NAME HAIDER ZAMAN

ID 14402

FINAL PAPER

TOTAL MARKS 50

Question no.1)

ELEMENTS OF USE CASE DIAGRAME

The elements of use case diagram are as follow

- Actors
- System boundary
- Include
- Extend

ACTORS:
An actor portrays any entity (or entities) that perform certain roles in a given system. The different roles the actor represents are the actual business roles of users in a given system.

SYSTEM BOUNDARY:

A system boundary defines the scope of what a system will be. A system cannot have infinite functionality.

Include:

When a use case is depicted as using the functionality of another use case in a diagram, this relationship between the use cases is named as
an include relationship.

EXTEND:

In an extend relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case.

QUESTION NO.6)

SINGLETON:

the singleton pattern is a software design pattern that restricts the instantiation of a class to one single instance.

```
Public Singleton
{
    Private static singleton uniqe;
    Private singleton() {
    Public static singleton  getInstance() {
        If (unique == null) {
          Unique=new Singleton();
}
Return unique;
}
}
```

SINGLETON
STATIC UNIQUE Instance
STATIC get Instance()

QUESTION NO.8

Observe Partner:

The Observer pattern defines a one-to-many dependency between objects so that one objects changes state, all of its dependents are notified and udated automatically.

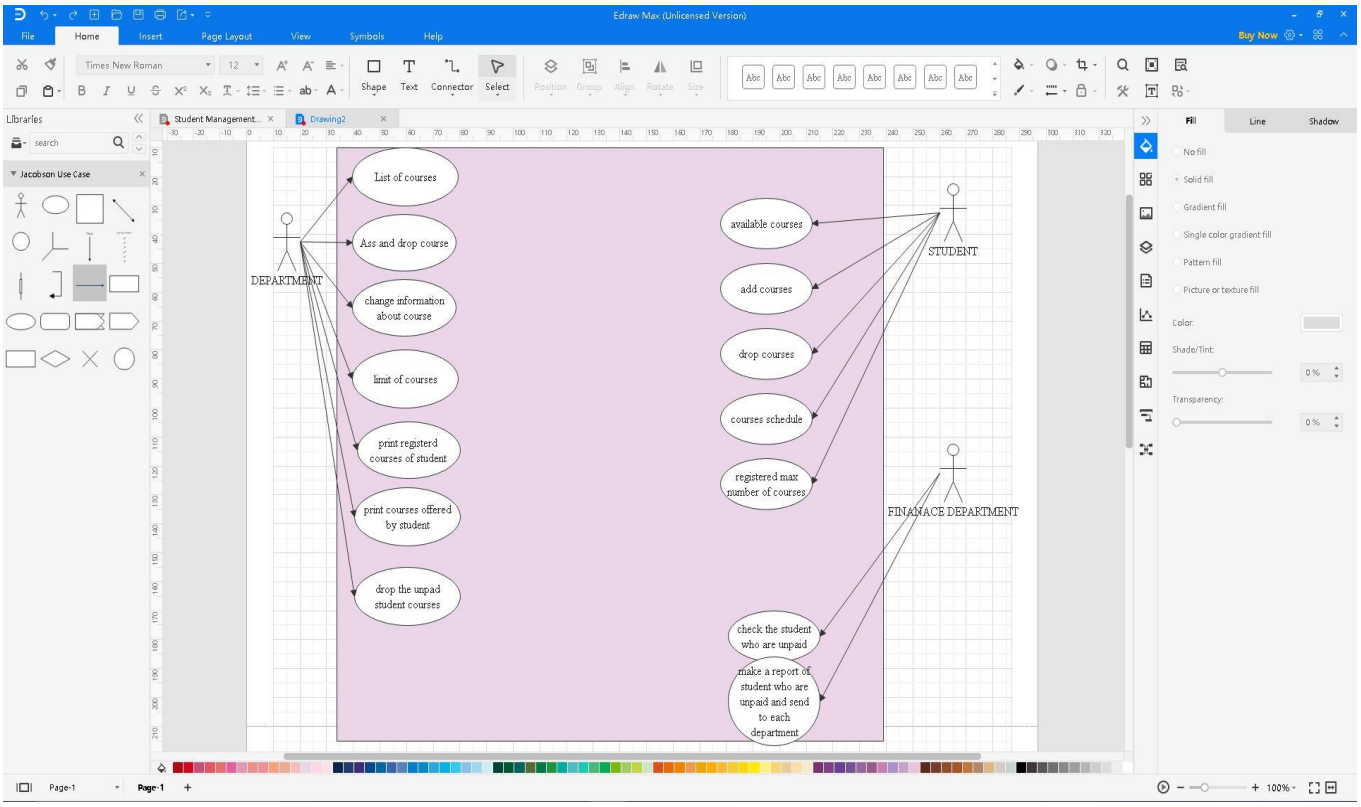
subject
Registered observer() Remove observer() Notify observer()

Observer
Update()

Concreate subject
Registered observer() Remover observer() Notify observer() getState() setState()

Concreate observer
Update()

Question no.5



Question no.4:

Modeling consists of building an abstraction of reality.

Abstractions are simplifications because:

They ignore irrelevant details and

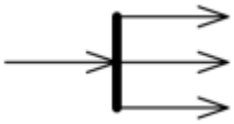
They only represent the relevant details.

What is relevant or irrelevant depends on the purpose of the model.

Question no.2

Fork node is a control node that has one incoming edge and multiple outgoing edges and is used to split incoming flow into multiple concurrent flows. Fork nodes are introduced to support parallelism in activities.

The notation for a fork node is a line segment with a single activity edge entering it, and two or more edges leaving it.



Fork node with a single activity edge entering it, and three edges leaving it.
The functionality of join node and fork node can be combined by using the same node symbol. This case maps to a model containing a join node with all the incoming edges shown in the diagram and one outgoing edge to a fork node that has all the outgoing edges shown in the diagram.

Question no.7

Our intent is to describe the most important ways in which software systems can offer privacy to their stakeholders. We express our privacy patterns as class diagrams in the UML (Universal Modelling Language), because this is a commonly-used language for expressing the high-level architecture of an object-oriented system. In this initial set of privacy patterns, we sketch how each of Westin's four states of privacy can be implemented in a software system.