Finnian O Neill

Student Number: 13430512

Supervisor: Dr. Stephen Blott

Abstract

Scout Ahead is a website to be used as a planning tool for people looking to explore the outdoors. It allows users to search a route, get directions, find the weather, see the elevation profile of the route and download the route as a GPX file which can then be used in their GPS device.

Scout Ahead

Technical Manual

A group of people standing in the dark

Description generated with high confidence

Table of Contents

[Motivation 2](#_Toc514623756)

[Glossary 2](#_Toc514623757)

[Research 3](#_Toc514623758)

[Design 4](#_Toc514623759)

[User Stories 4](#_Toc514623760)

[User Interface Design 5](#_Toc514623761)

[User Interface Storyboard 6](#_Toc514623762)

[Implementation 8](#_Toc514623763)

[Code Design 8](#_Toc514623764)

[Server Design 9](#_Toc514623765)

[Sample Code and Problems Solved 10](#_Toc514623766)

[Drawing the Route on the map 10](#_Toc514623767)

[Creating a GPX file for download 10](#_Toc514623768)

[Setting up the Tile Server 11](#_Toc514623769)

[Testing 13](#_Toc514623770)

[Unit Testing 13](#_Toc514623771)

[Smoke Testing 13](#_Toc514623772)

[User Testing 14](#_Toc514623773)

[Known Issues 15](#_Toc514623774)

[Future Work 15](#_Toc514623775)

[Acknowledgements 16](#_Toc514623776)

[Appendices 17](#_Toc514623777)

[Appendix #1 – Route Card Sample 17](#_Toc514623778)

[Appendix #2 – Sample Weather Download 17](#_Toc514623779)

[Appendix #3 – Sample GPX Download 18](#_Toc514623780)

[Appendix #4 – Same GPX being displayed on Garmin device 18](#_Toc514623781)

# Motivation

I am a leader in a Scout Group where we do 2-4 activities a week, these activities can range from hiking, camping, sailing, rowing etc. Out of these activities my personal favourite has always been hiking and when you lead groups in the mountains you need to have done quite a bit of preparation beforehand.   
  
The preparation would generally include, ensuring you know what the weather will be doing on the day so you can warn your group to dress appropriately. Look at the route in detail and ensure it’s suitable for your group. For example, check that it’s not too long or there’s not multiple steep climbs or one big steep climb etc. Generally, you would also fill out a thing called a Route Card (See Appendix #1 for an example). This is essentially a table that shows all the details of your route so you can give it to someone who is not coming with you, just in case of an emergency. The last part of preparation is always how are you going to get there, the mountains are rather remote and often difficult to get to so having directions on hand is essential.

The other side to doing all this preparation is you generally need to look up each of these pieces of preparation separately as there is no place that allows you to do it all at once. The other thing is that if you want to use a GPS you will have to recreate your route you did for your route card on your GPS software or on the device directly which is also very time consuming.

With all this in mind, I wanted to create something that would allow users to have a one stop shop for searching a route, seeing information like weather and the profile/elevation of the route all in one place while also being able to download the route so they do not have to recreate the route again.

# Glossary

**OSM** - OpenStreetMap

# Research

Before starting developing this project I spent a considerable amount of time researching what data was available from OpenStreetMap and how to interact with it.   
  
I also spent time researching how OpenLayers worked, what it’s architecture is and how the different layers of the map interacted with each other.

A close up of text on a white background

Description generated with high confidence

Some other parts I spent significant time on was researching was what sources were available for getting data.

# Design

“Has the project been designed to a high standard?   
Does the project clearly demonstrate the use of appropriate design principles?   
Is the design appropriate for the given problem? “

## User Stories

When designing Scout Ahead I made some sample Users to try to help me get a better understanding for the users that I feel would be the main type of people looking to use my project.

## User Interface Design

When considering exactly how my website should be designed I came up with various layouts for the website and then asked a variety of users to choose from the following designs.

Design #1 Design #2

A close up of text on a white background

Description generated with high confidence A close up of text on a whiteboard

Description generated with high confidence

Design #3 Design #4

A close up of text on a white background

Description generated with very high confidenceA close up of text on a whiteboard

Description generated with high confidence

The survey gave the following results. Showing design #4 being the most appealing with 36% of users.

A picture containing screenshot

Description generated with high confidence

### User Interface Storyboard

I then took Design #4 and developed it further by storyboarding exactly what I wanted to happen after each click or user interaction. Using this visualisation, I could start building a wireframe and create a basic structure for my code.

Design Storyboard

A close up of text on a whiteboard

Description generated with high confidenceA close up of text on a white background

Description generated with very high confidence

Initial UML Class Diagram

A screenshot of a social media post

Description generated with very high confidence

# Implementation

## Code Design

Process Flowchart

![A screenshot of a cell phone

Description generated with very high confidence]()

## Server Design

Architecture Diagram

A screenshot of a cell phone

Description generated with high confidence

# Sample Code and Problems Solved

Some of the more difficult sections of my project were the following:

## Drawing the Route on the map

To draw the route on the map I needed to add a new vector layer to the map. The vector layer sits on top of the base layer of the map and is the layer where all icons etc can be displayed. I initialised this layer in map.js, then once the finish location has been selected, displayRoute() would be called.

