

Exp 3:

EXAM REGISTRATION SYSTEM

AIM:

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

SOFTWARE REQUIREMENTS SPECIFICATION:

1.0 Hardware Requirements

1.1 Software Requirements

1.2 Problem Analysis and Project Plan

1.3 Project Description

1.4 Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLANNING

The 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.

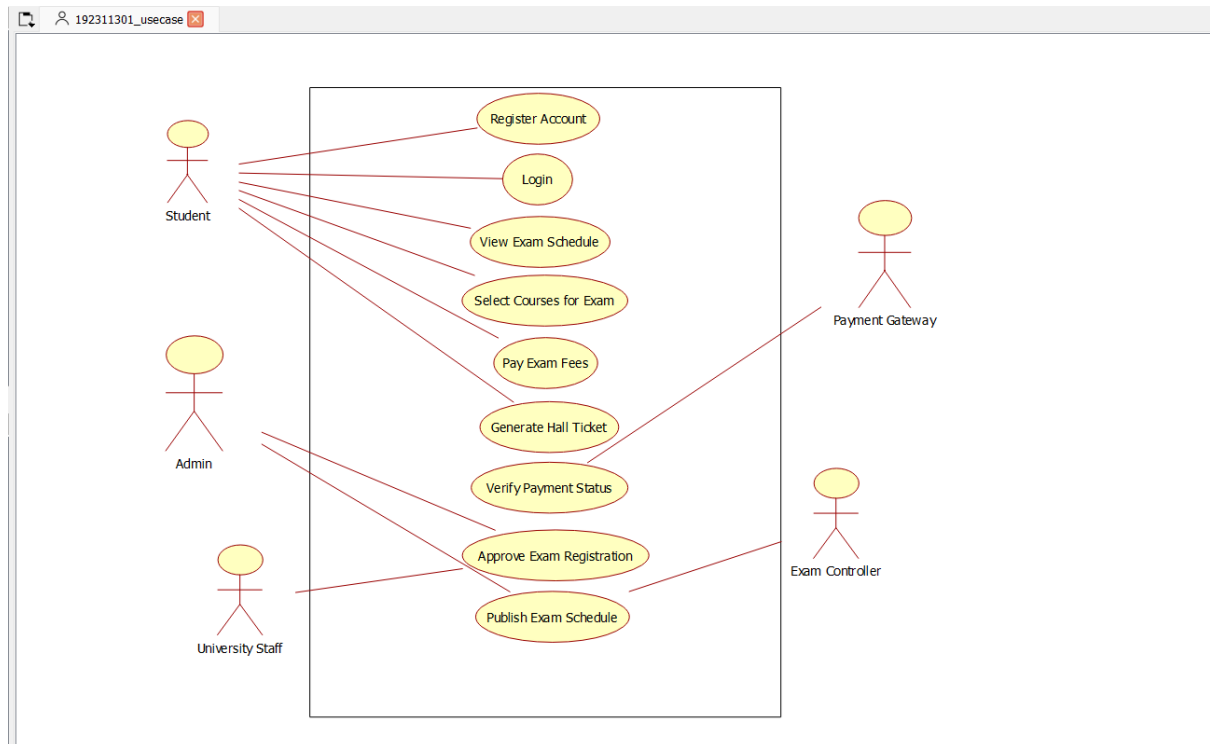
1.3 PROJECT DESCRIPTION:

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.

