

```
Public integer resume;
```

```
Public void HR()
```

```
{
```

```
}
```

OOAD LAB

REGISTER NO:

```
}
```

**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.

<b>EX:NO:11</b>	<b>CONFERENCE MANAGEMENT SYSTEM</b>
<b>DATE:</b>	

### **AIM:**

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

### **SOFTWARE REQUIREMENTS SPECIFICATION**

	<b>SOFTWARE REQUIREMENTS SPECIFICATION</b>
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:**

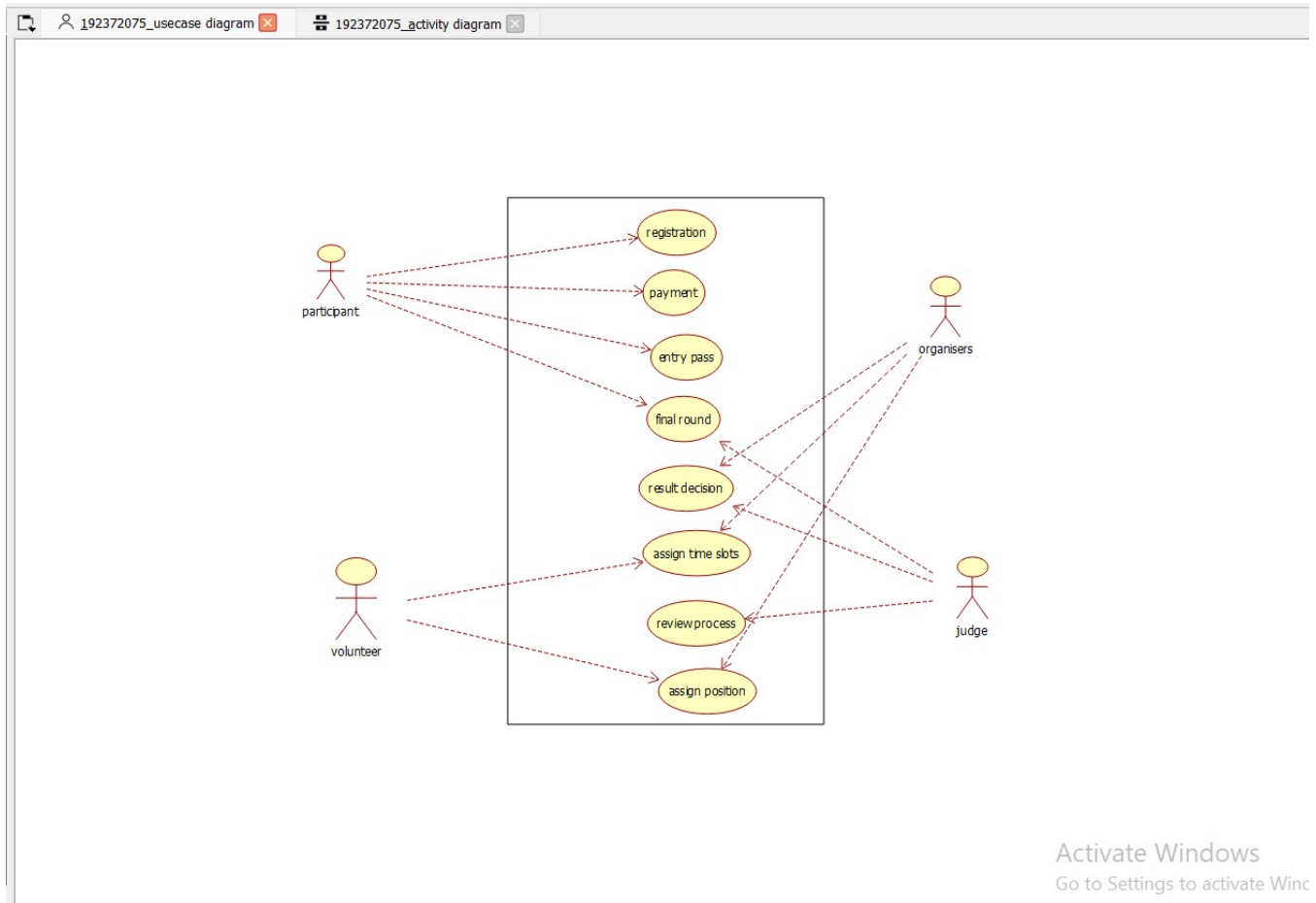
IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

