##### Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of **Honours B.Sc. in Computer Science** in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

##### Acknowledgment



Michael or David

##### Abstract

The goal of the project was to create an interactive and enjoyable fantasy hurling style application. It was to be for Hurling, but would be flexible in that the backend could easily be used for a different sport if and when it was needed. This made the project very modular. It was designed to be very intuitive and easy to use.

Some of the key features were the ability for the user to be able to create and maintain a fantasy hurling team, receive hurling new dynamically and interact with other users via an instant messaging style forum.

The result was an application that was very simplistic in its front end design to allow novice users and experienced users alike interact with all aspects of the application with ease. Administration of the application was simple using the administrator login to update and maintain all tables in the system. The application was very modular and could be expanded or used for a different sport very easily. It also tapped into the current phenomena of social media in its instant messaging style chat.

##### Table of figures

[Figure 1 Prototyping SDLC 5](#_Toc417771231)

[Figure 2 Gant Chart 9](#_Toc417771232)

[Figure 3 IBM Analytics Benchmark 12](#_Toc417771233)

[Figure 4 Desktop Browsers 12](#_Toc417771234)

[Figure 5 Moblle Browsers 13](#_Toc417771235)

[Figure 6 Console Browsers 13](#_Toc417771236)

[Figure 7 Consumption 20](#_Toc417771237)

[Figure 8 Website Traffic 23](#_Toc417771238)

[Figure 9 Web Application general design 31](#_Toc417771239)

[Figure 10 Fantasy Hurling Web Application in detail 33](#_Toc417771240)

[Figure 11 Use Case: User checks current score 34](#_Toc417771241)

[Figure 12 Use Case: User checks fixtures 35](#_Toc417771242)

[Figure 13 Use Case: User uses social media aspect 36](#_Toc417771243)

[Figure 14 Use Case: User creates team 37](#_Toc417771244)

[Figure 15 Use Case: User makes transfer 38](#_Toc417771245)

[Figure 16 Sequence: User checks score 39](#_Toc417771246)

[Figure 17 Sequence: User checks fixtures 39](#_Toc417771247)

[Figure 18 Sequence: User posts message 40](#_Toc417771248)

[Figure 19 Sequence: User chooses team 40](#_Toc417771249)

[Figure 20 Sequence: User makes a transfer 41](#_Toc417771250)

[Figure 21 User Interface: Login Page 42](#_Toc417771251)

[Figure 22 User Interface: Registration Page 43](#_Toc417771252)

[Figure 23 User Interface: Team statistics page 44](#_Toc417771253)

[Figure 24 User Interface: Player transfer page 45](#_Toc417771254)

[Figure 25 User Interface: Standings Page 46](#_Toc417771255)

[Figure 26 User Interface: Social Media area 1 47](#_Toc417771256)

[Figure 27 User Interface: Social Media area 2 47](#_Toc417771257)

[Figure 28 Fantasy Hurling database initial prototype 48](#_Toc417771258)

[Figure 29 URLs to resources 52](#_Toc417771259)

[Figure 30 JQuery Ajax Request Method 53](#_Toc417771260)

[Figure 31 HulingApi SQL server Azure 54](#_Toc417771261)

[Figure 32 HurlingApi Web server Azure 54](#_Toc417771262)

[Figure 33 CORS enabled for api/teams route 55](#_Toc417771263)

[Figure 34 Fantasy Hurling Database Schema 56](#_Toc417771264)

[Figure 35 Admin Page for Teams 57](#_Toc417771265)

[Figure 36 ConflictActionResult 59](#_Toc417771266)

[Figure 37 Conflict Result in code and action 59](#_Toc417771267)

[Figure 38 asynchronous method example 60](#_Toc417771268)

[Figure 39 HurlingApi OData support 60](#_Toc417771269)

[Figure 40 fantasyhurling Web Server dashboard Azure 61](#_Toc417771270)

[Figure 41 Test API client in action 62](#_Toc417771271)

[Figure 42 Login Dialog 64](#_Toc417771272)

[Figure 43 get user by username request 65](#_Toc417771273)

[Figure 44 register Form 65](#_Toc417771274)

[Figure 45 add a user request 66](#_Toc417771275)

[Figure 46 insert a new team request 67](#_Toc417771276)

[Figure 47 Player details 68](#_Toc417771277)

[Figure 48 Team details 68](#_Toc417771278)

[Figure 49 Team by user id request 69](#_Toc417771279)

[Figure 50 Team Table implementation 69](#_Toc417771280)

[Figure 51 User Standings 70](#_Toc417771281)

[Figure 52 Users sorted by overall points request 70](#_Toc417771282)

[Figure 53 Update user request 71](#_Toc417771283)

[Figure 54 Update User Dialog 71](#_Toc417771284)

[Figure 55 Chat Board 72](#_Toc417771285)

[Figure 56 Messages sorted by creation time request 72](#_Toc417771286)

[Figure 57 making a transfer 73](#_Toc417771287)

[Figure 58 remove a player from a team request 74](#_Toc417771288)

[Figure 59 Adding Player 74](#_Toc417771289)

[Figure 60 add a player to a team request 75](#_Toc417771290)

[Figure 61 insert a new team request 76](#_Toc417771291)

[Figure 62 session storage 2 76](#_Toc417771292)

****TABLE OF CONTENTS****

[Chapter 1: Introduction 1](#_Toc417771353)

[1.1 Introduction 1](#_Toc417771354)

[1.2 Aim and Objectives 2](#_Toc417771355)

[1.3 Main Research Questions 3](#_Toc417771356)

[1.4 Justification and Benefits 3](#_Toc417771357)

[1.5 Feasibility (Assumptions, Risk, Benefits) 4](#_Toc417771358)

[1.6 Proposed Methodologies 4](#_Toc417771359)

[1.7 Expected Results 5](#_Toc417771360)

[1.8 Work Breakdown Structure 7](#_Toc417771361)

[1.9 Gant chart 9](#_Toc417771362)

[1.10 Conclusion 9](#_Toc417771363)

[Chapter 2: Literature Review 11](#_Toc417771364)

[2.1 Introduction 11](#_Toc417771365)

[2.2 Modern Web 11](#_Toc417771366)

[2.3 Improving Website Design 15](#_Toc417771367)

[2.4 The effect of fantasy sport participation through social media. 18](#_Toc417771368)

[2.5 The effects of fantasy football participation on NFL (National Football League) consumption: A qualitative analysis. 19](#_Toc417771369)

[2.6 Using Social Media to Build Community 22](#_Toc417771370)

[2.7 HTTP Protocol Overview 28](#_Toc417771371)

[2.8 Conclusion 30](#_Toc417771372)

[Chapter 3: Analysis and Design 31](#_Toc417771373)

[3.1 Introduction 31](#_Toc417771374)

[3.2 Proposed Methodology 32](#_Toc417771375)

[3.3 Assignment of Individual Tasks 33](#_Toc417771376)

[3.4 Front-End 34](#_Toc417771377)

[3.5 Use Cases 34](#_Toc417771378)

[3.6 Sequence diagrams 39](#_Toc417771379)

[3.7 User Interface Design (Wireframes) 42](#_Toc417771380)

[3.8 Data Store 48](#_Toc417771381)

[3.9 Back End 48](#_Toc417771382)

[3.10 Versioning Control System: GIT and Github 49](#_Toc417771383)

[3.11 Conclusion 50](#_Toc417771384)

[Chapter 4: Implementation 51](#_Toc417771385)

[4.1 Introduction 51](#_Toc417771386)

[4.2 Methodology 51](#_Toc417771387)

[4.3 Prototype Version 1 (November 2014) 52](#_Toc417771388)

[4.4 Prototype Version 2 (December 2014) 54](#_Toc417771389)

[4.5 Prototype Version 3 (December 2014) 56](#_Toc417771390)

[4.6 Prototype Version 4 (January 2015) 59](#_Toc417771391)

[4.7 Prototype Version 5 (January 2015) 62](#_Toc417771392)

[4.8 Front end (Fantasy Hurling Web Page) 64](#_Toc417771393)

[76](#_Toc417771394)

[4.9 Conclusion 77](#_Toc417771395)

[Chapter 5: Testing and Evaluation 78](#_Toc417771396)

[5.1 Introduction david 78](#_Toc417771397)

[5.2 User Interface Tests michael 78](#_Toc417771398)

[5.3 Unit Tests ) Martin 78](#_Toc417771399)

[5.4 Integration Tests ) Martin 78](#_Toc417771400)

[5.5 Give it to some user tests david and michael 78](#_Toc417771401)

[5.6 Conclusion david 78](#_Toc417771402)

[Chapter 6: Conclusions and Further Work 79](#_Toc417771403)

[6.1 Introduction michael 79](#_Toc417771404)

[6.2 Achievements david 79](#_Toc417771405)

[6.3 Personal gain (shit learned) all of us should write a bit 79](#_Toc417771406)

[6.4 Further work ( possible improvements ) Michael 79](#_Toc417771407)

[6.5 Conclusion david 79](#_Toc417771408)

[Appendix A: Project Proposal 80](#_Toc417771409)

[Appendix B: Project Diary 81](#_Toc417771410)

[Appendix C: Code Listing 82](#_Toc417771411)

[Bibliography 83](#_Toc417771412)

# Introduction

## Introduction

Fantasy sports games are a fun and interactive experience for sports fans. Existing fantasy sports games allow fans to immures themselves in to a competitive environment and compete against other fans. Fantasy sports games operate with users registering, creating teams with their favorite sports stars and entering online leagues with fans that have also created fantasy teams of their own. Existing fantasy sports games such as fantasy premier league and fantasy baseball provide research statistics detailing the scale of online fantasy sports communities [missing citation] [missing citation].

In Ireland there are two major sports, Soccer and Hurling. Neither of these two sports have a large involvement with fantasy sports games. Although attempts exist online, there are no successful implementations of a fantasy hurling game. With no established game available, hurling fans are deprived the experience of an interactive web based social environment. The purpose of this project is to implement a fantasy hurling game that attempts to fulfill a successful version of an already successful fan experience and applying it to a sport that is lacking an online interactive gaming community.

This paper describes our research and study of fantasy games, online communities and online Gaelic sports websites. We have taken the findings of our research and have detailed how we applied them to the creating of what we feel is a more intuitive, simplified and better social experience for hurling fans. Our application allows hurling fans to have an interactive social gaming outlet.

Our implemented project provides fans with the ability to create, edit and manage their own fantasy team but a key feature is the facility to converse with other online gamers through a social chat environment. Our project follows existing structures for gameplay once the user creates an account. The simplified rules are as follows.

* The user is allocated a maximum budget
* The user uses his budget to create a full team of real life hurling players
* The user can create or enter an existing league of other gamers
* After each real life game week the users points are calculated and added to their current points total
* The user can edit the players on their team but must always be within budget
* The game ends at the end of the real hurling season and the user with the most points in the fantasy league is the winner

The addition of the chat feature allows users to compare and discuss their teams throughout the season. A key feature of the chat function is the ability for users to challenge one another in a head to head style competition each game week.

###### System Introduction

The system runs in two environments. The user has a front-end environment for playing the game playing through their browser. The administrator has a back-end environment for the maintaining and updating of scores as the game weeks happen.

The technologies used in the project are native to web based applications. Therefore we use HTML, JavaScript and CSS to design the interfaces. To connect the front-end user environment to the back-end database we have implemented .NET Web API and Entity Framework, we will host the application on Microsoft’s azure servers. These technologies are detailed in following chapters.

## Aim and Objectives

The objective of the project is to build a fantasy football hurling website, with an element of social media capabilities built in. The reasoning behind making this project is as follows. Anyone who plays fantasy football games knows that although they are quite interesting, a lot of the time is spent just logging in, checking your score, maybe making a change or two, and then logging out. The amount of time actually spent on the site is minimal, I feel this is a mistake in the design of these applications and games.

In this document we will provide some analysis and conceptual design of the product using UML diagrams and modelling techniques. We will look at the following:

* Comprehensive Use Case Diagrams
* Development Methodology
* Sequence Diagrams
* Class Diagram of the final product
* Activity diagrams
* User Interface Design
* Database table and schema design and description

## Main Research Questions

The main questions of research in regards this project are how can we improve on similar apps that are already out there, how much can we find out about the game of hurling, can we find out how to get live figures through a RSS feed, is it possible to use real Logos or are there licencing issues, and of course what technologies will we implement in our project. We should also try research if there are other better versions of what we might be able to create out there, we need to find a gap in the market, and if one does not exist the project is doomed. A big area of research for all of us will be how to use Java server to our advantage as we plan on using this in the project.

**Is there a potential gap in the social media market for a fantasy hurling game?**

With over 175,000 people on Facebook interested in the GAA. With a worldwide audience now able to watch live hurling games. Is there enough interest to justify the creation of a fantasy hurling game?

**Could a fantasy hurling game create revenue for the GAA?**

Could a successful fantasy hurling game be sold to the GAA, to capitalize on their social media presence and create revenue through their website?

**Could a fantasy hurling game create social media interest for the GAA?**

If a successful implementation of this project was achieved, Could it generate interest on social media and subsequently expand the number of people engaging with the GAA?

## Justification and Benefits

* There are no similar apps out there for hurling
* Hurling is a very well followed game
* The social media aspect can make it even more popular as outlined in the next section
* The instant messaging service would be unique to a fantasy gaming app
* The possibility of advertising revenues is very high
* Most importantly, hurling fans we have talked to want it!

## Feasibility (Assumptions, Risk, Benefits)

This project is team work of three third year Computer science students. Each participant must have access to a personal computer and internet connection. That should not pose a challenge to acquire. We should be able to finish the project using open source tools exclusively. There are available open source tools for any task this project would require.

At this point it’s not decided what technology (programming languages / frameworks) we are going to use for application development.

Because this project is part of the third year curriculum therefore we have dedicated time available for the project.

## Proposed Methodologies

* First we will need to carry out a preliminary research of similar projects, in other words we will need to see how fantasy sport sites or applications works and what features they offer.
* Secondly we will need to decide what social networking features we want to implement in the application, hence we should undertake additional research toward contemporary social network paradigms.
* Next challenge lies in the fact that either of us have any real experience developing web applications. Comprehensive study of web technologies and framework is therefore required.
* To carry out the research we will use internet and focus mainly on qualitative research methods
* To develop the application we should use adaptive approach.
* I would argue that Prototyping SDLC is the best approach to carry out the development. In this model developer basically re-analyzes, re-designs and re-implements application prototype until the product is accepted by the client. Considering our lack of experience with web application development we should expect major changes in design, but we should be able to have simple, but functioning prototype reasonably fast.

