

```
}
```

**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()
    {
    }
}
```

OOAD LAB

REGISTER NO:

**RESULT:**

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

<b>EX:NO:10</b>	<b>RECRUITMENT SYSTEM</b>
<b>DATE:</b>	

**AIM:**

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

**1.3 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:**

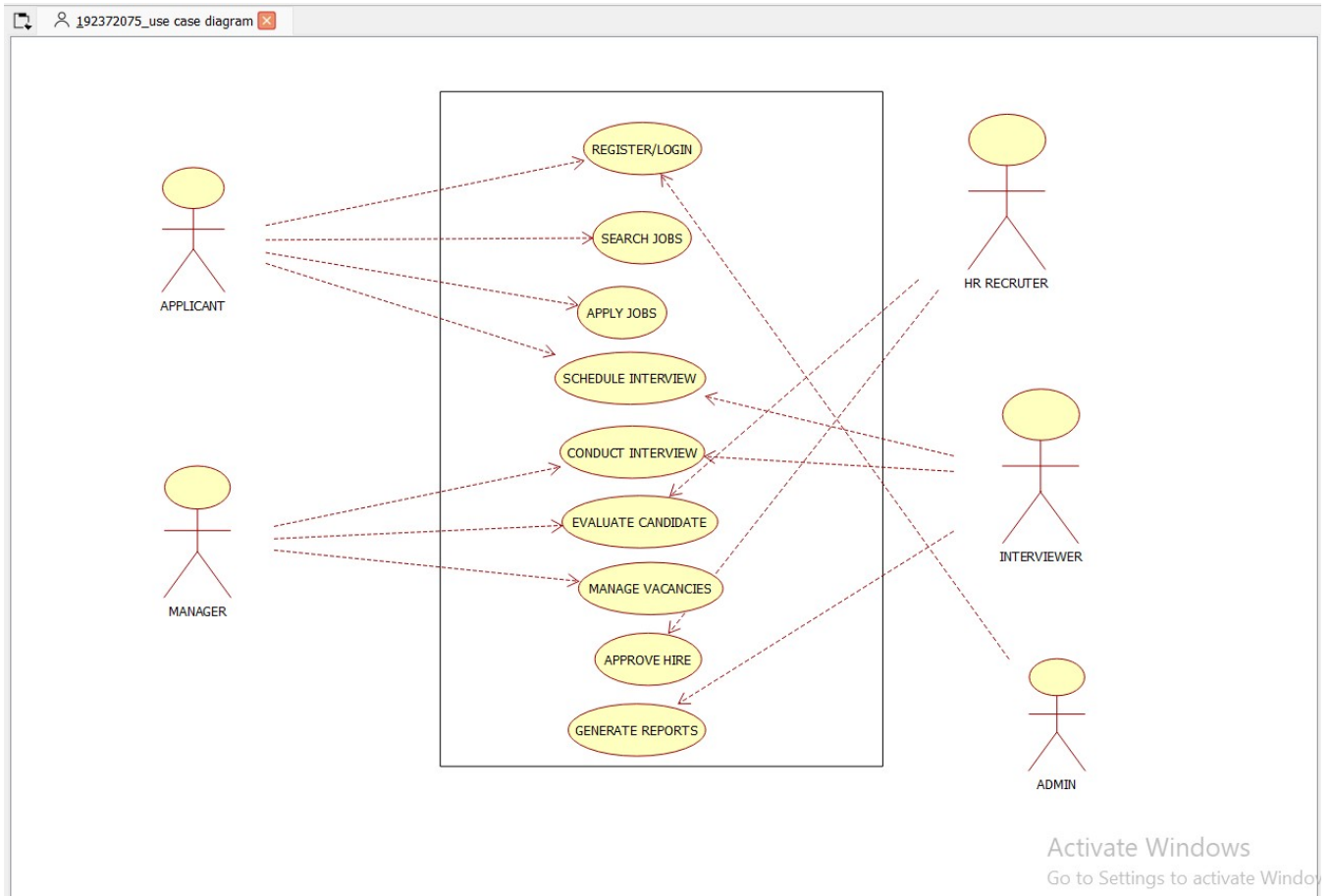
IEEE Software Requirement Specification format.

