	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri** **Department of Information Technology**

Subject name: Software Design Lab			Subject Code: ITL601
Class	TE IT	Semester –VI (CBCGS)	Academic year: 2018-19
Name of Student	**Kazi Jawwad A Rahim**		**QUIZ Score :**
Roll No	**27**	Assignment/Experiment No.	09
Title: **Designing of Component and Deployment Diagram.**			

1. Lab objectives applicable: LOB4.

2. Lab outcomes applicable: LO4

3. Learning Objectives:

1. To understand basics of Component diagram in UML.
2. To understand basics of Deployment diagram in UML.

4. Practical applications of the assignment/experiment: Diagrams are used in order to give an insight for the activities involved in the functionality of a system.

5. Prerequisites:

1. SRS

6. Hardware Requirements:

Windows operating system (Windows 7 or higher)

7. Software Requirements:

UML designing tool such as IBM Rational Rose/StarUML

8. Quiz Questions (if any): (Online Exam will be taken separately batch-wise, attach the certificate/ Marks obtained)

1. Which parameters are controlled by deployment diagrams?
2. What is the purpose behind using deployment diagram?
3. Which artifacts must be identified before drawing component diagrams?
4. What is the purpose behind using component diagram?

9. Experiment/Assignment Evaluation:

Sr. No.	**Parameters**		**Marks obtained**	**Out of**
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -			6
2	Neatness/presentation			2
3	Punctuality			2
Date of performance (DOP)	**02-04-2019**	**Total marks obtained**		**10**

Signature of the faculty

10. Theory:

Component Diagram

- **Overview:**

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model physical aspects of a system.

Physical aspects are the elements like executables, libraries, files, documents etc. which resides in a node.

So component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

- **Purpose**

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

1. Visualize the components of a system.
2. Construct executables by using forward and reverse engineering.
3. Describe the organization and relationships of the components.

- **How to draw Component Diagram?**

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc.

So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details.

Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.

This diagram is very important because without it the application cannot be implemented efficiently. A well prepared component diagram is also important for other aspects like application performance, maintenance etc.

So before drawing a component diagram the following artifacts are to be identified clearly:

1. Files used in the system.
2. Libraries and other artifacts relevant to the application.
3. Relationships among the artifacts.

Now after identifying the artifacts the following points needs to be followed:

1. Use a meaningful name to identify the component for which the diagram is to be drawn.
2. Prepare a mental layout before producing using tools.
3. Use notes for clarifying important points.

- **Where to use Component diagrams?**

We have already described that component diagrams are used to visualize the static implementation view of a system. Component diagrams are special type of UML diagrams used for different purposes.

These diagrams show the physical components of a system. To clarify it, we can say that component diagrams describe the organization of the components in a system.

Organization can be further described as the location of the components in a system. These components are organized in a special way to meet the system requirements.

As we have already discussed those components are libraries, files, executables etc. Now before implementing the application these components are to be organized. This component organization is also designed separately as a part of project execution.

Component diagrams are very important from implementation perspective. So the implementation team of an application should have a proper knowledge of the component details.

Now the usage of component diagrams can be described as:

1. Model the components of a system.
2. Model database schema.
3. Model executables of an application.
4. Model system's source code.

Deployment diagram

- **Overview:**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

- **Purpose**

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as:

1. Visualize hardware topology of a system.
2. Describe the hardware components used to deploy software components.
3. Describe runtime processing nodes.

- **How to draw Deployment Diagram?**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following parameters:

1. Performance
2. Scalability
3. Maintainability
4. Portability

So before drawing a deployment diagram the following artifacts should be identified:

1. Nodes
2. Relationships among nodes

- **Where to use Deployment diagrams?**

Deployment diagrams are mainly used by system engineers. These diagrams are used to describe the physical components (hardware), their distribution and association.

To clarify it in details we can visualize deployment diagrams as the hardware components/nodes on which software components reside.

Software applications are developed to model complex business processes. Only efficient software applications are not sufficient to meet business requirements. Business requirements can be described as to support increasing number of users, quick response time etc.

To meet these types of requirements hardware components should be designed efficiently and in a cost effective way.

