## Portfolio

## Student: Daniel Catania

## Student ID: 215189217

Contents

Portfolio 1

Student: Daniel Catania 1

Student ID: 215189217 1

Week 1 - Getting started 3

Xamarin 3

Cordova 4

Week 2 – Setting Up GitHub and IDE 6

Git Hub set up: 6

IDE 7

Week 3 –Event Handlers and UI 8

Week 4 – Debugging 11

Week 5 – Events 12

Week 6 – Exceptions 13

Week 7 – Data Persistence/ Data Storage 15

Week 8 – Networking 17

Week 9 – Security 20

Week 10 – Mobile Computing Conclusion 21

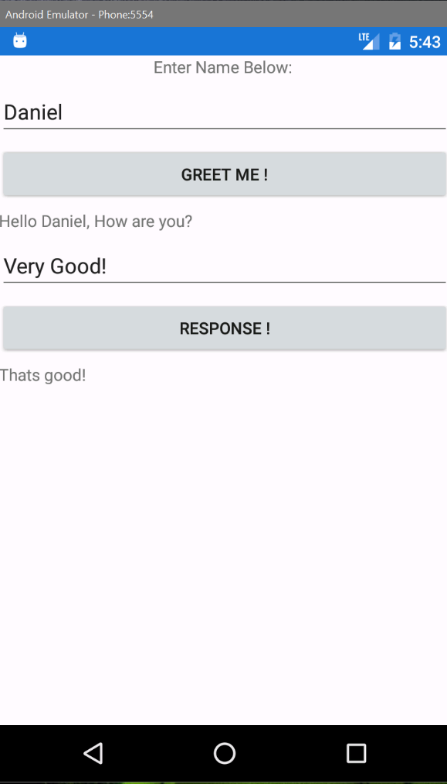
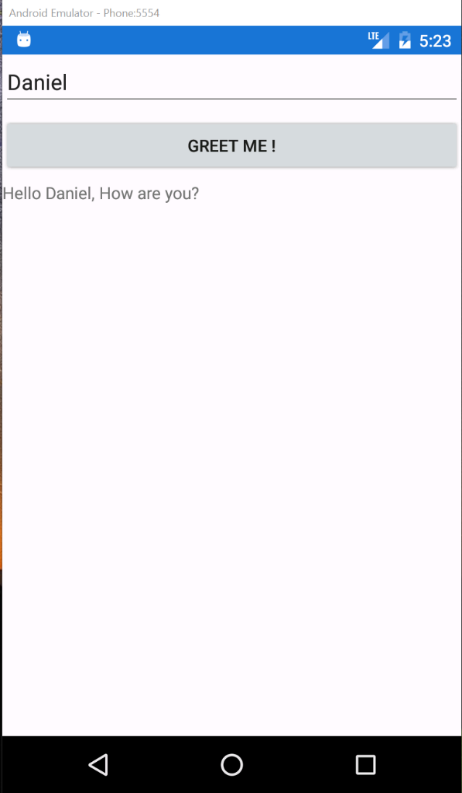
References – 23

# Week 1 - Getting started

To begin developing cross platform applications, programming software needs to be decided on, usually whatever the developer feels more comfortable with. They’re many types of programming languages, but within this portfolio, two will mentioned, one will be explained in more depth, and one will be used for examples throughout the document. These two programming languages include Xamarin, and Cordova.

## Xamarin

Xamarin uses C# and the chosen software to develop the application is Visual Studio’s, being a cross-platform based program. Below are examples of a basic functioning application using Xamarin.



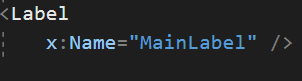
Here are code snippets from the above example, displaying how an input box, button and label are created in Visual Studio.



Input box:

## 

Button:

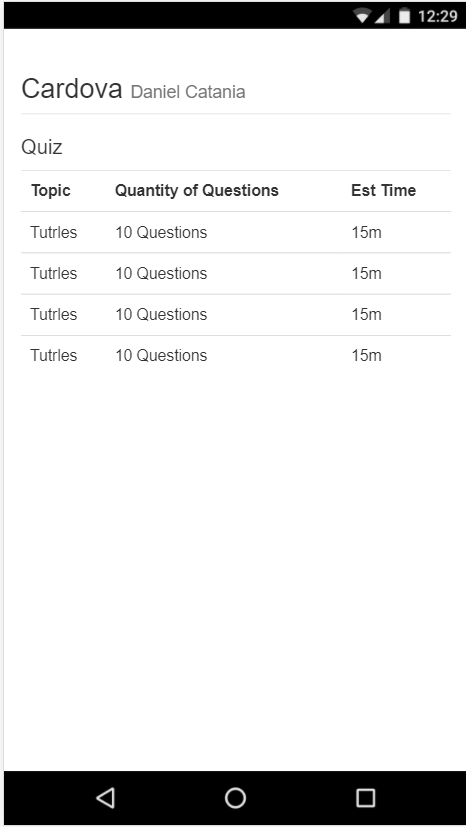
Label:

With Xamarin, compiling the application is as simple as downloading API’s, and implementing them into visual studio. Once the user has downloaded the required files and resources, simply choose a device that you want to test your application on, and compile.

## Cordova

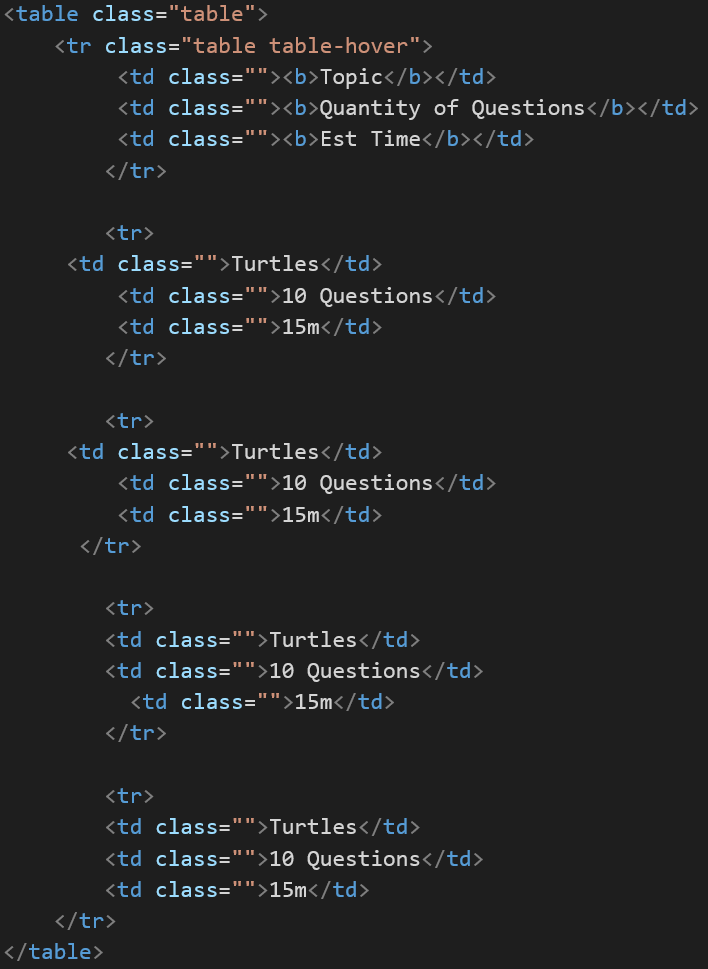
Cordova is a language that is programmed in Visual Studio Code, or BBEdit code, using HTML, JavaScript, PHP and CSS to create cross platform applications. Examples of developing a Cordova application are below, note that Bootstrap (CSS) can be used in order to make the user interface more user friendly.

This application is an example of a simple quiz User interface, with only the design being implemented.



With Cordova, they’re multiple ways of compiling, one can be as simple as compiling on a web browser (Chrome) and right clicking the page, choosing inspect, allowing the user to view their application in a phone layout and being able to view there errors through the console, or using terminal and either choosing IOS or Android to load your application into an emulator.

