**National College of Ireland**

**MSc in Web Technologies**

**2013/2014**

**ARIA**

**Project – Loyalty Card Holder**

**Entaria**

****

**Hugh Kelly 13117386**

**Adrian Mann 12110701**

**Brendan O’Brien 13122096**



Contents

[1 Motivation 3](#_Toc386883403)

[1.1 Project Scope 3](#_Toc386883404)

[1.2 Area of Contribution 3](#_Toc386883405)

[2 State Of The Art Review 4](#_Toc386883406)

[3 User Interface Design 14](#_Toc386883407)

[3.1 User Interaction Design Strategies 16](#_Toc386883408)

[3.2 User Testing 16](#_Toc386883409)

[4 Architecture 18](#_Toc386883410)

[4.1 Application Architecture 18](#_Toc386883411)

[4.2 Version Control 18](#_Toc386883412)

[4.3 Toolkits and Frameworks: 18](#_Toc386883413)

[4.4 Data Transfer Strategies: 18](#_Toc386883414)

[4.5 Evaluation and Testing: 19](#_Toc386883415)

[5 Summary 20](#_Toc386883416)

[References 21](#_Toc386883417)

# Motivation

## Project Scope

This project is a rich internet application that is intended to provide an interactive, responsive and modern front end to an enterprise application framework that is implemented for the business sector.

The scope of the project is to develop, in conjunction with the Enterprise Frameworks module, a cloud based server that will provide a loyalty card support system for small to medium sized businesses. The system shall allow businesses without IT specialism to partake in a loyalty card scheme without having to issues loyalty cards themselves.

In addition to this, and the focus for this module, we will design and implement a front end application that features many modern and upcoming frameworks that will be relevant in the year 2022.

The list of frameworks will include AngularJS which demonstrates a Model, View, Controller capability. We will also implement a number of APIs which will keep in theme with the offering of the application. We will also use the CanvasJS library to display json data to the user. There is also Google Map API features on the site. We have embedded a twitter feed to display relevant tweets. We also have created a form that posts data to a back end server.

## Area of Contribution

The team has identified a gap in the market for a Loyalty Card System that removes the need for multiple paper based cards and will satisfy many retailers across multiple store types. We aim to use an existing resource, in Iarnrod Eireann’s Leap Card, and this will benefit users and retailers alike.

We are interested in creating a front end application that users can interact with and one that will compel users to return to the application via interactivity, responsiveness and interesting features.

# State Of The Art Review

The rate of change of technologies used in the internet seems to be ever increasing. For example in 2007 Microsoft launched Silverlight as a framework to support RIAs but after quick acceptance in development it is now no longer seen as a practical framework for development of new projects, Microsoft have not made a major upgrade to it since May 2012 and Netflix, possibly the largest users of Silverlight, have announced that they will be moving to HTML5. AT this point in time, May 2014, the most important enablers for RIAs are HTML5, JavaScript and CSS3.

“This specification defines the 5th major revision of the core language of the World Wide Web: the Hypertext Mark-up Language (HTML). In this version, new features are introduced to help Web application authors, new elements are introduced based on research into prevailing authoring practices, and special attention has been given to defining clear conformance criteria for user agents in an effort to improve interoperability.” (W3C, 2014)

From this definition of HTML5 provided by the W3C working group we can see that more capability for rich internet applications has been included in HTML’s repertoire. HTML5 is the fifth generation of HTML, the mark-up language that has been used to describe the layout of web pages since the early 1990s. It is an elaboration of the previous generation of HTML designed to replace and extend HTML4, XHTML and DHTML. One of the main improvements of HTML5 over previous versions is its support for multimedia e.g. the introduction of support for <audio> and <video> tags (implying that HTML5 browsers will support the playback of media referred to via these tags). It also includes an API that can be called by JavaScript, this API allowing support for more advanced web applications than was possible heretofore (W3Schools, 2014).

Some of the capabilities supported by HTML5’s Web API include:

Web Storage: a simple key-value pair storage system that works in a similar way to cookies with a larger storage volume.

Geolocation: an API that can provide geographical position information (i.e. latitude and longitude co-ordinates) using various sources of location such as Wi-Fi (including other identifiable Wi-Fi hubs in range that are not necessarily being connected to), IP address, GPS and mobile phone cell ID depending on what is available to the browser from the client device

Web Sockets: a communication protocol that provides bi-directional (full duplex) traffic over one TCP/IP connection directly for within the browser.

Canvas: an API that allows for 2D drawing directly to the browser document.

Web Worker: an independently executed JavaScript thread that can run in the background suitable for use in browsers hosted on modern multicore systems.

WebRTC: Web Real-time Communication allows browser-to-browser connectivity to allow voice messaging, IM chat, video conferencing, etc. directly between browsers in a similar way to Skype connectivity but without using a third party application or plugin.

The increased capabilities of HTML5 are ideal for RIAs and reduce the need for ‘thick’ third party plugin-like frameworks such as Flash, Flex or Silverlight.



