Intro to Software Engineering
ECSE 321

Kevin Chuong, 260742781
Jeremy Davis, 260744431
Oscar Décéus, 260744646
Elias Al Homsi, 260797449

DELIVERABLE #1

Presented to Mr. Daniel Varro
Faculty of Engineering

McGill University
December 11, 2018

Table of Contents

1 System Requirements **3**

2 Actors and Use cases **7**

3 Domain Model **13**

4 Statechart for class "Tree" **14**

5 Requirements-level activity diagram(s) covering the main scenario for managing trees for all stakeholders (i.e. excluding forecasting) **14**

Android Front-End (Foresters and Residents) 14

Web Front-End (Scientists and Urban Decision Makers) 15

6 Work plan for remaining iterations **15**

Team Information 15

Work Plan 16

1 System Requirements

ID	R1
Description	The TreePLE system shall allow logged in users to manage trees.
Traceability	-
Kind	Functional
Priority	1

ID	R2
Description	The TreePLE system shall allow logged in users to specify information regarding a tree.
Traceability	-
Kind	Functional
Priority	1

ID	R3
Description	The municipal arborists or environmental scientists shall be capable of marking trees as diseased or to be cut down.
Traceability	-
Kind	Functional
Priority	1

ID	R4
Description	The TreePLE system shall store information about each tree.
Traceability	UC-TM-03

Kind	Functional
Priority	1

ID	R5
Description	The TreePLE system shall allow users to log in.
Traceability	UC-AM-02
Kind	Functional
Priority	1

ID	R6
Description	The TreePLE system shall encrypt the users' data.
Traceability	-
Kind	Non-Functional
Priority	2

ID	R7
Description	The TreePLE system shall allow users to sign up.
Traceability	-
Kind	Functional
Priority	1

ID	R8
Description	The TreePLE system shall list all trees.
Traceability	UC-TM-04
Kind	Functional
Priority	1

ID	R9
Description	The TreePLE system shall display forecasting information.
Traceability	-
Kind	Functional
Priority	2

ID	R10
Description	The TreePLE system shall plot each tree on a map
Traceability	-
Kind	Functional
Priority	1

ID	R11
Description	The residents shall be capable of managing trees(ie. Cut down a tree / Plan a tree).
Traceability	UC-TM-01
Kind	Functional
Priority	1

ID	R12
Description	The TreePLE system shall plot trees with specific status on a map.
Traceability	-
Kind	Functional
Priority	2

ID	R13
Description	The TreePLE system shall automatically calculate sustainability attributes.
Traceability	UC-TM-02
Kind	Functional
Priority	2

ID	R14
Description	The Tree PLE system shall have an average down time of 1 hour per month.
Traceability	-
Kind	Non-Functional
Priority	2

ID	R15
Description	The TreePLE system shall register transactions.
Traceability	-
Kind	Functional
Priority	1

