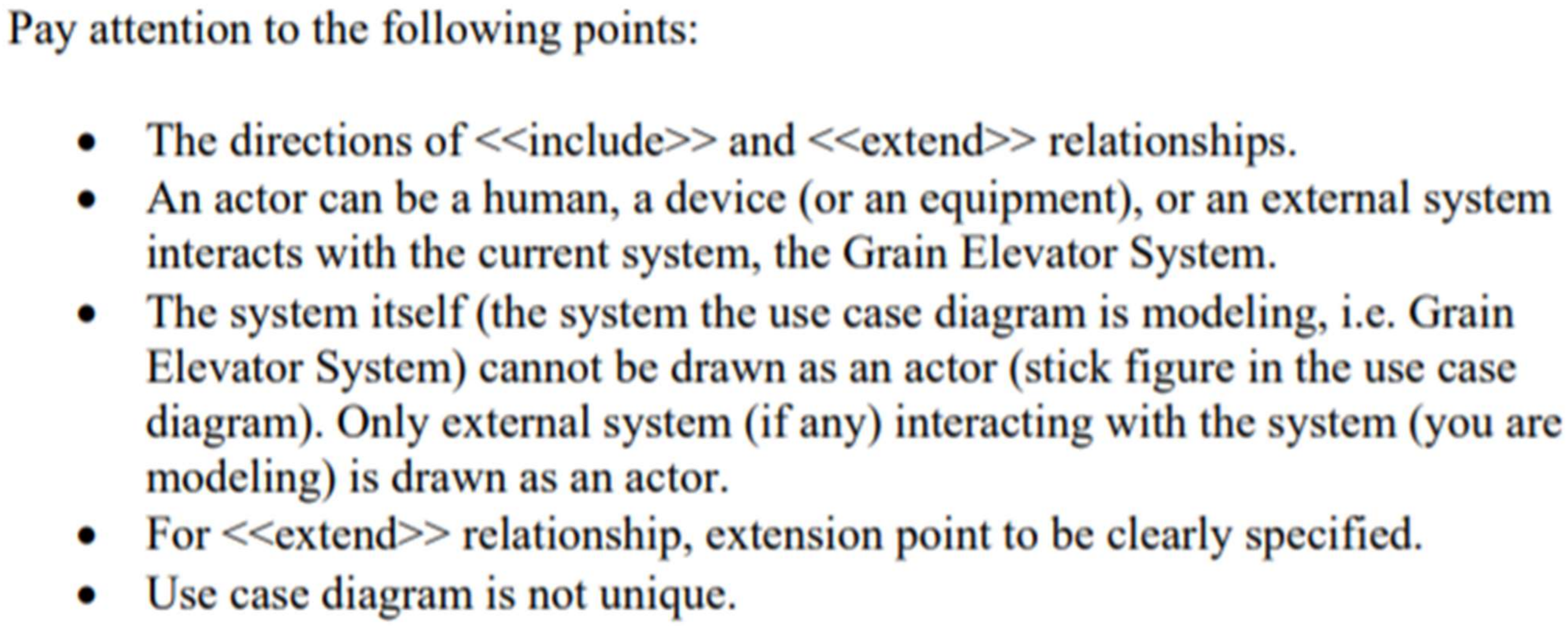
Tutorial 2: Requirements Elicitation – Use Case

Question 1: Grain Elevator System (GES)

(a) Make a complete use case model for the Grain Elevator System. Wherever appropriate, use include and extend relationships.

Points to take note:



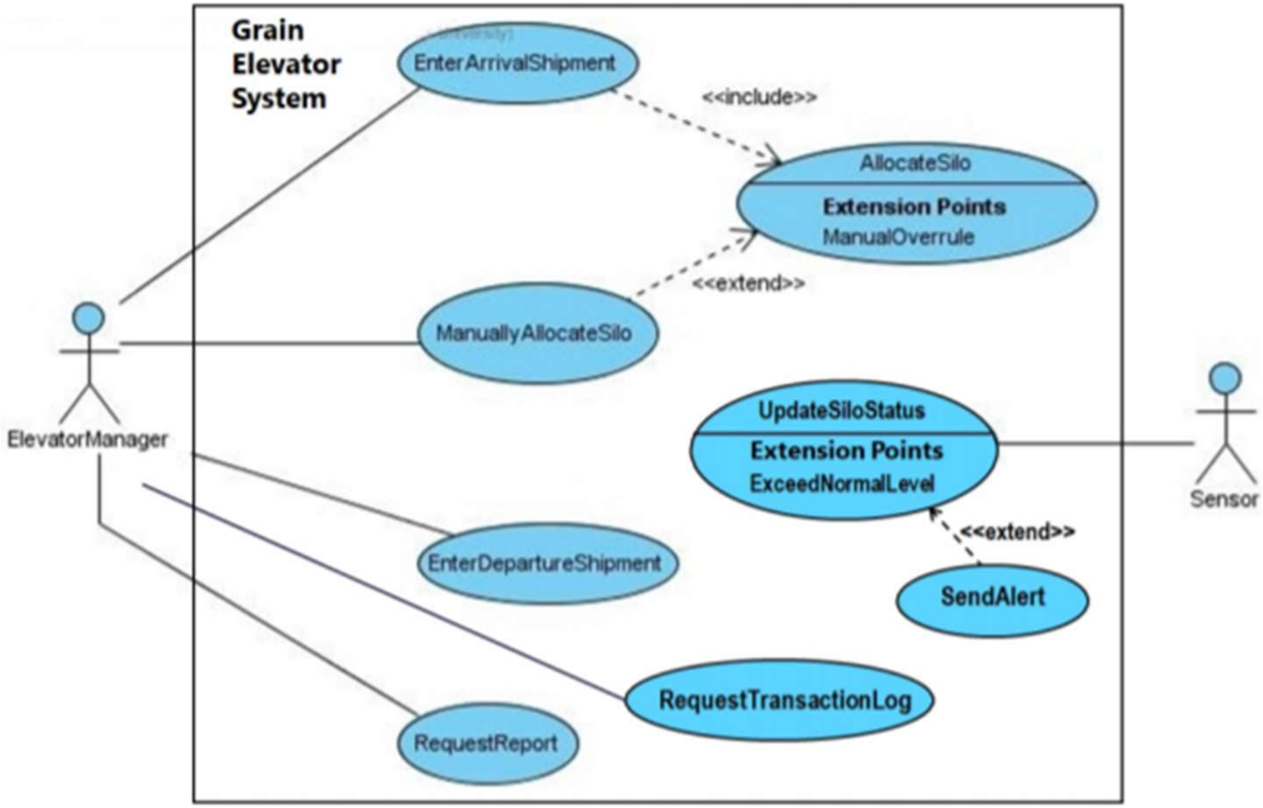
Pay attention to the following points:

- The directions of <<include>> and <<extend>> relationships.
- An actor can be a human, a device (or an equipment), or an external system interacts with the current system, the Grain Elevator System.
- The system itself (the system the use case diagram is modeling, i.e. Grain Elevator System) cannot be drawn as an actor (stick figure in the use case diagram). Only external system (if any) interacting with the system (you are modeling) is drawn as an actor.
- For <<extend>> relationship, extension point to be clearly specified.
- Use case diagram is not unique.

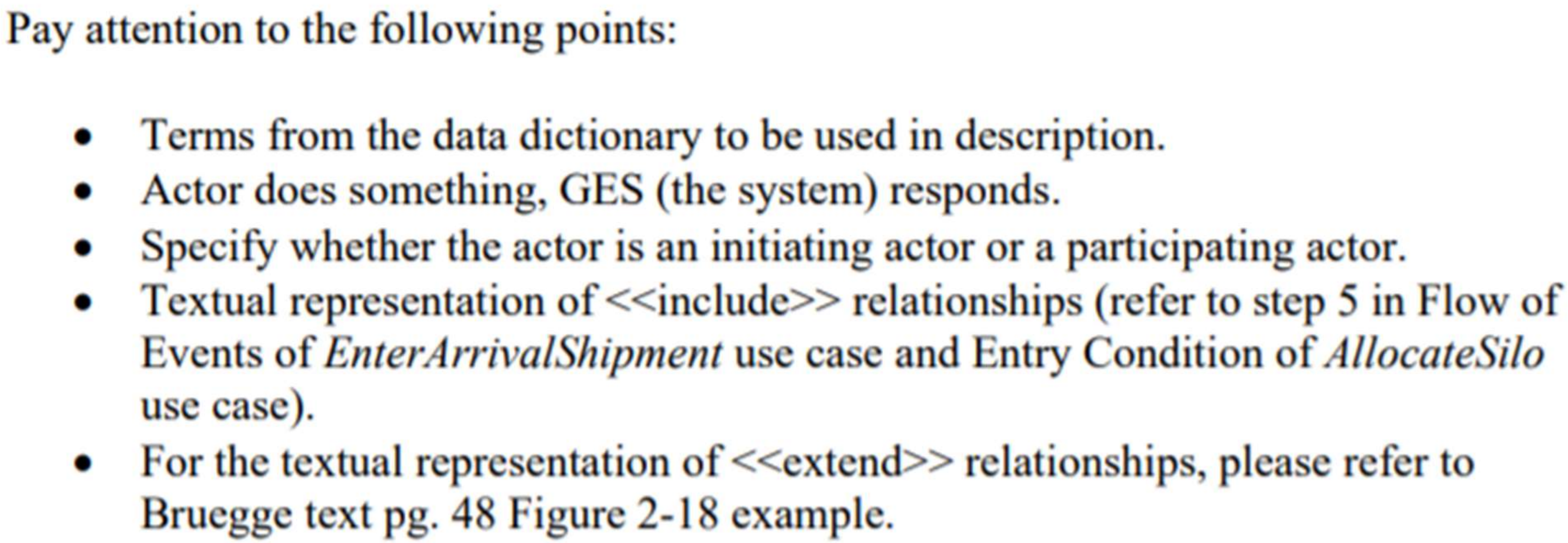
Tutorial 4: Requirements Elicitation – Use Case

