

ECSE 321  
Intro to Software Engineering

**Progress Report**

Group 1

Asma Alromaih - 260683238

Diana Serra -

Ilana Haddad - 260687661

Jessica Udo - 260627159

Thomas Hannaford - 260511654

## Table Of Contents

<b>1 Main Tasks Completed</b>	<b>3</b>
<b>2 Leadership roles</b>	<b>4</b>
<b>3 Work Hours</b>	<b>5</b>
3.1 Work Hours for Deliverable 1	5
3.2 Work Hours for Deliverable 2:	6
3.3 Work Hours for Deliverable 3:	7
3.4 Work Hours for Deliverable 4:	8
3.5 Work Hours for Deliverable 5:	8
3.6 Work Hours for Deliverable 6:	9
<b>4 Key Design Decisions</b>	<b>9</b>
4.1 Deliverable 1	9
4.1.1 Interface and functionalities	9
4.1.2 Use Cases	9
4.1.3 Activity Diagrams	10
4.1.4 Sequence Diagram	10
4.2 Deliverable 2	10
4.2.1 Changes to domain model	10
4.3 Deliverable 3	10
4.3.1 Changes to Domain Model:	10
4.4 Deliverable 4	11
4.5 Deliverable 5	11
4.6 Deliverable 6	11
<b>5 Meeting Logs</b>	<b>11</b>
5.1 Deliverable 1 Meetings	11
5.1.1 Meeting 1	11
5.1.2 Meeting 2	12
5.2 Deliverable 2 Meetings	12
5.2.1 Meeting 1	12
5.2.2 Meeting 2	13
5.2.4 Meeting 3	13
5.2.4 Meeting 4	14
5.2.5 Meeting 5	14
5.2.6 Meeting 6	14
5.2.7 Meeting 7	14
5.2.8 Meeting 8	15
5.3 Deliverable 3 Meetings	15

5.3.1 Meeting 1	15
5.3.2 Meeting 2	15
5.3.3 Meeting 3	16
5.3.4 Meeting 4	16
5.4 Deliverable 4 Meetings	17
5.4.1 Meeting 1	17
5.4.2 Meeting 2	17
5.5 Deliverable 5 Meetings	18
5.5.1 Meeting 1	18
5.5.2 Meeting 2	18
5.5.3 Meeting 3	18
5.5.4 Meeting 4	18
5.5.5 Meeting 5	19
5.5.6 Meeting 6	19
5.6 Deliverable 6 Meetings	19
5.6.1 Meeting 1	19

## 1 Main Tasks Completed

- Architecture:
  - Description of subsystems and interfaces
  - Evaluation of architectural styles
  - Block Diagram
- Detailed Design:
  - Design-Level Class Diagrams for Android and Web Applications
  - Description and operations of entity, controller, boundary classes and DTOs
- Back-end Implementation of “Plant Tree”, “Cut Down Tree”, and “List All Trees” features:
  - Implementation of back-end service
  - Implementation of REST controller
  - Implementation of persistence
  - Implementation of DTOs
  - Implementation of validation checks
  - Implementation of test cases (for backend service + REST API)
- Front-end Implementation of “List All Trees” feature for Web Application
- Implementation-Level sequence diagrams for “Plant Tree” and “List All Trees”

## 2 Leadership roles

While every member of the team has a responsibility to ensure the overall success of this project, we have assigned roles that best suit the expertise of each member. This allows us ensure accountability and efficiency within the team.

### **Project Manager/Backend Development Leader: *Ilana Haddad***

#### General Tasks

- Helps maintain schedule and organizational structure to team members and external influencers.
- Develops project plan to achieve project criteria.
- Schedules meetings, and ensures project documentation is always in order.

#### Software Tasks

- Implemented Java Spring backend implementation and developed test cases for the TreePLE test classes.

### **Web Frontend Leader: *Diana Serra***

#### General Tasks

- Oversees the entire project design and development teams and ensures interfacing of all software components.

#### Software Tasks

- Along with the project manager and the Test Manager, implemented Java Spring backend implementation and developed test cases for the TreePLE test classes.
- Designed the UMPLE diagram and contributed to the design-level class diagrams.
- Implementation of Web frontend.