## USE CASE DIAGRAM:

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

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

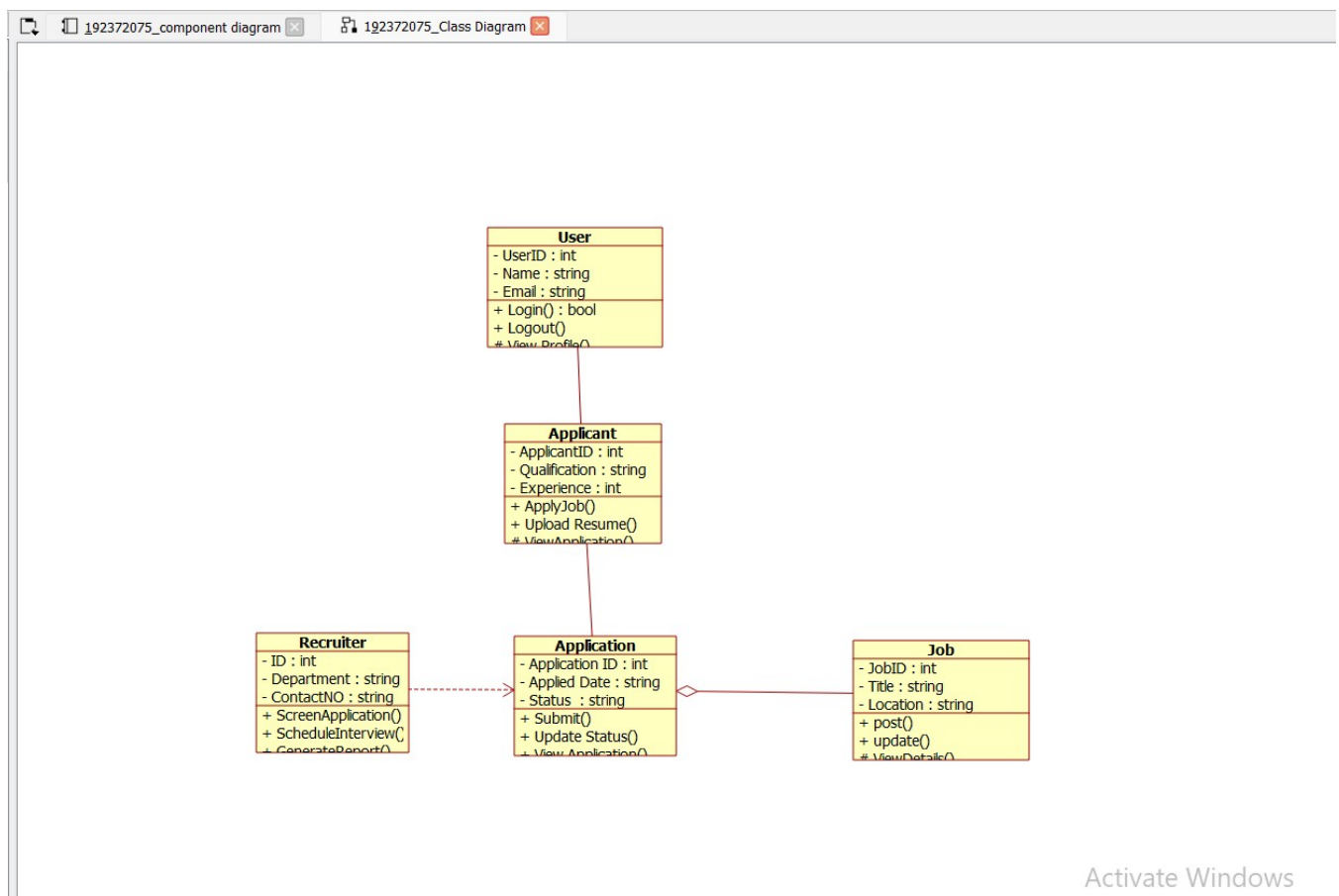
