SCOUT AHEAD

Technical Manual

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.

Finnian O Neill
Student Number: 13430512
Supervisor: Dr. Stephen Blott

Table of Contents

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.

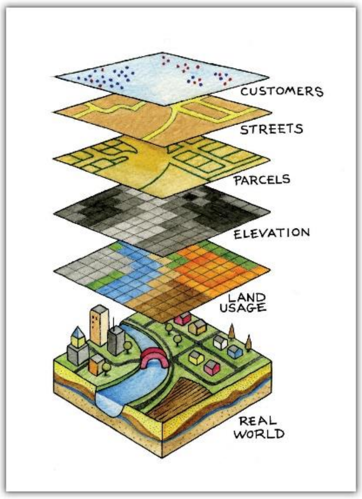
Glossary

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.



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

Design

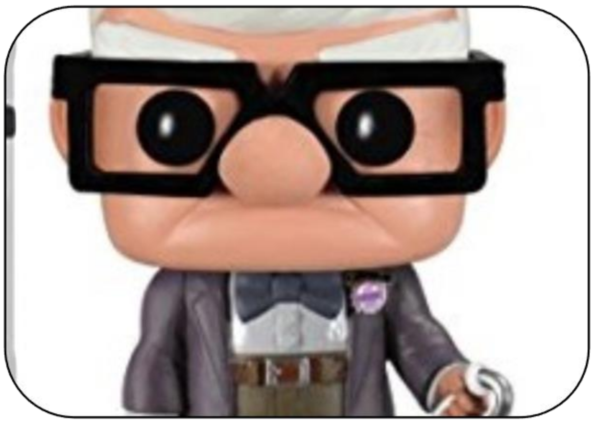
"Has the project been designed to a high standard?
Does the project clearly demonstrate the use of appropriate design principles?
Is the design appropriate for the given problem? "

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.

Name:	Shane
Age:	16
Profession:	Leaving Cert Student
Hobbies:	Sailing, Hiking, Camping
Looking for:	Easy to use, must be fast and responsive. Visually appealing.

Name:	Finn
Age:	24
Profession:	College Student
Hobbies:	Hiking, Board, Games, Beer
Looking for:	Must be fast, contain detailed information, Visually appealing



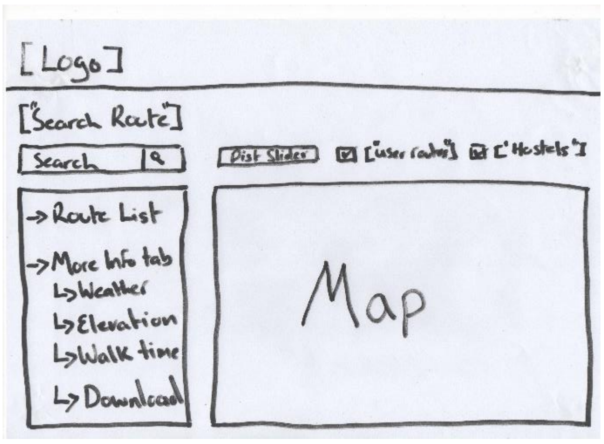
Name:	Colum
Age:	61
Profession:	Retired Salesman
Hobbies:	Walking, Swimming, Cooking
Looking for:	Easy to use, No small text, search for routes, profile, walk time

Name:	Stephen
Age:	42
Profession:	Engineer
Hobbies:	Outdoor Sports, Photography
Looking for:	Contain detailed information, child friendly, walk time? profile?

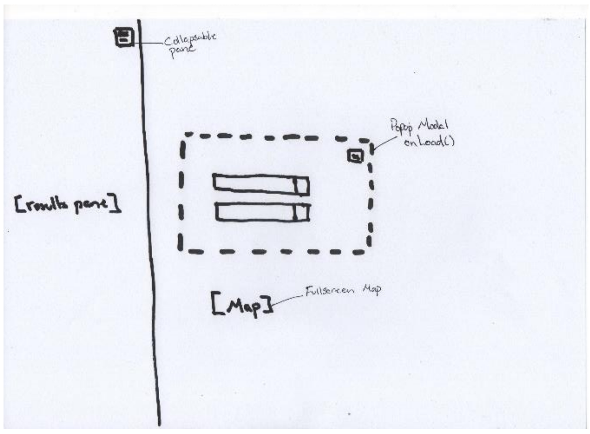
User Interface Design

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.

