

[visitour.azurewebsites.net](http://visitour.azurewebsites.net/)

Project Report

Author: Declan Feehily

S00145584

Software Development

PRJ400

May 2017

Project Supervisor: Colin O’Connor

B.Sc. in Computing (Software Development) Level 8 Year 4

Contents

[***Chapter 1*** Overview 4](#_Toc481353656)

[***Chapter 2*** Goal and Scope 5](#_Toc481353657)

[2.1 Goal 5](#_Toc481353658)

[2.2 Scope 5](#_Toc481353659)

[Characteristics of the Web Application 5](#_Toc481353660)

[Acceptance criteria 5](#_Toc481353661)

[Project Exclusions 6](#_Toc481353662)

[Constraints 6](#_Toc481353663)

[***Chapter 3*** Areas of Research 7](#_Toc481353664)

[3.1 Competitors 7](#_Toc481353665)

[3.2 Xamarin Android 8](#_Toc481353666)

[3.3 Microsoft Azure Cloud Computing 9](#_Toc481353667)

[3.4 Microsoft SQL Server 9](#_Toc481353668)

[3.5 ASP.NET MVC 10](#_Toc481353669)

[3.6 Google Maps API 11](#_Toc481353670)

[3.7 JavaScript/jQuery 12](#_Toc481353671)

[3.8 Toastr.js 13](#_Toc481353672)

[3.9 Etc.. 13](#_Toc481353673)

[***Chapter 4*** Design and User Interface 13](#_Toc481353674)

[4.1 User Interface 13](#_Toc481353675)

[Prototyping 13](#_Toc481353676)

[Implemented User Interface 21](#_Toc481353677)

[4.2 Overall Architecture 22](#_Toc481353678)

[4.3 Database Design if applicable 23](#_Toc481353679)

[4.4 Class Diagrams 24](#_Toc481353680)

[***Chapter 5*** Scrum 24](#_Toc481353681)

[5.1 Sprints 24](#_Toc481353682)

[5.2 Methodologies 24](#_Toc481353683)

[***Chapter 6*** Time Management 25](#_Toc481353684)

[***Chapter 7*** Problems Encountered 25](#_Toc481353685)

[***Chapter 8*** Results 25](#_Toc481353686)

[***Chapter 9*** References 27](#_Toc481353687)

[***Chapter 10*** Appendices 28](#_Toc481353688)

# Overview

Visual Itinerary Tour (VISiTOUR) is a Web based application that helps both tourists and locals to plan a day’s itinerary of places to see and go. In today’s world tourism has become a very broad category and is much harder to catering to those on their travels. You still have regular tourists who plan out their holidays or trips in advance before they leave but you also have those who may unexpectedly find themselves with a limited amount of time on their hands and wish to go sightseeing.

This Web application aims to provide for all types of travellers in planning out their ideal day’s itinerary that’s suited to their own needs, preferences and circumstances. Users will be able to choose their itinerary’s starting location from either through search or by current location. This will adjust the locations on display for a user’s to pick from that are of relevance and with reasonable proximity to their chosen location.

From there users can start choosing locations that take their interest or further filter down their choice through categories for example like swimming, hiking, restaurants or historic etc. Once they have selected all the places they hope to see and confirmed their selection the application will then show the user their intended route to take for their trip with a marker on each destination. It will also show a detailed route break down of each stop on the itinerary. This will show details like addresses, distance and rough estimate time of traveling to next stop.

The application will also cater to those that may like sightseeing but aren’t as adventurous or would prefer an expert’s opinion of places to go and see. For this the application provides Curated Packages. These are pre-defined location collections themed on certain interests or ideal itinerary trip categories like scenic, educational, A Day in the Town or Editor’s Choice. These would be created by site admins of the site, local tourist boards or from respected industry experts in the field of tourism.

Curated packages can be adjusted after selection. For example if there is something in the package that is of no interest to the user, they could remove its selection from the list of destination. The user will also be able to make additions to their selection if there is a particular destination they are wanting to see.

If the user is registered and logged into the site they are able to save their currently selected itinerary that can be accessed later at any time and can be updated at any time too. It will always appear as the first entry in curated packages if they user is logged in.

Owners of places of interest can register with the site. Once they have registered they can add their own locations to the database that will then appear for every one as a selectable destination for users to select. The rationality to include this functionality is to introduce a paid placement advertising system that would let owners pay to have their location appear more prominently on the site and be enough of a revenue income to at least cover server hosting costs and possibly become a profit in the future.

Users will access the application through any large screen device with web browser functionality. The implementation of the Bootstrap CSS framework allows the site to be viewed in a wide range of resolutions covering most desktops, laptops and large tablet devices.

The web application is created using the ASP.NET MVC framework. Along with Bootstrap the frontend is designed using HTML/CSS/JavaScript, Razor, jQuery, jQuery UI and Toastr.js JavaScript libraries. The Models of MVC are used in the backend when creating the database by using Entity Framework’s code-first migrations. The database is a SQL Server database which is, along with the web application all hosted on Microsoft Azure.

The application also incorporates a collection of Google Map API’s used to calculate Itinerary routes to display on the map and for the breakdown details between each destination.

# Goal and Scope

## Goal

The original goal of the project was to create a mobile application that would allow any user with a limited amount of time to be able to pick out a list of location near them and the application would create out a small itinerary for them to follow that could be started immediately if they so wished.

During the planning stage it was becoming unfeasible to create the application on mobile as I had no prior mobile development experience and in trying to set up Xamarin to develop with, I encountered too many problems with installation errors and following tutorials that would create errors when they shouldn’t. I decided it was best to move away from this plan as it would become a major hindrance if I were to continue. Instead I chose to develop it as a web application. The change in platform also altered the goal to not only cover the aim previously mention but also to users that may want to plan their itineraries in advance and thus would allow a greater coverage in the tourism market. Additionally this change, from what would have used small mobile devices to using bigger computer screens gave myself more screen retail to use in providing more detail and customization options to the user.

The other goal that will always be a constant no matter what platform I had chosen to develop on was that it is a forever expansive application and isn’t limited to just one place. It is designed so that it can start off as a small local information hub for just a small area to be used by locals or people just visiting that location but once attention, interest and possibly revenue increase the application itself will be able to dynamically grow at a steady and balanced rate in accordance with these positives.

## Scope

This project solves the problem of users visiting or planning to visit areas where they may want to get the most out of their time available to them or wish to have their journey orderly planned out to get the best experience of their trip’s possible.

### Characteristics of the Web Application

* Tourists/Locals/Travellers use the itinerary creation page to plan out their journey whether it be for use in the future or right then and there.
* Owners of places of interest can submit their location to be included as part of the application.
* The application is responsive to most devices, providing a satisfactory visual experience to any user.
* Users after having completed the journey can rate their experience of each destination which would be used in calculating an average rating system visible to all users as a star rating for each location.

### Acceptance criteria

The following criteria serves as satisfactory conditions for the web application to meet so that it is sufficiently complete to be used by end user and can go live :-

* A user can access the site as a guest and use the basic map and route functionality without needing to sign in/log in/register.
* A user can search for their starting location using the search box and the map will reflect their selection.
* A user can select locations from the side list and when then submit their selection for calculation and the app returns the route on the map and the itinerary breakdown between each destination is shown on the side.
* A user can select one of the Curated Packages and this auto-selects the locations within the package and auto calculates out the route.
* Once logged in as a regular user, they can save their current selected itinerary to be accessed later.
* A user is able to select their saved itinerary and have it auto calculate out the route.
* A user can log in as an owner of place of interest and add locations to the database.

### Project Exclusions

* I wasn’t able to create a user review system of each location that could be used as a search criteria when choosing due to time constraints.
* I wasn’t able to set up an advertisement payment system for owners to get more prominence in results for a certain fee.
* I wasn’t able to customise the map appearance with custom markers to make the application feel more personalized and stood out. This could have been just using a different marker compared to google maps default marker or unique markers depending on the location e.g. swimmer icon for beach areas.
* I wasn’t able to implement the editing of previously made location in the database due to the functionality breaking at one point without myself knowing and not having enough time left to assess the error and fix it.
* I wasn’t able to get the filtering system of locations to function correctly as intended and thus did not reach my criteria of being done and was excluded.
* I wasn’t able to create the users rating system of locations due to time constraints.

### Constraints

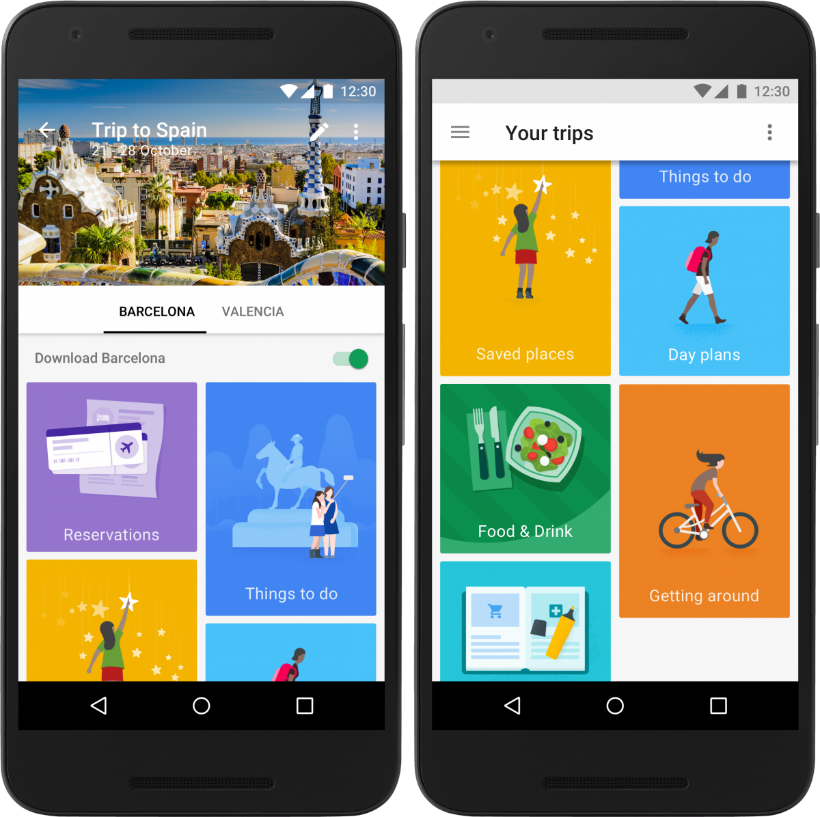
* One of the major constraints was the inability to create this application for mobile devices due to mobile development inexperience, constant errors in trying to install Xamarin Android for Visual Studio and at the time lack of or at least hard to find help and support online to simple problems when trying to create tutorial mobile applications. Solving errors sometimes took uninstalling and re-installing Xamarin with different components selected. This proved to be a time waster due to Visual Studio being a 7GB download installation already but Xamarin would increase this to 20GB.
* During writing of this report, I was collecting some of the diagrams required and for the database I had some difficulty. I discovered that SQL Server Databases on Azure don’t have implementation for creating databases diagrams. After some reading online the best solution was to script the entire database, create it locally and then create the diagram from there.
* Time was a major constraint during this project. I believe I could have created a much better and robust application that would have included most if not all of the mention exclusions above if I was given more time or if I only had this project to think about and not my many other assignments.
* The errors in connecting to the SQL Server database due to having to set up IP address in the rules of the firewall, ate into a lot of my time. It was very hard to diagnose in the beginning especially when it only occurred in some locations and not others which at the time gave me no indication as to what the changing aspect of locations was causing it to work sometimes and sometimes not.
* For myself a big time sink is anytime I am dealing with JavaScript/jQuery code. It is the hardest language I have ever encounter in my time of learning software development. At times I would spend hours trying to fix an error that would occur in the code that I would write. This was not helped when after eventually getting into the Google Chrome Debugger (the only way that it can be debugged) I would be met with where the errors occurred but be given no meaningful information as to what caused the problem. This would lead to spending ages trying to find answers to my problem which more often than not would be a simple syntax or position error that took a matter of seconds to fix.

