

EX NO:5	ONLINE COURSE RESERVATION SYSTEM
DATE:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Online course reservation system.

SOFTWARE REQUIREMENTS SPECIFICATION:

SL.NO	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 requirement form the customer is got and the requirements about the course registration are defined. The requirements are analyzed and defined so that is enables the student to efficiency select a course through registration system. The project scope is identified and the problem statement is prepared.

1.3 PROJECT DESCRIPTION:

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.

1.4 REFERENCES:

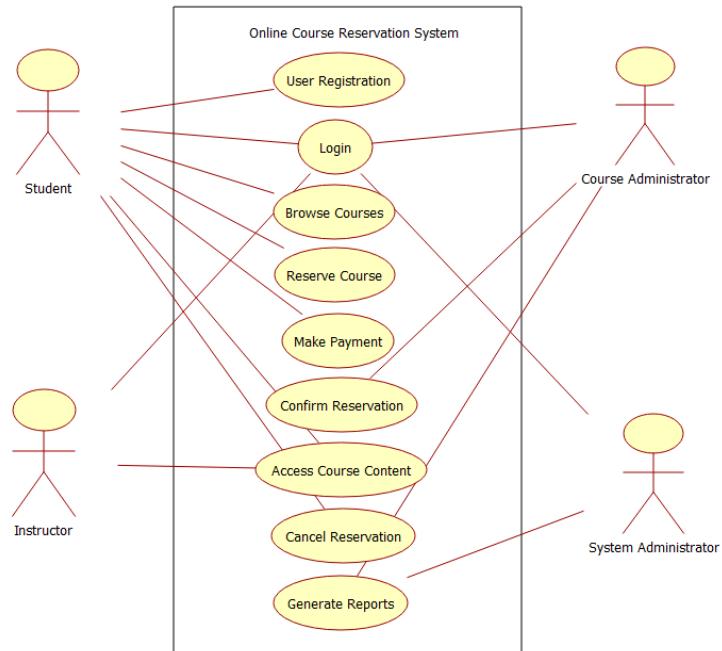
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Student, University.

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



CLASS DIAGRAM:

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

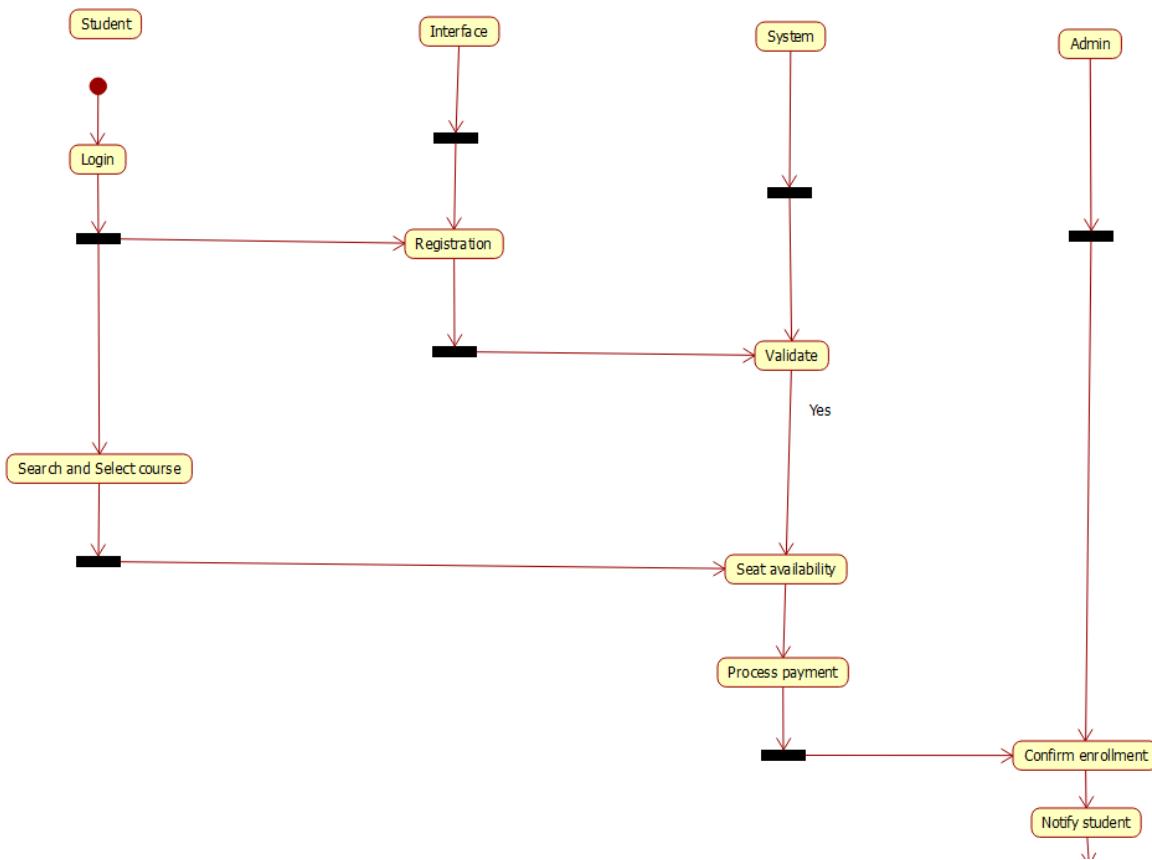
CLASSES	ATTRIBUTES	OPERATIONS
Central management system	Store details	Verify()
Student	Name and address	Browse()
University	Store details	Verify()