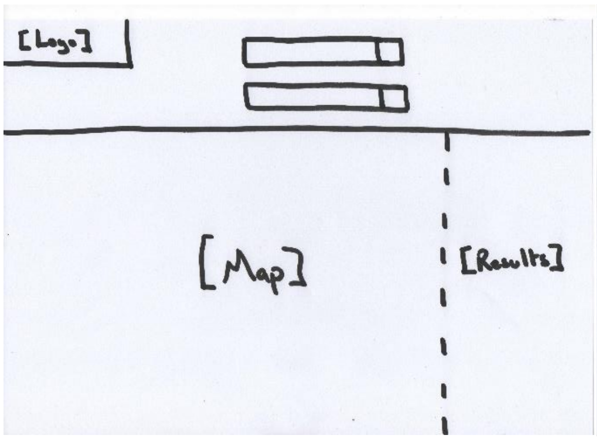
Design #1



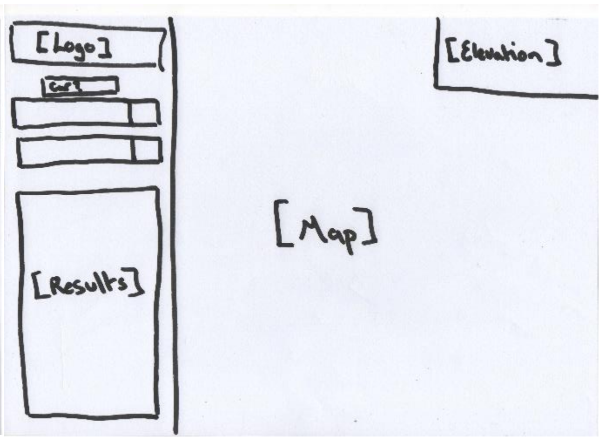
Design #2



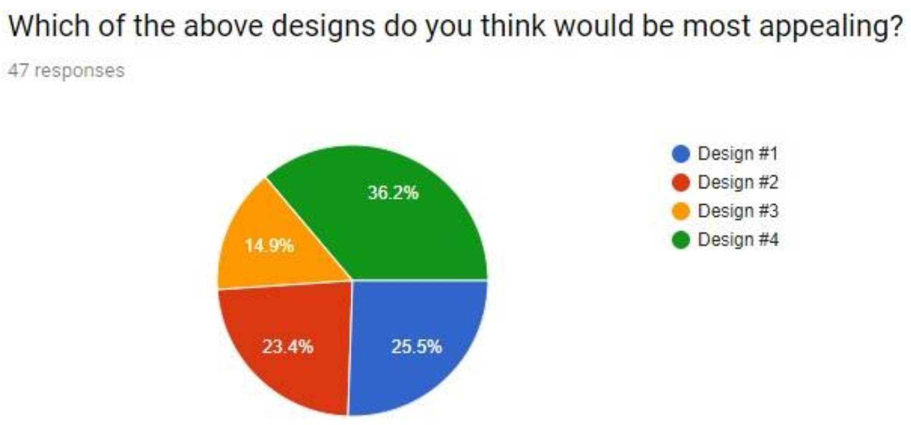
Design #3



Design #4



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

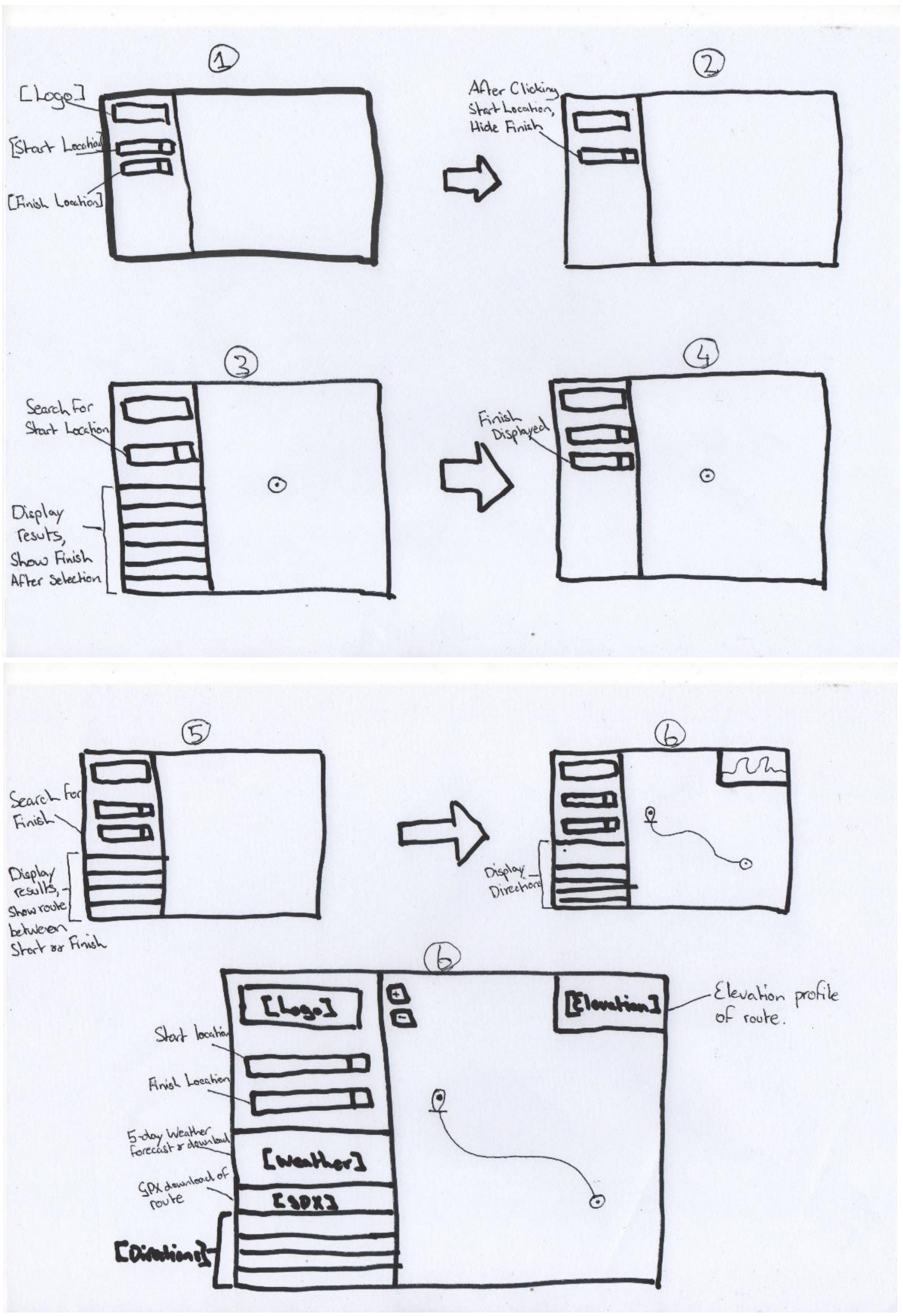

Which of the above designs do you think would be most appealing?