CSS is a simple and flexible way to style Web Pages and provide a number of feature which can be easily implemented. CSS3 is built on CSS2 and is backwards compatible. CSS3 is modular based and this allows CSS3 to be backwards compatible with CSS2 as each of these modules extend the CSS capabilities. CSS3 is the standard in effect of W3 working group.

CSS3 does not require any special tools to develop and can be easily integrate into web page design. CSS and HTML can be easily integrate with each other to enhance the visuals of a web page.

CSS3 supports the following.

·         Animations

·         User Interface

·         Text Effects

·         2D/3D Transformations

·         Box Model

·         Multiple Column Layout

·         Image Values and Replaced Content.

·         Selectors

·         Rounded Corners

CSS3 allows developer to implement some nice special effect such as Transformation and we have integrated a Spinning Cube which has an image on all four sides to show this capability.

CSS3 allows designers to implement some very complex manipulation with very little programming in JavaScript, which most designers shy away from. CSS3 allows programmers to create stylish pages using the CSS3 scripts. A designer can write 1 line of code which can replace 10 lines of code.

Media Queries take care some of the issues with view the web pages on different devices such as mobile, tablet or desktop. They can detect the device type based on the device attributes so the correct style sheet can be targeted directly to the device viewing the page.

CSS3 style sheets should be kept in separate files which allow these files to be compressed and cached on the browser side.

CSS3 is not HTML5 and vice versa HTML5 has a lot of great features but CSS3 is what does the really spectacular effects such as Rotations, scaling and animations. CSS3 continues to be developed using modules which allow more feature to be implemented on an ongoing basis.



Figure 2.1 CSS3 cube

JavaScript is not actually based on Java and this is a misleading in it description. JavaScript is the Client scripting Language of the Web.

JavaScript was developed by Netscape in 1995 and has now been standardised by EMCA and now JavaScript 5. JavaScript is a Curley Bracketed Language JavaScript that treat Functions as First Class Object.

JavaScript does not support inheritance in the OOP's manner as it does not have classes but does would prototypes and with some coding can get around the inheritance issue.

It is one of the top Ten Programming Language in the world and the web as we know it today could not exist without it.

It has spawned quite a number of libraries based around it and any problem you can think of usually has a JavaScript library to implement it or get around the problem.

JavaScript is supported by every Modern Browser.

JavaScript allows the browser perform computations functions within the browser and reduced the workload on the backend server.

JavaScript can manipulate the Dom, handle events and perform many computations and has support for string handling such as Regex and Mathematical functions which allow developer to perform many complex functions within browser itself.

Using AJAX allows JavaScript retrieve data from a host server and manipulate the element on the web page to display the new data without refreshing the page giving the user a richer experience.

JavaScript Functions can be executed inline in the HTML code or stored in external files that can be shared among other pages, embedding JavaScript within the HTM is frowned upon as it can lead to slow loading of a webpage, and it makes testing very difficult and makes for very messy coding when developing web Pages.

External files allow for caching on the local host which can speed up a web site performance in that the web pages does not have to reload them.

There are security risks associated JavaScript, malicious code can be injected in a web site using cross site scripting XSS and these potential issues should be guarded against and using Frames works can mitigate against of these issues but at the end of the day security is an issue that a developer should consider from the offset of the application.

 JavaScript has enable the development of a large number of libraries such as some mentioned below. With Great Power come great responsibility! (Stan Lee)

****

It is an open Source Library licensed under M.I.T and is free to use. It is a light weight footprint at only 32KB when minified and qzipped, it is Cross Browser and its CSS3 compliant.  It is pack with features, very fast, concise JavaScript Library.  It allows traversal of the HTML Document, AJAX and Event Handling. It has a plugin architecture, the API is fully documented.

It is used by 500K websites and is one of the most popular Libraries among a large developer community, with a very large number of contributors from developers. It is being used by large and small organisations alike.

It is easy to learn, and the JQuery philosophy is ***Write less, do more***. It is very friendly to use and helps avoids conflicts with other JavaScript libraries.

**** Jasmine is an open source testing framework for JavaScript. It aims to run on any JavaScript-enabled platform, to not intrude on the application nor the IDE, and to have easy-to-read syntax.

Jasmine aims to be easy to read. A simple hello world test looks like the code below, where describe () describes a suite of tests and it () is an individual test specification. The name "it ()" follows the idea of behaviour-driven development and serves as the first word in the test name, which should be a complete sentence. Usage follows syntax similar to that of RSpec.

describe('Hello world', **function**() {

it ('says hello', **function**() {

expect (helloWorld()).toEqual("Hello world!");

});

});

We have implemented some Jasmine tests for email validation. This test is in place as well as a number of other test that have not been executed.

AngularJS is an open source JavaScript framework that allows you to extend HTML vocabulary for your application. AngularJS enforces the Model View Controller pattern by dividing the application into MVC sections and then Angular connects the separate sections.

