

<b>EX NO: 3</b>	<b>EXAM REGISTRATION SYSTEM</b>
<b>DATE:</b>	

**AIM:**

To draw the diagrams [use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Exam registration system.

**SOFTWARE REQUIREMENTS SPECIFICATION**

<b>SL.NO</b>	<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 Exam Registration is an application in which applicant can register themselves for the exam. The details of the students who have registered for the examination will be stored in a database and will be maintained. The registered details can then be verified for any fraudulent or duplication and can be removed if found so. The database which is verified can be used to issue hall tickets and other necessary materials to the eligible students.

### **1.3 PROJECT DESCRIPTION:**

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.

### **1.4 REFERENCES:**

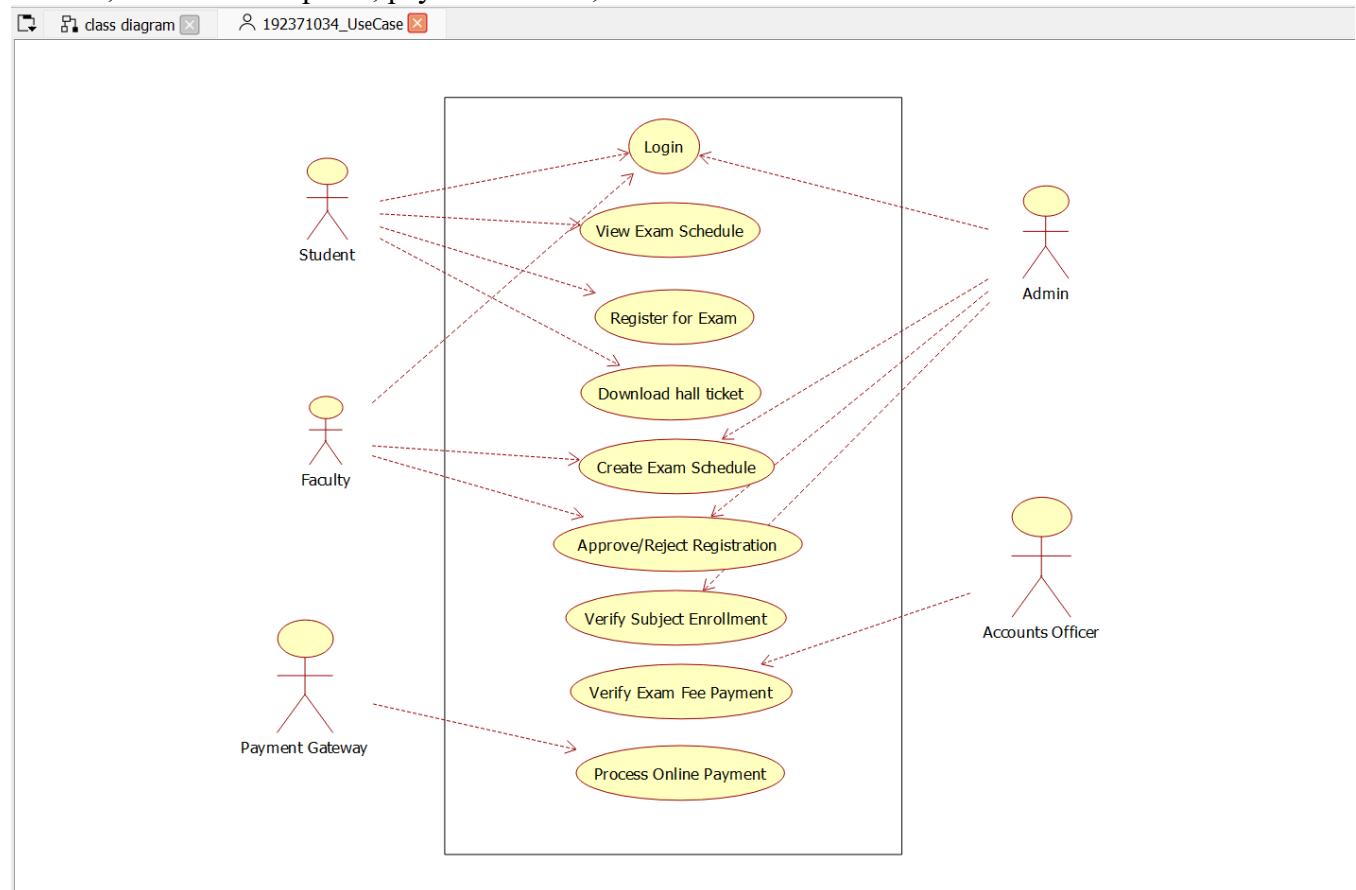
IEEE Software Requirement Specification format.