Example of a Cordova table:



|  |  |  |  |
| --- | --- | --- | --- |
| **Xamarin** | | **Cordova** | |
| Advantages | Disadvantages | Advantages | Disadvantages |
| * Can build all applications in visual studio * Can still be classified as native * Full hardware support * Open source technology * Simple maintenance and updating | * Smaller community, compared to IOS and Android * Need basic knowledge of native languages * Doesn’t support applications with high graphics | * Useful if you have a background with web development * Quick to prototype * Variety of library options * Built in terminal into compile applications | * A vast variety of different phone specifications * Complications with linking to other applications * Testing and debugging has to be manually done |

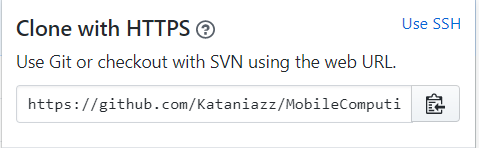
# Week 2 – Setting Up GitHub and IDE

Once a project has been created, the next step will be using an online source to backup your project. This process includes uploading your project files to an online backup source, for example Git Hub. Git hub allows other users as well as yourself, to view your project and use your application, at any location.

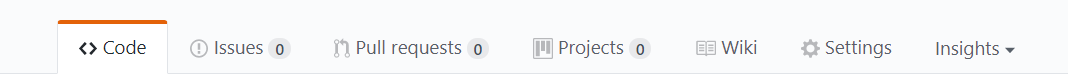
## Git Hub set up:

There’re a few steps in order to successfully upload a project to Git Hub, below is a step by step guide in order to achieve this. Once these steps have been completed, the user may push and pull their project from their Git hub account.

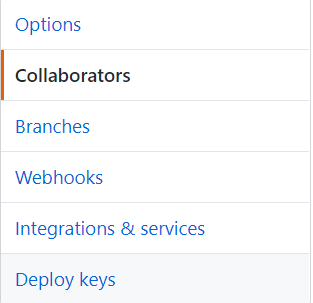
Steps:

1. Firstly, an account needs to be registered, this can be done by clicking on this link:
   1. <https://github.com/>
2. Secondly, a repository needs to be created, this being the storage space where you can access your project and all the included files. An already included file is the ReadMe document, this giving the viewer an overall about the project.
3. Once the above are completed, you need to open Git Bash, this is where the user will be able to set up their project into GitHub.
   1. Change working directory to the user’s local project
4. Initialize commend
   1. git init
5. Add files to repository
   1. git add .
6. Committing files
   1. git commit -m “First Commit”
7. find the link of the repository, for example [https://github.com/Kataniazz/MobileComputingAssignmentTwo.git](https://github.com/Kataniazz/MobileComputingAssignment.git).
   1. 
8. Add URL to Command prompt
   1. Git remote add origin URL Goes here
   2. Git remote -v
9. Finally, push the changes
   1. Git push origin master

Additionally, if you wanted to add another user to your repository, this can be easily done by

Clicking on settings,

Proceed to click on Collaborators,



Then search for either their username, or email address.

Lastly, additional Git processes that are important include:

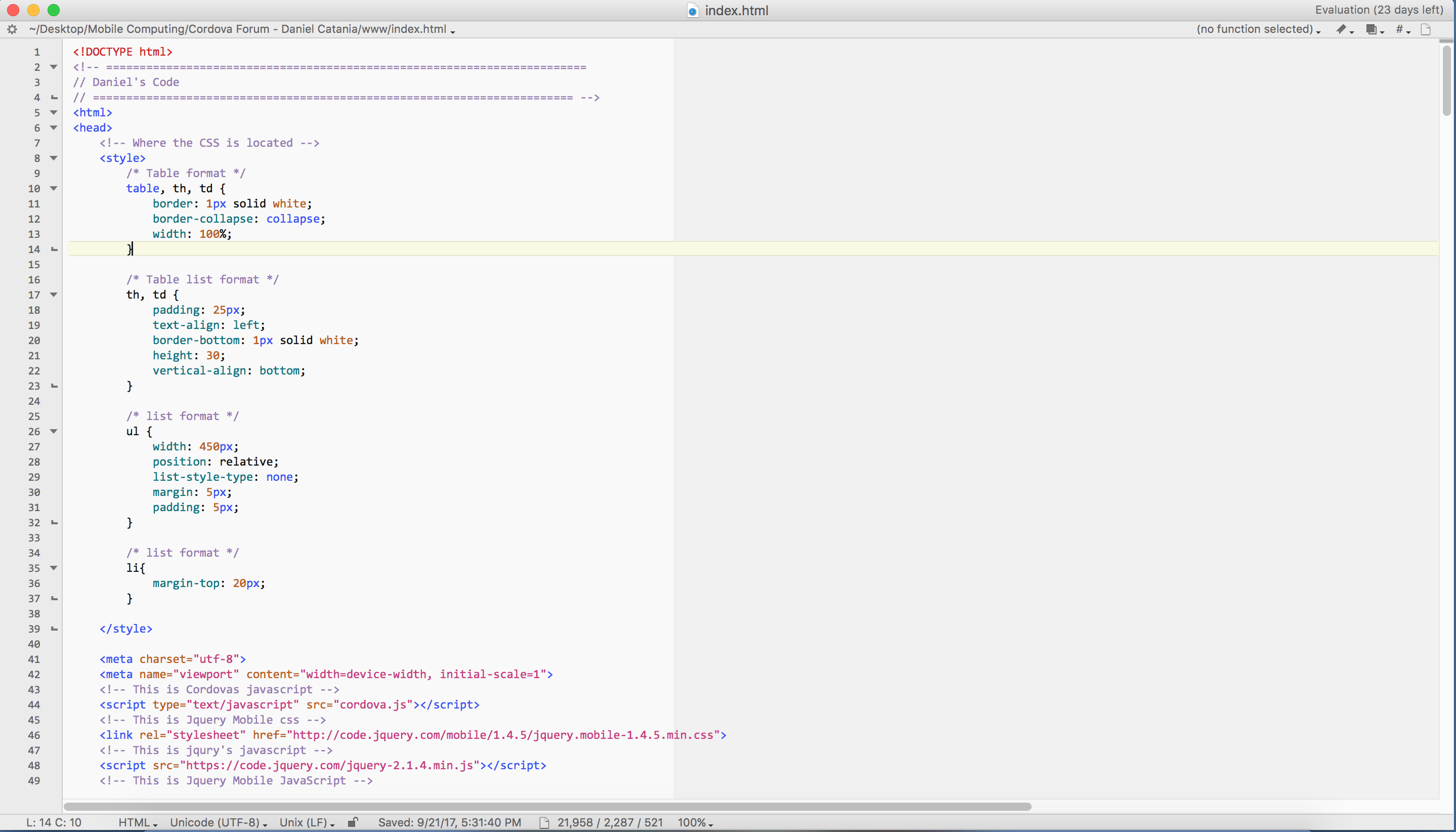
* Git cloning
  + Simply cloning an already existing git repository.
* Branches
  + A lightweight adjustable pointer to one of the commits.