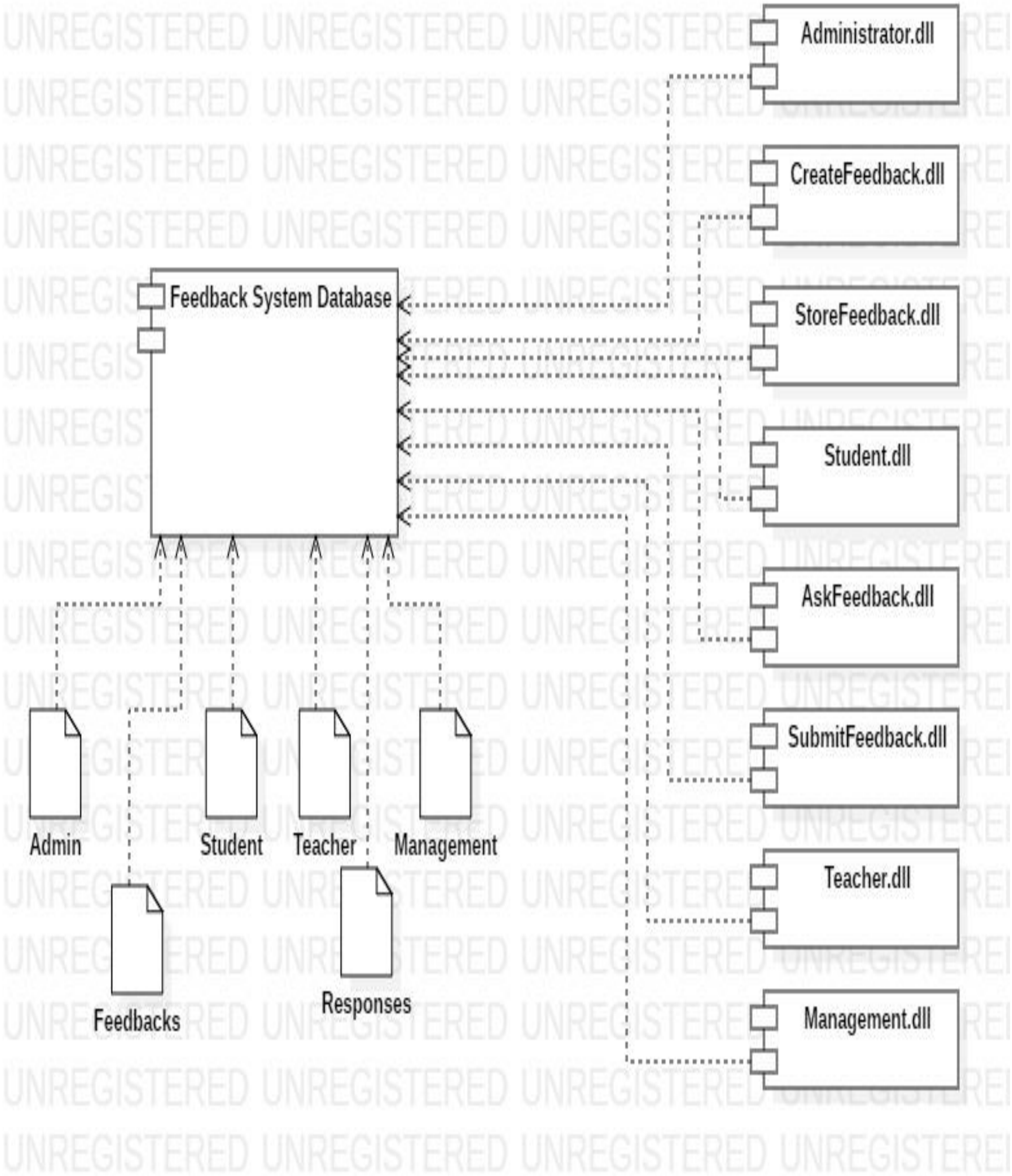
Now a day's software applications are very complex in nature. Software applications can be stand alone, web based, distributed, mainframe based and many more. So it is very important to design the hardware components efficiently.

