TOOL CREATION- UCEER

1. TECHNOLOGY

Technology: iOS SDK
Development Environment: Xcode
Programming Language: Objective C
Target Operating System: iOS7
Target Device: iPad

Compatible Devices with iOS7: -

- iPad2
- Third generation iPad
- Fourth generation iPad
- iPad Mini

Developer License Type: - University

2. IMPLEMENTATION CHOICES

From the discussions with our client Bob Wong, it was understood that iOS was the client's preferred technology and iPads (mobile) was the target deployment device.

Choices for mobile platform implementation: -

1. Web Application
2. Native iOS application written in Objective C language.

We decide to implement UCEER as a native iOS application.

Why we choose to go Native instead of a simple web application?

Short version: -

Primary reason- The UCEER project heavily depends on accessing the senses of the phone (camera, quick access to memory). Web applications have limited access to the device capabilities compared to a native application.**

Secondary reason- The User Interface experience

** Although the getUserMedia API was recently introduced into HTML5, which gives mobile web apps the capability to access the device camera capabilities, it is only a recent introduction (Dec 2013) and still doesn’t support all browsers. So it was deemed best not to take a chance with it before it has become well established.
LONG VERSION: -

The option of a web application implementation, although a first choice due to easier development, was dropped. The considerations are as follows:-

THE UPSIDES OF DOING A WEB APPLICATIONS FOR UCEER: -

All you need is a browser, such as Mobile Safari or Google Chrome.

They are cross-platform by default since the application runs in a browser.

Another advantage is the learning curve to create web applications. If you are familiar with web development, then you will be up and running in no time.

DOWNSIDES OF WEB APPLICATIONS THAT WE COULDN’T IGNORE: -

A web application has limited access to the capabilities of the devices they run on. Mobile web applications are quite powerful thanks to HTML5 and JavaScript. However, there are distinct restrictions in terms of what they can do.

Also, web applications typically must require a network connection to work.

Lastly, perhaps the biggest downside to exclusively building web applications is that, in general, they will feel significantly slower than native applications. The responsiveness of web apps is constantly improving, but the inherited delay added by the browser should not be overlooked or ignored.

Limited access to device capabilities translates to a lesser quality of the user interface and it was felt that an application with a lesser quality user interface was not the kind of products The Gapp Lab wants to deliver to its client.

GOING NATIVE iOS!

iOS SDK:

The UCEER application not only required access to the device camera but also Bob hinted that the value that he was expecting EAE to add to the app was a playful, well-designed and attractive user interface. Choosing to develop with the native SDK seemed to be the best choice to create applications that stand out and take full advantage of the device’s software and hardware capabilities.

Although this can be more work and complexity than developing a web app, opting for a native application such as iOS has the advantage of working in a robust development environment and that we can rely on Apple’s development tools, utilities, and support.

Unlike web apps, native apps are installed physically on a device and are therefore always available to the user, even when the device is in Airplane mode or when the Internet is unavailable.
3. APP FLOW AND FUNCTIONALITIES

From the above chart of the application flow, the UCEER application will have the following functionalities:

1. USER INTERFACE
2. FORM DATA INPUT
3. VIDEO PLAYBACK
4. VIDEO RECORDING
5. DATA COLLECTION
6. DATA UPLOADING TO A WEB SERVICE [BOTH VIDEO AND FORM DATA]
7. DYNAMIC DATA VISUALIZATION VIA GRAPHS, BAR, PIE CHARTS ETC

1, 2, 3, 4, 5, 7 are within the capabilities of the iOS SDK itself.

6 is where we are going to have the application talk to a web data service such as iCloud, Dropbox etc

Options available:

iCloud, Dropbox, GoogleDrive

Options we are strongly considering in terms of priority:-

1. Dropbox
2. iCloud
3. GoogleDrive
UCEER APP BREAKDOWN

APP SCREENS:

1. Start screen
2. Profile/Declaration
3. Video Screen (w/ controls)
4. 6 outcome questions screen
5. 3 Binary questions [(1 TEXT AUDIO BINARY followed by Text/Audio OPEN ENDED AV) * 3]
6. RESULTS/DISCOVERY screen
7. 3 Open Ended Reflection Questions [Text/Audio Open Ended AV * 3]
8. FINISH: Thank you screen

FUNCTIONALITY FROM A USER PERSPECTIVE:

1. Ability to input data (Screens 2, 4, 5, 6, 7)
2. Video Playback (Screen 3)
3. Data Manipulation in the Discovery screen (Screen 6)

FUNCTIONALITY FROM AN IMPLEMENTATION PERSPECTIVE:

1. User Interface Screens with controls
2. Animation
3. Form data collection
4. Data Saving on device
5. Video playback
6. Video Recording
7. Audio playback (Questions)
8. Data Plotting based on input.