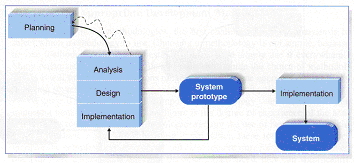


Figure 1 Prototyping SDLC

## Expected Results

With no current social gaming outlet for hurling fans this project could prove fruitful. The potential gap in the market could exist for a game like this to succeed. A successful game could function by itself but could also be sold to the GAA and implemented on their website in a similar fashion to the premier league fantasy football.

The GAA could be interested in integrating a social media game on their website to attract the number of followers they have on social media. A fantasy hurling game could increase traffic on the GAA website and generate user interaction, leading to increased advertising revenue.

With social media friends challenging, debating and boasting about their fantasy team though various social media outlets, the GAA can reap some free publicity and advertising for upcoming games.

###### Successful Project

A successful implementation of this project would create an exciting social environment for hurling fans. The fantasy hurling game would allow users to post their results and scores on social media sites. To be successful users should find the website interface easy to navigate. Playing the game should be intuitive and simple. Selecting to use your GPS coordinates to find other users in your area should be fast. Users should be able to retrieve their fantasy team’s results on the final day of a game week. The server should be responsive and load player information quickly. The user-to-user messaging service should be used frequently, for both challenging and talking about hurling. The system should be modular and deployable for a number of different sports.

The game should be complete and function well enough to potentially be sold to a third party, namely the GAA. Without deconstructing the system the game could be implemented on to a new website with little disruption. This would make the game profitable and marketable. This would be the ultimate goal and a truly successful project.

###### Successful System

A successful system would implement all the functionalities outlined in the methodology. Users could create and account, login, customize a profile, create a team, create a league, invite users, challenge users and use the messaging features within the system. The client side of the system would interact with the database. Data related to players scores and results would dynamically update on the client side after being updated by the administrator on the server. However if the game failed to attract interest from gamers and investors, the project may prove to have been created in vain. A successful system could be modified to adapt to different sports but without users playing the game the project would be considered incomplete.

###### Failed Project

A failure to create a functioning system that allows users to play the game would be considered a failed project. The project aims are to create fantasy hurling gaming system with social media capabilities that can operate as a standalone website. Potentially if the system created was successful and the market was there for hurling fans, users of social media and an investor to interact and back the game, the project would be considered successful. Failure to create a functioning system however has a knock on affect to potential investors. How can an investor buy a system that does not exist? A failed project can only occur if the team fails to build the system.

###### Incomplete Project

Failure to complete the system before the deadline could be considered a failed project. The difference between incomplete and failed is that the team mismanaged the time allocated to create the project. The system could still be completed in future iterations. If the system is completed outside of the allocated time and the market still exists for a fantasy hurling game with social media capabilities, the project may not be considered failed and could still potentially be distributed at a later date.

The plan is to deliver fully tested and working application implemented on the web in early May 2015. We would consider to be a great success if we managed to add few additional features and make the application stand out from the competition. The most important reason why we agreed on this project is an exposition to cutting-edge web technologies such as HTML5, Angular.js, GWT, Node.js and Rubi on Rails etc. We are planning to exhaust this opportunity for learning.

## Work Breakdown Structure

* 1. **Design**
     1. User Interface
        1. Specification
        2. Interface Model
        3. Design review
     2. SQL Database
        1. Identify table relationships
        2. Database model diagram
     3. User case, class and sequence diagrams
  2. **Development**
     1. Research on web technologies and frameworks
        1. Find most appropriate technology
     2. Front-end
        1. Code web pages
        2. Conduct unit tests
        3. Review web page design and functionality
     3. SQL Database
        1. Build database
        2. Review database
     4. Back-end and web services
        1. Code back-end
        2. Conduct unit tests
        3. Review back-end functionality
  3. **Quality Assurance**
     1. Front-end
        1. Perform integration tests
        2. Perform user acceptance tests
     2. SQL Database
        1. Perform integration tests
     3. Back-end and web services
        1. Perform integration tests
  4. **Implementation**
     1. Hardware
        1. Determine hardware needs
     2. Software
        1. Determine system software needs
     3. Deployment
        1. Make hardware production-ready
        2. Verify code
        3. Deploy
  5. **Post-Implementation**
     1. Verification
        1. Obtain user acceptance of production system
     2. Monitoring
        1. Verify performance and functionality
     3. Project Wrap-up

## Gant chart

Figure 2 Gant Chart

## Conclusion

In order to begin the analysis phase of the project the team must research all the data relevant to the proposed fantasy hurling game with social media capabilities, Information about social media interaction between the GAA and hurling fans. With no current game similar to the proposed project available, there appears to be room in the social media gaming market for a fantasy hurling game to exist and benefit fans of the sport and the organizers.

Fans can use the game as an outlet to interact with fellow hurling fans and feel involved in their sport. The GAA can use the game to attract visitors to their website. The project would need to consult the GAA over licensing laws and complying within the existing copyright laws.

The team involved has all the necessary skills to implement this project. Referencing the task time line in 6.2, the project is feasible within the allocated time. The most difficult parts of the project will be creating the website and updating the real life player ratings on the database.

We, a group of three Computer Science Students is going to develop a feature rich application which would bring the fantasy hurling game to fan audience all around the Ireland. We hope this app will bring people who love hurling closer together. Serious lack of fantasy hurling games on the web made us confident that our application will be welcomed among potential users. It is an exciting project and a great chance to pick up on modern application development skills and techniques.

# Literature Review

## Introduction

The project required extensive research of literature. We each picked a topic and reviewed a relevant piece of literature. One topic was social media, another was back end development. We used the knowledge gained in these reviews to help us in the overall project.

## Modern Web

###### Abstract

Today’s web is more complicated than ever. Web content is delivered to desktops, laptops, mobile devices, game consoles, and smart televisions. Each has a different screen size, performance capabilities and additional features. Web content is delivered in the form of dynamic web pages and feature rich applications in an environment without a common standard in web browsers. In addition to that, internet usage reaching billions online with all the human diversity additional complexity can be problematic. This paper purpose is to review some studies and white papers that address these problems. The research done in this field is immense and produced many solutions in the form of frameworks, tooltips, plugins, design patterns and methodologies and it’s out of the scope of this paper to cover all of those. I tried to name only general challenges in this review. Therefore I chose three white papers to study with additional research of relevant articles on the web. My work was driven by the need to acquire better understanding of concurrent state of the web, the web development challenges and methodologies. In conclusion I can claim that this paper delivers just that. It does not provide any specific solutions, but general recommendations.

###### Platforms: Mobile devices are on the move

Each month an increasing number of people are switching from PCs to tablets and mobile devices to browse the internet and by 2015 they’ll be using those more than any other device. Mobile traffic grew 120% between Q2 2013 and Q2 2014 while desktop traffic remained flat.

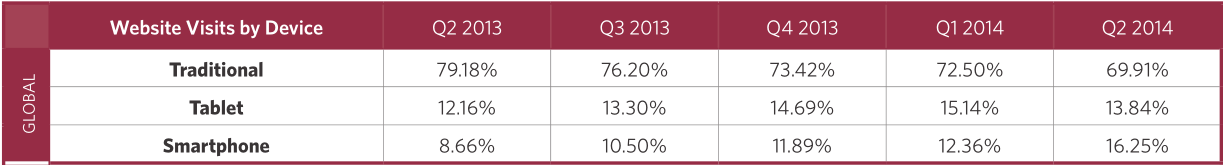


Figure 3 IBM Analytics Benchmark

From a development point of view certain areas on a website may function differently or may not function at all on mobile or tablet devices.

**Some of the issues are:**

* No Flash support
* Popups stops a website to being used on a small screen
* Tiny text links unusable on the touch screen
* Website elements overlapping on the small screen
* And many more…

Screen size and performance restrictions are the main reasons for design reevaluation of the application design. Web developers are using separate mobile pages, responsive web design and business apps to ensure a good user experience regardless of device type.

###### Browsers: Many knights, but no king

In the past, nobody questioned which browser ruled the land – it was Microsoft’s Internet Explorer. According to the latest analysis of the Web browsers, situation is very different now. Web browser market share varies from platform to platform.

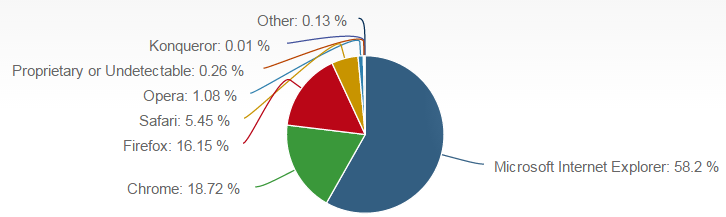


Figure 4 Desktop Browsers

Microsoft still wins desktop market with 58.2% share has acknowledged being critically late to mobile where his presence is almost negligible with 2.21% share. This led to domination of Webkit-based browsers: Safari, Chrome and Android with Apple’s Safari being the leader with 48.91% share.

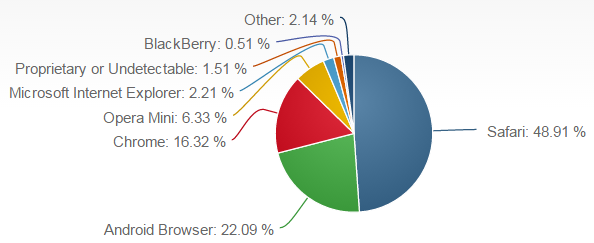


Figure 5 Moblle Browsers

Situation in console market is very different with most of the market shares in hands of proprietary browsers embedded in a particular device. Only Opera is getting a 4.51% share as a result of Nintendo implementing Opera browser in some of their consoles. (1).

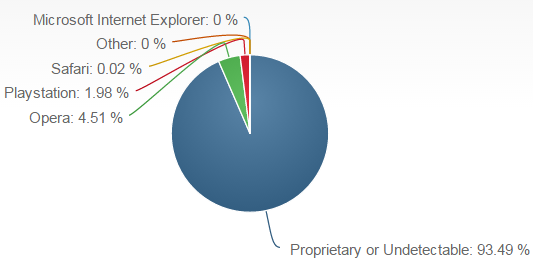


Figure 6 Console Browsers

It’s great that internet users have so many browser options. Unfortunately developers must put a lot of work into making sure that websites are compatible with as many browsers as possible.

**Here are some of the most common browser compatibility issues**Invalid source specified.**:**

* Elements not correctly positioned
* Browser specific CSS styles
* Lack of Valid HTML/CSS
* Javascript processing times
* Frames related problems

There is a vast research done to tackle these issues and many useful practices, methodologies, frameworks and tooltips have arisen in recent years. Technologies like HTML5 and CSS3 are making cross-browser problems more manageable and web developers are getting less concerned about browser compatibility issues (2).

###### Sources: More content is coming from third parties

Third-party content is a term that covers a lot of ground, from advertising to guest blog posts.

**The term frequently include:**

* Content delivery networks (CDNs)
* Advertising
* Web analytics
* Social Media
* Microblogs
* Miscellaneous Widgets

Analysis shows that the average web transaction involves 8.59 different third parties or hosts. (3). These web page components are dragging down the overall performance of the website. Various research was done to illustrate the impact that third-party additions have on web site performance. Some of the third-party widgets can impact a site’s performance by as much as 30% (4). In spite of these findings third-party components are virtually indispensable to many web sites today, whether they bring in revenue through advertising, or drive business by bringing in new users through social media. To tackle performance issues the best option includes continuous performance monitoring of the web site to pinpoint the components with the worst impact.

###### What developers think?

Recent study (5) took a different approach to name modern web challenges. They tried to find out what are the issues that web developers are discussing among themselves. To do so, they examined over 500,000 questions asked on <http://www.stackoverflow.com> over a four year period (2009-2013).

**Here are the big takeaways from their analysis:**

* Javascript is still most popular topic, but HTML5 and CSS is raising
* Cross-browser compatibility still big, but declining
* New HTML5 and Javascript features questions are raising
* Web technologies are growing in importance in mobile development

###### Conclusion

Modern web is a complex and lively medium. New technologies arise on daily bases. Users are connecting to web with various devices through different browsers. Webpages of today are dynamic applications with third party content interconnecting different domains. With all these fast paced changes, the contemporary web developers are facing many new and difficult challenges. Evolving and researching never stops in the life of web designers and developers. New frameworks, toolkits, methodologies are created to tackle major issues and improve web development productivity. Some of the examples of prominent frameworks are Bootstrap, Grails, jQuery, Node.js (6).

The user doesn’t care about all these technical issues. He wants to browse his web sites from laptop, mobile phone or game console. He expects a web page to perform quickly and properly every time and on every device. He wants to share content or buy a product by touching a widget. All this is possible on today’s Web.

## Improving Website Design

###### Who did what?

Melody Y.Ivory and Marti A.Hearst University of California, Berkeley 2002 investigated what makes a good design to a website, and developed a model to help users implement these changes that would make the website more pleasing to the eye and in terms of usability.

###### Summary

Two students undertook a paper to investigate current methods of website design, their effect on productivity and how it can be improved though better website design principles. They undertook this as part of the Web Tango project. They aimed to help steer the average web site builder away from poor design principles and toward an automated quality checking tool and a grammar checking tool. They made these tools available online at “webtango.berkeley.edu”.

Melody Y.Ivory and Marti A.Hearst (2002) found that a website is a complex mix of text, links, elements and formatting, surmising that all these aspects affect a websites quality and usability. They came to the conclusion that these principles are important in thinking about website design to begin with.

The students came up with a table that created a way of calculating how many measures for each element on a web page were needed in order to come up with a design that is pleasing and easy to use. For example, they came to the conclusion that there are 31 separate measures that are important to think about in regards a text element including amount, size and complexity. The table also related to site architecture and the performance of the page, all elements discussed had a set of measures that were important to that particular element to optimize design.157 Measures in total were found.

Melody Y.Ivory and Marti A.Hearst (2002) then ran their crawler tool and used this to gather sample web pages. Initially it ran on the home page of a website and randomly selected pages at successive levels starting at that page and only selected informational pages ignoring advertisements or flash pages totally. The analysis tool then runs on these pages and retrieves the information on each element in conjunction with a site metrics computation tool, forming the table they discussed with the elements and their measures.

The students found there were three main principles to successfully designing a webpage; navigation design, graphic design and experience design. From these 3 main principles a hierarchical pyramid model was built, surmising that on the top level is the site architecture, while on the bottom are the actual site elements themselves. Web design literature and user studies were used to come up with the final model. A tool was then developed from this model that could compute 157 site level measures. The accuracy of this tool was tested on many websites and it was found to be 86% accurate on 154 measures.

The students performed three studies to try to predict page and site ratings. From these 3 they developed a simple prediction model. They called it the WebTango model. Firstly they drew up an analysis of 428 web pages and found expert reviews and ratings on these pages from PC Magazine’s top 100 sites. They called sites either rated or un-rated and set out to find a way to predict which category a sample site would fall under. They then computed 12 quantitative measures related to page composition and design among other factors. They tried to see if they could predict with their model the page standings on this top 100. They found that 6 features were most important to design. The most prevalent were text cluster, reading complexity, and colour count and page size. They found that in rated sites these features needed to be tweaked a certain way to make the site very usable and stay in the rated group.

