| | |
|---|---|
| **EX NO:07** | **CREDIT CARD PROCESSING SYSTEM** |
| **DATE:** | |

**AIM:**

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package ] for Credit Card Processing .

**SOFTWARE REQUIREMENTS SPECIFICATION:**

| | **SOFTWARE REQUIREMENTS SPECIFICATION** |
|---|---|
| 1.0 | Hardware Requirements |
| 1.1 | Software Requirements |
| 1.2 | Problem Analysis anc |
| 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 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.

## 1.3 PROJECT DESCRIPTION:

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

## 1.4 REFERENCES:

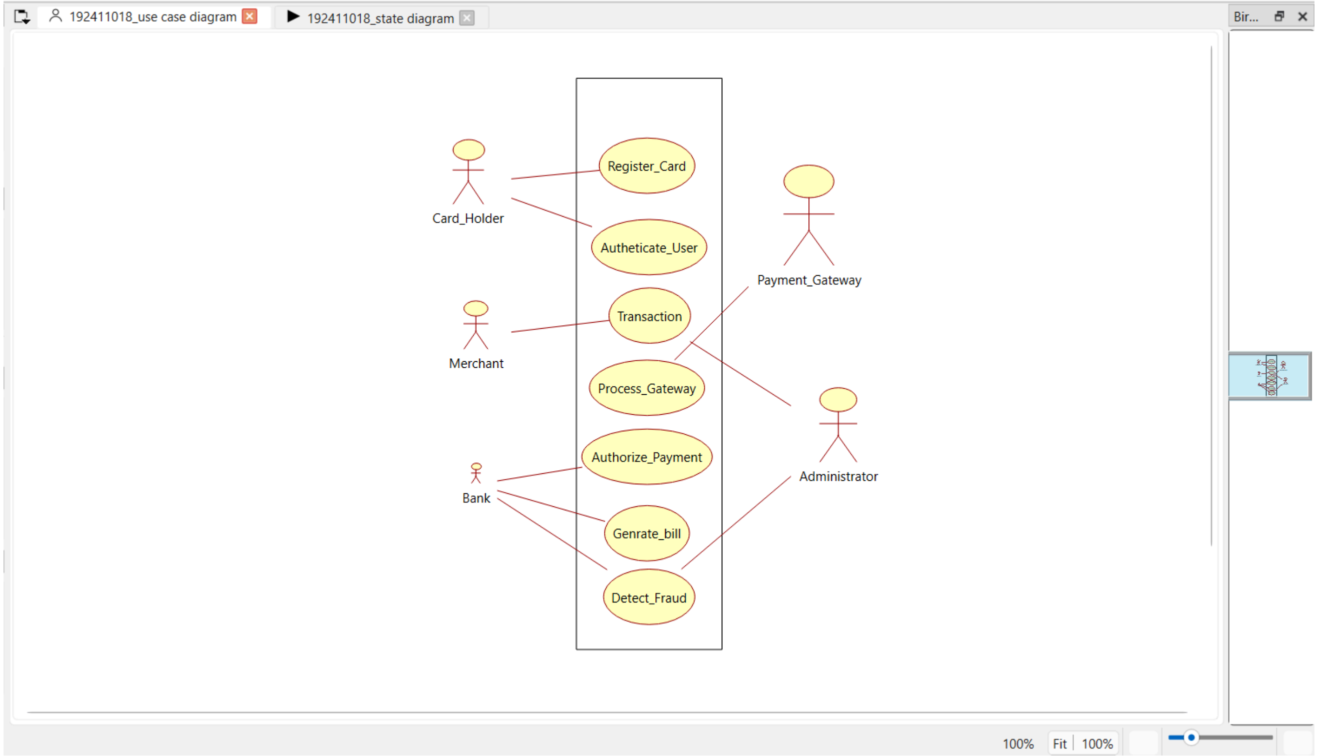
IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

