

# Functional Specification Contents

## ‘Vicinity’ -App for Finding Nearby Events

Adam McElroy: 16345753

Eimhin Dunne: 16386406

Supervisor: Paul Clarke

<b>0. Table of Contents</b>	<b>Pg.</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Overview	
1.2 Business Context	
1.3 Glossary	2
<b>1. General Description</b>	
2.1 Product / System Function	
2.2 User Characteristics and Objectives	
2.3 Operational Scenarios	3
2.4 Constraints	4
<b>2. Functional Requirements</b>	<b>5</b>
3.1 Getting data for event database	
3.2 Getting data from user database	
3.3 Getting data from event database	6
3.4 Create an account	
3.5 Use without an account	
3.6 Event recommendations	7
3.7 Finding event using filters	
3.8 Finding event or place using search	
3.9 Adding users as friends	8
3.10 In app communication	
3.11 Sharing events	9
3.12 Reviewing events and activities	
<b>3. System Architecture</b>	<b>10</b>
<b>4. High-Level Design</b>	<b>11</b>
5.1 Context Diagram	
5.2 Data Flow Diagram	
<b>5. Preliminary Schedule</b>	<b>12</b>
6.1 Schedule	
6.2 Gantt Chart	
<b>6. Appendices</b>	<b>13</b>

# **1. Introduction**

## **1.1 Overview**

The aim of Vicinity is to gather information from a wide range of sources on the internet relating to public events, places and activities within range of the user and recommend some of these activities for the user to do. The application is intended to make all the data retrieved from various sources easily read by a human in a list of events. This will make it easy for people to find events to do and organise outings with friends. Recommendations will be made to the user of new events that the system thinks they will like. A user will enter some of their interests upon sign up which will aid the system in identifying suitable events for them. It will also take into account past events they have attended as they use the system to learn about their behaviour. If multiple users are planning on doing something the system will take all of their interests into account and produce recommended events for them. Filters can be applied such as your budget for the day or is it accessible by public transport etc. People can share this data with each other by sharing it to group chats with other friends and organise days away. Data will be taken from APIs as well as OpenStreetMaps (OSM) and web scraping and put into a database. This data will be run through algorithms to classify it. The classification is to ensure a user can filter events if they want. We also want to add in a social feature where users can add each other as friends. Once a user has friends they can start conversations with the messaging feature and also create group chats with multiple users. Events, places and activities will all have a share option where you can share them to these chats for all of your friends to see. This will make organising days away with your friends easy as everyone will have access to the details in one area. Users can also leave comments under events for others to see. This will be helpful for people with any questions about an activity before they go. We plan on deploying this as an android application and also a web app.

## **1.2 Business Context**

The aim of the application is to inform a wide range of people about activities that are happening around them that they may want to go to but wouldn't have heard about anywhere else. They will hear about the event enough in advance that they will be able to organise everything they need. It will be usable by all ages and interests. People don't have to stick to looking at recommended events either. If you are travelling in a place you are not familiar with you can use the search feature to check is there anything that interests you nearby. For example, if you want to know if there is a park nearby you can simply search "park" and it will show you any nearby parks and information about them. We want to make it so that users won't be spammed with recommendations of events that don't interest them and instead learn about the user by looking at their interests and suggesting activities based on recent ones they may have done. We think it's a good way of promoting events in an area and would like to add a social aspect to it to make it even more effective in promoting. We plan on adding a messaging feature to it and the ability to share events into group chats to enable organising group outings.

### 1.3 Glossary

**OSM** - OpenStreetMap. Map based API for lots of data about public places.

**Firebase** - Cloud based storage by Google.

**Android Studio** - IDE for programming applications for Android Devices.

**API** - Application programming interface. Will be used for retrieving data.

**Web Scraping** - A method of retrieving content from a web page.

## 2. General Description

### 2.1 Product / System Functions

Below is a list of the main functions we believe are needed for the application:

- Login / Sign Up / logout
- Choose interests from list of categories
- Match activities to specific user interests
- Match activities to multiple user interests
- Categorise activities automatically
- Add users as friends
- Create chat between users
- Create a group chat with users
- Share events to a chat
- Show more details about certain event on click
- Find, classify and store data on the database

### 2.2 User Characteristics and Objectives

The user will not be required to have any technical expertise in order to use the application. It is intended to make the interface simple and clean and provide an intuitive experience for the user. Most of the heavy lifting will be done in the backend of the application to ensure there are no rigorous tasks to carry out for the user. We intend to create an android application as well as a web application to reach abroad user demographic.