A second study was conducted and asked 6 website design experts to examine 1898 pages from the Webby awards winner’s websites. These pages were judged on certain criteria including content, structure, navigation and visual aesthetics. They broke these pages into three groups, “good”, “not good” and “poor”. They wanted to see if their model, using the measures they had created, could predict which group a site would fall into. Predictive accuracy was 67%.

The third and final study was to analyse over 5000 pages from 300 sites. They used all 157 measures from their model and again had 3 groups, “good”, “average” and “poor”. They used the model to try predict which group the web pages would fall under. It was proven to be accurate on a page level 96% of the time, and accurate on a site level over 60% of the time.

Melody Y.Ivory and Marti A.Hearst (2002) talked about their final task of applying the model they had built to website design. They took a sample of 15 web pages and made minor tweaks to these based off their model parameters. They asked 13 people to analyse the pages both before and after the adjustments had been applied to them using the model. They made findings that 10 out of 13 people preferred the web pages after they had used to model to make adjustments to it.

The students then analysed their final tool, the WebTango system. They analysed how the tool worked and how it can be applied to a website. The tool compared all 157 site level measures from the website, and then makes suggestions on how to improve the website and also gives links to example websites who are similar in type but have been designed to a higher level.

The major findings were in essence that a model can be developed by 2 students with little or no web design experience that can enhance the usability and visual aspects of a website very quickly by identifying areas that can be improved. It found that although some studies have found automated tools find it difficult to find problems with usability issues, the tool developed is considered a practical and useful solution to solve design issues early in the design phase of a websites development.

## The effect of fantasy sport participation through social media.

Sport spectatorship is one of the largest forms of leisure behavior in the world [missing citation]. Large numbers of sports fans attend sporting events frequently and follow their sport ubiquitously through various mediums, TV, radio, newspapers, and magazines. With advancements in Internet technologies, fans are redirecting their consumption of sport to online platforms. Sports fans can now consume news and content via applications and social media on mobile devices. Fans can connect with each other through tweets, texts, blogs and Facebook.

Fantasy sport participation is not a new phenomenon; the history of fantasy sport is a topic of debate. In America the formation of a fantasy baseball league in the 1960’s by a man called Bill Gamson is considered to be the first documented account of a fantasy league [missing citation].

Today the exponential growth of fantasy sports participation is fuelled by the increased televised broadcasting of sport, accessibility of sport and statistics on the Internet and the advent of social media as a communication outlet for fans worldwide, because of this, fantasy sport platforms are now formally aligned with many official major sporting leagues globally.

Fantasy sport provides engaged fans a unique and personal opportunity to be involved while also building the interest and knowledge of new participants with lower prior involvement. Further, fantasy sport provides additional social interaction between players. It can serve as an education tool about the sport, teams and players and can also serve to increase the commitment and interaction of consumers with a sport.

###### 2.2 Existing Trends

In North America and Canada the fantasy sports industry has a body that monitors and provides information on all fantasy sports websites known as the Fantasy Sports Trade Association (FSTA) [missing citation]. The FSTA released figures on the number of participants in 2011, more than 35.9 million users are involved in some type of fantasy sport. The FSTA then surveyed the group to find out how many users participate using mobile devices. The survey found that 30% (12,000,000) of users are engaging through mobile devices. Further, of the 12,000,000 users that play through mobile devices 14% (1,680,000) use social media to discuss their fantasy games.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Members | Mobile Users | Social Media Users |
| BPL | > 3,000,000 | 900,000 | 125,000 |
| FSTA | >35,900,000 | 12,000,000 | 1,680,000 |

The Barclays Premier League (BPL) claims to have the biggest football fantasy game in the world [missing citation]. The Fantasy Premier League game has over 3,000,000 participants. If we apply the same calculations from FSTA to the Fantasy Premier League game we can get an estimate value of over 125,000 users that discuss their fantasy football game via social media.

###### Reason for Literature Review

This literature review is being conducted to ensure a thorough understanding of the connection between social media communities and fantasy gaming. I am currently working on a project that is combining a fantasy-hurling platform that enables social media capabilities.

Upon completion of this literature review I hope to identify potential areas of difficulty that may be unforeseen without specific research in to the topic of social media and fantasy gaming.

## The effects of fantasy football participation on NFL (National Football League) consumption: A qualitative analysis.

Joris Drayer, Stephen L. Shapiro, Brendan Dwyer, Alan L. Morse, Joel White. (2010) *Sport Management Review, 13*, 129-141. [missing citation]

Drayer, Shapiro, Dwyer, Morse and White proposed three areas of interaction that may be affected for fantasy football users and non-fantasy football users.

1. Identification and loyalty
2. Consumption
3. Attitudes and behavior

###### Identification and loyalty

To conduct a test on the fantasy gaming NFL fans, the authors must first define a norm. A fan that does not participate in fantasy football and the attributes of one such fan. A normal fan is considered to have allegiance to one team. The normal fan supports the team and the players that play for their team and no one else.

The authors expected gamers to have no loyalty or identification to a certain team, but rather have identification to a player. This was not the case, gamers still identified themselves with a team but their affiliation to other players via the fantasy game was just an extension.

Those interviewed stated that their loyalty to a fantasy player ended with the season.

Only 2 people of the 13 interviewed said that they carried affinity for a player in to the next season and rooted for them if their favorite team wasn’t playing.

###### Consumption

To conduct this test the authors once again had to define a norm. A normal fan is one that only consumes information about their favorite team.

The authors expected gamers to consume NFL information in a different manor. Having affiliation with many players in many different teams’ leads to greater consumption of information.

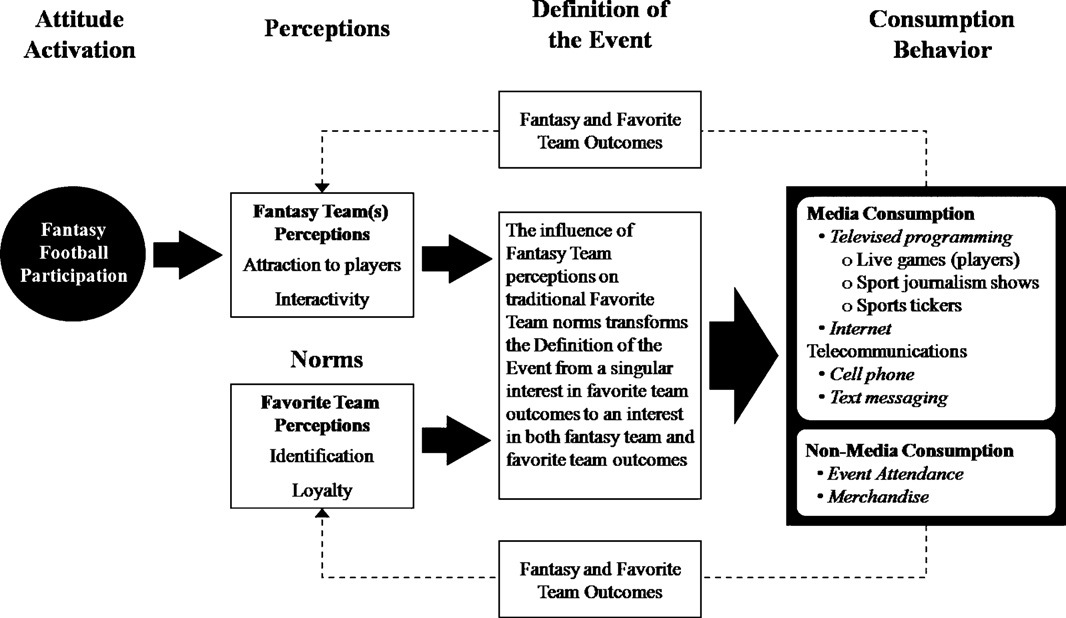


Figure 7 Consumption

Fantasy gamers have an increased media consumption of the NFL, spending time on the Internet and watching ESPN to gain information on starting rosters, injured players and to check which teams are playing in game weeks. Of those surveyed participants admitted to checking their mobile device and going on to the Internet every 10 minutes while a game is on. Some gamers paid for subscription sites to get informed advice on their fantasy teams. Fantasy football participants found themselves watching more matches on game weeks because they had numerous players playing in different games. When asked if they would watch as many games if not playing the fantasy football game, the answer was no.

No change was noted in the form of consumerism. Only one person said they would buy the jersey of a player because she began to like him in her fantasy team.

###### Attitudes and behavior

The control norm for this test was that of a fan, who regardless of their favorite teams result would support them.

The results found in the survey suggested that a fantasy football player would have allegiance for a team but if that team lost, the fantasy football player would adjust their attitude towards a real team to suit their fantasy game. When the 13 fantasy gamers were asked about their behavior towards an opposing player, 2 gamers admitted to supporting opposing players of their favorite team.

Those interviewed stated that if their fantasy football team began to lose and become uncompetitive they would lose interest and stop updating their team towards the end of the season.

###### Research Paper Conclusion

The research results show that fantasy gaming participation really does have an effect on how fans, consume, interact, behave and identify themselves with teams. A non-fantasy gamer does not consume as much information over a broad scale but rather focuses solely on their favorite team. With fantasy gamers this is not the case. Fantasy gamers spend more time online, on blogs, sports news websites, social media, TV, magazines trying to get information on their fantasy team. Fantasy gamers tend to be more fickle in their allegiance to a team and player, favoring and supporting teams and players that can generate points for them in their fantasy league.

## Using Social Media to Build Community

Kurt Komaromi, Scott Erickson (2011). *Competition Forum,* *October 27-29*, 325-333. [missing citation]

Komaromi and Erickson follow a template laid out by Culnan, McHugh and Zubillaga that assesses the effectiveness of an organizations social media strategy. Effective implementation of a social media strategy is one that attracts an audience enabling them to engage and interact with both the organization and each other.

Although the study is not conducted on a sports category, the essential requirements for a successful social media community apply in all communities. The study instead researches 18 insurance firms over a two-month period in 2011. The authors then present specific results for three firms that display a range of social media outlets to support a community.

The three insurance companies presented are, Progressive, Liberty Mutual and State Farm. The study begins by analyzing the company’s website homepages. The homepages all display a range of tools that allow customers to interact by getting quotes, mobile apps, price comparisons and calculations for various insurance plans. What is important on all these homepages is the prominence of social media resources. All three websites have social media icons to connect via Twitter, Facebook and YouTube. Liberty Mutual is the only site that has a LinkedIn page that allows customer access. State Farm has an additional social media outlet through Flickr.

Metrics for each homepage visit can be seen in the graph below. Progressive has the highest unique visit rate at over 4 million visits per month. State Farm is second with 2.5 million and Liberty Mutual is third with 1 million unique visits.



Figure 8 Website Traffic

###### Facebook

Facebook is the first social media outlet covered in the findings. All three insurance companies are present on Facebook.

Progressive have a Facebook page that primarily provides information about the company, offers and customer feedback. Progressive also have a current marketing campaign that attracts the most followers of the three companies Facebook accounts. “Flo” is the name of that marketing campaign and has a Facebook account that is updated daily and has interactive videos and games to engage customers. At the time of research, Progressive has a following of over 2.5 million on their Facebook accounts.

The Liberty Mutual Facebook page is similar to Progressive, They have an about section to provide information about the company. Further Liberty Mutual has videos, comments, polls and games on their page to engage customers. Their Facebook wall has daily posts from employees and brokers that can be interacted with by visitors via likes and comments.

State Farm’s Facebook page offers a diverse content including offers, promotions, and sponsorships for social causes, games to rate drivers and apps that can be played with friends. The site also has a driving game application to complete a safe driving discount offer. Their wall has many comments both by State Farm and visitors.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Progressive | Liberty Mutual | State Farm |
| Facebook | ~2,500,000 followers | ~10,000 followers | ~90,000 followers |

Although Progressive have an advantage with a larger hold on the market, the evidence is clear that an updated and well-maintained Facebook page that interacts with visitors gathers more followers. State Farm has a large number of followers because they engage their visitors with apps, polls, videos and respond to comments daily.

###### Twitter

Twitter is the second social media outlet researched in the report. All three insurance companies have an active twitter presence.

Progressive utilize twitter to connect with customers with posts daily and responding to complaints, questions and criticism. Progressives Twitter account post basic answers to very common customer questions in a timely basis.

The Liberty Mutual Twitter account is active once or twice a day. Their Twitter account is used to promote news and answer questions from customers. Liberty Mutual also use their Twitter account to promote sponsorship activities and encouraging followers to attend social events.

State Farm populates their Twitter feed with posts daily, ranging from comments, questions, complaints, and re-tweets from other third party sources. The emphasis is on responding to customer queries. State Farm is more concerned with customer comments and feedback than Liberty Mutual and Progressive. State Farm shares a lot of content from their other social media sites through twitter, connecting their Facebook, YouTube and Flickr accounts.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Progressive | Liberty Mutual | State Farm |
| Twitter | ~7000 followers | ~2500 followers | ~30,000 followers |

###### YouTube

|  |  |  |  |
| --- | --- | --- | --- |
|  | Progressive | Liberty Mutual | State Farm |
| YouTube | ~2,000,000 followers | ~250,000 followers | ~3,000,000 followers |

YouTube is the third and final social media outlet researched in this paper. All three insurance companies again utilize YouTube as a connection tool to their customers.

Progressive don’t only use their YouTube account as a video player but also implement a quote app link and link to their other social media outlets. The site has adverts, promotions and visitor uploads. Progressive had 97 videos uploaded to their account at the time of research and the comments section on their YouTube channel is very active with over 400 comments.

Liberty Mutual has 31 videos uploaded to their YouTube account, providing both advertising and more in-depth information. Liberty Mutual does not update or maintain their

YouTube account frequently with only a hand full of comments on their account page.

State Farm has the most amount of content on their YouTube account, with over 180 videos at the time of research. State Farms videos range from adverts, promotion offers, to custom content. Fail Fridays is a popular video series that compares customer reasons for claiming insurance. State Farm also provides instructional videos to prevent accidents. Comments are extensive across all their videos and a find-an-agent app is available.

###### Research Paper Conclusion

Social media connections to customers are of huge importance to building a community around a product. The statistical analysis of three insurance firms provides evidence of such a statement. State Farm update their social media accounts more frequently than the other two firms as a result the community surrounding State Farm is larger and engages more with the company. State Farm makes it important to respond and interact with followers daily and update followers with relevant information regarding their industry and company.

###### Research Paper Connection

Although the topics in both research papers are very different, the ubiquitous nature of social media is the connection. Taking advantage of social media to connect fantasy sport gamers and allow them to share their information is easier said than done. In order to understand the best methods of engaging fantasy sport fans and encourage them to use social media as a platform to create and interact with a community is the problem.

