The app automatically updates the displayed information in the background (using a background-thread) from the Glasgow City website, in every 70 seconds, to comply with the fair usage policy. This way when the application updates the car park information it doesn't "hang up", the update occurs seamlessly and the user doesn't even notice it.

The application also takes advantage of the underlying Android system's capability to handle different layouts in portrait and landscape device orientations. The app implements a landscape view designed to display information more efficiently in horizontal orientation.

In this setting, the list of car parks is displayed on the left and the details of the selected item on the right, eliminating the need of switching screens inside the application. (*fig4*)

Visually both approaches use a colour-palette that tries to incorporate the colours of the official website of the city.

## Testing Report

During the development of the project, as a general habit I was heavily relying on the IDE's built-in debug console feature, LogCat. As it can be seen in the code files, most of the critical areas have debug comment entries before and after them. This way it can be easily tracked if the part of the code run, or throw the application over.

Generally I have used two level of the Log. Feature, the warning for "informative" and "error" for the more curtail, progress-tracking purposes. However this can really quickly flood the console window, so nowadays I'm starting off my declaring a Boolean variable called "isDebugModeON" in all of my code, which essentially controls the output of the debug messages, by encapsulating the Log message in an if statement, as follows;

```
if (isDebugModeON) Log.w("Tom - report", "appstart test");
```

Appropriate testing of the software was also carried out on the storage class's functions, which handles the individual carpark instances. Using Whitebox testing I've implemented a test class, which evaluates the functions by supplying them with input values and checks them against the expected (pre-calculated and checked) outputs.

When designing the input/output values, since I knew the structure of the function being tested, I've targeted the potentially vulnerable areas, such as division by zero, checked that the iterations run on the whole array in all cases and also what happens if the supplied input data is "not valid", for example where the function used to crop off the last seven digits of the supplied string (to format the carpark name from the input stream) gets a string less than seven characters.

The test cases mentioned above are extensive, meaning that in normal situations cases like having zero total spaces in a carpark (in which case it's not a carpark ...) or having the names with the ending codes changed are really unlikely, but testing even for these assures the application can handle a variety of unexpected situations.

## Ambient Mode Report

This application uses colours and shapes to visualise numerical data for better and easier usability. I personally didn't find the idea of using different sound effects in my app appealing, as for me it would be more annoying.

After the application starts up and reads the necessary data, next to the name of the carpark a coloured dot appears. The colours chosen are already widely used in different applications to indicate levels, amounts and availability, making it almost intuitive for the user to interpret the information displayed.

Green dot appears when more than 60% of the spaces in that carpark are available, an orange if it's between 60 and 80% and red if the taken spaces are above 80%.

When a carpark is selected (either in portrait or landscape mode) along with the details of the selected carpark also a pie-chart appears, visualising the ratio of the available and taken places. The colour choice here helps the user quickly read the data, as the taken spaces are somewhat faded into the background, and the available spaces are highlighted and thus stand out from the background. Using a pie-chart, users can perceive the data just by looking at the shape of the diagram, without necessarily looking at the exact numbers.

## Program Code

carpark.java

```
/**
 * @author Zoltan Tompa - S1112414
 */
package org.me.myandroidstuff;
import java.lang.Math;

public class carPark {

    private String id;
    private String name;
    private int availableSpaces;
    private int totalSpaces;
    private int takenSpaces;
    private String status;

    public carPark() {
        // TODO Auto-generated constructor stub
    }

    //method to do in-class occupancy calculation
    public double calcOccupancy()
    {
        if (totalSpaces > 0.0)
        {
            double occu = (((double)takenSpaces) / totalSpaces); //getting
a percentage (0-1)
            occu = Math.round(occu *100*100.0)/100.0; //making it a
percentage 0-100 and rounding it to two decimals
            return occu;
        }
        else
            return 0.0;
    }

    //method to do general occupancy calculation
    public static double calcOccupancy(int taken, int total)
    {
        if (total > 0.0)
        {
            double occu = (((double)taken) / total); //getting a
percentage (0-1)
            occu = Math.round(occu *100*100.0)/100.0; //making it a
percentage 0-100 and rounding it to two decimals
            return occu;
        }
        else
            return 0.0;
    }

    public void trimName()
    {
        if(name.length() > 7)
        {
            name = name.substring(0, name.length()-7);
        }
    }
}
```

```

    }
}

/**
 * @return the id
 */
public String getId() {
    return id;
}

/**
 * @param id the id to set
 */
public void setId(String id) {
    this.id = id;
}

/**
 * @return the name
 */
public String getName() {
    return name;
}

/**
 * @param name the name to set
 */
public void setName(String name) {
    this.name = name;
}

/**
 * @return the totalSpaces
 */
public int getTotalSpaces() {
    return totalSpaces;
}

/**
 * @param totalSpaces the totalSpaces to set
 */
public void setTotalSpaces(int totalSpaces) {
    this.totalSpaces = totalSpaces;
}

/**
 * @return the takenSpaces
 */
public int getTakenSpaces() {
    return takenSpaces;
}

/**

```



```

        * @return the availableSpaces
        */
        public int getAvailableSpaces() {
            return availableSpaces;
        }

        /**
         * @param takenSpaces the takenSpaces to set
         */
        public void setTakenSpaces(int takenSpaces) {
            this.takenSpaces = takenSpaces;
        }

        /**
         * @param availableSpaces the available to set
         */
        public void setAvailableSpaces(int available) {
            this.availableSpaces = available;
        }

        /**
         * @return the status
         */
        public String getStatus() {
            return status;
        }

        /**
         * @param status the status to set
         */
        public void setStatus(String status) {
            this.status = status;
        }
    }
}

```

## MySimpleArrayAdapter.java

```

/**
 * @author Zoltan Tompa - S1112414
 * reference: http://www.vogella.com/tutorials/AndroidListView/article.html
 */

package org.me.myandroidstuff;

import java.util.List;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

```

```

import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class MySimpleArrayAdapter extends ArrayAdapter<String> {
    private final Context context;
    private final List<String> names;
    private final List<Integer> values;

    public MySimpleArrayAdapter(Context context, List<String> names,
List<Integer> values) {
        super(context, R.layout.row, names);
        this.context = context;
        this.names = names;
        this.values = values;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View rowView = inflater.inflate(R.layout.row, parent, false);
        TextView textView = (TextView) rowView.findViewById(R.id.listText);
        ImageView imageView = (ImageView) rowView.findViewById(R.id.icon);
        textView.setText(names.get(position));
        // change the icon for Windows and iPhone
        int s = values.get(position);
        if (s == 0) {
            imageView.setImageResource(R.drawable.red);
        }

        else if (s==1){
            imageView.setImageResource(R.drawable.orange);
        }
        else

