Deliverable 1

UML Assignment – Ontario Boat Rentals

## Evan Ling

[eling1@u.brockport.edu](mailto:eling1@u.brockport.edu)

CIS 427

Project Management with Practice

### Table of Contents

Introduction

Deliverable 1

UML Assignment

## Evan Ling

### Introduction

This deliverable is to demonstrate proficiency with the creation of various UML diagrams with the open-sourced software Open Modelsphere while following documentation standards. Ontario Boat Rentals’ system is used as an example.

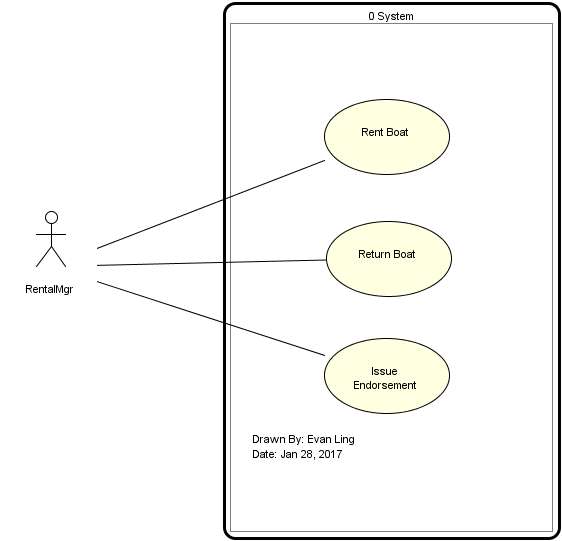
### Ontario Boat Rentals

SUNY has a contract to develop an information system for Ontario Boat Rentals (OBR). The initial description of how the system should work is as follows. Sailors can rent different sailboats from Ontario Boat Rentals (OBR). When a sailor has seen the available boats and made a decision about which to rent, a rental agreement, or contract, is produced and signed. When the boat is returned, it is checked back in and the customer contract completed.

In order to rent a boat, a sailor must have a sailing license and payment for that boat type on file with OBR. When a boat is to b erented the sailing license is checked to verify the sailor is capable of sailing the desired boat. If there is no license endorsement for the desired boat on file the customer must be checked out on the boat (tested) and a license issued, if a new customer, or endorsed, if an existing customer. The rental manager handles the rental and return process, testing, and licensing, and all the paperwork.

### Use Cases

There are three use cases involved in renting the boats: the actual act of renting the boat, returning the rented boat and issuing an endorsement. These use cases are outlined in Figure 1.



**Figure 1 - Use Case Diagram**

# *Rent Boat*

When a sailor hs selected a boat, the sailor’s sailing license is checked to verify the sailor can sail the chosen boat. The rental manager then produces a contract.

# Return Boat

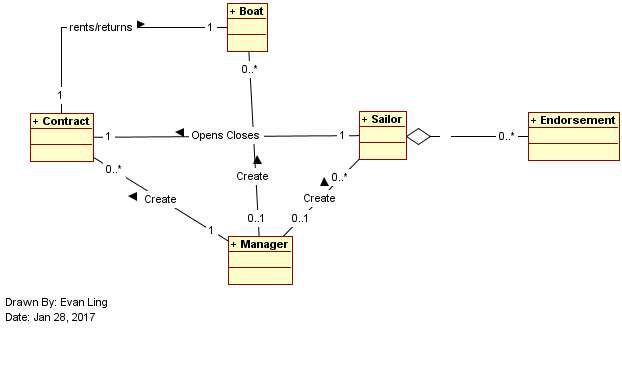
When a boat is returned, the rental manager notes the boats return and closes the contract

# Issues Endorsement

If there is no license endorsement for the desired boat on file the customer must be checked out on the boat (tested). If a new customer an endorsement is issued and customer information recorded. If an existing customer an endorsement is issued.

### Class Diagram

Analysis of the Ontario Boat Rental process and the use cases leads to a need for five classes. These classes hold and process the necessary data to track what boats are rented at any time and the qualifications of renters. Figure 2 shows the classes and the relationships between the class at the requirements level.



