

# **ONLINE COURSE REGISTRATION SYSTEM**

---

## **PROJECT REPORT**

### **18CSC202J- OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY**

**(2018 Regulation)**

**II Year/ III Semester**

**Academic Year: 2022 -2023**

**By**

**DHRUV DESHMUKH (RA2111028010125)**

**PAPAI MONDAL (RA2111028010116)**

**Under the guidance of**

**Dr. L.N.B. SRINIVAS**

**Associate Professor**

**Department of Networking and Communications**



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Kancheepuram**

**NOVEMBER 2022**

## **BONAFIDE**

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled “**ONLINE COURSE REGISTRATION SYSTEM**” is the bonafide work of **DHRUV DESHMUKH (RA2111028010125), PAPAI MONDAL (RA2111028010116)** who undertook the task of completing the project within the allotted time.

### **Signature of the Guide**

Dr. L.N.B. Srinivas

**Associate Professor**

Department of NWC,

SRM Institute of Science and Technology

### **Signature of the II Year Academic Advisor**

Dr. Saravanan M

**Professor and Head**

Department of NWC,

SRM Institute of Science and Technology

### **About the course:-**

18CSC202J- Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

### **Objectives:**

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

### **Course Learning Rationale (CLR): The purpose of learning this course is to:**

- 1.Utilize class and build domain model for real-time programs
- 2.Utilize method overloading and operator overloading for real-time application development programs
- 3.Utilize inline, friend and virtual functions and create application development programs
- 4.Utilize exceptional handling and collections for real-time object-oriented programming applications
- 5.Construct UML component diagram and deployment diagram for design of applications
- 6.Create programs using object-oriented approach and design methodologies for real-time application development

### **Course Learning Outcomes (CLO): At the end of this course, learners will be able to:**

- 1.Identify the class and build domain model
- 2.Construct programs using method overloading and operator overloading
- 3.Create programs using inline, friend and virtual functions, construct programs using standard templates
- 4.Construct programs using exceptional handling and collections
- 5.Create UML component diagram and deployment diagram
- 6.Create programs using object oriented approach and design methodologies

**Table 1: Rubrics for Laboratory Exercises**

(Internal Mark Splitup:- As per Curriculum)

<b>CLAP-1</b>	5=(2(E-lab Completion) + 2(Simple Exercises)( from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
<b>CLAP-2</b>	7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
<b>CLAP-3</b>	7.5=(2.0(E-lab Completion(80 Pgms)+ 2.0 (Simple Exercises)( from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	<b>2 Mark - E-lab Completion 80 Program</b> Completion from 10 Session (Each session min 8 program) <b>2 Mark - Code to UML</b> conversion GCR Exercises <b>3.5 Mark - Hacker Rank</b> Coding challenge completion
<b>CLAP-4</b>	5= 3 ( Model Practical) + 2( Oral Viva)	<ul style="list-style-type: none"> <li>• <b>3 Mark – Model Test</b></li> <li>• <b>2 Mark – Oral Viva</b></li> </ul>
<b>Total</b>	25	

### COURSE ASSESSMENT PLAN FOR OODP LAB

S.No	List of Experiments	Course Learning Outcomes (CLO)	Blooms Level	PI	No of Programs in each session
1.	Implementation of I/O Operations in C++	CLO-1	Understand	2.8.1	10
2.	Implementation of Classes and Objects in C++	CLO-1	Apply	2.6.1	10
3.	To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram.	CLO-1	Analysis	4.6.1	Mini Project Given
4.	Implementation of Constructor Overloading and Method Overloading in C++	CLO-2	Apply	2.6.1	10
5.	Implementation of Operator Overloading in C++	CLO-2	Apply	2.6.1	10
6.	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams	CLO-2	Analysis	4.6.1	Mini Project Given
7.	Implementation of Inheritance concepts in C++	CLO-3	Apply	2.6.1	10
8.	Implementation of Virtual function & interface concepts in C++	CLO-3	Apply	2.6.1	10
9.	Using the identified scenarios in your project, draw relevant state charts and activity diagrams.	CLO-3	Analysis	4.6.1	Mini Project Given
10.	Implementation of Templates in C++	CLO-3	Apply	2.6.1	10
11.	Implementation of Exception of Handling in C++	CLO-4	Apply	2.6.1	10
12.	Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram.	CLO-5	Analysis	4.6.1	Mini Project Given
13.	Implementation of STL Containers in C++	CLO-6	Apply	2.6.1	10
14.	Implementation of STL associate containers and algorithms in C++	CLO-6	Apply	2.6.1	10
15.	Implementation of Streams and File Handling in C++	CLO-6	Apply	2.6.1	10

## **LIST OF EXPERIMENTS FOR UML DESIGN AND MODELLING:**

**To develop a mini-project by following the exercises listed below.**

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

## **Suggested Software Tools for UML:**

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

# ABSTRACT

This system will maintain a **course registration database** which will allow the student to view the course details, to register the course and select the course. The students enter details of her or him profile and choice of selecting the course while going to manual method it is so difficult.