```

```

        {
            imageView.setImageResource(R.drawable.green);
        }

        return rowView;
    }
}

```

## UpdaterThread.java

```

/**
 * @author Zoltan Tompa - S1112414
 * reference: TimerProjectActivity
 */

package org.me.myandroidstuff;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.StringReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;

import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;

public class UpdaterThread extends Thread{

    private boolean updateRunning;    // The "updateRunning" status of the
    counter

    //private String time;

```

```

        private Handler parentHandler; // The handler class to deal with
interaction with the parent

        private boolean isDebugModeON;
        private String result; //string to hold xml data

        //other variables
        //private String sourceListingURL = "http://tomzoy.me/tmp/parking.xml";
        private String sourceListingURL =
"http://open.glasgow.gov.uk/api/live/parking.php?type=xml";

        private int updateIntervall = 70000; //should be: 70000 for 70sec

        private int currentCarParkList;

        public UpdaterThread() {

            updateRunning = false;

        }

        public UpdaterThread(Handler parentHandler) {

            this.parentHandler = parentHandler;
            updateRunning = false;

        }

        //*****

        private Handler myThreadHandler = new Handler()
        {
            // Handle messages from the parent
            public void handleMessage(Message msg)
            {

                if (msg.what == 0)
                {
                    updateRunning = true;
                }

                else
                {
                    updateRunning = false;
                }
            }
        }
    }

```

```

        }
    }
};

public Handler getHandler()
{
    return myThreadHandler;
}

//
// This is where the main work of the class is done
//
public void run()
{

    super.run();

    isDebugModeON = true;
    currentCarParkList = 0;

    try
    {
        while(true)
        {

            readData();

            if (updateRunning == true)
            {
                Log.e("Z","readDataStarted");
                readData();
                Log.e("Z","readData DONE");
            }
            else
            {
                Log.e("Z"," else path reached");
                Log.e("Z"," M 1");
                Message messageToParent = new Message();
                Log.e("Z"," M 2");
                Bundle messageData = new Bundle();
                Log.e("Z"," M 3");
                messageToParent.what = 0;
                Log.e("Z"," M 4");
                messageData.putInt("currentCarParkList",
currentCarParkList);

                // Package Data and send it

```

```

        messageToParent.setData(messageData);
        Log.e("Z", " M 5");
        parentHandler.sendMessage(messageToParent);
        Log.e("Z", " M 6");
    }

    currentCarParkList ++;
    if (currentCarParkList ==2) {currentCarParkList=0;}
    sleep(updateIntervall);
} // End of while
}
catch (Exception ae)
{
    Log.e("MyTag","Time loop exception");
}

} // End of run

//-----
-----

private void readData()
{
    try{
        // Get the data from the RSS stream as a string
        if (isDebugModeON)Log.w("Tom - report","start reading data");
        result = sourceListingString(sourceListingURL);

        if (isDebugModeON)Log.w("Tom - report","point 1");

        // start parsing data
        XmlPullParserFactory pullParserFactory;
        try {
            pullParserFactory = XmlPullParserFactory.newInstance();
            XmlPullParser parser = pullParserFactory.newPullParser();

            parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES,
false);

            parser.setInput(new StringReader ( result ));

            if (isDebugModeON)Log.w("Tom - report","before parser");

            parseXML(parser);
            if (isDebugModeON)Log.w("Tom - report","after parser");

```

```

        } catch (XmlPullParserException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    if (isDebugModeON)Log.w("Tom - report","parser run with NO
exception");

    if (isDebugModeON){
        /*
        Log.e("Tom - report -List","Total carparks: "+
Integer.toString(CarParkListingTestActivity.carParkList1.size()));
        Log.e("Tom - report -List","Total Occupied Spaces: "+
Integer.toString(CarParkListingTestActivity.calc_global_OccupiedPlaces()));
        Log.e("Tom - report -List","Total Spaces: "+
Integer.toString(CarParkListingTestActivity.calc_global_TotalPlaces()));

        Log.e("Tom - report -List","Total Occupancy: "+
Double.toString(carPark.calcOccupancy(CarParkListingTestActivity.calc_global_Occup
iedPlaces(), calc_global_TotalPlaces())));
        */
    }
}
catch(IOException ae)
{
    //TODO make this a pop-up

}

updateRunning = false;

}

// Method to handle the reading of the data from the XML stream
private static String sourceListingString(String urlString)throws IOException
{
    String result = "";
    InputStream anInStream = null;
    int response = -1;
    URL url = new URL(urlString);
    URLConnection conn = url.openConnection();

    Log.w("Tom - report","con 1");

```

```

// Check that the connection can be opened
if (!(conn instanceof HttpURLConnection))
{
    Log.w("Tom - report", "con 2");
    throw new IOException("Not an HTTP connection");
}
try
{
    // Open connection
    Log.w("Tom - report", "con 3");
    HttpURLConnection httpConn = (HttpURLConnection) conn;
    Log.w("Tom - report", "con 4");
    httpConn.setAllowUserInteraction(false);
    httpConn.setInstanceFollowRedirects(true);
    httpConn.setRequestProperty("Accept-Charset", "UTF-8");
    Log.w("Tom - report", "con 5");
    httpConn.setRequestMethod("GET");
    Log.w("Tom - report", "con 6");
    httpConn.connect();
    Log.w("Tom - report", "con 7");
    response = httpConn.getResponseCode();
    Log.w("Tom - report", "con 8");
    // Check that connection is Ok
    boolean needLine = false;
    Log.w("Tom - report", "con 9");
    if (response == HttpURLConnection.HTTP_OK)
    {
        // Connection is Ok so open a reader
        anInStream = httpConn.getInputStream();
        InputStreamReader in= new InputStreamReader(anInStream, "UTF-
8");

        BufferedReader bin= new BufferedReader(in);

        // Read in the data from the XML stream
        String line = new String();

        while (( (line = bin.readLine())) != null)
        {
            if (needLine == true)
                result = result + "\r\n" + line;
            needLine = true;
        }
    }
}
catch (Exception ex)
{
    throw new IOException("Error connecting");
}