### **Backend Developer: *Jessica Udo***

#### General Task:

- Overall management of the documentation.

#### Software Tasks:

- Designed Implementation-level sequence diagram for the “List All Trees” method.
- Implemented the “List All Trees” use case (backend + web) components.
- Implementation of controller methods.

### **Web Frontend Developer: *Thomas Hannaford***

#### Software Tasks:

- Responsible for building the android front end and the web front end application for the TreePLE, the backend, and all infrastructure that sits behind them.
- Designed design level class diagrams for web & Android applications.

### **Android Frontend Leader: *Asma Alromaih***

#### General Tasks:

- Updates the progress report, including updating meeting minutes, design decision etc, and uploads it on GitHub.

#### Software Tasks:

- Designed the block diagram for the software architecture.
- Designed Implementation-level sequence diagram for the “Create Tree” method.
- Implemented the sustainability report service methods and tests.
- Implementation of Android frontend.

## 3 Work Hours

### *3.1 Work Hours for Deliverable 1*

Members	Hours	Summary
<b>Asma</b>	10	Overall structure of the document(1.5 hrs), brainstorming sessions(use case diagram sketch, requirements, Uml model sketch)(5hrs), Uml model & class diagram(2hr), editing requirements(1.5hr).
<b>Diana</b>	11	Two group meetings for requirements, use case diagram sketch, and domain model (5 hours), Use Case Diagram descriptions (5 hours) and helping with others' tasks (1 hour)
<b>Ilana</b>	15	Two group meetings for requirements, use case diagram sketch, and domain model (5hrs), State Chart (5hrs), Use Case Diagram completion (2hrs), contributing to others' tasks (3hrs)
<b>Jessica</b>	9	Two group meetings (5hrs), writing out documentation for meeting logs and progress report (3hrs), editing and completing deliverable (1hr).
<b>Thomas</b>	11	Two group meetings consisting of brainstorming sessions on the use case diagrams, requirements and the domain model (5 hrs), activity diagrams (3 hrs), various help and editing (3 hrs)

### *3.2 Work Hours for Deliverable 2*

Members	Hours	Summary
<b>Asma</b>	27.5	Overview of the project and tasks – Group meeting (1hr), Design level class diagrams for web & Android– Group meeting(7hrs), Block diagram (6hrs), Android front-end basics (2hrs), Architectural style documentation (10hrs), Sequence diagram for “createTree” (1.5hrs)

<b>Diana</b>	40.5	Group Meetings (14 hrs) Back-end implementation of controllers, services and tests, (10 hrs), implementation of REST services (10.5 hrs), touch ups on JUnit tests (3hrs), touchups on Domain Model (3 hours)
<b>Ilana</b>	41	Group Meetings 1-4 (14hrs), Draw.io the Design-Level Class diagrams (3hrs). Back-end implementation of controllers, services, and tests (10hrs), implementation of REST services (5.5 hrs), Final touches of web app + descriptions for boundary, entity, controller classes (6hrs), Touch ups on sequence diagrams, formatting, writing work plan for this deliverable and future iterations (2.5hrs)
<b>Jessica</b>	22	Group Meetings 1-4 (12hrs). Contribution to the back-end implementation (2hrs), and sequence diagrams (4hrs). Documentation (4hrs).
<b>Thomas</b>	24	Overview of the project and tasks – Group meeting (1hr), Design level class diagrams for web & Android– Group meeting(7hrs), Web Front-End(11hrs), Web app + descriptions for boundary, entity, controller classes (4hrs), Editing and formatting(1hr)

