UNIVERSITY OF ULSTER

Bootily

COM553 Project Report

Interactive Multimedia Design

Peer group one

Jonathan Wallace

James Nelson | B00584231

Year 4

April 2015

Acknowledgements

My final year studying Interactive Multimedia Design has been both a challenge in my own technical skills and an emotional challenge. From overcoming the death of my Father whom was not just a father, but also my closest friend, I am incredibly proud of what I have been able to achieve and I hope he would have been to. To my mother who has supported me through out my life and encouraged me to continue with my education I cannot thank you enough. To my partner Gillian and her four-year-old son (soon to be five) Andrew I thank you for providing a loving home and all your support through out this year, Thank you for providing me with the time I needed to complete this year, I only wish more of it was with you. Thanks to the Northern Ireland Scout's team, especially Noel Irwin for allowing me to conduct user research and obtain user feedback. To my mentor Jonathan Wallace who was always willing to provide his help, even when it overlapped with his own personal time, thank you for all your time and may I add, it was an absolute pleasure working with you. To our course director Peter Nichol I thank you for the guidance you have provided for the past four years. Peter without hesitation provides his support when it's needed and in my own personal time of need, he acted more like a friend than a lecturer. Lastly I would like to thank my Father Samuel Nelson who sadly passed away October 2014. He always supported and encouraged my own education and without his support, I would not be who I am today. He was a close friend who I will remember fondly and miss dearly. My thanks to my Father for everything he has done for me in my life.

ACKNOWLEDGEMENTS ..... **II**

1. INTRODUCTION **7**

1.1. PROJECT INTRODUCTION 7

1.1.1. Project aim *7*

1.1.2. Project objectives *8*

1.2. OVERVIEW OF WORK UNDERTAKEN 8

1.3. OVERVIEW OF REPORT 9

2. CONCEPT DEFINITION **10**

2.1. IDEA GENERATION AND EVOLUTION 10

2.2. PROJECT METHODOLOGY 11

2.2.1. Agile *11*

2.2.2. Waterfall *11*

2.2.3. Rapid Application Development *11*

2.2.4. Methodology evaluation and comparison *12*

2.2.5. Conclusion *12*

3. REQUIREMENTS SPECIFICATION AND ANALYSIS **13**

3.1. STAKEHOLDERS 13

3.1.1. Primary Users *13*

3.1.2. Secondary Users *14*

3.2. PROJECT CONSTRAINTS 14

3.2.1. Implementation environment *15*

3.2.2. Anticipated workplace environment *15*

3.2.3. Naming conventions and definitions *16*

3.3. FUNCTIONAL REQUIREMENTS 16

3.3.1. System Boundary *16*

3.3.2. User requirements *20*

4. CONCEPT TESTING **22**

4.1. PAPER PROTOTYPING 22

4.1.1. Purpose of paper prototyping *22*

4.1.2. Tools used *22*

4.1.3. Sample pencil and paper prototypes *22*

4.1.4. Sample Mockup prototypes *25*

4.1.5. Initial user testing findings *26*

4.2. FEASIBILITY TESTING 27

4.2.1. Purpose of test *27*
4.2.2. Technology's / tools tested *27*
4.2.3. Feasibility test outcome *28*

5. APPLICATION DESIGN **29**
5.1. USER EXPERIENCE DESIGN EVOLUTION 29
5.1.1. Initial UX designs *29*
5.1.2. Branding *31*
5.1.3. Refined high fidelity designs *32*
5.2. SYSTEM TECHNOLOGIES 36
5.2.1. HTML 5 ..... *36*
5.2.2. SASS *36*
5.2.3. Foundation *36*
5.2.4. JavaScript & Jquery *37*
5.2.5. Google Maps JavaScript API v3 ..... *37*
5.2.6. QR Scanner / Reader Module *37*
5.2.7. QR Creator Module *37*
5.2.8. PDF generator module *37*
5.2.9. Git & Github *38*
5.2.10. PHP & MySQL *38*
5.3. DESIGN PATTERNS 38
5.3.1. SMACCS *38*
5.3.2. BEM *38*
5.3.3. Grid system *39*
5.4. LOGIC DESIGN 39
5.4.1. Database design & ERD *39*

6. IMPLEMENTATION **41**
6.1. PLATFORM AND TOOLS USED 41
6.1.1. Hardware *41*
6.1.2. Development tools *41*
6.2. TECHNICAL IMPLEMENTATION 42
6.2.1. Google Maps API *42*
6.2.2. SMACCS Structure *42*
6.2.3. QR Scanning / Reading / Generation *43*
6.2.4. PDF generation (FPDF) *43*
6.2.5. PHP & MySQL *44*
6.2.6. JavaScript & Jquery *44*

6.2.7. Foundation *44*
6.2.8. Git & GitHub *45*
6.3. NOTABLE CHALLENGES 45
6.3.1. Dynamic search radius *45*
6.3.2. Generate dynamic PDF *47*
7. TESTING 49
7.1. TESTING APPROACH 49
7.2. REQUIREMENTS TESTING 49
7.3. CROSS BROWSER TESTING 51
7.4. USABILITY TESTING 51
7.5. TECHNICAL ISSUES 52
7.5.1. Drag markers issue *52*
7.5.2 QR browser compatibility issues *52*
7.5.3 CSS issues *53*
7.5.4 Navigation highlighting / Breadcrumbs *53*
8. EVALUATION 54
8.1. TECHNOLOGY EVALUATION 54
8.1.1. Google Maps API *54*
8.1.2. SASS and Foundation *54*
8.1.3. Git and Github *55*
8.1.4. PHP & MySQL *55*
8.1.5. QR Scanner module *55*
8.1.6. PDF generator module *55*
8.2. USER TESTING RESULTS 56
8.3. METHODOLOGY EVALUATION 57
8.4. PROJECT OUTCOMES 57
9. CONCLUSION 59
9.1. SUMMARY 59
9.2. ACHIEVEMENTS 59
9.2.1. Time management *59*
9.2.2. Technical ability *59*
9.3. FURTHER WORK 60
REFERENCES 61
APPENDICES 63
APPENDIX A – PRIMARY USER PERSONAS 63
APPENDIX B – SECONDARY USER PERSONAS 66

APPENDIX C – USER REQUIREMENTS (1-19) 67
APPENDIX D – PAPER PROTOTYPES 72
APPENDIX E – BALSAMIQ MOCKUP'S PROTOTYPES 74
APPENDIX F – INITIAL UX DESIGN 75
APPENDIX G – REFINED UX DESIGN 79
APPENDIX E – USER SURVEY QUESTIONNAIRES 82

1. Introduction

1.1. Project introduction

Creating and participating in treasure trails is a hobby to some and enjoyable for almost all age groups. Treasure trails are enjoyable family events in which various 'clues' have been placed in and around an area. Providing participants with a starting clue, which gives direction on how to find the succeeding clue, the treasure trail is complete once all clues are found. Various church related groups within the UK, especially youth groups regularly set up treasure trails for members of the public to partake in. In saying so, treasure trails setup are by no means restricted to youth groups and such, as many people enjoy creating personal treasure trails for the enjoyment of their own family and friends.

The process of planning and setting up of a treasure trail can be time consuming and in practice, these events are commonly restricted to a single day. With the rise in popularity of hand-held mobile devices we can easily obtain and make usage of the user's location at any point. Commonly implemented within these devices are cameras, video and sound recorders, which give users of these devices the ability to create their own multimedia. Thus by taking advantage of modern technology, this project aims to deliver an experience in which the user can create their own digital treasure trails. The project enables users to provide their own personalized multimedia based clues whether it is text, video or image to each clue point within a treasure trail. Participating users are able to easily partake in user created treasure trails and discover the wealth of content this project provides.

1.1.1. Project aim

With the rise of popularity in mobile technology a wealth of devices are now commercially available which make usage of a range of operating systems. Mobile device based application development restricts distribution of the end product to the specific platform / operating system the application was developed in. For example, developing an Application in Apple's X-code would restrict distribution of the application to Apple's IOS (operating system of Apple's mobile products) based products. Not wishing to exclude potential users it was decided that the project was to be developed as a web application, thus producing an application available to all users who have access to a web browser. Therefore the aim of this project is design and implement an inclusive

online application that provides users with the ability to create, play and enjoy user created treasure trails.

1.1.2. Project objectives

In order for this project to be successful, project objectives where established and measured as a form of project completion with this project's evaluation. The objectives of this project are defined in table 1.1.

Table 1.1:Project objectives

Goal ID	Description
1	To identify the users of this project.
2	To extract user and project needs from these users.
3	To analysis user needs and obtain the technical needs of the application.
4	To research and enhance my own knowledge of the application.
5	To design a testable paper prototype.
6	To test the products feasibility via a developed throw away application.
7	To design and develop a functional application.
8	To test the end product.
9	To evaluate and reflect on the completed product.

1.2. Overview of work undertaken

Due to multi-disciplinary scope of University of Ulster's 'Interactive Multimedia Design' the work that was undertaken during this project varied largely between design and development. The project began with the unfortunate difficulties in defining the project's scope. The project had potential to become a solution for various problems, including the potential of becoming a digital tool for creating and organizing orienteering events. Once the project's scope had been narrowed down to a digital treasure trail application, research began on finding potential users and interviews with these users where had. The needs of the project where extracted and analyzed in order to develop the project's functional requirements. A functional prototype was then developed in order to test the technical feasibility of the product. Project requirements and the results of the feasibility testing where then used to design and develop a usable solution.

1.3. Overview of report

Chapter two discusses the project's idea generation process. The chapter concludes with an evaluation and selection of a software development methodology.

Chapter three discusses the gathering and analysis of the various functional projects requirements. Various stakeholders involved within the project's development are defined and discussed. Potential constraining factors of the projects are discussed, along with the concluding section defining the project's functional requirements.

Chapter four discusses the process of paper prototyping, along with a short test of an interactive paper prototype mockup. The feasibility of the product and its potential technologies are tested within the feasibility section of this chapter.

Chapter five presents an overview of the projects design. The evolution of product's user experience design is presented along with a high level overview of the project's technical design and technological needs.

Chapter six discusses the implementation process of the application. Various sections of functionality implemented within the project's end product is discussed in detail along with an overview of programming achievements achieved during implementation.

Chapter seven covers the testing of the project's end product. The project is firstly tested to ensure it has met the functional requirements defined in chapter three. The chapter then proceeds with user testing and cross browser testing and finally concluding with the findings of these tests.

Chapter eight provides an evaluation of the project that has been developed. User feedback results are analyzed along with an evaluation of the development methodology and the technologies used. The chapter then concludes with an evaluation of the project's outcomes.

Chapter nine concludes the report with a critical reflection of the project and the work undertaken during project initialization to a developed product.

2. Concept definition

2.1. Idea generation and evolution

The idea behind this project originated from a feature that was implemented into a recent remake of a video Game known as 'The Legend Of Zelda: The Wind Waker" [1]. The game itself involved the main character sailing around a vast fictional ocean and completing quests. The feature that the project's idea originated from provided users with the ability to create a message ranging from an in-game tips to an in-game image. These messages are then placed within a bottle and the player has the ability to cast these messages out to sea. Other players of the game, provided they have and are using an active Internet connection are then able to find and interact with these user created bottled messages.

The idea of allowing users to place multimedia based messages through out the world was discussed with this project's supervisor, thus creating the idea of a public digital treasure trail / hunt experience. Further discussion however led to a change in scope, which resulted in the project being aimed towards the possibilities of digital orienteering. Although the idea was well-received, early user research findings found in order for such a product to be useful, the product must provide the ability to output complex map attributes i.e. land elevation, terrain type etc. Providing such features proved to be not technically feasible, thus further change in scope was inevitable.

Although the potential solution for digital orienteering organizing had failed, whilst researching the sport of orienteering I found it intriguing how the sport is played with multiple waypoints. Building upon the idea originally generated, the idea evolved to enable the creation of treasure trails with multiple way points, as opposed to a single treasure hunt experience consisting of many non connected trail points. It was this idea that was developed and researched further and thus becoming the topic of this report.

2.2. Project Methodology

Within modern software development there are several development methodologies used in order to control and manage a product's development. Within the following several popular development methodologies have been briefly discussed. This section is then concluded with a comparison of these methodologies along with a methodology selection for the project to follow.

2.2.1. Agile

Agile is a time boxed, interactive approach to software development. It relies on software being built incrementally rather than delivering a completed product in a single development cycle. Within Agile, the potential functionality features of a product are broken down into smaller components of functionality that are prioritized and worked upon by a developer or a development team. Agile allows for a very user centered development process and has proven effective in large team based projects [2].

2.2.2. Waterfall

The 'Waterfall' approach is a sequential approach to software development. The waterfall methodology is linear and expects the proceeding sections to be completed before continuing to the next section. When developing with the waterfall it is non-advised to revise previously completed steps in the process. The typical flow of the waterfall methodology is outlined below [3].

1. Project requirements gathering
2. Project design
3. Development and implementation
4. Testing
5. Development
6. Continuous maintenance

2.2.3. Rapid Application Development

Rapid Application Development (RAD) is an incremental software development methodology, which incorporates an iterative approach to the development of the product itself. Within RAD software developers create a prototype that is iteratively build upon until the product has been developed. Once the iterative product development cycle has complete, testing begins followed by the implementation of the finished product [4].

2.2.4. Methodology evaluation and comparison

The following table, which has been used to select this project's software development methodology, describes the advantages and disadvantages of each methodology. Information sourced from [3], [4], [5].

Table 2.1: Methodology pros and cons

Methodology	Pros	Cons
Waterfall	Simple and easy to understand and use. Easy to manage. Phases are completed one at a time.	Requires customer's needs and requirements to be well understood. Very linear and inflexible to change. Software is not produced until late development.
Agile	Working software is produced frequently. Changes in requirements are welcomed.	Not particularly useful in small teams. The project can easily be taken off target.
Rapid application development	Reduced development time. Reusability of components. Quick initial error reviews occur. Encourages customer feedback.	Depends on strong team and individuals. Performances for identifying business requirements. Requires highly skilled developers and programmers. High dependency on modelling skills.

The above findings indicated that both Agile and RAD where better suited for team based software development projects. Although RAD did have the advantage of delivering a reduced development time, it requires technical expertise to be effective. Agile had the benefit of producing working software frequently, which is useful for gaining regular and early customer insight. However with the time constraints in place of this project the processes involved within Agile may in-fact result in a slower development time in comparison with RAD or the Waterfall approach. Although the waterfall model is almost restively linear and leaves little to no room for previously iterations, it is familiar, easy to manage and closely resembles how I've worked in the past. Due to the project only consisting of one designer / developer (myself) it was deemed wise to use a methodology that is familiar and will not add unnecessary issues during the project's development.

2.2.5. Conclusion

After consideration it was decided that the project's development was to follow the Waterfall approach. Both Agile and RAD are unfamiliar, require technical expertise and require multiple to many developers in order to be truly effective. The waterfall approach is familiar, easy to manage and will allow for development to effectively meet deadlines.

3. Requirements specification and analysis

This following chapter discusses the project's requirement gathering process and its findings. To ensure the project's end product was designed to ensure a heightened usability, a user centered design approach was taken. Social mediums such as Facebook and Twitter were used to gather quantitative information on potential user interest. These findings were analyzed and used to construct the user personas in which the project's design and development was centered around.

3.1. Stakeholders

With a broad range of potential users, this project made use of 'user personas' to define the user bases that would influence design decisions through out the project's design and development. User Personas are a representation of the behavior and goals of a hypnotized group of users [15]. The following section sees the project's stakeholders separated into two main persona groups – primary and secondary users.

3.1.1. Primary Users

A project's primary user group refers to the group or type of people that the project or product is primarily to be aimed at. The following section outlines the project's primary user groups along with reasoning behind scoping the project to meet their needs. Please note, user personas have been created for each primary user mentioned below. Please see user persona's for each listed user within Appendix A.

Youth group i.e. Scouts Northern Ireland- Regularly within youth groups, treasure trail events are held during camping trips /outings. This project would provide an engaging way to create treasure trails for both youth group leaders and children to enjoy.

Teachers / classroom assistants - Teachers and classroom assistants within schools enjoy getting their pupils outside of the classroom to conduct outdoor based activities. This project would create a perfect opportunity to get children exploring the outdoors in a fun and engaging manner.

Children - Young children whether they be creating or playing treasure hunt's will play a large role within the user base of the application. Such an application will aid in getting children out-doors more.

3.1.2. Secondary Users

The secondary user group of a project refers to a group of people that may not have as much influence in decisions as the primary users, however are still relevant stakeholders of the product. The following section outlines the project's secondary user's along with reasoning behind scoping the project to meet their needs. Please note, secondary personas have been created within Appendix B for each primary user listed below.

Fathers / Mothers – With the youth of today becoming more confined to indoor activities i.e. video games. The application delivered in conjunction with this project will help in getting children out more, spending more time with their family in a fun, engaging and interesting way.

3.2. Project constraints

Further project background research sourced potential constraints in which the end product must meet in order to be successful. These constraints are listed within table 3.1.

Table 3.1: Project constraints

Constraint ID	Requirement
C-1	The application must be developed as a web application in order to be usable by various devices and platforms.
C-2	Due to the numerous screen sizes of devices, the developed application must be responsive and provide a consistent, usable experience across all devices.
C-3	The application must not be reliant on WI-FI, wired based connections and must provide an optimal experience in conditions where 3G may only be available.
C-4	The application will provide QR codes to allow for physical placement of treasure trail clue points.
C-5	The application shall provide the ability of reading and analyzing QR codes.
C-6	Due to the potential varying range of environments the application will be used in the application's design must provide good contrast.

3.2.1. Implementation environment

As stated within the project's constraints, the project was to be developed as a web application available from specific URL (Uniform Resource Locater). By developing the project as a web application the project can ensure that potential users are not excluded due to their preference of device or operating system use. The screen size and screen resolution can vary quite significantly in modern mobile devices. Therefore the application must be developed to respond to screen size and provide a consistent experience across all platforms (as mentioned in constraint C-2).

3.2.2. Anticipated workplace environment

The application's workplace environment will vary quite significantly. As the project's purpose is to allow digital creation the playability of treasure trails, a large number of users will use the product in an outdoor environment. Due to the vast variation of natural lighting in such an environment, it is important to design an interface that is visible both in bright sunlight and dull lighting (See constraint C-6). The application is to make use of QR codes as mentioned in constraint C-4. The potential condition of placed QR codes will unfortunately be at the mercy of environmental factors such as the weather.

While setting up a treasure hunt, user research may find that users prefer such tasks on a larger monitor perhaps while at home on a desktop computer. While setting up a trail it is likely that the user will be devoting his / her attention to the application. Therefore during the 'create treasure trail' user journey it is important the product provides rich visual feedback. The desktop experience should be optimized for the usage of a keyboard and a mouse however must support the usage of touch screen for such devices as a touch screen monitor. The product's interface must be designed to maintain consistency between desktop and mobile devices (Constraint C-1, C2).

3.2.3. Naming conventions and definitions

The following table lists definitions, terms and acronyms used by stakeholders involved within the project.

Table 3.2: Naming conventions and definitions

Term	Definition
Clue	A multimedia item associated with a marker.
Finish marker	The finishing point of a user created treasure trail.
Way point Hunt	The middle point(s) of a user created treasure trail.
Maps	A digital interactive map provided by Google Maps API.
PDF	A Portable Document Format (PDF) is digital file format. Within the application PDF's will be used to output QR codes and treasure trail information.
QR Code	QR code (abbreviated from Quick response Code) is a graphic matrix barcode used within the application to uniquely identify a treasure trail point [6].
Responsive	Coined by Ethan Marcotte, the term responsive refers to responsive web design. Responsive web design refers to a web development and design technique that allows the design of a website to respond to different screen sizes and resolutions [7].
Starting Point	The starting point of a treasure trail.
Treasure trail	A user created treasure trail.
Zoom level	The level the map is currently zoomed in at.

3.3. Functional requirements

3.3.1. System Boundary

The following section portrays two 'Use Case Diagrams (UCD)' that identify the boundaries and interactions between the users and the application in two main 'application use' contexts. The first UCD (figure 3.1) describes the interactions involved in the 'create a treasure trail' user journey. The second UCD (figure 3.2) describes the interactions involved in the 'find a treasure trail' user journey.

