

IC470, Software Engineering

Lab 2

Due: As per the course syllabus

Lab Focus: Object Oriented Analysis

- UML Use Case Modeling,
- Scenarios,
- Noun Extraction, and
- UML Class Modeling

Lab Partners: If there are other mids in this section from your capstone team, you may work together as a lab group. If not, pair up with another free mid from this section and work together on this lab. Include the names of your lab team members on each piece of paper you turn in for this lab.



System Description: The Food Services Manager wants you to create the Drydock/Steorage Ordering System (DSOS). DSOS allows users to place an order to either restaurant via a mobile device and requires them to pay for their food with either a credit card or their Yard Card at the time of ordering. Orders are sent to either Steorage or Drydock as appropriate, where a cook makes the food. An initial text message is sent to the user at the time of ordering with an expected time at which the food will be ready for pickup, with updates sent when the food is ready or delayed. Users must present the initial text message when picking up their food. Food not picked up by the user within one hour of the "food is ready" message being sent is offered for sale at a discounted price, with the proceeds donated to MWR, and no refunds are given to the user who placed the order.

Assumptions: You may make any reasonable assumptions about the system being described, but must list all such assumptions.

Lab assignment:

Part I: Use-Case Modeling

1. (20 pts) Using Visual Paradigm, create a UML Use Case Diagram for the DSOS. Show all services that your system will directly provide to the actors, for all the actors involved in the scenarios given in the System Description. Identify each actor's category as either: Principal, Secondary, Integral Hardware, or Other Systems. You must have at least one actor per category.
2. (10 pts) Give the first 6 steps of the "food not picked up in one hour" scenario given in the System Description.

Part II: Class Modeling. Give each stage of a noun extraction for the DSOS and then, using Visual Paradigm, produce a UML Class Diagram based on your noun extraction:

1. (10 pts) Stage 1: Concise Definition (describe the system in a single sentence)
2. (10 pts) Stage 2: Informal Strategy (expand Stage 1 sentence into a single paragraph, incorporate constraints)
3. (10 pts) Stage 3: Formalize the Strategy (identify & cull Stage 2 nouns, completing a table as described below).
 - a. List all the nouns from your Stage 2 in the first column. Use only singular forms, ignore duplicates, and list/group synonyms together. Ignore pronouns themselves, but ensure you list the noun to which the pronoun refers.
 - b. Draw a line through all nouns in your list that are outside the problem boundary.
 - c. For all remaining nouns, either:
 - i. list the abstract nouns in the second column (and include them as attributes in your UML Class Diagram below), or
 - ii. identify them as candidate classes in the third column

Example table headers:

Noun List	Abstract Nouns	Candidate Classes

4. Produce a first cut of a UML Class Diagram based on what remains after culling the nouns as per b and c above.
 - a. (30 pts) Your UML Class Diagram must make at least one valid use *each* of is-a (inheritance), has-a (composition), and an association in your UML Class Diagram. Use proper UML notation for all is-a, and has-a relationships, and label all other associations.

- b. (10 pts) If you have not already done so, add a database component to your UML Class Diagram to store order information. Your UML Class Diagram must abstract any database you use, the system, and the data stored in the database from each other.
 - i. This means that the system must go through a “database controller” or similar class to access data in the database, and that the database must be modeled separately (i.e. as a separate class) from the data contained within it (which is also modeled as a different class).
 - ii. Such abstractions promote software reuse and facilitate modifications to portions of the system that may change during development or after delivery.

Deliverables: Turn in your answers for Parts I and II (be sure to show all three stages of your noun extraction to include the culling), as well as your UML Use Case and Class Diagrams. Include the names of your lab team on each piece of paper you turn in.