EX:NO:4	STOCK MAINTAINANCE SYSTEM
DATE:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, collaboration, deployment, state chart , package] for the Stock maintainence system.

SOFTWARE REQUIREMENTS SPECIFICATION:

SL.NO	SOFTWARE REQUIREMENTS SPECIFICATION
1.0	Hardware Requirements
1.1	Software Requirements
1.2	Problem Analysis and Project Plan
1.3	Project Description
1.4	Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLANNING :

The Stock Maintenance System, initial requirement to develop the project about the mechanism of the Stock Maintenance System is caught from the customer. The requirement are

analyzed and refined which enables the end users to efficiently use Stock Maintenance System. The complete project is developed after the whole project analysis explaining about the scope and the project statement is prepared.

1.3 PROJECT DESCRIPTION:

This software is designed for supporting the computerized stock maintainence System .In this system, the customer can place order and purchase items with the aid of the stock dealer and central stock system. This orders are verified and the items are delivered to the customer.

1.4 REFERENCES:

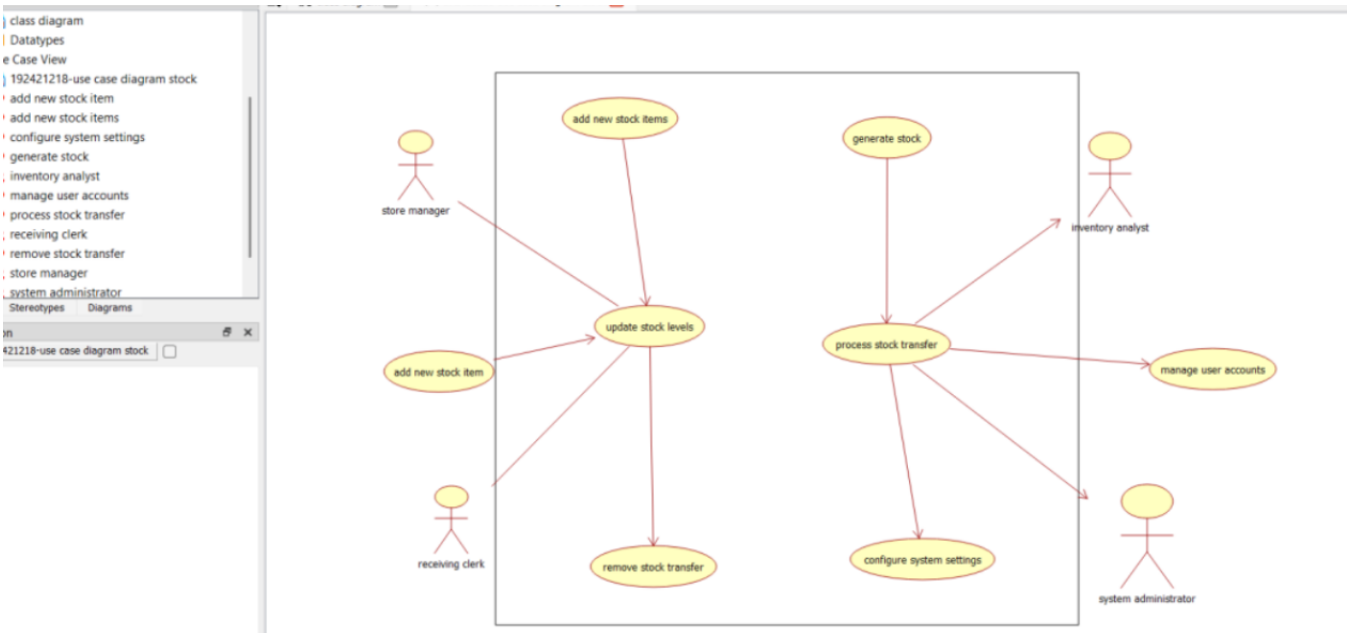
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

This diagram will contain the actors, use cases which are given below

Actors: Customer, Stock dealer, central stock system.

Use case: purchase order, verification of order, payment ,delivery of items.