# Areas of Research

## Competitors

My initial research of travelling apps lead me to a clutter of different ones and a lot of places just showing lists of travelling apps. While some had similar functions as to what I was planning none seemed to meet the need of those who may want to plan out an itinerary immediately for where they were, most just seemed to be a holiday planner between cities for flights etc. and not locally customisable.

It wasn’t until about a month or two into development that my supervisor Colin sent me a link on a new Google Application called Google Trips. They had launched it near the end of September and its main focus was similar to what I was hoping to achieve. This shows how there is a considerate market for this kind of application if Google is only recently entering the market.

 (Jani, 2017)

## Xamarin Android

I had originally planned to develop this application using Xamarin Android due to the projects original goal of using locations then and there which would have been best suited on a mobile device. Recently Microsoft bought Xamarin and integrated it under their Visual Studio Collection which made the option all the more appealing to try out and use but unfortunately it became less and less appealing as I tried to set it up.

I found it very difficult just to get Xamarin installed due to its huge size of around 13GB alone and some components that wouldn’t install correctly. Eventually after trial and error I manged to get it to install and test out a downloaded app from their site to see if it would run correctly.

During this I ran into the problem of getting it to display as none of the premade virtual devices would run the app nor could I find a tutorial that definitively told me how to go about doing it so. After consideration I thought that if I could debug it on my own device this should be fine, which I was able to set up with little effort.

The last step I took to try it out was to follow the step by step guide of creating a simple Hello, Android application to understand the basic fundamentals of application development. (Xamarin, 2017) This again proved a hard task where I would follow the instructions carefully but it would throw errors on the xml page design of the code and searching the error message left me with no options. To test I even tried coping and pasting their code exactly but to no avail.

This code I tried over and over to get correct but it would tell me some of the android xml attributes would be incorrect or where they referenced did not exist. This shouldn’t have been the case with such a simple section of code. All this displays is a pre-filled text area and a button. Items you could also just drag straight from the Visual Studio Toolbox but still wouldn’t work.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:text="Enter a Phoneword:"

android:textAppearance="?android:attr/textAppearanceLarge"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/textView1" />

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/PhoneNumberText"

android:text="087MMAJAMM" />

<Button

android:text="TranslatE"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/TranslateButton" />

<Button

android:text="Call"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/CallButton" />

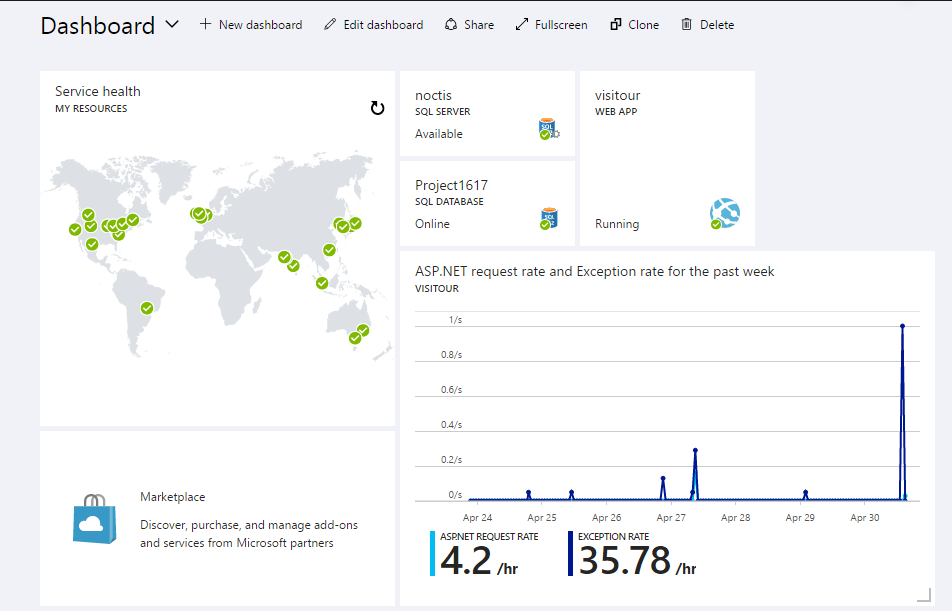
</LinearLayout>

I found out over time that the tutorials weren’t all up to scratch as they didn’t cover common errors that may occur and how best to solve them. Which is to be expected if someone is learning these simple tutorials they are more than likely new at it. (Xamarin, 2017) At this point I decided continuing further down the road and moved onto a different platform.

## Microsoft Azure Cloud Computing

Once I had chosen the platform I would develop on I knew I wanted to set up the hosting features of the Web application, SQL Server and SQL Database. Last year in my group project we had great difficulty in acquiring Azure Students key to gain access to some of the resources we needed. This included waiting months for the keys to be sent and only to discover the account didn’t have access to the SQL Server and SQL Database resources. Although reading more into new offers provided by Microsoft I discovered the newly offered Visual Studio Dev Essentials range of products to help out all developers. (Microsoft, 2017) This got me past all the setbacks of hosting in last year’s project and set me to get started straight away.

Talking with a past class mate who eventually managed to host part of our project last year on azure told me the basics of hosting on azure to get started and some of the errors he first was challenged with. This along with my supervisor who was able to find a free Microsoft Azure Essentials eBook on the Fundamentals of Azure (Collier & Shahan, 2017), allowed great ease in the setup of all the services I required and was live immediately from creation.



Further into development I had problems of not being able to connect to the database with access denied error messages. Looking into it I discovered that the firewall setting for IP address for accessing the database needed to be updated accordingly to where you were located or open them all to allow access to all. (Microsoft, 2017)

## Microsoft SQL Server

Most of what I know about SQL already came from the modules during years 2 and 3 but I was a little rusty with not having coded in T-SQL almost a year ago. I decided to go back over assignments I had done to for a refresher. Eventually I had to research into data types within SQL that I hadn’t used before like Var-binary to understand its behaviours and constraints in using those types. Var-binary was needed for the storing of images within the database. (Microsoft, 2017)

To be able to save a picture into a Var-binary column I had to convert the inputted picture by the user into a byte array and thus was in a suitable data type to be saved to the database.

HttpPostedFileBase file = Request.Files["iPicture"];

byte[] imageBytes = null;

BinaryReader reader = new BinaryReader(file.InputStream);

imageBytes = reader.ReadBytes((int)file.ContentLength);

I also had to find the best decimal datatype that would coincide with a C# type as sometimes conversion between the two can be difficult. An example of this was the storage of latitude and longitude co-ordinates of locations. During development I found numbers would be truncated or not be saved at all to the database. I eventually found the solution when reading into numerical data types for the correct one I should be using. (w3schools, 2017)

## ASP.NET MVC

Most of the knowledge I used in creating the MVC Web application came from code I had created and notes written from classes and assignments initially during my time previously in the Software Development Course.

Creating most of the tables within the database came from using Entity Frameworks Code First. This comes from taking classes created within the project solution and are turned into tables. Below shows the Locations class created in C# included with a Database Context connection to the database.

public class LocationViewModel

{

[Key]

public int Id { get; set; }

[Required]

public string Name { get; set; }

[Required]

public float Latitude { get; set; }

[Required]

public float Longitude { get; set; }

[Required]

[Display(Name = "Picture")]

public byte[] Picture { get; set; }

public string Description { get; set; }

[Required]

public string OwnerId { get; set; }

public string Website { get; set; }

[Display(Name = "Category")]

public int CategoryId { get; set; }

}

public class LocationContext:DbContext

{

public DbSet<LocationViewModel> Locations { get; set; }

public DbSet<PlannedTripsModel> Trips { get; set; }

public LocationContext() : base("DefaultConnection") { }

}

One area I did have to read into was refreshing of partial views which had never worked for me in my past assignments and made a determination to get it to work. Reading up on examples and people with similar errors I was able to find out I could refresh the views in combination with simple jQuery but still never got to fully work. (stackoverflow, 2017)

$.post("@Url.Action("AddPlace", "Owner")", { data: test.toString() }, function (data2) {//alert(data); });

I did find similar results with Ajax but once again my luck with Ajax continues of not getting it to work.

$("#Edit").click(function (e) {

$.ajax({

url: "@Url.Action("AddPlace", "Owner")",

type: "GET",

dataType: 'string',

data: $(this).siblings()[0].value.toString()

}).done(function (partialViewResult) {

$("#addEditForm").empty();

$("#addEditForm").html(partialViewResult);

});

});

Another area I had to research into was the processing of images being saved to database and returning the data back and getting it to display correctly. To begin with I was just saving a URL string to pictures online to be displayed for each location. Later in semester 2 when I was working on my mobile application (which I finally got Xamarin to work for) I came across the conversion of pictures through byte arrays and Var-binary data types that would allow the upload of selected picture and displaying of that picture. (Chindam, 2010)

<img id="thumbimgp" src="data:image;base64,@System.Convert.ToBase64String(item.Picture)">

## Google Maps API

This was my first time using Google Map API’s and so had to read into a lot of documentation on the varies functions and as I found all the different APIs available. (Google, 2017) To find the exact functionality you were looking for took a while due to the vast amount of APIs Google provides but it has good reason. If you needed just one small map function but were forced to implement the entire JavaScript API Library would become very cumbersome. Instead they breakdown the functionalities that are similar into API packages. For example all Street View functions would be under the Street View API or if you just wanted maps to appear that weren’t intractable nor function dynamically you can use the Static Maps API.