The first paper reviewed states that fantasy sports gamers clearly consume information at a higher level than that of a normal spectator. Fantasy sports gamers gather more information regarding a sport through magazines, TV, radio, blogs and online. A part of making the transition to social media is by creating a community where user content is updated and fans can interact with one another, sharing information, tactics, statistics and analysis.

The second paper reviewed contains data and an analysis of social media communities that already exist. This data gives an insight in to how a social media community works.

State Farm take advantage of their social media presence more than the other two firms. The number of followers proves that maintaining a regular social media presence attracts followers. Having a Facebook page and Twitter account is essential to connecting with an online community. All three firms in the research paper use these social media outlets regularly to post comments and just as importantly to receive questions, input and criticism.

###### Gaps in Research Papers

After doing research external of this literary review I would like to draw attention to some gaps that I feel exist in both research papers. Although the connection between the two papers is very real and an outlet for fantasy sport gamer discussion can be managed in a social media platform, questions must be asked about the data analyzed.

The research paper by, Joris Drayer, Stephen L. Shapiro, Brendan Dwyer, Alan L. Morse, Joel White. **“The effects of fantasy football participation on NFL (National Football League) consumption: A qualitative analysis”** is a research paper about fantasy football in America and not sports globally, including hurling. Questions about a fantasy sports player consumption of information must be asked in order to get a complete view of the fantasy sport industry.

* Do all fantasy sports players consume more information about their sport than that of a normal fan?
* Do fantasy sports games accommodate gamer interaction?

The survey documented in the research paper reports on only 13 test users. The information could be incorrect or corrupted by such a small number of test users.

Kurt Komaromi, Scott Erickson. **“Using Social Media to Build Community”** researched the social communities surrounding three large insurance firms in America. The data gathered in this research paper is informative and formidable but questions still exist in the form below.

Can the social community of an insurance firm be relevant to that of a fantasy sport community?

* How do insurance customers differ from fantasy sports gamers?
* Have all forms of social media been researched?

###### Literature Review Conclusion

It appears that fantasy sports gamers may consume information and data of their sport on a different level to that of a normal fan. Fantasy sports fans follow a greater number of players and teams connected to their fantasy game. A platform to cater for an online community of gamers could exist to allow gamers to share information with fellow gamers. Fantasy gamers spend more time on mobile devices during games and between game weeks researching real life player and team data.

Fantasy players can still identify themselves normally with one team but can also extend their identification to a number of other teams during the season. Gamers however have a tendency to lose interest in their fantasy games if their fantasy team is performing poorly or their real life favorite team is performing well.

Social media communities flourish when content is updated regularly and interaction between community members is high. Building a social community takes many platforms through YouTube, Facebook and Twitter. Connecting social media outlets together such as linking videos on Twitter to videos on YouTube engages the community to interact within itself.

Having platforms high in content produced by community such as comments and questions creates a larger following. Users tend not to follow social media outlets that rarely post content. The insurance company with the smallest amount of content on social media platforms had the fewest number of followers. The insurance company that had the largest amount of content, videos, posts, tweets and applications had the largest number of followers. The connection between a maintained up to date social media presence directly relates to the number of followers in the community.

## HTTP Protocol Overview

Hypertext Transfer Protocol is a stateless application layer protocol for communicating between distributed systems (7). HTTP is specified by RFC 2616 (8).

HTTP is connectionless, stateless and media independent protocol. A client sends a request and server answers with a response. After that the connection is closed and communication is forgotten on both sides. Current version of HTTP is 1.1.

HTTP is using Uniform Resource Identifiers (9) for resource requests.

###### Generic URI format

URI = "http:" "//" host [ ":" port ] [ abs\_path [ "?" query ]]

Example

https://www.youtube.com/watch?v=BNgU-ZaF06w

###### HTTP request

A Request-line

Zero of more header fields ending with CRLF

An empty line ending with CRLF

A message body (optional)

Request Line generic syntax

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

###### Request methods

* GET: asking server for a resource located at given URI
* HEAD: similar to GET request method, but only header is returned from server
* POST: asking to send a data in request body to server
* PUT: asking to replace a resource located at given URI with new data in request body
* DELETE: asking to delete a resource located at given URI
* CONNECT: asking to establish a tunnel to server identified by given URI
* OPTIONS: describes the communication options for resource at given URI
* TRACE: test the resource availability

###### HTTP Request Simple example

GET <http://www.itb.ie/> HTTP/1.1

User-Agent: Fiddler

Host: <www.itb.ie>

User-Agent and Host are some of many predefined header fields. Custom fields can be introduced as well. This request message doesn’t contain any message body.

###### HTTP response

A Status-line

Zero of more header fields ending with CRLF

An empty line ending with CRLF

A message body (optional)

Status Line generic syntax

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

###### Status Code

It’s 3 digit integer. First digit defines the class of the status.

* 1xx : Informational
* 2xx : Success
* 3xx : Redirection
* 4xx : Client Error
* 5xx : Server Error

###### HTTP Response example:

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: 14887

Content-Type: text/html

Server: Microsoft-IIS/7.5

Set-Cookie: ASPSESSIONIDQAADTRCR=KIIFFBMDCAOAICBNIDKKACIH; path=/

X-Powered-By: ASP.NET

Date: Sun, 05 Apr 2015 09:36:25 GMT

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

Rest of the html omitted…

Example above is the response to <http://www.itb.ie> request. Response returned with status code 200 which indicates the success. After Status Line the various header fields follows. Then there is an empty line followed by message body. The message body contains actual html document of requested web page. Most of the html document is omitted in this example.

## Conclusion

All the reviews were useful in research towards the eventual project itself. The reviews were of both technical and observational variety. The reviews themselves gave us fresh impetus in working on the project as they seemed to back up the idea that social media in this form would be something people would be interested in. The HTTP research was obviously vital to the project also as the front end heavily relies on these concepts.

# Analysis and Design

## Introduction

Fantasy hurling project is basically fantasy sport game. It’s rich web application not a static web page. As a player manages his fantasy hurling team a content dynamically changes. At the same time it must have a persistence capabilities. Basically a changes player made must be stored somehow and not get lost when player finishes to play. Data must outlive the application runtime.

Basically we will need to have three main layers.

* Front-End (code running in browser delivered by initial http request from web server)
* Back-End (code running on server listening and answering to http requests)
* Persistence Storage (relational database server listening and answering CRUD queries)

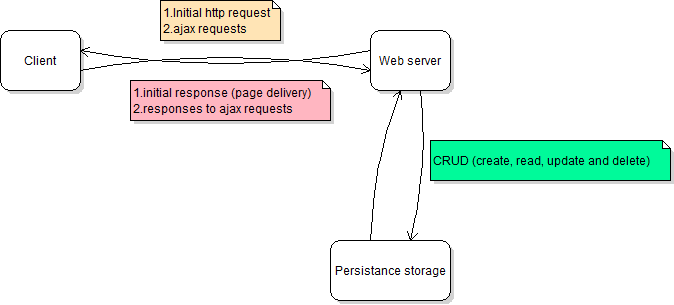


Figure 9 Web Application general design

## Proposed Methodology

In this section we are going to discuss the methodology of software development life circle (SDLC) we chose to use to develop this project. After initial research we decided for Prototyping SDLC.

In this approach the development team implements a ‘sample’ which have only very limited functionality of proposed project and show it to the customer. Customer provides the suggestions of improvements and finally the development team implements these. This circle then repeated itself numerous times until the project is fully functional, tested and accepted by customer. This approach differs from other methodologies by avoiding doing the ‘big design in advance’ followed by implementation, testing and deployment phase. The project is rather developed by mutating the prototype with numerous design, implementation testing and deployment phases until the final product is build (10).

We wanted to avoid ‘big design in advance’. To design web application in advance it requires experience. Only experienced developers who already worked on projects with similar functionality and scope are able to do that. If ‘an amateur’ tries to design the application he will realize during the implementation phase that his design has flaws and redesign is required. This will lead to lose of work hours and any codebase the developer had already implemented. Basically you need an architect to design the house. We amateurs. We never build web application of this scope. With prototyping approach we be able to redesign the sample as many times as needed.

We will learn with each prototype iteration. In our case we first build a simple but working sample build from prototypes. Data store prototype (relational database layer), data model prototype (objects to data mapping layer), controller prototype (business logic layer), and the view prototype (front end client layer). We make sure it all works together. Then we pick one proposed functionality and implement it whole way down through all the layers. And we learn from it. Implementation of next functionality will be easier and we get more productive over time. Hopefully after various iterations we will be able to call ourselves ‘the web developers’.

## Assignment of Individual Tasks

###### David Kelly

* Front-End design ***responsibility*** (wire frames, photo shop, HTML5, CSS3)
* Documentation involvement
* Testing involvement

###### Michael James

* Front-End development ***responsibility*** (HTML5, CSS3, vanilla JavaScript, JQuery, JQuery UI)
* Documentation involvement
* Testing involvement

###### Martin Zuber

* Back-End ***responsibility***(MySQL database, .NET Web API 2, Entity Framework ORM mapper, Azure Deployment)
* Documentation involvement
* Testing involvement

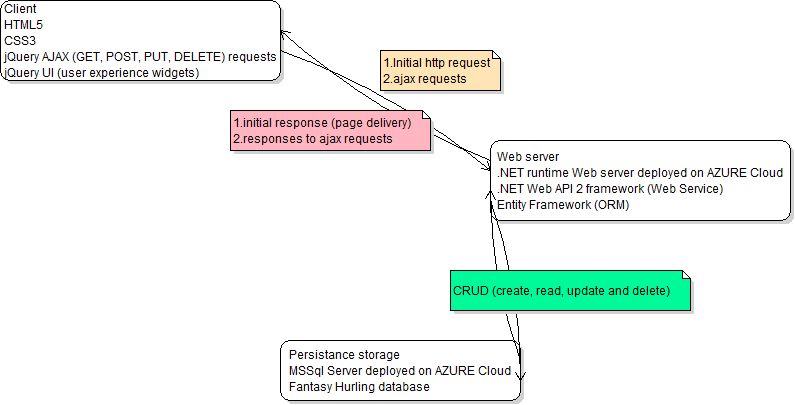


Figure 10 Fantasy Hurling Web Application in detail

## Front-End

First front end prototype is simple web page using HTML5 (11) and CSS3 (12)***.***AJAX (13) request are made through simple web forms using JQuery (14)***.*** In prototyping iteration this web page will gradually turned to user friendly application using JQuery UI (15)widgets and methods.

Front-End development will be done using IDE. Some recommended options are Sublime Text 2 (16)or Aptana Studio (17)

## Use Cases

###### User checks current score

Use case specification:

1: User logs in

1a: User logs off

1b: User not registered and is prompted to register

2: User checks their current score

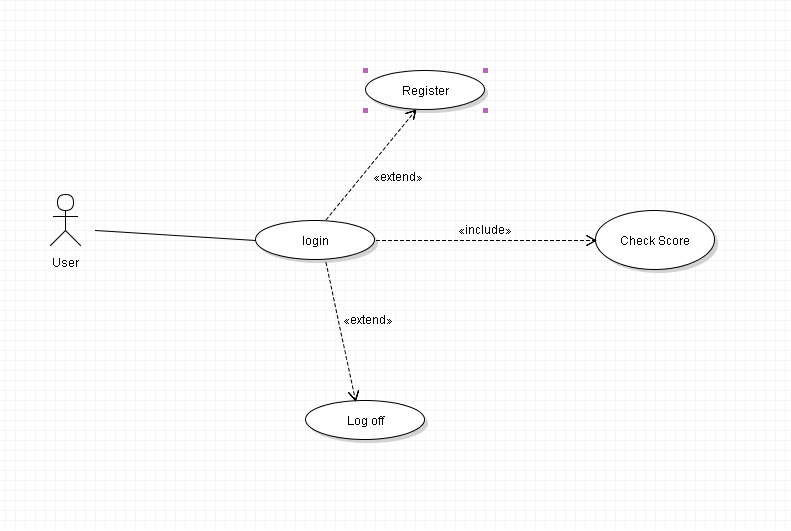


Figure 11 Use Case: User checks current score

###### User checks fixture

Use case specification:

1: User logs in

1a: User logs off

1b: User not registered and is prompted to register

2: User checks fixtures coming up

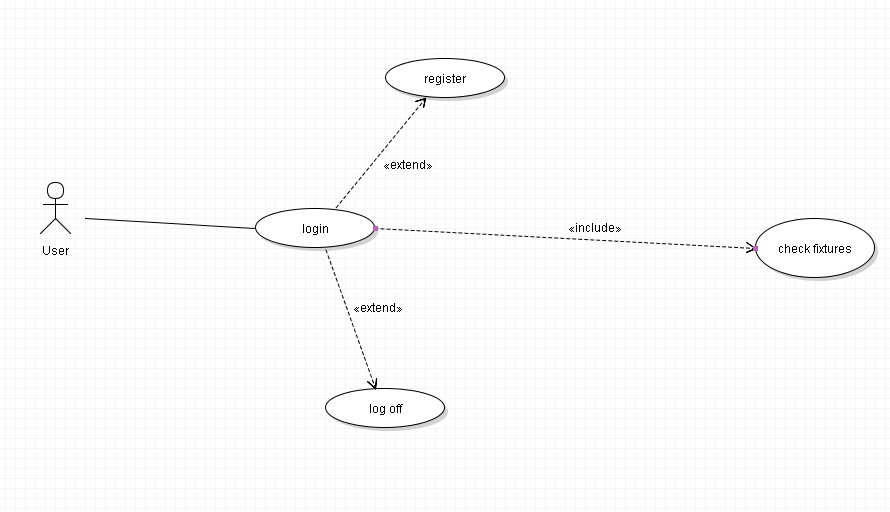


Figure 12 Use Case: User checks fixtures

###### User uses social media aspect

Use case specification:

1: User logs in

1a: User logs off

1b: User not registered and is prompted to register

2: User sends message

2a: User posts in the forum

2b: User sends an instant message

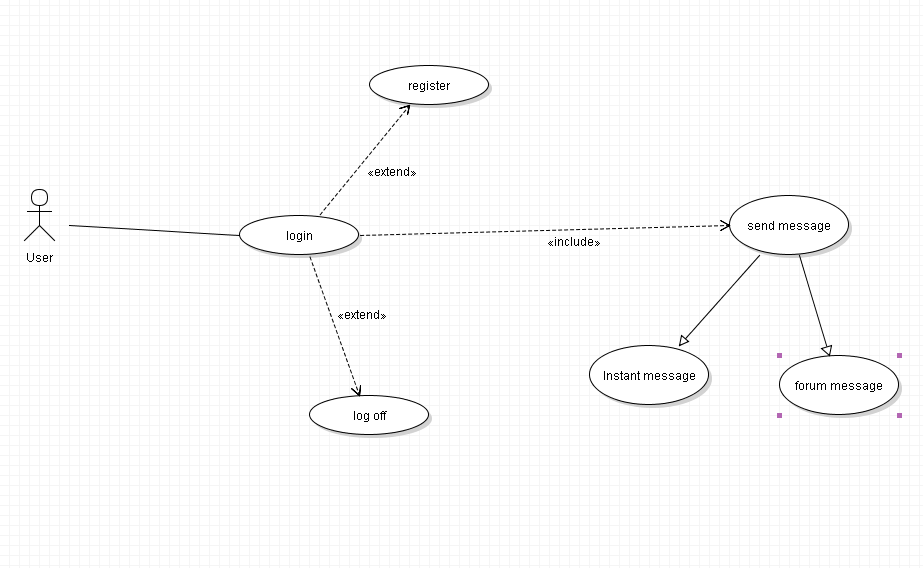


Figure 13 Use Case: User uses social media aspect

###### User creates team

Use case specification:

1: User logs in

1a: User logs off

1b: User not registered and is prompted to register

2: User makes team

3: User chooses squad

4: User chooses team name

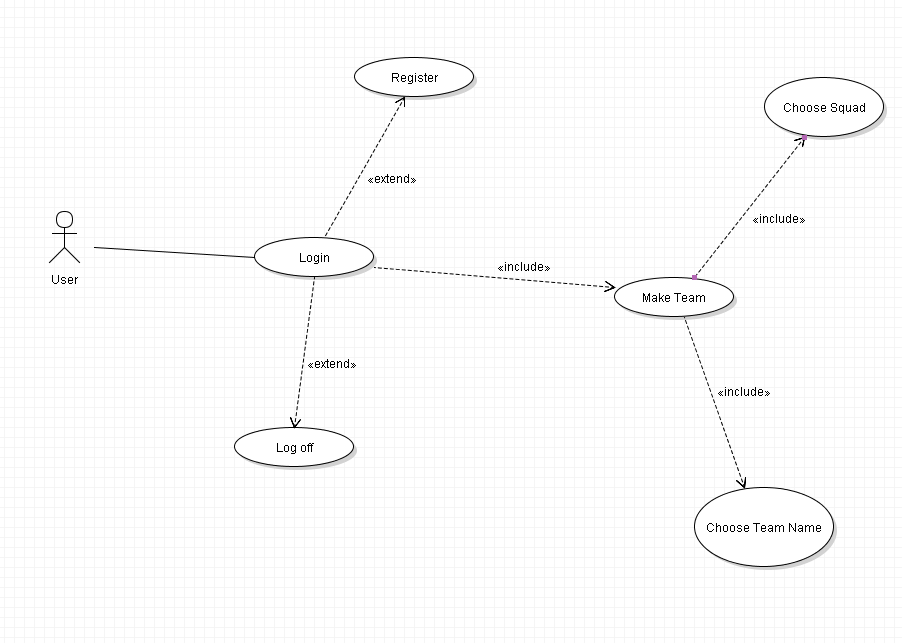


Figure 14 Use Case: User creates team

###### User makes transfer

Use case specification:

1: User logs in

1a: User logs off

1b: User not registered and is prompted to register

2: User makes transfer

2a: User has not enough funds to make transfer

2b: User selects and invalid team choice

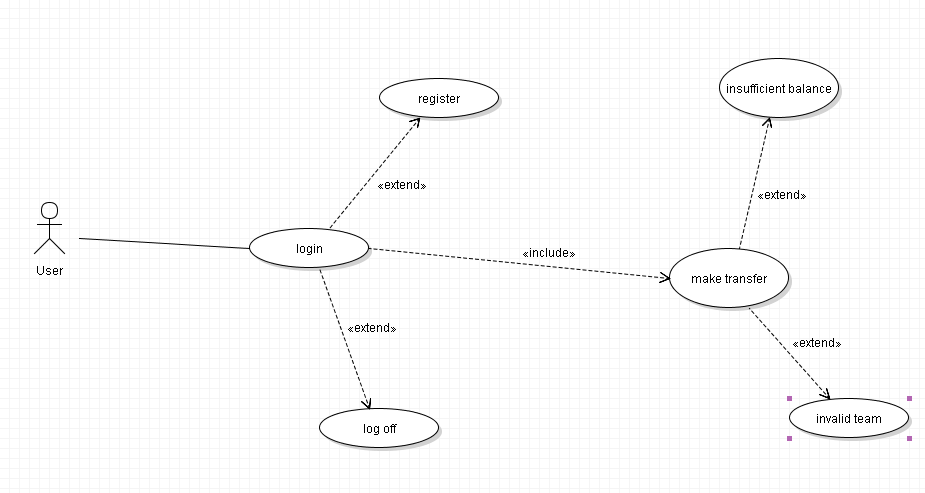


Figure 15 Use Case: User makes transfer

## Sequence diagrams

###### User checks score

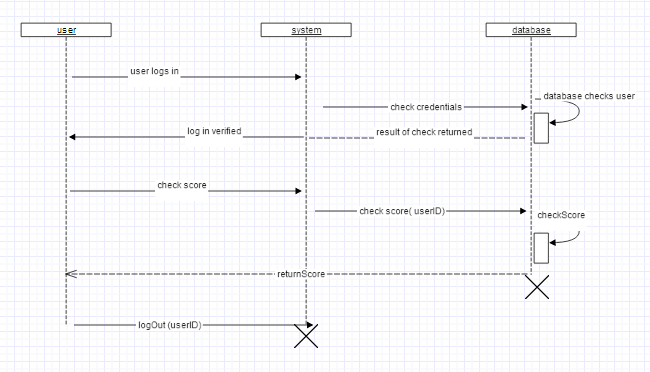


Figure 16 Sequence: User checks score

###### User checks fixtures



Figure 17 Sequence: User checks fixtures

###### User posts message

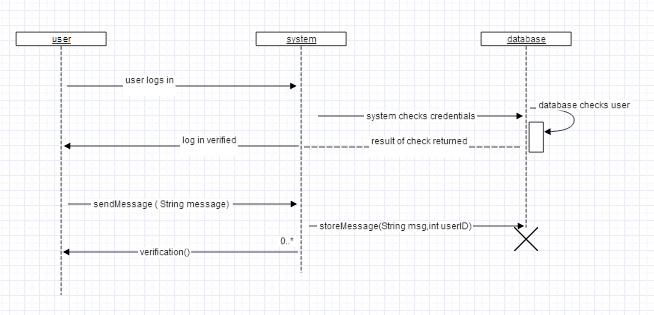


Figure 18 Sequence: User posts message

###### User chooses team

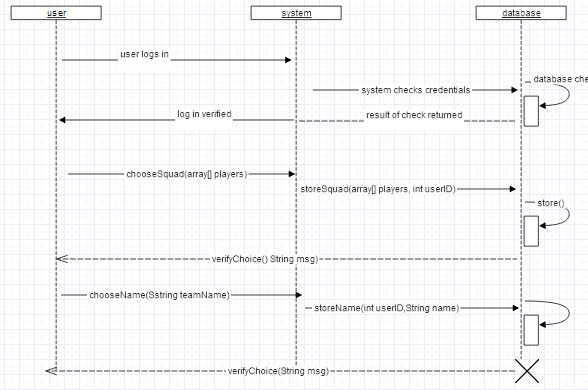


Figure 19 Sequence: User chooses team

###### User makes a transfer



Figure 20 Sequence: User makes a transfer

## User Interface Design (Wireframes)

###### Login Page

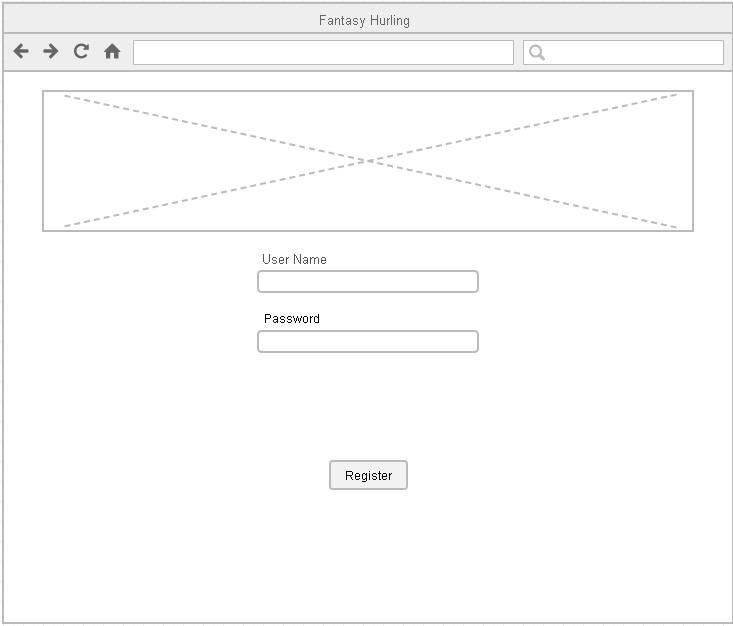


Figure 21 User Interface: Login Page

On this page the user can log into the system. They enter a username and password and are logged in. There is an image also.

Elements needed:

* 2 Text boxes
* One button
* One image

###### Registration Page



Figure 22 User Interface: Registration Page

On this page the user can register if they have not already.

Elements needed:

* 5 Text boxes
* One button
* One image

###### Team statistics page

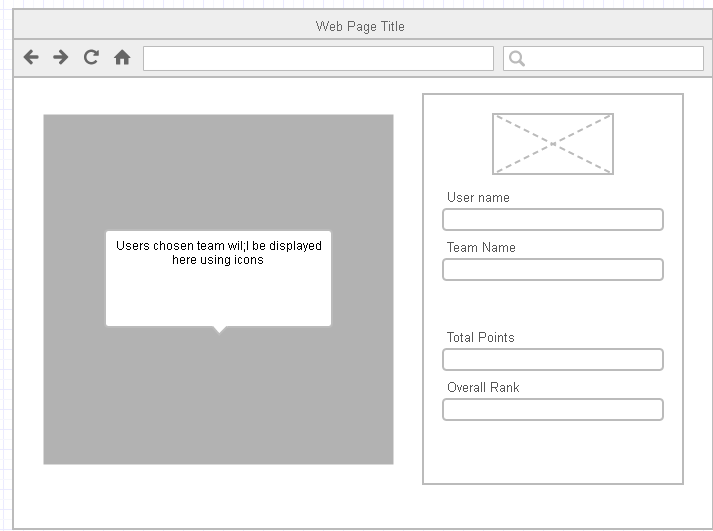


Figure 23 User Interface: Team statistics page

On this page the user can view their team’s current standing and overall points. It shows the user name and team name and the team the user has selected on the left

Elements needed:

* One image
* Team box made up of images and text
* Some basic Divs to show some information based on the user

###### Player transfer page



Figure 24 User Interface: Player transfer page

On this page the user can make transfers if they want to. They can search through the database of players, remove a player from their team, and see player score and value. Nothing is confirmed till the user clicks the add player button.

Elements needed:

* One combo box
* Basic Div. to show stats about the player
* One image to show player picture
* One button to confirm change
* A box on the left with all the players selectable and removable from the players team

###### Standings Page

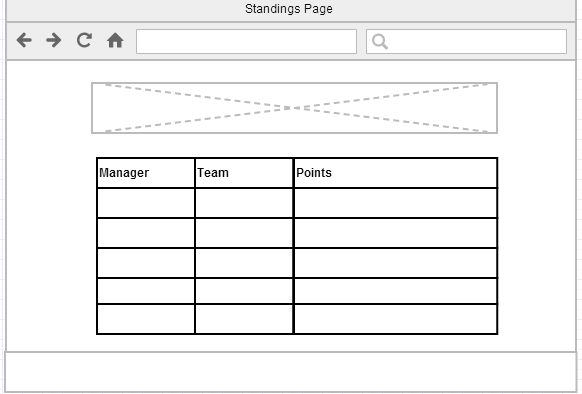


Figure 25 User Interface: Standings Page

On this page the player can view the top players in the game by points scored. He may be in this list, it will be a simple SQL query displayed in a HTML table.

Elements Needed:

* HTML table
* One image
* Header and footer ECT

###### Social Media area

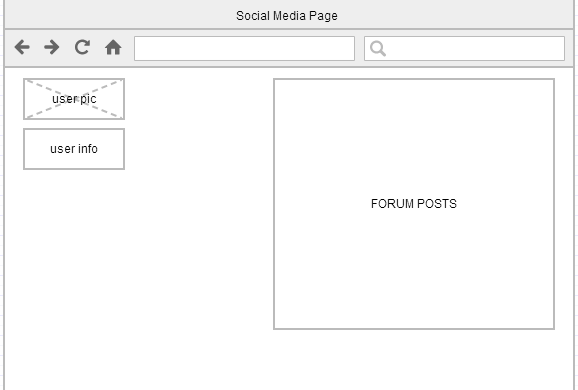


Figure 26 User Interface: Social Media area 1

On this main social media page the user can view all recent forum posts and has some information about their profile displayed also, when the user clicks a post, which will be displayed as a title with a link they are brought to a separate page with that post and all replies in it. Here the user can reply to the post also.



Figure 27 User Interface: Social Media area 2

Elements needed:

* One image
* One text area
* An area to store forum posts
* A button to post the message

## Data Store

We store Fantasy Hurling data in relational database tables. Initial prototype is design to mimic football like sport team and player statistics and it will gradually mutate into final ‘hurling’ state during prototyping iterations.