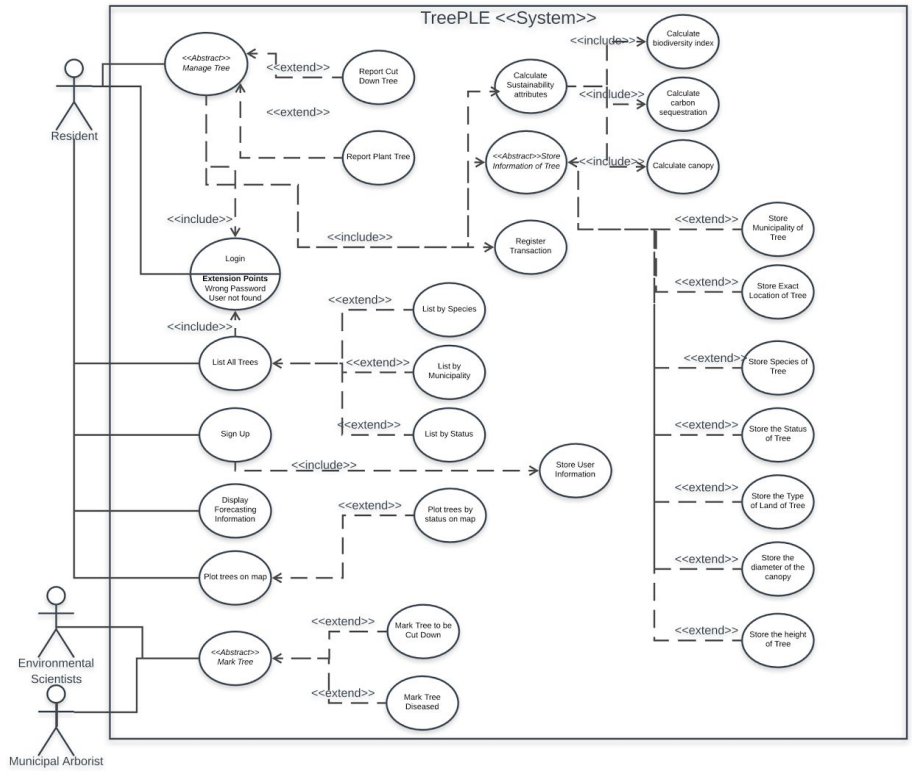
ID	R16
Description	The TreePLE system shall timeout users after 10 minutes of inactivity.
Traceability	-
Kind	Nonfunctional
Priority	2

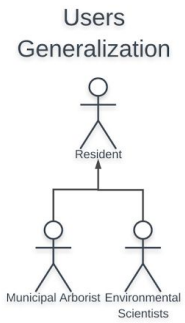
ID	R17

Description	The TreePLE system shall be usable in the french and english language
Traceability	-
Kind	Non-Functional
Priority	2

ID	R18
Description	The TreePLE system shall store information of signed up users.
Traceability	-
Kind	Functional
Priority	1

2 Actors and Use cases





Descriptions of Use cases

Use Case: Report a Tree as Planted <<extends>> Manage Tree
Successful Outcome: Primary Actor reports a Tree as planted.

Use Case Package	Tree Management
ID	UC-TM-01
Use Case Goal	Primary actor successfully reports a Tree as planted
Actor(s)	Primary Actor: Resident
Level	User-Goal
Precondition	Resident has successfully logged in as per {UC-AM-02}
Domain Entities	Tree, Resident

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to report a Tree as planted.	
2	System prompts Primary Actor to enter new Tree location.	
3	Primary Actor performs Store Information of Tree {UC-TM-03}	
4	System generates a unique Tree ID with the information gathered in *Main Success Scenario step 3*. Saves the new Tree.	
5	*Use case ends successfully*.	

Alternative Flows:

Step	Action	Notes
3a.1	Primary Actor Enters data that causes Store Information of Tree {UC-TM-03} to fail.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

Step	Action	Notes
3b.1	Primary Actor Enters datathat causes Store Information of Tree {UC-TM-03} to fail.	
3b.2	User Aborts.	
3b.3	*Use Case ends unsuccessfully.*	

Use Case: Login
Successful Outcome: Primary Actor logs into TreePLE system.

Use Case Package	Account Management
ID	UC-AM-02
Use Case Goal	Primary Actor successfully logs into TreePLE system.
Actor(s)	Primary Actor: Resident
Level	User-Goal
Precondition	Resident has successfully completed a Sign Up as per {UC-AM-01}
Domain Entities	Resident, Login

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to login to the TreePLE system.	
2	System prompts Primary Actor to enter login information.	
3	System validates credentials (ie. username, password).	
4	Server saves user object into a session.	
3	*Use case ends successfully*.	

Alternative Flows:

Step	Action	Notes
3a.1	Primary Actor enters information that causes credential	

	validation to fail.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

Use Case: Calculate Sustainability Attributes <<include>> Calculate biodiversity, Calculate carbon sequestration, Calculate canopy
Successful Outcome: TreePLE system calculates sustainability attributes.

