

## Deliverable 1: Requirements Document

Introduction to Software Engineering

ECSE 321

### Team Members:

Marine Huynh (260743073)

Erion Hysa (260746699)

Iliana Katinakis (260740917)

Anthony Laye (260729851)

Wenzong Xia (260732931)

<b>Functional and Non-Functional System Requirements</b>	<b>3</b>
ID: R1	3
ID: R2	3
ID: R3	3
ID: R4	3
ID: R5	3
ID: R6	3
ID: R7	3
ID: R8	4
ID: R9	4
ID: R10	4
ID: R11	4
ID: R12	4
ID: R13	4
ID: R14	4
ID: R15	5
ID: R16	5
ID: R17	5
ID: R18	5
ID: R19	5
ID: R20	5
<b>Use Case Diagrams</b>	<b>6</b>
Register planted tree to municipality	6
Store status of tree	7
List all trees	8
Report Planted Tree	10
Report Cut Down Tree	11

<b>Activity Diagram</b>	<b>12</b>
<b>Domain Model and Class Diagram</b>	<b>12</b>
<b>Statechart for Class Tree</b>	<b>13</b>

## **Functional and Non-Functional System Requirements**

ID: R1

Description: The TreePLE system shall register a planted tree to a municipality.

Traceability: UC-01

Kind: Functional

Priority: High

ID: R2

Description: The TreePLE system shall store the status of a planted tree, i.e. whether the tree is healthy, diseased, planted, marked to be cutdown.

Traceability: UC-02

Kind: Functional

Priority: High

ID: R3

Description: The TreePLE system shall list all trees in a given municipality.

Traceability: UC-03

Kind: Functional

Priority: High

ID: R4

Description: The TreePLE system shall allow all residents to report if they have planted a tree within their own property.

Traceability: UC-04

Kind: Functional

Priority: High

ID: R5

Description: The TreePLE system shall allow all residents and scientists to report if they cut down a tree within their own property.

Traceability: UC-05

Kind: Functional

Priority: High

ID: R6

Description: The TreePLE system shall allow residents and scientists to log in.

Traceability: UC-06

Kind: Functional

Priority: Medium

ID: R7

Description: The TreePLE system shall register the date and scientist responsible for the most recent survey.

Traceability: UC-07

Kind: Functional  
Priority: High

ID: R8

Description: The TreePLE system shall allow users to specify the species of a newly planted tree.

Traceability: UC-08  
Kind: Functional  
Priority: Medium

ID: R9

Description: The TreePLE system shall store latitude and longitude of a planted tree.

Traceability: UC-09  
Kind: Functional  
Priority: Medium

ID: R10

Description: The TreePLE system shall store the height and diameter of a planted tree.

Traceability: UC-10  
Kind: Functional  
Priority: Medium

ID: R11

Description: The TreePLE system shall list all trees of a given species.

Traceability: UC-11  
Kind: Functional  
Priority: Medium

ID: R12

Description: The TreePLE system shall allow municipal arborists or environmental scientists to report the state of the tree; if a tree is healthy, diseased or to be cut down.

Traceability: UC-12  
Kind: Functional  
Priority: Low

ID: R13

Description: The TreePLE system shall plot each planted tree on the island of Montreal on a map.

Traceability: UC-13  
Kind: Functional  
Priority: Low

ID: R14

Description: The TreePLE system shall be compatible with the Android and Web platforms.

Traceability: UC-14  
Kind: Non-Functional

Priority: Medium

ID: R15

Description: The TreePLE system shall be able to calculate the biodiversity index of a given area within 8 seconds, 90% of the time.

Traceability: UC-15

Kind: Non-Functional

Priority: Medium

ID: R16

Description: The TreePLE system shall be capable of handling 500 simultaneous logged in users.

Traceability: UC-16

Kind: Non-Functional

Priority: Medium

ID: R17

Description: The TreePLE system shall ensure that any addition or removal of trees update the database within 5 minutes of the change, 92% of the time.

Traceability: UC-17

Kind: Non-Functional

Priority: Medium

ID: R18

Description: The TreePLE system shall provide a backup for the database every month.