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

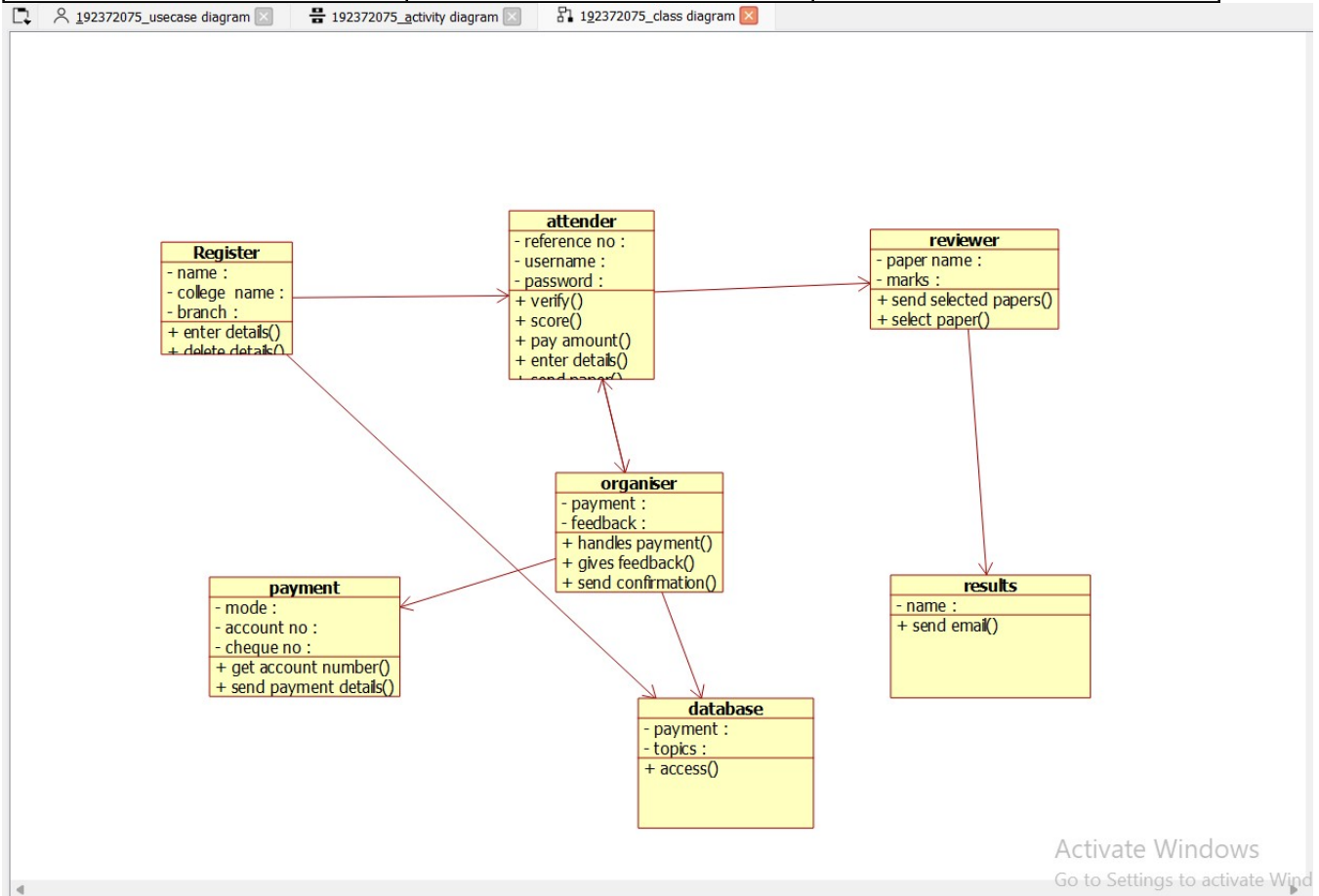
**Use case:** planning, invite delegates, allocate seats, presenting paper, prize distribution