### 3.3 Work Hours for Deliverable 3

Members	Hours	Summary
<b>Asma</b>	22.25	First meeting - overview of the classes and methods(2hrs), second meeting - brainstorming new required methods (2hrs), Meeting 2 minutes (15mins), Meeting 3 (3hrs), Meeting 4 (5hrs), sustainability report calculations (2hrs), Unit test(3hrs), Documentation(2hr), Meeting 5 (3hrs).
<b>Diana</b>	22	First meeting - overview of the classes and methods(2hrs), second meeting - brainstorming new required methods (2hrs), Meeting 3 (3hrs), Meeting 4 (5hrs), Meeting 5 (3hrs), Fixing old Tests (3 hrs), Creating New Tests (3 hours), Documentation of tests (1 hour)
<b>Ilana</b>	24	Meetings 1&2 (4hrs), Formatting and updating documents (1hr), Meeting 3 (3hrs), Writing unit tests for updateTreeData (6hrs), Meeting 4 (6hrs), Finished individual tests (1.5hrs), Meeting 5 (2.5hrs)
<b>Jessica</b>	22	First meeting - overview of the classes and methods(2hrs), second meeting - brainstorming new required methods (2hrs), Meeting 3

		(3hrs), Meeting 4 (5hrs), Formatting and updating documents (1hr), Writing Unit tests (4hrs), Meeting 5 (3hrs), Finishing Documentation and Researching Integration Strategy (2hrs)
<b>Thomas</b>	19	First meeting - overview of the classes and methods(2hrs), second meeting - brainstorming new required methods (2hrs), Meeting 3 (3hrs), Front-end development(2hrs), Meeting 4 (6hrs), Methods implementation and testing (2hrs), Meeting 5(2hrs)

### 3.4 Work Hours for Deliverable 4

Members	Hours	Summary
<b>Asma</b>	11	Group Meeting 1 (3hrs), Documentation (2hrs), Group Meeting 2 (2hrs), Backend (sustainability report) (4 hrs)
<b>Diana</b>	11.5	Group Meeting 1 (3hrs), Documentation (3hrs), Group Meeting 2 (3hrs), Office hours (1.5 hrs), Backend Development (1hr)
<b>Ilana</b>	21.5	Re-organizing backend structure for tests, making issues/project TODOs on github (2hrs), Group Meeting 1 (3hrs), Office hours (1.5hrs), Android development (3hrs), Jenkins automation (9hrs), Group Meeting 2 (3hrs)
<b>Jessica</b>	6	Group Meeting 1 (3hrs), Research Web Deployment Tools and attempt using WebPack to facilitate front-end automation (3hrs)
<b>Thomas</b>	10	Group Meeting 1 (3hrs), Documentation (3hrs), Group Meeting 2 (3hrs), Office Hours(1hr)

### 3.5 Work Hours for Deliverable 5

Members	Hours	Summary
<b>Asma</b>	9	Presentation preparation
<b>Diana</b>	9	Presentation preparation
<b>Ilana</b>	9	Presentation preparation

<b>Jessica</b>	9	Presentation preparation
<b>Thomas</b>	9	Presentation preparation

### 3.6 Work Hours for Deliverable 6

Members	Hours	Summary
<b>Asma</b>	71	Backend - ReportServices and corresponding tests (9hrs), Backend -SurveyService editSurvey and corresponding tests (3hrs), TreeManagerService moveTree & its tests (1hr) Meeting 1 (5hrs) Android interface + meetings (45hrs), Android to backend (12hrs)
<b>Diana</b>	120	Web Frontend + meetings (45), Web Frontend(75hrs)
<b>Ilana</b>	150	Backend and tests + meetings(45), Backend(105hrs)
<b>Jessica</b>	23	Documentation(2hrs), Backend controller methods + meetings(21.5)
<b>Thomas</b>	45	Web Frontend + meetings(25hrs), Web Frontend(20hrs)

## 4 Key Design Decisions

### 4.1 Deliverable 1

When formulating our requirements and designing our diagrams we had to make several key design decisions in order to clarify any ambiguities and fully understand the needs of the client. Our design decisions were therefore taken based on the two following principles:

- The Android app would be primarily used to enter information and data relating to each new or existing tree
- The Web application would be primarily used to browse, filter and compile reports on all the information and data entered by users in the Android app

#### 4.1.1 Interface and functionalities

Using these principles, some of the key design decisions taken that were not explicitly outlined in the requirements include:

- The web application will include a map rather than a list for a more aesthetic user interface.