The view is constructed using HTML and we can also specify which controllers to use for each HTML element or multiple elements. In other words, lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology (Docs.angularjs.org, 2014).

Another benefit of angular is that is reduces the disconnection between the static HTML and how we can modify and perform actions on this HTML via data binding, DOM control structures and grouping of HTML into reusable components.

Angular is a complete client side solution which includes everything needed to build a CRUD application as well as unit testing, end to end testing, seed application with directory layout and test scripts as a starting point (Docs.angularjs.org, 2014).

We have recognised the benefits of the AngularJS framework and have implemented it in the application. The slider is operating with AngularJS as seen below:



Figure 2.2. AngularJS Slider

CanvasJS is an easy to use HTML5 & JavaScript Charting library built on Canvas element. Graphs can render across devices including iPhone, iPad, Android, Windows Phone, Desktops, etc. This allows you to create rich dashboards that work on all devices without compromising on maintainability or functionality of your web application. Charts include several good looking themes and is over 10x faster than conventional Flash and SVG Charts – resulting in lightweight, beautiful and responsive dashboards (Urs, S. (2014))

CanvasJS offers many different chart types that we have illustrated in the application with some jQuery which allows the user to interact and choose how they would like to view the data that we have provided.

Below is how a minimal basic Column Chart would look like. Here are important things to remember

1. Instantiate a new **Chart** object by sending the ID of div element where the chart is to be rendered.
2. Pass all the Chart related **“options”** to the constructor as the second parameter.
3. Call **chart. Render()** method to render the chart

Chart **“options”** mainly contains 4 important items.

1. **Title** object with its **text** property set.
2. **dataPoints** – which is an array of all data items to be rendered
3. **dataSeries** – parent of dataPoints that also defines type of chart and other series wide options
4. **Data** – array element which is collection of one or more dataSeries objects. Here we have only one dataSeries.

Though we are going to use Column Charts to learn various concepts, it should be easy to change them into any kind of chart that you wish! Just change the type property of dataSeries to any chart type that you need – bar, area, line, scatter, stacked Colum, etc. (Urs, S. (2014))

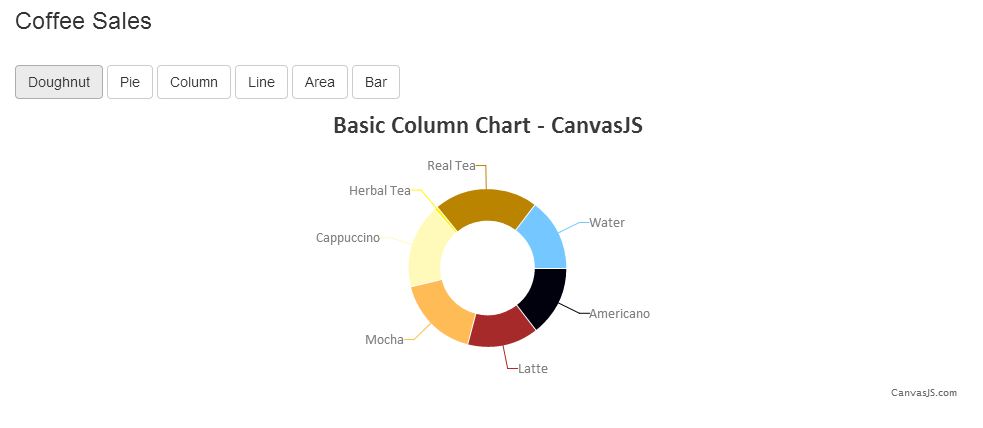
****

Figure 2.3. CanvasJS Interactivity Chart

The Google Maps JavaScript API allows users to build highly customisable maps with your own content and imagery and to use these creation to create rich applications and stunning visualisations of your data, leveraging the comprehensiveness, accuracy, and usability of Google Maps and a modern web platform that scales as you grow (Developers.google.com, (2014)).

We have found extremely efficient to implement the Google Maps API due to the ease of use and the extensive documentation that is provided on the Maps section of the Google Developers website. However, it should be noted, for non-technical users, that an understanding of JavaScript is recommended before getting started with this API.

We firstly had to obtain a key which allows us to monitor our application’s Map API usage. If your application's Maps API usage exceeds the [Usage Limits](https://developers.google.com/maps/documentation/javascript/usage#usage_limits), you must load the Maps API using an API key in order to purchase additional quota (Developers.google.com, (2014)).

All instances of a Google Map are centered on a few basic points as can be seen below:

· We declare the application as HTML5 using the <! DOCTYPE html> declaration.

· We include the Maps API JavaScript using a script tag.

· We create a div element named "map-canvas" to hold the Map.

· We create a JavaScript object literal to hold a number of map properties.

· We create a JavaScript map object, passing it the div element and the map properties.

· We use an event listener to load the map after the page has loaded.

With the aid of numerous tutorials and our team’s experience with JavaScript and Google Maps we implemented this feature to the site which will display a map that is centered on your current location and display coffee stores nearby.