## 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()

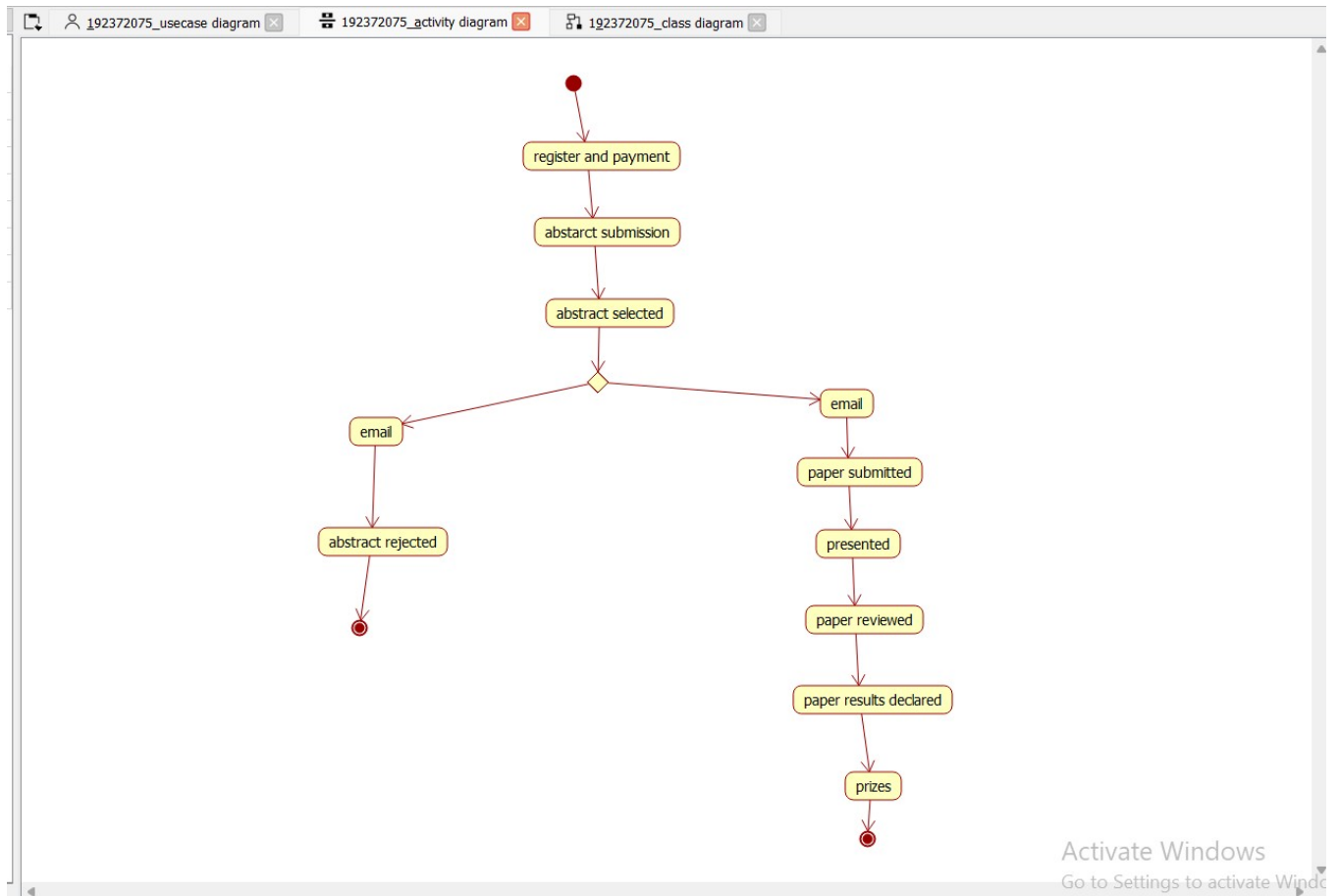


**ACTIVITY DIAGRAM:**

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

