

ECSE 321 – Iteration 1 Deliverables

PROJECT TEAM 9 – NADA MARAWAN, FOUAD BITAR, ALJULANDA AL-ABRI,
AHMED ELEHWANY, IMAD DODIN

Contents

Functional and non-functional system requirements	2
Actors and Use Cases	5
Use Case Descriptions.....	6
Use Case Description 1:	6
Successful Outcomes:	6
Main Success scenario:	6
Alternative Flows:.....	6
Use Case Description 2:	7
Successful Outcomes:	7
Main Success scenario:	7
Alternative Flows:.....	7
Use Case Description 3:	8
Successful Outcomes:	8
Main Success scenario:	8
Alternative Flows:.....	8
Use Case Description 4:	9
Successful Outcomes:	9
Main Success scenario:	9
Alternative Flows:.....	9
Use Case Description 5:	10
Successful Outcomes:	10
Main Success scenario:	10
Alternative Flows:.....	10
Domain Model.....	11
Requirements-Level Activity Diagram: Managing All Trees	11
Domain-Level State Chart for class “Tree”	12
Work Plan	13

Functional and non-functional system requirements

ID	R1
Description	The app shall allow users to register planted trees.
Traceability	UC-1
Kind	Functional

ID	R2
Description	The app shall allow users to register cut-down trees.
Traceability	UC-2
Kind	Functional

ID	R3
Description	The app shall allow users to mark trees with disease.
Traceability	UC-3
Kind	Functional

ID	R4
Description	The app shall allow users to mark trees to be cut down.
Traceability	UC-4
Kind	Functional

ID	R5
Description	The system shall load tree data from a text file.
Traceability	UC-5
Kind	Functional

ID	R6
Description	The system shall update tree data within the database.
Traceability	UC-6
Kind	Functional

ID	R7
Description	The system shall list all or a subset of the trees for display to the user
Traceability	UC-7
Kind	Functional

ID	R8
Description	The system shall plot all or a subset of the trees on a map for display to the user.
Traceability	UC-8
Kind	Functional

ID	R9
Description	The system shall calculate sustainability attributes.(biodiversity and emission reduction) and displays result to the user.
Traceability	UC-9
Kind	Functional

ID	R10
Description	The system shall provide to the user, forecasts for what-if scenarios.
Traceability	UC-10
Kind	Functional

ID	R11
Description	The system shall be able to validate a user's credentials and dictate whether a user is a citizen or arborist.
Traceability	-
Kind	Non-Functional

ID	R12
Description	The system shall be able to load data from text files in under 500ms.
Traceability	-
Kind	Non-Functional

ID	R13
Description	The system shall be able to write data to the database in under 1s.
Traceability	-
Kind	Non-Functional