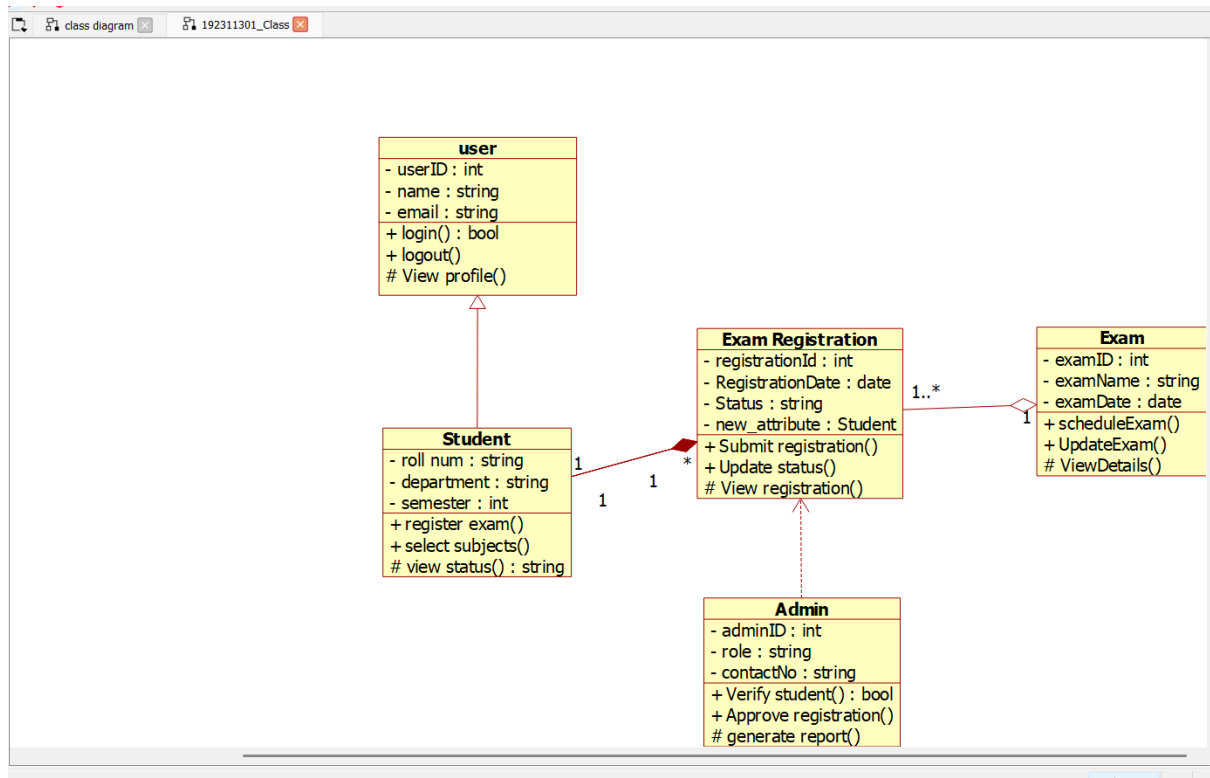
1.4 REFERENCES:

IEEE Software Requirement Specification format.

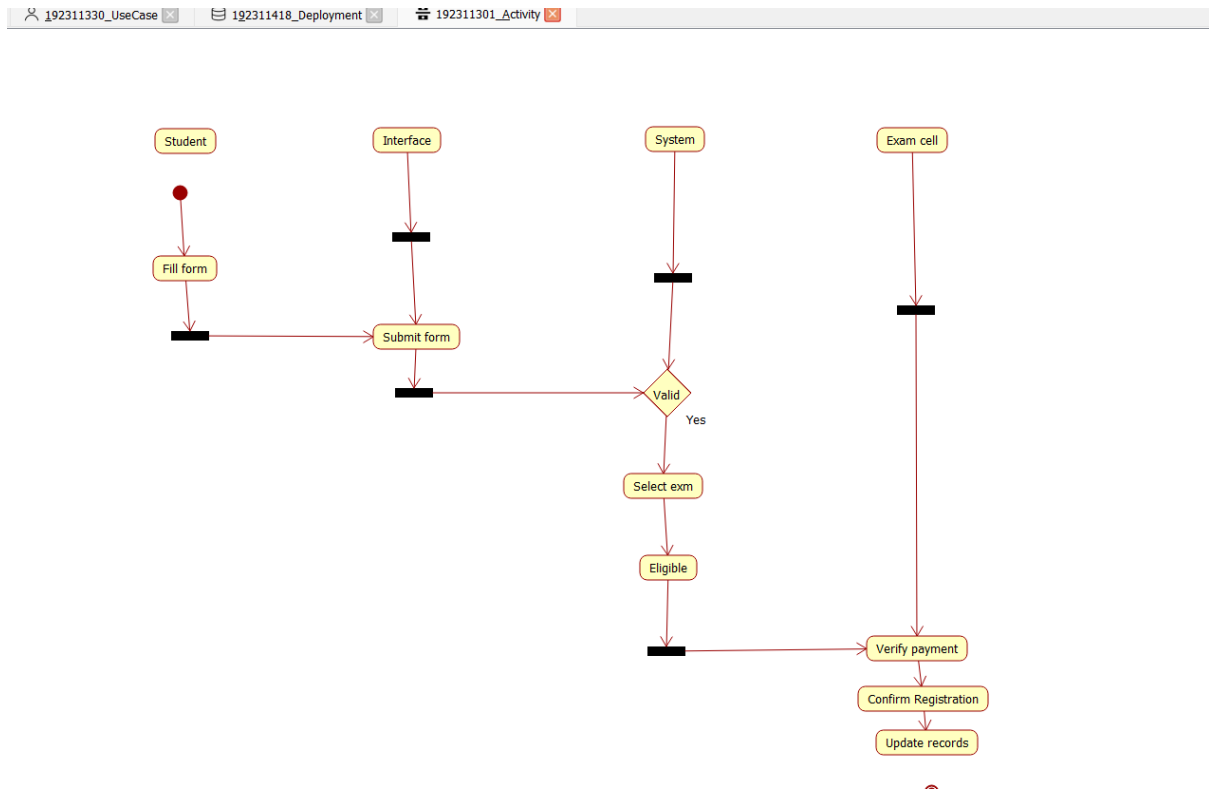
Use case diagram:



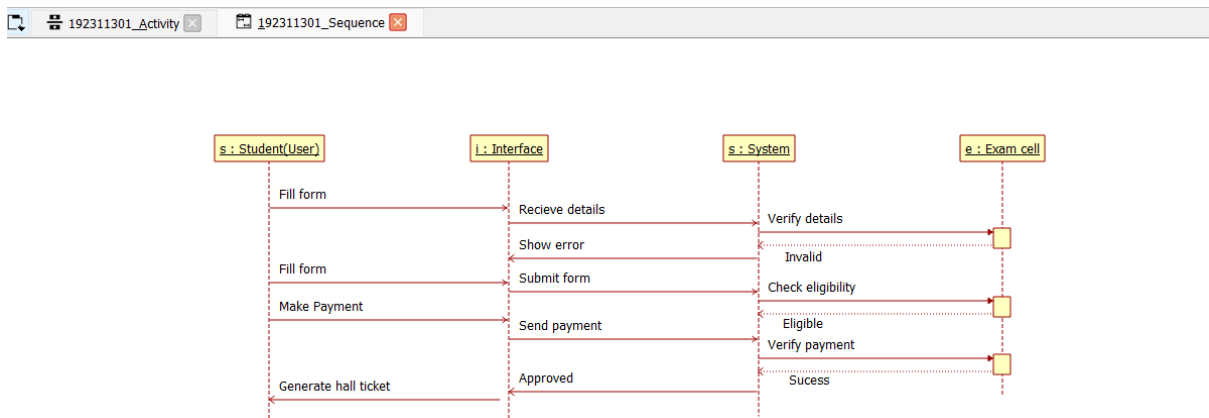
Class Diagram:



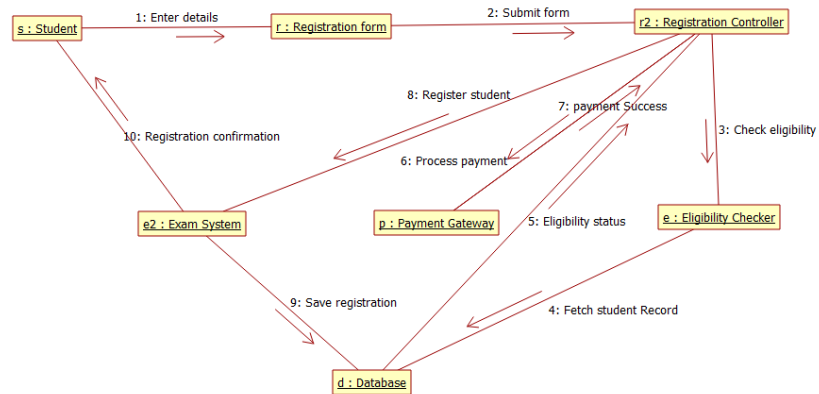
Activity Diagram:



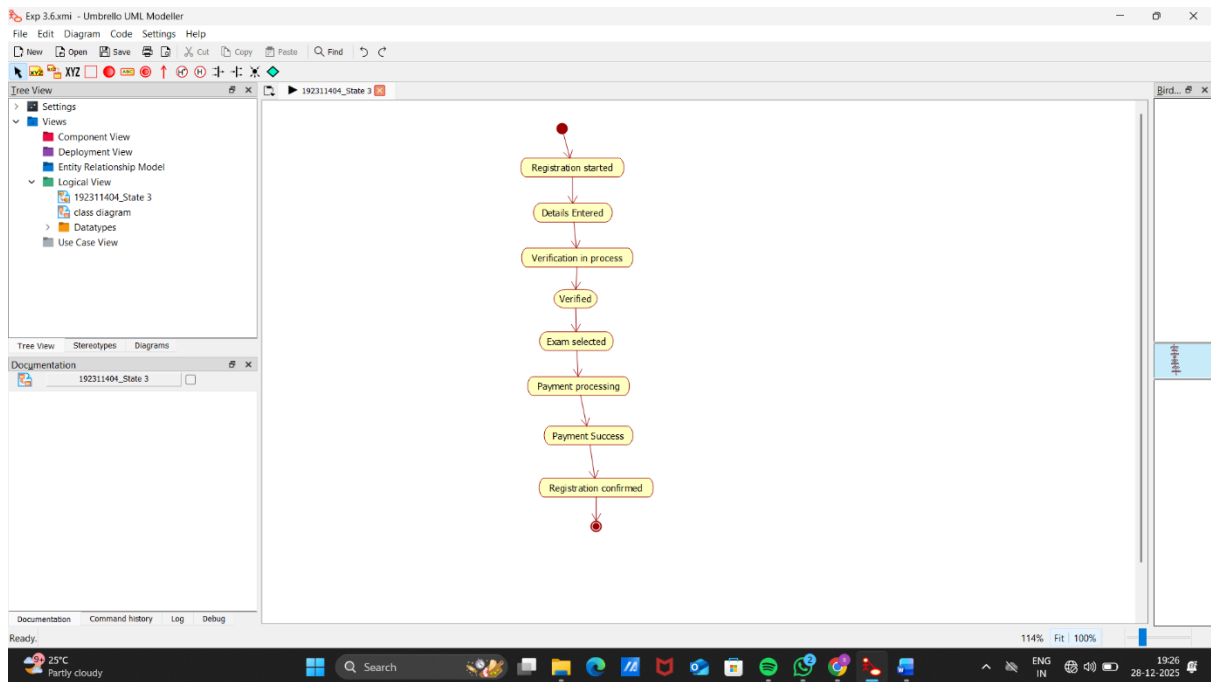
Sequence diagram:



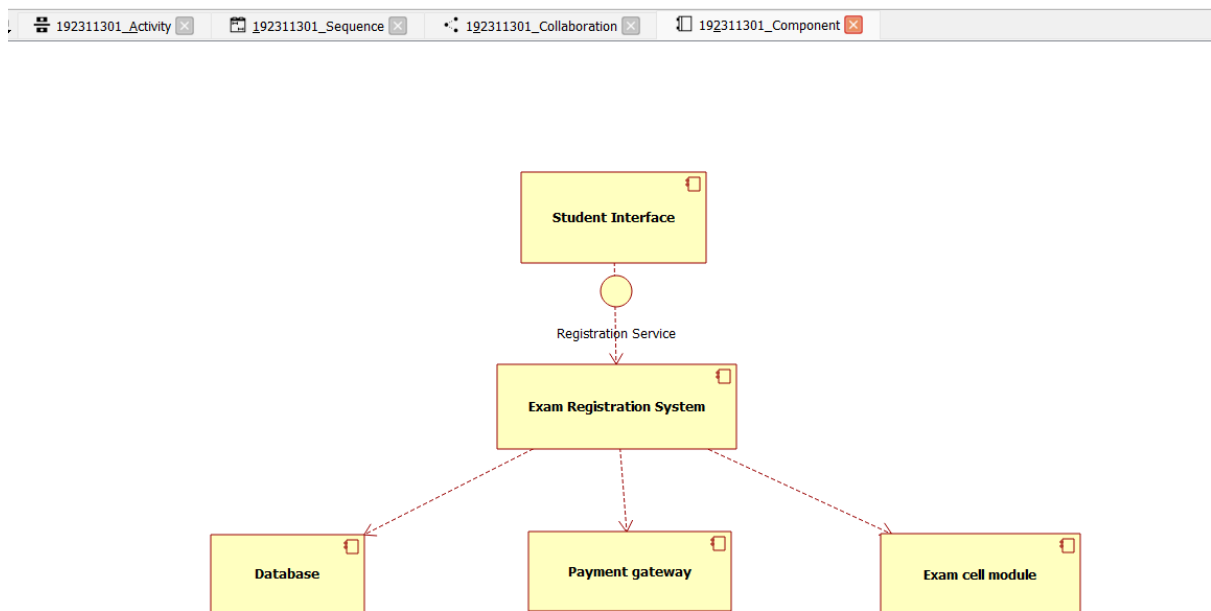
Collaboration Diagram:



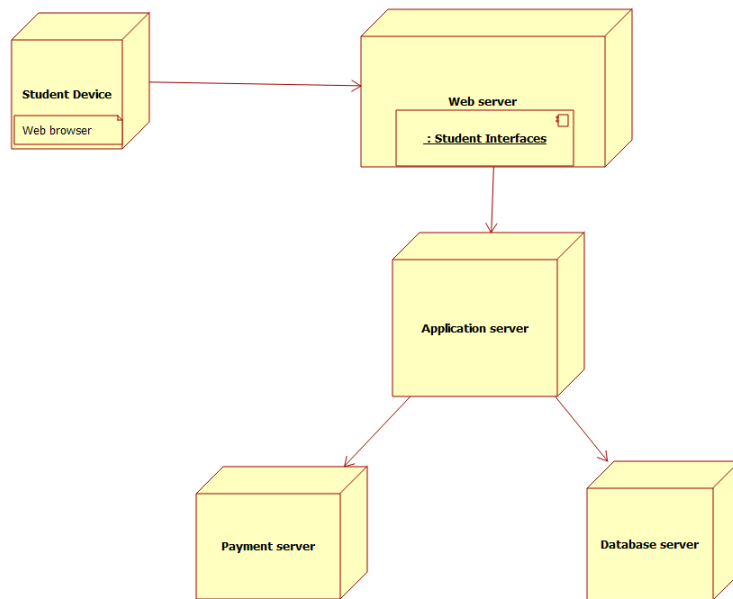
State Diagram:



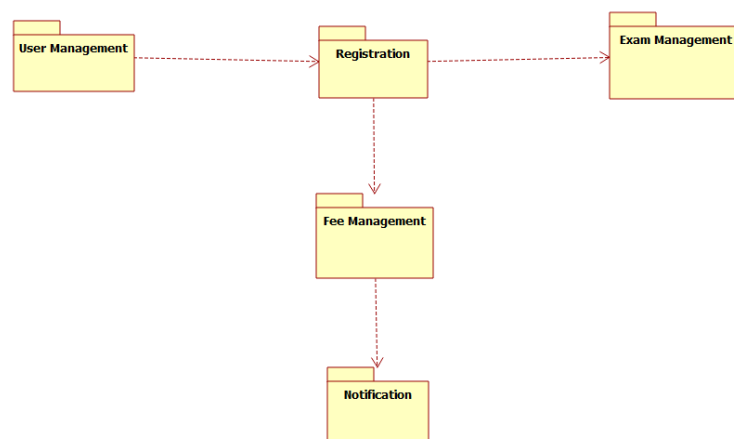
Component Diagram:



Deployment Diagram:



Package Diagram:



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.