Software Requirements Specification

for

Java Script &

JSON Route Editor

Version 1.1

Prepared by

Group Name: Zuntechi

|  |  |  |
| --- | --- | --- |
| Sebastian Hilton | 500244541 | eeuc47@bangor.ac.uk |
| Emmanuel Onah | 500252011 | eeuc24@bangor.ac.uk |
| James Wright | 500214469 | eeua56@bangor.ac.uk |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| Instructor: | Steven Marriot |
| Course: | Computer Science |
| Lab Section: | Software Hut |
| Client: | Ryan Brookes |

|  |  |
| --- | --- |
| Version 1.0.5 Print Date: | 13/02/12 |

Contents

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

Appendix B - Group Log 6

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | James Wright | Initial creation of document | 30/01/12 |
| 1.0.1 | Sebastian Hilton | Update Appendix B – Group Log | 07/02/12 |
| 1.0.2 | James Wright | Section 2 draft | 12/02/12 |
| 1.0.3 | Emmanuel Onah | Section 1 draft | 12/02/12 |
| 1.0.4 | Sebastian Hilton | Section 3 draft | 12/02/12 |
| 1.0.5 | Group | Review and update of sections | 13/02/12 |
| 1.0.5.1 | Sebastian Hilton | Update Appendix B – Group Log | 15/02/12 |
| 1.0.5.2 | Emmanuel Onah | Create use case diagram | 22/02/12 |
| 1.0.5.3 | Sebastian Hilton | Work on initial prototype | 22/02/12 |
| 1.0.5.4 | James Wright | Testing type research with explanation | 24/02/12 |
| 1.0.5.5 | Emmanuel Onah | Create Use case Diagram | 30/02/12 |
| 1.0.5.6 | Sebastian Hilton | Set up Xampp and gitHub project repository | 01/03/12 |
| 1.0.5.7 | James Wright | Identify testing methods | 01/03/12 |

# 

# Introduction

## Document Purpose

The purpose of this document is to present a detailed description of a user-editable climbing guidebook website. It will explain the purpose and features of the website, the interfaces of the website, what the website will do, the constraints under which it must operate and how the website will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## Product Scope

This document has been written for a developer, as one which describes the requirements of the system for validation of the final system delivered. Any changes to the requirements and this document must be processed and approved, then listed above.

## Intended Audience and Document Overview

This document is intended for use by System Architects, Project Managers, Developers and Test engineers.

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## Definitions, Acronyms and Abbreviations

**JSON:** JavaScript Object Notation

**REST:** Representational State Transfer

## References and Acknowledgments

|  |  |
| --- | --- |
| 1. | IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1993 (Revision of IEEE Std 830-1984) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Overall Description

## Product Perspective

There have been several previous implementations of the route editor however none have been successful. Our product is a new product and will effectively be the first of its kind. It will be entirely self-contained.

## Product Functionality

In this section we will describe the main functions our product will perform.

* Allow the user to view all routes from our database.
* Allow the user to upload an image of a crag
* Allow the user to add climbing routes to any image
* Allow the user to add points of interest (belay, difficult section etc.) to the routes.
* Product must be user friendly and very easily operated
* Product’s presentation must be of an extremely high standard

## Users and Characteristics

Our product is primarily designed for use by climbers, specifically those with little to no computing experience. It is crucial that our product can be used by anyone and choosing this target audience will ensure that. We anticipate a large variety of users will use our product with the heavier users being frequent climbers and the lighter users being occasional climbers. We also expect our product to be at least tried by walkers, ramblers, hikers, explorers and other members of the outdoors community.

## Operating Environment

Our product is designed to be viewed in a web browser on a desktop computer or laptop. It will also be primarily designed to run in Windows 7 (service pack 1) and viewed in Google Chrome. It will however also be operational in Internet Explorer and Mozilla Firefox. Smart phones with Javascript capabilities will also be able to view the application but a stand-alone app will not be created.