So this system uses several programming and database techniques to elucidate the work involved in this process. He has the certain privileges to add the course status and to approve the issue of course. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of course.

Course Registration System will help the student to gather information about a particular course and then they can easily register them self in a particular course.

The management of the institution can easily see the records of the students and course and fees.

This presentation is about the analysis of Course Registration System. Diagram's purpose is to present System clearly and completely as possible.

# PURPOSE

- System will allow the registration of students in particular course.
- System has inbuilt validation system to validate the entered data.
- After successful submission, system will give unique registration no for each student.
- Student can login into system by using username and check the details of course, faculty and department.

# ACTORS INVOLVED

The various actor involved in this process are student, admin, server and database.

**Student:** User who register the course.

**Admin:** Check whether a student is eligible or not.

**Server:** Provide the required data from database.

**Database:** Store the registered data and act accordingly to the server.



# MODULE DESCRIPTION

The user enters the username and password and chooses if the user is student or administrator. If entered details are valid, the user's account becomes available. If it is invalid and appropriate message is displayed to the user.

A student can search all the courses available to him and he chooses the best course he wants. The student can view course duration, faculty and department of the course he may choose. When a student has successfully chosen a course, he can register to that course. Upon registration, the student's details are stored in database.

After registration to any course, the student may see the details of current course. He may wish to know details about fees and other information. The administrator also has the privilege to display details of the students and the corresponding course for which they have registered.

The administrator has to perform duties of maintaining the course details any change to the course structured is maintained by the administrator. Each failure processes are informed to the user on the screen based on the above information.

# USECASE DIAGRAM

Use case diagram is a combination of actors and their functionality represented by graphical components. Here the various actors involved are student, admin, server, data base.

The course registration system has the following use-cases

## **ACTORS INVOLVED:**

1. Student
2. Registrar

## **USE-CASE NAME: LOGIN**

The user enters the username and password and chooses if the user is student or Registrar. If entered details are valid, the user's account becomes available. If it is invalid, an appropriate message is displayed to the user.

## **USE-CASE NAME: COURSE SELECT**

In this use case, a student can search all the courses available to him and choose the best course he wants. The student can view the course duration, faculty and department of the courses he may choose.

## **USE-CASE NAME: TIME SELECTION**

When a student has successfully chosen a course, he can register to that course. Upon registration, the student's details are stored in the database.

