2013

ECWM506 Mobile Computing Principles

COURSEWORK 1 MOBILE WEB APPLICATION

W1387769 MOHAMED RAHIM BARAKY

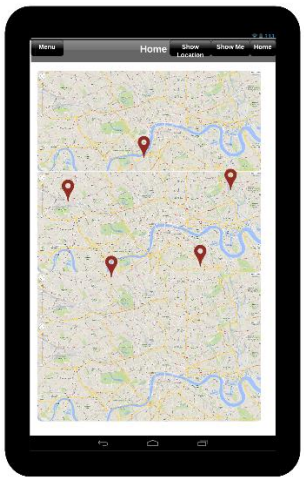
Contents

Web application: http://users.wmin.ac.uk/~w1387769/mobileCW/place.html
Xml file: http://users.wmin.ac.uk/~w1387769/mobileCW/places.xml
Xsl file: http://users.wmin.ac.uk/~w1387769/mobileCW/Transformer.xsl

1)

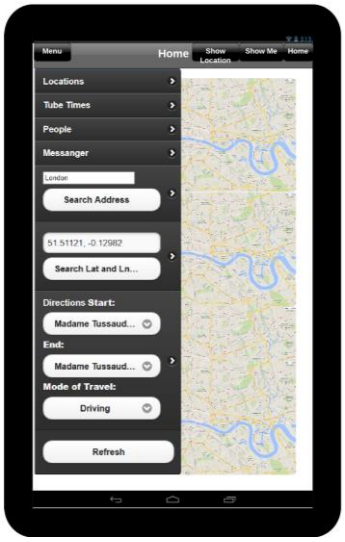
"*Design the Screens for the application, and the navigation flow, outlining which of Ben Schneider man's Eight Golden Rules were followed and why, and how these influenced your design*."

This is the main page. As soon as you open the application Google maps will appear in the centre. You will have a bar at the top with the option menu on the left hand side and on the right-hand side show locations, show me and home. Also the browser/phone will ask if you would like to share your location with application.



When the user selects the show locations option, markers will appear on the page showing all the tourist locations, the user may then click on any of the markers and an info window with the name of the tourist attraction will open.

When the user selects the show me option, a text bubble marker will appear pointing at the position at which the user is, it will also display the words "you're here".

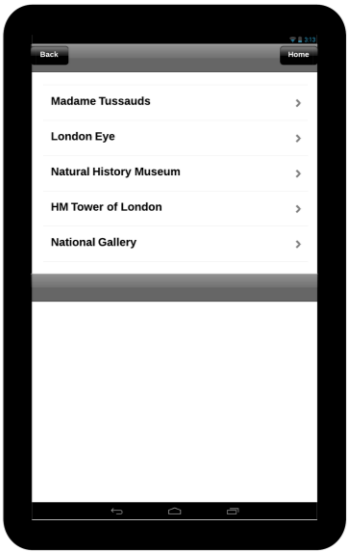


When the user selects the menu option, a panel will appear to the left, the first four options direct the user to another page. These options are

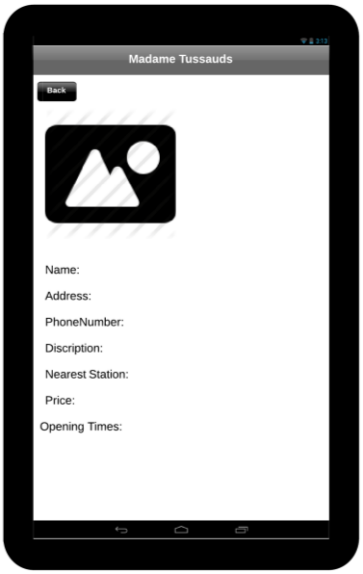
- Locations- which displays a list of all the tourist locations and their details.
- Tube times- this direct the user to the Transport for London website to check the tube times.
- People and messenger and not implemented at this time.

The other four options performance functions.

