Connects Life & Learning

**SRI SHANMUGHA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(APPROVED BY AICTE, NEW DELHI & AFFILIATED TO ANNA UNIVERSITYAND ACCREDITED BY NAAC & NBA(ECE,CSE,MECH)**

**Tiruchengode-Sankari Main Road, Pullipalayam, Morur(Po), Sankari (Tk), Salem (Dt) Pin: 637 304**

# RECORD NOTE BOOK

## (CS8582)

**OBJECT ORIENDED ANALYSIS AND DESIGN LABORATORY**

**NAME :**

**REG NO :**

**YEAR :**

Connects Life & Learning

**SRI SHANMUGHA COLLEGE OF ENGINEERING AND TECHNOLOGY**

(APPROVED BY AICTE, NEW DELHI & AFFILIATED TO ANNA UNIVERSITYAND ACCREDITED BY NAAC & NBA(ECE,CSE,MECH)

Tiruchengode-Sankari Main Road, Pullipalayam, Morur(Po), Sankari (Tk), Salem (Dt) Pin: 637 304

# RECORD NOTE BOOK

REG NO. □

Certified that this is a bonafide record of Practical work done by

Mr/Ms …………………………………………. of the ……………………

Semester ……………….. Branch during the Academic year …………

in the ……………………………………………… Laboratory

Staff-in-charge Head of the Department

Submitted for the Anna University Practical

Examination held on……………..

Internal Examiner External Examiner

# CS8582-OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

DEGREE / BRANCH**: B.E / CSE** YEAR / SEM**: III /V**

## TABLE OF CONTENTS

| S.NO | DATE | NAME OF THE EXPERIMENT | PAGE NO | SIGNATURE |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| S.NO | DATE | NAME OF THE EXPERIMENT | PAGE NO | SIGNATURE |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# PASSPORT AUTOMATION SYSTEM

**EX NO : 1**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the PASSPORT AUTOMATION SYSTEM with UML diagrams using Object Oriented Designing Methodology.

## PROBLEM ANALYSIS:

### Problem planning and Program Analysis:

This software is designed for the verification of the passport details of the applicant by the central computer. The details regarding the passport will be provided to the central computer and the computer will verify the details of applicant and provide approval to the office. Then the passport will issue from the office to the applicant.

## SOFTWARE REQUIREMENTS SPECIFICATION (SRS):

### Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| | 1.0 | Analysis on requirements | CSE |
| | 1.0 | Creation of documentation | CSE |
| | 1.0 | Designing | CSE |
| | 1.0 | Coding | CSE |
| | 1.0 | Implementation of System | CSE |
| | 1.0 | Testing of system | CSE |

# SOFTWARE REQUIREMENTS SPECIFICATION

## 1. Introduction

### 1.1 Purpose

This document is meant to delineate the features of OSM, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

### 1.2 Scope

The purpose of passport automation system is to verify the details of applicant and to issue passport. It interacts with the central system to carry out the verification details of the applicant and provide approval to the office.

### 1.3 Definitions, Acronyms and Abbreviations

### 1.4 References

IEEE SRS Format.

### 1.4 Technologies to be used

Microsoft Visual Basics 6.0 &
Microsoft Access
Rational Rose tool
Eclipse IDE

### 1.6 Overview

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

## 2. Overall Description

This section of the Modern SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in section 3, and makes them easier to understand. Include such items as product perspective, product functions, user characteristics, constraints, assumptions and dependencies, and requirements subsets.

### 2.1 Use-Case Model Survey

This section contains an overview of the use-case model or the subset of the use-case model that is applicable for this subsystem or feature. This includes a list of names and brief descriptions of all use cases and actors, along with applicable diagrams and relationships. This section describes the use-case model comprehensively, in terms of how the model is structured into packages and what use cases and actors there are in the model. If you are using packages, the document shows the model structure hierarchically.

- Actor:

  An Actor represent a set of rules that user of a case play when interacting with the usecases. Actor identifies the Applicant, Enquiry officer.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

  The usecase here it describes:

  - Applicant enters the details. Enquiry officer verifies the detail and issue the passport.
  - Required operations are performed

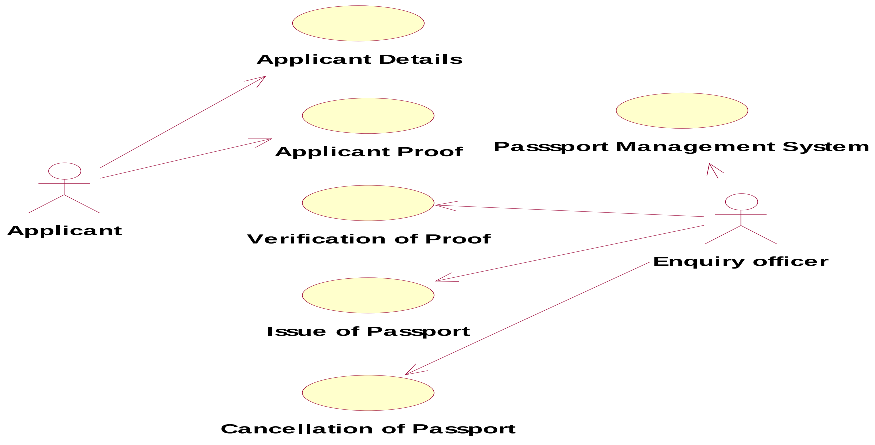
## 2.2 Assumptions and Dependencies

We assume that the system is not very focused on security and there is another system is used as a forum in the system. All the information inputted by the operator is well enough to use.

## 3. Specific Requirements

This section of the Modern SRS contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, the majority of these requirements are captured in the use cases.

## 3.1 Use-Case Specifications



In use-case modeling, the use cases often define the majority of the functional requirements of the system, along with some non-functional requirements. For each use case in the above use-case model, or subset thereof, enclose the use-case specification here.

**Actors:**

- Applicant
- Enquiry Officer

**Use cases:**

- Applicant details
- Applicant proof
- Verification of proof
- Issue of passport
- Cancellation of passport

## REQUIREMENTS:

### FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software systemon its component. A function is described as a set of input,the behaviour and output.

1. **REQUIREMENTS**

   The system should have the requirements of the project. The developer should prepare the requirements of the project. The should prepare the requirements which are need for the software.

2. **ANALYSIS**

   Analyze the requirements whether it provides properoperations/output and performs the task.

3. **DESIGN**

   Project manager should design the layout of the project before going to implement time allocation,cost allocation and staff allocation wil coming under design process.

4. **IMPLEMENTATION**

   After encomposing all the diagrams,we have to generate code for each and every diagrams i.e from usecase to deployment.

5. **TESTING**

   After implementing the diagram with domain language,we haveto test the particular projects.

6. **MAINTAINENCE**

   The system should be easily updated.The system should maintain the cost andtime schedule of the project.

## NON FUNCTIONAL REQUIREMENTS

Nonfunctional requirements define the needs in terms if performance,logical database requirements, design constraints, standard compliance, reliability,availability, security, maintainability and portability.

## SYSTEM SPECIFICATION:

### Software requirements:

Operating System: Windows XP

Front end: Rational Rose Enterprise suite

Back end: Microsoft Access

### Hardware requirements:

Processor: Intel Pentium @3.06 GHZ

RAM: 512 MB DDR

Hard disk: 80 GB SATA

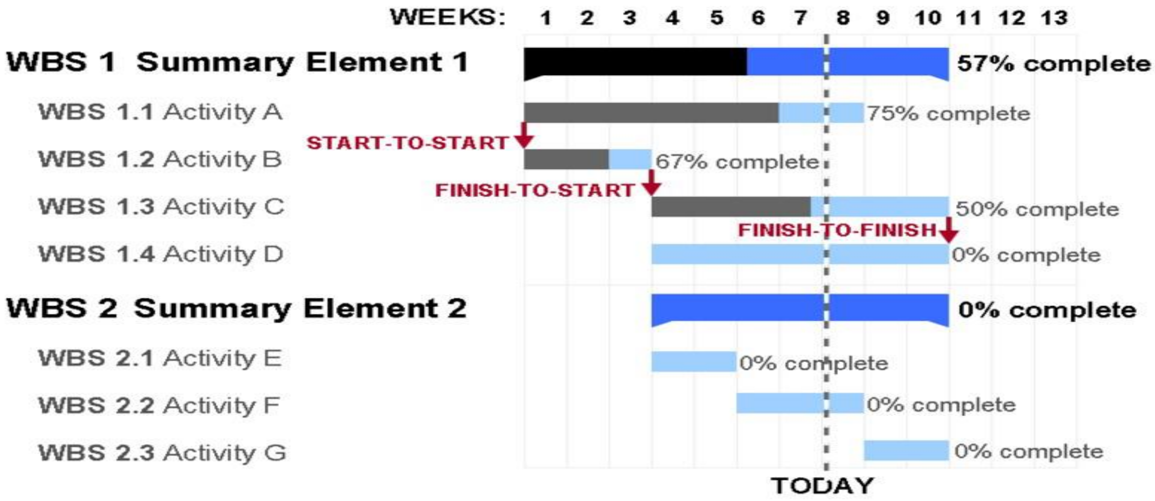
Monitor: 15’’ TFT

Keyboard: Multimedia Keyboard

Mouse: USB Optical type

## GANTT CHART :

A Gantt Chart is a type of bar chart that illustrate a project schedule.Gantt chart illustrate the start and finish dates of the terminal elements and summary elements of a project.Terminal elements and summary elements comprise the work breakdown structure of the project.Some Gantt charts also show the dependency(i.e., precedence network)relationships between activities.Gantt charts can be used to show current schedule status using percent complete shadings and a vertical “TODAY” line as shown here.

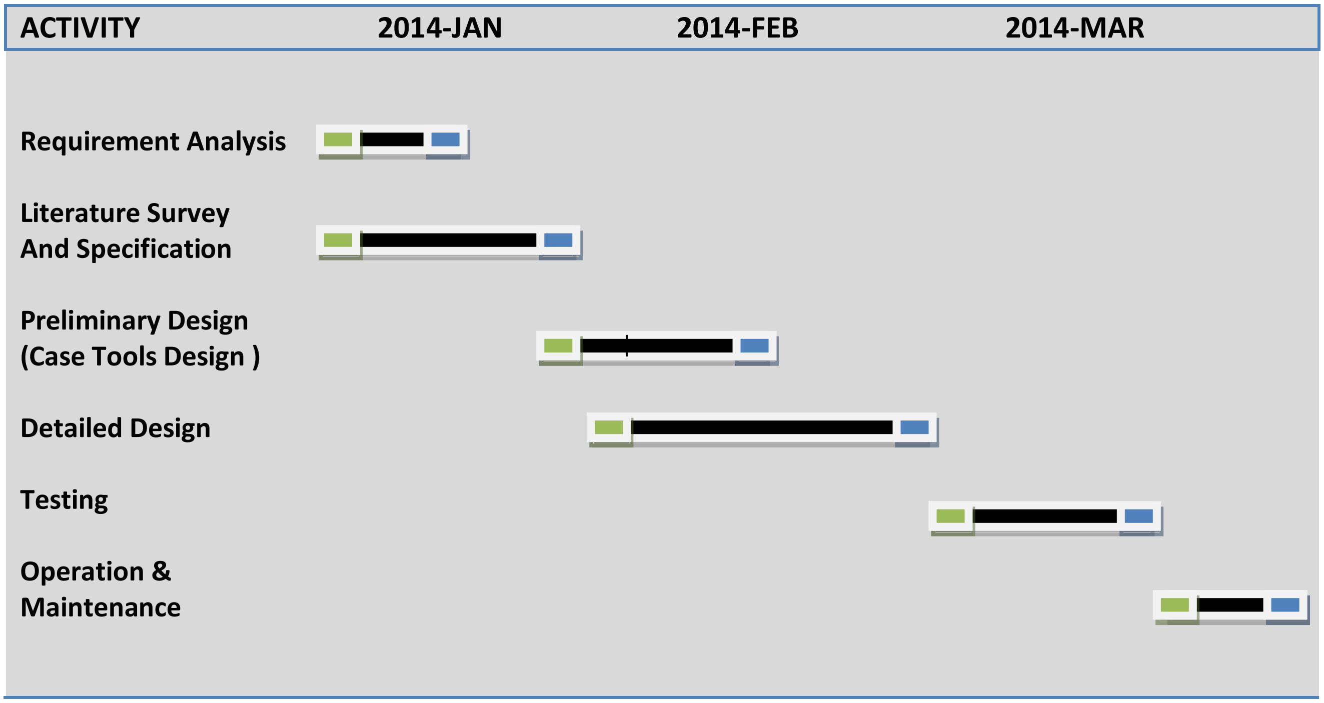


**Passport Automation System :**

For the project, the activities considered are :

1. Requirement Analysis
2. Literature Survey and Specification
3. Preliminary Design
4. Detailed Design
5. Testing
6. Operation and Maintenance

## GANTT CHART FOR PASSPORT AUTOMATION SYSTEM



## DOMAIN MODEL:

A domain model is a visual representation of conceptual classes or real situations object in a domain.

- In object oriented analysis, the domain model is the mostimportant.
- It illustrates the concept in the domain.
- It act as a source of inspiration for designing some softwareobjects.

## PARTIAL LAYERED ARCHITECTURE:

Sequence diagram is an interaction overview diagram. It provides a bigpicture overview of how a set of interaction are related in terms of logic and processflow.

This Partial layer architecture shows the interface of the sequencediagram, here the administrator shows the interface by displaying actor symbol.

## LOGICAL ARCHITECTURE:

The Logical architecture is large scale organization of software class intopackages, sub system layer, It is called logical architecture because there is no directionabout how these elements are display on different operating system.

## RISK ACTIVITY:

- Personal short falls.
- Unrealistic schedules and budgets.
- Developing the wrong functions and programs.
- Developing the wrong user interface.
- Continuing stream of requirements changes.
- Short falls in externally furnished component.
- Real time performance short falls.
- Straining computer science capabilities.

## RISK ASSESSMENT:

- Figure out what the risks are and what to focus on.
- Making a list of all of the potential dangers that will affect the project.
- Assessing the probability of occurrence and potential loss of each item listed.

## RISK CONTROL:

- Monitoring the effectiveness of the strategies and the changing levels of risk throughout the project.
- Techniques and strategies to migrate the highest ordered risks.

## TECHNICAL SERVIVES LAYER:

Technical service layer describes the relationship between different actors,components of the software process for any admin seek the registration for new visitor.So that the new visitor can login the website and search for book and if need can buy it.

## DOMAIN SERVIVES LAYER:

After technical services layer from partial layered architecture, we aregoing to generate the coding in java or VB the project domain is now finalized inJAVA/VB.

## USER INTERFACE LAYER:

- Passport management system
- Verification details

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

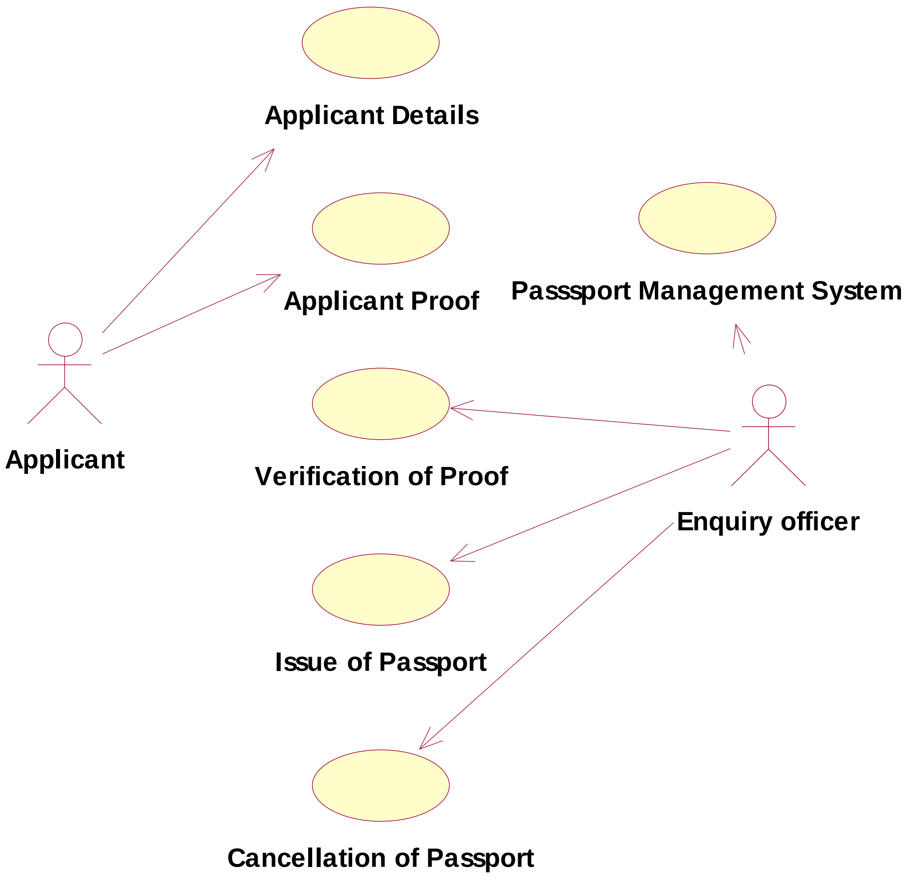
- Actor:

  An Actor represent a set of rules that user of a case play when interacting with the usecases..
- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Applicant, Enquiry officer.

**Usecase**:Applicant details, Applicant proof, Verification of proof, Issue of passport,Cancellation of the passport.



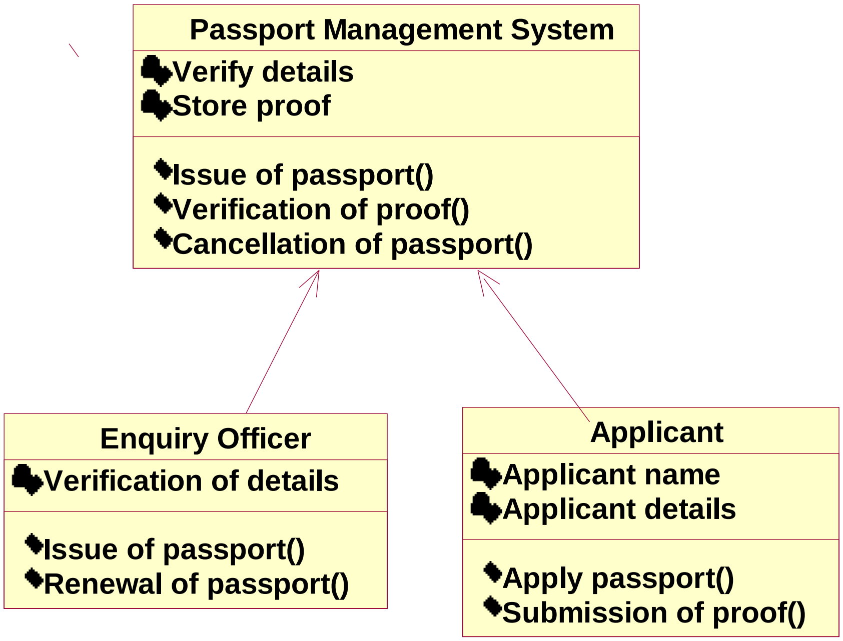
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Passport management system | Verify details, store proof | Verification of proof() |
| Enquiry officer | Applicant Details | Issue of passport() |
| Applicant | Name, details | Apply passport() |

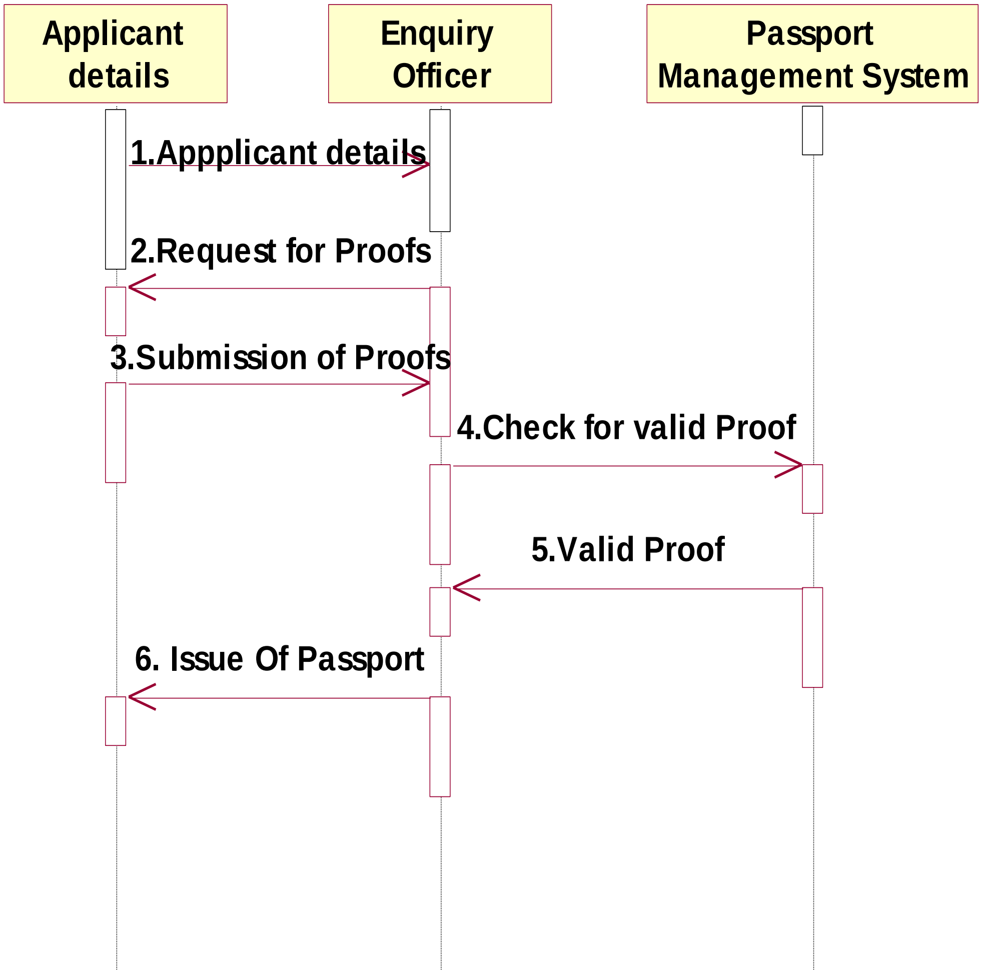


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Applicant, Enquiry officer, Passport management system



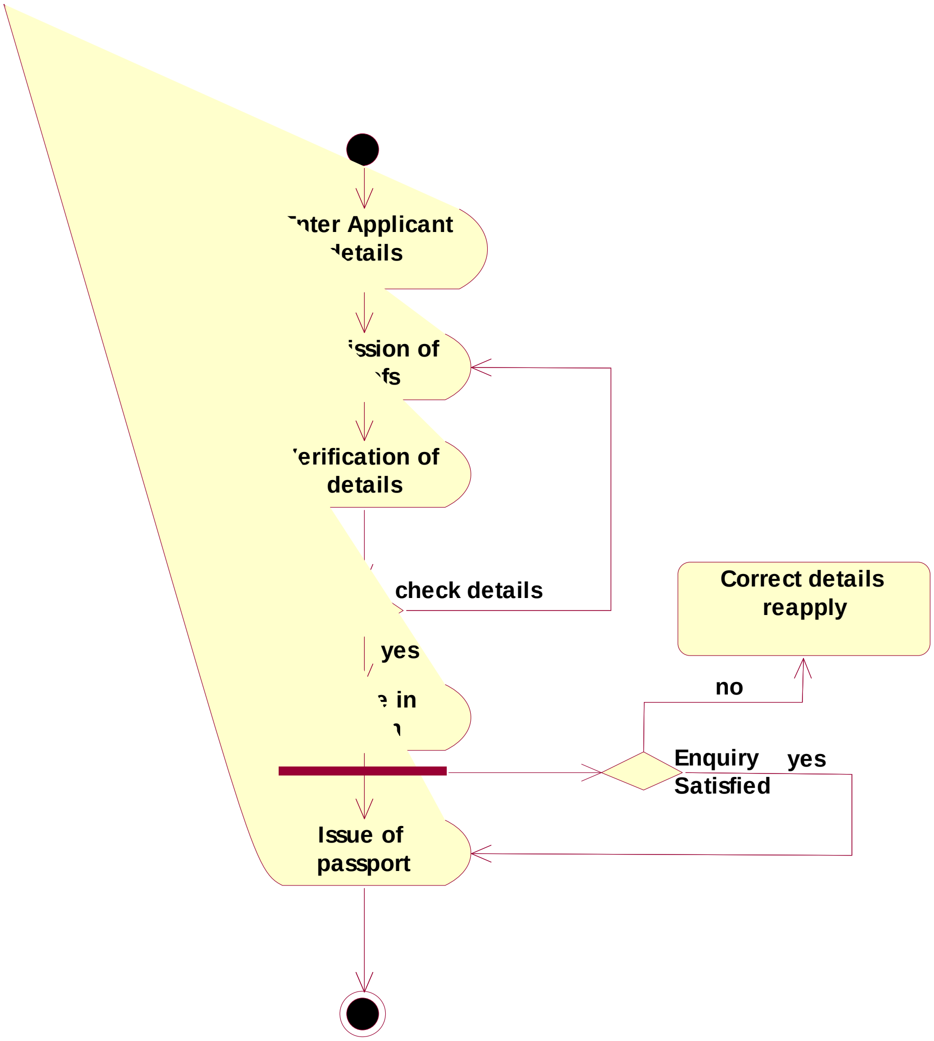
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Enter applicant details, Submission of proof, Verification of details, issue of passport.

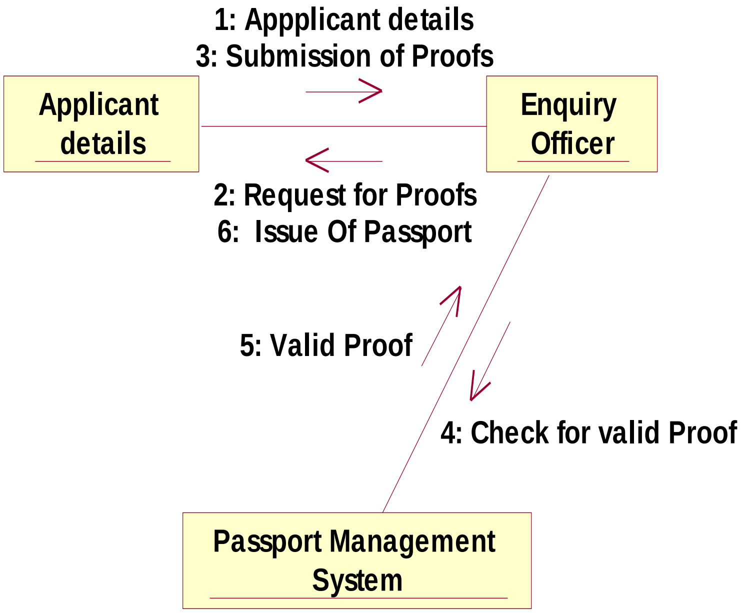
**Decision box:** Check details whether it is correct or not.



## Collaboration Diagram:

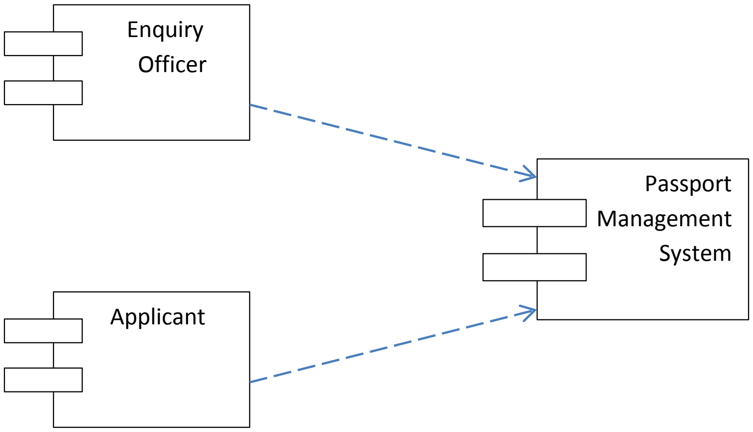
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



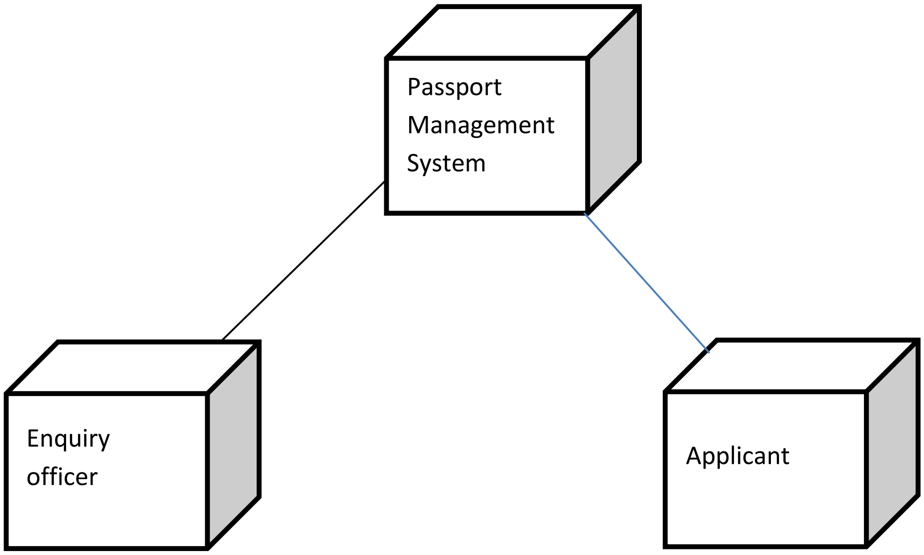
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



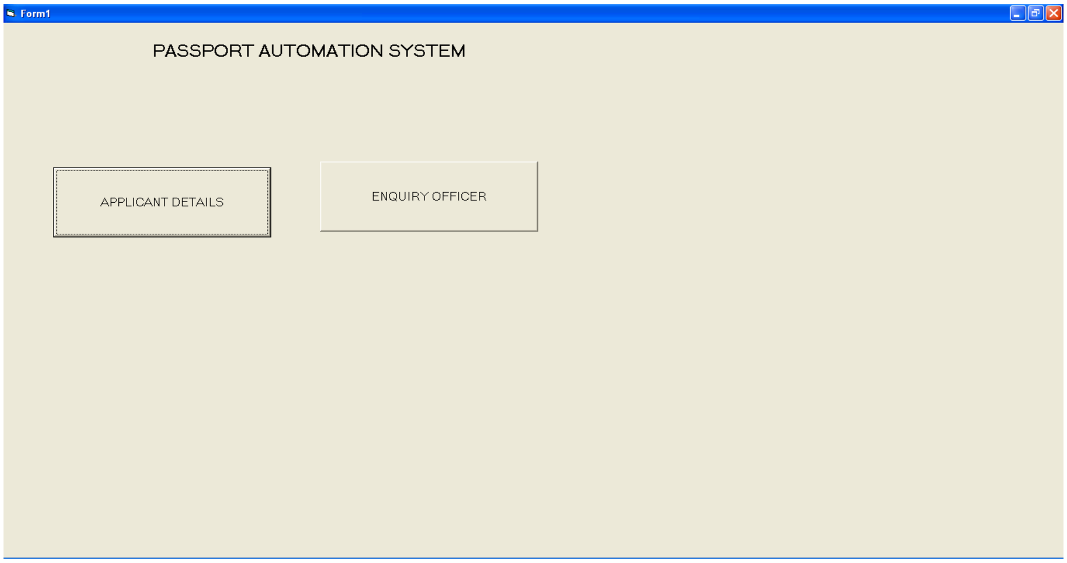
## Deployment Diagram:

Deployment diagram shows the physical configurations of software and hardware.

