# FACULTY OF SCIENCE AND ENGINEERING

**SEMESTER 2, 2018**

**IAB330 - Mobile App**

**Development**

**Assignment 2: Project management**

**(10 marks)**

**Due Date: Monday, September 10, 11:59 pm**

**Submission Coversheet Declaration**

You must sign below. By signing this form you agree to the following:

* We declare that all of the work submitted for this assignment is our own original work except for material that is explicitly referenced and for which we have permission, or which is freely available (and also referenced).
* We agree that QUT may archive this assignment for an indefinite period of time, and use it in the future for educational purposes including, but not limited to: as an example of previous work; as the basis for assignments, lectures or tutorials; for comparison when scanning for plagiarism, etc.
* We agree to indemnify QUT and hold it blameless if copyright infringements are found in this work and the copyright owner takes action against QUT that is not covered by the normal terms of Educational Use.

The assignment should be completed in a team of 4 students. Please consult with your tutor and unit coordinator if you have any issues.

|  |  |  |
| --- | --- | --- |
| **Chosen Project Name:** Geo App | |  |
| **Team Member Details** | |  |
| **Student Number** | **Student Name** | **Signature** |
| n9656286 | Dean McHugh | dean mchugh |
| n9720316 | Patric Marchant | Patric Marchant |
| n9967273 | Donghyeon Kim | Donghyeon Kim |
| n9460314 | Jade Kelley-Reskic | Jade Kelley |

Overview

In Assignment 2, you are expected to submit a detailed and scoped-down project plan of your Assignment 1 proposal. The project planning phase includes an executive summary of the minimum viable product (MVP), the breakdown of tasks, estimation and planning, quality assurance, and planning of future resources.

1. Executive Summary of MVP

Present a scoped down version of your original project that can be realised within the timeframe of the unit.

o Discussion of Scoping Criteria

Discuss and justify why particular features were included/excluded from the MVP.

In terms of the overall functionality and features of the application, our team decided to focus less on implementing new features or additions, and more on polishing core functionality and design described by the stakeholders for the project. because of this, there were not many features we originally planned, but decided against due to time constraints. One of the features that we originally planned, but we may not have time for is accuracy testing. Because of the inaccuracy of GPS technology (usually accurate of within 10 meters), this functionality would provide users with a way to test the location of their GPS to compare against where the location of the device actually is. In the end, this feature was shifted to a “maybe”, as while it would assist users seeking more accurate plotting of points, accuracy down to the meter would not be necessary for most people.

In the end, our Minimum Viable Project was modeled very closely from our interview with the project stakeholders; the features that were the core of the applications functionality, as well as other features they said were necessary to implement were listed as a “must implement”. Our final MVP was being able to plot points, lines, and polygons and store them temporarily in the app’s internal storage while offline, then to be able to export the plotted points if/when an internet connection is present, the user should also be able to fill input fields to leave information about what the plotted data actually describes, such as a name and other metadata including a float, integer, and string field. The user should also be able to import plotted data from other users, and view previously logged data either on the main screen of the application on a interactive map, or in tabular format, each piece of data will be able to be selected and viewed in further detail to show the description of the points, as well as the device ID of the user who logged it. This application described is the MVP, as it includes/implements all of the features described as “musts” by the stakeholders in our meeting.

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.

|  |
| --- |
| 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 (Future) 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.

|  |
| --- |
| 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.*** |

**3. Project Timeline**

**Week 4 slides**

**Exceptional project management plan that captures all required aspects**

**The timeline is well structured and shows the tasks of each team member, and the task breakdown.**

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

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 affected 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.

4. **Security and privacy requirements**

To provide a secure and private application we will focus of four areas of

weakness identified while discussing the build process. The four areas will be securing the code, securing the device, securing the data, and secure transactions.

The application will be a native app so the code will be stored in the users device. This will expose the code to potential attacks from other applications or malware on the device. Measures to manage this vulnerability given the limited time frame for building will be continuous testing of the code. This process will allow us to identify any weaknesses in the application while performing functions. The code could potentially be downloaded and modified by third parties, from here, the modified app could then become available for download by unsuspecting users causing potential damage to the users device and is a breach of device data. Ensuring the original application is easily available to users will reduce confusion in which application to download.

The three types of data we need to secure is data in transit, at rest and in use. Data at rest will consist of the data on the user device and in the central database. This data will primarily be safeguarded using inbuilt protection such as firewalls and anti-virus programs. Data in use will be the location data being used by the application while collecting points of interest in a designated area. This data will be secured by limiting access with authenticated login to the collectors device until it is uploaded to the central database. Finally data in transit will be the location data collected by the device being sent to the central database, this will be the point when the data is most vulnerable. The most simple option to protect the data in transit is to send it through email using end to end HTTPS protocols. Using this process will encrypt the data while in transit.

The privacy of the users will also need to be protected, this includes personal information about the user, logins and passwords. This can be achieved by minimising the amount of information the user needs to provide for verification. Primarily the user should only need to provide a email, username and password when creating an account. The user can then be allocated a user number for identification, this number can then be attached to all points collected by the user to ensure privacy and accountability for the data generated.

**5. Testing and Quality Assurance Strategy**

Testing will be important for the project as it ensures that all features are working as expected and that end users get a stable application and not encounter any bugs. The most fundamental part of testing any software is unit testing, which ensures that a segment of the code is providing the right output for the given inputs. Unit testing is the most appropriate type of testing for the GeoApp given the limited time of development as it’s the simplest and the most convenient. In Visual Studio, code can be tested at the unit level (functions and methods) by setting breakpoints which allows the development to be more appropriate as opposed to running the emulator every time a piece of code needs to be checked. In the case for our application, unit test will be run when implementing the features described in the user stories and only mark a story as complete when test cases are passed. A template like the one below will used when undertaking unit testing of features.

User Story:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Input | Expected Output | Actual Output | Result |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Functional testing will also be undertaken as it will ensure that a certain feature works properly with simulated human interaction. Xamarin accommodates for this type of testing for mobile apps through simulation of user inputs and behaviour on the application emulator. Code will be written to test possible interaction such as clicking a button, inputting data, exporting file etc. Once the application is in its final stages of completion, members of the development team will test the app in the field to ensure that it is functional. While field testing will not be scripted, members will still be assigned loosely defined task so it can be insightful rather than each member doing random test. Furthermore, students who will be using the application will be contacted throughout the development and testing phase to get more insight on how they interact with it and possibly have them participate in the field testing.