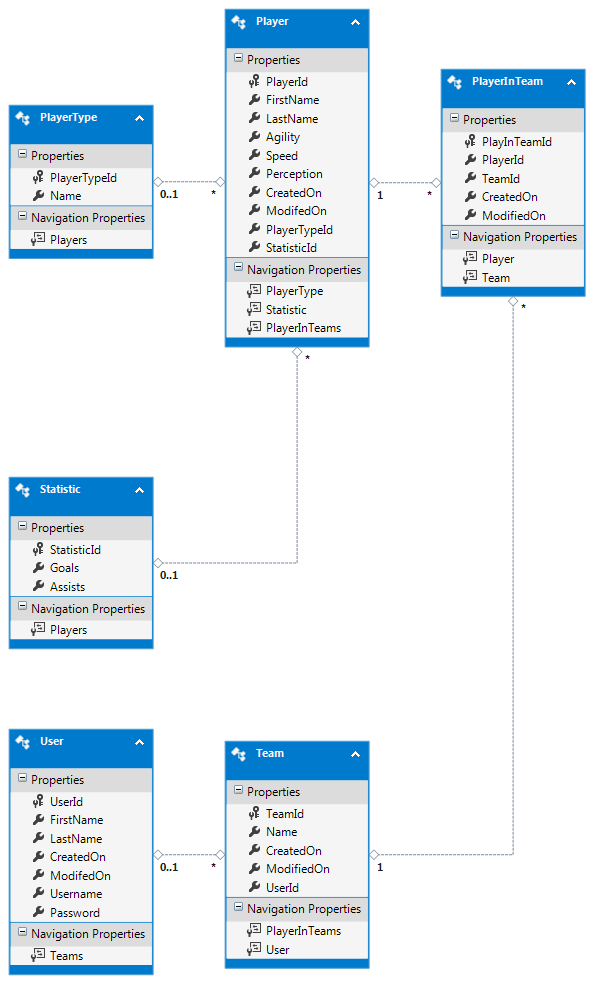


Figure 28 Fantasy Hurling database initial prototype

Database will be deployed on Azure Data Store (18)***.*** At the end of development data store gets migrated to Azure Europe North Datacenter located in Dublin (19)***.***

## Back End

Extensive research was done on web application back end development and we decided to build Web API Service (20)***.*** Initial idea was to use Node.js framework (21) to build the API, but after additional research a decision was made to use ASP.NET Web API (22)technology. Some of the reasons for this change include:

* Complexity of task versus maturity of Node.js technology
* Potential lack of documentation if a problem in code arises
* Lack of quality connectors to relational databases, Mongo DB (json based object database) is preferred with Node.js
* Additional frameworks to learn building Web API (express framework and more)
* Visual Studio 2013 Community edition free for use now
* 10 web servers free on Azure Cloud with Visual Studio installation
* Visual Studio fully implements Azure storage and web server deployment (on click in a menu and application and database is up on cloud)
* Comprehensive documentation and tutorials for all required frameworks and technologies.
* Technical support and huge community of developers if a coding problem arise.

I must add here that we expect a usage of additional tools and technologies during the project development as a need arises.

## Versioning Control System: GIT and Github

We are developing this application using Git distributed revision control system.

Fantasy hurling upstream repository is available at:

<https://github.com/Michaelcj10/Fantasy_Hurling>

Web API prototype upstream repository is available at:

https://github.com/zubidlo/HurlingApi

These two repositories are going to get merged in later state of development.

## Conclusion



# Implementation

## Introduction



## Methodology

As explained in design section, we choose to implement this application using Prototyping methodology. This approach allowed us to build a working prototype pretty early in the process. Because our application is a collaboration of many different web technologies, first step was to make sure we can actually make them work together on something remotely resembling fantasy sport game. If that’s the case we would then mutate the working prototype toward the final project objective. During the each iteration we will add a new functionality or improve the existing one. Because this is a mutli-tier web application with a pipeline where data flow from persistent storage through server toward the clients and vice versa, we must simultaneously improve each tier.

## Prototype Version 1 (November 2014)

1. A simple database with a few mock tables and simple relations between them was created in Microsoft SQL Server Compact on localhost.
2. Connection to database was assured.
3. New .Net Web API 2 with Entity Framework project was created in Visual Studio 2013 with a connection string to mock database.
4. One ORM Entity and Web API Controller with GET, POST, DELETE and UPDATE methods was created.
5. URLs to resources were crafted accordingly to a fantasy game like logic. The server will return JSON representation of requested table rows in the case of GET request or an appropriate http response for POST, DELETE, UPDATE request. 

Figure 29 URLs to resources

1. Application was tested until worked
2. Simple HTML and Javascript client which would consume the API was created.
3. JQuery Ajax function for GET, POST, UPDATE, DELETE requests was crafted, tested and refactored to acceptable form.



Figure 30 JQuery Ajax Request Method

**Prototype Version 1 Summary**

* a “working prototype” of an AJAX application consuming REST like web service resources from the persistent storage

We knew we can do this on localhost. Next step was to make sure we can deploy this on internet.

## Prototype Version 2 (December 2014)

1. SQL Server was created on Azure Cloud North Europe datacentre and the localhost mock database was migrated there.

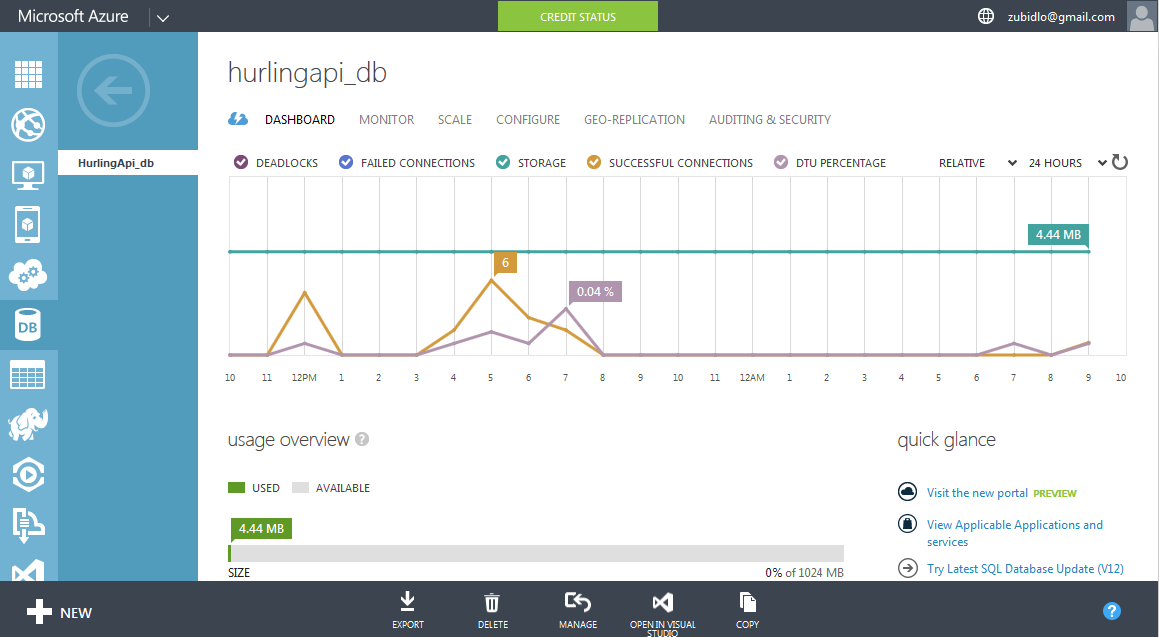


Figure 31 HulingApi SQL server Azure

1. Visual Studio Web API project was configured with a new connection string to the database on the cloud.
2. A web server was created on Azure Cloud North Europe datacentre and Visual Studio was configured to publish Web API there.

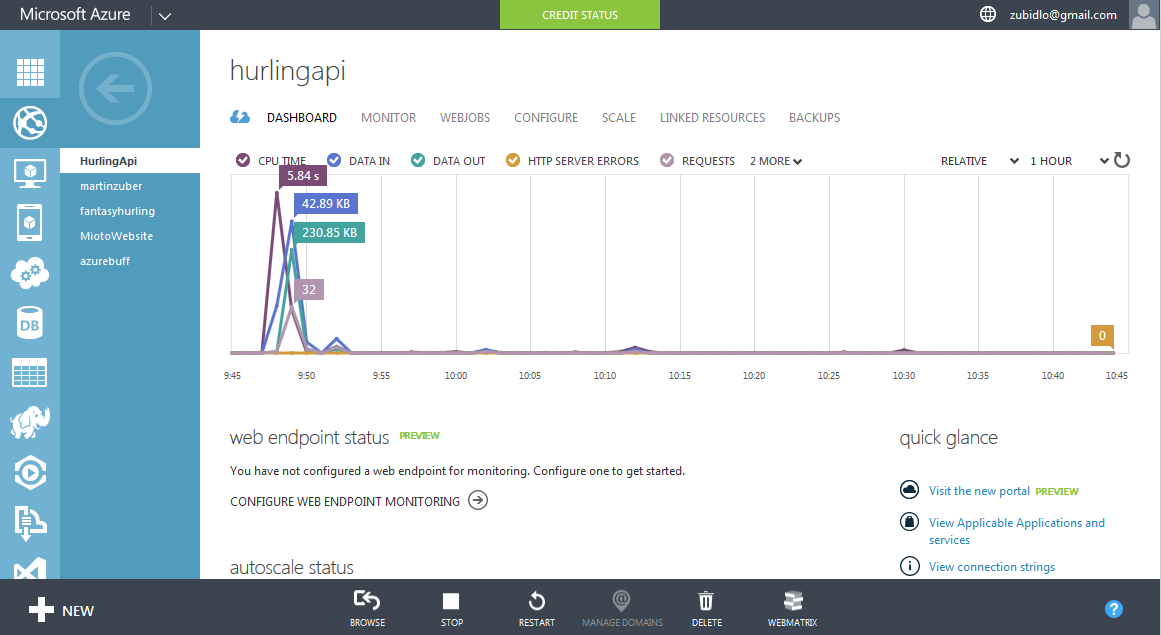


Figure 32 HurlingApi Web server Azure

1. Deployment was tested.
2. Simple web forms and tables were created for our client. At this point we were turning the client into “database admin pages”. These will serve for populating the tables of the final version of fantasy hurling database.
3. At this point we faced first serious problem. Our application turned into Cross-Origin resource sharing (CORS) application. When the client running on localhost (Domain 1) requests the resources from Web API on Azure (Domain 2) the browser will deny the request.
4. After the research on CORS in .Net Web API, the CORS were allowed for each resource route.



Figure 33 CORS enabled for api/teams route

**Prototype Version 2 Summary**

* REST like web service deployed on the web
* simple admin pages client running on localhost

It was a time to start to turn this into ‘fantasy hurling’ application

## Prototype Version 3 (December 2014)

1. With fairly good idea how ‘fantasy hurling’ should work, the database relations were crafted. It turned out there must be a many-to-many relationship between team and player entities, which added additional complexity to the project. But overall we design the database schema well and we never needed to change it in the future except for some cosmetic changes like renaming or adding the columns. So at this point we had final version of database deployed on Azure.

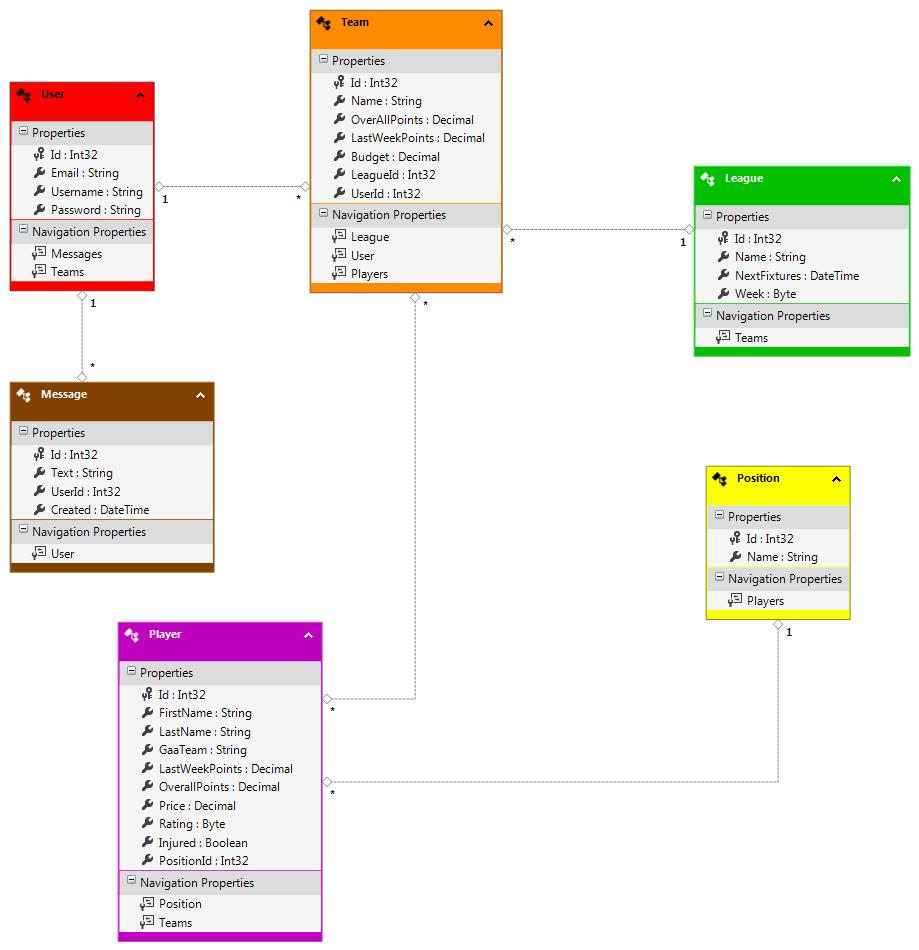


Figure 34 Fantasy Hurling Database Schema

1. Now the development process turned into the circles of
   1. Populate a database table with few rows of data.
   2. Mapping the table columns to plain old c# object POCO object with Entity Framework which is an Object-Relational Mapper (ORM)
   3. Writing the controller which would route the requests to POJO object or collection of POJO objects and return custom crafted http response with JSON representation of POJO object in the http message body.
   4. Creating client ‘data admin page’ for each database table.

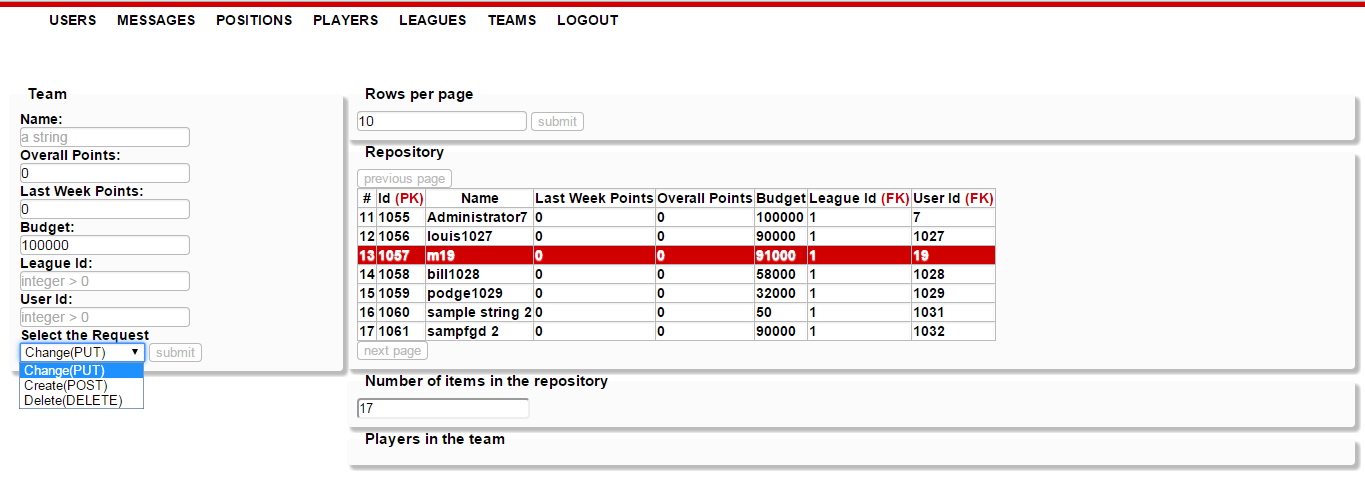


Figure 35 Admin Page for Teams

1. There is 6 tables in our database so I needed to apply steps mentioned above for each and with each new iteration of the process I learned how to do it more efficient and code it more nicely. I faced a number of challenges during this process mainly because each step uses different languages and tools.
   1. SQL insert queries
   2. Entity Framework with C# with design patterns such as Abstract Factory, Repository. Visual Studio 2013
   3. C# Web API 2 Framework with C#, LINQ C# extension, Visual Studio 2013
   4. HTML5 , CSS3, Javascript, JQuery, HTTP protocol
2. It became apparent that I will need to implement some kind of ‘business logic’ to the controllers in later prototype. For example if an administrator using admin pages client would like to delete an player which is registered in one or more teams the Entity Framework will throw exception with 3 page long stack trace. This exception will be then inserted into http response body and send back to client. So I will need to check for all such a cases and create custom http responses like *“you cannot delete this player because he is registered with one or more teams”*
3. Other issue was ‘cyclic entity references’ when Entity framework will parse never ending data to JSON. For example a team has references to players and those have references to teams they play for and those have references to same players again and so on. This issue was handled by one more layer of object mapping where POCO entities are mapped to much simpler Data Transfer Objects (DTOs) and those are parsed to JSON and send to client.