While I always find it hard to work with anything that uses JavaScript I found Google Maps implementation easy to understand and manipulate to how I wanted it to function. Initializing the map is just simply creating the map variable, customising its options and the attaching to the element you want from HTML.

var directionsService = new google.maps.DirectionsService;

var directionsDisplay = new google.maps.DirectionsRenderer;

var map = new google.maps.Map(document.getElementById('map'), {

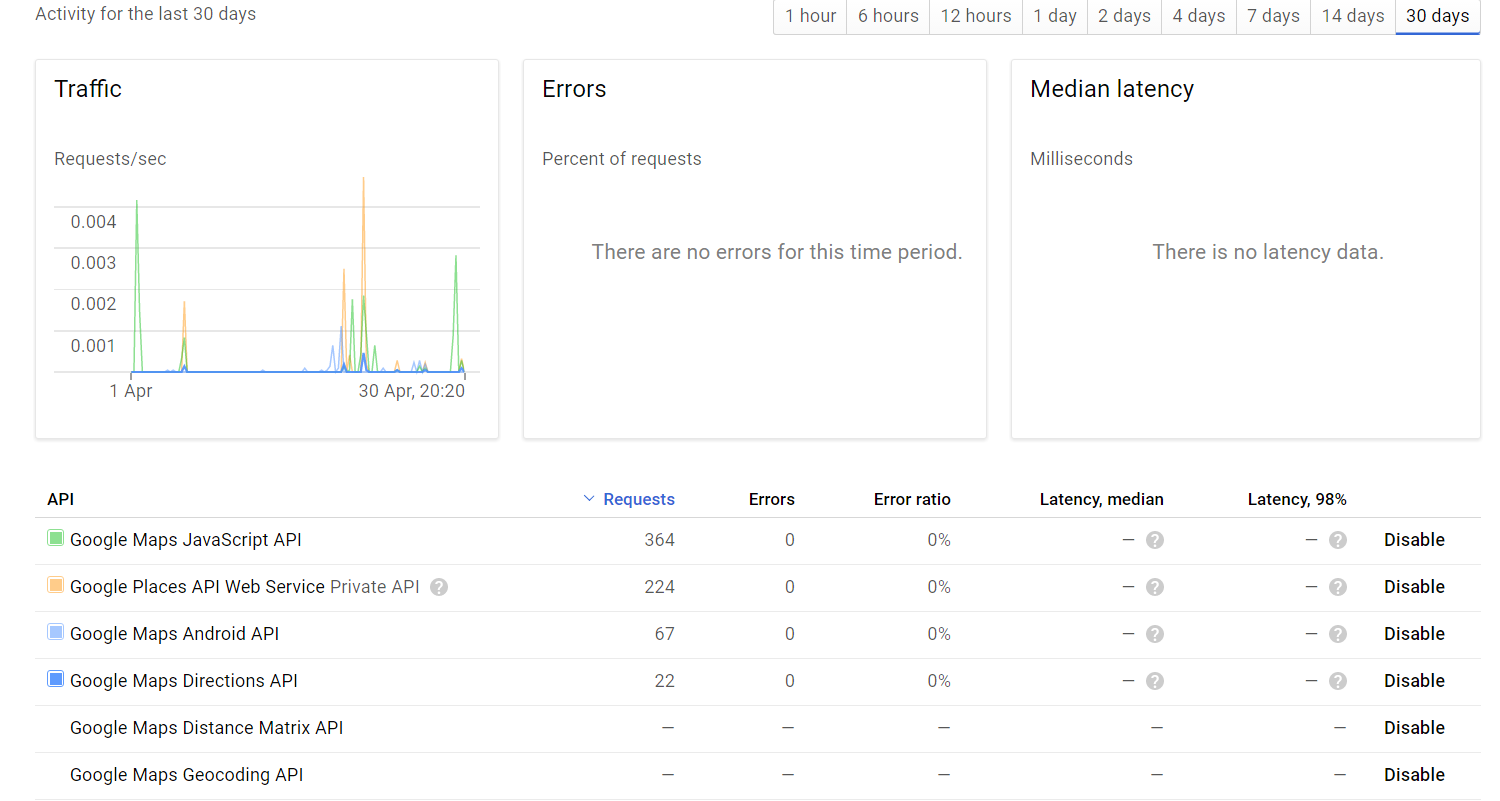
zoom: 14,

center: { lat: 54.27, lng: -8.47 }

});

directionsDisplay.setMap(map);

To use any of the APIs it required you to set up a Google account if you didn’t have one already, create a project and then set up your own personal API KEY for the project. (Google, 2017) This would then be used for all APIs you wanted to use by enabling them on your project and thus would be connected to your API key.



Creating the calculations was only a matter of following the code already provided by Google Map tutorials and to adjust it to your own needs. For my implementation it would take all the destinations that I had selected and place their values into an array which would be used as part of the waypoint attribute of the direction Service function. If it was successful it would carry out certain code or would display an error alert.

function calculateAndDisplayRoute(directionsService, directionsDisplay) {

var waypts = [];

var checkboxArray = document.getElementsByClassName('checkbox');

for (var i = 0; i < checkboxArray.length; i++) {

if (checkboxArray[i].checked) {

waypts.push({

location: checkboxArray[i].value,

stopover: true

});

}

}

directionsService.route({

origin: GlobalVars.coords,

destination: GlobalVars.coords,

waypoints: waypts,

optimizeWaypoints: true,

travelMode: 'DRIVING'

}, function (response, status) {

if (status === 'OK') {

. . . . .

} else {window.alert('Directions request failed due to ' + status);

}

});

}

A difficulty in implementing the use of APIs was not knowing some of the dependencies some of them required. When choosing a particular API to carry out a function you wanted it would list the API that is needed. For example calculating distance and time between selected points on a map mainly uses the Distance Matrix API. (Google, 2017) Once all code was created to calculate this all out it would continue to fail on me during the JavaScript execution which isn’t the best for giving back error feedback nor was the Google API Dashboard at the time. I was able to find out that it needed the Directions API (which in retrospect appears more obvious that was needed) to help in the calculations. (Google, 2017) Once I had enabled it on the dashboard, it immediately refreshed the error graph for the API and showed all of my previous attempts that had failed.

This had fixed the problems that were occurring. During development I always kept this as part of my error checks to carry out in case it reoccurred which it did one other time and saved me a lot of time in trying to diagnose the problem.

## JavaScript/jQuery

For most of the more than basic functions that I wanted to carry out in JavaScript/jQuery I would usually have to look into getting it to work properly as it never would the first time. An example of this was selecting or deselecting checkboxes on locations in relation to what curated packages were being chosen. It required the heavy use of the Document Object Model (DOM) of the webpage in getting parents and sibling of certain elements with unique identifier values. (Stackoverflow, 2017)

$(".PlannedTrips").click(function (e) {

$.post("@Url.Action("GetPlannedTrips", "Map")", { id: $(this).find('#TripID').val() }, function (data) {

//alert(data);

$(".checkbox").each(function () {this.checked = false; });

jQuery.each(data, function (i, val) {

$("#LocationID[value='" + val + "']").siblings()[0].checked = true;

});

});

});

## Toastr.js

I was looking into finding something that would give some feedback to the user that their saving action of itineraries would show. The best one I found from recommended lists was Toastr.js, with a very simple implementation and design. They have their own demo that allows you to immediately test and customise the kind of messages you want to appear and then just let you copy and paste the code of options to put in your own code. (Toastr, 2017)

toastr.options = {

"closeButton": false,

"debug": false,

"newestOnTop": false,

"progressBar": false,

"positionClass": "toast-top-right",

"preventDuplicates": false,

"onclick": null,

"showDuration": "300",

"hideDuration": "1000",

"timeOut": "5000",

"extendedTimeOut": "1000",

"showEasing": "swing",

"hideEasing": "linear",

"showMethod": "fadeIn",

"hideMethod": "fadeOut"

}

if (data == 1) {

toastr["success"]("Your Saved Itinerary Has Been Updated")

}

else if (data == 2) {

toastr["success"]("You Have Saved An Itinerary")

}

else{

toastr["error"]("Error has occured!")

}

# Design and User Interface

## User Interface

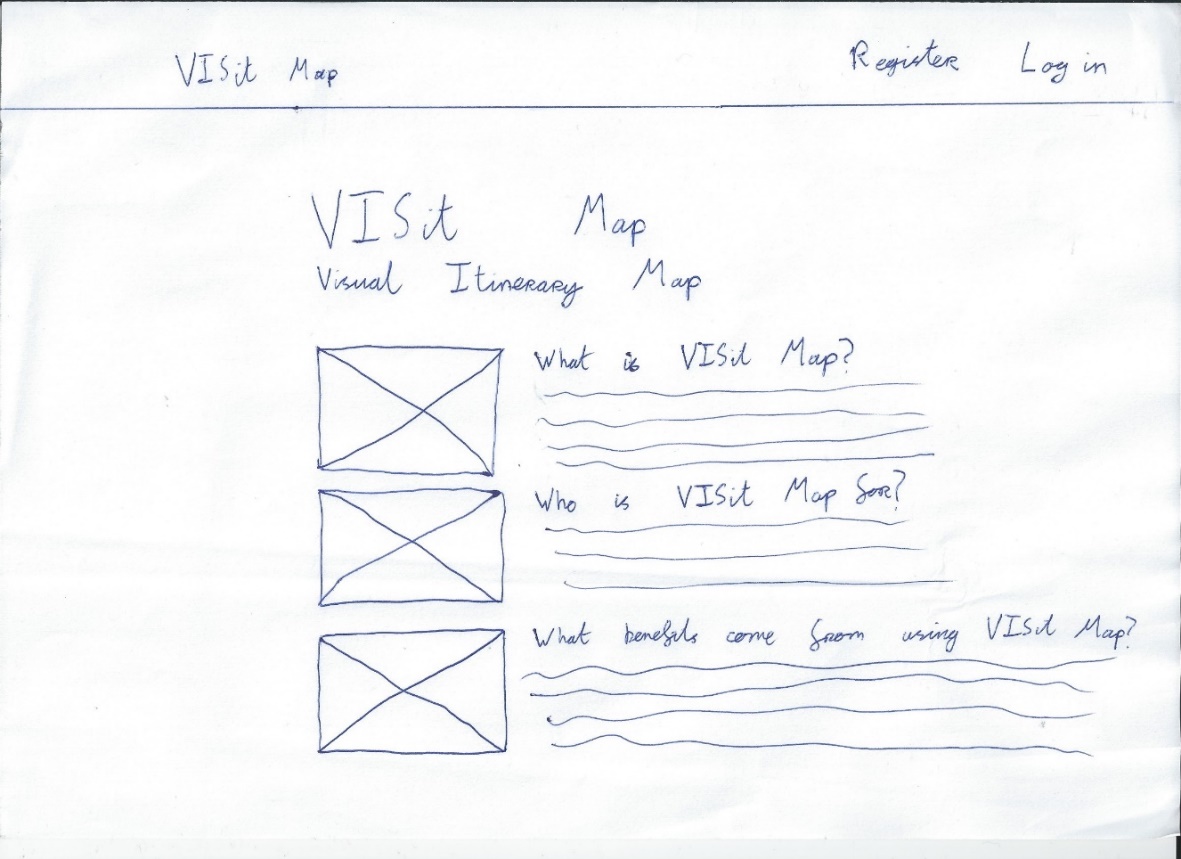
The main goal when creating the user interface that I always followed during development was making it as accessible to as many different sizes of devices as possible. This insures that no one is limited to gaining access to the application.