**Figure 2 - Class Diagram**

The boat class maintains data on the individual boats. There is an instance, an object of this class for each boat. This class/object maintains the boat identification details (number and type) and the rental status of the boat. When new boats come into service the manager creates new boat objects (this capability is outside the scope of the current project).

The sailor class maintains data on boat renters. There is an instance of this class for every sailor that is authorized to rent a boat from Ontario Boat Rentals. The sailor objects maintain the name of the sailor and have a relationship with the sailor’s endorsement objects.

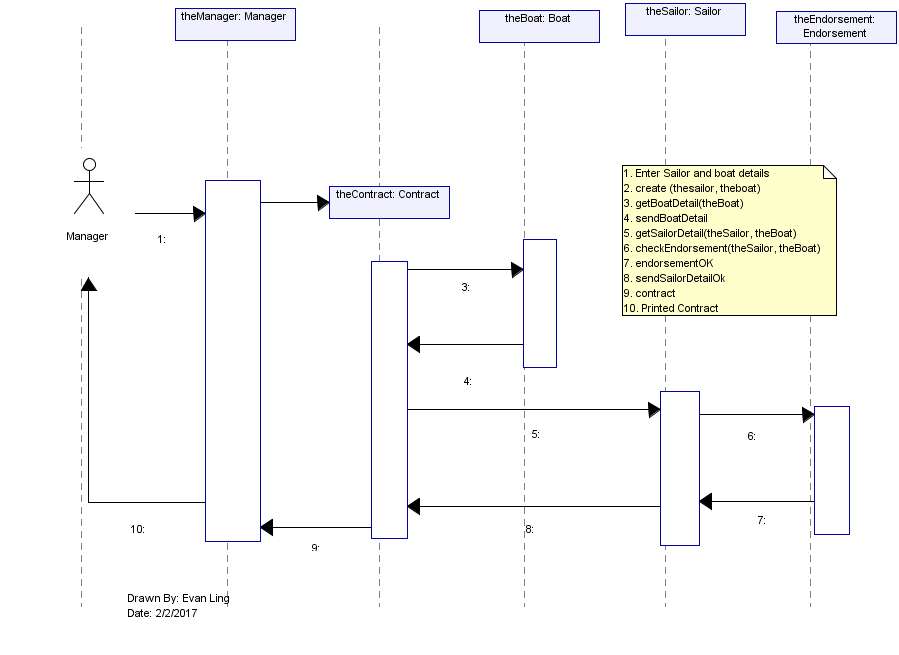
The instances of the endorsement class are in an aggregation relationship with their associated sailors. This class maintains the boat types that a sailor is qualified to rent.

Instances of the contract class is created when a sailor decides to rent a particular boat that he is qualified to rent. These contract instances remain until the sailor returns the boat.

The manager class is the information system’s representation of the user. There is a single instance of this class. It does not maintain any specific information, rather it interacts with the other classes to approve and create new rental contracts, initiate the creation of new instances of sailors and endorsements, and in the future create new boats.

### Sequence Diagrams

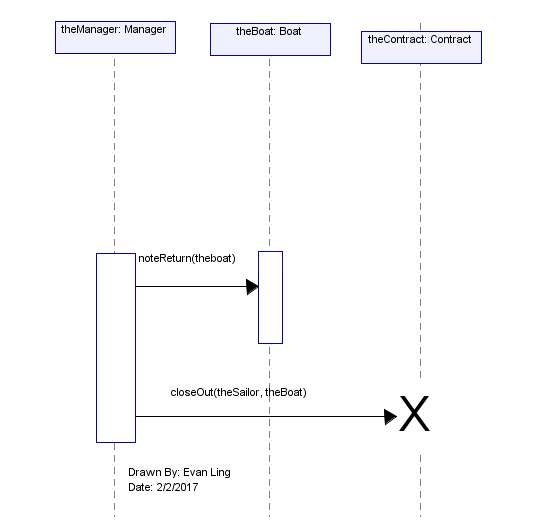
The sequence diagram shows the passage f=of messages between object and the order in which the messages are passed. In Open Modelsphere in addition to the classes/objects that take place in a sequence, the actor or external entity that initiates the sequence or receives the results of the sequence is also show. This demonstration of UML includes four requirements level sequence diagrams as they apply to Ontario Boat Rentals. The four sequences diagrams show all the workings of the information system needed to meet the requirements of the three use cases. The most complex sequence diagram is for renting a boat (Figure 3).