- The first one allows you to search for an address.
- The second option allows you to search latitude and longitude coordinates.
- The third option allows you to get directions to and from any location as well as your own position, you might also select the mode of transport such as driving, walking, bicycling and transit.
- The fourth option allows you to refresh the application.



When the user selects the locations option, a new page will appear with a list of all the tourist locations. The user then may select whichever location he or she wishes. This page also contains, in the top bar, a back button which will return the user to the previous page and the home button which will return the user back to the homepage.



Should the user then click on a location such as Madame Tussauds a new page will appear the header of which will display the name of the location and underneath will be a back button leading back to the list of locations. The main body of the page contains details about the selected location, there will be an image showing the location (i.e. the building, the Ferris wheel etc.) Followed by the Name, Address, Phone Number, Description, Nearest Station, Price and Opening Times.

Throughout my work I used Ben Schneider man's Golden rules in order to make my work better. I used rules one, five, six, seven and eight

- The first rule of his which I used was striving for consistency, throughout my work I have kept the colour scheme the same in order to be consistent as well as using similar colour panels and buttons.
- The second rule was to enable frequent users to use shortcuts, I did not use this rule simply because my application doesn't have any areas in which shortcuts could be added. Everything one would need to access can be accessed very quickly either from the home screen or the panel.
- The third rule is to offer information feedback for every user action, I did not implement this rule because I do not foresee any situation in which to respond time will be noticeable, the application is very simple and does not have long loading times for options.
- The fourth rule is to design dialogues to yield closure, my application does not have any areas in which dialogues could be added for actions such as sending emails etc. therefore I did not implement it.
- The fifth rule is to offer error prevention and simple error handling, I implemented this by if the user was to select "my location" and the system was unable to provide a location a text box will appear saying that it was unable to locate the uses the device.
- This sixth rule was to permit easy reversal of actions, I implemented this such that every page has either a back button or a home button to allow the user to easily revert to the previous page.
- The seventh rule is to support internal locus of control, I implemented this in the form of whenever a marker has a text box the user may easily close this text box by clicking the cancel button in the dialogue.
- The eighth rule is to reduce short-term memory load, I implemented this by keeping all my displays as simple as possible in order to reduce the memory it took in the system.

2)

"Create an XML file to hold the attraction information for any 5 tourist attractions displayed on this website. The information should be held in a minimum of 10 XML tags and at least one of these must contain more than one child node. The information held should contain as a minimum:

a. Attraction name
b. Address
c. Picture
d. Tube station
e. Price
f. Short Description
g. Long Description
h. Opening times
i. Category"

