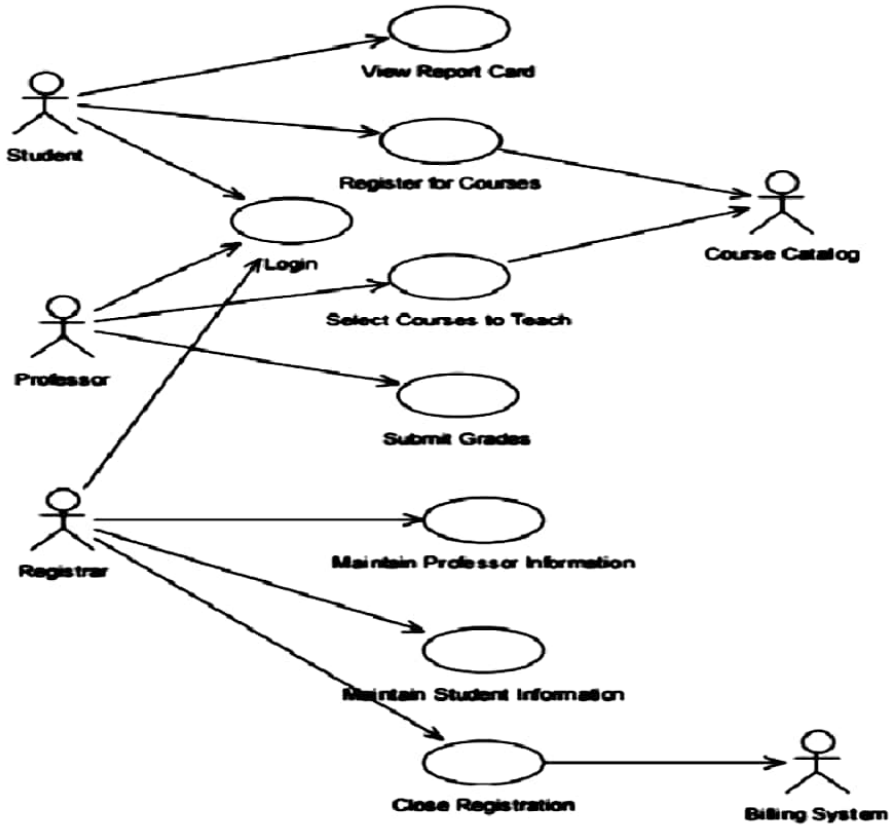
- **Front-End: HTML5, CSS3, JavaScript (frameworks like Angular, React, or Vue.js)**
- **web application framework like Django(Python) to streamline development and provide structure to the codebase**