Traceability: UC-18

Kind: Non-Functional

Priority: Medium

ID: R19

Description: The maintenance of the TreePLE system shall not cost more than \$2,500 per month.

Traceability: UC-19

Kind: Non-Functional

Priority: Medium

ID: R20

Description: The TreePLE system shall be able to expand the application which covers a broader surface area than the island of Montréal.

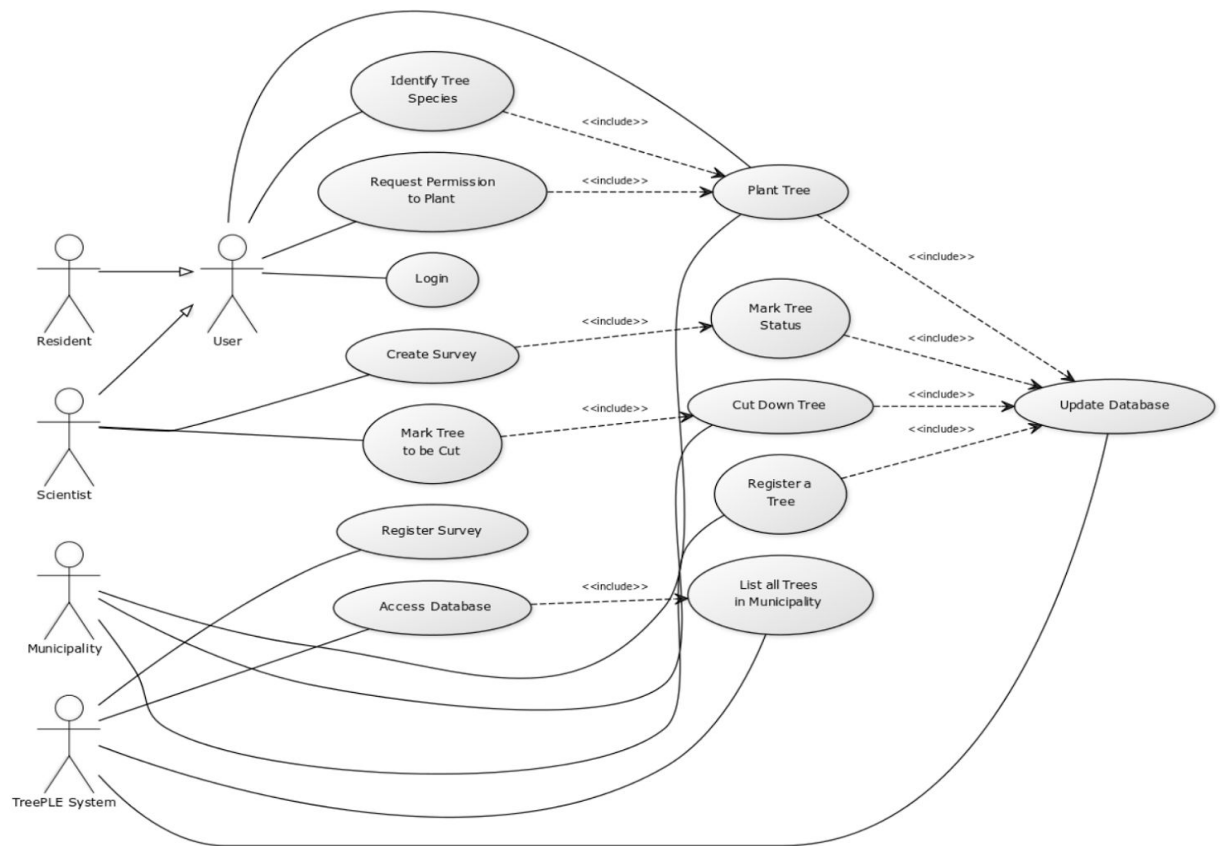
Traceability: UC-20

Kind: Non-Functional

Priority: Low

## Use Case Diagrams

The various use cases for this project can be compiled into the following diagram:



For clarity, the important use cases each have a more detailed use case diagram detailed below.