**Use case:** Aptitude, Group discussion, Technical skills, Personal specification, Short list, Result



### 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()

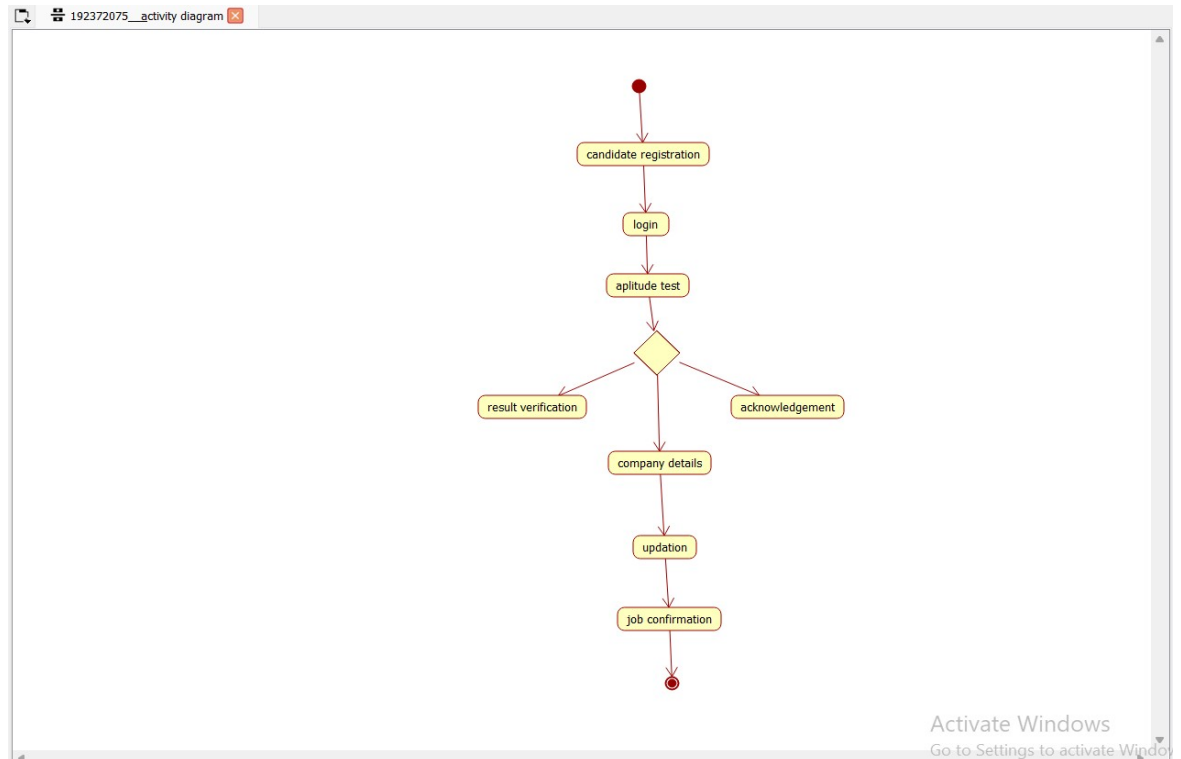


