WESTERN UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SE 3352A – SOFTWARE REQUIREMENTS AND ANALYSIS

Assignment 2: Developing the Entity Class Diagram for Self-Start

Due Date: December 4, 2017

In assignment1, you developed the SRS for Self-Start system in which the main functionalities of Self-Start are modeled using UML use cases. In this assignment, you will refine the functional model and derive the analysis object model. The analysis object model focuses on the individual concepts that are manipulated by the system, their properties and their relationships.

Once you get familiar with the requirements of this assignment, update the Responsibility Matrix file exist in your GitHub account to reflect the new works distributions among your team members.

Task 0

1. Each team member will work on his/her assigned tasks as follows:
    a. Create a new branch for the local repo to be named by the student's '*uwoID*-assignment2'.
    b. All works in editing the workbook document sections(s), sub-sections(s), creating diagrams, or editing the Excel sheets need to be committed to this branch.
    c. When all tasks are finished, publish this branch to the GitHub.
    d. Create pull request to the GitHub.
2. Only one team member will ***MANUALLY*** merge all pull requests (don't use automatic merge) from all branches and will make sure that the all documents are complete.
3. Do not delete any of the created branches after the merge process.

Task 1

1. Consider the vision document of Self-Start and the UC diagram you provided in assignment1. Examine the use case description for each UC, then identify the Self-Start system entity classes.
2. Using the workbook document (provided in GitHub account) add the following table in the section titled "Potential Entity Classes".
   For each identify entity class complete the following table in which you need to write the class name and brief description for each class.

Class name	Brief Description

3. Note that, entity classes represent the persistent information tracked by the system. The following heuristics can be used to determine and identify entity classes:
    - Is it a container for data?
    - Does it have separate attributes that will take on different values?

- Would it have many instance objects?
- Is it in the scope of the application domain?

Task 2

1. Based on your understanding of the Self-Start system requirements, you need now to identify a set of attributes for each entity class you have in the previous table.
2. In your workbook section titled 'Classes Attributes", record the attributes for each class using the following table.

Attribute name	Attribute Type	Brief Description

3. Specify the type of each attribute, as appropriate. Stick to a simple set of types for your attributes, such as the Java primitives (int, float, char, boolean) and simple classes (e.g. String). Avoid recording attributes that can be derived from other attributes (for example, a Circle has a radius and a diameter, but we could pick either as an attribute and the other would be derived).

Task 3

Consider all the classes and their attributes you have identified in Task 1 and Task2 above and draw the UML class diagram in your workbook section titled "Entity Class Diagram". You diagram should include the following:

1. The relationships between pairs of entity classes that seem to be strongly connected. You may use inheritance, association, aggregation, or composition.
2. If there is a candidate for inheritance, consider introducing abstract or interface classes as necessary (for elegance and for future extension).
3. If you find a class that is associated with an existing relationship between two other classes, consider making it an association class.
4. For every relationship, except inheritance, you should now consider adding some kind of description:
   - An association name and direction, describing what the association represents.
   - An association end names (role name), describing the source and/or target end names.
5. Specify the multiplicity of objects involved in each class's relationship (except inheritance, which is implicitly one-to-one).

Task 4

Develop two traceability matrices, the former is to link the Self-Start system features to the use cases you identified in assignment1, and the later is to link these use cases to the entity classes you identified in the above tasks. Use the MS Excel sheets provided in GitHub to complete this task.

Hand In

1. After finishing all tasks, download the final version of (1) the workbook document, (2) FEAT-UC matrix, (3) UC-EntityClass matrix, and (4) Responsibility matrix from your account in GitHub.
2. Use OWL to submit these files by the due date mentioned above.
3. Only one submission is need per each team.