CI301

Interim Planning and Research Report

Name:	Charlie Grinsted
Student Number:	11832016
Project Title:	A web app for managing sports teams
Project Supervisor:	Dr. Gulden Uchyigit
Second Reader:	Prof. John Howse

Contents

Project Scope

Proposed Project

The aim of this project is to create a web app to address organisational issues that arise for small-scale sports clubs. These issues stem from both my own personal experience and that of my peers in arranging both friendly and competitive sporting events. The key issues surmised prior to research were a lack of centralised and detailed communication between players regarding practical and financial arrangements.

Ultimately, the project is intended to be a demonstration of full-stack development with the end product as a summation of the skills I have developed during the degree course. This process runs from initial idea conception through research, design, client-side development, server-side development and testing.

While the feature set is subject to change, basic functionality of the app will allow users to create an account, join their team(s) and respond to match invitations. Before further developing the project scope, research was conducted to ascertain the required and/or desired features of said application.

The project calls for development in both front-end user interactivity and a back-end accessing a database. As the basic premise will require the creation of CRUD (Create, Read, Update, Delete) functionality with an API (application programming interface) endpoint, I have considered using an MVC (Model, View, Controller) framework, either Ruby on Rails or a Node.js. I shall choose whichever is more appropriate when my research has been furthered. Other aspects of this project will explore the user experience along with front-end development in HTML/CSS & Javascript.

Stakeholders

The project stakeholders are primarily myself and Gulden Uchyigit, my project supervisor. Additionally, the project will be considered by a second reader at key stages. This project is independent of any client, meaning there are no financial or managerial influences. Prospective users are notable stakeholders - those in the target audience which I shall be aiming the final product towards. Those with whom I conduct research will be involved with the project, but not all participants will directly hold a vested interest in the product; only those who became users would be beneficiaries of the research outcome, and therefore project stakeholders.

Communication with Gulden will be conducted via email and through face-to-face meetings. The latter have been provisionally scheduled into my Project Timeline (see **Appendix A**) as approximate times - these could potentially be deviated by the unavailability of either party.

Methodology

My own personal experience with web development has included several development methodologies, notably the two most common, Waterfall and Agile. When choosing a methodology, there are several variables to consider, including the available resources, timescale and stakeholder involvement. The timescale is set by the assessment deadlines but the other two variables are flexible. As mentioned, during project development the stakeholders are myself and the project supervisor, with resources being skill and time, since I will be assuming the roles of strategist, designer, developer and tester throughout the process.

Agile & Scrum

Agile methodology, to give a brief overview, is an iterative methodology which involves rapid development towards short-term goals, with an emphasis on communication and feedback.

"It means regularly releasing small pieces of functionality; constant communication between team members, and using equipment that displays progress being made by your team" [1]

In terms of communication, traditional Agile methodology is not expressly relevant; there will not be a large amount of communication since I am the only active contributor to the project. However in terms of setting short-term 'sprint' goals, Agile should prove to be helpful and motivational not only for meeting targets, but also assessing the product as the project progresses.

The concept of sprint or scrum methodology is harder but not impossible to apply to solo work. This idea is explored by Dustin Wax in his article 'Scrum for One':

"Give yourself a time limit and set a reasonable goal – reasonable, but meaningful – to reach by the end of that period…

…Sprinting the way Scrum teams do it won't really work for individuals – you probably have a lot of different roles to play on a day-to-day basis, which means focusing on a single project to the exclusion of everything else is going to be difficult, if its even possible." (Wax, 2009)[2]

In short, the quick and iterative process remains intact, but relies on constant self-reflection in place of the team communication.

Measuring Success

Although there is no budget to speak of, success can be measured by adherence to the schedule and scope of the project. Scope-creep will be difficult to manage since justifiable extra features may emerge or develop after development has begun. As can be seen in the Project Timeline (**Appendix A**), sections haven't been sub-divided into particular features; the banner headings of "client-side development" and "server-side development" will encompass many smaller tasks. In order to successfully manage this, I will set weekly targets based on incremental changes to the project codebase.