1. Register planted tree to municipality

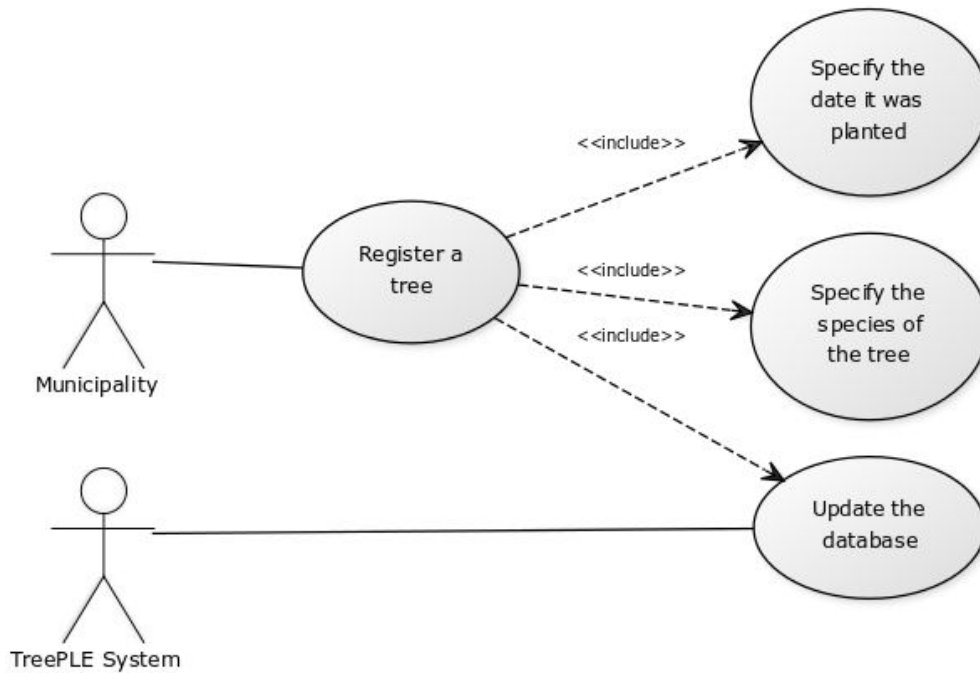
ID: UC-01

Use Case Goal: Primary Actor successfully register a tree to the system when a tree is planted.

Actor(s): Primary Actor: Municipality Secondary Actor: TreePLE System

Precondition: User, here member of the municipality, has successfully logged in as per [UC-06].

Domain Entities: Tree, Municipality, TreePLE System, Person



**Main Success Scenario:**

Step	Action
1	Primary Actor register a tree.
2	Primary Actor specifies the information on the tree, such as the species and the date it was planted.
3	Secondary Actor TreePLE System updates database to add the specified tree to the database [UC-17].
4	<i>Use case ends successfully.</i>



2. Store status of tree

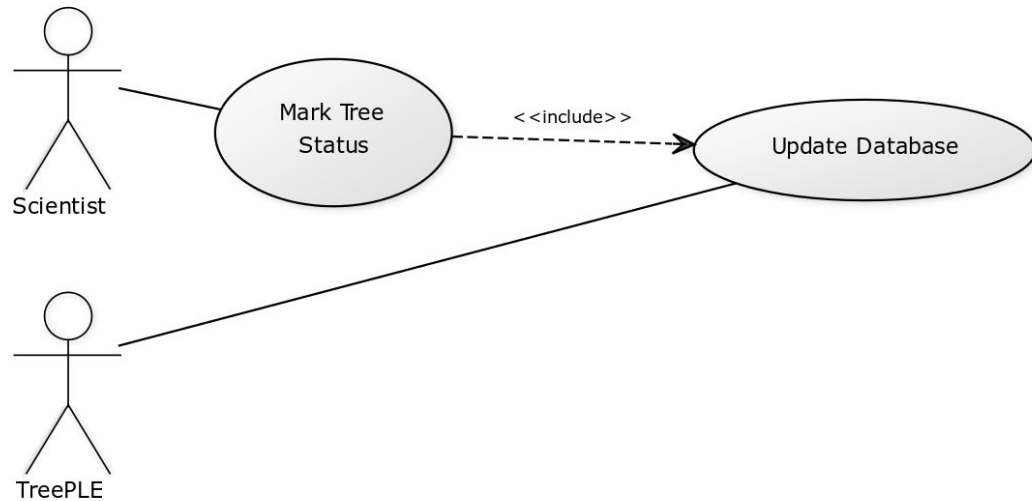
ID: UC-02

Use Case Goal: Primary Actor successfully updates the tree status to the system when the scientist completes the survey.