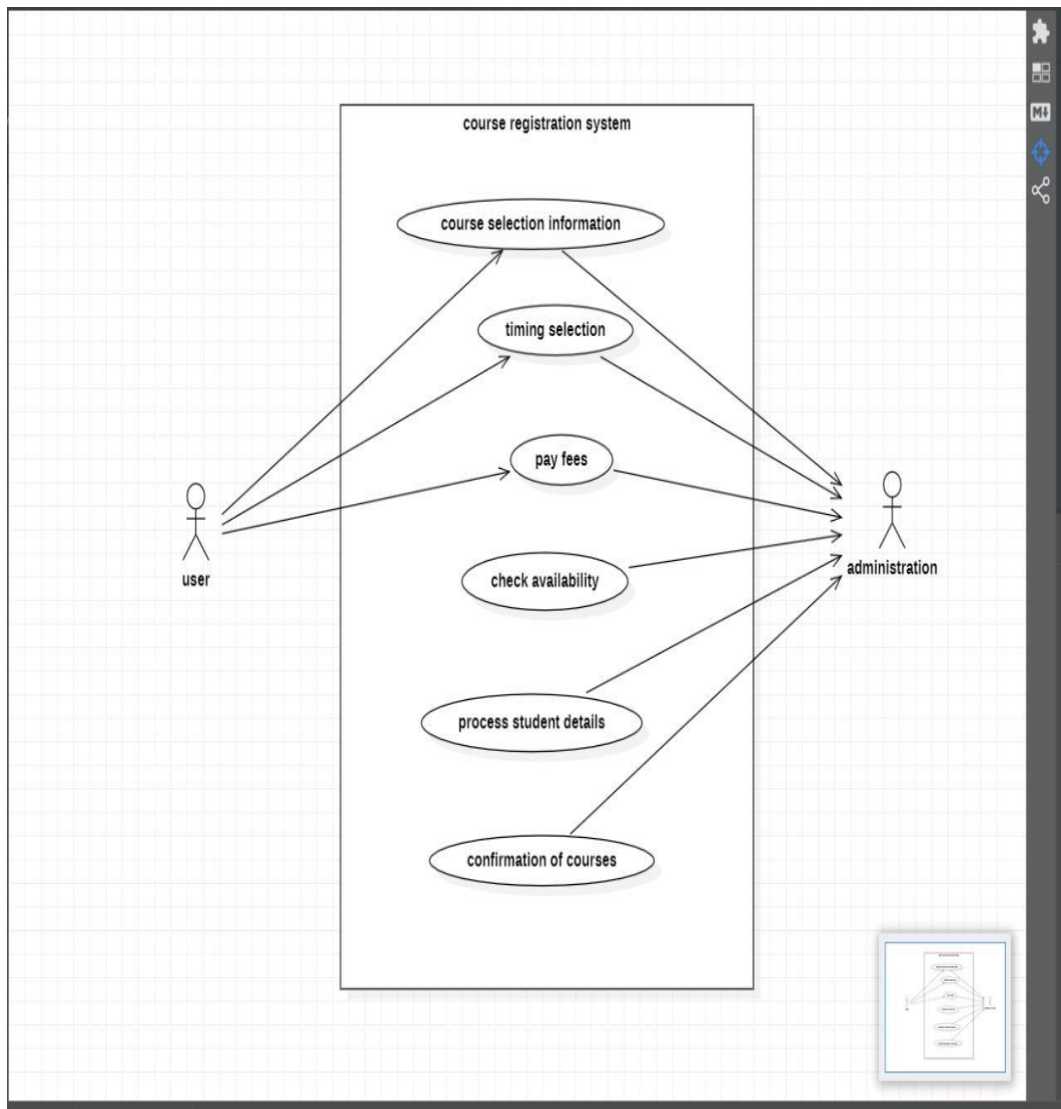
## **USE-CASE NAME: PAY FEE**

After registration to any course, the student may see the details of his current course. He may wish to know details about fees and other information.

## **USE-CASE NAME: CHECK CONFORMATION**

The student tries to check the status in which category applied. The system displays the status information to the student

