

SYS466 Fall 2018

Professor: Barb Czegel

Lab 5 (1% of Final Grade): Introduction to Object level Sequence Diagrams

Due: At the end of the lab session

Objectives:

- Use a sequence diagram to document the object interaction involved in carrying out a scenario
- Use controllers to show interaction of layers
- Use proper UML syntax for sequence diagrams.
- Learn these additional constructs: loop, guard conditions, message to self

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:

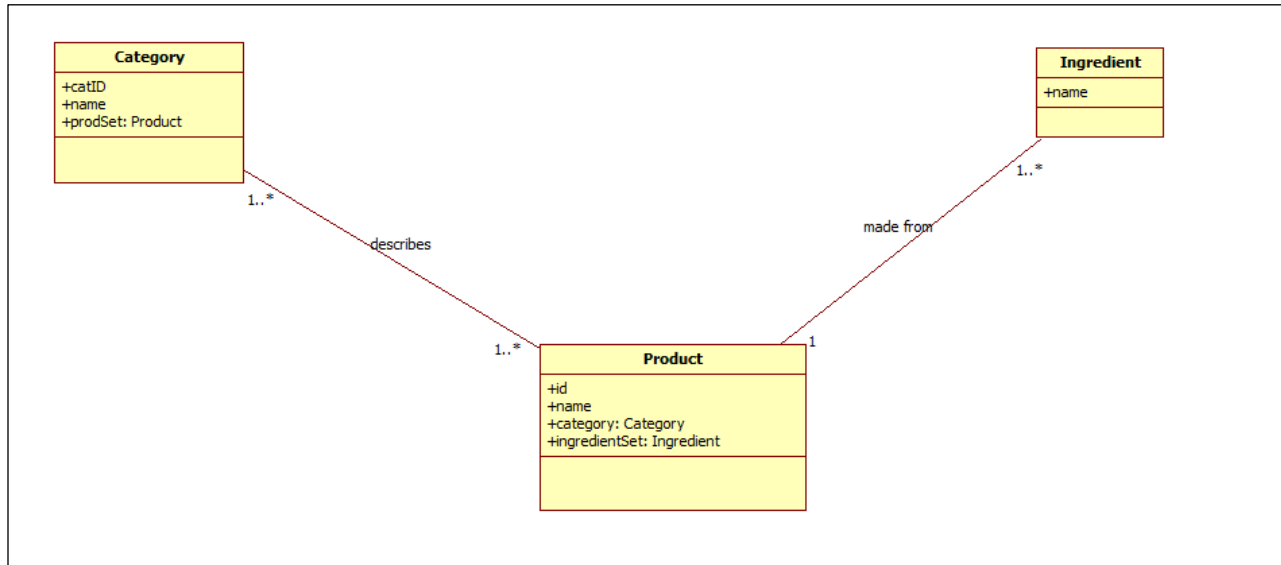
- Only those students who sign in will get credit for the lab. Submission is to be done via Blackboard. Email submissions WILL NOT be accepted.
- **Name your submission name_Lab5.uml, for example JohnSmith_Lab5.uml**

Special Instructions For This Lab:

- You will be working in the **Bakeshop Model of the uml file you have been given**. In that model you will find three class diagrams and 3 empty sequence diagrams. Please use these for your answers. Do not create any new diagrams
 - The class diagram named Controllers will contain the 3 controller classes you need.
- You will create each SD by pulling in (from the right sidebar in model explorer view) an actor, the 3 controllers, and the objects from the classes you need. The Entity Manager will be on the far right. In this lab we do not use it but it is excellent practice to have it there.

Exercise 1:

For this exercise you are creating a sequence diagram – please put it into the sequence diagram titled ViewIngredients in the model you have been given. To create the sequence diagram you will use the Ingredients class diagram and the View Ingredients scenario.



Scenario: View Ingredients

Precondition – a list of products has already been retrieved from the database and is displayed on the screen. Ingredients and category information are not displayed but have been retrieved from the database.

Actor – Baker	System
Selects a product	Displays the names of the ingredients for that product as well as product category name.

Hints and Notes:

- You will have to loop through all ingredients for the product. **NOTE: The product should have responsibility for getting all the ingredients.** For the loop use the “Frame” tool in the toolbox. In the Properties window for the frame set FrameKind to “Loop”, and Name to “All ingredients” (without the quotes). This will help a reader understand this loop.
- You will have to get category name. The product will have responsibility for doing this.

Exercise 2:

For this exercise you are creating a sequence diagram – please put it into the sequence diagram titled FindIngredient in the model you have been given. Please use the scenario below and the same Ingredients class diagram you used in the previous question to create the sequence diagram.

Scenario: Find all products having a specific ingredient

Precondition -- the list of products and their ingredients have already been retrieved from the database.

