

Requirements Document

2012-13

Project Name: Geocaching

Team Number: 3

James Camden

James King

Emma Nugee

Alice Smiddy

Charles Wilson



# 0 Document Information

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Project Name:** | Geocaching |  |  |
|  |  |  |  |
| **Prepared By:** | James Camden | **Preparation Date** |  |
|  | James King | **5/12/2012** |  |
|  | Emma Nugee |  |  |
|  | Alice Smiddy |  |  |
|  | Charles Wilson |  |  |
|  |  |  |  |
| **Email / Phone:** | cs-seg3@durham.ac.uk |  |  |
|  |  | **Document Version Date** |  |
| **Document Version No:** | 2.6 | **5/12/2012** |  |
|  |  |  |  |

## 0.1 Version History

| Ver. No | Ver. Date | Revised By | Description |
| --- | --- | --- | --- |
| 0.1 | 16/11/12 | Charles Wilson | Created executive summary, and sections of domain analysis. |
| 0.2 |  | James King | Created hardware and software, and sections of domain analysis. |
| 0.3 |  | James Camden, James King | Created functional solution requirements. |
| 0.4 |  | Alice Smiddy | Created domain analysis. Edited solution requirements. |
| 1.0 | 21/11/12 | Emma Nugee | Requirements Document created; added and revised executive summary, created project plan. |
| 1.1 | 22/11/12 | Emma Nugee | Added the hardware and software section. Filled in the responsibility matrix of the project plan. Updated project plan to reflect meeting decisions. |
| 1.2 | 23/11/12 | Emma Nugee | Added domain analysis. First draft handed in to manager for review. |
| 2.0 | 27/11/12 | Emma Nugee | Updated project plan and executive summary to reflect manager feedback. Minor formatting changes to domain analysis. |
| 2.1 | 27/11/12 | Emma Nugee | Added definitions and references. |
| 2.2 | 28/11/12 | Emma Nugee | Updated domain analysis to reflect manager feedback. |
| 2.3 | 30/11/12 | Emma Nugee | Created logo |
| 2.4 | 2/12/12 | Emma Nugee | Began adding solution requirements |
| 2.5 | 4/12/12 | Charles Wilson, Emma Nugee | Added and formatted functional solution requirements |
| 2.6 | 5/12/12 | James King | Updated hardware and software |
| 2.7 | 5/12/12 | Emma Nugee James Camden Charles Wilson | Revised domain analysis – added the background section. Also minor changes to the executive summary. |

# 0.2 Table of Contents

0 Document Information 2

0.1 Version History 2

0.2 Table of Contents 3

1 Executive Summary 4

2 Domain Analysis 5

2.1 Background 5

2.2 Analysis Process 5

2.3 Analysis 5

2.4 Stakeholders 7

3 Project Plan 8

3.1 Responsibility and Deadline Matrix 8

3.2 Gantt Chart 9

4 Hardware and Software Platforms 10

4.1 Android Application 10

4.2 Central Server 11

4.3 Central Server 11

5 Solution Requirements 12

5.1 User Accounts 12

5.2 Game Mechanics 17

5.3

5.4

6 Definition of Terms 27

7 References 30

# 1 Executive Summary

Poppleton Ltd, herein the client, wishes to expand their social program through the development of a mobile game application. This document details the requirements for this mobile game application, as well as setting out the plan of this project and analysis of the domain relevant to the application. The document will be referred to throughout the project to guide the design and development process.

The game application is to be called Fortitude, and will take its inspiration from geocaching, a location based game that requires users to travel to interact with other users and the game as a whole. The increasing ownership of smartphones by the general populace, and the permanent removal of artificial limitations on the accuracy of the GPS coordinates these phones can receive, has led to the creation of virtual geocaching. In this, users are able to place an object known as a virtual geocache at the GPS coordinates of their current location, or interact with a virtual geocache if at the location of an already placed cache.

Fortitude will run on the Android operating system and will provide a social, competitive and interactive experience for users, allowing them to use the inbuilt technologies of their smartphone devices – namely location finding services – to participate in the virtual geocaching game. The application will be supported by a website that provides greater functionality than is feasible on a mobile device.

The aim of this project is to design and create a working version of the application and website that supports the full functionality that is laid out in this document. However, the version created in this project may not be the final version; in particular, the population of the game with caches in this project will be focussed on the local area of Durham whereas the full game will have no such limitation. Moreover, the project will run off a very simple server and as such issues such as advanced server security are not within the scope of this project, although basic measures will be considered. Moreover, the project will run off a very simple server and as such issues such as server security are not within the scope of this project.Moreover, the project will run off a very simple server and as such issues such as server security are not within the scope of this project.Moreover, the project will run off a very simple server and as such issues such as server security are not within the scope of this project.It should also be noted that while the visual appeal of the application and website will not be ignored in this project, it is the functionality that is the main priority.

Detailed functional and non-functional requirements and the research that was used to help outline them are to be found within this document. The research includes an analysis of games and applications similar to what will be developed during this project, the stakeholders that are involved, and research that was conducted into hardware and software information relevant to our application.

# 2 Domain Analysis

### 2.1 Background information

Geocaching is an activity where participants create caches and leave them at a certain location for other participants to find. A cache usually consists of a box containing a small object and a log book for others to record that they found the cache. Often, the object is taken as a prize and replaced with another object for the next time the cache is found. The location of a cache is made available to the geocaching community via a website or application, and is usually given in the form of GPS coordinates.

In this project we will create an android app for virtual geocaching, an alternative form of geocaching where all physical caches are replaced with virtual caches, accessible through the app when the user is at the correct location. The users can place their own caches and get GPS coordinates of caches which they must physically travel to. In the application that will be developed for the client, a cache will act as a marker of an area of land owned by a user. A user approaching a cache they do not own may attempt to conquer the cache, and if they are successful, ownership of the cache transfers to them and they must then defend it from subsequent attacks by other users. With this project we have a unique opportunity to extend the game for non-geocachers, as the client is sub-contracting us to build this app for their social programme, which in this project we will take to mean the general public. We are therefore aiming to make a functional, accessible and exciting app to be used by people of all interests.

### 2.1 Analysis Process

We used a variety of analytical methods to research the domain of this project. These included:

* Thoroughly examining the project mandate, to ensure requirements more than satisfy the client.
* Using the current Geocaching app, giving us first-hand knowledge of how apps are used in conjunction with an outdoor activity, and the practical implications associated with it.
* Interviewing current geocachers to give us an idea of the successful features of existing apps.
* Research in similar areas, which we used this to hone our ideas on virtual geocaching with the hope of being equally or more popular and successful as existing products.

### 2.2 Analysis

The most popular geocaching game currently available is Geocaching, owned by Groundspeak Inc. It consists of a website and a mobile application, although not all members own the application. In 12 years Geocaching has gained over 5,000,000 players and there are almost 2,000,000 caches. Another popular game is Munzee – Scavenger Hunt, which has managed to gain over 61,000 players in just over a year. Munzee is a game similar to Geocaching where caches are replaced with munzees with QR-code style barcodes.