```



```

        // Return result as a string for further processing
        return result;

    }

    // Method for parsing the XML data to class instances
    private void parseXML(XmlPullParser parser) throws
    XmlPullParserException, IOException
    {

        if (isDebugModeON) Log.w("Tom - report", "point 3");

        boolean isCarParkIdentityNext = false;
        boolean iscarParkStatusNext = false;
        boolean isoccupiedSpacesNext = false;
        boolean istotalCapacityNext = false;

        String Pname = null;
        int totalSpaces = 0;
        int takenSpaces = 0;
        String status = null;

        int eventType = parser.getEventType();

        CarParkListingTestActivity.carParkListArray.get(currentCarParkList).clear();

        while (eventType != XmlPullParser.END_DOCUMENT) {
            if(eventType == XmlPullParser.START_DOCUMENT) {

                if (isDebugModeON) Log.w("Tom - report", "Start document");

            } else if(eventType == XmlPullParser.START_TAG) {

                if (isDebugModeON) Log.w("Tom - report", "Start tag:
                "+parser.getName());

                if (parser.getName().contains("carParkIdentity"))
                {
                    isCarParkIdentityNext = true; //trigger data filed "pull"
                }
            }
        }
    }

```

```

        else if (parser.getName().contains("carParkStatus")) {
            iscarParkStatusNext = true; } //trigger data filed "pull"
        else if (parser.getName().contains("occupiedSpaces")) {
            isoccupiedSpacesNext = true; } //trigger data filed "pull"
        else if (parser.getName().contains("totalCapacity")) {
            istotalCapacityNext = true; }; //trigger data filed "pull"

    } else if(eventType == XmlPullParser.END_TAG) {

        if (isDebugModeON) Log.w("Tom - report","End tag:
"+parser.getName());

    } else if(eventType == XmlPullParser.TEXT) {

        if (isDebugModeON) Log.w("Tom - report","data: "+parser.getText());

        if (isCarParkIdentityNext == true)
        {
            Log.e("Tom - report","the stuff that I
need:"+parser.getText());
            Pname = parser.getText();
            isCarParkIdentityNext = false;
        }

        else if (iscarParkStatusNext == true)
        {
            Log.e("Tom - report","the stuff that I
need:"+parser.getText());
            status = parser.getText();
            iscarParkStatusNext = false;
        }

        else if (isoccupiedSpacesNext == true)
        {
            Log.e("Tom - report","the stuff that I
need:"+parser.getText());
            takenSpaces = Integer.parseInt(parser.getText());
            isoccupiedSpacesNext = false;
        }

        else if (istotalCapacityNext == true)
        {
            Log.e("Tom - report","the stuff that I
need:"+parser.getText());

```

```

        totalSpaces = Integer.parseInt(parser.getText());
        istotalCapacityNext = false;

        // construct new object
        carPark currentCarPark;
        currentCarPark = new carPark();
        currentCarPark.setName(Pname);
        currentCarPark.trimName();
        currentCarPark.setStatus(status);
        currentCarPark.setTakenSpaces(takenSpaces);
        currentCarPark.setTotalSpaces(totalSpaces);
        currentCarPark.setavailableSpaces((totalSpaces-takenSpaces));

        CarParkListingTestActivity.carParkListArray.get(currentCarParkList).add(currentCarPark);

        if (isDebugModeON) Log.e("Tom - report-names","from OBJECT:
"+currentCarPark.getName());
        if (isDebugModeON) Log.e("Tom - report-occu","occu:
"+currentCarPark.calcOccupancy());
    };

    }
    eventType = parser.next();
    if (isDebugModeON) Log.i("Tom - report","jumpedToNext");
}
}
}
}

```

#### CarParkListingTestActivity.java

```

/**
 * @author Zoltan Tompa - S1112414
 */
package org.me.myandroidstuff;

import java.util.ArrayList;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.res.Configuration;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;

```

```

import android.webkit.WebView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewSwitcher;

@SuppressLint("NewApi")
public class CarParkListingTestActivity extends Activity implements
OnItemClickListener
{
    //variables bound to GUI
    private Button backButton;
    //private TextView response;
    private TextView errorText;
    private TextView textLoading;
    private TextView textViewTotalOccu;
    private TextView textViewTotalTotal;
    private TextView textViewTotalOccup;
    private TextView textViewS2Name;
    private TextView textViewAvaliable;
    private TextView textViewS2Spaces;
    private TextView textViewS2Occup;
    private TextView textViewS2Status;
    private TextView textViewS2Ava;
    private WebView pieChartWeb;

    private ViewSwitcher avw;

    public static ArrayList<carPark> carParkList1 = new ArrayList<carPark>();
    // a list to store the carPark instances 1
    public static ArrayList<carPark> carParkList2 = new ArrayList<carPark>(); // a
list to store the carPark instances 1

    public static ArrayList<ArrayList<carPark>> carParkListArray = new
ArrayList<ArrayList<carPark>>();
    public int currentCarParkList;

    private UpdaterThread myThread;

    private boolean isDebugModeON = true;

    /** Called when the activity is first created. */
    @Override

```

```

public void onCreate(Bundle savedInstanceState)
{

    if (isDebugModeON) Log.w("Tom - report","appstart test");

    super.onCreate(savedInstanceState);

        //init containers
        init();

        //read data
        startUpdater();

} // End of onCreate

private void init()
{
    //myThread = new UpdaterThread(mainHandler);
    //myThread.start();

    setContentView(R.layout.main);

    //set up viewSwitcher
    avw = (ViewSwitcher) findViewById(R.id.vwSwitch);

    // Get the View object on which to display the results
    //TODO sort this out
    errorText = (TextView)findViewById(R.id.author);

    textViewTotalOccu = (TextView)findViewById(R.id.textViewTotalOccu);
    textViewTotalTotal = (TextView)findViewById(R.id.textViewTotalTotal);
    textViewTotalOccup = (TextView)findViewById(R.id.textViewTotalOccup);

    textViewS2Name = (TextView)findViewById(R.id.Parkname);
    textViewAvaliable = (TextView)findViewById(R.id.TextViewAvaliable);

    textViewS2Spaces = (TextView)findViewById(R.id.S2Spaces);
    textViewS2Ava = (TextView)findViewById(R.id.S2Ava);
    textViewS2Occup = (TextView)findViewById(R.id.S2Occup);
    textViewS2Status = (TextView)findViewById(R.id.S2Status);
    textLoading = (TextView)findViewById(R.id.texLoading);

    pieChartWeb = (WebView)findViewById(R.id.PChartWeb);

    if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_PORTRAIT)