- In order to offer as much flexibility as possible when compiling reports by area/location, the user will be able to select the area they want to analyze by delimiting a perimeter on the map. This will use the trees' information relating to latitudes and longitudes.
- Trees that are cut down will remain in the system as historical data. They are therefore not 'deleted' or suppressed from the system. Our system currently does offer a 'delete' functionality for mistakes should a user have entered the wrong information. We will assess whether this is needed in subsequent meetings.

#### 4.1.2 Use Cases

- We have designed two separate use case diagram for the *Web* and *Android* applications for simplicity/
- There are two actors included in the android use diagram: The *Resident* and the *Professionals*. As per, the project description document, these are the users with access to the TreePLE from the android application.
- Since, the forester, scientist and municipal arborist have similar access rights, we decided to categorize them together for our Android use case diagram.
- Only the Professionals, however, have access to the TreePLE system from the Web front end, as indicated in the Web use case diagram.
- Each user acts upon multiple use cases respective to their functionalities mentioned in the Requirements section of this document.

#### 4.1.3 Activity Diagrams

- Each user, whether a local resident or professional is required to login before accessing the system.
- This log in allows us authenticate the user. Moreover, since it is required that only local residents can mark tree as planted or cutdown, it also allows us validate the location of the user and the ownership of a tree by the user.

#### 4.1.4 Sequence Diagram

- For convenience and concision, we decided to design sequence diagrams for each main user action.
- The sequence diagram may contain actions from multiple user classes to emphasize the sequence.

### 4.2 Deliverable 2

#### 4.2.1 Changes to domain model

- Add Municipality class in order to list all trees in that municipality, change attribute in tree accordingly.
- Change Local class to LocalResident
- Change Report class to SustainabilityReport
- Change location attributes to associations with multiplicity 4: associate to LocalResident and SustainabilityReport
- Add Forecast class:
  - Associate it to Location with multiplicity 4 in order to get a rectangle of area to forecast
  - Associate to tree: existingTrees, treesToCutDown, treesToPlant

- Associating new Municipality class to Forecast class
- Add Version class that is associated to Tree, Forecast, and SustainabilityReport
- Added a TreeManager class that is associated to Tree, Survey, User, Municipality, Location

### 4.3 Deliverable 3

#### 4.3.1 Changes to Domain Model:

- New class VersionManager that manages the different version of TreeManagers
- Add association from Version 1--1 TreeManager
- Remove association from Version to Tree
- Add boolean to TreeManager: isCurrent
- Removed Version class and instead just have version and versionYear as attributes to TreeManager as we have decided only the TreeManager needs to keep track of the versions
- changing association from professional 1 -- sustainability report 0..\*
- Added age as Tree attribute
- Add biodiversity index, canopy, and carbon sequestration as attributes to SustainabilityReport
- Added association from TreeManager 1 -> \* sustainability report

### 4.4 Deliverable 4

- The method calculateSustainabilityReport takes an input any arbitrary perimeter of any shape as long as the coordinates are given as a location array. It used to be only a rectangular shape.

### 4.5 Deliverable 5

- Not implementing forecast of forecast, only make a forecast of a base version 1.0, 2.0 etc.
- Not implementing age tree, because of the complications corresponding to the algorithm, as well as lack of resources online.

### 4.6 Deliverable 6

- Web application allows for adding trees and creating surveys for tree statuses.
- Added attribute newStatus to Survey class in order to keep a history of status changes for surveys of a certain tree. We realised the way we had it before we would return the tree's status so for all surveys of a tree, the status returned was the tree's current status instead of a history of status changes.

## 5 Meeting Logs

### 5.1 Deliverable 1 Meetings

#### 5.1.1 Meeting 1

Date: February 6, 2018

Time: 7:30-9:30pm - 2hrs

Location: Trottier 5103

Purpose: Deliverable 1 preparation

Members: Everyone

Meeting minutes:

- Introductions
- Overview
- Listing of requirements
- Created use models for Android and Web
- Action Items for next meeting
  - We should all come up with our own domain models
  - Brainstorm the other models to be completed in Deliverable 1
- Next Meeting
  - Scheduled for Saturday, February 10th @ 16:00

### 5.1.2 Meeting 2

Date: February 10, 2018

Time: 4:00-7:00pm (3hrs)