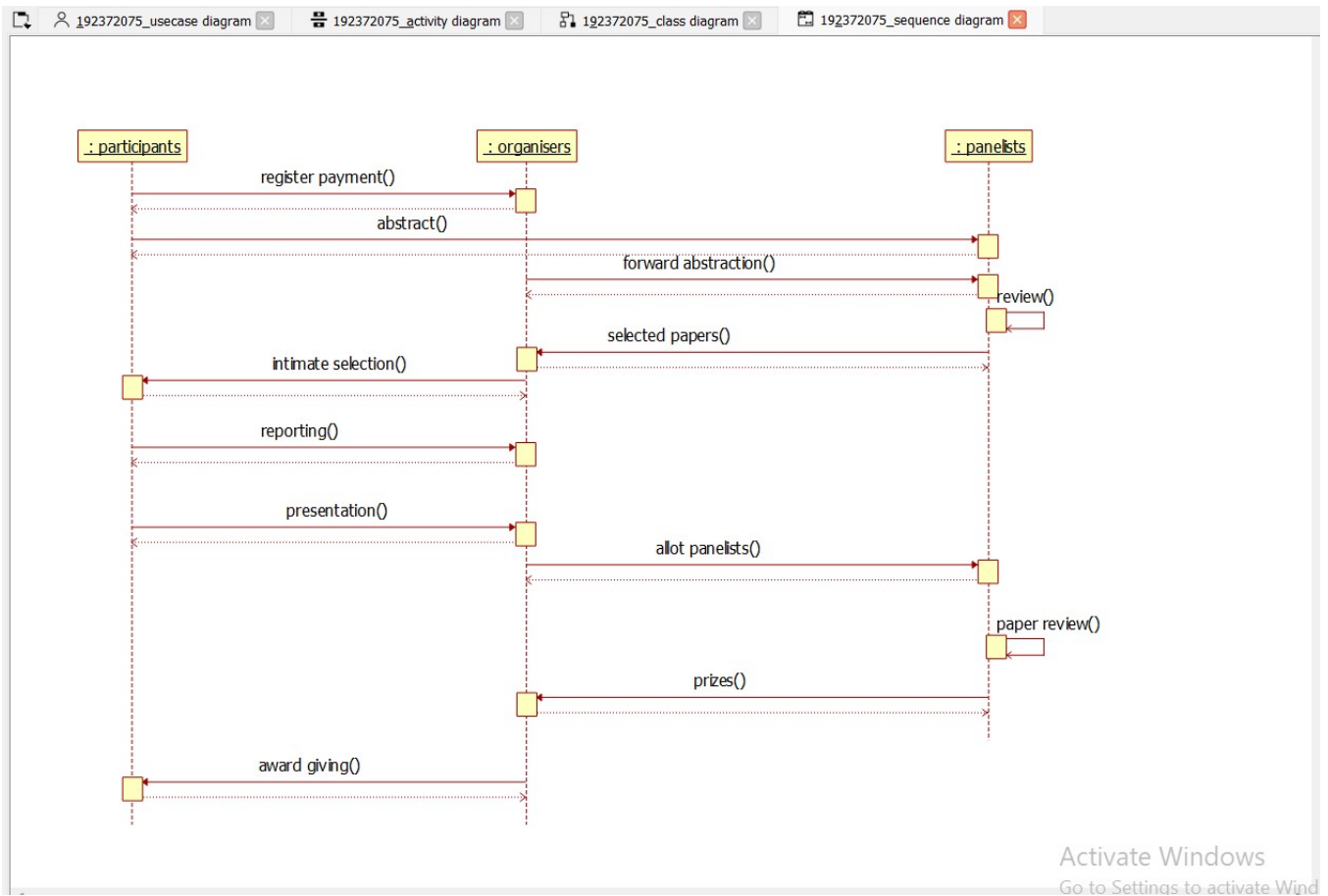
**Decision box:** Whether it is reserved or not, Whether the presentation is good or not