Success could also be gauged by my own personal skill development, although this may prove hard to qualify.

Finally, to measure the success of the project itself, a test-driven development process will be adopted. In line with the Agile methodology, each iteration of the product will be tested after a sprint. A successful product will be one which adheres to the initial scope and functions as expected.

[1] *Gov.uk, (2014). Agile - Government Service Design Manual.*

[2] *Wax, D. (2014). Scrum for One.*

Specification

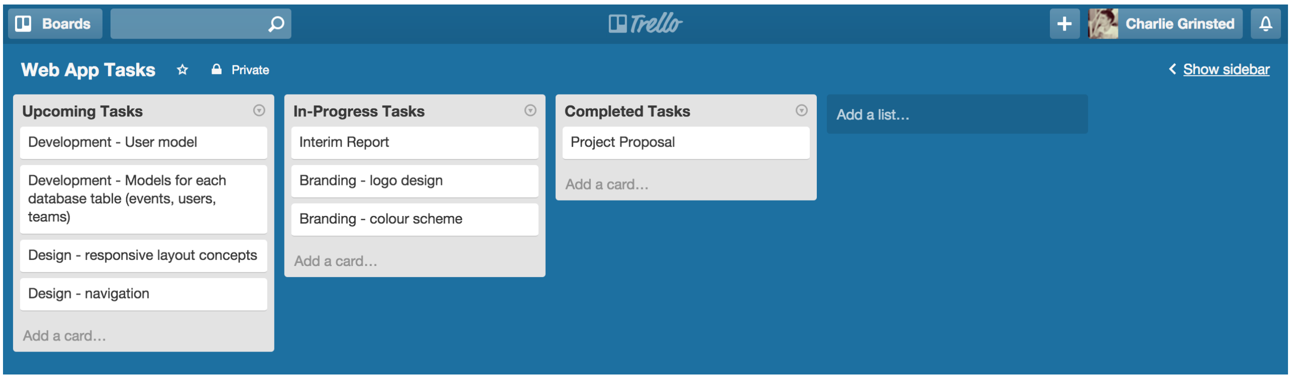
Project Timeline

In order to ensure that the project stays on track, I have drawn up a Gantt chart including key tasks and their deadlines, covering the 32 weeks available from the start of October until the final deadline in May. The Project Timeline is available as **Appendix A**.

Deliverables

The nature of this project is such that the order of the deliverables is not of critical importance. While the work has been divided into the relevant sub-sections (see Project Timeline in **Appendix A**), the order in which they are completed will not ultimately alter the rate of progress. Naturally, there are some exceptions which require completion prior to others commencing, most notably the creation of a front-end style guide prior to front-end template development. Additionally, front-end development cannot commence before a server-side API exists to test against.

As explained previously, my use of Agile and Scrum methodology will result in short-term goals and regular self-reflection. Accordingly, intermediate deliverables will be based on small targets set throughout the process. Ultimately, this will materialise into tasks such as "Finish login system" or "Complete user profile page" being set and reviewed on a weekly basis. In order to effectively manage this, I will use Trello[3], (pictured below) to keep track of completed, in-progress and upcoming tasks, adding new ones as they emerge also.



Upcoming Tasks

- Development - User model
- Development - Models for each database table (events, users, teams)
- Design - responsive layout concepts
- Design - navigation

In-Progress Tasks

- Interim Report
- Branding - logo design
- Branding - colour scheme

Completed Tasks

- Project Proposal

[3] *Trello.com, (2014). Trello.*

Project Description

As outlined in the project proposal, the solution will be a web app using MVC (Model, View, Controller) software pattern. Software planning is a new experience for me, but I became aware from my peers on other Computing courses that one valuable method is UML (Unified Modelling Language). Accordingly, I have produced Class Diagrams (**Appendix C**), as further explained below.

Data Modelling

Understanding the required values and functions for the web app is paramount to building a successful product. Though these are subject to change, the software has been planned in the form of Class Diagrams (**Appendix C**). In order to complete this in a logical and readable fashion, I first simply wrote out the desired functions for each.