# USECASE DIAGRAM

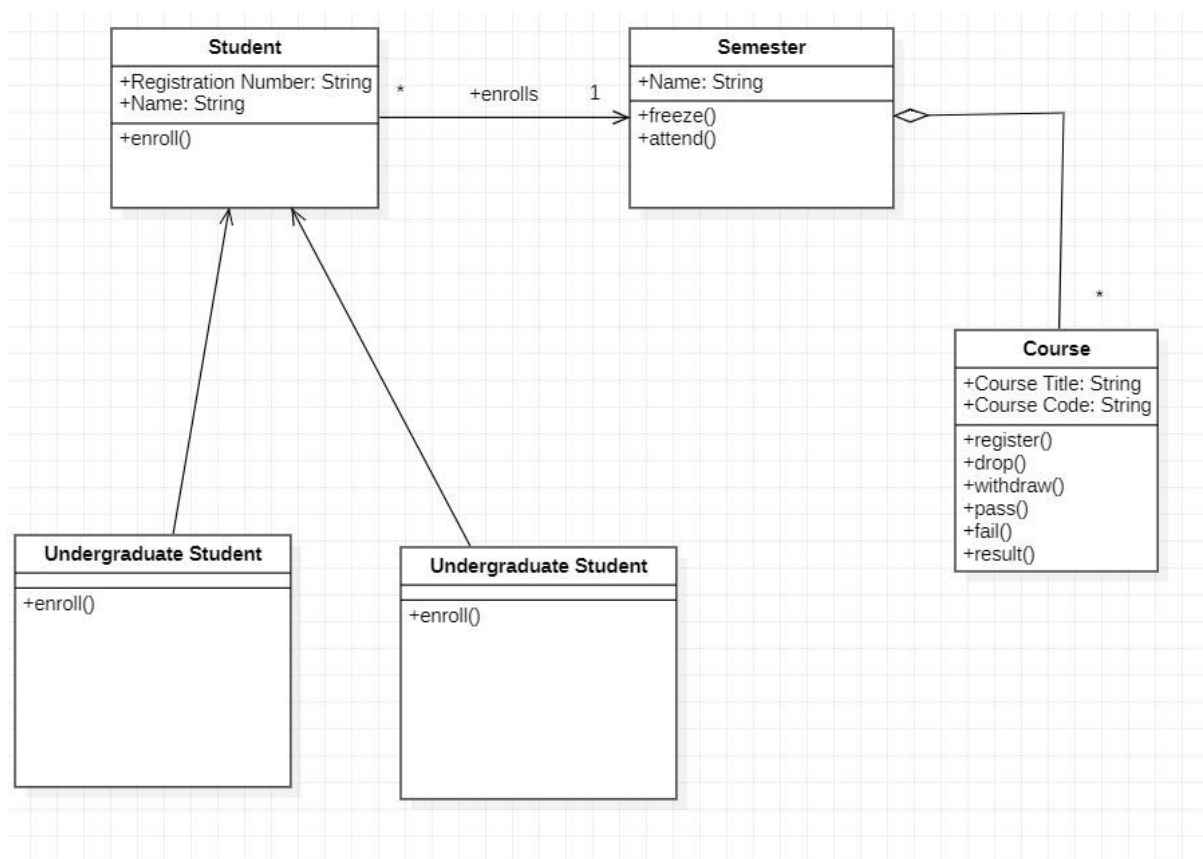


# CLASS DIAGRAM

The class diagram is a graphical representation of all classes used in the system and their operations, attributes and relations.

□ The course register make use of following classes:

- **Student**
- **Semester**
- **course**
- **undergraduate student**



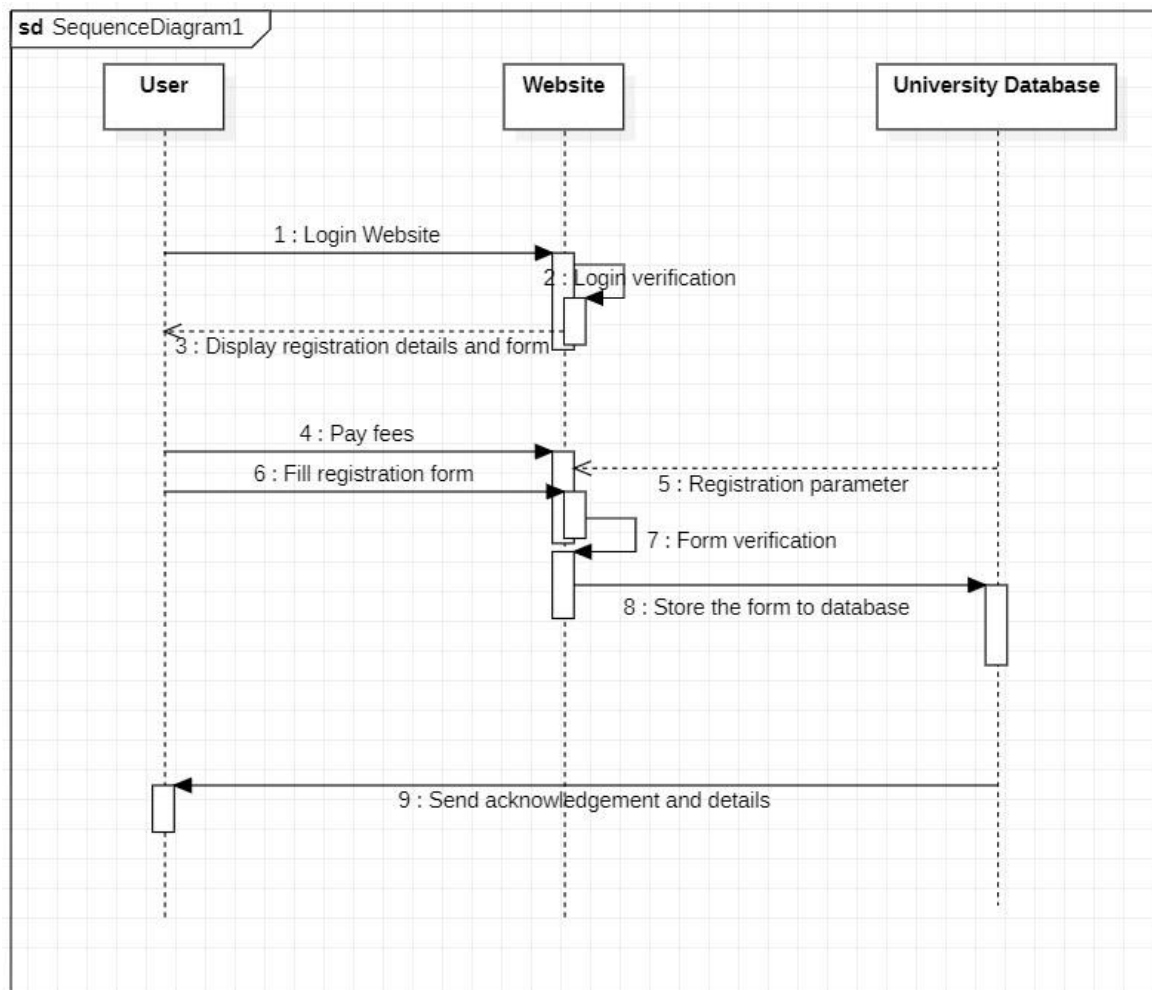
□ Thus from the class diagram it describes the Online Course Registration System which has classes,