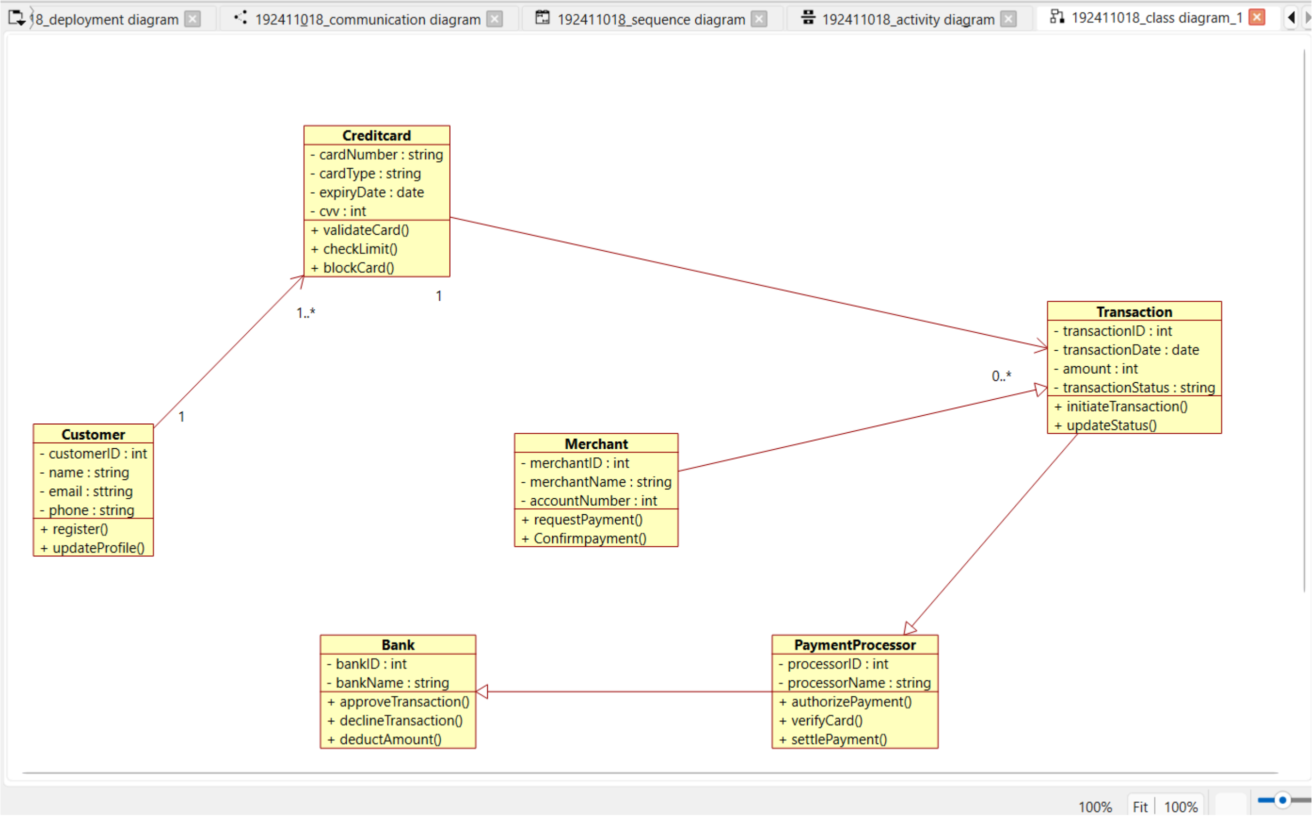
**Actors**: Cardholder, Cashier, Central system.

**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.

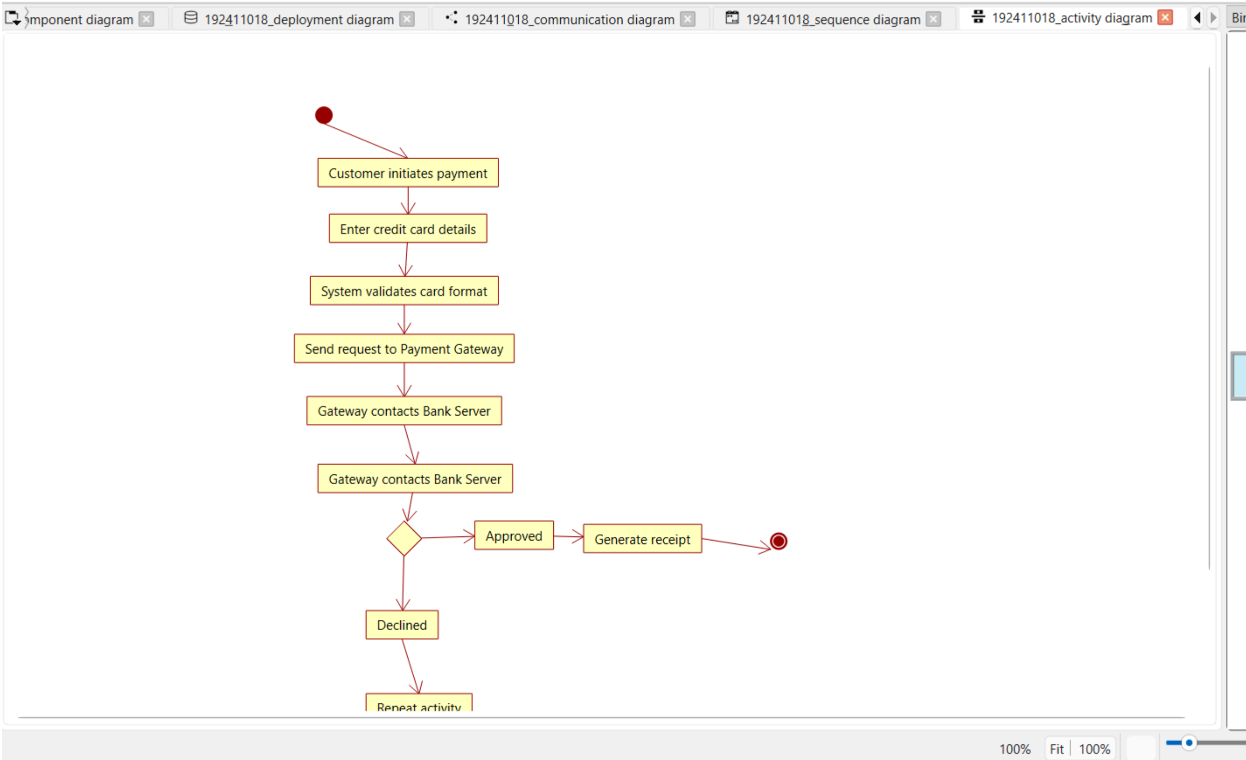
| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central system | product name,<br>product details | Print bill(),<br>Validate card() |
| Cashier | Product name,<br>Cost of the product | Enter amount(),<br>Swipe card(),<br>Print bill(),<br>Deliver product() |
| Card holder | Item purchased,<br>Validate card | Give card(),<br>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