Question 1: Grain Elevator System (GES)

(a) Make a complete use case model for the Grain Elevator System. Wherever appropriate, use include and extend relationships.



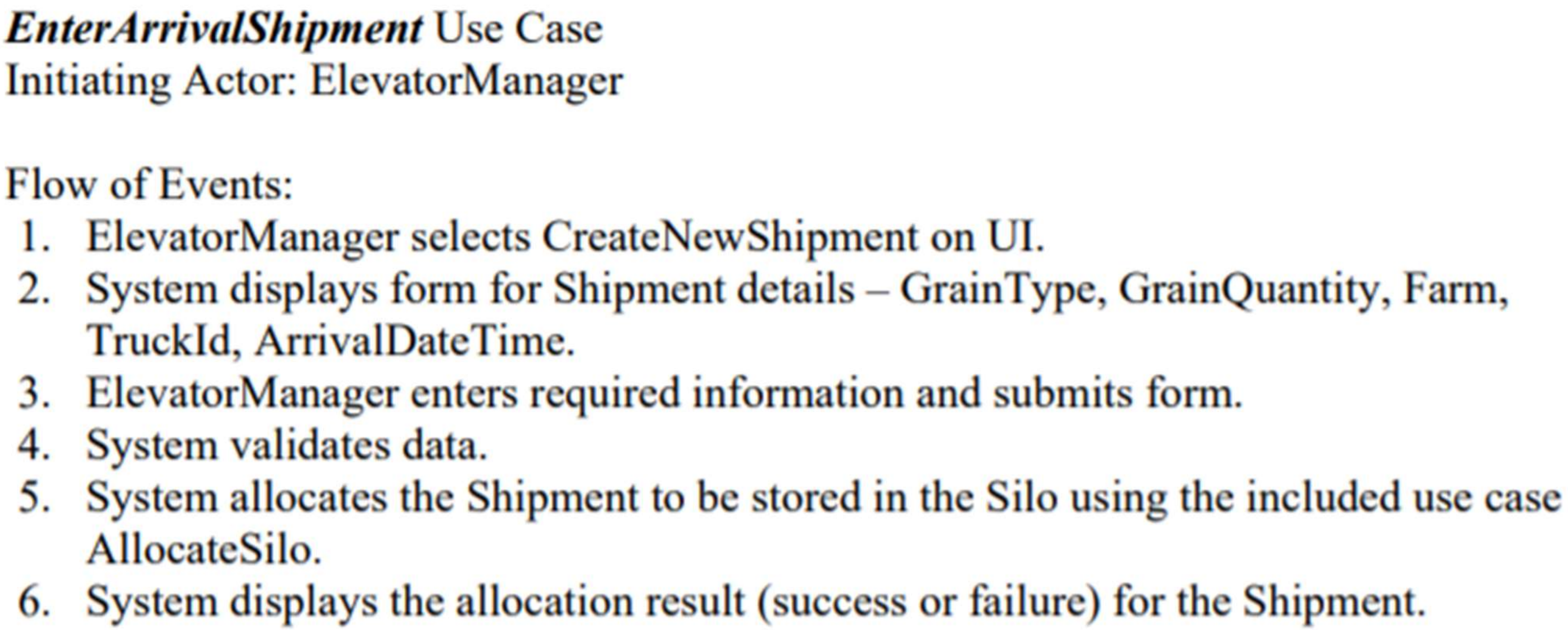
Question 1: Grain Elevator System (GES)
(b) Use Case Description



Pay attention to the following points:

- Terms from the data dictionary to be used in description.
- Actor does something, GES (the system) responds.
- Specify whether the actor is an initiating actor or a participating actor.
- Textual representation of <<include>> relationships (refer to step 5 in Flow of Events of *EnterArrivalShipment* use case and Entry Condition of *AllocateSilo* use case).
- For the textual representation of <<extend>> relationships, please refer to Bruegge text pg. 48 Figure 2-18 example.

Question 1: Grain Elevator System (GES)
(b) Use Case Description (cont'd)

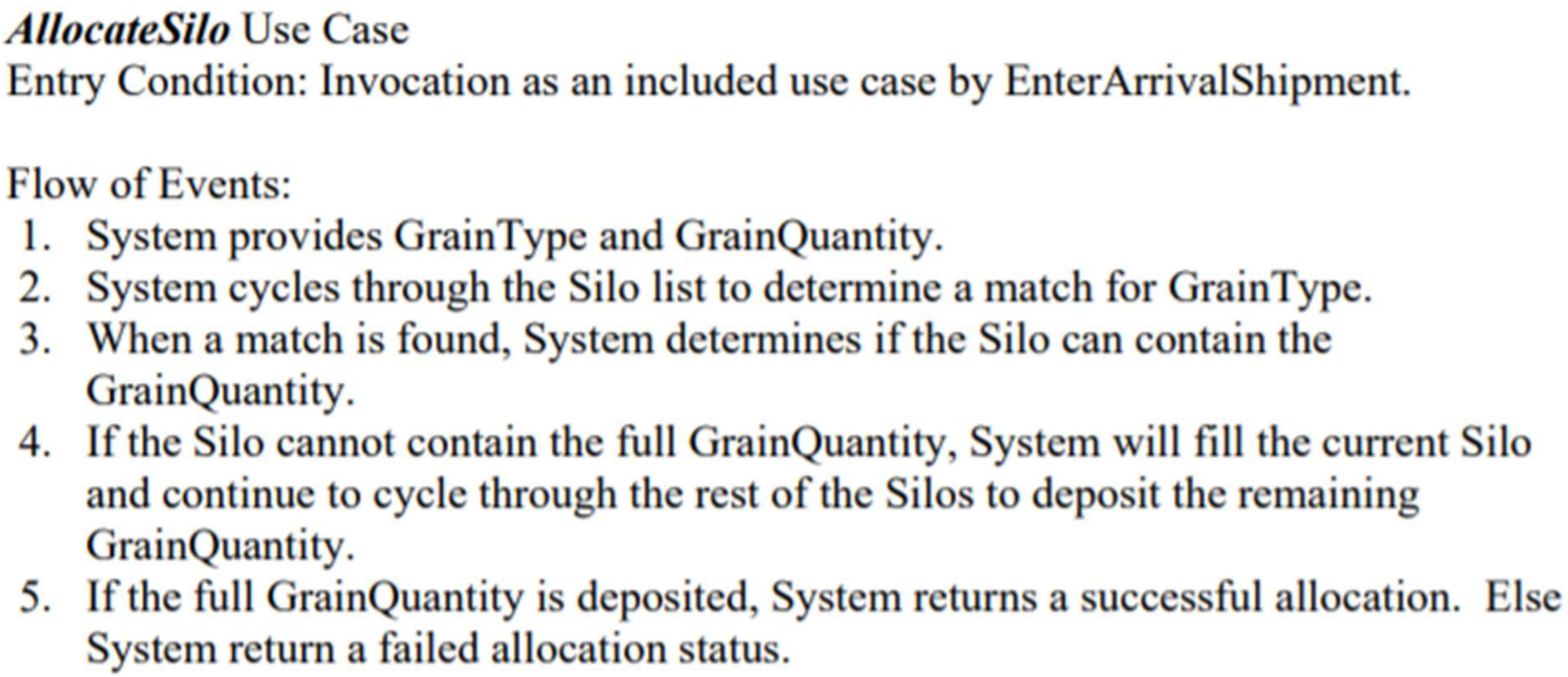


EnterArrivalShipment Use Case
Initiating Actor: ElevatorManager

Flow of Events:

1. ElevatorManager selects CreateNewShipment on UI.
2. System displays form for Shipment details – GrainType, GrainQuantity, Farm, TruckId, ArrivalDateTime.
3. ElevatorManager enters required information and submits form.
4. System validates data.
5. System allocates the Shipment to be stored in the Silo using the included use case AllocateSilo.
6. System displays the allocation result (success or failure) for the Shipment.

Question 1: Grain Elevator System (GES)
(b) Use Case Description (cont'd)

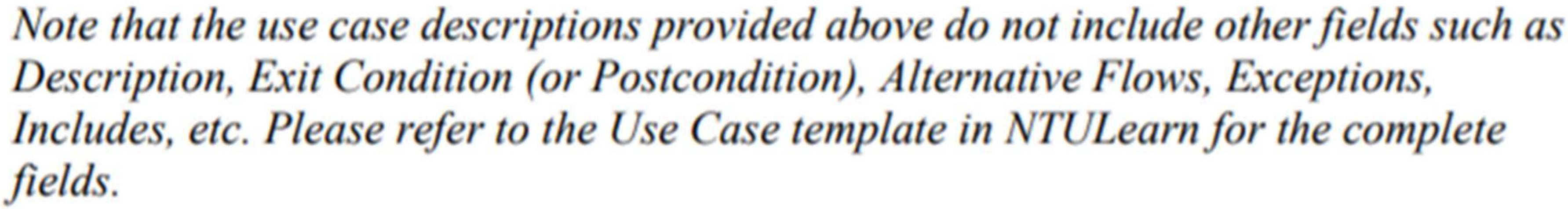


AllocateSilo Use Case
Entry Condition: Invocation as an included use case by EnterArrivalShipment.

Flow of Events:

1. System provides GrainType and GrainQuantity.
2. System cycles through the Silo list to determine a match for GrainType.
3. When a match is found, System determines if the Silo can contain the GrainQuantity.
4. If the Silo cannot contain the full GrainQuantity, System will fill the current Silo and continue to cycle through the rest of the Silos to deposit the remaining GrainQuantity.
5. If the full GrainQuantity is deposited, System returns a successful allocation. Else System return a failed allocation status.

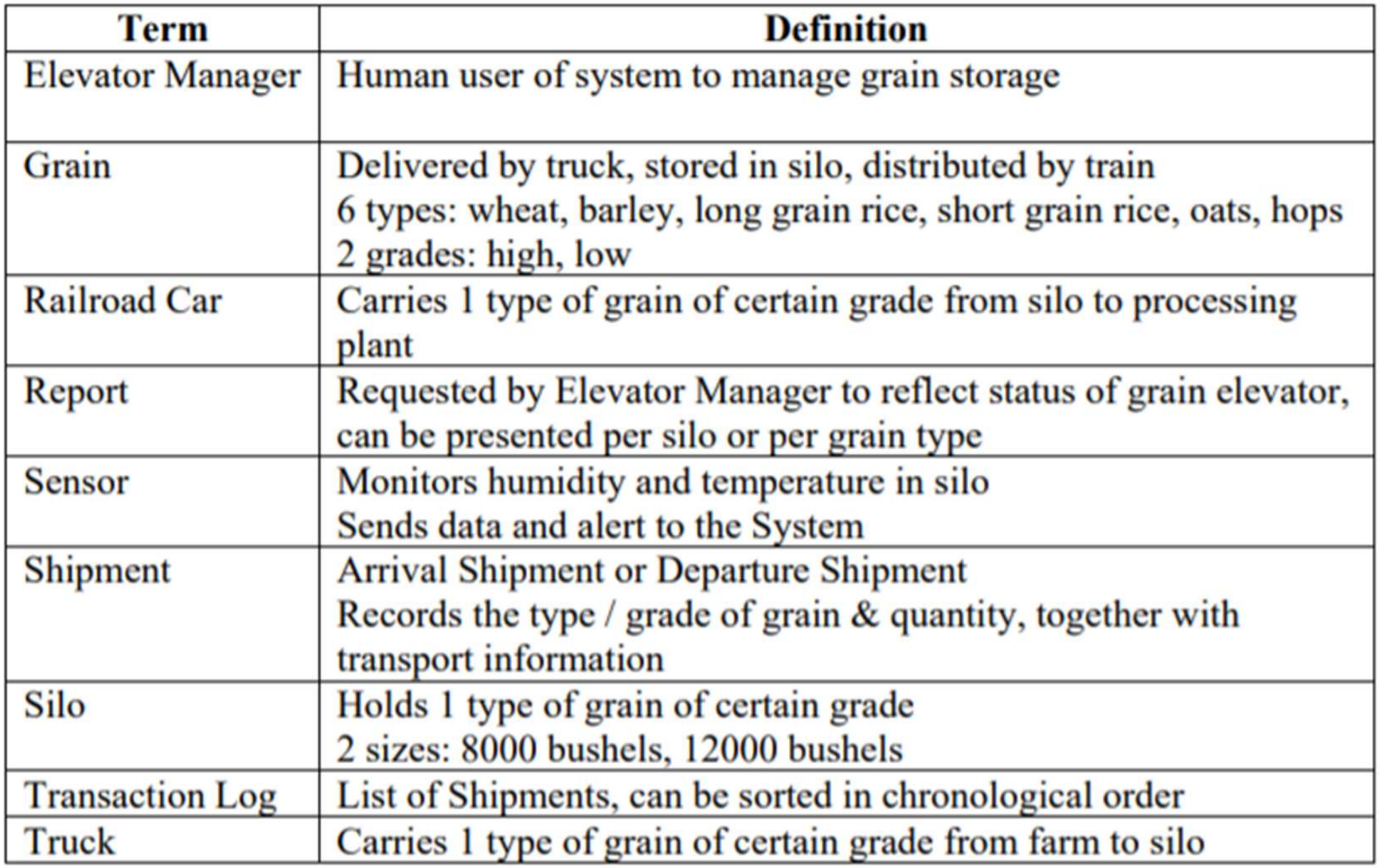
Points to take note:



Note that the use case descriptions provided above do not include other fields such as Description, Exit Condition (or Postcondition), Alternative Flows, Exceptions, Includes, etc. Please refer to the Use Case template in NTULearn for the complete fields.

Question 1: Grain Elevator System (GES)

(c) Data Dictionary



Term	Definition
Elevator Manager	Human user of system to manage grain storage
Grain	Delivered by truck, stored in silo, distributed by train 6 types: wheat, barley, long grain rice, short grain rice, oats, hops 2 grades: high, low
Railroad Car	Carries 1 type of grain of certain grade from silo to processing plant
Report	Requested by Elevator Manager to reflect status of grain elevator, can be presented per silo or per grain type
Sensor	Monitors humidity and temperature in silo Sends data and alert to the System
Shipment	Arrival Shipment or Departure Shipment Records the type / grade of grain & quantity, together with transport information
Silo	Holds 1 type of grain of certain grade 2 sizes: 8000 bushels, 12000 bushels
Transaction Log	List of Shipments, can be sorted in chronological order
Truck	Carries 1 type of grain of certain grade from farm to silo

Tutorial 4: Requirements Analysis - Conceptual Model in ClassDiagram

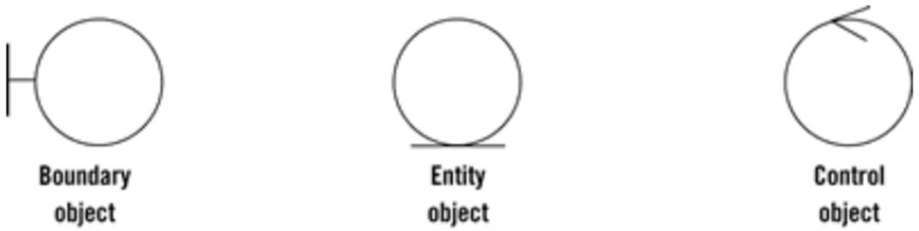
Question 2: Refer to the shipment arrival and allocation functionality use cases

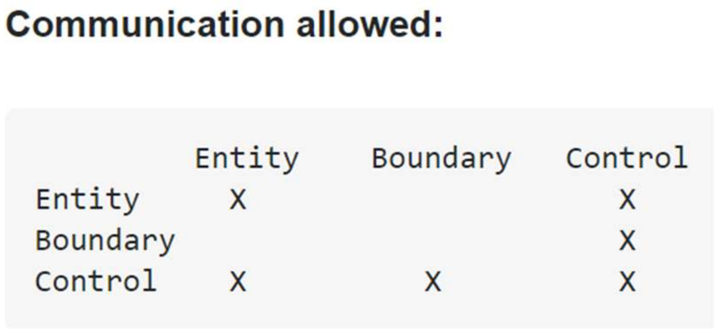
(a) From the use-case description and data dictionary (see Question 1), identify the classes to produce a conceptual model.

Points to node:

- The conceptual model class diagram is developed during the analysis phase of the SDLC. Thus, it is not a detail class diagram which is to be developed during the design phase.
- The main focus of the conceptual model is to identify the boundary, control, and entity classes, and the associations and/or dependencies between these classes.
- Attributes and operations are not mandatory to be included in the conceptual model.

(a) From the use-case description and data dictionary, identify the (analysis) classes to produce a conceptual model. (cont'd)





Communication allowed:

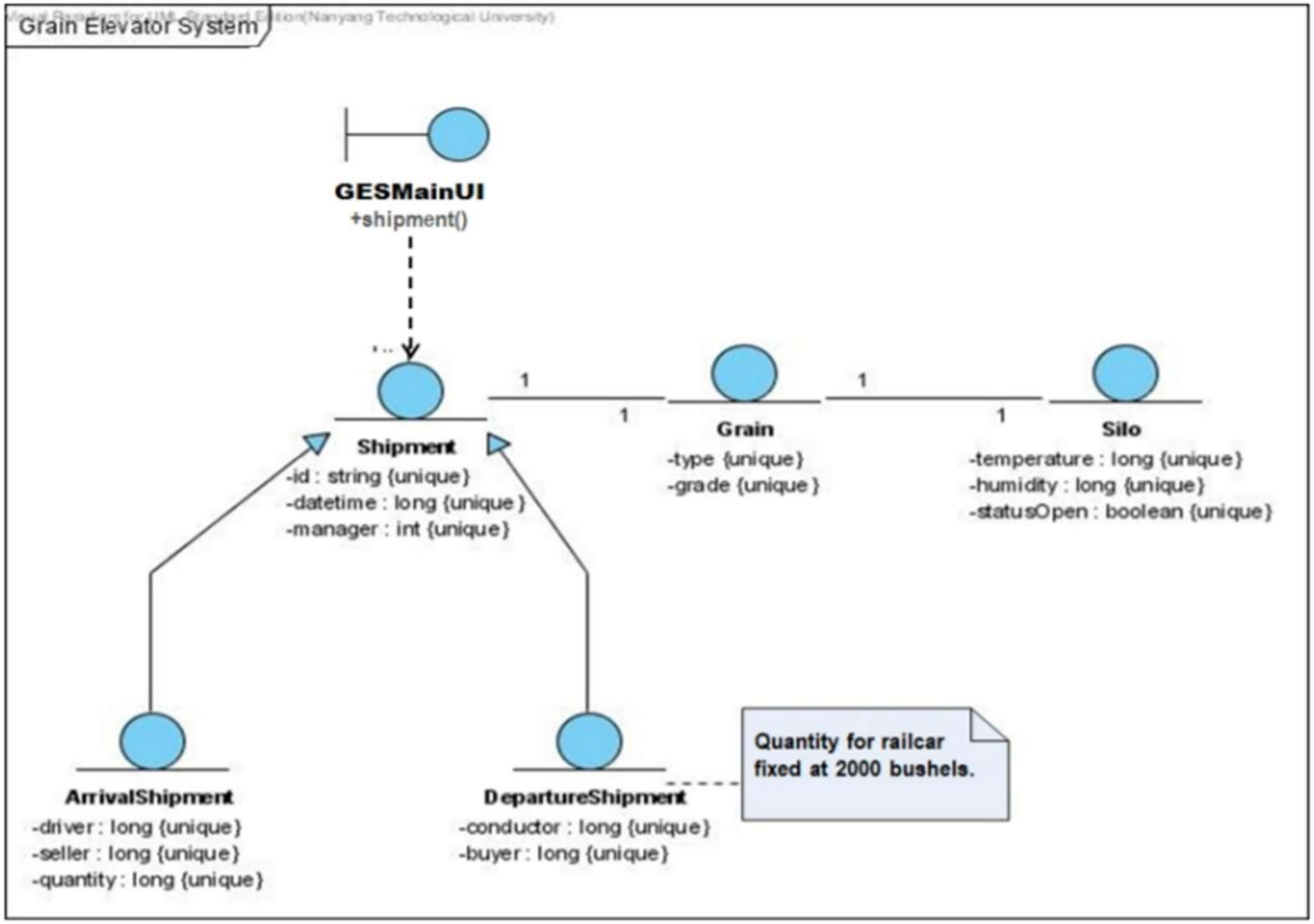
	Entity	Boundary	Control
Entity	X		X
Boundary			X
Control	X	X	X

Entities: **Objects representing system data, often from the domain model.**

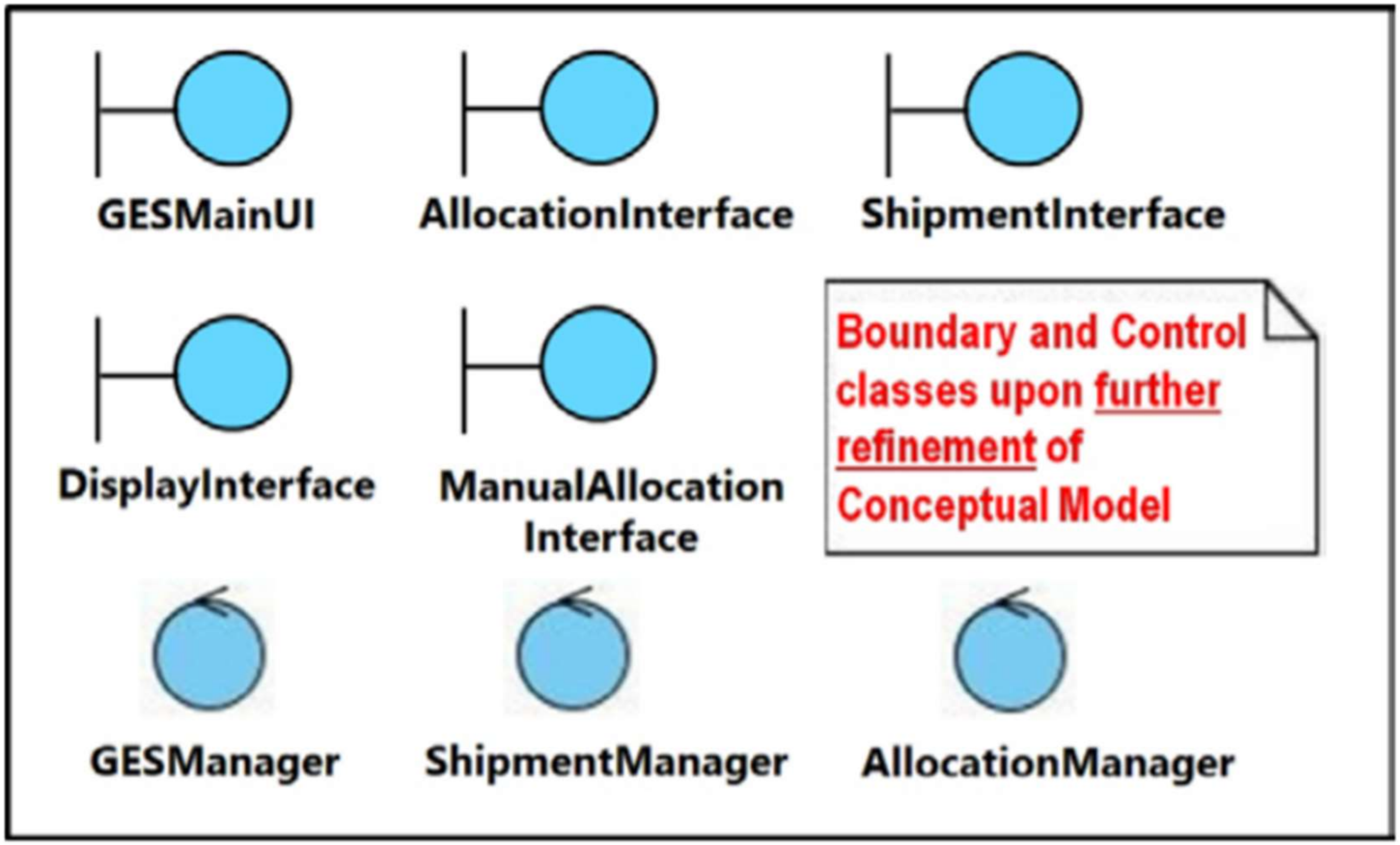
Boundaries: **Objects that interface with system actors (e.g. a user or external service). [Windows, screens and menus are examples of boundaries that interface with users.]**

Controls: **Objects that mediate between boundaries and entities. These serve as the glue between boundary elements and entity elements, implementing the logic required to manage the various elements and their interactions.**

Question 2: Grain Elevator System (GES)
(b) Initial class diagram



(b) Initial class diagram (cont'd)



Boundary and Control classes upon further refinement of Conceptual Model