## **1. Use Case Diagram – AI-LMS Overview**

**Legend:**

* **Actors:**
  + **User:** Regular learner who enrolls in courses, takes assessments, views recommendations, and earns certificates.
  + **Admin:** Manages learning paths, courses, modules, assessments, and users.
* **Use Cases:** Actions performed by each actor, such as “Enroll in Course” or “Create Assessment.”
* **System Boundary:** Box around all use cases represents the LMS itself.

**Explanation:** This diagram maps the interaction between users and the LMS. It visually separates admin capabilities from user capabilities, making it clear that admins manage the content and users consume it. The bot is embedded only during course sessions to give contextual tutoring.

## **2. Class Diagram – Core Entities**

**Legend:**

* **Classes:** Represent data models (e.g., User, Course, Module).
* **Attributes:** Key properties for each entity (minimal GDPR-friendly set).
* **Methods:** Key operations handled by that entity.
* **Relationships:**
  + **Solid line:** Association (e.g., a Course contains many Modules).
  + **Diamond:** Composition (e.g., a LearningPath contains Courses).

**Explanation:** This diagram shows the system’s data structure, including relationships between learning paths, courses, modules, and assessments. It reflects rules such as “A course can belong to multiple learning paths” and “A module belongs to exactly one course.” Assessment questions link directly to modules for targeted recommendations.

## **3. ERD (Entity-Relationship Diagram)**

**Legend:**

* **Entities:** Represent database tables.
* **Primary Key (PK):** Unique identifier for each record.
* **Foreign Key (FK):** Links between tables.
* **Crow’s Foot Notation:** Shows relationship cardinality (1-to-many, many-to-many).

**Explanation:** The ERD is a database-focused view of the class diagram. It models how entities store and relate data, ensuring database compatibility across PostgreSQL and MySQL. It includes join tables for many-to-many relations (e.g., course\_learning\_path).

## **4. Activity Diagrams**

### **4.1 User Takes Course & Assessment**

**Legend:**

* **Rounded rectangles:** Actions performed by the user or system.
* **Diamonds:** Decision points (branching).
* **Repeat Loops:** Represented using repeat while.
* **Start/Stop:** Begin and end of the process.

**Explanation:** This diagram walks through the full learning experience - from login to course completion. It also shows how recommendations are generated only for failed questions in an assessment.

### **4.2 Admin Creates Course & Assigns Content**

**Legend:**

* Same as above.
* Special focus on repeated actions for adding modules and questions.

**Explanation:** Maps the content creation workflow. It shows the admin flow from creating learning paths to publishing courses, maintaining strict module ordering, and linking assessment questions to recommended modules.

### **4.3 User Registration & Login with OTP**

**Legend:**

* Split into two partitions: Registration and Login.
* OTP flow clearly marked with decision checks.

**Explanation:** This diagram covers authentication using JWT and email OTP. It ensures that even in your simplified setup, security is explicitly represented in the report.

## **5. System Architecture Diagram**

**Legend:**

* **Rectangles:** System components or layers.
* **Arrows:** Communication paths between components.
* **Notes:** Special clarifications on behavior or constraints.

**Explanation:** Illustrates the high-level system layout - from user’s browser (React) to backend (Express) to database (Postgres/MySQL), file storage, and external APIs (Botpress, SMTP). It emphasizes that Botpress is only active during course sessions and that the system is deployed on a single shared hosting server.