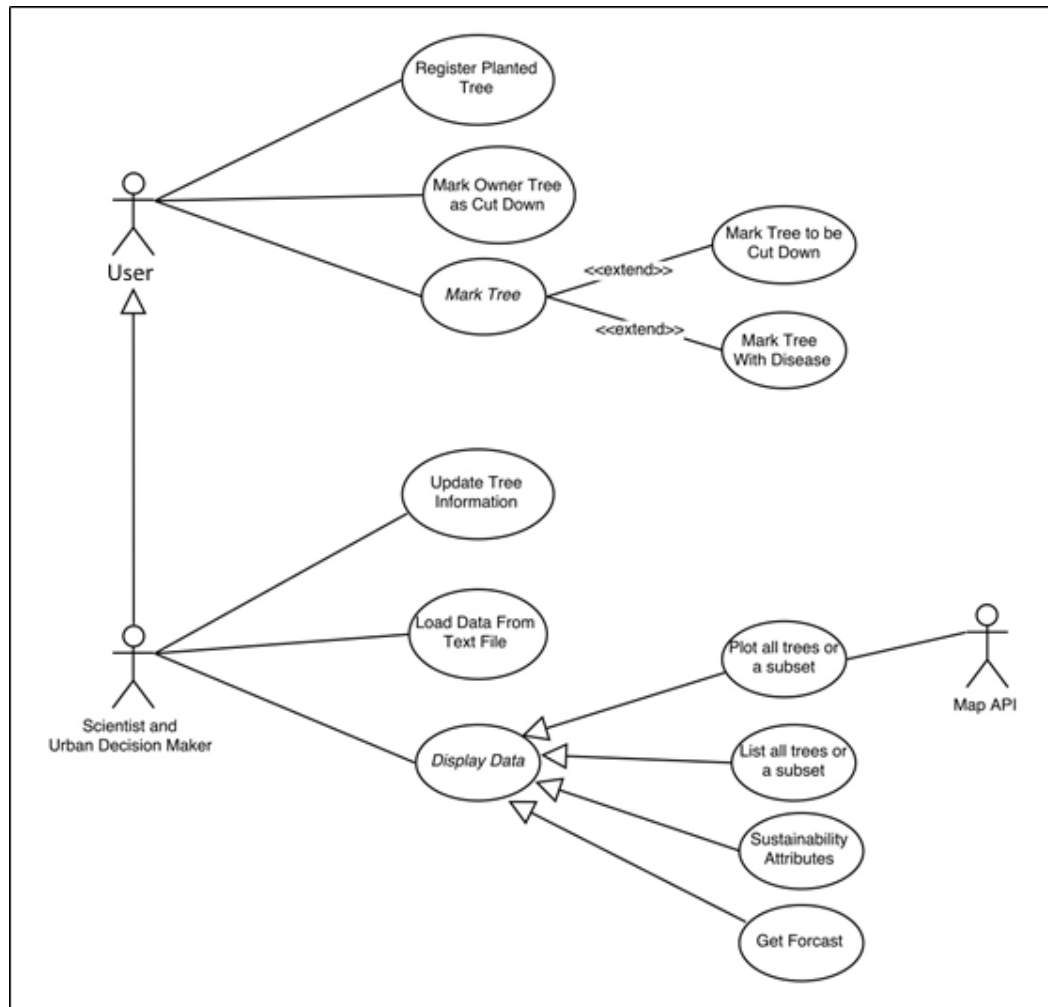
ID	R14
Description	The system shall be accessible with a success rate of above 90%.
Traceability	-
Kind	Non-Functional

ID	R15
Description	The system shall be able to retrieve and display an online map to the user in under 600ms.
Traceability	-
Kind	Non-Functional

ID	R16
Description	The system shall be able to generate and display tree markers on a map in under 2s.
Traceability	-
Kind	Non-Functional

ID	R17
Description	The system shall be able to allow multiple users to input data at any given instant of time.
Traceability	-
Kind	Non-Functional

Actors and Use Cases



Use Case Descriptions

Use Case Description 1:

Successful Outcomes:

Use Case Package	TreeApp
ID	UC-1
Use Case Goal	Primary Actor successfully reports planted tree.
Actor(s)	Users
Level	User-Goal
Precondition	Local launched the app
Domain Entities	Tree, Local/Specialist, Status, Municipality, LandUse, Species

Main Success scenario:

Steps	Action
1	Actor indicates intention to report planted tree
2	System/App prompts primary actor to input tree information
3	Primary Actor performs Create Tree Details
4	System generates a unique tree ID, assigns it to the tree and registers the tree with the information given into the system.
5	Use case ends successfully

Alternative Flows:

3a.Initialize Tree Details ends unsuccessfully

Step	Action	Notes
3a.1	Primary Actor enters information that causes Initialize Tree Details to fail.	
3a.2	System returns control to Main Success Scenario step 2.	

3b.Initialize Tree Details ends unsuccessfully

Step	Action	Notes
3b.1	Primary Actor enters information that causes Initialize Tree Details to fail.	
3b.2	User aborts.	
3b.3	Use case ends unsuccessfully.	

Use Case Description 2:

Successful Outcomes:

Use Case Package	TreeApp
ID	UC-2
Use Case Goal	Primary Actor successfully reports cut-down tree.
Actor(s)	Users
Level	User-Goal
Precondition	Local launched the app
Domain Entities	Tree, local/specialist, status,

Main Success scenario:

Steps	Action
1	Actor indicates intention to report cut-down tree.
2	System/App prompts primary actor to input tree ID or street Address.
3	System retrieves back tree information from the database and displays it for the user to make sure the system retrieved the right tree.
4	System asks user if he wants to confirm the status of the tree to "cut-down".
5	Use case ends successfully

Alternative Flows:

3a.Retrieving Tree ends unsuccessfully

Step	Action	Notes
3a.1	Primary Actor enters invalid tree ID that causes Retrieving Tree to fail.	
3a.2	System returns control to Main Success Scenario step 2.	

3b.Initialize Tree Details ends unsuccessfully

Step	Action	Notes
3b.1	Primary Actor enters invalid tree ID that causes Retrieving Tree to fail.	
3b.2	User aborts.	
3b.3	Use case ends unsuccessfully.	