This calls for routeShape file from [www.mapquestapi.com](http://www.mapquestapi.com). The routeShape file is a file that contains all the latitude and longitude points of the selected route. I then transform those points to EPSG:4326 which are the type of latitude and longitude we are used to seeing and the type that is easy to plot on our map.

A screenshot of a cell phone

Description generated with high confidence

We add these points to our vector layer, add the vector layer to the map, find the size of the map and determine what level zoom to display.

A picture containing text

Description generated with high confidence

## Creating a GPX file for download

In order to create a GPX file I found a sample Garmin GPX file and copied its tag names and attributes. I created a file with the root tag being <gpx> I could not find a way to app the xml prolog so I had to do this a roundabout way.

I setup the initial file then, got the translated route shape files that we collected earlier and added all the latitude and longitude information. I then took the elevation data we downloaded earlier and added that information for each latitude, longitude pair.

A screenshot of a cell phone

Description generated with high confidence

After that loop finished I converted the XML document to a string, added the xml prolog to the start of the file and created a download link for the file.

A screenshot of text

Description generated with high confidence

## Setting up the Tile Server

Setting up the OSM Tile Server was quite a finicky process. I started by setting up an Ubuntu Linux v18.04 virtual machine, I then started by installing PostgreSQL and adding my account to be the owner of the table.

After that I needed various other tools to import and manage moving OSM data in a database, I used osm2pgsql. I cloned the git repository from OSM and installed it. After we installed Mapnik (which will create our map stylesheets), mod\_tile (an apache module that handles requests for tiles) and renderd (is a daemon that renders tiles when mod\_tile requests them).

After that was all installed, I downloaded the basic OSM-Carto stylesheet from their git and installed it. Once installed I converted the project to a Mapnik XML stylesheet.

Then I began to load data into my database, I got all the information for Ireland from <http://download.geofabrik.de/europe/ireland-and-northern-ireland.html> loading all this data took several hours to process.

Once processed I downloaded the shapefiles for when low-zoom country boundaries are needed. I then downloaded several fonts as some businesses use things like emojis in the shop names.

Then I started to configure my apache server. I pointed apache to all the mod\_tile configuration files, I also added timeouts for renderd in case tiles are missing from my database and then I added my website to the www directory.

Lastly, I changed the source of my Tile Sever in my map.js from default to my servers IP.

# Testing

All of tests can be found at “~/src/Testing”.

## Unit Testing

I only did a handful of unit tests because I didn’t find this type of testing suited my project very well and instead I primarily focused on Smoke Testing and User Testing.

For my Unit tests I used Jasmine and they can be found under “~/src/Jasmine”.

## Smoke Testing

For my smoke testing I used Selenium. When testing my focus was to ensure that the user cannot input anything that may be harmful or cause my application to crash.

A screenshot of a cell phone

Description generated with very high confidence

## User Testing

For my user testing I sent my website along with a link to a google form where they could log issues they had with the website to 20 Scout Leaders of varying ages and backgrounds.

A screenshot of a cell phone

Description generated with very high confidence

As a result of my user testing I found that MapQuest has trouble routing through walking paths. I also found that I had no error message displaying when there was no route available or there was an issue with the route.

# Known Issues

A full list of all the known issues with Scout Ahead can be found at my GitLab Issues Board (<https://gitlab.computing.dcu.ie/oneilf27/2018-ca400-oneilf27/boards>).

Some of the main issues is as of 19th May 2018 there is an issue with the MapQuest servers creating sessions as they expire immediately. This issue has direct effect on the functionality of my project as this means I cannot retrieve the ShapePoints File which allows me to draw my elevation chart and create my GPX file.

Another issue is the page is being reloaded after downloading the GPX file.

A screenshot of a cell phone

Description generated with very high confidence

# Future Work

Some things I would like to do if time was not a factor would be to implement my own routing algorithm so it would allow for routing through uploaded user traces and by adding a significant amount of user traces I could build frequently walked routes through mountains as most are not mapped out with walking in mind.

I would also like to implement the ability for users to draw their own routes for when users want to walk through the open mountains.

# Acknowledgements

My project uses various API’s and libraries. Here is a link to all third parties used within my project.

OpenLayers: <https://openlayers.org/en/latest/apidoc/>

MapQuest: <http://www.mapquestapi.com/>

Open-Elevation: <https://open-elevation.com/>

OpenWeatherMap: <https://openweathermap.org/api>

jsToPDF: <https://github.com/iszroil/JSTOPDF>

Bootstrap 4: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

# Appendices

## Appendix #1 – Route Card Sample

A screenshot of a cell phone

Description generated with very high confidence

## Appendix #2 – Sample Weather Download

A screenshot of a cell phone

Description generated with very high confidence

## Appendix #3 – Sample GPX Download

A screenshot of a cell phone

Description generated with very high confidence

## Appendix #4 – Same GPX being displayed on Garmin device

A close up of a map

Description generated with high confidence