## **2.3 Operational Scenarios**

### **Log In / Sign Up**

Upon opening the app for the first time the user will be greeted with a Log In / Sign Up landing page. From here the user can enter their email and password to login or sign up for a free account. To sign up a user will be able to create a Vicinity account by entering an email, username and password or they can select to use an existing google or facebook account to sign in. They can also skip this step if they wish to just browse events on the application without an account.

### **Select Interests**

Once a user first signs in they will have the option to select what they are interested in from a list of predefined topics. They can also skip this step if they do not wish to enter their interests at this time. They will still have the option to modify these settings later in case they wish to add/remove certain topics.

### **Browse Events**

The main screen will be displayed to the user once signed in. The main part of this screen will be dedicated to the users current recommended events. The page will also contain a section where you can select different categories and a search bar to filter events further.

### **Pick Event**

If a user sees an event that they are interested in they can click it to open up more information about it. This will display a broader description of the event and provide information such as location, accessibility, prices etc.

### **Searching**

Users can search events by category as well as by text. They can search for different users as well as events by place, date etc.

### **Add Friends**

The user will have the ability to add friends. This will promote the social aspect of the application. User will be able to share events with friends and also message them.

### **Message Friends**

Users will have the ability to create a chat with friends in order to message them directly through the app. It will also be possible to create group chats with various members to talk in.

### **Create Group Chat**

A user will have the ability to create a group chat where various people can message each other at once. These group chats can also be used to share events into.

### **Share Events**

If a user finds an event that they think might be suitable for a group of people they will have an option to share the event to a group in order to inform them. This will make it easier to plan in advance.

### **Confirm going to event**

A user can select if they are going to an event that has been recommended to them. This data will be used in order to help the system recommend similar events in future.

### **Review**

Users can leave a brief review on an activity in the form of a comment. This will help other users to learn any extra information about the activity. (eg. bring a towel). They can also rate how the activity matched their interests to help our app predict better activities that suit them in the future.

## **2.4 Constraints**

**Internet Connectivity** - The user will need to have an active internet connection in order to use the app as the data is constantly being updated in the background.

**Missing Data** - Some data might be missed for small specific events that a niche number of people might want may not make it onto the database.

### 3. Functional Requirements

#### 3.1 Getting data for event database:

**Description** - This will be one of the most technical parts of the project. Once a day the application will search for new data and classify it so it can be filtered. This data will be taken from APIs and other sources like OSM and web scraping. This data will then be stored in the database so it can be filtered and searched through later.

**Criticality** - This is one of the most important aspects of the project, without it the app would just be an app to text friends. This data gathering and classification will be the backbone that the rest of the project will be built around.

**Technical issues** - This part of the project will be connected to every aspect of the app. It will be connected to the database to store the data so this has to be quick and efficient to how the data is ordered and organised. Users will be able to see this data so it has to be classified with tags so it can be filtered and used to search. This data will also be used in the recommendation part of the app.

**Dependencies with other requirements** - This will be dependent on both the event database and the user database. When the user is filtering and searching and recommendation.

#### 3.2 Getting data from user database

**Description** - When the user logs in, tries to log in or create an account they will be interacting with the user database. This database will also be used to store the users past events, conversations and interests. The recommendation feature will use this database to recommend events to that specific user.

**Criticality** - Important as user data needs to be secure.

**Technical issues** - This component will have to be secure so as to not break any gdpr guidelines that the user expects of the application.

**Dependencies with other requirements** - This will rely on the user making an account and then using the account. The more the user uses the app and interacts with the app the user database will be added to storing their interests and chats.

### 3.3 Getting data from event database

**Description** - As the user interacts with the app and applies a filter or a search the backend will call this database returning the data that is stored that matches this criteria. Users can add to this in the form of reviews and ratings and this will be stored in this database.

**Criticality** - This is also very important as without it we would have data but no way to see it

**Technical issues** - The retrieval of this data might be slow if it has to search through a large database slowing down the application.

**Dependencies with other requirements** - This will be dependent on the user searching and filtering as well as recommendations being made to the user.

### 3.4 Create an account:

**Description** - To make the most of the app an account will be needed to make use of the recommendation aspect as well as the messaging component of the app. The user will be able to log in with facebook, google and a Vicinity account. This will record things like name, gender, location and interests for the recommendation system.