**Prototype Version 3 Summary**

* Database in final form deployed on the web.
* Web API working prototype deployed on the web.
* Data admin pages prototype on localhost.

## Prototype Version 4 (January 2015)

1. Custom http action results were created need for business logic.



Figure 36 ConflictActionResult

1. These results will be returned by controllers instead of internal server error 500 with a stack trace.



Figure 37 Conflict Result in code and action

1. Controller methods were turn into Asynchronous Tasks. This approach will not block the server thread.



Figure 38 asynchronous method example

1. Web API 2 framework supports Open Data Protocol (OData). A controller methods can return OData query able collections. OData is powerful and well-crafted protocol which adds a lot of functionality for the web service. For example sorting, grouping, conditions, server side paging and much more.



Figure 39 HurlingApi OData support

1. Database admin pages client was finished with highlight-able, click-able and page-able tables. Simple login dialog was created. Admin pages were deployed on new Azure web server.



Figure 40 fantasyhurling Web Server dashboard Azure

**Prototype Version 4 Summary**

* Database in final form deployed on the web.
* Asynchronous Web API with OData support prototype deployed on the web.
* Database Admin Pages in final form deployed on the web.

## Prototype Version 5 (January 2015)

1. With Database Admin pages deployed we started to populate database tables.
2. WebAPI business logic was finished and testing and debugging process on running service started. At this point we had fairly well working Web service, which could be considered as final product when properly debugged. There are some design decision which should be reconsidered: I decided to tread following as additional functionalities and will be implemented only if time allows.
   1. The business logic should be pushed one level lower to the repository dbContext. That way the controllers would be decoupled and could be tested more easily with help of Dependency Injection pattern.
   2. Instead of manually mapping relation Entities to DTOs, an auto-mapper tool could be used.
   3. There is a question of authentication. Web API 2 supports various methods over HTTP and HTTPS, OAuth2 protocol included. After some initial research I decided to leave the Web service without the authentication, because it would introduce the complexity we are as a team not equipped to handle just yet. The web page clients will have their own authentication mechanism implemented using HTLM5 session storage.
3. Web API have a default home page client with automatically generated help page. I decided to add ‘API Test’ client to it, which can be imported from NuGet package manager in Visual Studio 2013. This client will allow the front end developers to see how craft the requests toward the API.

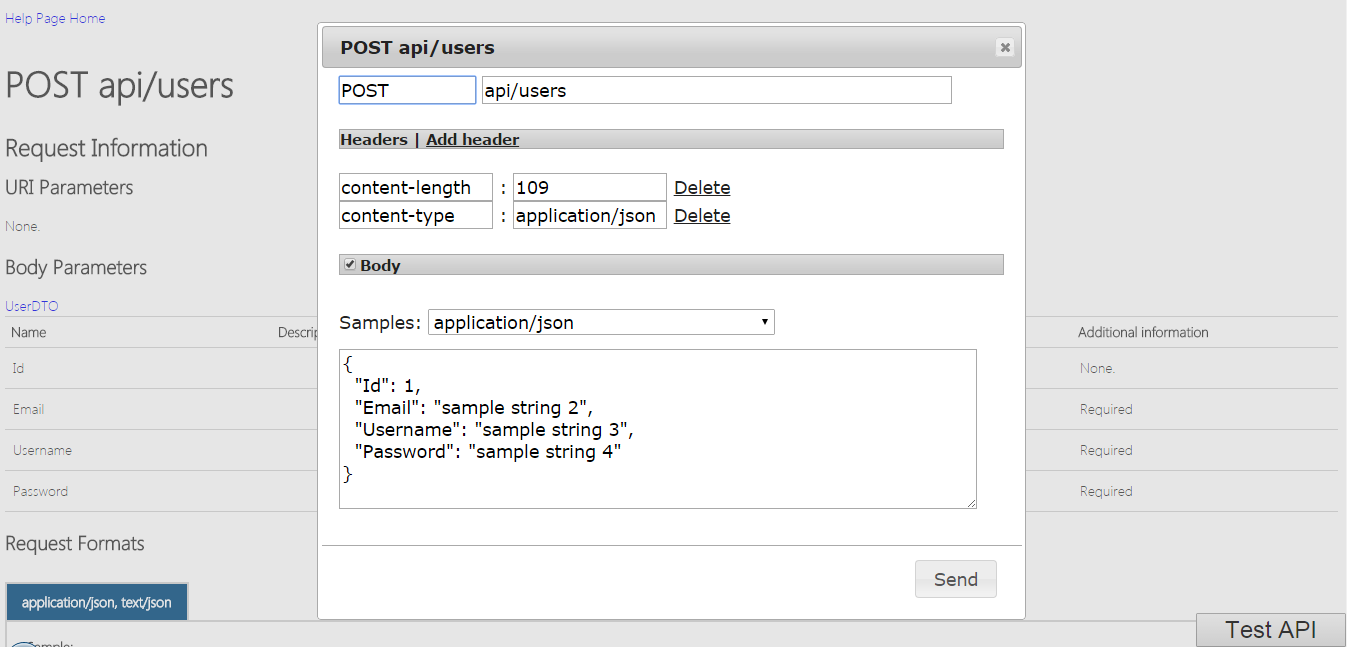


Figure 41 Test API client in action

**Prototype Version 5 Summary**

* Database in final form deployed on the web.
* Web API in final form deployed on the web:

<http://hurlingapi.azurewebsites.net/>

* Database Admin Pages in final form deployed on the web:

<http://fantasyhurling.azurewebsites.net/consume_api_examples/data_admin/login.html>

At this point we are ready for development of Fantasy Hurling Client

## Front end (Fantasy Hurling Web Page)

There were some key elements to creating the front end. A series of simple but cleanly designed web pages were built. JQuery UI widgets were used, and JavaScript was used to connect the front end to the back end database. This was done using an API that was built by one of the team members in C#. We won’t go into this now as it will be explained in detail later. Navigation on the site was made as simple as possible and there were help icons on each page to explain how the user could interact with the system. We will look at the site in detail now and fragments of code will be shown and explained.

###### Login and Register

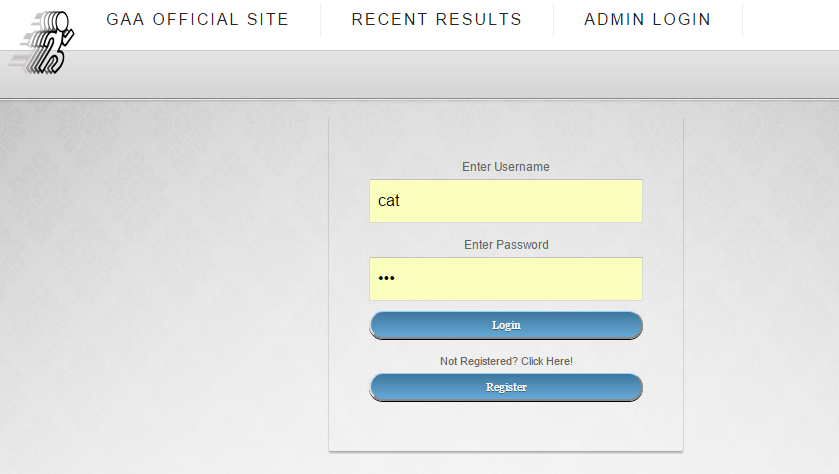


Figure 42 Login Dialog

The user needed a password and username to login and interact with their team. The login page was kept minimal and straightforward to understand. The following is an example of the method that would log the user in if that username and password was correct. A URL was specified with the user ID the user had entered in the text field as their own and an AJAX request retrieved that users details if he existed. If he did exist, it retrieved the password associated with that account and checked it against the password the user entered, if it matched, the user was logged in, if not, a message prompted the user that he needed to re-enter the details.



Figure 43 get user by username request

If the user did not have a login, he could register to create one. He had to enter a username, password and a valid email address. The API would not allow duplicate usernames so long as it did not already exist, the account was created. A simple function was needed to add a new user to the account. The information to put into the database via the AJAX POST request was retrieved and stored in an array. This information was what the user had entered In the HTML fields.

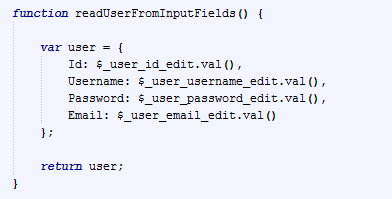


Figure 44 register Form

The function to add the user into the database via the API was a simple POST request using the JSON data.



Figure 45 add a user request

###### Session Storage

With local storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Local storage is more secure, and large amounts of data can be stored locally, without affecting website performance. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server. Local storage is per domain. All pages, from one domain, can store and access the same data.

Session storage was used largely in this application. The bonus to the session storage was that it got rid of the need to pass variables between pages in the style of PHP which made accessing variables such as the username very easy.

Example of setting is below, we set our current user ID to be called “id” in session storage. We could not access this any time we wanted across the entire site with a very simply command.

Set session storage

sessionStorage.setItem ("id", id);

Get session storage value

Var user = sessionStorage.getItem ("id");

Session storage could be cleared at the end of the session using a simple command.

sessionStorage.clear ();

The user’s players in his team were stored in this manner also which allowed them being interacted with very easily. For example when the players in the users team were being retrieved each player was assigned a session variable of the data retrieved for player position, so the first player out was called “1” as his player position is 1. The value for this name of 1, was his player ID value. The entire team could be stored this way in session storage and then accessed at any time.

sessionStorage.setItem ("" + object.PositionId, "" + object.Id);

###### Creating a new Team for the user

When the user navigates to either the view team page or the transfer’s page, the application checks if a team exists with that user ID. If the team exists then nothing extra happens and the application acts as normal. If it does not exist, a new team is created for the user. The name is simply the username and the user ID combined. The type of request to do this is a POST request. The team ID is then stored in session storage.



Figure 46 insert a new team request

###### How to view the team

The user could view his team using the team view page. In here there was a graphical representation of his team using icons which were movable. Each time a user hovered over a player a baseball style card would pop up showing details about that player.

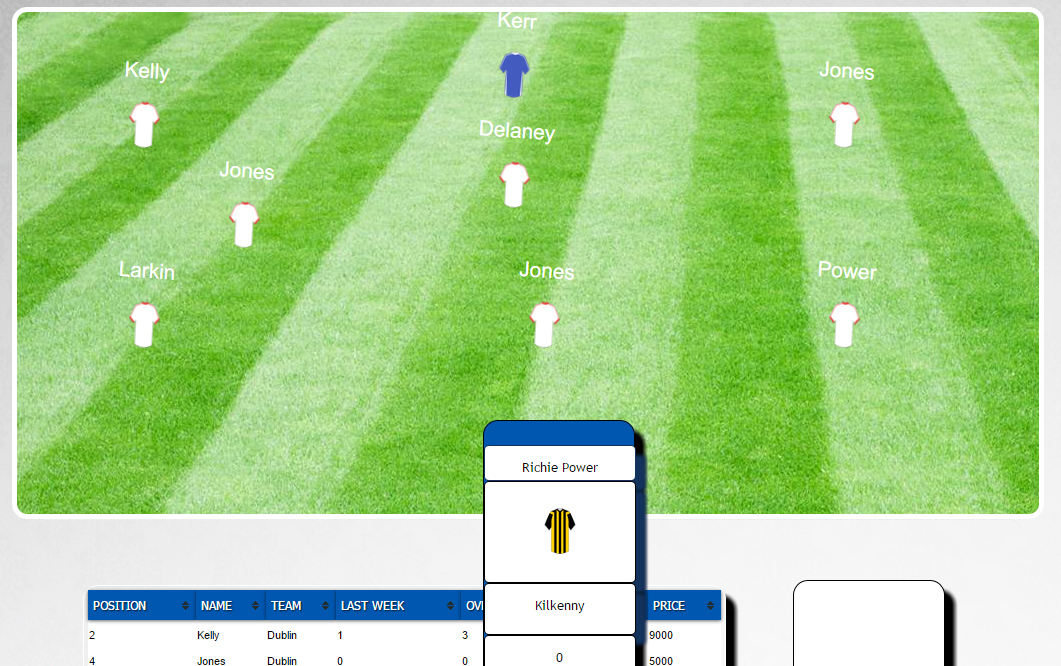


Figure 47 Player details

More detailed stats about the team were also shown in the form of a sortable table and some widgets showing information such as the top scorer or most expensive player.