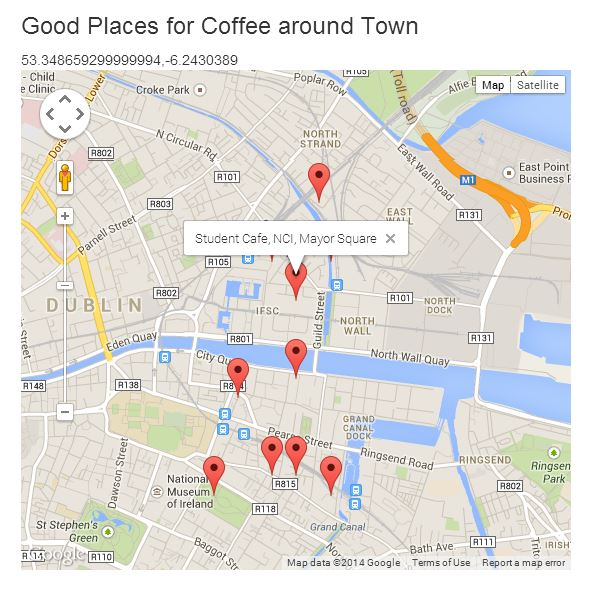


Figure 2.4. Google Map feature

The Application has a requirement for a mapping service, MapBox offers a Cloud based service for providing a Mapping service based on OpenStreetMap [] for Web and native Applications.

It has full browser support across all platforms.  They provide a very simple to use JavaScript API, the API allows for extension by using plugin to enhance it feature, reducing the size of the base library so you only load what you need. They claim within minutes you can have a map service in your website. They also have a web service which can be consumed by any language. They have tools kit for IOS and X OS also.

It allows the designer/developer design customer markers which could be used to display a customer’s logo such as a Restaurant for instance to make their locations stand out when view the map. (See Below)

It has full multi lingual support which allows easy translation for use in other languages allowing users to customise their view to their native language.

The also claim it is very easy to move from Google Maps.

It is not exactly a pay as you go but works on a monthly fee and the fee determines the number of hit per month, the developer account will give 3K hits a month with a very low feature count such as points lines and polygons.

The Premium account give 1M hits a month with 2K geographic features but at a cost of $499 per month. Google Map's is the market leader "by Streets" and allows a very large number of map hits per day and if your map requirements are for basic mapping then MapBox might not be the best selection but if your requirements are a little more complicated, MapBox would provide a very powerful service and applications such as environmental or Planning Departments to show geoFencing or importing GIS information from ESRI which is the de facto system used by most Local Authorities/Utility Companies  to map their Services such as Water, Gas Telecoms infrastructure which is hidden under ground.

We did not incorporate this feature due to time constraints but the benefits of Map Reduce would greatly enhance our application and we will introduce this to the project in the future.



Figure 2.5. Map Canvas example

We have included a Twitter timeline that again highlights the theme of our application and was quite easy to implement due to the excellent documentation available.

Embeddable timelines make it easy to syndicate any public Twitter timeline to your website with one line of code. Just like timelines on twitter.com, embeddable timelines are interactive and enable your visitors to reply, Retweet, and favourite Tweets directly from your pages. Users can expand Tweets to see Cards inline, as well as Retweet and favourite counts. An integrated Tweet box encourages users to respond or start new conversations, and the option to auto-expand media brings photos front and centre. You can embed a timeline for Tweets from an individual user, a user's favourites, Twitter lists, or any search query or hashtag (Dev.twitter.com, (2014)). Our implementation of the embedded timeline is shown below.



Figure 2.6. Map Canvas example

Twitter Bootstrap is the most popular front-end framework for developing responsive, mobile first projects on the web. Bootstrap easily and efficiently scales your project with one code base, from phones to tablets to desktops. Furthermore, with bootstrap you get extensive and beautiful documentation with hundreds of live examples, code snippets, and more (Mark Otto, (2014)).

We had deliberated over whether we should incorporate SkelJS or Twitter Bootstrap. We concluded that the application would gain more benefit from using Bootstrap because we receive more in terms of the style of the application than Bootstrap while keeping the same functionality. Our team members also have more experience in dealing with Bootstrap and there is a larger community of users of Bootstrap with more documentation available also.

We have included screenshots that demonstrate the responsiveness of our application below on different browser window sizes.