### Prototyping

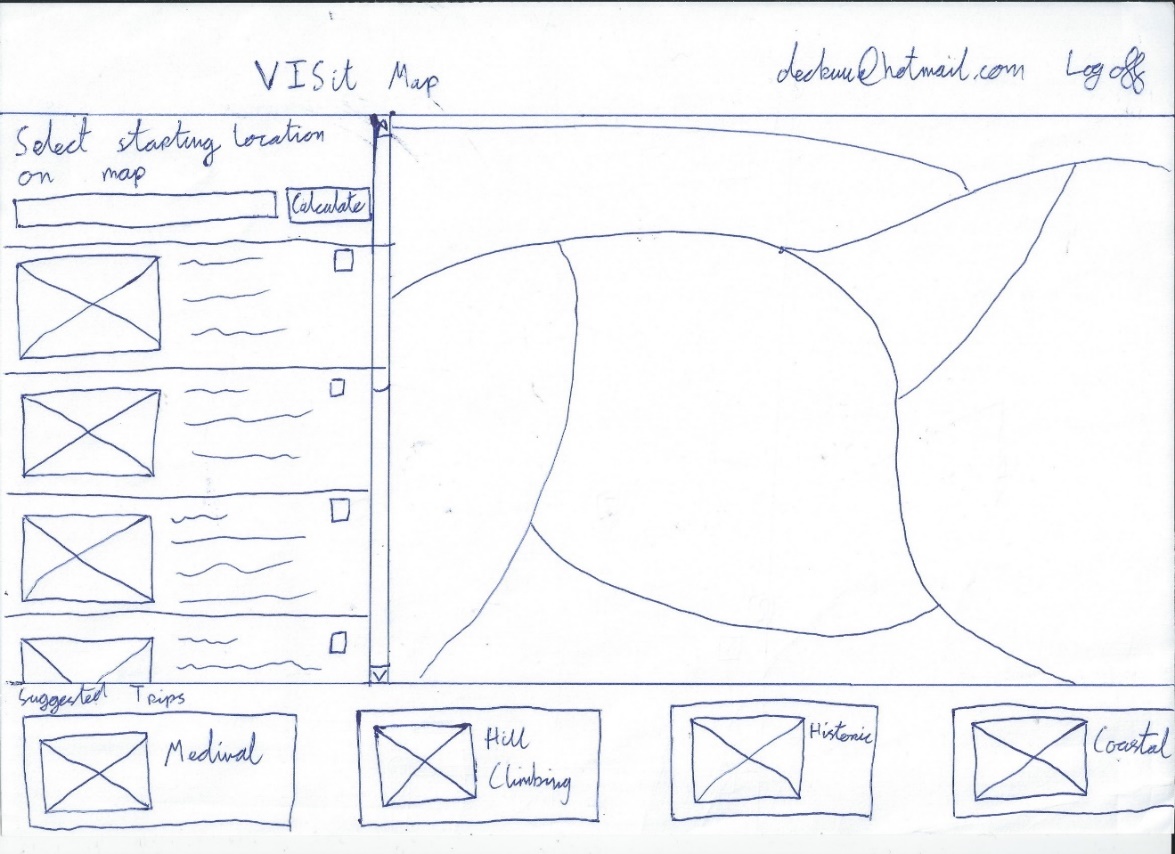
In creating the user interface I decided to create and conduct prototype tests of both low and high fidelity. The aim of this was to get visual and design feedback from a range of test subjects to help in creation of the user interface.

#### Prototype

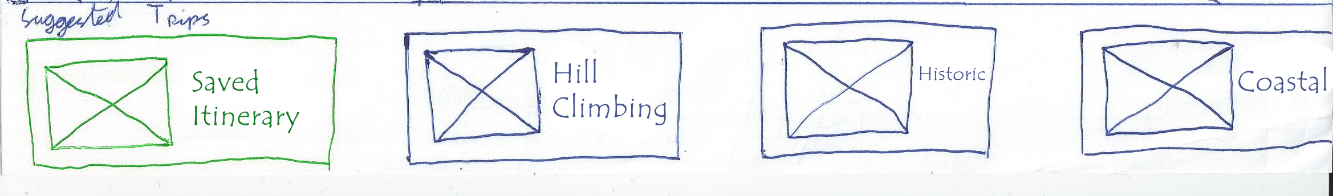
The first page is landing page that invites new potential users into using the service by explaining its features in a clear concise manner that even those not tech savvy could understand. This will include pictures showing off the features in a simplified form and adds an air of professionalism. This page will allow users to register to the site if they wish or login if already a registered user. Owners must log in but users can use the itinerary calculation planning as a guest but won’t be able to save their route or other functions that a registered user can.



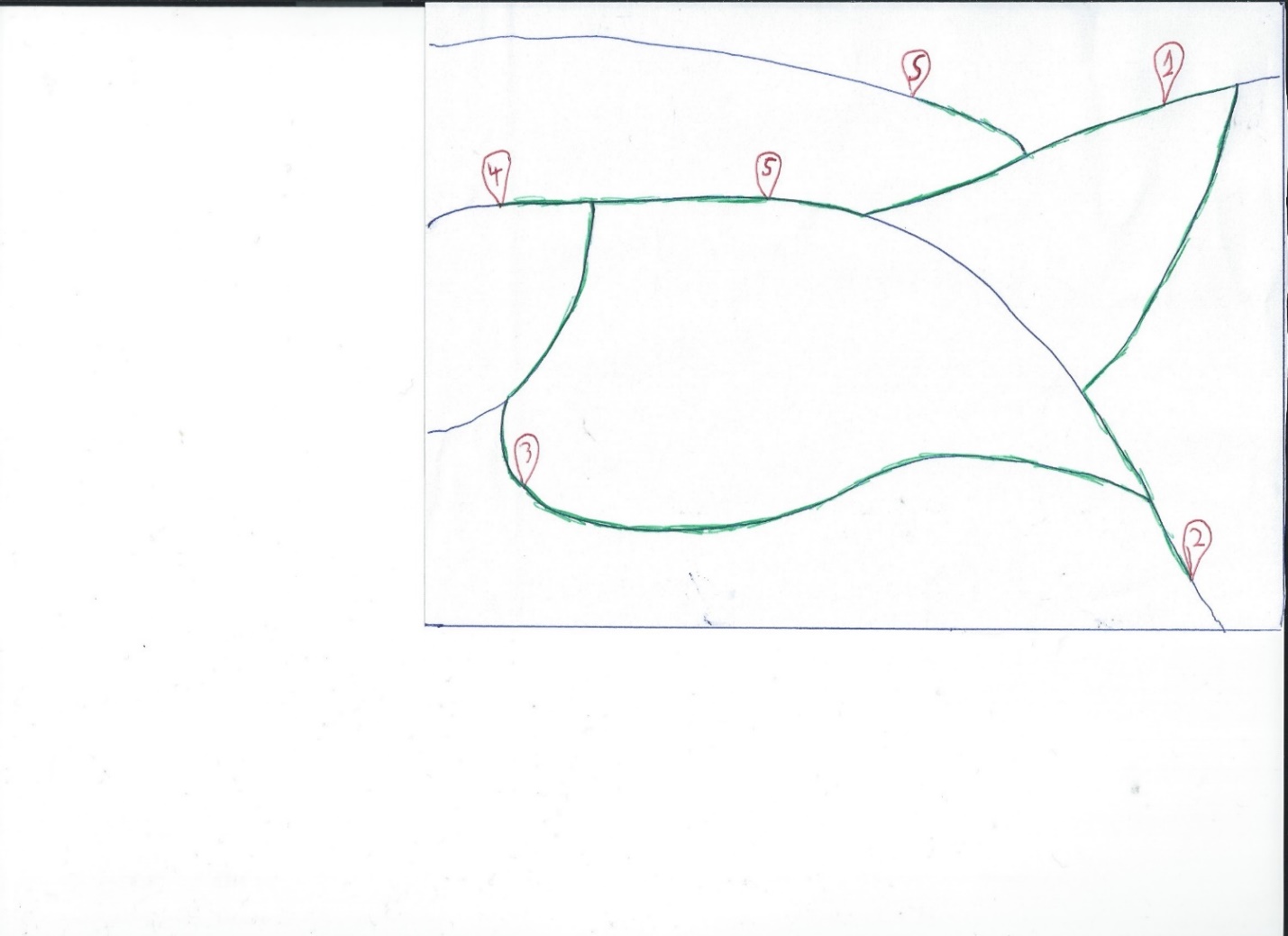
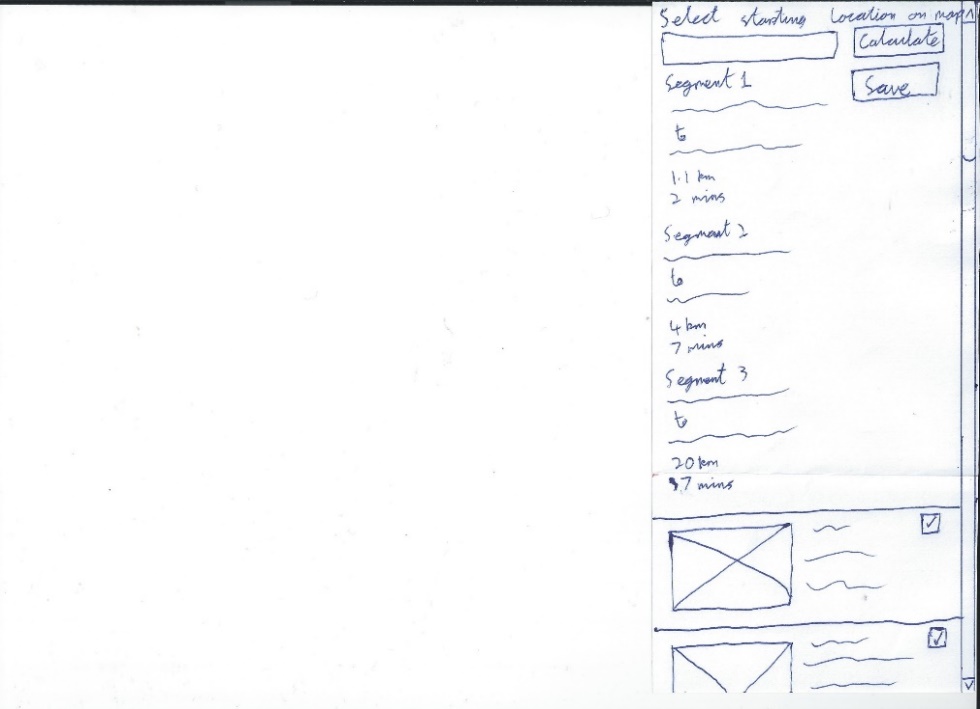
The map page encompasses most of the functionality all in one screen so a clean, non-cluttered user interface is very important. Each function of the user interface is separated into its own section on the page. The side list bar holds the search box for users to select their start/end location that autocompletes when a user is typing. This then reorganises the list of possible destinations below it to those that are the closest to it. Each destination has an image, name, address and distance from the chosen starting point. A user can tick each one they want to visit and once they click on the submit button it will calculate out the route of the journey. When this is done the list of destinations will be pushed down with a save option and an itinerary segment summary appearing just below the submit button. This shows in detail each segment on the journey that includes the addresses of the start and end of each segment, the distance between each and how long the journey would take depending on the users chosen method of transport. The submit button allows the user to save their chosen itinerary for future use. It will be added to the horizontal curated package bar and always appear as the first entry if they are logged in.