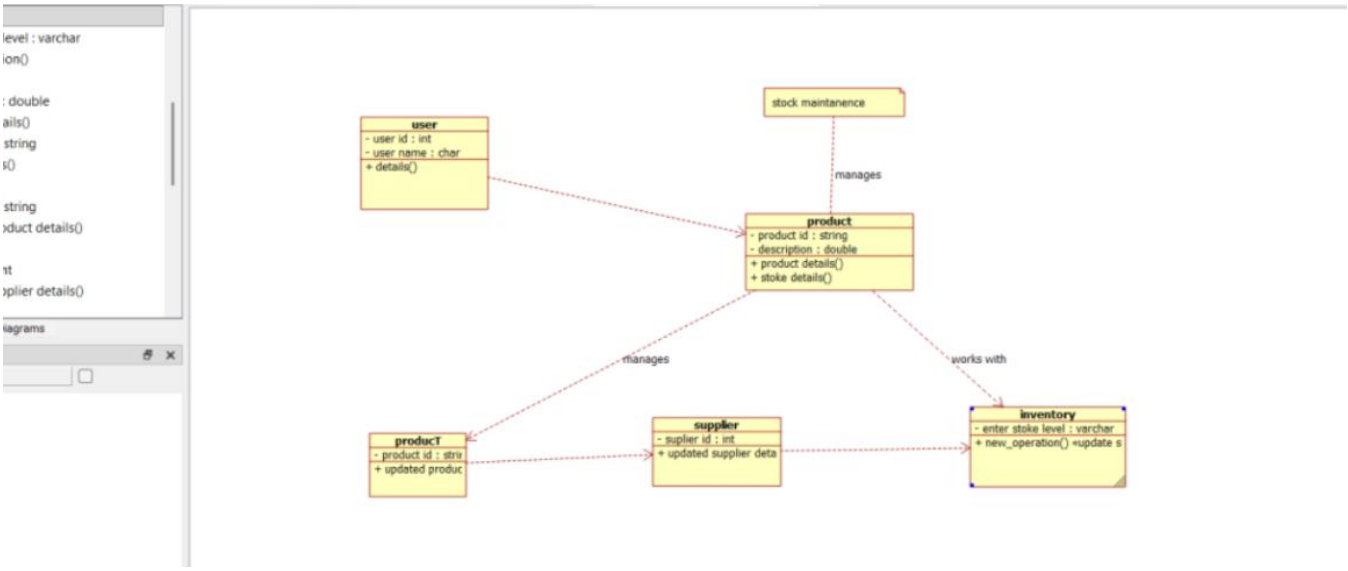


CLASS DIAGRAM:

This diagram consists of the following classes, attributes and their operations.

CLASSES	ATTRIBUTES	OPERATIONS
Central stock system	Store stock details	Print bill()

Stock dealer	Take order	Deliver item()
Customer	Place order	Payment()