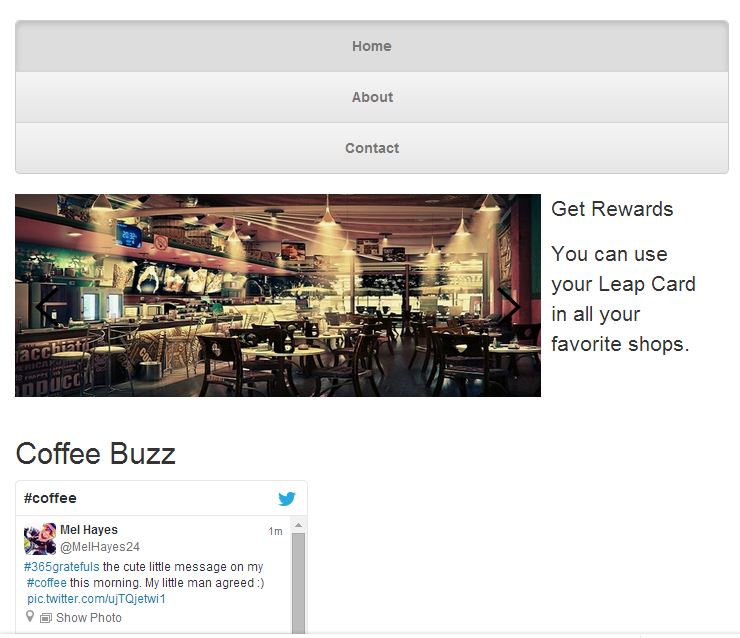


Figure 2.7. Medium browser window size example

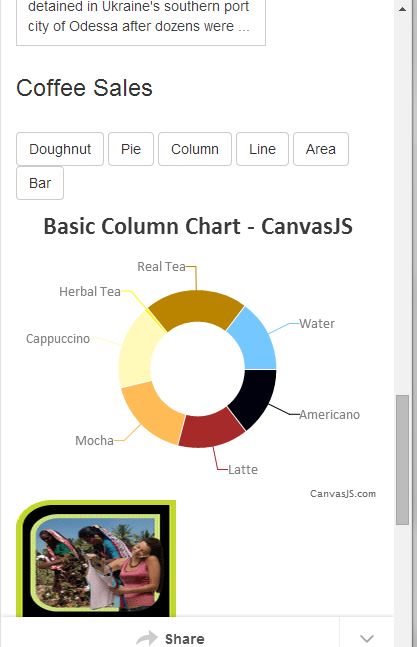


Figure 2.8. Narrow browser window size example

# User Interface Design

The User Interface design for the ARIA continuous assessment was undertaken in conjunction with the Enterprise Frameworks project. In the formative stages of the development process we completed initial sketches of a possible user interface design and we refined these sketches through the wire framing process using the Balsamiq software package. An example of the sketches that were done can be found below in Figure 3.1.