Actor(s): Primary Actor: Scientist Secondary Actor: TreePLE System

Precondition: The scientist has successfully logged in as per [UC-06].

Domain Entities: Scientist, Tree, TreePLE System, Survey



**Main Success Scenario:**

Step	Action
1	Scientist starts surveying the tree
2	Scientist marks tree status.
3	Secondary Actor TreePLE System updates the status of the specified tree to the database [UC-12].
4	<i>Use case ends successfully.</i>

### 3. List all trees

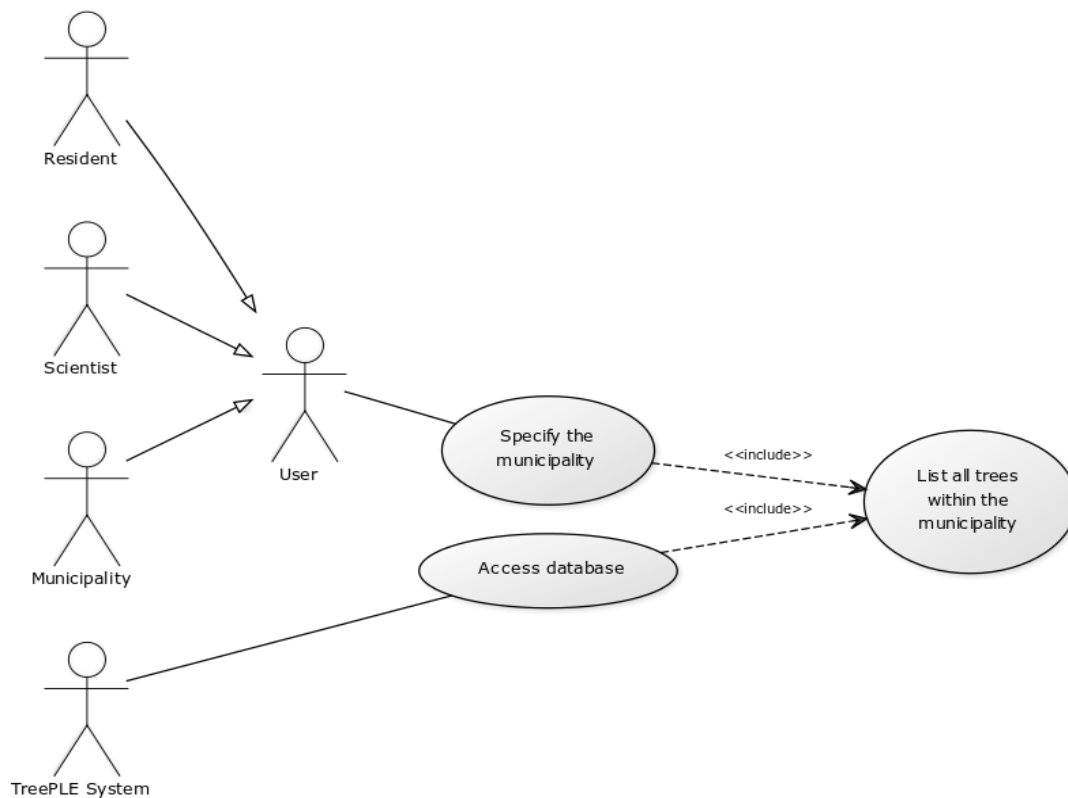
ID: UC-03

Use Case Goal: Primary Actor successfully list out all trees in a given municipality

Actor(s): Primary Actor: User, Secondary Actor: TreePLE System

Precondition: User has successfully logged in as per [UC-06]

Domain Entities: Person, TreePLE System, Tree, Municipality



#### Main Success Scenario:

Step:	Action
1	Primary Actor inputs a municipality
2	Secondary Actor TreePLE system accesses database
3	TreePLE system list out all trees within a specified municipality
4	<i>Use Case ends successfully</i>