### **USE CASE DIAGRAM:**

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

**Actors:** Student, educational officer..

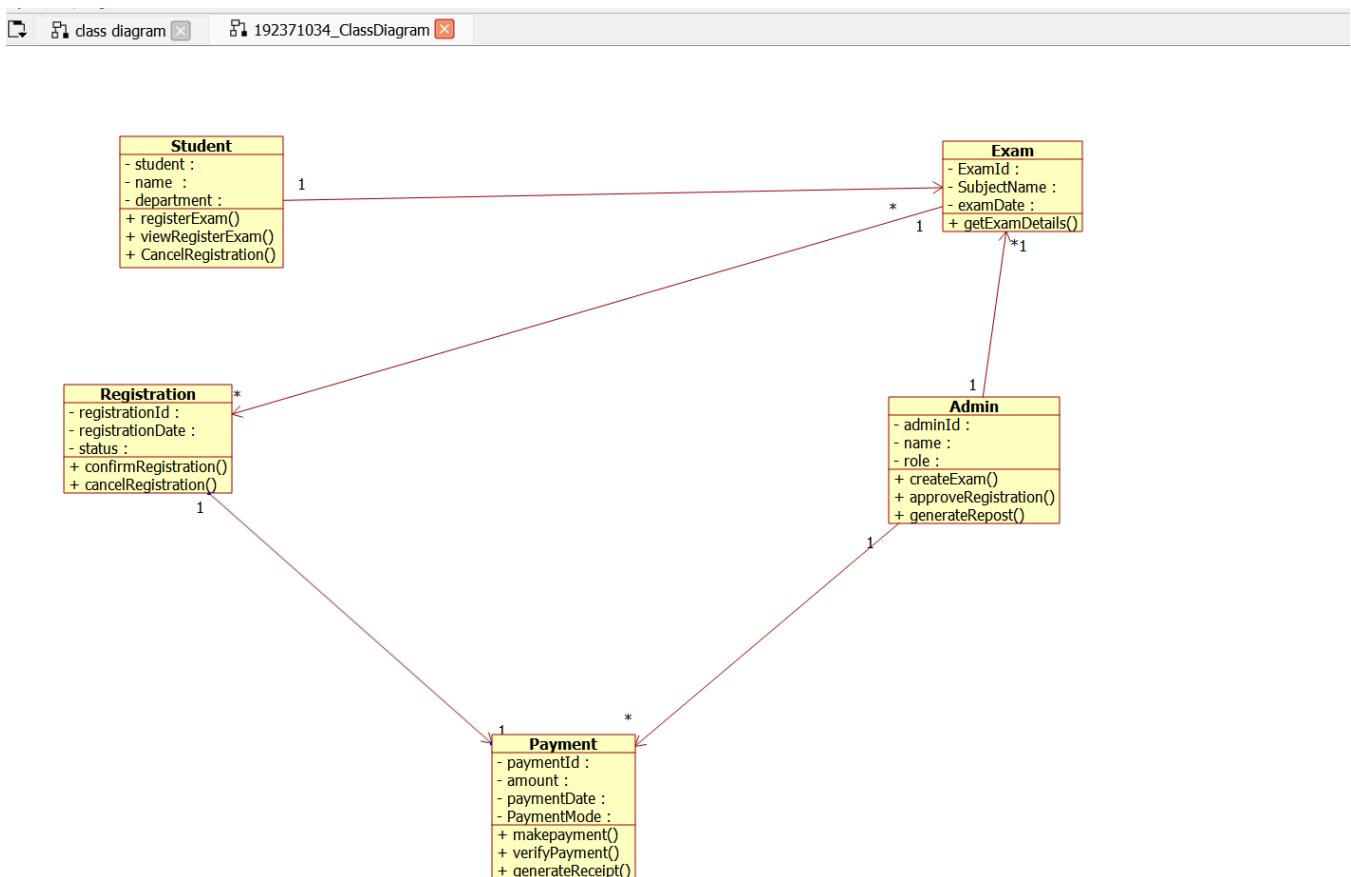
**Use case:** Student details, student photo, student proof submission of proof ,verification of proof, payment of fees, issue of hall ticket