Use Case Package	Tree Management
ID	UC-TM-02
Use Case Goal	Primary Actor successfully calculates sustainability attributes.
Actor(s)	Primary Actor: TreePLE system
Level	System-Goal
Precondition	Tree has been successfully planted as per {UC-TM-01}
Domain Entities	Tree, Municipality

Main Success Scenario:

Step	Action	Notes
1	Resident indicated intention to view sustainability attributes.	
2	System prompts user to enter the desired Municipality.	
3	Resident enters Municipality name.	
4	System calculates Biodiversity Index, canopy and carbon sequestrations of the given Municipality.	
5	System displays sustainability attributes to the Resident.	
7	*Use case ends successfully*.	

Alternative Flows:

Step	Action	Notes
3a.1	Resident enters an invalid municipality name.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

Use Case: Store Exact Location of Tree <<extends>> Store Information of Tree
Successful Outcome: Primary Actor stores the exact location of a Tree.

Use Case Package	Tree Management
ID	UC-TM-03
Use Case Goal	Primary Actor successfully stores the exact location of a Tree
Actor(s)	Primary Actor: Resident
Level	User-Goal
Precondition	Tree has been successfully planted as per {UC-TM-01}
Domain Entities	Resident, Tree

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to store the exact location of Tree.	This will happen when a user decides to report a Tree as Planted {UC-TM-01}.
2	System prompts Primary Actor to enter Tree coordinates (Latitude, Longitude).	
3	System performs Create Location {UC-TM-06}. Saves the new Location.	
4	System draws association between Tree and Location.	
5	*Use case ends successfully*.	

Alternative Flows:

Step	Action	Notes
3a.1	Primary Actor Enters invalid information for longitude and/or longitude.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

Step	Action	Notes
3b.1	Primary Actor Enters invalid information for longitude and/or longitude.	
3b.2	User Aborts.	
3b.3	*Use Case ends unsuccessfully.*	

Use Case: List By Municipality <<extends>> List All Trees.
Successful Outcome: TreePLE system lists each and every Tree in a Municipality

Use Case Package	Tree Management
ID	UC-TM-04
Use Case Goal	Primary Actor successfully lists each Tree in a Municipality
Actor(s)	Primary Actor: TreePLE system
Level	System-Goal
Precondition	Tree has been successfully planted as per {UC-TM-01} Municipality has been successfully created as per {UC-TM-05}
Domain Entities	Tree, Municipality