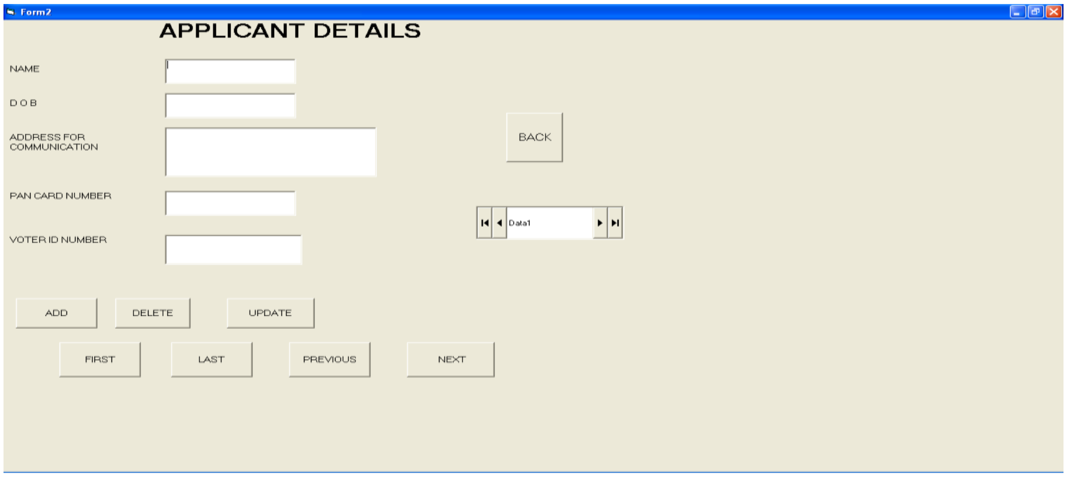


## VB DESIGN &CODING :

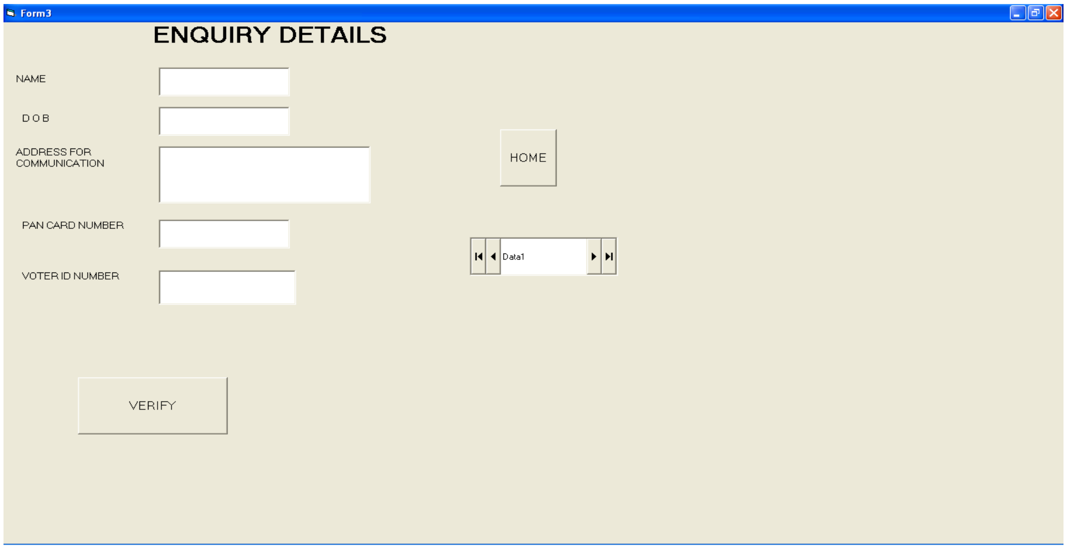
## FORM 1



## FORM 2



## FORM 3



## FORM 1:

```
Private Sub cmdAPPLICANTDETAILS_Click()
 Form2.Show
End Sub

Private Sub cmdENQUIRYOFFICER_Click()
 Form3.Show
End Sub
```

**FORM 2:**

```
Private Sub cmdADD_Click()

 Data1.Recordset.AddNew
End Sub

Private Sub cmdDELETE_Click()
 Data1.Recordset.Delete
 If Data1.Recordset.EOF Then
    Data1.Recordset.MoveLast
ElseIf Data1.Recordset.BOF Then
    Data1.Recordset.MoveFirst
 Else
    Data1.Recordset.MoveNext
 End If
MsgBox "Record Deleted Successfully", vbInformation
End Sub

Private Sub cmdFIRST_Click()
 Data1.Recordset.MoveFirst
 If Data1.Recordset.BOF Then
MsgBox "Beginning Of File", vbCritical
 End If
End Sub

Private Sub cmdLAST_Click()
Data.Recordset.MoveLast
 If Data1.Recordset.EOF Then
MsgBox "End Of File", vbCritical
 End If
End Sub

Private Sub cmdNEXT_Click()
 Data1.Recordset.MoveNext
 If Data1.Recordset.EOF Then
    Data1.Recordset.MoveLast
MsgBox "End Of File", vbCritical
 End If
End Sub

Private Sub cmdPREVIOUS_Click()
 Data1.Recordset.MovePrevious
 If Data1.Recordset.BOF Then
    Data1.Recordset.MoveFirst
MsgBox "Beginning Of File", vbCritical
```

```
 End If
End Sub

Private Sub cmdUPDATE_Click()
 Data1.Recordset.Edit
 Data1.Recordset.Update
End Sub

Private Sub lableBACK_Click()
Form1.show
Me.hide
End Sub
```

**FORM 3:**

```
Private Sub cmdVERIFY_Click()
IfData1.Recordset.Text5.Text = Form3.Text5.Text Then
Data1.Recordset.Text1.Text = Form3.Text1.Text
Data1.Recordset.Text2.Text = Form3.Text2.Text
Data1.Recordset.Text3.Text = Form3.Text3.Text
Data1.Recordset.Text4.Text = Form3.Text4.Text
MsgBox "DATA MATCHED"
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
End  If
End Sub
```

**JAVA CODING :**

**PassportManagementSystem.java**

```
public class PassportManagementSystem
{
  private int VerifyDetails;
  private int StoreProof;

  public PassportManagementSystem()
  {    }

  public void IssueOfPassport()
```

```
    {    }

       public void VerificationOfProof()
    {  }

       public void CancellationOfPassport()
    {   }
}
```

**EnquiryOfficer.java**

```
public class EnquiryOfficer
{
   private int VerificationOfDetails;
   public PassportManagementSystem thePassportManagementSystem;

   public EnquiryOfficer()
 {    }

   public void IssueOfPassport()
{    }

    public void RenewalOfPassport()
{    }
}
```

**RESULT:**

Thus the project PASSPORT AUTOMATION SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# BOOK BANK REGISTRATION SYSTEM

**EX NO : 2**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the BOOK BANK REGISTRATION SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

This software is designed for the verification of the details of the student by the central computer. The details regarding the student will be provided to the central computer through the administrator in the book bank and the computer will verify the details of student and provide approval to the office. Then the books that are needed by the student will issue from the office to the him.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

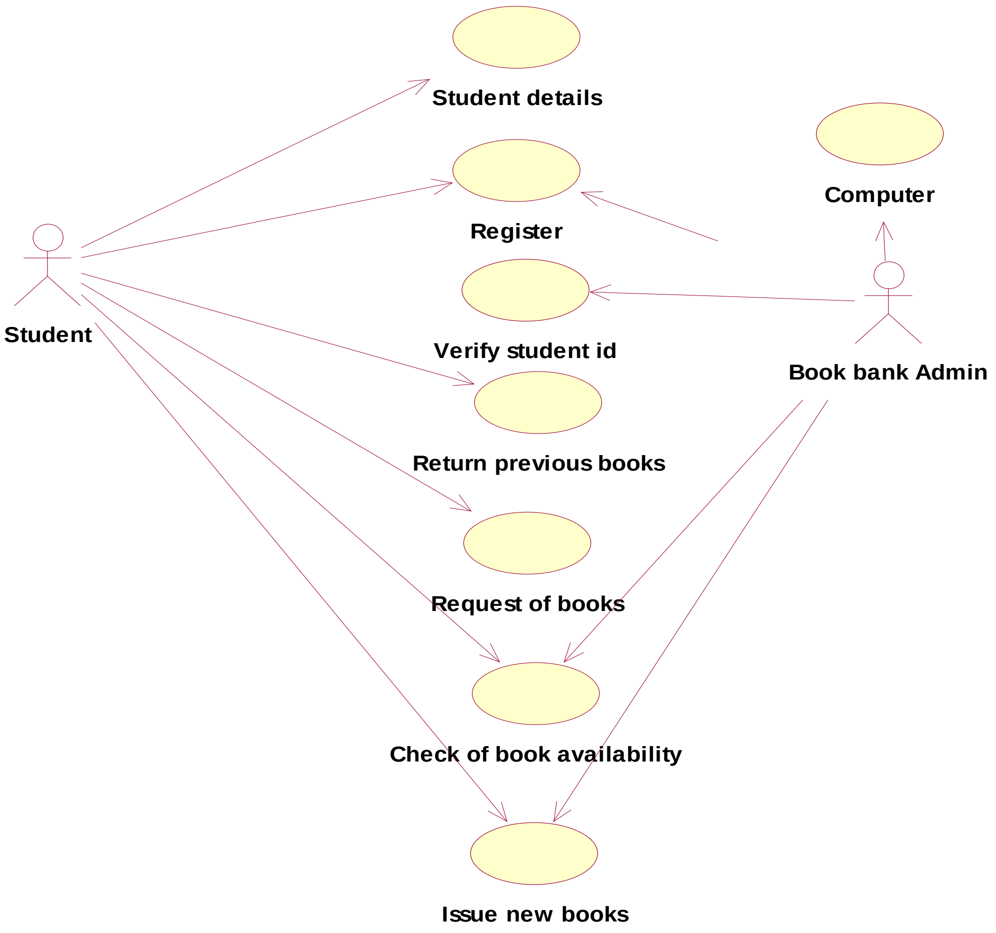
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Student, book bank admin.

**Usecase**: Student details, register, verify student id, return previous books, request of books, issue of books, check of book availability.



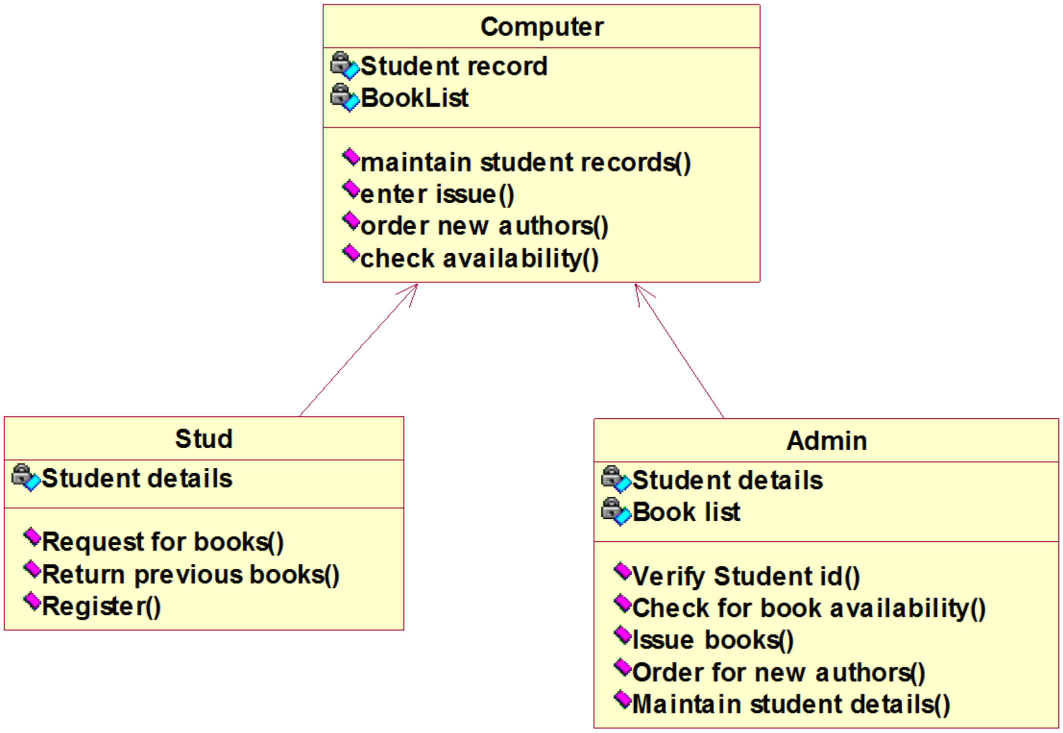
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Computer | Student record, booklist | Enter issue(),check availability() |
| Stud | Student Details | Request for books(), register() |
| Admin | Student Details, book list | Verify student id(), issue books() |

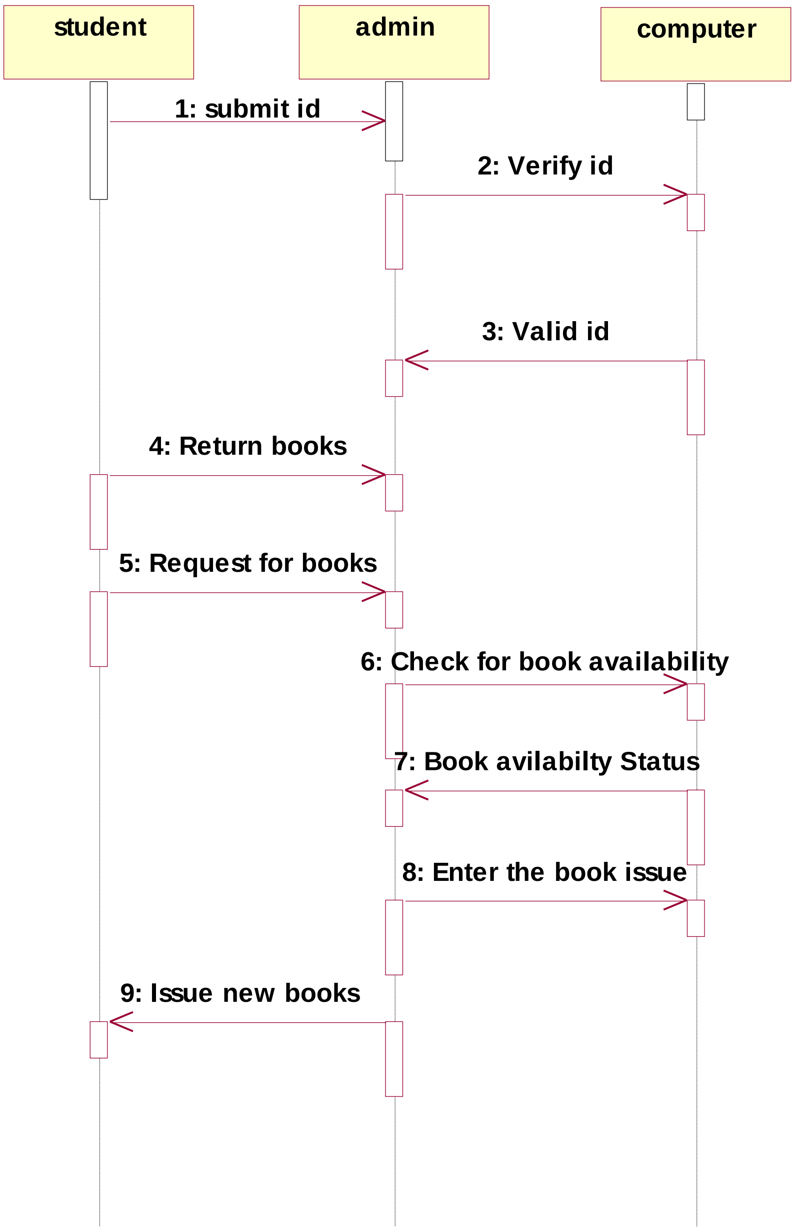


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Stud, admin, computer



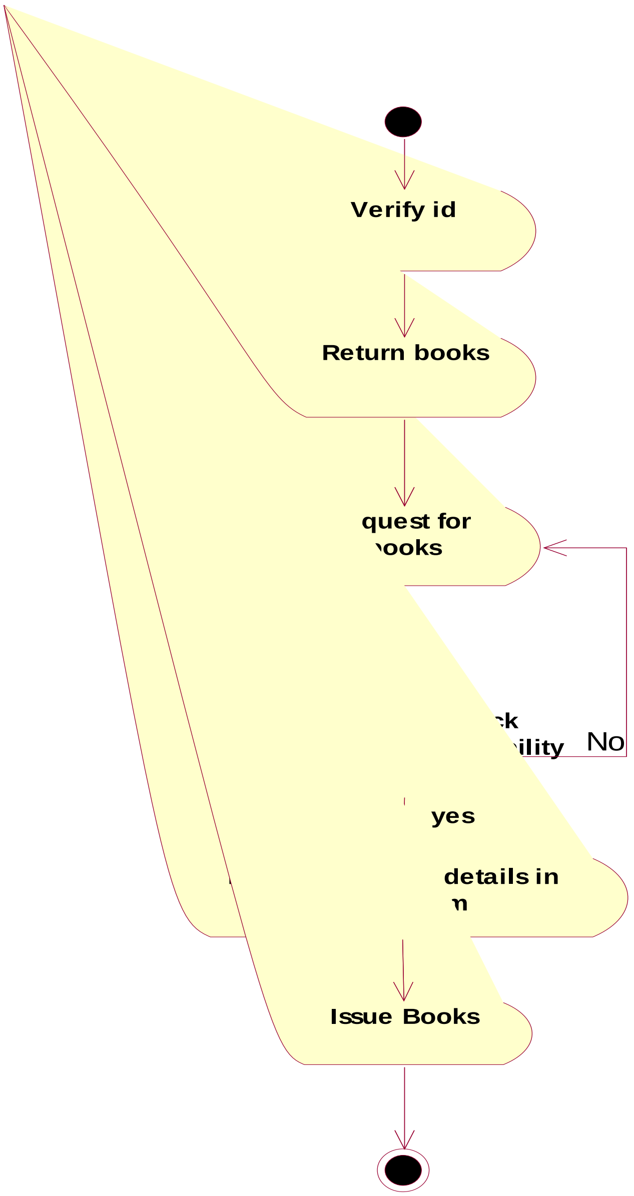
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Verify id, return books, request for books, enter book issue details in system, issue books

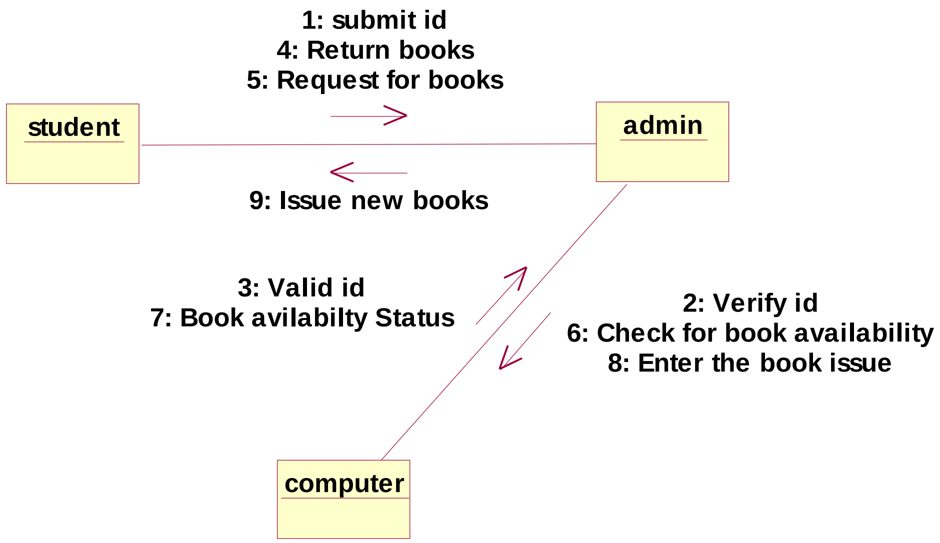
**Decision box:** Check availability of books whether it is present or not.



## Collaboration Diagram:

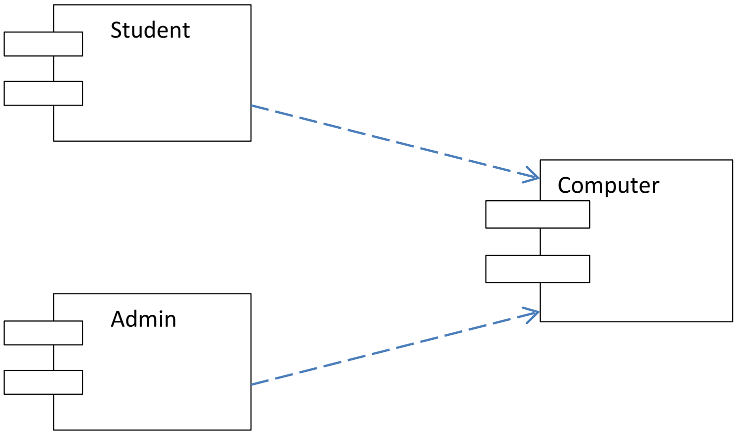
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



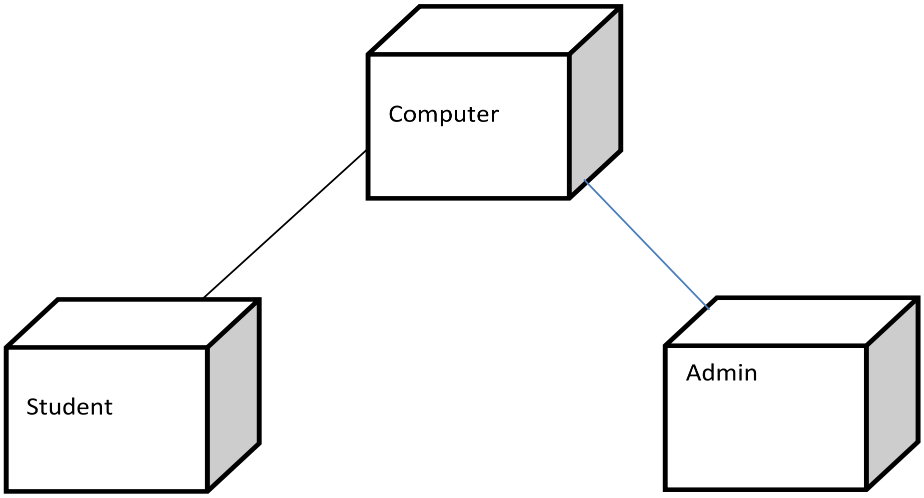
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**Computer.java**

```
public class Computer
{
  private int StudentRecord;
  private int BookList;
  public Admin theAdmin;

 public Computer()
 {   }

  public void maintainStudentRecords()
  {     }

  public void enterIssue()
  {    }

  public void orderNewAuthors()
  {   }

  public void checkAvailability()
  {       }
}
```

**Stud.java**

```
public class Stud
{
  private int StudentDetails;
  public Computer theComputer;

  public Stud()
  {     }

  public void RequestForBooks()
  {     }


  public void ReturnPreviousBooks()
  {     }

  public void Register()
  {     }
}
```

**RESULT:**

Thus the project BANK BOOK REGISTRATION SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# EXAM REGISTRATION SYSTEM

**EX NO : 3**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the EXAM REGISTRATION SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

This software is designed for the verification of the details of the candidate by the central computer. The details regarding the candidate will be provided to the central computer through the administrator and the computer will verify the details of candidate and provide approval .Then the hall ticket will be issued from the office to the candidate.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

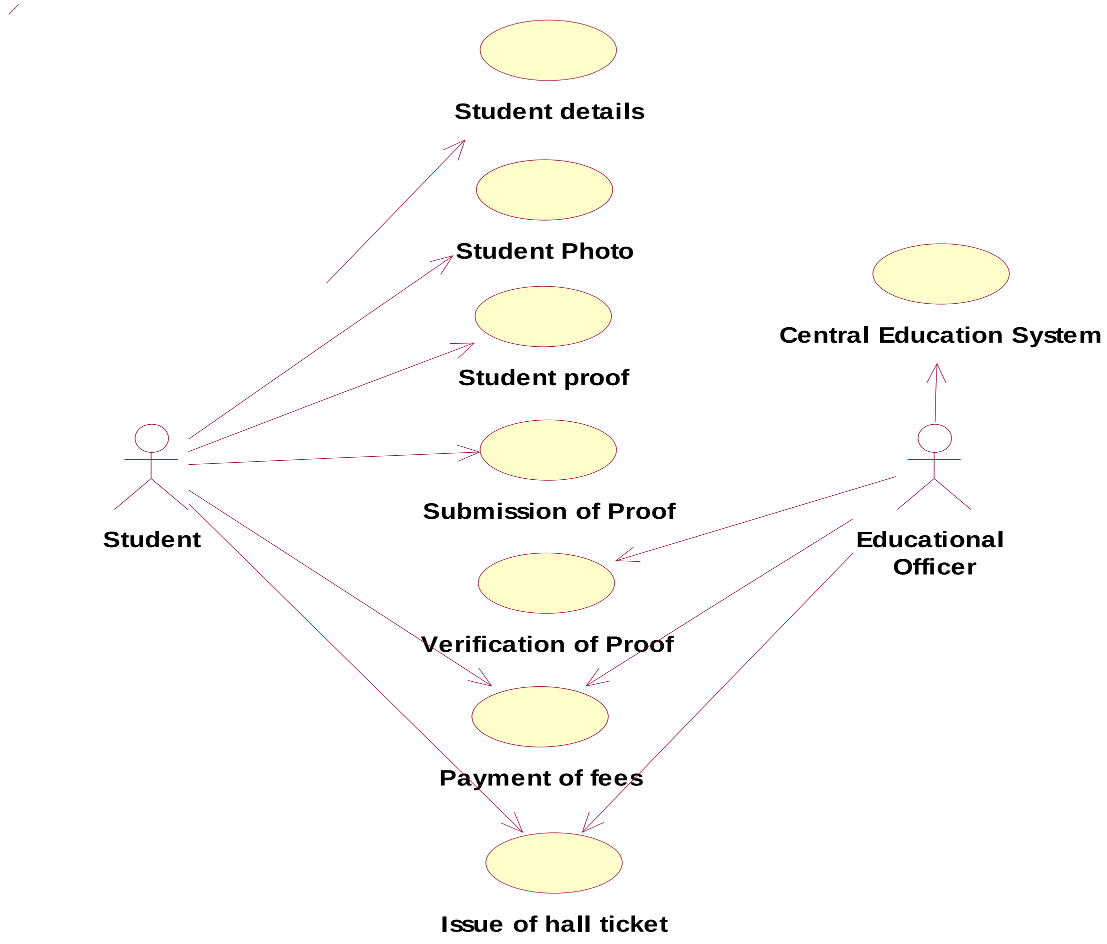
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Student, educational officer.

**Usecase**:Student details, student photo, student proof submission of proof ,verification of proof, payment of fees, issue of hall
ticket.



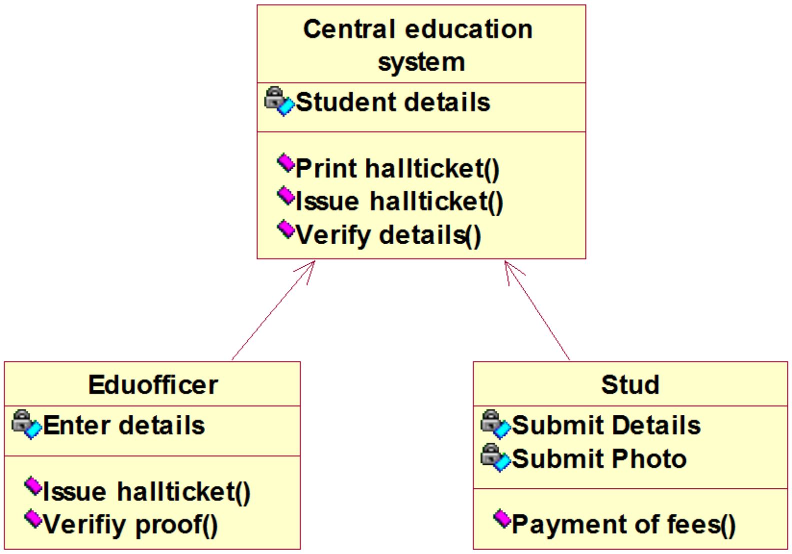
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central educational system | Student details | Print hall ticket(), issue hall ticket() |
| Stud | Submit details(),submit photo() | Payment of fees() |
| Eduofficer | Enter details | Issue hall ticket(), verify proof() |

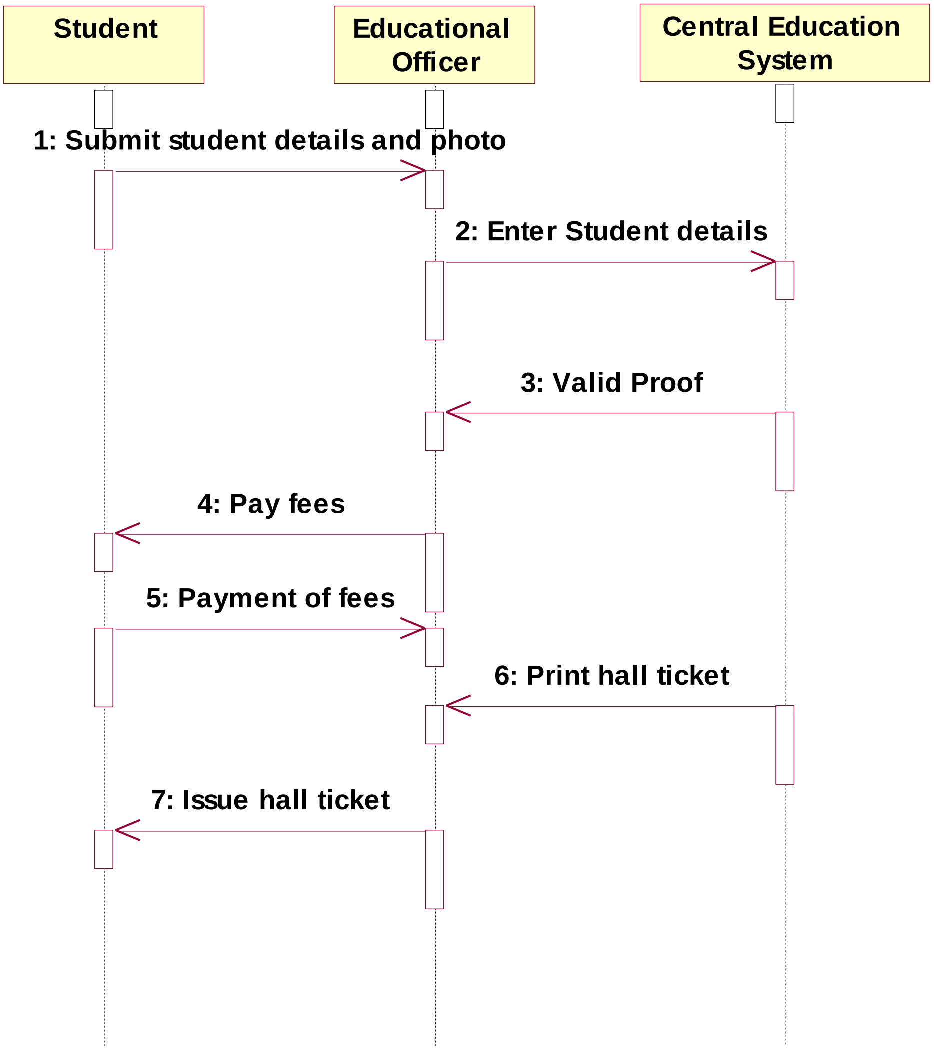


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** student, educational officer, central education system



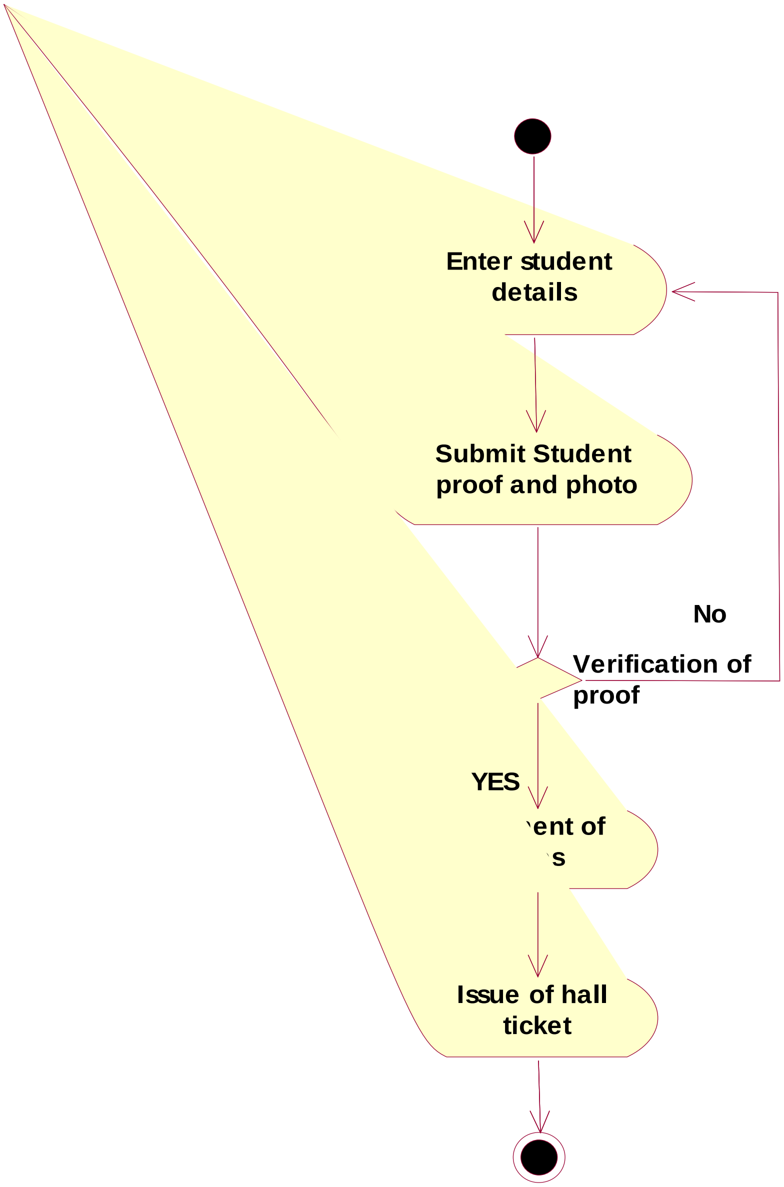
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Enter student details , submit student proof and photo, payment of fees, issue of hall ticket.

