CS3354 – Fall 2017 – Assignment 3

Due date: Thursday, Oct 12 at 11:55 p.m.

Project Title: Shipping Store (stage 3)

Goal: In this assignment, student will analyze the requirements of a shipping store management software, and come up with an appropriate UML design, which can be later translated into code. This version of the software will have added functionality compared to the previous assignment. More specifically, the software should allow five following operations, see details below:

1. Inventory Management
2. Transaction Management
3. Accounting Management
4. User Management
5. Shipping Store Web site

1. Inventory Management

The software should maintain information about the packages currently in the inventory of the shipping store. A package is in the inventory if it has been shipped but not yet delivered to its final destination. The package types and required fields for specific types are the same as defined in Assignment 2. All packages also have an extra field, tracking location. Tracking location provides the address where the last scan of the package happened, and it is a string type.

2. Transaction Management

A shipping transaction can be created either by a customer through the store's website (see below) or by an employee. A transaction can be in three different states:
1. *Pending*: when the shipping transaction record has been created but the package has not been dropped off to the store yet, and it has not been added to the inventory.
2. *Active*: when the package has been dropped off to the store and it has been added to its inventory.
3. *Completed*: when the package has been delivered to is final destination and has been removed from the inventory.

For each transaction, the system keeps the following information:
- Customer id: who shipped the package
- Package Tracking Number
- Cost of shipping
- Employee ID: employee who collected the package (empty if package has not been collected yet)
- Date shipping transaction was created
- Date of delivery: empty if the package has not yet been delivered
- Employee ID: employee who delivered the package (empty if the package has not yet been delivered)
- Status: Pending, Active, Completed

When the employee collects the package from the customer, the status of the transaction is set to Active and the package is added to the inventory. When an employee delivers a package to its final destination, the package is removed from the inventory and the status of the transaction is set to Completed. In other words, the package should exist in the inventory only when the shipping transaction status is Active.

3. Accounting Management

The accounting management part of the software, allows the software administrators to keep track of the income, expenses and profit, and print Financial Documents like Balance Sheets, Statements of Cash Flow and Invoices. To keep things simple, we will assume that all the income comes from package shipping transaction sales and all the expenses come from employee salaries and space rental. Every time a new package is shipped or an expense is paid the accounting must be updated and optionally a financial document can be printed.

4. User Management

The system now supports three types of users: Customers, Employees and Administrators. For all user types, the system keeps some basic information, like their Name, Phone Number and Email. In addition, for customers, the system keeps record of their address. For employees, the system keeps records of their SSN, Monthly Salary, their Bank Account Number, a flag indicating if there are Active or Inactive and their PIN number. Past employees remain in the system but are considered inactive. Finally, the administrators are special types of employees with some extra system access rights. For example, an administrator can add/edit an employee or another administrator to the system and can access the accounting management part of the software, from where they can pay expenses and print financial documents at any time. The simple employees, can only create shipping transactions, collect and deliver packages, add/edit customer info to the system, and reply to customer messages.

5. Shipping Store Website

The shipping store has a website which allows customers to create an account (and edit it), create shipping transactions, track packages, and contact the shipping store through a web form about possible questions that they may have. When a customer creates a shipping transaction through the website, initially the Employee ID is left empty (since there is no employee involved yet) and the transaction status is set to Pending. Later, when the package is dropped off to the store, the employee id is populated with the id of the employee who collected the package, and the transaction status is set to Active. Note that a shipping transaction can be created either by a customer through the website or by an employee at the store. The customer (and/or the employee) can also optionally print a receipt and/or a shipping label when they create the shipping transaction.

The website also offers the option to customers to contact the store by sending a message through a web form. To send a message, the customers need to have first created an account. The message is added to a queue of unanswered messages. Subsequently, an employee (whoever sees the message first) can respond to the customer via email. The response of the employee is also recorded, along with the Employee ID. Each answer is associated with one customer question and is added to a pool of answered messages.

Tasks:

Assume that the above are the requirements of the system, as described by the owner of the Shipping Store.

You are asked to do the following:

1. [30 points] Come up with a set of Use Cases that best capture the requirements as described above. Use free text to describe flows of events for each individual use case. Create a Use Case Diagram that shows the interaction between actors and uses cases, as well as the relationships among use cases (communication, inclusion, extension, inheritance).
2. [30 points] Derive and show a set of Class-Responsibility-Collaboration (CRC) cards that show the classes that you have identified, as well as their responsibilities and collaborators. Based on your CRC cards, create a Class Diagram that shows the relationships between classes (dependences, inheritance, aggregations, compositions, multiplicities, etc.). In each class of the class diagram, you may show the most important fields and methods.
3. [20 points] Draw a Sequence Diagram for the sequence of operations taking place when an employee creates a new shipping transaction and collects the package from the customer. You can assume the customer already has an account in the system. You should come up with appropriate method names, and specific method calls to complete the task.
4. [20 points] Assume a class named “Transactions” which represents shipping transactions. Create a State Machine Diagram that shows the possible states of objects of that class and the transitions based on different events from the moment a shipping transaction is first created by a customer through the Website, until the moment the package has been delivered to its final destination.

Logistics:

Submit your solutions as a single MS Word or PDF document. To create your diagrams, you can use any UML editor of your preference, which is compatible with the notation used in class (StarUML is recommended). You may export the diagrams as images and add them to your document, or simply take screenshots of your UML editor and crop them appropriately.

There is no single correct solution. All solutions that are reasonable, well documented and follow the standards that we saw in class, will be accepted. If you are unsure about certain decisions and need to make assumptions, please state your assumptions clearly in your solution document.

This assignment can be done either individually or with a partner. If you are working with a partner, only one needs to submit, and specify both names in the submitted document.

Submit an electronic copy only, using the Assignments tool on the TRACS website for this class.