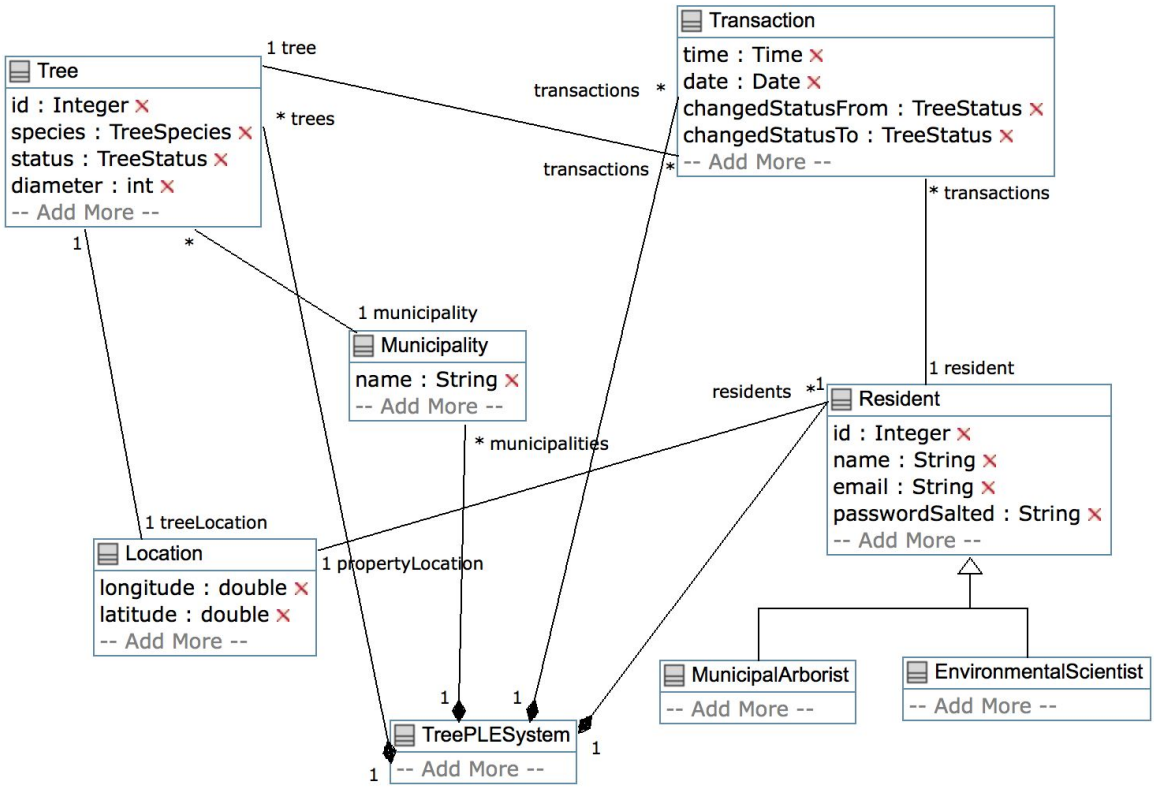
Main Success Scenario:

Step	Action	Notes
1	Resident indicated intention to view all trees by Municipality.	
2	System prompts user to enter the desired Municipality.	
3	Resident enters Municipality name.	
4	System retrieves list of all Tree objects in given Municipality.	
5	System displays list of all Tree objects in the desired Municipality to the Resident.	
6	*Use case ends successfully*.	

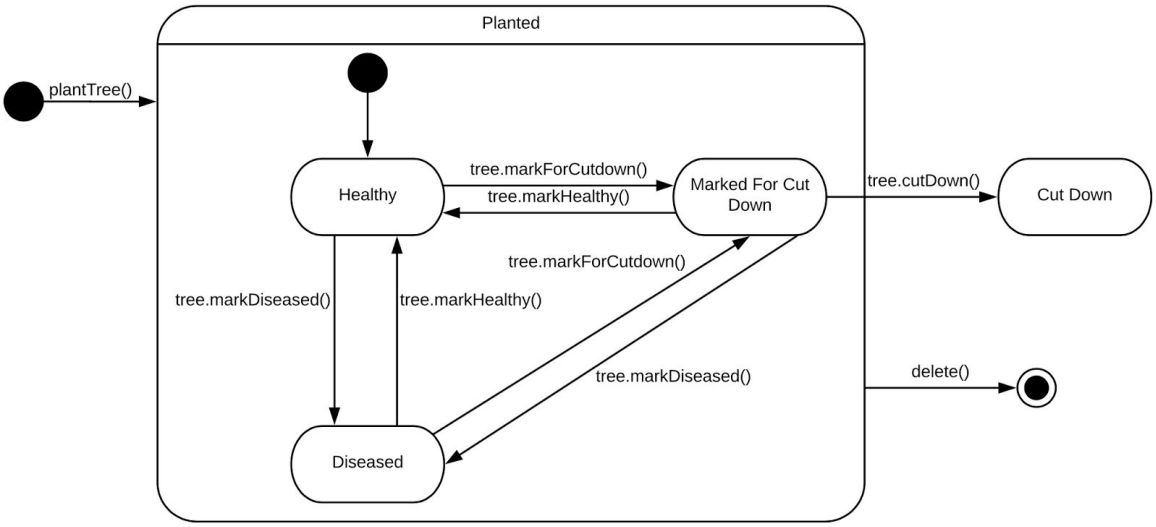
Alternative Flows:

Step	Action	Notes
3a.1	Resident enters an invalid municipality name.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

3 Domain Model

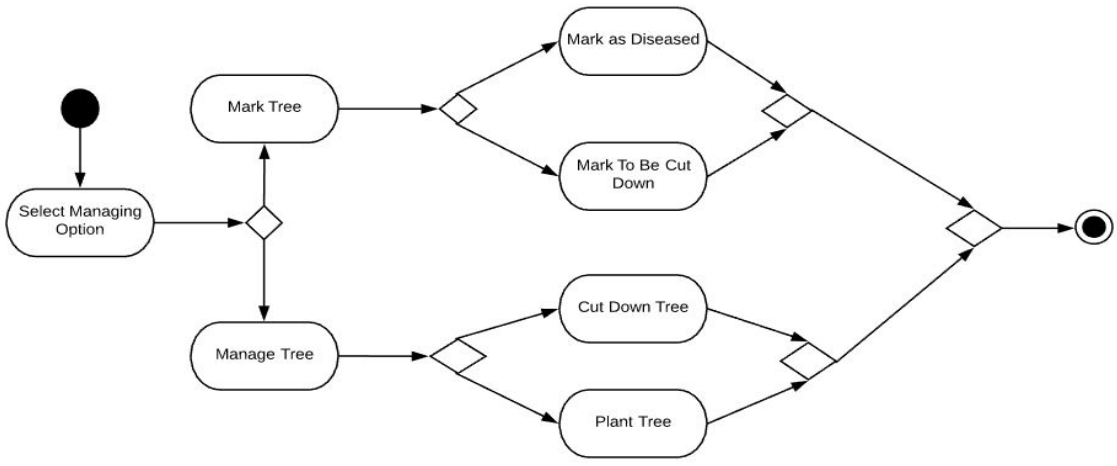


4 Statechart for class “Tree”

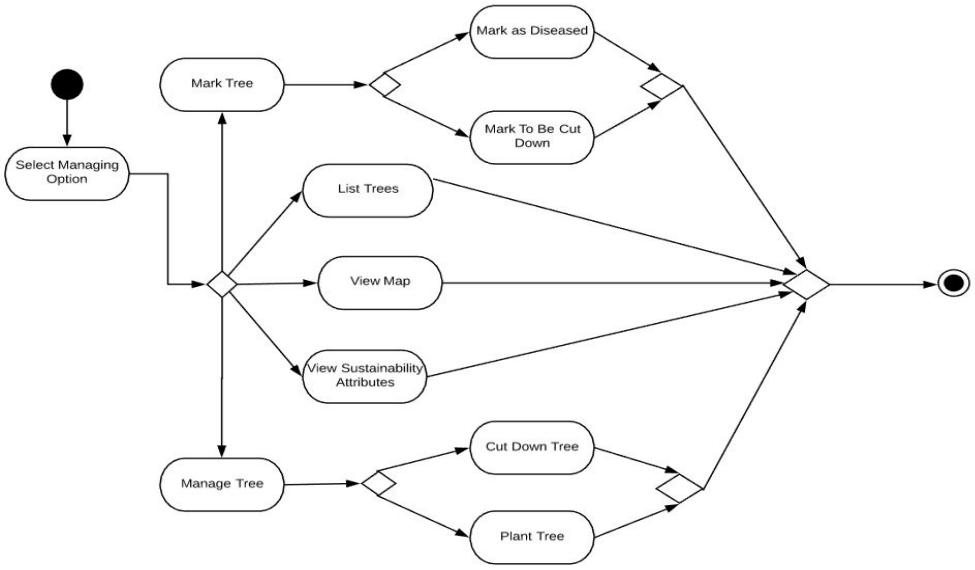


5 Requirements-level activity diagram(s) covering the main scenario for managing trees for all stakeholders (i.e. excluding forecasting)