```

```

        {
            backButton= (Button) findViewById(R.id.s2backbutton);
            backButton.setOnClickListener(this);
        }

        carParkListArray.add(carParkList1);
        carParkListArray.add(carParkList2);
        currentCarParkList = 0;
    }

//*****

// Handle messages from the Timer Thread
public Handler mainHandler = new Handler()
{
    public void handleMessage(android.os.Message msg)
    {
        // Update UI with new time and progress of progress bar
        if (msg.what == 0)
        {
            currentCarParkList =
msg.getData().getInt("currentCarParkList");
            //*currentCarParkList
            //display data
            Log.e("Ztag","disp. data invoked");
            Log.e("Z",Integer.toString(currentCarParkList));
            displayData();

            if (isDebugModeON)Log.e("Tom - report","program run DONE");

        }
        else
        {

        }

    }

};

//*****

public void startUpdater()
{
    // Send initial time

```

```

        // Start the thread
        myThread = new UpdaterThread(mainHandler);
        myThread.start();

        // Send start message to timer class
        Bundle abundle = new Bundle();
        //*abundle.putInt("timerCount",value);
        Message messageToThread = new Message();

        messageToThread.what = 0;
        messageToThread.setData(abundle);

        myThread.getHandler().sendMessage(messageToThread);

    }
    /*******
    private void displayData()
    {
        //disable loading text
        textLoading.setVisibility(View.GONE);

        //displaying "Global-Total" values
        textViewAvaliable.setText("Total Avaliable Spaces: "+
        Integer.toString(calc_global_AvaliablePlaces())+" / " +
        Integer.toString(calc_global_TotalPlaces()));
        textViewTotalOccu.setText("Total Occupied Spaces: "+
        Integer.toString(calc_global_OccupiedPlaces())+" / " +
        Integer.toString(calc_global_TotalPlaces()));
        //textViewTotalTotal.setText("Total Spaces: "+
        Integer.toString(calc_global_TotalPlaces()));
        textViewTotalTotal.setVisibility(View.GONE);
        textViewTotalOccup.setText("Total Occupancy: "+
        Double.toString(carPark.calcOccupancy(calc_global_OccupiedPlaces(),
        calc_global_TotalPlaces()))+"%");

        //constructing the listview

        final ListView listview = (ListView) findViewById(R.id.listview);
        final ArrayList<String> namelist = new ArrayList<String>();
        final ArrayList<Integer> occuList = new ArrayList<Integer>();

        //building a name-list for the listview
        for (int i = 0; i < carParkListArray.get(currentCarParkList).size(); ++i)

```

```

{

namelist.add((carParkListArray.get(currentCarParkList).get(i)).getName());
}

//building occulist for the listview
int o = 0;
for (int i = 0; i < carParkListArray.get(currentCarParkList).size(); ++i)
{
    //if carpark is bigger than 80%
    if (carParkListArray.get(currentCarParkList).get(i).calcOccupancy() >
80.00) {o = 0;}
    //if more than 60% is free
    else if
(carParkListArray.get(currentCarParkList).get(i).calcOccupancy() < 60.00) { o=2;}
    else { o=1;}
    occuList.add(o);
}

//assign the adapter

    /**final StableArrayAdapter adapter = new
StableArrayAdapter(this,android.R.layout.simple_list_item_1,namelist);

    /** WORKS final ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,R.layout.row,R.id.listText,namelist);

    final MySimpleArrayAdapter adapter = new
MySimpleArrayAdapter(this,namelist,occuList);

listview.setAdapter(adapter);

listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    public void onItemClick(AdapterView<?> parent, final View
view, int position, long id) {

        final String itemName = (String)
parent.getItemAtPosition(position);

        //debug toast
        Toast.makeText(getApplicationContext(),
itemName+" selected", Toast.LENGTH_SHORT).show();

        //fill in the next screen and switch
genrateDetScreen(itemName);

```



```

        if (getResources().getConfiguration().orientation
== Configuration.ORIENTATION_PORTRAIT)
            avw.showNext();
    }

});

} // end of displayData()

// Method for filling in the details screen with the selected instance's
values in PORT VIEW
public void genrateDetScreen(String carpName)
{
    //get the selected carPark from the arraylist
    int count = 0;
    carPark selected = new carPark();
    selected = carParkListArray.get(currentCarParkList).get(0);
    while (selected.getName()!=carpName)
    {
        count++;
        selected = carParkListArray.get(currentCarParkList).get(count);
    }

    //fill in text-fields
    textViewS2Name.setText(selected.getName());
    textViewS2Ava.setText( Integer.toString(selected.getavaliabileSpaces()));
    textViewS2Spaces.setText( Integer.toString(selected.getTakenSpaces())+" /
"+Integer.toString(selected.getTotalSpaces()));
    textViewS2Occup.setText( Double.toString(selected.calcOccupancy())+"%");
    textViewS2Status.setText(selected.getStatus());

    //generate the pie-chart
    int tmpOccup = ((int)selected.calcOccupancy());

    String chartURL =
"http://chart.apis.google.com/chart?cht=p&chs=250x150&chl=taken|free&chd=t:";
    chartURL += Integer.toString(tmpOccup) + "," + Integer.toString((100-
tmpOccup));
    chartURL += "&chco=1c7c78,f9d63e&chf=bg,s,4c9eb6ff";

    //draw pieChart
    pieChartWeb.loadUrl(chartURL);

```

```

        if (isDebugModeON)
        {
            Log.e("Tom - LOG",selected.getName());
            Log.e("Tom - LOG",chartURL);
        }
    }

    //Method for calculating global_AvaliablePlaces
    public int calc_global_AvaliablePlaces()
    {
        int count = 0;

        for(int i=0; i<carParkListArray.get(currentCarParkList).size();i++)
        {
            count = count +
carParkListArray.get(currentCarParkList).get(i).getavaliabileSpaces();
        }

        return count;
    }

    //Method for calculating global_OccupiedPlaces
    public int calc_global_OccupiedPlaces()
    {
        int count = 0;

        for(int i=0; i<carParkListArray.get(currentCarParkList).size();i++)
        {
            count = count +
carParkListArray.get(currentCarParkList).get(i).getTakenSpaces();
        }

        return count;
    }

    //Method for calculating global_TotalPlaces
    public int calc_global_TotalPlaces()
    {
        int count = 0;

        for(int i=0; i<carParkListArray.get(currentCarParkList).size();i++)
        {
            count = count +
carParkListArray.get(currentCarParkList).get(i).getTotalSpaces();