Figure 3.1: Use Case Diagram – Create treasure trail

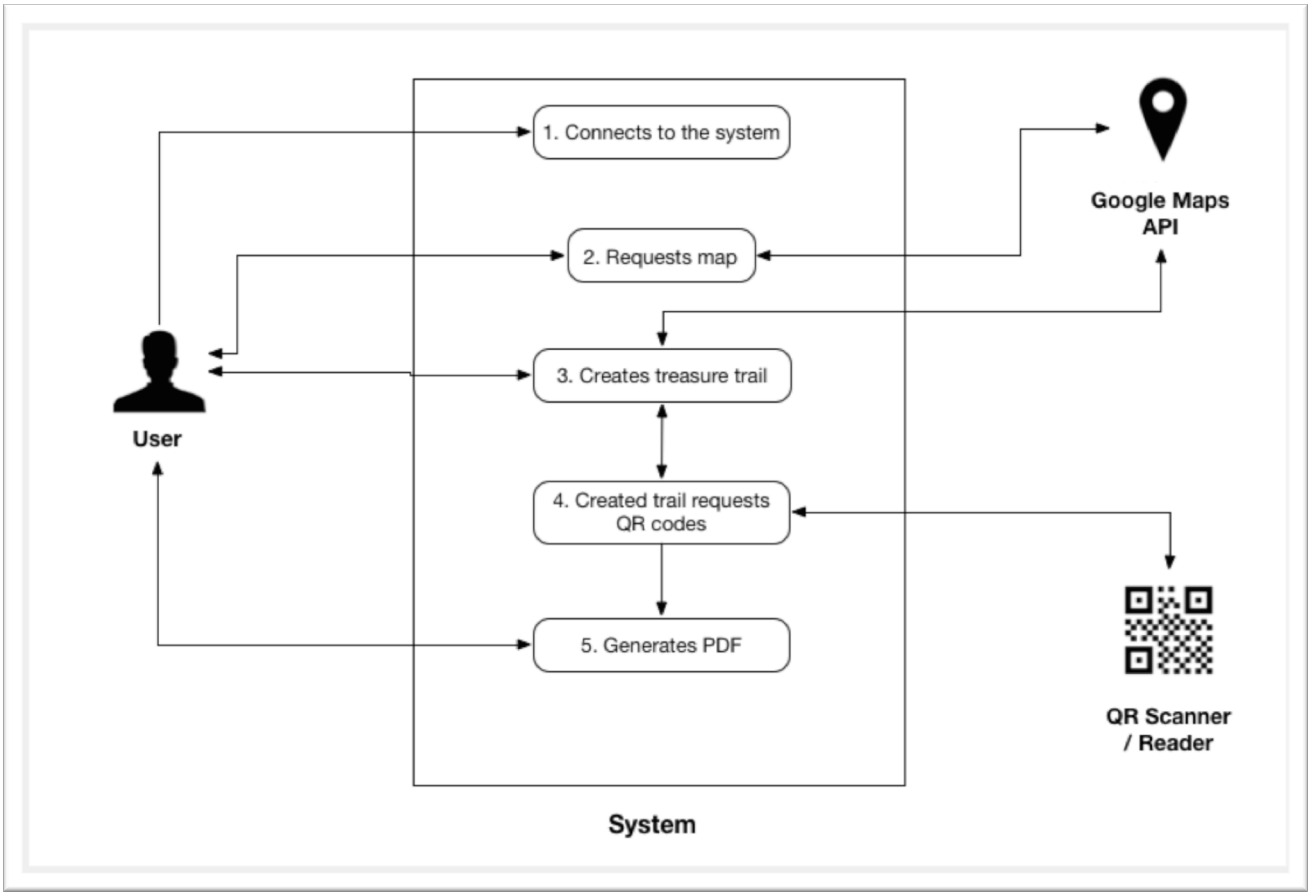


Figure 3.1's UCD portrays three main actors, these being the user, the Google Maps API and the QR Scanner / Reader module. A description on each of these actors is listed below:

User - A person whom is making usage of the application.
Google Maps API - A mapping Application Program Interface provided from Google that allows the embedment and usage of Google Maps within a third party application.
QR Scanner / Reader - A QR Code scanning and reading module that allows QR codes to be created, read and used by the system.

Table 3.3 provides a short narrative description of the use cases defined in figure 3.1.

Table 3.3: Create trail UCD description

Number	Use Case title	Description
1	Connects to the system	A connection between the user and the system is established.
2	Request map	The user request functionality from the system, which requires the usage of the Google Maps API. A request is sent to the API and an instance of Google Maps is returned.
3	Creates treasure trail	User creates a treasure trail using the systems functionality.
4	Created trail request's QR codes	Each clue added during treasure trail creation will request a separate QR code to be created in order for the clue to be uniquely identified via the QR reader module.
5	Generates PDF	Once a treasure trail has been created a PDF will be generated for the usage of the treasure trail creator.

Upon creating a treasure trail the user of the system may wish to find and play a pre-existing treasure trail. The following use case diagram explores the interactions of a user playing a treasure trail.

Figure 3.2: Use case diagram – find a trail

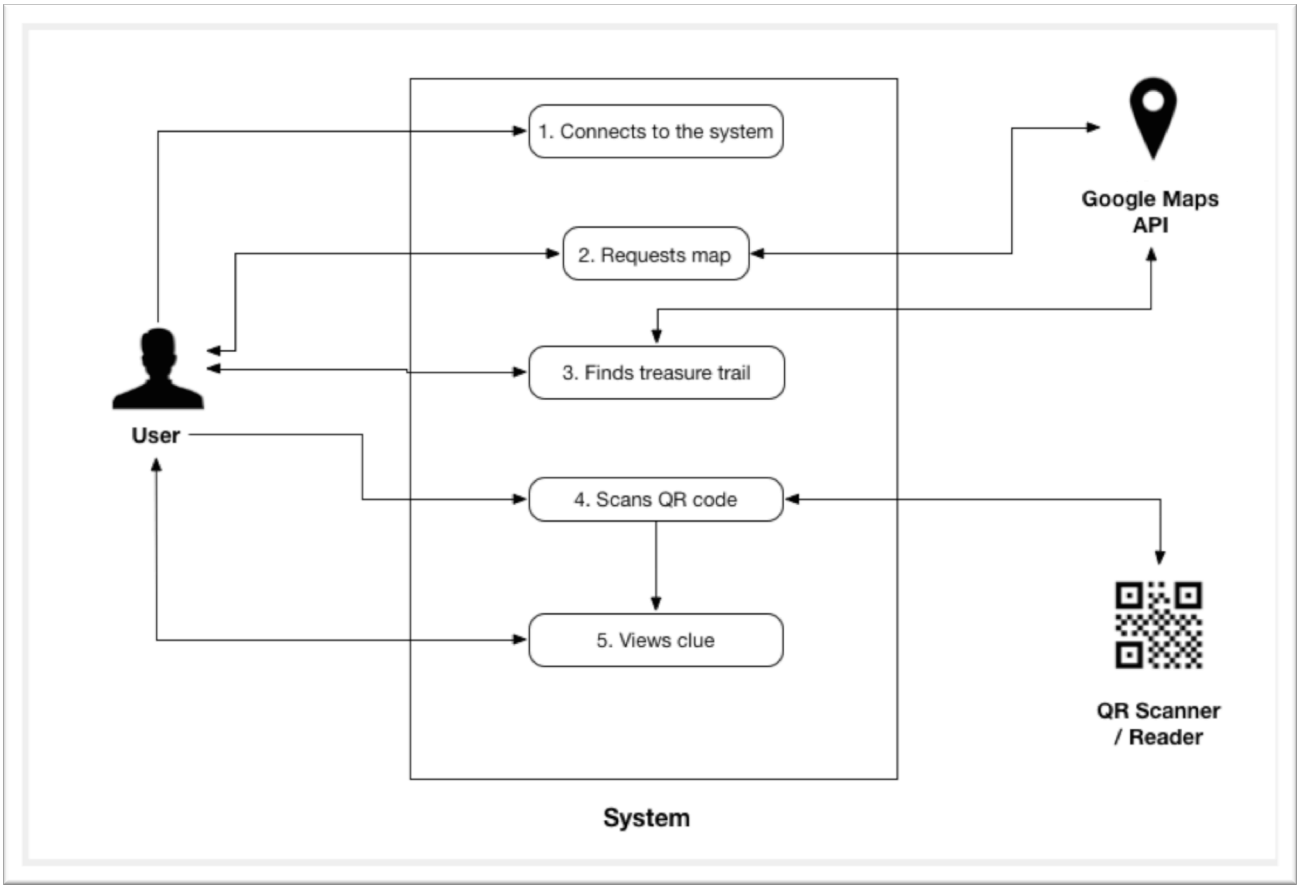


Figure 3.2 makes usage of the same actors defined within figure 3.1, therefore please refer to the actors description provided with figure 3.1 if needs to be. Table 3.4 provides a description of the use cases portrayed within figure 3.2.

Table 3.4: Find a trail UCD description

Number	Use Case title	Description
1	Connects to the system	A connection between the user and the system is established.
2	Request map	The user request functionality from the system, which requires the usage of the Google Maps API. A request is sent to the API and an instance of Google Maps is returned.
3	Finds treasure trail	User finds a treasure trail using the systems functionality.
4	Scans QR code	User attends the hunt's location and finds a QR Code. The user can now use the systems QR scanning functionality to scan the QR code.
5	Views clue	The system attempts to make a match between the QR code scanned and the QR codes held within the database. If a match is found the user will be provided with the ability to view the found clue.

3.3.2. User requirements

The following section see's five of the project's functional user requirements listed. These requirements have been gathered during user interviews, project background research and from the UCD's provided within figure 3.1 and figure 3.2. Please note, the remaining user requirements can be found within Appendix C.

Requirement ID - #1
Type - User authentication
Description - The product shall allow users to create an account.
Rationale - To allow users to enjoy a personal experience with the product.
Fit criteria - In order for the user to interact with the product they must create an account.
Dependencies -
Priority - 10

Requirement ID - #2
Type - User authentication

Description - The product shall allow users to log in.

Rationale - To allow users to view their account.

Fit criteria - In order for the user to interact with the product they must log in to account.

Dependencies - #1

Priority - 10

Requirement ID - #3

Type - User authentication, security

Description - The product will allow user to sign out of an account.

Rationale - To allow users to sign out of their account.

Fit criteria - For security reasons, the product must allow users to log out of their account from all pages whilst using the product.

Dependencies - #1, #2

Priority - 10

Requirement ID - #4

Type - Security

Description - The product must automatically sign a user out if idle for a certain amount of time.

Rationale - To protect the user's information.

Fit criteria - For security reasons, the product must force a logout on a certain timescale to ensure the safety of user data.

Dependencies - #2, #3

Priority - 7

Requirement ID - #6

Type – Technical functionality

Description - The product must be able to accurately retrieve the user's location.

Rationale - To retrieve localized results.

Fit criteria - In order to provide localized data the product will need to know the location of the user.

Dependencies –

Priority - 8

4. Concept testing

4.1. Paper prototyping

Paper prototyping is a technique used by many designers, especially those involved in UX design whilst exploring the many potential design directions a project could follow. The following section discusses the purposes of paper prototyping along with the process of paper prototyping in conjunction with this project.

4.1.1. Purpose of paper prototyping

The concept of paper prototyping allows designers to get initial, testable ideas out early in order to test the usability and feasibility of a product. Within the field of interaction design, user feedback is crucial in creating a truly, user centered product. Being able to test a product on paper allows designers to quickly create a low fidelity mockup that if proven to fail, can simply be discarded or re-worked. Thus being a much cheaper approach to initial product testing in comparison with the development of a functioning prototype. A notable issue within interaction design is, the higher the fidelity of a product presented to a user during user feedback sessions, the more reluctant users tend to become in providing critical feedback. Presenting a user with a low fidelity paper mockup encourages the user to give much more honest critical feedback.

4.1.2. Tools used

The most common tools used within paper prototypes are of course paper and a pencil. I have found however that creating and user testing with an interactive low fidelity mockup provides a better understanding of the product to the user. Balsamiq Mockup's is an excellent design tool that enables the creation of interactive low fidelity prototypes [8]. During the process of paper prototyping both paper / pencil and digital mockup's where used.

4.1.3. Sample pencil and paper prototypes

Figure 4.1 - 4.3 portray early prototyping of the application in the context of mobile device usage. Please note, during early prototypes treasure trails where known as 'bottles'. This terminology was aborted in future design as it proven to fail during early user testing (see conclusion). Figure 4.1 displays an early adaption of a feature that would allow users to see nearby treasure trails. The idea

of having a collapsible panel (pull able) to view nearby treasure trails was explored. Doing so would open a non-obstructive panel that would allow for quick access to near-by treasure hunts.

Figure 4.1: Explore nearby paper prototype

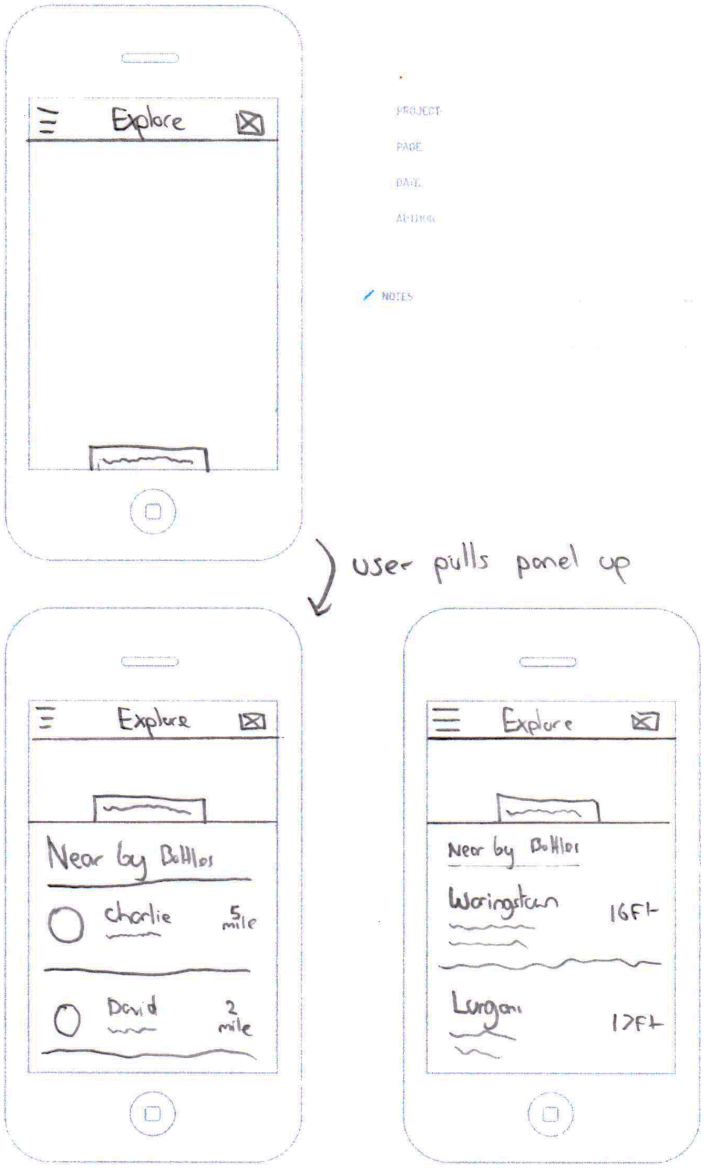


Figure 4.2 portrays an early adaption of the ‘create a treasure hunt’ user journey. Within the context of figure 4.2, the user has clicked on the ‘add’ button which provides the user with the ability to add a treasure trail clue. The user is then presented with a screen that provides image-adding functionality.

Figure 1.2: Add a clue paper prototype

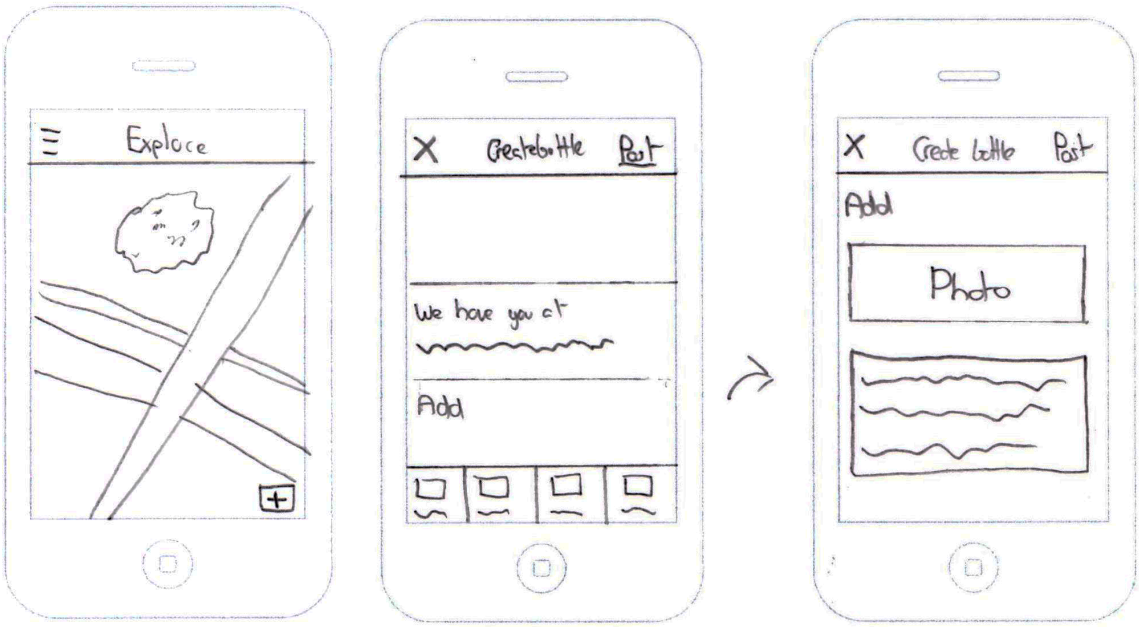
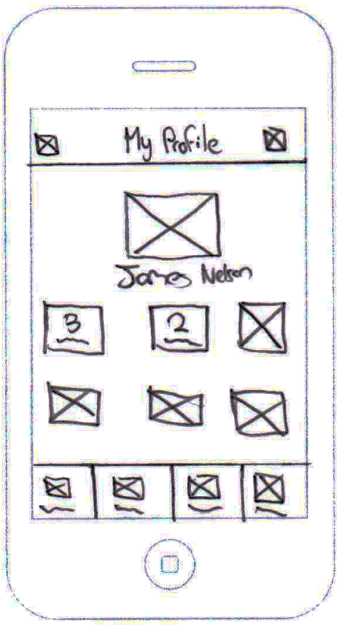


Figure 4.3 shows an early adaption of the applications potential profile page within the context of a mobile device.

Figure 4.3: Naming conventions and definitions



To view a complete list of paper based prototypes please refer to Appendix D.

4.1.4. Sample Mockup prototypes

The following section presents a sample of screens of the interactive low fidelity prototype designed and used in the project's initial user testing. Alike the paper prototype section, the following designs continue the exploration of the application's design in the context of mobile device interactivity. As mentioned earlier within this report, the following screens where designed using Balsamiq Mockup [8].

Figure 4.4 portrays the 'forgotten password' user journey. Within this user journey, the potential layout for the login and create an account pages are displayed.