## ACTIVITY DIAGRAM:

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

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

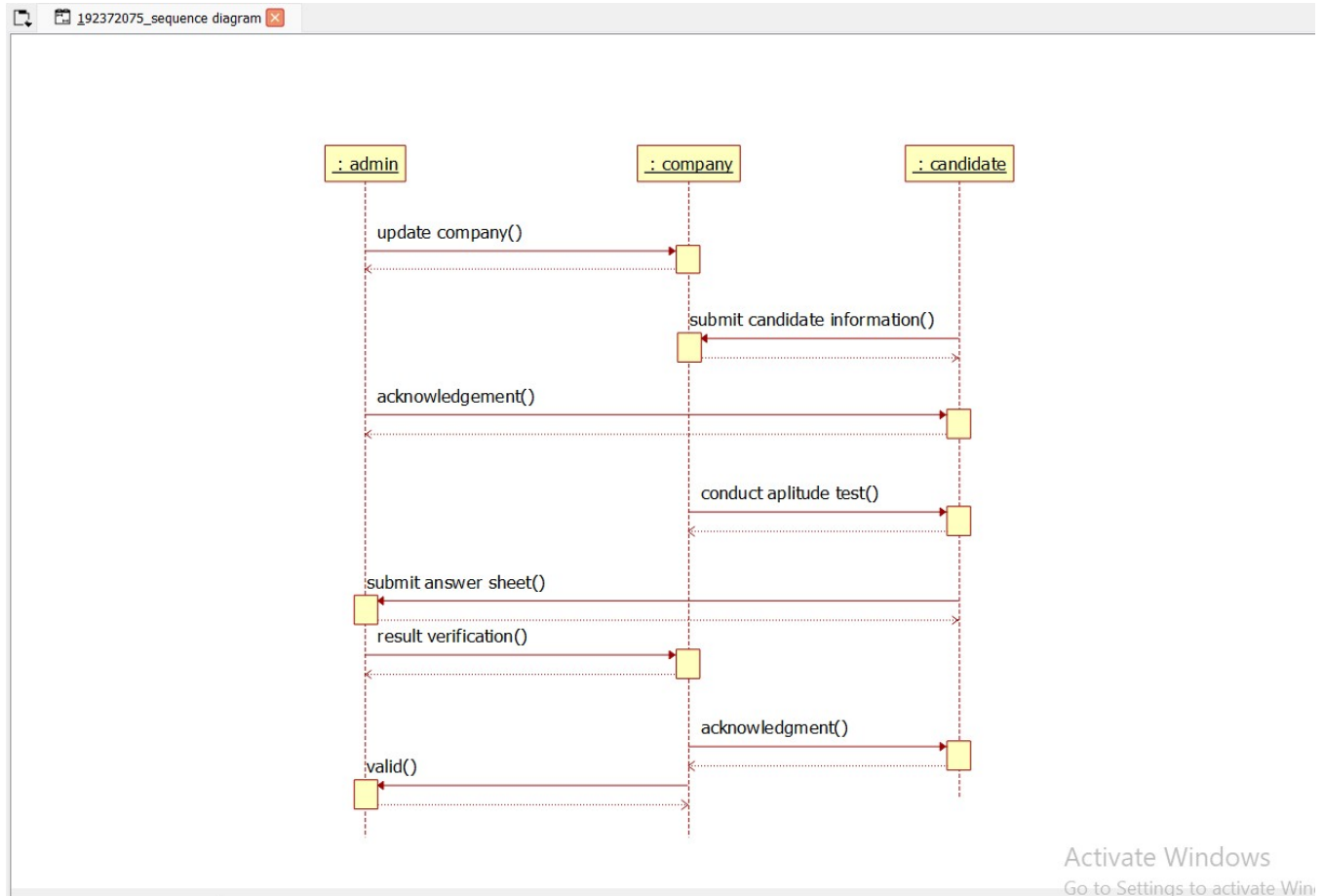