47 responses

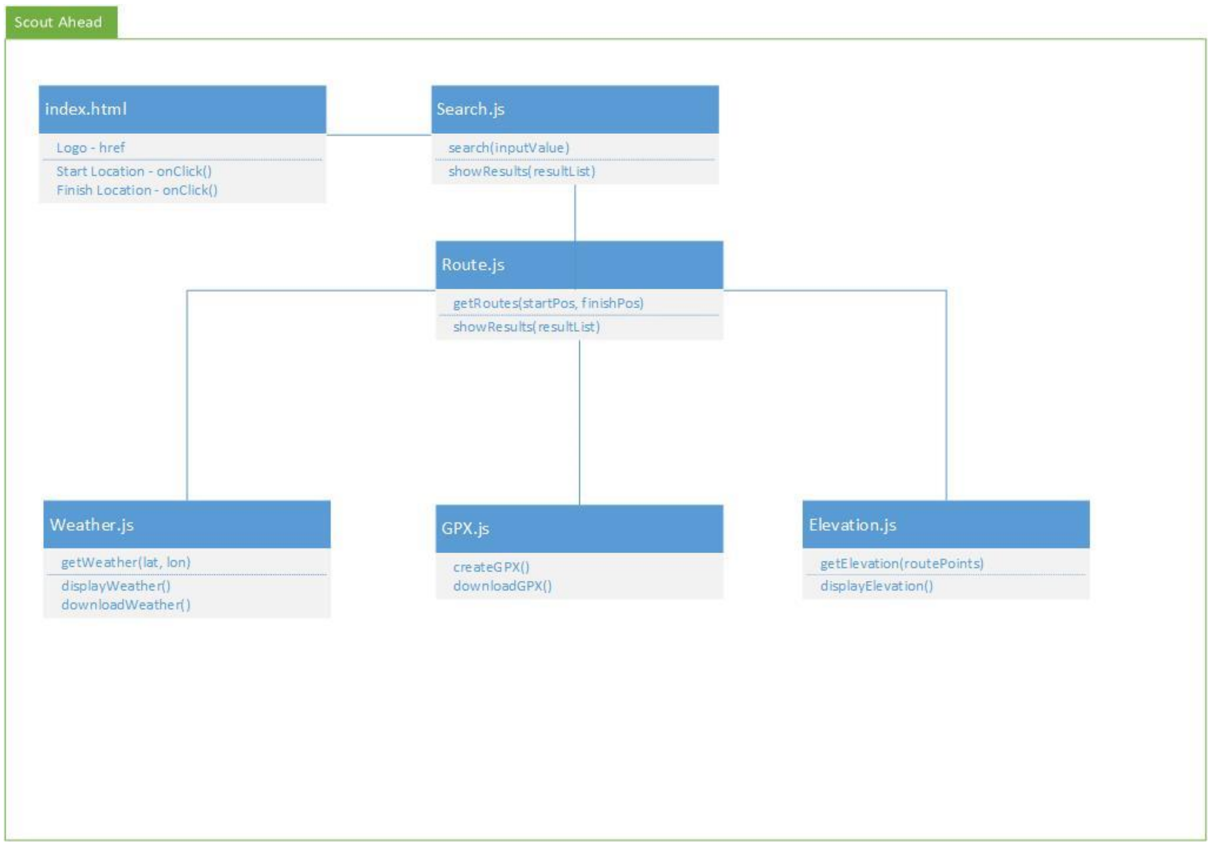
User Interface Storyboard

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.

<u>Design Storyboard</u>


1
[Logo]
[Start Location]
[Finish Location]
2
After Clicking Start Location, Hide Finish
3
Search For Start Location
Display results, Show Finish After Selection
4
Finish Displayed
5
Search for Finish
Display results, Show route between Start & Finish
6
Display Directions
6
[Logo]
Start location
Finish Location
5-day Weather forecast & download
[weather]
GPX download of route
[GPX]
[Directions]
[Elevation]
Elevation profile of route.