Figure 4.4: Naming conventions and definitions

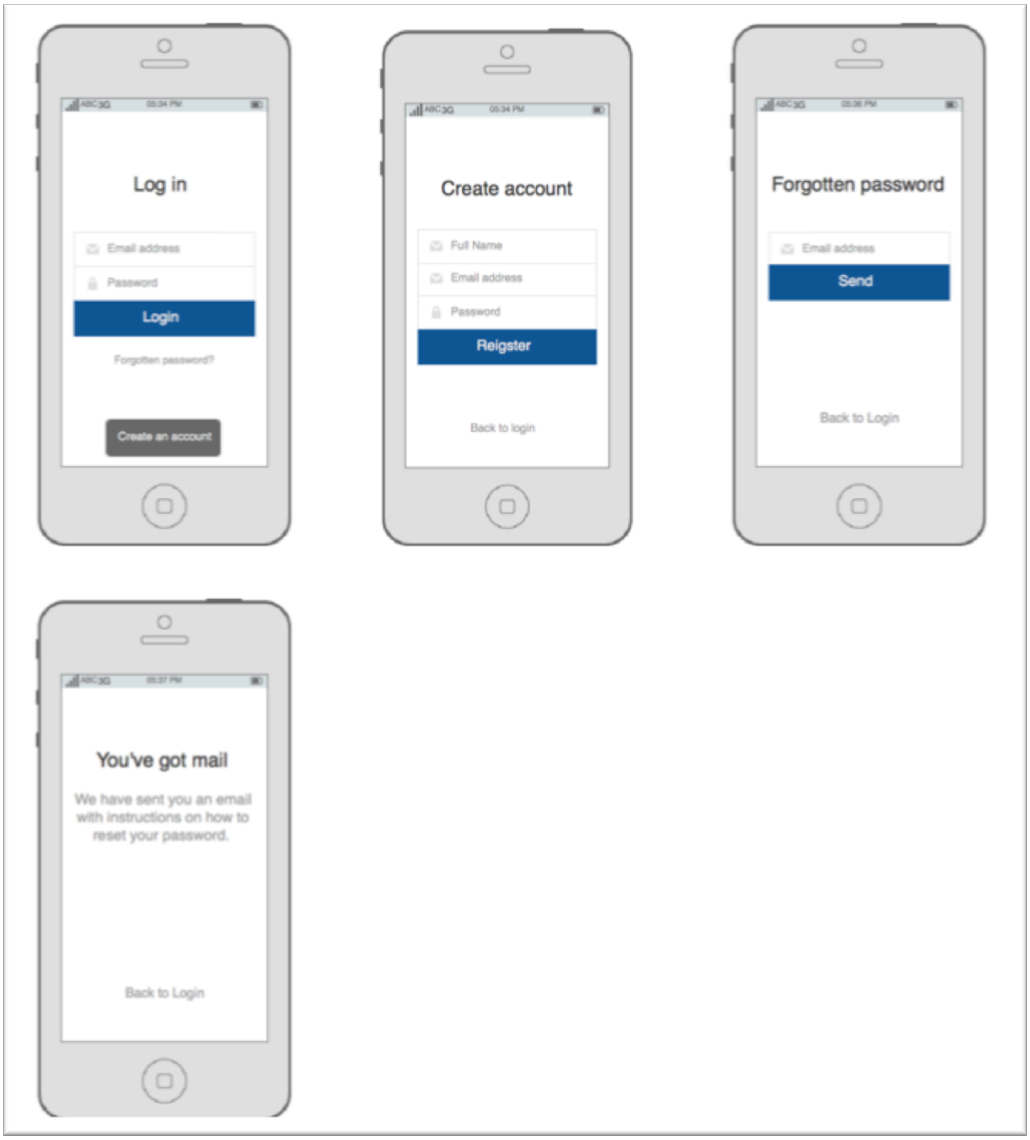
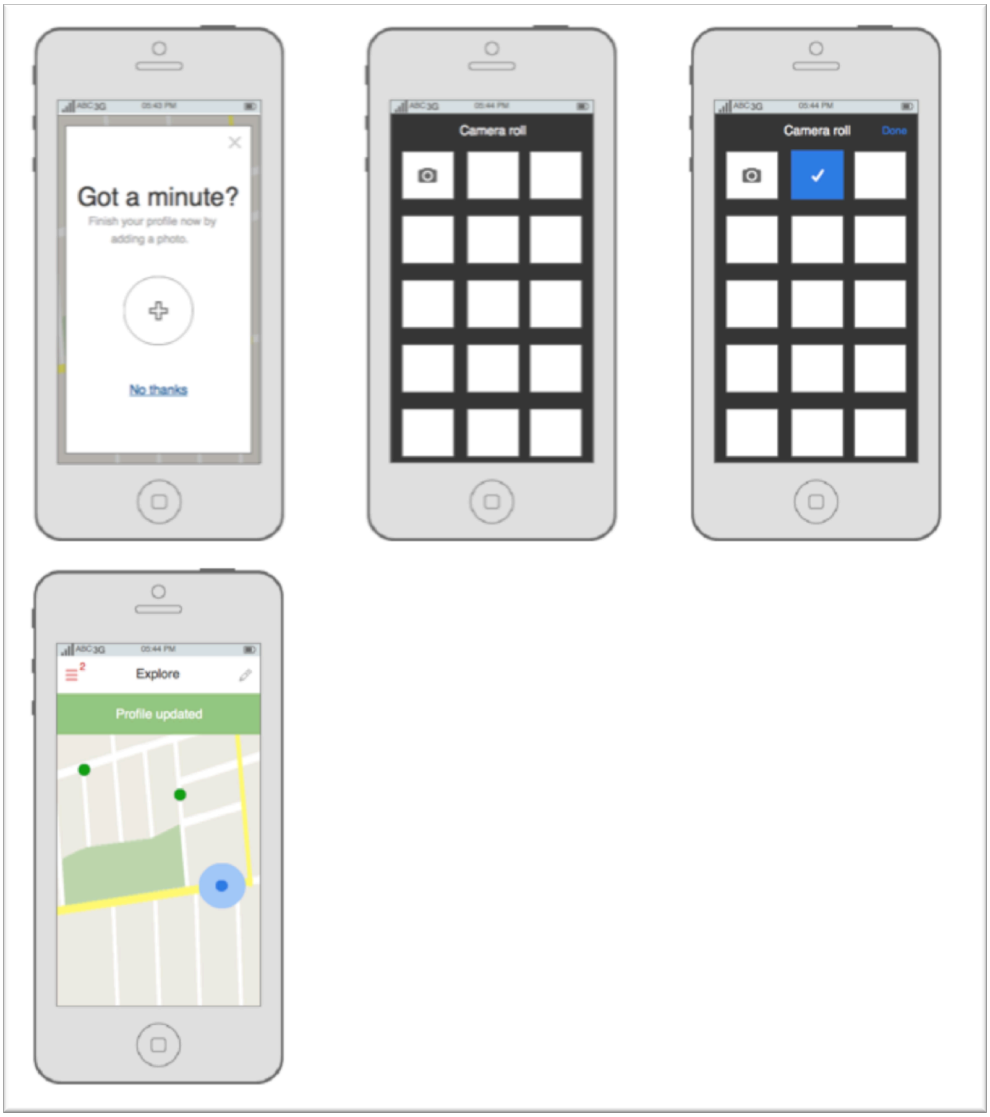


Figure 4.5 explored an experimental feature within the application provided the user had logged in and had as of yet, had not uploaded a profile image. Figure 4.5 displays the user being prompted with a modal that requests a profile image to be added. Providing the user does not close this modal,

the user then proceeds to add an image using their computing device (image context - IPhone). Upon doing so the user is provided with subtle visual feedback indicated that the image upload was successful.

Figure 4.5: Upload image paper prototype



To view an expanded list of Balsamiq prototypes please refer to Appendix E.

4.1.5. Initial user testing findings

On product conception, an immediate search to find potential users that closely resembled the attributes defined within the project's user personas (see chapter 3) began. Noel Irwin of the Scouts group (Lisburn district) was as a result selected for user testing and regularly provided his time for user feedback sessions. Noel was also able to provide access to other Scout group members to aid with the project's user testing. Provided below is a short description on Noel Irwin:

Noel Irwin – Noel is a middle-aged man who describes himself to in having basic to intermediate proficiency in modern computing. He is currently a team leader of the Scouts (Lisburn district) and is committed to providing engaging activities for his group.

Whilst testing the low fidelity mockup with Noel and other members of the Scouts group, it became obvious that the terminology of 'bottles' used to describe treasure hunts was not intuitive. Findings indicated that the overall purpose of the product was not made obvious through the project's initial design and the usage of the terminology 'bottles' resulted in further user confusion. It was thus clear that the terminology 'bottles' was to be aborted in future design and development. User feedback findings however indicated that the application's layout and form was highly usable and users praised the application's ease of navigation. An interesting observation found during user testing was participants tended to evade the 'upload an image modal' upon user login (please see figure 4.6). Upon being presented with the image upload modal, the majority of testing participant's immediate reaction was to close the modal. It became clear that the implementation of the 'image request modal' may become obtrusive and frustrating to the user rather than helpful. Thus it was decided the future product should only provide image upload functionality within the user's profile section.

4.2. Feasibility testing

4.2.1. Purpose of test

With the project's requirements established, testing the technical feasibility and limitations of the technologies required for the product's implementation was essential. Running a technical feasibility test provided the following benefits:

- A better understanding of how the project's required technologies work is developed.
- The compatibility of these technologies with other application technologies is tested.
- The ambitiousness of the project is made easier to evaluate.

4.2.2. Technology's / tools tested

Due to the lack of technical experience with the Google Maps API and the project's obvious future reliance on this technology, it was important to become familiar with the API before major

development began. The project was to attempt to make usage of a MVC (Model View Controller) framework known as Angular.JS. Such a framework would help to organize code effectively and would further extended the functionality of JavaScript. Lastly it was decided that the project should make use of a HTML/CSS based framework - Foundation. This would provide the project with a reliable responsive grid system and usage of JavaScript based components.

4.2.3. Feasibility test outcome

In order to test the feasibility of these technologies, a functional prototype was created which experimented on the Google Maps API, Angular.JS and Foundation's grid framework. Experimenting with the Google maps API successfully proved that integration with the API was realistically feasible. The Google Maps API was found to be well documented and implementation of the mapping features essential for success of this project would be relatively straightforward.

Whilst experimenting with Google Maps API, HTML5's geo-location feature was tested to ensure the technology could accurately obtain the user's location. The Geo-location feature was found to be very accurate and if needed, the feature can accurately track a user's location down to five meters. Experimenting with the basics of Angular gave insight on the usefulness of this technology. Features such as two-way data binding, easy filtration of objects / arrays and generally more manageable code are provided from the framework. Experimenting with Angular indicated an exceptionally steep learning curve as the technology proved to be exceedingly complex. In regards to application development, Angular tends to become the 'roots' of a project, which without having an experienced working knowledge of the technology can lead to many unnecessary development issues. Due to the lack of knowledge of Angular.JS it was felt that development should unfortunately proceed without the usage of Angular.JS.

Lastly, the Foundation framework was also experimented with, which was found to be generally straightforward to use. A criticism against foundation and HTML/CSS frameworks in general is they tend to bring a large amount of potentially unused code to projects. To save on potential bandwidth a customized version of Foundation was to be downloaded and used to ensure unnecessary files and modules are not included within the finished product.

5. Application Design

5.1. User Experience design evolution

5.1.1. Initial UX designs

Building upon user research findings, prototype testing and feasibility testing the design of the project began to move towards a much more higher fidelity and richer experience. Whilst designing initial high fidelity user journeys, a focus was made upon the desktop experience of the application. The following section provides a sample of the high fidelity mockups designed before further design refinement occurred.

Figure 5.1: Initial UX – Sign up

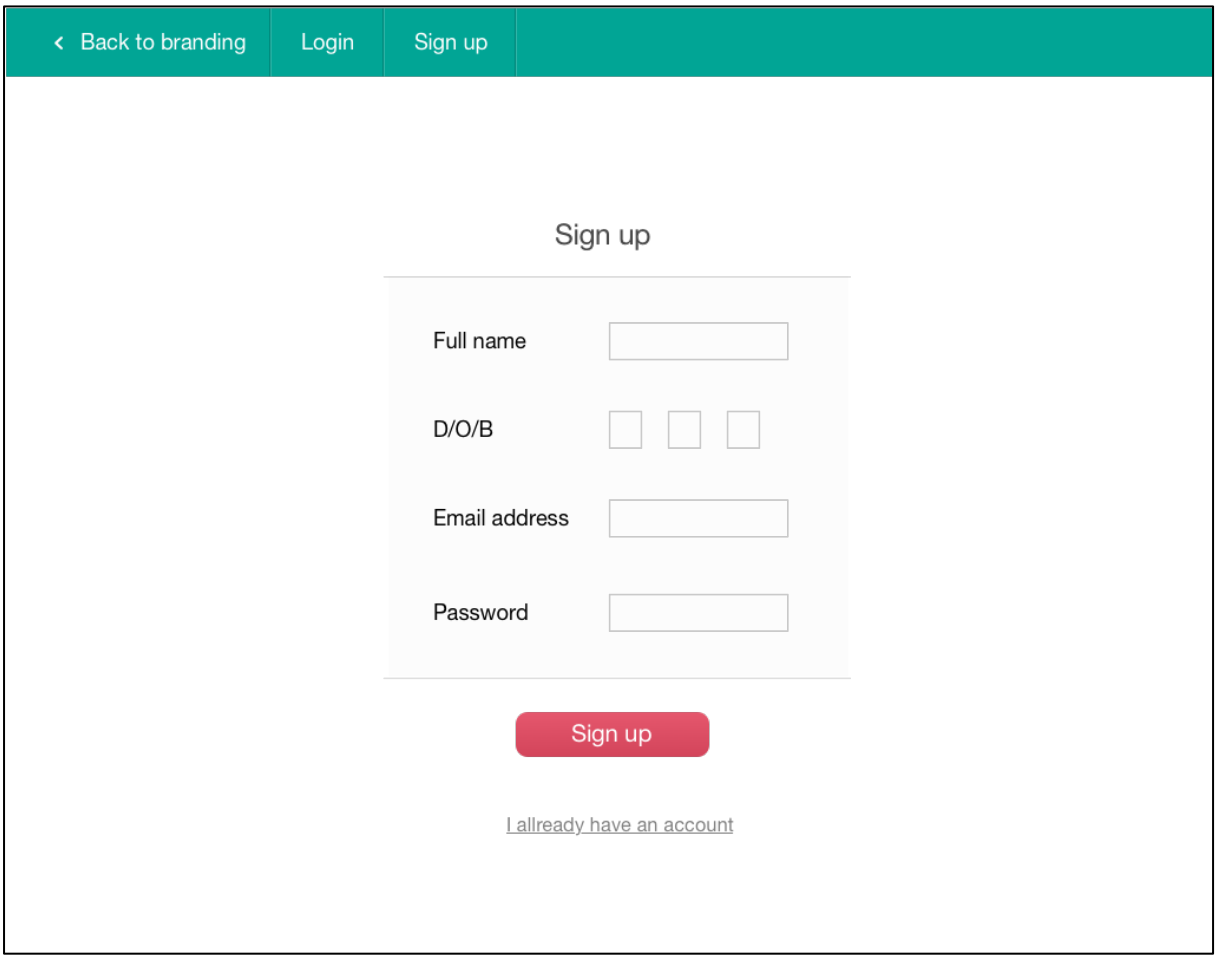


Figure 5.1 displays the initial sign up page. Input boxes are used to provided users with the ability to successfully register with the application. A button labeled ‘sign up’ is provided in order to post

user data to the future connected database. Navigation located at the top of the page allows the user to return back to the home page and the log in page if needs to be.

Figure 5.2: Initial UX – Create a trail

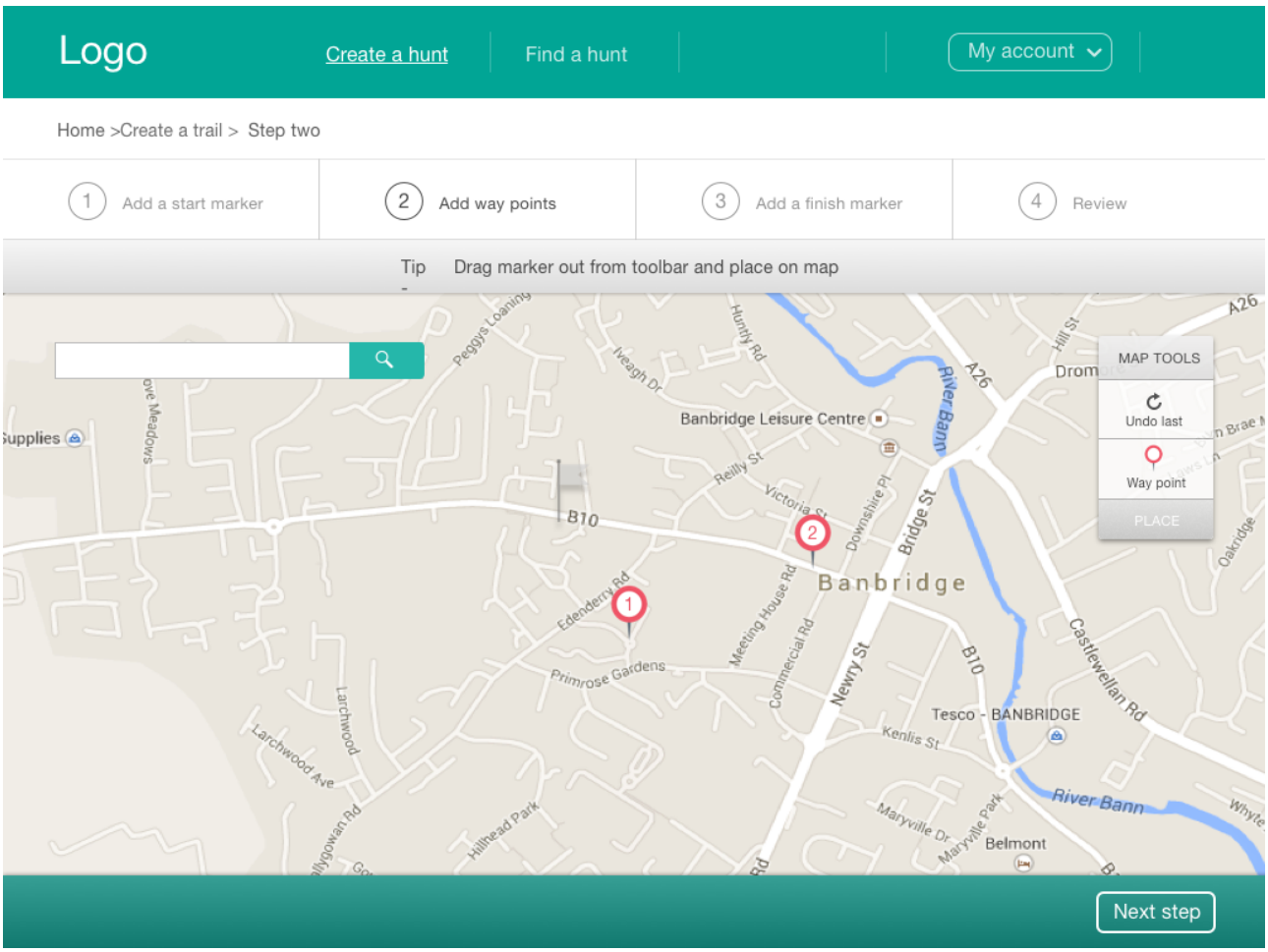


Figure 5.2 displays a use case in which a user is creating a treasure trail. Creating a treasure trail has been designed as a multi step process, hence the reasoning for the step progress component located under the breadcrumbs. In the context of figure 5.2, the user drags markers out from the map tools panel and places these markers onto the mapping area. Once the user is satisfied with their marker placement, they click on the 'next step' button to continue to the succeeding step.

Figure 5.3: Initial UX – find a treasure trail

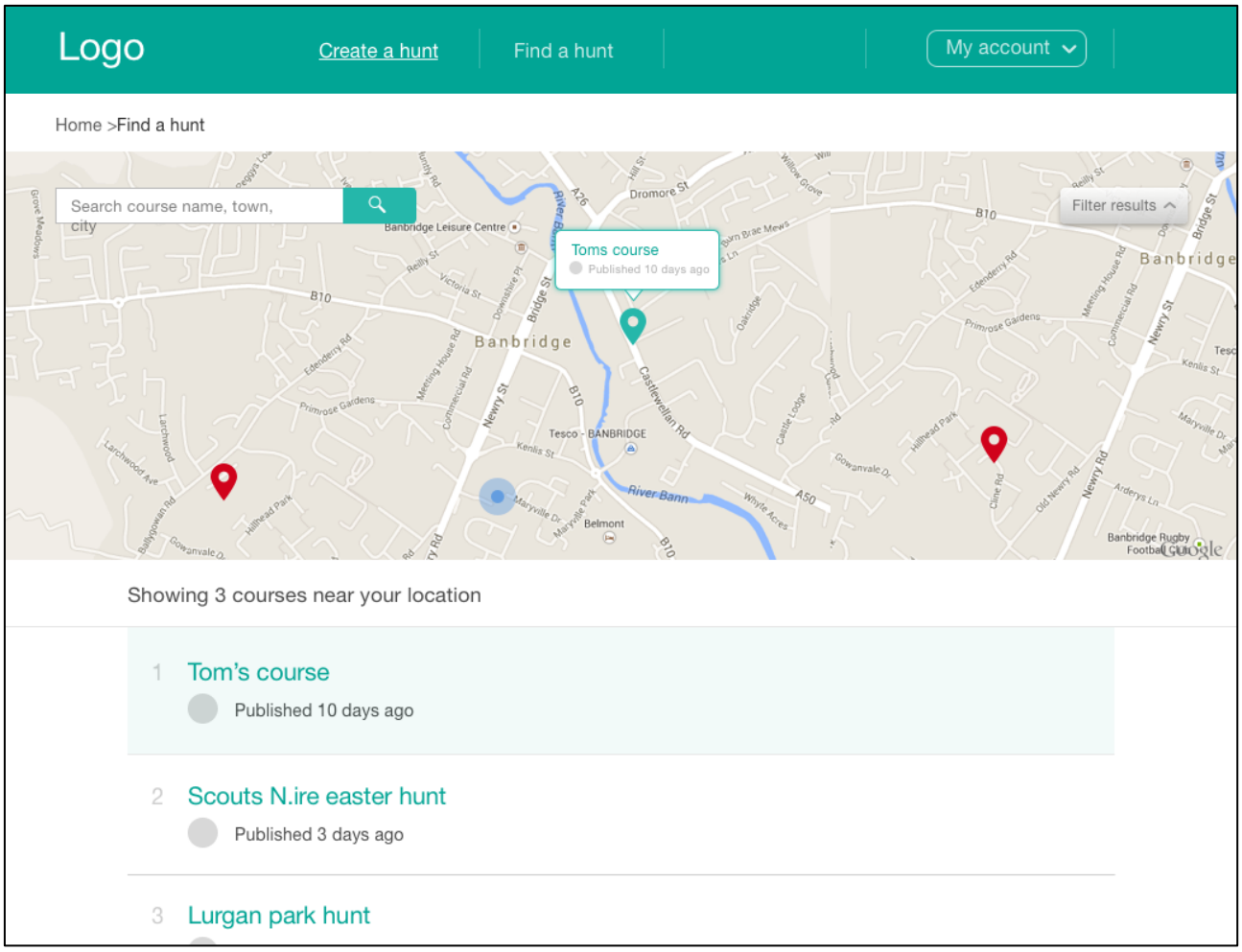


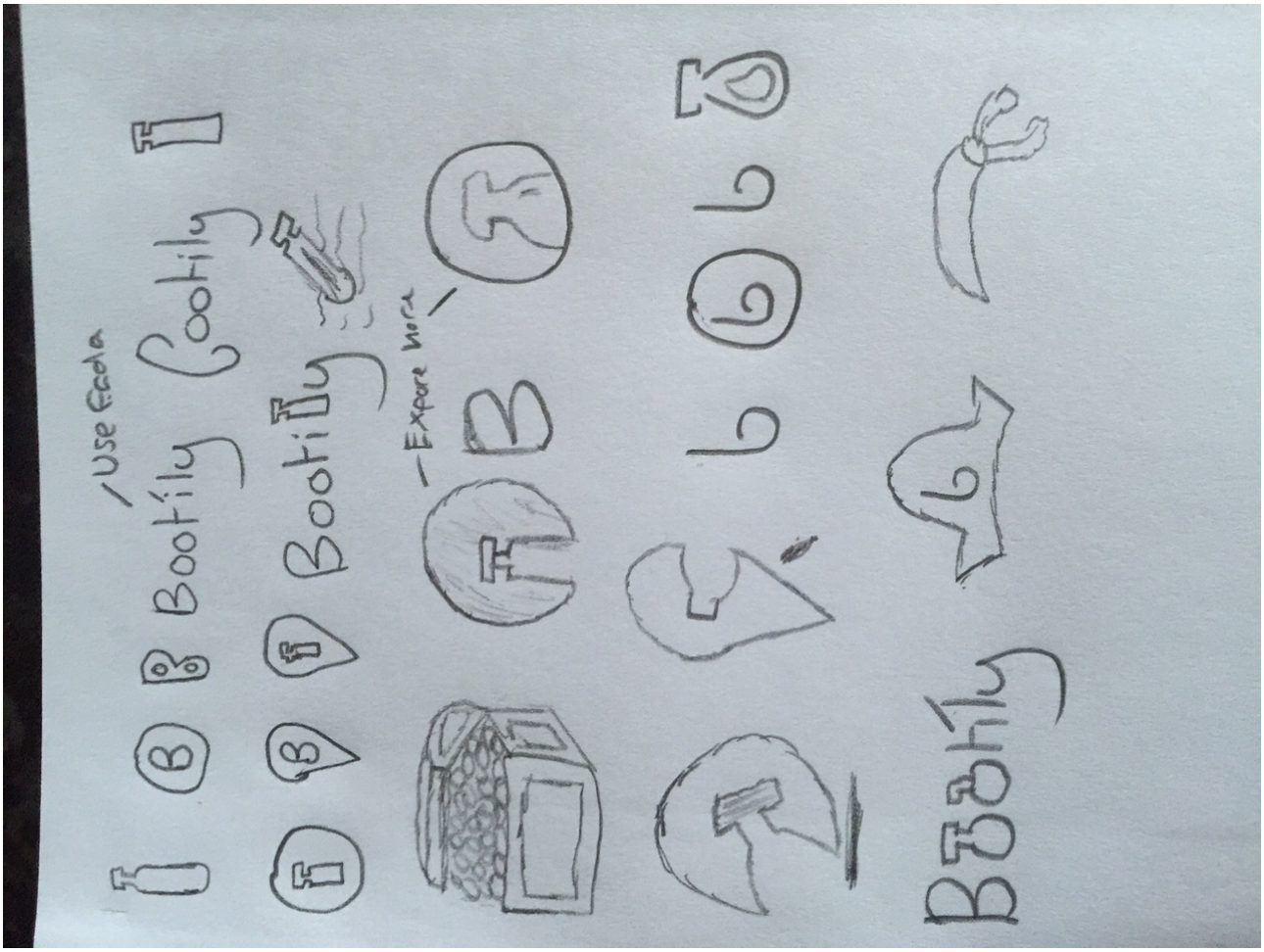
Figure 5.3 displays a use case in which the user is finding a trail. In the context of figure 5.3, the user has selected a trail he / she is interested in which has highlight the relevant information.

