CSM2220/SEM2220 Assessed Assignment 2018-19

Assignment One: Mobile Web – *I'm A Tourist* web app

Chris Loftus
Hand-out date: Monday 18th February 2019
Hand-in date: Tuesday 12th March 2019
Feedback by: Tuesday 2nd April 2019
20% of overall assessment
(about 17 hours)

The problem

In the worksheets you developed a simple conference web app using jQuery Mobile, HTML5 etc. That app used hardcoded session data in the *index.html* page. The attached basic example (*conference_example.zip* and *ConferenceBrowser-debug.apk*) creates that session list dynamically based on a SQLite database.

Your task is to study this code and produce something similar, but for the *I'm A Tourist* app, i.e. you may wish to reuse and adapt the provided code. Note that the code does not use IE6, unlike code used in workshops. It is up to you as to whether you switch to IE6, or use 'old school' Javascript.

This prototype app should allow the tourist to take notes and a photo of a location. The short description, notes, photo, date/time and location information are then stored in a local database. You may use the WebSQL API for this (rather than IndexDB, although see the Flair part of the marking grid). The tourist is then able to list all the entries recorded so far (rather like the *sessions* list in the conference app). Furthermore, a map will be displayed that also shows entries as pins.

The detail of how you present this functionality is up to you. I would be happy with a tabbed approach, e.g. three tabs: *Record Visit*, *Visits* and *Map*, but you are welcome to try something else if you wish.

Package up your app as a PhoneGap application and then test it on either a real Android phone or an Android emulator. I will need to see evidence.

Record Visit functionality

For the prototype this function should allow a tourist to add notes in a text area. There must be a button to allow the tourist to open the camera and take a picture. The tourist can see the image on the page and can then decide whether to save or cancel the page entry. The tourist should be able to take the picture again if they are not happy with the first attempt. The saved data is then stored in the local database.

List Visits functionality

This displays a list of visits in date/time order. A filter field should be included. Each row should show the date/time, the short description and a thumbnail image. Clicking on the short description replaces the page with the full description. For the prototype there is no edit facility.

Map functionality

When the map page is displayed (e.g. by tapping on the *map* tab) then a map of the current location is displayed, showing where you are located. Pins corresponding to the visits in the database are also displayed (i.e. those relevant for the area of map being displayed). You will need to create a Google account and obtain a Google maps API key.

Approach

Step One: Start by running the example project (attached zip and apk files on Blackboard). The example is not robust, but you should be able to see three tabs: Home, Sessions and Map. The Home and Sessions will show data. The Map tab will show an error. This is because it lacks an Map API key. You will need to add your own key.

Step Two: Analyse the example code.

Step Three: Build code for the three functional areas. I suggest you don't try and implement everything at once. I also recommend that you leave the map functionality until last.

Make sure you use the browser Developer Tools when debugging your code. I find it essential to step debug when writing Javascript. See the worksheet on testing.

Step Four: After testing the app on browsers and a mobile web emulator (e.g. Chrome developer tools) package up the app as a PhoneGap wrapped app (as we did in a worksheet) and run on an Android emulator or a real device.

Step Five: Write a 1,000 – 1,500 word report that tells the story of how you went about implementing the assignment, problems encountered, what you learned and screen shots showing the solution running either on a real mobile device or Android emulator. Make sure you include an additional cover page with your name etc.

If you used an alternative mobile web framework (not jQuery Mobile) then provide justification for that choice.

Please indicate in your report the mark you think you should be awarded for each item in the marking table (see below) and why.

Step Six: Package up your code, the apk file generated by PhoneGap and the report (as a pdf) in a **zip** file and upload to Blackboard by 4pm Tuesday 12th March 2019. I will need to deploy your apk files on my phone.

Note: this is an "individual" assignment and must be completed as a one-person effort by the student submitting the work.

This assignment is **not** marked anonymously.

When you submit your report to Blackboard, you will be shown the wording regarding the **Declaration of Originality;** these are the same as those found on the Anonymous Marking Sheet, which is available on the resources page of the

Department's Intranet: http://www.aber.ac.uk/~dcswww/intranet/staff-students-internal/teaching/resources.php. When you click submit, we will assume that you have agreed to the terms of that statement. You do not need to complete a separate form regarding the Declaration of Originality.

Learning outcomes

By undertaking this assignment, and the worksheets it builds on, you will:
1. Learn how to use the Web SQL Database API.
2. Learn how to use some of the jQuery Mobile API.
3. Learn how to test web apps.
4. Learn how to create a hybrid web app that will run on a mobile device.
5. Enhance your Javascript and HTML5 abilities.

Mark breakdown

Assessment will be based on the assessment criteria described in Appendix AA of the Student Handbook:
http://www.aber.ac.uk/~dcswww/Dept/Teaching/Handbook/AppendixAA.pdf.
However, the following table gives you some indication of the weights associated with individual parts of the assignment. This will help you judge how much time to spend on each part. Note that I will award full marks for each category below if you fully complete the requested functionality. However, the flair marks represent work that goes beyond the requested features.

Documentation	Does the documentation convey a convincing and detailed story of how the code was implemented, problems encountered, what was learned and an indication of the mark that should be awarded and why?	20%
Implementation (record visit function)	Correct implementation with all features asked for and reasonable user interface. Quality of code written.	20%
Implementation (list visits function)	As above.	15%
Implementation (map function)	As above.	15%
Testing	You provided evidence of testing using browser developer tools with some discussion in the report of how you undertook the testing.	10%
Flair	I am looking for applications that show flair, creativity or innovation. This addresses the Appendix AA 80% to100% assessment criteria that state: "and will **more** than completely fulfil the functional requirements." This could include using one of the following: IndexdDB instead of WebSQL; RWD that supports tablets as well as smartphones; more sophisticated use of the map and map pins.	20%