## IDE

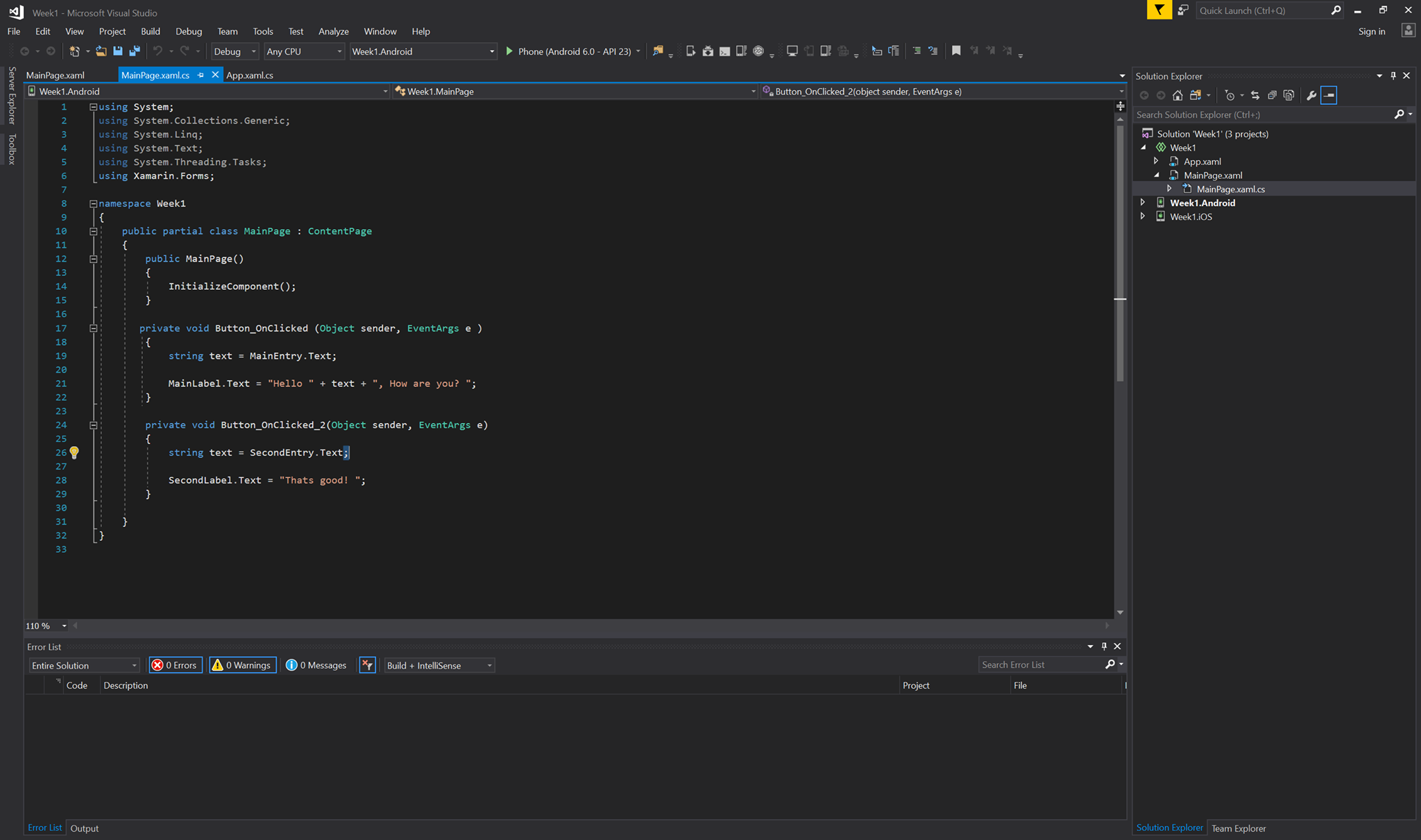
An IDE (integrated development environment) is software that allows the user to use its facilities, to create an application of there choice. An IDE usually consist of source code, compile automation software tools and a debugging system that guides the user to find their errors, and allows the build of their application.

An example of IDE’s for both Xamarin and Cordova are Visual Studio, and BBEdit, one allowing the language of C# to be used, and the other being HTML based. Below are screen shots on what both IDE’s user interface looks like.

Visual Studio Code:



Visual Studio:

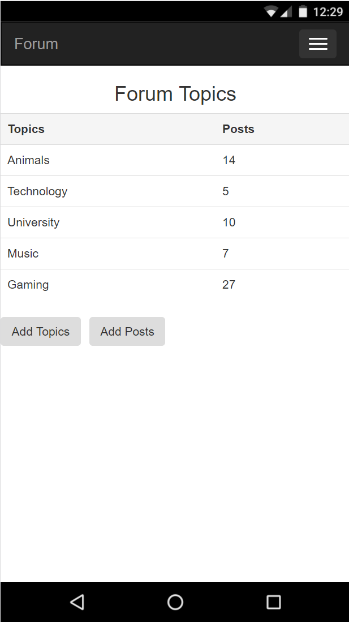
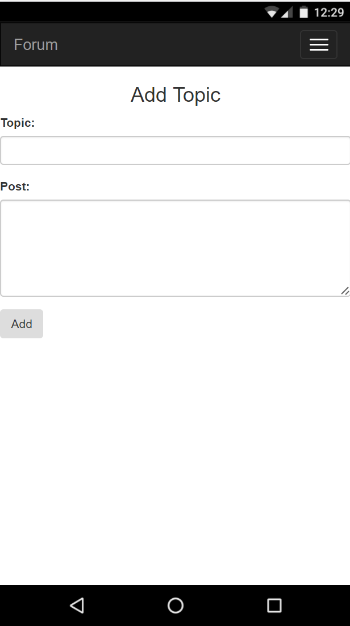
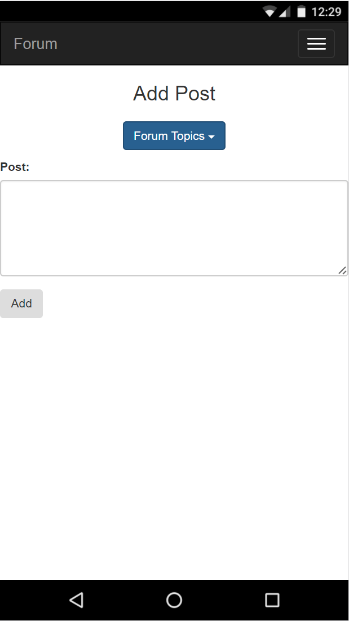
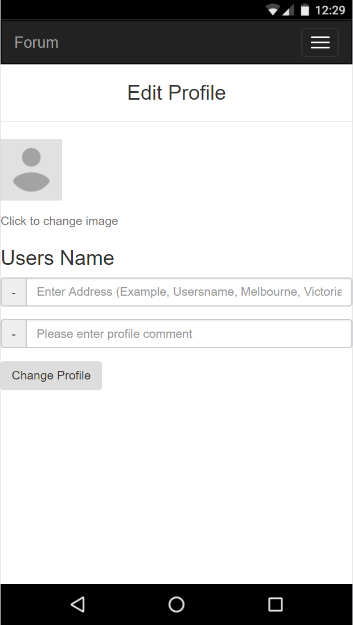
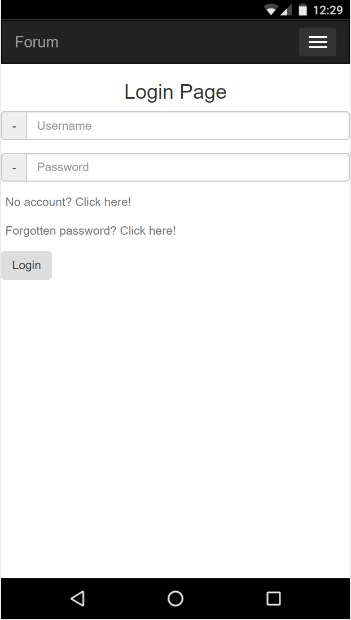
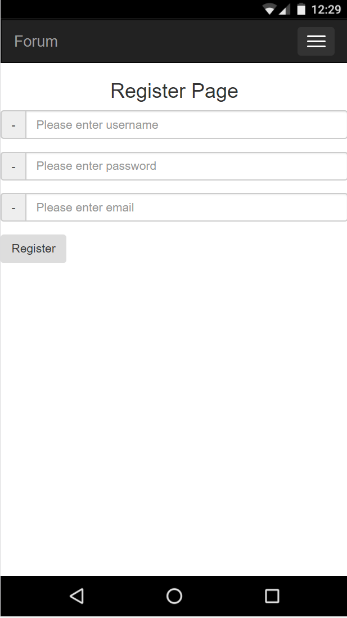
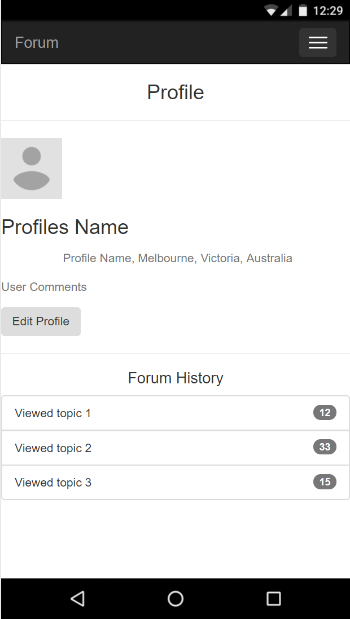
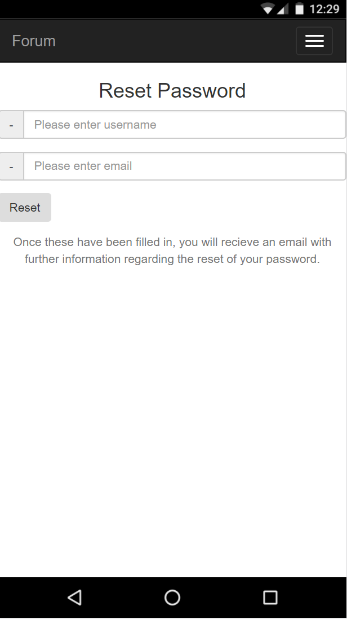