□ Their attributes, and the relationship among the object.

□ The main classes of the course registration system are Course, registration, fees, and student.

## SEQUENCE DIAGRAM

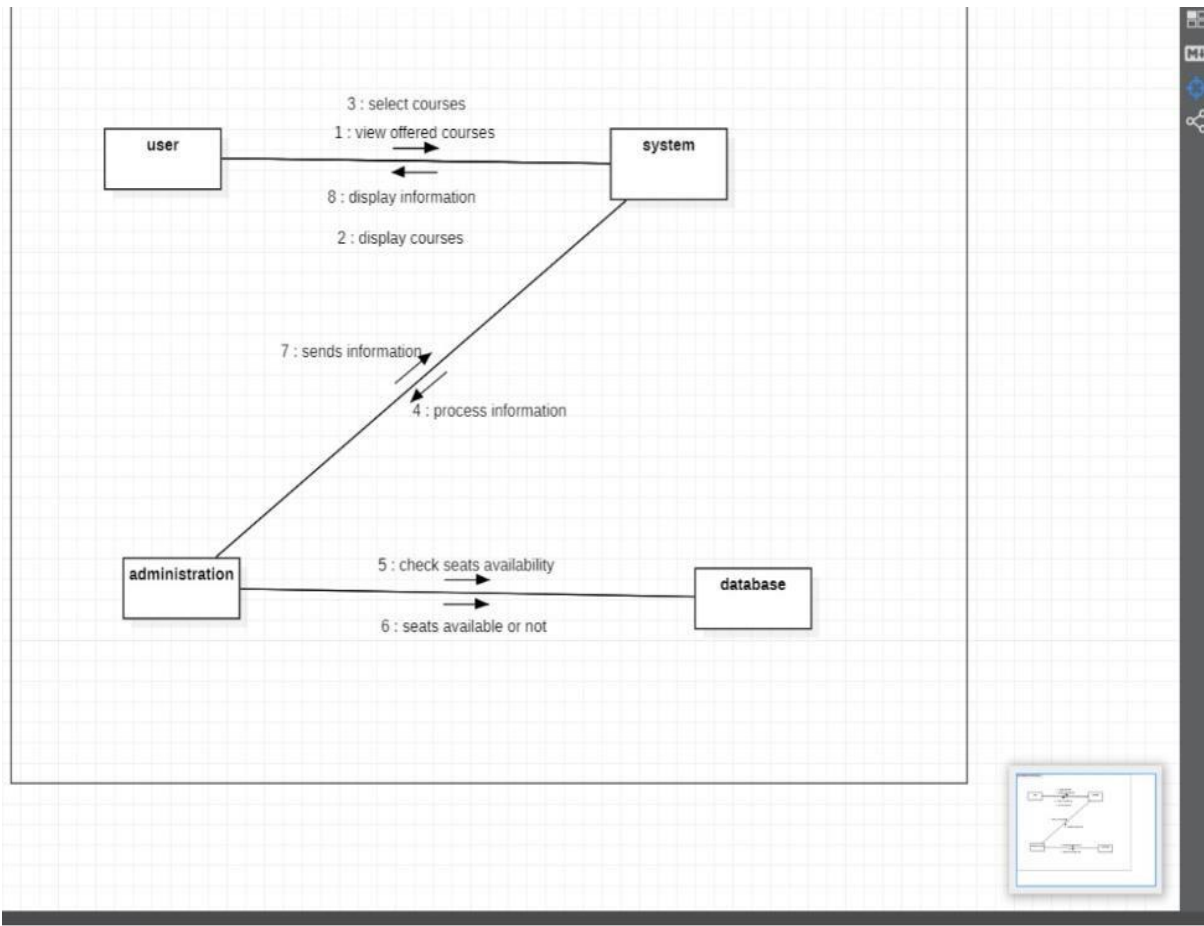
A sequence diagram shows an interaction arranged in time sequence. It shows object participating in interaction by the message. The exchange arranged in time sequence critical dimension represent time and horizontal dimension represent object.