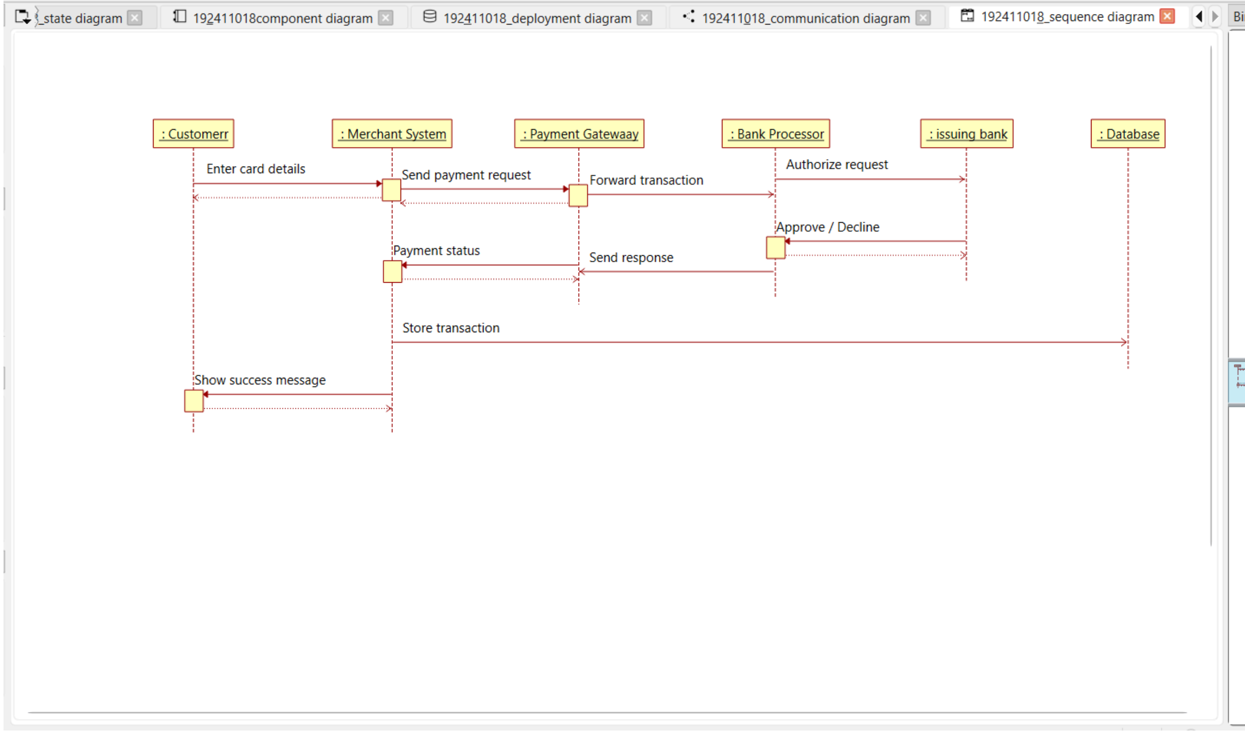


mponent diagram
192411018_deployment diagram
192411018_communication diagram
192411018_sequence diagram
192411018_activity diagram
Bi
Customer initiates payment
Enter credit card details
System validates card format
Send request to Payment Gateway
Gateway contacts Bank Server
Gateway contacts Bank Server
Approved
Generate receipt
Declined
Repeat activity
100%
Fit
100%

## SEQUENCE DIAGRAM:

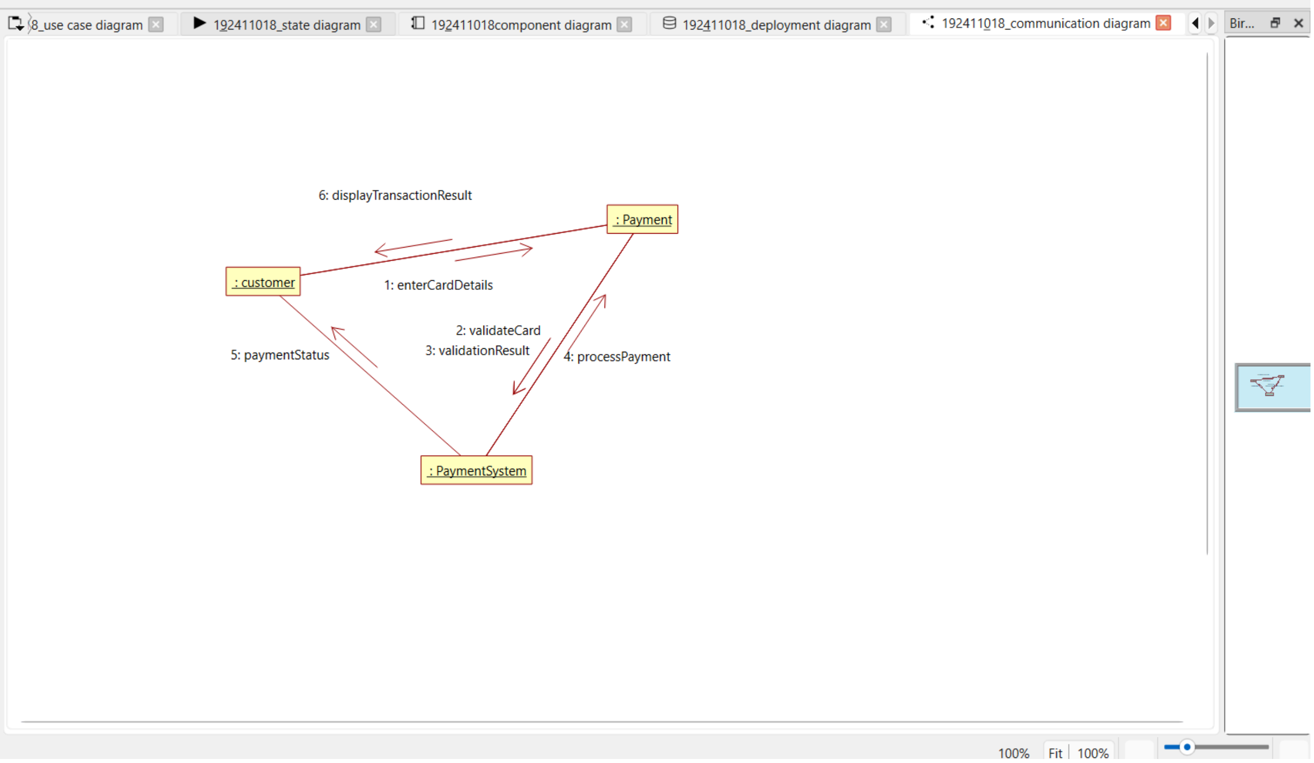
This diagram consists of the objects, messages and return messages.