**Figure 3 - Sequence Diagram (Rent Boat)**

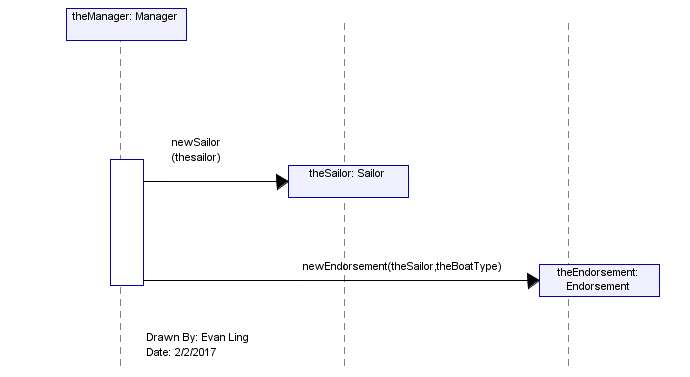
In the process (sequence) of renting a boat, the manager-user enters the sailor’s name and the number of the boat to be rented into the system, specifically to the manager object. The manager object then creates a contract object then asks the specific boat object if they are available to be rented. After receiving a positive reply and boat number the manager object queries the sailor object and its endoresements to ensure the sailor is qualified to rent that type of boat. Receiving a positive reply the contract is completed with sailor’s name and the boat’s number. The contract is passed to the manager object, which then prints a hardcopy for the user manager.

After the renting sailor is finished with the boat they return it to the user manager. The user manager finds the boat and the sailor’s contract, and through the manager object notes that the boat is again available to rent and closes-out the contract. The contract object is deleted. The sequence diagram is Figure 4. (X indicates the deletion of an object.)



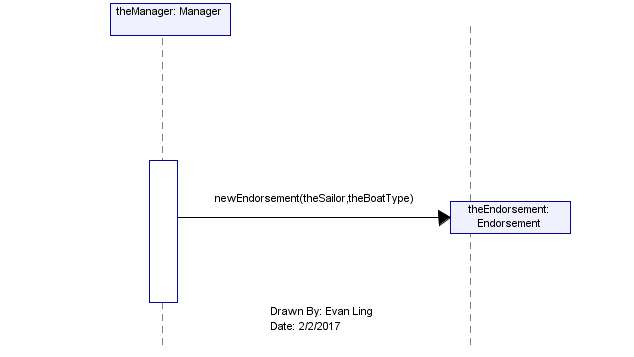
**Figure 4 - Sequence Diagram (Return Boat)**

New Sailors are coming to Ontario Boat Rentals to rent boats all the time. The information system must be able to record these new sailors and the types of boats they are authorized to rent. This process involves the manager conducting a test with the new sailor to decide on the sailor’s abilities. Of course this is a physical action of going for a ride in the boat to get checked out. After the manager decides what types of boats the sailor will be allowed to rent, this data must be recorded. Figure 5 shows the sequence of actions to record and endorse a new sailor.



**Figure 5 - Sequence Diagram (New Sailor)**

Sometimes an old sailor, one that has been renting boats, becomes qualified on a new type of boat. In this case, the sailor is already in the information system, only his endoresements need to be updated so he can rent this new type of boat. Figure 6 shows the sequence of actions in the information to add an endorsement to an existing sailor.



**Figure 6 - Sequence Diagram (New Endorsement)**

### Conclusion

While this report is a good practice in getting familiar with the format of deliverables, the creation of diagrams was a straining process not because of the requirements itself, but because of roadblocks that have occurred during construction. Open Modelsphere is a poorly documented program that suffers from a very unintuitive user interface – I had to spend an hour simply to find out how to change the size of the sequence diagram, and another hour to figure out how to delete role lines.

Given the plethora of options for design programs available currently, (some enterprise-level applications free for student use) there really isn’t any reason for us to use such an antiquated piece of software. Going forward, I would strongly suggest the use of another UML diagramming program for the benefit of my fellow colleagues and I.