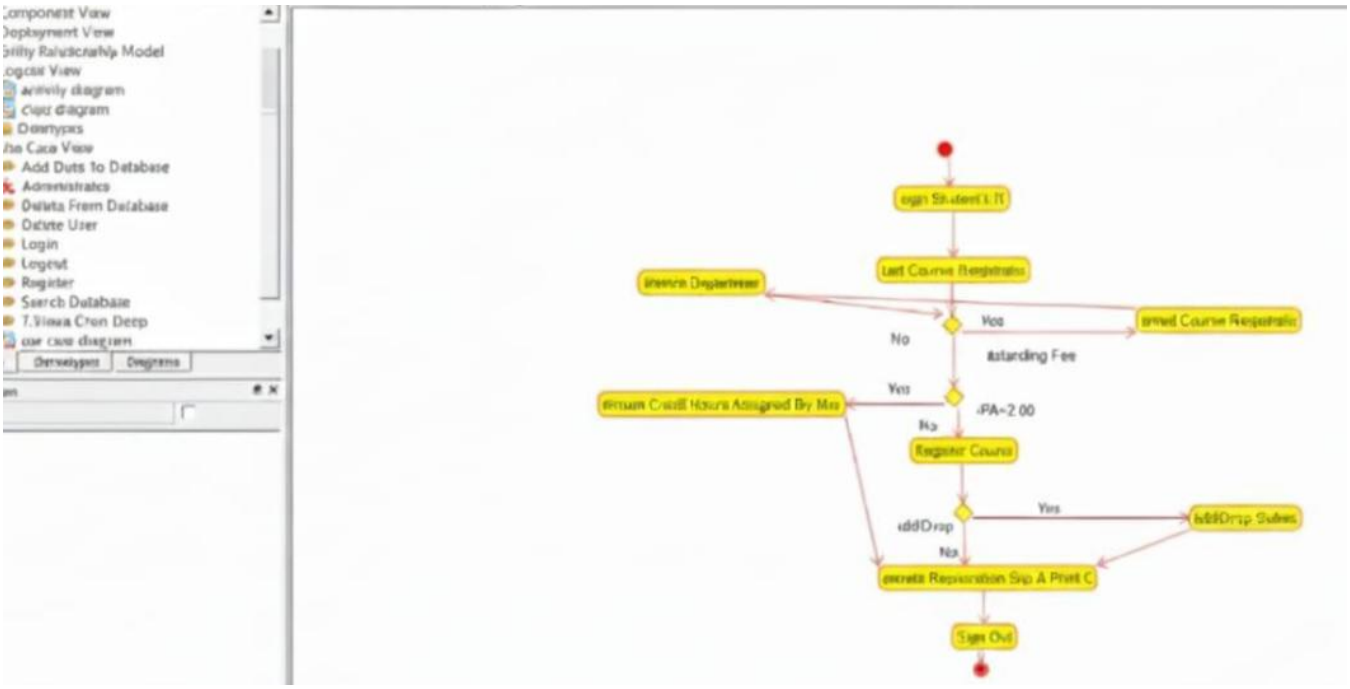


ACTIVITY DIAGRAM:

This diagram will have the activities as Start point ,End point, Decision boxes as given below:

Activities: Purchase order, payment , delivery of items.

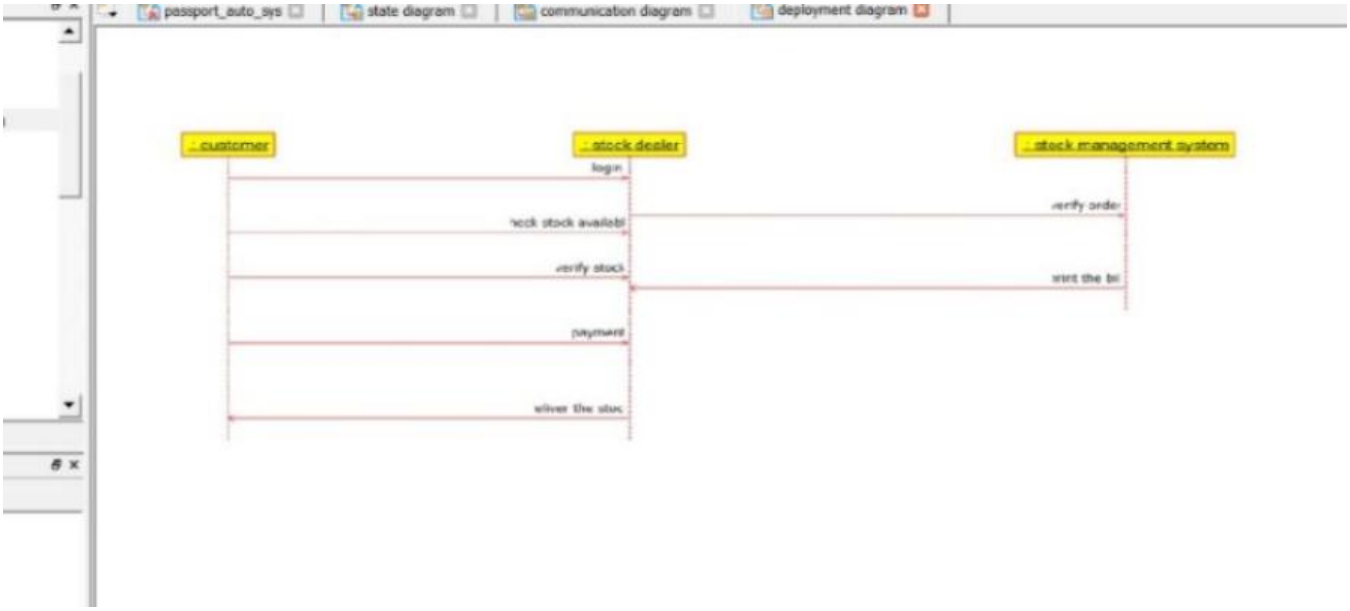
Decision box: Valid or not



SEQUENCE DIAGRAM:

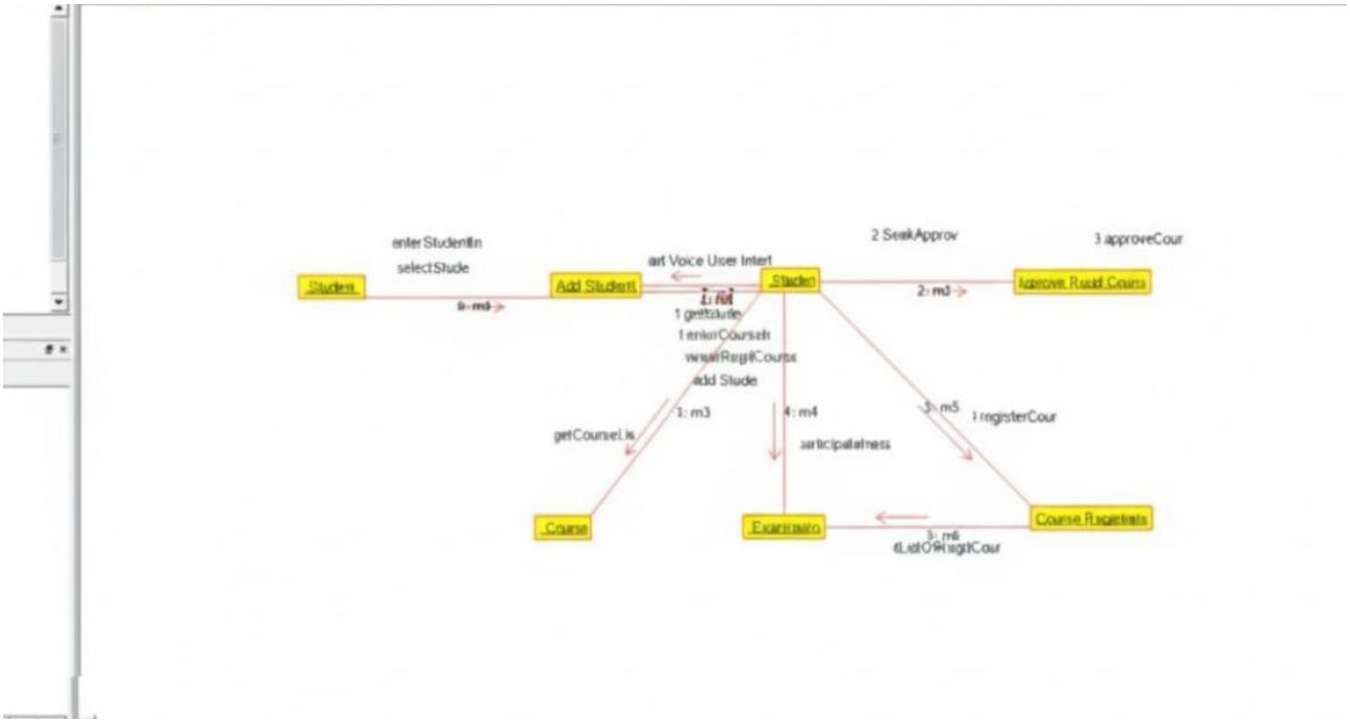
This diagram consists of the objects, messages and return messages.