A user needs to be able to:

- Create an account
- Update their personal details
- Join or create a team
- Administrate teams which they have created:
  - Create / update / delete events
  - Add / remove users from the team
  - Post updates and other information for the team
  - Pick players for events
- Delete their account
- Be notified when requested to play

The theory is that every player/member from the associated team(s) will be added as potential players. The app will then display these players to the administrator. The players will have a global status (such as "Away on holiday for 2 weeks" with a timestamp) that will allow the administrator to see if the player is available. The player will also have a per-event-status showing their current commitment to the event (not selected, selected but unpaid, paid).

Initially, the system will require three data models, with variables as follows:

Users	**Teams**	**Events**
Unique ID	Team ID	Event ID
Full Name	Administrator(s)	Teams
Email Address	Players	Time/Date
Encrypted Password	Events	Attendees
Teams administered		
Teams joined		

This is, of course, subject to change as the project develops.

Technology Choice

Knowing that the project requires an MVC architecture is one thing, but making a choice of a framework from the abundance of available offerings presents a challenge in itself. As mentioned in my project proposal, choosing the right one is paramount to the success of the end product.

In September 2009, Julia Plekhanova of Temple University wrote an evaluation of web development frameworks[4]. Technology has advanced in the last 5 years, but the three platforms which she assessed are still prevalent in 2014. Django (Python-based)[5], Ruby on Rails (Ruby-based)[6] and CakePHP (PHP-based)[7] all offer an MVC development framework in their respective languages.

I have gained very brief experience with Ruby on Rails and through the use of an in-depth tutorial[8] I gained a wealth of understanding in terms of how MVC applications are structured. However, having explored the technology and the potential pitfalls of a Rails project, I opted to venture into Node.js[9] and the options available under that umbrella. This choice is a very much form over function, as the achievable function is not limited by the Node.js platform as opposed to Ruby on Rails. Moreover, Node.js can potentially provide a higher level of interactivity on a real-time basis by comparison.

Above all, the appeal of having a codebase entirely written in Javascript is too appealing to ignore. Unifying client and server side development under one language not only makes development faster working solo, but also improves scalability should the project development ever expand beyond just myself.

Further investigation into Node.js led me to several pieces of middleware, built as frameworks on top of Node.js, in a similar fashion to how Rails is built atop Ruby. These included GeddyJS[10], Express[11] and SailsJS[12]. Sails builds upon the functionality offered by Express. Having explored the documentation for each, I concluded that Sails was the best choice.

[4] *Plekhanova, J (2009). Evaluating web development frameworks: Django, Ruby on Rails and CakePHP.*

[5] *Python.org, (2014)*

[6] *Rubyonrails.org, (2014). Ruby on Rails.*

[7] *Cakephp.org, (2014). CakePHP: the rapid development php framework.*

[8] *Hartl, M. (2014). Ruby on Rails Tutorial (3rd Ed.)*

[9] *Nodejs.org, (2014). node.js*

[10] *Geddyjs.org, (2014). Geddy | The original MVC Web framework for Node - a simple, structured way to create full stack javascript applications.*

[11] *Expressjs.com, (2014). Express - Node.js web application framework.*

[12] *McNeil, M. (2014). Sails.js | Realtime MVC Framework for Node.js.*

Competitor Research

Direct Competitor Research

In order to appreciate the existing options currently marketed, I looked at several applications/services that have features similar or partially related to my envisioned offering.

Nike Football mobile app (Available on iOS and Android)[13]

Nike have created a sleek and polished offering with their Nike Football app. Unfortunately, the app serves primarily as a marketing vehicle for Nike, with the majority of the content focusing on new Nike equipment or sponsored athletes. However, a section for a 'Crew' is included, allowing a user to connect with other users of the app to form a small team - functionality akin to core features of this project.

The GAME by Ronaldo (Available on iOS, Android and web browser)[14]

A creation heavily endorsed by Real Madrid and Portugal forward Cristiano Ronaldo, 'The GAME by Ronaldo' is a web application promoting a global football competition. The premise encourages 'grass-roots' football with amateur players connecting online, arranging competitive matches and progressing through the stages of an eventual 'global tournament'. This functionality is similar in principle but is tied to an external factor, the worldwide competition, providing no facility for private organisation.