### **CLASS DIAGRAM:**

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

CLASSES	ATTRIBUTES	OPERATIONS
Central educational system	Student details	Print hall ticket(), Issue hall ticket()
Stud	Submit details, Submit photo	Payment of fees()
Edu officer	Enter details	Issue hall ticket(), Verify proof()

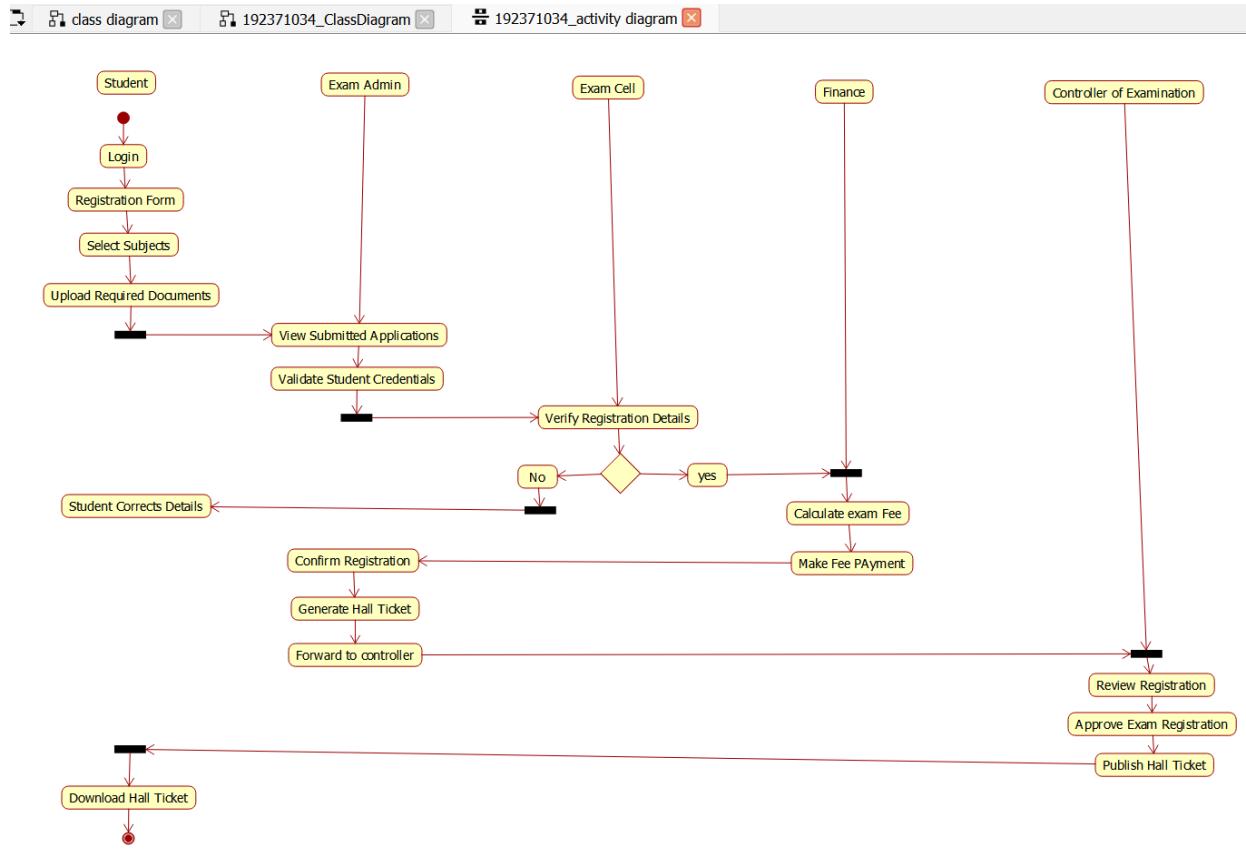


### ACTIVITY DIAGRAM:

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

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

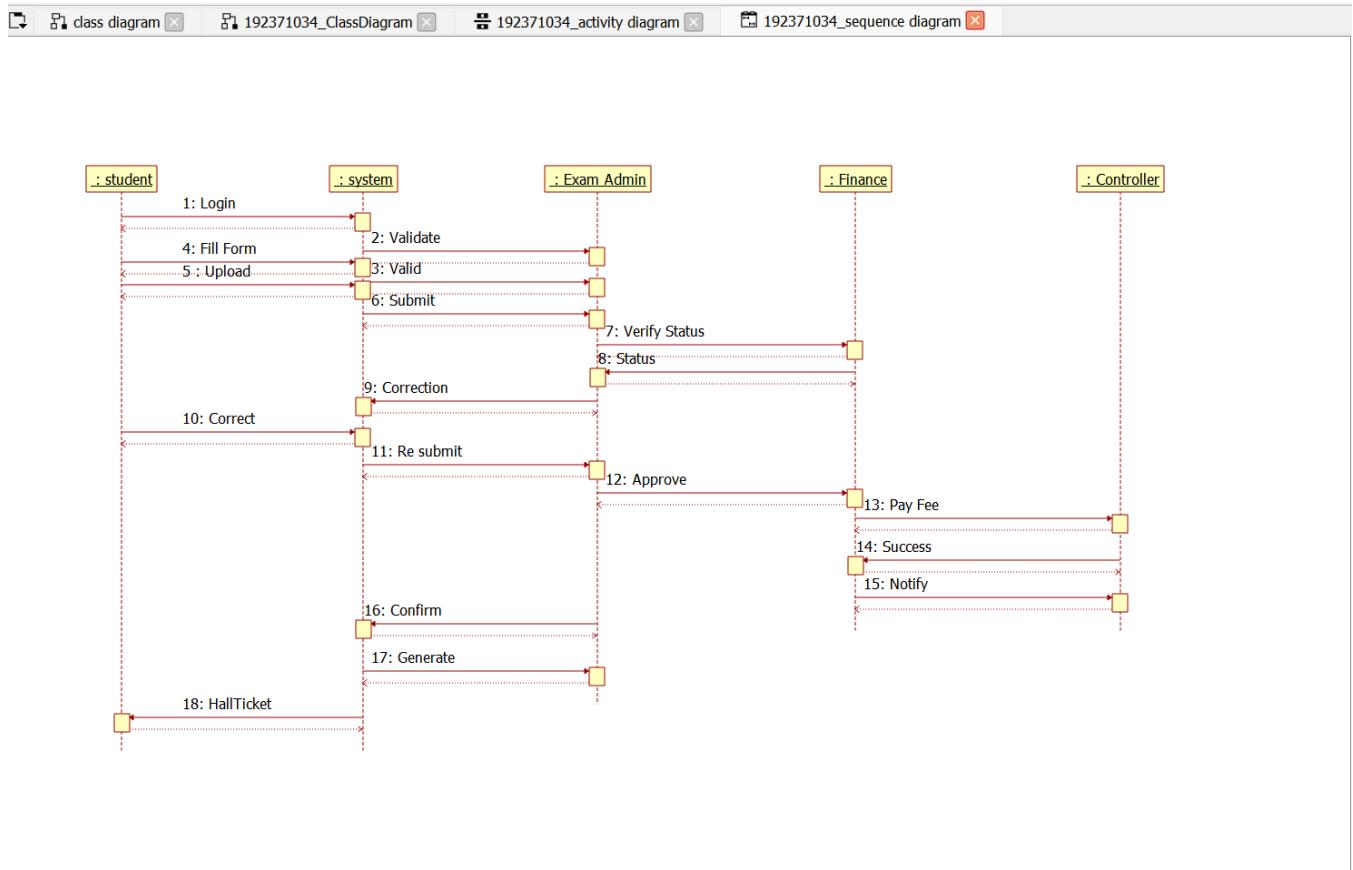
**Decision box:** Verification of proof.