places.xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<?xml-stylesheet type="text/xsl"  href="Transformer.xsl"?>
<places>
  <attraction>
      <name>Madame Tussauds</name>
      <address>
      <lineOne>Marylebone Road</lineOne>
      <lineTwo>London</lineTwo>
      <postcode>NW1 5LR</postcode>
      <location>51.52297,-0.15505,</location>
      <phonenumber>+44 (0) 871 894 3000</phonenumber>
      </address>
      <picture>MadameT.jpg</picture>
      <tubeStation>Baker Street</tubeStation>
      <price>£30</price>
      <shortDis>Waxwork Museum</shortDis>
      <longDis>Madame Tussauds is a major tourist
        attraction in London, displaying waxworks of
        historical and royal figures, film stars, sports
        stars and infamous murderers. Madame Tussauds is
        owned an operated by Merlin Entertainments.</longDis>
      <!--http://en.wikipedia.org/wiki/Madame_Tussauds-->
      <openingtimes>
        <monday>9:30-17:30</monday>
        <tuesday>9:30-17:30</tuesday>
        <wednesday>9:30-17:30</wednesday>
        <thursday>9:30-17:30</thursday>
        <friday>9:30-17:30</friday>
        <saturday>9:30-17:30</saturday>
        <sunday>9:30-17:30</sunday>
      </openingtimes>
      <category>Museum</category>
  </attraction>
  <attraction>
      <name>London Eye</name>
      <address>
      <lineOne>Westminster Bridge Road</lineOne>
      <lineTwo>London</lineTwo>
      <postcode>SE1 7PB</postcode>
      <location>51.50282,-0.11925,</location>
      <phonenumber>+44 (0)871 781 3000</phonenumber>
      </address>
      <picture>LondonEye.jpg</picture>
      <tubeStation>Waterloo station</tubeStation>
      <price>£19.20</price>
```

eXtensible Markup Language file

As you can see from the screenshot on the left I have created a XML file which contains information about tourist locations. The information it contains is the name, the address, the telephone number, the nearest tube station, the price, the short and long description of the location, the opening times and the category. You cannot see it from the screenshot that this XML file contains the information of five tourist locations as specified in the coursework.

3)

"*Create an XSL file that transforms the information held in the XML file into a jQuery Mobile application that presents the information to the tourists and corresponds to your design. You must therefore make use of jQuery Mobile UI elements and not the standard HTML elements.*"

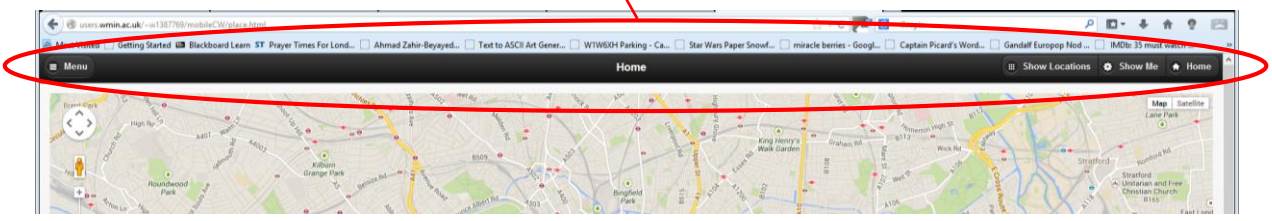
Transformer.xsl

```
<?xml version="1.0" encoding="utf-8"?>
<!-- DWXMLSource="Places.xml" -->
<!DOCTYPE xsl:stylesheet  [
<!ENTITY nbsp   " ">
<!ENTITY copy   "&#169;">
<!ENTITY reg    "&#174;">
<!ENTITY trade  "&#8482;">
<!ENTITY mdash  "&#8212;">
<!ENTITY ldquo  "&#8220;">
<!ENTITY rdquo  "&#8221;">
<!ENTITY pound  "&#163;">
<!ENTITY yen    "&#165;">
<!ENTITY euro   "&#8364;">
]>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="http://
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

        <style type="text/css">
         #map-canvas { width:100%; height: 800px }

        </style>
        <title>London Sites</title>
        <link href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" rel="stylesheet" type="text/css" />
        <script src="http://code.jquery.com/jquery-1.9.1.min.js" type="text/javascript"></script>
        <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js" type="text/javascript"></script>
  </head>
      <body>