# Week 3 –Event Handlers and UI

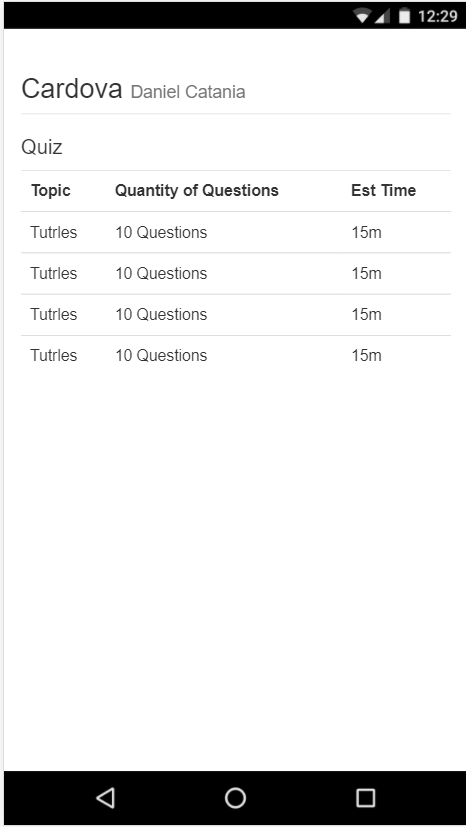
In order to create a fully functioning application in Cordova, event handlers are needed. Event handlers are simply anything within the application that allows user interaction, examples include a button, input box, search bars, etc. These will allow the user to navigate through the application. Basically without event handlers, the application will not function properly, this creating an inefficient connection to the user, this leading to poor user feedback.

What users always see when loading an application is the UI (user interface), basically allowing the user to use the application, the stronger the UI, the better the app. We need a UI because it’s the source of the user interaction, allowing the user to use the application. In the process of using UI’s, there are two types, static and dynamic. The static UI doesn’t depend on the user state, this meaning that if the user interacts with an event handler, for example a button, the action will not be applicable at that point. The dynamic UI does depend on the user state, so if the user locates a button, it will not be applicable, but after some sort of action, the button will appear or disappear.

What’s shown below is a basic forum application, with each page being simple and straight to the point, to avoid confusion. The only functionality this application has is the navigation, no storage and security has been implemented, the only intentions, are the UI.

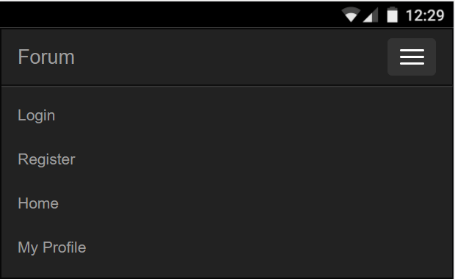
Example 1,

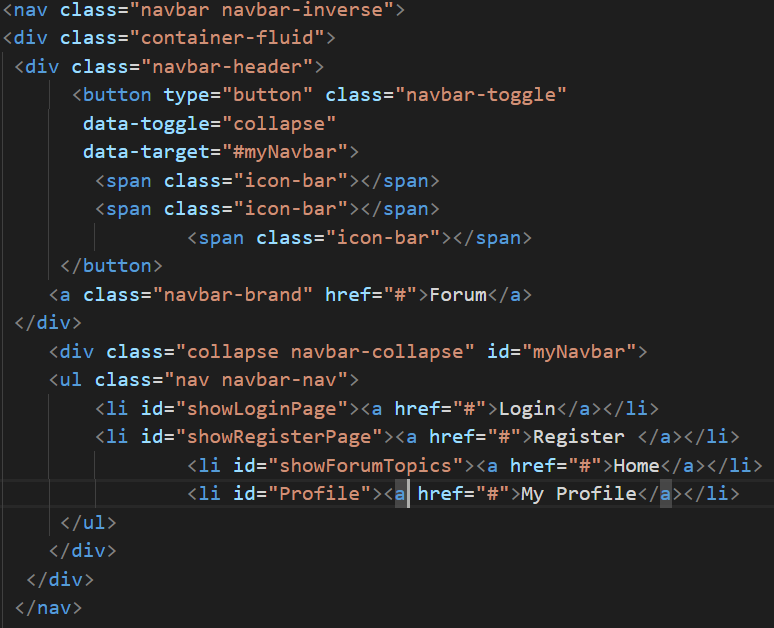
Below is an example of a poorly constructed application, the design being rushed, and none user friendly in the slightest way, the user doesn’t know how to function it as its not properly labelled, and has no source of navigation.



As mentioned in example 1, the only functionality within this application is the navigation, this being done by the navigation bar, which is located at the top right of every page. Below are snippets and images on how this is implanted,

Menu design,



Menu code,

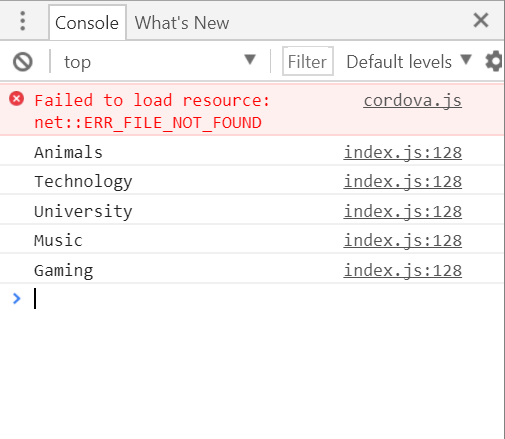
# Week 4 – Debugging

Throughout ones programming experience in Cordova, unfortunately, there’s going to be many errors, and while proof reading usually eliminates these errors, usually this will not be enough, so we need to take a more effective approach to this. This approach can be debugging, which is the process of going through your source code, inputting console’s or alert boxes, and removing/ editing all inefficient or corrupted code. If this is done correctly, your project will be functioning successfully without any errors. An advantage on debugging is simply to decrease the chance of getting an error, once the user familiarises themselves with this process, their application will be clean and organised.

So a simple way to debug areas of your project, is to use a console line,

console.log("Blah Blah");

This line of code will allow the user to find out what errors are displayed. Simply input this line of code within every function, if its displayed in the console, then that means your project is working, if it doesn’t, you’ve found your problem. Below is an example of what the console looks,



Note ~ the only reason why the error above is displayed, is because the simulator is being used in Chrome, instead of a proper emulator, therefore Cordova will not work.

# Week 5 – Events

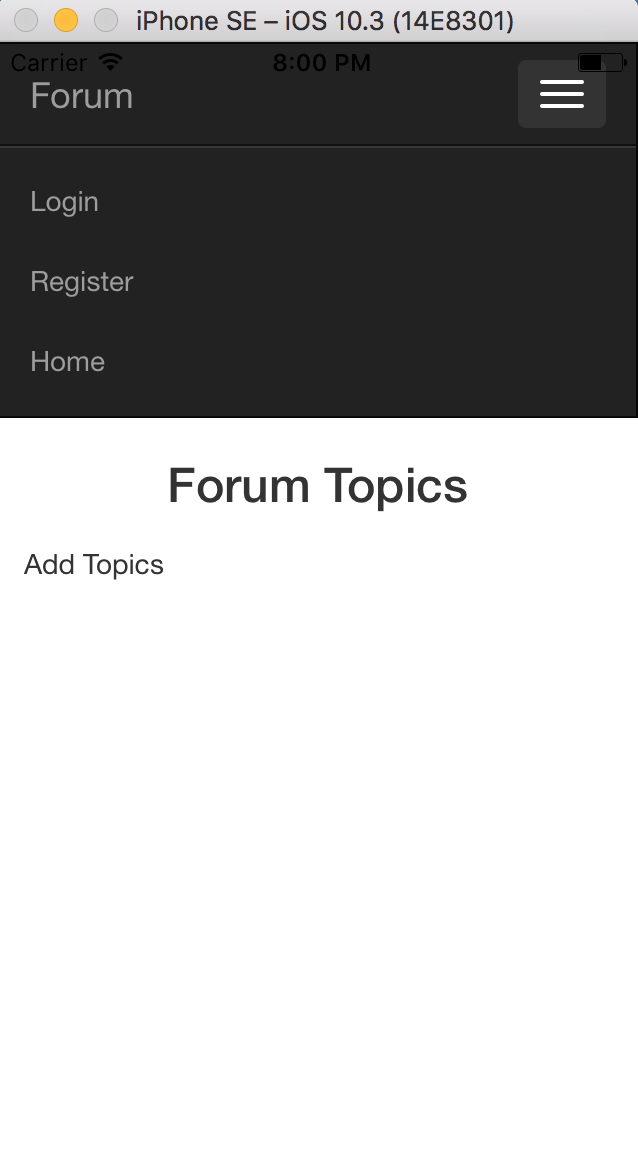
Events, in simple terms, are just whatever the user can physically interact with in the application, where actions from the outside world (finger tapping, keyboard, messages from other applications, etc.) are handled by the program as a sequence of events. Theoretically, without events, the user would be very limited in what they can do within the application. For example, if an application were created without any buttons or input boxes, navigation as well as user input would be eliminated; therefore a useless application is being created.