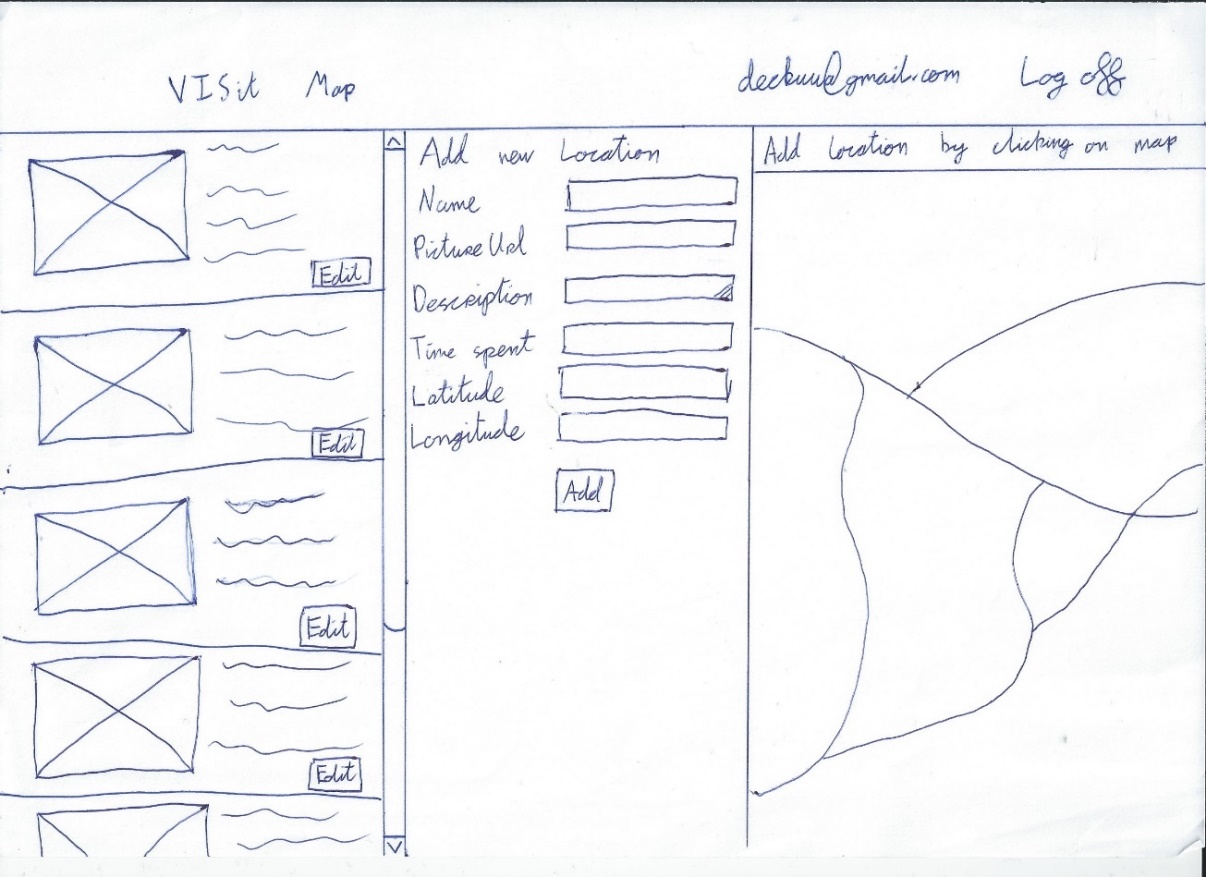
Their saved itinerary will be added to the horizontal curated package bar and always appear as the first entry with a slight modification to make it stand out differently from the rest of the packages if they are logged in. It is a horizontal scrolling list of pre-defined packages. Once a user selects one of these packages it will automatically tick all the destinations in the sidebar and update the map with the itinerary. This can then still be customized by adding or removing destinations by the user.



The map section will load up with the person’s current location initially so that users that are in their intended place to go travelling can see immediately their surrounding area. When a user uses the search box to choose their start/end location the map will change to that location and place a marker upon it. When the user has selected their intended destinations and submitted it, the map will then change to show the entire route with each destination having its own marker.



The owner page will look like the map page to maintain layout consistency with a few differences. The horizontal list at the bottom of the page is removed with its area used up by the other functions of the page. The maps prominent size is reduced to allow for a form to be added between it and the sidebar list. This form is used to enter in details of a new location being added to the database. The sidebar list of destinations will be populated with the owner’s uploaded locations with the same information that appears on the map page. In the form, all information is entered by the user accept for the location co-ordinates. This is entered when an owner clicks on a location on the map for their intended new addition which reads in the co-ordinates and places these in un-editable text boxes. Once all fields are entered and the owner clicks submit this will update the sidebar list to include the new destination. Each entry on the list will have an edit button that will populate the form with its details and allow the user to edit that locations information and save it.



#### Method, Tasks and Test Procedure

I conducted my first tests of the paper prototype with random people during one of my society group evenings. I would inform them as to the purpose of the site and their goals to reach in these 3 tasks:

* Task 1: User selects locations or pre-defined trips and it is displayed on map.
* Task 2: Users save their selected itinerary and load it up again at another time.
* Task 3: Business owners adding/editing a location in the database to be used in the application.

I would present them with the landing page and just explained the content that was on view but none of the functionality so that they could discover it from themselves and let me observe how they would interact with as little engagement from me as possible. Responsiveness of their actions was mimicked by moving different sheets of paper around to show the changes that would be made during after their interacted choice.

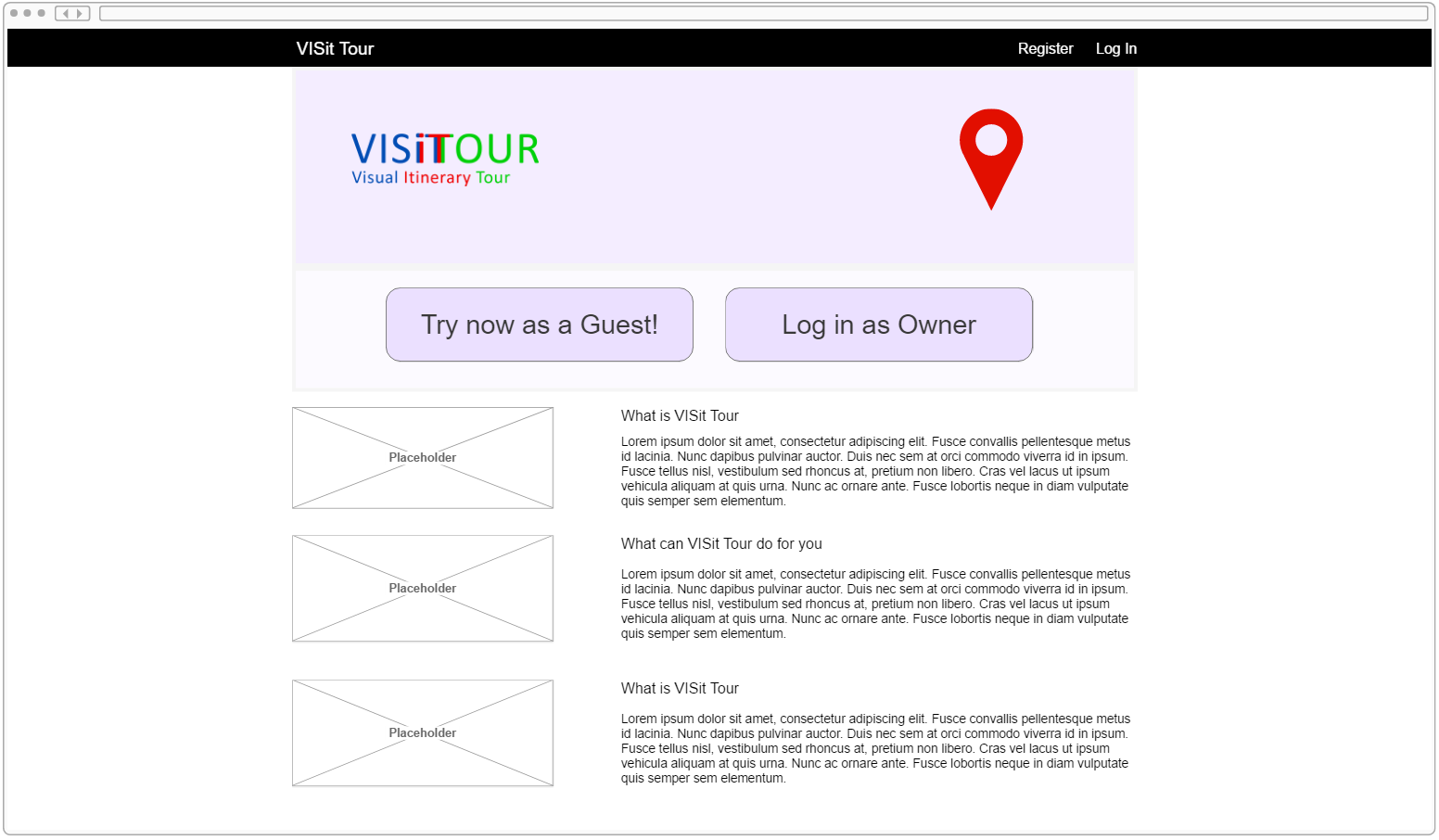
After each task, I would ask them to give a rating from (Low) 1-10 (High) of how they felt on the completeness of the task they felt they had achieved.

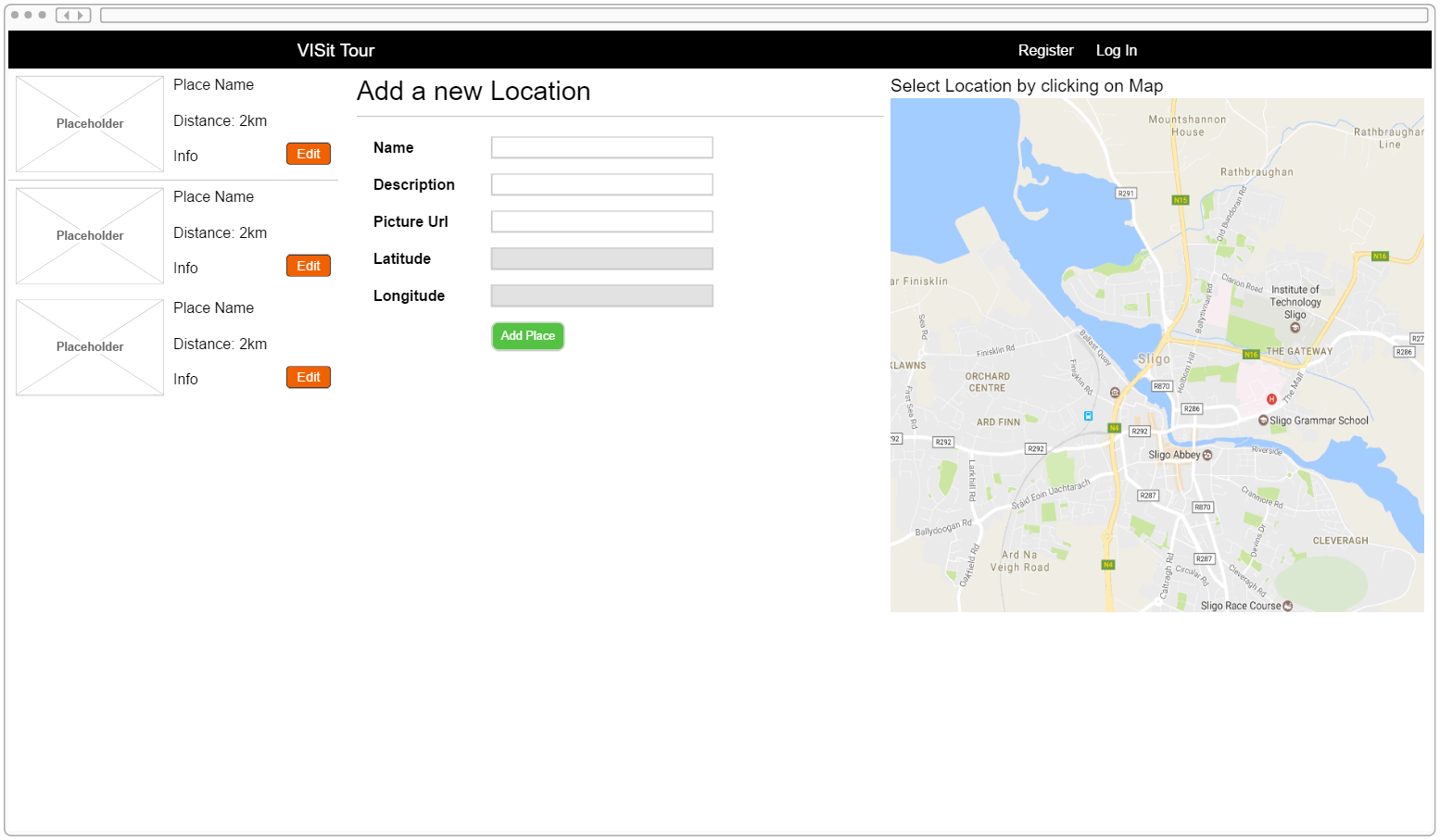
|  |  |  |  |
| --- | --- | --- | --- |
|  | Key Task #1 | Key Task #2 | Key Task #3 |
| Subject #1 | 5 | 8 | 8 |
| Subject #2 | 4 | 7 | 7 |
| Subject #3 | 3 | 9 | 5 |
| Subject #4 | 4 | 8 | 6 |
| Subject #5 | 5 | 8 | 7 |
| Subject #6 | 7 | 7 | 7 |
| Subject #7 | 4 | 7 | 5 |
| Subject #8 | 3 | 8 | 4 |
| Subject #9 | 7 | 9 | 8 |
| Subject #10 | 4 | 9 | 6 |
| Average Score | 4.6 | 8.0 | 6.3 |