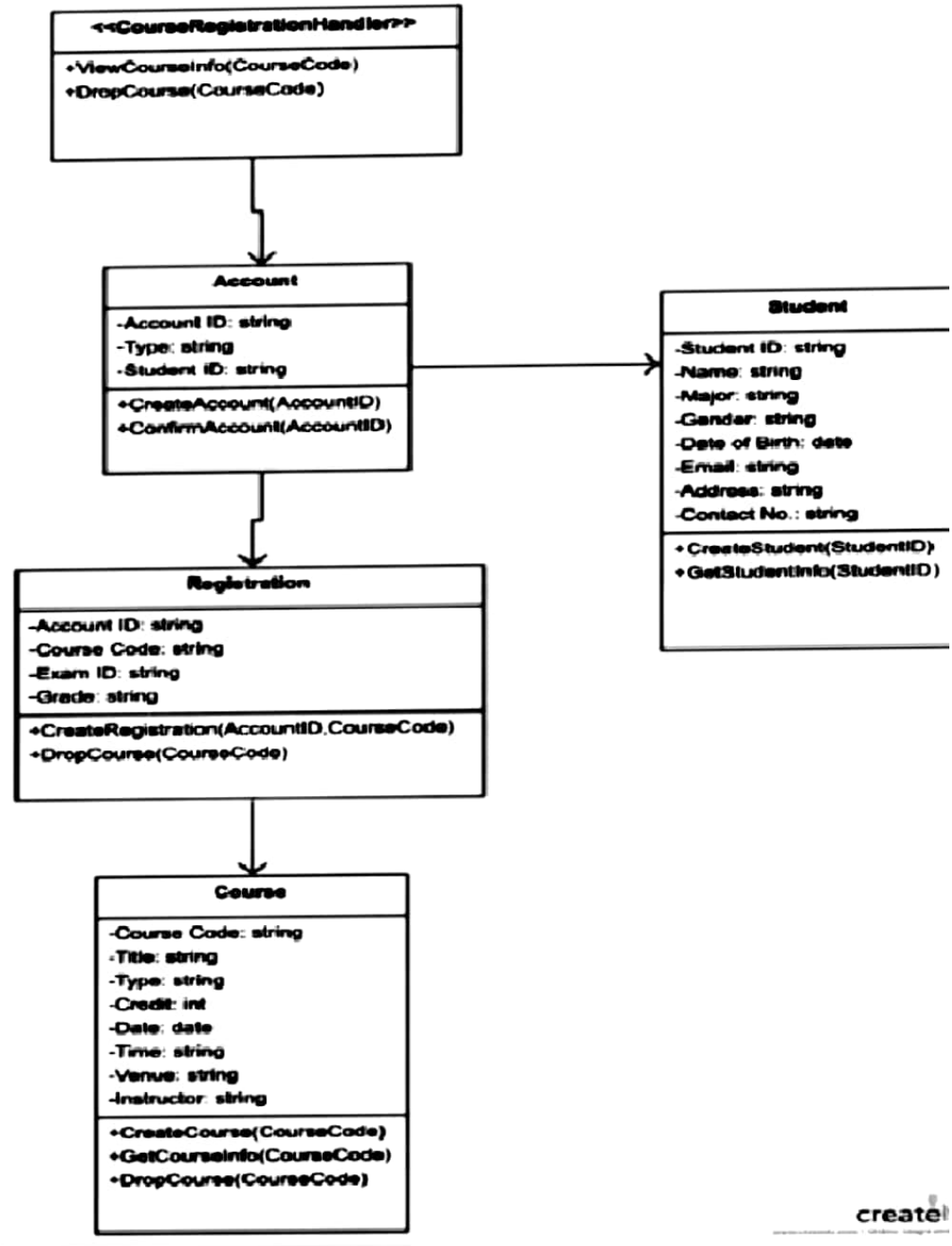
USECASE DIAGRAM

The main actors are represented at the top: "Student," "Administrator," and "Professor."
The "Course Registration System" is the central system being interacted with.
The use cases are represented below the system, listing the actions that each type of user (actor) can perform within the system.
Key actions include user authentication (login), course management (add, modify, close), course registration, monitoring, generating UML diagrams, and more



CLASS DIAGRAM:

A class diagram describes the type of objectors in the system the various kinds of static relationship that exist among them.