### **SEQUENCE DIAGRAM:**

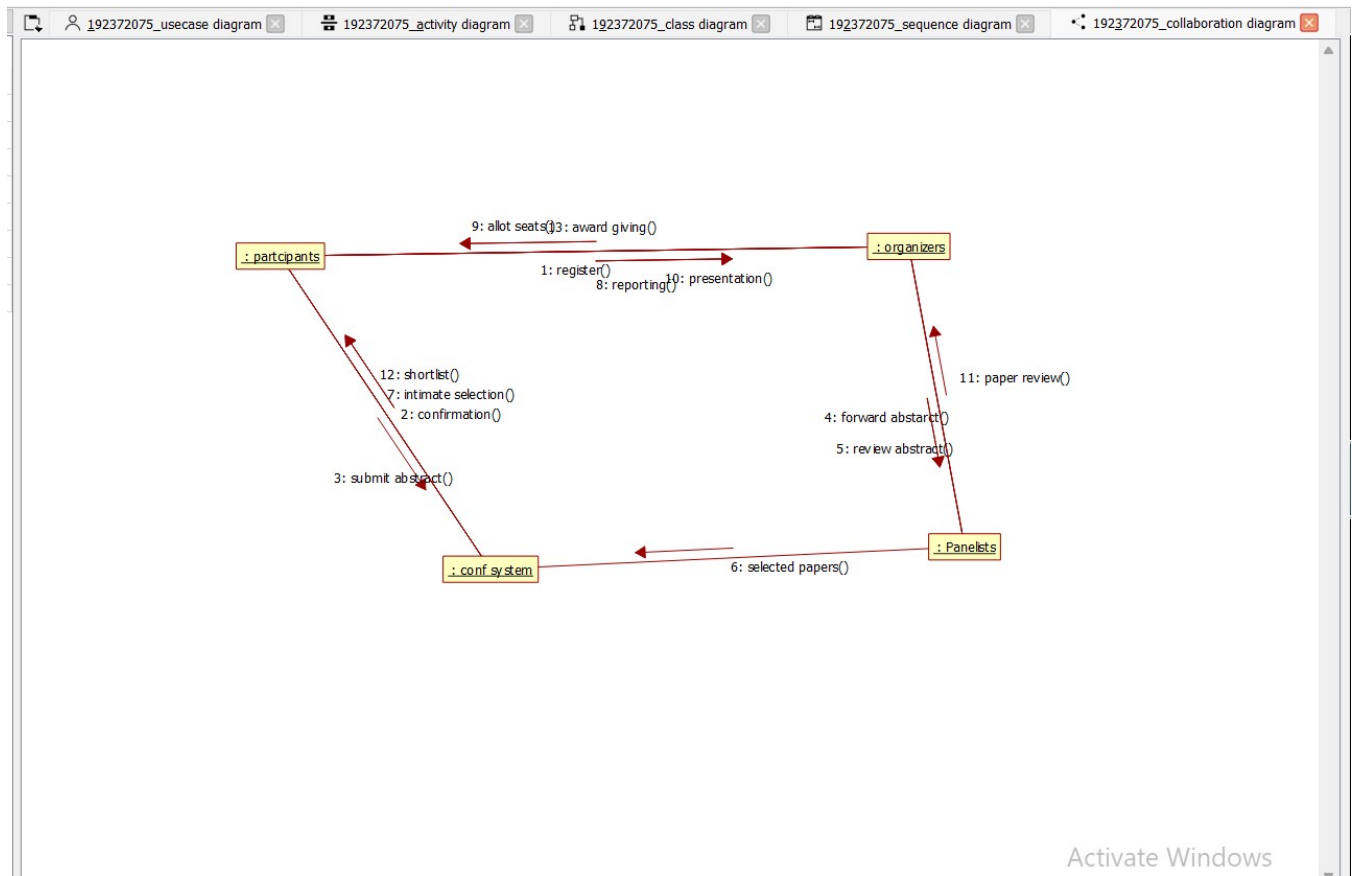
This diagram consists of the objects, messages and return messages.