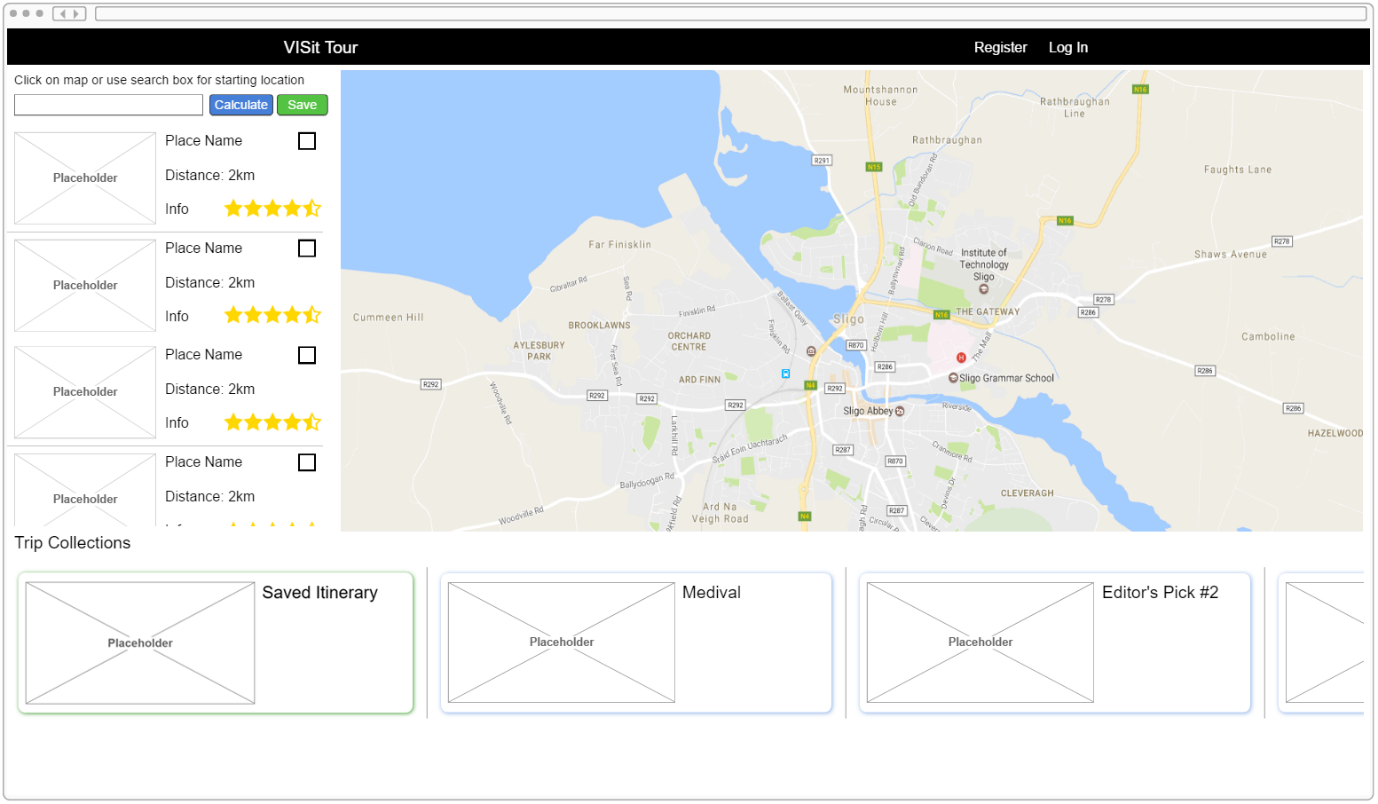
One problem I noticed people were having was there was no call to action in getting people to use the main functionality of the application. The functionality was hidden behind the register/login which was just in one corner. This is a problem with regards to the 3-click rule. It is an unofficial rule with regards to web user interface design where no functionality or information should be more than 3 clicks away from the homepage. This combined with forcing people to sign up which can lose people straight away needed to be addressed. I decided to allow users to access the itinerary planner directly from the main page as a guest and not force them to register/login. Any of the account functionality like saving itineraries I disabled and if guests wanted to use them they would be directed to the register page.

Another problem was an oversight on my part of leaving out buttons that would be needed to carry out some functions like for example allowing an owner to edit their locations which I never supplied.

These changes that were addressed were brought over to when I created the High-fidelity prototype. It was created and tested on a computer screen using Proto.io prototyping software that expertly copies the environment of the device that the project is intended for. In this case, it is for a web browser. It wasn’t possible to implement in the prototype stage getting correct aspect ratios and a responsive layout to cater for the various sizes and resolutions of screens but in the during development this is handled with the bootstrap CSS framework to cover from HDTVs down to small tablets.







With the new High-fidelity prototypes I went and tested it with a mixture of new and previously used test subjects getting them to carry out the same goals as last time and have them write down their rating on the feeling of completeness for each task.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Key Task #1 | Key Task #2 | Key Task #3 |
| Subject #1 | 6 | 9 | 7 |
| Subject #2 | 6 | 7 | 8 |
| Subject #3 | 4 | 9 | 7 |
| Subject #4 | 5 | 9 | 8 |
| Subject #5 | 7 | 8 | 7 |
| Subject #6 | 7 | 9 | 8 |
| Subject #7 | 8 | 9 | 6 |
| Subject #8 | 5 | 8 | 6 |
| Subject #9 | 6 | 8 | 6 |
| Average Score | 6.0 | 8.5 | 7.0 |

#### Results

I found that the functionality of the web application was received well by most of the test subjects. All of them generally found the navigation between pages to be fine. Most of the errors came in the layout of material on display and how it was displayed. The majority had the problem with the functionality that was hidden behind the register/login options. There were some occurrences of spotting small mistakes like missing buttons due to some of the test subjects being students within IT Sligo and having an understanding of computers, which a regular person who mightn’t be up-to-date with computers wouldn’t have spotted immediately. The point brought up the most though was the not knowing where to start and what was the order of carrying out the function of creating the itinerary with all sections of the page being visible and enabled all at the same time when page is loaded first.

#### Analysis

As a logical person, I lack the creativity to express and convey the delivery of content in a good and meaningful way. This I knew would play into effect when creating the prototypes and was proven correctly for the missing of simple items and unordered layout that I could understand but other more casual computer users may not. Conducting these tests proved to be a necessary part for my project as the design aspect of projects are my weakest area and will ensure that I apply more prototyping in future projects. The proof came from the difference in the scores collected between the two tests. Even though I had one less test subject the second time around the average percentage for all 3 tasks increased. Task 1 had the best improvement with 30.43% increase, Task 2 had a 6.25% increase and Task 3 had an increase of 11.11%.

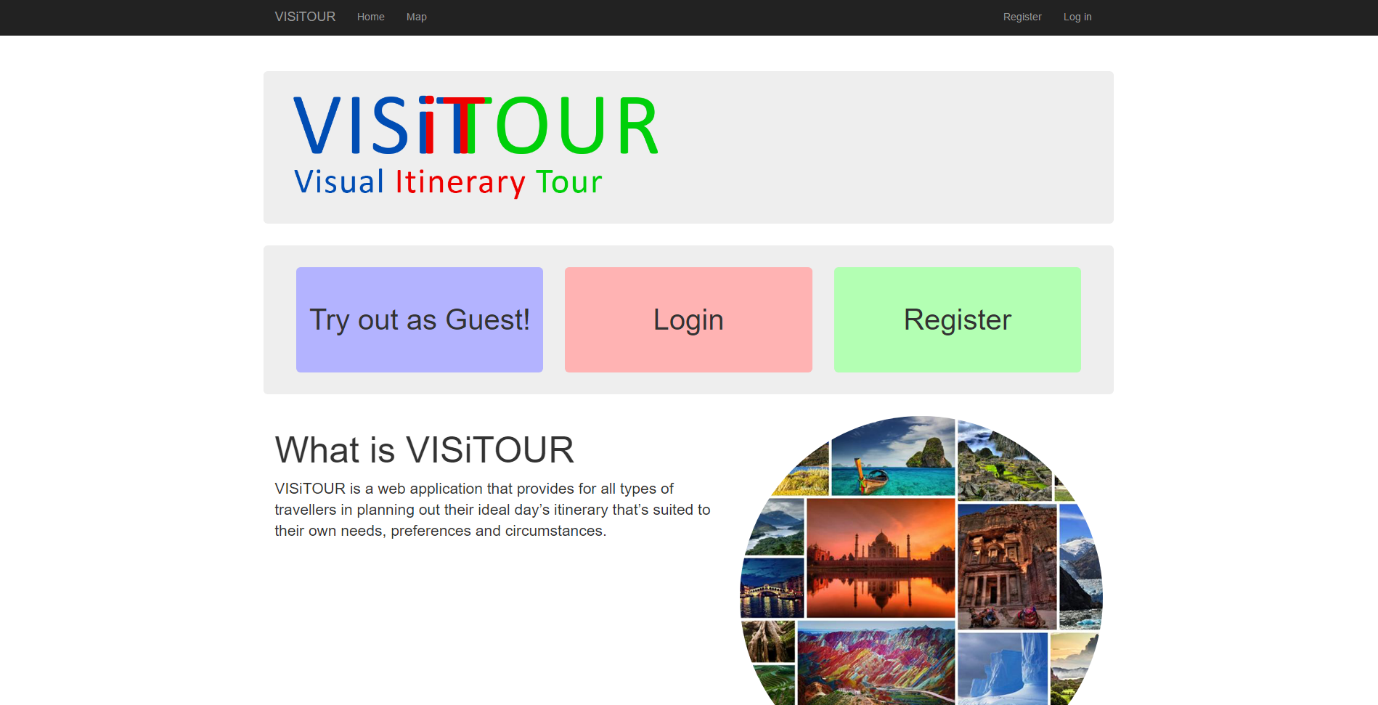
I learned while observing people just how many people can take things up in diverse ways or how they carry out the procedures to get to the goals. This shows how you must cater for the multitude of ways that all lead to the same goal due to every person being different.

