|  |  |
| --- | --- |
|  |  |
|  |  |



**GROUP PROJECT**

**MSc in Web Technologies**

**National College of Ireland**

**ARIA Application for a Gymnasium or Chain of Gymnasiums**

Module: Advanced Rich Internet Applications

**Report**

**Technology:**  HTML5, CSS5 and Javascript frameworks: SkelJS, AngularJS, Jasmine Testing Framework

**Group:**

Thomas Dowling x12113441

John Fitzpatrick x00113647

Kean Wei Ho x12104574

Ewa Marciniak x11114533

**Supervisor:**  Adrian Skehill

Contents

[1. Introduction 3](#_Toc385846417)

[2. Motivation [5%] 3](#_Toc385846418)

[2.1. Poject Scope 3](#_Toc385846419)

[2.2. Area of contribution 3](#_Toc385846420)

[3. State of the Art Review [30%] 3](#_Toc385846421)

[4. User Interface Design [20%] 3](#_Toc385846422)

[5. Architecture [25%] 3](#_Toc385846423)

[5.1. Application Architecture 3](#_Toc385846424)

[5.2. Security 4](#_Toc385846425)

[5.3. Toolkits and Framework 4](#_Toc385846426)

[5.4. Data Transfer Strategies: 4](#_Toc385846427)

[5.5. Evaluation and Testing: 4](#_Toc385846428)

[6. Summary [10%] 4](#_Toc385846429)

[7. References [10%] 4](#_Toc385846430)

# 1. Introduction

# 2. Motivation [5%]

## 2.1. Poject Scope

This is a short description of why your innovation is useful and what it might achieve.

## 2.2. Area of contribution

This should describe the general problem area. For example, what is a problem in educational environments and how might they be better developed through the application of a RIA.

# 3. State of the Art Review

## 3.1 Technologies used for implementations of RIA

On the technology are all these frameworks used to build and deploy rich internet applications (RIA) in this project:

1. AngularJS ( [Fig. 1](#Fig1) )
2. SkelJS ( [Fig. 2](#Fig2) )
3. CanvasJS [( Fig. 3](#Fig3) )
4. Google Calendar API [( Fig. 4](#Fig4) )
5. Google Map API ( [Fig. 5](#Fig5))

(Rowell, 2010)

## 3.2 Overview of state of the art “Adopted”

Below are all the frameworks with the relevant technologies used to develop and deploy our project:

### 3.2.1 What is AngularJS ?



**Figure 1: AngularJS**

* Definition: AngularJS is a JavaScript MVC (Model View Controller) framework that is created by Google to build good Web application. Web app build by AngularJS are well structured, testable, and maintainable front-end applications.
* Distinctive Features:

(a) In layman terms: AngularJS is Framework, HTML Compiler, and JS library.

1. In Angular, you directly design your app (architecture) your view, unlike JQuery ( Design then manipulate DOM ).
2. Usually, JQuery could be integrated with Angular JS fairly easy, Thus augmentation is possible like long lines of codes for JQuery plugins could be rewritten with smaller number of codes.
3. With AngularJS, you tends to “think –architecture”. For instance: Single-page applications are applications, not webpages; therefore We must emphasize on both the front and server side as a Developer. The distinctive characteristics are :
4. Divide the app into individual , extensible, and testable parts.
5. Data Binding – A set of functions associated with a scope or in simple terms is a process of connectiona between app UI and Business Logic. One good example: Automatically update view.
6. Distinct Model Layer—completely separate from view. Maintains separation of concerns and good for testing.
7. Separation of concerns—Views (Shows what happen); Model represents data; A service layer that perform reusable tasks; Do DOM manipulation and augment the view with directives; lastly glue everything together with controllers.
8. Dependency injection—providing objects that an object needs (its dependancies) instead constructing them itself,thus a very useful technique for testing.
9. Test Driven Development—Both iterative and test-driven.

* Advantages: (a) Good tool for testing and debugging—Jasmine works on top of AngularJS. Furthermore, Angular which utilizes dependency injection that is used for unit testing .

(a) Angular was an internally developed by Google. Now it is open sourced but have continued development support by Google.

(b) Ease of Learning—many materials available in the internet.

(c) Compatibility and Extensibility – Angular is compatible with other

Frameworks.

1. Uses primitive JavaScript.
2. Loaded with functionality.
3. Easy setup.
4. Make browsers to operate the same way as angular.

* Disadvantages: (a) The learning curve is longer for Angular to various common libraries: Knockout and Backbone.