Android Front-End (Foresters and Residents)



Web Front-End (Scientists and Urban Decision Makers)



6 Work plan for remaining iterations

Team Information

Team Members	**Main Tasks**	**Roles**	**Work hours**
Oscar Décéus	Activity Diagram, State Chart “Tree”	Testing Lead	7.5
Jeremy Davis	Use Case Descriptions, Documentation	Quality Assurance	8.0
Elias Homsi	Use Case Diagram, Domain Modeling	Software Lead	7.5
Kevin Chuong	Use Case Diagram, Documentation. (Meetings, etc)	Documentation Manager	8.0

Work Plan

Iteration	Use Case	Implementation Efforts	Estimate Date Completion of Implementation
1	Manage Tree(Plant and Cut down)	Easy	February 16
2	List All Tree	Easy	February 20
3	Mark Tree (Diseased / Cut down)	Easy	February 22
4	Store Information of Tree	Medium	February 25
5	Login / Signup	Medium	February 28
6	Store User Information	Easy	March 1
6	Register Transaction	Easy	March 2
7	Plot trees on map	Hard	March 4
8	Calculate Sustainability Attributes	Easy	March 9
9	Display Forecasting Information	Really Hard	March 18

For key decisions* see Meeting II.

Meeting Log

Group 13
February 2nd, 2018
14:30
MC10 (McConnell Engineering Building)

MEETING I

PRESENT: Kevin Chuong, Jeremy Davis, Oscar Deceus, Andrei Guevorkian
ABSENT: Armen Stepanian

1. **CALL TO ORDER**
    - **a.** 14:30
2. **OPEN ISSUES**
    - **a.** N/A
3. **INTRODUCTIONS AND CLARIFICATION OF PROJECT**

   a. Distinguished non-functional requirements and functional requirements have been clarified.
   b. Created template for requirements document on Google Docs
   c. Identified tasks for this deliverable
4. **IDENTIFICATION OF REQUIREMENTS**
   a. Began discussions on possible requirements
5. **AGENDA FOR NEXT MEETING**
   a. Establish roles for individual members.
   b. Assign respective tasks to be completed for this project deliverable.
6. **ADJOURNMENT**
   a. 15:33

Group 13
February 9th, 2018
14:35
Schulich Library 5th Floor

MEETING II

PRESENT: Kevin Chuong, Jeremy Davis, Oscar Deceus, Elias Al Homsi
ABSENT: N/A

1. **CALL TO ORDER**
   a. 14:35
2. **OPEN ISSUES**
   a. Leaving of former members, therefore no tasks assigned.
   b. Agenda from previous meeting still need to be addressed.
3. **REINTRODUCTIONS AND INITIAL VERSION OF REQUIREMENTS DOCUMENT**
   a. Quickly re-clarifying what are the tasks at hand
   b. Decided that everyone shall work at the same time on each task
      i. Reason: Everyone will understand each part as we all work together concurrently.
   c. Completed initial version of requirements document (need to revisit)
4. **CREATION OF USE CASE DIAGRAM**
   a. Brainstormed on possible use cases in the use case diagram
   b. Created initial use case diagram template with actors
5. ***KEY DESIGN DECISIONS**
   a. Generalization of certain requirements that shared the same goal (Mark tree)
   b. Introduce log in and sign up features to distinguish users (Residents/Arborists/Scientists)
   c. Record date of last survey and by whom using transactions (Transaction class)
6. **AGENDA FOR NEXT MEETING**
   a. Revisit and finalize requirements document
   b. Complete use case diagram and descriptions
   c. Create and complete a statechart diagram for “Tree” class
   d. Create and complete an activity diagram
7. **ADJOURNMENT**
   a. 17:45