Please see append D to view additional initial UX mockups.

5.1.2. Branding

The project's branding began as rough ideas sketched onto a small notepad from various points in the project's development life cycle. From project initialization it was clear that the 'message in a bottle' concept would make its way in some form into the project's branding. Sketches displayed in figure 10 portray the initial ideas in which the project's branding evolved from.

Figure 5.4: Project's branding – initial ideas



Further digital refinements of the sketches in figure 5.4 resulted in the project's branding being created and thus used within the system (figure 5.5).

Figure 5.5: Project's branding

5.1.3. Refined high fidelity designs

Upon obtaining further customer feedback from Noel Irwin and members of the Scouts youth group, feedback indicated that the current look and feel of the application felt a bit 'serious'. Due to the fact that children would be making use of the application it was decided that the application was to be redesigned to be more approachable for children. User feedback also provided evidence that users had been struggling to understand the 'create a trail' functionality. It was found that the

'create a trail' user journey felt to complex and needed to be refined. The following chapter provides refined UX visuals that had been created in response to user feedback.

Figure 5.6: Refined UX – Login page

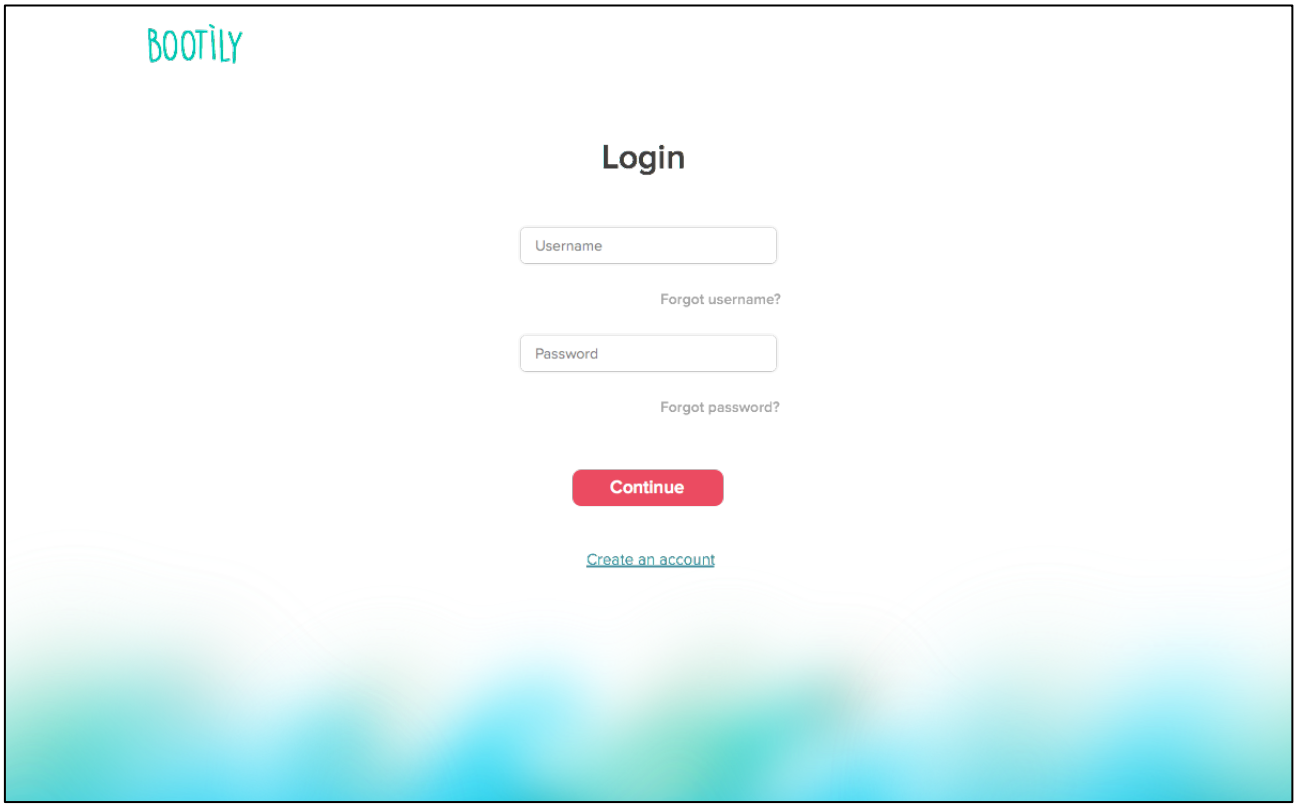


Figure 5.6 displays a refined login page. As shown in figure 5.6, the aesthetics of the application had been updated to be much more vibrant and colorful. The project's branding has also been utilized throughout these refined designs. The context of figure 5.6 portrays a log-in screen which enables users to log into the system. A link is further provided to enable the creation of an account along with provided navigation elements.

Figure 5.7: Refined UX – find hunts

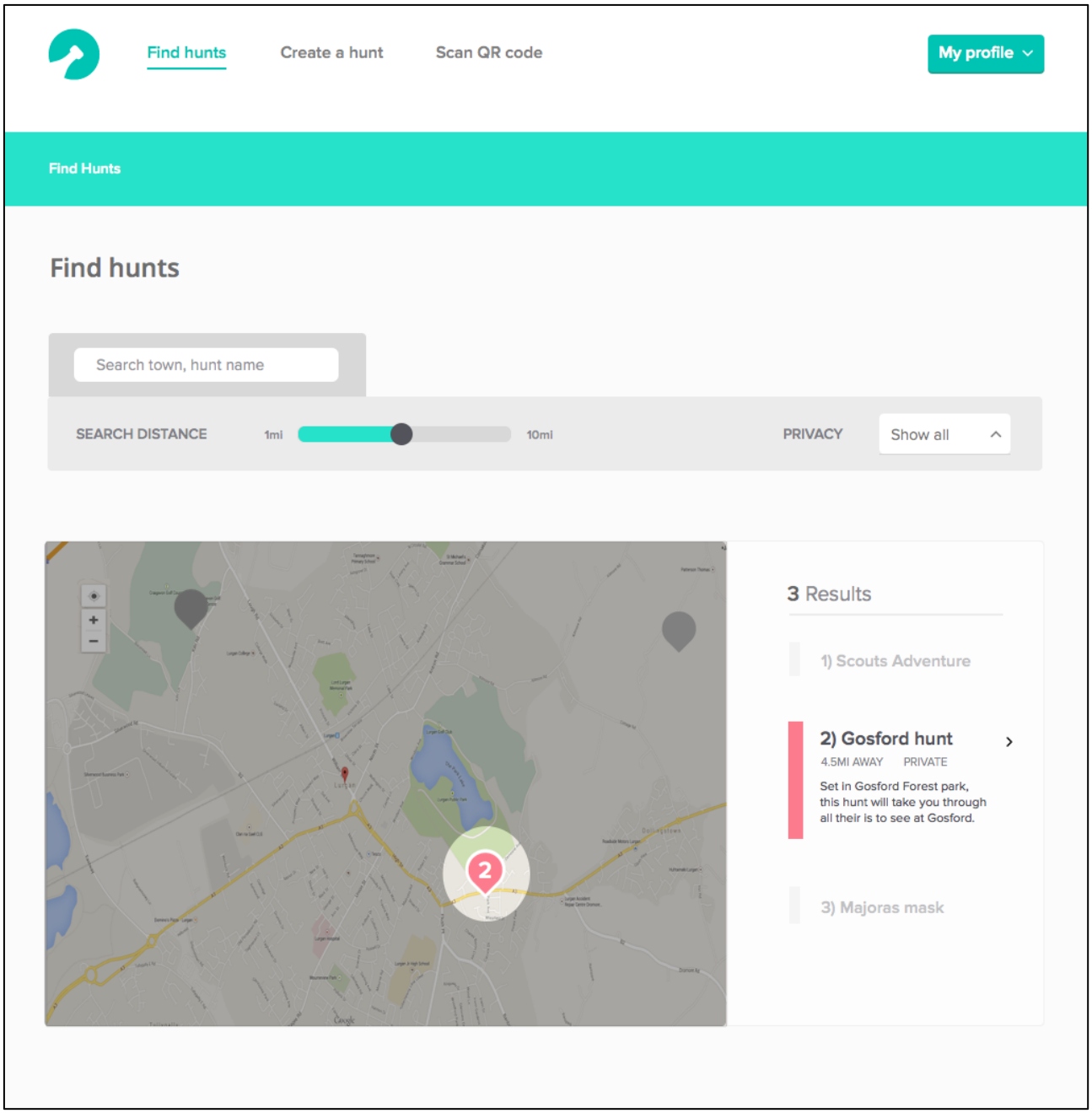


Figure 5.7 portrays a refined ‘find a treasure trail’ mockup. Search filters have now been implemented as a connected component rather than appearing as the resulting interaction of a dropdown menu. The layout utilizes a matching colour scheme between map markers and list items to provide an intuitive connection between the search results appearing on the map and on the search panel.

Figure 5.8: Refined UX – Create a hunt

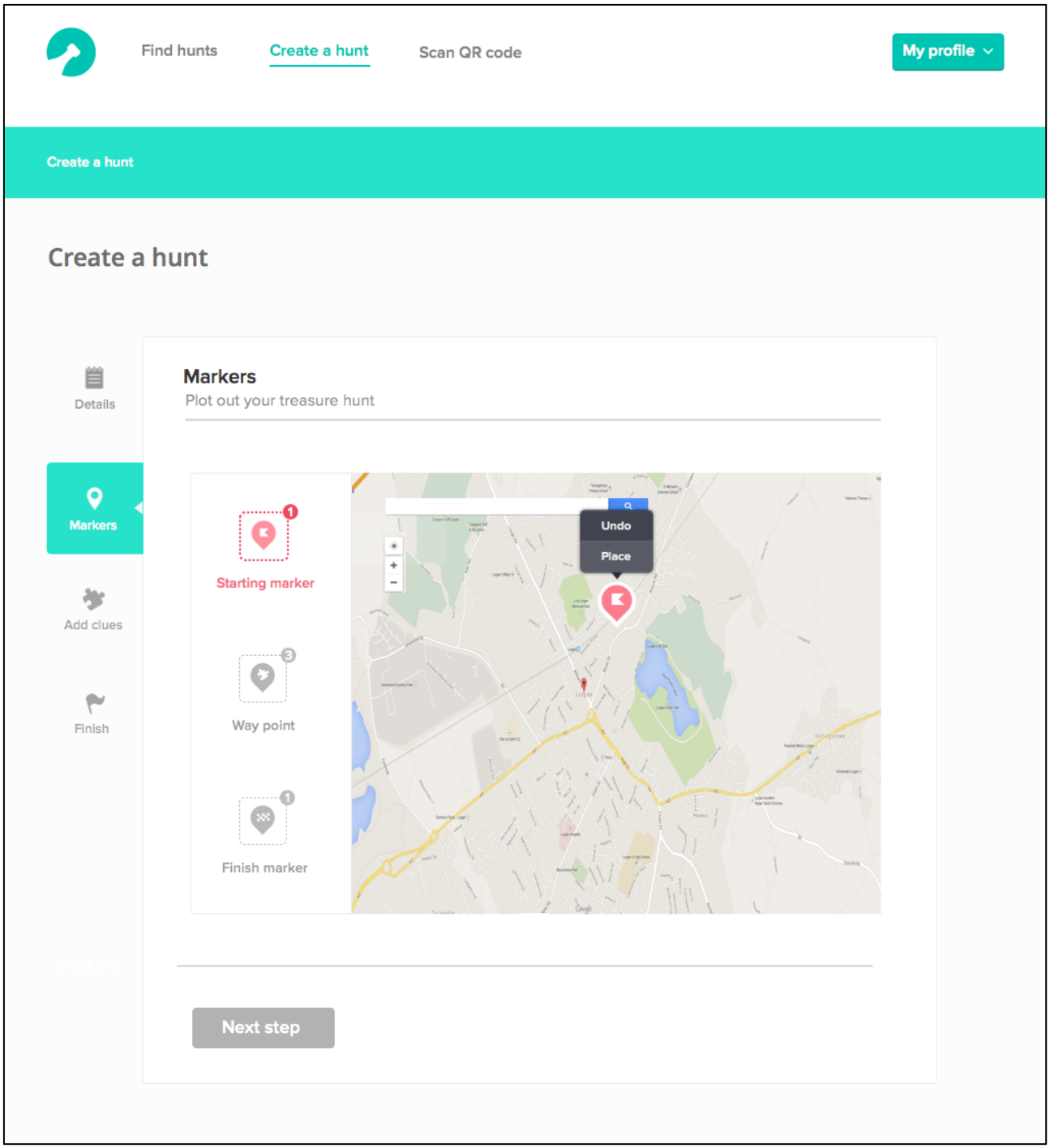


Figure 5.8 displays a refined screen from the 'create a treasure trail' user journey. This mockup allows users to simply drag markers from the provided map marker side panel onto the map. Once a marker has been dropped within the map, the user can simply drag the marker again to ensure the marker is in the desired location.

Please see appendix E to view additional refined UX screens.

5.2. System technologies

The following section describes the technologies that have been used during application development.

5.2.1. HTML 5

HTML (Hyper Text Markup Language) is a scripting language used within web development to develop web pages and web applications. Although HTML5 is not yet a released W3C standard, HTML5 is regualry used in modern web development. HTML5 provides a number of benefits over preceding HTML versions including additional web structural tags, which in practice will result in heightened web page semantics. Further benefits of HTML5 include a simpler method than it's JavaScript alternative in regards to capturing the user's location (Geo-location). The web development framework this project made use of (Foundation) relies on the usage of HTML5's DOCTYPE. Therefore it was essential that HTML5 was to be used to structure web pages during this project's development.

5.2.2. SASS

Syntactically Awesome Style Sheets (SASS) is a Cascading Style Sheets (CSS) complier that allows SASS authors to write more effective and more manageable CSS. Syntax for SASS and CSS are largely fairly similar, however SASS provides additional functionality that makes it superior. A competing CSS complier known as Less exists, however SASS currently provides a substantial amount of additional features than its competitor (Less).

5.2.3. Foundation

Foundation by Zurb is a web development framework, which provides pre-created essentials for creating modern web applications. The framework includes a responsive grid system, JavaScript based components such as modals, alerts, accordions etc. The usage of Foundation will increase the project's development speed due to required web elements being provided within the framework. Various competing web development frameworks such as Twitter's Bootstrap, Blueprint etc. do exist however background research has shown that Foundation provides modules / components which are easily customizable with SASS, the CSS complier in which this project relies on.

5.2.4. JavaScript & Jquery

JavaScript is a programming language most commonly used in conjunction with front-end web development. JavaScript provides the means of interactivity implementation through out modern web development. Jquery is a JavaScript based library, which provides a simplified approach to JavaScript programming. Both JavaScript and Jquery are used to develop usable and understandable web interactions through out this project.

5.2.5. Google Maps JavaScript API v3

Google Maps API is a well documented JavaScript based API that is used in conjunction with web applications to provide an interactive mapping component to the project. The Google Maps API provides the functionality necessary in order to create and visually portray mapping marker objects, map routing etc.

5.2.6. QR Scanner / Reader Module

In order for the system to successfully scan and read QR codes, a third party QR scanner had been implemented into the system. After research was conducted the QR scanner developed by Sangmin Shim [9] proved to be an effective module for usage and implementation within the project.

5.2.7. QR Creator Module

In order for the system QR reading / generation functionality to be application inclusive, a QR code constructor was needed. After conducting research a PHP based QR code generator was discovered and has since been used within the system [10].

5.2.8. PDF generator module

Upon creating a treasure trail a system requirement was to generate a printable document for real world environment placement. The PDF generator known as FPDF was found to provide functionality of dynamic PDF generation [11]. It has since then been implemented and used within the application.

5.2.9. Git & Github

Git is a lightweight Linux based version control system that works well with online repositories such as Github, Bitbucket etc. The usage of Git and Github provides version control and more flexibility to a project as well as providing an almost universal backup system to a project.

5.2.10. PHP & MySQL

PHP is a web scripting language that is both powerful yet easy to use, it has been used within this project to communicate with the database. Although there are various technologies which would achieve the same results, the likes of PDO (PHP Data Objects) have proven to be simpler to use and more efficient than PHP, the familiarity of PHP avoids the risk of development being stumped due to a steep learning curve. Lastly MySQL has been used to create the projects back end relational database. PHP is used in combination with the SQL based database in order to extract data from this database.

5.3. Design patterns

5.3.1. SMACCS

SMACCS is an abbreviation of Scalable and Modular architecture for CSS is a CSS architectural guideline for creating complex projects. SMACCS incorporates a programming practice of the separation of concerns while structuring / marking up web pages. By doing so the reusability of common design patterns used through out the project is ensured [12].

5.3.2. BEM

BEM is an abbreviation for Block Element Modifier. It is a naming convention practice to be used in combination with HTML and CSS. BEM attempts to put in place a much more consistent naming pattern which can easily be used in conjunction with CSS architectural guidelines such as SMACCS [13].

5.3.3. Grid system

Due to a wide range of devices capable of viewing web content it is important to implement and make usage of a responsive grid system. Although it is possible to develop a custom grid system, usage of the grid system supplied by Foundation ensures a high level of reliability. The usage of a custom grid system the project may suffer from slower development speeds due to the unnecessary time fixing potential issues and grid optimization.

5.4. Logic design

5.4.1. Database design & ERD

From the needs of the project's database extracted from the project's requirements we can assume the following is true:

- One user can play many trails (One to many).
- One user can create many trails (One to many).
- One trail can have many users (One to many).
- One clue object belongs to one trail (One to one).
- Many clue objects belong to one trail (Many to one).
- One trail can have many clue objects (One to many).
- One clue object has one QR code (One to one).
- One QR code identifies one clue object (One to one).

The relationships discussed above have been used to model the application's database. Figure 5.9 includes an Entity Relationship Diagram (ERD) portraying the design of the projects database.

