

DR B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR



LAB FILE OF OBJECT ORIENTED ANALYSIS and DESIGN CSX-429

SESSION: JUL - DEC 2020

SUBMITTED TO-
Dr. Jagdeep Kaur
CSE Department

SUBMITTED BY-
Amit Kumar (17103009)
Ankit Goyal (17103011)
Group - G-1
Branch - CSE

INDEX

S. No	Table of contents	Page Numbers	Remarks
1.	Problem Statements	3	
2.	Software Requirement Specification	4	
3.	Use case Diagram	11	
4.	Class Diagram	13	
5.	Sequence Diagram	15	
6.	Activity Diagram	17	
7.	Collaboration Diagram	19	

PRACTICAL NO. 1

Aim: To understand the problem statement and prepare a detailed statement by exploring different aspects of the problem.

Student registration System will be a software which will be helpful for students as well as the school authorities. Proposed student registration system will eliminate all the manual intervention and increase the speed of whole process.

In the proposed system, students can add their details by themselves. So the overhead of the school authorities and the teachers is become less. The objective of Student Registration System is to allow the administrator of organization to edit and find out the personal details of a student and allows the student to keep up to date his profile.

System will allow student to fill the registration forms online. System will have inbuilt validation system to validate the entered data. It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc.

After successful login, student will be able to add the courses in which he/she is interested. He can view all the added courses along with the related course details like Faculty name, Start date, Course Fee etc. After adding the courses, student can submit the Fee of courses. System will give unique registration number for each student. Student can login into system by using their registration number and check the course and fee details.

The main functional elements of Student Registration System are Register Student, Submit Fee, Edit Profile, Change Password, Add or Remove the courses, View Fee Deposition history. Overall, it'll make Student registration easier.

PRACTICAL NO. 2

Aim: To prepare a Software Requirement Specification (SRS) Document.

1. INTRODUCTION

Student registration System will be a software which will helpful for students as well as the school authorities. Student will be able to register them using this system. Students will be able to register themselves. They would be able to submit fee, change password, add or remove the courses also.

1.1 Purpose

The purpose of this project is to develop the student registration system. The main purpose of this project is to develop a system which can facilitate the student registration process. Student can check the available courses and register to them. He is also able to submit the fee for these courses. Moreover, the user can check all these features from their home.

1.2 Intended Audience and Reading suggestions

The intended audience of this document is a developer's group or system administrators who would use the student registration system.

1.3 Project Scope

- Manually doing the registration causes a significant workload on the administration. Using the student registration system will help doing the registration online in fast in easy way.
- Student can check their details related to courses, fee submission history.
- The students can register from their homes without visiting the college/organization.

1.4 Definitions, Acronyms and Abbreviations

1. Courses -> Subject available in the college
2. SQL -> Structured query Language
3. IDE -> Integrated Development Environment

1.5. Technologies to be used

- HTML
- CSS
- JavaScript
- PHP
- MySql

1.6. Tools to be used

- VS Code Editor

1.7. Overview

SRS includes two sections overall description and specific requirements

- Overall Description will describe major role of the system components and interconnections
- Specific Requirements will describe roles & functions of the actors.

2. OVERALL DESCRIPTION

2.1 Product Perspective

The proposed Student registration system will make the job of Institute Administrator easier. It will eliminate all the manual intervention and increase the speed of whole process. So the overhead of the institute authorities and the teachers is become less. Institute authorities can get information of any student at any time. It is also easy for student for their registration because they will not have to waiting in queue for fee payment and form submission.

2.2. Product functionalities

R.1: Register

1. Description: First the user will have to register/sign up. There are Three different type of users.
2. Professor: Professor have to provide details about his/her name, course, address, phone number, email id.
3. Student: The users have to provide details about his/her name of address, phone number, email id.
4. Admin: Admin have to provide details about his/her name, course, address, phone number, email id.