**Criticality** - An account is very important as without it the messaging and recommendation components would be useless.

**Technical issues** - Storing this data in a secure manor will be important here.

**Dependencies with other requirements** - This will be heavily dependent on the user database.

### 3.5 Use without an account:

**Description** - To make the app as user friendly as possible we want it to be usable without an account. This will restrict a few of the features like chat and recommendations but it will let the user view the database, apply basic filters and search through the database.

**Criticality** - Not massively important but adds a lot to the user friendly aspect of the app.

**Technical issues** - This functionality will be available to users with accounts so we just need to remove some of the account users ability.

**Dependencies with other requirements** - This will require the event database to work letting these users search through it.

### 3.6 Event recommendations:

**Description** - The app will recommend activities that it thinks a user might like based on their previous activity. The more frequently a user interacts with the app the more accurate the app will be in predicting suitable events for that user in future.

**Criticality** - This is a fairly important part of the app because we want to make so a user is given ideas of what to do quickly rather than having to manually search themselves.

**Technical issues** - If a user has not used the app much before and has not entered any interests on sign up then they system may not be able to recommend appropriate activities right away.

**Dependencies with other requirements** - This will depend on the user database to find user interests and the event database to match these interests against.

### 3.7 Finding event using filters:

**Description** - We wish to add the ability to filter through events for ease of navigation through the app. An example would be a user who wishes to find a music event near them. They will have the option to filter by music and the app will only display music events for them to view.

**Criticality** - Not overly critical but nice to have for improved usability.

**Technical issues** - Will have to ensure that events get correctly filtered into categories so they can be displayed properly.

**Dependencies with other requirements** - This will depend on events being automatically categorized once retrieved from the internet.

### 3.8 Finding event or place using search:

**Description** - This is similar to filtering an event except the user has more control over it. They will have the ability to type keywords into a search bar in order to try to find an event that matches it.

**Criticality** - Not overly critical but nice to have for improved user experience.



**Technical issues** - The request may have to be a general term. If a user enters a very specific search request they may not find what they are looking for even though the event in question may exist but just not match exactly what the user has entered.

**Dependencies with other requirements** - This will depend on all events having some keywords in which they can be located with

### 3.9 Adding users as friends:

**Description** - Users who have made a profile will be able to add other users as friends. This can be used in order to direct message people through the app and make groups with people.

**Criticality** - This is an important feature for the app as it adds a social aspect to it and also allows for sharing events between people.

**Technical issues** - Searching a person's username may yield many results so you will have to know the users ID in order to add the correct friend.

**Dependencies with other requirements** - This will be dependent on the user database having users and being able to connect them with each other and share their data with each other.

### 3.10 In app communication:

**Description** - This will involve users being able to direct message each other from within the app. Users can add each other as friends if they are logged into their account and message people once friends with them. Users will also have the option to create group chats where multiple people can message each other in the one space.

**Criticality** - This is not a critical part of the app but it adds a social aspect to it which fits in well as a lot of events and activities are often better in groups.

**Technical issues** - Users will not be able to send or receive messages unless they have an active internet connection.

**Dependencies with other requirements** - This will depend on users being able to add each other as friends first.

### 3.11 Sharing events:

**Description** - This feature allows users to share events between friends. If a user sees an event and thinks that a friend or a group of friends might be interested in it also they will have the option to share it to a chat in order to show their friends. This will provide a link to that event once shared in the group.

**Criticality** - This is an important part of the app as it is a way of users aiding the app in recommending events for users in future by networking the events themselves.

**Technical issues** - Sharing events that are over will have to be dealt with.

**Dependencies with other requirements** - A direct message chat or a group chat with other users will have to be made before sharing an event.

