EX NO:7	CREDIT CARD PROCESSING SYSTEM
DATE:23/1/26	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package] for the credit card processing system

SOFTWARE REQUIREMENTS SPECIFICATION:

	SOFTWARE REQUIREMENTS SPECIFICATION
1	Hardware Requirements
2	Software Requirements
3	Problem Analysis and Project Plan
4	Project Description
5	Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.0 PROBLEM ANALYSIS AND PROJECT PLANNING

The Credit Card Processing System which is use to purchasing an item from any shop mall, and it is used to maintain the limitation of credit card balance and current transaction process could be update via credit card machine. This project mainly used for large amount of item can be easy to buy from anywhere and required transaction process should be maintained them.

1.1 PROJECT DESCRIPTION:

This software is designed for supporting the computerized credit card processing System .In this system, the cardholder purchases items and pays bills with the aid of the credit card. The cashier accepts the card and proceeds for the transaction using the central system. The bill is verified and the items are delivered to the cardholder.

1.2 REFERENCES:
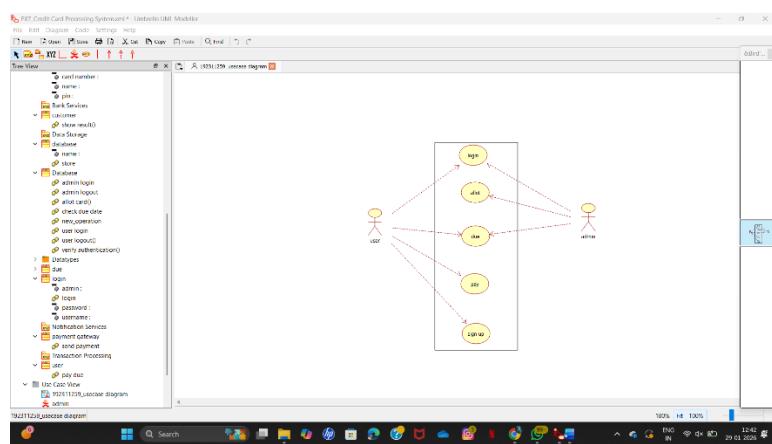
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Cardholder, Cashier, Central system.
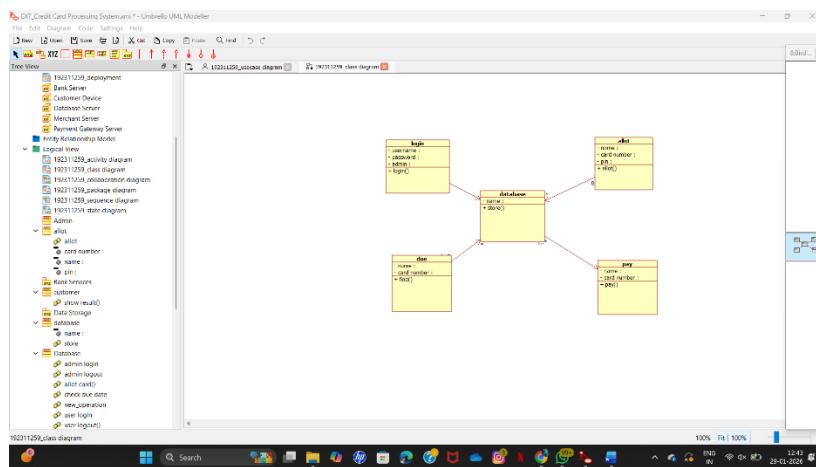
Use case: Receive bill, Give card, Enter card number, Enter amount, Transaction, Receive Receipt.

CLASS DIAGRAM:

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

| CLASSES	ATTRIBUTES	OPERATIONS
Central system	product name, product details	Print bill(), Validate card()
Cashier	Product name, Cost of the product	Enter amount(), Swipe card(), Print bill(), Deliver product()
Card holder	Item purchased, Validate card	Give card(), Sign bill()
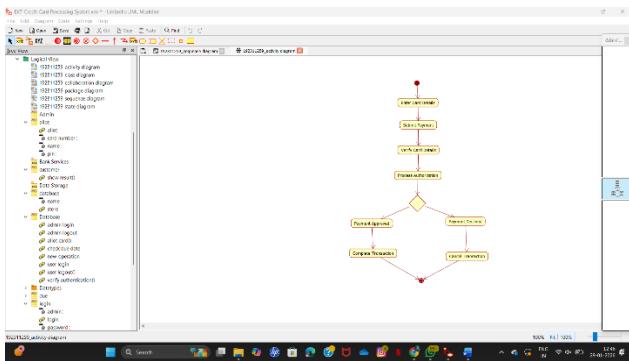
ACTIVITY DIAGRAM:

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

Activities: Receive Bill, Give card, Enter the card number, Enter the amount, Transaction, Receive Receipt
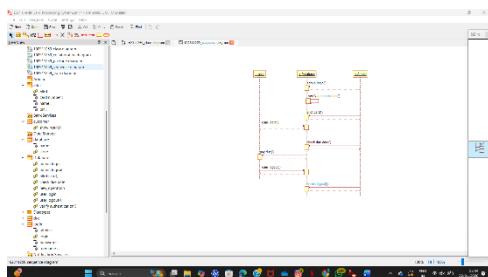
Decision box: Verification of card


SEQUENCE DIAGRAM:
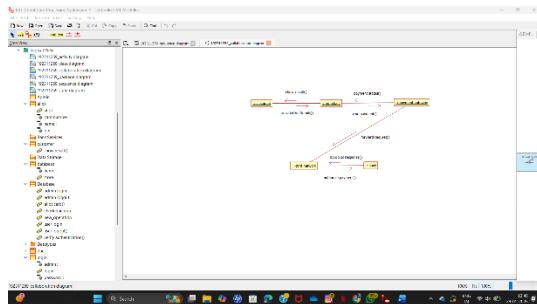
This diagram consists of the objects, messages and return messages.