R.1.1: Sign up

1. Input: Detail about the user as mentioned in the description.
2. Output: Confirmation of registration status and a registration number and password will be generated and mailed to the user.
3. Processing: All details will be checked and if any error are found then an error message is displayed else a registration number and password will be generated.

R.1.2: Login

1. Input: Enter the registration number and password provided.
2. Output: User will be able to use the features of software.

R.2: Manage registration process by users.

R.2.1: For Student.

1. Add Course: Student can add the semester course.

2. Drop Course: Student can drop the course which is not related to semester.
3. View Registered Course: Student can view list of registered course.
4. Display Course: Student can view All the courses.
5. Fee Payment: Student can pay the institute fee.

R.2.2: For Professor

1. Display Course: A list of courses displayed.
2. Select Course: Select the courses to teach.
3. View Selected Course: A list of selected courses displayed
4. View course Offer: A list of offered courses displayed.

R.2.3: For Admin

1. Create New course: A new course is added in previous course list.
2. Remove course: Admin can remove the course which are not in curriculum.
3. Modify the course: Admin can modify the previous courses.

2.3. Operating Environments

This is a web-based application. It can run on browsers such as chrome, firefox or edge.

2.4. General constraints, assumptions and dependencies

• Constraints

- A user cannot register more than once with same email id.
- Must use email or mobile no. as a unique identifier for a user account.

• Assumptions and Dependencies

- The users and Administrator must have basic knowledge of computers and English Language.
- All users require to Login for registration.

3. SPECIFIC REQUIREMENTS

3.1. External interface requirements

3.1.1. User Interface

After login, Student can have their dashboard where they can Add, Drop, View registered Courses and also can pay the course fee. For Professors, they can enroll student, select course to teach and can get student details. For Admin he can add new courses, remove course and can also remove users.

3.1.2. Hardware interfaces

The server is directly connected to the client systems. The client systems have access to the database in the server.

3.1.3. Software interfaces

- **Front-end Client:** The applicant and Administrator online interface is built using HTML, CSS and Javascript. The administrator's local interface is built using NodeJS.
- **Web Server:** Heroku Hosting Server
- **Back End:** PHP, Mysql Database

3.1.4. Communication interfaces

Login checks authentication and authorization of the party. Furthermore, the password created by the user is allowed only if it follows standard protection rules. All the inputs entered by the user are checked and then validated. A user sends an HTTP request for required information. The information sent through a network that hides the data through RSA encryption.

3.2. Functional requirements

- **Making Account:** User sign up to the website and register by providing the details.
- **Select courses:** Student and professor select their respected semester course.
- **Registration:** When Student do all the process of registration at they have to pay the fee for successful registration.

3.3. USER CHARACTERSTICS

We have 3 levels of users:

1. Student: In the user module, user will perform following task.
 - Add Course
 - Drop Course
 - Display Course
 - Fee Payment
 - View Registered Course
2. Professor:
 - Display course
 - View Offer Course
 - Select Course
 - View selected Course
3. Admin:
 - Add New Course
 - Remove Course
 - Modify Course

3.4. User Documentation

1. You need to register first to use the application. Registration can be done via your email Id for security reasons.
2. After registering, you need to confirm your email from your inbox. Then, you need to fill in your login credentials to sign in.
3. The User Interface will open. All the features corresponding user displayed.
4. Now users can perform their task.

4. Other Non-Functional Requirements

1. **Usability Requirement:** The system shall allow the users to access the system from the phone or laptops. The system uses a web application as an interface. Since all users are familiar with the general usage of web app, no special training is required. The system is user friendly which makes the system easy.
2. **Availability Requirement:** The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

3. **Efficiency Requirement:** Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.
4. **Accuracy:** The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.
5. **Performance Requirement:** The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submitted. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.
6. **Reliability Requirement:** The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.

PRACTICAL NO. 3

Aim: To prepare a Preliminary and Detailed Use Case Diagram.