### 3.12 Reviewing events and activities:

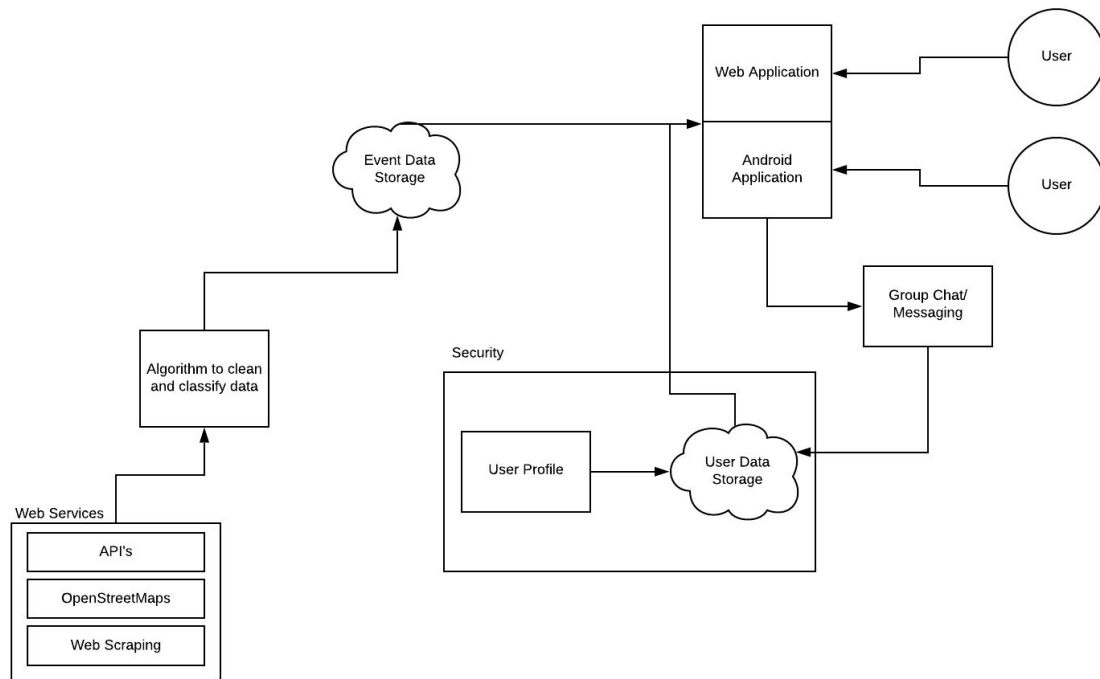
**Description** - This feature allows users to leave reviews in the form of comments under activities. This will help in providing key information about the activity that a user may not have seen in the description and help a user identify anything that needs to be done in preparation before attending as well as helping them choose if the event would suit them.

**Criticality** - This is not critical as information about the event will already be provided but it is still helpful for the users to see.

**Technical issues** - False comments and reviews will have to be dealt with and removed if found.

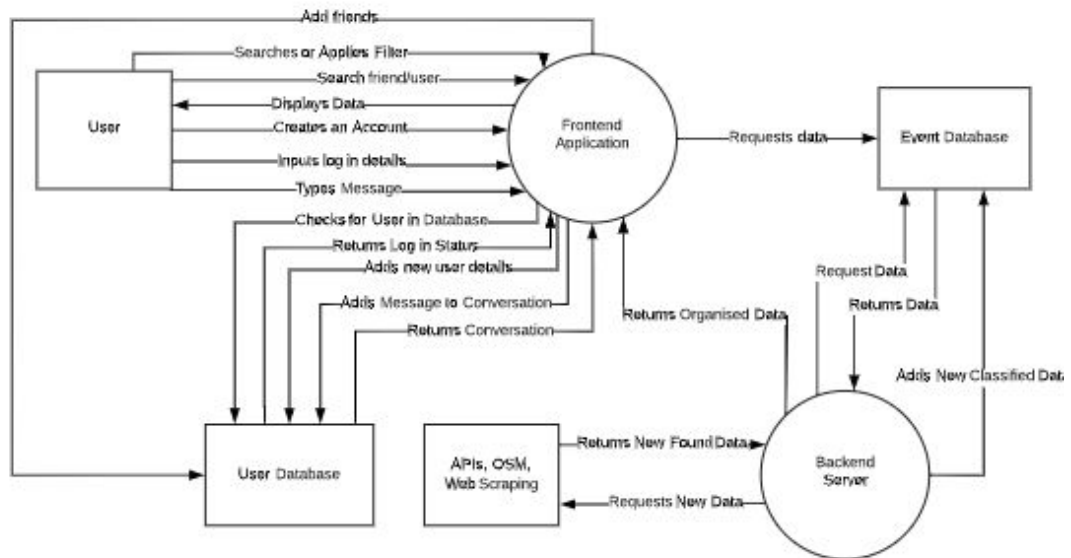
**Dependencies with other requirements** - This depends on the user being logged in for them to leave a comment.