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



### **COLLABORATION DIAGRAM:**

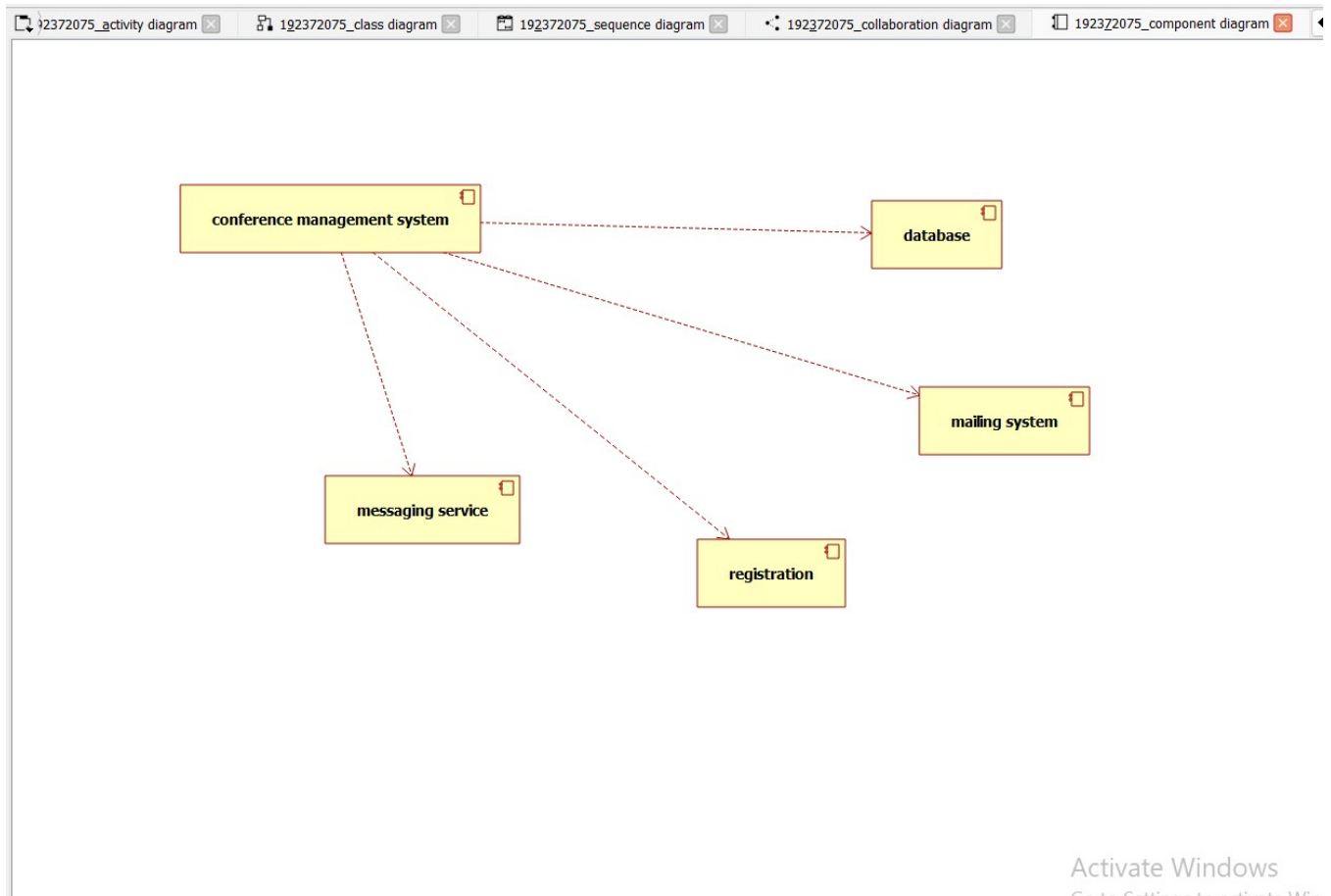
This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