**Decision box:** Verification of the qualities



## SEQUENCE DIAGRAM:

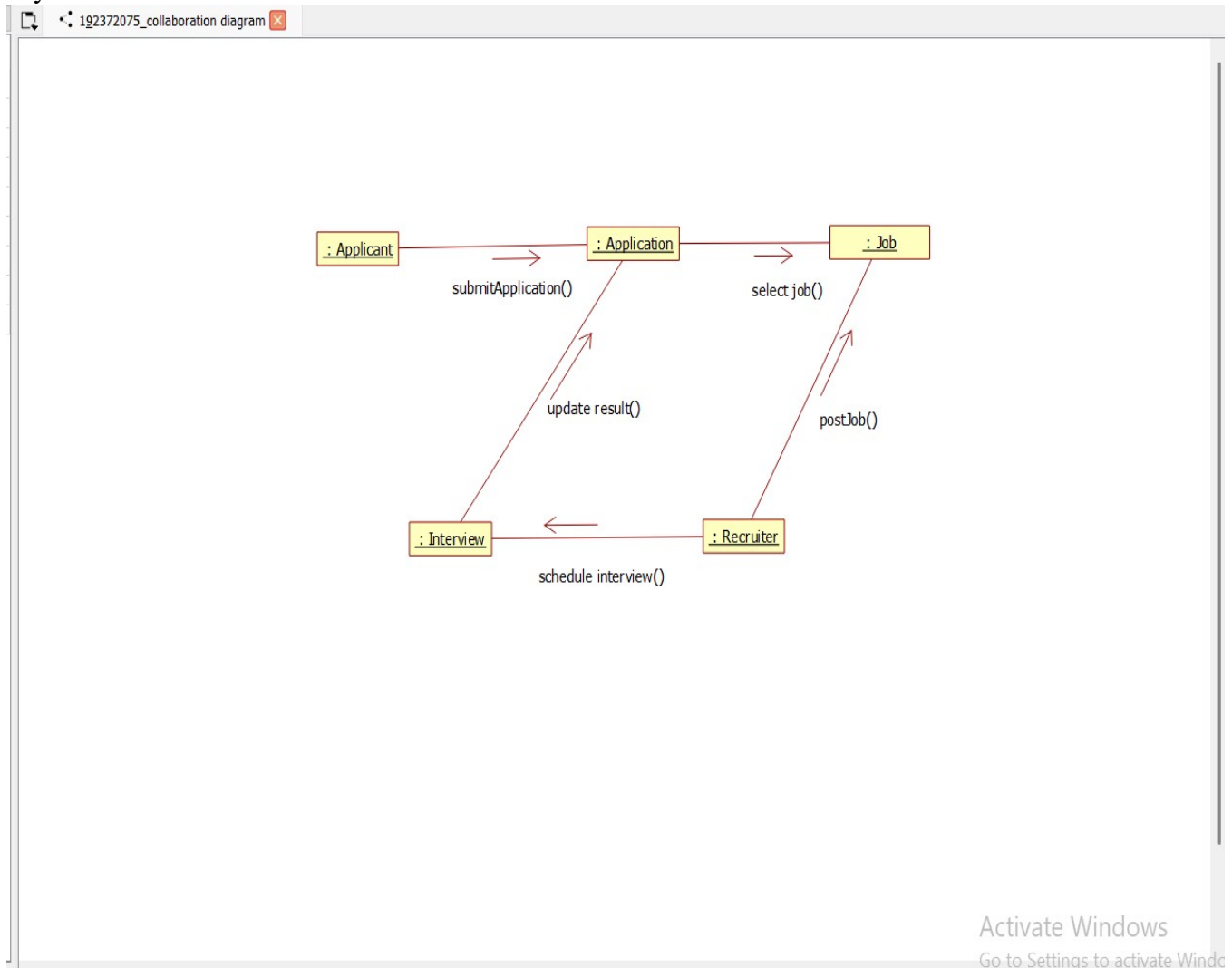
This diagram consists of the objects, messages and return messages.