Theory:

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

The purpose of the use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

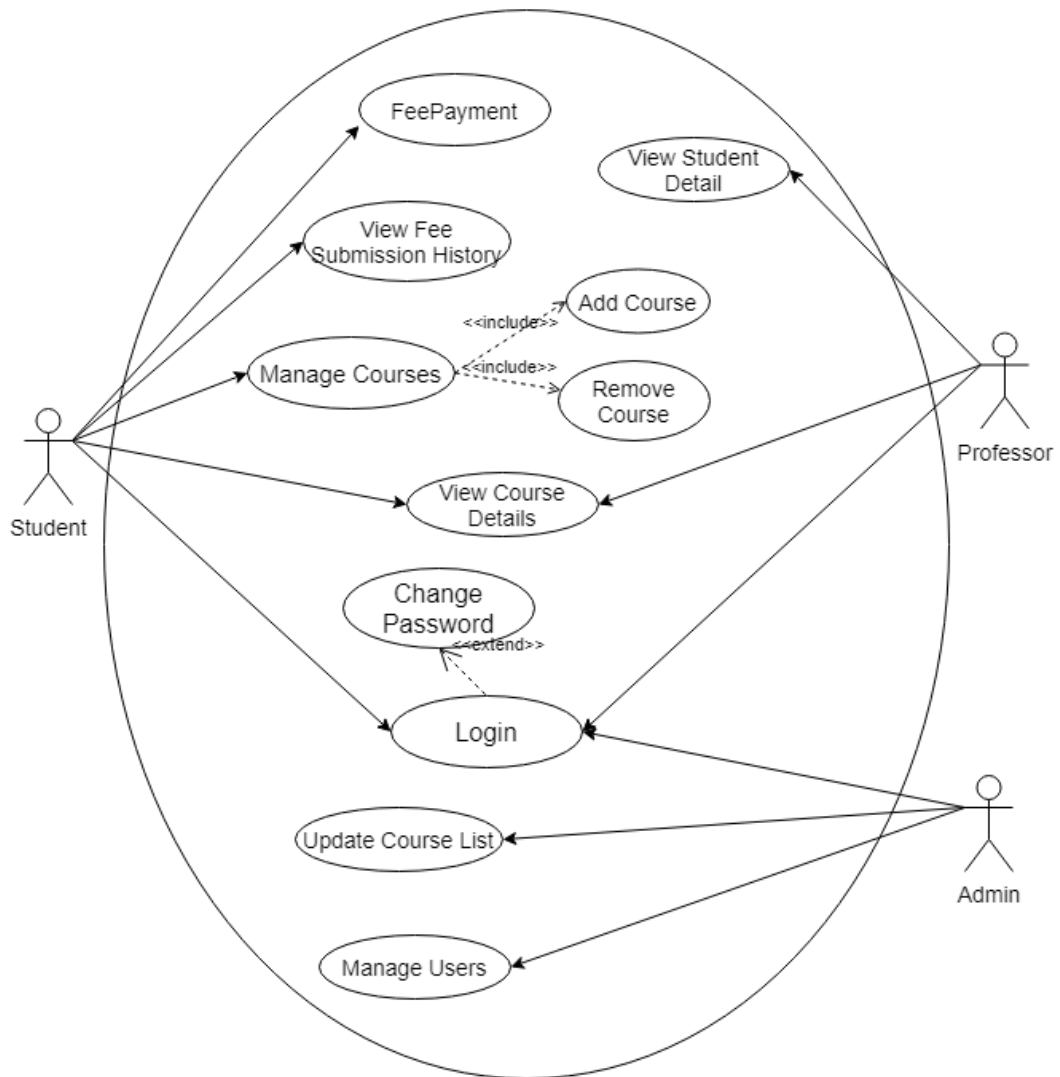
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements of the actors.

Usecase Diagram:



PRACTICAL NO. 4

Aim: To prepare a Class Diagram for the Student Registration System.