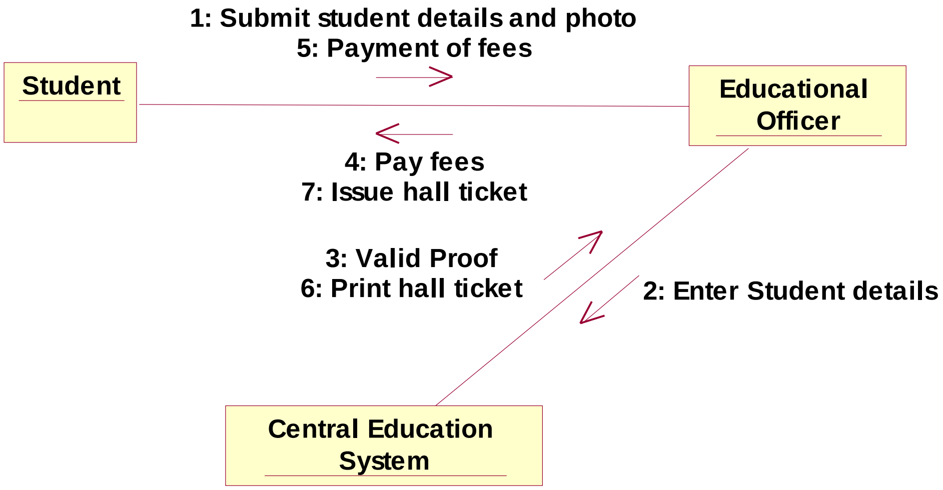
**Decision box:** Verification of proof.



## Collaboration Diagram:

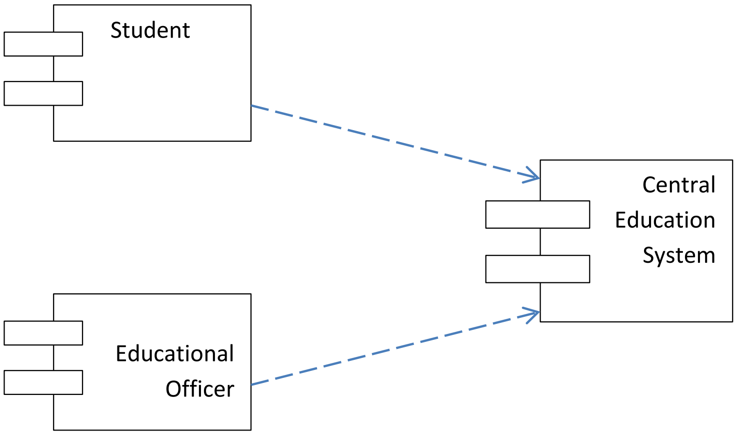
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



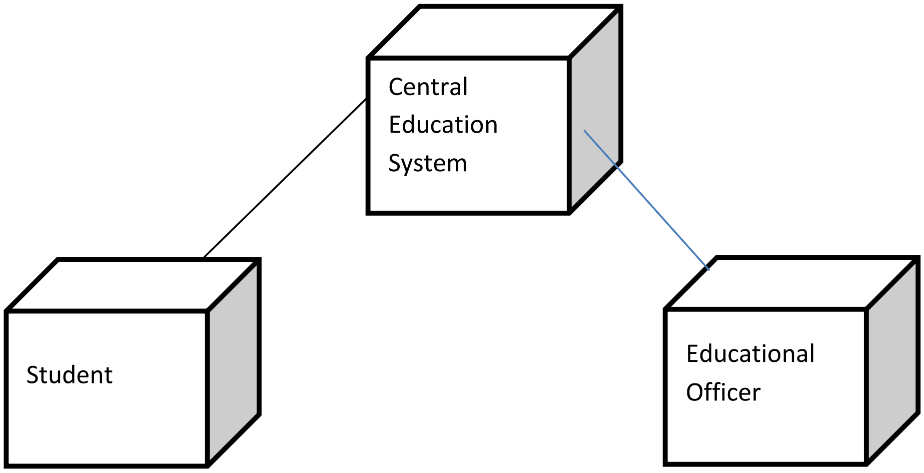
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**Eduofficer.java**

```
public class Eduofficer
{
  private int EnterDetails;
  public CentralEducationSystem theCentralEducationSystem;

  public Eduofficer()
  {     }

   public void IssueHallticket()
  {     }

   public void VerifiyProof()
  {   }
```

**CentralEducationSystem.java**

```
public class CentralEducationSystem
{
  private int StudentDetails;


  public CentralEducationSystem()
  {    }

  public void PrintHallticket()
  {    }

   public void IssueHallticket()
  {    }

     public void VerifyDetails()
  {  }
}
}
```

**RESULT:**

Thus the project EXAM REGISTRATION SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# STOCK MAINTENANCE SYSTEM

**EX NO : 4**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the STOCK MAINTENANCE SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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. The orders are verified and the items are delivered to the customer.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

**DESIGN :**

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

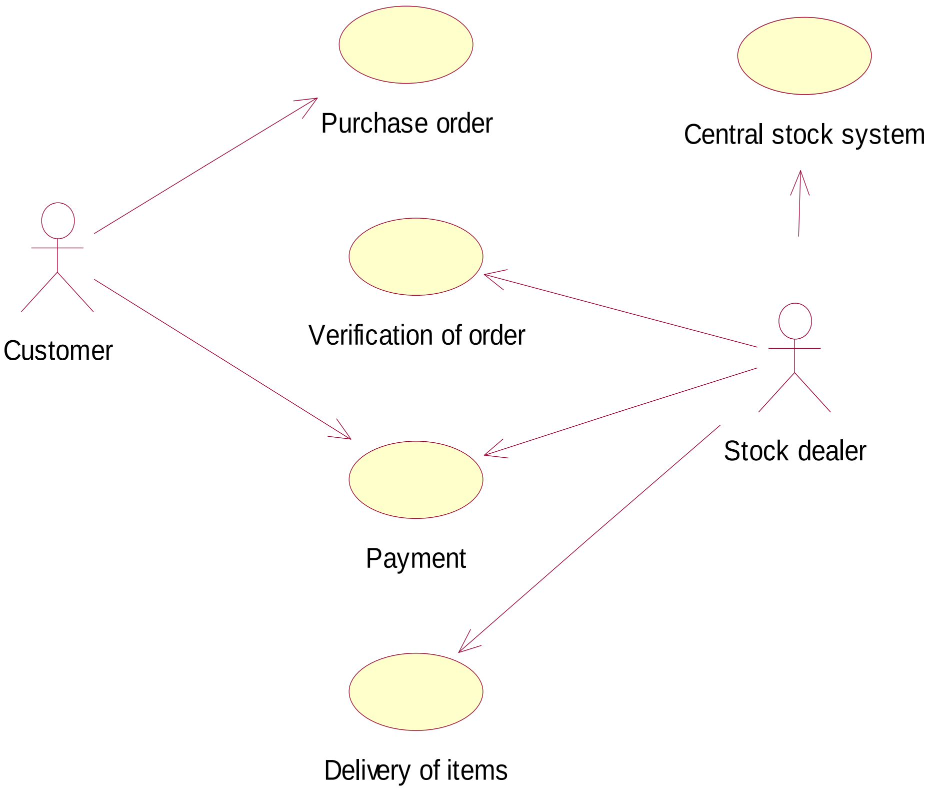
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Customer, Stock dealer, central stock system.

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



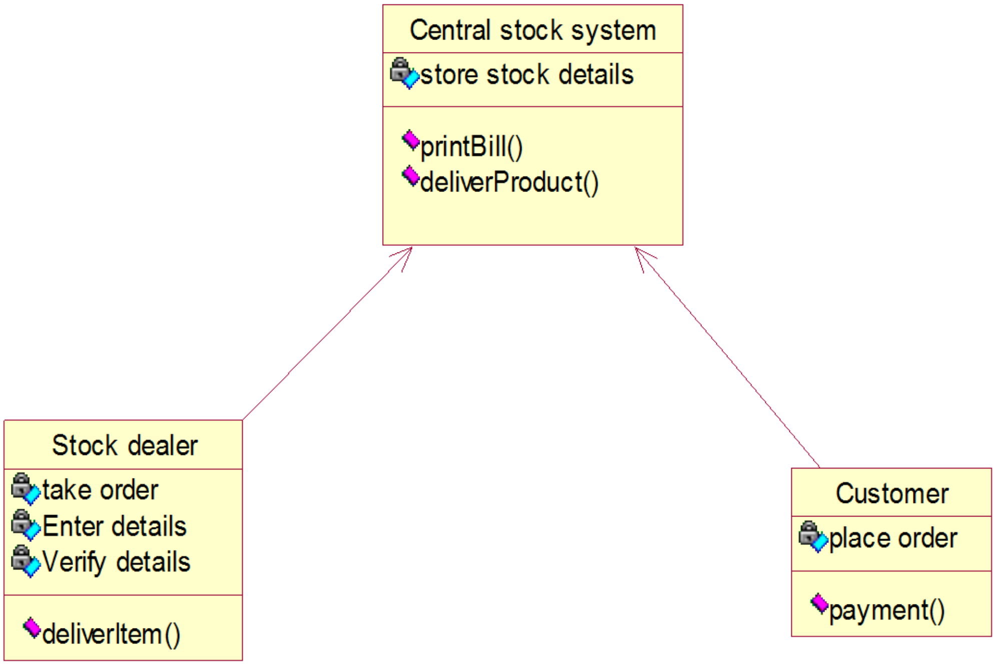
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central stock system | Store stock details | Print bill() |
| Stock dealer | Take order | Deliver item() |
| Customer | Place order | Payment() |

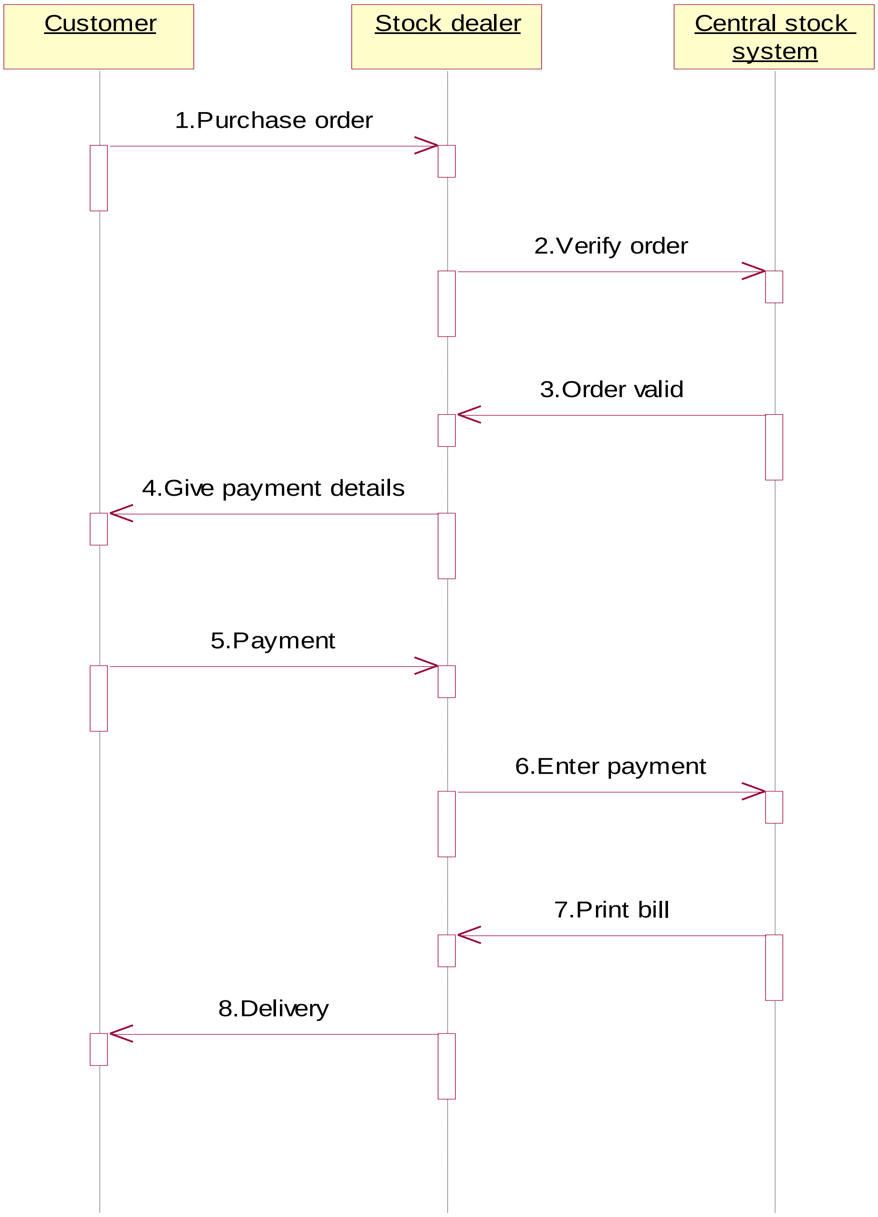


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Customer, Stock dealer, Central stock system



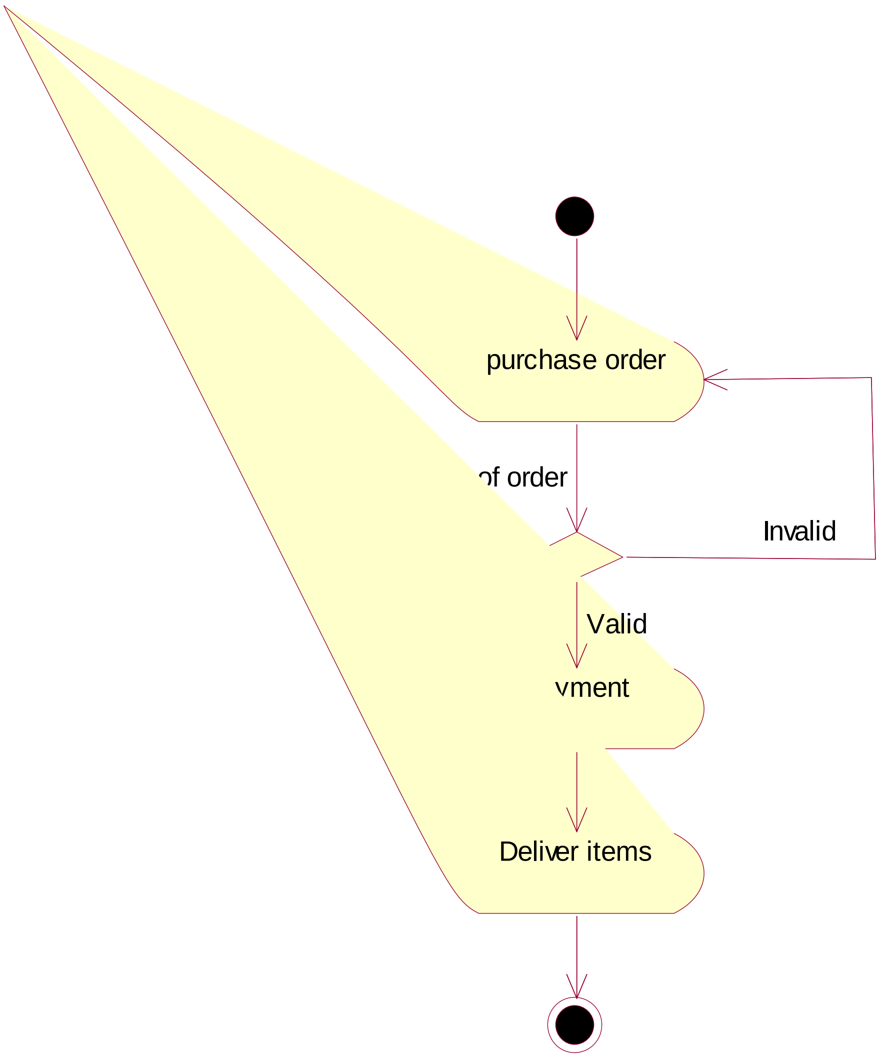
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities**: Purchase order, payment, delivery of items.

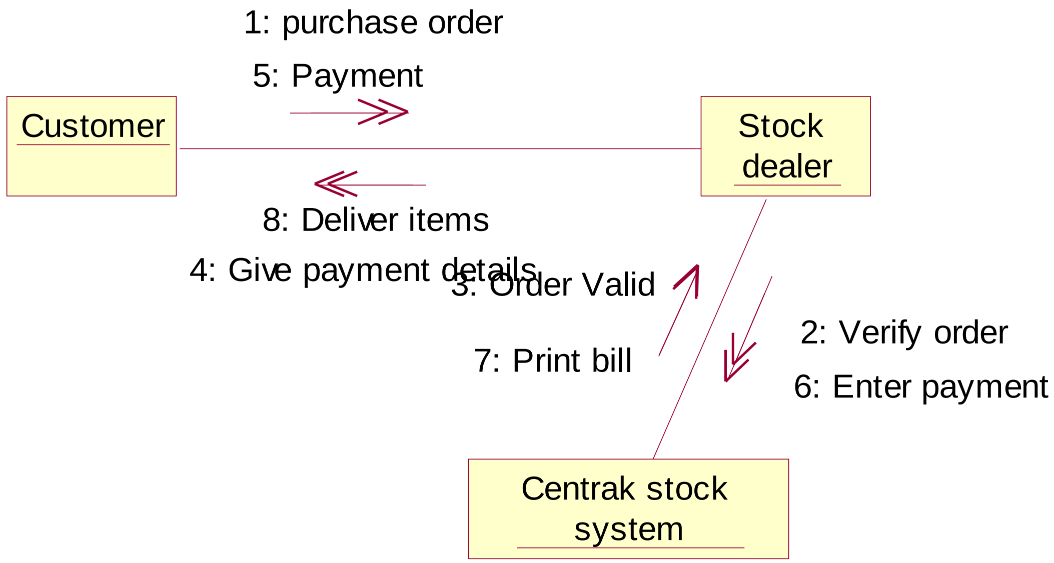
**Decision box:** Valid or not



**Collaboration Diagram:**

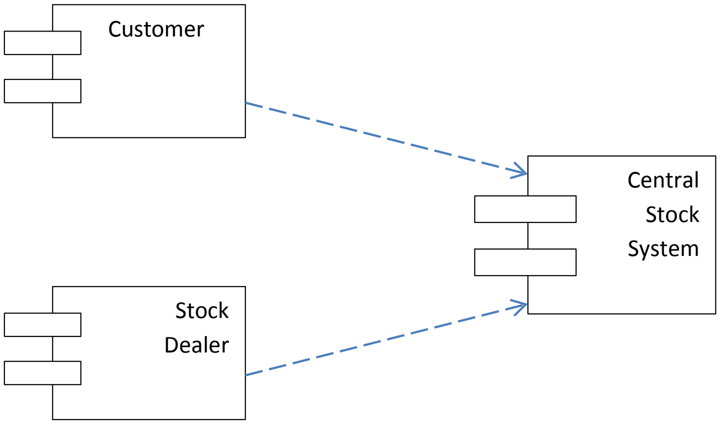
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



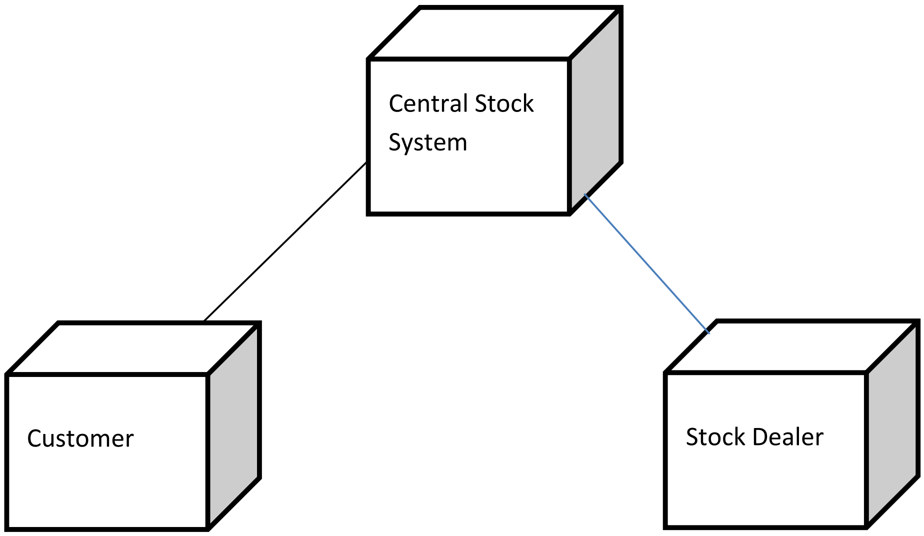
**Component Diagram:**

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**Customer.java**

```
public class Customer

{

   private int placeOrder;

   public CentralStockSystem theCentralStockSystem;

     public Customer()

   {   }

    public void payment()

   {   }

}
```

**RESULT:**

Thus the project STOCK MAINTENANCE SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# ONLINE COURSE RESERVATION SYSTEM

**EX NO : 5**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the ONLINE COURSE RESERVATION SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

This software is designed for supporting online course reservation system. This system is organized by the central management system. The student first browses and select the desired course of their choice. The university then checks the availability of the seat if it is available the student is enrolled for the course.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

**DESIGN :**

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

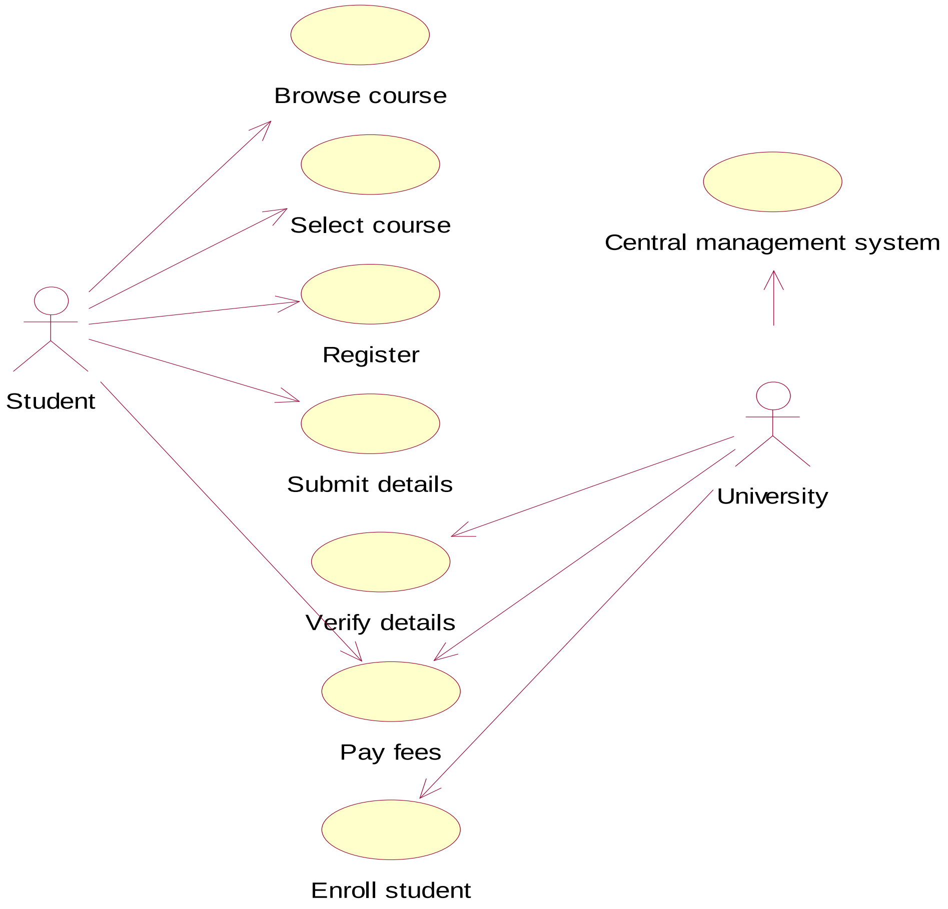
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Student, University

**Use case**: Browse course, select course, register, submit details, verify details, pay fees, enroll student



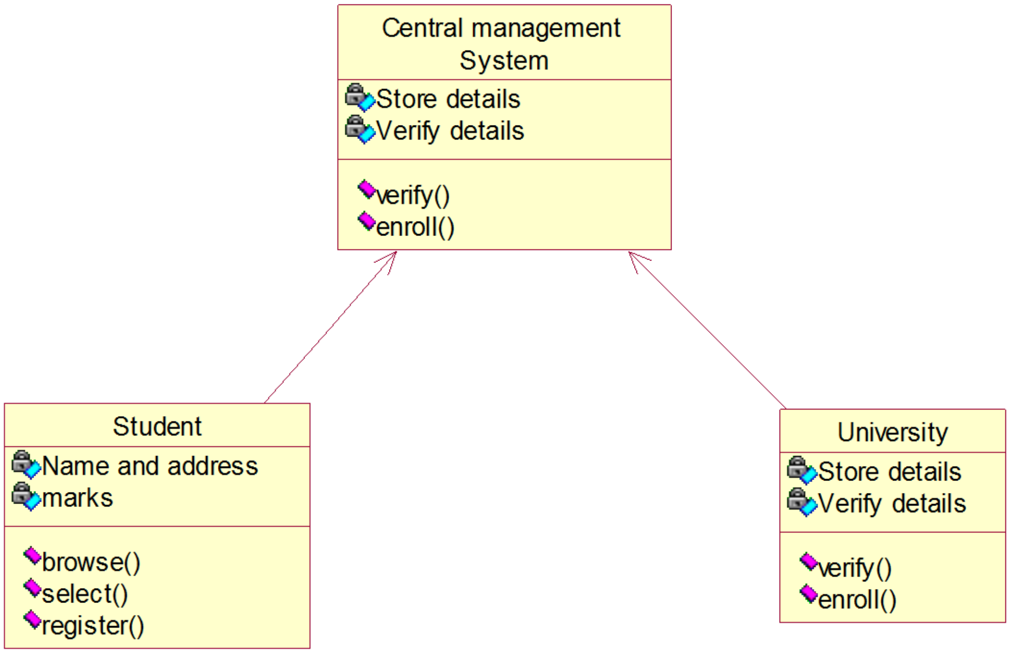
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central management system | Store details | Verify() |
| Student | Name and address | Browse() |
| University | Store details | Verify() |