Location: Trottier 5103

Purpose: Deliverable 1 preparation (continued)

Members: Everyone

Agenda:

- Compare all our individual domain models and amalgamate them into one model
- Draw our use case diagrams on draw.io & describe in natural language
- Activity diagram
- Traceability between diagrams and requirements

Meeting minutes:

- Finished domain model and began defining it with *Umple*
- Collaboratively created class diagram, activity diagrams, use case diagrams and state charts.
- Assigned remaining tasks for the "*Deliverable 1*" document
- Tasks Assigned:
  - o Convert hand-written use case diagrams to digital format: *Ilana*
  - o Convert domain model into Umple: *Asma*
  - o Change format of requirements to table + trace them to use cases: *Asma*
  - o Write use cases description for each use case: *Diana*
  - o Create domain-level state-chart: *Ilana*
  - o Create requirement-level activity diagram: *Thomas*
  - o Write progress report: *Jess*
  - o Complete meeting logs in project documentation: *Jess*

## 5.2 Deliverable 2 Meetings

### 5.2.1 Meeting 1

Date: Friday February 16, 2018

Time: 17:30-18:30 – 1 hour

Location: Trottier Building, Room 5106

Purpose: Deliverable 2 overview and allocation of tasks

Members: Everyone

Meeting minutes:

- Reading over requirements for deliverable 2
- Allocating tasks for Architecture and Detailed Design
- Work plan for next week:
  - Finish Architecture and Detailed Design (40% of deliverable) by end of Tuesday (Feb 20<sup>th</sup>) night, using Tuesday's office hours for help if needed
- Next Meeting
  - Scheduled for Monday, February 19th @ 15:30

### 5.2.2 Meeting 2

Date: Monday February 19, 2018

Time: 15:30-19:30 – 4 hours

Location: Trottier Building, Room 5106

Purpose: Completing 40% of deliverable 2: Architecture and Detailed Design

Members: Ilana, Thomas, Diana, Asma

Agenda: Everyone works on their task individually but also in coordination with others for help, guidance, and conformity

Meeting minutes:

- Creating docs in github
- Updating class diagram and key design decisions
- Updating use case diagrams
- Generated java code from .ump file with updated domain model
- Created project in eclipse and linked it to github
- Work plan for next meeting:
  - Asking all our questions during Tuesday's office hours

### 5.2.4 Meeting 3

Date: Tuesday February 20, 2018

Time: 16:30-22:30 – 6 hours

Location: McConnell 627 & Schulich 5<sup>th</sup> floor

Purpose: Office hours + design-level class diagrams

Members: Everyone

Agenda: Ask questions relating to deliverable 2

Meeting minutes:

- During office hours

- Clarification relating to forecasting. Suggestion that there needs to be a version control for each model whether it is the current situation or future models. Sustainability reports can then be evaluated based on the model that is being pointed to. This version control would also allow to create forecasting of hypothetical models and not only forecasting of the current model (Russian-doll forecasting)
- Difference between creating a tree and planting a tree
- Should try and reduce redundancies because they are essentially the same function
- ca.mcgill.ecse321.TreePLE.model instead of TreePLE.model is best practice
- Clarification relating to architecture block diagrams
- Schulich group meeting
  - Completed design-level class diagram for Android and Web

#### 5.2.4 Meeting 4

Date: Wednesday February 21, 2018

Time: 20:30-23:30 – 3 hours

Location: McLennan Library, Room M3-17B

Purpose: Begin Development of Features

Members: Everyone

Agenda: Start Implementation of features

Meeting minutes:

Splitting up tasks for implementation:

- Asma: Android front-end for plant tree and cut down tree, block diagrams
- Thomas: Web front-end for list all trees, 2 sequence diagrams
- Ilana: all of backend implementation for plant tree
- Diana: all of backend implementation for cut down tree
- Jess: all of backend implementation for list all trees

#### 5.2.5 Meeting 5

Date: Thursday February 22, 2018

Time: 14:30-15:30 – 1 hour

Location: Trottier Building, Room

Purpose: Office Hours with Mentor

Members: Diana, Ilana, Thomas

Agenda: Ask questions about class diagrams (design-level and domain model)

