David Slupko (BFEI Student)

20192323

**Mobile Technology Essay**

***Mobile application lifecycle***

The first step of developing a mobile is to do market research. Before you can move your app into the design and development stage you should first do some deep research into your own business and target audience and also research and take note of all your competitors. Once this is done gather your development team and other relevant departments and give them the information you have recorded. This is done to get feedback so that is would make a clear path for the development process.

The next step is to establish the goals and objectives of the mobile application. Once your goals and objectives have been determined the app will proceed into important objectives in your mobile marketing scheme. This step is very important int the planning stage of the mobile app. You need to set objectives such as: What queries will your app solve for your users? What features will your app have? What makes your mobile app so appealing?

The next step is setting up wireframes and storyboards. This is the stage where you make the layout all the pages of your app. You should have a pretty good idea about what the various pages in your app will look like and what features you to include. Once you have this figured out your goal is to create wireframes of the app and develop a clear picture.

The next step is to define the backend of your app. The wireframes and storyboards should serve as a guide for the backend structures you will need for your app, things such as API’s, data diagrams, servers and data integration.

The next step is to finalise your apps wireframe and to test the prototype. At this stage, you should make your final changes to your wireframes which might have been identified during the backend planning process. Once that is done you should build an interactive prototype and put through testing.

Then comes the development stage. At this phase, your designer will make the app screens used in your mobile app. The high-end wireframes that you made will be used to represent the interface of your app that your customers will finally use to interact with the app, your app should include all the idea and feedback that was recorded in the earlier testing phase.

After the development stage, it’s time to test again. In this stage, you need to thoroughly test your app in a diversity of real-world scenarios to check for any technical errors and to correct them. Make sure you go through your original design and planning documents and go through every feature to check for any mistakes or errors. In the testing phase, it’s a good idea to observe the user as they test the app as they might question and or feedback tweaks that aren’t that obvious. Once you are content in your app in all scenarios and happy with the design and usability, its time to make final arrangements.

The next stage is to prepare for the launch of the app. This may have the greatest effect on your app’s success. In this phase, it's important to get your marketing department involved early in this process. The market department will help with keyword research, which is very important for SEO and app store optimization. The research you gain from the keyword research will help guide your choice for an app title. It is also important to get good quality screenshots of the app and a promotional video for your app. In this stage you will also need a website or a landing page, these will be used to support and promote your app. The components your app’s landing page or website should include your apps name and icon, icons and names of the stores that can be used to download your app, the screenshots and promo video should be included and contact and support information.

The final phase is the official release. The official release date should be the peak of the app marketing efforts. In this phase you should be creating some write-ups and articles by known bloggers and journalists, an announcement should be made to everyone who was interested in your app prior to launch. You should have a clear station for feedback and respond to any comments or concerns your users have. Updates should be done monthly to help retain your users.

***Phone Gap API***

We use a phone gap API to allow our applications to use the PhoneGap Build web service so that you could create, build and download PhoneGap apps. This API easily integrates into IDEs, shell scripts, app builders and elsewhere. In an app, there are parameters required. Title must be given to your app as it takes precedence when specified in the XML package. The app is created by either uploading a file with the app content in it or you have a remote repository with the app content in it.

Signing your builds on PhoneGap Build, you must first upload a couple of keys through the web interface or through POST. You can get all the keys that connect with your account by sending a GET request to the URL your app is connected to. In JSON (one method which is used to serialize and structure data) you use to build the server, you can specify in detail the keys by id per platform. To update description and version you do it via config.xml file. To update the build settings is to send a JSON object as the parameter data, options that are available in the JSON object is to debug and check the privacy to see if the app has restricted visibility.

Lastly, currently there are two competitive well-known OS’s in this era, IOS and Android. To assign keys for IOS, the following are required:

* A p12 certificate file
* A mobile provision file
* The password to access your certificate
* A title for your certificate-profile pair

And to assign Android Keys you need:

* A Keystore file
* The alias used for that Keystore
* Private or normal password which are both optional
* A title for your key

To conclude, Phonegap is created by Adobe System and is a software development framework that is used to develop mobile applications.

***PhoneGap’s ability to compile apps for various operating systems***

2013 was the year where things started to change. The release of iPhone 5 began making hybrid path an option to choose. Thought-provoking, this was straight after Facebook abandonment of HTML5 for native.

Hybrid with a high percentage success rate works very well, as the majority of the businesses do not target making high performance, graphics-heavy game apps. They target the average human population and so they create, read, update and delete data and add a bit of UX sprinkled on top.

Apps that are created with PhoneGap can be assigned to different application stores such as App store on IOS(Apple) and downloaded on the user's device like any other native application. Each merchant(vendor) administers a different toolchain, and every PhoneGap release is functional with a specific set of tools. This is the power of PhoneGap.

Native SDKs are needed in apps that are developed locally and maintaining multiple SDKs are time-consuming. PhoneGap handles this in seconds and there is no stress and is minimal-effort! PhoneGap Build will eliminate the required SDK.

To conclude, PhoneGap provides a bridge between JavaScript and the native. This helps the application to communicate with the native operating system without writing any native code. A PhoneGap applications output is a binary application archive that contains all the necessary HTML, CSS, & JavaScript assets for you to function.

***Include user interface considerations and deployment requirements***

Using HTML, CSS, and JavaScript the user interface is developed for PhoneGap applications.

A PhoneGap application's UI layer is a web browser view which accounts for 100 percent of device width and 100 percent of device height.

Think of this as a browser that is "chromeless". It allows HTML content, without a standard web browsers "chrome" or window decoration. To take advantage of this area, you build your framework, and integrate navigational/interactive/content elements and chrome framework into your user interface based on HTML and CSS.

PhoneGap's web view is the same Web view as the native operating system uses. IOS uses Objective-C UIWebView class and Android uses android.webkit.WebView. As there are variations between operating systems in the web view rendering engines, make sure that you account for this in the implementation of the UI.

To upload onto PhoneGap, you should not upload any native application code files such as java or .c. Have no plugin files or .js as PhoneGap Build will inject any files required by your plugins, into the root www. It is because these files differ depending on PhoneGap versions and whatever plugins you use. However, you do need to source phonegap.js or cordova.js from your HTML files (both are available and are identical).

Minimum requirement for your application is to contain an index.html and it will be packaged as an app. For example, you can upload a file called index.html and it will be packaged as an app.

Although this is most likely a minimum requirement, you will need a much more complex framework that will include your javascript files, splash screens and icon images, media files, and any other payloads that your device needs.

.pgbomit is a folder that you use to store any files needed during the PhoneGap Build until the compile step. A common case of usage is a directory that includes the icons and splash screens for an app. .pgbomit file is a placeholder file only, its function is to highlight a directory and the file itself is not read.

And lastly to configure your application, you'll need a config.xml and you’re app package to configure how the app will be built. This could be PhoneGap version, splash screens, icons and platforms.