We started by looking at the current Geocaching app. Each user of the Geocaching app must have an account to sign in to which they can create through the app. Once the user has logged into their account, they do not have to log in again until they sign out. The app can be used to search for caches by your current location (found using GPS, Wi-Fi positioning and cell towers), a chosen location or by the unique code of a cache. Once this is done, the user is presented with a list of caches near to their selected location. The list only displays the 20 closest caches, but this can be increased upon request. Following this, the user has the option to display these caches on a map using the Google Maps API.

The user can navigate around the map and display caches near to where they are looking by pressing a “Search Here” option. If you select a cache on the map or from the displayed list, information about it will be displayed. This includes the distance and direction from you, a description about the cache, the date it was placed and by whom. Ways of navigating from your current position to a cache include a map that shows where you are and where the cache is, a compass that has an arrow pointing towards the cache and text stating its distance from the current location, and directions using Google Maps Navigation (Beta). There is a feature which allows the user to download a limited number of caches which can be viewed at a later date without an internet connection. This is particularly useful as users rarely geocache opportunistically and instead plan their routes at home. Finally, at any point in time a user can write a message about a cache that will be visible to the cache’s owner. In this message users can mark that they found the cache and it will be logged to their profile, even if they haven’t actually found it or been there. Users can create new caches, but this can only be done through the website due to the large number of restrictions placed by Groundspeak.

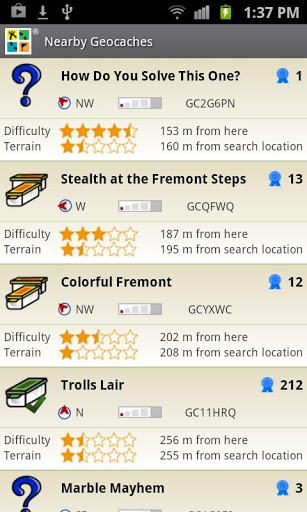
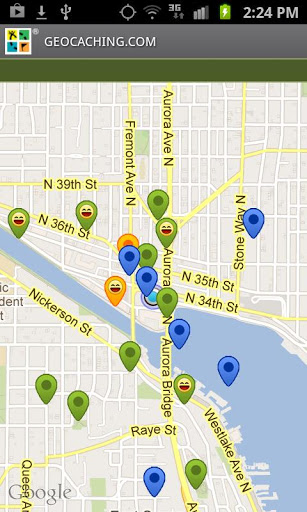


Fig 2.1: Screenshots from Groundspeak’s

Geocaching Android application

We have also used the Munzee application and feel it is very similar to Groundspeak’s app, although certain aspects are different such as the lack of an offline feature. Navigation is done through a map only, and the app makes use of symbols and button labels which some users at times find lack clarity. The primary aim of Munzee is to find munzees, which when scanned reward the user who found, and the user who placed, the munzee with points. This lends the game a competitive aspect with leader boards tracking the most successful users, displayed on the website. Munzees can be deployed directly from the app.

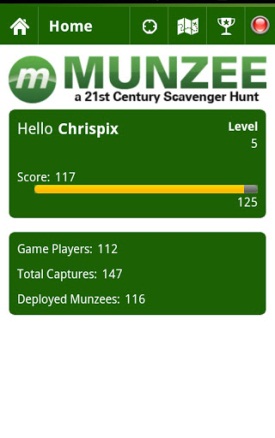


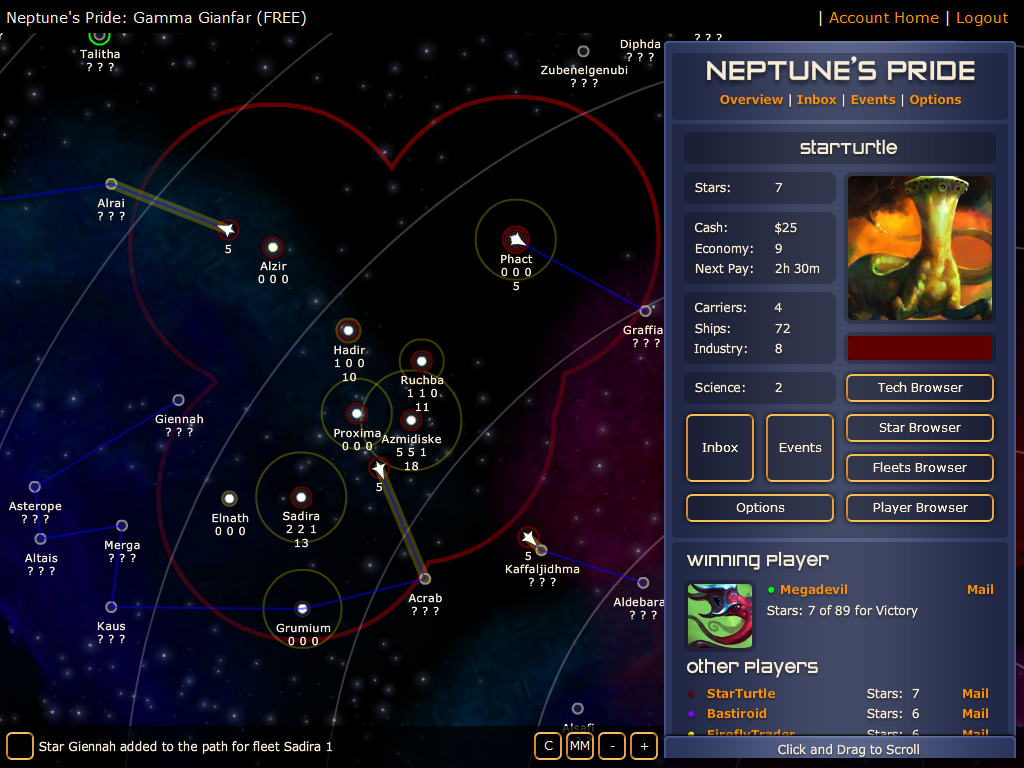
Fig 2.2: Screenshots from Munzee LLC’s game, Munzee – Scavenger Hunt

Based on our own experiences and interviews with other geocachers, the interface of Groundspeak’s app is generally easy to use, but at times the Munzee app is less intuitive. The compass is the preferred means of navigation, with the map only occasionally being used when the compass option is available. The points and leader boards system of the Munzee app are very appealing to users, and those who used Groundspeak’s app were also positive about a competitive aspect to the game. The ability to place and register caches easily is also valued, with the method of placing them directly via the app being preferred.

Another popular app was Fable 3: Kingmaker, in which users plant virtual flags at their GPS coordinates to claim land for one of two sides. The other side are able to retaliate by planting a flag there. The side with the most land won gold. More gold could be won by individuals going to designated locations. Although it only ran from October 2010 to January 2011 in 8 Countries, over 3 million flags were planted. This is an example of a truly virtual and competitive app that utilizes a user’s position and has been successful.

To further our understanding, we researched successful existing online computer games. A good example of this is the multiplayer game Neptune’s Pride, where users receive an amount of money per amount of real time (resource cycle). Each user starts with a fixed number of solar systems and money and their systems can be upgraded by spending money. This causes the system to produce more units and points per

resource cycle. A big challenge, which a lot of users enjoy, is the need to have units be physically transferred between systems by the user and not just appear instantly. Travel time is large causing a slow pace, allowing complex strategies and player alliances to be formed.