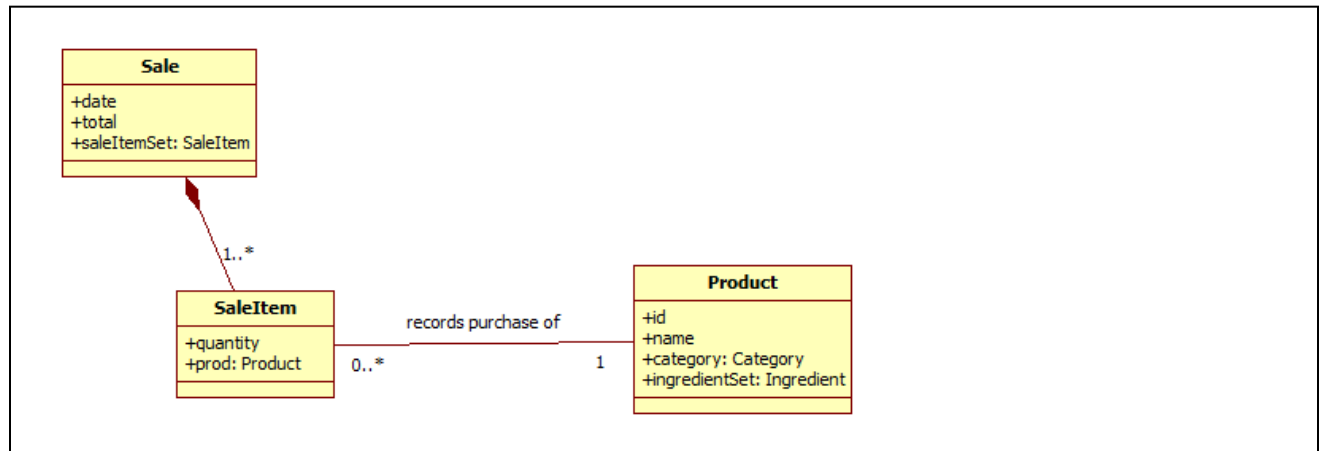
Actor – Baker	System
Enters an ingredient name and requests to find all products containing this ingredient.	Displays the names of the products which contain the given ingredient

Hints and Notes:

- You will have to use a loop within a loop – loop through all products and also all ingredients for each product. Use the frame tool as previously described – one frame will fit within another.
- **NEW CONCEPT:** if you find the given ingredient, you will need to get the product name. In this case it is good to use what is called a guard condition. The way to do this is as follows (ignore quotes):
 - Your message will look like this: “[ingredient found]: get name()”
 - How to set this up: in the properties window for the message, the message name will be “get name” as usual but you must then set the Branch to “ingredient found” (no [] brackets; the software will put these in)
 - This technique is good for simple situations; if the processing is more complex then a separate SD is probably called for.

Exercise 3:

For this exercise you are creating a sequence diagram – please put it into the sequence diagram titled **SalesReportByQuantity** in the model you have been given. Use the Sales class diagram and the “Report on quantities of product sold within a given date range” scenario to create the sequence diagram.



Scenario: Report on quantities of product sold within a given date range

Precondition – a list of sales has already been retrieved from the database. Sale items and products have also been retrieved from the database.

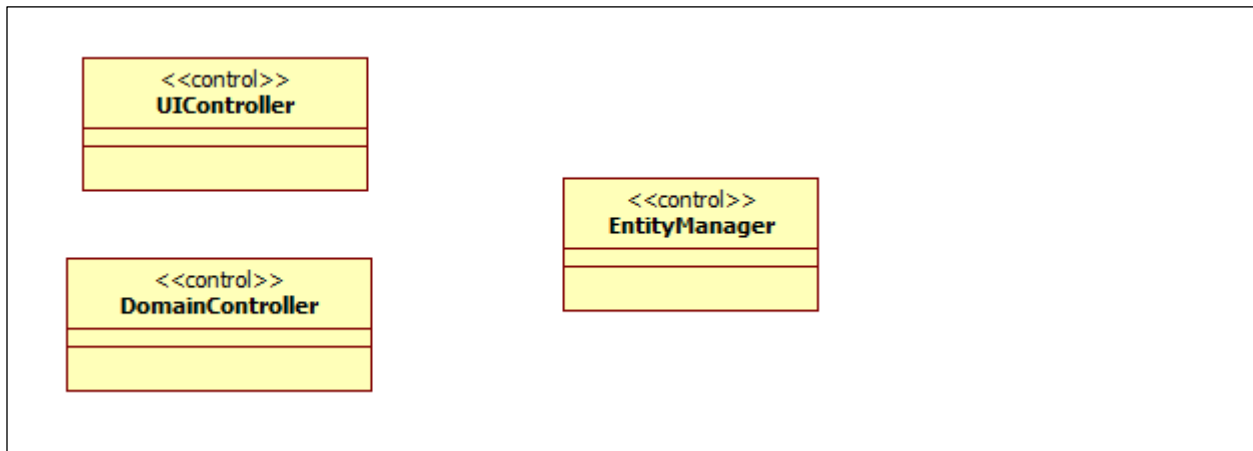
Actor – Manager	System
Enters date range and requests report	<p>Reads through all sales – if the sale is within the given date range the system collects the quantities and product names for all items in the sale.</p> <p>The system sums the sale quantities for each product and displays product name and total sale quantity for each product.</p>

Hints and Notes:

- You will have to use a loop within a loop – loop through all sales and also all sale items for a product. Use the frame tool as previously described.
- The **SaleItem** will have responsibility for getting product name. Note that nothing can access **SaleItem** except for **Sale** because it is part of a composition but **SaleItem** can access **Product**.
- You will probably need a guard condition to indicate that a sale within the range has been found.
- NEW CONCEPT – Message to self**
 - The data retrieved here must be summed and sorted. There are many ways of doing this. A simple way is to use a message to self (“self stimulus” in the tool box).
 - Put this message after the loops – it could be a Domain Controller message so that all data can be sorted and summed before being returned.

Exercise 4:

For this exercise you are modifying the class diagram titled **Controllers**.



Instructions:

- Look at the **ViewIngredients sequence diagram**. Using the information there please update the **DomainController** class, in the Controllers class diagram, with reference attributes and operations.

BIG HINT:

- You should have ONE reference attribute and ONE operation