**Object:** Card Holder,Cashier , Central system



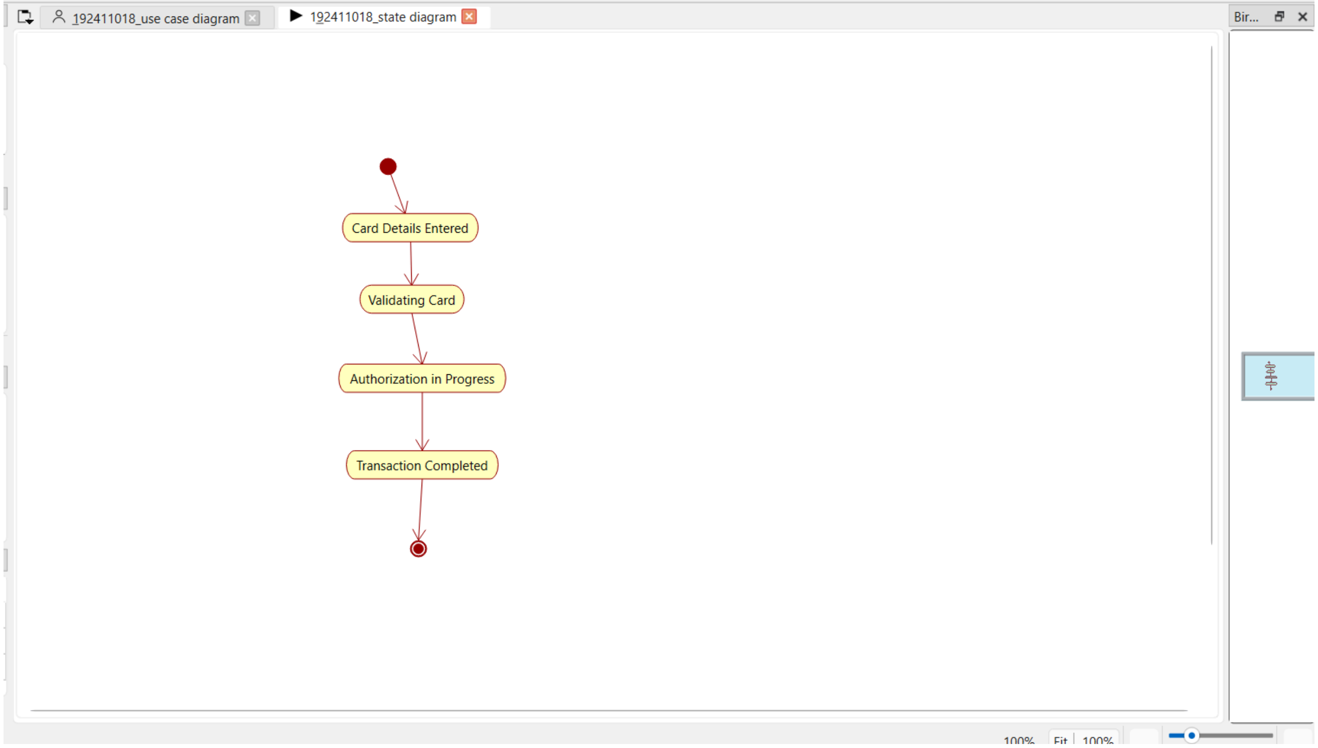
## COLLABORATION DIAGRAM:

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



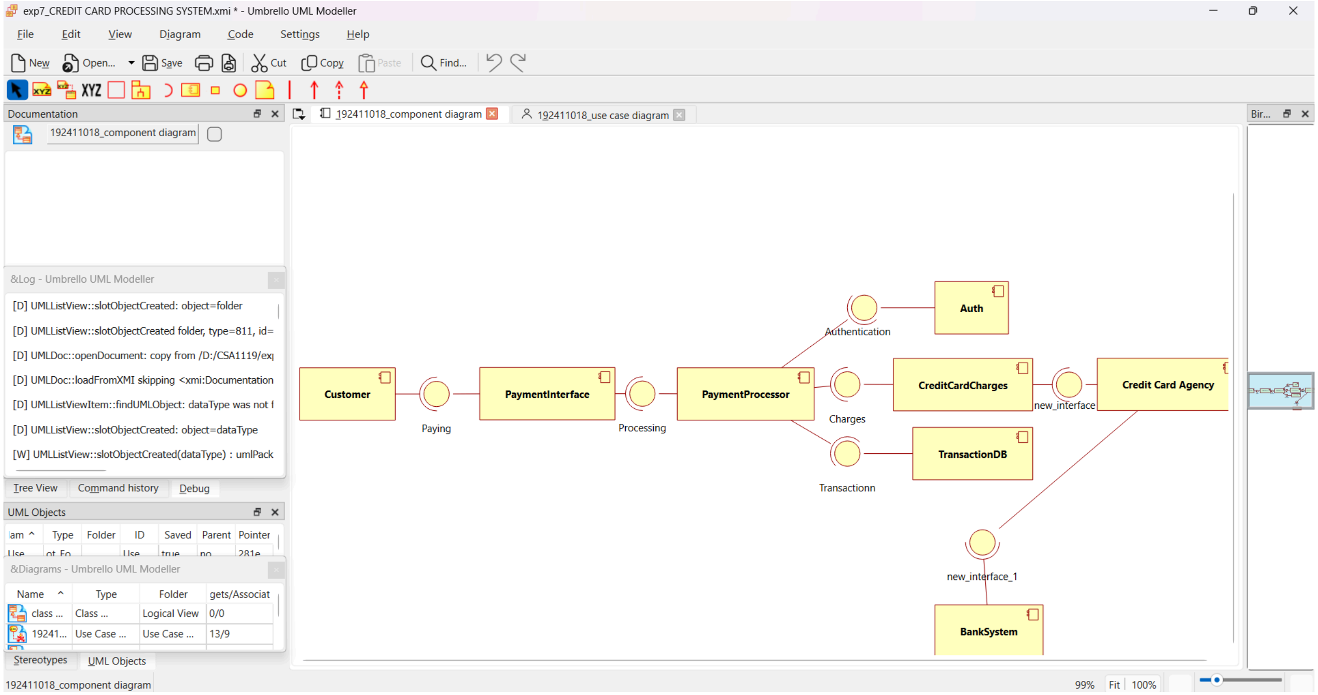
## STATE CHART DIAGRAM:

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 diagram 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 systems. It is represented by boxed figure. Dependencies are represented by communication association



exp7_CREDIT CARD PROCESSING SYSTEM.xmi * - Umbrello UML Modeller

```
[D] UMLListView::slotObjectCreated: object=folder
[D] UMLListView::slotObjectCreated folder, type=811, id=
[D] UMLDoc::openDocument: copy from /D:/CSA1119/ex
[D] UMLDoc::loadFromXMI skipping <xmi:Documentation
[D] UMLListViewItem::findUMLObject: dataType was not f
[D] UMLListView::slotObjectCreated: object=dataType
[W] UMLListView::slotObjectCreated(dataType) : umlPack
```

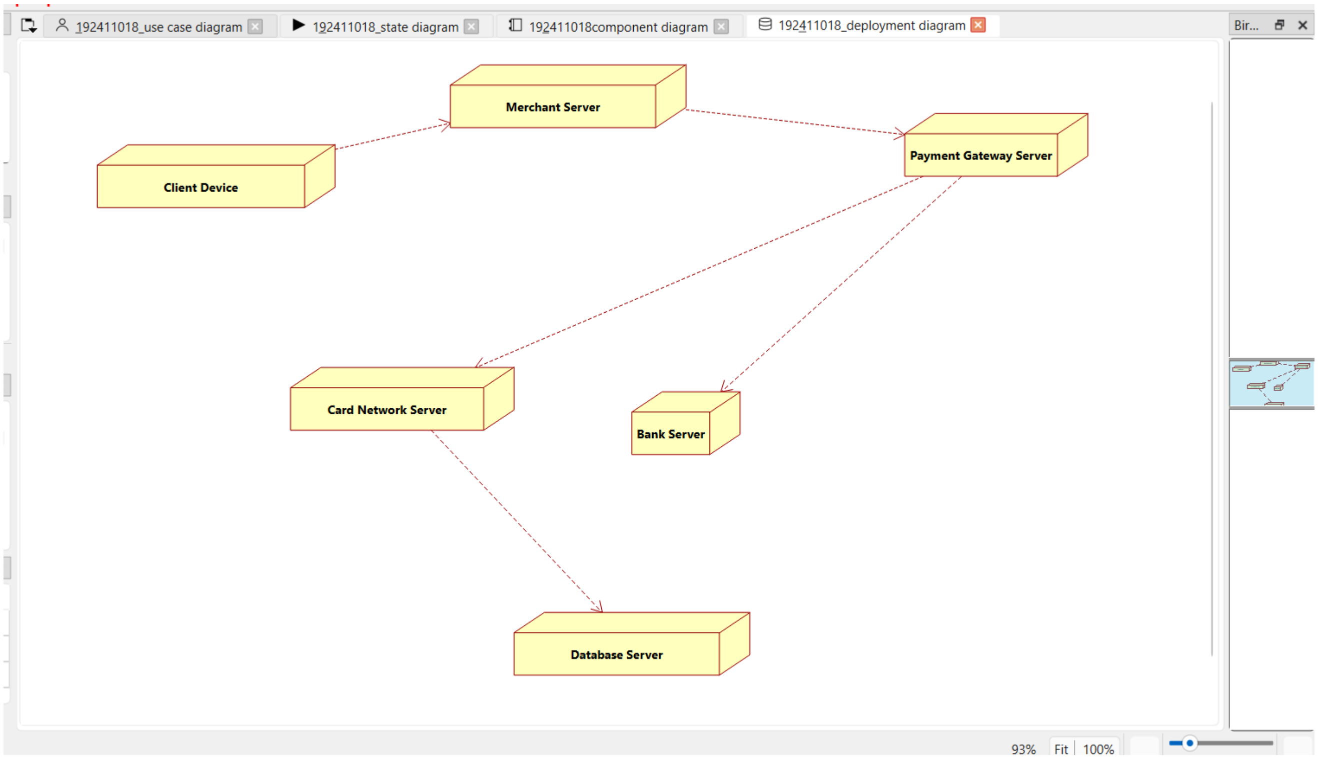
| am ^ | Type | Folder | ID | Saved | Parent | Pointer |
|---|---|---|---|---|---|---|
| Use | ot Fo | | Use | true | no | 281e |

| Name ^ | Type | Folder | gets/Associat |
|---|---|---|---|
| class ... | Class ... | Logical View | 0/0 |
| 19241... | Use Case ... | Use Case ... | 13/9 |