Teamstuff

Teamstuff is the most feature-complete to the solution required by the project's initial problem. In fact, they outline the issue near verbatim on their homepage:

"*Any coach or manager on a social or youth sporting team knows how much time is consumed by administration*"[15]

Exploring the functionality provided by Teamstuff will aid the scoping of this project and potentially highlight pitfalls prior to their occurrence.

Abstract Competitor Research

Many applications exist which offer functionality similar to that of my proposed project, but for different disciplines or audiences. Despite being unrelated to the project subject, their structure and features bear great relevance to the project development.

Basecamp (Available on iOS, Android and web browser)[16]
Basecamp is a project management tool available across a number of platforms. Despite not being relevant to sports, Basecamp is a textbook example of an MVC application and is in fact the application for which Ruby on Rails was developed. It offers tools for companies to manage projects across a variety of disciplines and essentially presents users with a proverbial blank canvas with options such as calendar reminders, group messaging and to-do lists.

[13] *Nike.com, (2014). Nike Football App.*

[14] *Gamebyronaldo.com, (2014). The GAME by Ronaldo.*

[15] *Teamstuff.com, (2014). Free Sports Team Management Software | Teamstuff.*

[16] *Basecamp.com, (2014). Basecamp is everyone's favorite project management app.*

Competitor Research with regard to Visual Design

Since this project will also involve the branding and visual design of the product, research into popular sports brands was required to get a feel for their aesthetic.

Nike, Adidas & Puma

Three of the largest, most popular sports brands all exhibit a similar style. Typography is bold and sans-serif, often printed over contrasting imagery - either of professional, sponsored athletes or aspirational shots of generic figures. The latter are worth considering when it comes to the design of the website, particularly in the engaging, pre-signup stage.

A mood-board embodying these ideas can be found as **Appendix B**.

Images sourced from: puma.com adidas.co.uk nike.com/gb

Common elements found across these brands were:

- Clear sans-serif typography
- Emphasis on aspirational photography
- Bright, bold colours
- Clear and contrasting calls to action

User Research

Questionnaire

In order to answer a few key questions about the behaviour of the target audience, a survey[17] was designed using SurveyMonkey and 50 participants were invited. Unfortunately, of those 50, only 43 responses were collected. Furthermore, 11 of the 43 respondents were screened out of the survey due to a lack of involvement with sports teams.

Despite this, the survey gathered useful information regarding internet use in general, but also use of technology when organising sporting activities. The resultant data is included in **Appendix D**.

The questionnaire revealed much about the internet usage habits of the target audience. Over 78% of respondents claimed to access the internet from a mobile device **daily**, with only 6% stating that they **never** use such platforms. One can easily draw the conclusion that mobile devices are a part of everyday life for the target audience of the product. Stephen Hay, a Netherlands-based web developer wrote an article[18] back in 2011 on this subject, surmising[19] that there is no distinction between mobile and desktop use cases. The notion is that everyone is mobile, just for different proportions of time. For some people, mobile is 80% of the time and for others, the desktop is the primary device.

Another proponent of this inclusive, unified development style is Ethan Marcotte who wrote[20] even further back, in 2010, coining the term 'Responsive Web Design'. Since then, this practice has become more widespread in the web development and is one which I have practiced in previous projects. Given the results of my research confirming the emerging, nay dominant mobile platform, there is every reason for me to apply such a philosophy to this project.

Interviews

As part of the SurveyMonkey questionnaire, email addresses were collected from participants. A handful of these participants were then invited to take part in more extensive research in the form of an interview. Participants were selected based on their response to Q2 of the questionnaire, having identified themselves as either a Manager or Organiser.

Ultimately, 3 individuals were interviewed one-on-one around the topic of team organisation. Of these participants, one was involved with playing ultimate frisbee and the other two regularly arrange football matches. The interviews took the form of informal conversations in order to allow the discussion to be directed organically.