Fig 2.3: Screenshot from Neptune’s Pride.

Another game is the Facebook app Farmville, which is very simple but extremely popular. The user accesses the game through their Facebook account and develops and expands a farm. The user plants crops and waits until they grow, they then must return to harvest them before the crops die. Successfully harvesting your crops gives you money, to expand your farm and grow more crops. This sets up an addictive game, where users have to play the game regularly in order to progress. It has become popular by playing on how the average person likes success, progression, ownership and expansion.

An account system is used in all these apps and games. Most websites with account registration need a user to have a valid email address. This is often done by sending users an activation email, which deters a user from creating many accounts, as unique email addresses are needed. It also helps to stop spam accounts being created by an automated system.

It is important to try to prevent malicious users from gaining access to other people’s accounts. This can be done by using complicated, hard to guess passwords which use a mixture of upper case letters, lower case letters and numbers. However this is useless if it can be intercepted while being sent to the server. To stop this passwords are usually hashed with a hashing function that is very difficult or impossible to reverse. The server only sees the hash of a password meaning passwords are protected if the server database is compromised.

If user data is stored the Data Protection Act (1998) requires organisations holding personal information to remove that information when the subject of the data requests it. This is often done by allowing the user to delete their account through an automated process.

### 2.4 Stakeholders

We have identified several stakeholders, represented in an onion ring diagram. At the centre are the general public, the users of the application, and it is important that we consider their views and interests. These users may include current members of the geocaching community, whose opinion is therefore relevant. Private land owners may be affected if caches are placed on their property, although users will be discouraged from doing this. The client will be administrating the app, and have contracted us to create it. Therefore we must be careful to keep to what they have asked us to do in the project mandate. We, the creators of the app, are responsible for making a product which conforms to the client’s requirements and ensuring it is delivered without fault.

Fig 2.4: An ‘onion ring’ diagram to represent the influence that the system will have on key stakeholders. The closer to the centre a stakeholder is, the more influence the app will have on them.

App users

Private land owners

Administrators

Creators of the app and website

# 3 Project Plan

The project has been divided into tasks and sub tasks, each with an associated deadline as shown below and on the following page. Both the soft deadlines and the members responsible for each task may be subject to change if the needs of the project suggest a more efficient use of time and resources.

### 3.1 Responsibility and Deadline Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Sub-task** | **Start Date** | **Soft Deadline** | **Hard Deadline** | **Duration (days)** | **Members Responsible** |
| **Requirements Document** | Elicitation | 25/10/12 | 15/11/12 | 6/12/12  17:00 | 21 | All |
| Sections | 15/11/12 | 22/11/12 | 7 | All |
| Collaborate | 22/11/12 | 29/11/12 | 7 | EN |
| Editing | 29/11/12 | 4/12/12 | 5 | All |
| **Prototype** | Server side functionality | 6/12/12 | 24/1/13 | 7/2/13  14:00 | 49 | JK, JC, CW |
| Client side functionality | 15/12/12 | 24/1/13 | 40 | AS, EN, JC |
| Testing | 17/1/13 | 7/2/13 | 21 | All (CW) |
| **Design Document** | Sections | 6/12/12 | 7/2/13 | 7/3/13  17:00 | 63 | All |
| Collaborate | 7/2/13 | 21/2/13 | 14 | JK |
| Editing | 21/2/13 | 4/3/13 | 11 | All |
| **Finished Product** | Server side functionality | 7/2/13 | 7/3/13 | 14/3/13  14:00 | 28 | JK, JC, CW |
| Client side functionality | 7/2/13 | 7/3/13 | 28 | AS, EN, JC |
| Testing | 21/2/13 | 14/3/13 | 21 | All (CW) |
| **Website** | Scripting | 20/12/12 | 7/2/13 | 14/3/13  14:00 | 49 | JK, JC |
| Web design | 17/1/13 | 21/2/13 | 35 | EN |
| Testing | 31/1/13 | 14/3/13 | 35 | All |
| **Poster** | Creation | 14/3/13 | 26/4/13 | 2/5/13  17:00 | 43 | All |
| Editing | 26/4/13 | 1/5/13 | 5 | All |

Fig 3.1 : Responsibility and Deadline Matrix. ‘Client side functionality’ refers to the graphical user interface   
and the app internal workings. The Members Responsible column shows initials, where JC = James Camden,

JK = James King, EN = Emma Nugee, AS = Alice Smiddy, CW = Charles Wilson, All (CW) = All, headed by CW

### 3.2 Gantt Chart

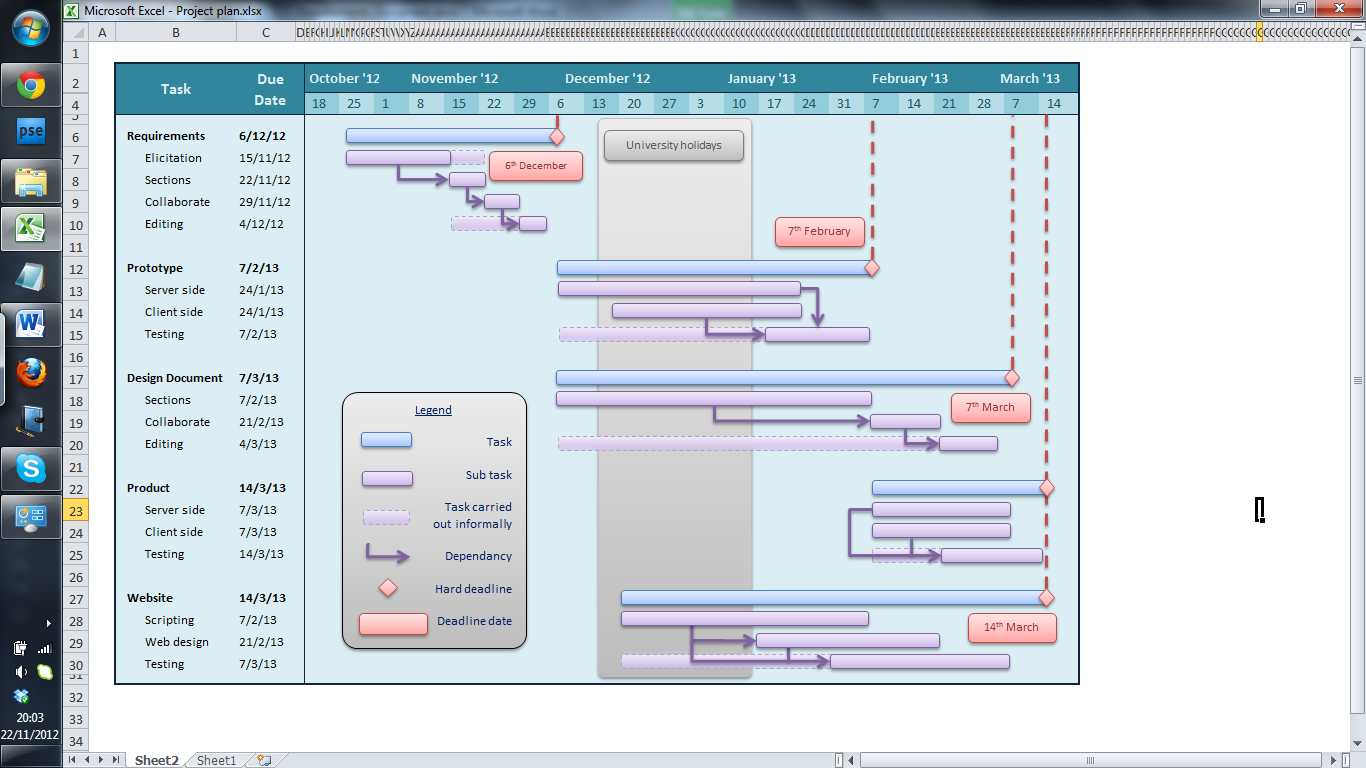


Fig 3.2 : Gantt chart showing the deadlines and time spent on each section of the project, not including the poster.