Meeting minutes:

- Asked our mentor for guidance, tips, and help confirming our class diagram was headed in the right direction

### 5.2.6 Meeting 6

Date: Thursday February 22, 2018

Time: 15:30-00:30 – 9 hours

Location: Trottier 5th floor

Purpose: Implementation of back-end

Members: Diana, Ilana, Asma

Agenda: Work together to implement back-end needed for features

Meeting minutes:

- Asma: finished block diagram
- Diana and Ilana: setting up controllers, services, DTOs

### 5.2.7 Meeting 7

Date: Friday February 23, 2018

Time: 13:00-18:30 – 5.5 hours

Location: Schulich 5th floor

Purpose: Implement REST services for back-end for createTree

Members: Diana and Ilana

Agenda: finish implementing REST services

Meeting minutes:

- Wrote POST requests for creating a new tree, a new municipality
- Wrote GET requests for all municipalities

### 5.2.8 Meeting 8

Date: Saturday February 24, 2018

Time: 18:00-00:00 – 6 hours

Location: Redpath Library Main Floor

Purpose: Wrap-up deliverable

Members: Everyone

Agenda:

- Showcase our individual progress so far
- Re-allocate tasks to finish deliverable requirements

Meeting minutes:

- Implemented ListAllTrees GET request + service
- Connected back-end with front-end
- Tested Web app, made minor design changes, completed implementation of list all trees feature for web app
- Wrote descriptions + operations for both design-level class diagrams
- Finished TestSurveyService and TestTreeManagerService

## 5.3 Deliverable 3 Meetings

### 5.3.1 Meeting 1

Date: Thursday March 8, 2018

Time: 14:00-15:30 – 1.5 hours

Location: McLennan Library, Cybertheque Pod 1

Purpose: Deliverable 2 wrap-up & Intro to Deliverable 3

Members: Everyone

Agenda: Make sure everyone understands all code written for deliverable 2, start talking about deliverable 3 and splitting up tasks (if time permits)

Meeting minutes:

- Went through every class in backend and explained their functionality
- Scheduled next meeting for Monday March 12th @ 15:30
  - Todo for next meeting: read deliverable 3 document

### *5.3.2 Meeting 2*

Date: Monday March 12, 2018

Time: 15:30-18:00 – 2.5 hours

Location: Trottier 5106

Purpose: Deliverable 3

Members: Everyone

Agenda: Start working on deliverable 3

Meeting minutes:

- Brainstormed most of the required methods, such as generating forecasts and calculating sustainability reports.
- Wrote pseudo-codes representing some of the method
- Updated the umple model to contain all the desired changes
- Generated a list of questions to be asked during the office hours:
  - How to deal with tree age and attributes changing as the trees grow (manually using editTreeData or automatically?)
  - Does our way of implementing forecast work? Explain with VersionManager
- Scheduled next meeting for Wednesday March 14th @ 15:30

### *5.3.3 Meeting 3*

Date: Wednesday March 14, 2018

Time: 20:30-23:30 – 3 hours

Location: Trottier 5106

Purpose: Deliverable 3

Members: Everyone

Agenda:

- Agreeing on a list of detailed methods, and ensuring all members understand them completely, and then
- Assigning the methods on members so that each member is responsible for creating the test cases of them.

Meeting minutes:

- Wrote list of all business methods to test for deliverable
- Finalized desired implementation of Forecast, pseudo code:  
createForecast(String baseVersion, String futureVersion, int futureDate)

```

For every tm in versionmanager
    Find tm that has that version → tm0
    if(tm0.isCurrent)
        Tm1 = Create duplicate tm of baseVersion (1.0 → 2.0) (copy all data)
        tm1.setIsCurrent = true
        tm0.setIsCurrent = false
    TmForecast : setVersion(futureVersion)
    tmForecast.version.setDate(futureDate)
    Copy all tree data from tm0 to tmForecast
    For every tree in tmForecast
        ageTree(constant*height, constant*diameter)

```

- Assigned two non-trivial business methods to each member to write unit tests for