Use Case Description 3:

Successful Outcomes:

Use Case Package	TreeApp
ID	UC-5
Use Case Goal	Primary Actor successfully loads data from a text file.
Actor(s)	Specialists
Level	User-Goal
Precondition	Local accessed the web front end successfully.
Domain Entities	Tree, Local/Specialist, Status, Municipality, LandUse, Species

Main Success scenario:

Steps	Action
1	Primary Actor indicates intention to create database by loading data from a file.
2	System/App prompts primary actor to input a file path.
3	System loads the text file, reads it and extracts the tree data.
4	System asks Primary Actor if he wants to create a database using the extracted data.
5	Use case ends successfully

Alternative Flows:

3a.Loading text file ends unsuccessfully

Step	Action	Notes
3a.1	Primary Actor enters invalid file path that causes loading text file to fail.	
3a.2	System returns control to Main Success Scenario step 2.	

3b.Reading text file ends unsuccessfully

Step	Action	Notes
3b.1	System's file reading function crashes while reading the file.	
3b.2	User aborts.	
3b.3	Use case ends unsuccessfully.	

Use Case Description 4:

Successful Outcomes:

TreeApp	TreeApp
ID	UC-8
Use Case Goal	Primary Actor successfully plots trees on a map.
Actor(s)	Specialists
Level	User-Goal
Precondition	System loaded the map successfully from an external API.
Domain Entities	Tree

Main Success scenario:

Steps	Action
1	Primary Actor indicates intention to plot trees on a map.
2	System prompts primary actor to choose between plotting all trees or specifying a subset to be plotted only.
3	System plots the trees on the loaded map using tree coordinates data from the database.
4	System asks Primary Actor if he wants to save the map.
5	Use case ends successfully

Alternative Flows:

3a.Loading map from API ends unsuccessfully

Step	Action	Notes
3a.1	System cannot load the map due to unsuccessful API calls.	
3a.2	System displays an error in loading map.	
3a.3	Use case ends unsuccessfully.	

3b.Reading text file ends unsuccessfully

Step	Action	Notes
3b.1	System's plotting function crashes due to invalid tree coordinate data.	
3b.2	System returns plotting error.	
3b.3	Use case ends unsuccessfully.	

Use Case Description 5:

Successful Outcomes:

Use Case Package	TreeApp
ID	UC-9
Use Case Goal	System successfully updates biodiversity index.
Actor(s)	Specialist
Level	User-Goal
Precondition	System has a database with updated data.
Domain Entities	Tree, Species

Main Success scenario:

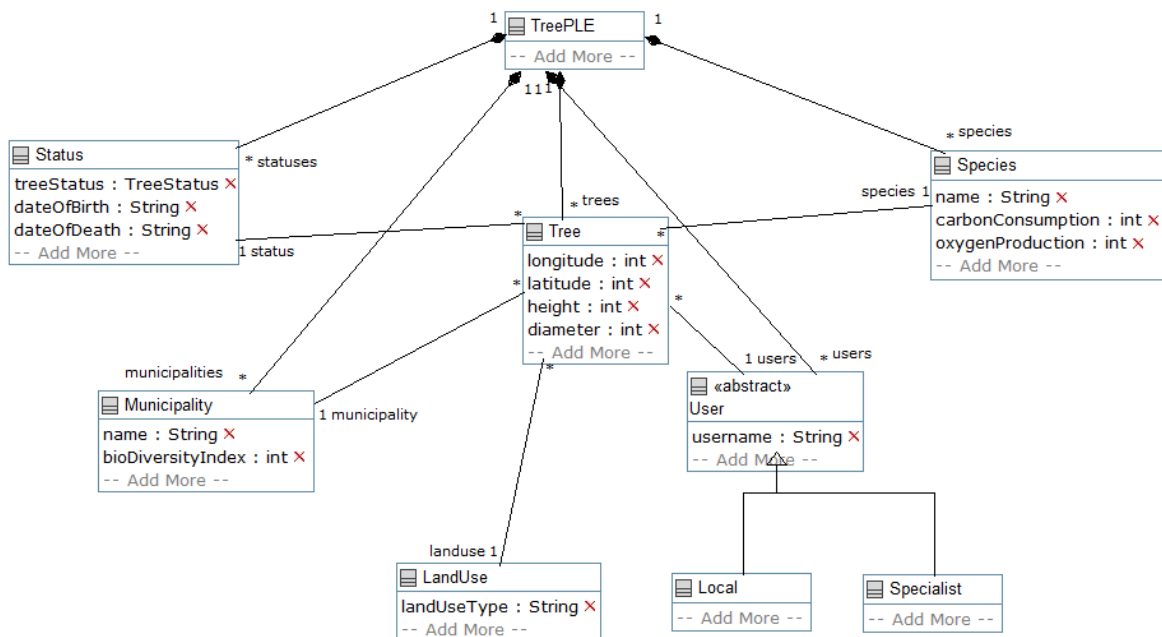
Steps	Action
1	Primary Actor indicates intention to calculate biodiversity index over an area.
2	System prompts primary actor to input the range of geographical coordinates.
3	System calculates a biodiversity index based on tree variation in the area.
4	Use case ends successfully

