**1. Use Case Diagram and Descriptions**

The system is called **Training Management System**. It allows different users (Finance Manager, Trainer, and Trainee) to manage and participate in training activities.

**Must-Have Use Cases (Core Diagram):**

* **Create Training Modules** – Trainer creates training content.
* **Enroll in Modules** – Trainee enrolls in a training course.
* **Upload and Access Resources** – Trainers upload resources, and trainees access them.
* **Track Progress** – Trainer monitors trainee progress.
* **Generate Certificates** – Finance Manager issues certificates after module completion.
* **Participate in Discussion Forum** – Trainers and trainees communicate.

**Actors and Descriptions:**

* **Trainer**: Creates modules, tracks progress, uploads content, joins discussions.
* **Trainee**: Enrolls in modules, accesses resources, joins discussions.
* **Finance Manager**: Manages users and generates certificates.

Each actor is connected to their relevant use cases. All must-have functions are shown in the main use case diagram.

**2. ERD and Explanation**

The ERD diagram shows all entities, attributes, and relationships needed to support the system’s use cases. Below is a summary of each entity:

**Entities:**

* **User**
  + Attributes: user\_id (PK), name, email, role (FK)
  + Description: Stores all users of the system.
* **Role**
  + Attributes: role\_id (PK), role\_name
  + Description: Defines user types (Trainer, Trainee, Finance Manager).
* **TrainingModule**
  + Attributes: module\_id (PK), title, description, created\_by (FK)
  + Description: Stores each training module’s details.
* **Enrollment**
  + Attributes: enrollment\_id (PK), user\_id (FK), module\_id (FK), enrollment\_date
  + Description: Tracks who enrolled in what module.
* **Progress**
  + Attributes: progress\_id (PK), user\_id (FK), module\_id (FK), completion\_status, last\_updated
  + Description: Tracks each trainee’s progress.
* **Resource**
  + Attributes: resource\_id (PK), module\_id (FK), title, file\_url
  + Description: Files uploaded to each module.
* **Certificate**
  + Attributes: certificate\_id (PK), user\_id (FK), module\_id (FK), issue\_date
  + Description: Issued after completing training.
* **Discussion**
  + Attributes: discussion\_id (PK), module\_id (FK), user\_id (FK), message, timestamp
  + Description: Stores messages posted in module forums.

**Relationship Summary:**

* One **user** has one **role**.
* One **trainer** creates many modules.
* **Trainees** enroll in many modules (via the Enrollment table).
* Resources, progress, certificates, and discussions are linked to modules and users.

This database structure fully supports all use cases described above.

**3. Test Coverage**

Each use case in the system has a matching test file written using .feature format (Cucumber). This ensures the system works as expected for:

* Creating modules
* Enrolling in modules
* Uploading and accessing resources
* Tracking progress
* Generating certificates
* Participating in discussions

Each .feature file includes scenarios to test the core flow of the use case.