Assignment 2 Documentation

Design Documentation.

Design choices were made to ensure the application was easy to understand and operate, Once the user log’s in they are presented with their recent posts and a large single button that describes the next action. This is to help make clear what the user should do next. Originally, I had planned for all action to be automatic, but time and my own understanding limited this, so I opted for large clear buttons that are easy to follow.

Text is divided into clear and manageable chunks for the user to consume and ensuring a high contrast between the text and background colours was chosen to ensure readability. The text is also left justified to make it easy to read.

Colours were chosen to give a clean and easy on the eye’s look. Not too bright and not overly dark they were chosen for a simple design.

The entire designed to remain minimalistic to not distract the user all information presented is at the centre of the screen and easy to find.

Technical Documentation.

Three API’s where used in the app, Facebooks Graph API, The Key phrase extraction from Microsoft Azure Text Analytics API and Flickr’s Image search API.

The graph API is used to retrieve the user’s past 25 post’s the information from this is processed into a JSON object formatted to suit the requirements of the Text analytics API which only accepts JSON input. The Text analytics API returns another JSON object which Identifies the key phrases of each post which are then added to an array and then sorted, and the common key phrase returned. This phrase is then used for Flickr’s search API which returns 10 recent images with the key phrase in its tags.

The most difficult challenge that I over came was understanding how to implement the Text Analytics API as it wasn’t clear how to access the returned JSON file correctly. In the end I had it store in a global variable and used to second function to call the analysis that then used setTimeout to pause for a second before retuning the variable. A better method would have been to use a callback function or an asynchronous function, both of which I attempted but could never get to work properly and as eventually with the due date looming I had to settle for using buttons with .click to call each stage of the process individually.

It would have been better code and a much simpler user experience had I been able to properly implement working call-back function. I also would of like to build proper error handling and remove blank posts from the array if I had more time.