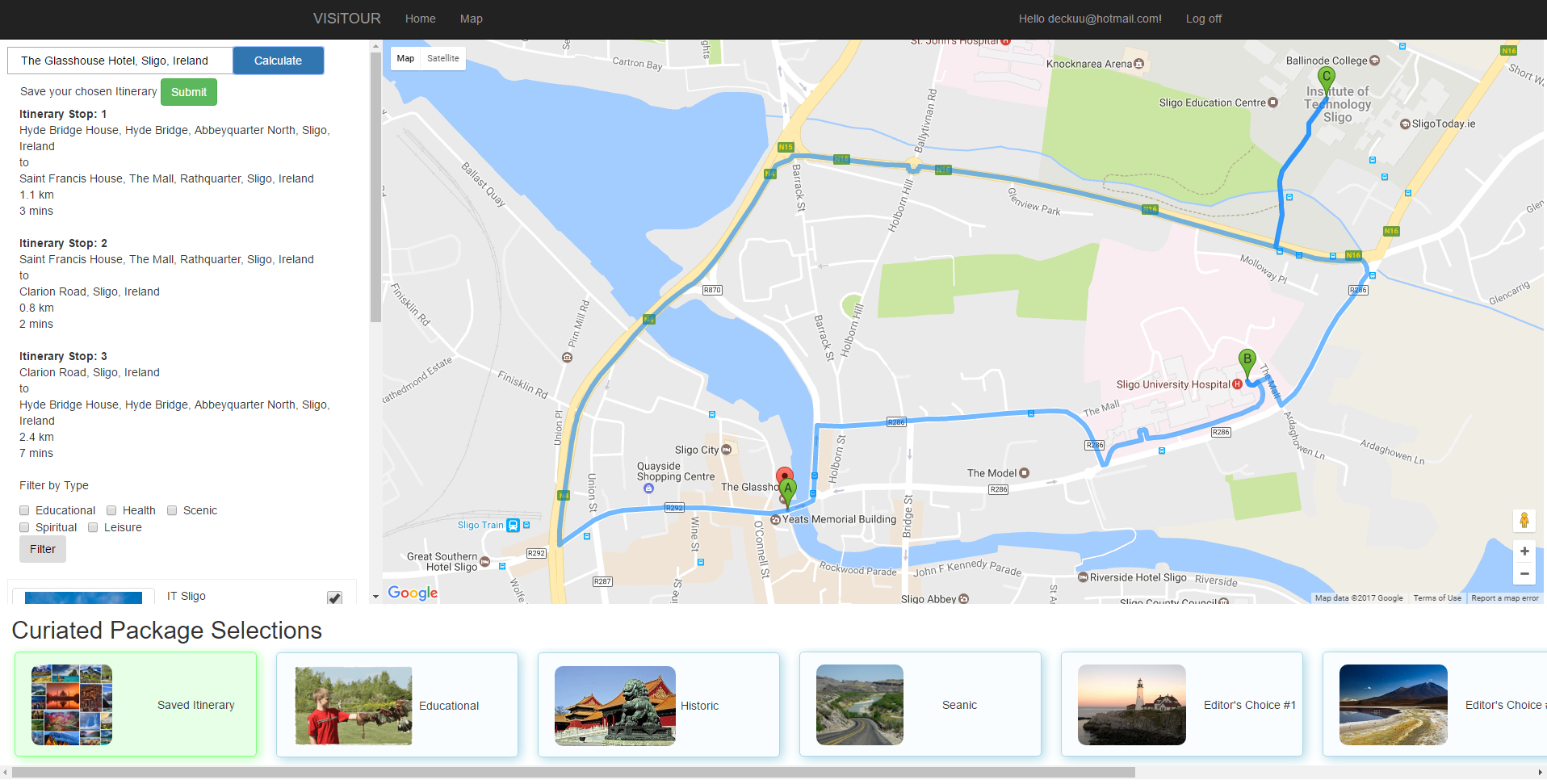
Using Low-fidelity prototyping allowed me to create a user interface that was easy and relatively took not much time at all to create. It also allowed for changes to be made easily and functions like scrolling with just folding up paper that otherwise would have required some coding that could have taken away time for an outcome that may not necessarily work and require more time again to change. This allowed me to focus my user interface to fit in with the platform I had designated for the project and by keeping it relatively simple allowed me to save time with changes in this stage than have to make changes that would take up much more time later in development.

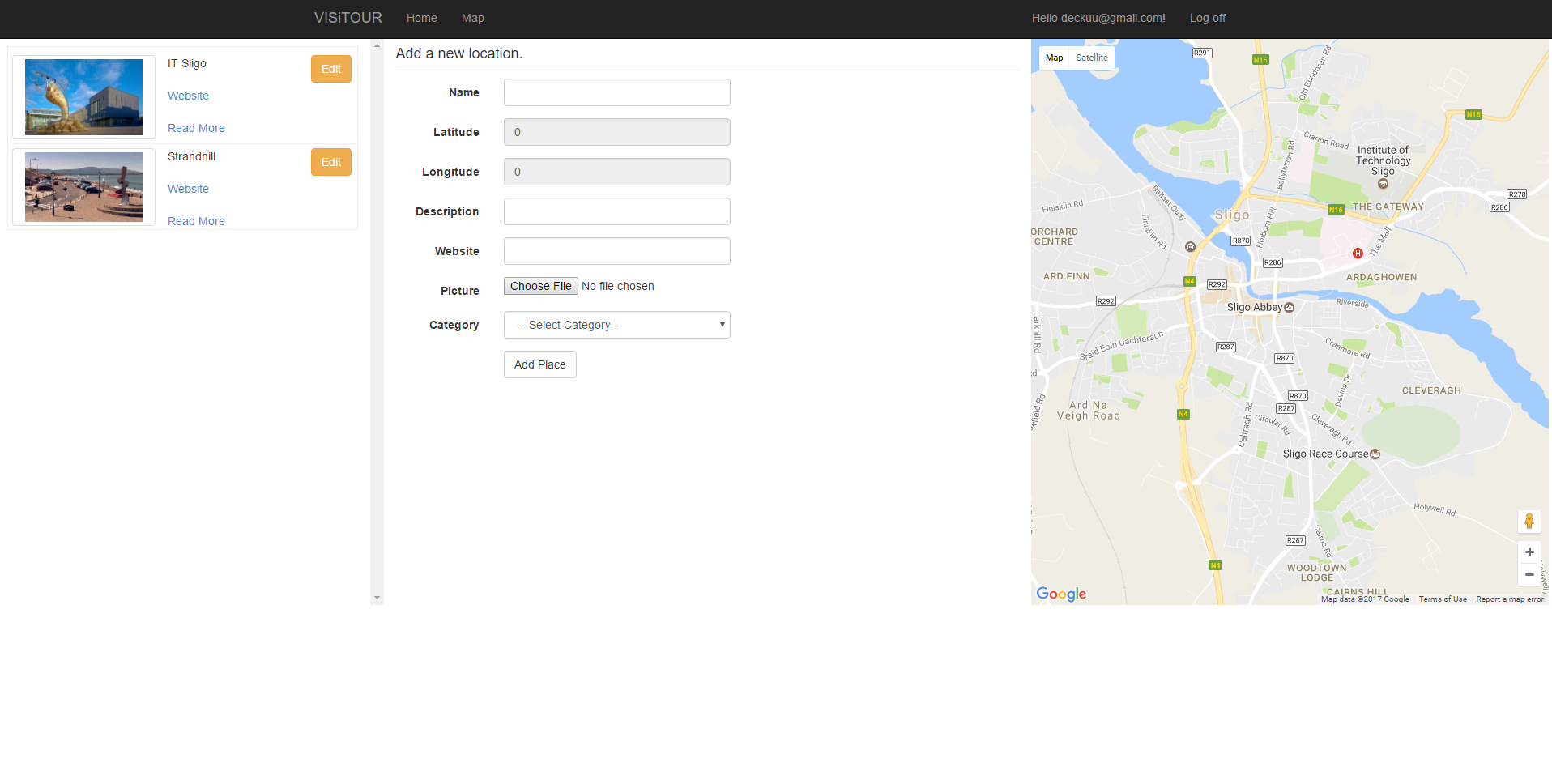
The High-fidelity prototyping allowed for easy converting of turning the Low-fidelity into High-fidelity while keeping the user interface and general functionality intact. This was especially helpful with having used Proto.io previously. Before this project I had never personally created a low-fidelity prototype for anything but after doing so Proto.io was helpful in easily implementing the user interface of what you have already created. With having the same advantages of Low-fidelity it allowed the user interface and function of the project resemble closer to the final product.

### Implemented User Interface

Through review of test subjects comments and views on both prototypes and with some minor adjustments this is the User Interface that was implemented.







Colour is a major part of the experience of using the Web Application. Itis a quick and effective way to inform a user of the functions they can carry out and this has been used mostly throughout the application. The colours convey meaning that are implicit e.g. green is good or orange is warning. The colours also help in directing user’s attention to certain area of the application.