Figure 5.9: Entity relationship diagram

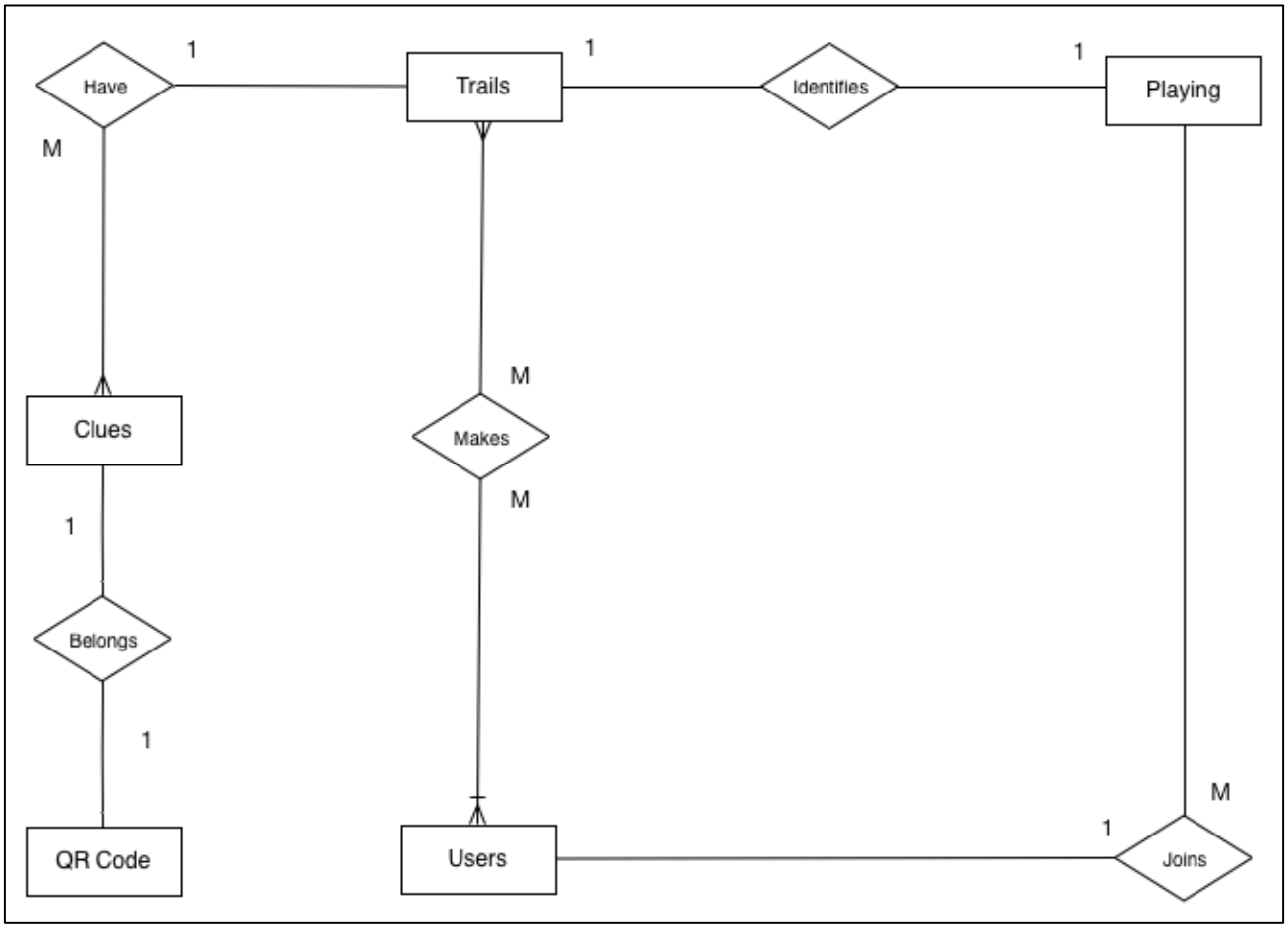


Figure 5.9 describes a total of five entitles that are used to model the project's relational database. These entities being 'Users', 'Trails, 'Playing', 'Clues', 'QR code'. Relationships are made and graphically portrayed in figure 5.9 which closely relate to the relationships defined in the proceeding section. Through the process of normalization, this database structure was created and utilized through out the system. Primary key's are used within each table entity in order to provide the means of uniquely identifying a row. While foreign keys constraints are used as means of creating relational links between one or more tables.

6. Implementation

6.1. Platform and tools used

The following section outlines both the hardware and software (development tools) that had been used in conjunction with the design and development of this project.

6.1.1. Hardware

Table 6.1 provides insight to the project's implementation environment.

Table 6.1: Project constraints

Type	Hardware
System	13" Macbook Pro running OX X Yosemite
System	IPhone six Plus running IOS 8
System	Ipad Air running IOS 8
Monitor	Apple 27" Thunderbolt display

6.1.2. Development tools

Table 6.2 lists the software tools used during project implementation and design.

Table 6.2: Project constraints

Type	Software	Usage
Code editor	Brackets 1.2	Creating and editing project programming code.
Design package	Sketch 3.3	UX design of the application.
Terminal emulator	Terminal	Communicating with Git / Github and SASS compiling.
Database management	PHP MyAdmin	Management of SQL database.
Version control	GitHub	Online hosted version control software.

6.2. Technical implementation

The following section describes how the technologies defined in section 5.2 – system technologies had been used during project implementation.

6.2.1. Google Maps API

The Google Mapping API is fundamental to the project's end project and is the means of which the user finds and creates a treasure trail. Within the application's 'find a treasure trail' functionality, pre-existing treasure trails are visually displayed via custom designed mapping markers placed on the map. Each treasure trail marker is positioned on the map relative to the treasure trails starting marker's latitude and longitude value (obtained during treasure trail creation). Doing so results in map markers accurately portraying the location of where the treasure trail begins.

The process of treasure trail creation makes use of the Google Maps Marker in a more complex manor by providing drag and drop marker functionality. To elaborate further, clue points are placed onto the map via a drag and drop interaction, which sees the user dragging markers from a HTML based element onto a Google Map object. Upon doing so, program logic is ran which calculates the DOM offset of the point in which the marker was dropped on and translates this point into a Google Maps latitude and longitude location value. This value is then easily be used by the Google Maps API to accurately position the resulting Google map marker.

Lastly throughout the application, functionality native to the Google Maps API has been used to portray a 'polyline' between trail points. By doing so the route between a treasure trail's points can be accurately visually portrayed to the user.

6.2.2. SMACCS Structure

The application made use of the SMACCS architectural system in order to design and implement a sustainable and maintainable architecture for the project's CSS. The version of SMACSS implemented was a slightly modified approach to the original concept defined by Jonathan Snook. This was due to the architectural needs of this project's CSS being not as complex as the architectural structure was originally intended for, [12]. All CSS based components where separated in four main components described below:

Base - The base folder holds base styles which where unlikely to change e.g typography, grid.
Modules - The modules folder holds modular styles for such components as custom divs, ordered and un-ordered lists etc.
States - The states folder controls the visual styles for components that may vary due to interaction state e.g a link on mouse-over.
Util – The utility folder houses useful utility functions such as clearfix, image text replacement functionality etc.

Each folder (listed above) includes an index file, which makes usage of SASS'S @import function to import all files within the folder's directory. At the root of the architectural system is an index file, which imports from each folder's index file. Doing so results in a filing system, which will compile SASS to CSS in a sequential order determined by styling importance.

6.2.3. QR Scanning / Reading / Generation

Whilst creating a treasure trail, functionality implemented allows for the generation of dynamic QR codes. Upon generation these QR codes are placed within the project's image folder directory. In order for this functionality to be possible, a third party module was implemented and used within the project [10]. The QR generator module was built and uses object orientated PHP to successfully provide a reusable function for means of QR code generation.

Upon generating a QR code, these QR codes can then be read by application's QR reader functionality. The implemented QR code reader module is built in JavaScript as opposed to the PHP based QR code generator. Alike the QR code generator, the QR code reader functionality was provided from another third party module [9]. This module has been used within the application via providing a facility in which the user (in combination with a camera), scans found QR codes. Upon scanning a QR code, program logic is ran to check if the found QR code resides within the database. If a QR code match is found the user is then offered the opportunity of viewing the found treasure trail clue.

6.2.4. PDF generation (FPDF)

Upon creating a treasure trail a PDF file is generated containing important treasure trail information. This functionality was implemented by making usage of a third party module known as FPDF [11]. FPDF is a PHP object orientated PDF generation tool, which provides a pre-made PHP prototype used to generate custom PDFs. Upon creating a treasure trail, further functionality

implemented within the application allows for re-generation of a treasure trail using both PHP and the FPDF module.

6.2.5. PHP & MySQL

PHP and MySQL where used through out the application as means of relational database communication. PHP Sessions have also been heavily used through out the application as means of transporting data from multiple pages. The application's architecture consists of many PHP files that had been used in a combination with AJAX to obtain and make usage of database information. Lastly whilst handling sensitive customer information such as their password, PHP based MD5 encoding had been used to ensure sensitive information is kept secure.

6.2.6. JavaScript & Jquery

Both JavaScript and Jquery where primarily used through out the application as means of front end interactivity development. Jquery based AJAX had been used through out the application in a combination with Jquery as means of form validation. AJAX was also utilized through out the application as means of obtaining information from the application's back end database. Jquery was then used to present this information to the user via DOM manipulation. Both the Google Maps API and the QR reader used within the application had been built using JavaScript. JavaScript was used as means of extending these technologies in order to develop the application's necessary functionality.

6.2.7. Foundation

Many features provided by Zurb's Foundation where utilized through out the project. The grid system provided by foundation was used to ensure the layout of the application adapts to the screen size / resolution of the viewing device. Foundation's alerts where used to provide visual feedback on user activity, such as the successful completion of treasure trail creation or an unsuccessful form submission. Modal's have been used within the 'create a treasure trail' user journey and the QR scanner functionality in order to unobtrusively obtain further information when needed. Lastly a collapsible navigation component provided by Foundation has been implemented to allow for viewing context specific navigation elements.

6.2.8. Git & GitHub

Git in combination with GitHub was used during project implementation to provide means of version control. Doing so provided the project with an almost universal backup system along with the ability to resort back to a previous development alliteration / version if needed.

6.3. Notable challenges

The following section discusses technical obstacles that where found during implementation and how they had been resolved.

6.3.1. Dynamic search radius

A component implemented into this product allowed the users to search for trails closest to their location via a radius search. The user could then modify the search radius by interacting with a range slider implemented into the UI. In order for the functionality of this component to be possible, a search filter object was set up which on default contains non-restrictive default search data. Interacting with the search filter UI component updated the searchOptions object, shown in figure 6.1.

Figure 6.1: SearchOptions object

```
9    //Set inintal search options
10   var searchOptions = {distance:20000,searchText:'',privacy:'Show all', results: 0};
```

As see in figure 6.1 the default search radius is initially set to '20000' which is equivalent to a 200-mile radius. Figure 6.2 displays a function that handled interactions with the distance radius slider.

Figure 6.2: Project constraints

```
    huntName.distanceFromUser = google.maps.geometry.spherical.computeDistanceBetween (marker.position,
huntName.position);
    //If the marker location is within searchoptions then show
    if(huntName.distanceFromUser <= searchOptions.distance){
        markersList.push(huntName);
        setResultMenu(markersList);
        removeMarkers(map);
    }
}
```

The Google Maps geometry library is included with the find trail page in order for the Google Maps API to have access to the 'geometry' function used in the above (figure 6.2) code snippted. The

geometry library is required in order to accurately compute a distance between the user's current location and a (or multiple) marker(s). Once an accurate distance is generated, each marker is only place upon the map if it's distance is less than the default search radius (200 miles).

Figure 6.3: Search radius change handler

```
//If the user changes data slider
$('[data-slider]').on('change.fndtn.slider', function(){
    var value = $(this).attr("data-slider"); //Get value
    searchOptions.distance = value * 1000; //Times by 1000 to = mile value
    circle.setRadius(searchOptions.distance); //Set the circle radius object to equal new distance
    filter();//Run the filter function
});
```

Upon interacting with the radius slider the logic portrayed in figure 6.3 would begin to run. Firstly the data-slider attribute of the slider component is selected and assigned to the variable 'value'. The searchOptions object displayed in figure 6.1 is updated to include the latest / updated search value. Before assignment this value is multiplied by 1000 to return a value that can be translated into miles. The Google Maps circle object (visual search radius) is then updated to reflect the changes.

Figure 6.4: Filter function

```
//Filters the search based on user interaction with input
function filter(){
    //Assigns filter values
    var text = searchOptions.searchText;
    var option = searchOptions.privacy;
    if(option == "Private"){
        option = 1;
    }else{
        option = 0;
    }
    showList = [];
    //Follwoing code checks for matches in text against all markers currently on display. Also checks
for privacy and distance from user
    var distance = searchOptions.distance;
    for (var i = 0; i < markersList.length; i++) {
        if(text != ""){
            if(markersList[i].title == text && markersList[i].private == option &&
markersList[i].distanceFromUser <= distance){
                showList.push(markersList[i]);
            }
        }else{
            if(markersList[i].private == option && markersList[i].distanceFromUser <= distance){
                showList.push(markersList[i]);
            }
        }
    }
    removeMarkers(null);
    showMarkers(map);
    setResultMenu(showList);
}
```

The filter function makes use of the searchOptions object displayed in figure 6.4 to limit the search results. In the context of the search radius, within the filter function's 'for loop', the currently displayed marker array (markerList) containing the location of treasure trails are looped through and calculations are performed to ensure that only map markers that have a distance less than the current search distance are pushed into the ShowList array. The showMarkers function and the setResultsMenu are then ran which feed from the ShowList array to append marker information to the DOM.

6.3.2. Generate dynamic PDF

Upon completion of creating a treasure trail the user is presented with a PDF containing information on their newly created treasure trail. This information is used for the physical placement of QR codes. Once a treasure trail had been created, the user can re-download their treasure trail PDF's from within their profile. The functionality that has been developed allows for dynamic generation of PDF's at any given point of application use. Figure 6.5 portrays a snippet of the functionality developed in order to do so.

Figure 6.5: PDF generator part one

```php
<?php //generatepdf_profile.php
require('../php_setup/config.php');
include(ROOT_PATH."php_setup/rnfunctions.php");
include(ROOT_PATH."php_setup/fpdf17/fpdf.php");

//Sets up a new connection the database
$con = mysqli_connect('localhost','root', 'root', 'bootily');
mysqli_select_db($con, "clues"); //Connects to the clues table
session_start();
$link = $_SERVER['REQUEST_URI'];
//Retrives the hunt ID from the url
$huntID = end((explode('.php/?', $link)));
$huntID = intval($huntID); //converts value to int
//Selects unqiue clueID from clues
$query = "SELECT DISTINCT clueID FROM clues WHERE huntID = '$huntID'";
$result = mysqli_query($con,$query);
//For statement selects all clueID's as query may select more than one
for ($i = 0; $i < mysqli_num_rows($result); $i++){
    $row[$i] = mysqli_fetch_row($result);
}
//Sets up the three clues and converts to int
$clueIDOne =  intval($row[0][0]);
$clueIDTwo =  intval($row[1][0]);
$clueIDThree =  intval($row[2][0]);
```

Figure 6.5 displays part one of the functionality implementation to create a dynamic PDF. Firstly a connection to the database is made and a PHP session is started. The Server's full URL is then stored within a variable named 'link'. The value succeeding the '.php/?' from within the link is then assigned to the variable 'HuntID'. Expanding on this further, when this component is called the treasure trail ID is passed into the URL e.g. bootily/generatepdf.php/?5. The number succeeding the '.php/?' refers to the huntID which is used to obtain information on the treasure trail.

A select SQL statement is then ran to select all treasure trail clue information by ID, data is then returned, processed and stored within the 'row' array. Each clue point ID is stored within the clueId One - Three variables and converted to interger values. A final Select statement is ran to select the treasure hunt name which is then assigned to the variable 'huntName'. Sessions are then created for usage within the FPDF (PDF generator) component.

7. Testing

7.1. Testing approach

Testing began by firstly testing that the project's end product met the needs of the functional requirements defined and discussed within chapter three. Please note, a full list of these tests can be found in Appendix C. The application was then tested with three testing participants. These participants where then required to fill out a short survey documenting their findings.

7.2. Requirements testing

Table 7.1 lists tests ran to ensure the project's end product has met the project's functional requirements. A full list of tests can be found in Appendix C.

Table 7.1: Requirements testing

Test ID	Req ID	Test	Expected result	Result
1	1	Application should allow users to create an account.	An account is created and added to the database.	Pass
2	2	Application allows users to log in.	User can log in with known authentication credentials.	Pass
3	3	Application allows for user sign out.	User can successfully sign out.	Pass
4	4	Application must be able to retrieve users location.	Application can successfully retrieve users location.	Pass
5	5	Application must be able to retrieve all treasure trails.	Application retrieves treasure trails from the database and visually displays these as markers on the map.	Pass
6	6	Application must allow for specific treasure trail search.	Usage of the search filter's search by name functionality will return a specific treasure trail.	Pass
7	7	Application must display treasure hunts trail to them.	Usage of the search radius slider should narrow down	Pass

			search results.	
8	8	Application must allow a user to see more detail on a specific hunt.	Application will provide details on a specific hunt.	Pass
9	9	Application must allow a user to join a specific hunt.	Clicking on the 'join hunt' button within a specific hunts detail page will ensure user is playing the hunt.	Pass
10	10	A user must be able to create their own treasure trails.	Create treasure trail step process will successfully upload a complete hunt to the database.	Pass
11	11	A QR code must be generated for each clue within a treasure trail.	QR codes generated are unique to the specific clue and placed in database.	Pass
12	12	QR codes must be printable	QR codes are successfully generated within a PDF format.	Pass
13	13	QR scanning facility is included with the system.	A functioning QR code scanner is implemented with the system.	Pass
14	14	Scanning a QR code must update the user's progression within a specific treasure trail.	QR scanned will unlock specific clue within a treasure trail.	Pass
15	15	User progression in conjunction with treasure hunts is stored.	Clues found by a user will be updated within the database.	Pass
16	16	Users must be able to upload a photo to their profile.	Photo uploading is successful.	Pass
17	17	Users must be able to update their personal information.	Users can provide and update their description, location, twitter and Facebook link.	Pass

As seen from table 7.1, the project's requirements were successfully met by the project's end product.

7.3. Cross browser testing

Due to the fact that modern web applications may be used on a varying combination of devices and operating systems, the application was tested across a few of the more popular web browsers on various platforms. Table 7.2 lists the cross browser testing results.

Table 7.2: Cross browser-testing results

Testing device	Browser used	Result of test	Result
Macbook Pro	Chrome	Application works as expected.	Pass
Macbook Pro	Firefox	Application works as expected.	Pass
Macbook Pro	Opera	Application works as expected.	Pass
Macbook Pro	Safari	QR Scanner seems to not function correctly.	Fail.
Ipad Air	Safari	QR Scanner seems to not function correctly. Minor CSS issues present.	Fail.
Ipad Air	Chrome	Application works as expected. Minor CSS issues present.	Pass
IPhone	Safari	QR Scanner seems to not function correctly. Minor CSS issues present.	Fail

From the results portrayed in table 7.2 it was apparent that the QR scanner module was having an issue in combination with Safari. It was also apparent that the system suffered from CSS based issues in relation to smaller mobile devices. These issues are discussed and investigated further within section 7.5 – technical issues.

7.4. Usability testing

Three participants where assigned with the task of testing the system. These test's where ran to make sure the product was usable and to spot any potential technical issues that where not found during previous tests. Upon testing the system, participants where asked to fill out a questionnaire based on developed system. Please see Append E to view these questionnaires.

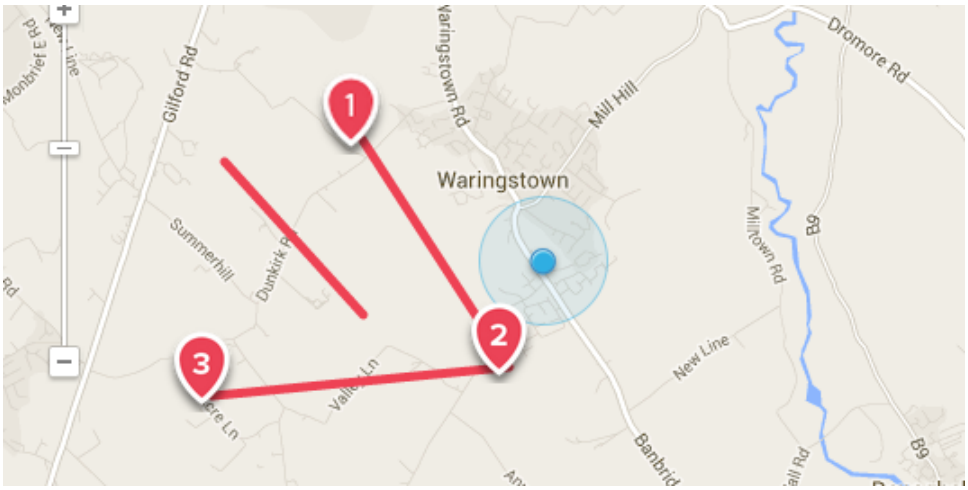
The overall feedback from participants was positive and participants enjoyed the experience provided. The participants did however note a few technical issues. Please see the following section for further details.

7.5. Technical issues

7.5.1. Drag markers issue

An issue which resides within the 'create a treasure trail' functionality shows the Google maps polyline object not correctly resetting the route when a marker is dragged upon placement of all markers. Strangely, the issue only seemed to occur once, as functionality seems to run normally once the issue has occurred. A screenshot portraying this issue is provided in figure 7.1.

Figure 7.1: Drag marker issue



Several attempts where made in order to resolve this issue however due to unfortunate time constraints placed on the project's completion, this issue still exists within the end product. The issue is occurring due to the initial polyline connection still existing within the array object this polyline route is formed from upon further drag events. A hard reset of this array has been and implemented within the program's logic however doing so does not fix the error. As the project's developer I believe that a fix to this issue would require this function to be re-written.

7.5.2 QR browser compatibility issues

Noted by participants during system testing and found whilst cross browser testing, the QR scanner facility seems to fail to request camera access permission while browsing in Safari. Without having

access to the camera, the QR scanner module can and does not function. Further investigation of this issue indicated that the cause of this issue originated from a recent update in the QR reader's library. It is likely that in the near future a fix will be released which will easy be easily merged into the system to address this browser compatibility issue. The reliance on third party modules and API's unfortunately results in the product's stability and functionality being at the mercy of the stability of these third party modules.

7.5.3 CSS issues

Due to browsers representation of DOM slightly differs from one browser to another, padding / margin defects are present. Issues found are minor and do not effect the application's ability to function correctly by any means. An attempt to fix these errors has managed to resolve the majority of CSS error's found however due to the time constraints in place, it was simply not feasible to fix all issues. As a result several minor CSS based issues still exist within the project's end product.