        <!-- ////////////////////////////////////////////////////////////////////////////////////////////////////////////
        <!-- ////////////////////////////////////////////////////////////////////////////////////////////////////////////
        <!-- ////////////////////////////////////////////////////////////////////////////////////////////////////////////
        <div data-role="page" id="page1">
          <div data-role="panel" id="panel1">
            <ul data-role="listview"  data-inset="false">
              <li data-role="list-divider" role="heading">
              </li>
              <li data-theme="a">
                <a href="#page7" data-transition="slide">
                  Locations
                </a>
              </li>
              <li data-theme="a">
```

eXtensible Markup Language file — length : 17935 lines : 488 — Ln : 9 Col : 27

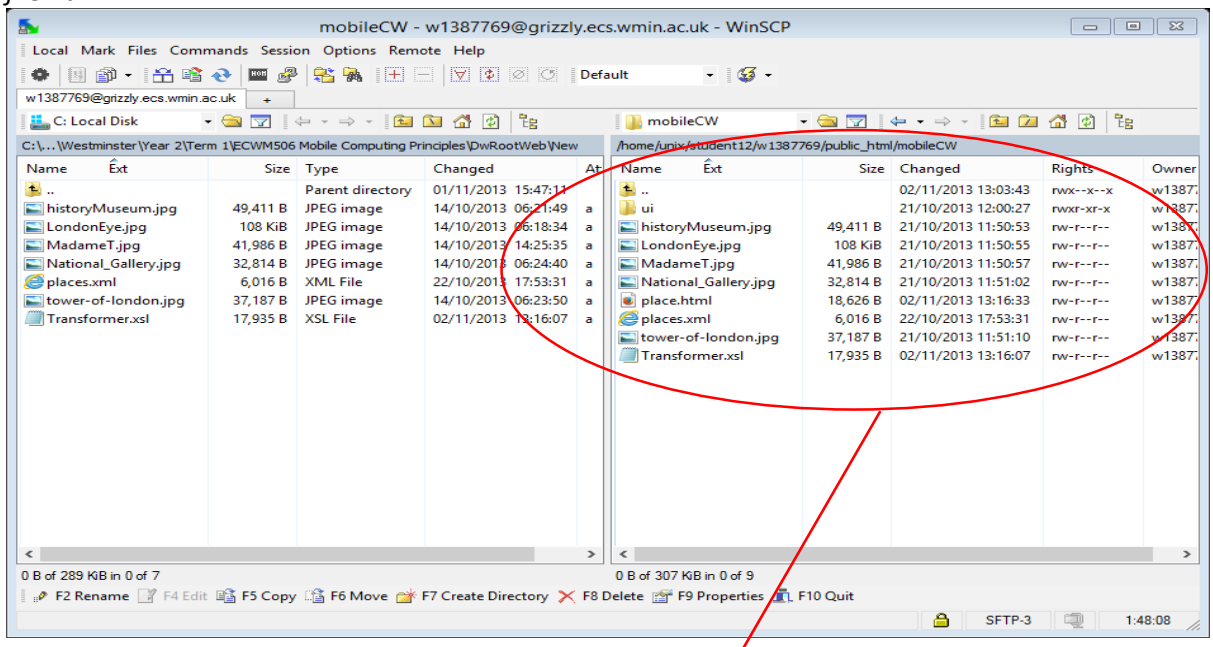