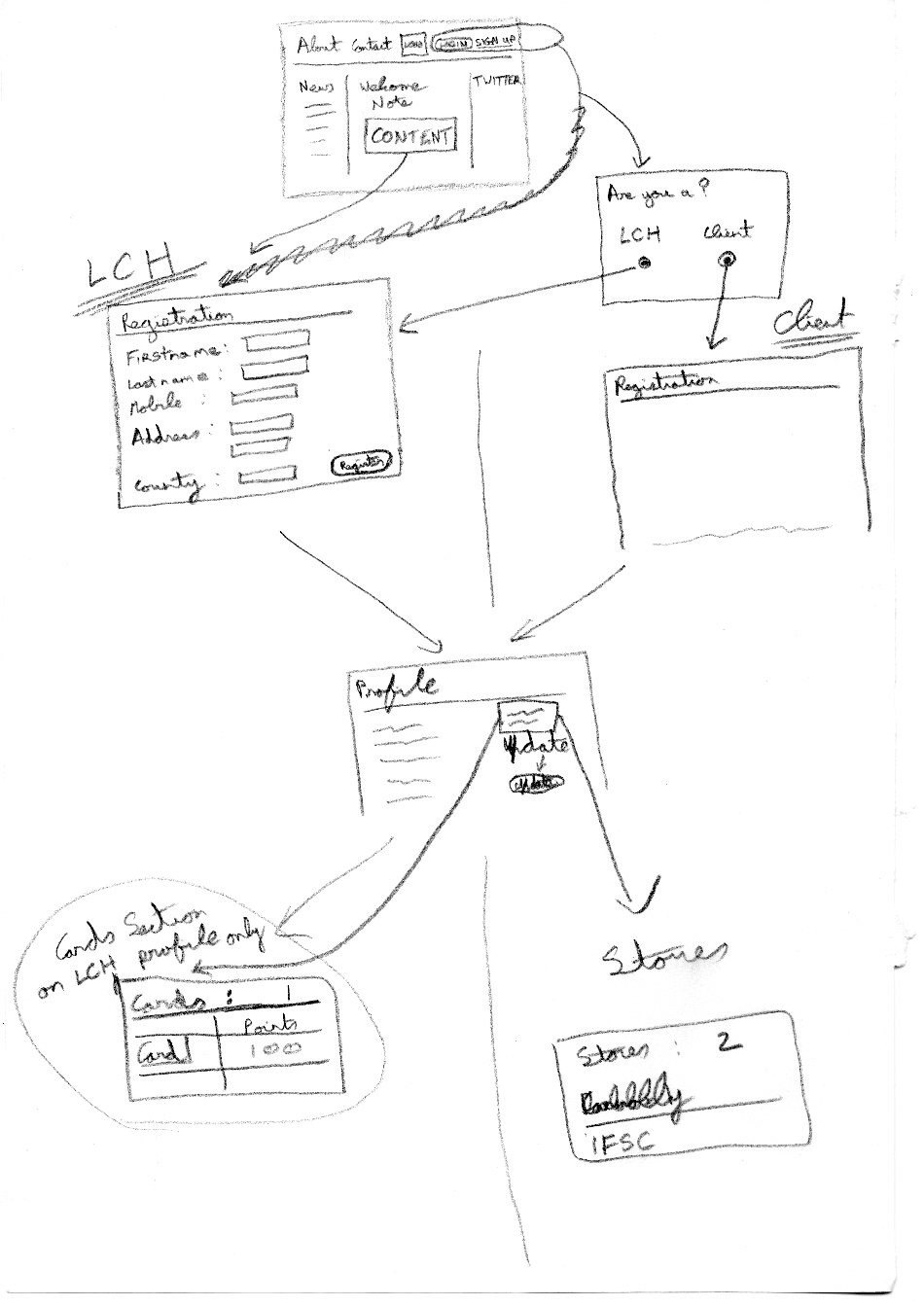


Figure 3.1 Application Sketches

As mentioned above, these sketches were further enhanced in the higher fidelity wire frames that were produced with Balsamiq and an example of the home page wire frame can be seen in figure 3.2 below.

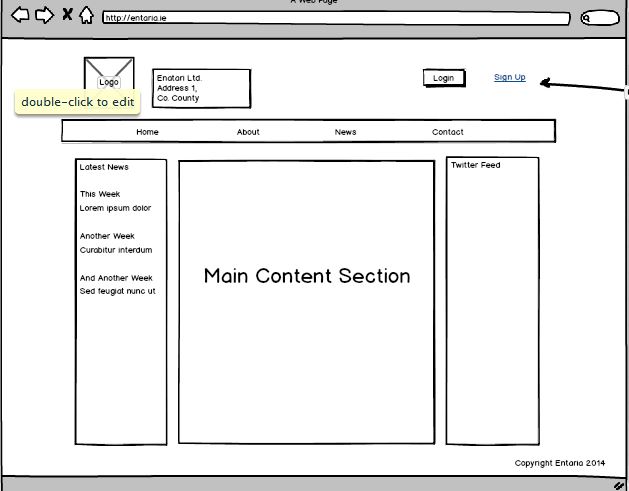


Figure 3.2 Landing Page Wire Frame

Due to the agile nature of our development process it is evident that there were some adjustments made between the first and second stages of the user interface design process. This pattern was continued as we implemented our interaction design strategy. Therefore, the result that we have deployed contains some features that were not including in the original mock ups and/or sketches. However, these have not only improved the design of the application but also show cased some of the technologies that we have encountered in tutorials and lectures. The final home page design can be seen in Figure 3.3 below.