## SEQUENCE DIAGRAM:

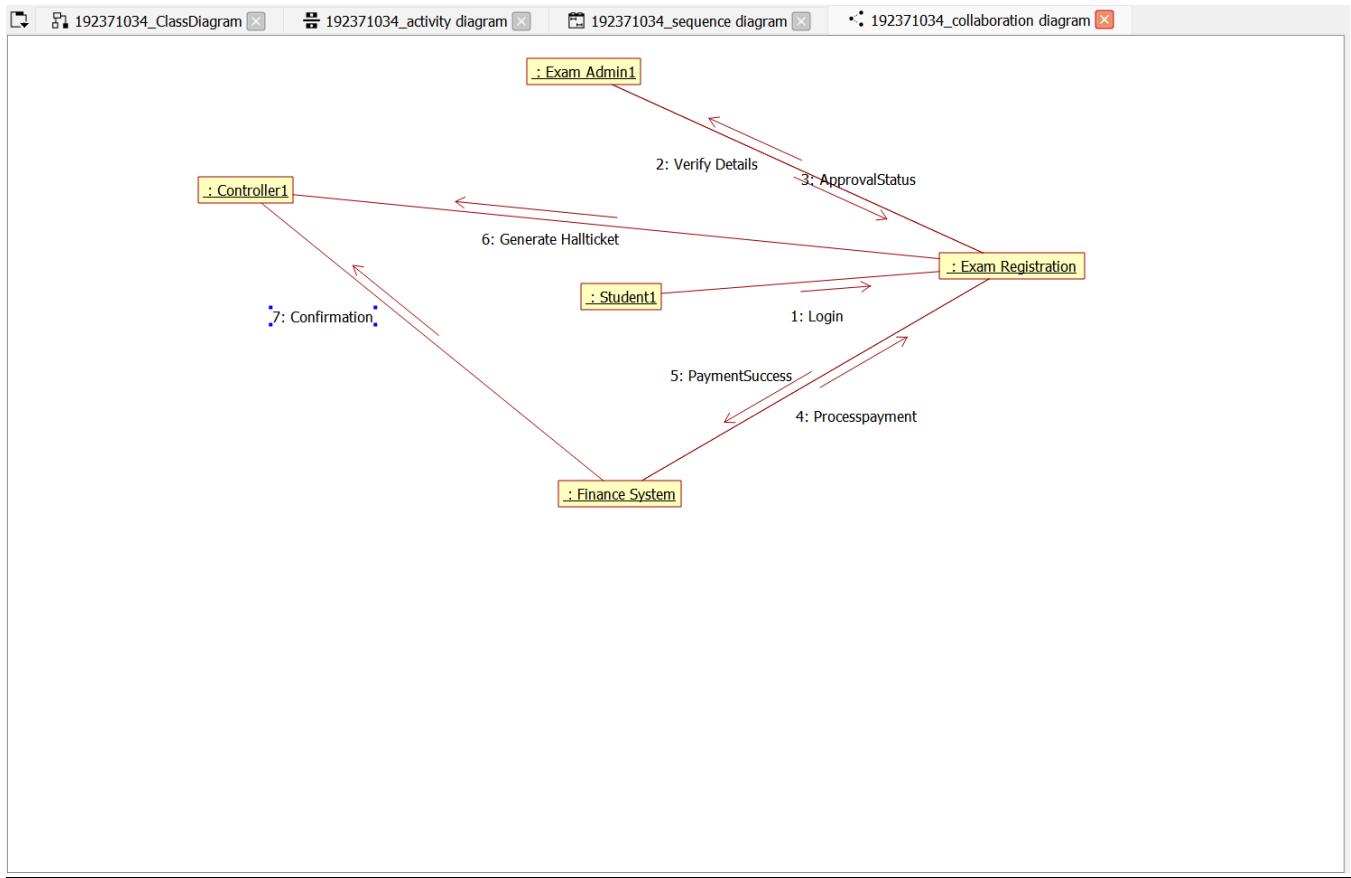
This diagram consists of the objects, messages and return messages.