(Bégaudeau, 2012)

(Tim Cunningham, 2011)

(angularjs.org, 2014)

(Trager & Kagan, 2013)

### 3.2.2 What is SkelJS ?

******

**Figure 2: SkelJS**

* Definition: “SkelJS is a lightweight frontend framework for building responsive sites and apps. It consists of only a single JS file (weighing only 20Kb )
* Distinctive Features: There are four powerful components.

1. CSS 12 column grid system = “concise, uncluttered syntax, adjustable gutters, unlimited nesting support and many more cool features”.
2. Lots of responsive breakpoints = “A streamlined replacement for CSS media queries designed to handle all of your responsive needs”.
3. Lots of CSS shortcuts =”handy shortcuts with all common CSS tasks, like normalizing browser CSS (via normalize.css)” .
4. Extendable via plugins=”a platform for extending skelJS without compromising its lightweight footprint ; eg: Panels plugin and overlays plugin.

* Analysis For skel.JS:

1. Skel.JS uses a JQuery library (jquery.min.js) which is a fast, small, and feature-rich JavaScript library.
2. It is MIT licensed.

(skels.org, 2014); (thewebsites.info, 2014)

### 

### 3.2.3 What is CanvasJS?



**Figure 3: CanvasJS**

* Definition: The canvas.js module is a simple and robust JavaScript charting libraries for the HTML5 <canvas> element, which can be used to generate interactive 2D graphics in a web browser, using lines , shapes, paths, images and text.
* Distinctive Features:

1. CanvasJS Charts have simple and cool JavaScript API
2. CanvasJS preload with several fantastic looking themes that you have options by setting a single parameter.
3. CanvasJS supports 18 different types of HTML5 Graphs.
4. CanvasJS can render 100,000 Data-Points in just around 100 milliseconds.( high performance HTML5 and JS Charts ).

* Advantages:

1. Very easy to use.
2. Can work cross-platform.
3. Provide good performance.
4. Have lots of support like Documentation, and directly from developers.
5. CanvasJS comes with beautiful themes and is around 10x faster than conventional Flash and SVG charts – resulting in light and responsive dashboards.
6. For non-commercial usage like non-profits organization or students ; it is free.

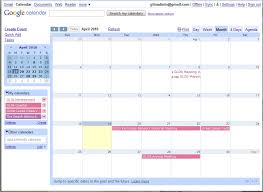
* Disadvantages:

1. It is not free when used for commercial purposes

(canvasjs.com, 2014)

(Panda, 2013)

### 3.2.4 What is Google Calendar API?



**Figure 4: Google Calendar API**

* Definition:

The Google Calendar API allows a program to perform many of the operations available via Google Calendar web interface. In the simplest term API ( Application Programming Interfaces ) are sets of requirements that govern how one application can talk to another. In layman terms, APIs are crucial for developers can create new apps that “use or share” web services like Google Maps or Dropbox.

* Distinctive Features:

1. A calendar that have single event that have information like title of event, start and end times, and attendees.
2. A lists of calendars in the Calendar UI
3. A calendar that have options like user preference from the Calendar UI, such as user’s time zone.
4. A list of colors to choose for the events and the calendars.
5. A feature that show that you are Free/Busy in the set of time or set of calendars.

* Advantages:

1. Saves a lot of time and money. APIs are free make it easy to use huge infrastructure instantly for app.
2. Potential for innovative ideas. For example can integrate Google Calendar with minimum costs.

* Disadvantages:

1. Dependence on external services. If the service or the company shuts down it would cause the app’s usability at risk.
2. Too many APIs and platforms could be confusing because every API providers do not adhere to standards.

(developers.google.com, 2014); (googledevelopers.blogspot.ie, 2014); (Gagem, 2013)

### 3.2.4 What is Google Map API?



**Figure 5: Google Map API**

* Definition:

Google map API is a specification used by software components to communicate with each other. The API also describe how a particular task is performed as well.

* Distinctive Features:

1. Free API key from Google.
2. Create a Basic Google Map is an easy task.
3. Maps Overlays has options where you can add a marker in the Google Map.
4. Add Maps Events so that users can click the marker to zoom and attached event handlers to Google Map.
5. A Google Maps Controls that with the default set like Zoom, Pan, MapType, and Street View.
6. There are 4 types of Google Maps : Roadmap, Satellite, Hybrid, and Terrain.

* Advantages:

1. Most APIs are free or very cheap.
2. Leveraging the resources of the web could same a lot of time in terms of development Time.