## 4. System Architecture

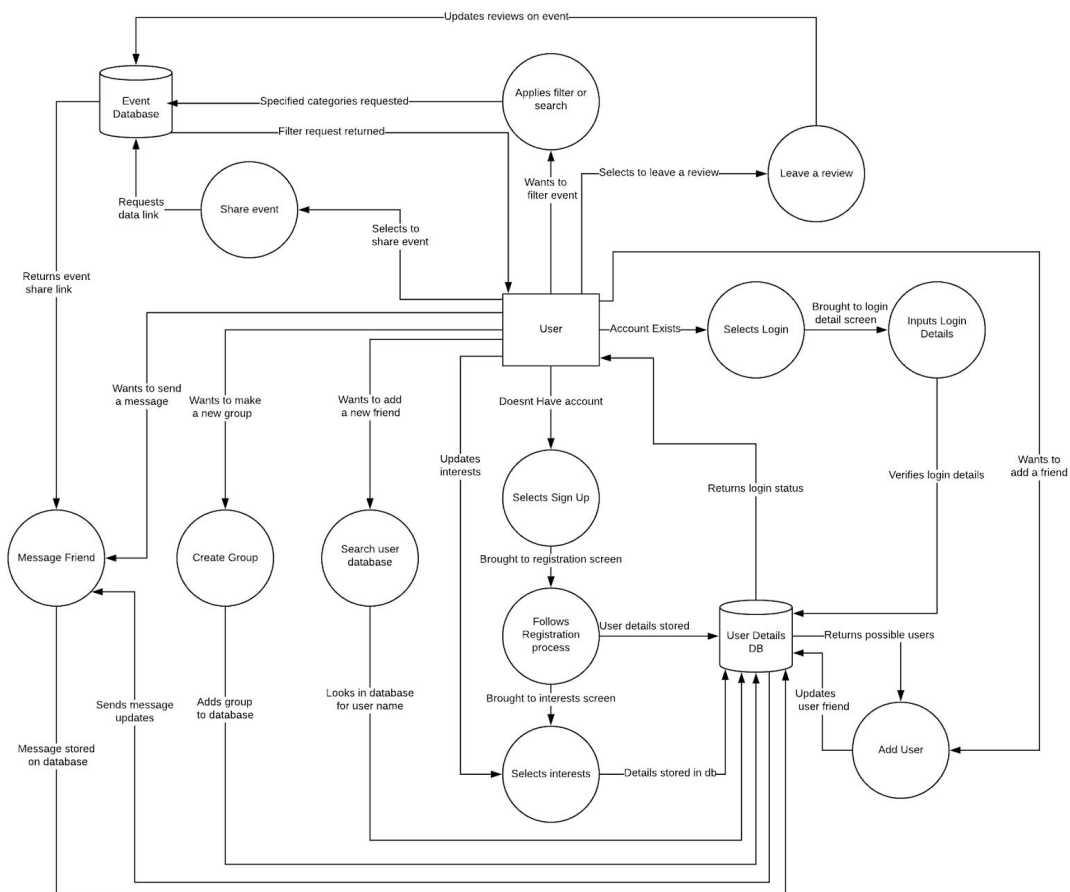


## 5. High-Level Design

### 5.1 Context Diagram



### 5.2 Data Flow Diagram

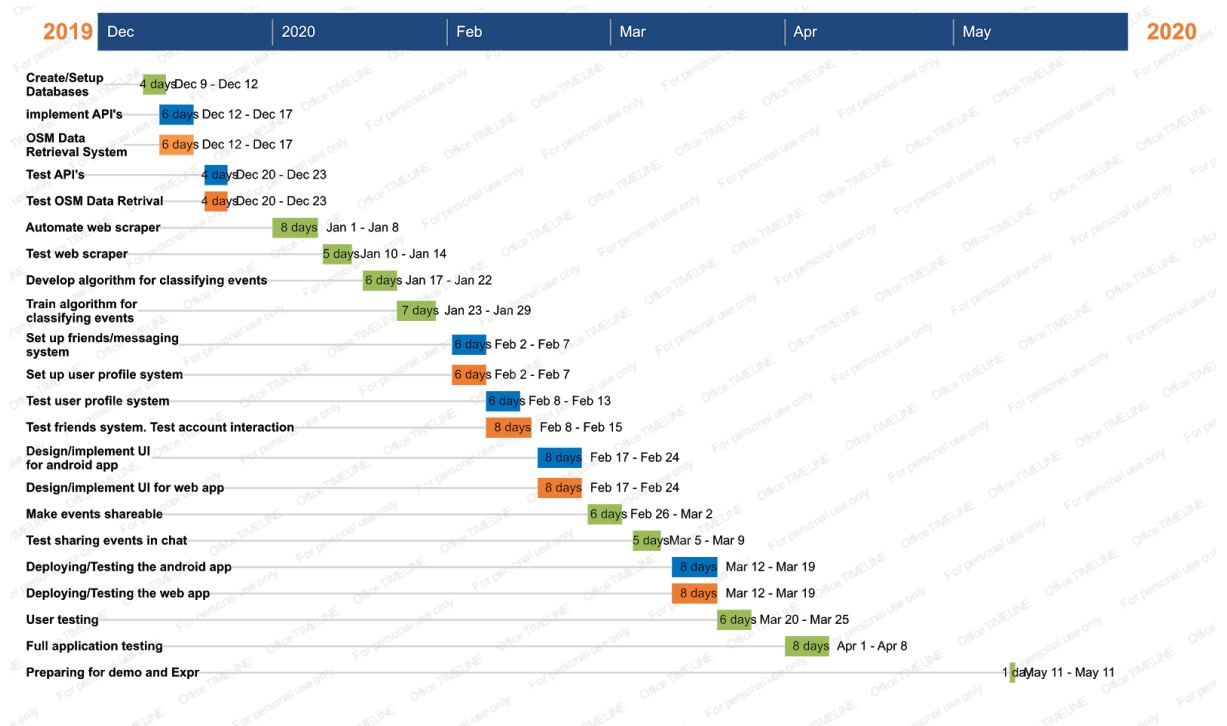


## 6. Preliminary Schedule

### 6.1 Schedule

Task No.	Task name	Duration (Days)	Start Date	End Date	Predecessors	Person
1	Create/Setup Databases	3	09/12/2019	12/12/2019		Adam McElroy, Eimhin Dunne
2	Implement API's	5	12/12/2019	17/12/2019	1	Adam McElroy
3	Test API's. Check data is going into database correctly	3	20/12/2019	23/12/2019	2	Adam McElroy
4	Develop System for retrieving OSM data	5	12/12/2019	17/12/2019	1	Eimhin Dunne
5	Test system for OSM data	3	20/12/2019	23/12/2019	4	Eimhin Dunne
6	Automate web scraper	7	01/01/2020	08/01/2020	1	Adam McElroy, Eimhin Dunne
7	Test web scraper	4	10/01/2020	14/01/2020	6	Adam McElroy, Eimhin Dunne
8	Develop algorithm for classifying events	5	17/01/2020	22/01/2020		Adam McElroy, Eimhin Dunne
9	Train algorithm for classifying events	6	23/01/2020	29/01/2020	9	Adam McElroy, Eimhin Dunne
10	Set up user profile system	5	02/02/2020	07/02/2020	1	Adam McElroy
11	Test user profile system	5	08/02/2020	13/02/2020	10	Adam McElroy
12	Set up friends/messaging system	5	02/02/2020	07/02/2020	1	Eimhin Dunne