Theory:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

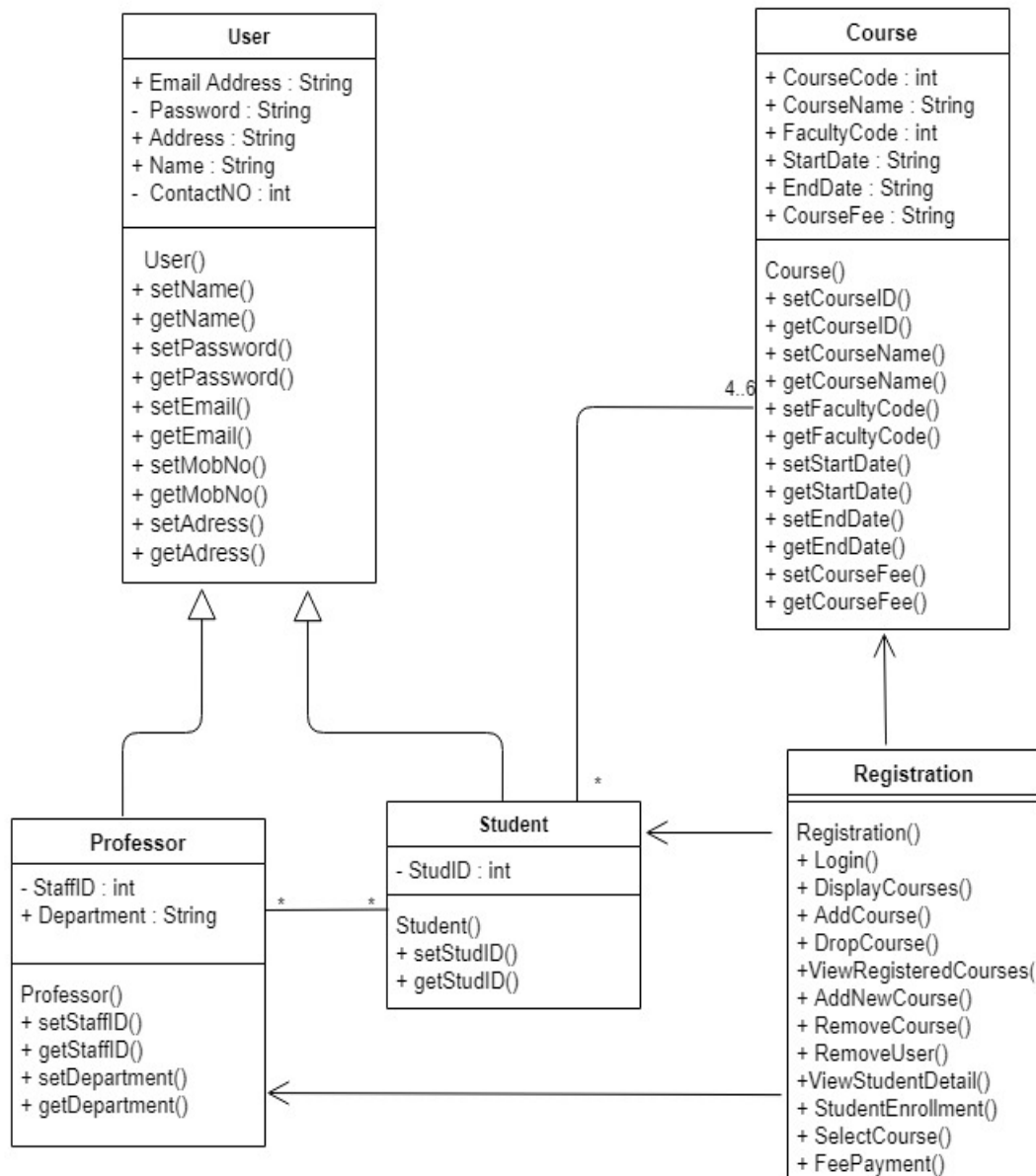
Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagrams is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Class Diagram:



PRACTICAL NO. 5

Aim: To prepare an Activity Diagram for Student Registration System.

Theory:

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modelling how a collection of use cases coordinates to represent business workflows.