As you can see from the screenshot above I have implemented J query in the code in order to style my work.

As you can see from the screenshot below I have made use of J query mobile UI elements.



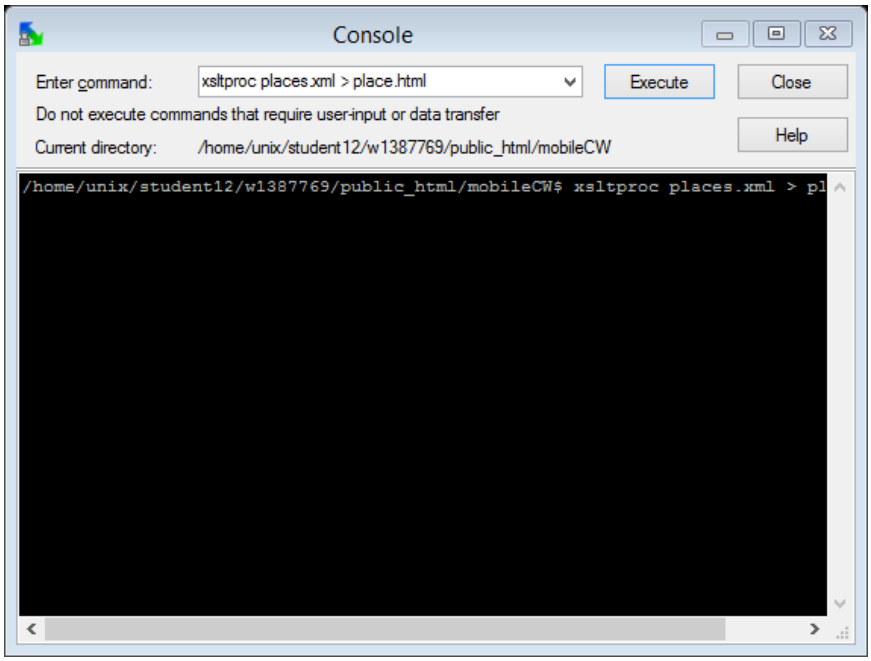
4)

"We will use xsltproc to do the transformation on the server. To do this, you will also need to save your XML and our XSL files in your public_html area on UNIX. You can transfer those files using SFTP (host name: grizzly.ecs.wmin.ac.uk). Once these are on the server, run xsltproc your_file.xml > your file.html"



As you can see above all my work has been saved on my UNIX account.

The screenshot below shows that I have done the xsltproc transformation on the server.



5)

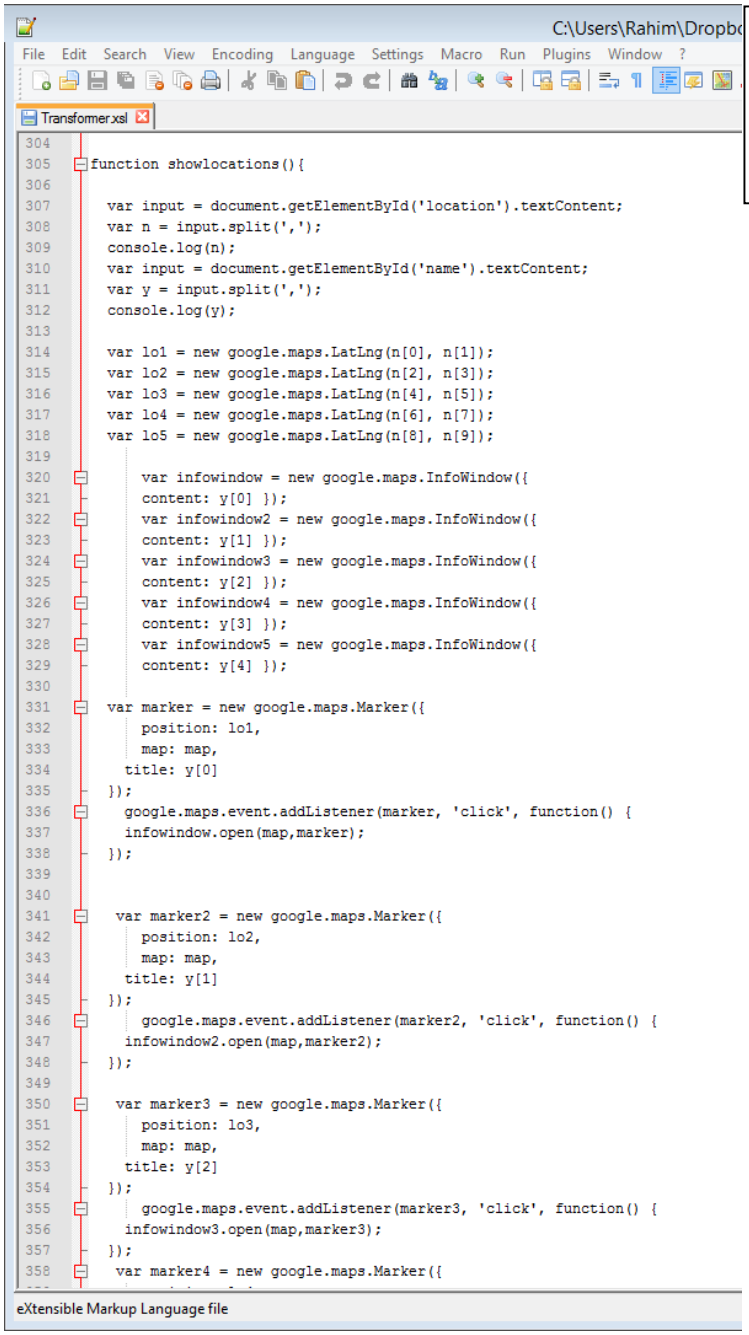
"Modify your XSL file to:

j. Add a page to display a Google Map and the attraction's location