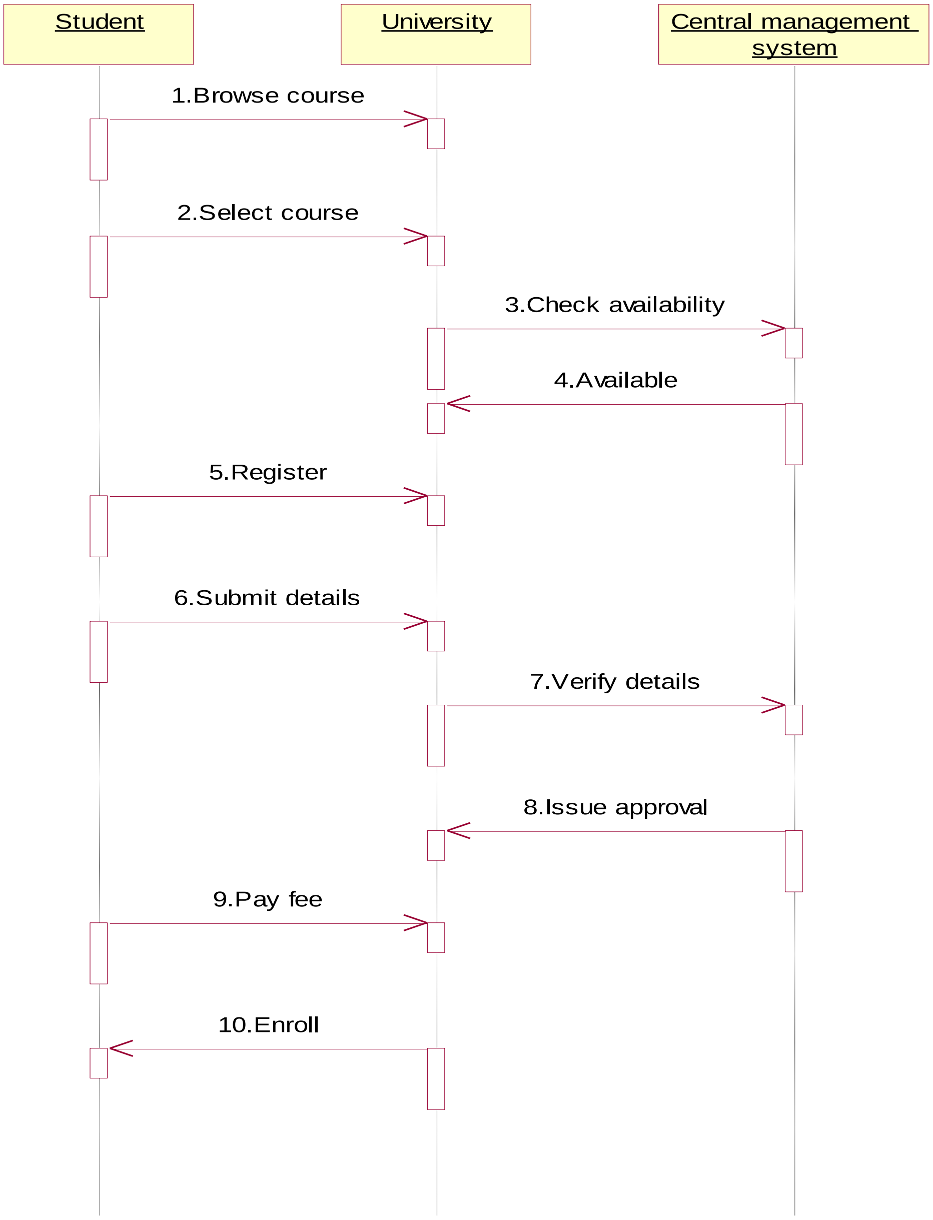


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Student, University, Central management system



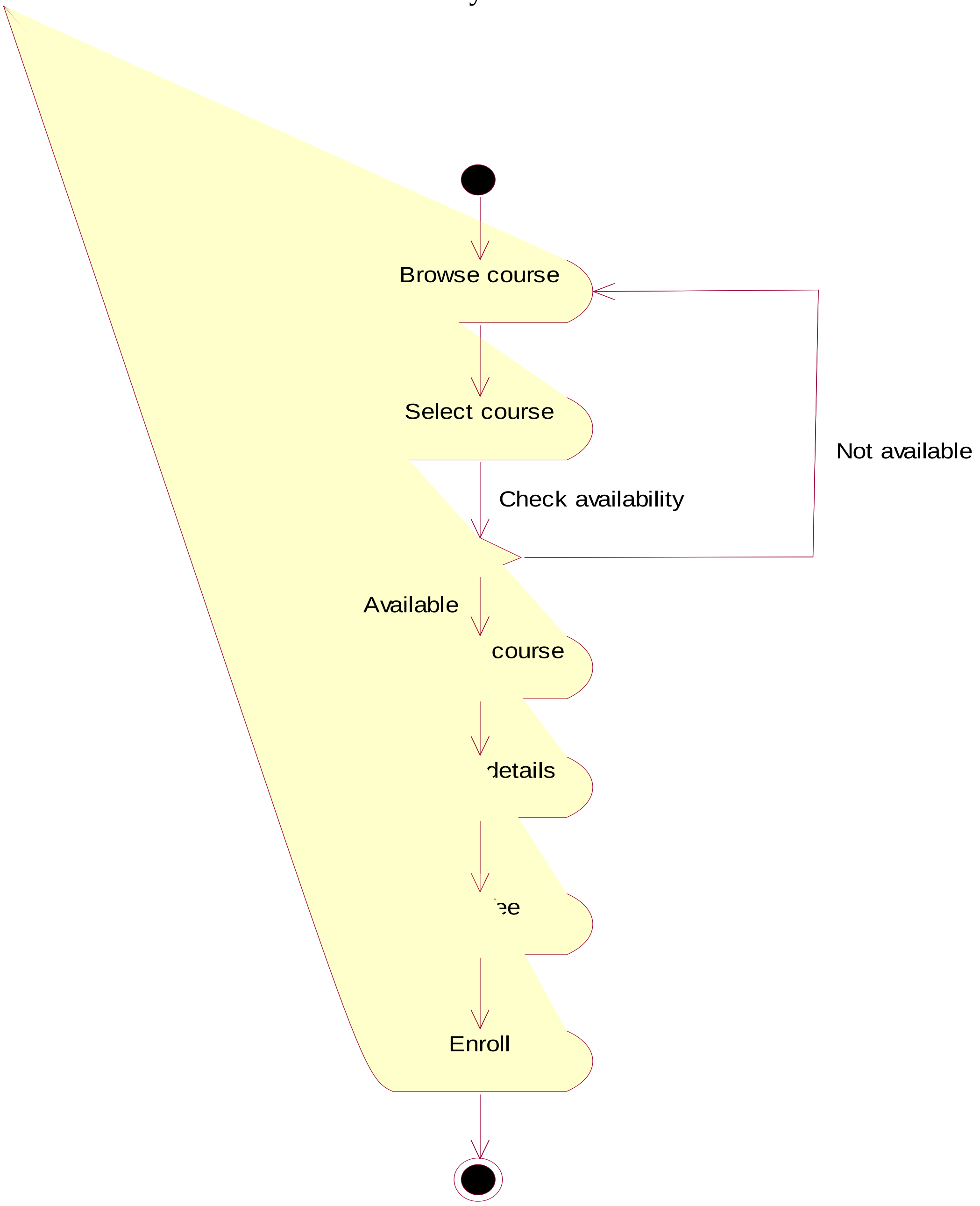
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities**: Browse course, select course, register course, submit details

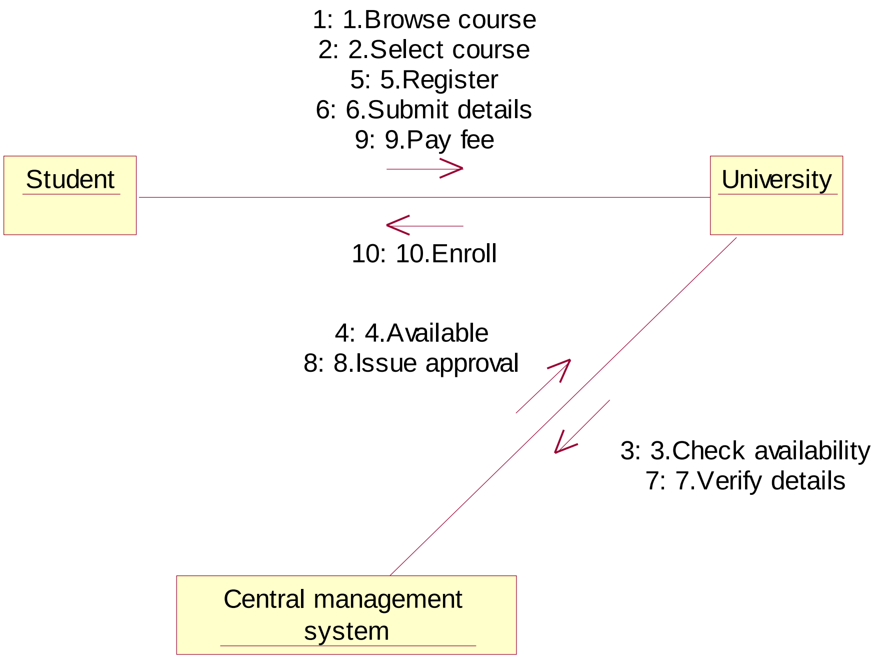
**Decision box:** check availability or not



## Collaboration Diagram:

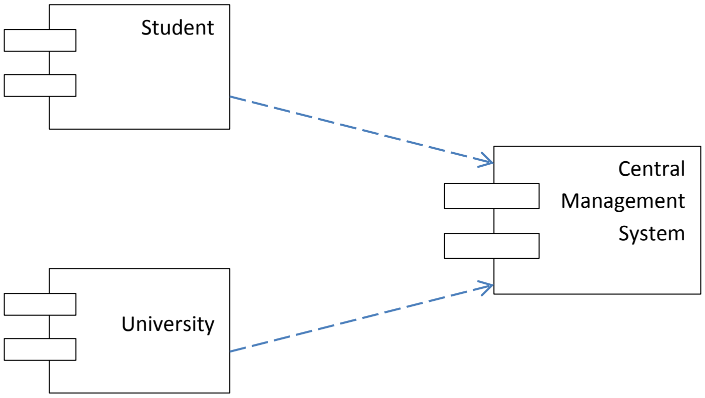
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



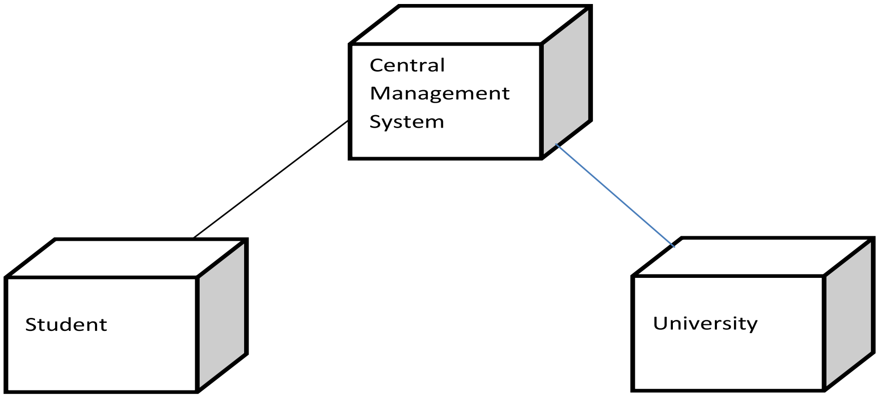
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



## Deployment Diagram:

Deployment diagram shows the physical configurations of software and hardware.



## JAVA CODING :

**CentralManagementSystem.java**

```
public class CentralManagementSystem

{

  private int StoreDetails;

  private int VerifyDetails;

  public CentralManagementSystem()

  {   }

 public void verify()

  {   }

   public void enroll()

  {    }}
```

## RESULT:

Thus the project ONLINE COURSE RESERVATION SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# E - TICKETING

**EX NO : 6**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the E- TICKETING with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

This software is designed for supporting the computerized e-ticketing.This is widely used by the passenger for reserving the tickets for their travel. This E-ticketing is organized by the central system. The information is provided from the railway reservation system.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

**DESIGN :**

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

  An Actor represent a set of rules that user of a case play when interacting with the usecases.

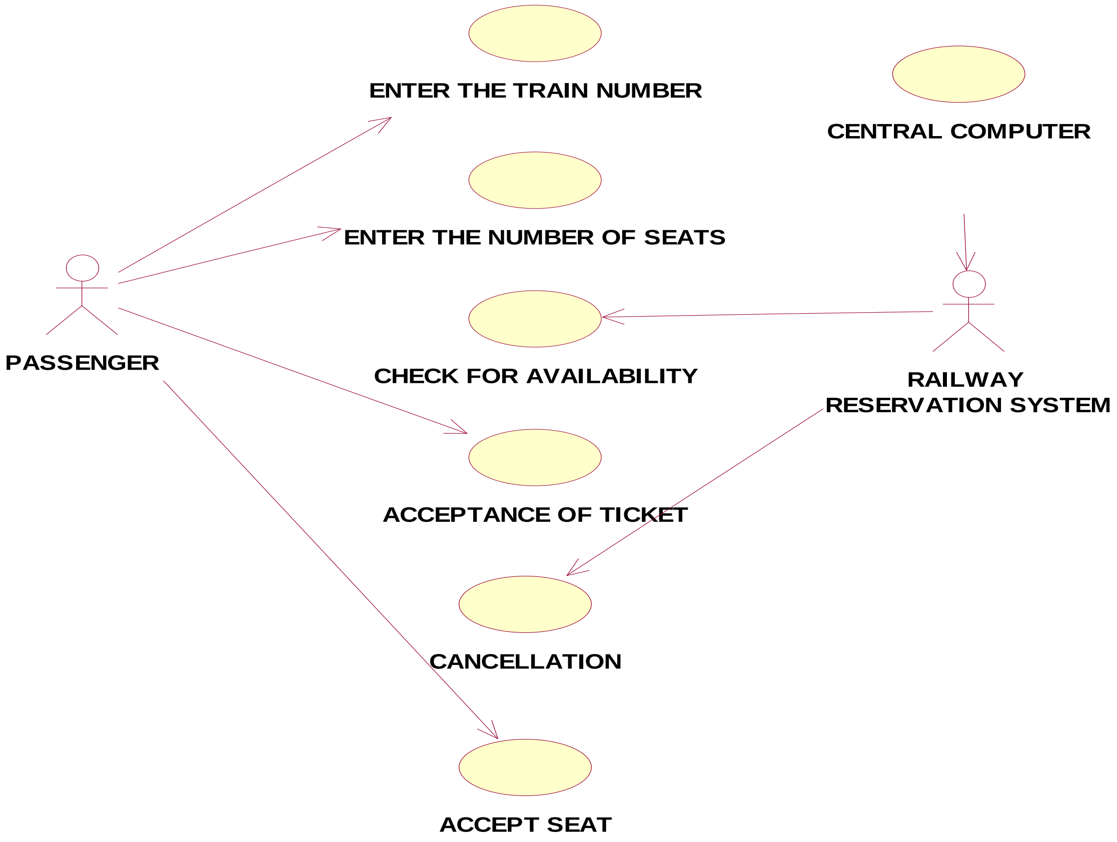
- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor

  .

**Actors**: Passenger, Railway reservation system.

**Use case**: Status, reservation, cancellation, enter the train number, enter the number of seats, availability of seats, acceptance of ticket.



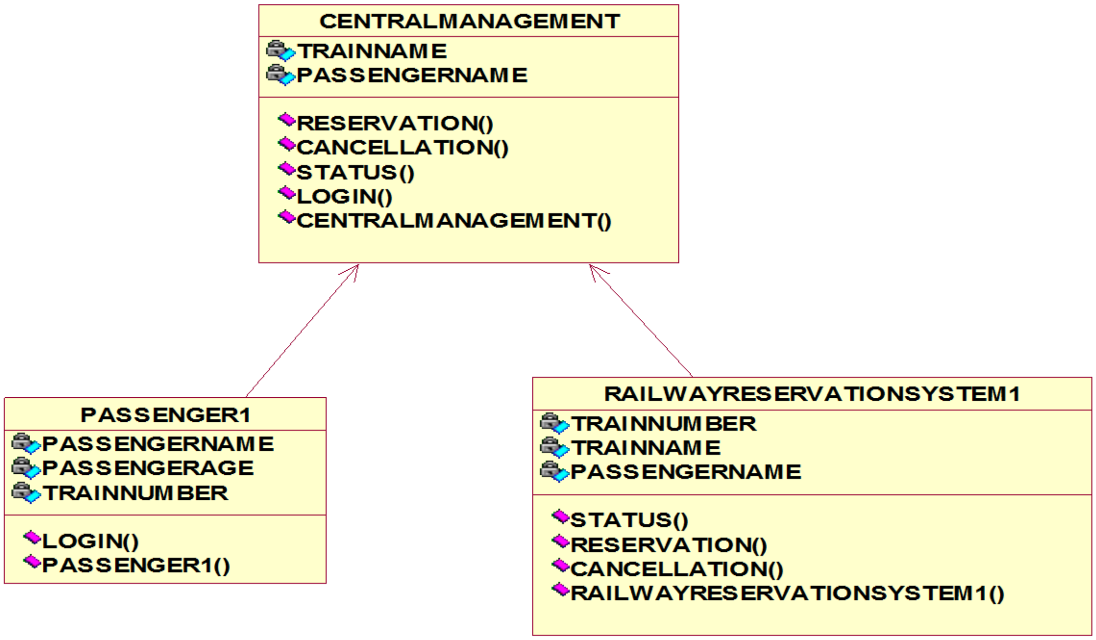
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central computer | Train name, passenger name | Reservation(),login() |
| Passenger | Passenger age | Login() |
| R\ailway reservation system | Train number | Cancellation() |

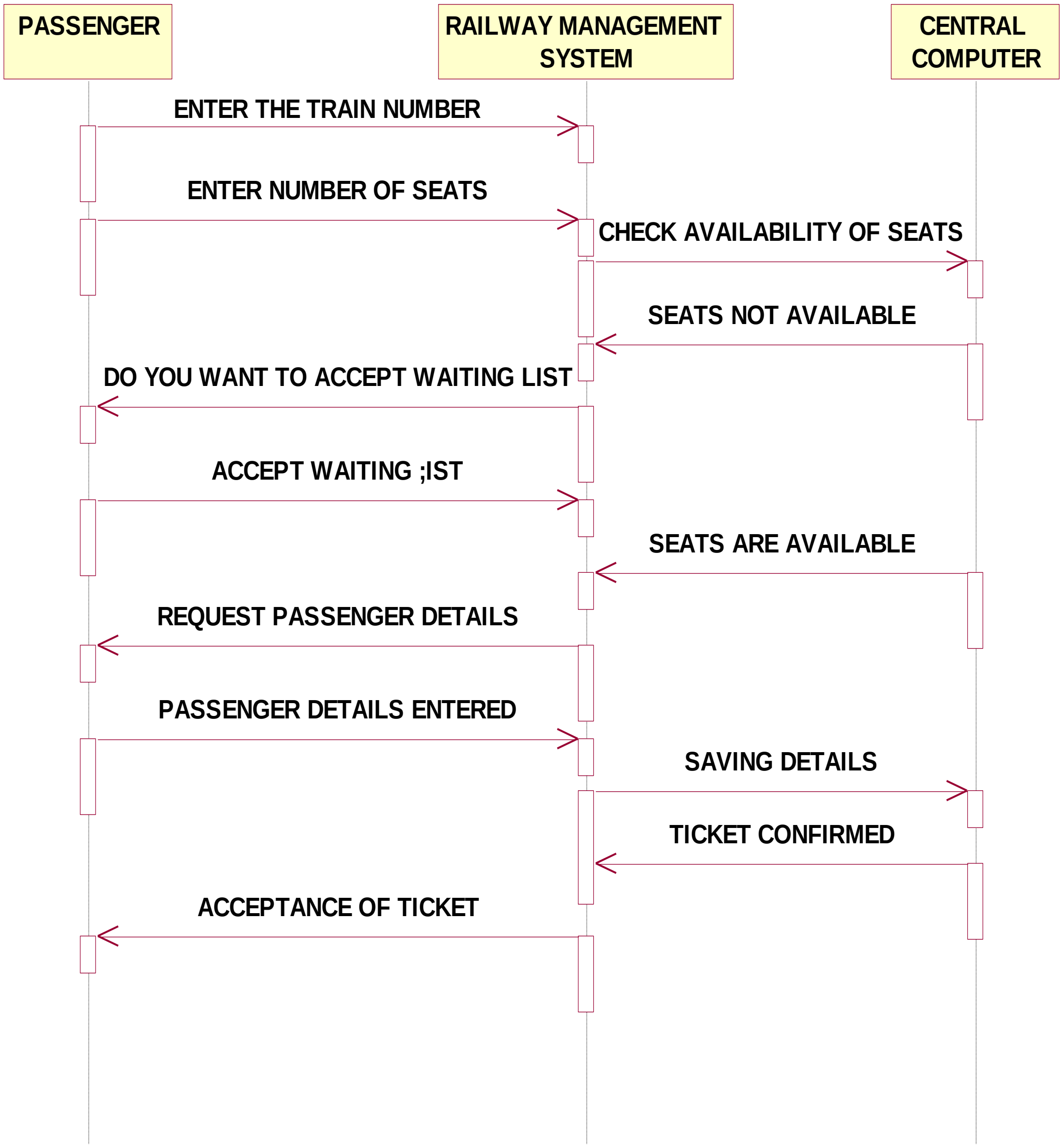


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Passenger, Railway reservation system, Central computer



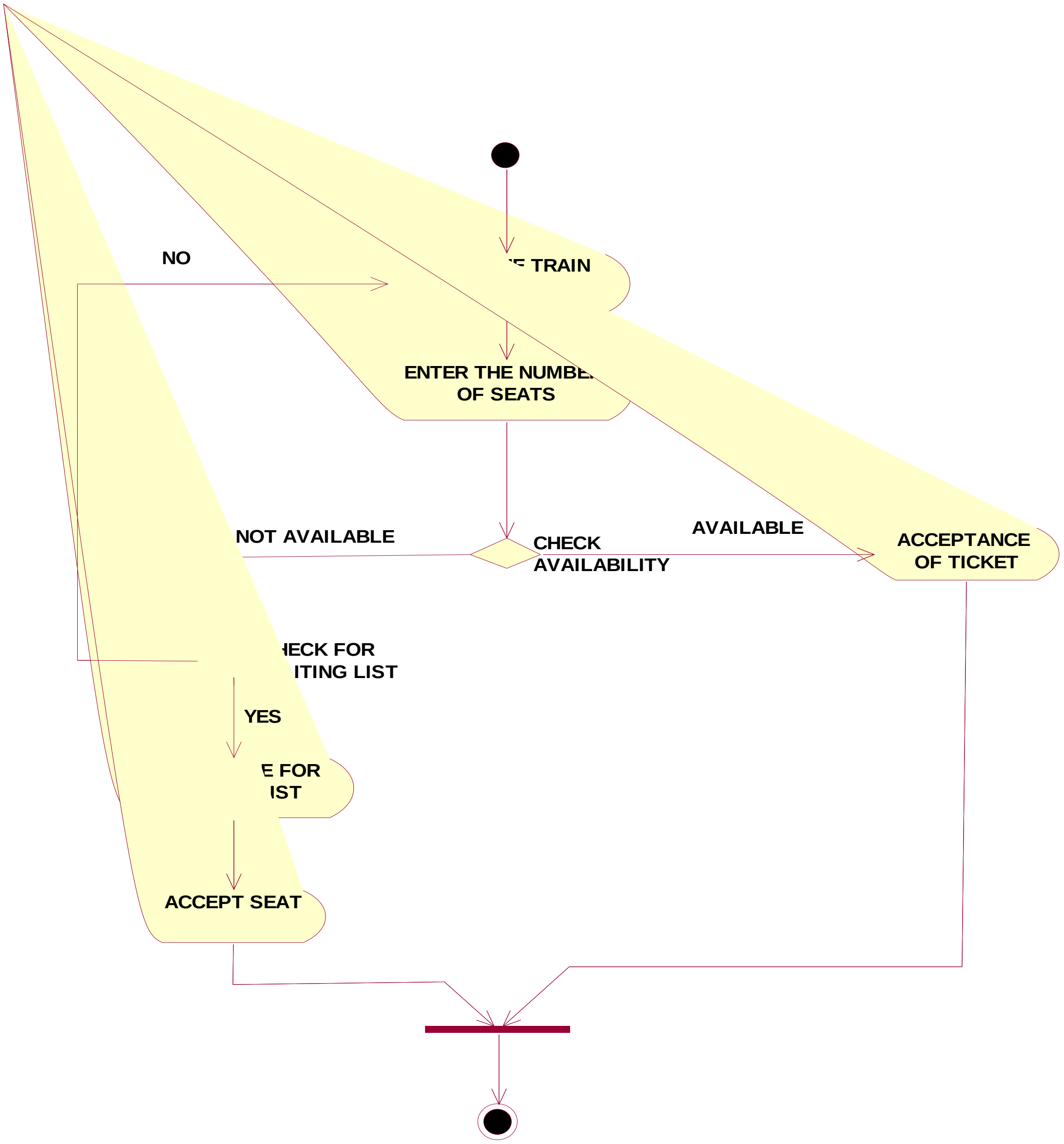
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities: E**nter the train number, Enter the number of seats, Acceptance of ticket,Accept seat.

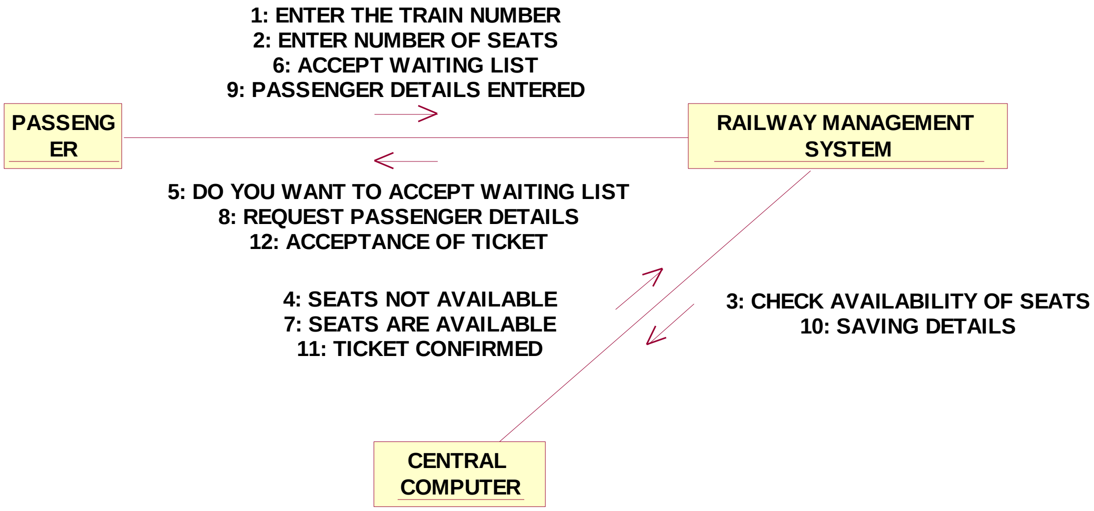
**Decision box:** Check availability of seats whether it is present or not



## Collaboration Diagram:

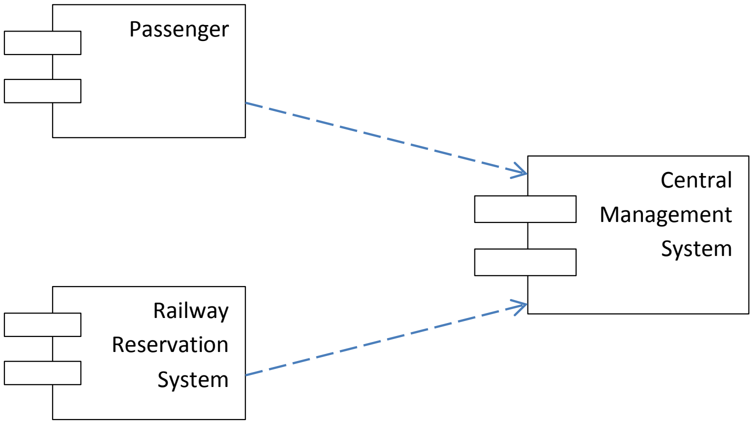
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



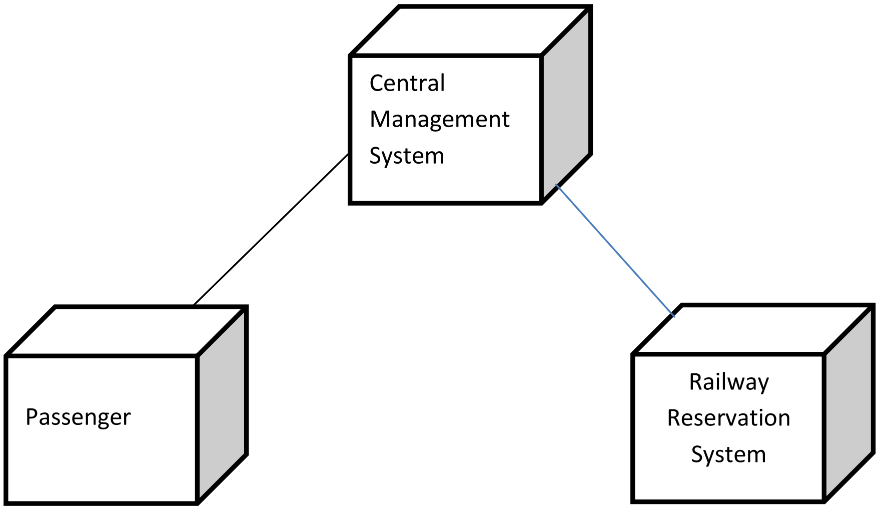
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



## JAVA CODING :

**RailwayManagementSystem.java**

```
public class railwayManagementSystem
{
  private int trainNo;
  private int trainName;
  private int name;
  public centralcomputer theCentralcomputer;


  public railwayManagementSystem()
  { }

 public void status()
  {  }

  public void reservation()
  {   }

  public void cancellation()
```

```
    {  }
  }
```

**Centralcomputer.java**

```
  public class centralcomputer
  {
    private int trainName;
    private int passengerName;

    public centralcomputer()
    {  }

    public void cancellation()
    {   }

public void status()
    {   }

    public void login()
    {   }
   }
```

**RESULT:**

Thus the project E -TICKETING was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# CREDIT CARD PROCESSING

**EX NO : 7**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the BOOK BANK REGISTRATION SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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**.**

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

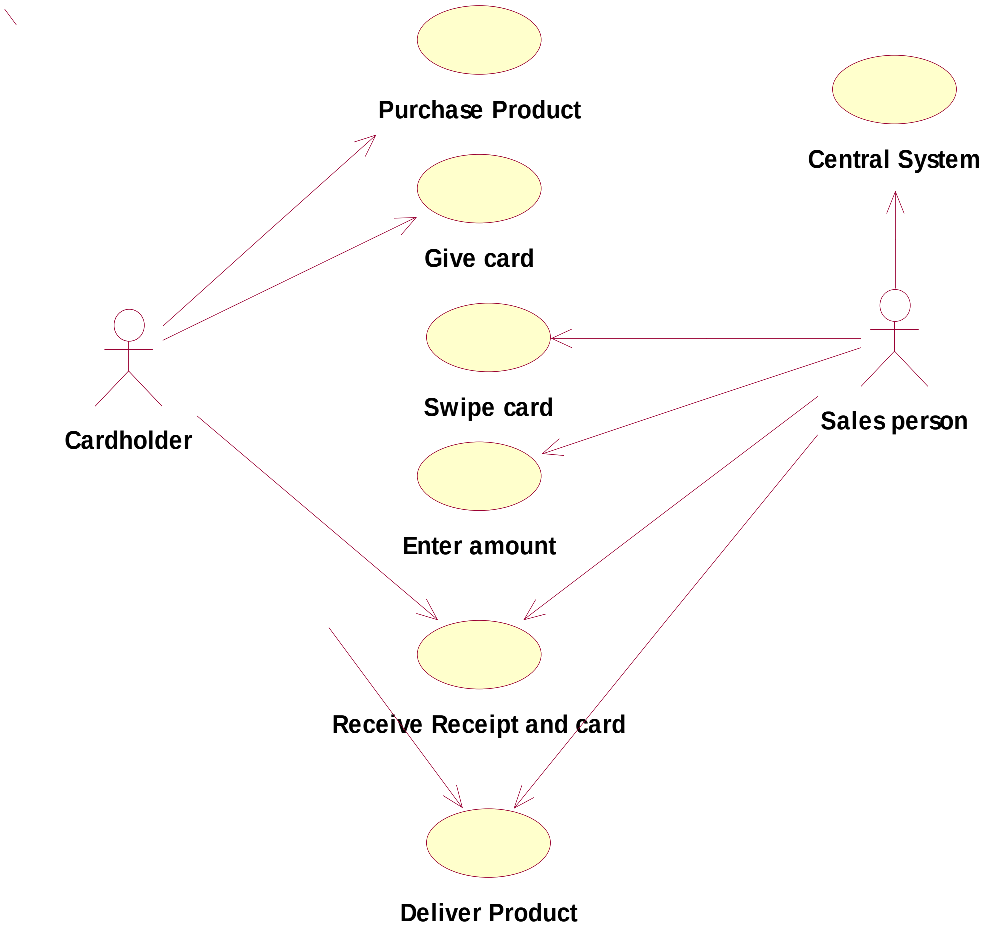
- Actor:

  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

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



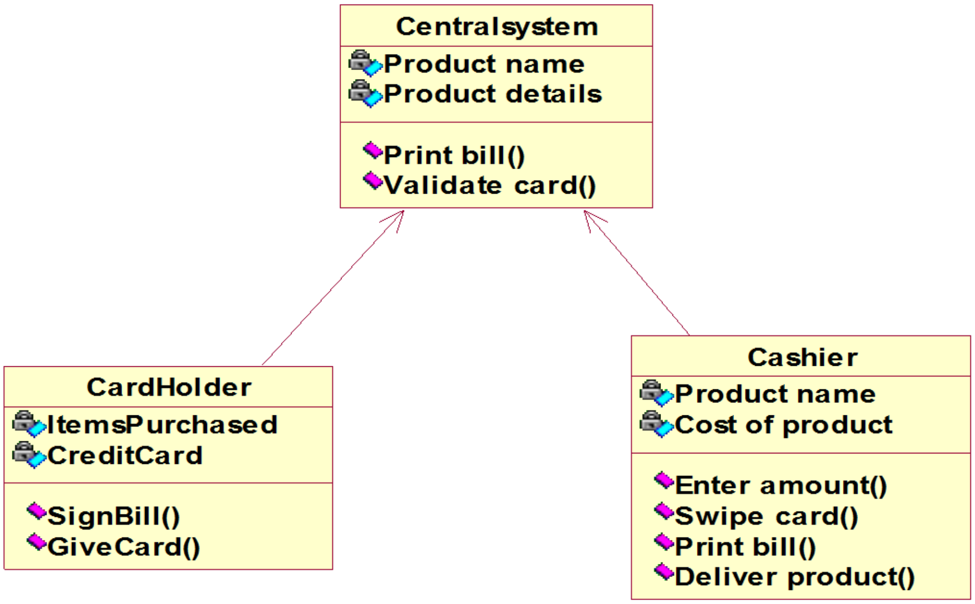
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of 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() |
| CardHolder | Items Purchased<br>Validate Card | Give card()<br>Sign bill() |

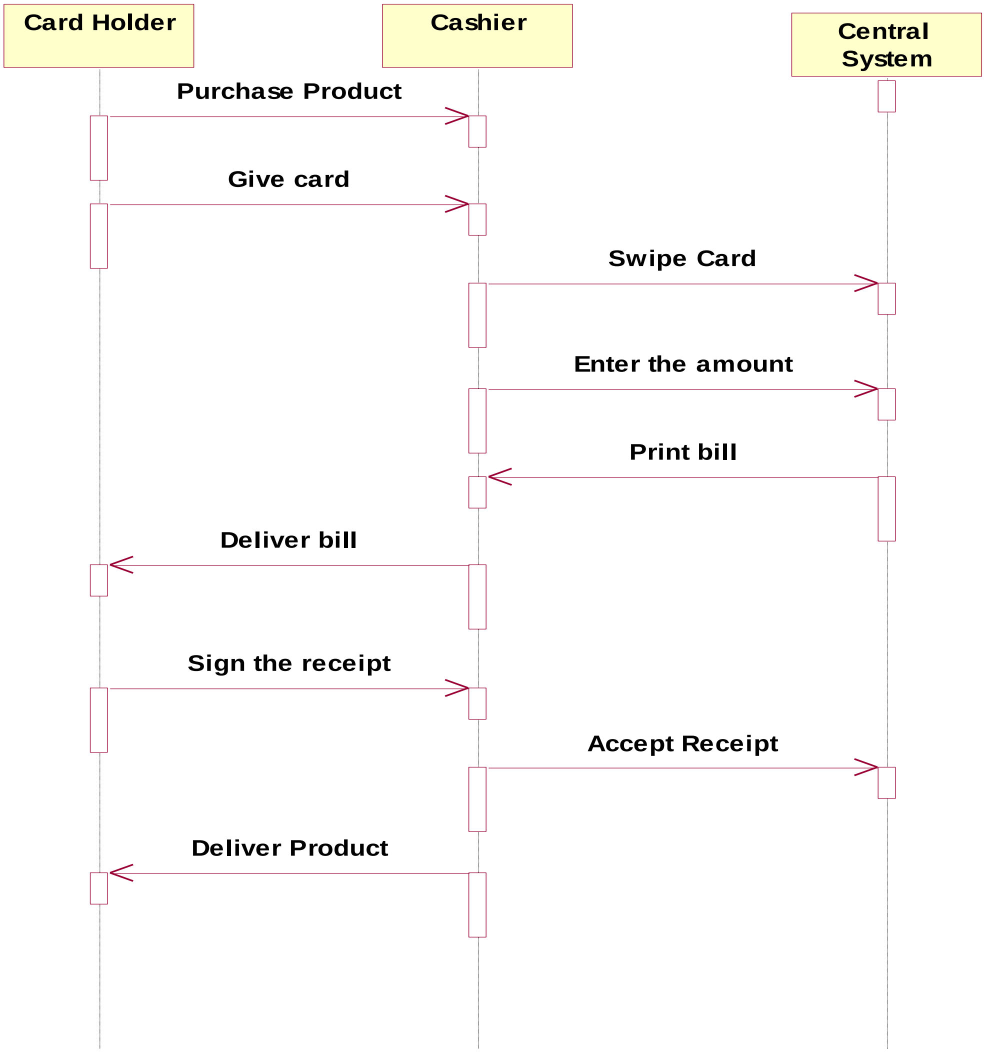


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

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



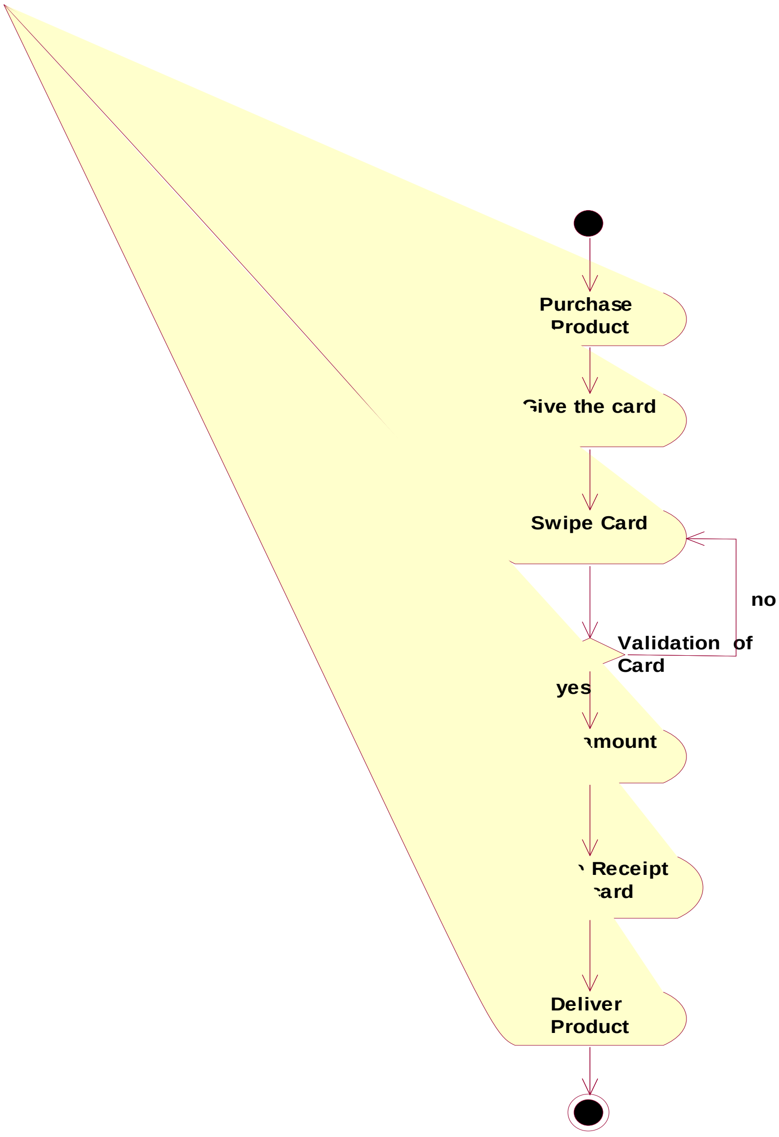
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Receive Bill, Give card, Enter the card number, Enter the amount, Transaction, Receive Receipt

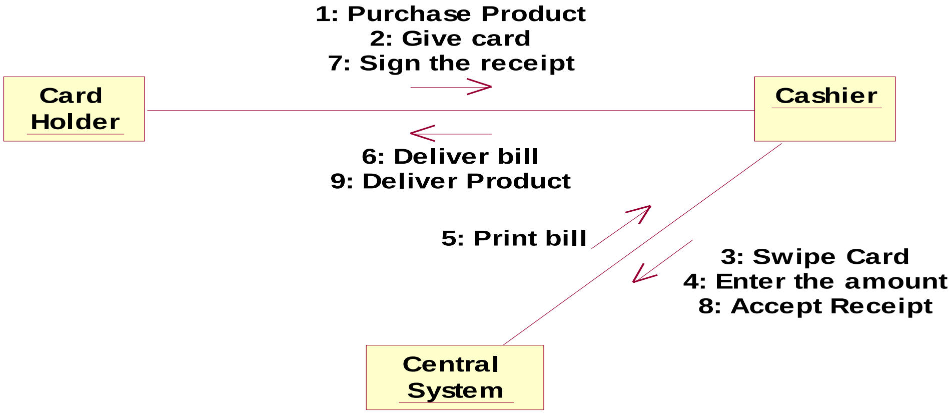
**Decision box:** Verification of card



## Collaboration Diagram:

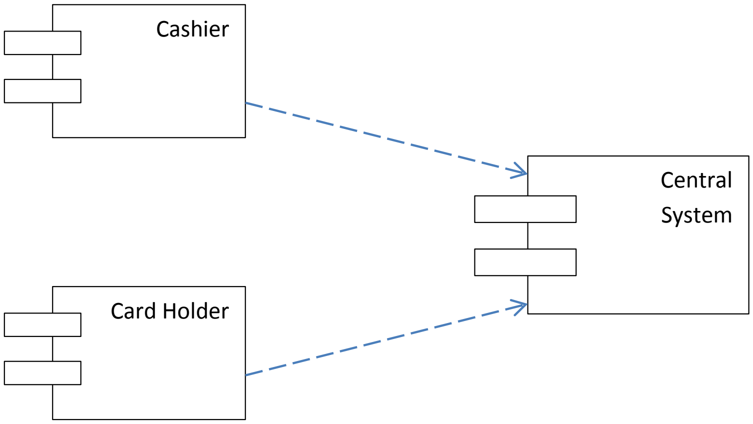
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



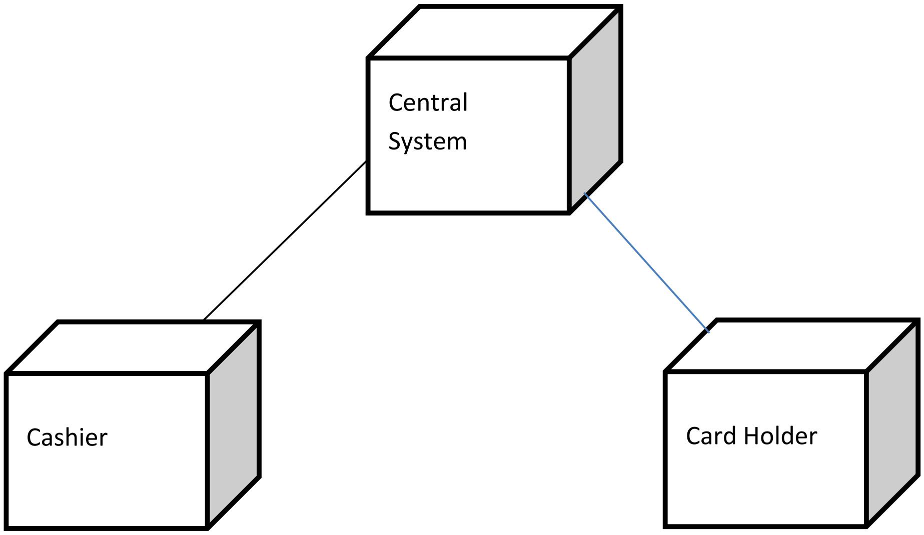
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**Cashier.java**

```
public class Cashier
{
   private int ProductName;
   private int CostOfProduct;
   public Centralsystem theCentralsystem;

   public Cashier()
   {    }
   public void EnterAmount()
   {   }

   public void SwipeCard()
   {   }

   public void PrintBill()
   {   }

   public void DeliverProduct()
   {   }
}
```

**RESULT:**

Thus the project CREDIT CARD PROCESSING was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# SOFTWARE PERSONNEL MANAGEMENT SYSTEM

**EX NO : 8**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the SOFTWARE PERSONNEL MANAGEMENT SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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 crosschecking the person's details.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15'' TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

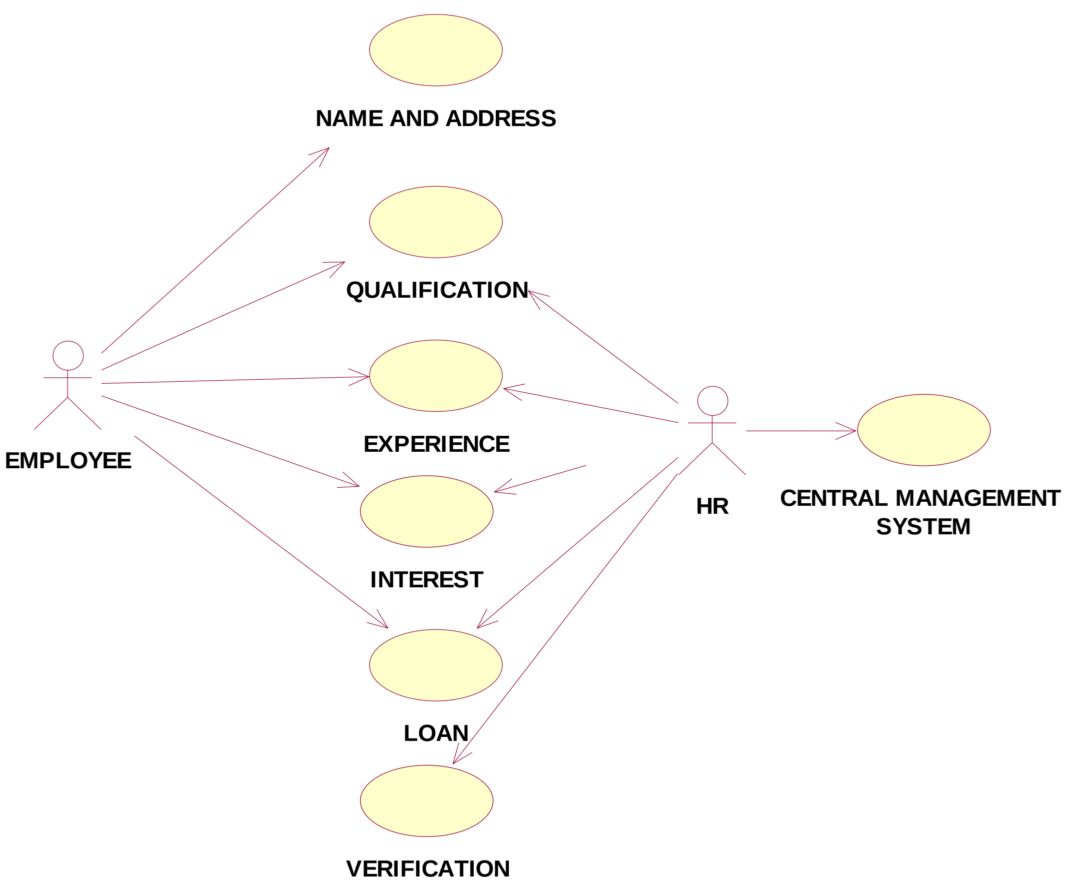
- Actor:

  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Employee, HR, Central system.
**Use case**: Name and address ,qualification ,experience, internet, loan, verification



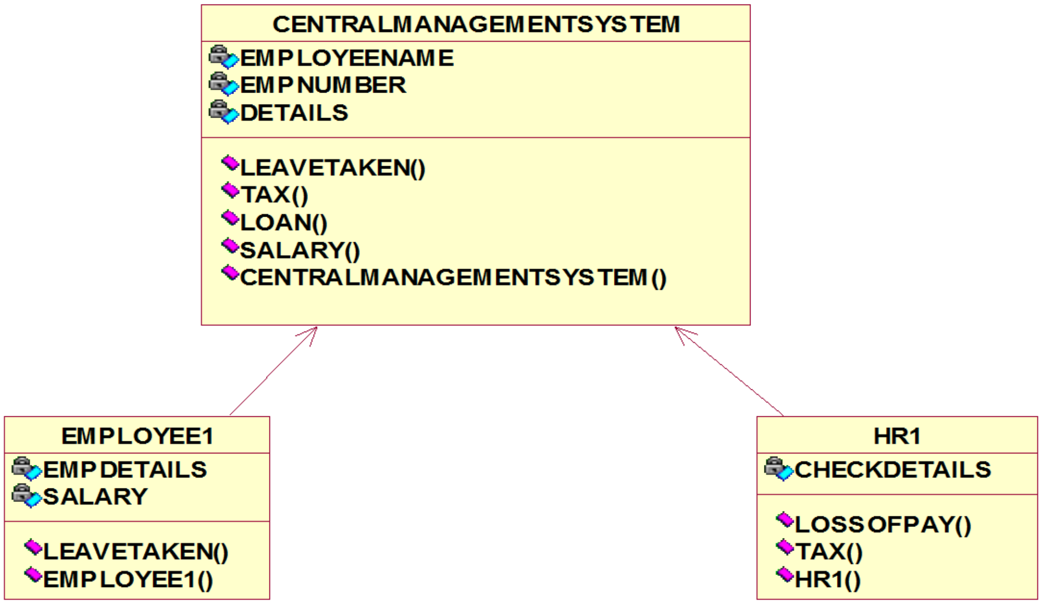
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| CLASSES | ATTRIBUTES | OPERATIONS |
|---|---|---|
| Central management system | Employeename,<br>Employeenumber | Tax()<br>Loan() |
| Employee1 | Employee details | Leave taken() |
| HR | checkdetails | Loss of pay() |

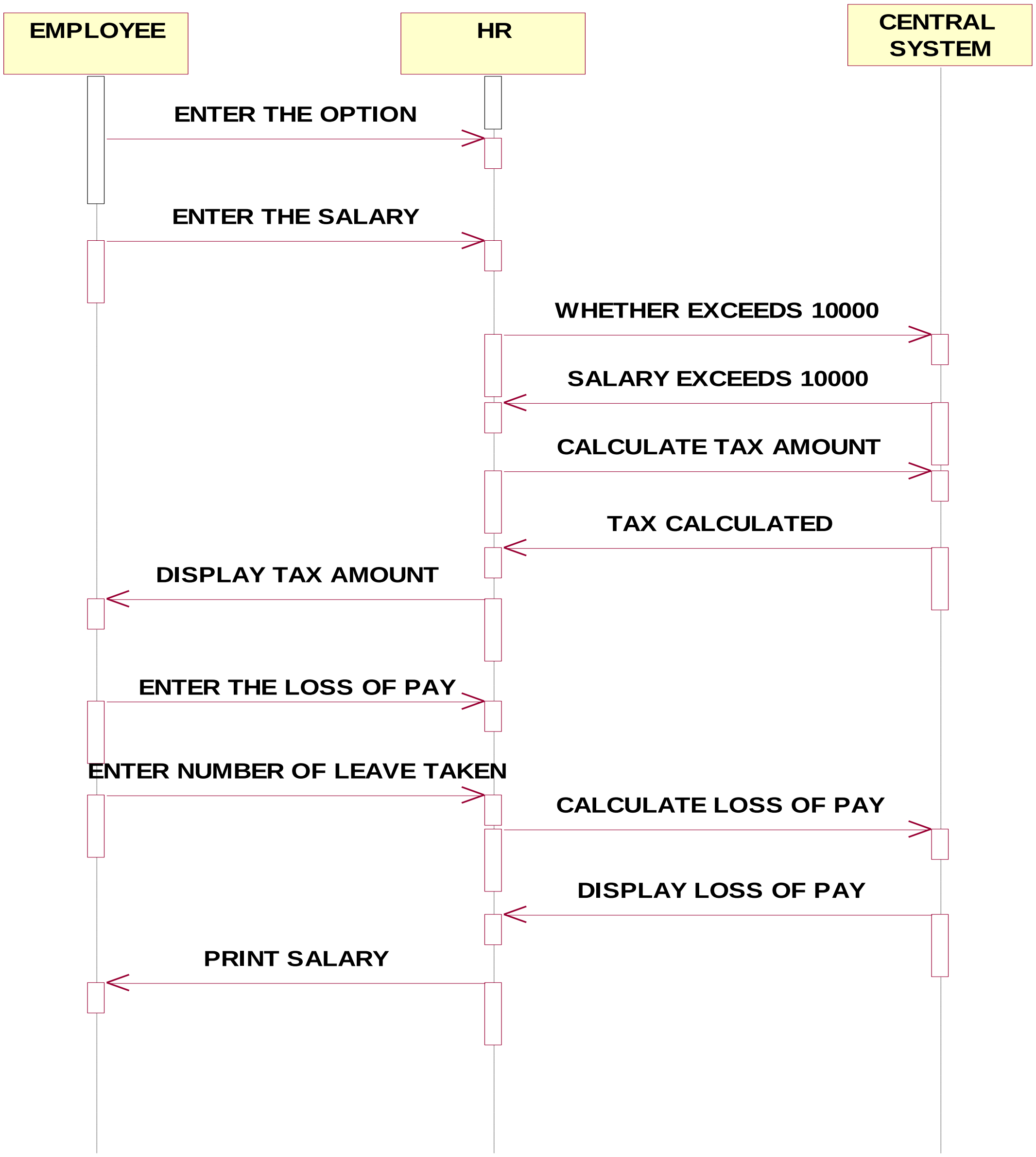


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object :**Employee, HR, Central system



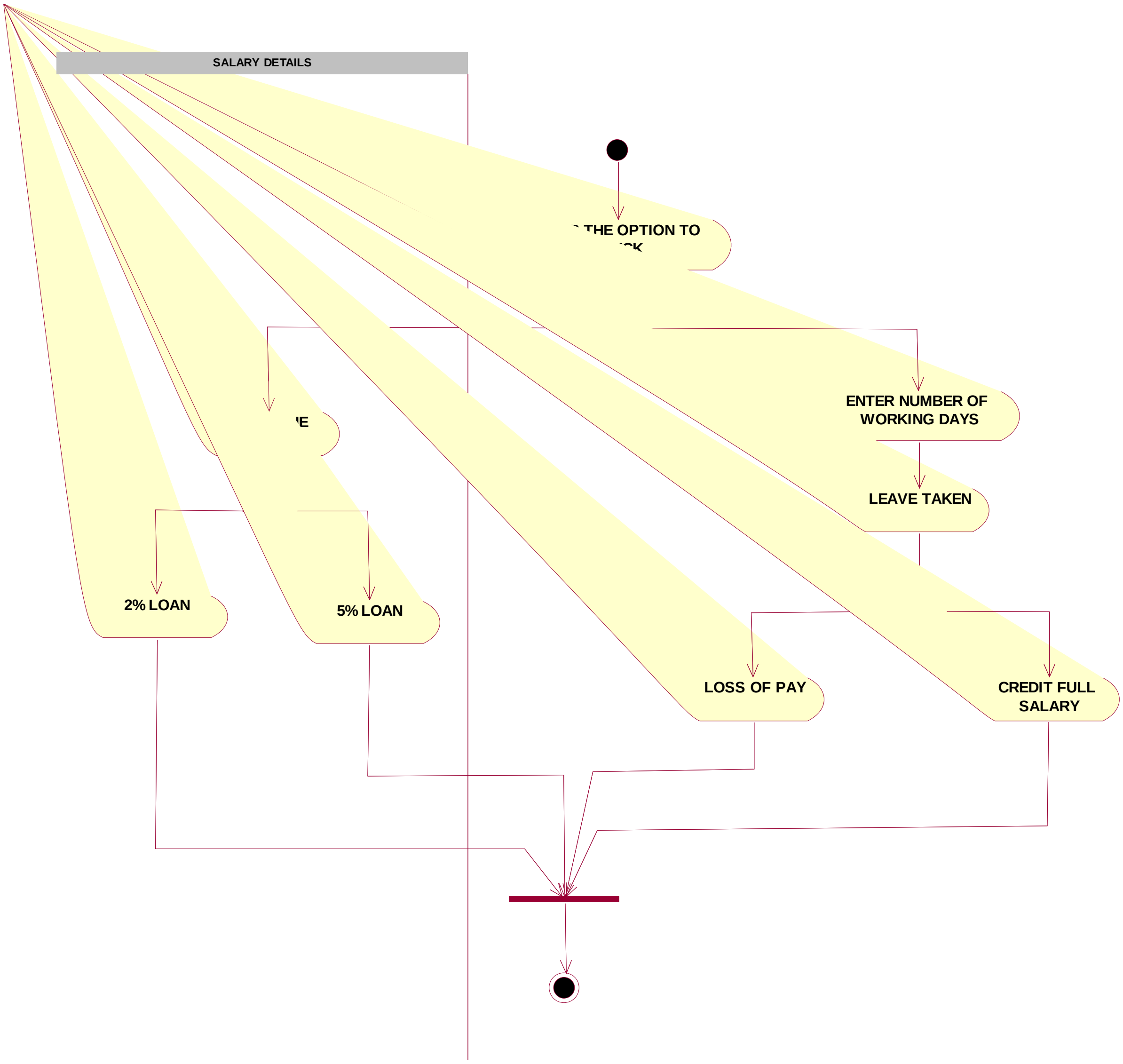
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

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

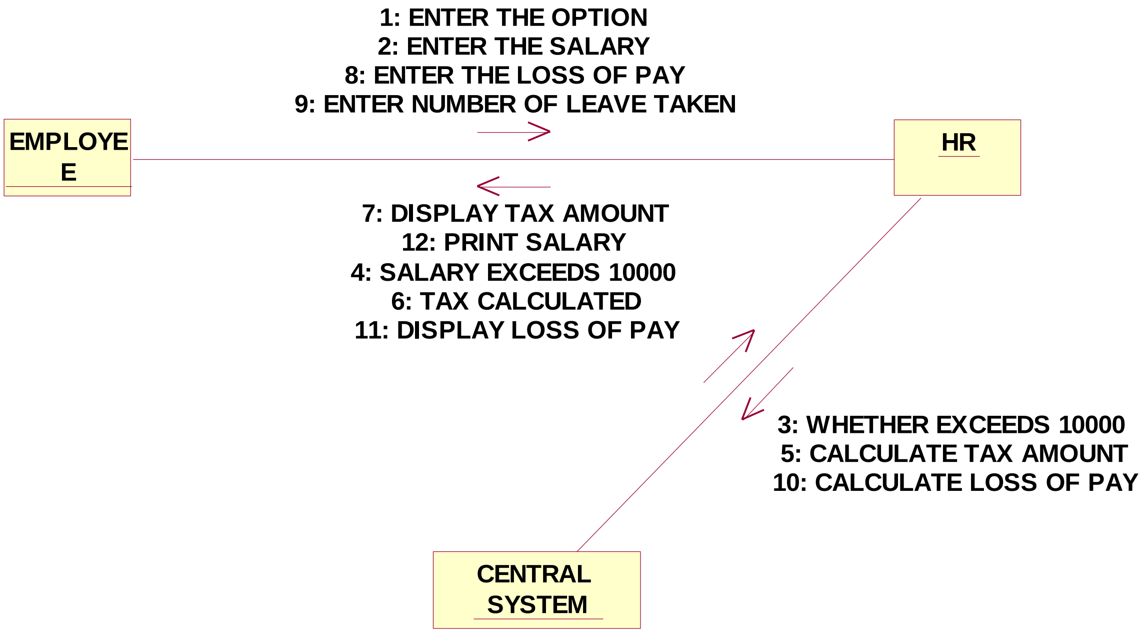
**Decision box:** Option to check



## Collaboration Diagram:

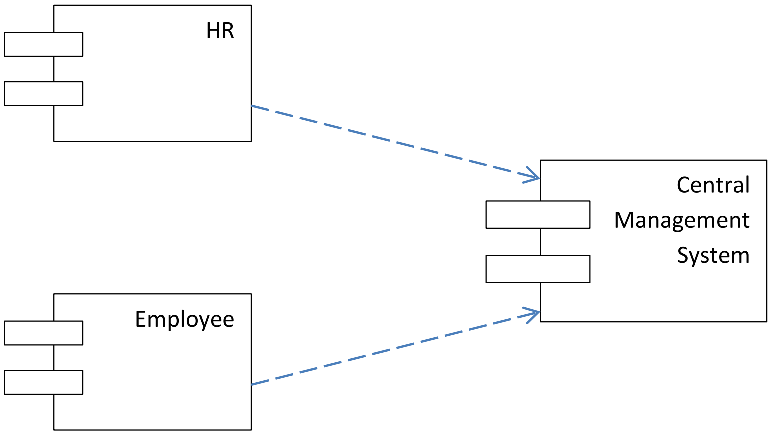
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



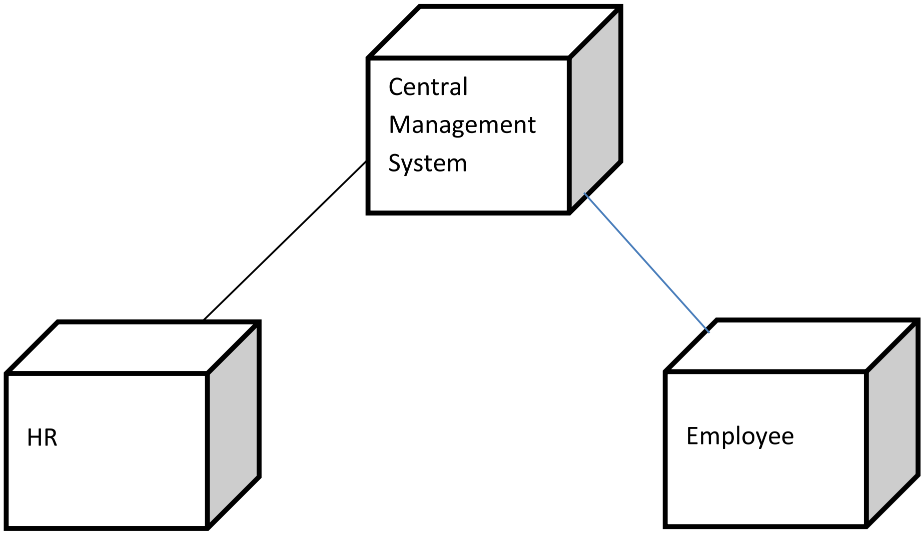
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**CENTRALMANAGEMENTSYSTEM.java**

```
public class CENTRALMANAGEMENTSYSTEM
{
  private int EMPLOYEENAME;
  private int EMPNUMBER;
  private int DETAILS;


  public CENTRALMANAGEMENTSYSTEM()
  {  }

  public void LEAVETAKEN()
  {   }

  public void TAX()
  {     }

  public void LOAN()
  {   }
  public void SALARY()
  {    }
}
```