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



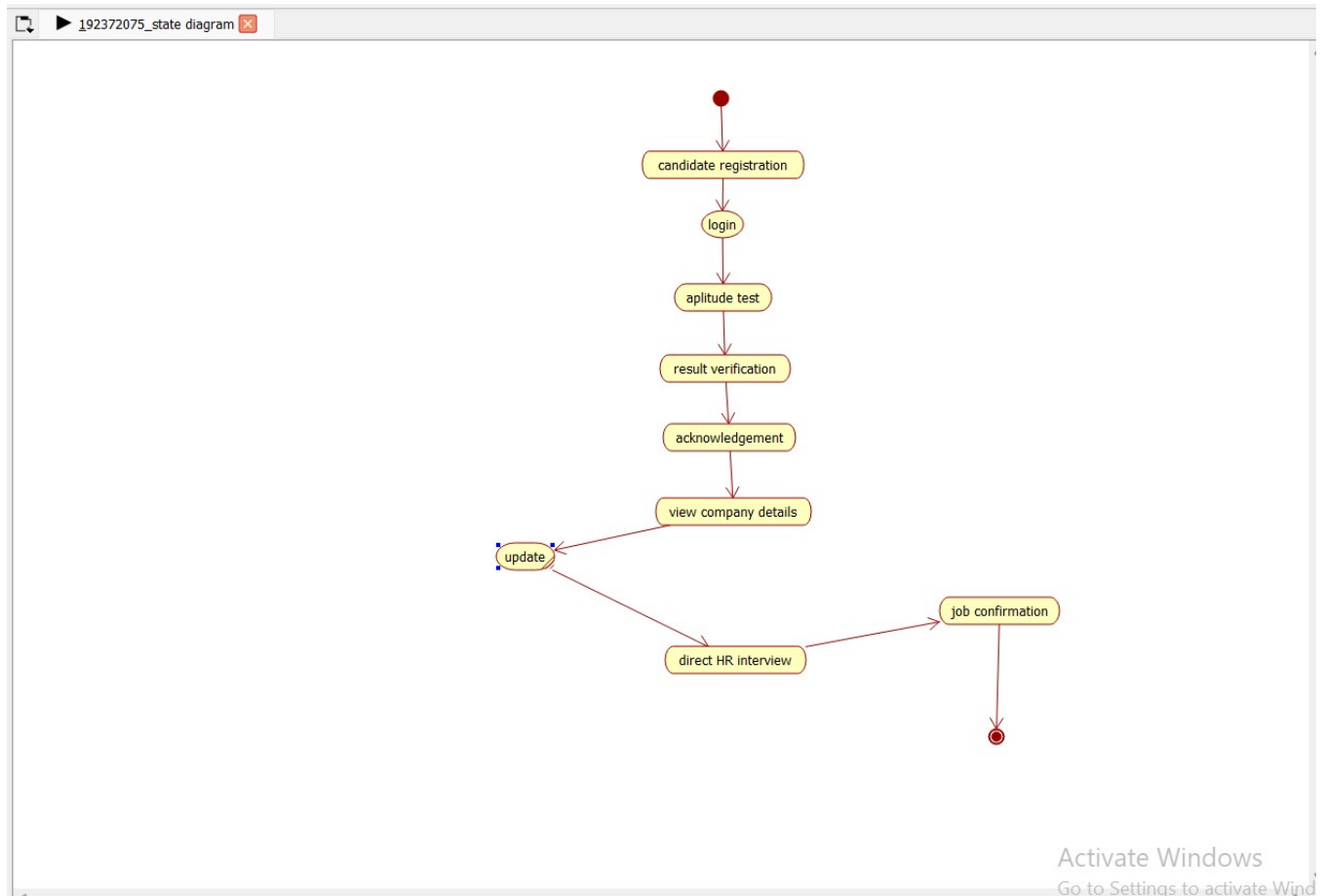
## COLLABORATION DIAGRAM:

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:

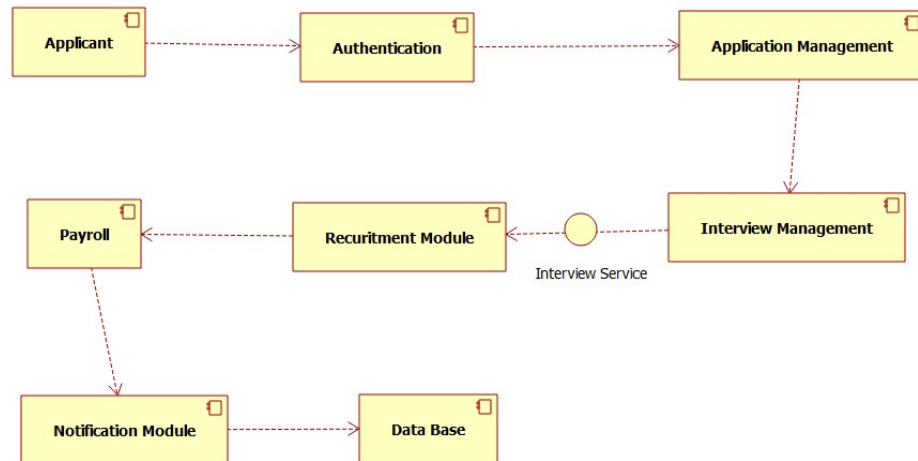
It is a technique to describe the behavior of the system. It describes all the possible states that