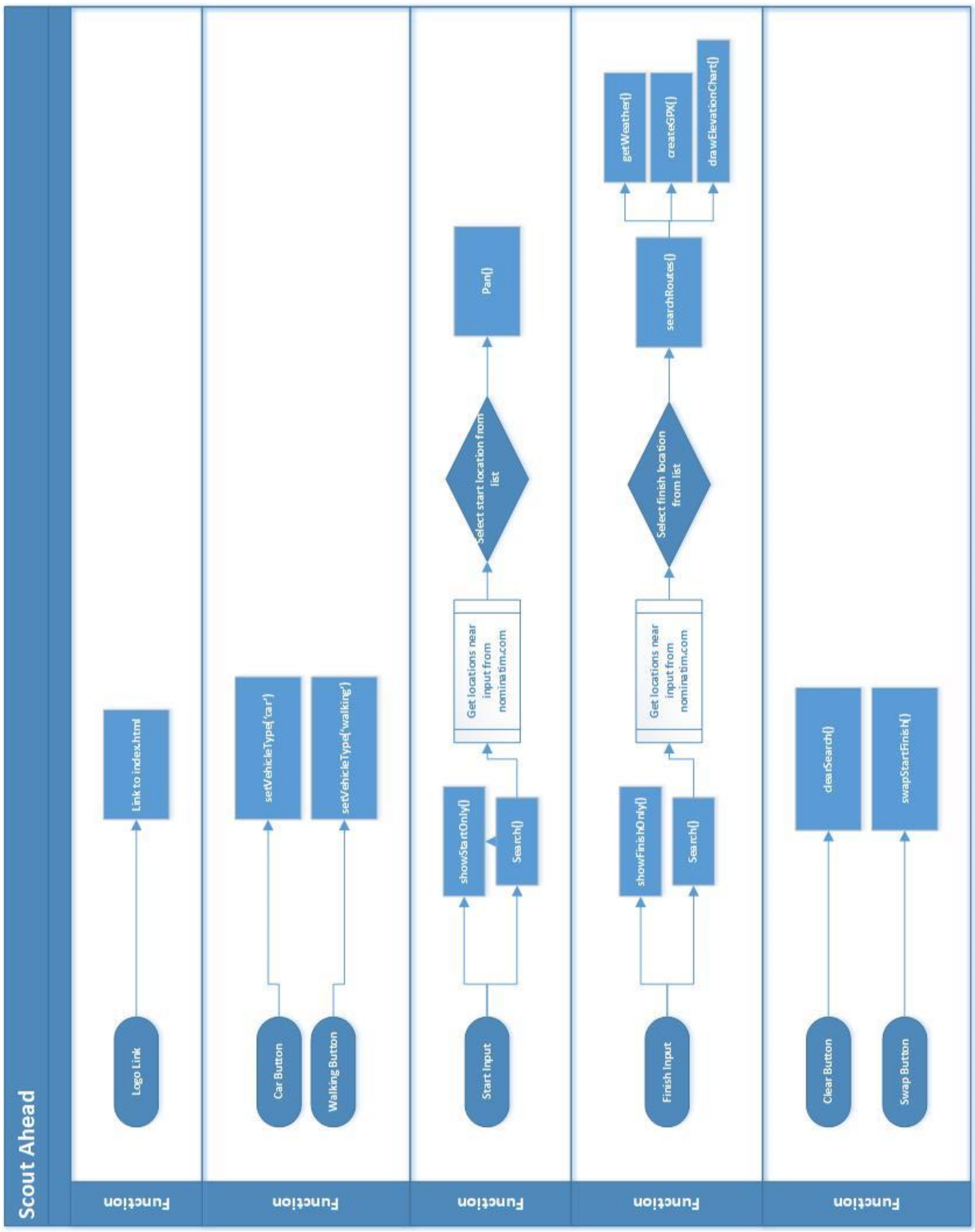
Initial UML Class Diagram



Implementation

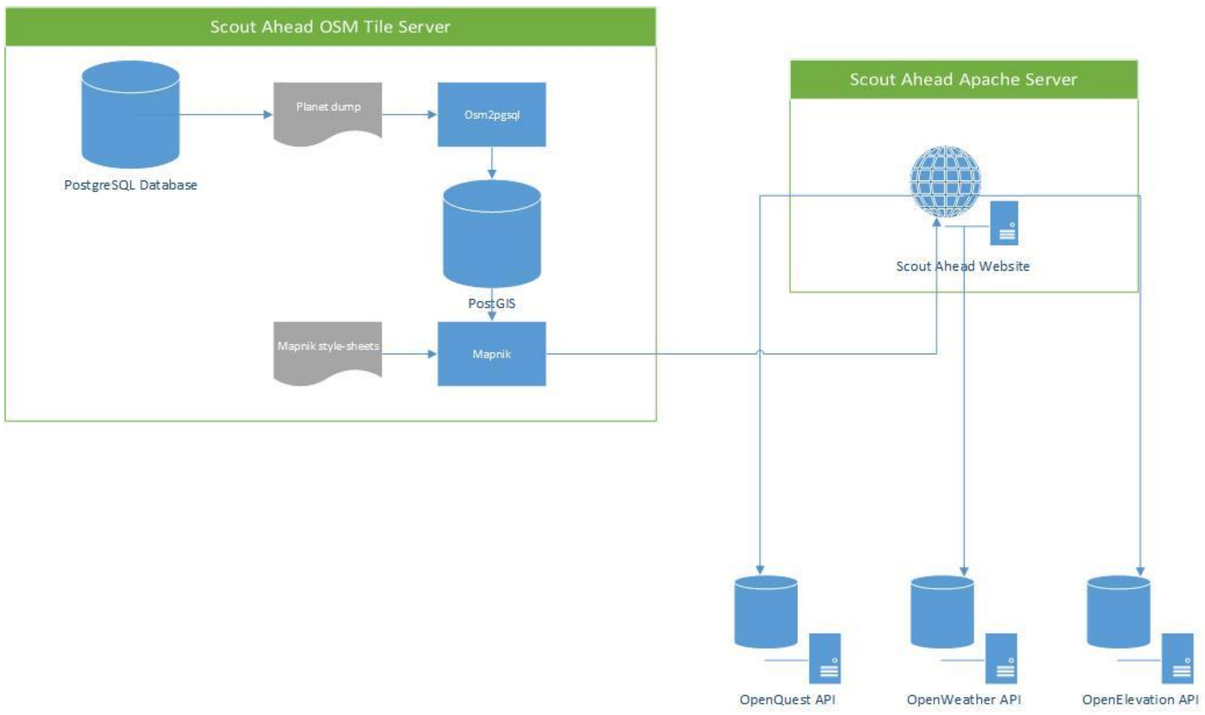
Code Design

Process Flowchart



Server Design

<u>Architecture Diagram</u>



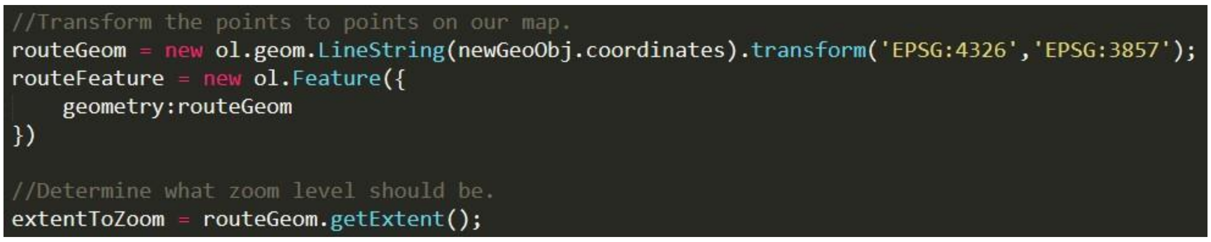
Sample Code and Problems Solved

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.

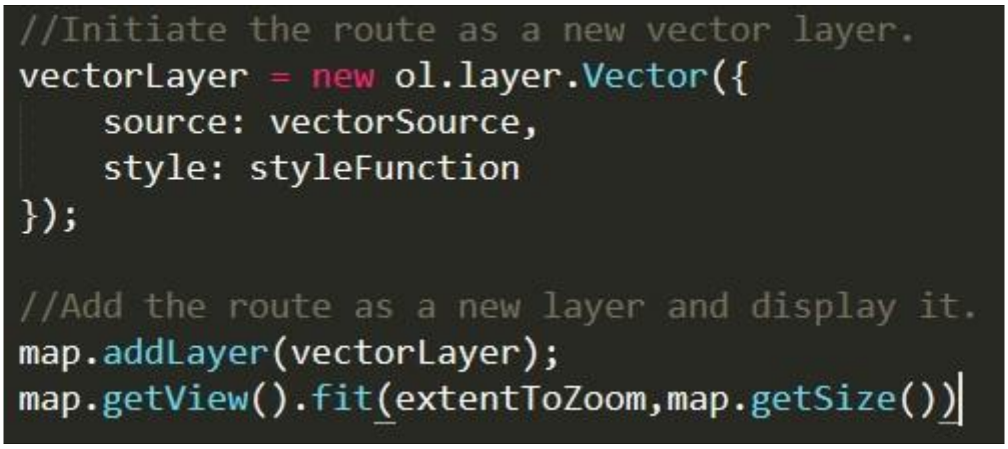
This calls for routeShape file from www.mapquestapi.com. The routeShape file is a file that contains all the latitude and longitude points of the selected route. I then transform those points to EPSG:4326 which are the type of latitude and longitude we are used to seeing and the type that is easy to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



```
//Transform the points to points on our map.
routeGeom = new ol.geom.LineString(newGeoObj.coordinates).transform('EPSG:4326','EPSG:3857');
routeFeature = new ol.Feature({
    geometry:routeGeom
})

//Determine what zoom level should be.
extentToZoom = routeGeom.getExtent();
```