- Identify candidate use cases, through the examination of business workflows
- Identify pre- and post-conditions (the context) for use cases
- Model workflows between/within use cases
- Model complex workflows in operations on objects
- Model in detail complex activities in a high level activity Diagram

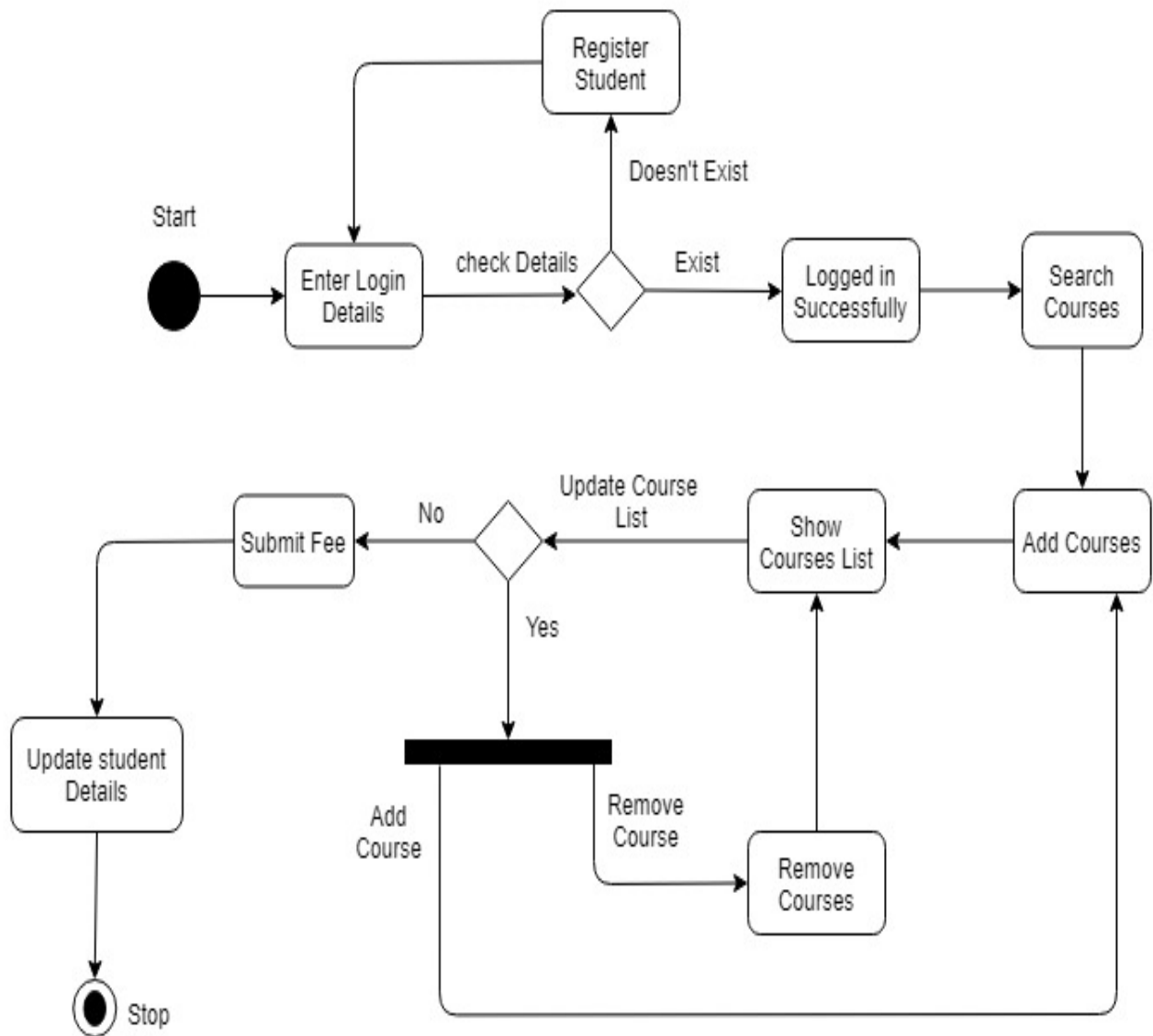
The basic purpose of activity diagrams is similar to the other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Activity Diagram:



PRACTICAL NO. 6

Aim: To prepare a Sequence Diagram for the Student Registration System.