Object: Customer, Stock dealer, Central stock system



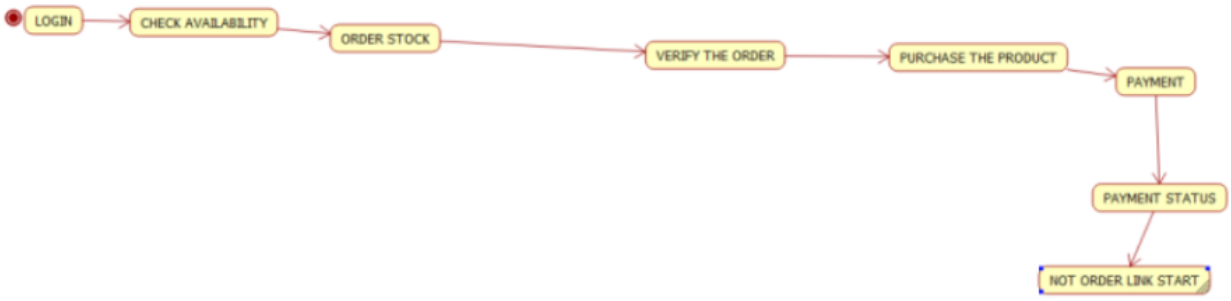
COLLABORATION DIAGRAM:

This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



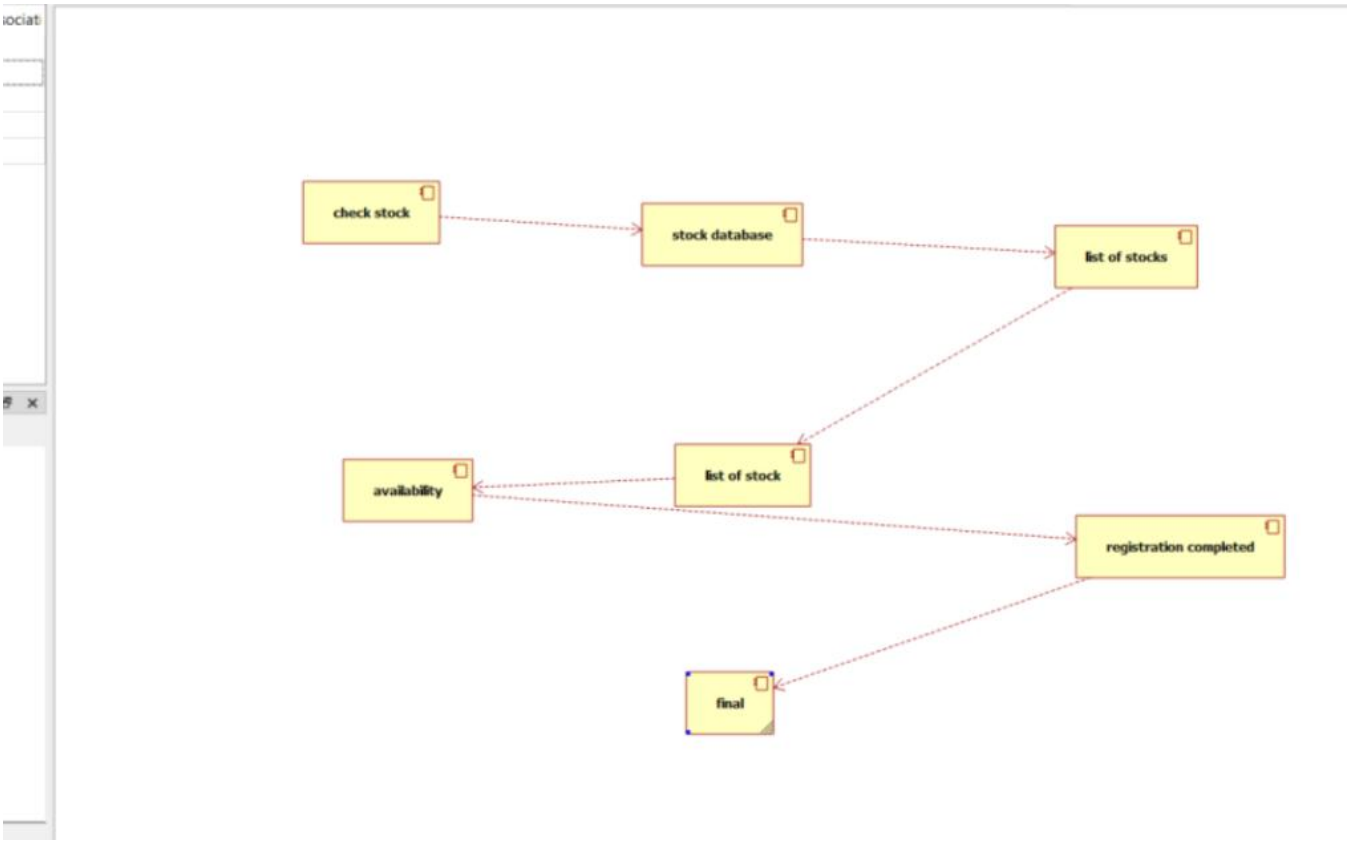
STATE CHART DIAGRAM:

The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle surrounding a small dot.