This research concluded the following points:

- Players don't always pay the required fees

The biggest issue and the only one mentioned in all three discussions was lack of payment. This obviously hits the organiser hard as any financial shortfall has to be filled from their own funds.

- Players don't always attend

Secondary to the payment issue, attendance is also an issue. While some players turn up but forget payment, others don't attend at all.

[17] *Hay, S. (2011). There is no Mobile Web.*

[18] *Hay, S. (2014). Stephen Hay on Twitter.*

[19] *Marcotte, E. (2010). Responsive Web Design.*

[20] *Grinsted, C. (2014). Small-Scale Sports Team Logistics Survey (direct link to survey)*

- Players are not always friends

Using Facebook or a similar tool works well, to an extent, but often not all players will be friends outside of the game and therefore not Facebook friends. Having a system separate from Facebook would be an improvement in terms of privacy.

The results of the user research will prove valuable in forming personas and scenarios to develop use cases for the product.

Bibliography

Books

- Marcotte, E. (2011). *Responsive Web design*. New York: A Book Apart.

Marcotte justifies the need for responsive web design and covers a variety of topics including grid-systems and typography.

- Wroblewski, L. (2011). *Mobile first*. New York: A Book Apart.

Wroblewski discusses the benefits of taking a mobile-first approach to client-side web development. The issues raised in this book are thought-provoking in terms of both technical development and content strategy.

- Hartl, M. (2011). *Ruby on Rails 3 tutorial*. Upper Saddle River, NJ: Addison-Wesley.

Hartl provides an overview and tutorial for the Ruby on Rails framework, explaining key MVC concepts and conventions. This information proved useful when making a choice of technology to base the project on.

- Rauch, G. (2012). *Smashing Node.js*. Chichester: John Wiley & Sons Inc.

Rauch takes a comprehensive look at Node.js as a platform, explaining its benefits and pitfalls; understanding this was useful for making a technology choice.

Journals

- Plekhanova, J (2009). *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP*. Fox School for Business, Temple University.

Plekhanova discusses the merits of a variety of web development frameworks. Despite being published five years ago, this proved interesting reading regarding technology choice.

- Razzaq, A., Latif, K., Ahmad, H., Hur, A., Anwar, Z. and Bloodsworth, P. (2014). *Semantic security against web application attacks*. Information Sciences, 254, pp.19-38.

This article considers security against attacks on web applications and will have further implications in the development process of this project.

Government Publications

- Great Britain. Parliament (1998). *Data Protection Act 1998*. London: Stationery Office.

The Data Protection Act [1998] defines UK law on processing personal information. Portions of this will be relevant to application development since the application will be collecting and storing potentially sensitive details from users.

Websites

- Gov.uk, (2014). *Agile - Government Service Design Manual*. [online] Available at: https://www.gov.uk/service-manual/agile [Accessed 17 Oct. 2014].

The Agile manual available at gov.uk proved to be a strong resource for understanding and explaining methodology choices for this project.

- Microsoft, (2014). *UML Class Diagrams: Reference*. [online] Available at: http://msdn.microsoft.com/en-us/library/dd409437.aspx [Accessed 29 Oct. 2014].

A source of information regarding UML class diagrams, providing an understanding of software planning.

- Expressjs.com, (2014). *Express - API Reference*. [online] Available at: http://expressjs.com/4x/api.html [Accessed 26 Oct. 2014].

The ExpressJS documentation provides an insight into how the middleware functions and is best used. I have used this to further my understanding of the framework upon which the server-side code will be built.

- Mike McNeil, a. (2014). *Sails.js | Realtime MVC Framework for Node.js*. [online] Sailsjs.org. Available at: http://sailsjs.org/#/documentation/concepts [Accessed 26 Oct. 2014].

More specific than the ExpressJS documentation, the Sails.js documentation provides direction for developing with the Sails.js framework.

- Hernandez, A. (2013). *An Introduction To Full-Stack JavaScript - Smashing Magazine*. [online] Smashing Magazine. Available at: http://www.smashingmagazine.com/2013/11/21/introduction-to-full-stack-javascript/ [Accessed 26 Oct. 2014].