## Design and Implementation Constraints

Our product is a community driven project. This immediately outlines our main constraint which is the quality of the user’s contributions. At product launch our database of crag images and routes will be very minimal and this will affect the products usability. However as the product sees more and more activity the product will grow and thus the usability will grow.

## User Documentation

One of the main functions of our product is that it is incredibly easy to use by anyone. Because of this we will not be including any tutorials but we will be supplying a user manual with the documentation. This will contain full instructions on operating all functions of the product. The user interface of our product will be designed in a helpful manner and will aid users as they operate the product.

## Assumptions and Dependencies

Our product is independent of all other software however it will be implemented within a website. As previously stated, our product is community driven. We are therefore designing the route editor under the assumption that the community will consist of honest trustworthy contributors. Our product relies on users like this in order to reach its full potential.

# Specific Requirements

## External Interface Requirements

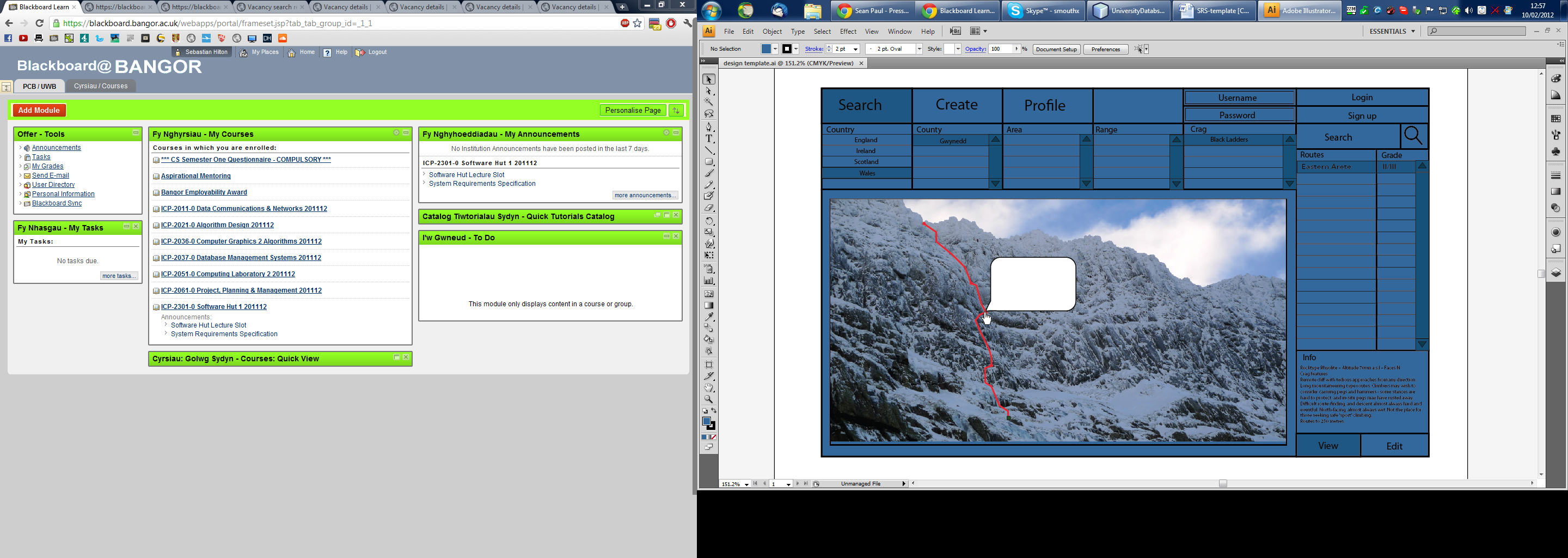
### User Interfaces

As this project will be web based, we will be implementing twitter's bootstrap layout style. In order to keep it simple there will only be 3 possible pages;

* Search: Set as the home page and used to search for routes.
* Create: Very similar to layout of search but with some modifications in order to allow user editing.
* Profile: This is the page with least priority as it will be an added feature.