**EMPLOYEE1.java**

```
public class EMPLOYEE1
{
  private int EMPDETAILS;
  private int SALARY;
  public CENTRALMANAGEMENTSYSTEM theCENTRALMANAGEMENTSYSTEM;


  public EMPLOYEE1()
  {     }

  public void LEAVETAKEN()
  {   }
```

**RESULT:**

Thus the project SOFTWARE PERSONNEL MANAGEMENT SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# E- BOOK MANAGEMENT SYSTEM

**EX NO : 9**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the E-BOOK MANAGEMENT SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

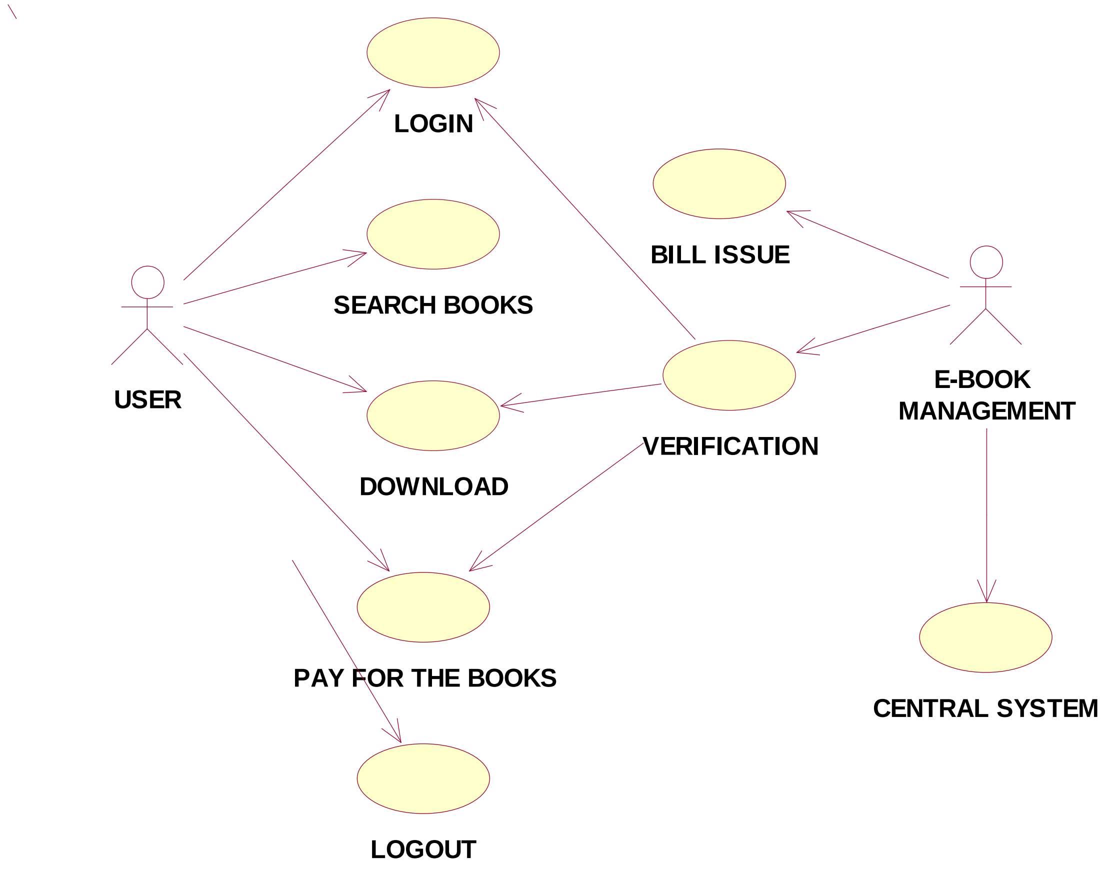
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

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

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



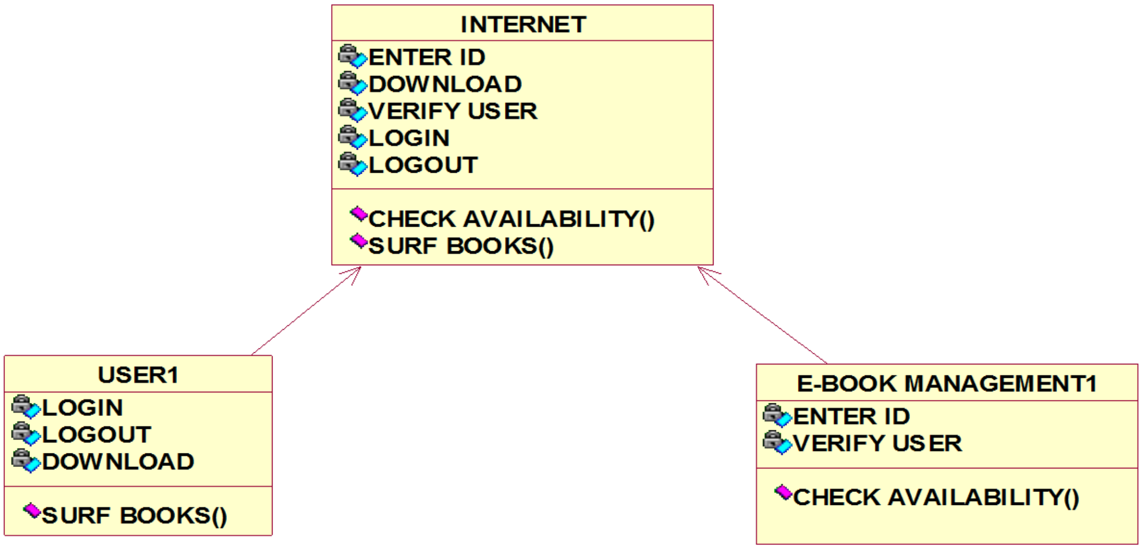
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

| **CLASSES** | **ATTRIBUTES** | **OPERATIONS** |
|---|---|---|
| Internet | Enter id<br>Login, logout | Surf books() |
| User | Login ,logout | Surf books() |
| E-book management | Verify user | Check availability() |

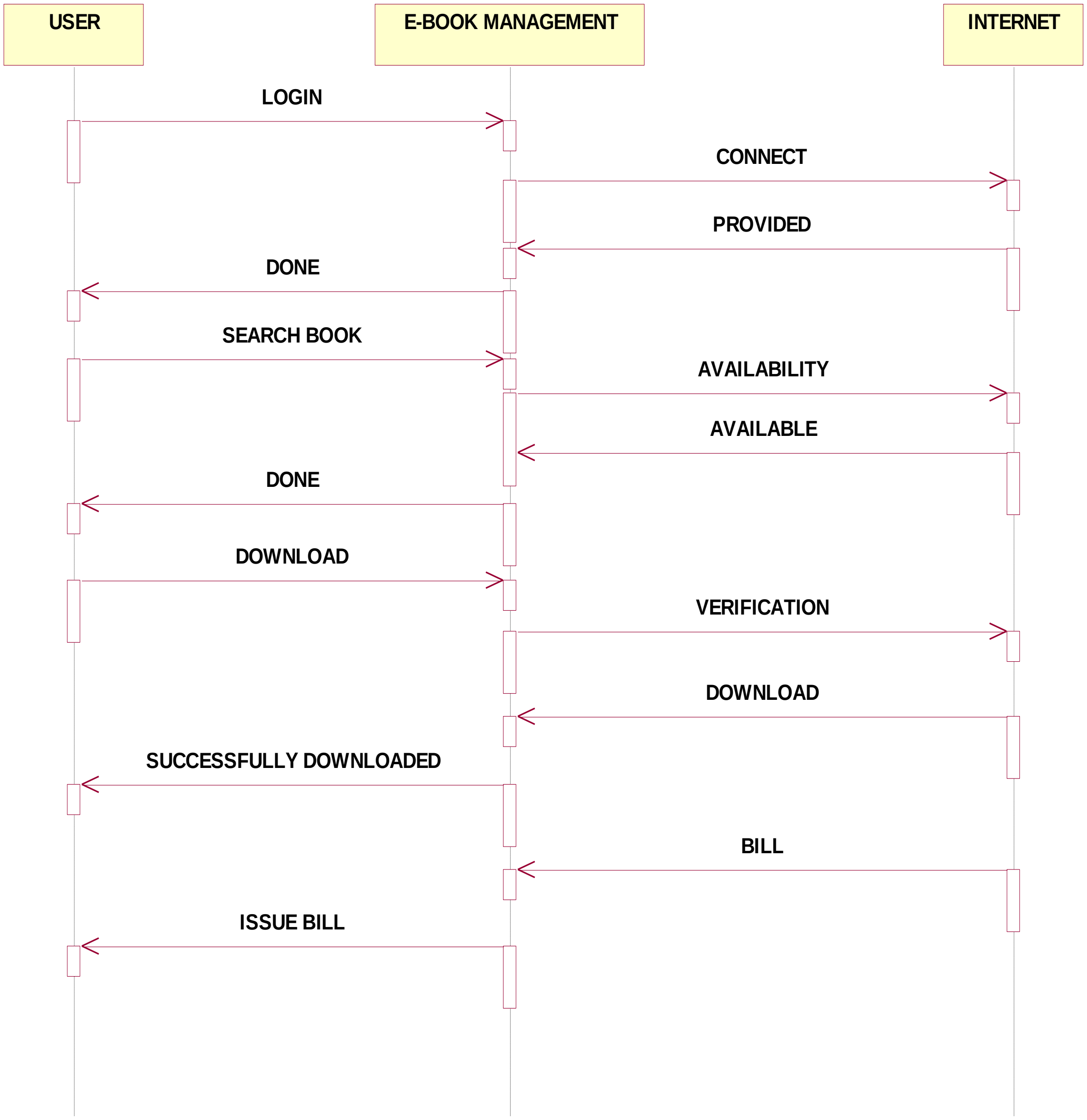


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

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



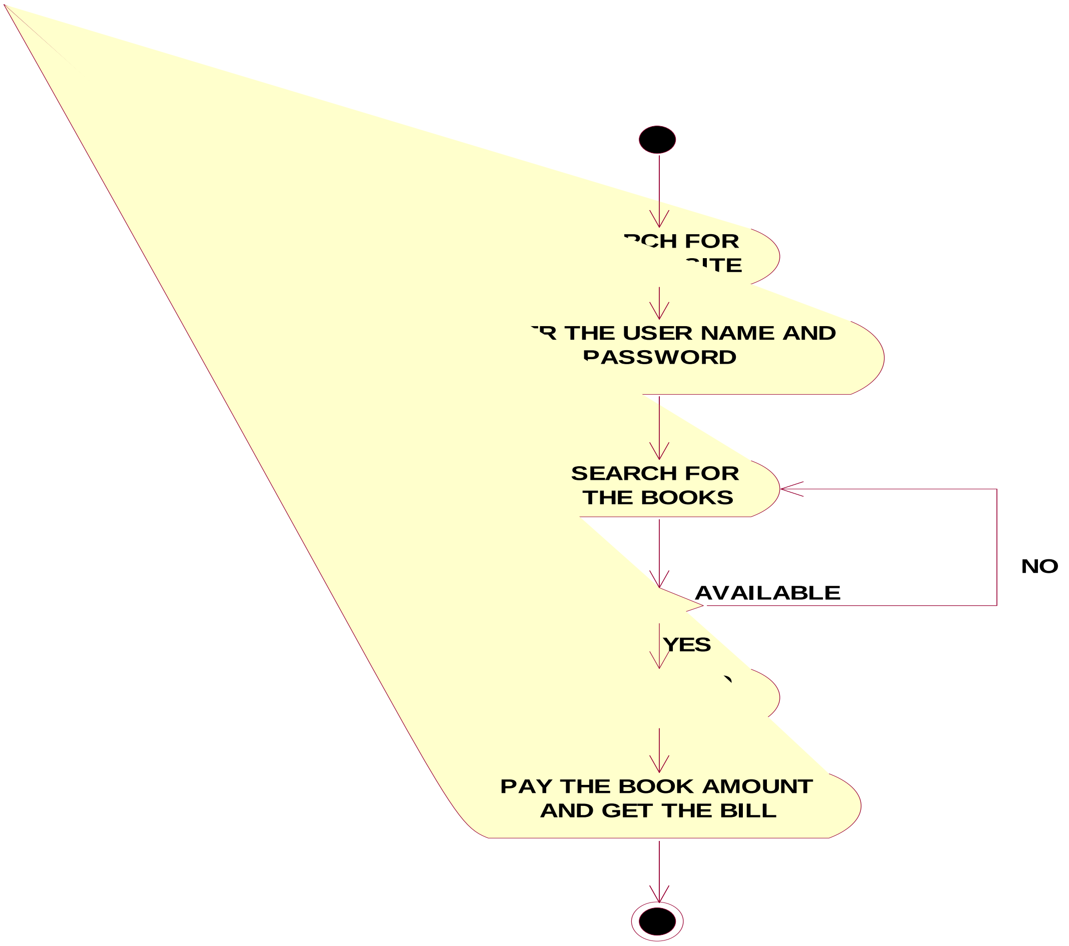
## Activity Diagram:

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

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

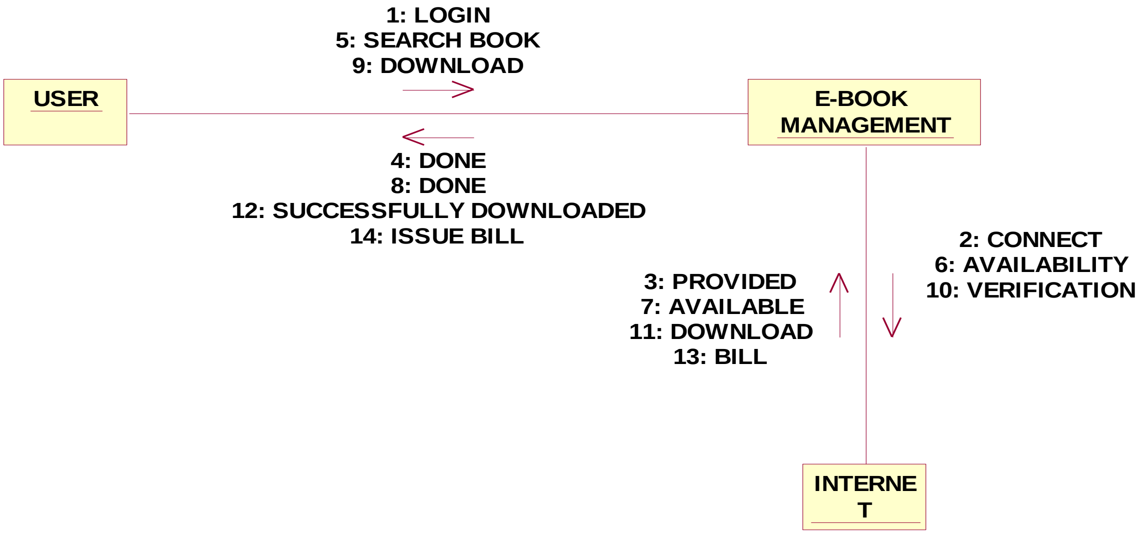
**Decision box:** check availability



## Collaboration Diagram:

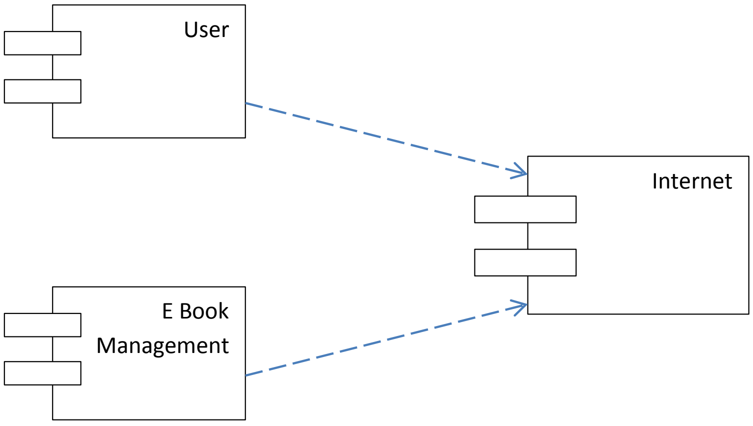
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



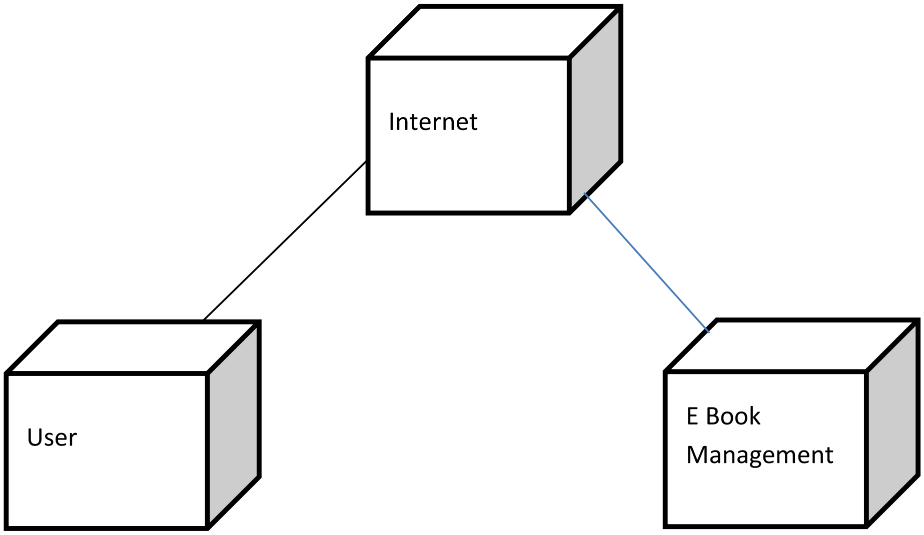
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**USER1.java**

```
public class USER1
{
   private int LOGIN;
   private int LOGOUT;
   private int DOWNLOAD;
   public INTERNET theINTERNET;

   public USER1()
   {     }

   public void SURFBOOKS()
   {   }
}
```

**RESULT:**

Thus the project E-BOOK MANAGEMENT SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# RECRUITMENT SYSTEM

**EX NO : 10**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the RECRUITMENT SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

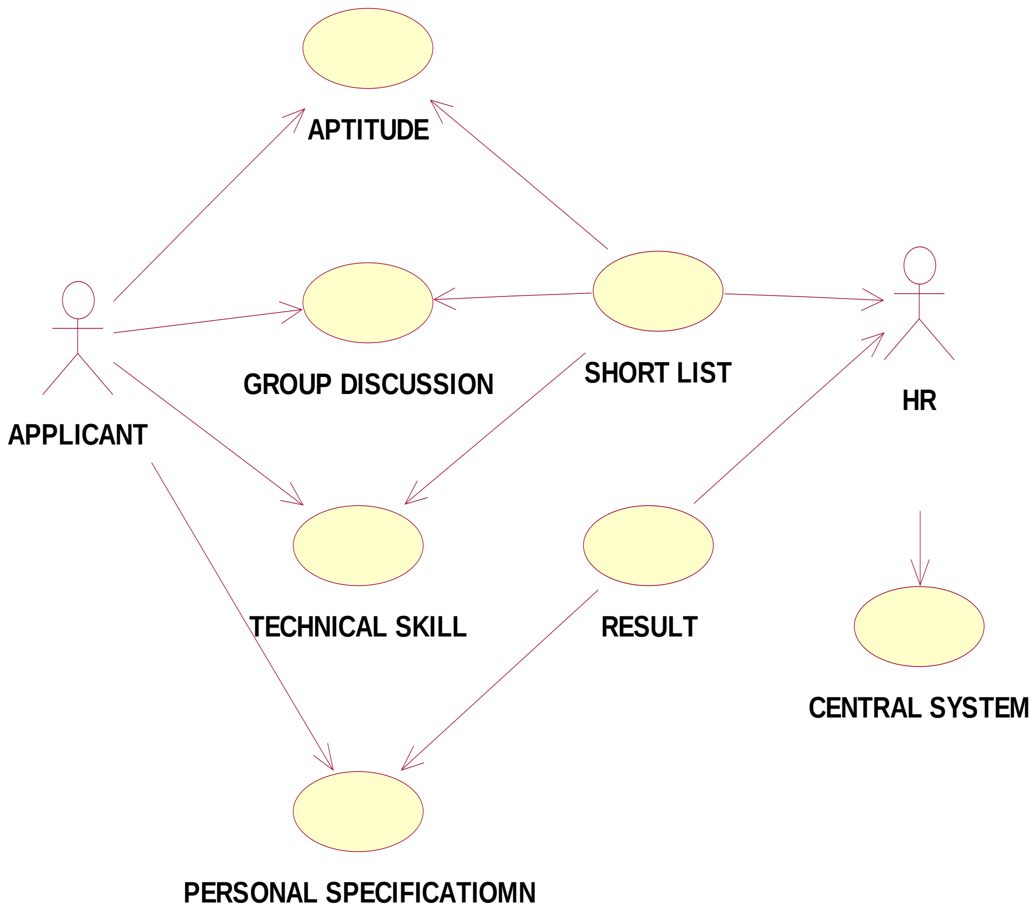
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

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

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



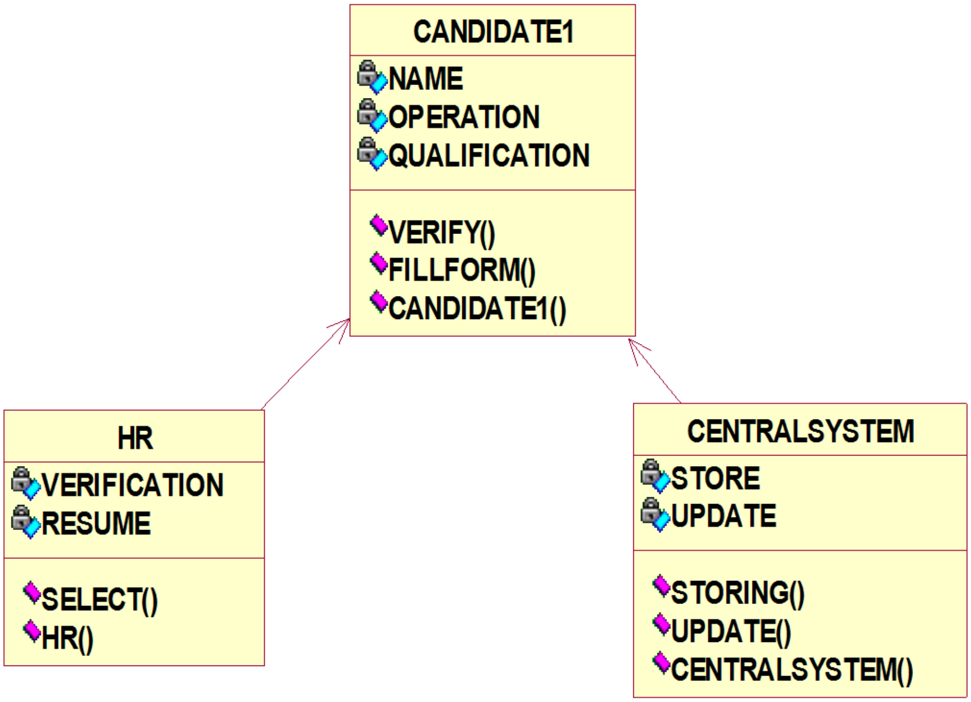
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

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

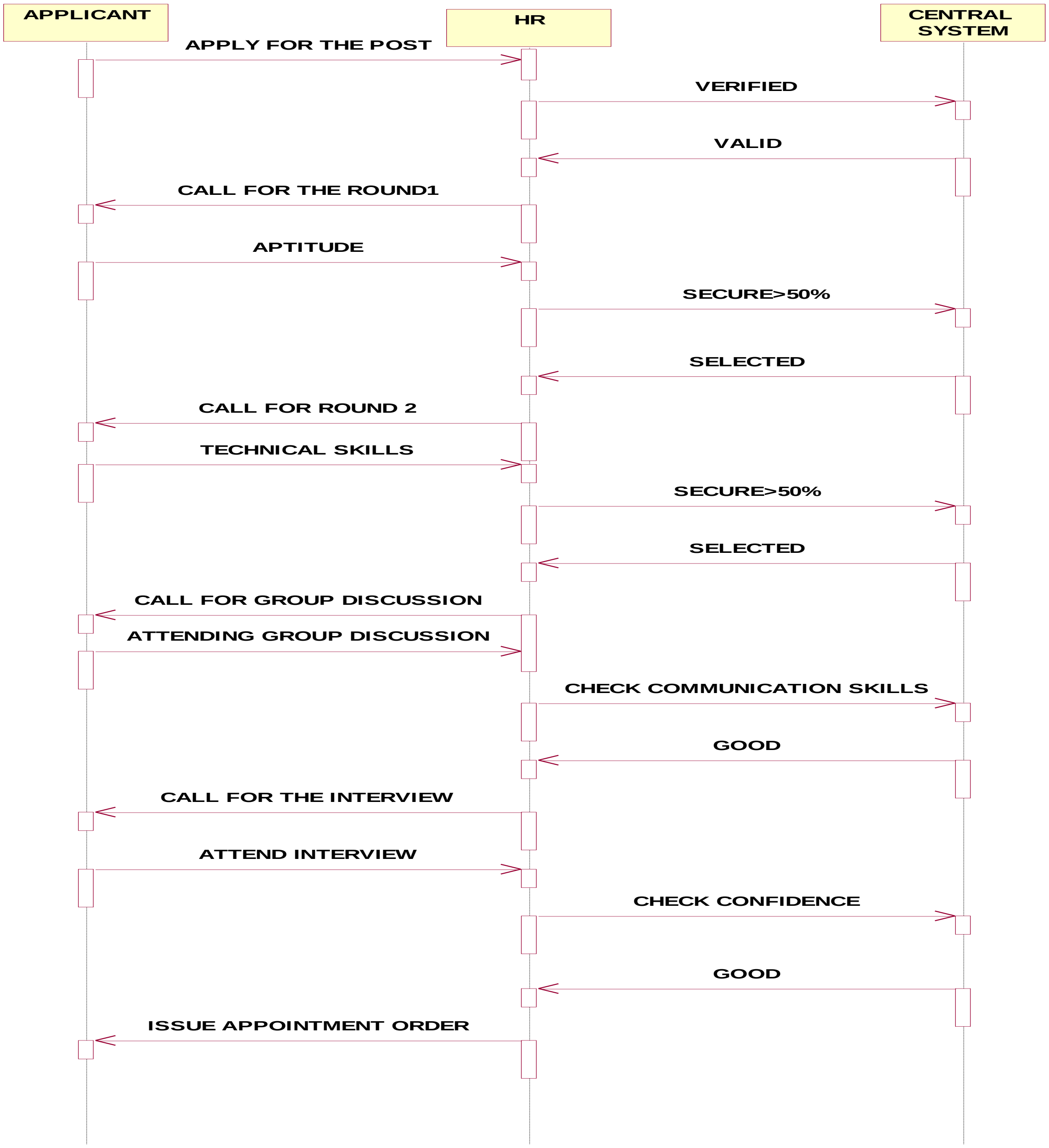


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object:** Candidate, HR, Central system



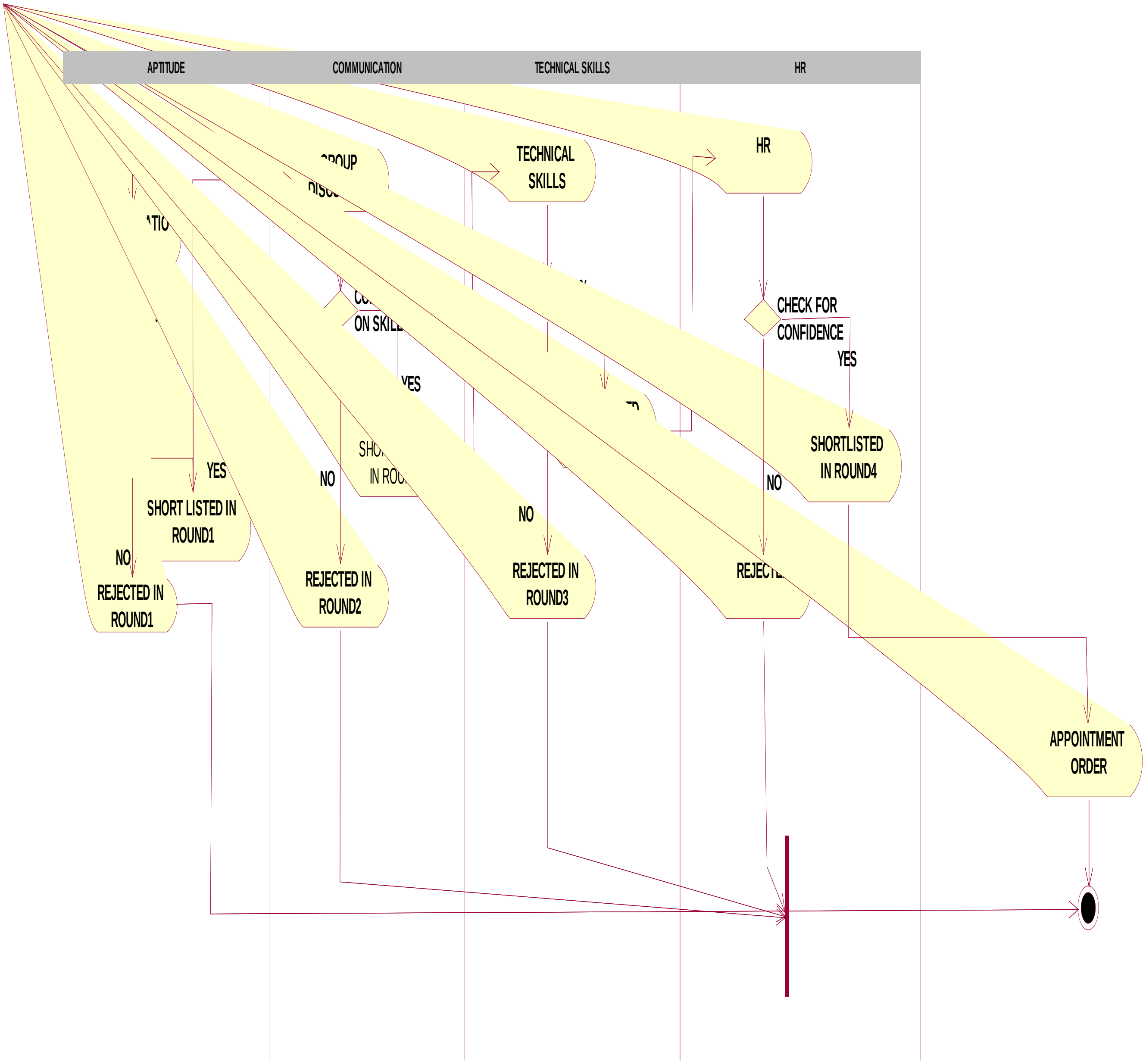
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