Where work has been scheduled over the university holidays, the soft deadline for that

task is sufficiently early that it can be extended should holiday work not be completed.

# 4 Hardware and Software Platforms

The main product shall take the form of an Android app - a self contained program running on an Android mobile device. This is what the user will directly interact with. Our product will also require a central server in order to function, and a web server to support a website with additional functionality. The Android app must be able to transfer information to and from the central sever.

### 4.1 Android Application

##### 4.1.1 Hardware

The device provided for testing the Android app is a HTC Desire C, and the application should perform well on similar devices. This device is considered a very basic model, and the majority of handsets that we are targeting are at least as capable. If an application functions well on the Desire C, it will function well on most devices. The relevant hardware capabilities of the Desire C are listed below.

|  |  |
| --- | --- |
|  |  |
| **Screen Resolution** | 320 x 480 pixels |
| **Processor Speed** | 600 MHz |
| **Memory** | 512 MB |
| **SD Card Storage** | 4 GB + |
| **Battery Life** | 10 - 20 hours of active use |
| **Location Service** | Internal GPS Antenna |
|  |  |

Fig 4.1 : Hardware capabilities of the HTC Desire C handset

The app is unlikely to require more capable hardware than what is provided by the device because any intensive processing of data can be offset to the central server. The most that the device will need to handle is some trivial real-time graphical operations, and for that the hardware is more than adequate. Memory should not be a concern because the device has a relatively large amount for a hand-held device, and this product will not require excessive amounts of data to be stored. The app should not exhaust the battery supply of the device to a prohibitive extent, and some effort should be made during development to limit the usage of battery draining resources. The product requires the ability to find its current position using a GPS system, which is an ability of the testing device.

##### 4.1.2 Software

The application will be written in the programming language Java using the development environment Eclipse, and will target the Android operating system version 4.0 as a minimum as specified by the client. Java will be used because it is the language the application developers are most proficient in, and Eclipse was chosen for its superior support for developing Android applications using the Android Developer Tools plugin.

### 4.2 Central Server

##### 1.2.1 Hardware

The central server program will reside on a conventional computer, and because this one machine will handle all client requests it will need much more advanced hardware than the mobile devices. This server shall have internet connectivity in order to communicate with the Android app clients. The server will also need to be almost constantly active, with downtime only occurring at times of the day when few clients will want to connect. The machine running the server requires a large upload bandwidth in order to service many clients in a small time frame, and this should be complemented by a high enough CPU and memory access speed to reduce the processing time of client requests. A lower bandwidth can be compensated for by reducing the size of data being sent to clients. The server should have enough free storage space to allow for expansion of the product to cover more area of the world. At a minimum, the server must have at least two processor cores to utilize multi-threading in the server software, 2 GB of memory to be able to process a lot of data simultaneously and also support the operating system, 16GB of available storage space to hold account and game data.

##### 1.2.2 Software

The server will be written in the programming language C#, and built on the .NET Framework version 4.5 using the Visual Studio 2012 development environment. C# was chosen for its superior performance relative to Java, and because it is the language the primary server developer is most experienced with. Visual Studio 2012 will be used for its support of both the .NET Framework 4.5 and many tools to aid development.

### 4.3 Website

##### 1.3.1 Hardware

The web server will need to be hosted on the same machine as the game server to allow rapid transferral of data between them. This means the server will need the same hardware capabilities.

##### 1.3.2 Software

The web server software will need to serve HTML pages with dynamic content pulled from the game server's data. Because the game server is also servicing HTML requests, they may be merged into one program. This means the software requirements are the same as for the central server. The dynamic portion of the server will be written in C#, using the same development environment as the central server. For clients accessing the website, they will need a web browser that supports the HTML 5 Canvas element. This includes the latest versions of Chromium, Firefox, Opera, Safari and Internet Explorer.

# 5 Solution Requirements

### 5.1 User Accounts

##### 5.1.1 Account Registration

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | New users must be able to create an account which is stored on the server. |
| **Priority** | High |
| **Input** | The user will give a username, email address and password. |
| **Operations** | The server will check that the username and password conform to any length and composition restraints, the email is structured correctly, and that the user-name and email address are both unique. |
| **Expected Results** | If the given information is valid, a new account with the details entered by the user will be stored. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.2 Account Activation

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | New accounts must have their email addresses verified with a verification email. The account will not be `active' - it will not provide access to restricted resources - until the user verifies their email. |
| **Priority** | Medium |
| **Pre-conditions** | Account Registration |
| **Input** | The user will click a link on the verification email to activate the account. |
| **Operations** | The server will check to see if the account has already been activated, or if the account has expired because it had not been activated for an amount of time. |
| **Expected Results** | The account will be marked as active if the account exists and has not yet been activated. If the account is not activated within a set amount of time of the email being sent, the account is removed from the system. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.3 Administrator Accounts

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It must be possible to mark an account as being owned by an Administrator. These accounts will be authorised to perform more actions than normal user accounts. |
| **Priority** | Medium |
| **Pre-conditions** | Account Registration |
| **Input** | An existing user account will be selected to be designated as an administrator account. |
| **Operations** | The server will ensure that the user account selected exists and is not already an administrator. |
| **Expected Results** | An account marked as being an Administrator account will have additional abilities as defined by later requirements. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.4 Resend Activation Email

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user must be able to request that the activation email is resent. This will invalidate the previously sent email. |
| **Priority** | Low |
| **Pre-conditions** | Account Activation |
| **Input** | The user will supply their email address, which is where the activation email will be sent. |
| **Operations** | The server will check that the email address is registered to an existing account that is not already active. |
| **Expected Results** | If the email is registered to an inactivated account, an activation email is sent to the user with a link to activate the account. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.5 User Authentication

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user with an activated account must be able to authenticate themselves in order to access resources restricted to account owners. |
| **Priority** | High |
| **Pre-conditions** | Account Registration |
| **Input** | The user will give their username and password. |
| **Operations** | The server will look for an account in the database which matches the username and password provided. |
| **Expected Results** | If a matching account is found, the user is authenticated as the account owner and they may access some otherwise restricted parts of the system. Otherwise, the user will be informed that the credentials that they supplied are incorrect. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.6 Account Removal