Coding event handlers can be very useful in terms of a successful functioning application. For example, when creating a forum application, in order for it to function properly it would need to save user inputs, this being as simple as saving forum topics and being able to view them for future use, and by linking the information to a database will allow this to work. Examples of events are shown below,

IOS Keyboard event, this event will allow the user to enter characters,



Touch events, being able to tap on certain areas of the screen and interacting with them. Clicking on this symbol will allow the user to navigate from the home page to different pages,



Using a program scope may also be helpful in terms of developing an application, this being the overall plan on how the application is going to look as well as function. The scope is important when using event handlers as well because you want to know what each handler’s purpose is. Without planning what does what, your application will be unorganized, as well as unsuccessful.

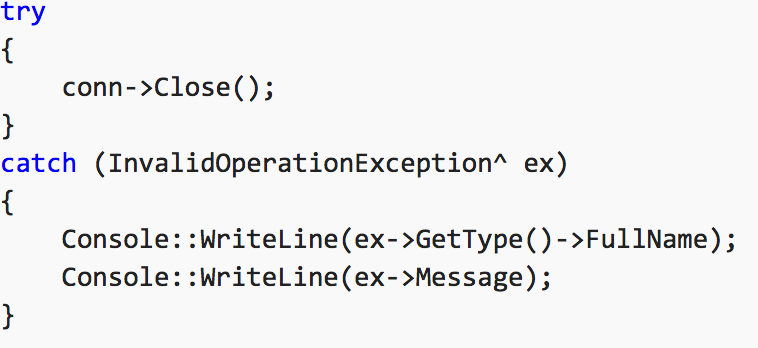
# Week 6 – Exceptions

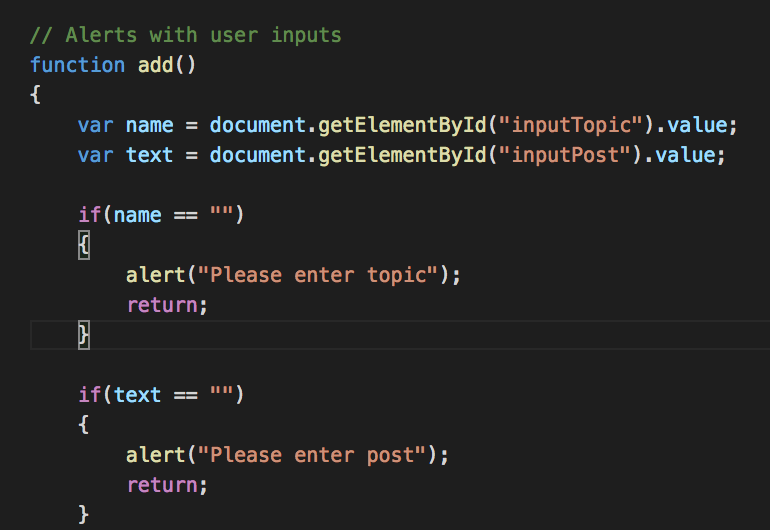
When programming with either Cordova or Xamarin, there’s always going to be problems, these either being errors with the network, software, or hardware. When something like this occurs, they’re a variety of ways to resolve this, if there were to be a software issue, a simple re download/ installation may resolve the issue, or downloading different packages/ files that may also help the problem is usually the solution.

List of potential exceptions,

* User error
* Application logical error
* Hardware/ API issues
* Network
* Syntax errors

Examples of error catching,

Try catch:

Alert boxes,

# Week 7 – Data Persistence/ Data Storage

This week we covered data persistence and data storage, data persistence being the data that’s being accessed and not likely to be modified, and data storage being the information being saved from the user input. The data persistence process is important for evaluating different data store systems. A poorly informed choice when dealing with data storage may lead into data loss, or substantial downtime. When using data persistence and data storage, the use of files and directories are needed. Moreover, a file, an object on a device that stores data and information with the devices program, will also be involved with local storage. The reason why we use files is because when using an application, in order to store, retrieve, delete and edit information, a file needs to be created in order for the application to retrieve the data and therefore allowing modifications. There’re two types of files, binary, and text files, binary files just include binary (a series of bytes), therefore majority of files are binary, but when using characters, symbols and numbers, it is known to be called a text file. And for directories, them simply being a system cataloging structure containing references to other computer files.

Some scenarios where data persistence can be very useful in terms of creating an application, is using databases or networks. As shown in local storage and networking, we display the use of both servers and databases. When wanting to deal with local storage (storing within the phones systems), a file needs to be created in order for data to be saved. But when dealing with saving data to a server, it works a little different.

Example of data persistence,

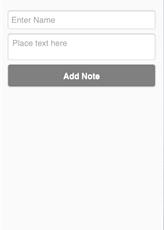
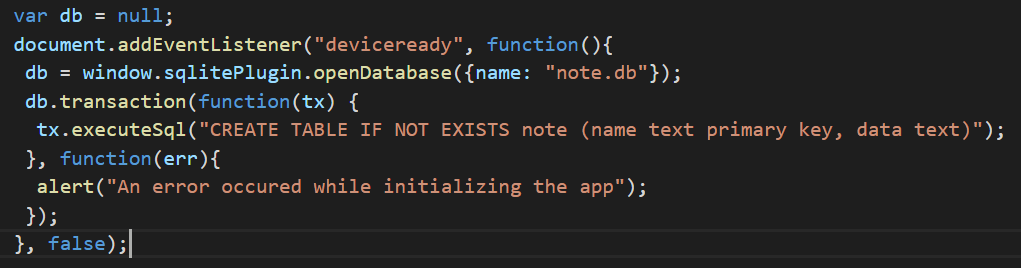
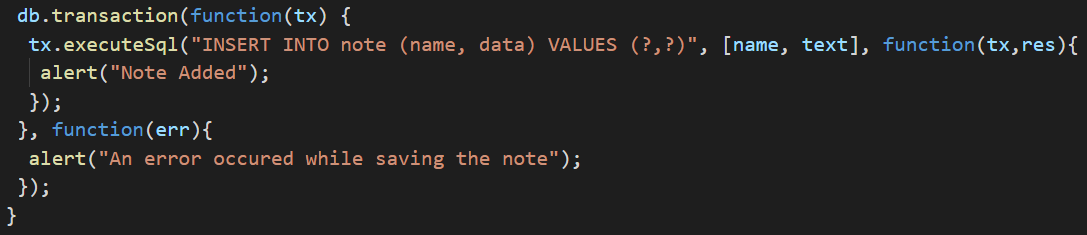


Image 1 Image 2 Image 3

Within this example, the user can input information such as a title, and a short comment (image 2), once that’s completed, it will be saved to the database, and recovered once the user clicks on the Display Notes button (image 3), allowing them to click on their note, viewing their comment. If inputted correctly, this project will allow an effective database to be created.



This will allow the creation of the database, as well as check if there is a table to store data or not, if the table doesn’t exist, then one is created.



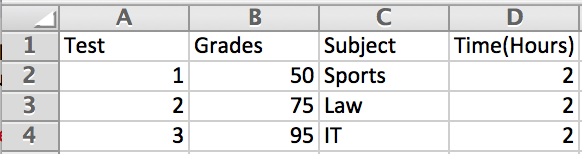
This snippet ensures that the input fields are not empty, and proceeds to add a row to the table. If everything goes according to plan, the data that the user enters will be saved into the table.