So the usage of deployment diagrams can be described as follows:

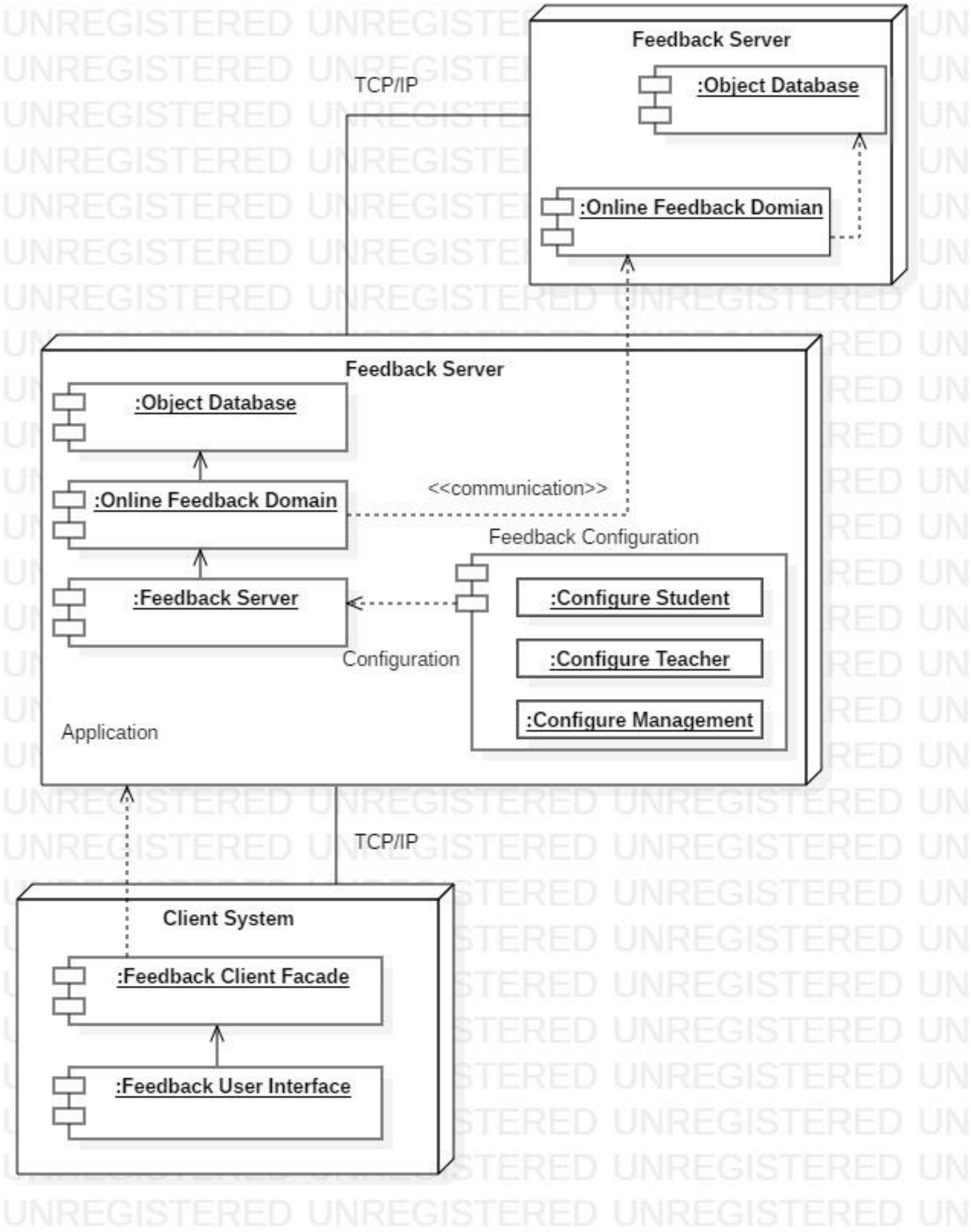
1. To model the hardware topology of a system.
2. To model embedded system.
3. To model hardware details for a client/server system.
4. To model hardware details of a distributed application.
5. Forward and reverse engineering.

11. Source code / Diagrams:

Component Diagram



Deployment Diagram



References:

[1]. https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
[2]. https://www.tutorialspoint.com/uml/
[3]. http://vlabs.iitkgp.ernet.in/se/6/theory/