Hernandez covers a variety of topics relating to developing web applications in Javascript. Points raised in this article helped to direct my choice of framework to build the project upon.

- Docs.mongodb.org, (2014). *The MongoDB 2.6 Manual - MongoDB Manual 2.6.4*. [online] Available at: http://docs.mongodb.org/manual/ [Accessed 26 Oct. 2014].

The MongoDB documentation covers the functionality and syntax used in MongoDB, and has served as a useful reference during server-side experimentation.

References

1. Gov.uk, (2014). *Agile - Government Service Design Manual*. [online] Available at: https://www.gov.uk/service-manual/agile [Accessed 17 Oct. 2014]

2. Wax, D. (2014). *Scrum for One*. [online] Lifehack. Available at: http://www.lifehack.org/articles/productivity/scrum-for-one.html [Accessed 10 Nov. 2014].

3. Trello.com, (2014). *Trello*. [online] Available at: http://www.trello.com [Accessed 10 Nov. 2014].

4. Plekhanova, J (2009). *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP*. Fox School for Business, Temple University.

5. Python.org, (2014). *Welcome to Python.org*. [online] Available at: https://www.python.org/ [Accessed 10 Nov. 2014].

6. Rubyonrails.org, (2014). *Ruby on Rails*. [online] Available at: http://rubyonrails.org/ [Accessed 10 Nov. 2014].

7. Cakephp.org, (2014). *CakePHP: the rapid development php framework*. [online] Available at: http://cakephp.org/ [Accessed 10 Nov. 2014].

8. Hartl, M. (2014). *Ruby on Rails Tutorial (3rd Ed.)*. [online] Softcover.io. Available at: http://ruby.railstutorial.org/ [Accessed 10 Nov. 2014].

9. Nodejs.org, (2014). *node.js*. [online] Available at: http://nodejs.org/ [Accessed 10 Nov. 2014].

10. Geddyjs.org, (2014). Geddy | The original MVC Web framework for Node - a simple, structured way to create full stack javascript applications. [online] Available at: http://geddyjs.org/ [Accessed 10 Nov. 2014].

11. Expressjs.com, (2014). *Express - Node.js web application framework*. [online] Available at: http://expressjs.com/ [Accessed 10 Nov. 2014].

12. McNeil, M. (2014). *Sails.js | Realtime MVC Framework for Node.js*. [online] Sailsjs.org. Available at: http://sailsjs.org [Accessed 10 Nov. 2014].

13. Nike.com, (2014). *Nike Football App*. [online] Available at: http://www.nike.com/gb/en_gb/c/football/footballapp [Accessed 10 Nov. 2014].

14. Gamebyronaldo.com, (2014). *The GAME by Ronaldo*. [online] Available at: https://gamebyronaldo.com/ [Accessed 10 Nov. 2014].

15. Teamstuff.com, (2014). *Free Sports Team Management Software | Teamstuff*. [online] Available at: https://teamstuff.com [Accessed 10 Nov. 2014].

16. Basecamp.com, (2014). *Basecamp is everyoneâ€™s favorite project management app.*. [online] Available at: https://basecamp.com/ [Accessed 10 Nov. 2014].

17. Hay, S. (2011). *There is no Mobile Web*. [online] The Haystack. Available at: http://www.the-haystack.com/2011/01/07/there-is-no-mobile-web/ [Accessed 10 Nov. 2014].

18. Hay, S. (2014). Stephen Hay on Twitter: "There is no Mobile Web. There is only The Web, which we view in different ways. There is also no Desktop Web. Or Tablet Web. Thank you.". [online] Twitter.com. Available at: https://twitter.com/stephenhay/status/23350345962889216 [Accessed 10 Nov. 2014].

19. Marcotte, E. (2010). *Responsive Web Design*. [online] Alistapart.com. Available at: http://alistapart.com/article/responsive-web-design [Accessed 10 Nov. 2014].

20. Grinsted, C. (2014). *Small-Scale Sports Team Logistics Survey*. [online] Surveymonkey.com. Available at: https://www.surveymonkey.com/s/92HRWTR [Accessed 10 Nov. 2014].