a particular object gets into the object oriented technique. State diagram are drawn for a single class to show to the lifetime behaviour of a single objects



### **COMPONENT DIAGRAM:**

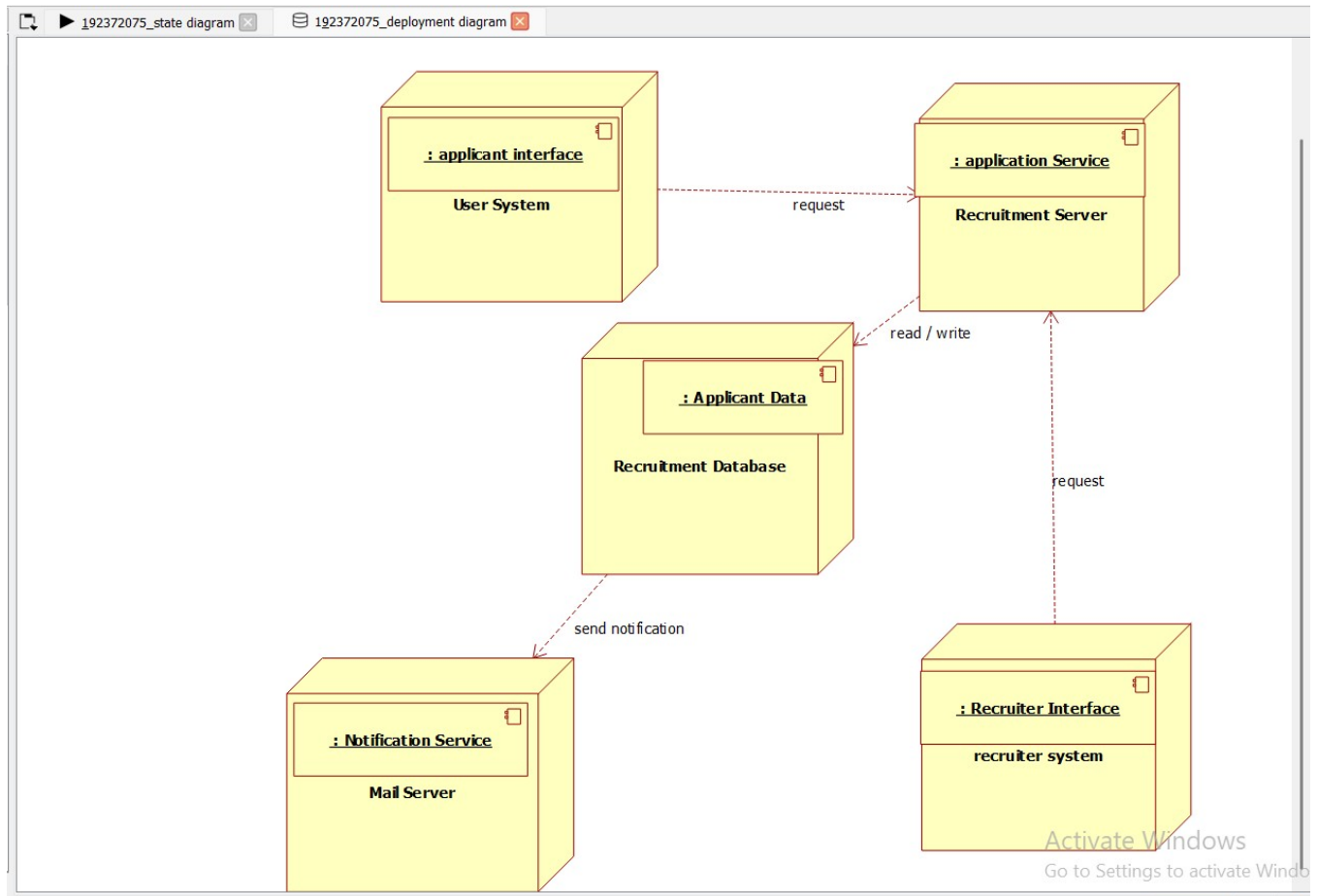
The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication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