### **COMPONENT DIAGRAM:**

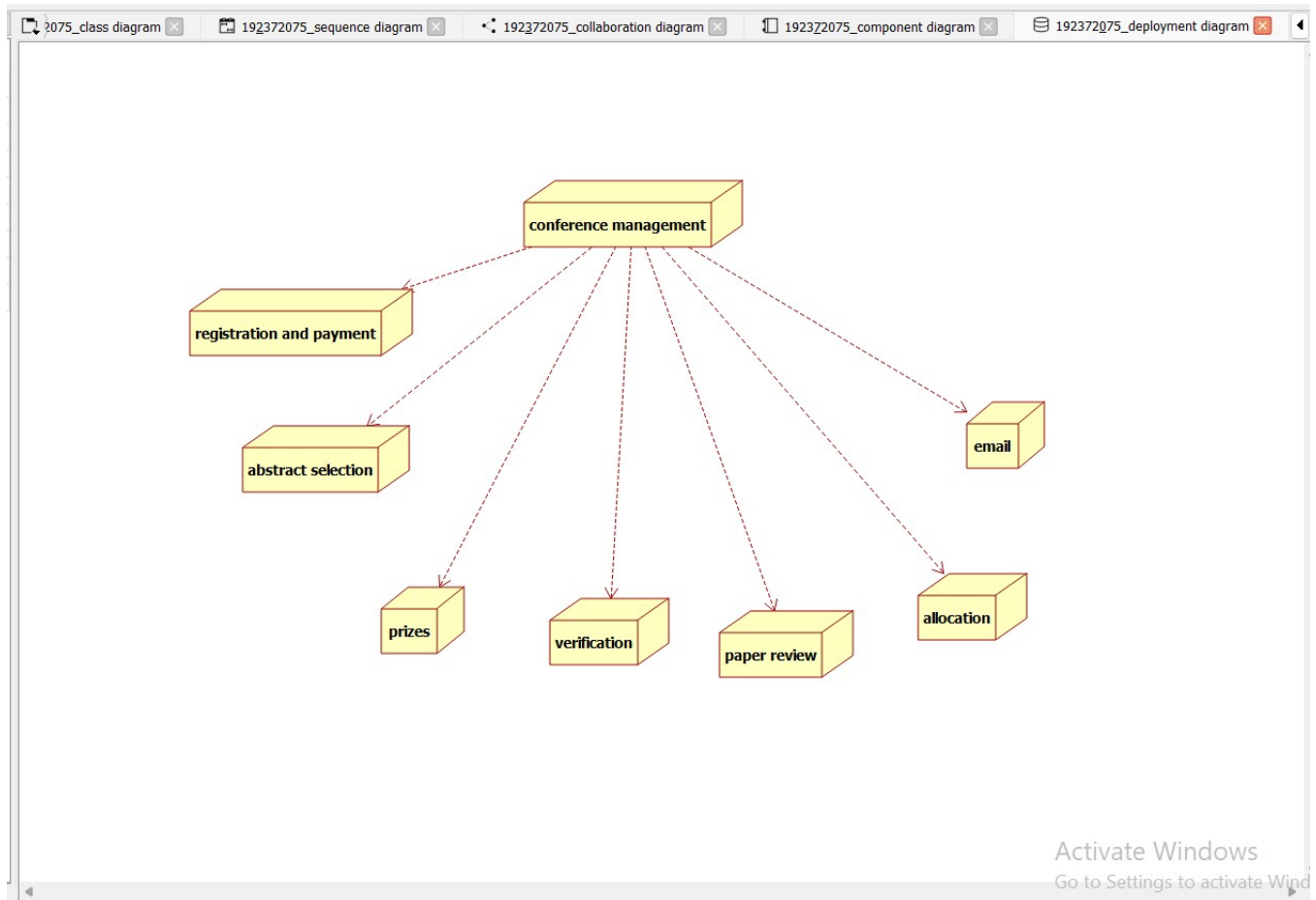
The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication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