Bellow is an example of what the Search page will look like.

The scrollable option panels works by filtering out the JSON URL's and then showing a specific routes.



### Hardware Interfaces

All the TOPO data will be stored on a server in the form of a JSON URL. We need to ensure that the app works efficiently on all browsers. The choice of implementing the app in java script rather than java is because mobile devices are often incompatible with java run web apps.

### Software Interfaces

The web application will need to be able to access the server and the databases stored on this. We are still in the process of selecting an appropriate database manager as this will contain many pictures and we would want to make the data acquisition as fast as possible while maintaining a compressed storage space.

Appendix B - Group Log

**Week 1:**

**Choice of projects and project bid**  **(23/01/12)**

In attendance: SH,JW,EO

To do:

* Select main project and two back up projects.
* Write bid on main project (approx. 500 words).

**Week 2:**

**Initial Meeting (30/01/12)**

In attendance: SH,JW,EO

To do:

* Identify all software technologies required.
* Contact client to gather further information.
* Familiarise with Java-Script, JSON and REST.
* Select person in charge of UML. (Sebastian)
* Requirement specification document basic skeleton.

**Week 3:**

**Meeting with Client (06/02/12)**

In attendance: SH,JW,EO

To do:

* Look further into Java-Script , JSON and REST
* Meet with project sponsor
* Update requirements specification

Meeting with client (Ryan Brookes)

Point to consider:

* Include ascent and descent routes as well as details on how to access the crag
* Take into account previous attempt to develop project using DOJO
* The high priority of good usability and GUI
* Identify; clip in points, belay sections, hazards, change in difficulty, etc.
* Completely open-source with anonymous editing, choice of membership if wanted.
* Version control in order to protect system from malicious activity.
* Possible implementation of twitter’s open source style; Bootstrap
* Beware of problems with printing the routes with Java-Script
* Possible host: XAMPP
* Look into using Grails & Groovy as well as Bitbucket.

**Week 4:**

**Standard Meeting (13/02/12)**

In attendance: SH,JW,EO

To do:

* Develop a System Architecture diagram with client
* Make connection to server
* Setup a basic web site, interacting with Java-Script
* Read up on software testing; agree on format of forms (James)

**Group Work (15/02/12)**

In attendance: SH,JW,EO

Client was contacted about server connectivity and file sharing using gitHub. A server hasn’t yet been set up. However, we shall be connecting locally and testing this using XAMPP.

James Wright:

* Software testing research.
* Decided on Software testing technique – JSUnit.

Sebastian Hilton

* Familiarise with twitter bootstrap.
* Set up basic navigation bar.

Emmanuel Onah:

* Familiarise with JSON syntax.
* Look into changing JSON text files into Java-Script objects.

**Development Work (16/02/12)**

In attendance: SH

* Research JQuery and JavaScript technologies and continue with web page layout

**Week 5:**

**Standard Meeting (20/02/12)**

In attendance: SH,JW,EO

To do:

* Use case diagram with text; with aid of UML tool. – E.O.
* Working prototype; load and plot on image. – S.H.
* Contact Ryan – E.O.
  + Image library (who will supply?)
  + Demonstrate User Interface and ensure client is happy with design
  + Mention API’s and Libraries that have been researched
  + Server setup clarification
* Provide a list of 10 types of testing with an explanation. - J.W.

**Week 6:**

**Standard Meeting (27/02/12)**

In attendance: SH,JW,EO

To do:

* Group review of Use Case Diagram
* Communicate with client
* System architecture diagram (informal)
* Server connection test if possible
* Group review of Requirement Specification Documentation
* Continue with development of initial prototype - S.H.
* ArgoUML diagram example – E.O.
* HTML development for prototype – J.W.