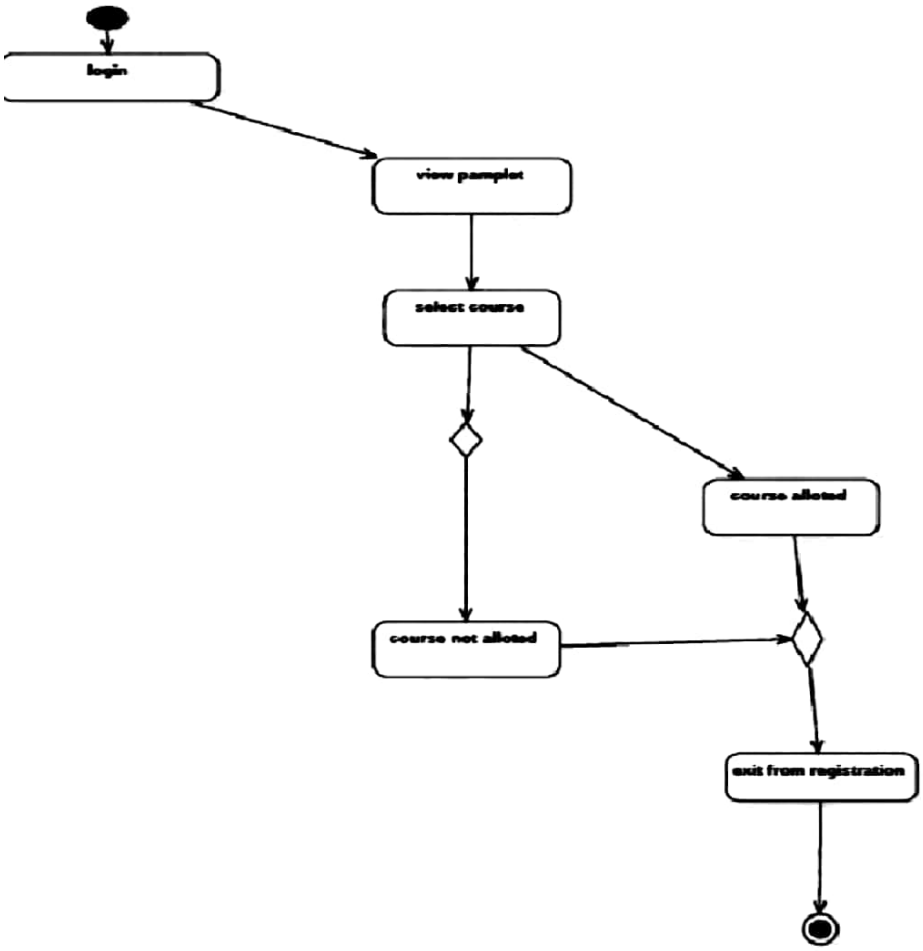


ACTIVITY DIAGRAM:

This diagram represents the graphical representatior stepwise activities and actions with support for cho concurrency. It shows the overall flow of control.

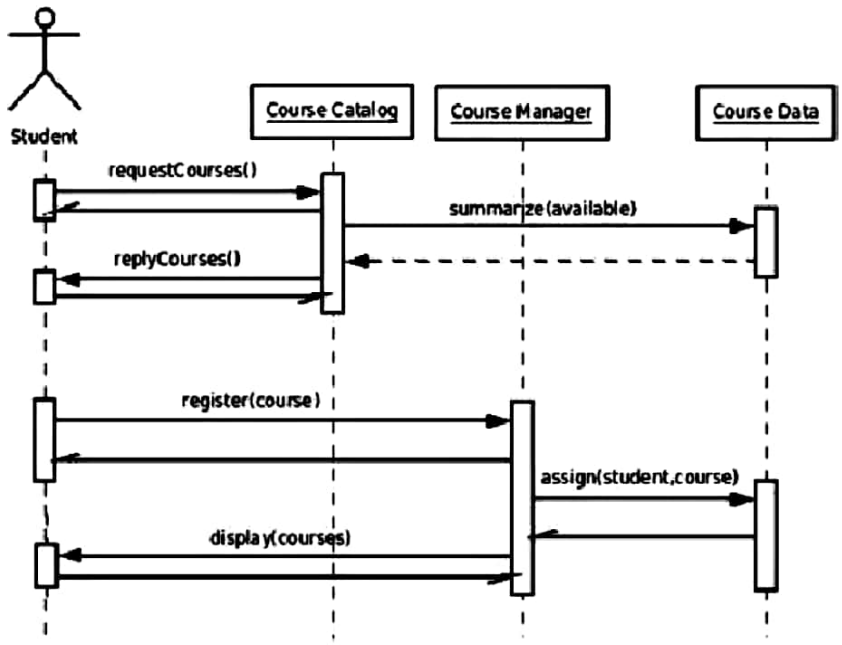
The process starts with the "Start Process" activity.
The student is required to log in to the system for authenti
After logging in, the student can search for available cours

: student can also view their course schedule.
: process ends with the "End Process" activity