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | An authenticated user must be able to request the deletion of their account along with all personal data. This request must be confirmed by the user through a confirmation email within a week. |
| **Priority** | Medium |
| **Pre-conditions** | User Authentication |
| **Input** | The user will need to be authenticated and also click on a link in a confirmation email. |
| **Operations** | The request will only be fulfilled when the user has clicked a link in a confirmation email that was sent when they initiated the request and it has been under a week since it was sent. |
| **Expected Results** | When the user has confirmed the action, their account will be deleted from the server along with any related personal data. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.7 Administrator Account Removal

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user authenticated as an administrator must be able to delete any  non-administrator account. |
| **Priority** | Medium |
| **Pre-conditions** | Administrator Accounts |
| **Input** | The administrator will select which account they wish to delete. |
| **Operations** | The server will check that the selected account exists and the user making the request is an authenticated administrator. |
| **Expected Results** | The selected account will be removed from the system. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.8 Password Reset

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user should be able to request to reset their password. They do not need to be authenticated to do so. Upon initiating the request, an email will be sent to confirm the action and allow them to choose a new password. |
| **Priority** | Low |
| **Pre-conditions** | Account Registration |
| **Input** | The user will enter their email address. |
| **Operations** | The given email address will be checked to see if it is registered to a user account. |
| **Expected Results** | If the email address does not match an account, the user is informed with a message. Otherwise, an email will be sent to the given email address containing a link to reset their password. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.9 Activation Email Delay

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | After a user requests an activation email or password reset email, the email must have been sent to them within a minute. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.10 Ease of Account Creation

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | A user with no experience of the system but with at least basic knowledge of operating a computer must be able to create an account including activation unassisted and within 5 minutes. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.1.11 Response Times

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | The server must not take longer than 2 seconds on average to process each individual request. |
| **Pass / Fail** |  |
| **Remarks** |  |

### 5.2 Game Mechanics

##### 5.2.1 Account Balance

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Each user account must have a point balance associated with it. When authenticated, the user that owns the account must be able to view the balance at any time. |
| **Priority** | High |
| **Pre-conditions** | Account Registration |
| **Input** | A user makes a request to view their account balance. |
| **Operations** | The server will check that the user is authenticated |
| **Expected Results** | An authenticated user will be able to view their account balance. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.2 Account Transactions

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It must be possible for points to be added and removed from a user's account balance by other components of the system. |
| **Priority** | High |
| **Pre-conditions** | Account Balance |
| **Input** | The amount of points to be added or removed is specified. |
| **Operations** | A withdrawal transaction will not occur if the number of points to remove exceeds the number of points stored in the account. |
| **Expected Results** | When a withdrawal or deposit occurs, the updated amount of points in the account will be updated immediately. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.3 Cache Ownership

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A cache is able to be owned by at most one user at a time. Users must be able to see which caches are owned, and who owns them. |
| **Priority** | High |
| **Pre-conditions** | Account Registration |
| **Input** | A user makes a request to view the owner of a cache. |
| **Operations** | The server will check to see if the cache has an owner. |
| **Expected Results** | If the cache has an owner, the current owner will be displayed. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.4 Cache Balance

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A cache must have a point balance associated with it, which must be visible to authorised users, including the owner of the cache, when requested. |
| **Priority** | High |
| **Input** | A user makes a request to view the point balance of a cache. |
| **Operations** | The server checks to see if the user is authorized to view the point balance of the cache. |
| **Expected Results** | The balance of a cache will be displayed to authorised users when requested. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.5 Cache Transactions

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It must be possible for points be added or removed from the cache by other components of the system. |
| **Priority** | High |
| **Pre-conditions** | Cache Balance |
| **Input** | When points are being added or removed, the amount is specified. |
| **Operations** | A withdrawal transaction will not occur if the number of points to remove exceeds the number of points stored in the cache. |
| **Expected Results** | When a withdrawal or deposit occurs, the updated amount of points in the cache will be visible to authorised users when requested. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.6 Cache Withdrawal

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The owner of a cache must be able to transfer points from the cache to their account when they physically visit the location of the cache. An owned cache becomes unowned if the owning user withdraws all points from the cache. |
| **Priority** | High |
| **Pre-conditions** | Account Balance, Cache Balance |
| **Input** | The user will specify how many points to withdraw. |
| **Operations** | The server will check to see if the user performing the transaction is within a given radius of the cache. The server will also check to ensure the point balance of the cache has at least the number they wish to withdraw. |
| **Expected Results** | If the user is not within a given radius of the cache or the cache has less points stored in it than the requested amount to withdraw, the user will be informed and the transaction will not occur. Otherwise, the cache and user account balances are updated after the transaction. If a withdrawal leaves a cache empty then the cache is marked as unowned. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.7 Cache Depositing

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | When they physically visit the location of a cache that is owned by them or does not have an owner, a user must be able to transfer points to the cache from their account. After transferring one or more points to an unowned cache, the unowned cache will become owned by the user. |
| **Priority** | High |
| **Pre-conditions** | Account Balance, Cache Balance |
| **Input** | The user will specify how many points to deposit. |
| **Operations** | The server will check to see if the user performing the transaction is within a given radius of the cache. The server also checks to see if the user has enough points. |
| **Expected Results** | If the user is not within a given radius of the cache or the user owns less points than they requested to deposit, the use is informed and the transaction does not occur. Otherwise the cache and user account balances are updated after the transaction. If a deposit leaves a previously empty cache populated then the cache is marked as owned. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.8 Cache Placement Cost

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | When placing a cache, a number of points must be deducted from the user's account balance. |
| **Priority** | High |
| **Pre-conditions** | Account Balance, Cache Placement |
| **Input** | A user attempts to place a cache. |
| **Operations** | The server will check to see if the user has enough points in their account to place the cache. |
| **Expected Results** | If the user can afford it, the cache is placed and points are removed from the placing user's account. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.9 Cache Scouting

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | If a user physically visits the location of a cache owned by a different user, they must be able to view the current point balance of that cache. |
| **Priority** | Medium |
| **Pre-conditions** | Cache Balance |
| **Input** | The location of the user is supplied. |
| **Operations** | The distance of the user from the cache is checked to ensure they are sufficiently near the cache. |
| **Expected Results** | If the user is within a given radius of the cache, they will be shown the number of points that is stored in the cache. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.10 Cache Attacking

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | After scouting a cache, a user must be given the option to trigger an attack on that cache using points from their account. |
| **Priority** | High |
| **Pre-conditions** | Cache Scouting |
| **Input** | The attacker chooses how many points from their account they will attack with. |
| **Operations** | The server checks that the attacker's distance to the cache is less than a given radius, and has input a number of points between one and the number of points in their account. The server will decide which party survives the encounter, and how many points they lost in the conflict. |
| **Expected Results** | If the request was valid, an attack is initiated on the cache by that user with the specified number of points. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.11 Successful Attack

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | If the attacker wins, the ownership of the cache must pass to them. All defending points will be lost, and the surviving attacking points will transferred to the balance of the cache. |
| **Priority** | High |
| **Pre-conditions** | Cache Attacking |
| **Expected Results** | The surviving points from their attack will be moved to the cache's point balance, and the cache becomes owned by the attacker. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.12 Successful Defence