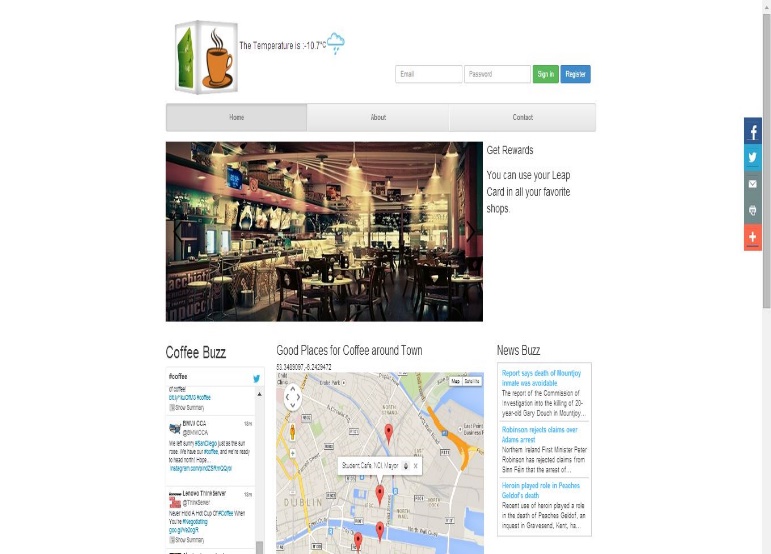
****

Figure 3.3 Landing Page – Final Design

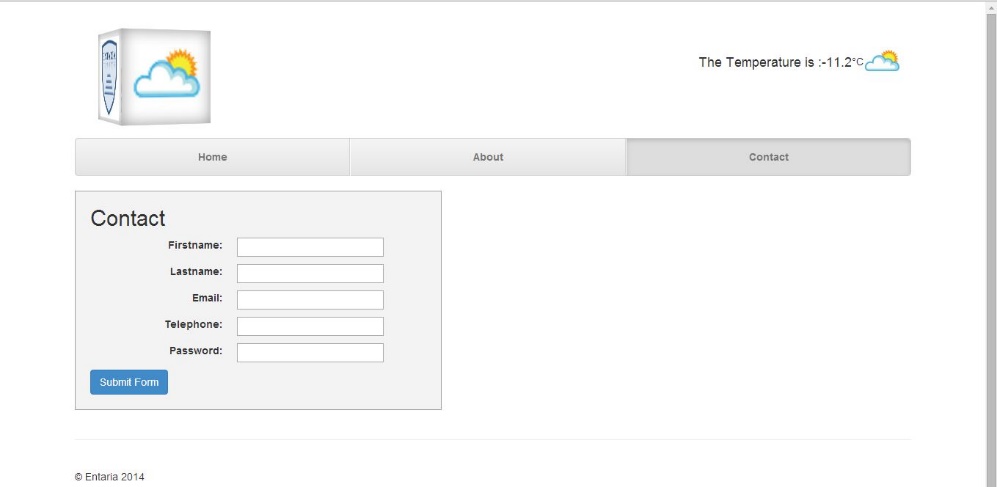


Figure 3.4 Contact Page – Final Design

The layout of the site and the responsiveness have been implemented with the Twitter Bootstrap front end framework. This framework will reduce the work load in terms of designing a layout that is interactive and provided a large amount of sleek styling.

The final layout incorporates many features of a modern website in a style and format that is familiar to users of web applications yet one that we anticipate will be suitable in the year 2022. As is evident in the image of the homepage above, the basic layout shows a header with navigation, a slider and basic three column layout.

The theme of the site and the nature of the application that we are providing was heavily focused on and is evident throughout the style of the application. The theme of coffee is also noticeable in the news, twitter feed, weather, google map coffee shop locations and data display on the homepage.

## User Interaction Design Strategies

As previously stated, we have focused on creating an interactive design that is available to as many users as possible on as many devices as possible. A major feature of the application is that it is a SPA or Single Page Application which was implemented using JavaScript. This improves load time and increases the enjoyment of using the application.

The Google Map is situated purposefully in the centre of the application to gain the users attention and to drive traffic to stores which is the goal of the system for both of our participants. In relation to this we have focused on the overall flow of the site. The lack of a page refresh, and the columnar layout of the site does not disrupt the user flow. The page refresh creates an artificial break in the action – or a break in the user’s flow (Scott, B & Niel, T, 2009).

We encourage users to share our site as much as possible with social media, email and print icons that are located in a menu that is always evident on the screen to encourage use.

## User Testing

The user testing focused on a think aloud process. I have documented the scenarios below that our users had to undertake.

**Scenario 1**

Navigate to the Coffee Sales section of the website. View the data that is displayed on the doughnut chart that is displayed. Identify the buttons above the chart and click on one of your choice to display the same information in a different style of graph.

**Scenario 2**

Navigate to the Twitter section of the website. View the Tweets that are displayed and navigate with through the list of coffee related tweets.

**Scenario 3**

Navigate to the header section of the website and identify the menu navigation. Click on the “Contact” tab. When you are taken to the “Contact” page, identify the Contact form. Enter your details into this form and hit submit.

Each user who completed the Think Aloud testing was recorded using the Cam Studio software.

**Satisfaction**

Each user was also asked each participant in the study to complete as systems usability test. The test is documented below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
| 1. I found the system unnecessarily complex. |  |  |  |  |  |
| 1. I thought the system was easy to use. |  |  |  |  |  |
| 1. I think that I would need the support of a technical person to be able to use this system. |  |  |  |  |  |
| 1. I found the various functions in this system were well integrated. |  |  |  |  |  |
| 1. I thought there was too much inconsistency in this system. |  |  |  |  |  |
| 1. I think that I would like to use this system frequently. |  |  |  |  |  |
| 1. I would imagine that most people would learn to use this system very quickly. |  |  |  |  |  |
| 1. I found the system very cumbersome to use. |  |  |  |  |  |
| 1. I felt very confident using the system. |  |  |  |  |  |
| 1. I needed to learn a lot of things before I could get going with this system. |  |  |  |  |  |

# Architecture

## Application Architecture

We have deployed our website to Azure Web Sites. Azure Web Sites is a fully managed Platform-as-a-Service offering that enables you to deploy and scale Web Apps in seconds. Focus on your application code, and let Azure take care of the infrastructure to scale and securely run it for you (Azure.microsoft.com, (2014)).

## Security

As this site is being designed from the ground up, it was intended to implement Best Practices when it came to security right from the beginning.   Best Practices must be employed using the Security recommendations from such organisations as OWASP [19] and W3C to make sure that the site is secure from any number of attacks which may cause the site to be compromised.  We should have implemented counter measures to known site attacks. Security is the job of the developer and the administration to ensure that Best Practices are adhered to and that security is a fundamental part of the Web Site Development.

## Version Control

Our group used the Github source control environment for the project. This was used regularly throughout the development of the project and was a major aspect used in the weekly progress reports and planning process of the project. GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both paid plans for private repositories, and free accounts for open source projects.