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

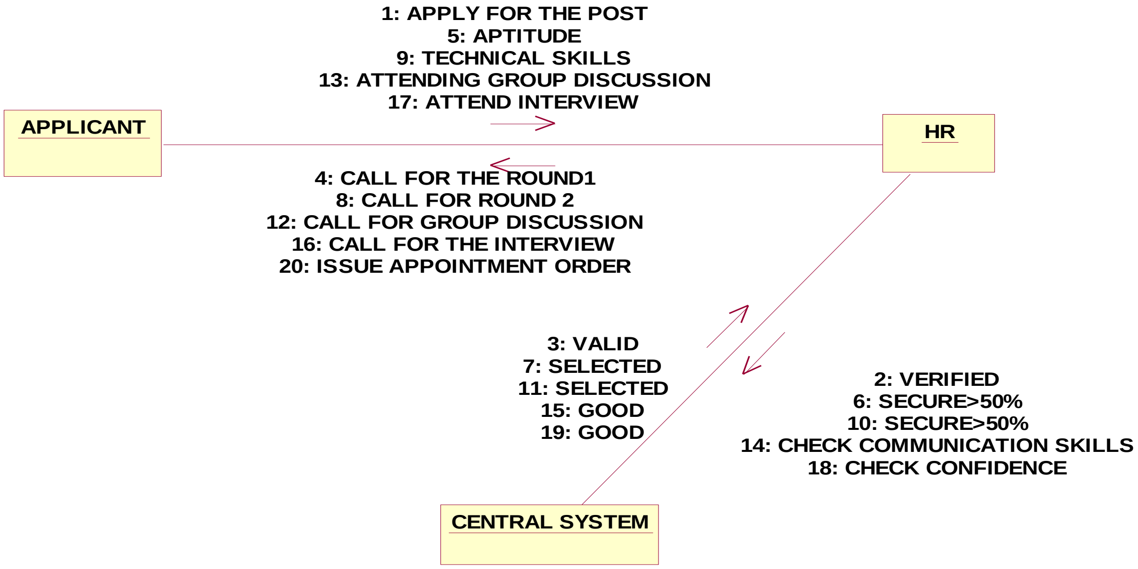
**Decision box:** Verification of the qualities



## Collaboration Diagram:

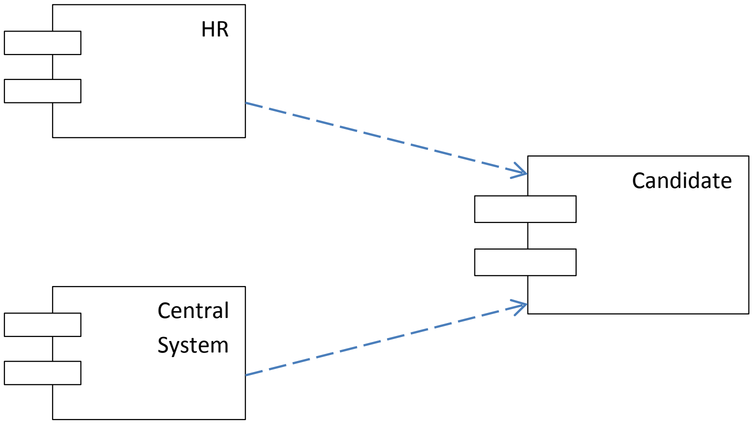
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



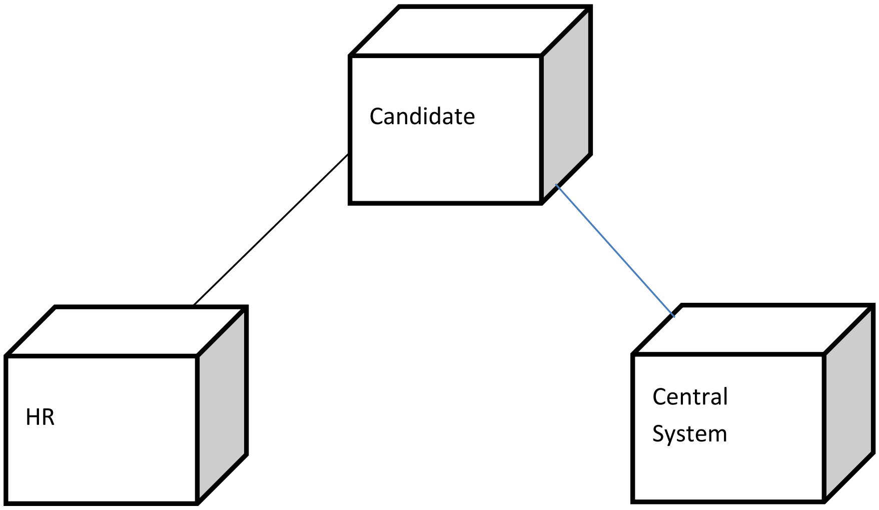
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**CANDIDATE1.java**

```
public class CANDIDATE1
{
   private int NAME;
   private int OPERATION;
   private int QUALIFICATION;

   public CANDIDATE1()
   {    }

   public void VERIFY()
   {    }

    public void FILLFORM()
   {   }
}
```

**HR.java**

```
public class HR
{
  private int VERIFICATION;
  private int RESUME;
  public CANDIDATE1 theCANDIDATE1;


  public HR()
  {    }

  public void SELECT()
  {}
}
```

**RESULT:**

Thus the project RECRUITMENT SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# CONFERENCE MANAGEMENT SYSTEM

**EX NO : 11**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the CONFERENCE MANAGEMENT SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

**DESIGN :**

**Use case diagram:**

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

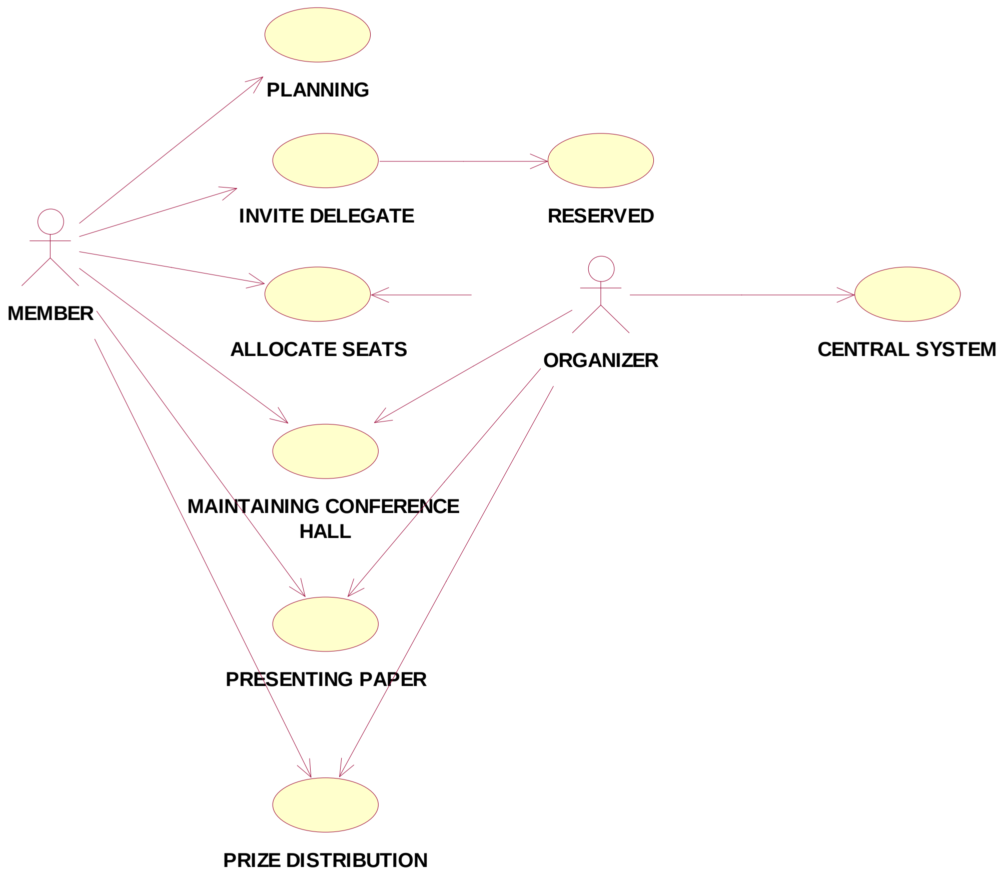
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

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

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



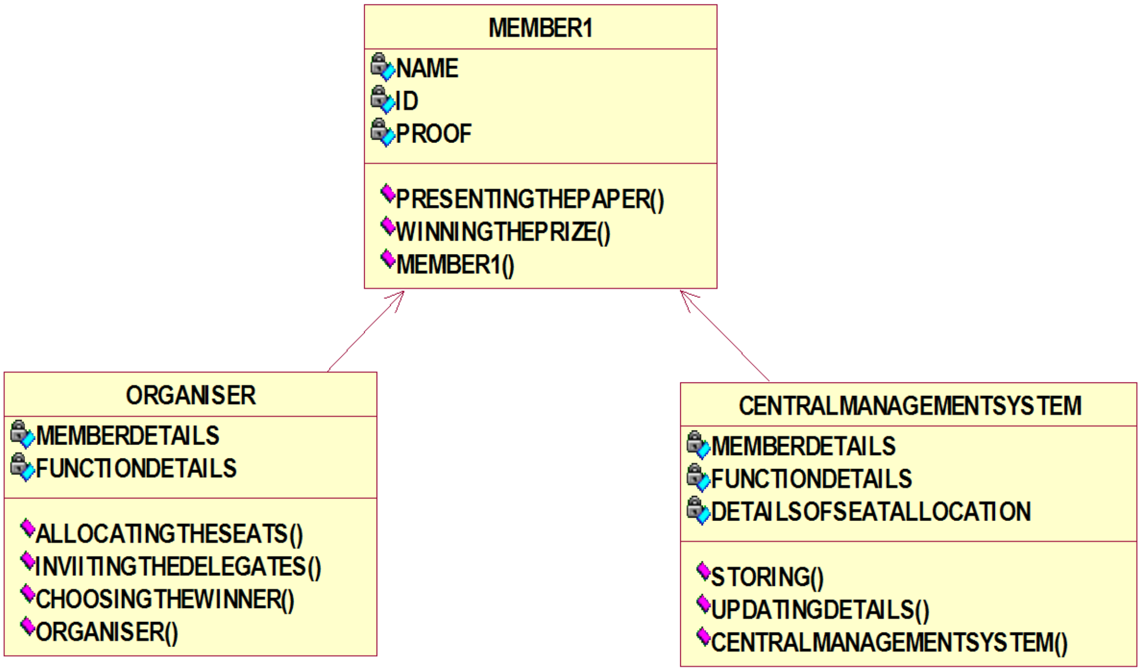
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

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

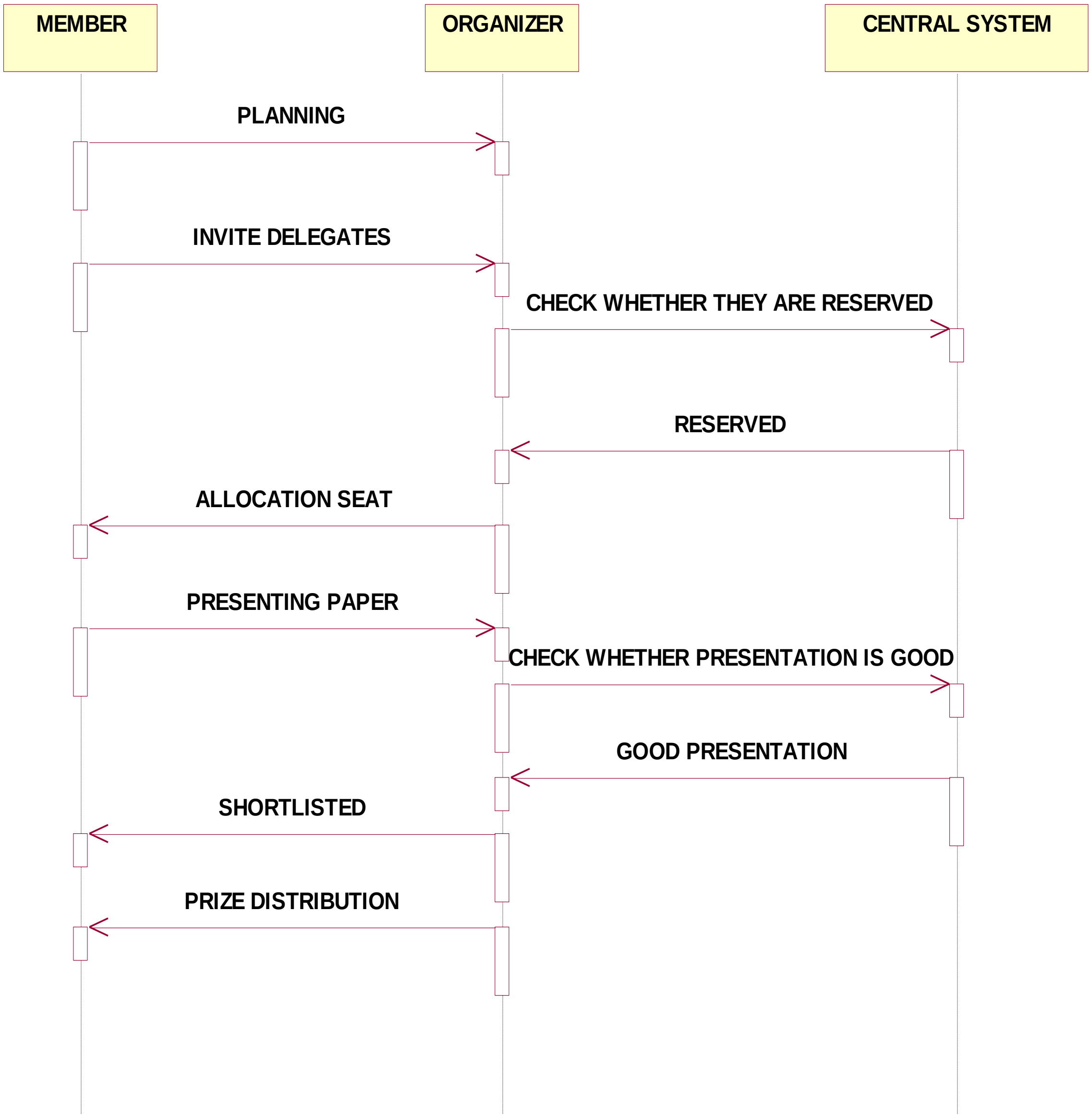


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

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

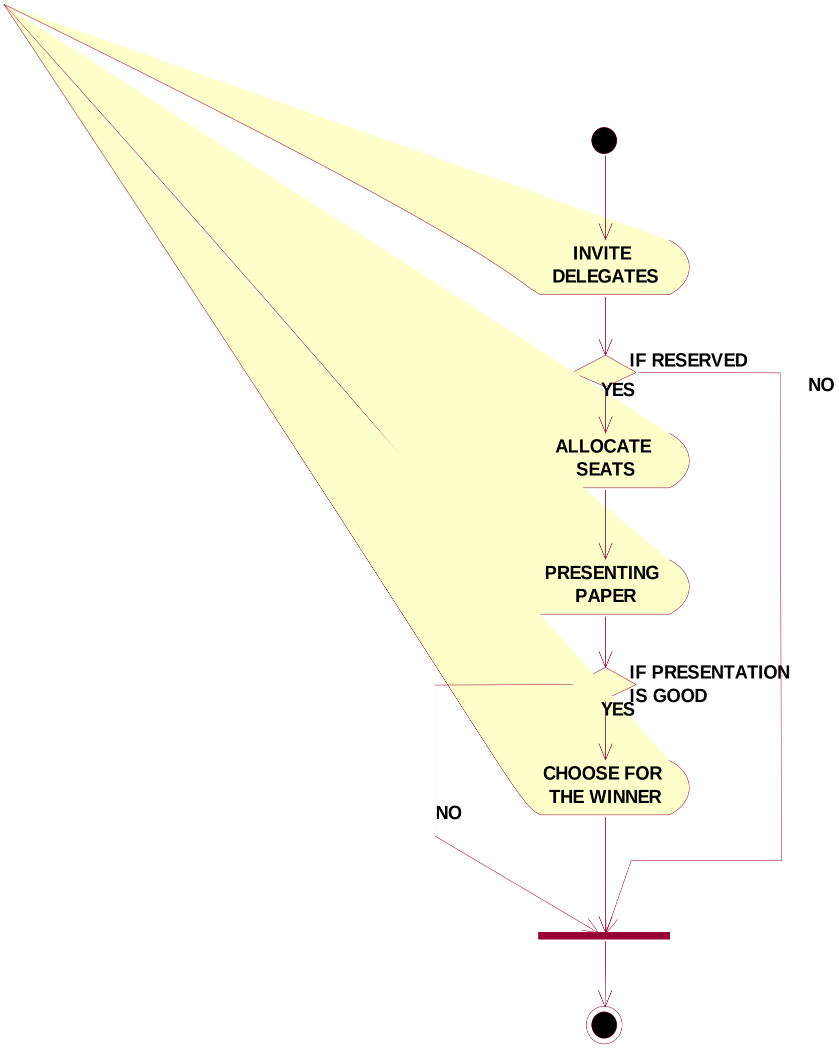


**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

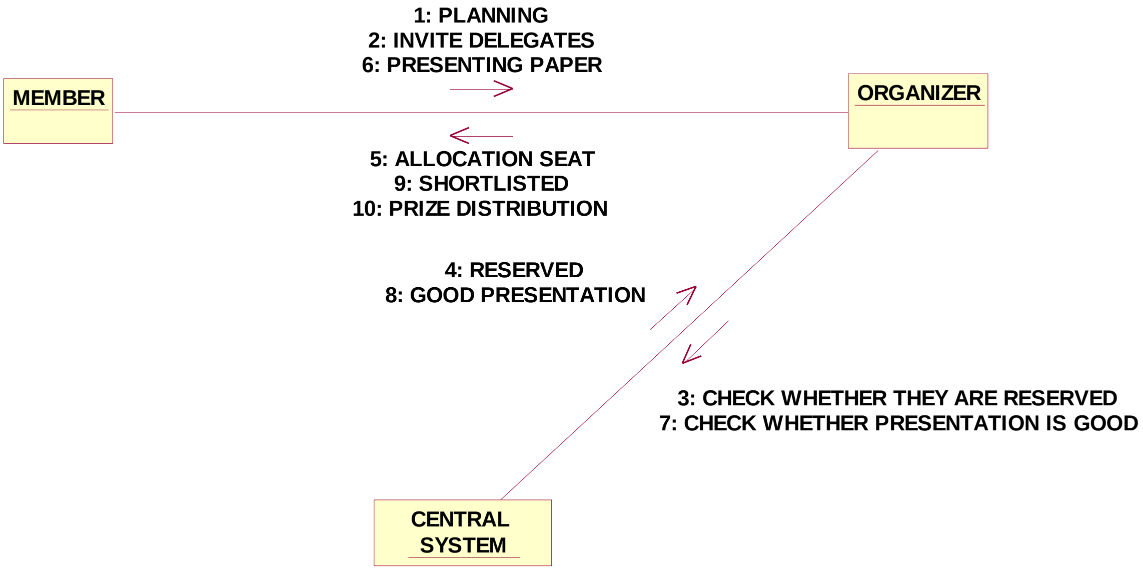
**Activities:** Invite delegates, Allocate seats, Presenting paper, Choose the winner
**Decision box:** Whether it is reserved or not, Whether the presentation is good or not



## Collaboration Diagram:

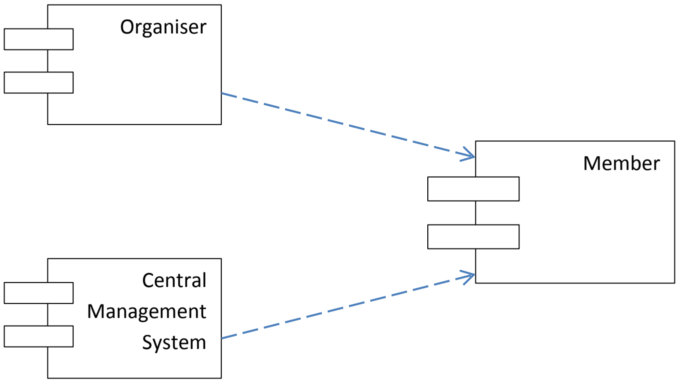
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



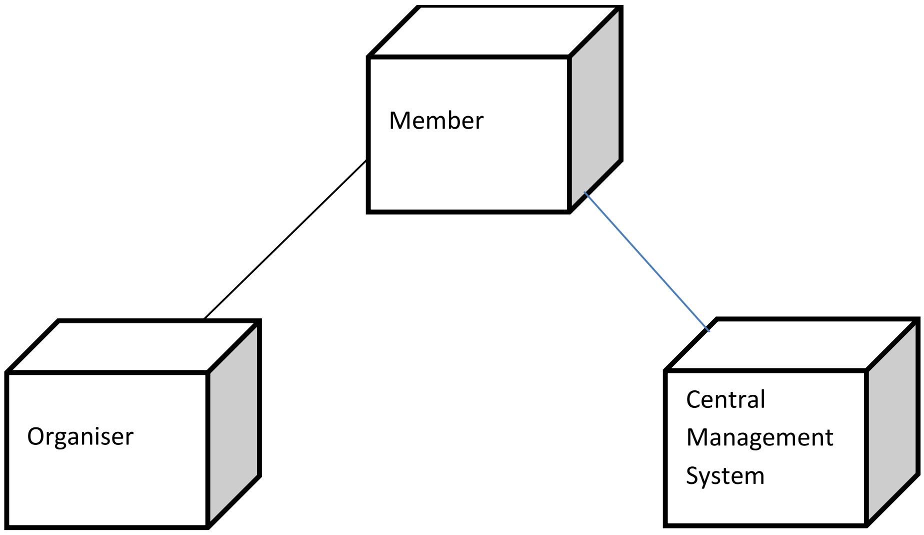
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**MEMBER1.java**

```
public class MEMBER1
{
  private int NAME;
  private int ID;
  private int PROOF;

  public MEMBER1()
  {   }

  public void PRESENTINGTHEPAPER()
  {   }

  public void WINNINGTHEPRIZE()
  {   }
}
```

**ORGANISER.java**

```
public class ORGANISER
{
  private int MEMBERDETAILS;
  private int FUNCTIONDETAILS;
  public MEMBER1 theMEMBER1;

  public ORGANISER()
  {   }

  public void ALLOCATINGTHESEATS()
  {   }

  public void INVIITINGTHEDELEGATES()
  {   }

  public void CHOOSINGTHEWINNER()
  {   }
}
```

**RESULT:**

Thus the project CONFERNECE MANAGEMENT SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# FOREIGN TRADING SYSTEM

**EX NO : 12**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the FOREIGN TRADING SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

This software is designed to maintain the details about the trading system that exists between the foreign countries. This details are hold by the trading management system.The details to the system are provided by the customer and the supplier

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

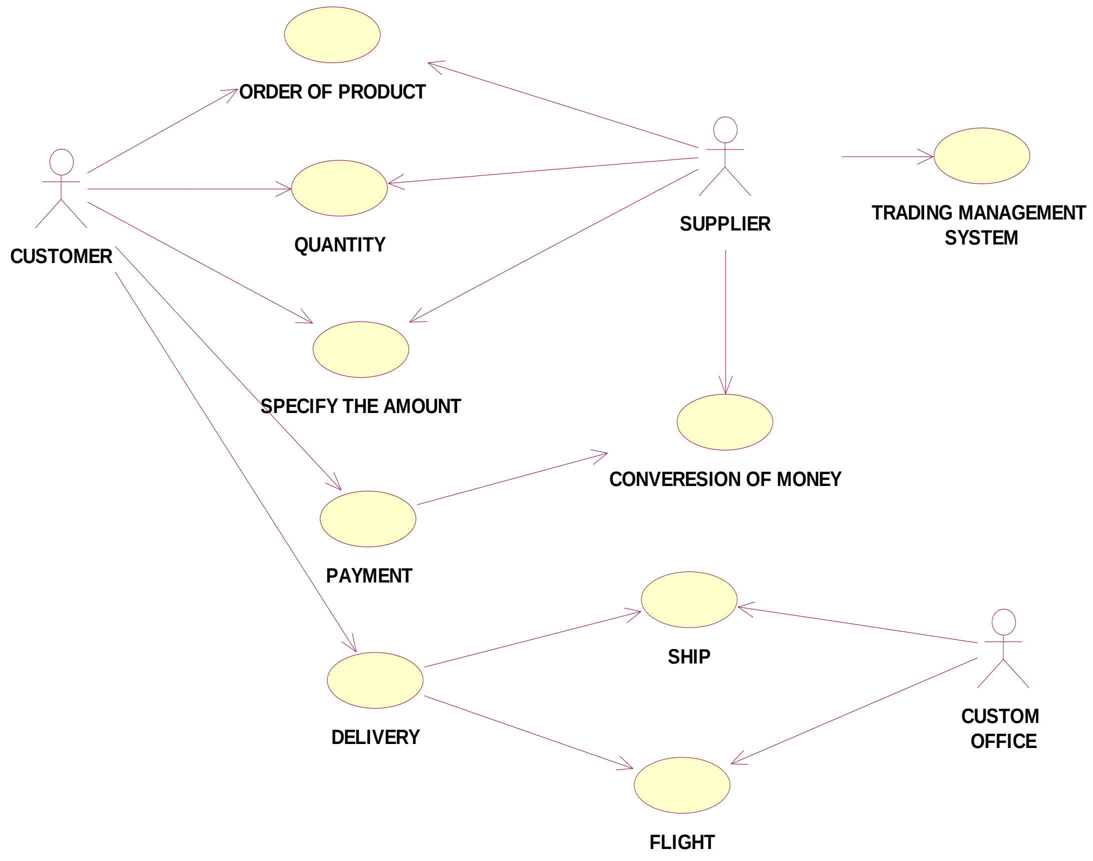
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors**: Customer, Supplier, Custom officer

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



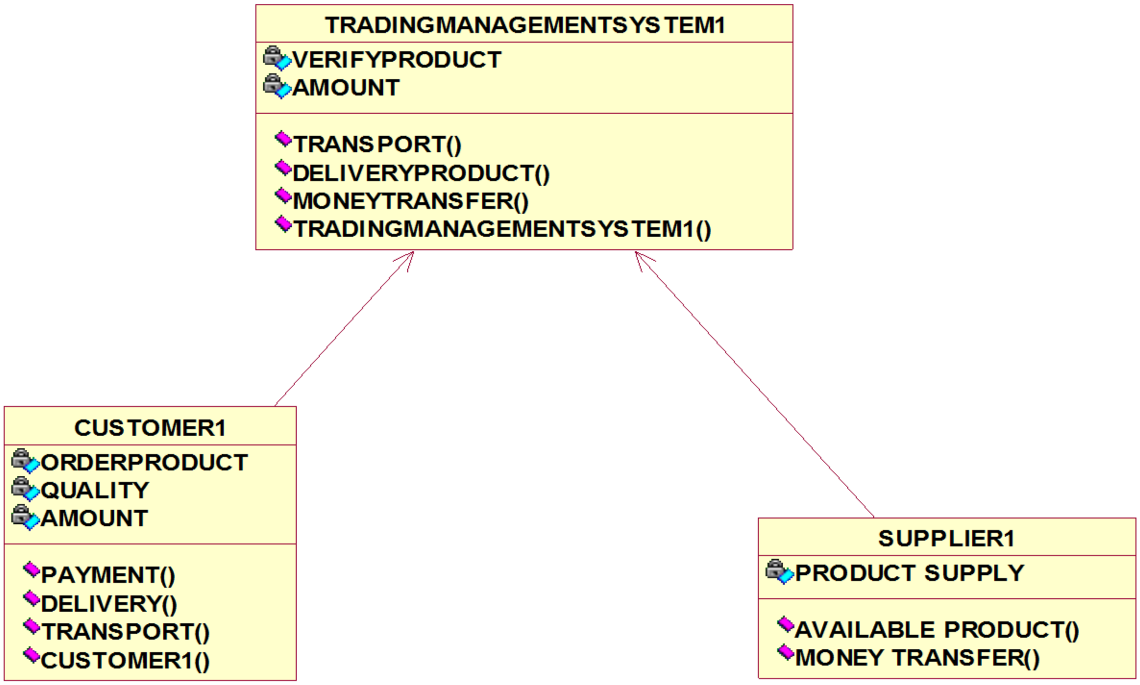
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