#### Alternative Flows:

##### 2.a List all trees ends unsuccessfully

Step 2a.1	User inputs an invalid municipality
Step 2a.2	<i>User returns to main scenario Step 1</i>
Step 2a.3	<i>User case ends unsuccessfully</i>

#### 4. Report Planted Tree

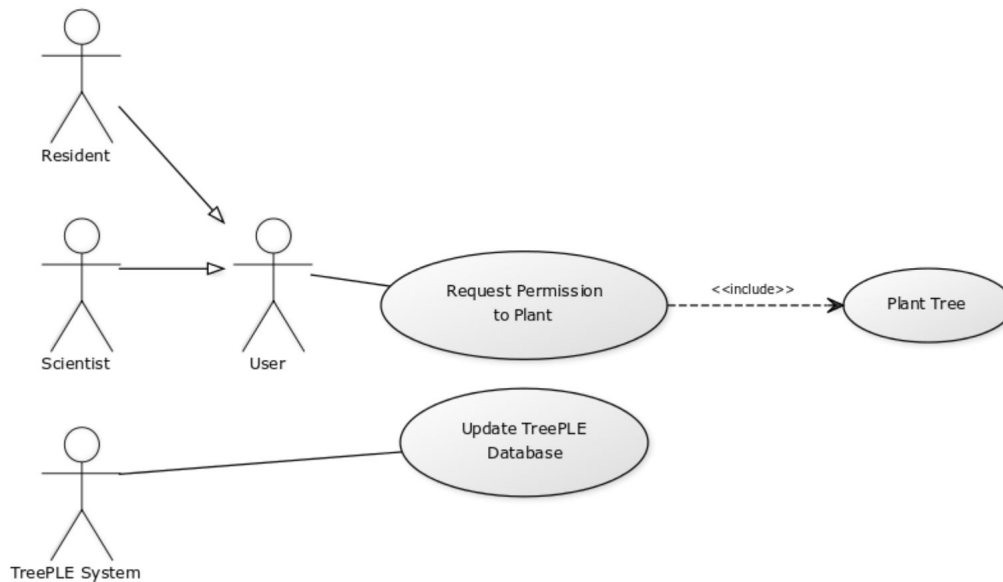
ID: UC-04

Use Case Goal: Primary Actor successfully reports a tree as planted.

Actor(s): Primary Actor: User, Secondary Actor: TreePLE System, municipality

Precondition: User has successfully logged in as per [UC-06]

Domain Entities: Person, TreePLE System, Tree, Municipality



#### Main Success Scenario:

Step	Action
1	Primary Actor selects location to plant tree
2	Primary Actor requests permission to plant tree
3	Primary Actor plants tree
4	Secondary Actor TreePLE System updates database by adding the specified tree to the database [UC-17].
5	<i>Use case ends successfully</i>

## Alternative Flows:

### 2a. Report Planted Tree ends unsuccessfully

Step	Action
2a.1	User is not authorized to plant tree in given location
2a.2	User returns to <i>Main Success Scenario Step 1</i> .
2a.2	<i>Use case ends unsuccessfully.</i>