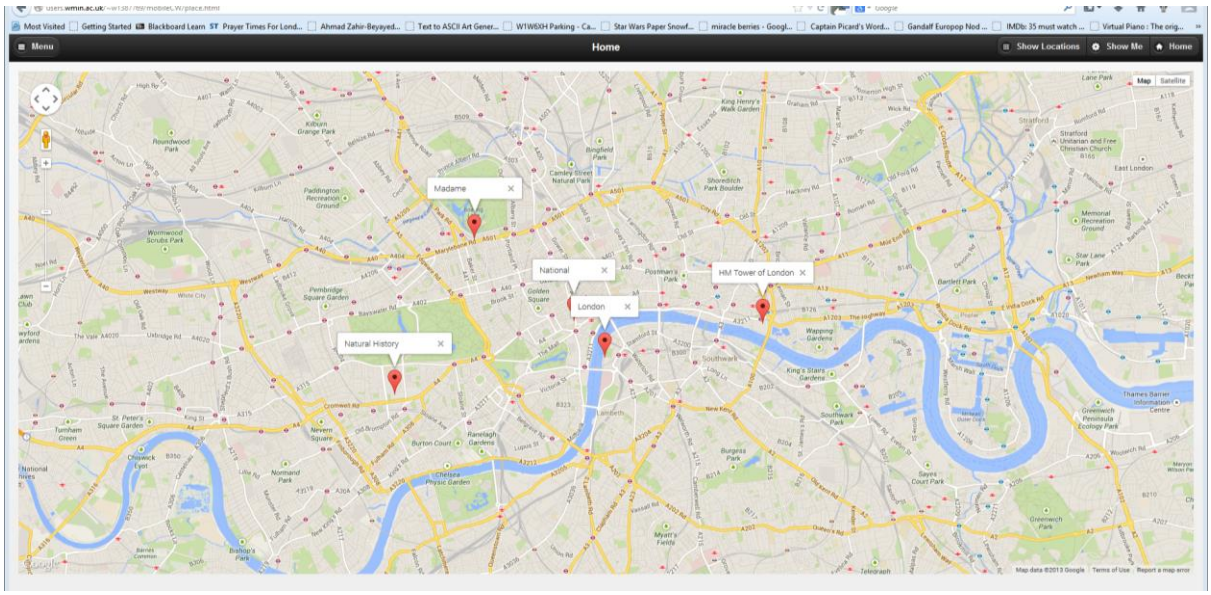
k. add an option to display the user's current location, and implement this function

l. Add an option to display the directions between the user's current location and a chosen attraction, and implement this function."

j)



The screenshot to your left shows the function which I created which shows that attractions locations on the Google maps.



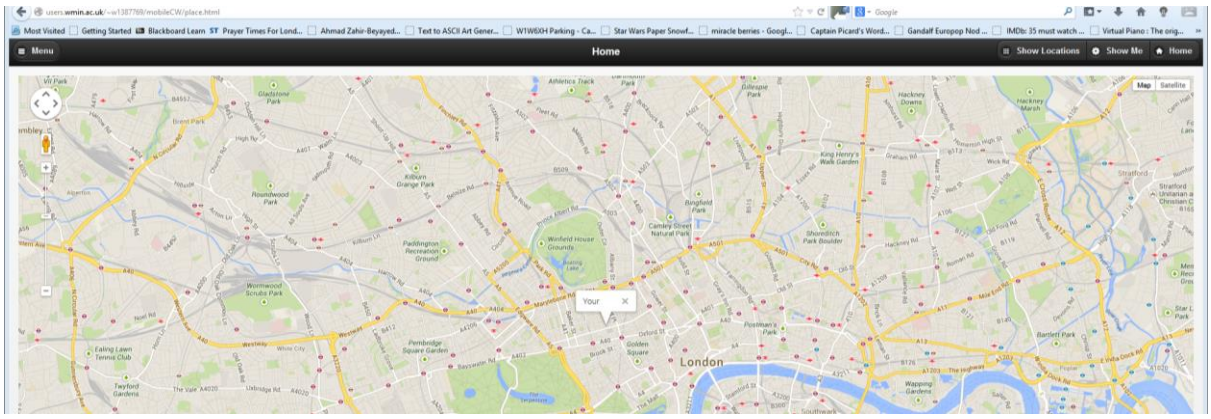
The screenshot above shows my implemented function within the application, when the user wishes to view the tourist locations he or she simply selects on show locations, at which point markers will appear on the map pointing to each locations position. The user then may click on the marker in order to bring up the name of the tourist attraction (i.e. the London eye, national Gallery etc.) They can dismiss this information by clicking on the X symbol.

k)

Transformer.xsl