- ☐ It is also represented by the order in which things occur and how the objects in the system send message to one another.
- ☐ The sequence diagram for each USE-CASE that exists when a user administrator, check status and new registration about course registration system are given.
- ☐ User have to first login to the system before performing any operation. The user has to provide the necessary details to the system for login.
- ☐ Next the database send some parameters like Registration to the website inorder to verify and store from the data and finally send all necessary detail.

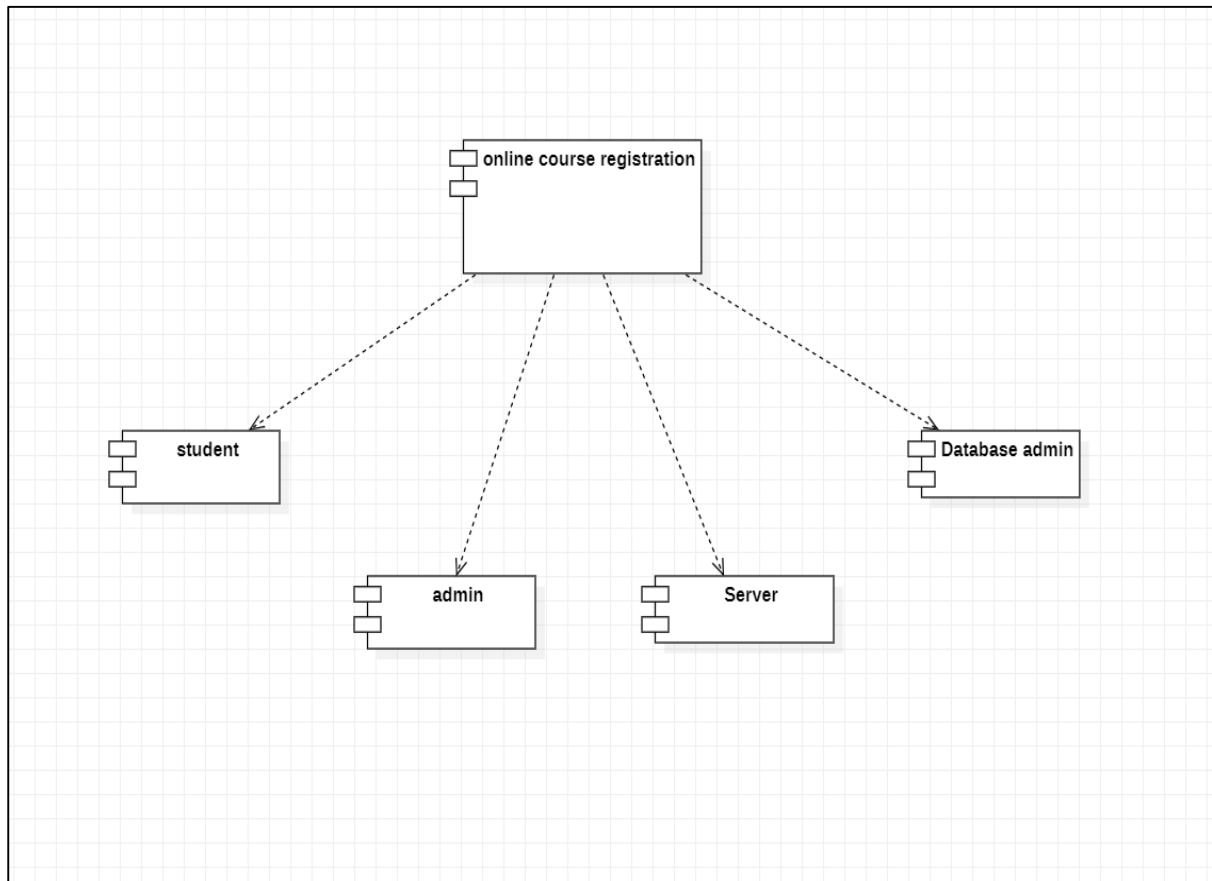
# COLLABRATION DIAGRAM

Communication diagram illustrate object interact on a group or network format in which the object can be placed where a diagram in collaboration diagram the object can be placed in anywhere on the diagram. The collaboration diagram comes from the sequence.