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



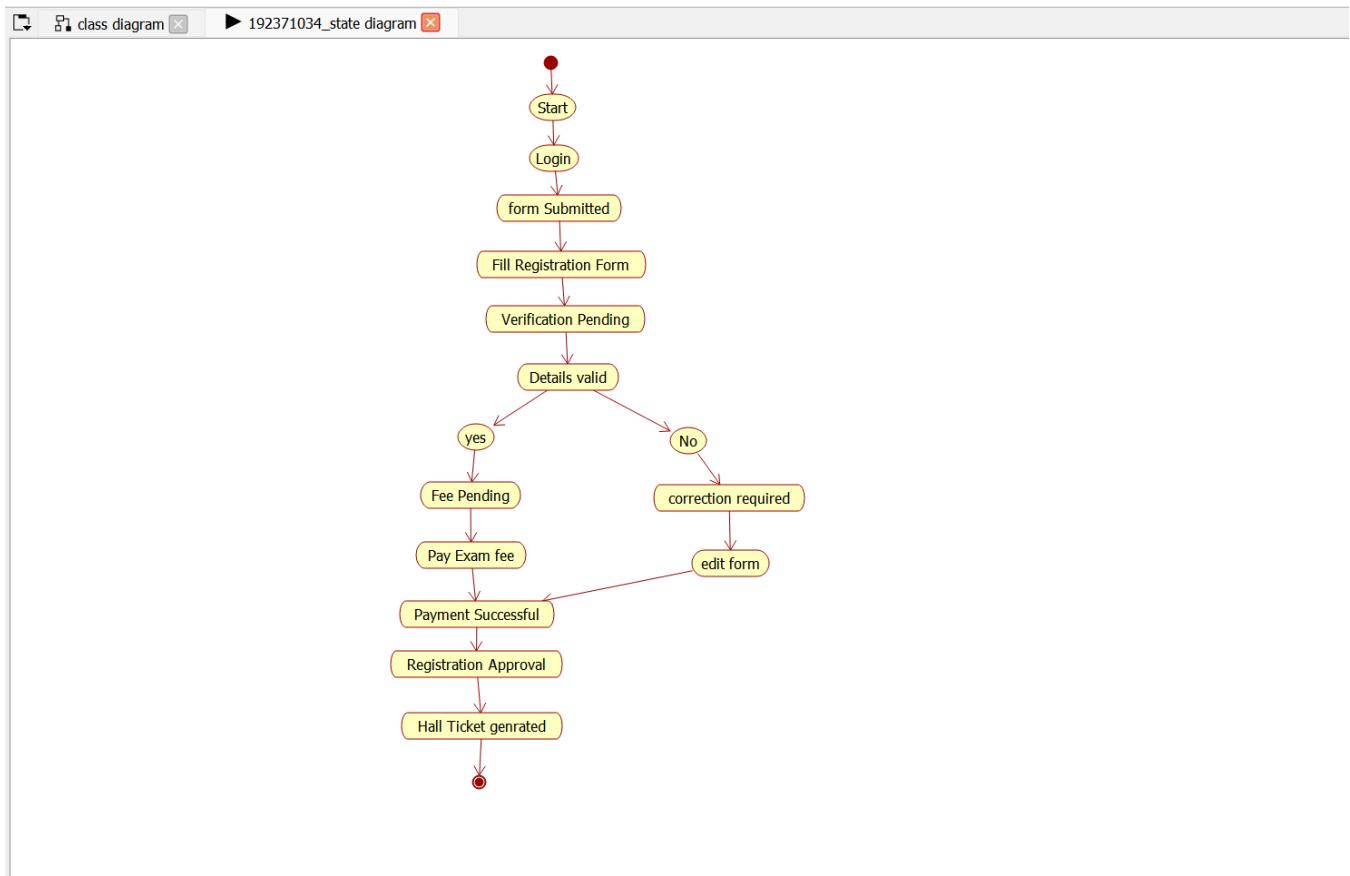
## **COLLABORATION DIAGRAM:**

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



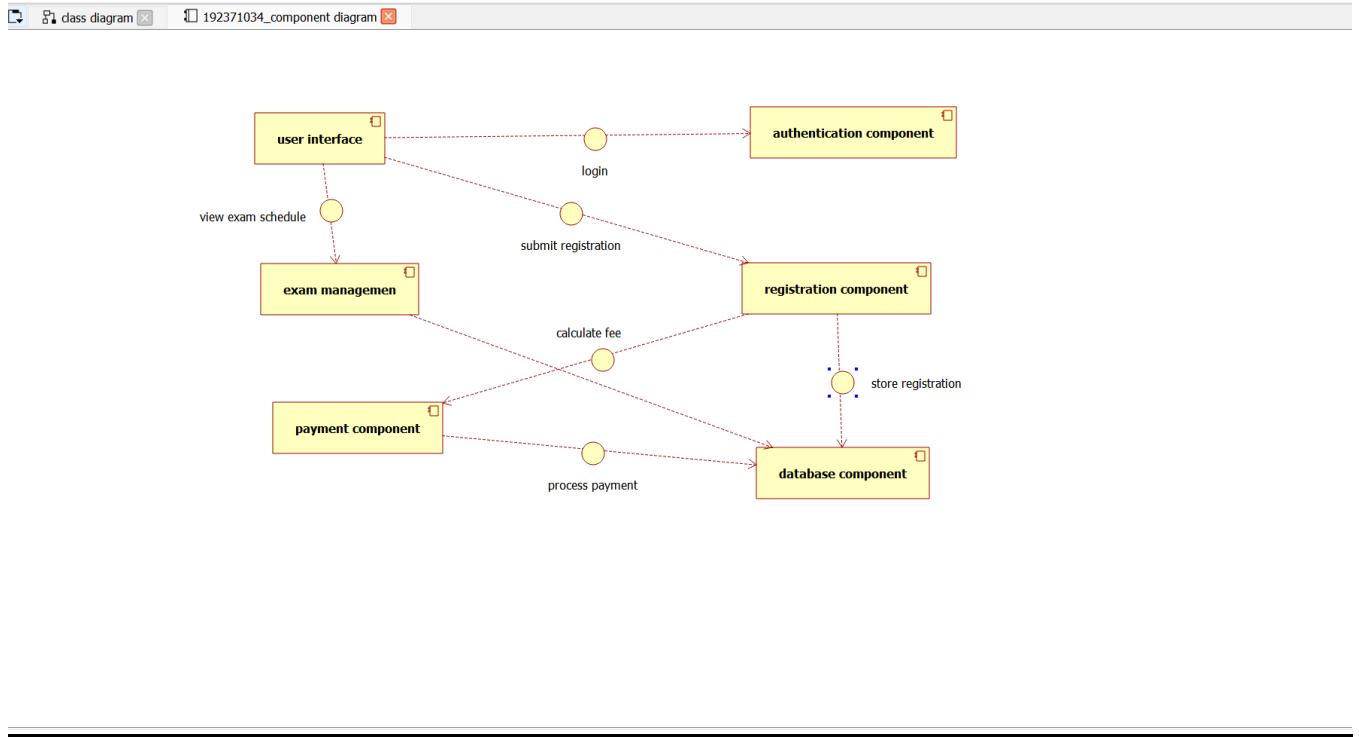
## STATE CHART DIAGRAM:

The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle surrounding a small dot.



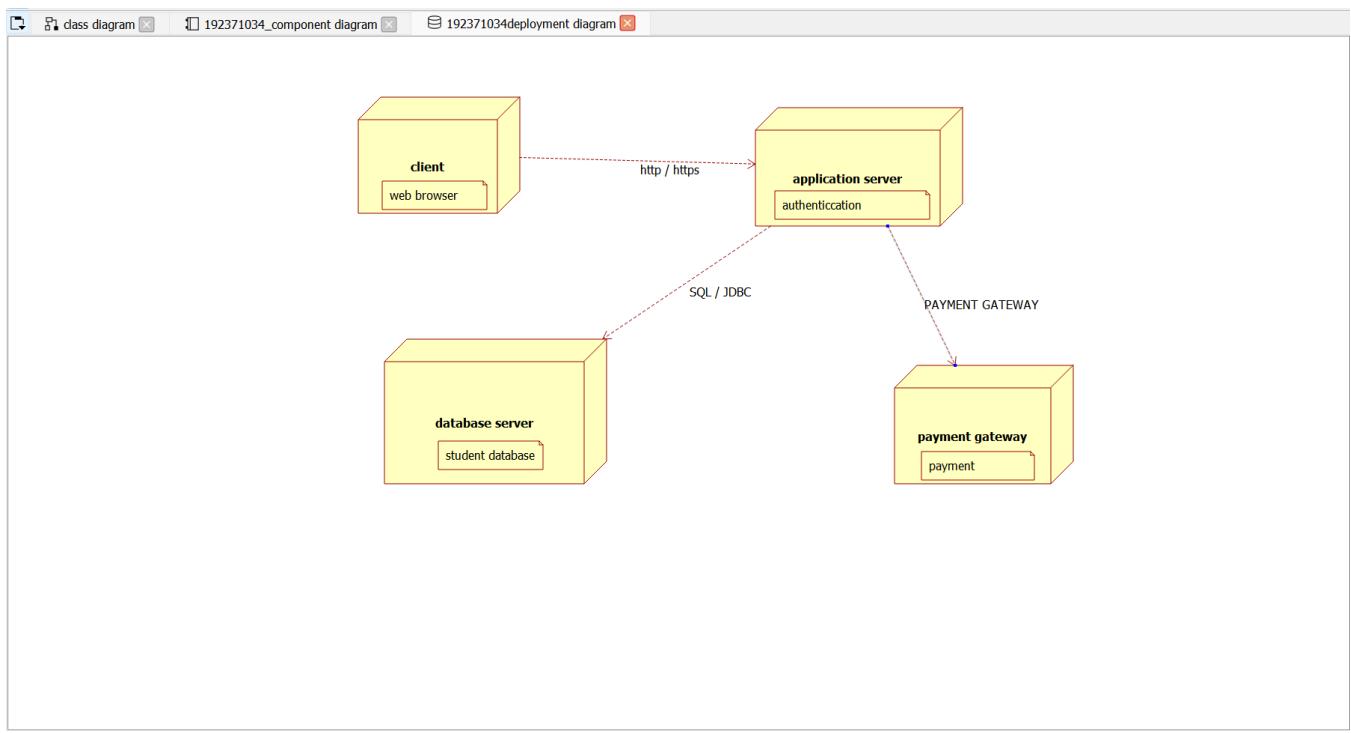
## **COMPONENT DIAGRAM:**

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association



## **DEPLOYMENT DIAGRAM:**

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

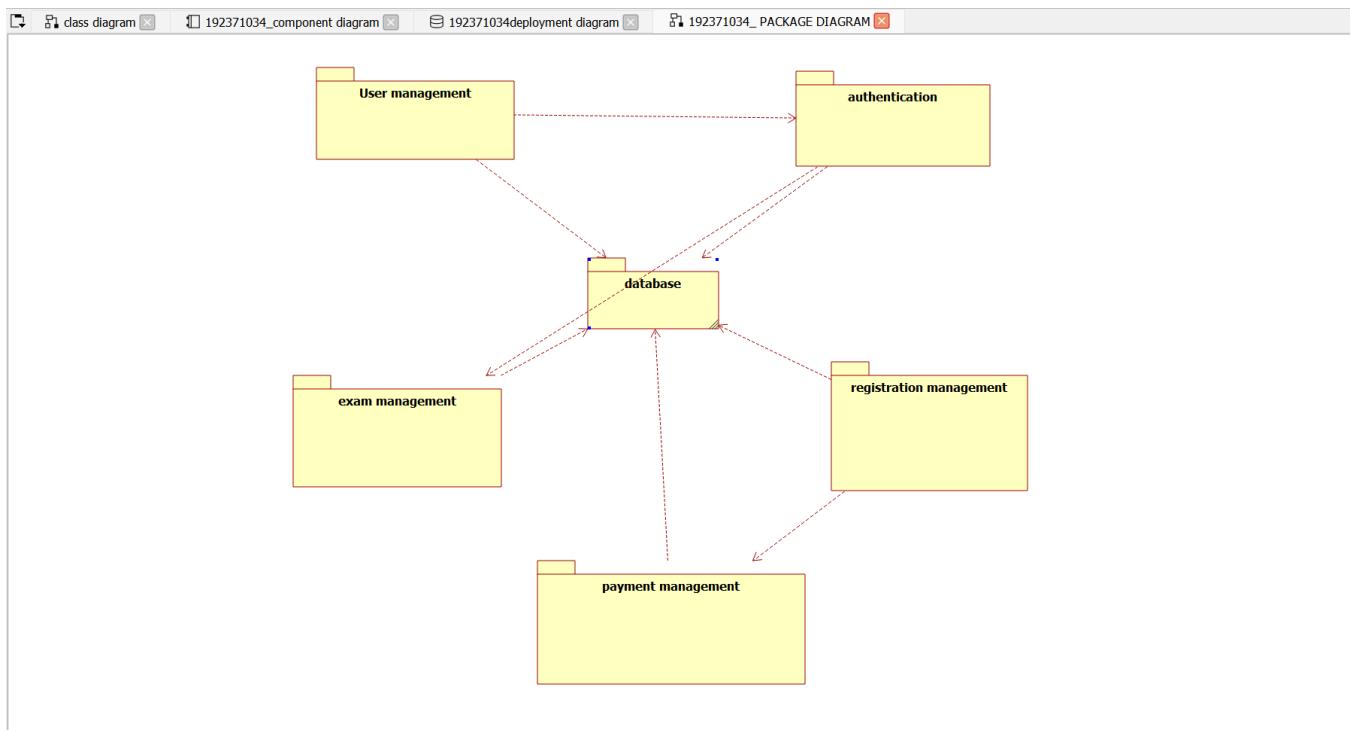


## PACKAGE DIAGRAM:

A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

There are three types of layer. They are

- o User interface layer
- o Domain layer
- o Technical services layer



## **PROGRAM CODING:**

### **CENETRAL EDUATIONAL SYSTEM:**

Public class central educational system

{

    Public integer student details;

    Public void valid proof()

    {

    }

}

### **EDUCATIONAL OFFICER:**

Public class educational officer

{

    Public integer id no;

    Public string name;

    Public void verification of proof()

    {

    }

    Public void issue hall ticket()

    {

    }

}

### **STUDENT:**

Public class student

{

    Public integer student details;

    Public void payment of fees()

    {

```
}

Public void receive hall ticket()

{

}

}
```

**RESULT:**

Thus the diagrams [use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Exam registration system.