```

```

    }

    return count;
}

//Method to handle button clicks
public void onClick(View arg0)
{
    if (arg0 == backButton)
    {
        avw.showPrevious();
    }
}

//to disable the "built-in" destroy function upon screen rotation, and
redraw the elements but skip the data download
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation ==
Configuration.ORIENTATION_PORTRAIT){
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }

    //init containers
    init();

    //display data
    displayData();
}
} // End of Activity class

```

testCarpark.java

```
/**
 * @author Zoltan Tompa - S1112414
 */
package org.me.TESTmyandroidstuff;

import junit.framework.*;
import org.me.myandroidstuff.carPark;
/**
 * @author TomZoy
 */
public class testCarpark extends TestCase {

    /**
     * Test method for {@link org.me.myandroidstuff.carPark#calcOccupancy()}.
     */
    public void testCalcOccupancy() {

        int input[] = new int[] {10,0,0,0,0,10,5,10,70,80 };
        double expected[] = new double[] {0.0,0.0,0.0,50.0,87.5};
        double temp;
        for (int i = 0; i < expected.length;i++ )
        {
            temp = carPark.calcOccupancy(input[i],input[i+1]);
            assertEquals("test failed: ", expected[i], temp);
        }
    }

    // Test method for {@link org.me.myandroidstuff.carPark#trimName()}.

    public void testTrimName() {
        carPark testInstance1 = new carPark();

        String input[] = new String[] {"123456789","abc","1234567","" };
        String expected[] = new String[] {"12","abc","1234567","" };
        for (int i = 0; i < expected.length;i++ )
        {
            testInstance1.setName(input[i]);
            assertEquals("test failed: ",
expected[i],testInstance1.getName());
        }
    }
}
```

## row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Author: Zoltan Tompa - S1112414 -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/ListText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingBottom="5dp"
        android:paddingLeft="15dp"
        android:paddingTop="5dp"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="@color/c_bright"
        android:textSize="30sp" />

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginLeft="4px"
        android:layout_marginRight="10px"
        android:layout_marginTop="4px"
        android:paddingLeft="30dp"
        android:src="@drawable/icon" >

</ImageView>

</LinearLayout>
```

## Main.xml (port)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Author: Zoltan Tompa - S1112414 -->
<ViewSwitcher xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/vwSwitch"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/c_bg"
    android:padding="10dp" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="match_parent"
        android:gravity="left"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/author"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:layout_gravity="right"
        android:layout_marginLeft="150dp"
        android:text="@string/author"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@color/c_orange" />
```

```
<TextView
```

```
    android:id="@+id/TextViewAvaliable"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="1dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="18sp" />
```

```
<TextView
```

```
    android:id="@+id/textViewTotalOccu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="1dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="18sp" />
```

```
<TextView
```

```
    android:id="@+id/texLoading"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:clickable="false"
    android:duplicateParentState="false"
    android:gravity="center"
    android:longClickable="false"
    android:text="@string/firstLoadText"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="@color/c_red"
    android:textSize="25sp"
    android:textStyle="bold" />
```

```
<TextView
```

```
    android:id="@+id/textViewTotalTotal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="1dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="18sp" />
```

```
<TextView
```

```
    android:id="@+id/textViewTotalOccup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="1dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="18sp" />
```

```
<TextView
```

```
    android:id="@+id/errorText"
```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/placeholder"
        android:textColor="@color/c_red"
        android:visibility="invisible" />

<ListView
    android:id="@+id/Listview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<TextView
    android:id="@+id/lastupdate"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/placeholder"
/>

</LinearLayout>

<!-- details screen starts HERE -->

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal|left"
    android:orientation="vertical" >

    <Button
        android:id="@+id/s2backbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:background="@color/c_bright"
        android:text="@string/backbutton"
        android:textColor="@color/c_dark" />

    <TextView
        android:id="@+id/Parkname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:padding="10dp"
        android:paddingTop="20dp"
        android:text="@string/placeholder"
        android:textColor="@color/c_orange"
        android:textSize="35sp" />

    <TextView
        android:id="@+id/staticTextS0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center"

```

```
        android:text="@string/s2ava"  
        android:textColor="@color/c_dark"  
        android:textSize="15sp" />
```

```
<TextView  
    android:id="@+id/S2Ava"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:gravity="center"  
    android:paddingBottom="15dp"  
    android:text="@string/placeholder"  
    android:textColor="@color/c_yellow"  
    android:textSize="20sp" />
```

```
<TextView  
    android:id="@+id/staticTextS21"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="@string/s2spaces"  
    android:textColor="@color/c_dark"  
    android:textSize="15sp" />
```

```
<TextView  
    android:id="@+id/S2Spaces"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="@string/placeholder"  
    android:textColor="@color/c_yellow"  
    android:textSize="20sp" />
```

```
<TextView  
    android:id="@+id/staticTextS22"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:paddingTop="20dp"  
    android:text="@string/s2status"  
    android:textColor="@color/c_dark"  
    android:textSize="15sp" />
```

```
<TextView  
    android:id="@+id/S2Status"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="@string/placeholder"  
    android:textColor="@color/c_yellow"  
    android:textSize="20sp" />
```

```
<WebView  
    android:id="@+id/PChartWeb"  
    android:layout_width="250dp"
```



```

        android:layout_height="150dp"
        android:layout_gravity="center"
        android:contentDescription="@string/s2chart"
        android:gravity="center"
        android:paddingBottom="50dp"
        android:paddingTop="50dp"
        android:src="@drawable/icon" />

<TextView
    android:id="@+id/staticTextS23"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="@string/s2occu"
    android:textColor="@color/c_dark"
    android:textSize="15sp" />

<TextView
    android:id="@+id/S20occup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="20sp" />
</LinearLayout>

</ViewSwitcher>

```

## Main.xml (land)

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Author: Zoltan Tompa - S1112414 -->
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/GridLayout1"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:background="@color/c_bg"
    android:columnCount="2"
    android:orientation="horizontal"
    android:paddingTop="5dp"
    android:rowCount="13" >

    <TextView
        android:id="@+id/author"
        android:layout_column="1"
        android:layout_gravity="right/top"
        android:layout_row="0"
        android:paddingLeft="20dp"
        android:paddingRight="5dp"
        android:text="@string/author"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@color/c_orange" />

    <TextView

