**2. Feature list**

**Required Features**

+ Easy user interface and straight forward application usability.

+ The application must be OS and Android compatible

+ Application is REQUIRED to work offline.

+ Ability to import / export data by inputting an email address to send to.

* It will be exported as GeoJSON file, where users can then clean the dataset then use it for mapping / analysis
* Users should be able to import GeoJSON type file from their local storage, which will then be synchronized into the application’s local database.

+ Data collection

* Data can be captured by three types of geometry; lines, points or polygon.
* One line input can have multiple lines. For each line it should include line’s name, location and metadata.
* Point input should be a single point. For each point it should have point’s name, location and metadata.
* Polygon should have multiple points with a closing point. For each point it should have point’s name, location and metadata, simply click a button to extend each points.
* Meta-data should be interpreted into 4 basic fields; text, integer, float and customizable user input.
* An unique ID (gathered from user login), logged date, time and location (latitude, longitude) should be automatically be filled by using GPS.

+ Users should be able to filter view their dataset by location and type of data.

* Users should also be able to view all of their logged data in a tabular format listing its ‘title (or short description)’ and ‘location’
* They should be able to filter results by location or data type

+ Accurate location data, with an icon or a label indicating GPS is currently active.

* GPS should be highly accurate, and it should alert the user if GPS signal gets weak, as precise locations are important.

+ Each user must have a form of identification, so when they log the data, they hold accountability.

* User should be able to only log in once and not be asked to log in again unless they have manually signed out or exited the application.

+ Application must be able to view, gather, log data whether the user is offline or online.

* Application should store the data in a local storage, then make synchronizations with the main database every few cycles or once connection is established. This will prevent; noisy data, loss of data, frustration from the users.

**Complementary Features**

+ Live update of maps using OSM (Open Street Map) showing current location

* It should show the current location, with an icon of indicator.
* It should display collected data on the map using different types of icons related to the dataset.

+ The application should have some form of map interface, where users could visualize their dataset on the map (only focus on gardens point area at the moment)

* Ability to open / import GeoJSON polygon file from email and display its contained data set on the map

+ Exporting their data to a cloud storage

+ Data collection

* Data collection should be flexible to inputs, (e.g. students should be able to name their own data provided with common fields, like trees, bins, Wi-Fi and etc.)

+ Data storage

* It should accept pictures, videos and voice files as its input values.

**3. Project Timeline**

Present a timeline for the implementation of your user stories that reflects the task breakdown and the distribution of work between team members.

Development tools that will be used to support our projects are. Git for source code repository, our group will be utilizing github, by using branches, each member will have their own branches and we will work on it until a major functionality has been successfully completed which will then be merged into the master branch, this way, it will prevent merge errors where other member’s code could be effected by this merge. Github will also be used for project management, users are able to create tasks through the official github website and are able to create separate tabs which could be ‘todo’, ‘in progress’ and completed.

Group will be frequently communicating through facebook group messenger and will be holding several physical meet ups throughout the project.

|  |
| --- |
| Compulsory User Story |
| As a *GeoApp user* I want to be able to *easily understand UI layout* so that *I can operate the application without difficulties* |
| As an *GeoApp user* I want to be able to *use the application without the requirement of internet connection so* that *I don’t lose my progress if my internet disconnects.* |
| As a *GeoApp user* I want to be able to *run GeoApp through the use OS and Android operating systems* so that *I am not limited to one OS platform* |
| As a *GeoApp user* I want to be able to *import and export my collected data through email as a GeoJSON file type* so that *I can easily transfer or analyse data on my computer.* |
| As a *GeoApp user* I want to be able to *collect data sets of different types of geometry (lines, points and polygon)* so that *I can collect wide range of data types.* |
| As a *GeoApp user* I want to be able to *view my unique ID on the datasets* so that *I can claim accountability for my datasets* |
| As a *GeoApp user* I want to be able to *view all of my collected datasets in tabular format and be able to filter results* so that *I can view what kind of data I have collected and which type / location.* |
| As a *GeoApp user* I want to be able to *have an form of indication of GPS* so that *I can be aware when the GPS signal is lost.* |

|  |
| --- |
| Complementary User Story |
| As a *GeoApp user* I want to be able to *view my collected dataset through an OSM (open street map)* so that *I can visualize my datasets.* |
| As a *GeoApp user* I want to be able to *import / export my collection of datasets to cloud storages* so that *I conveniently access and manage my data without having to login to email.* |
| As a *GeoApp user* I want to be able to *input a customizable dataset of common data types* so that *I can be flexible when inputting data.* |
| As a *GeoApp user* I want to be able to *input picture, video or voice files as an input field* so that *I can provide additional information to the dataset.* |