### **DEPLOYMENT DIAGRAM:**

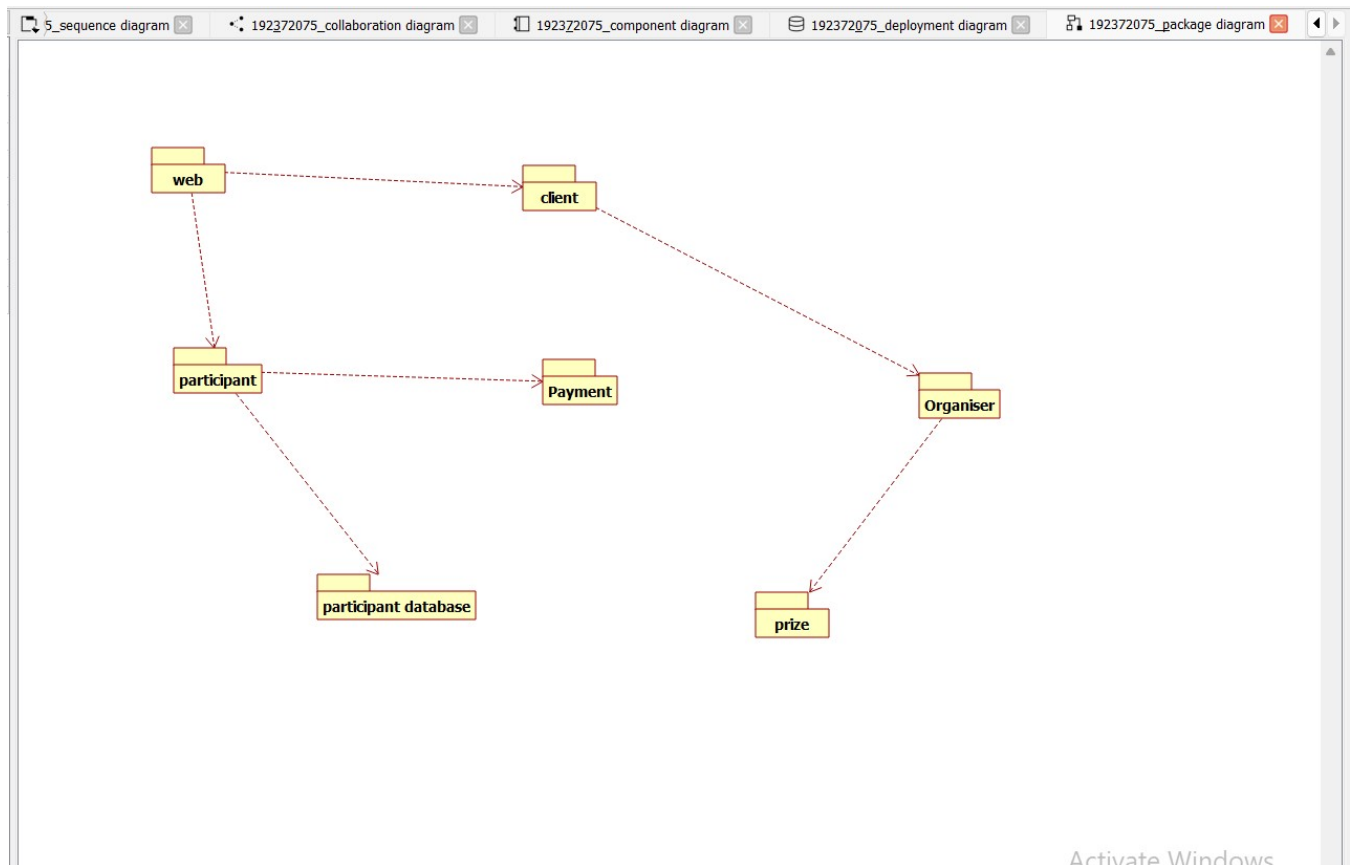
A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimensional box. Dependencies are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



OOAD LAB

REGISTER NO:

### **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 {

OOAD LAB

REGISTER NO:

```

    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.