- ☐ Collaboration diagrams emphasize the structure of object interaction instead of the Interaction sequences.
- ☐ Object (User): The interaction between objects takes put in a system.
- ☐ Relation/Association: Association among objects is connected by connecting them.
- ☐ Messages: An arrow that commencing from one object to the destination object.

# COMPONENT DIAGRAM



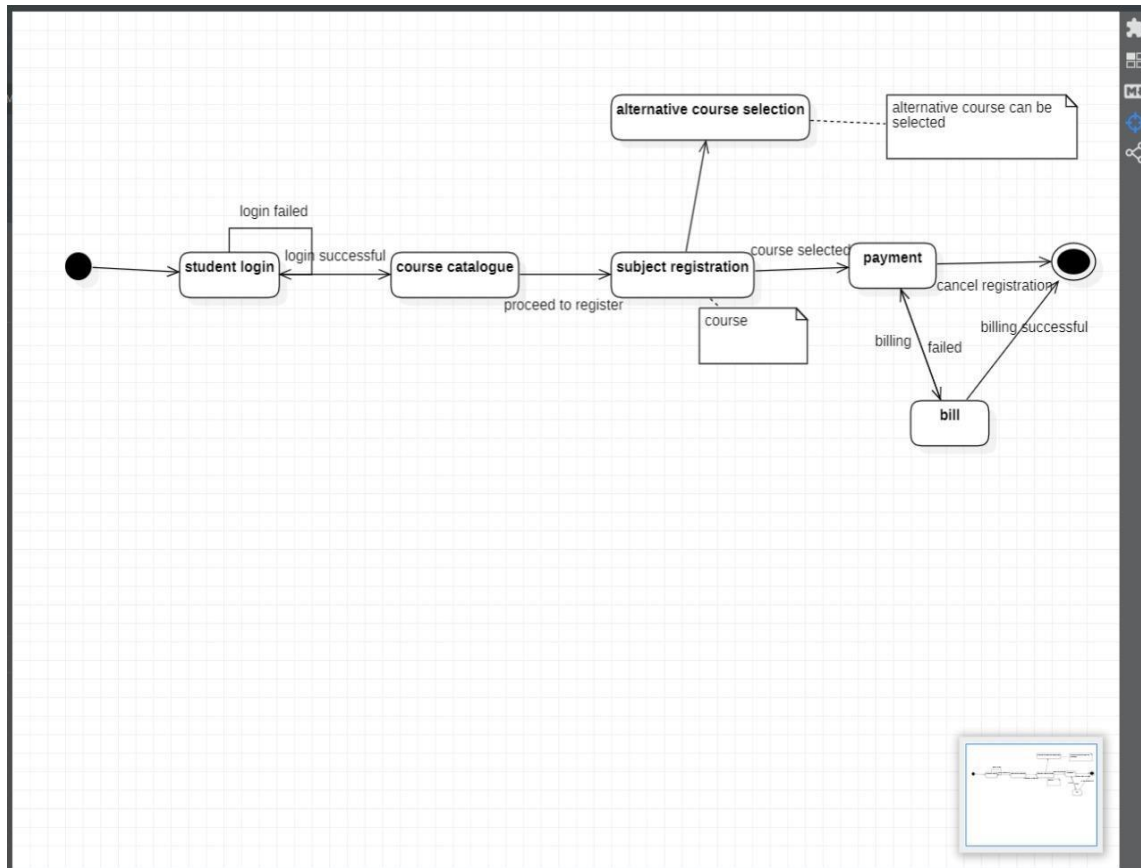
This is the component diagram which shows components, provided and required interfaces, ports and relationships between the Online courses, Database, Admin, Server and the Students. Component diagram describes the organization and wiring of the physical components in a system.

It consists of

- ☐ Course Component
- ☐ Fees Component
- ☐ Registration Component
- ☐ Admin's Component
- ☐ Students Component

## STATE CHART DIAGRAM

State chart diagram describes the flow of control from one state to another state. A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.