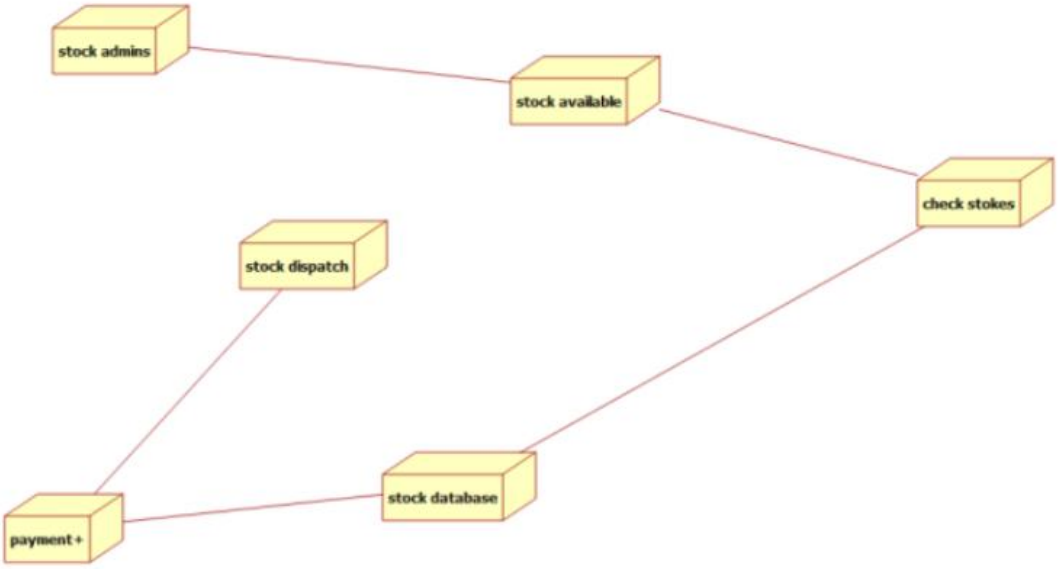
COMPONENT DIAGRAM:

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association



DEPLOYMENT DIAGRAM:

A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3- dimentional box. Dependencies are represented by communication association.

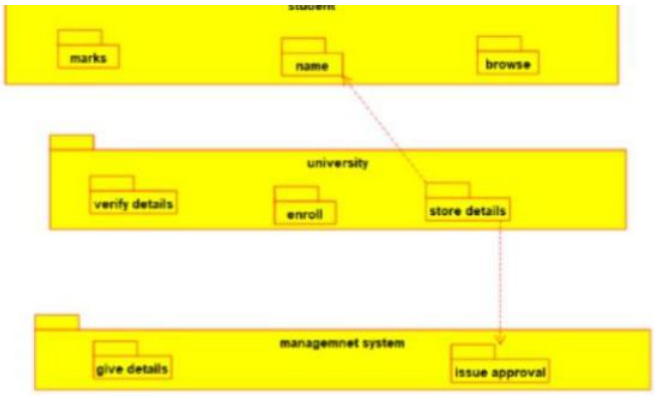


PACKAGE DIAGRAM:

A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

There are three types of layer. They are

- o User interface layer
- o Domain layer
- o Technical services layer



PROGRAM CODING: CENTRAL STOCK SYSTEM:

```
Public central stock system

{

Public integer store stock details; Public void print bill()

{

}

Public void deliver product()
```

{

}

}

CUSTOMER:

Public class customer

{

Public integer place order; Public void payment()

{

}

}

STOCK DEALER:

Public class stock dealer

{

Public integer take order; Pubic integer enter details; Public integer verify details; Public void deliver item()

{

}

}

RESULT:

Thus the diagrams [usecase, activity, sequence, collaboration, class, collaboration, deployment, compoenent, statechart, package] for the Stock maintainence system.