## 5. Report Cut Down Tree

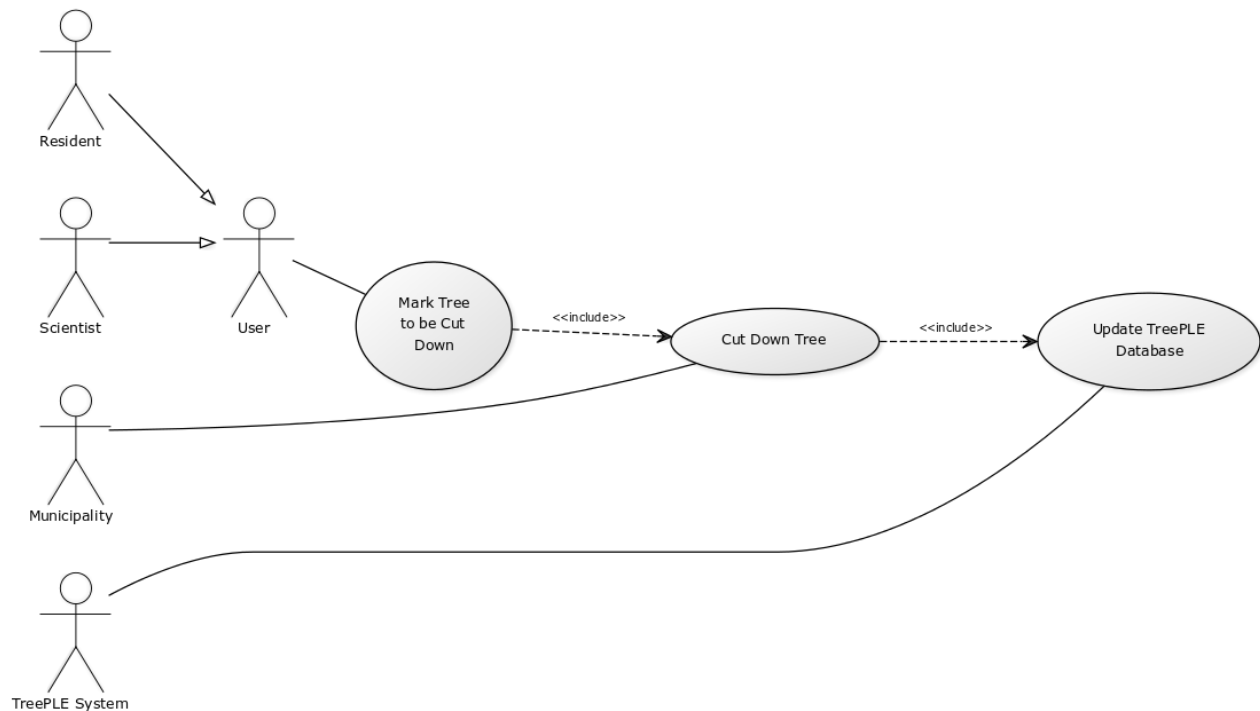
ID: UC-05

Use Case Goal: Primary Actor successfully reports a tree as cut down.

Actor(s): Primary Actor: User, Secondary Actors: Municipality, TreePLE System

Precondition: User has successfully logged in as per [UC-06].

Domain Entities: Person, Municipality, Tree, TreePLE System



## Main Success Scenario:

Step	Action
------	--------

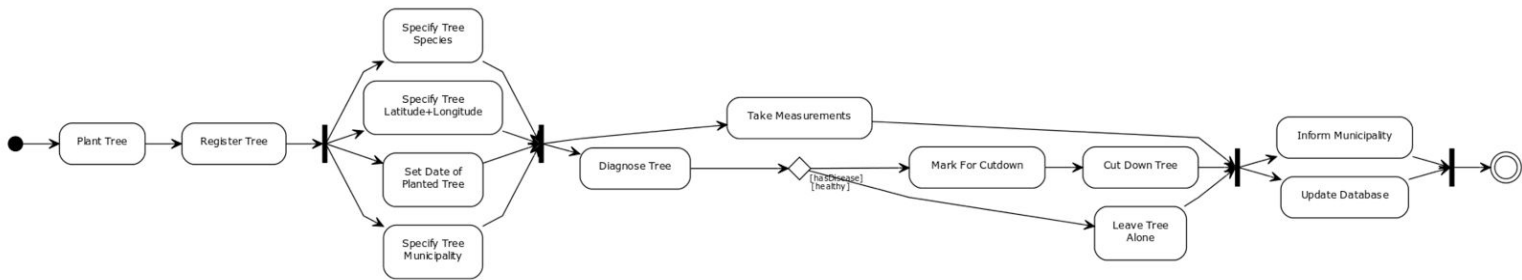
1	Primary Actor selects tree.
2	Primary Actor reports specified tree to be cut down.
3	Secondary Actor Municipality cuts down tree.
4	Secondary Actor TreePLE System updates database by removing the specified tree from the database [UC-17].
5	<i>Use Case ends successfully.</i>