#### 5.3.4 Meeting 4

Date: Saturday March 17, 2018

Time: 18:00-00:00 – 6 hours

Location: Trottier 5106

Purpose: Deliverable 3

Members: Everyone

Agenda: Write deliverable 3 report together

Meeting Minutes:

- Wrote Unit test plan, integration test plan, and system test plan in deliverable 3 report
- Finished most of the documentation for the deliverable
- Ensured members are writing accurate tests to the accurate methods
- Decided which tests are non-trivial

#### 5.3.5 Meeting 5

Date: Sunday March 18, 2018

Time: 11:00-14:00 – 3 hours

Location: Trottier 5107

Purpose: Deliverable 3

Members: Asma, Diana, Ilana, Thomas

Agenda: Merge all branches on github

Meeting Minutes:

- More on documentation
- Merged the branches and resolved all conflicts on GitHub in preparation for submission
- Finalized the document to be ready for submission

### 5.4 Deliverable 4 Meetings

#### 5.4.1 Meeting 1

Date: Monday March 26, 2018

Time: 21:00-00:00 – 3 hours

Location: McLennan Room M3-37A



Purpose: Deliverable 4

Members: Everyone

Agenda: Start documentation for deliverable 4

Meeting Minutes:

- Reading deliverable 4 requirements
- Started documentation for release pipeline plan
- Split up tasks for documentation and gradle/webpack implementation

#### **5.4.2 Meeting 2**

Date: Wednesday March 28, 2018

Time: 20:30-00:00 – 3.5 hours

Location: Redpath Library RM-19B

Purpose: Deliverable 4

Members: Ilana, Thomas, Asma, Diana

Agenda: Finish all requirements for deliverable 4

Meeting Minutes:

- Finished documentation for deliverable 4 report

### **5.5 Deliverable 5 Meetings**

#### **5.5.1 Meeting 1**

Date: Sunday April 1, 2018

Time: 13:00-20:00 - 7 hours

Location: Trottier 5104

Purpose: Continue implementation

Members: Everyone

Agenda: Plan tasks and divide work

Meeting Minutes:

- Completed plan for web display
- Web/android frontends

#### **5.5.2 Meeting 2**

Date: Monday April 2, 2018

Time: 13:00-20:00 - 7 hours

Location: Trottier 5105

Purpose: Continue implementation

Members: Everyone

Agenda: Planning web frontend

Meeting Minutes:

- Plan for web display and functionality
- General web/android frontends
- Divided tasks

### 5.5.3 Meeting 3

Date: Thursday April 4, 2018

Time: 20:00-21:30 - 3.5 hours

Location: Trottier 5104

Purpose: Continue implementation

Members: Everyone

Agenda: Plan tasks and divide work

Meeting Minutes:

- Completed plan for web display
- Web/android frontends

### 5.5.4 Meeting 4

Date: Saturday April 7, 2018

Time: 13:00-1:00 - 12 hours

Location: Redpath hall

Purpose: Finishing web and Android frontends

Members: Everyone

Agenda: Frontend ready for presentation

Meeting Minutes:

- Attempts to resolve issues in Android
- Preparing web frontend, implementing reports for sustainability

### 5.5.5 Meeting 5

Date: Sunday April 8, 2018

Time: 13:00-1:30 - 13.5 hours

Location: Redpath hall

Purpose: Continue implementation

Members: Everyone

Agenda: Finishing off the parts for the presentation

Meeting Minutes:

- Implementing forecast
- Web/android frontends

### 5.5.6 Meeting 6

Date: Monday April 9, 2018

Time: 15:30-23:30 - 8 hours

Location: Trottier 5107

Purpose: Preparation for presentation

Members: Everyone

Agenda: Complete preparation of the presentation

Meeting Minutes:

- Completed slides
- Divided parts

- Practiced the parts with timing

## **5.6 Deliverable 6 Meetings**

### **5.6.1 Meeting 1**

Date: Sunday April 15, 2018

Time: 12:00 - 3:00 - 3 hours

Location: Redpath Hall

Purpose: Preparation for submission

Members: Everyone

Agenda: Finish documents for final submission

Meeting Minutes:

- Filled the documentation on GitHub
- Creating Javadocs for the overall documents
- Finalizing documents and codes for submission