192411018_component diagram

## 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.



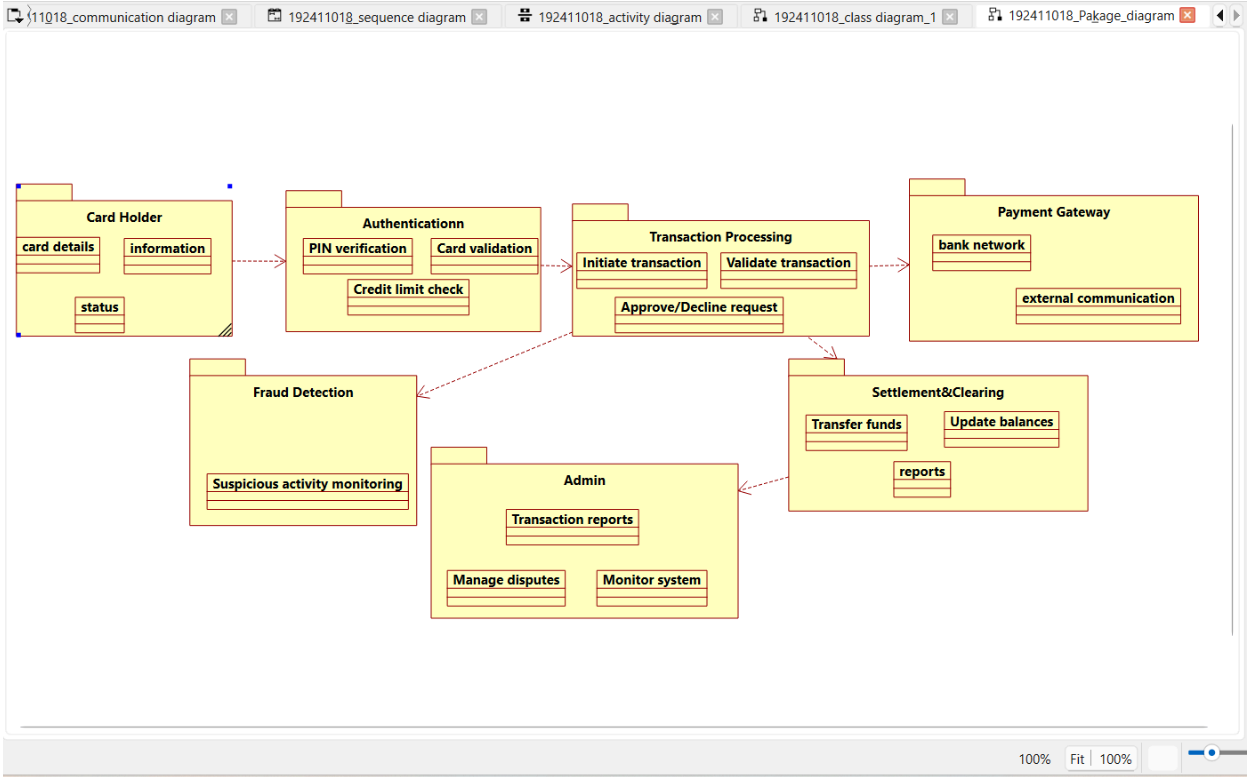
## PACKAGE DIAGRAM:

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:**

**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;
          public Integer 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.

| | |
|---|---|
| **EX:NO**:08 | **SOFTWARE PERSONNEL MANAGEMENT SYSTEM** |
| **DATE:** | |

**AIM:**

To draw the diagrams [ Usecase, Class, Activity, Sequence, Collaboration, State chart, Deployment, Component, package] for software personnel management system.

**SOFTWARE REQUIREMENTS SPECIFICATION:**

| | **SOFTWARE REQUIREMENTS SPECIFICATION** |
|---|---|
| 1.0 | Hardware Requirements |

## 1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

## 1.1 SOFTWARE REQUIREMENTS:

Rational Rose/Argo UML

## 1.2 PROJECT DESCRIPTION:

This software is designed for the process of knowing the details of a person works in a software company. The details are being stored in the Central Management System for the cross checking the person's details.

## 1.3 REFERENCES:

IEEE Software Requirement Specification format.

## USECASE DIAGRAM:

This diagram will contain the actors, usecases which are:

***ACTORS***: Employee, HR, Central System

***USECASE***: Name and address, qualification, experience, internet, loan, verificatio

## CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central Management System | Employee name<br>Employee number | Tax()<br>Loan() |
| Employee1 | Employee details | Leave taken() |
| HR | Check details | Loss of pay() |

## ACTIVITY DIAGRAM:

This diagram will have the activities as start point, end point, decision boxes as:

***ACTIVITIES:*** Enter the option to check, enter the salary, enter the working days, leave taken, loss of pay

***DECISION BOX***: Option to check

## SEQUENCE DIAGRAM:

This diagram consists of the objects, messages and return messages

***OBJECT:*** Employee, HR, Central System

## COLLABORATION DIAGRAM:

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

## STATECHART 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 diagram 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 systems. 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:**

**EMPLOYEE:**

```
Public class employee

 {

        Public integer employee details;

        Public integer salary;

        Public void leave taken()

         {

         }

        Public void employee()

         {

         }

}
```