ACTIVITY DIAGRAM:

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

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

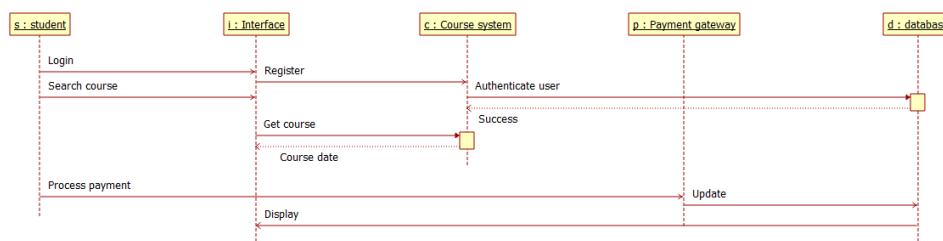
Decision box: check availability or not.



SEQUENCE DIAGRAM:

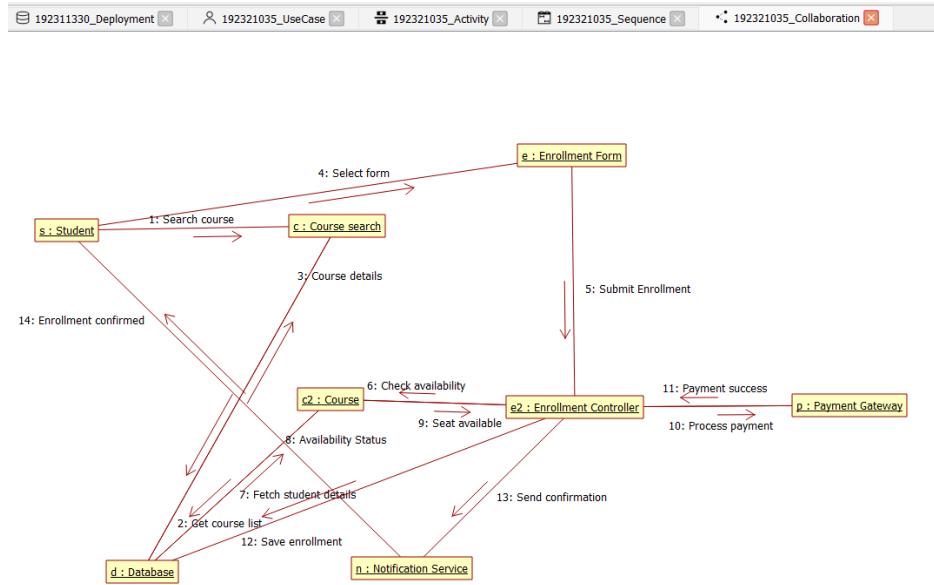
This diagram consists of the objects, messages and return messages.

Object: Student, University, Central management system



COLLABORATION DIAGRAM:

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

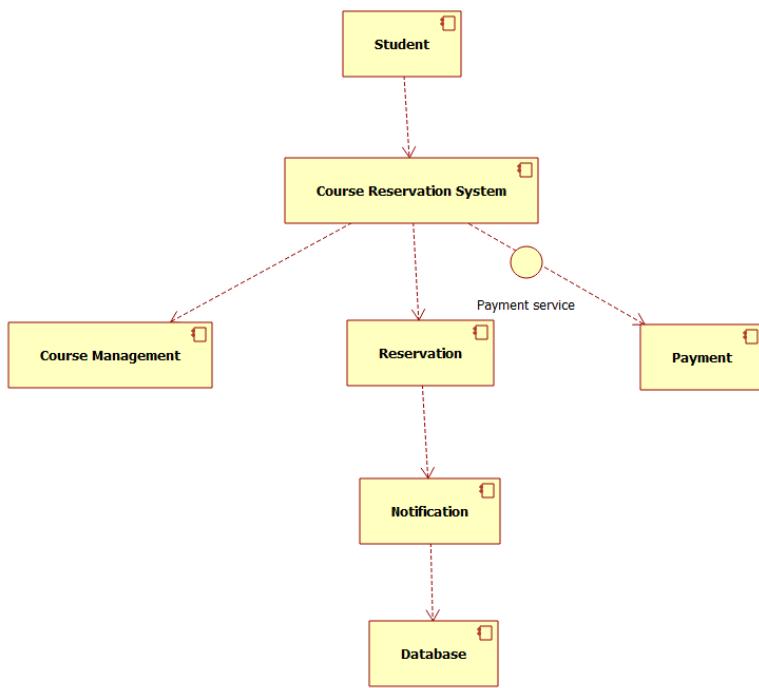


STATE CHART DIAGRAM:

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 the lifetime behaviour of a single objects

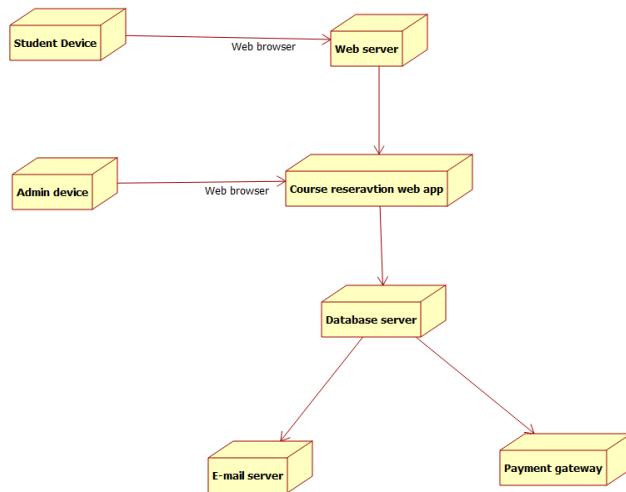
COMPONENT DIAGRAM:

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-dimentional box. Dependencies are represented by communication association.

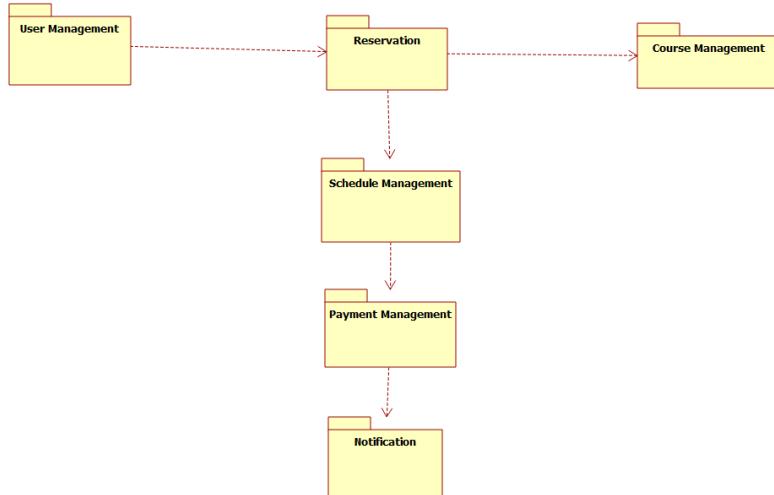


PACKAGE DIAGRAM:

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:

CENTRAL MANAGEMENT SYSTEM

Public class central management system

{

 Public integer details;

 Public integer verify details;

 Public void verify()

 {

 }

 Public void enroll()

 {

 }

}

STUDENT:

Public class student

{

```
Public integer name;  
Public integer address;  
Public integer marks;  
Public void browse()  
{  
}  
Public void select()  
{  
}  
Public void register()  
{  
}  
}
```

UNIVERSITY:

```
Public class university  
{  
    Public integer store details;  
    Public integer verify details;  
    Public void verify()  
    {  
    }  
    Public void enroll()  
    {  
    }  
}
```

RESULT:

Thus draw the diagrams [usecase, activity, sequence, collaboration, class,state chart, component, deployment , package] for the Online course reservation system has been designed executed and output is verified.