'ERACTION DIAGRAM

: student starts by logging into the system.
: system authenticates the student's credentials.
: student searches for available courses, which results in a list of courses.
: student can view the details of a specific course.
: student registers for a course, drops a course, or modifies their course istration, with each action resulting in a confirmation.
: student can also view their course schedule, which provides informatio out the courses they are registered for

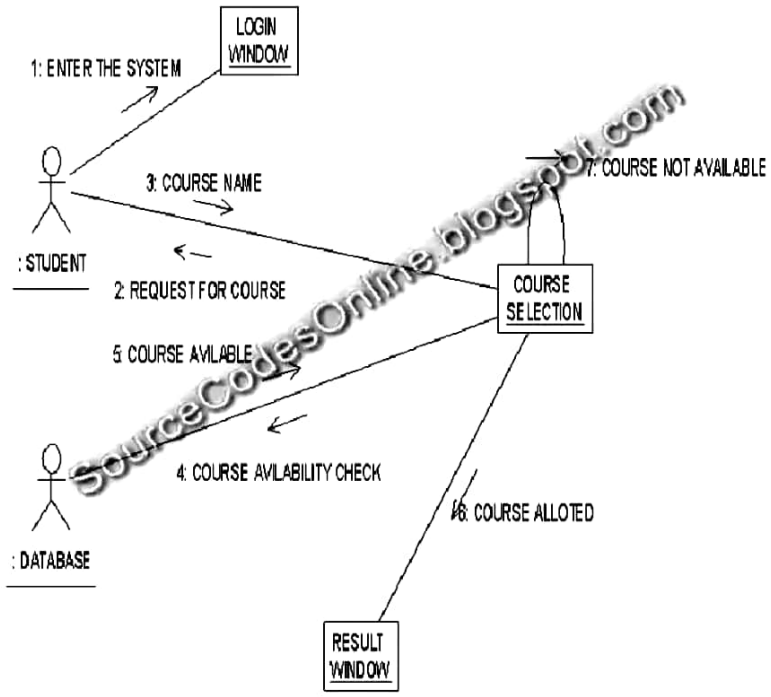


COMMUNICATION DIAGRAM

- **The "Student" and "Course Registration System" are represen**
- **The arrows between the objects indicate the flow of communi interactions.**
- **Messages exchanged between objects are labeled with the co actions, such as "Login," "Authentication," "Search for Course**

D

Ad



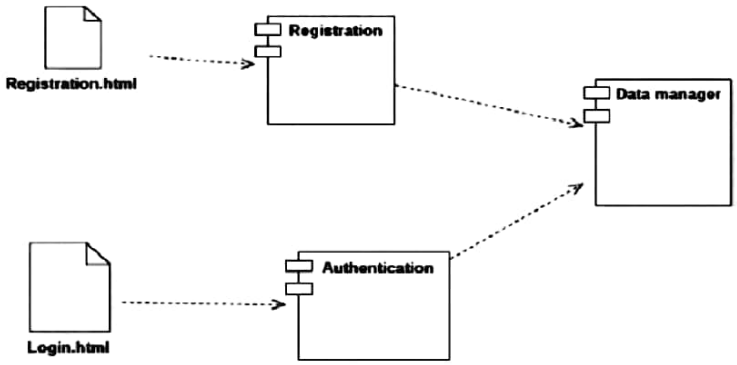
COMPONENT DIAGRAM:

Component diagrams are used to visualize the organization and relationship among components in a system.

The "Course Registration System" depends on the "Web Server" and "Application Server" to function.
The "Application Server" interacts with both the "Database Server" for data storage and "UML Tools" for UML diagram generation.