**HUMAN RESOURCES:**

```
Public class HR

 {

        Public integer check details;

        Public void loss of pay()
```

```
         {

         }

        Public void tax()

         {

         }


        Public void HR()
         {

         }

}
```

**CENTRAL MANAGEMENT**

**SYSTEM:** Public class central

management system {

```
        Public integer employee name;

        Public integer employee no;

        Public integer details;

        Public void leave taken()

         {

         }

        Public void tax()

         {
```

```
         }

        Public void loan()

         {

         }
        Public void salary()
         {
         }
}
```

**RESULT:**

Thus the diagram [ usecase, class, activity, sequence, collaboration, state chart, component, deployment, package] for the Software Personnel Management System has been designed, executed and output is verified.

| | |
|---|---|
| **EX.NO:09** | **E-BOOK MANAGEMENT SYSTEM** |
| **DATE:** | |

**AIM:**

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package ] for E-book management system.

**SOFTWARE REQUIREMENTS SPECIFICATION**

| | **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

E-book Management System gives an idea about how books are maintained in the particular websites. The books that are to be purchased, the books that are to be sold are maintained here. . Further some additional details of the current books that is available in the store are also given. E book Management System in this project is done in an authorized way.
The password and user id has been set here.

## 1.3 PROJECT DESCRIPTION:

This software is designed to manage the books that were read through the internet. This consists of the details of the e-book that were read by the user online. It will be controlled by the central system. This system act as a backup of all details together.

## 1.4 REFERENCES:

IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

**Actors**: user, e-book management

**Use case**: login ,search books, download ,pay for the books, logout .

## CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Internet | Enter id, login, logout | Surf book() |
| User | Login, logout | Surf book() |
| E-book management system | verify user | check availability() |

## ACTIVITY DIAGRAM:

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

**Activities**: Search for the e-book site,search for the book,download book

**Decision box:** check availability

## SEQUENCE DIAGRAM:

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

**Object:** User ,E-book management ,Internet

## 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.

## STATECHART 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 diagram 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 systems. 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:

**E-BOOK MANAGEMENT:**

```
 public class e-book management extends internet
  {

             public Integer enterid;
```

```
            public Integer verifyuser;

            public void
             checkavailability() {
             }
}
```

**USER:**

```
public class user1 extends internet
 {
            public Integer login;

             public Integer logout;

            public void surfbooks()
              {
             }

}
```

**CENTRAL SYSTEM:**

```
public class central system
 {

             public Integer enterid;

             public Integer download;

             public Integer login;

              public Integer logout;

             public void verify()
             {
             }
```

```
        public void status()
         {
         }

}
```

**RESULT**:

Thus the draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package ] for E-book management system.

| | |
|---|---|
| **EX:NO:10** | **RECRUITMENT SYSTEM** |
| **DATE:** | |

**AIM:**

To draw the diagram[Usecase, Activity, Sequence, Collaboration, Class, StateChart, Component and Deployment, package] for recruitment system.

**SOFTWARE REQUIREMENTS SPECIFICATION:**

| | **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

## 1.3 PROJECT DESCRIPTION:

This system is designed to recruit the particular job to the person in a company .It was controlled by the central management system to manage the details of the particular candidate that one has to be recruited for a company.

## 1.4 REFERENCES:

IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

**Actors**: Applicant, HR, Central management system.

**Use case**: Aptitude, Group discussion, Technical skills, Personal specification, Short list, Result

## CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Candidate | Name, qualification | Verify() |
| HR | Verification, resume | Select() |
| Central system | Store, update | Update() |

## ACTIVITY DIAGRAM:

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

**Activities**: Aptitude, Group discussion ,Technical skills,HR

**Decision box:** Verification of the qualities

## SEQUENCE DIAGRAM:

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

**Object:** Candidate, HR, 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 diagram 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 systems. 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 SYSTEM:

```
Public class central system

 {
        Public integer store;

        Public integer update;

        Public void central management system()

         {

         }

}
```

**HR:**

```
Public class HR

  {

        Public integer verification;
        Public integer resume;

        Public void HR()

         {

         }
```

```
}
```

**CANDIDATE:**

```
Public class candidate

 {

        Public integer name;

        Public integer operation;

        Public integer qualification;

        Public void verify()

         {

         }

        Public void candidate()

         {
        }
}
```

**RESULT:**

To draw the diagram [Use case, Activity, Sequence, Collaboration, Class, State Chart, Component and Deployment, package] for recruitment system has been designed and output is verified.

| | |
|---|---|
| **EX:NO:11** | **CONFERENCE MANAGEMENT SYSTEM** |
| **DATE:** | |

**AIM:**

To draw the diagrams [use case, activity, sequence, collaboration, class, component, deployment, package] for Conference management system

**SOFTWARE REQUIREMENTS SPECIFICATION**

| | **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 Conference Management System is an online website in which candidate can submit the paper and register themselves and then attend the conference. The paper will be reviewed. The details of the conference, date and time will be made available to them through the website. After getting the confirmation details the candidate should submit the revised and camera ready paper. Then the registration process will be done.

