SYS466 Winter 2019

**Lab 7 (2% of Final Grade): Object level Sequence Diagrams –advanced topics**

**Objectives:**

* Create sequence diagrams for generalized and specialized objects
* Create sequence diagrams for objects in a composition relationship
* Use service controllers to show interfaces with subsystems

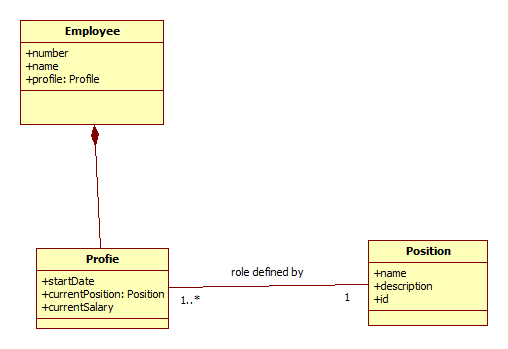
This is an individual lab and must be done in the lab room itself. **INDIVIDUAL SUBMISSIONS ONLY. PLEASE DO NOT USE INTERNET EXPLORER TO ACCESS THIS LAB – USE FIREFOX OR CHROME.**

**Submission and Credit:**

* Only those students who sign in will get credit for the lab. Submission is to be done via Blackboard. Email submissions WILL NOT be accepted.
* **Name your submission name\_Lab7.uml, for example JenSmith\_Lab7.uml**

**Special Instructions For This Lab:**

* You will be working in the **Company and Plowing models of the uml file you have been given**. In that model you will find diagrams and empty sequence diagrams. Please use these for your answers. Do not create any new diagrams
  + The class diagram named Controllers will contain the 4 controller classes you need. It is in the Company model but you can use it in the Plowing model also



**Exercise 1: Generate Employee List**

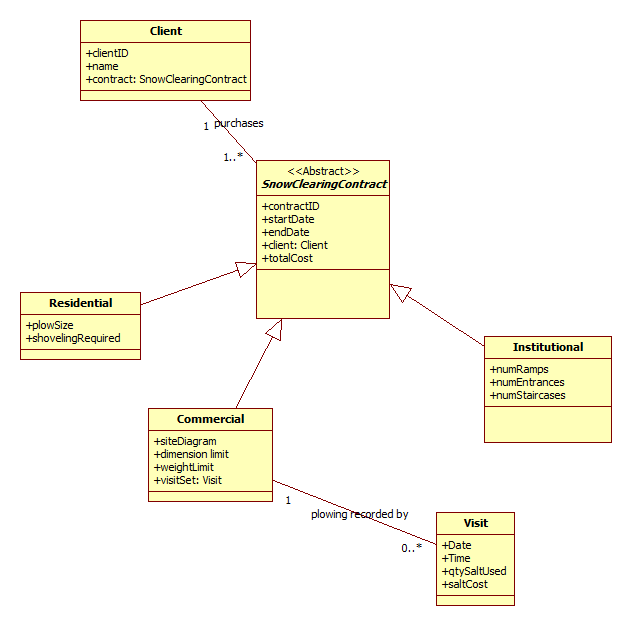
**Put into Employee model into SD titled Generate Employee List**

|  |  |
| --- | --- |
| **Manager** | **System** |
| Requests employee report. | Generates a report showing all employees (name and number of each employee) and the position of each employee (position name). |

**Exercise 2: Update Employee Position**

**Put into Employee model into SD titled Update Employee Position**

|  |  |
| --- | --- |
| **Actor – Manager** | **System** |
| Enters employee number. | Displays employee name and employee position name |
| Requests to see all positions | Displays all position names |
| Chooses a position and requests to replace the employee’s position with the new position. | Changes the employee’s position to the new position.  *(Removes old position from employee Adds the new position to employee)*  Saves the changes to the DB. |



**Exercise 3: Report on Contract Value**

**Put into Plowing model into SD titled Report on Contract Value**

|  |  |
| --- | --- |
| **Actor – Manager** | **System** |
| Enters a date | Gets total value (sums total cost) of all contracts with start date greater than entered date. |

**Exercise 4: Report on Salt**

**Put into Plowing model into SD titled Report on Salt**

|  |  |
| --- | --- |
| Actor – Manager | System |
| Enters commercial contract ID | Returns value of total salt used so far in that contract – quantity and $ value. |

**Exercise 5: Create Commercial Contract**

**Put into Plowing model into SD titled Create Commercial Contract**

|  |  |
| --- | --- |
| Actor – Manager | System |
| Enters client ID | Shows client name |
| Enters start date, end date, total cost and requests to create Commercial contract | Creates contract and displays the contract so far. |
| Adds site diagram, weight limit, dimension limit | Add site diagram, weight limit, dimension limit to contract. Displays contract so far. |
| Checks everything, and indicates to go ahead with contract. | Saves contract to the DB.  Sends contract to payment system for payment processing |

***Hint: don’t forget reference attributes***