Alternative Flows:

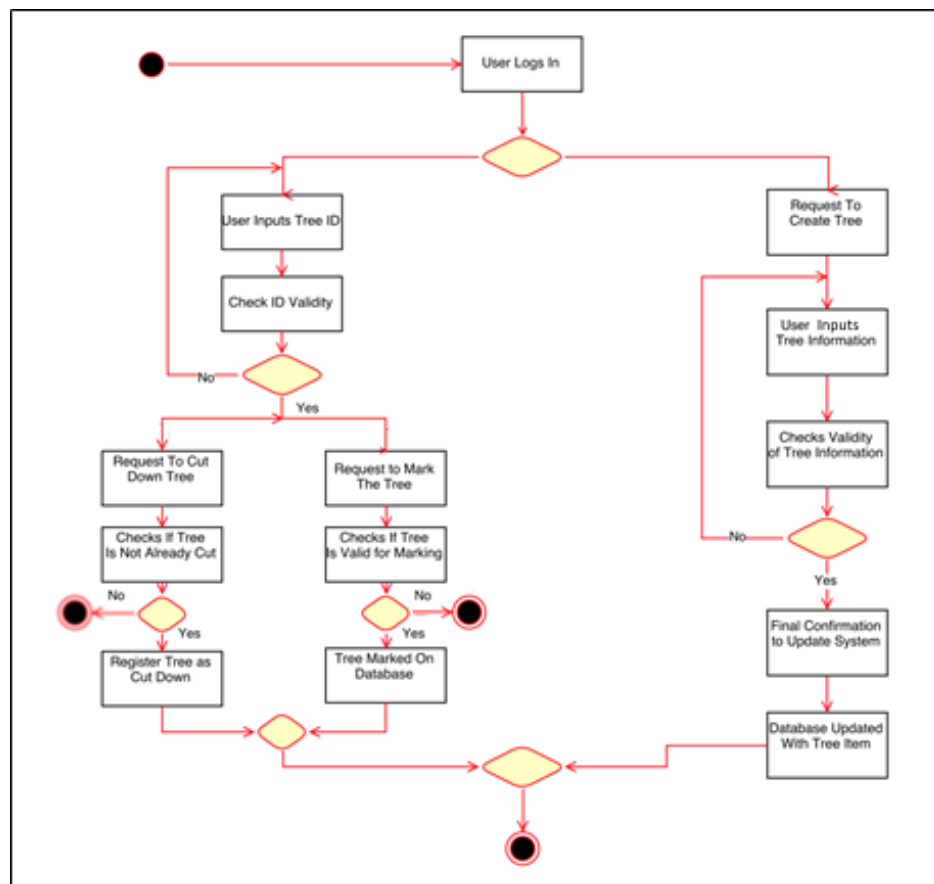
3a.Calculating index ends unsuccessfully

Step	Action	Notes
3a.1	System's calculating function crashes due to invalid geographical range input.	
3a.2	System returns control to Main Success Scenario step 2.	