7.5.4 Navigation highlighting / Breadcrumbs

Due to the project being developed on a local server, when moved onto the University's server the navigation and breadcrumbs have not translated well. The breadcrumbs had been developed as folder root relative, resulting in breadcrumbs items feeding or obtaining content from the project's folder directory / structure. The project's original implementation environment was restricted to being placed within a single folder, named bootily (project's branding). Therefore whilst in development, breadcrumb list items would begin with 'Bootily'. When the project was moved to the University's server, the folder structure differs significantly, as all University projects are to be placed within a root folder named 'mp'. In the current state of the breadcrumbs component, breadcrumb items feed from and output the University's server folder structure.

As a result of this problem, the navigation-highlighting feature cannot effectively pick up the user's current page location. Therefore resulting in the application failing to visually highlight the user's current location. A fix for this feature would require a re-arrangement of folder structure, which unfortunately is not recommended by the University. Unfortunately due to project time constraints, further work in fixing this issue was not feasible and as a result this issue is still present.

8. Evaluation

The following section evaluates the technologies used through out the project's implementation along with an evaluation of the user testing results and the project's methodology.

8.1. Technology evaluation

8.1.1. Google Maps API

The Google Maps API was found to be effective in providing highly usable mapping functionality throughout the application. The API provided the ability of easily removing UI components that I did not feel where essential. The API also successfully allowed for the usage of custom designed map markers to aid in the portrayal and emphasis of the application's branding. The documentation provided with the API was found to well documented and easy to understand. Due to the popularity of the API, Google Map API related resources where widely available on the Internet, which helped in developing a solid understanding of the API. It was found that the API's map initialization times could sometimes be lengthy and consume large amounts of bandwidth. On further investigation the API's load time varied significantly depending on Internet connection method. This is obviously due to the complexity of the API itself and something that I am sure Google are continuously working on to heighten the API's efficiency.

8.1.2. SASS and Foundation

The usage of SASS in conjunction with this project's development resulted in an increase of development speed. Upon reviewing the complied CSS file, it was evident that the CSS output was much more DRY (Don't repeat yourself) and efficient as opposed to a non-SASS generated file. SASS was an excellent choice to use during this project's development.

The Foundation framework used in conjunction with the project's development further aided in speeding up the project's development time. Many pre-made components of the framework where used during development - modals, alerts, drop downs etc which where found to be highly customizable. The grid system included with the framework was found to be quite restrictive however and generally inflexible to the project's layout needs. In future projects it may be wise to design, develop and make usage of a custom grid more suited to the product.

8.1.3. Git and Github

Git used in a combination with Github proved undoubtedly beneficial during project development. Git allowed for development progress to be easily tracked and if needed, the project could default to early revisions, which in multiple times in this project's development saved time removing non-functioning logic. GitHub provided a universal backup system that gave comfort in knowing that if the project was to somehow become corrupted, a backup version was at all times available and downloadable at relative ease.

8.1.4. PHP & MySQL

PHP as a scripting language was found to be both powerful and frustrating at times. PHP offered such features as Sessions, which allowed for the development of the application's multi-page process. Its syntax is both highly usable and legible, and makes for a great programming experience. However unlike other programming languages if an error occurred PHP refuses to provide an indication of where this error may be occurring, resulting in frustration. PHP's compatibility and communication with the project's MySQL relational database however was made simple. Despite the issue briefly discussed above, the usage of PHP and MySQL was relatively straightforward, however in hindsight the usage of PDO may of brought additional benefits to the project's development.

8.1.5. QR Scanner module

Although I am slightly disappointed in this module's recent update breaking functionality within Safari, this module was well constructed and functions effectively. Modifying the file had proven to be quite difficult as the programming practices implementing in within module where not easily understandable. Whilst modifying the module I found myself logging out many variables to Google Chrome's developer tools in attempt to understand various functions. In regards to the modules functionality, the threshold in which the scanner functions was found to be quite narrow. It would be beneficial to see further updates of this module expanding the scanner threshold and perhaps implementing better programming practices.

8.1.6. PDF generator module

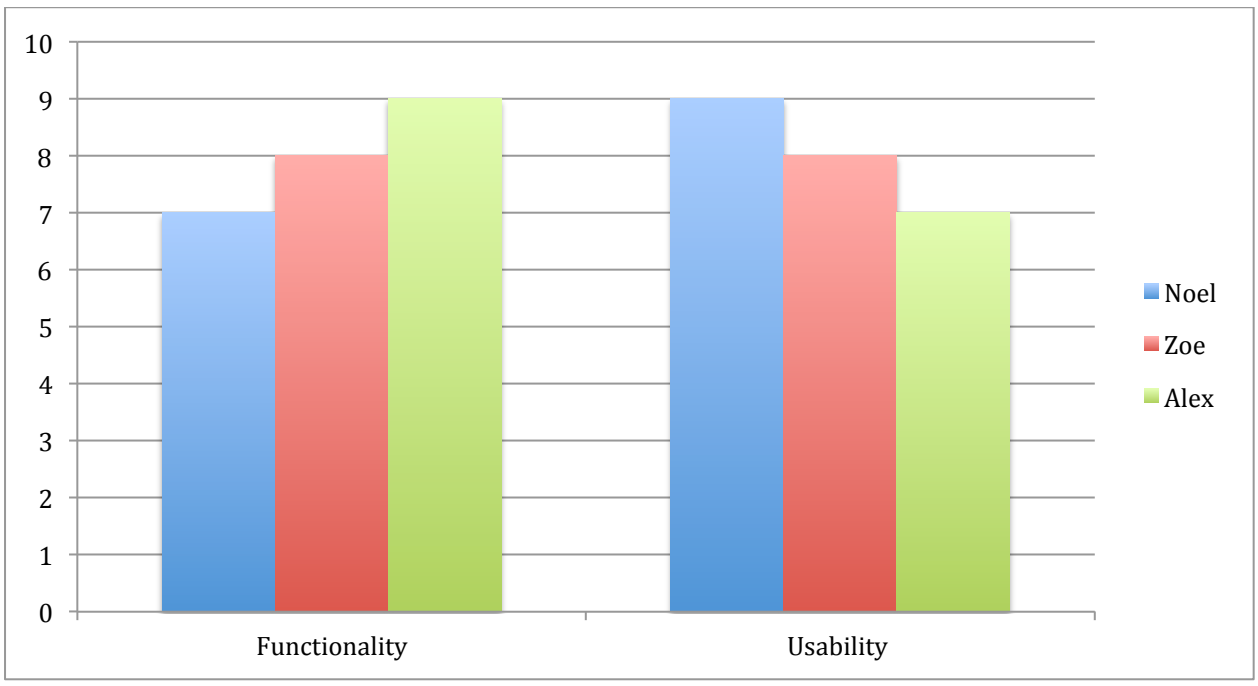
The PDF generator module (FPDF) [10] effectively provided the functionality needed to generate PDF output. The module itself is powerful and thus seemed quite complex at initial glance however the provided documentation effectively explains the module's usage in great detail. Overall I found

FDPF relatively easy to work and as a result I will be using this module again if future projects require such functionality.

8.2. User testing results

Testing performed within chapter seven has successfully proven that the completed solution met each defined project requirement, thus proving that implementation of the product had been a success. Although functionality and styling issues had been found, user feedback retrieved from questionnaires (see Appendix E) proved that the application was usable and again proved that functionality requirements had been successfully met. Within the questionnaires provided to testing participants, participants where requested to provide a rating (1-10) of both the products usability and functionality. The findings of these questionnaires have been analyzed and are portrayed in the following graph (figure 8.1)

Figure 8.1: Usability / Functionality results



As you can see from the results displayed in figure 8.1, users rated both the overall functionality and usability of the application highly. The markings measured are scaled one through to ten, One representing a negative view and ten representing a positive view on the application's functionality / usability. Interestingly the results shows that on average both the usability and functionality of the

system were equally balanced. Thus the results from this graph indicate that the project has been successful in meeting the user's usability and functionality expectations of the application.

8.3. Methodology evaluation

The waterfall methodology ensured steps required to progress in development where completed fully before continuing to the succeeding step. Thus resulting in time and effort being spent in areas that may have been rushed through if a non-sequential methodology had been utilized. Using the waterfall methodology allowed the application's User interface design to be fully developed and explored, before continuing onto development, which thus speed up development time. A criticism of the methodology is that the testing stage proceeds only when the entire application has been developed. Doing so resulted in a wealth of technical issues needing fixed. During testing of this application it was found that some technical issues would have been easier to fix within earlier stages of development. By using an iterative methodology such as RAD, testing would been conducted regularly. Thus eliminating the majority of technical issues through out the project's development. In saying so, the amount of technical issues within a project is relative to the skill of the developer. As I become more skilled in web development I may find a significant reduction in technical / logic errors regardless of methodology utilized.

8.4. Project outcomes

Within chapter one of this report, the project's objectives had been defined. The following table analyses these goals to test if the completed project had successfully met these goals.

Table 8.1: Project goals

Goal ID	Description	Result
1	To identify the user's of this project.	Pass
2	To extract user and project needs from these users.	Pass
3	To analysis user needs and obtain technical needs of the application.	Pass
4	To research and enhance my own knowledge of the application.	Pass
5	To design a testable paper prototype.	Pass
6	To test the products feasibility using a developed throw away applications.	Pass

7	To design and develop a functional application	Pass
8	To test and eliminate bugs within the product	Fail
9	To evaluate and reflect on the completed product	Pass

As portrayed in table 8.1 the developed end product has successfully resulted in meeting the majority of product goals. However during the testing phase of the project a wealth of issues where discovered and with project time constraints in place I did unfortunately not have the time to resolve all issues. As mentioned above, the majority of these issues could have been fixed easily and in a more timely fashion during early development as opposed to development completion. However due to waterfall methodology, testing was to be conducted upon project completion. With the complexity and ambitiousness of the project I am exceeding proud of what has been created, regardless of existing technical issues.

9. Conclusion

The following section concludes the work undertaken and suggests plans for the future of this application.

9.1. Summary

Upon project completion, despite technical issues present within the system, it can be said that the project has been a success. Playing many roles during the development of this project I was able to successfully gather, analysis project requirements and develop these into a working solution. I am exceeding proud of the completed project's complexity and scale and I'm eager to continue further development of this project. The following section outlines my main achievements achieved during development of this project.

9.2. Achievements

9.2.1. Time management

The project as a whole required countless hours being spent on research gathering, design and development, which was restricted to a 20-week deadline. In order for this project to be successful my time had to be effectively managed which the project's completion can prove I was successful in doing so. I was able to effectively prioritize work and thus focus solely on the more important tasks before moving onto additional tasks.

9.2.2. Technical ability

Prior to this project, my technical knowledge in various web technologies utilized throughout this project such as the Google Maps API was next to non-existent. During development of this project I was able to effectively research and implement complex program logic in conjunction with various API's and modules I was unfamiliar with. Being tasked with the usage of a range of unfamiliar technologies, I found that I was able to quickly develop an intermediate understanding of these technologies. Doing so has heightened my confidence in my own programming ability and understanding of web development in general. From completing this project I am now confident in

my working knowledge of modern web development technologies and will continue to work these technologies utilized within the future.

9.3. Further work

Although this project was developed as a web application, in hindsight this project may of been better suited for mobile device implementation. However due to a limited knowledge on IOS software development it was advised that this project was to be developed using modern web technologies. The product that has been created, I believe is a great Minimum Viable Product (MVP) and great proof of concept. I believe that there is potential for this product to become somewhat of a product that many use and enjoy, however in order to reach that potential the product needs to be ported into the context in which it will be used – mobile.

My future plans for this project is to redesign the product for IOS implementation. Doing so will give the project access to additional useful potential functionality. Apple's IPhone for example make use of a magnetometer and an accelerometer, which would provide the project with access to such on-board features as an accurate compass. Such a feature could be used to develop a much more interesting manner of finding treasure trail clues. Features such as achievements are planned to be included to reward the user for playing treasure trails in order to provide reasoning for continuous use. I will begin to delve into Objective C programming and in the near future hopefully deliver a commercial IOS application that I can continue to literate and be proud off.

References

[1]Nitendo (2013). The Legend of Zelda: The Winder Waker.

[2] Jonathan Rasmusson. (2015). *Agile in a nutshell.* Available: http://www.agilenutshell.com. Last accessed 22nd April 2015.

[3] ISTQB Exam certification (2014) What is waterfall model . Available: http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/. Last accessed 18 November 2014

[4] ISTQB Exam certification (2014) What is prototype model . Available: http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/. Last accessed 18 November 2014

[5] ISTQB Exam certification (2014) *What is agile model*. Available: http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/. Last accessed 18 November 2014

[6] Wikipedia. (2015). *QR code.* Available: https://www.google.co.uk/search?client=safari&rls=en&q=pdf&ie=UTF-8&oe=UTF-8&gfe_rd=cr&ei=0Ps3VeHUCYu7-QaphICQAQ#safe=off&rls=en&q=qr+code. Last accessed 22nd April 2015.

[7] Ethan Marcotte. (2010). *Responsive web design* . Available: http://alistapart.com/article/responsive-web-design. Last accessed 22nd April 2015.

[8] Balasmiq. (2015). *Balasmiq Mockups.* Available: https://balsamiq.com/products/mockups/. Last accessed 23rd April 2015.

[9] Sangmin Shim. (2015). *Qrcode.js.* Available: http://davidshimjs.github.io/qrcodejs/. Last accessed 23rd April 2015.

[10] Dominik Dzienia. (2015). *PHP QR Code.* Available: http://phpqrcode.sourceforge.net. Last accessed 23rd April 2015.

[11] FPDF Library. (2015). *PDF generator.* Available: http://www.fpdf.org. Last accessed 23rd April 2015.

[12] Jonathan Snook (2014). Scalable and Modular Architecture for CSS. : . p2-199.

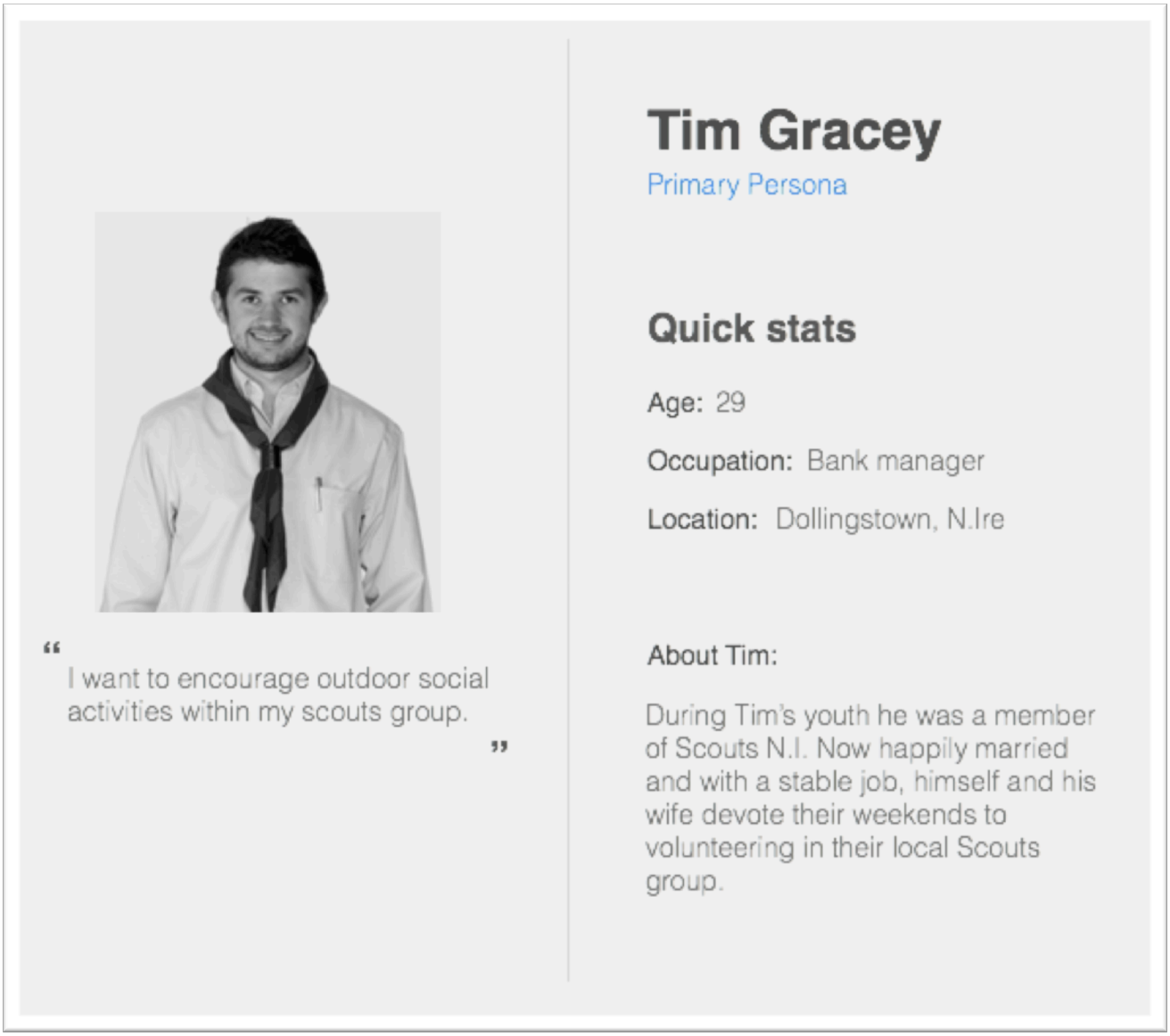
[13] BEM. (2015). *Block Element Modifier* . Available: http://bem.github.io/bem-method/html/all.en.html. Last accessed 23rd April 2015.

[14] Username - Dr Mollie. (2014). *Drag & Drop an object into Google Maps.* Available: http://stackoverflow.com/questions/18912744/drag-drop-an-object-into-google-maps-from-outside-the-map-marker-not-put-at. Last accessed 24th April 2015.

[15] Wikipedia. (2015). *Persona (user experience).* Available: http://www.neilstoolbox.com/bibliography-creator/reference-website.htm. Last accessed 30th April 2015.

Appendices

Appendix A – Primary user Personas



"I enjoy seeing students off my class working together."

Annie Tom

Secondary Persona

Quick stats

Age: 30

Occupation: Georgraphy teacher

Location: Portadown, N.Ire

About Annie:

Annie is passionate about teaching georgraphy. Yearly she teaches her pupils about maps and compasses and enjoys her yearly orienteering outing with her pupils.

“I enjoy creating and sharing digital content.”

Paul Esler

Primary Persona

Quick stats

Age: 12

Occupation: Student

Location: Lisburn, N.Ire

About Tim:

Paul is a fan of the latest technology and enjoys playing with applications on his Iphone. He enjoys applications that enable him to be creative and create content for others to enjoy.

Appendix B – Secondary user Personas

"I want to encourage my kids to understand that their is more to do with their free time than play video games."

David O'Reilly

Secondary Persona

Quick stats

Age: 45

Occupation: Taxi driver

Location: Lisburn, N.Ire

About David:

David is a hardworker and rarely takes times off. When he does he enjoys spending his time with his two kids. He has found recently that they have became more intrested in video games than they have in David's favourite past time - camping.

Appendix C – User requirements (1-19)

Requirement ID - #1

Type - User authentication

Description - The product shall allow users to create an account.

Rationale - To allow users to enjoy a personal experience with the product.

Fit criteria - In order for the user to interact with the product they must create an account.

Dependencies -

Priority - 10

Requirement ID - #2

Type - User authentication

Description - The product shall allow users to log in.

Rationale - To allow users to view their account.

Fit criteria - In order for the user to interact with the product they must log in to account.

Dependencies - #1

Priority - 10

Requirement ID - #3

Type - User authentication

Description - The product will allow user to sign out of an account.

Rationale - To allow users to sign out of their account.

Fit criteria - For security reasons, the product must allow users to log out of an account anywhere from any page whilst using the product.

Dependencies - #1, #2

Priority - 10

Requirement ID - #4

Type - GPS

Description - A product must be able to accurately retrieve the user's location.

Rationale - To retrieve localized results.

Fit criteria - In order to provide localized data the product will need to know the location of the user.

Dependencies –

Priority - 8

Requirement ID - #5

Type - Functional

Description - A product must be able to retrieve a list of user produced courses.

Rationale - To provide orienteering courses to the user.

Fit criteria -Users may be using the product to look for orienteering courses near by to them.

Dependencies - #6

Priority - 9

Requirement ID - #6

Type - Functional

Description - The product must allow users to search for specific hunts.

Rationale - To provide the tools needed to find a specific hunts.

Fit criteria - Product users may be looking for a particular hunts, search functionality must be provided.

Dependencies - #7

Priority - 10

Requirement ID - #7

Type - Functional

Description - The product must provide hunts near to the user.

Rationale - Users may be simply exploring the application to find hunts near to them.

Fit criteria - Product users may be exploring the applications treasure trail list, therefore it makes sense to order these in terms of proximity.

Dependencies - #6, #7

Priority - 9

Requirement ID - #8

Type - Functional

Description - The user must be able to view a hunt.

Rationale - Users will want to know more details on a specific hunt.

Fit criteria - Once a user has found a hunt, they must be able to view the course to retrieve more details.

Dependencies - #6

Priority - 10

Requirement ID - #9

Type - Functional

Description - Users must be able to join a hunt.

Rationale - Once a hunt has been found a user may want to participate in a hunt.

