COEN 160/275 OO Analysis, Design and Programming Winter2014

Project (Groups of two)

Total points (250) with an EC of up to 50pts

EcoRe is a company that wants to promote and market products for recycling of material. The company wants to design and manufacture, an ***EcoRecycle system***, consisting of *recycling machines* and *recycling stations* which monitor the recycling machines.

You are required to help *EcoRe* design and implement the Recycling machines and Recycling Stations. As a part of that requirement, you are given the job of creating a simulation of how the Recycling machines and Stations typically work where they are installed.

Given below is the description of the functionality of the different components in this system.

1.0 A Recycling machine (**RCM**) is similar to a vending machine and is designed to accept recyclable items that are aluminum and glass products where a user is paid a small amount of money for each type of item. The recycling machine is equipped with an interface to display the items accepted by the machine, the amount paid for each item and slot(s) to accept the items and return the money to the user. These machines are installed in offices, schools, hospitals and large buildings. Each **Recycling Machine** has a machine id, location, a list of items that it can accept, price paid for each item, capacity (in weight), total weight of items currently in the machine, last time the machine is emptied of items.

1.1A **Recycling Monitoring Station** (**RMOS**) is a software program that runs on a workstation that is connected to monitor typically ten different recycling machines within a radius of one mile. It is used to activate each recycling machine in the group (it monitors) to accept items and keep track of the status of each individual recycling machine. The status includes whether the machine is in operational status (operational or down), current weight of recycled items in the machine, amount of money in the machine, time last emptied. RMOS updates the capabilities of the RCMs – this may include changing or adding new types of recyclable items and changing the price. RMOS collects statistical information about usage of the RCMs it monitors. This may include the no. of times the machine was emptied in a specific duration (in no. of hours), weight of items collected in a specific duration, no. of items collected by type (aluminum, glass and so on).

You are required to design and implement a simulation of a group of Recycling machines and a recycling Station that monitors them, using OO analysis and design techniques and implementation in Java.

2.0Your simulation should demonstrate the following:

Provide at least two **Recycling machines (RCM)** and one **Recycling Monitoring Station (RMOS)** with Graphical User Interfaces and functionality as described below.

2.1 **A Recycling Monitoring Station (RMOS)** with a graphical user interface to perform its tasks, by an administrator. Please note that each RMOS monitors a group of RCMs (in your demo, you should have at least 2 RCMS in the group). In the discussion below, we will refer to this group as **RCM group**.

The **GUI interface** should provide the following functionality:

- A graphical interface, where each of the individual machines in the group (monitored by this workstation) can be represented. Each machine is identified by its location and an id. Each machine should store the time of the last removal of its items.
- Add a new machine to the group.
- Remove a machine from the group. (Not required for single person teams)
- Check the operational status of each machine.
- Change/add new types of recyclable items. (Not required for single person teams)
- Change the price of an item. (Not required for single person teams)
- Check the amount of money in a specific RCM.
- Check the current (and available) capacity (by weight or volume) of an RCM. This indicates whether an RCM is full and has to be emptied.
- Get number of items returned by a specific machine (or all the machines in the group) in a month.
- Show the time a specific RCM was emptied last time.
- Get the location of recycling station and id of the machine that was used the most (in the last n days) (Not required for single person teams)
- Display the usage statistics for each RCM in the RCM group. This will include the total weight of recycled items by machine (per day, week ..), total value (cash or coupons) issued, using a visualization (a graph for

example); show the number of times a particular RCM was emptied in a specific duration.

A typical scenario for using a RMOS by an administrator is

- The administrator logs in with a username.
- Adds a recycling station to the group to be monitored.
- Sets (change) the list of recyclable items for the recycling machines. This may include adding new item types, removing item types, changing the price and so on.
- Activates the recycling station in the group to accept items.
- Displays the statistics of usage for the RCM group as described above.

2.2 A Recycling Machine (RCM) with a graphical user interface to perform its tasks. The GUI interface should provide the following functionality:

- Display the list of recyclable item types and the price paid for each of the item types, by weight. *Note*: As a developer, choose and set a price for each of the item types, by their weight. For example, $1.00 for 2lbs of glass.
- Allow the user to start inserting the item in the designated receptacle.
- Does not allow an item that is not permitted and shown as a recyclable item by the RCM.
- Does not allow an item if RCM is full (capacity to hold the items has been reached).
- Displays the item type, weight (in lbs) and price due to the customer.

- A customer (user) may choose to view the weight in Metric units. (Not required for single person teams)
- Returns the money cash if there is enough in the machine. Otherwise, prints a coupon that can be redeemed in a number of designated stores. The user may select to have a coupon instead of cash.

A typical scenario for using a RCM is

- The user drops the item to be recycled in the designated receptacle.
- The user interface shows the type of item, weight of item and money given to the user. If there is no money in the machine, a coupon is given to the user; the coupon is redeemable in the designated stores.
- When the user has many items to recycle, they can indicate the start of the session (with a button click, for example, drop the items and indicate the end of a session (with a button click, for example). Money/coupon will be returned for the value of all items returned within the session.
- A user may indicate the value for the returned items to be given in coupons instead of cash.
- A user may try to recycle an item which is not designated as acceptable (electronics for example) at the machine; in that case, the machine should display a message and not accept the item. (Not required for single person teams)

Note:

You are free to add any enhancements (and make reasonable assumptions) of your choice to the system described above.

An extra credit of **up to 50 pts** is possible for creativity and usability of the system.

Design and implement (in Java) the ***EcoRecycle* system** for *EcoRe* using **OO analysis and Design Techniques and Java**.

Deliverables:

Part 1: The Analysis and Design documents (100 pts)

The Analysis and Design document should include the following:

1. Use-Cases to illustrate the functionality of the system (20 pts) **(Deliverable 1)**
2. CRC cards to show a list of the most important classes and a brief description of their responsibilities. (15 pts) **(Deliverable 1)**
3. Class Diagrams. (25 pts) **(Deliverable 2)**
4. State Transitions (if any) and Interaction Diagrams for the Usage scenarios. (25 pts) **(Deliverable 2)**
5. The Logical partitioning of the system into packages (15 pts) **(Deliverable 2)**

NOTE: UML notation is to be used.

Part 2 (Demos will be during the 10th week) (150 pts)

An implementation with a final demo of the project is required. (100 pts) **(Deliverable 3)**

Source Code (Java) (50 pts) **(Deliverable 3)**

Due Dates

Deliverable 1: 18th Feb

Deliverable 2: 7th March

Deliverable 3: Week 10