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | If the defender wins, the cache must remain theirs. All attacking points are lost, and the surviving defenders remain in the cache. |
| **Priority** | High |
| **Pre-conditions** | Cache Attacking |
| **Expected Results** | The surviving defenders remain in the cache, which remains owned by the defending user. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.13 Battle Breakdown

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | After an attack on a cache, the attacking user will be able to view a breakdown of the results of the attack. |
| **Priority** | Low |
| **Pre-conditions** | Cache Attacking |
| **Input** | An attack on a cache has concluded. |
| **Operations** | The server will provide information such as the number of units lost by each side, and the reward for winning if the attacker was victorious. |
| **Expected Results** | A breakdown of the results of the battle will be displayed to the attacking user. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.14 Cache Operation Chronology

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | All operations on a given cache (transactions, attacks and deletions) must occur in chronological order in respect to when they were received by the server. |
| **Priority** | High |
| **Pre-conditions** | Cache Transactions, Cache Attacking |
| **Input** | An operation on a cache performed by a user or administrator. |
| **Operations** | The server will only process one request concerning operations on a cache at a time. |
| **Expected Results** | An operation will not be initiated until all previously received requests concerning operations on that cache have been completed. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.15 Point Generation

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Points must be periodically supplied to each user based on their current performance in the game. |
| **Priority** | High |
| **Pre-conditions** | Account Balance |
| **Input** | The point allocation system will use a user's current cache number, the events of recent battles involving the user, and other relevant data. |
| **Operations** | The server will use the given information to decide how many points to give the user. |
| **Expected Results** | Each user will periodically receive points based on their performance in the game. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.16 Administrator Placed Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It must be possible for administrators to place caches without needing to be at the location. These caches behave as a user placed cache, but the administrator can place it with as many points in them as they wish, including none. |
| **Priority** | Medium |
| **Pre-conditions** | Cache Attacking |
| **Input** | Administrators will specify the longitude and latitude of a new cache to place, along with how many points will be stored there. |
| **Operations** | The server will validate the location given and if the number of units placed is non-negative. |
| **Expected Results** | The cache will immediately be placed at the given location with the specified number of points in its balance. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.17 Non-Player Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It must be possible for administrators to place caches which may be attacked by users, but not claimed after a victory. |
| **Priority** | Medium |
| **Pre-conditions** | Cache Attacking |
| **Input** | Administrators will specify the longitude and latitude of a new cache to place, along with how many points will be stored there. |
| **Operations** | The server will validate the location given and the point balance given is non-negative. |
| **Expected Results** | The cache will immediately be placed at the given location with the specified number of points in its balance. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.18 Attacking Non-Player Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user attacking a non-player cache must receive a number of points if they are victorious, but it will not become theirs. After the victory, the number of points in the cache balance should be reset. |
| **Priority** | Medium |
| **Pre-conditions** | Non-Player Caches |
| **Input** | Users may trigger an attack in the same was as they would on a user controlled cache. |
| **Operations** | The process for deciding the outcome of an attack on a non-player cache will take the same form as one on a user cache. Each attack is treated as a separate instance and each user will have to wait a period of time before they can attack the same non-player cache again. |
| **Expected Results** | The attacking player will receive a point reward directly to their account if they are deemed to have won the battle, and the point balance in the cache will be reset. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.19 Scouting Non-Player Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | If a user that has attacked a non-player cache and the minimum delay between attacks on a non-player cache has not elapsed for that cache, the user will not be allowed to scout (and by extension, attack) the cache. |
| **Priority** | Medium |
| **Pre-conditions** | Attacking Non-Player Caches |
| **Input** | A user attempts to scout a non-player cache that they attacked a period of time ago that is less than the minimum amount of time that they must wait in between attacking non-player caches. |
| **Operations** | The server will check the amount of time since that user last attacked the cache (if at all). |
| **Expected Results** | If the user has never attacked that cache, or the time since the last attack is greater than the amount of time a user must wait between attacks on a non-player cache, they are able to scout the cache as normal. Otherwise they cannot. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.20 Special Event Placement

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | It should be possible for administrators to define areas by the MAC address of a nearby wireless network. This area will define a collection point for a one-time-only reward which will be limited to a given number of users. |
| **Priority** | Low |
| **Pre-conditions** | Find MAC Address |
| **Input** | Administrators will specify the MAC address of the new cache, the reward, and how many users may claim that reward. |
| **Operations** | The server will validate that the given address is a valid MAC address, and that the reward and number of users which can claim it are greater than zero. |
| **Expected Results** | A new special event area will be available for users to be notified of and claim rewards from. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.21 Special Event Rewards

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | When a user enters the effective range of a wireless network whose MAC address has been designated as a special event placement, they must be able to claim a reward from it. If it is still available, they will receive points directly to their account balance. Once the maximum number of users have claimed the reward the cache is removed and no longer visible or available. |
| **Priority** | Low |
| **Pre-conditions** | Special Event Placement |
| **Input** | A user claims a reward at a given wireless network. |
| **Operations** | The server will check to see if the reward is still available. |
| **Expected Results** | If the reward is still available, a user can claim the reward. The value is credited directly to their point balance and the special event area is removed after the reward has been claimed by a designated number of users. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.22 Cache Reporting

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Users should have the ability to mark a cache as being placed unfairly. Administrators will be alerted to the reported cache. |
| **Priority** | Medium |
| **Pre-conditions** | Cache Placement |
| **Input** | A user will select a cache that they wish to report. |
| **Operations** | The server will check that the cache is not owned by the user themselves, has not already been reported, and is not a non-player cache. |
| **Expected Results** | If the report is valid, an administrator will be alerted to the reported cache, with information including the owner, location, and who reported it. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.23 Administrator Cache Deletion

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Administrators must have the ability to delete any cache from the system. |
| **Priority** | Medium |
| **Pre-conditions** | Administrator Accounts |
| **Input** | An administrator will select a cache that they wish to remove from the system. |
| **Operations** | The server will check that the selected cache exists and that the user making the request is an authenticated administrator. |
| **Expected Results** | The selected cache will be deleted from the server and will no longer be visible to users. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.24 Account Deletion Cache Removal

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | When a player account is removed from the system, all caches owned by them must also be deleted. |
| **Priority** | Medium |
| **Pre-conditions** | Cache Ownership, Account Removal |
| **Input** | The process will be triggered by a user account being removed |
| **Operations** | The server will find all caches owned by that account. |
| **Expected Results** | All caches owned by the deleted player will be removed from the system. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.24 Anti-Cheating Measures

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The system must include a method of ensuring that a user is at the location they claim they are when they perform a location dependent action. |
| **Priority** | Medium |
| **Input** | The user makes a request that requires the user to be at the location they claim to be. |
| **Operations** | The server performs some validation to ensure the user is where they are claiming to be. |
| **Expected Results** | If the request passes the server's validation, the request is performed. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.26 Distinguishing Cache Owners

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | A new user must be able to identify which caches are owned by themselves, which are owned by opponents, and which have no owners at all without assistance. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.27 Difficulty Curve