* Disadvantages:

1. The reliability of the system is depends on the provider, since it is free, the provider do not have the responsibility to keep your app operational.

(Stone, 2010); (developers.google.com, 2014)

# 4. User Interface Design [20%]

The main goal when deigning the User Interface (UI) for the project was to make the user interaction simple, intuitive and efficient. The following section describes the process employed by the team to arrive at the design of the UI.

The formulation of the UI design was influenced by a number of factors. At the start of the project the group researched and discussed the requirements of the potential users. Due to the tight time constraints and limited budget, the project requirements were finalized based on the research, rather than a full scale functionality requirement gathering techniques such as interview or questionnaires. It was agreed that this application must be simple and functional and must deliver specific gym information to its members outside of the gym. This list included a list of gym locations nearby, class schedules, workout statistics, etc. Users tend to search for the nearby gym location when traveling, check the class schedule on the way back home, and check their own workout statistics when on the bus. This sort of content would normally be accessed by the users “on the go” and at various locations outside the gym, not only on PCs but also tablets and mobiles.

In the era of growing popularity of smart mobile devices and easy access to the Internet, the importance of a mobile-friendly design cannot be understated. Creating a number of separate applications specific to each type of device was never an option as it would not be sustainable in the long term to maintain such a large number of applications. Also due to the tight time constraints this was not possible. The chosen solution was to implement a responsive design approach that would provide a unified user experience throughout the site across multiple devices, and for a number of different screen sizes.

A responsive design can be implemented quite efficiently using one of the many JavaScript frameworks such as SkelJS or Foundation. Due to the level of skill and experience recently acquired by the members during this course module, we decided to use SkelJS.

*“SkelJS is a lightweight frontend framework (…) consisting of only a single JS file - weighing in at just 20kb as of this version.”* (Skeljs.org, 2014).

It is characterized by a sophisticated 12-column CSS [grid](http://www.jqueryscript.net/tags.php?/grid/) system which allows proportional scaling of UI elements and includes a breakpoint handler which intelligently replaces the CSS queries depending on the type of media the app is displayed on. SkelJS was very easy to implement into the app.

Another factor that influenced the UI design was the growing user-experience expectations.“*People are beginning to demand a higher standard of user experience (UX) quality from the software systems they use in their business and personal lives.”* (Anderson, McRee and Wilson, 2010). Single page applications (SPA) with desktop like behavior implemented into web based applications are rapidly growing in popularity and slowly becoming the new standard. Users no longer are required to wait for the page to reload and instead are presented with the data almost instantaneously.

In order to oblige with the current expectations the project was designed and implemented as a SPA configuration using AngularJS. AngularJS is a powerful JavaScript framework that applies the MVC structure to the project’s front-end business logic and provides a number of comprehensive features such as dependency injection, custom HTML templates, and two way binding. The implementation of this framework did not shape as much the form of the interface, as it improved the user experience. When selecting menu tabs, users are seamlessly taken from one page to another without the page load waiting time.

Based on the established requirements the team concentrated on forming the information architecture. Due to the relatively simple nature of the app, it has been decided that the navigational structure will be flat and will consist of a 5 tabs, each corresponding to a separate section. Due to the personal and confidential aspect of the workout statistics displayed on one of the pages the team decided to implement a simple authorization system. Unfortunately due to the problematic nature of back end implementation and limited time constraints this was later abandoned. The login Controls initially added to the interface were consequently removed.

One of the goals of the user interface design was to implement both usability and accessibility principles into the application. “*Usability is about designing products to be effective, efficient, and satisfying”* and its key aspect is creating positive user experience. “*The goal of web accessibility is to make the Web work well for people, specifically people with disabilities”* (W3.org, 2014). Combining both usability and accessibility meant creating both satisfying and disabled-people-friendly applications. SPA implementation was a definite step towards the improved user experience and increased usability. Additionally the graphic design was supposed to serve the purpose of enhancing both usability and accessibility.

The simplicity of the structure and design was meant to enhance both the accessibility and usability of the application. Additionally the design incorporates the following features:

### Provision of text alternatives for any non-text content

### Minimal amount of color to prevent difficulties for people with color blindness

### Use of relatively large font and use of highly contrasting colors (black and white) on the text and background to help people with poor eyesight

### Images contain the alt values (Heng, 2014)

* Consistency in the design

The team aspired to design the User interface to the best principle and standards, as well as to provide a high aesthetic appeal of the design at the same time balancing the visual and functional aspects of the system. Although the design could be further improved we feel that we have achieved some success in fulfilling these aspirations.