Fit criteria - Once a user has found a hunt the user they may want to continue by playing this hunt.

Dependencies - #6, #10

Priority - 10

Requirement ID - #10

Type - Functional

Description - Users must be able to create their own hunts.

Rationale – Users will want to design and create their own treasure hunts.

Fit criteria – Users will have the end goal of creating their own treasure hunt.

Dependencies - #15, #16, #17

Priority - 10

Requirement ID - #11

Type - Functional

Description - QR codes should be generated for each clue point.

Rationale - Participants must be able to complete hunts.

Fit criteria - Orienteering planners will need an object participants can interact with.

Dependencies -

Priority - 10

Requirement ID - #12

Type - Functional

Description - QR codes must be printable.

Rationale – Treasure hunt creators will want to print and place marker QR points.

Fit criteria – Treasure hunt creators will need QR codes to be printable.

Dependencies -

Priority - 10

Requirement ID - #13

Type - Functional

Description - Scanning facilities must be provided within the application.

Rationale - Tools the user needs must be easily accessible.

Fit criteria - Having to rely on third party applications to complete a task that could be completed by one application can damage the products credibility.

Dependencies -

Priority - 10

Requirement ID - #14

Type - Functional

Description - Scanning a QR code must update users progression within a course.

Rationale - QR scanning is required for the application.

Fit criteria - Once a user has scanned a QR code this information will need fed back to the system.

Dependencies -

Priority - 10

Requirement ID - #15

Type - Functional

Description - User progression is stored.

Rationale - As the user participates in a treasure hunt there progression must be stored.

Fit criteria - In order to keep track of treasure hunt completion, user activity will need to be stored.

Dependencies -

Priority – 9

Requirement ID - #16

Type - Functional

Description - Users must be able to upload a photo to their profile.

Rationale - Users will want to customize their account.

Fit criteria - Photos are to be used as avatars within the product, users must be able to upload a photo.

Dependencies - #1

Priority - 7

Requirement ID - #17

Type - Functional

Description - Users must be able to update their personal information.

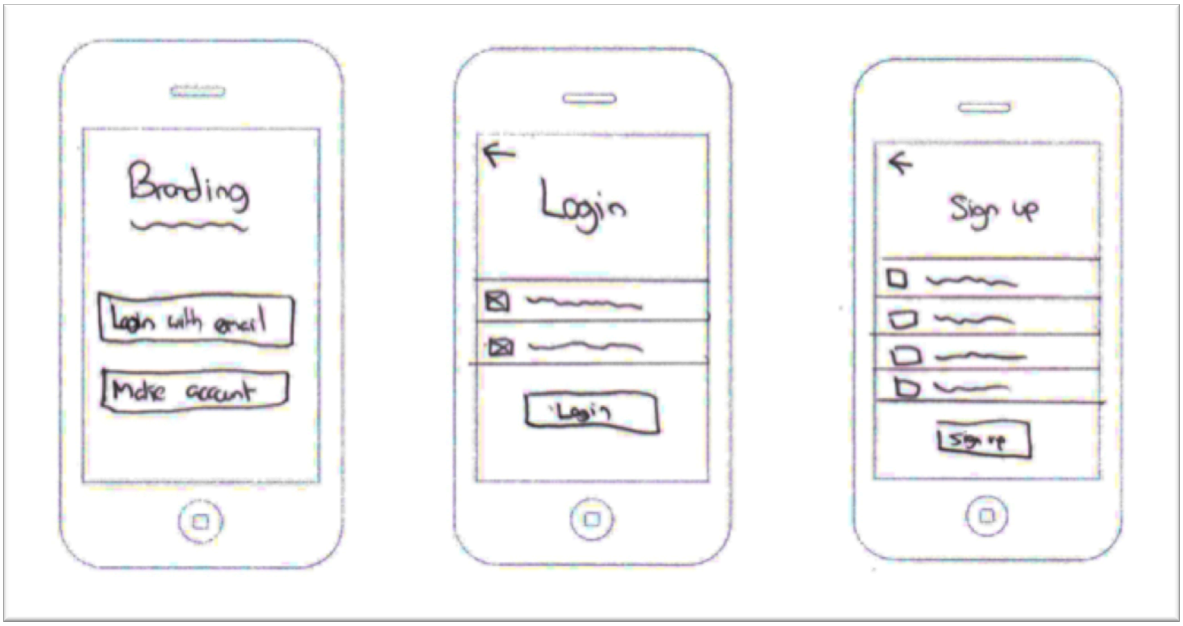
Rationale - Users will want to customize their account.

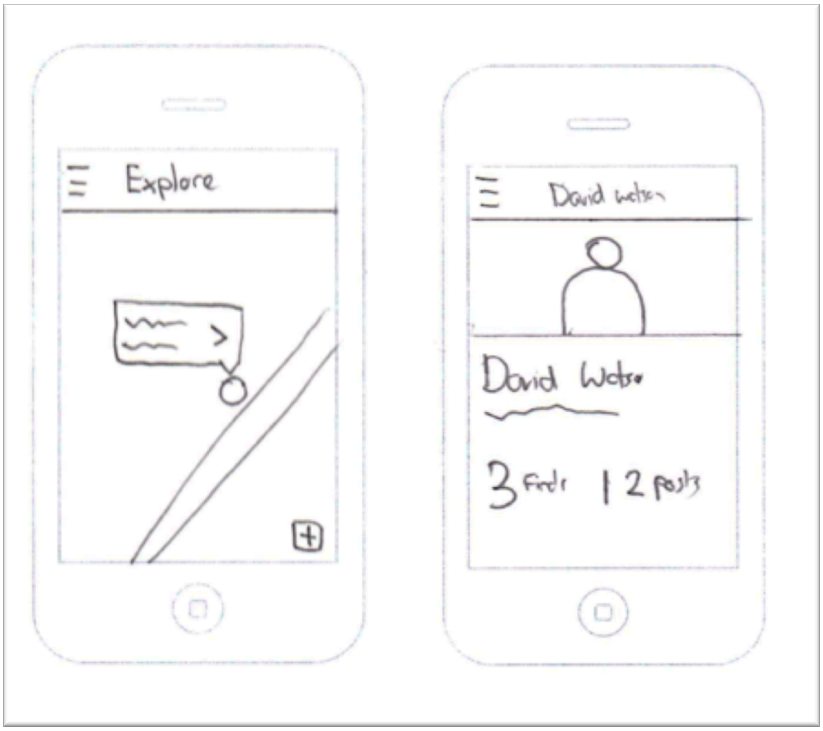
Fit criteria - Users must be able to update their account information.