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | There should be an inverse relationship between the ease with which a user can expand their territory (own new caches) and the number of caches that they own. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.28 New Player Protection

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | There must be some form of protection for new players to deter older ones from easily eliminating them, or a disincentive to make preying on new players unprofitable. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.29 Cache Action Delay

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | Any action performed by an administrator or user on a cache (such as placement, transactions or an attack) must take effect within 10 seconds. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.2.30 Data Usage

|  |  |
| --- | --- |
| **Type** | Non – Functional |
| **Description** | The application should, while in active use, transfer on average less than 5MB an hour with the server. |
| **Pass / Fail** |  |
| **Remarks** |  |

### 5.3 Application

##### 5.3.1 Display Location

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The application must allow the user to have displayed to them their current location in the context of a map provided by the Google Maps API. |
| **Priority** | High |
| **Input** | The user will request for the Google Maps API powered map to navigate to their current location. |
| **Operations** | The application will check to see if the user has an Internet connection and that their GPS antenna is enabled. |
| **Expected Results** | If the user has an internet connection and their GPS antenna is enabled, the application will show the user's location on a map provided by the Google Maps API. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.2 Nearby Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The application must be able to show the user, on the map, the locations of caches near to a specified location. |
| **Priority** | High |
| **Pre-conditions** | Display Location, Server Connectivity |
| **Input** | The user will request to see the locations of caches near to a specified position. |
| **Operations** | The application will check that the device's GPS antenna is enabled and there is Internet connectivity. If there is a connection and the antenna is enabled, the application will send a request to the Server for a list of coordinates of caches near to the specified position. |
| **Expected Results** | The application will mark on the map, provided by the Google Maps API, all caches near to the specified position. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.3 Map Zooming

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The application should allow the user to view a larger or smaller area of the map. |
| **Priority** | Medium |
| **Pre-conditions** | Nearby Caches |
| **Input** | The user will be able to specify the `zoom level'. |
| **Operations** | The application will ensure that the specified zoom level is within defined limits. |
| **Expected Results** | The map is displayed at different levels of zoom, as specified by the user. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.4 Path Finding

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The application should allow the user to view a path between the user's current position and a target cache they specify. |
| **Priority** | Low |
| **Pre-conditions** | Display Location, Nearby Caches |
| **Input** | The user will select which cache they want to navigate to. |
| **Operations** | The application will use the Google Directions API to find a path to the target cache. |
| **Expected Results** | A path will be drawn on a map to show which route to take. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.5 Cache Placement

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | The application must allow the user to request the placement of a cache at their current location. |
| **Priority** | High |
| **Pre-conditions** | Server Connectivity |
| **Input** | The user will trigger a request to place a cache at their current position. |
| **Operations** | The server will validate the request to place a new cache. |
| **Expected Results** | The application will send a cache placement request to the central server, and if it is successful all users will be able to see the new cache. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.6 Activity Recording

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Any game activity performed by a registered user should be recorded and can be viewed by that user when requested. Users will also be able to view activities by other users which have a direct effect on them. |
| **Priority** | Medium |
| **Pre-conditions** | User Accounts |
| **Input** | A user can request the activities that have occurred since a specified time and date. |
| **Operations** | The server will locate all activities related to that user since they time they gave. |
| **Expected Results** | Any activities that occurred since the given time and date will be displayed to the user. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.7 Interface Style Uniformity

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The application's graphical user interface must have a consistent overall style, including colour scheme, usage of images, and structure. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.8 Interface Scaling

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | All elements of the user interface should adhere to the screen dimensions of the host device. No elements of the interface should not be visible due to being cropped off-screen. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.9 Interface Feedback Delay

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The user interface should provide feedback of some form within 0.25 seconds for all user actions that require a response. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.10 Start-up Time

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The application should, on average, be ready for user input in less than 10 seconds after starting. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.3.11 Battery Usage

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The application should be optimised such that, when started on a device with a full battery, it can be active for at least 4 hours without completely depleting the host device's battery. |
| **Pass / Fail** |  |
| **Remarks** |  |

### 5.4 Peripheral Functionality

##### 5.4.1 User Communication

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Users may be able to send messages to other users. |
| **Priority** | Low |
| **Pre-conditions** | User Accounts |
| **Input** | An authenticated user will specify the message subject, and the message to send. |
| **Operations** | The server will check that the subject and message body are not empty. |
| **Expected Results** | A message will be created which is visible to all intended recipients. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.2 Communication Reporting

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | Users should have the ability to report communications sent between users as being inappropriate. Administrators will be alerted to the reported communication. |
| **Priority** | Low |
| **Pre-conditions** | User Communication |
| **Input** | The user will specify which message to report. |
| **Operations** | The server will check that the message has not already been reported. |
| **Expected Results** | If it has not already been reported, an administrator will be alerted to the reported message, including information such as the sender, message content and who reported it. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.3 Website

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | There must be a website that is accessible by users. |
| **Priority** | Medium |
| **Pre-conditions** | User Accounts |
| **Input** | A user will make a request to access a certain page of the website through a web browser. |
| **Operations** | The server will provide the requested page if it exists. |
| **Expected Results** | A user will be able to view a requested web page if it exists. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.4 Website Authentication

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user must be able to authenticate themselves to their account through the website to be able to access restricted content on the site. |
| **Priority** | Medium |
| **Pre-conditions** | User Accounts |
| **Input** | The user will give their username and password. |
| **Operations** | The server will look for an account in the database which matches the username and password provided. |
| **Expected Results** | If a matching account is found, the user is authenticated as the account owner and they may access some otherwise restricted parts of the website. Otherwise, the user will be informed that the credentials that they supplied are incorrect. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.5 Viewing Owned Caches

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | A user, when authenticated on the website, must be able to view a list of all caches that are currently owned by them, and details about each one, including the number of points stored inside and its location. |
| **Priority** | Medium |
| **Pre-conditions** | Website Authentication, Cache Ownership |
| **Input** | A user will request to view the web page containing a list of the caches they own. |
| **Operations** | The server will ensure the user is authenticated. |
| **Expected Results** | If the user is authenticated, a web page containing a list of all caches owned by that user's account is displayed. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.6 Viewing Caches Details

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | On the website, an authenticated user must be able to view information about a specified cache. This information will include a graphical representation of the location of that cache on a map. If the cache is owned by the user that made the request, further information such as the number of points in the cache and a history of attacks made against the cache while it was owned by that user are also displayed. |
| **Priority** | Low |
| **Pre-conditions** | Website Authentication, Cache Ownership |
| **Input** | A user will request to view a web page containing additional information about a specified cache. |
| **Operations** | The server will check that the user is authenticated, the specified cache exists, and if currently owned by the user. |
| **Expected Results** | A web page with details about that cache will be displayed, including a graphical representation of the cache's location on a map and the current owner. If the cache is owned by the user that made the request, additional information will be displayed including the number of points in the cache and a history of its recent events. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.7 Overview Map

|  |  |
| --- | --- |
| **Type** | Functional |
| **Description** | An authenticated user must be able to view a map, using the Google Maps API, which displays a subset of all caches. The user will be able to specify a filter to decide which caches are displayed. The user should also be able to use this map to select a cache to view information about it. |
| **Priority** | Medium |
| **Pre-conditions** | Viewing Cache Details |
| **Input** | A user requests to view a web page containing the overview map using a specified filter. |
| **Operations** | The server will ensure that the user is authenticated. If they are authenticated, the server will find the collection of caches that matches the specified filter. |
| **Expected Results** | If the user is authenticated, they will be provided with a web page containing the overview map. The map will only contain the caches that match the given filter, and will allow the user to select caches on the map to access the cache details page for that cache. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.8 Website-Application Style Uniformity