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.



```
//Initiate the route as a new vector layer.
vectorLayer = new ol.layer.Vector({
    source: vectorSource,
    style: styleFunction
});

//Add the route as a new layer and display it.
map.addLayer(vectorLayer);
map.getView().fit(extentToZoom,map.getSize())
```

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.

```
var trkseg = document.createElement('trkseg');
for (var i = 0; i < shapePoints.length; i++) {
    var trkpt = document.createElement("trkpt");
    trkpt.setAttribute('lat', parseFloat(shapePoints[i].childNodes[0].childNodes[0].nodeValue));
    trkpt.setAttribute('lon', parseFloat(shapePoints[i].childNodes[1].childNodes[0].nodeValue));

    var elevTmp = document.createElement("ele");
    elevTmp.appendChild(document.createTextNode(elevationData[i]));

    var timeTmp = document.createElement("time");
    timeTmp.appendChild(document.createTextNode(Date.now()));

    trkpt.appendChild(elevTmp);
    trkpt.appendChild(timeTmp);
    trkseg.appendChild(trkpt);
}
```

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.

```
var serializer = new XMLSerializer();
var xmlString = serializer.serializeToString(xmlDoc);

xmlString = xmlString.substring(5,xmlString.length);
xmlString = xmlString.substring(0, xmlString.length-6);

xmlProlog = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>";
xmlString = xmlProlog.concat(xmlString);

download(start.value+" to "+finish.value+".gpx", xmlString);
```

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".

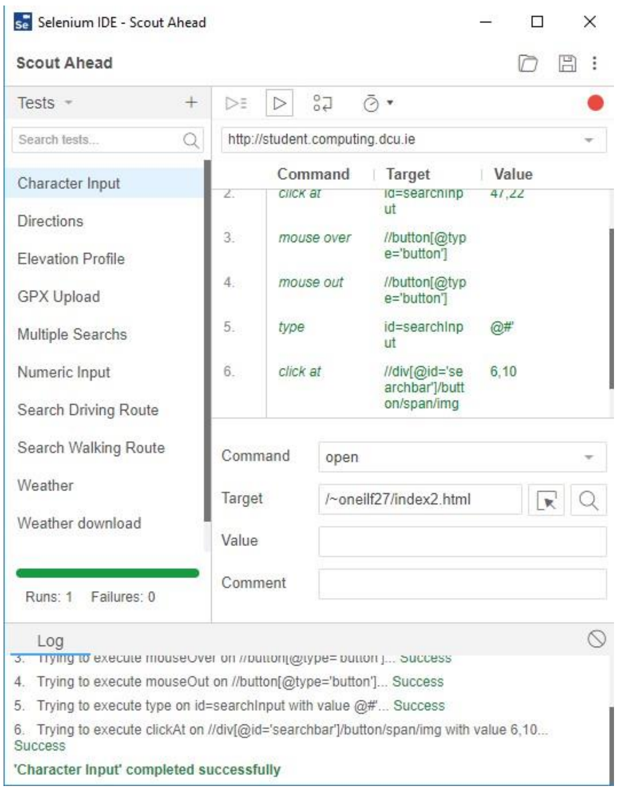
Unit Testing

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.



http://student.computing.dcu.ie

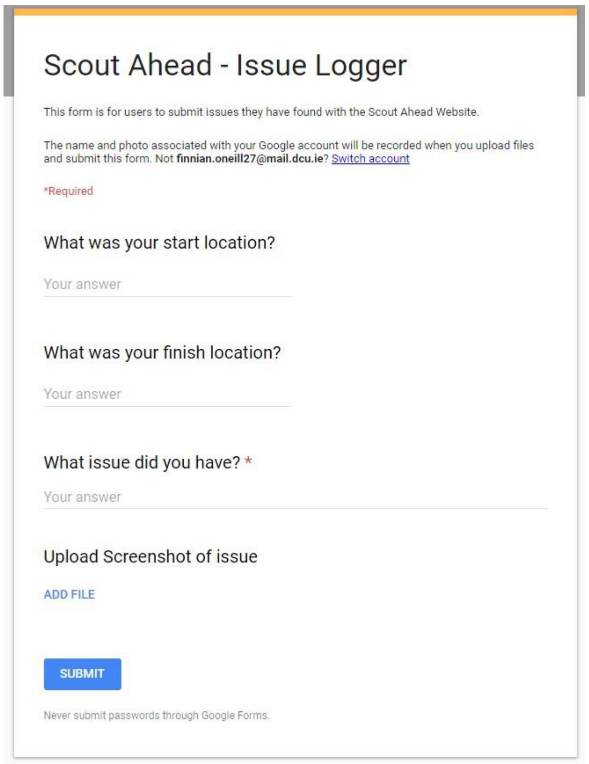
	Command	Target	Value
2.	click at	id=searchInput	47,22
3.	mouse over	//button[@type='button']	
4.	mouse out	//button[@type='button']	
5.	type	id=searchInput	@#'
6.	click at	//div[@id='searchbar']/button/span/img	6,10

Command	open
Target	/~oneilf27/index2.html
Value	
Comment	

```
3.  Trying to execute mouseOver on //button[@type='button']... Success
4.  Trying to execute mouseOut on //button[@type='button']... Success
5.  Trying to execute type on id=searchInput with value @#'... Success
6.  Trying to execute clickAt on //div[@id='searchbar']/button/span/img with value 6,10...
Success
'Character Input' completed successfully
```

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.