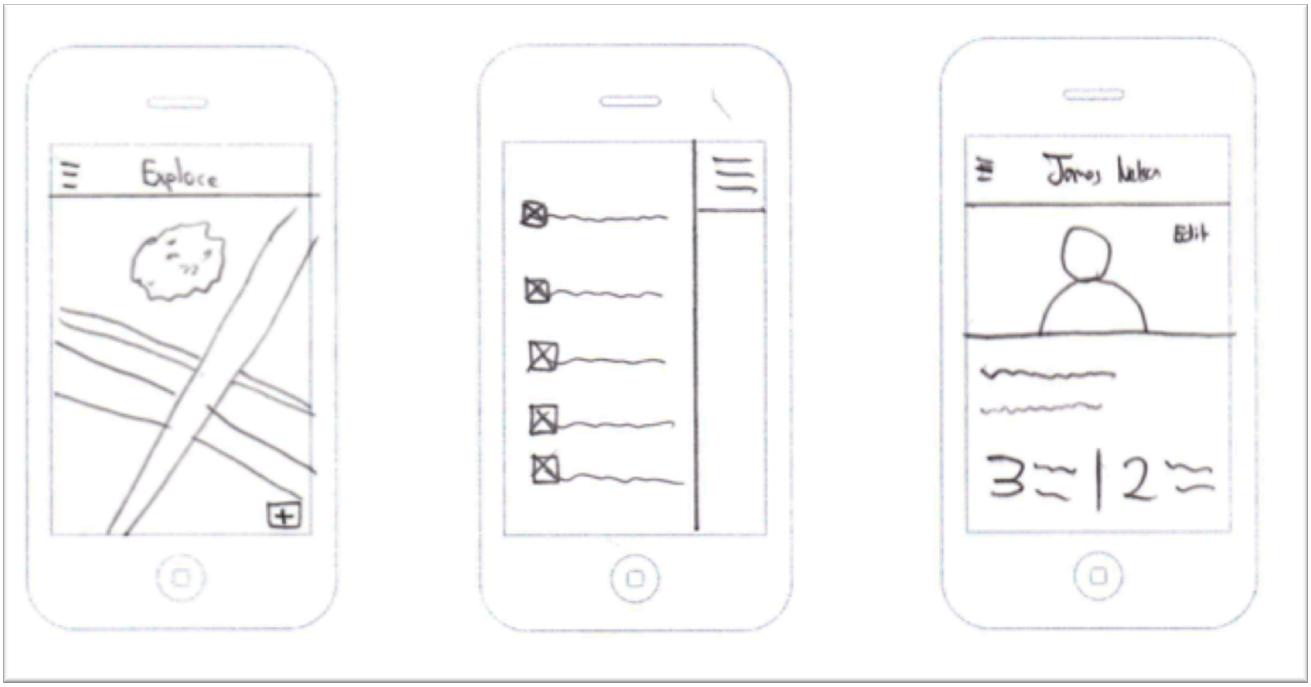
Dependencies - #1

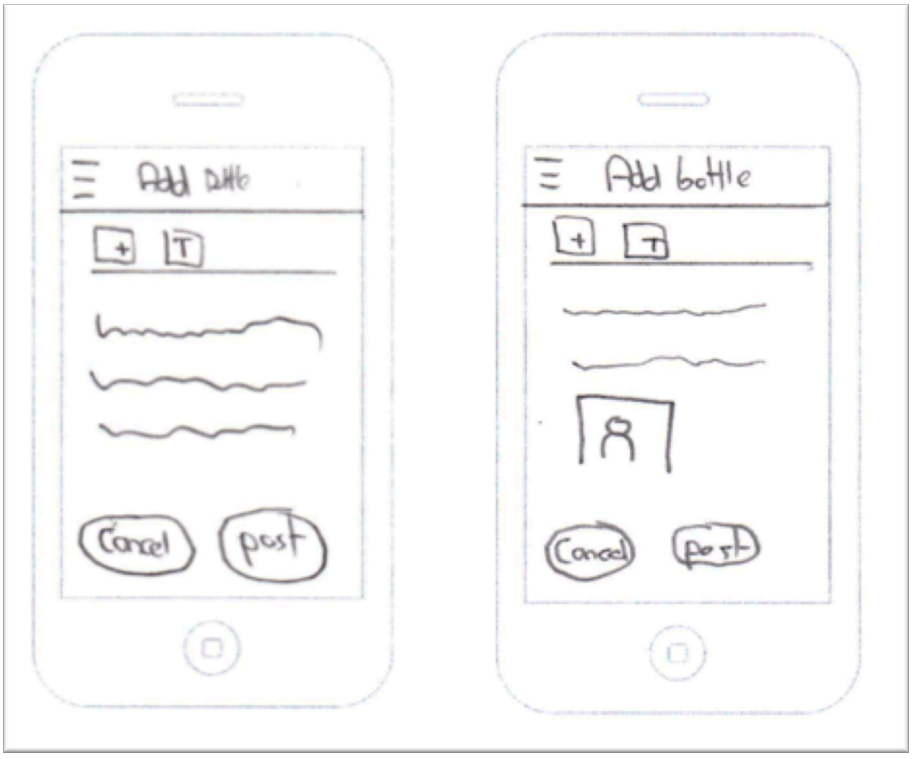
Priority – 5

Appendix D – Paper prototypes

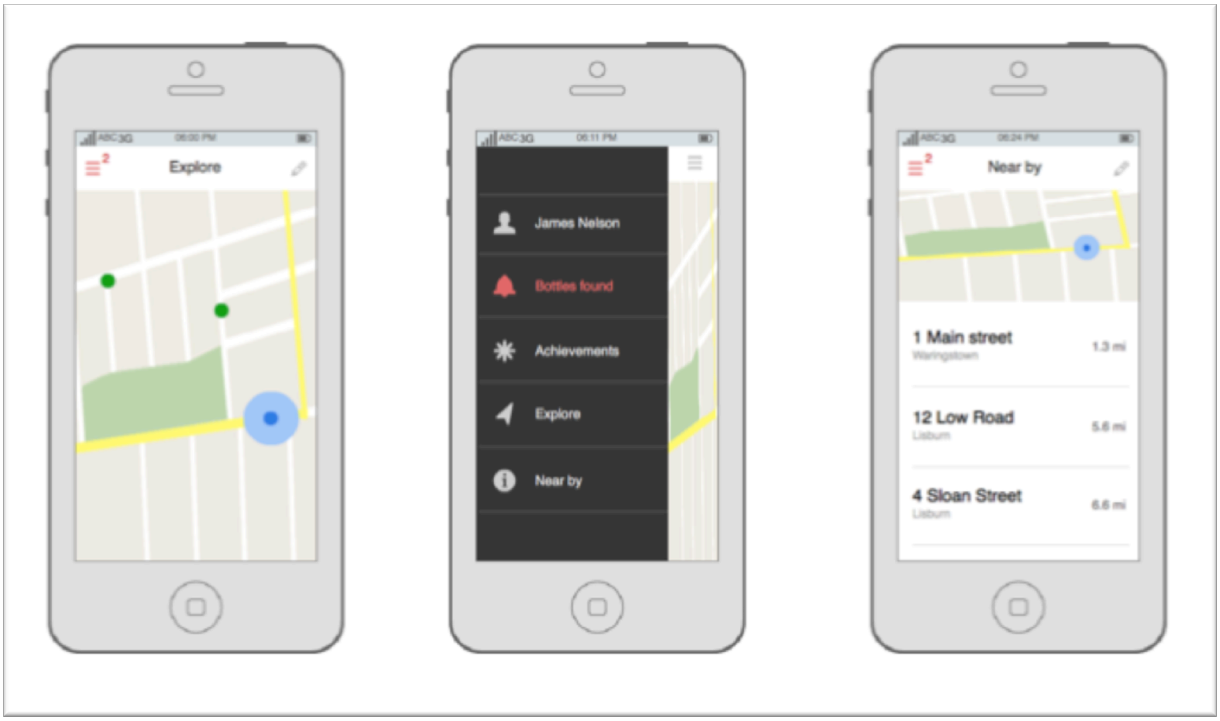


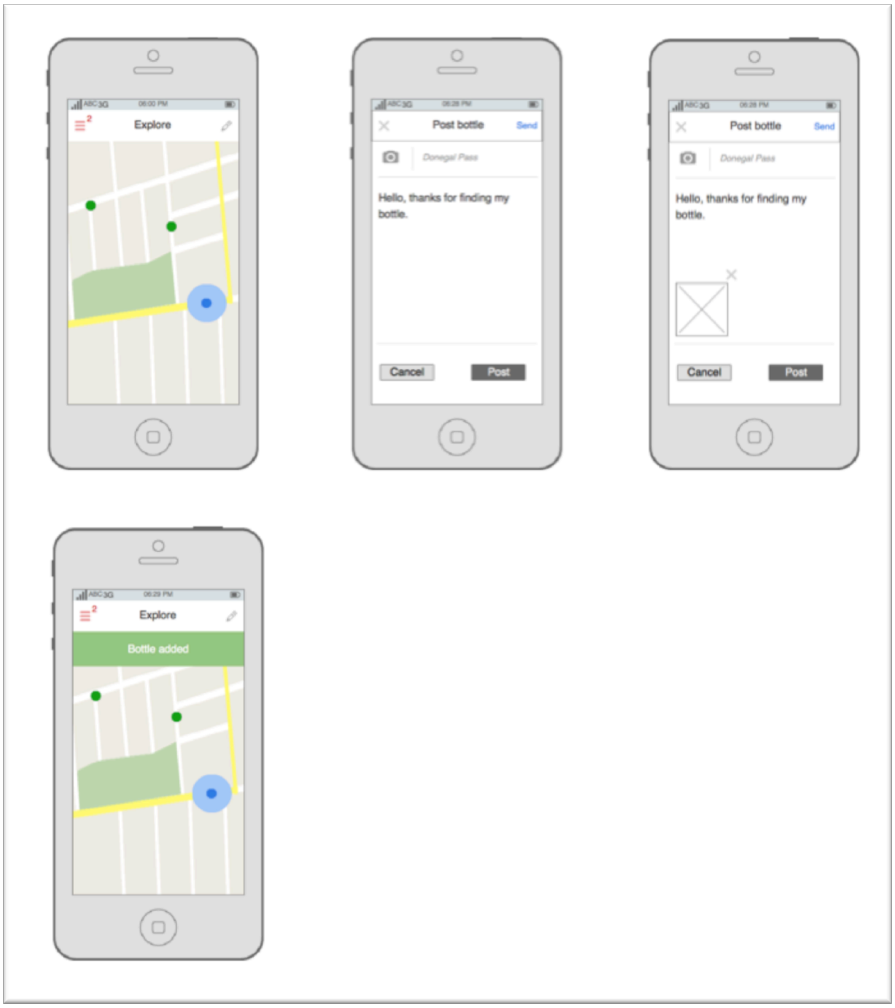



Explore
Edit
3
2

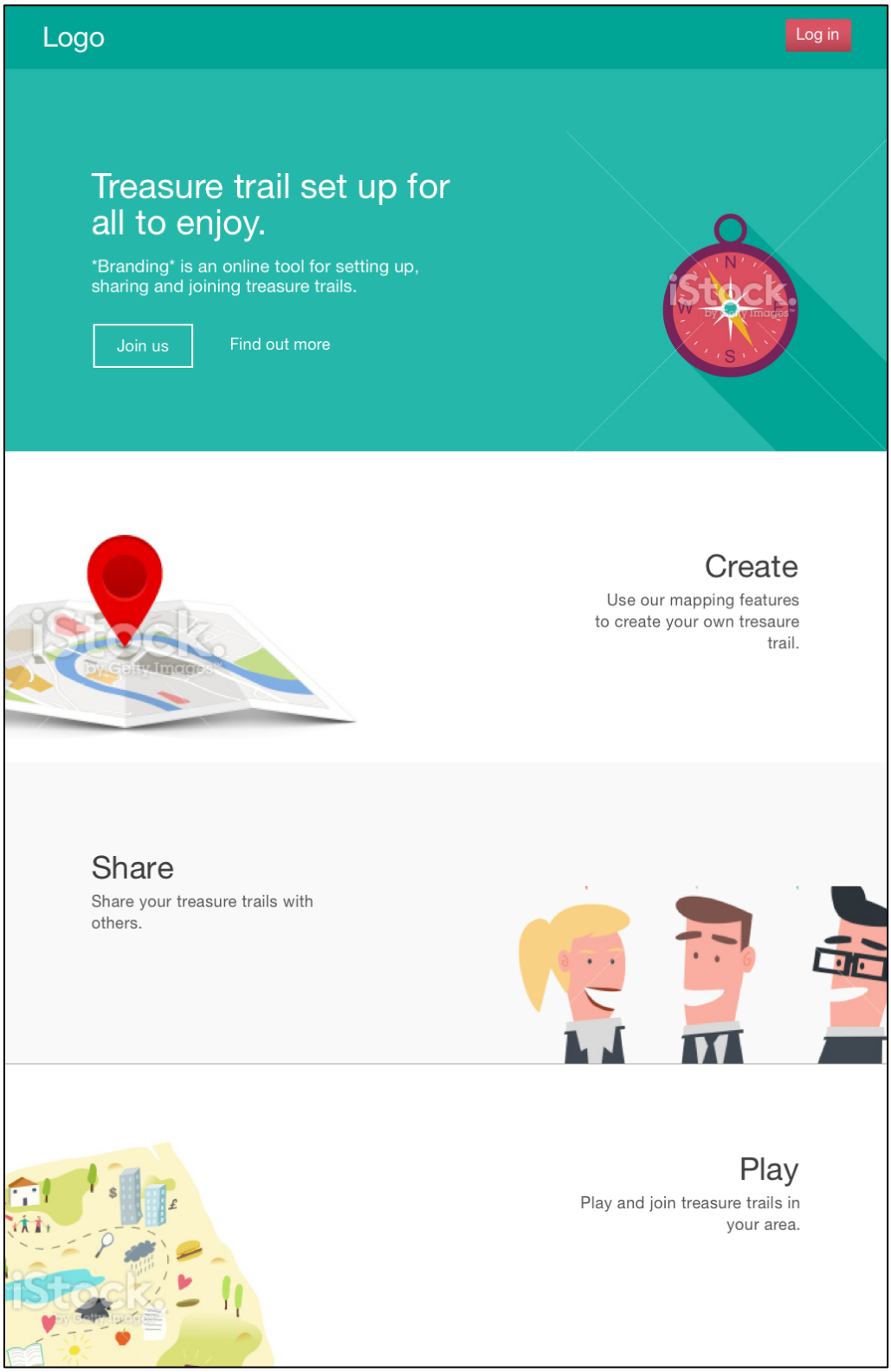

Add bottle
Cancel
post
Add bottle
Cancel
post

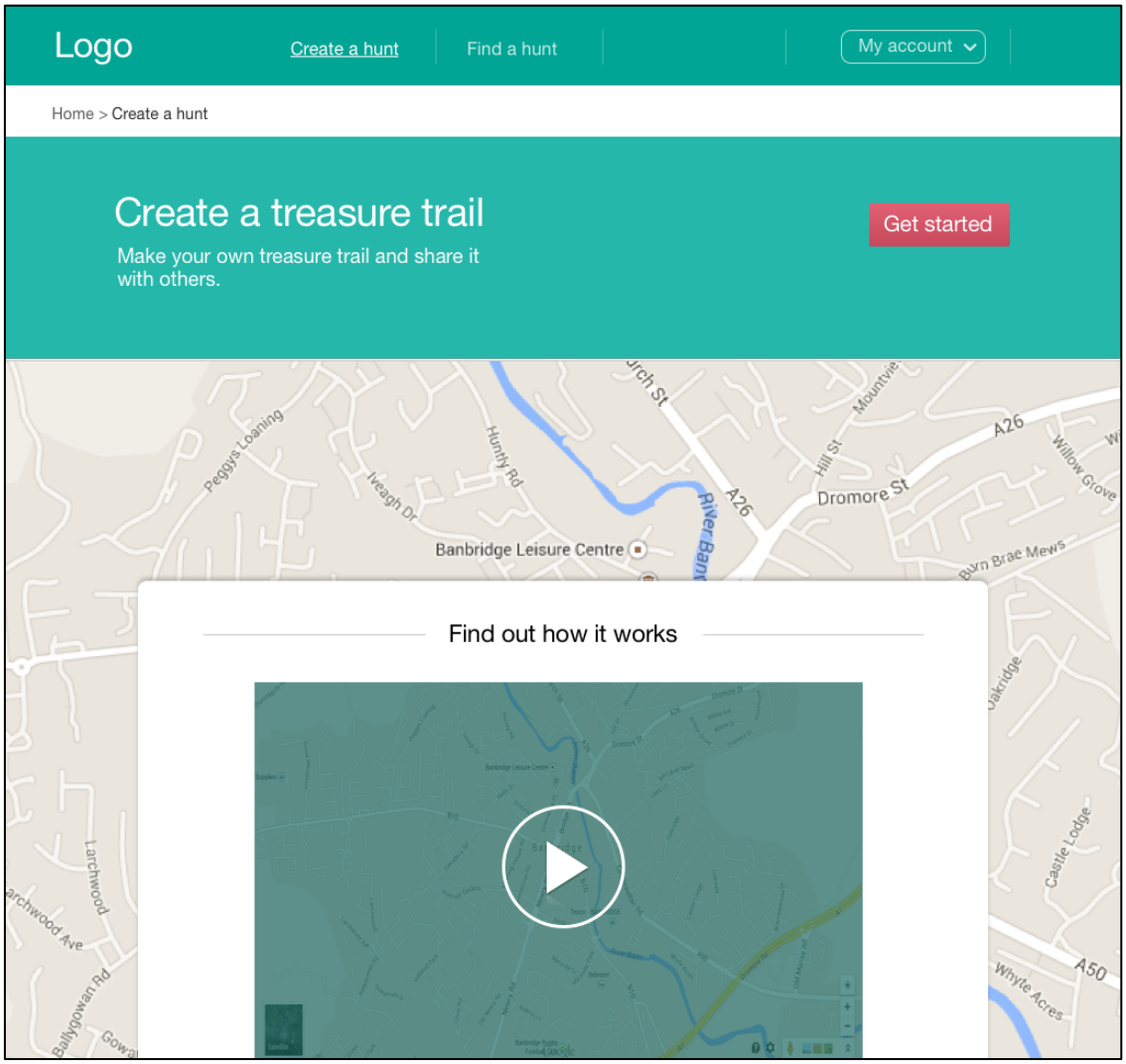
Appendix E – Balsamiq Mockup's prototypes





Appendix F – Initial UX design




Logo
Create a hunt
Find a hunt
My account
Home > Create a hunt
Create a treasure trail
Make your own treasure trail and share it with others.
Get started
Banbridge Leisure Centre
Dromore St
A26
Find out how it works

Logo

Scouts NI Orienteering trip

QR Codes

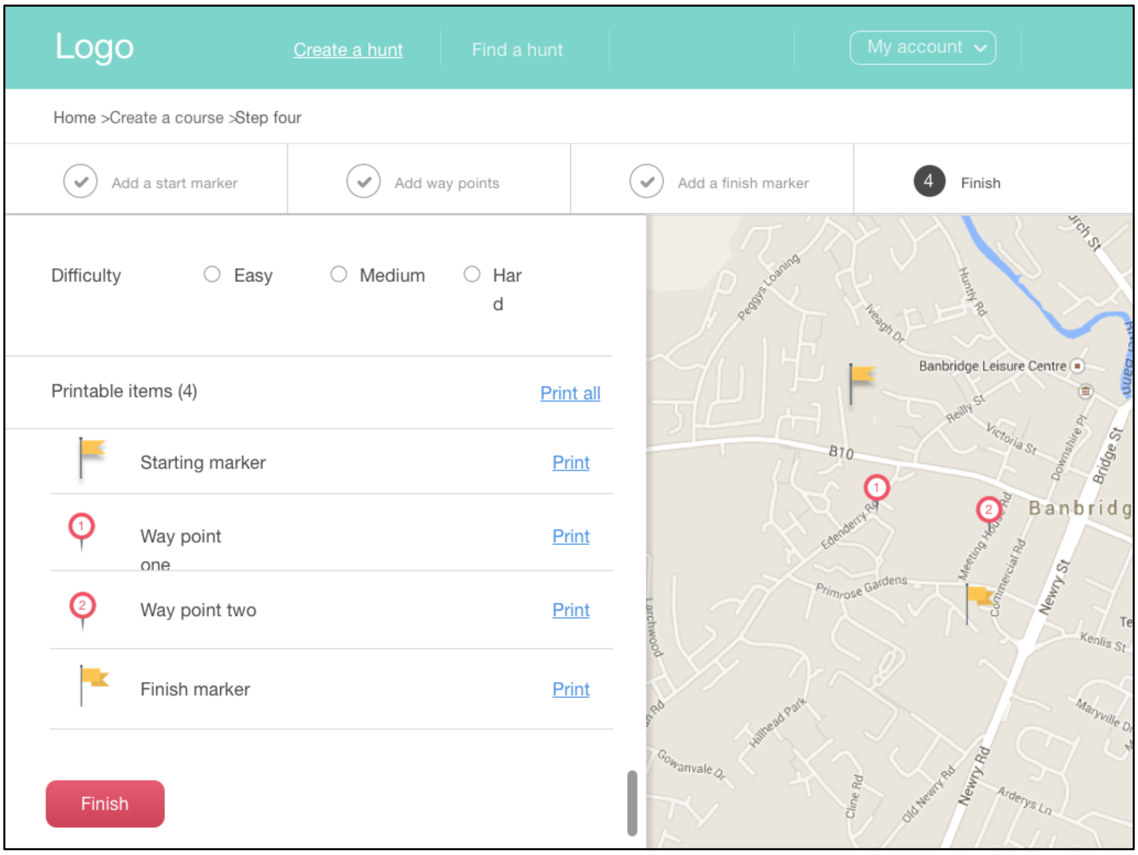
Print out your codes, cut between the cut marks and place at each mark within the hunt.

Scouts Ni orienteering trip

Starting point

Scouts Ni orienteering trip

Way point one


Logo
Create a hunt
Find a hunt
My account
Home >Create a course >Step four
Add a start marker
Add way points
Add a finish marker
4
Finish
Difficulty
Easy
Medium
Har
d
Printable items (4)
Print all
Starting marker
Print
Way point
one
Print
Way point two
Print
Finish marker
Print
Finish
Banbridge Leisure Centre
B10
Banbridg
Newry St
Bridge St
Newry Rd
Primrose Gardens
Victoria St
Commercial Rd
Kenlis St
Gowanvale Dr
Hillhead Park
Old Newry Rd
Arderys Ln

Appendix G – Refined UX design

BOOTILY

Sign in

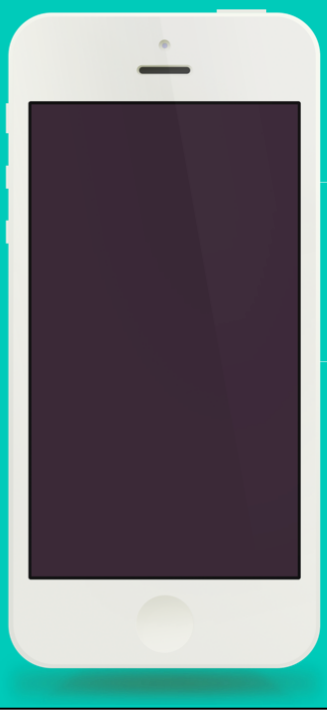
Play and create your own treasure hunts.

Bootily is the easiest way to create and play digital treasure hunts.

Sign me up

What is bootily?

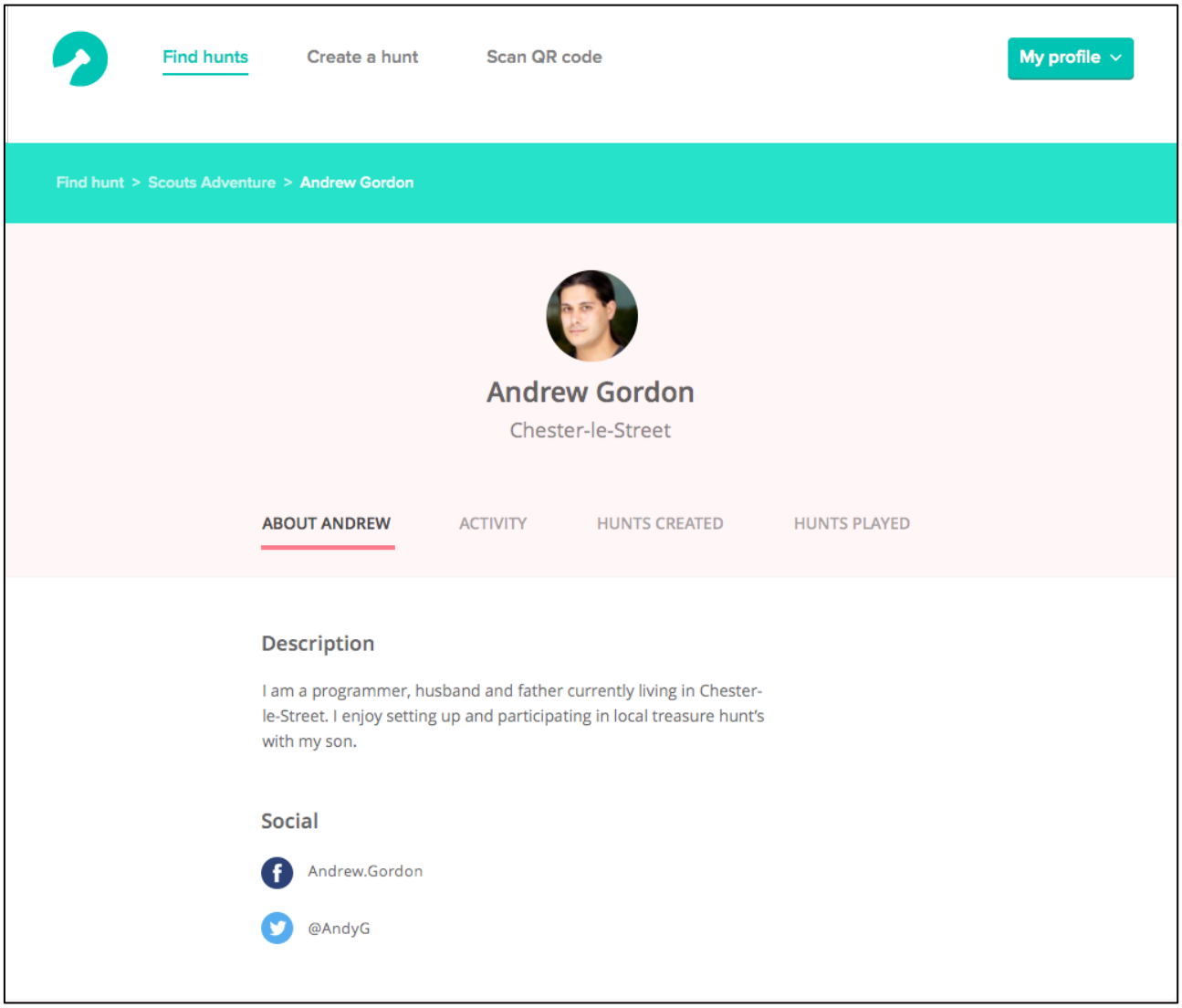
Bootily is a new digital application for setting up and playing online treasure hunts. Treasure hunts can be customised with your own video, text and voice to give that personal touch to your creations.

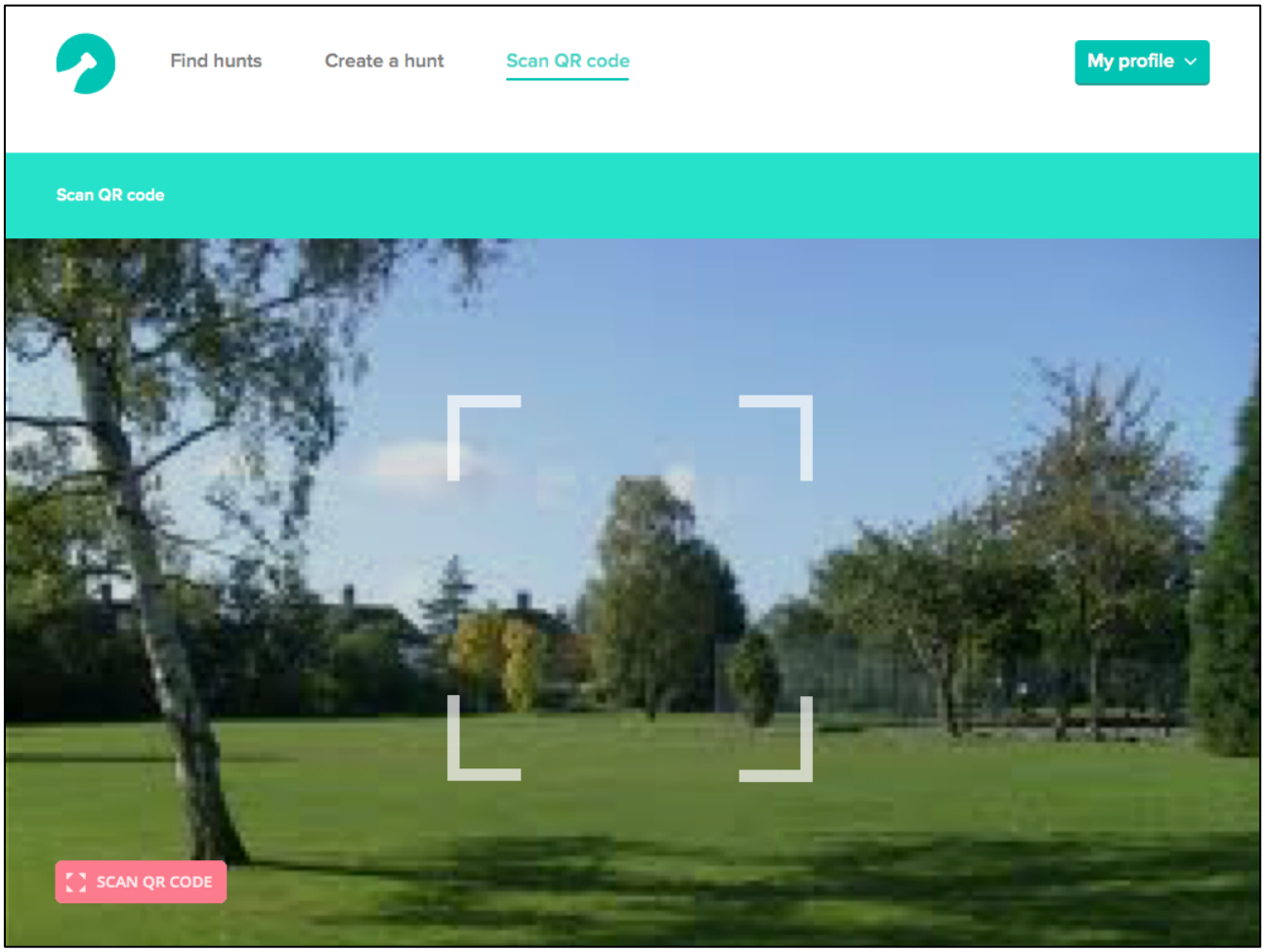


Create custom treasure trails and add custom content.

Share creations with friends through social media.

Use a mobile device's GPRS to complete treasure trails.


Find hunts
Create a hunt
Scan QR code
My profile
Find hunt > Scouts Adventure > Andrew Gordon
Andrew Gordon
Chester-le-Street
ABOUT ANDREW
ACTIVITY
HUNTS CREATED
HUNTS PLAYED
Description
I am a programmer, husband and father currently living in Chester-le-Street. I enjoy setting up and participating in local treasure hunt's with my son.
Social
Andrew.Gordon
@AndyG


Find hunts
Create a hunt
Scan QR code
My profile
Scan QR code
SCAN QR CODE

Appendix E – User survey questionnaires

Instructor:	James Nelson	Name:	Noel Irwin
		Occupation:	Sales assistant
Results:	All requirements passed	Date:	15th April 2015

Upon testing Bootily please fill out the following survey. Whilst answering please place a capital 'X' next to the correct answer. Please email survey back to jamesnelson117@gmail.com upon completion.

7. Where you able to successfully create an account?
   a. Yes X
   b. No

8. Where you able to successfully log into your account?
   c. Yes X
   d. No

9. Where you able to sign out of the application?
   e. Yes X
   f. No

10. Where you able to find a trail?
    g. Yes X
    h. No

11. Where you able to view more detail on a treasure trail?
    i. Yes X
    j. No

12. Where you able to search for a specific hunt?
    k. Yes X
    l. No

13. Where you able to create a treasure trail?
    m. Yes X
    n. No

14. Where you able to print PDF output from trail creation?
    o. Yes X
    p. No

15. Where you able to scan QR codes included with the PDF?
    q. Yes X
    r. No

16. Where you able to upload an image to your profile?
    s. Yes X
    t. No

17. Where you able to edit your personal information?
    u. Yes X
    v. No

18. Did you encounter any issues?
    w. Yes
    x. No X

19. If so please include details in the textbox below

No issues encountered. Great product overall.

20. How would you rate the overall ease of use of the product (1-10)?

7

21. How would you rate the overall functionality of the system(1-10)?

9

22. Is there a feature / piece of functionality that you feel should be implemented in a future release?

I would really like to see this application available on my mobile phone as an application.

Instructor:	James Nelson	Name:	Zoe McClausland
		Occupation:	Student
Results:	All requirements passed	Date:	11th April 2015

Upon testing Bootily please fill out the following survey. Whilst answering please place a capital 'X' next to the correct answer. Please email survey back to jamesnelson117@gmail.com upon completion.

1. Where you able to successfully create an account?
    a. Yes X
    b. No

2. Where you able to successfully log into your account?
    c. Yes X
    d. No

3. Where you able to sign out of the application?
    e. Yes X
    f. No

4. Where you able to find a trail?
    g. Yes X
    h. No

5. Where you able to view more detail on a treasure trail?
    i. Yes X
    j. No

6. Where you able to search for a specific hunt?
    k. Yes X
    l. No

7. Where you able to create a treasure trail?
    m. Yes X
    n. No

8. Where you able to print PDF output from trail creation?
    o. Yes X
    p. No

9. Where you able to scan QR codes included with the PDF?
    q. Yes X
    r. No

10. Where you able to upload an image to your profile?
    s. Yes X
    t. No

11. Where you able to edit your personal information?
    u. Yes X
    v. No

12. Did you encounter any issues?
    w. Yes X
    x. No

13. If so please include details in the textbox below

The QR scanner did not seem to work on my Iphone's browser. I then tried on my phone's browser (Chrome) and it worked fine. Design problems where found on my Iphone.

14. How would you rate the overall ease of use of the product (1-10)?

8

15. How would you rate the overall functionality of the system (1-10)?

8

16. Is there a feature / piece of functionality that you feel should be implemented in a future release?

I would like to see more Social features. Perhaps I could connect with my friends and share my creations with them. Or see and play treasure trails that my friends have made.

Instructor:	James Nelson	Name:	Alex Torrance
		Occupation:	UX designer
Results:	All requirements passed	Date:	11th April 2015

Upon testing Bootily please fill out the following survey. Whilst answering please place a capital 'X' next to the correct answer. Please email survey back to jamesnelson117@gmail.com upon completion.

1. Where you able to successfully create an account?
    a. Yes X
    b. No

2. Where you able to successfully log into your account?
    c. Yes X
    d. No

3. Where you able to sign out of the application?
    e. Yes X
    f. No

4. Where you able to find a trail?
    g. Yes X
    h. No

5. Where you able to view more detail on a treasure trail?
    i. Yes X
    j. No

6. Where you able to search for a specific hunt?
    k. Yes X
    l. No

7. Where you able to create a treasure trail?
    m. Yes X
    n. No

8. Where you able to print PDF output from trail creation?
    o. Yes X
    p. No

9. Where you able to scan QR codes included with the PDF?
    q. Yes X
    r. No

10. Where you able to upload an image to your profile?
    s. Yes X
    t. No

11. Where you able to edit your personal information?
    u. Yes X
    v. No

12. Did you encounter any issues?
    w. Yes X
    x. No

13. If so please include details in the textbox below

James – CSS issues present in various parts of the application and the navigation does not seem to highlight current page.

14. How would you rate the overall ease of use of the product (1-10)?

9

15. How would you rate the overall functionality of the system (1-10)?

7

16. Is there a feature / piece of functionality that you feel should be implemented in a future release?

James, this is a great MVP and as a product I believe this has potential. Continue to work on this product and eventually release it perhaps as an IOS app.