```
427       infowindow.open(map,marker);
428     });
429       } else {
430         alert('No results found');
431       }
432     } else {
433       alert('Geocoder failed due to: ' + status);
434     }
435   });
436 }
437
438 function here(){
439     <!--https://developers.google.com/maps/documentation/javascript/examples/map-geolocation -->
440   // Try HTML5 geolocation
441   if(navigator.geolocation) {
442     navigator.geolocation.getCurrentPosition(function(position) {
443       var pos = new google.maps.LatLng(position.coords.latitude,
444                                        position.coords.longitude);
445
446
447       var infowindow = new google.maps.InfoWindow({
448         map: map,
449         position: pos,
450         content: 'Your here'
451
452       });
453
454       map.setCenter(pos);
455     }, function() {
456       handleNoGeolocation(true);
457     });
458   } else {
459     // Browser doesn't support Geolocation
460     handleNoGeolocation(false);
461   }
462 }
```

The screenshot on the left shows the function which I created which displays the user's current location.



The screenshot above shows my implemented function within the application, when the user wishes to find out his or her position they would simply select show me, at which point an information box will appear pointing to the location of the user's device, it will also display the text you're here.

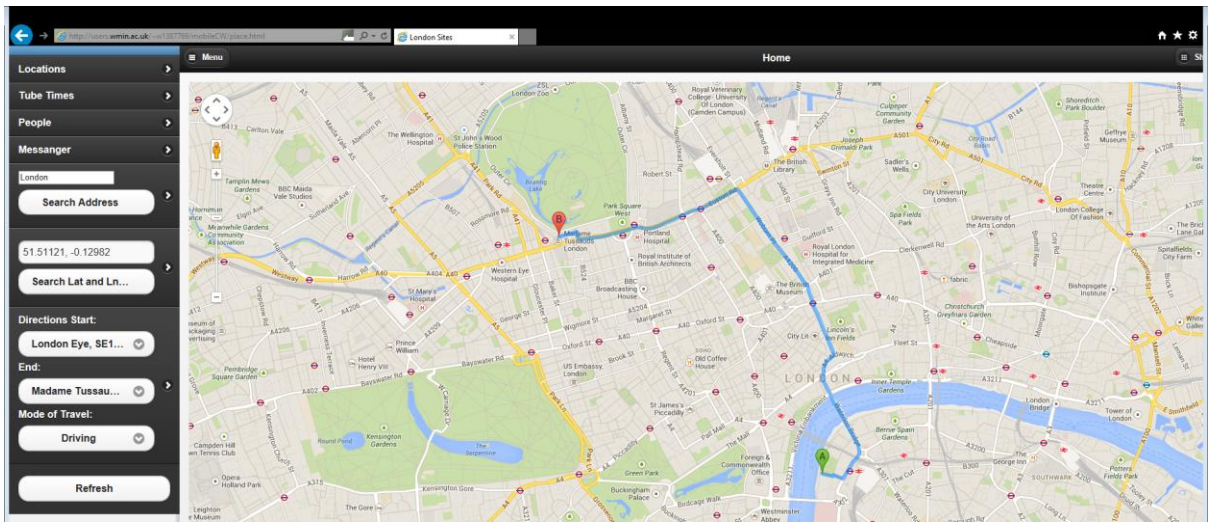
I)

Transformer.xsl