Theory:

Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

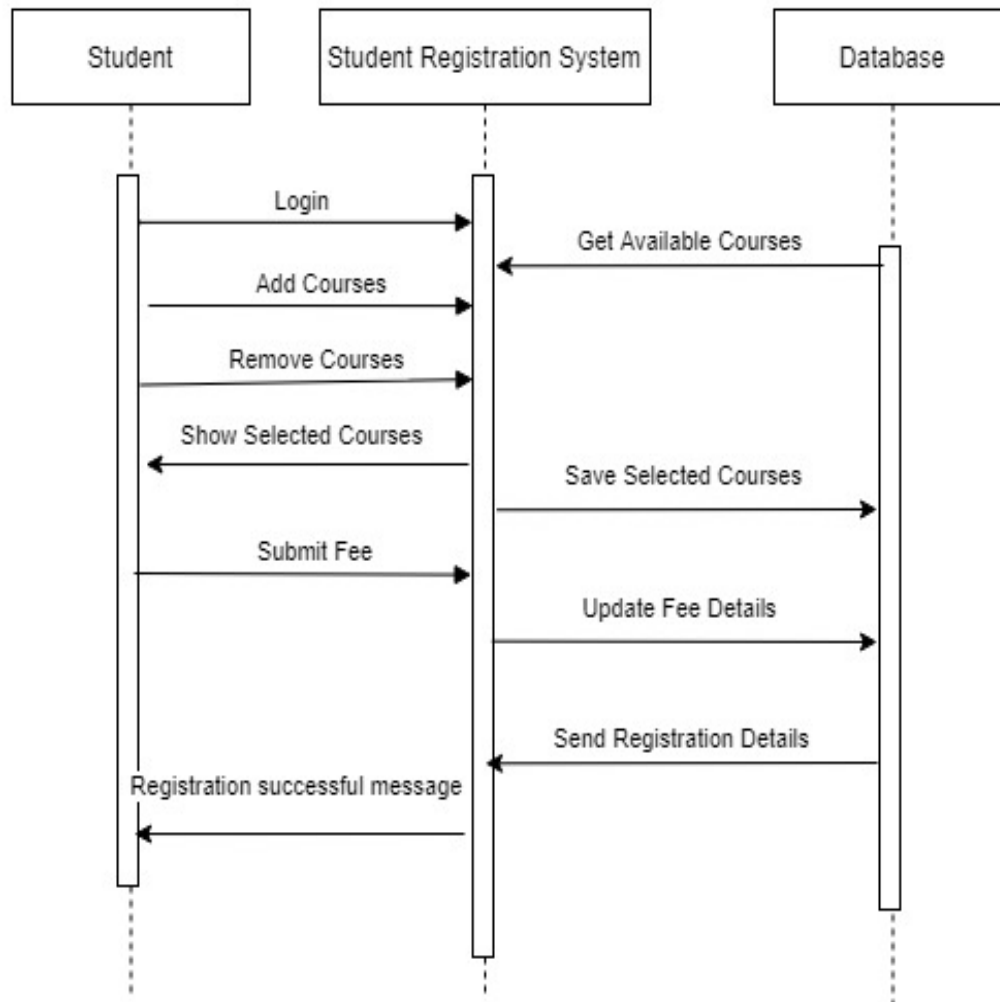
The purpose of Sequence diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle.

The purpose of interaction diagram is –

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

Sequence Diagram:



PRACTICAL NO. 7

Aim: To prepare a Collaboration Diagram for the Student Registration System.

Theory:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

Notations of a collaboration diagram:

- **Objects-** Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
- **Actors-** Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
- **Links-** Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.
- **Messages-** Messages between objects are shown as a labelled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

Collaboration Diagram:

