Tutorial #1 – Requirements Elicitation – Functional / Non-Functional Requirements

1. Atomise the following descriptions for an Inventory and Asset Tracking System as functional requirements. Check that the atomized requirements are also verifiable.

   The System maybe queried. A query may contain a user number or a serial number. If the query contains a user number, all equipment assigned to that user is reported. If the query contains a serial number, the assignment for that computer is reported. All query results show the name, office and user number of a user, followed by the serial numbers and types of all computers assigned to that user, and the date of each assignment. If a user is not assigned a computer or a computer not assigned to a user, this is reported.

2. Identify several non-functional requirements for the above Inventory and Asset Tracking System.

1) 1 User must input query
1.1 query must contains user number
1.1.1 all equipment assigned to that user is reported
1.2 query must contains serial number
1.2.1 assignment for that computer is reported
1.3 query results must show name, office n user number of user
Followed by serial numb n type of all Comp assigned to user
1.4 System must report user is not assigned to Comp
1.8 System must report Comp not assigned to user

2) Non Functional requirement
↳ query must respond within 1 minute
↳ System take 2 minute to boot

Tutorial 1: Requirements Elicitation – Functional/Non-Functional Requirements

Question 1: Atomize the requirements for an Inventory and Asset Tracking System.

Check that the atomized requirements are also verifiable.

The System may be queried. A query may contain a user number or a serial number. If the query contains a user number, all equipment assigned to that user is reported. If the query contains a serial number, the assignment for that computer is reported. All query results show the name, office and user number of a user, followed by the serial numbers and types of all computers assigned to that user, and the date of each assignment. If a user is not assigned a computer or a computer not assigned to a user, this is reported.

Question 2: List several non-functional requirements for the above Inventory and Asset Tracking System

Question 1: Atomize the requirements

This tutorial is an exercise in splitting (atomizing) requirements statements and organizing them in a logical hierarchy. There should also be consideration for how the requirements can be verified.

Requirements specification heuristics:

- Write complete, simple sentences in the active voice.
- Define terms clearly and use them consistently.
- Group related material into sections.
- Express all requirements using the words "must" or "shall".
- Write verifiable requirements.
- Write atomized requirements.

1. Users shall be able to query the System.

    1.1. Users must query the System by user number or by serial number.

        1.1.1. A report for a query containing a user number must report all equipment assigned to the user with that user number.

        1.1.2. A report for a query containing a serial number must report the assignment for the computer with that serial number.

    1.2. All computer assignment reports must have the same format.

        1.2.1. A report of computer assignments must display user data followed by computer data.

1.2.2. The user data display must consist of information about a user, or an indication that the computer is unassigned.

1.2.2.1. The user data displayed for an assigned computer must consist of user's name.

1.2.2.2. The user data displayed for an assigned computer must consist of office number.

1.2.2.3. The user data displayed for an assigned computer must consist of user's number.

1.2.2.4. If the user is not assigned a computer, the user data display must indicate that no computer is assigned to that user.

1.2.3. The computer data display must consist of a list of data for each computer assigned.

1.2.3.1. The data for an assigned computer must consist of the computer's serial number.

1.2.3.2. The data for an assigned computer must consist of the types of computers.

1.2.3.3. The data for an assigned computer must consist of the date of the assignment.

1.2.3.4. If the list of assigned computers is empty, the computer data must indicate that no computer is assigned to the user.

Question 2: Non-functional requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system

Security: Authentication

Reliability: Consistency, accuracy

Performance: Response time

Maintainability: Easy to restart / recover

Scalability: Up capacity easily

Usability: Friendliness

Tutorial #2 – Requirements Analysis – Use Case & Conceptual Model in a Class Diagram

1. Refer to the description of the Grain Elevator System (GES).

   (a) Identify the Actors and their main Use Cases and draw a complete Use Case diagram for the Grain Elevator System. Wherever appropriate, use generalization, include and extend relationships.

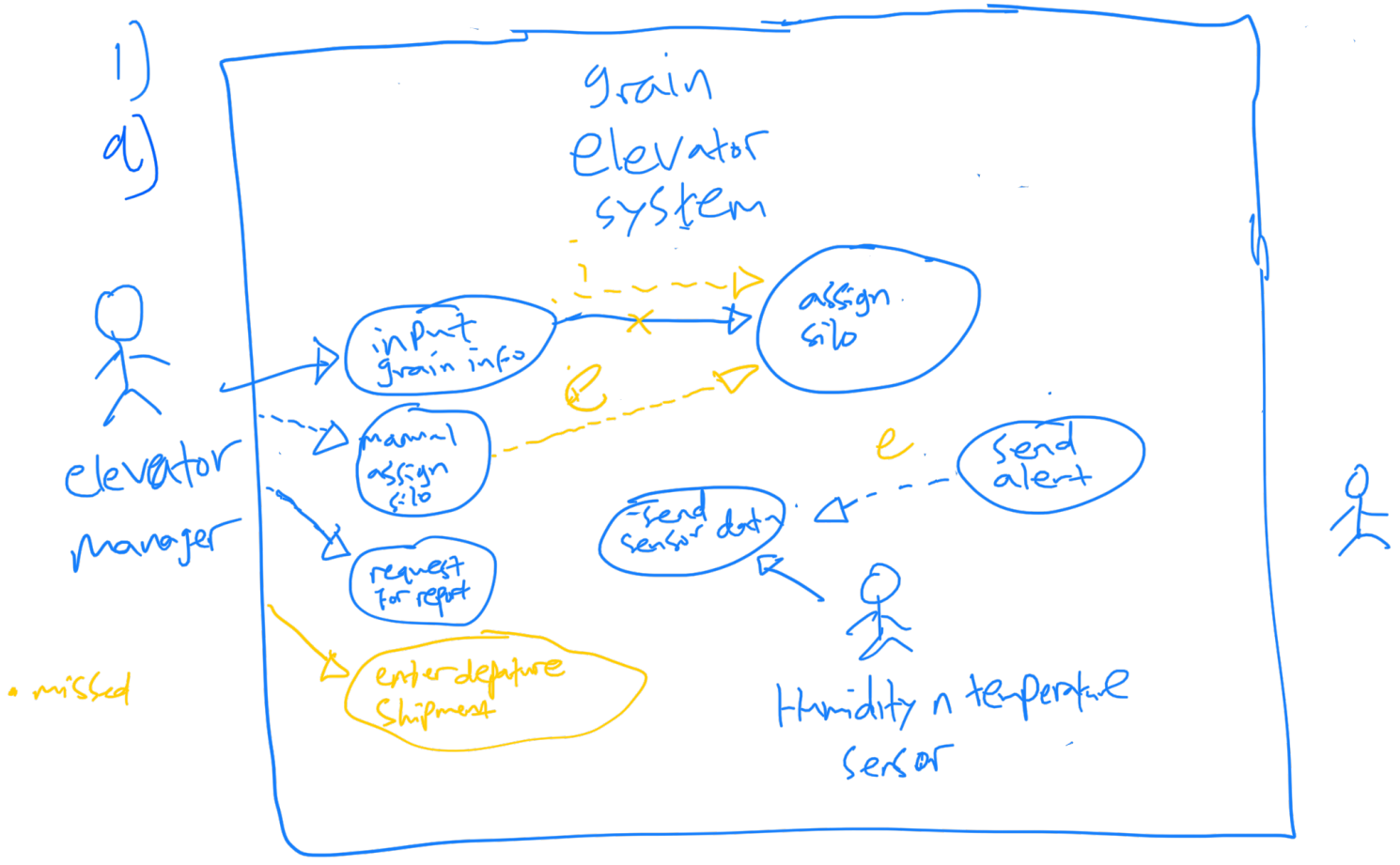
   (b) Write the Use Case description for the use case(s) that encompasses the grain shipment arrival and subsequent allocation functionality.

   (c) Determine and document important terms in the Data Dictionary

2. Refer to the shipment arrival and allocation functionality obtained in the above question.

   (a) From the Use Case description and Data Dictionary, identify the classes to produce a conceptual model.

   (b) Depict the classes identified in part (a) and the associations/dependencies between them in a Class Diagram.



b)

c)

term	definition
elevator manager	human
silo	hold 1 type of grain
Transaction log	

2) a)

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:

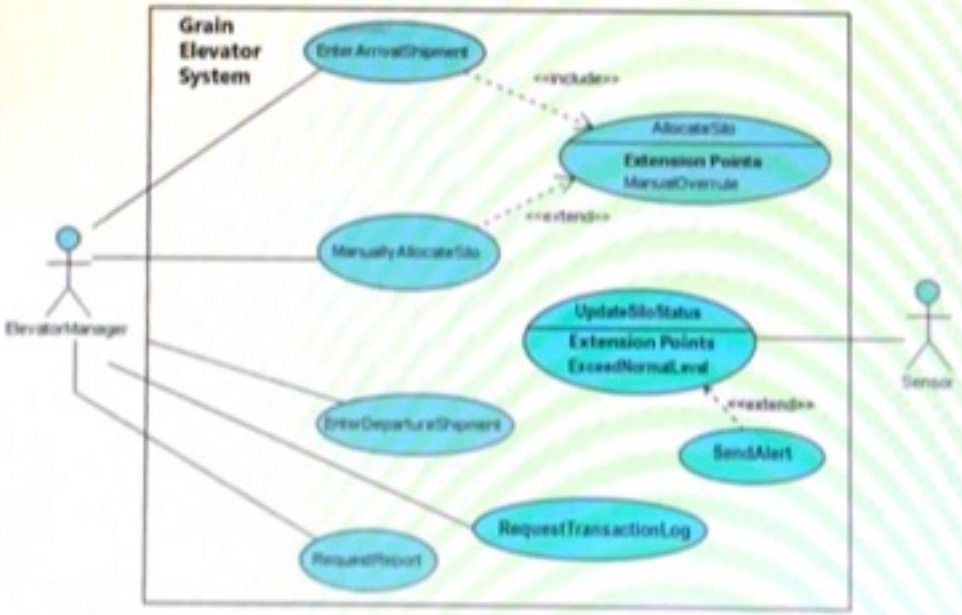
Pay attention to the following points:

- 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.

more ?

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

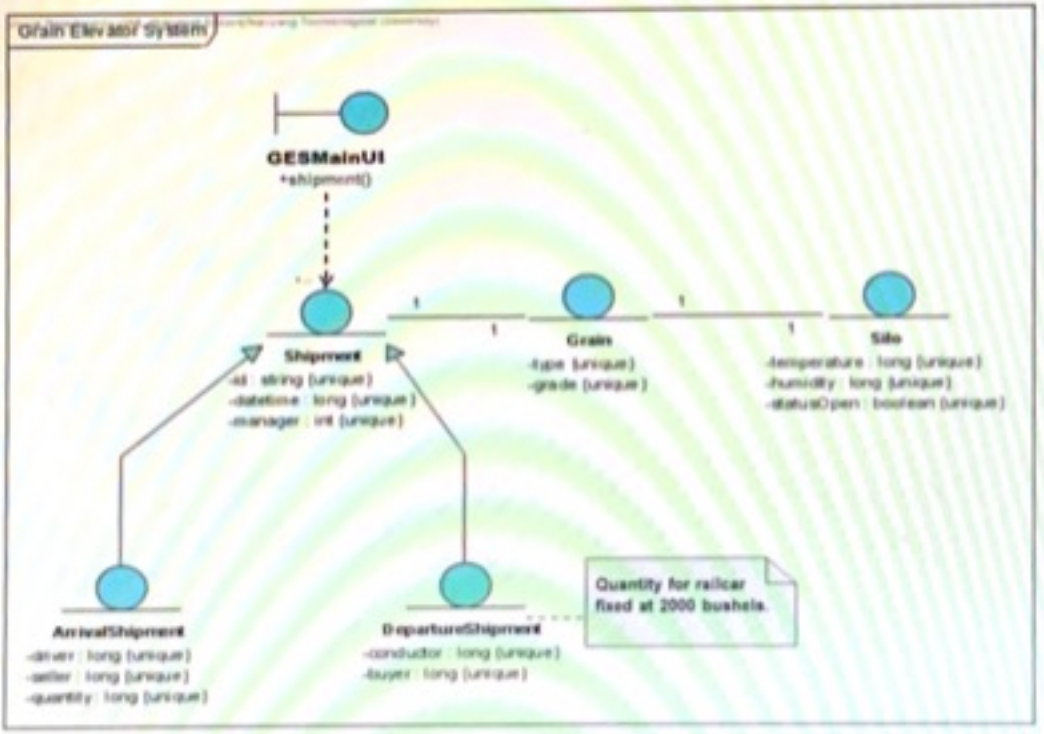
Entities: Objects representing system data, often from the domain model.

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.]

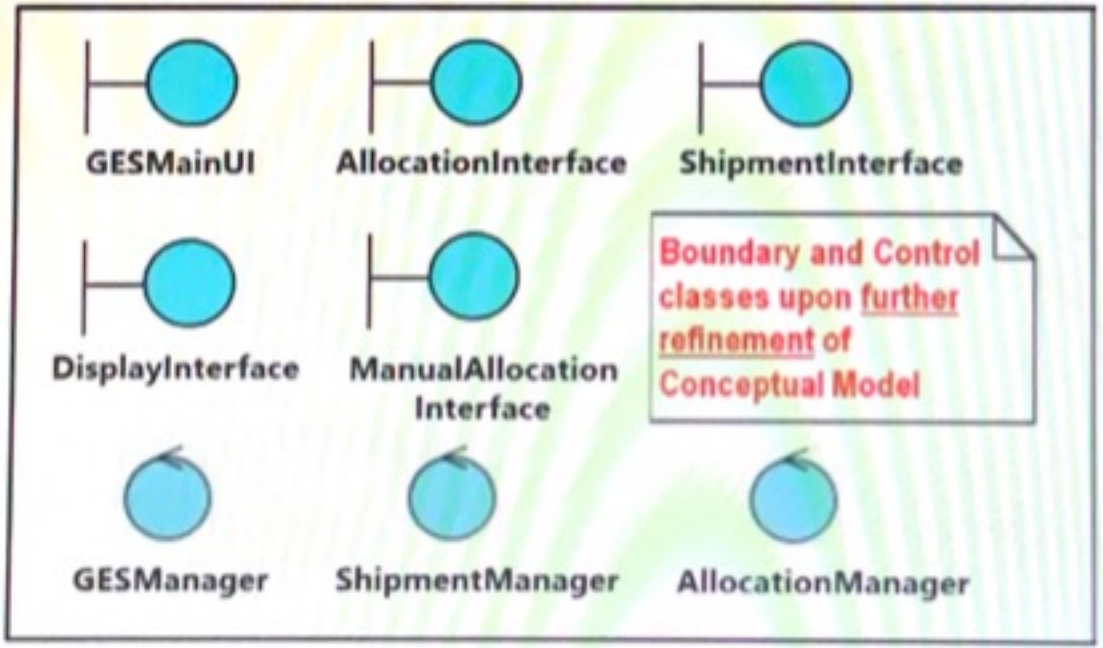
Controls: Objects that mediate between boundaries and entities. These serve as the glue between boundary elements and entity elements, implementing the logic required to manage the various elements and their interactions.

Question 2: Grain Elevator System (GES)

(b) Initial class diagram



(b) Initial class diagram (cont'd)



Tutorial #3 – Requirements Analysis – Dynamic Models in Sequence Diagram and State Machine Diagram

1. Grain Elevator System (GES). Refer to the shipment arrival and allocation functionality obtained in Tutorial #4.

   (a) In a Sequence Diagram, model the interaction amongst boundary / control / entity classes to enact the shipment arrival functionality.
   Demonstrate how the messages passed between objects enact the Use Case's flow of events.

   (b) Refine the conceptual model obtained in Tutorial #4 using details uncovered in question 1(a) above by adding:
   - associations,
   - generalization relationships,
   - properties / attributes,
   - operations, and
   - new classes (if any)

2. 8:30am lectures are usually frustrating for the lecturer. Projectors, having been turned off at the end of the previous day and take a long time to become ready for use.

   Here's what Professor Jane does at the Projector Controller Panel, in order to project slides from her laptop:-

   - Turn on the projector
   - As the projector warms up, it first gives a beep (sound) and a red indicator light continues to blink
   - When ready, the indicator light turns to green; the projector takes input from the PC by default
   - Jane presses the "LAPTOP" switch

   After the lecture, Jane could turn off the projector by pressing the "OFF" button. However, to save her colleagues the angst, she decides to leave it ON.
   Model the projector's system behavior using a State Machine diagram (can also be called a dialog map.

1) a)

Tutorial #4 – Software Processes

1. Why are software development costs so high?

2. Why do we spend so much time and effort maintaining existing software programs?

3. Discuss on the characteristics of software that affect its development.

4. What are the two distinct differences between software development and hardware manufacturing?

5. Sum up Agile manifesto in four sentences, then discuss the key Agile principles.

6. Discuss the following characteristics of eXtreme Programming (XP):
   i. When to use XP
   ii. Pair programming
   iii. Test-driven development (TDD)
   iv. Code refactoring

Tutorial #5 – Scrum

1. What are the considerations crucial to deciding whether to adopt Agile/Scrum development in a project?

2. Discuss the ways of estimating Velocity in Scrum.

3. What are the characteristics of a good product backlog?

4. What does the INVEST mnemonic of Agile software development stand for?

5. Under Scrum methodology, a Product Increment is a piece of software that is both complete and potentially shippable. Why is it important to have a working software delivered at the end of each Sprint?

6. The total size (in points) of a list of product backlog items is 64. The Scrum Team consists of Product Owner, Scrum Master and six members in the Development Team. The cost of each member in the Scrum Team for each Sprint is $2K. Eight Sprints is determined to be needed to complete the project. Calculate the velocity and cost of this project.

Tutorial #6 – Software Architecture and Strategy Pattern

1. Grain Elevator System (GES)

   a) Refer to the conceptual model developed in Question 1 of Tutorial#4.

      Use proper architectural styles to model dependencies among boundary, control, and entity objects.

2. Grain Elevator System (GES)

   The elevator manager uses a mobile device to process shipment and allocation. This mobile device must deal with a variety of network access protocols (LAN, WiFi, Bluetooth, 3G, 4G). Furthermore, we want to be able to deal with further network protocols with minimal impacts on the application.

   a) Identify the design problem and apply appropriate design pattern to address this problem

   b) Draw a UML class diagram depicting the classes in the design pattern and explain their roles

Tutorial #7 –Observer Pattern and Factory Pattern

1. Grain Elevator System (GES)

   Once grain arrives or leaves, GES will notify processing plants. The processing plants can register their interests in specific types of grains. GES will notify the registered processing plants using the plant preferred communication means, including email, SMS, or the combination of these means.

   Note that the interests and preferred communication means of the processing plants vary greatly, and the processing plants can change their interests and preferred communication means over time. Furthermore, GES should be easily extended to support new communication means once they become available.

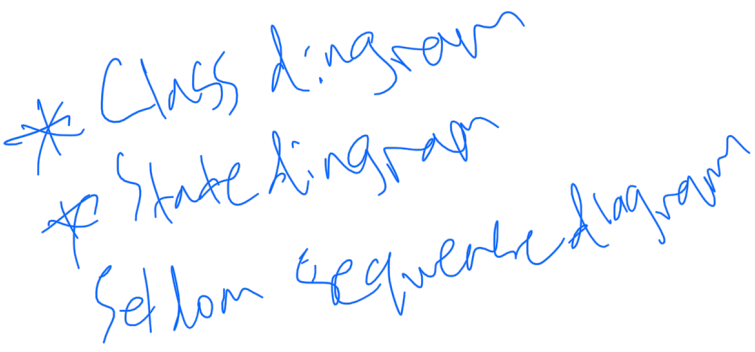
   a) Please identify the design problem in this feature and suggest a design pattern for addressing this problem.

   b) Illustrate your solution in a class diagram and explain the roles of each class.

2. Refer to the design pattern solution developed in Question 2 of Tutorial#6.

   a) Discuss what is missing in the solution

   b) Add a relevant design pattern to the design solution to complete the design



Tutorial #8 – Equivalence Class and Boundary Value Testing

1. Grain Elevator System

   A silo cannot accept any more grain if its temperature or humidity exceeds normal levels. Humidity ranges 0-100%, the normal humidity is 35 – 50%.
   Temperature ranges -10 – 100 C, the normal temperature is 40-60 C.

   a) Determine equivalence classes and boundary values for humidity and temperature

   b) Systematically design a set of test cases to test whether silo can accept grain or not based on its temperature and humidity.

Tutorial #9 – Control Flow Testing

1. A Java method to compute the sum of 1 .. n is shown in the following Java code below:

```
public int sum(int n, int upperbound){
   1  int result, i;
   2  result = 0;
   3 i = 0;
   4 if(n < 0) {
   5    n = -n;
   6  }
   7 while(i<n && result <= upperbound){
   8    i = i + 1;
   9    result = result + i;
   10 }
   11 if(result <= upperbound){
   12   System.out.println("The sum is " + result);
   13 } else {
   14   System.out.println("The sum is too large!");
   15 }
   16 return result;
}
```

   a) Draw the corresponding control flow graph for the `sum(int n, int upperbound)` method

   b) Calculate the Cyclomatic Complexity of the `sum(int)` method

   c) Design test cases to achieve 100% statement coverage and 100% branch coverage

   d) List the basis set of linearly independent paths, along with a test case (input parameters) and expected outcome for each of the path in the basis path set