## Toolkits and Frameworks:

Knockout.js was created by Steve Sanderson in 2010 based on an MVVM framework that provides data-bindings. It is used to keep the UI code separated from the data model. We can then implement custom behaviour and, as it is backed by Microsoft it is compatible with most browsers including IE6. It has less complexity than either Ember or Angular and is quick for a developer to learn and begin coding.

We have used this framework on the SPA feature of this application. This greatly enhanced the user experience of the site and was one of many possible methods of how we could have created the SPA. KnockoutJS was our choice because it was suitable for data binding used for implementing SPA functionality.

We also considered using the following framework and have documented a summary on this and explained why we did not incorporate this into the project.

Ember.js is a recently released (2011) modular framework that follows the principle of 'convention over configuration'. It has strong templating capabilities based on Handlebars.js the designers of ember.js see it “not [as] a framework for building traditional websites [but to] build desktop-like experiences”. It is a larger download than the other frameworks at 69kB but contains many features already built in.

We did not use this framework because its major drawback is that as it is still so new it has not yet reached maturity as a framework (e.g. data handling has yet to be incorporated) yet it is being used by companies Yahoo!, Groupon and Square (Emberjs.com, 2014).

## Data Transfer Strategies:

Our data transfer strategy is to use JSON (JavaScript Object Notation)., throughout  the application, we are consuming a number of API’s using Angular.JS HTTP and JQuery. Get.  All data retrieval willbe performed using AJAX and JSON.  JSON is a lightweight data-interchange format. It is human readable, it is completely language independent. Unlike xml, JSON can support Objects, arrays, strings and numbers.  We are also sending back data to the server using JSON which is then inserted into the CardHolderDetails table on the Azure Website.

JSON allows data to be sent straight into the application with little or no parsing which reduced the load on the Host Processor this is what make it ideal for Mobile devices with limited horsepower.

JSON is much more suitable for use with JavaScript over XML as it can support data types and arrays objects.

We have implemented local storage as the data source for the CanvasJS element of the application.

## Evaluation and Testing:

We have organised our peers from other groups in the module to sit down and evaluate the rich internet application. These test users were more than willing to not only see what application we had created but systematically analyse the project and provide feedback via the methods above. The response from our test users was largely positive.

We used the Jasmine framework for testing various JavaScript methods as part of a TDD (Test Driven Development) process that was chosen for this application’s development.  We found that, while Jasmine is excellent for testing standalone JavaScript methods, it is difficult to apply to http testing and UI testing.  For debugging we used Firebug in Mozilla for html and JavaScript debugging and we used Fiddler2 to examine http verbs testing web API calls.

# Summary

This has been an excellent project in the sense that we have enjoyed applying what we have learned to the project. We have learned about responsiveness, the state of the art web triad HTML 5, CSS 3 and JavaScript. We are conscious of the languages, toolkits and frameworks that are in demand in the real world and we tried to apply as many of these as possible.

We recognise that the future of rich internet applications will focus heavily on mobile and single page applications and hence our application focused on these. Giving the time scale, personnel and time constraints we were unable to implement some of the features that would be expected in an ARIA project such as security and limited testing.

This is also the first time that we have worked as part of team using source control for both the Advanced Rich Internet Applications and Enterprise Frameworks modules.

# References

W3C (2014). *HTML5* [online] Available at http://www.w3.org/TR/html5/ [accessed 2nd May 2014]

W3Schools (2014). *HTML5 Introduction* [online] Available at http://www.w3schools.com/html/html5\_intro.asp

Docs.angularjs.org, (2014). *AngularJS*. [Online] Available at: https://docs.angularjs.org/guide/introduction [Accessed 2 May. 2014].

Urs, S. (2014). *Beautiful HTML5 & JavaScript Charts | CanvasJS*. [Online] CanvasJS. Available at: http://canvasjs.com/ [Accessed 2 May. 2014].

Developers.google.com, (2014). *Google Maps Web APIs — Google Developers*. [Online] Available at: https://developers.google.com/maps/web/ [Accessed 3 May. 2014].

Dev.twitter.com, (2014). *Embedded Timelines | Twitter Developers*. [Online] Available at: https://dev.twitter.com/docs/embedded-timelines [Accessed 3 May. 2014].

Azure.microsoft.com, (2014). *Web*. [Online] Available at: http://azure.microsoft.com/en-us/solutions/web/ [Accessed 3 May. 2014].

Mark Otto, a. (2014). *Bootstrap*. [Online] Getbootstrap.com. Available at: http://getbootstrap.com/ [Accessed 3 May. 2014].

Wikipedia, (2014). *GitHub*. [Online] Available at: http://en.wikipedia.org/wiki/GitHub [Accessed 3 May. 2014].

Owasp.org, (2014). *OWASP*. [Online] Available at: https://www.owasp.org/index.php/Main\_Page [Accessed 3 May. 2014].