Scout Ahead - Issue Logger

This form is for users to submit issues they have found with the Scout Ahead Website.

The name and photo associated with your Google account will be recorded when you upload files and submit this form. Not **finnian.oneill27@mail.dcu.ie**? Switch account

*Required

What was your start location?

Your answer

What was your finish location?

Your answer

What issue did you have? *

Your answer

Upload Screenshot of issue

ADD FILE

SUBMIT

Never submit passwords through Google Forms.

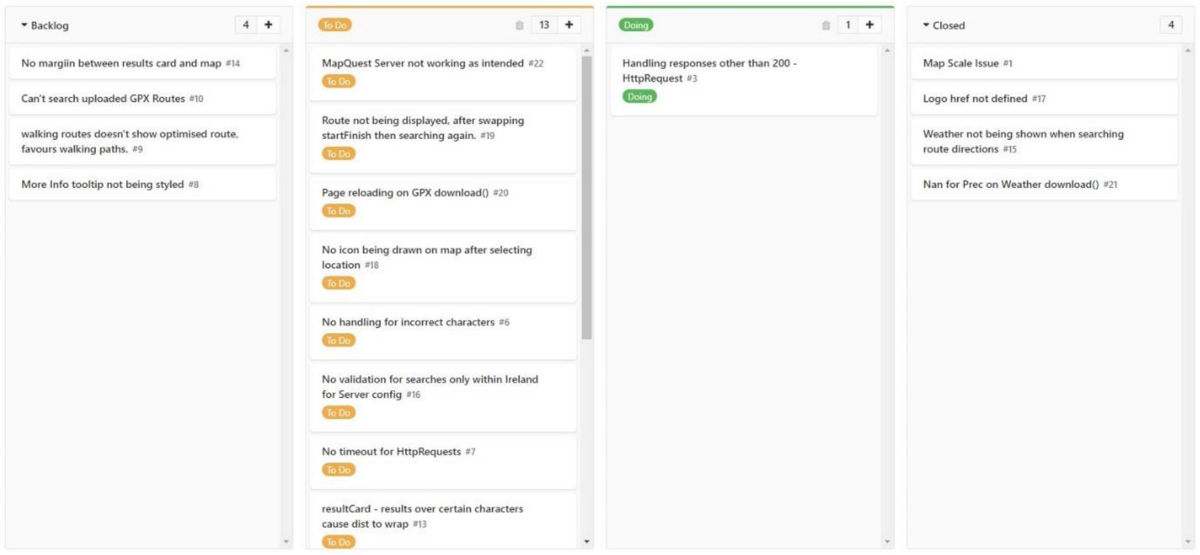
As a result of my user testing I found that MapQuest has trouble routing through walking paths. I also found that I had no error message displaying when there was no route available or there was an issue with the route.

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.



Backlog 4 +

No margiin between results card and map #14

Can't search uploaded GPX Routes #10

walking routes doesn't show optimised route, favours walking paths. #9

More Info tooltip not being styled #8

To Do 13 +

MapQuest Server not working as intended #22 To Do

Route not being displayed, after swapping startFinish then searching again. #19 To Do

Page reloading on GPX download() #20 To Do

No icon being drawn on map after selecting location #18 To Do

No handling for incorrect characters #6 To Do

No validation for searches only within Ireland for Server config #16 To Do

No timeout for HttpRequests #7 To Do

resultCard - results over certain characters cause dist to wrap #13 To Do

Doing 1 +

Handling responses other than 200 - HttpRequest #3 Doing

Closed 4

Map Scale Issue #1

Logo href not defined #17

Weather not being shown when searching route directions #15

Nan for Prec on Weather download() #21

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/

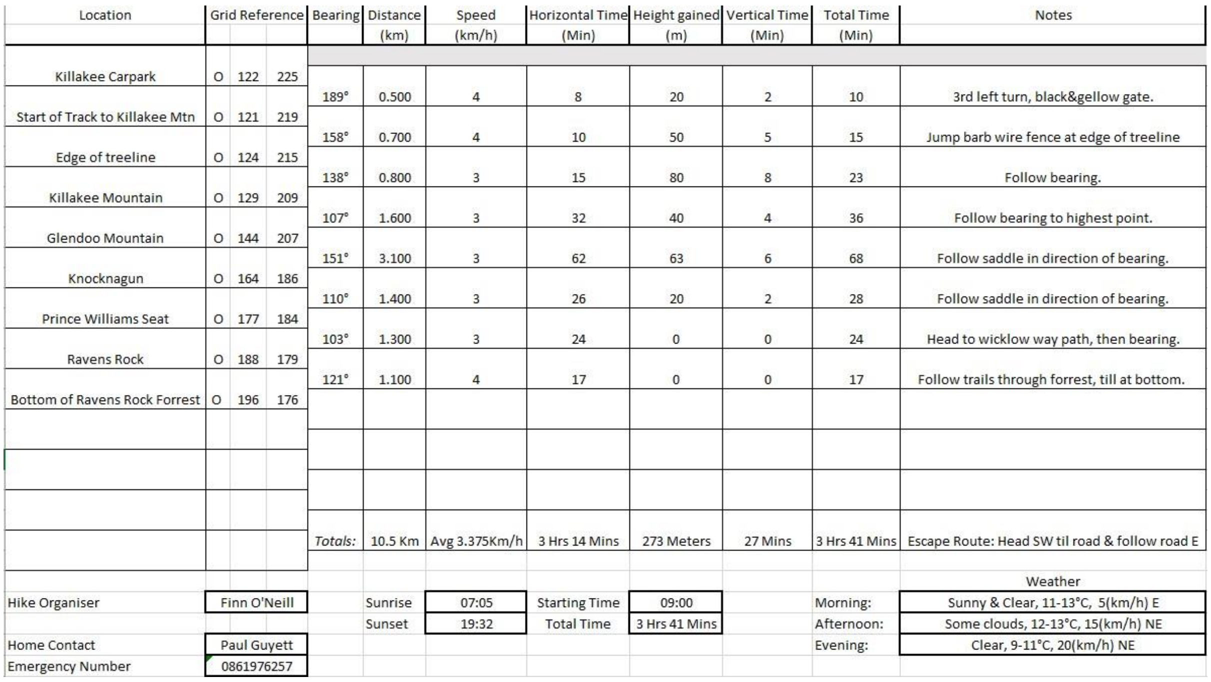
OpenWeatherMap: https://openweathermap.org/api