13	Test friends system. Test account interaction	7	08/02/2020	15/02/2020	12	Eimhin Dunne
14	Design/Implement UI for android app	7	17/02/2020	24/02/2020		Adam McElroy
15	Design/Implement UI for web app	7	17/02/2020	24/02/2020		Eimhin Dunne
16	Make events shareable	4	26/02/2020	02/03/2020	2,4,6	Adam McElroy, Eimhin Dunne
17	Test a shared event is successful in chats	4	05/03/2020	09/03/2020	16	Adam McElroy, Eimhin Dunne
18	Deploying/Testing the web app	7	12/03/2020	19/03/2020	All of the above	Eimhin Dunne
19	Deploying application as android app	7	12/03/2020	19/03/2020	All of the above	Adam McElroy
20	User testing	5	20/03/2020	25/03/2020	All of the above	Adam McElroy, Eimhin Dunne
21	Full application testing	7	01/04/2020	08/04/2020	All of the above	Adam McElroy, Eimhin Dunne
22	Preparation for demo and Expo					

### 6.2 GANTT Chart



Green = Adam, Eimhin

Blue = Adam

Orange = Eimhin

## 7. Appendices

Android Studio documentation: <https://developer.android.com/docs>

Firebase Documentation: <https://firebase.google.com/docs>

OpenStreetMap documentation: [https://wiki.openstreetmap.org/wiki/Main\\_Page](https://wiki.openstreetmap.org/wiki/Main_Page)