```

```
    android:id="@+id/TextViewAvaliable"
    android:layout_column="0"
    android:layout_gravity="left/top"
    android:layout_row="0"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow" />
```

```
<TextView
    android:id="@+id/textViewTotalOccu"
    android:layout_column="1"
    android:layout_gravity="top/left"
    android:layout_row="0"
    android:paddingLeft="40dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow" />
```

```
<TextView
    android:id="@+id/textViewTotalTotal"
    android:layout_column="1"
    android:layout_gravity="top/center_horizontal"
    android:layout_row="0"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow" />
```

```
<TextView
    android:id="@+id/textViewTotalOccup"
    android:layout_column="1"
    android:layout_columnSpan="1"
    android:layout_gravity="right/top"
    android:layout_row="0"
    android:paddingRight="200dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow" />
```

```
<ListView
    android:id="@+id/Listview"
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:layout_column="0"
    android:layout_row="1"
    android:layout_rowSpan="12" />
```

```
<!-- details (right side) starts here -->
```

```
<TextView
    android:id="@+id/Parkname"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="1"
    android:paddingTop="25dp"
    android:text="@string/selectone"
    android:textColor="@color/c_orange"
    android:textSize="35sp" />
```

```
<TextView
    android:id="@+id/staticText0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1"
```

```
android:layout_gravity="center_horizontal"
android:layout_row="2"
android:gravity="center"
android:text="@string/s2ava"
android:textColor="@color/c_dark"
android:textSize="15sp" />
```

```
<TextView
    android:id="@+id/S2Ava"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1"
    android:layout_gravity="center_horizontal"
    android:layout_row="3"
    android:gravity="center"
    android:paddingBottom="15dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="20sp" />
```

```
<TextView
    android:id="@+id/staticTextS21"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="4"
    android:text="@string/s2spaces"
    android:textColor="@color/c_dark"
    android:textSize="15sp" />
```

```
<TextView
    android:id="@+id/S2Spaces"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="5"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="20sp" />
```

```
<TextView
    android:id="@+id/staticTextS22"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="6"
    android:paddingTop="15dp"
    android:text="@string/s2status"
    android:textColor="@color/c_dark"
    android:textSize="15sp" />
```

```
<TextView
    android:id="@+id/S2Status"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="7"
    android:paddingTop="15dp"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="20sp" />
```

```
<TextView
```

```
        android:id="@+id/staticTextS23"
        android:layout_column="1"
        android:layout_gravity="center_horizontal"
        android:layout_row="8"
        android:paddingTop="15dp"
        android:text="@string/s20ccu"
        android:textColor="@color/c_dark"
        android:textSize="15sp" />
```

```
<TextView
```

```
    android:id="@+id/S20ccup"
    android:layout_column="1"
    android:layout_gravity="center/top"
    android:layout_row="9"
    android:text="@string/placeholder"
    android:textColor="@color/c_yellow"
    android:textSize="20sp" />
```

```
<TextView
```

```
    android:id="@+id/texLoading"
    android:layout_column="1"
    android:layout_gravity="center"
    android:layout_row="11"
    android:text="@string/firstLoadText"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textColor="@color/c_red"
    android:textStyle="bold" />
```

```
<android.support.v7.widget.Space
```

```
    android:id="@+id/space1"
    android:layout_width="1dp"
    android:layout_height="15dp" />
```

```
<android.support.v7.widget.Space
```

```
    android:id="@+id/space2"
    android:layout_width="1dp"
    android:layout_height="15dp" />
```

```
<WebView
```

```
    android:layout_column="1"
    android:layout_gravity="center"
    android:layout_row="10"
    android:id="@+id/PChartWeb"
    android:layout_width="250dp"
    android:layout_height="150dp"
    android:contentDescription="@string/s2chart"
    android:gravity="center"
    android:paddingBottom="50dp"
    android:paddingTop="50dp"
    android:src="@drawable/icon" />
```

```
</GridLayout>
```

## Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Author: Zoltan Tompa - S1112414 -->
<resources>
    <string name="hello">Hello World, HttpTestActivity!</string>
    <string name="app_name">Glasgow Carpark App</string>
    <string name="author">Zoltan Tompa - S1112414</string>
    <string name="placeholder"></string>
    <string name="selectone">Please select a Carpark from the list</string>
    <string name="firstLoadText">The program is loading... please wait.</string>

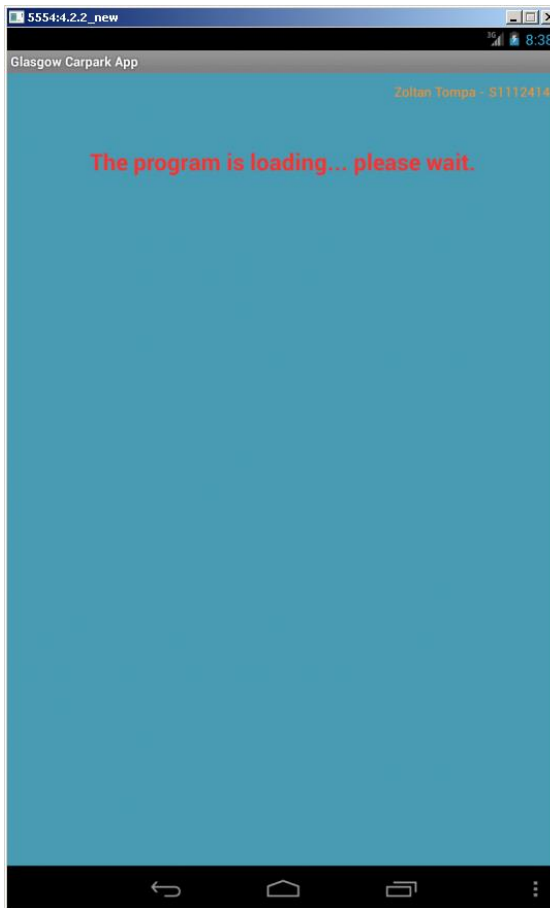
    <string name="s2chart">piechart</string>
    <string name="s2ava">Spaces Available: </string>
    <string name="s2occu">Carpark Occupancy: </string>
    <string name="s2status">Carpark Status: </string>
    <string name="s2spaces">Spaces Occupied / Total: </string>
    <string name="backbutton">&lt; back</string>