# 5. Architecture [25%]

## 5.1. Application Architecture

Outline the solution architecture for your application. You sould justify the approach you took, and how it supports the project scope. This means mapping back to the state of the art review when describing the features and functionality of your system.

## 5.2. Security

Evaluate industry standard error handling, and outline how you integrated these approaches within your application.

## 5.3. Toolkits and Framework

Discuss the toolkits and frameworks used and the justification for using them.

## 5.4. Data Transfer Strategies:

This section looks at the access and transporting of data to be consumed/created by the RIA.

## 5.5. Evaluation and Testing:

You must include a short description of how your application was evaluated for its audience.

Critically analyse the testing methodology employed, as well as any debugging techniques you used in building the application.

# 6. Summary [10%]

This is a short section that includes a brief summary of what was achieved so far. Evaluate the approach you took, the tools you used, and the implementation of your applications. You should describe what changes you would make or future work that would benefit your application.

# 7. References

Anderson, J., McRee, J. and Wilson, R. (2010). *Effective UI*. 1st ed. Beijing: O'Reilly.

angularjs.org, 2014. *What Is Angular?.* [Online] Available at: https://docs.angularjs.org/guide/introduction [Accessed 30 Apr 2014].

Bégaudeau, S., 2012. *Everything you need to understand to start with AngularJS.* Obeo, France: Stéphane Bégaudeau. canvasjs.com, 2014. *CanvasJS.* [Online]   
Available at: http://canvasjs.com/[Accessed 1 May 2014].

developers.google.com, 2014. *Google Maps API.* [Online] Available at: https://developers.google.com/maps/ [Accessed 2 May 2014].

developers.google.com, 2014. *Google Calendar API.* [Online] Available at: https://developers.google.com/google-apps/calendar [Accessed 2 May 2014].

Gagem, S. v., 2013. *Developing applications with APIs – pros and cons.* [Online]   
Available at: https://www.developergarden.com/en/blog/articles/article/developing-applications-with-apis-pros-and-cons/ [Accessed 2 May 2014].

Heng, C. (2014). *How to Improve the Accessibility of Your Website (thesitewizard.com)*. [online] Thesitewizard.com. Available at: http://www.thesitewizard.com/webdesign/improve-accessibility.shtml [Accessed 30 Apr. 2014].

googledevelopers.blogspot.ie, 2014. *Improving the Google Calendar API.* [Online]   
Available at: http://googledevelopers.blogspot.ie/2014/03/improving-google-calendar-api.html [Accessed 2 May 2014].

Panda, P., 2013. *Introducing CanvasJS – Beautiful, Platform-Independent & Easy-To-Make Charts and Graphs.* [Online] Available at: http://thetechpanda.com/2013/03/04/introducing-canvasjs-beautiful-platform-independent-easy-to-make-charts-and-graphs/#.U2JTT\_ldVqU [Accessed 1 May 2014].

Rowell, E., 2010. *Building Rich Internet Applications with Frameworks.* [Online]   
Available at: http://sixrevisions.com/web-development/building-rich-internet-applications-with-frameworks/ [Accessed 29 April 2014].

Skeljs.org, (2014). *skelJS : Documentation*. [online] Available at: http://skeljs.org/docs [Accessed 28 Apr. 2014]

Stone, A., 2010. *Pros and Cons of Outsourcing Your Apps to APIs.* [Online]   
Available at: http://blog.programmableweb.com/2010/10/28/pros-and-cons-of-outsourcing-your-apps-to-apis/[Accessed 2 May 2014].

thewebsites.info, 2014. *ANALYSIS SUMMARY OF SKELJS.ORG.* [Online]   
Available at: http://thewebsites.info/website-analysis/skeljs.org#general  
[Accessed 1 May 2014].

Tim Cunningham, 2011. *AngularJS (What HTML would have been if it were written for Web Apps).* Central Georgia, USA, cfmumbojumbo.Trager, B. & Kagan, R., 2013. *The Battle of Modern Javascript Frameworks – Part IV: Angular.js.* [Online]   
Available at: http://www.softfinity.com/blog/battle-of-modern-javascript-frameworks-part-iv-angularjs/  
[Accessed 1 May 2014].

W3.org, (2014). *Web Accessibility and Usability Working Together*. [online] Available at: http://www.w3.org/WAI/intro/usable [Accessed 30 Apr. 2014].