# Supporting Document 1: Design Methodologies & Diagrams

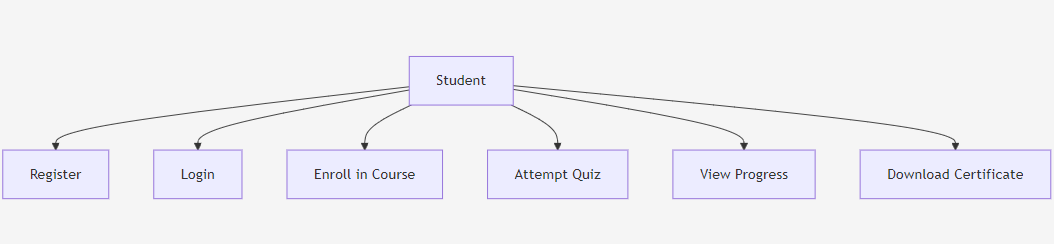
## 1. Design Methodologies

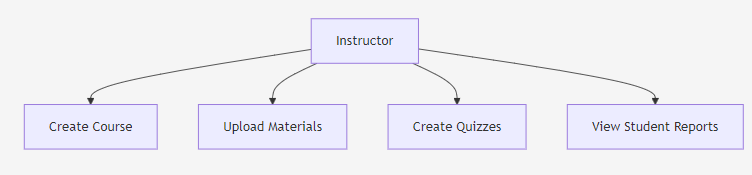
* UML (Unified Modeling Language): Used to model system structure and user interactions clearly. (added in the worksheet)
* Database Design : Ensures data consistency and eliminates redundancy in tables like users, courses, and quizzes.

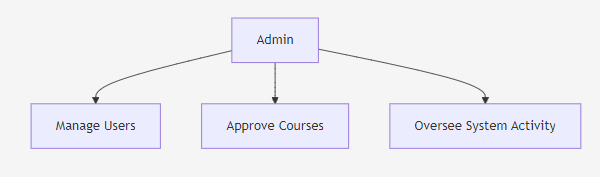
## 2. Diagrams

### Use Case Diagram – E-Learning Platform

Actors: Student, Instructor, Admin







Main Use Cases:

* Student: Register, Login, Enroll in Course, Attempt Quiz, View Progress, Download Certificate
* Instructor: Create Course, Upload Materials, Create Quizzes, View Student Reports
* Admin: Manage Users, Approve Courses, Oversee System Activity

### Database Design (ER Diagram)

### 

* **User** *(user\_id, name, email, password, role)*
* **Course** *(course\_id, title, description, Instructor\_id)*
* **Quiz** *(quiz\_id, course\_id, title)*
* **Question** *(question\_id, quiz\_id)*
* **Certificate** *(certificate\_id, student\_id, course\_id, issue\_date)*

**Relationships:**

* A Student can enroll in many Courses (1–Many)
* An Instructor can create many Courses (1–Many)
* An Admin can approve/deny many Courses (1–Many)
* A Course can have many Quizzes (1–Many)
* A Quiz has many Questions (1–Many)
* A Student can earn many Certificates (1–Many)

# Supporting Document 2: Types of Testing Techniques

## Unit Testing

Tests small, independent modules like quiz grading or login validation.

## Integration Testing

Ensures combined modules (e.g., course enrollment + quiz system) work correctly together.

## System Testing

Validates that the complete platform meets all functional and non-functional requirements.

## User Acceptance Testing (UAT)

Confirms that the platform performs as expected for actual users before launch.

## Performance Testing

Checks system responsiveness and stability when accessed by multiple users simultaneously.