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

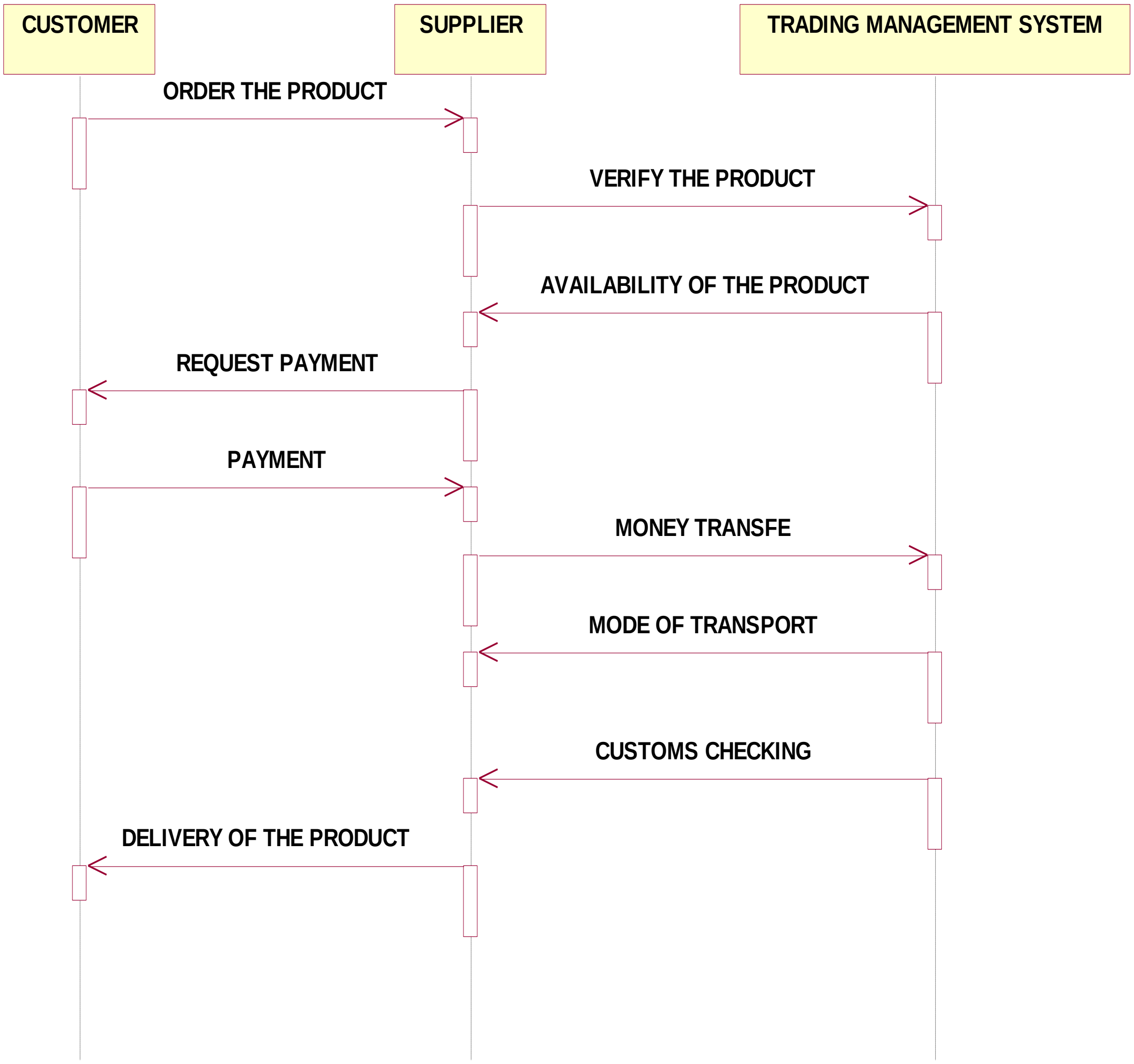


**Sequence Diagram:**

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

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



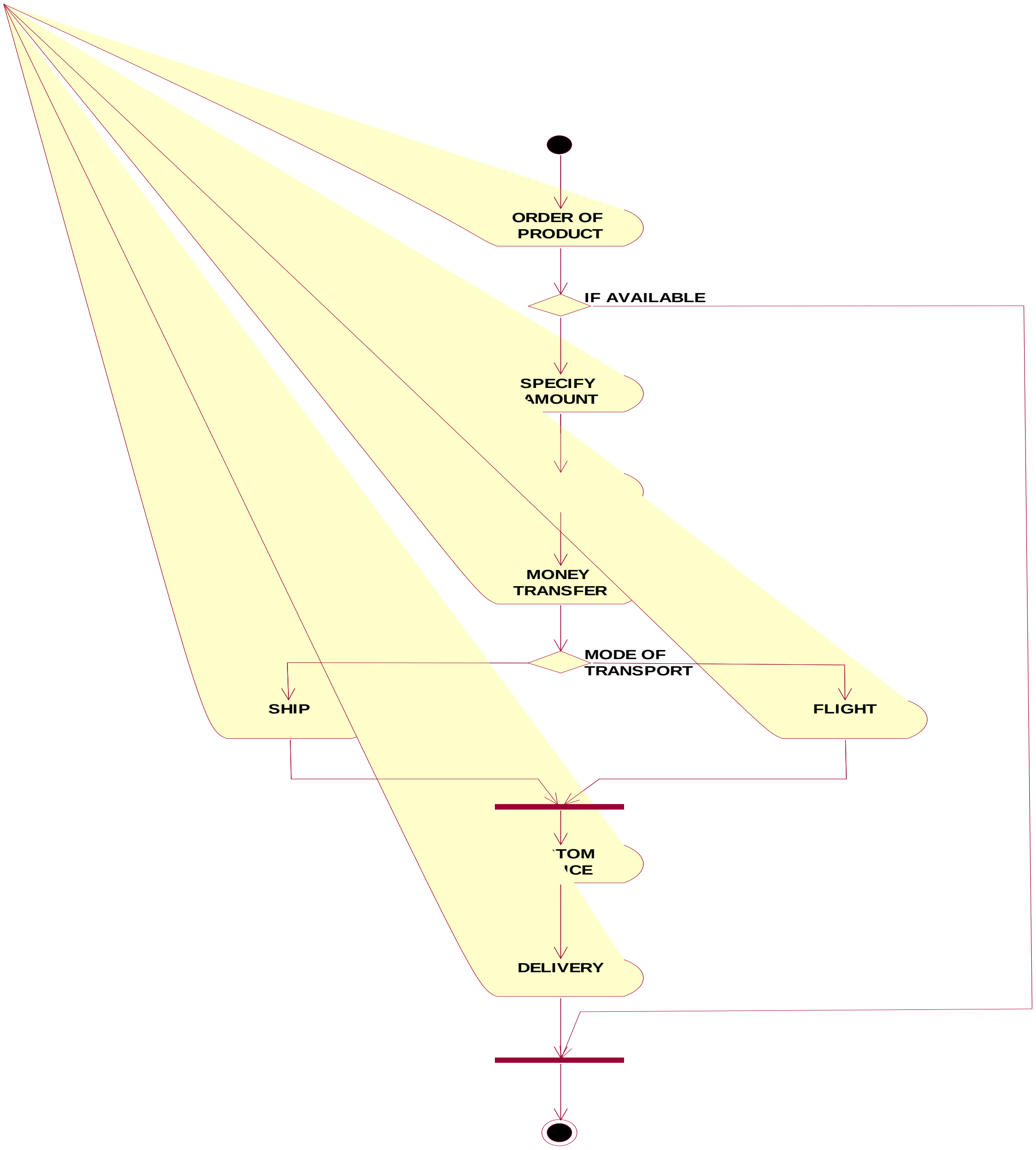
## Activity Diagram:

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Order of the product, Specify amount, Payment, Money transfer

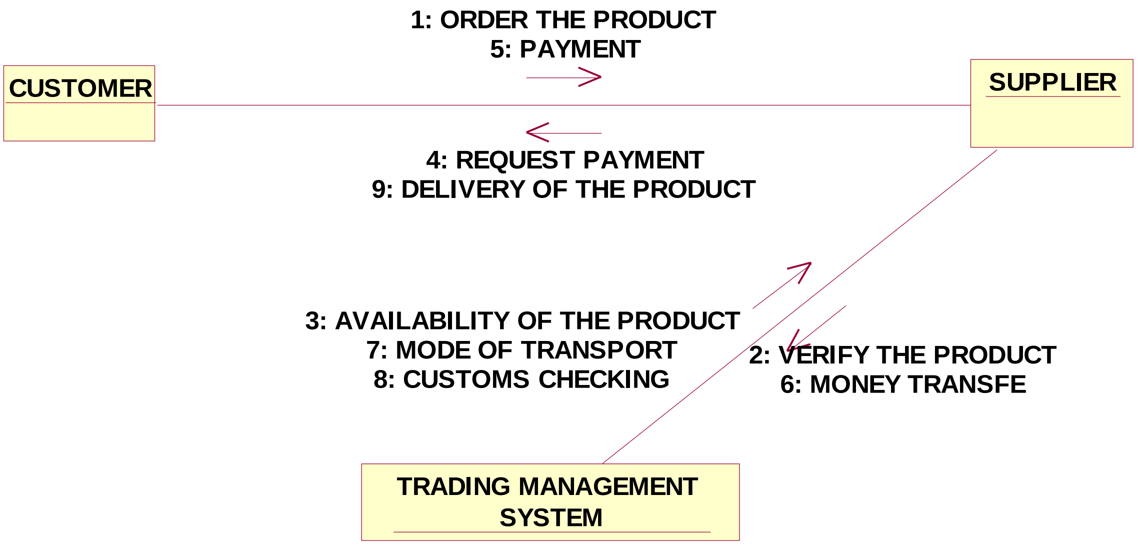
**Decision box:** Check for availability



## Collaboration Diagram:

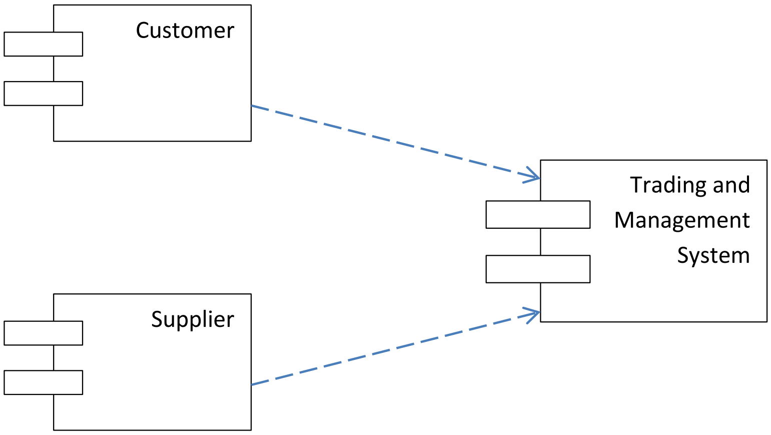
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



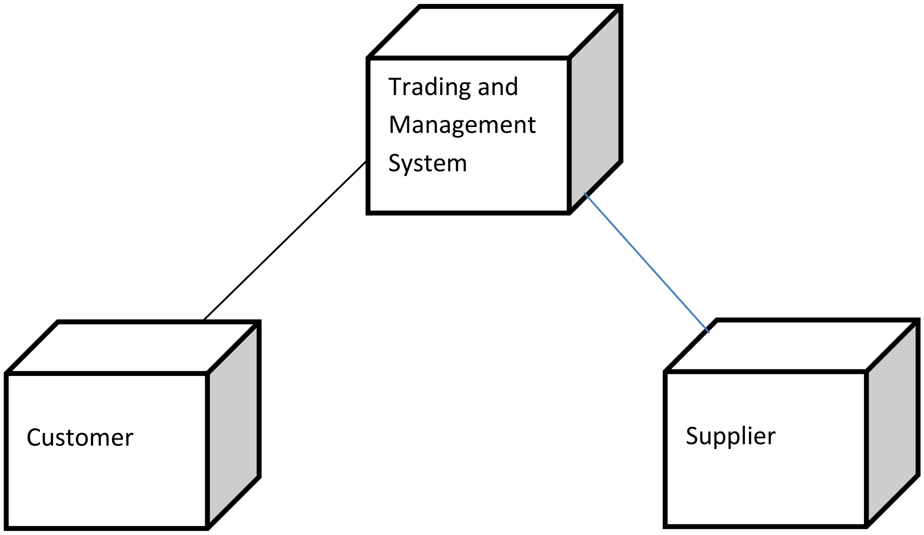
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**TRADINGMANAGEMENTSYSTEM1.java**

```
public class TRADINGMANAGEMENTSYSTEM1
{
  private int VERIFYPRODUCT;
  private int AMOUNT;

  public TRADINGMANAGEMENTSYSTEM1()
  {   }

  public void TRANSPORT()
  {   }

  public void DELIVERYPRODUCT()
  {   }

  public void MONEYTRANSFER()
  {   }
}
```

**CUSTOMER1.java**

```
public class CUSTOMER1
{
  private int ORDERPRODUCT;
  private int QUALITY;
  private int AMOUNT;
  public TRADINGMANAGEMENTSYSTEM1 theTRADINGMANAGEMENTSYSTEM1;

  public CUSTOMER1()
  {   }

  public void PAYMENT()
  {   }

  public void DELIVERY()
  {    }

  public void TRANSPORT()
  {    }
}
```

**RESULT:**

Thus the project FOREIGN TRADING SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.

# BPO MANAGEMENT SYSTEM

**EX NO : 13**

**DATE :**

**AIM:**

To prepare necessary documents and to develop the BPO MANAGEMENT SYSTEM with UML diagrams using Object Oriented Designing Methodology.

**PROBLEM ANALYSIS:**

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 system.

**SYSTEM SPECIFICATION:**

**Software requirements:**

Operating System: Windows XP
Front end: Rational Rose Enterprise suite
Back end: Microsoft Access

**Hardware requirements:**

Processor: Intel Pentium @3.06 GHZ
RAM: 512 MB DDR
Hard disk: 80 GB SATA
Monitor: 15’’ TFT
Keyboard: Multimedia Keyboard
Mouse: USB Optical type

## DESIGN :

### Use case diagram:

Use case diagram is a graph that describes the association between actors and the usecases and a generalization among the usecases.

- Actor:

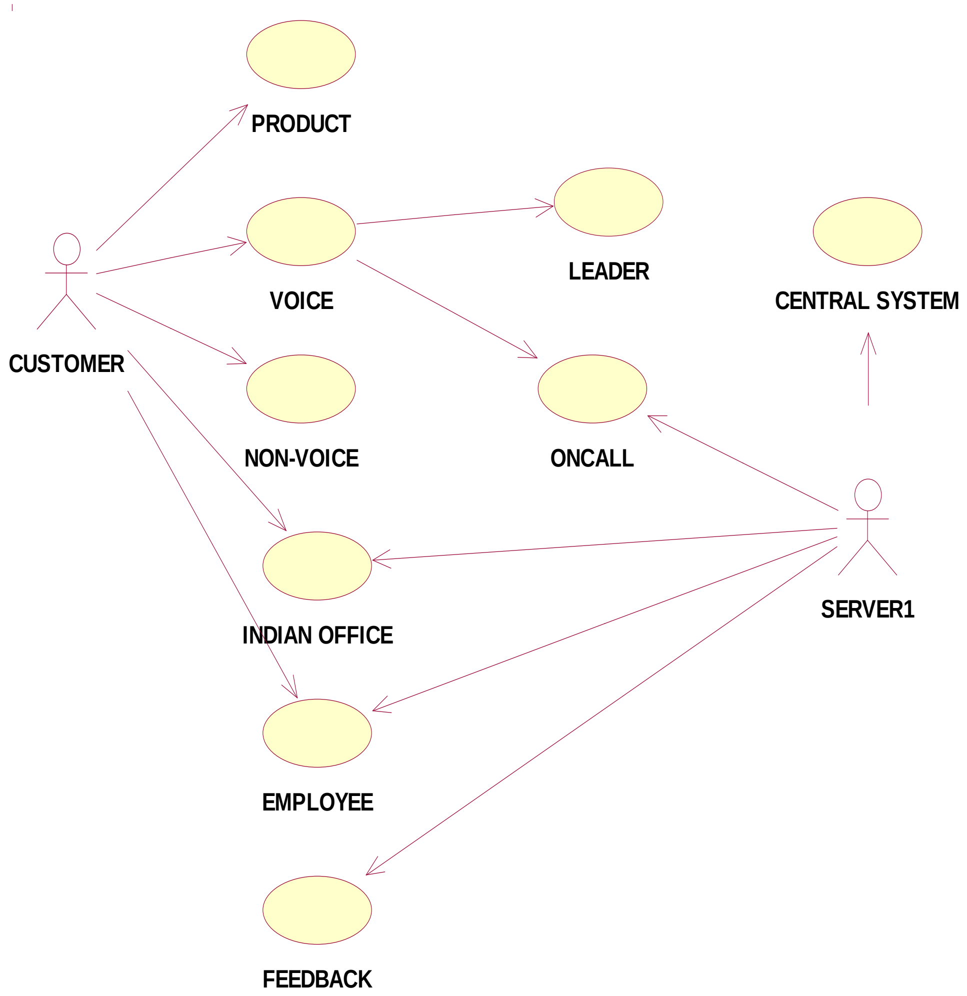
  An Actor represent a set of rules that user of a case play when interacting with the usecases.

- Usecase:

  Usecase is a description of a set of sequence of actions that a system performs of yield result of value to an actor.

**Actors:** Customer, Server, Central system

**Usecase**: Product, Voice, NonVoice,Indianoffice,Employee,Feedback.



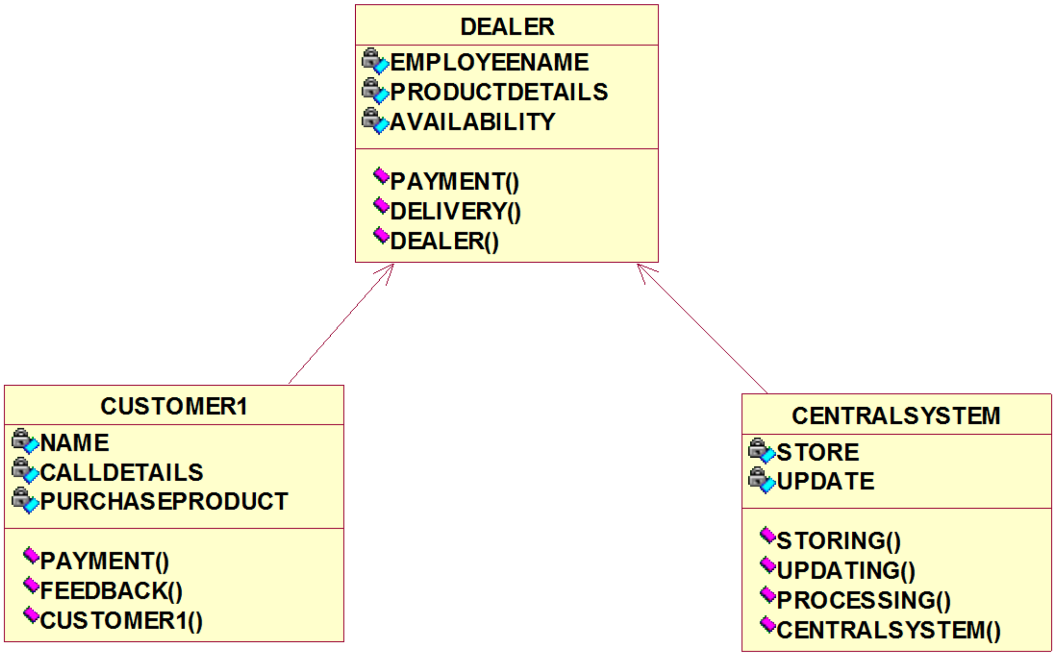
**Class diagram:**

UML class diagram is the main static analysis diagram. This diagram shows the static structure of the model. This is the collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents.

A class is drawn as a rectangle with 3 fields:

- Top field contains class name.
- Middle field contains attributes.
- Bottom field contains the list of operations.

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

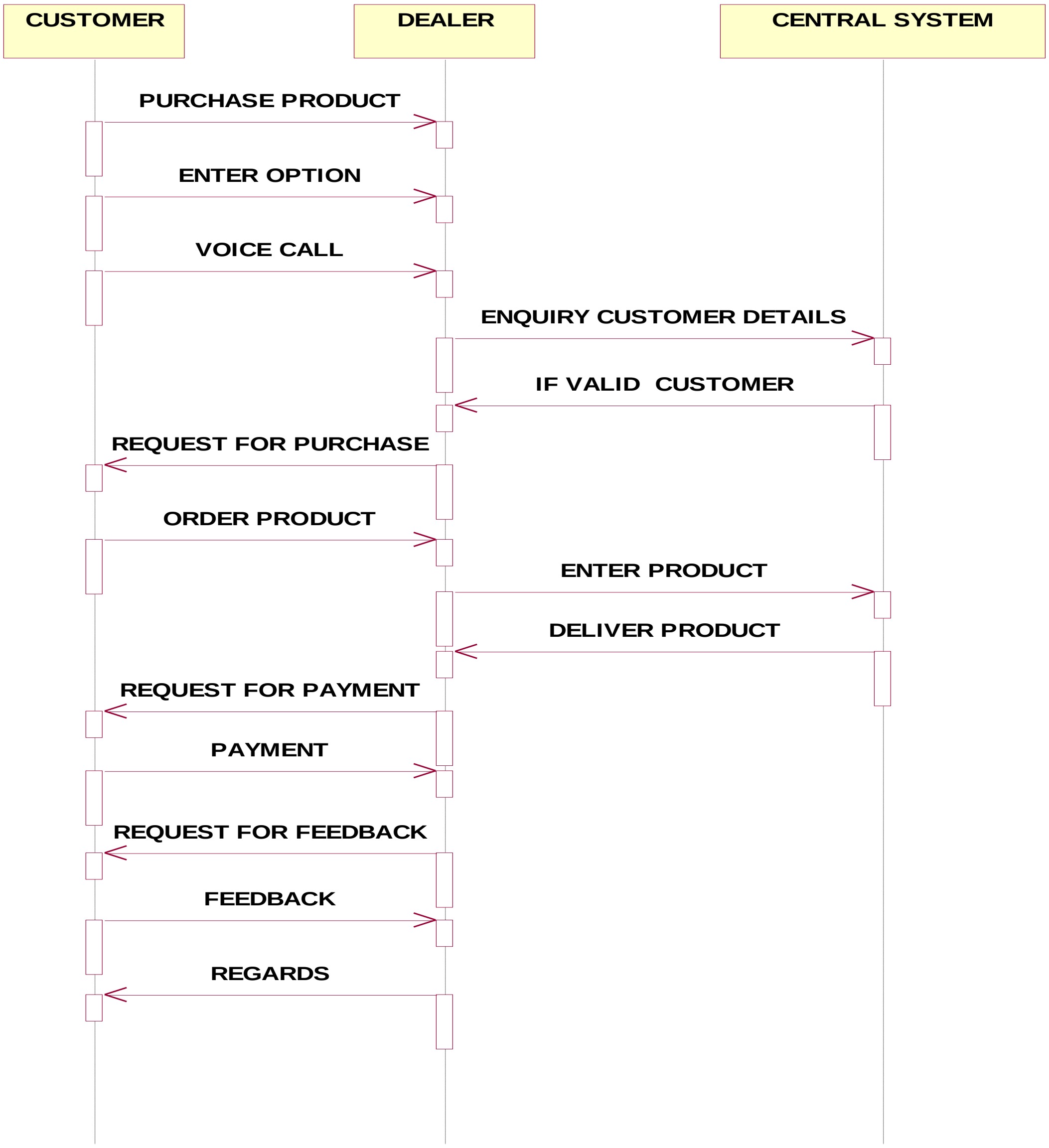


## Sequence Diagram:

Sequence diagrams are easy and intuitive way of describing the behavior of the system by viewing the interaction between the system and its environment.

It shows an interaction arranged in sequence order.

**Object :**Customer, Dealer, Central system



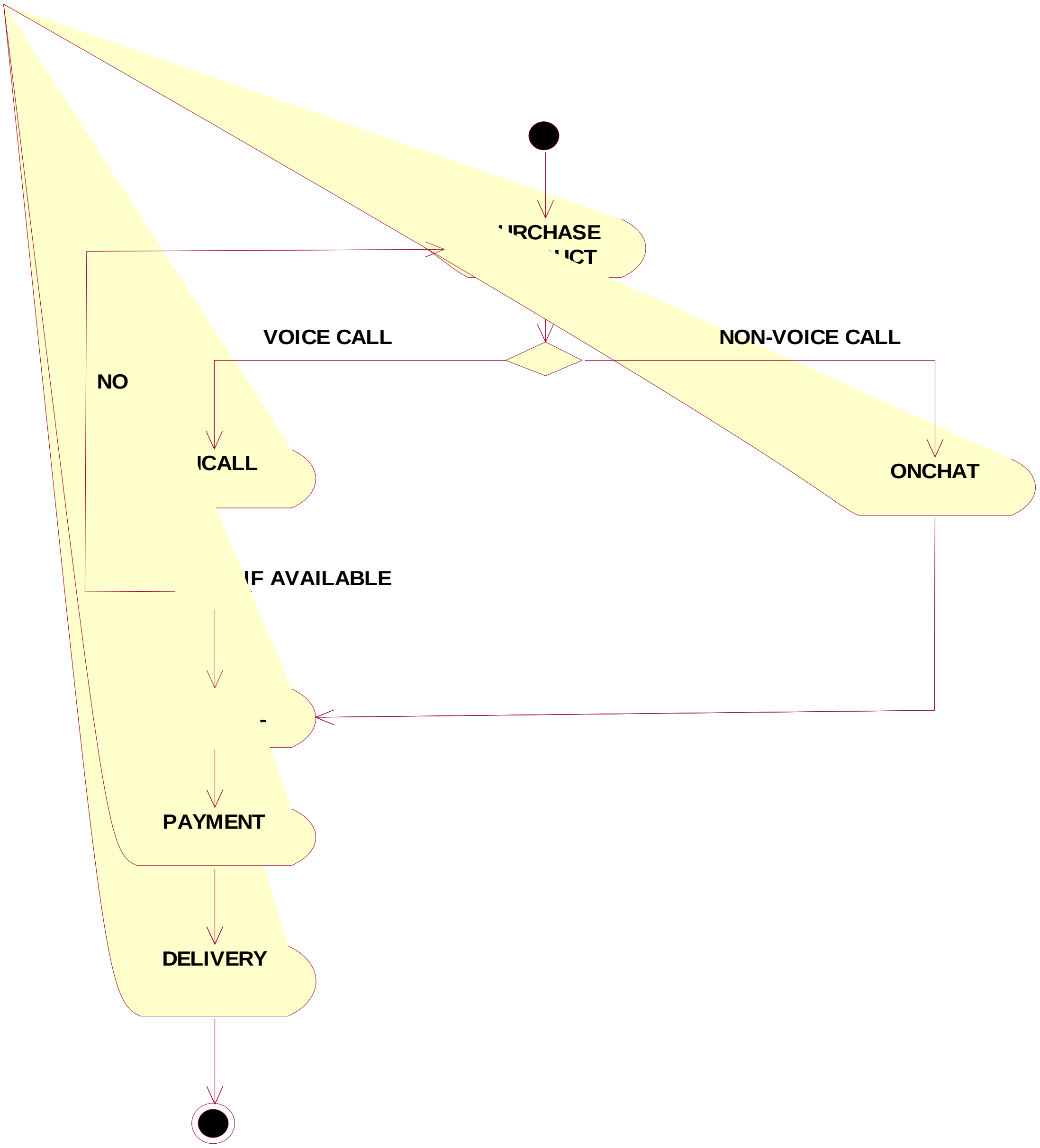
**Activity Diagram:**

The states are activities representing the performance of operations and the transactions are triggered by the completion of the operations. This diagram provides the way of flows and what is going on inside a username or among several classes.

The activity diagram is used to describe the various activities taking place in an application.

**Activities:** Purchase product, oncall, onchat

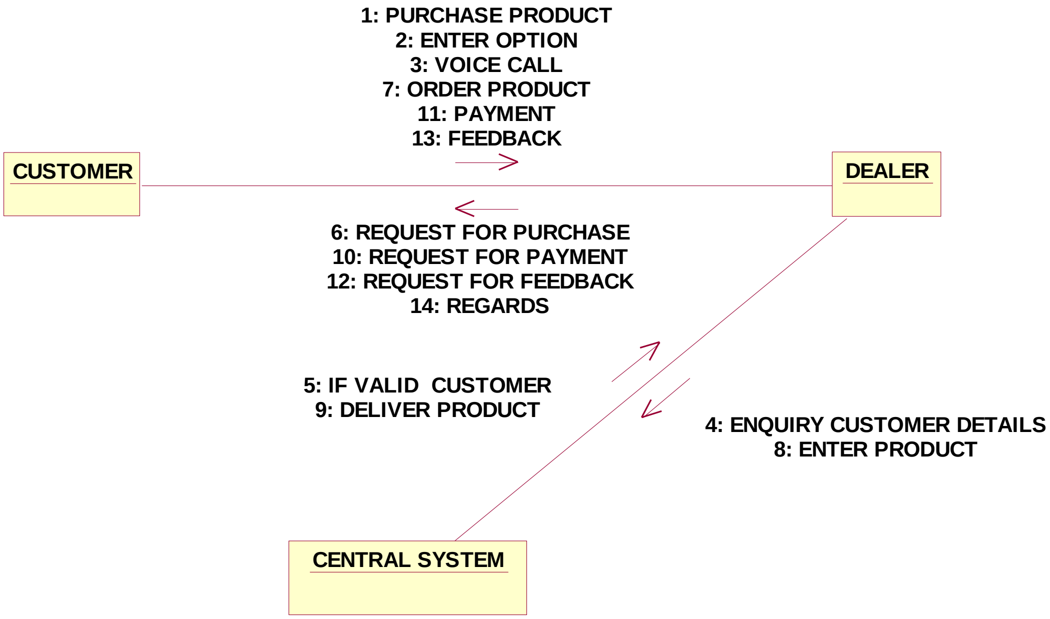
**Decision box**: Option to check



## Collaboration Diagram:

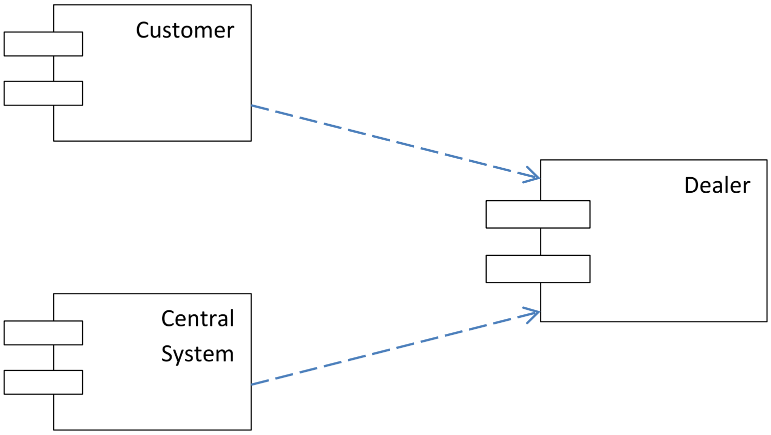
A collaboration diagram represents a collaboration, which is a set of objects related in a particular context and interaction, which is a set of messages exchanged among the object within the collaboration to achieve a desired outcome.

It describes interacting or collaboration with each other objects, it gets represented by solid line drawn between them.



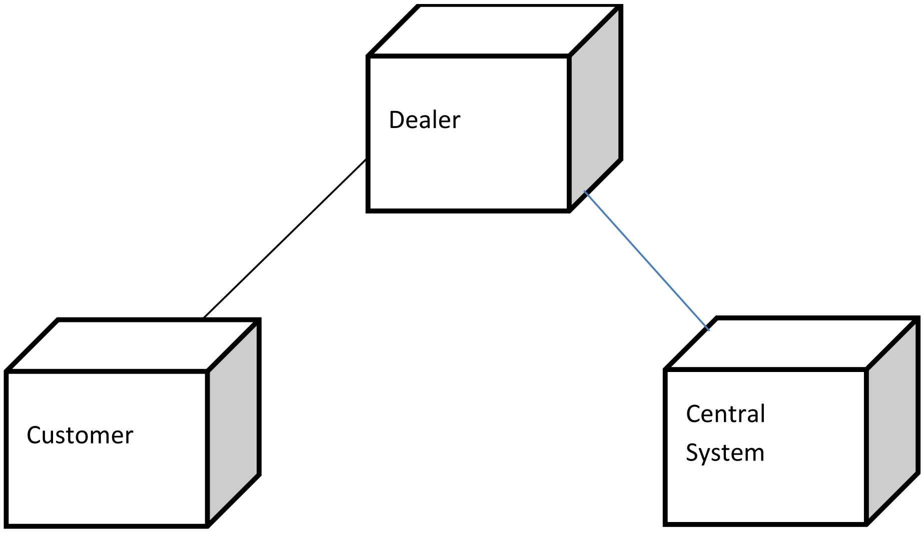
## Component Diagram:

A component is a code module. Component diagrams are physical analogs of class diagram.



**Deployment Diagram:**

Deployment diagram shows the physical configurations of software and hardware.



**JAVA CODING :**

**CENTRALSYSTEM.java**

```
public class CENTRALSYSTEM
{
  private int STORE;
  private int UPDATE;
  public DEALER theDEALER;
  public CENTRALSYSTEM()
  {   }
  public void STORING()
  {   }
  public void UPDATING()
  {   }
  public void PROCESSING()
  {

  }
}
```

**RESULT:**

Thus the project BPO MANAGEMENT SYSTEM was developed with all necessary documents and UML diagrams using Object Oriented designing methodology.