```
function calcRoute() { <!--https://developers.google.com/maps/documentation/ja
-->
        navigator.geolocation.getCurrentPosition(function(position) {
      var pos = new google.maps.LatLng(position.coords.latitude,
                                       position.coords.longitude);



var selectedMode = document.getElementById('mode').value;
var input = document.getElementById('start').value;
  var e = input.split(',');
  console.log(e);
var input = document.getElementById('end').value;
  var g = input.split(',');
  console.log(g);

 if(document.getElementById('start').value == 01) {

var request = {
      origin:pos,
      destination:g[01],
      travelMode: google.maps.TravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    }
  });

    } else if (document.getElementById('end').value == 01){
        var request = {
      origin:e[01],
      destination:pos,
      travelMode: google.maps.TravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    }
  });
        } else {
 var request = {
      origin:e[01],
      destination:g[01],
      travelMode: google.maps.TravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    }
  }); }
})
}
```

eXtensible Markup Language file

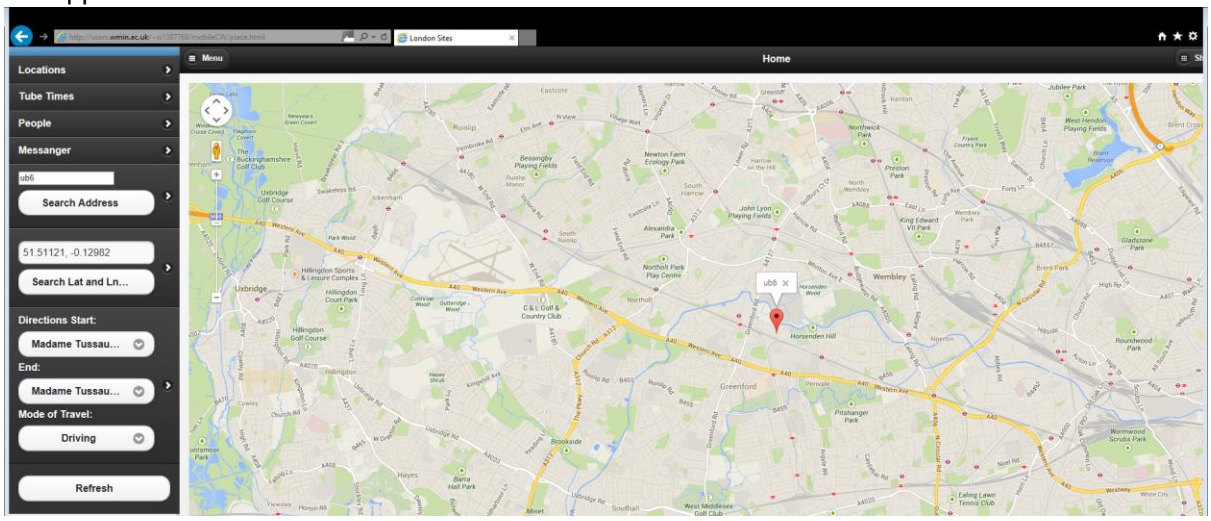
The screenshot on the left shows the function which I created to show directions between the different tourist locations and/or from the user's current position.



The screenshot above shows my implemented function within the application, when the user wishes to obtain directions he or she simply need to select the start of the directions, either a pre-set location or his or her location, and then select the end destination. He or she may also alter the mode of travel, the options being driving walking bicycling or transit.

6) (Additional features)

In my mobile application I added two additional features, the first feature allows the user to locate an address by entering the address in the address box and selecting search address, this will then create a marker on the map pointing to the location of the address, on top of the marker a text box will appear with the address the user entered as shown below.



This second additional future I added was to allow the user to locate an address by entering a longitude and latitude coordinate. Once entered the user simply press search lat and long, at which point a marker will appear on the map pointing to the location of the coordinates, above this marker a text box will appear with the nearest address to that coordinate as shown in the screenshot below.