# Another important aspect of using a database is data retrieval, simply retrieving the stored data from the user inputs and displaying it somewhere on the application, either being another page, or the same page where the user has inputted the information.

Data serialization and data encoding are more processes included within the whole application development experience. Data serialization being the process of translating data structures into some sort of format that may be stored or transmitted, being reconstructed for future use on another device. And data encoding, being the process of sequencing characters into a specialized format for effective storage.

When dealing with information, there’s different ways to format the files they’re being saved to, for example, using CSV (comma-separated values), involving spreadsheets or databases. When using CSV’s, data can be imported to and exported from programs that store data in tables, such as Microsoft Word and Excel. And example of this is shown below. Another form of CSV is TSV (tabbed-separated values), this simply being just import or export text.

Example of CSV,



When dealing with devices and applications, escaped characters can be used; this allowing characters to invoke an alternative interpretation on subsequent characters within a character sequence. Examples of escaped characters in JavaScript are:

* Single quote \’
* Double quote \”
* Backslash \\
* New line \n
* Carriage return \r

If you don’t want to use either CSV or TSV, JSON (JavaScript Object Notation) can be another option, this simply being a lightweight data-interchange format.

# Week 8 – Networking

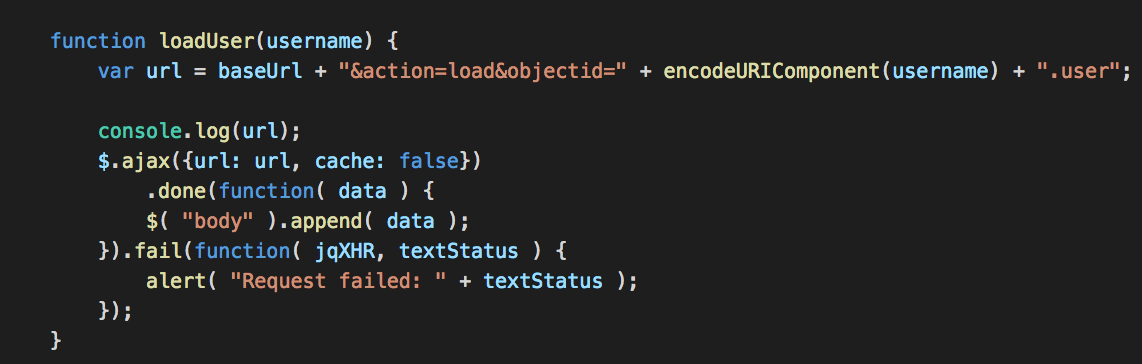
Another large influence when developing an application, is networking, this being because of how useful it can be when dealing with online user storage. When dealing with anything network related, usually there will be a source, and a destination, because as a user you want to create information as well as share or store it. The client, who usually creates the information, will want to store their information somewhere, this can be either locally as mentioned in week 7, or this destination could e the server, which will be explained this week. When using servers, we think the required storage area, where all information from the user input will be stored. Having a server will benefit the user in many ways; one of these ways includes having data sent through a network and always being able to access the information. The data wont be stored locally, therefore it will get to the desired destination almost immediately without hassle.

|  |  |  |  |
| --- | --- | --- | --- |
| **Networking** | | **Local Storage** | |
| Advantages | Disadvantages | Advantages | Disadvantages |
| * Easy communication speeds * Access at any location * Security | * Internet connection needed * Data loss may occur * Breakdowns * Security threats * Bandwidth issues | * Fast storage speeds * Full backup control | * Storage is limited * Expensive if more storage is added * If device is damage, so is the content |

When using networking, users usually have experiences with HTTP (hypertext transfer protocol), now this is just hypermedia information system with the involvement of the World Wide Web, using this for communication. When dealing with HTTP, users may come across requests, these are sent to the server containing information. The client usually provides the information, this being what they have chosen to input into the server, once that information has been created, it is then requested and sent to the server. Furthermore, URI component encoding relates to networking by replacing each character of certain numbers by one, two or three escaped sequences.

Moreover, http uses headers that allows the client and the server to pass additional information requests, this being useful because. Another area relating to networking is MIME (Multipurpose internet Mail Extensions), which identifies the nature of files according to their nature as well as their format. Lastly, they’re many HTTP codes that users may come across; these include the 1 1xx (informational response), 2 2xx (Success) and 3 3xx (Redirection), all-relating to HTTP.

Example of JSON, the loadUser function will allow the user to simply load a created user.

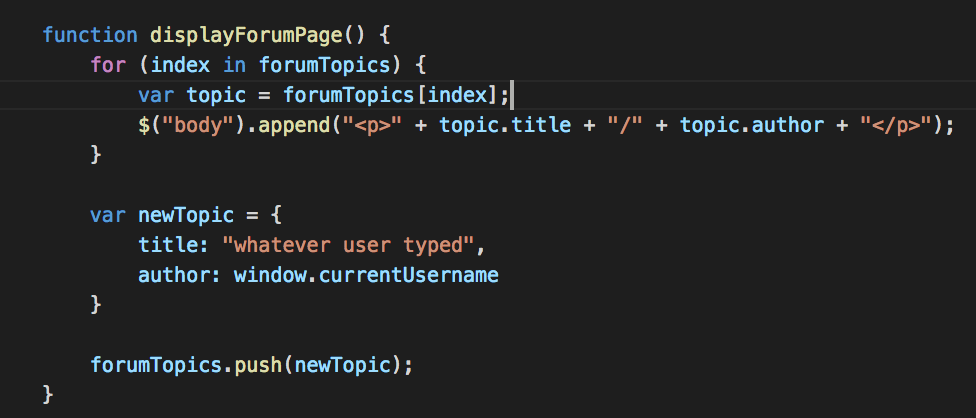


Below is the createUser function, allowing the user to create a user to be displayed,

The function loadForumTopics will simply display the created forum topics to the user,



The displayForumPage will display the forum page,



When all combined, a simple networking example is created, allowing the user to create users through logging in and registering.

# Week 9 – Security

Lastly, one of the most important areas of developing an application is making sure it’s secure. Without security, your application is prone for hacking, and data loss, but having a secure security method will prevent this. As explained earlier, they’re two ways to store the users information, local storage and networking. The risks of storing the information, such as a password, within a network or local storage are theft, loss and neglect.

A method to keep a users information safe is encryption. They’re two types of encryption, symmetric, being a secret key, which usually uses strings and random characters to change the content in a particular way. In order tov successfully use symmetric encryption, both parties need to know the secret key in order to decrypt it back to original form. The advantages of using symmetric encryption are that as long as the secret key isn’t misplaced then a successful transfer of information in a secure manner will occur. The disadvantage to this is that as mentioned earlier, if that secret key is lost, then the second party cannot decrypt the information.

And asymmetric, instead of using one key, uses two related keys. The first key being a public key that allows the users to send a message, and the second key, this being private which is kept in secret, so that only the user knows it. The advantages of using asymmetric encryption is not worrying about the passing of public keys over the internet, however, a disadvantage is the speed of the processes compared to symmetric, it required more processing power to both encrypt and decrypt the content of the information.

Another security method that may be of use when creating an application is one-way hashing. It involves a variable-length input string and converts it into some sort of binary sequence. An advantage of one-way hashing is that the process is hard to reverse, therefore making the information secure.

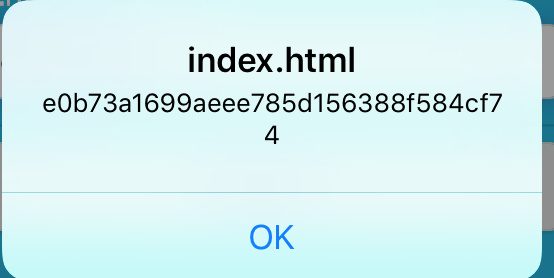
Example of Security,

Register example, below encrypts the user inputs from a register page, encrypting the password, when the password is used again for when the user logs in, the second example displays the decryption.

var EncryptedText = md5(registerPassword);

Login Example, as explained above, the password is encrypted on the server, and in order to re access it, the user needs to input the password on the login page. Once this has been completed, the user may log in as their password is decrypted.