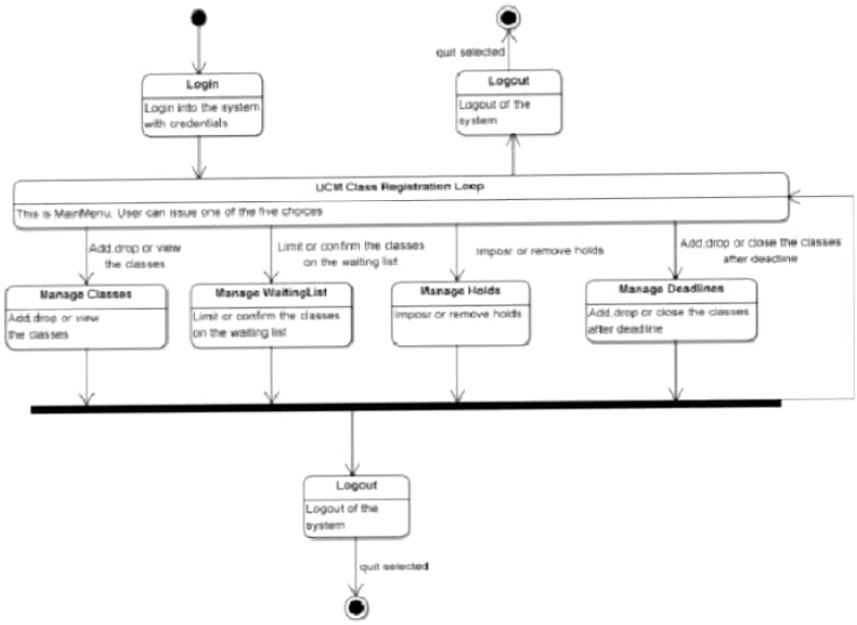
Component diagrams are used to visualize the organization and relationship among components in a system.

The "Course Registration System" depends on the "Web Server" and "Application Server" to function.
The "Application Server" interacts with both the "Database Server" for data storage and "UML Tools" for UML diagram generation.



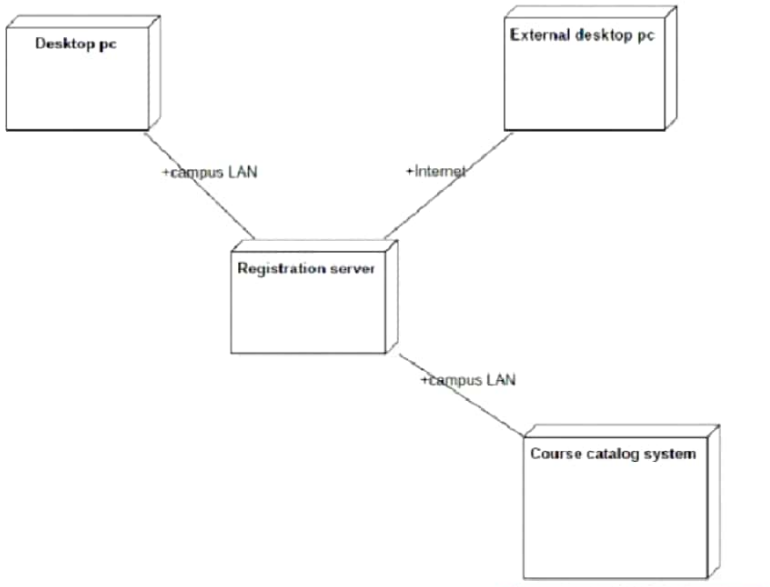
STATECHART DIAGRAM:

- **The "Course" entity has four states: "Open Enrollment," "Closed Enrollment," "In Progress," and "Completed."**
- **Transitions between states occur based on specific eve example:**
  - **A course starts in the "Open Enrollment" state when it's available for registration.**
  - **When the enrollment period ends, it transitions to the "Closed Enrollment" state.**
  - **When the course is actively being taught, it enters the "In Progress" state.**
  - **Finally, when the course is completed, it moves to the "Completed"**



PACKAGE DIAGRAM

- The "Course Registration System" is represented as the top-level package.
- Sub-packages are used to group related functionalities:
  - "Users Management" encompasses classes related to user authentication and management.
  - "Course Management" includes classes related to course creation, modification, and viewing.
  - "Registration Management" represents classes responsible for handling student registrations and course enrollment.
  - "Schedule Management" includes classes for managing course schedules and time slots.
  - "UML Tools Integration" represents the package r integrating UML diagram generation tools



RESULT:

Thus the Course Registration System application was successfully designed and the output was verified.