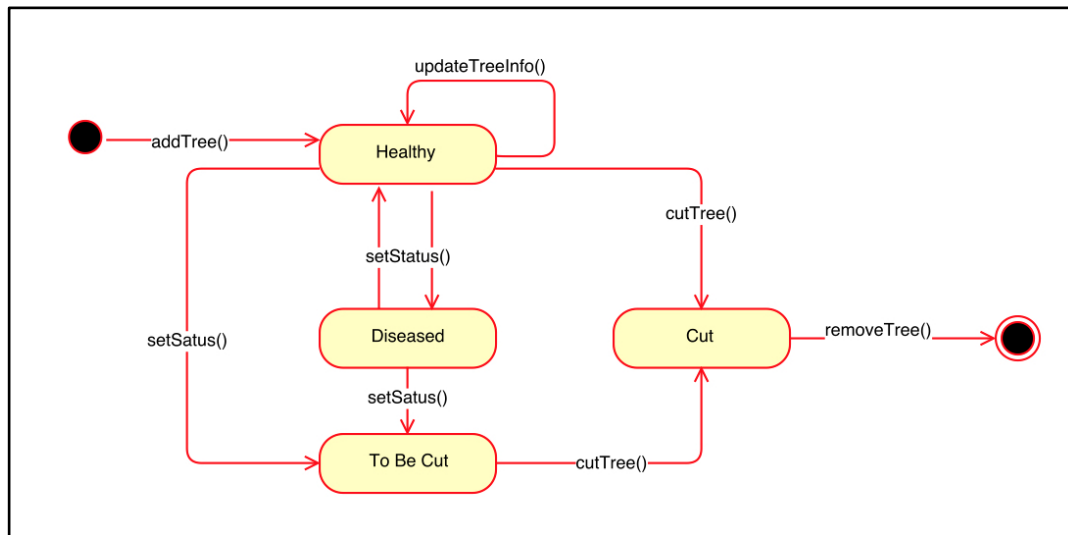
Domain Model



Requirements-Level Activity Diagram: Managing All Trees



Domain-Level State Chart for class “Tree”



Work Plan

Deliverable 2 – Design Specification and Prototype (8%) (due February 26)

5 1.5-hour meetings: 15, 18, 19, 25 and 26 of February. The iteration should be complete by the 25th of February. The 26th will be for final polishing.

Main goals:

- 1- Description of architecture and proposed solution. 2- Prototype for “plant tree” and “cut down” using Spring and Android. 3- Prototype for “list all trees” using Spring and web. 4- Sequence diagrams for “plant tree” and “List all trees”.

Use cases to be addressed:

- UC1: Primary actor successfully report planted tree. UC2: Primary actor successfully report cut-down tree. UC7: List all trees or a subset. UC8: plot all trees or a subset.

Deliverable 3 – Quality Assurance Plan (5%) (due March 14)

4 1.5-hour meetings: 1, 10, 11 and 14th of March. The iteration should be complete by the 11th of March.

The meeting on the 14th of March will be for finalizing minor details.

Main goals:

- Description of testing (unit, component and system)

Use cases to be addressed:

- All use cases.

Bonus:

- 1- Finalize forecasting model. 2- Implement “mark to be cut down” and “mark diseased” using Android and Spring. 3- Implement data loading from the xml files.

Deliverable 4 – Release Pipeline Plan (5%) (due March 25)

3 1.5-hour meetings: 17, 18 and 25th of March. The iteration should be complete by the 25th of March.

Main goals:

- Release pipeline.

Use cases to be addressed:

- No specific use cases.

Bonus:

- 1- Implement the forecasting model and sustainability calculations. 2- Implement “update tree data” within the database. 3- Implement “locate all trees” (or subset) and “locate a single tree” on the map.

Deliverable 5 – Presentation (8%) (scheduled on April 03)

3 1.5-hour meetings: 31/3, 1 and 2nd of April. The presentation should be ready by the 1st of April. The 2nd of April will be for practicing.

Main goals:

- prepare and practice the presentation.

Use cases to be addressed:

- No specific use cases.

Deliverable 6 – Final Application (12%) (due April 11)

5 1.5-hour meetings: 5, 6, 7, 8 and 9th of April. The full implementation should be ready by the 8th of April. The 9th of April will be for code revision.

Main goals:

- 1- Full implementation of code. 2- Peer revision of code. 3- Final round of testing. 4- If time permits: additional features.

Use cases to be addressed:

- All use cases.

MAIN TASKS AND LEADERSHIP ROLES. ITERATION 1:

- Ahmed: Use Case descriptions and Activity diagram.
- Fouad: State chart and Actor and Use cases diagram.
- Imad: Domain Model
- Nada: Functional and non-functional system requirements.
- Aljulanda: Requirements and work plan.

Each team member worked at least 6 hours (3 work sessions 2 hours each) during this iteration.

Key Decision:

- 1- Architecture of persistence layer. 2- Additional features to be implemented. 3- Sustainability attributes.

MEETINGS LOG. ITERATION 1:

- 7/2/2018. Discussed the project in general. Assigned tasks. Finished Uml model and requirements.
- 10/2/2018. Worked on use case diagrams, traceability and specifications. Finished requirement level activity diagram.
- 11/2/2018. Finalized domain model. Worked on statechart of Tree class.