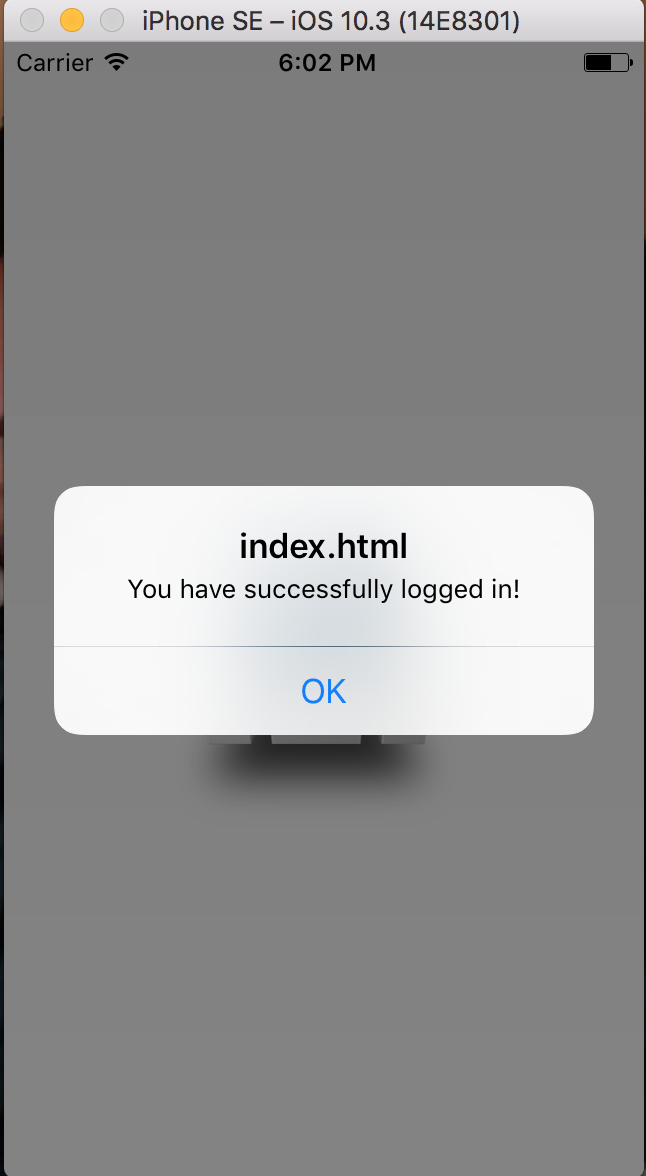
var EncryptedText = md5(loginPassword);