**Alternative Flows:**

**2a. Report Cut Down Tree ends unsuccessfully.**

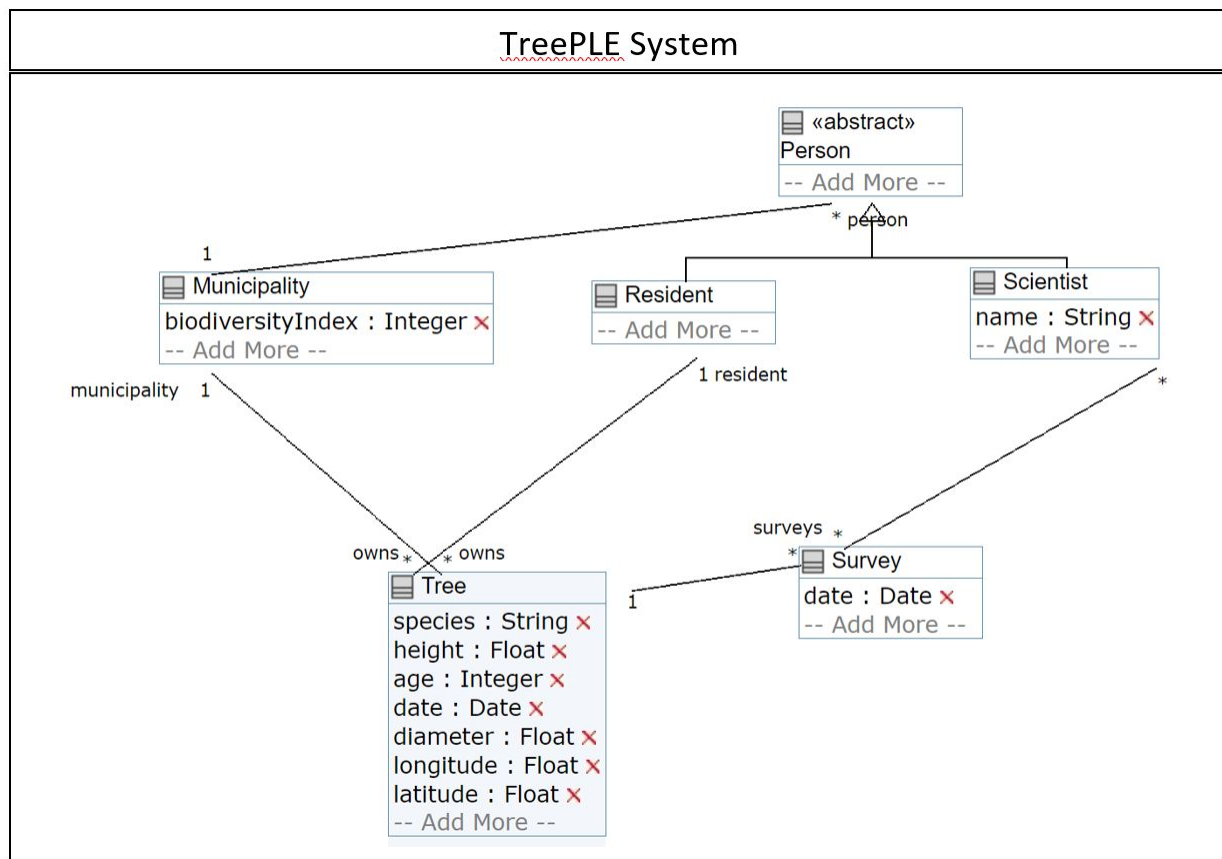
Step	Action
2a.1	User is not authorized to manage selected tree.
2a.2	User returns to <i>Main Success Scenario Step 1</i> .
2a.2	<i>Use case ends unsuccessfully.</i>

## Activity Diagram



## Domain Model and Class Diagram

In order to have a proper architecture for the classes that will be used to meet the requirements stated above, the following Domain Model has been drawn to understand the relationship between classes. These classes are generated using Umlple to have the model classes for the TreePLE System. The detailed code generated can be found in the zipped file in the repository Project-14.



## Statechart for Class Tree

Finite state machines are efficient to implement for the computer to manage the state of an object in the class. By responding to certain inputs, the finite state machines can change the state of the class. The following diagram is a representation for the state of a tree.