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The overall style of the website must match that chosen for the Android application, with the exception of the general structure which may differ. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.9 Page Request Response Time

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | The server must, on average, take less than 2 seconds to produce a requested web page after receiving a request. |
| **Pass / Fail** |  |
| **Remarks** |  |

##### 5.4.10 Ease of Navigation

|  |  |
| --- | --- |
| **Type** | Non - Functional |
| **Description** | It must be possible to travel between any two pages in no more than three page transitions (assuming the user attempting to access the desired page is authorised to). |
| **Pass / Fail** |  |
| **Remarks** |  |

# 6 Definitions

**Account** The means by which a user accesses a system. An account stores data particular to that user and is typically accessible only by a unique ID and password combination.

**Android** A Linux-based operating system currently developed by Google which is designed primarily for touchscreen mobile devices.

**App, application** A program or piece of software designed to fulfil a particular purpose.

**Bandwidth** The transmission capacity of a computer network or other telecommunication system

**Cache, Geocache** A hidden container that usually contains a log book for geocachers to sign. A cache may also include an item. Within the context of this document, cache may also refer to a virtual cache (see below).

**Canvas** The canvas element is part of HTML5 and allows for dynamic, scriptable rendering of 2D shapes and bitmap images. It is a low level, procedural model that updates a bit map and does not have a built in scene graph.

**Cell towers** A fixed transmitter/receiver location, also known as a base station or a cell site, which establishes communications between a wireless system and a wireless device using radio links. The cell tower includes an antenna tower, transmission radios and radio controllers.

**C#, C Sharp** A multi paradigm programming language developed by Microsoft.

**Data Protection Act** A United Kingdom act of parliament defining UK law on the processing of data on identifiable living people.

**Eclipse** A software system for the development and deployment of constraint programming applications.

**Fable 3, Fable III** An action role-playing open world video game developed by Microsoft Game Studios where players must gain and retain control of a kingdom.

**Fable 3: Kingmaker** The title of smart phone application corresponding to the video game Fable 3. The app differs from the video game by being tied to real world locations.

**Farmville** A real-time simulation game developed by Zynga, accessible through Facebook and as a smart phone app.

**Geocacher** One who geocaches. This may also include one who virtual geocaches.

**Geocaching** A form of treasure hunt using GPS, in which an item is hidden somewhere in the world and its coordinates posted on the Internet, so that GPS users can locate it.

**Google Maps** A web mapping service application and technology provided by Google that powers many map-based services.

**Google Maps API** Standing for Google Maps Application Program Interface. A set of routines, protocols and tools for building software applications using Google Maps.

**GPS** Standing for Global Positioning System. An accurate worldwide navigational and surveying facility based on the reception of signals from an array of orbiting satellites.

**Groundspeak Inc.** The company that owns the largest geocaching site.

**GUI** Standing for Graphical User Interface. A user interface based on graphics rather than just text or a command line interface.

**Hashing function** A function which converts an arbitrary block of data into a fixed size bit string from which the original data is unrecognisable.

**HTC Desire C** An Android smartphone manufactured by HTC, released in May 2012. The HTC Desire C is seen as a low-end or entry level device.

**HTML5** The new standard of HTML, introducing several new features and retiring some old ones. HTML5 is designed to remove the need for plugins and addons and provide better structuring for HTML documents. Not all browsers have full HTML5 support.

**Java** A network orientated programming language invented by Sun Microsystems. Java in the language that most apps are written in for use on Android smart phones.

**Multi-threading** A technique by which a single set of code can be used by several processors at different stages of execution

**Munzee** 1) A real world scavenger hunt game where items are found in the real world and captured by a smart phone with the use of QR-codes.

2) The name given to a cache in the game Munzee.

**Neptune's Pride** A slow-form multiplayer strategy game where players battle each other in order to conquer a galaxy.

**.NET Framework** A software framework developed by Microsoft.

**Profile** The means of displaying data associated with a user account. Profiles may be public or private to that user.

**QR-code** Standing for Quick-Response codes. A two dimensional or matrix barcode that is readable by mobile phones.

**Server** A computer or computer program that manages access to a centralized resource or service in a network.

**Smartphone** A mobile phone with more advanced computing ability and connectivity than a basic phone.

**Software** The programs and other operating information used by a computer.

**Stakeholders** A person or other entity with an interest or concern in something, especially a business.

**Virtual cache** A geocache which exists only in the context of a virtual geocaching game. A virtual geocache is only accessible from a set location and provides some form of game functionality once accessed.

**Visual Studio 2012** The most recent release of an integrated development environment from Microsoft which can be used to develop console and graphical user interface applications.

**Wi-Fi positioning** A location finding system which makes use of wireless access points, used where GPS is inadequate.

# 7 References

CDG Resources, *Acronyms & Definitions*. [online] Available at: <http://www.cdg.org/resources/acronyms.asp> Accessed November 2012

Fable 3 Kingmaker, *Fable 3 Kingmaker on Twitter.* [online] Available at: <https://twitter.com/Fable3Kingmaker> Accessed November 2012

Geocaching, *Groundspeak’s Geocaching Android Application Tutorial.* [online] Available at: <http://www.geocaching.com/android/tutorial/default.aspx> Accessed November 2012

HM Government Publishing, *Data Protection Act 1998*. [online] Available at: <http://www.legislation.gov.uk/ukpga/1998/29/contents> Accessed November 2012

HTC Corporation, *HTC Desire C Specs*. [online] Available at: <http://www.htc.com/uk/smartphones/htc-desire-c/#specs> Accessed November 2012

Iron Helmet Games, *Neptune’s Pride (BETA 2).* [online] Available at: <np.ironhelmet.com> Accessed November 2012

Munzee LLC, *Munzee*. [online] Available at: <http://www.munzee.com/> Accessed November 2012

Munzee LLC, *Welcome to the World of Munzee.* [online] Available at: <https://play.google.com/store/apps/details?id=com.munzee.android.client&hl=en> Accessed November 2012

Oxford Dictionaries. *Oxford Dictionaries*. [online] Available at: < http://oxforddictionaries.com/> Accessed November 2012

SanSambal, 2010, *Fable II Kingmaker*. [online] Available at: <http://www.youtube.com/watch?feature=player\_embedded&v=RlfvMJlZFqM> Accessed November 2012

ScreenTrend. *Words to know, further reading and useful links.* [online] Available at: <http://screentrend.com/resources.html> Accessed November 2012

WebCamStore. *Webcam Glossary*. [online] Available at: <http://www.webcamstore.com/consumer/tutorial/glossary.php> Accessed November 2012