Object: Card Holder,Cashier , Central system
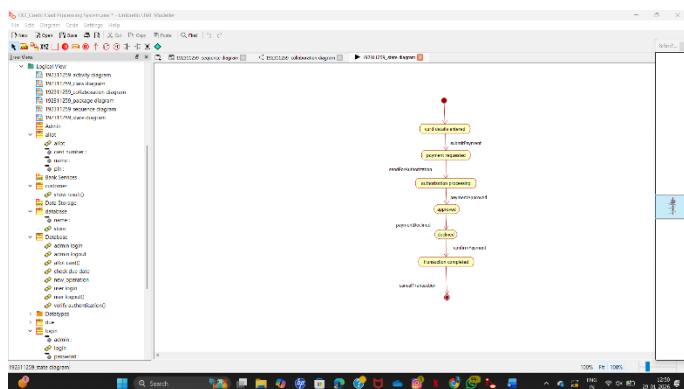
COLLABORATION DIAGRAM:

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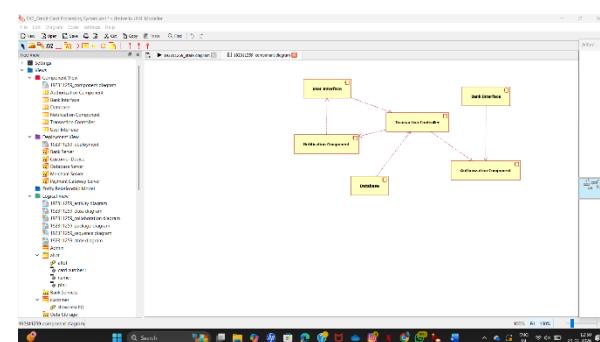
STATE CHART DIAGRAM:

It is a technique to describe the behavior of the system. It describes all the possible states that a particular object gets into the object oriented technique. State diagrams are drawn for a single class to show to the lifetime behaviour of a single objects.
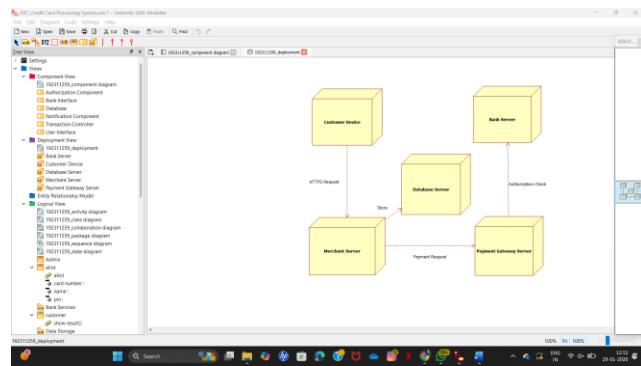
COMPONENT DIAGRAM:

The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a 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 a 3- dimensional box. Dependencies are represented by a communication association.
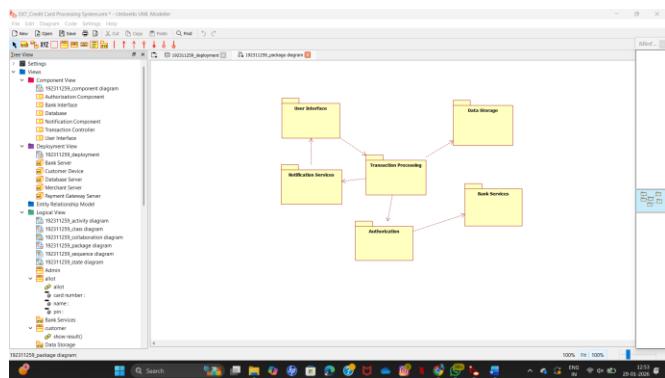

PACKAGE DIAGRAM:

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 layers. They are

1. User interface layer
2. Domain layer
3. Technical services layer

PROGRAM CODING:

CASH HOLDER:

public class cash holder

```
{
    public Integer

    itemspurchased;

    public void signbill()

        {
        }

    }
```

CASHIER:

public class cashier

```
    {

    public Integer

    name; public

    Integer cast;

    public void

    amount()


            {

            }



}


CENTRAL SYSTEM:

public class central sys

    {

    Private


            Integer

            productname;

            publicInteger

            productdetails;

            public Void
```

printbill()

{

}

public void validatecard()

{

}

RESULT:

Thus the diagrams[use case, activity, sequence, collaboration, class, state chart, component, deployment, package] for the credit card processing system has been designed, executed and output is verified.