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

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

Appendices

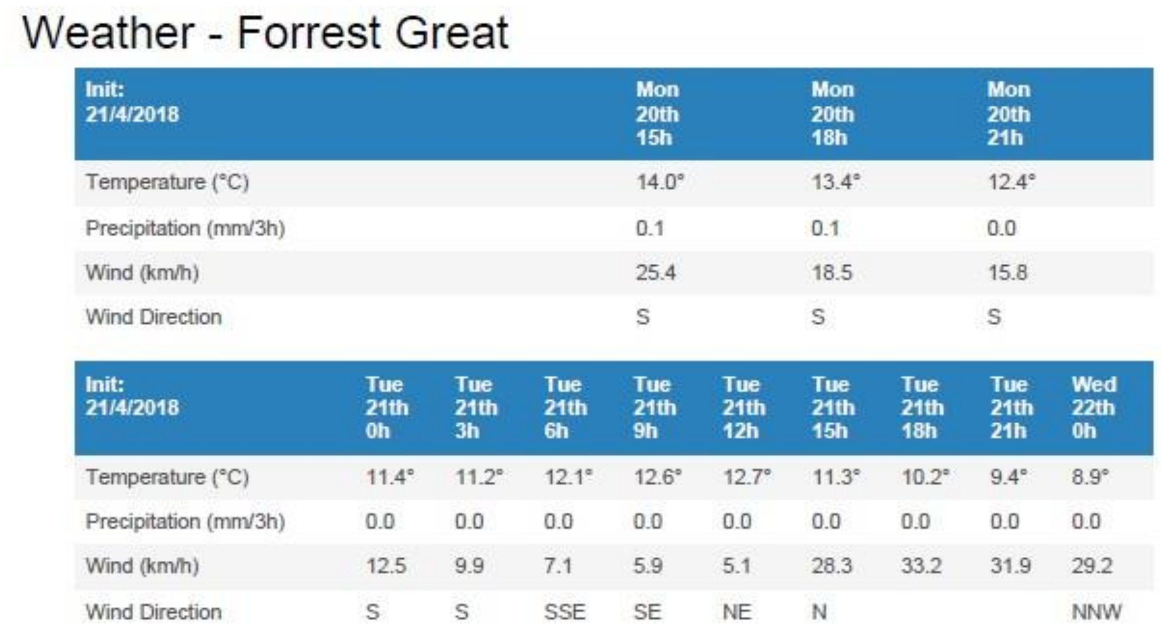
Appendix #1 – Route Card Sample



Location	Grid Reference			Bearing	Distance (km)	Speed (km/h)	Horizontal Time (Min)	Height gained (m)	Vertical Time (Min)	Total Time (Min)	Notes
Killakee Carpark	O	122	225								
				189°	0.500	4	8	20	2	10	3rd left turn, black&gellow gate.
Start of Track to Killakee Mtn	O	121	219								
				158°	0.700	4	10	50	5	15	Jump barb wire fence at edge of treeline
Edge of treeline	O	124	215								
				138°	0.800	3	15	80	8	23	Follow bearing.
Killakee Mountain	O	129	209								
				107°	1.600	3	32	40	4	36	Follow bearing to highest point.
Glendoo Mountain	O	144	207								
				151°	3.100	3	62	63	6	68	Follow saddle in direction of bearing.
Knocknagun	O	164	186								
				110°	1.400	3	26	20	2	28	Follow saddle in direction of bearing.
Prince Williams Seat	O	177	184								
				103°	1.300	3	24	0	0	24	Head to wicklow way path, then bearing.
Ravens Rock	O	188	179								
				121°	1.100	4	17	0	0	17	Follow trails through forrest, till at bottom.
Bottom of Ravens Rock Forrest	O	196	176								
				Totals:	10.5 Km	Avg 3.375Km/h	3 Hrs 14 Mins	273 Meters	27 Mins	3 Hrs 41 Mins	Escape Route: Head SW til road & follow road E

							Weather
Hike Organiser	Finn O'Neill	Sunrise	07:05	Starting Time	09:00	Morning:	Sunny & Clear, 11-13°C, 5(km/h) E
		Sunset	19:32	Total Time	3 Hrs 41 Mins	Afternoon:	Some clouds, 12-13°C, 15(km/h) NE
Home Contact	Paul Guyett					Evening:	Clear, 9-11°C, 20(km/h) NE
Emergency Number	0861976257						