## 1.3 PROJECT DESCRIPTION:

This software is designed to manage the details of the process that will be taken place in the conference in a place. It works along with the organizer, who arranges all these program and central management system, which consists of the all the details of the member who participates in the presentation

## 1.4 REFERENCES:

IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

**Actors**: Member, Organizer, Central system

**Use case**: planning, invite delegates, allocate seats, presenting paper, prize distribution

## CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Member | Name, id | Presenting paper() |
| Organizer | Member details | Allocating seats() |
| Central management system | Member details | Updating() |

## ACTIVITY DIAGRAM:

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

**Activities**: Invite delegates, Allocate seats, Presenting paper, Choose the winner

**Decision box:** Whether it is reserved or not, Whether the presentation is good or not

## SEQUENCE DIAGRAM:

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

**Object:** Member, Organiser, Central management 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.

## 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 systems. 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:

### MEMBER 1:

```
public class member
 {
        public integer name;
        public integer id;

        public integer proof;

        public void winning prize()

         {

         }

        public void member()

         {

         }

}
```

**ORGANIZER:**

```
public class organizer

 {
        public integer member

        attributes; public integer

        function details;

        public void choosing for

         winner() {

         }

}
```

**CENTRAL MANAGEMENT**

**SYSTEM:** public class central

management system {

```
        public integer function details;

        public integer detail of seat allocation;

        public void storing()
        {
        }
        public void updating details()
        {
        }
}
```

**RESULT:**

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

**DATE:**

**EX:NO:12**

## FOREIGN TRADING SYSTEM

**AIM**:

To draw the diagrams [Use case, Activity, Sequence, Collaboration, Class, State chart, Component, Deployment, package] for foreign trading system

**SOFTWARE REQUIREMENTS SPECIFICATION**

| | **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 PROJECT ANALYSIS AND PROJECT PLANNING

The initial requirements to develop the project about the mechanism of the Foreign Trading System is bought from the trader. The requirements are analyzed and refined which enables the analyst (administrator) to efficiently use the Foreign Trading System. The complete project analysis 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 to maintain the details about the trading system that exists between the foreign countries. These details are hold by the trading management system. The details to the system are provided by the customer and the supplier.

## 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, Supplier, Custom officer

**Use case**: Order of product, Quantity, Specify the amount

## CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Trading management system | Verify product | Transport() |
| Customer | Quality | Payment() |
| Supplier | Product supply | Money transfer() |

## ACTIVITY DIAGRAM:

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

**Activities**: Order of the product, Specify amount, Payment, Money transfer
**Decision box:** Check for availability

## SEQUENCE DIAGRAM:

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

**Object:** Customer, Supplier, Trading management 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.

## STATECHART 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 diagram 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 systems. 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

- User interface layer
- Domain layer
- Technical services layer

## PROGRAM CODING:

### TRADING MANAGEMENT SYSTEM:

```
public class trading management system
 {
          public integer verify product;
          public integer amount;
          public void transport()
           {
           }
          public void money transfer()
           {
           }
}
```

### CUSTOMER:

```
public class customer
```

```
 {
          Public integer order product;
          Public integer amount;
          Public void payment()
           {
           }
          Public void delivery()
           {
           }
 }
```

**SUPPLIER:**

```
Public class supplier
 {
          Public integer supply;
          Public void available product()
           }
```

**RESULT:**

Thus thethe diagrams [Usecase, Activity, Sequence, Collaboration, Class, Statechart, Component, Deployment, package ] for foreign trading system has been designed, executed and output is verified.

| | |
|---|---|
| **Ex.no** 13 | **BPO MANAGEMENT SYSTEM** |
| **Date:** | |

**AIM:**

To draw the diagrams [ Use case , Class, Activity, Sequence, Collaboration, State Chart, Component, Deployment, package ] for the BPO Management System .

## SOFTWARE REQUIREMENTS SPECIFICATION

### 1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

### 1.1 SOFTWARE REQUIREMENTS:

Rational rose /Argo UML

### 1.3 PROJECT DESCRIPTION:

This software is designed to know about the process that were taking place in the BPO office. This system holds the details of the customer who and all approaches to it.It is managed by the central systems.

### 1.4 REFERENCES:

IEEE Software Requirement Specification format.

### USECASE DIAGRAM:

This diagram will contain the attributes as start point, end point, decision box as given below

***ACTORS:*** Purchase product, Server, Central system

***USECASE***: Product, Voice, Non-Voice, Indian office, Employee, Feedback.

### CLASS DIAGRAM:

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

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central System | Store, update | Storing(), updating() |
| Dealer | Employee name | Delivery() |
| Customer | Details | Feedback() |

## ACTIVITY DIAGRAM:

This diagram will contain the activities as start point, end point, decision boxes as given below

***ACTIVITIES***: Purchase Product, On call, On chat

***DECISION BOX:*** Option to check

## SEQUENCE DIAGRAM:

This diagram consists of the objects, messages and return messages

***Object:*** Customer, Dealer, Central System

## COLLABORATION DIAGRAM:

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

## STATECHART 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 diagram 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 systems. 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 associatio

## 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 SYSTEM*:*

```
import java.util.Vector;

public class central system

  {

            public Integer store;

             public Integer update;

             public Vector mydealer;

             public void updating()

             {

             }
```

```
        public void processing()
          {
         }
 }
```

**CUSTOMER:**

```
import java.util.Vector;
public class customer
  {
        public Integer name;
        private Integer product;
        public Vector mydealer;
         public void feedback()
         {
          }
         public void customer()
         {
          }
 }
```

**DEALER:**

```
import java.util.Vector;
public class dealer
  {
        public Integer employeename;
        public Integer availability;
```

```
            public Integer newAttr;

             public Vector mycustomer;

             public Vector mycentral system;

            public void payment()

              {

             }

             public void delivery()
```

```
              {

             }

}
```

**RESULT:**

The diagrams [Use case, Class, Activity, Sequence, Collaboration, State Chart, Component, Deployment, package] for the BPO Management system has been designed, executed and output is verified.

OOAD LAB