## Overall Architecture

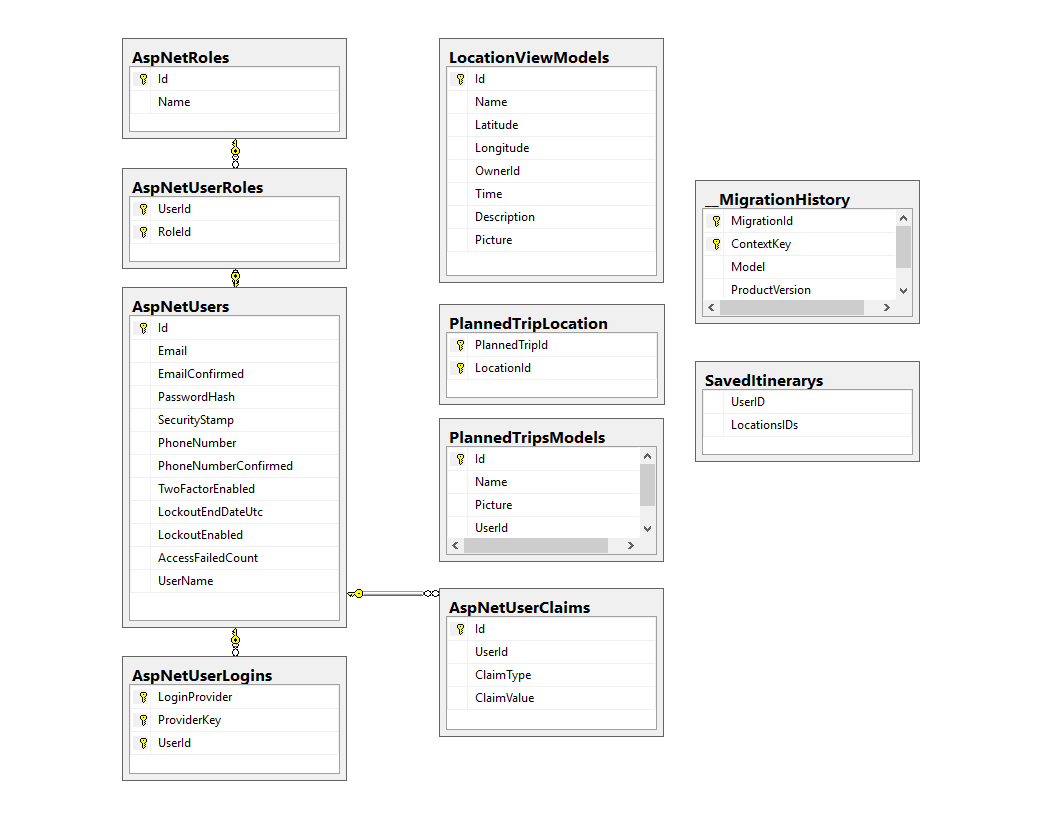
SQL Server Database

Hosted on Microsoft Azure

Google Map APIs

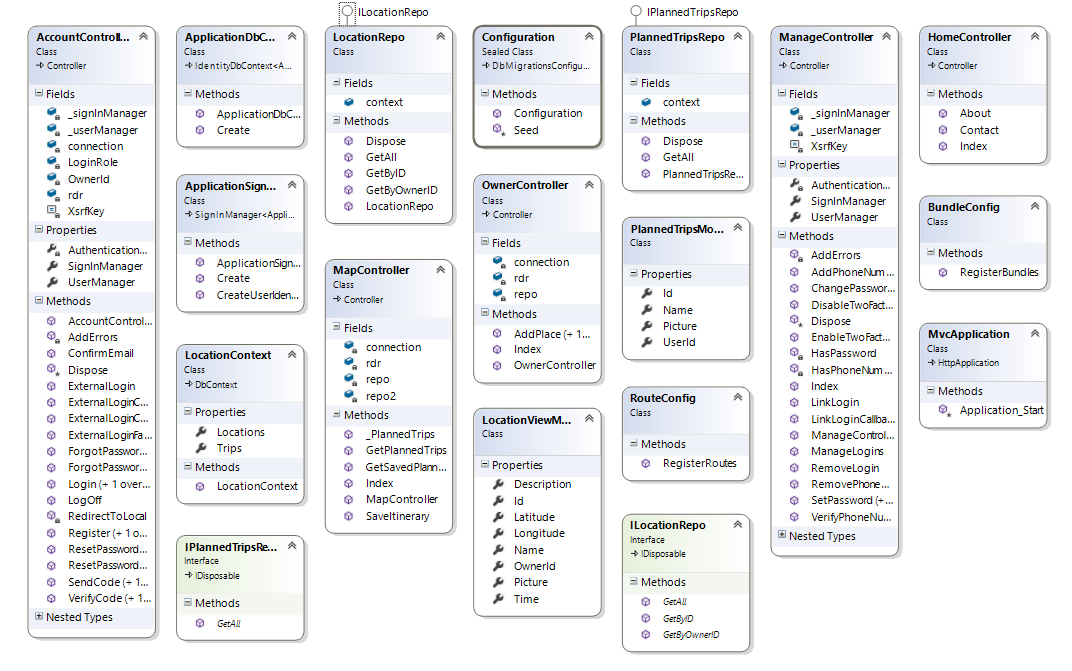
MVC Web Application

## Database Design if applicable



The Asp.Net Tables were auto created by the ASP.NET identity system using entity framework code first. These were all created initially with creating the MVC Web Application in Visual Studio.

## Class Diagrams



Some of these classes were created as part of the ASP.NET identity system that is implemented on the creation of the MVC Web Application.

# Scrum

During development I, as best I could implemented the SCRUM methodologies in getting my work done.

## Sprints

I set myself either one or two weeks to each sprint depending on what I intended to get done or my workload. This was just the right amount of time that would be enough to get most of the tasks done and wouldn’t give myself enough time to be wasting time. I carried over this methodology from last year’s group project that kept us working at our best and concisely on our assigned tasks. For myself personally I saw it as a really good practice of development. It certainly helped me be realistic with the tasks I wanted to complete and allow me to precisely gauge how much work I could get done in a certain amount of time.

## Methodologies

While I personally believe the SCRUM methodology is very helpful it wasn’t possible for me to fully implement every aspect of it. Due to not being able to work on the project full time and having many other assignments to complete, it wasn’t possible to directly assign work to each day and assess all of what I had done and what I was going to work on next, thus each sprint was irregular in length being at least a week or longer at times. An example of a technique used during the development I used was the MoSCoW gauging scale of deciding on all the features wanted for the project and if they were a Must, Should, Could or Won’t.

# Time Management

Last year I had a Software Project Management module which helped in orderly planning out the development and timing of the project. At the start I had no set timing for completing actions as I was only just conducting research. This time was used mainly for research into Mobile development which led me to Xamarin. As of course this led to nothing I moved onto technologies I knew so I could just go and get started from there and make up for the lost time.

This began the working of laying out the sprints and the work needed for each and carrying on from there. Sometimes I may not have fully completed an intended function or task by the end of a sprint which depending on its necessity was either finished at start of next sprint or was tackled at the end of another sprint I may have gotten finished early.

Unfortunately I was unable to actually set exact times to dedicate to working on the project due to personal life and mental health issue. Sometimes I had been travelling in cars on long journeys continuing to work on the project.

I met with my supervisor Colin once a week where we would discuss what was done and what was intended to get done soon. It also helped in asking for advice or solutions to any problems I may have been having at the time.

I used Trello to keep track of my progress on what was done, needed doing and what I possibly will plan to do. It effectively functions as an electronic version of the general characteristics of SCRUM meetings. It mimics sticky notes for each task to carry out, under headings of what section of the project it belongs to and under further notes of who is tasked with completing it.

# Problems Encountered

Setting up of Xamarin at the beginning was a major problem due to the wasting initial project time. For whatever reason installations and tutorials just weren’t enough to solve the many problems that occurred and just led to frustration and apathy towards the project as a whole throughout development which I found difficult to deal with.\*

During development I lost some of the functionality which included editing of previously made locations in the database. This had been previously working but for whatever reason broke over time and once I had realised this I carried out rollbacks and error checking but this wasn’t enough to fix the problem that had caused it. With time running out for completion of the project I was unable to solve this issue.

Again during development I lost functionality that had been previously working. This time it was the filtering system of locations and again not solution to solve the problem were able to help before the deadline.

# Results

The finished Web Application at the end of this projects contains many of the features that were intended to be implemented. However, due to the time constraints, other assignments that at least required some attention but other times requiring time for long periods and difficulties I personally had with certain platforms and coding languages I was unable to implement everything I set out hoping to achieve.

The obvious one I had hoped to implement was Xamarin Android. This I felt would have be a worthwhile accomplishment with having put in a considerable amount of time researching and wanting to try out something different that had been thought on the course. Now being at the end of this project I know it would have made for a much better application\*

\*During the course of this year, my second semester module Mobile Development required us to make a mobile application of our choice and develop it however we wanted. I choose to try my hand at Xamarin Android to see if it would be possible this time to set out what I had hoped to achieve at the start of this year. After getting to the same installation point I had gotten to where downloaded code example would work on my mobile I delved deeper into anything I could find that would give me the exact information I would need to develop on it. This time it was a complete success with implementing everything I had wanted to achieve. Using Xamarin also helped me to understand Java which I had never looked into before. It allowed me to look up coding examples if I was stuck that were in Java and convert them for use in c#. Not only was I able to create a substantial mobile application but was also informed by a lecturer that it was of the standard of a 4th year final project worth a high grade.

The design for this web application would have been similar to the Google Trips mobile application that is pictured in the research section.

Having learned what I know now, I would completely redesign this Web application as a Mobile Application. Doing this I would have found it considerably easier to develop for and possibly not only completed the exclusions that I had no choice but to leave but I also would have had more time to consider adding further functionality.

The filtering of location categories and edit functions that were working were a disappointment to not be able to fix or rollback to a point that at least showed that at one point it was working.

I am however glad to have achieved what I was able to get implemented for the gruelling hours, work and perseverance I placed into this project even with all the frustration moments I encountered.

That users of all kinds no matter their level of tourism interest or knowledge of computer interaction will be able to use the system with ease and get the most out of what they seek. The ultimate satisfactory level of having users use this product is if they are provided with the best user experience of not only the web application but of their actual itinerary tour as well.

I am also glad to have accomplished creating an application that is capable of being a forever expansive application. As I had hoped I wanted it to be able to start out small and in time be able to grow however much it needed. Currently it is only focussed to around the areas around Sligo but the application can further grow from beyond this project and gain traction to cover more and more areas to which it could become worldwide.

# References

Chindam, P., 2010. *Inserting & Retrieving Images from SQL Server Database without using Stored Procedures.* [Online]   
Available at: http://www.c-sharpcorner.com/uploadfile/e628d9/inserting-retrieving-images-from-sql-server-database-without-using-stored-procedures/

Collier, M. S. & Shahan, R. E., 2017. *Free ebook: Microsoft Azure Essentials: Fundamentals of Azure, Second Edition.* [Online]   
Available at: https://blogs.msdn.microsoft.com/microsoft\_press/2016/09/01/free-ebook-microsoft-azure-essentials-fundamentals-of-azure-second-edition/

Google, 2017. *Directions API.* [Online]   
Available at: https://developers.google.com/maps/documentation/directions/

Google, 2017. *Distance Matrix API.* [Online]   
Available at: https://developers.google.com/maps/documentation/distance-matrix/

Google, 2017. *Embed Google Maps features and functionality in your sites..* [Online]   
Available at: https://developers.google.com/maps/web/

Google, 2017. *Get API Key.* [Online]   
Available at: https://developers.google.com/maps/documentation/javascript/get-api-key

Jani, D., 2017. *Now Make Bus, Train Reservations On Google Trips.* [Online]   
Available at: http://www.vishwagujarat.com/technology/now-make-bus-train-reservations-google-trips/

Microsoft, 2017. *Azure SQL Database server-level and database-level firewall rules.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-firewall-configure

Microsoft, 2017. *Storing Images and Photos in SQL Server.* [Online]   
Available at: https://social.technet.microsoft.com/wiki/contents/articles/6519.storing-images-and-photos-in-sql-server.aspx

Microsoft, 2017. *Visual Studio Dev Essentials.* [Online]   
Available at: https://www.visualstudio.com/dev-essentials/

Stackoverflow, 2017. *jQuery get parent sibling for this element only.* [Online]   
Available at: http://stackoverflow.com/questions/8591887/jquery-get-parent-sibling-for-this-element-only

stackoverflow, 2017. *Updating partial view with Jquery in ASP.NET MVC C#.* [Online]   
Available at: http://stackoverflow.com/questions/11122268/updating-partial-view-with-jquery-in-asp-net-mvc-c-sharp

Toastr, 2017. *Demo.* [Online]   
Available at: http://codeseven.github.io/toastr/demo.html

w3schools, 2017. *SQL General Data Types.* [Online]   
Available at: https://www.w3schools.com/sql/sql\_datatypes\_general.asp

Xamarin, 2017. *Hello, Andoid.* [Online]   
Available at: https://developer.xamarin.com/guides/android/getting\_started/hello,android/

Xamarin, 2017. *Xamarin.Android guides.* [Online]   
Available at: https://developer.xamarin.com/guides/android/