Appendix #2 – Sample Weather Download

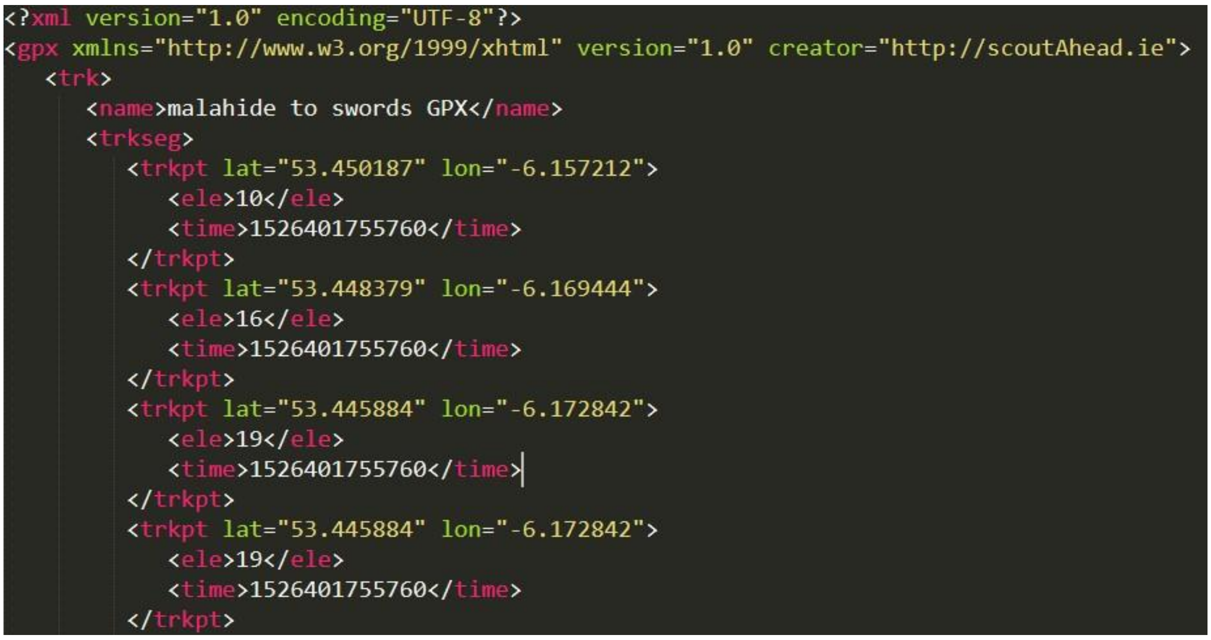


Weather - Forrest Great

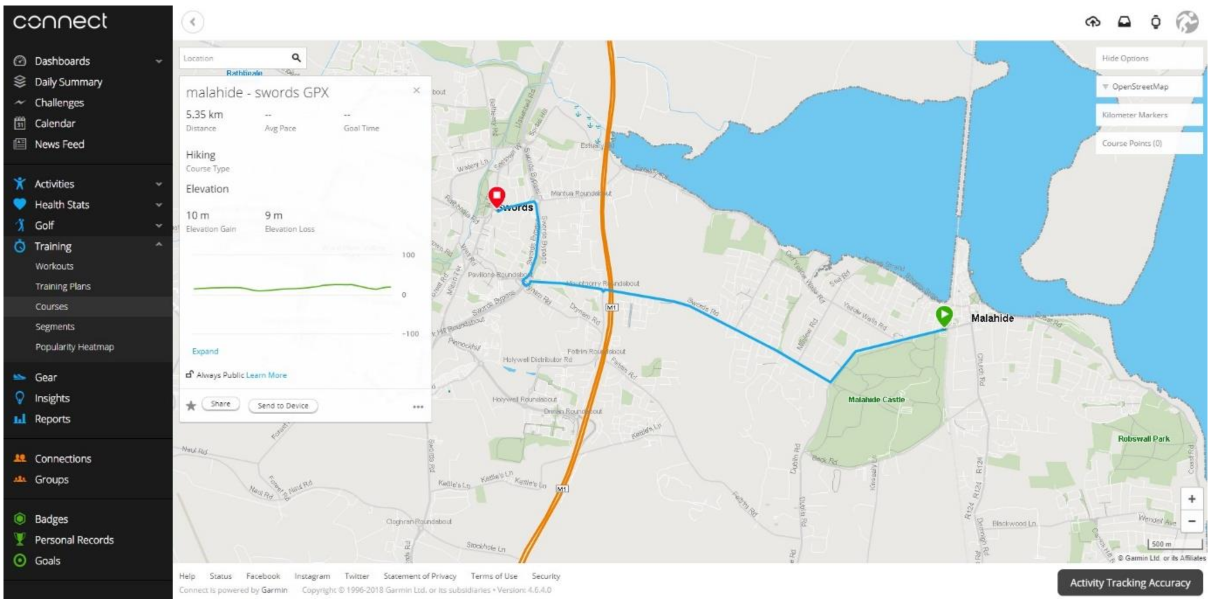
Init: 21/4/2018	Mon 20th 15h	Mon 20th 18h	Mon 20th 21h
Temperature (°C)	14.0°	13.4°	12.4°
Precipitation (mm/3h)	0.1	0.1	0.0
Wind (km/h)	25.4	18.5	15.8
Wind Direction	S	S	S

Init: 21/4/2018	Tue 21th 0h	Tue 21th 3h	Tue 21th 6h	Tue 21th 9h	Tue 21th 12h	Tue 21th 15h	Tue 21th 18h	Tue 21th 21h	Wed 22th 0h
Temperature (°C)	11.4°	11.2°	12.1°	12.6°	12.7°	11.3°	10.2°	9.4°	8.9°
Precipitation (mm/3h)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wind (km/h)	12.5	9.9	7.1	5.9	5.1	28.3	33.2	31.9	29.2
Wind Direction	S	S	SSE	SE	NE	N			NNW

Appendix #3 – Sample GPX Download


```
<?xml version="1.0" encoding="UTF-8"?>
<gpx xmlns="http://www.w3.org/1999/xhtml" version="1.0" creator="http://scoutAhead.ie">
   <trk>
      <name>malahide to swords GPX</name>
      <trkseg>
         <trkpt lat="53.450187" lon="-6.157212">
            <ele>10</ele>
            <time>1526401755760</time>
         </trkpt>
         <trkpt lat="53.448379" lon="-6.169444">
            <ele>16</ele>
            <time>1526401755760</time>
         </trkpt>
         <trkpt lat="53.445884" lon="-6.172842">
            <ele>19</ele>
            <time>1526401755760</time>
         </trkpt>
         <trkpt lat="53.445884" lon="-6.172842">
            <ele>19</ele>
            <time>1526401755760</time>
         </trkpt>
```

Appendix #4 – Same GPX being displayed on Garmin device



malahide - swords GPX

5.35 km Distance
-- Avg Pace
-- Goal Time

Hiking
Course Type

Elevation
10 m Elevation Gain
9 m Elevation Loss

Always Public Learn More

Connect is powered by Garmin   Copyright © 1996-2018 Garmin Ltd. or its subsidiaries • Version: 4.0.4.0