    <color name="c_bg">#4c9eb6</color>
    <color name="c_orange">#ff9034</color>
    <color name="c_yellow">#f9d63e</color>
    <color name="c_dark">#0c7672</color>
    <color name="c_bright">#fff9e3</color>
    <color name="c_red">#ff3434</color>

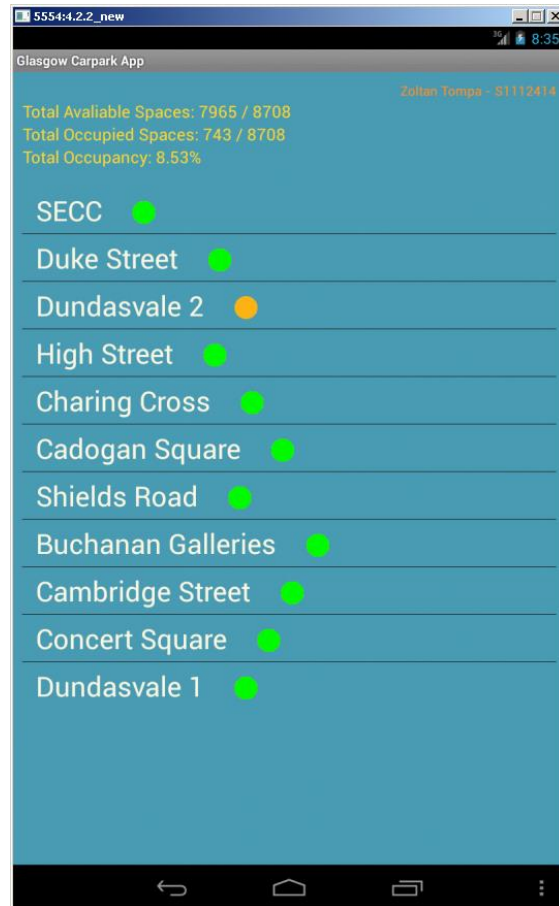
</resources>
```

## Appendix

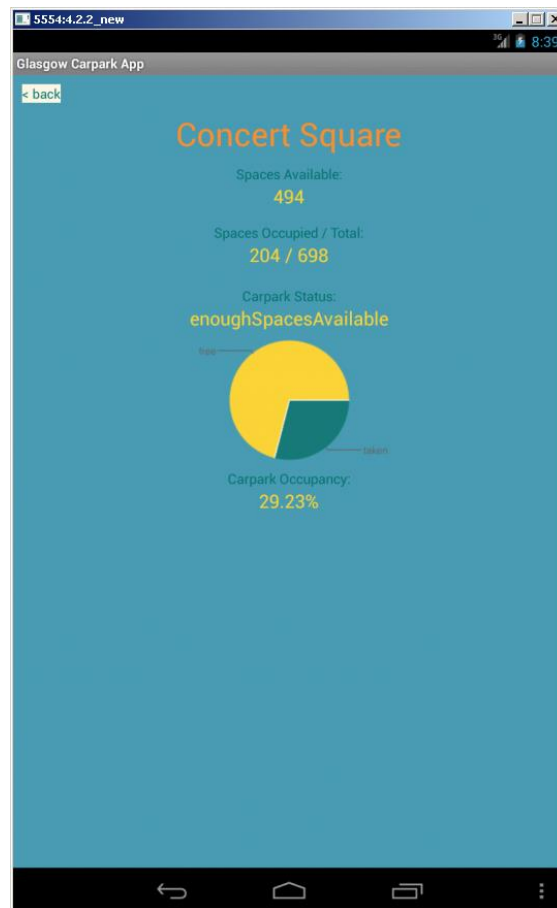
### Screenshots



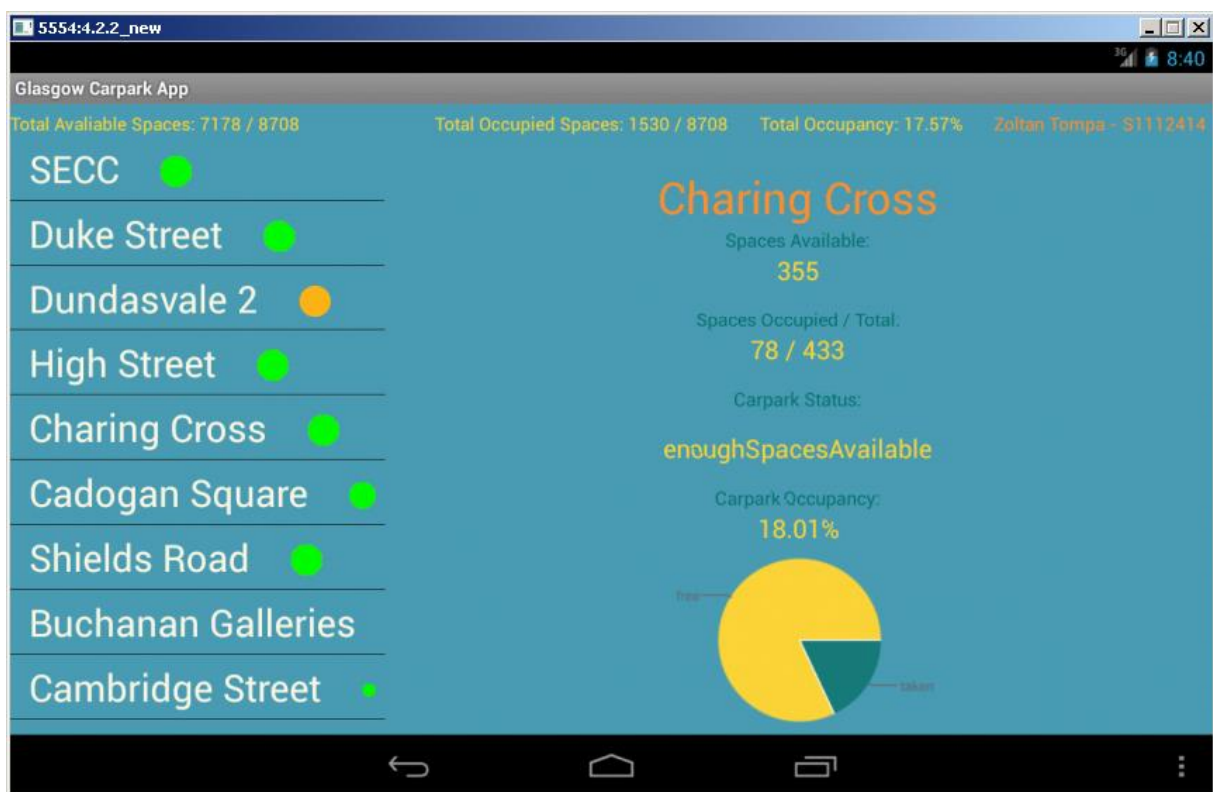
/ fig. 1 – loading screen /



/ fig. 1 –carpark selection /

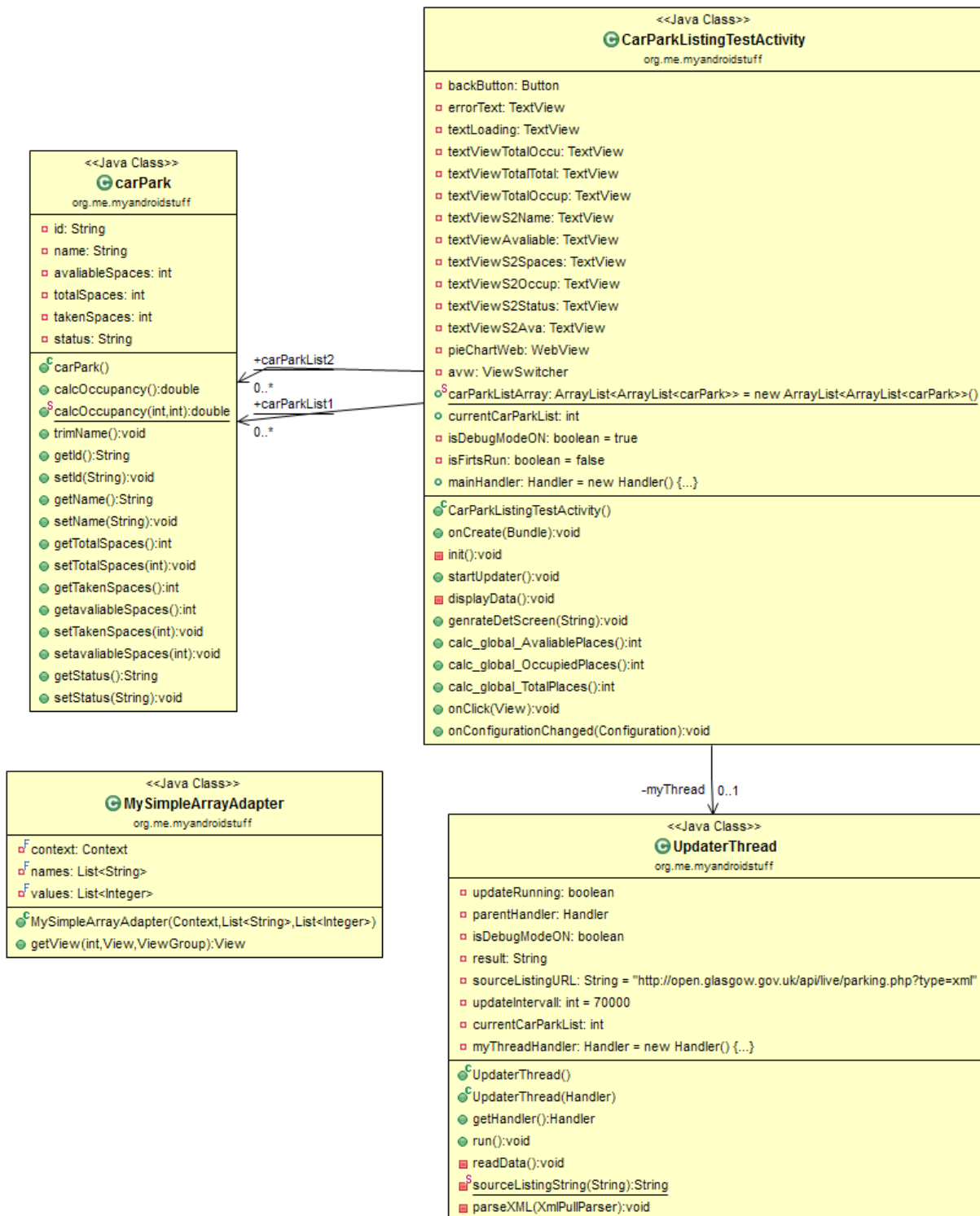


/ fig. 3 –details screen /



/ fig. 4 –landscape view /

## Class Diagram



/ fig. 5 – Class Diagram /



## References

<http://stackoverflow.com/questions/11701399/round-up-to-2-decimal-places-in-java>  
<http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html>  
[http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html#substring\(int, int\)](http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/String.html#substring(int, int))  
<https://developer.android.com/reference/java/util/ArrayList.html>

<http://www.sitepoint.com/learning-to-parse-xml-data-in-your-android-app/>  
<http://www.androidhive.info/2011/11/android-xml-parsing-tutorial/>  
<http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html>

chart;

<http://androidroadies.blogspot.in/2013/05/generate-3d-pie-chart-using-google.html>  
<http://psychopyko.com/tutorial/how-to-use-google-charts/>  