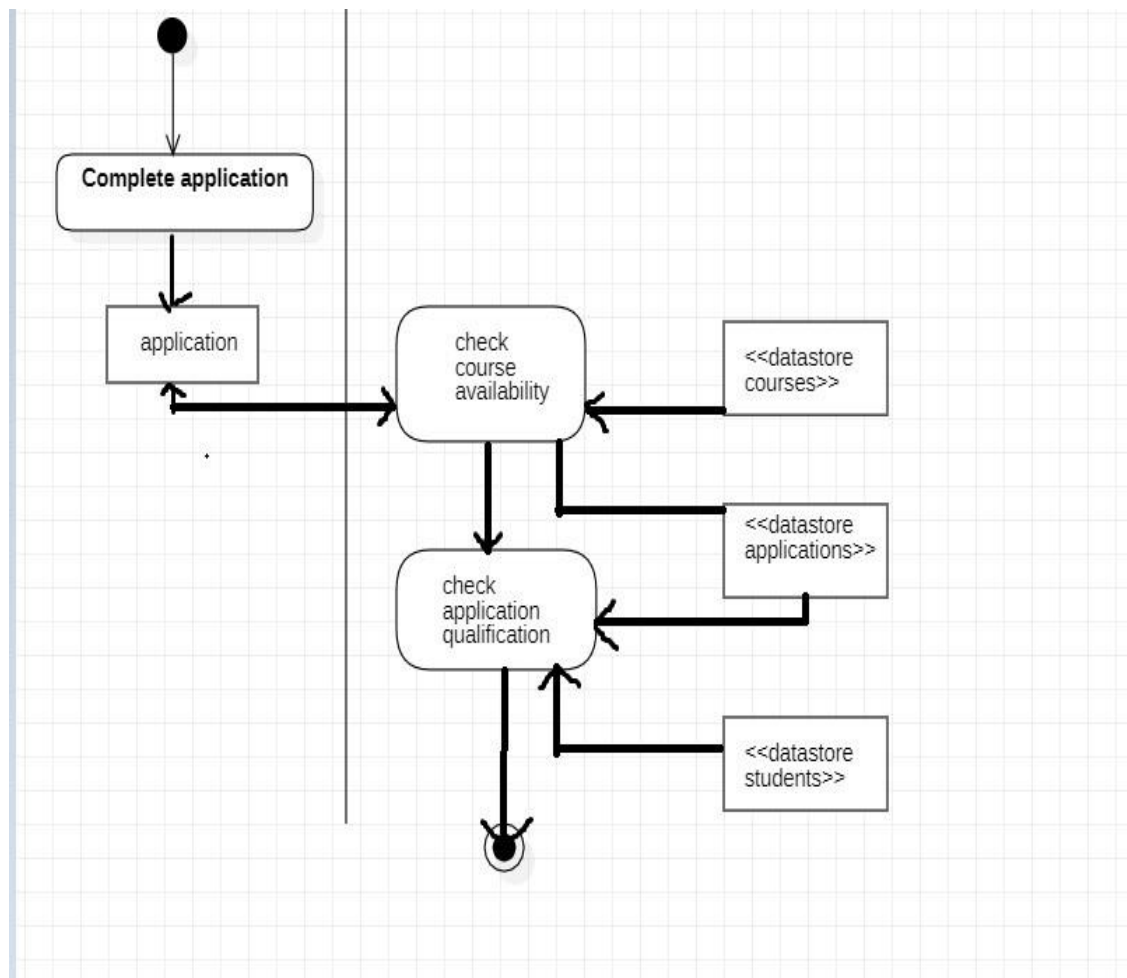
- ☐ State chart diagram is a graph of nodes connected by a communication association.
- ☐ It represent the intermediate stages such as student login, course catalogue, registration course, check alternate course, register accepted, fees payment, logout.
- ☐ Here, we used a black filled circle represent the initial state of a system
- ☐ The rectangles with rounded corners that are labelled with the name of the state.
- ☐ In this we used an Transition in order to get and make the connections.
- ☐ We used the frames and finally reached to the final state with the circle inside the dotted one



## ACTIVITY DIAGRAM

The UML activity diagram for course registration system is a diagram that presents the flow of system activities. It is one of the methods used to document the system behaviour in terms of activities and development.

□ Additionally, the activity diagram uses symbols to define the overall workflow of the course registration system. It is composed of activities, decisions, and paths (flows).



- The symbols that we used in this activity diagram are starting state, action state, control flow.
- Advantages are in this the complex stage or steps in a software system can be Explained easily and diagrammatically.

## **CONCLUSION**

Thus the various UML diagram for the online course registration system was successfully created.

## **REFERENCE**

□ Here are the following UML diagrams done in this report:

- **USE CASE DIAGRAM**
- **CLASS DIAGRAM**
- **SEQUENCE DIAGRAM**
- **COMMUNICATION DIAGRAM**
- **STATE CHART DIAGRAM**
- **ACTIVITY DIAGRAM**

□ In this we used the StarUml as it is a software engineering open source tool for system modelling Purpose

**\*\*\*\*\*THE END\*\*\*\*\***