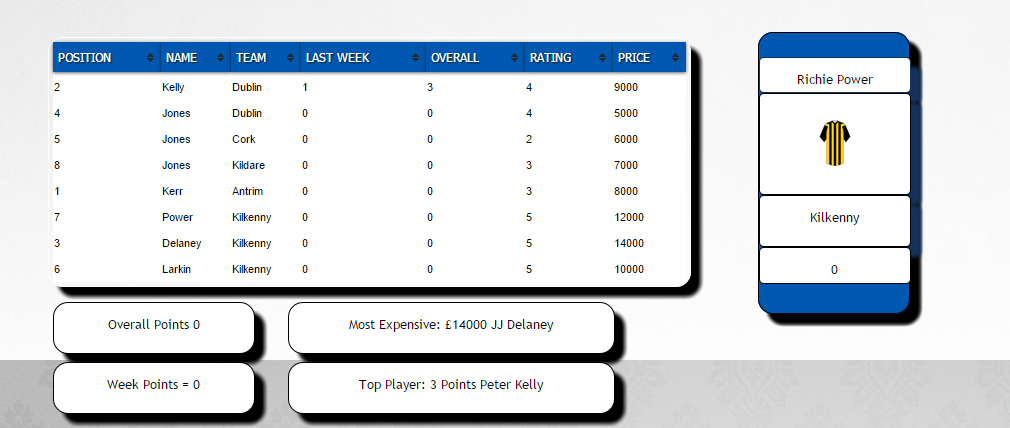


Figure 48 Team details

All this information was retrieved using the API and a simple GET request using the appropriate URL. Below is an example of how the full team is retrieved. As can be seen simple local variables were used to decide which the most valuable player was and so on, each iteration the player was checked against the current most expensive player, if he was worth more, he was set as the most expensive. When the request was finished, we then had the most expensive player, top player and so on in variables which were used to update the inner HTML of the DIV elements in the HTML page.



Figure 49 Team by user id request

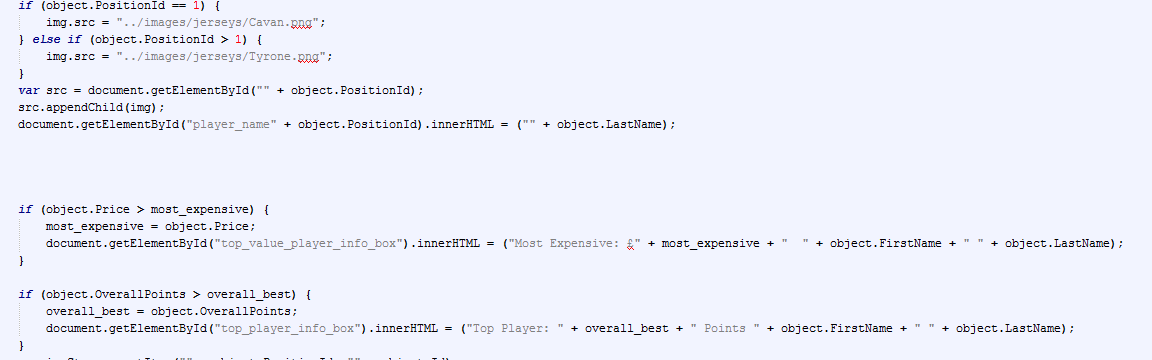


Figure 50 Team Table implementation

###### Viewing the standings

The user could view his overall position and the top 10 teams in the game in the standings page.

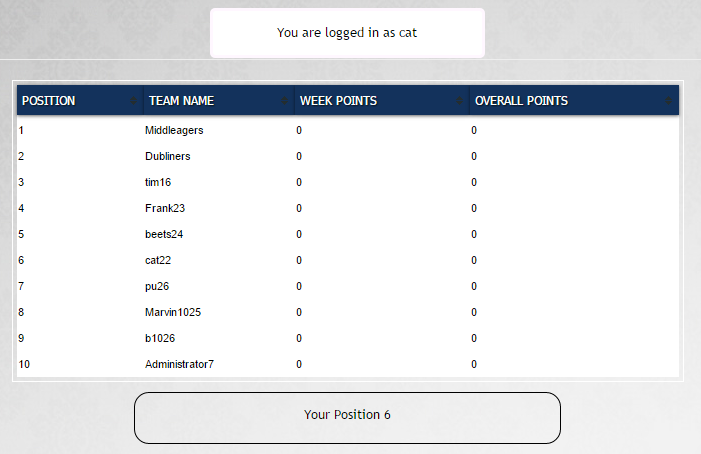


Figure 51 User Standings

The simple GET request returned the top 10 users based on overall points. A second request was done to retrieve all users, a counter variable was incremented each time a user was found, when the user found matched the current user ID stored in session storage, that counter contained the current users overall position out of all users. Below is an example of the function to retrieve the top 10 users in the database sorted by the highest value for the user’s points.



Figure 52 Users sorted by overall points request

###### Changing user details

A page was created to allow the user change his login details if they wished. The request was made in the usual way, a PUT request was used this time however.

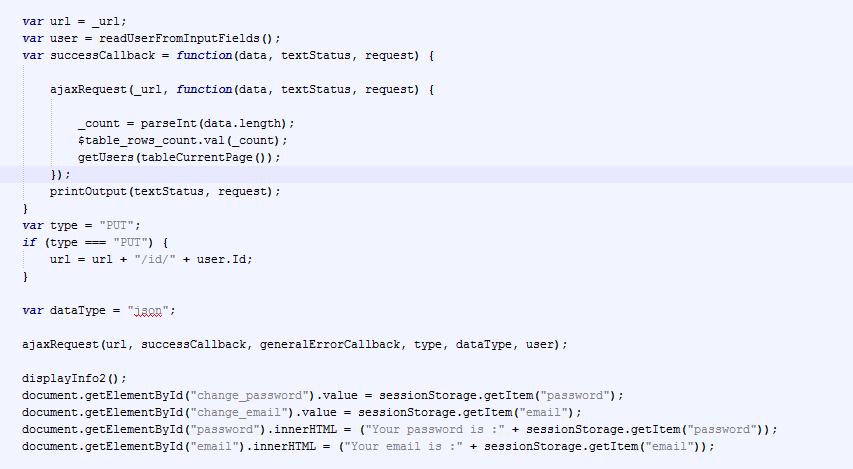


Figure 53 Update user request

The page was kept simple with HTML DIV elements showing the current users details and text fields allowing the user to enter new details. On click of the button the request created a new JSON array of data, and PUT this data into the user’s details via an AJAX request. Below is an example of the HTML styling in the displayed DIV containing the current details and the fields and button required to change these details.

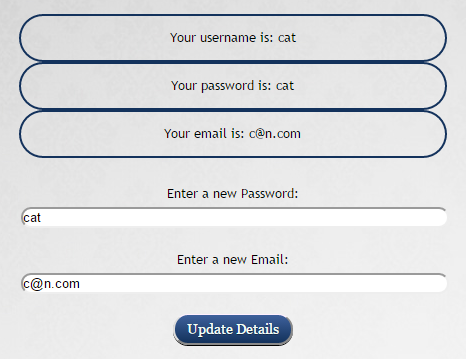


Figure 54 Update User Dialog

###### Instant messaging in the application

Social media was integrated into the application in the form of instant messaging on each page. Every functional page on the website has a small neat DIV containing the last 3 posted messages and a text area and button to allow the user post a new message.

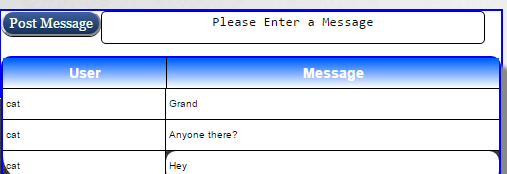


Figure 55 Chat Board

The information is retrieved using a GET request in an AJAX function. Only the 3 newest messages are retrieved keeping the feeling of the application being in real time, and keeping the table itself small and efficient. An example of the code can be seen below.



Figure 56 Messages sorted by creation time request

###### Making a transfer

The transfer’s page is the most complex. On initially entering the page a function loads the user’s players. It knows what players to load based off the team ID which was set to session storage earlier. These players are stored in JQuery UI widgets which are resizable and moveable. An example of this can be seen below.



Figure 57 making a transfer

###### Deleting a player from the team

To delete a player, the user clicks on his icon. On click fires a function with the parameter of player position given. We now have the player’s position. We then run a new function iterating through the users team looking for the player who has a position equal to what we passed through originally. When we find it, we take his player ID and pass it to another function. This function removes the player with the ID we pass through from the team with the ID we have stored in session storage! Easy!

We can see below an example of the function to delete a player. It is of course another AJAX request and is of type DELETE. Once the delete function finishes we reload the page after a delay of 600 milliseconds which will reload the entire team again, which of course no longer includes the player we deleted.



Figure 58 remove a player from a team request

###### Adding a player to the team

To add a player to the team, a user simply uses the table on the right to make searches based on team, when he finds a player he wishes to add, he simply clicks the button to add the player.

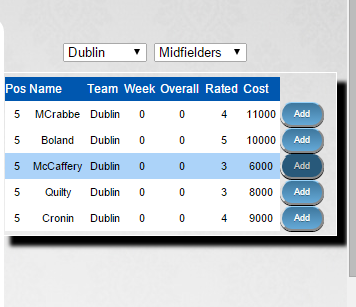


Figure 59 Adding Player

On click of the add button, a function is fired that retrieves the player ID. Another function is run then to actually add the player. It is of type PUT. If the user has enough budget and the position is vacant the player is added to the team.



Figure 60 add a player to a team request

###### Dealing with if a user has not enough players in his team

One issue that became apparent early on is that if a user creates his team and adds some players, he may not add in the full complement of 8. This should not be allowed. We came up with a simple solution, when the user tries to do anything on the site a check is done, essentially an AJAX request retrieves all the players from the user team and counts them with a simple local variable. If there is not 8 players on the team, then he is redirected to the transfer’s page and he must add into his team additional players.

###### Creating the user team

When the user registers, he now has a username and password. At this stage he has no team created in the database. There were a number of solutions to this, but the simplest was to simply do a check when the user tried to view his team or make a transfer if he actually had a team. The user ID is passed into the URL that retrieves all teams, and if the AJAX request retrieves nothing, the user must have no team and one is automatically created for him. It is given a name derived from his username and user ID. We can see below the code to input the new team and the fields it requires in the function below that again.

## 

Figure 61 insert a new team request

###### Keeping track of the session

Another issue was making sure that if the session was ended, that if the user tried to navigate back to that page, he would be brought back to the login page. Also the user should not ever be allowed simply type in for instance the URL for the transfer’s page directly into the browser and be brought there without logging in. A simple way to do this was to use the session storage variables. The first thing that happened on all the pages was a check was done to see if there was a user ID in session storage, if there was not, then the user was re-directed to the login page. This avoided any potential glitches. We can see below a user tries to navigate to the home page, if username is not there then he is brought to the login page again automatically.

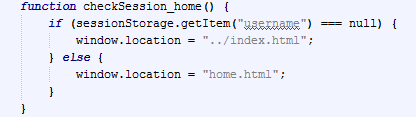


Figure 62 session storage 2

## Conclusion



# Testing and Evaluation

## Introduction david

## User Interface Tests michael

## Unit Tests ) Martin

## Integration Tests ) Martin

## Give it to some user tests david and michael

## Conclusion david

# Conclusions and Further Work

## Introduction michael

## Achievements david

## Personal gain (shit learned) all of us should write a bit

## Further work ( possible improvements ) Michael

## Conclusion david

### Project Proposal

### Project Diary

### Code Listing

### Bibliography

1. **Technologies, Market Share Statistics for Internet.** Browser market share 2014. [Online] 2014. [Cited: 11 30, 2014.] http://www.netmarketshare.com/browser-market-share.aspx?qprid=0&qptimeframe=Y&qpcustomb=&qpcustomd=2.

2. **Jonson, Phil.** Web developers less concerned about browser-compatibility, more concerned with HTML5. [Online] 2014. [Cited: 11 25, 2014.] http://www.itworld.com/article/2693441/big-data/web-developers-less-concerned-about-browser-compatibility--more-concerned-with-html5.html.

3. **Gomez.** Thriving in a multi-browser world. [Online] 2010. [Cited: 11 21, 2014.] http://www.cio.com.au/campaign/370042?content=%2Fwhitepaper%2F370263%2Fthriving-in-a-multi-browser-world-ensuring-optimal-website-performance-for-all-end-users%2F%3Ftype%3Dother%26arg%3D0%26location%3Dfeatured\_list.

4. **keynote.com.** The Benefits of Third-Party Content Monitoring. [Online] 2014. [Cited: 11 23, 2014.] http://www.keynote.com/resources/white-papers/benefits-of-3rd-party-content-monitoring.

5. **Kartik Bajaj, Karthik Pattabiraman, Ali Mesbah.** Mining Questions Asked by Web Developers. [Online] 2013. [Cited: 11 15, 2014.] http://salt.ece.ubc.ca/publications/docs/kartik-msr14.pdf.

6. **devrates.com.** devrates projects list. [Online] 2013. [Cited: January 1, 2013.] http://devrates.com/project/list.

7. **Podila.** HTTP: The Protocol Every Web Developer Must Know. [Online] 2013. [Cited: April 2, 2015.] http://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177.

8. **R., Fielding.** Hypertext Transfer Protocol. [Online] September 1, 2004. [Cited: April 4, 2015.] http://www.w3.org/Protocols/rfc2616/rfc2616.

9. **Berners-Lee, T.** Uniform Resource Identifiers (URI): Generic Syntax. [Online] August 1998. [Cited: April 5, 2015.] Uniform Resource Identifiers (URI): Generic Syntax.

10. **Model, Prototype.** What is Prototype model- advantages, disadvantages and when to use it? *ISTQB EXAM CERTIFICATION.* [Online] [Cited: December 15, 2014.] http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it.

11. **W3C.** HTML5. *A vocabulary and associated APIs for HTML and XHTML.* [Online] October 28, 2014. [Cited: December 14, 2014.] http://http://www.w3.org/TR/html5.

12. —. Introduction to CSS3. *W3C Working Draft, 23 May 2001.* [Online] May 23, 2001. [Cited: December 14, 2014.] http://www.w3.org/TR/2001/WD-css3-roadmap-20010523.

13. **Mozilla.** Ajax. [Online] 2005. [Cited: December 24, 2014.] https://developer.mozilla.org/en/docs/AJAX.

14. **jQuery Foundation.** jQuery. [Online] [Cited: December 23, 2014.] http://jquery.com.

15. —. jQuery User Interface. [Online] [Cited: December 24, 2014.] http://jqueryui.com/.

16. **Sublime Text.** Sublime Text. [Online] [Cited: December 25, 2014.] http://www.sublimetext.com.

17. **Appcelerator, Inc.** Aptana. [Online] [Cited: December 14, 2014.] http://www.aptana.com.

18. **Microsoft.** Storage documentation. [Online] [Cited: December 15, 2014.] http://azure.microsoft.com/en-us/documentation/services/storage.

19. —. Azure Regions. [Online] [Cited: December 14, 2014.] http://azure.microsoft.com/en-us/regions.

20. **Anonymous.** Web API. [Online] [Cited: December 15, 2014.] http://en.wikipedia.org/wiki/Web\_API.

21. **Node.js. [Online] [Cited: December 20, 2014.] http://nodejs.org.**

**22. Microsoft. ASP.NET. [Online] [Cited: December 15, 2014.] http://www.asp.net/web-api.**

**23. monetate.com. Ecommerce Quarterly EQ2 2014. [Online] [Cited: April 1, 2014.] http://www.monetate.com/research/#ufh-i-25962265-ecommerce-quarterly-eq2-2014.**