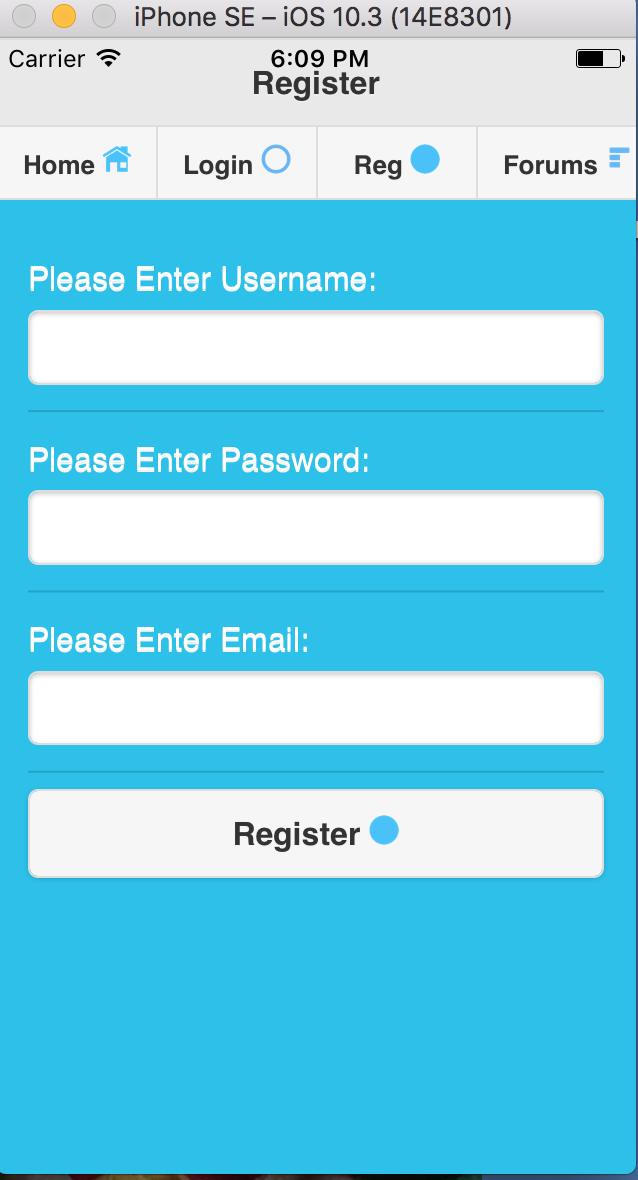
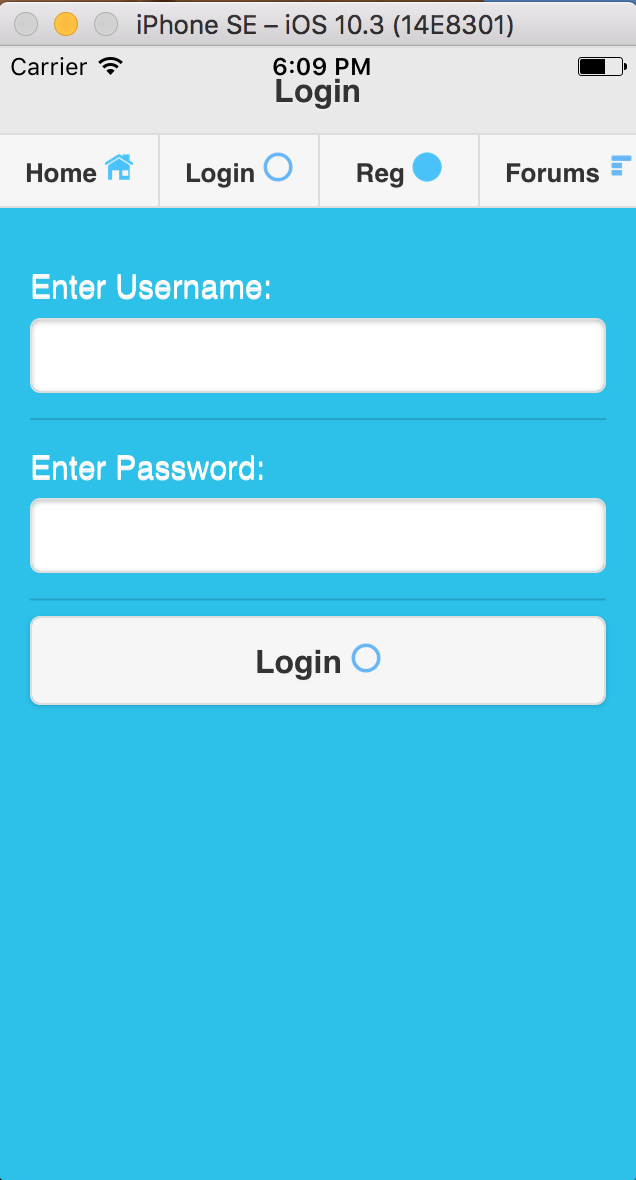
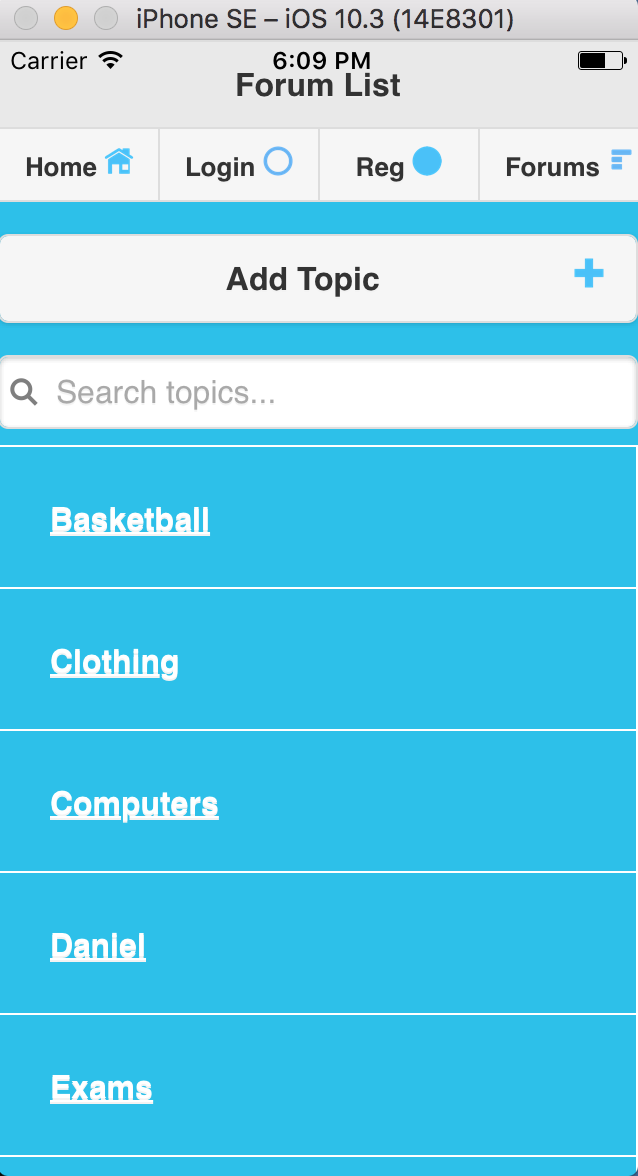
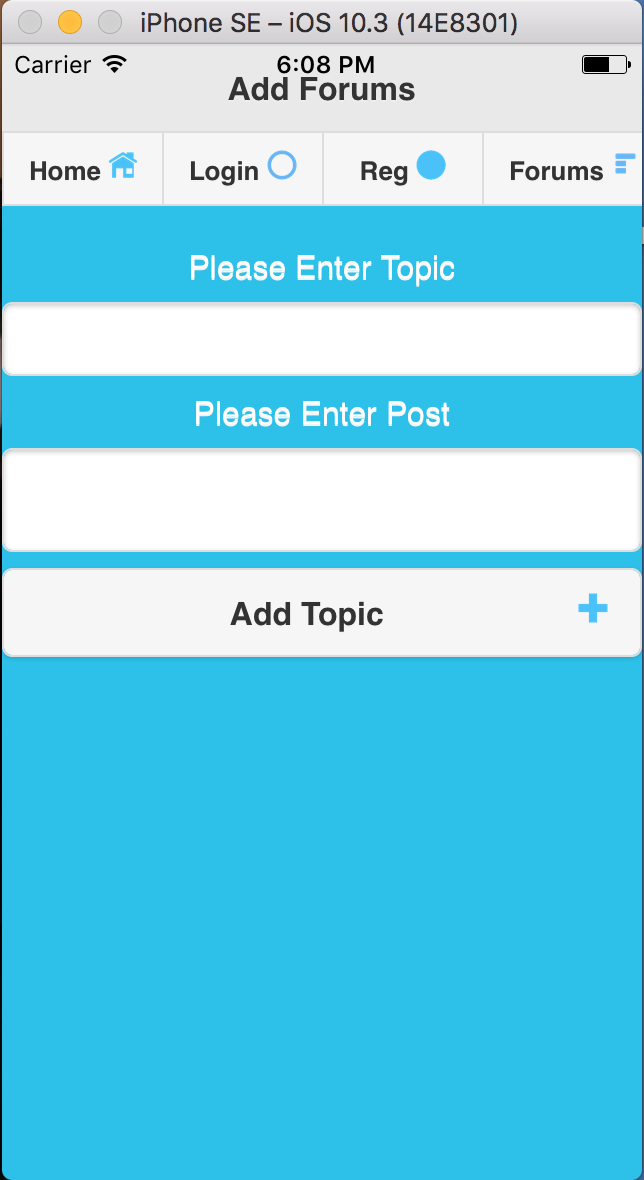
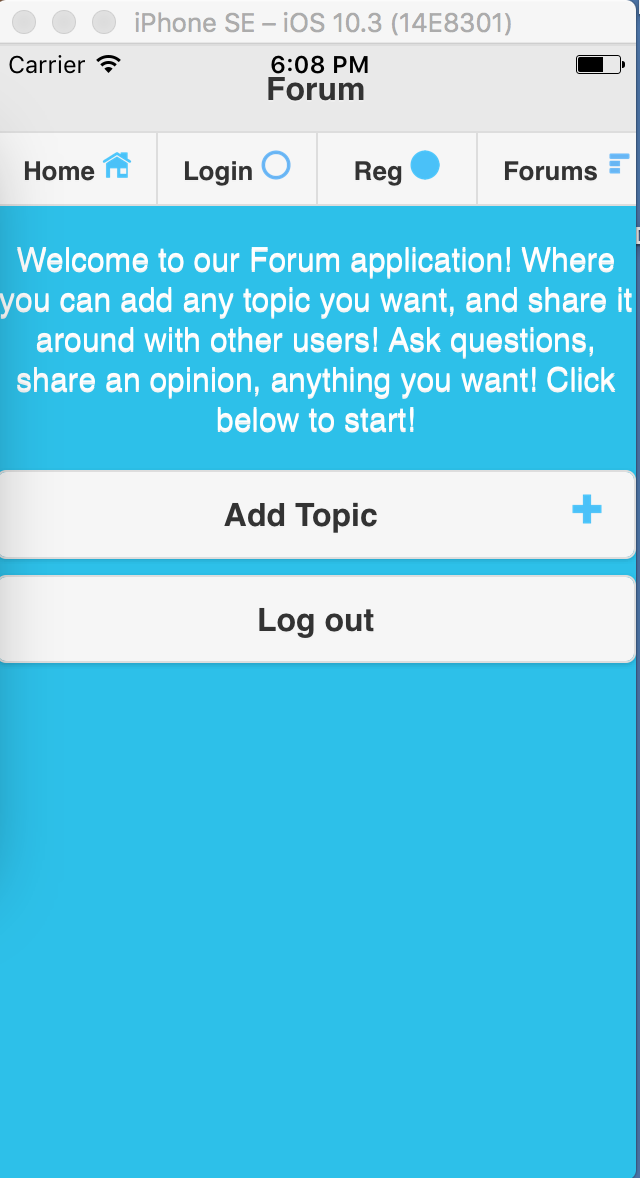
Below is an alert box that displays if the password has been encrypted, and what it’s changed to,

# Week 10 – Mobile Computing Conclusion

In conclusion, what’s explained within this portfolio is a step-by-step process of creating an application with key components, these being local storage, networking and security, being the main aspects.

Throughout this unit I have learnt many key lessons for developing a cross platform application. As categorized within this portfolio, there’re many steps in developing a cross-platform application, and all are just as important. From previous units, this unit’s way of learning is completely different; this is because it’s more independent than other units. Instead of providing students with weekly resources, they expect you to gather information about the topic by yourself, allowing students to use real world skills. Networking has been very challenging in terms of getting it to save to the server, as I’ve never used JSON within my previous applications, but has been very useful in terms of this assignment. In terms of code, I have researched myself. When developing an application, I would now start off with planning my application, this including designing the user interface, so I would know how the application will look, then ill proceed to create a empty project that will upload to Git Hub, so I can access and have a back up of the project. By doing all this ill have an organized, as well as an effective process with upcoming projects.

Below are images from the forum application I have created, with all of the above contents functionality,



# References –

Bootstrap: <http://getbootstrap.com/getting-started/>

Daniel Catania’s GitHub: <https://github.com/Kataniazz/MobileComputingAssignmentTwo.git>

Local storage: <https://www.sitepoint.com/storing-local-data-in-a-cordova-app/>

Security: [https://github.com/blueimp/JavaScript-MD5](https://l.facebook.com/l.php?u=https%3A%2F%2Fgithub.com%2Fblueimp%2FJavaScript-MD5&h=ATMIjMOyha5szIifskHOuEmGPh8Sx5AzNTMB8Pvt6RtGsgKczbNXUIzzvioqB0qP5ZxyVgjbXNH2boi_M4O9ERZF-Gpun8ASWxCOuF29azDvD0zcysaF-2TIJtoDQ3PwQpb1Xf0oYPerUfXHVgND8AW_Pn88XGGCrFw" \t "_blank)