[https://developers.google.com/chart/image/docs/making\\_charts](https://developers.google.com/chart/image/docs/making_charts)  
[http://markinns.com/articles/full/using\\_googles\\_chart\\_api](http://markinns.com/articles/full/using_googles_chart_api)  
<http://chart.apis.google.com/chart?cht=p&chs=300x150&chl=A|B|C&chd=t:20,20,60&chco=80C65A,224499,FF0000>

<http://chart.apis.google.com/chart?cht=p&chs=250x150&chl=taken|free&chd=t:28,72&chco=1c7c78,f9d63e>

orientation;

[http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)  
<http://stackoverflow.com/questions/5407752/android-layout-folders-layout-layout-port-layout-land>  
<http://stackoverflow.com/questions/5443304/how-to-change-an-android-apps-name>

//avoid reloading:

<http://stackoverflow.com/questions/5913130/dont-reload-application-when-orientation-changes>

expandable listview;

<http://www.vogella.com/tutorials/AndroidListView/article.html>  
<http://theopentutorials.com/tutorials/android/listview/android-expandable-list-view-example/>

icons

<http://developer.android.com/design/style/iconography.html>

async task

[http://www.vogella.com/tutorials/AndroidBackgroundProcessing/article.html#overview\\_basics](http://www.vogella.com/tutorials/AndroidBackgroundProcessing/article.html#overview_basics)  
<https://developer.android.com/reference/android/os/AsyncTask.html>