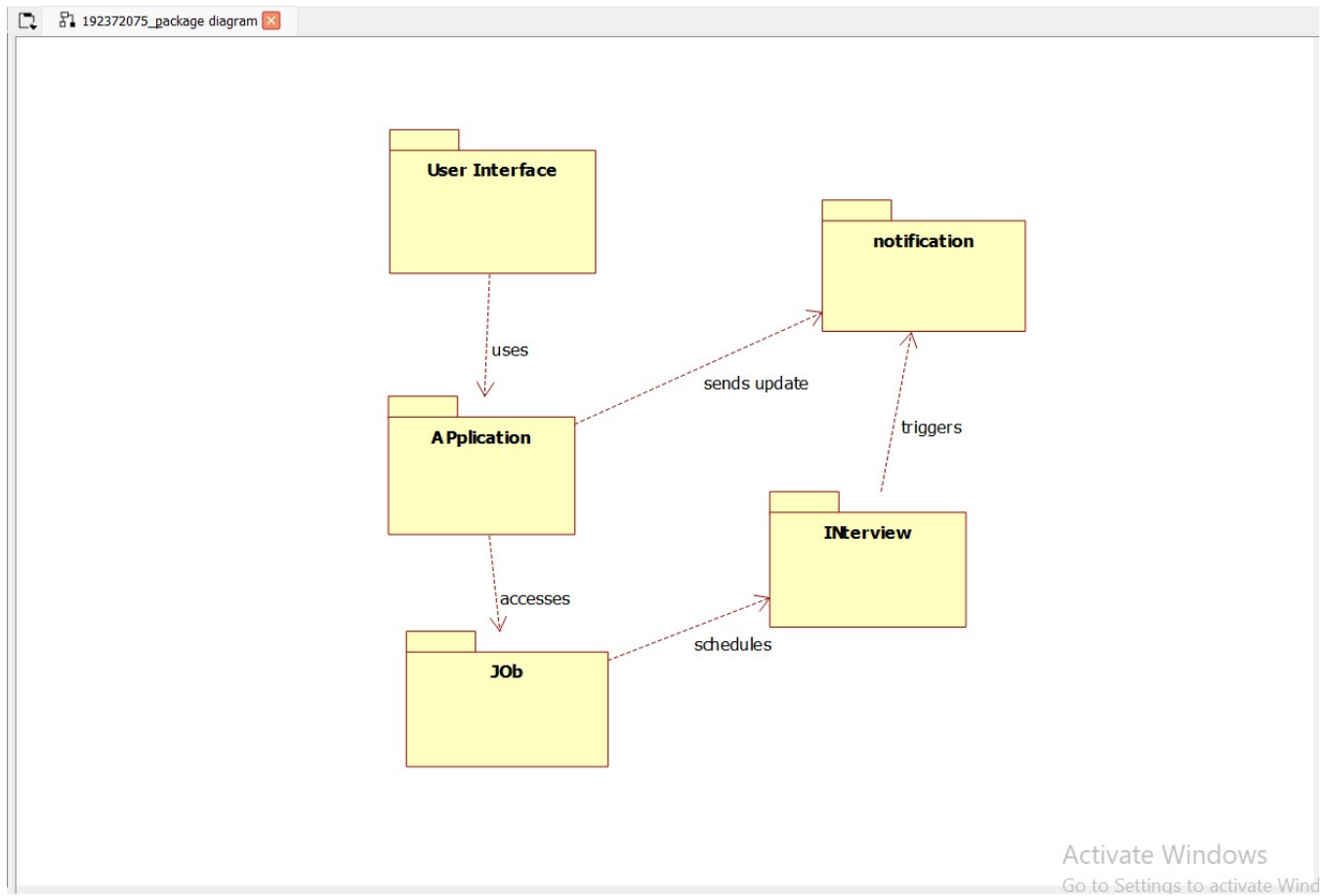
artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimensional box. Dependencies are represented by communication association

## 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



### **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;
```