**Course**: M.Sc. in Computing (Communications Software)

**Module**: Mobile Web Development

**Name**: Michael O Brien

**Student Number**: 98273850

**Assignment 2 -** Photoshare App

The app I designed and developed is called photoshare and is a take on instagram. The app can be used to take a photo using the camera of the smartphone and uploaded to the photoshare server. The photo can then be viewed by other photoshare users. The user can also upload an image file from their smartphone library.

Another feature of the app is to record the geolocation of photos as they are taken using the app and display a map showing the location of each of the photos.

On the social media side, users can log into the photoshare app using their twitter or facebook accounts. On successfully logging in the using one of these social networks, the ability to share on the corresponding social network is unlocked.

**Photoshare Design and Development**

The screens of Photshare were designed using wireframes on the balsamiq.com site. These can be found in the wireframes.png file. The app incorporates 5 screens as follows:

* **Home** – displays the latest images uploaded to photoshare server
* **Capture** – provide functionality for the user to take a picture using the device camera, select a picture from the device library, upload the recently taken picture or the selected one from the device library.
* **Share** – the user can share a photo on twitter or facebook
* **Map** – a map displaying the geolocation of the photos taken using the photoshare app
* **Info** – display some general information.

**App versions and technologies**

The photoshare app was developed as a mobile web app and as a hybrid android app. The mobile web app was developed using html 5, css3, jQuery mobile, jQuery, iscroll, nginx, node.js, backbone.js, underscore.js and mondodb. A hybrid version of the photoshare app was also developed for android devices. This was implemented using cordova, specifically version 1.6.0 in conjunction with the mobile web version of the app.

A hybrid version of the photoshare app was also developed for android devices. This was implemented using cordova, specifically version 1.6.0. Flurry is used in the hybrid app to record user activity.

**Hosting**

The mobile web app is deployed to an Amazon ec2 micro instance where nginx is running as the proxy to a node.js server. The mobile web app can be accessed at <http://ec2-46-51-152-146.eu-west-1.compute.amazonaws.com/photoshare/>.

The hybrid, android app is uploaded to amazon s3 and can be downloaded from <https://s3-eu-west-1.amazonaws.com/mickobrien99photoshare/photoshare.apk> However, the default settings on android devices is not to allow installation of apps that are not downloaded from the android marketplace.

Amazon s3 storage is used by the app to store pictures.

**Analytics**

Google Analytics is used in the mobile web app to record user activity.

Flurry is used in the hybrid app to record user activity.

**Source Code**

The source code of the app is preserved in github.com at <https://github.com/mickobrien99/mwd-photoshare>

**Testing and Devices used**

I used a laptop with Windows 7 for development. For the development/testing of the mobile web app I used desktop safari 5.1.5 and firefox 12.0, both on the laptop. I also used safari on my iphone 4 (iOS 5.1) and using the browser on an HTC Vision (Android 2.3.7). I do not have an Apple Developer licence so could not develop for my iphone, however I did create an android hybrid app and tested this on the HTC Vision.

**Achievements and Issues**

During the development of the app a number of obstacles were encountered some of which were solved and others which were not.

Google analytics was quite easy to set up however this could not be used for a hybrid app as the app is not running in a browser. Flurry is the option chosen to capture analytics for the hybrid app however, the instructions on the flurry website advise to add a couple of lines of code to the Activity class. However, as it is a hybrid app and not a native app I had difficulty finding the correct location to add this code.

The main feature of the photoshare app is to take a photo using the device camera nad upload it to the photoshare server. After some tinkering with cordova the functionality to take a photo with the camera and display it in the app was achieved. However, I had a lot of difficulty trying to upload the image file to the photoshare server. Using the cordova ft.upload function to upload the image file to the server, I could see that the call was sending data to my node.js server. I set up an api method to try and take this image file but after over a week of trying and searching the web for some help I was unable to process the incoming image file. My plan was to save the image file locally to the server, temporarily, and then upload it to amazon s3 using the knox module for node.js.

The feature to allow the user to select a photo from their library to upload is not functioning correctly. Using the cordova library the user can tap the Select from library button to select a file from their library however, on my HTC Vision this then gave an error that it could not find the file. I feel this may be due to the images being stored on an sdcard and not on the device itself.

The social network integration allows the user to log into the app using their facebook or twitter account. I managed to get this working perfectly when running locally on desktop safari. The user was then able to post a message to face book or a tweet to twitter. I encountered a problem with this when I deployed the app to amazon ec2 and I was not able to fix this. When selecting to log in with either facebook or twitter, the request does not get to node.js to process.

To display the latest images to the user on the Home tab, I needed to get the images in the node.js server from amazon s3 storage and transfer them onto the user. However as I was unable to get the streaming working for uploading the images I was unable to get it working for downloading them also. Currently the home sreen displays random images that are available in the Amazon s3 bucket <https://s3-eu-west-1.amazonaws.com/mickobrien99photoshare/>...

Setting up the Amazon ec2 instance took some time. This was due to the micro instance having a very small amount of computing power for the software that needed to be installed. I set up a git repository to store my code and transfer it to the ec2 instance. I had a number of problems with the git repository when I was deleting files but managed to overcome these. Overall I found the amazon service very good and somewhat easy to use after the learning curve is overcome.