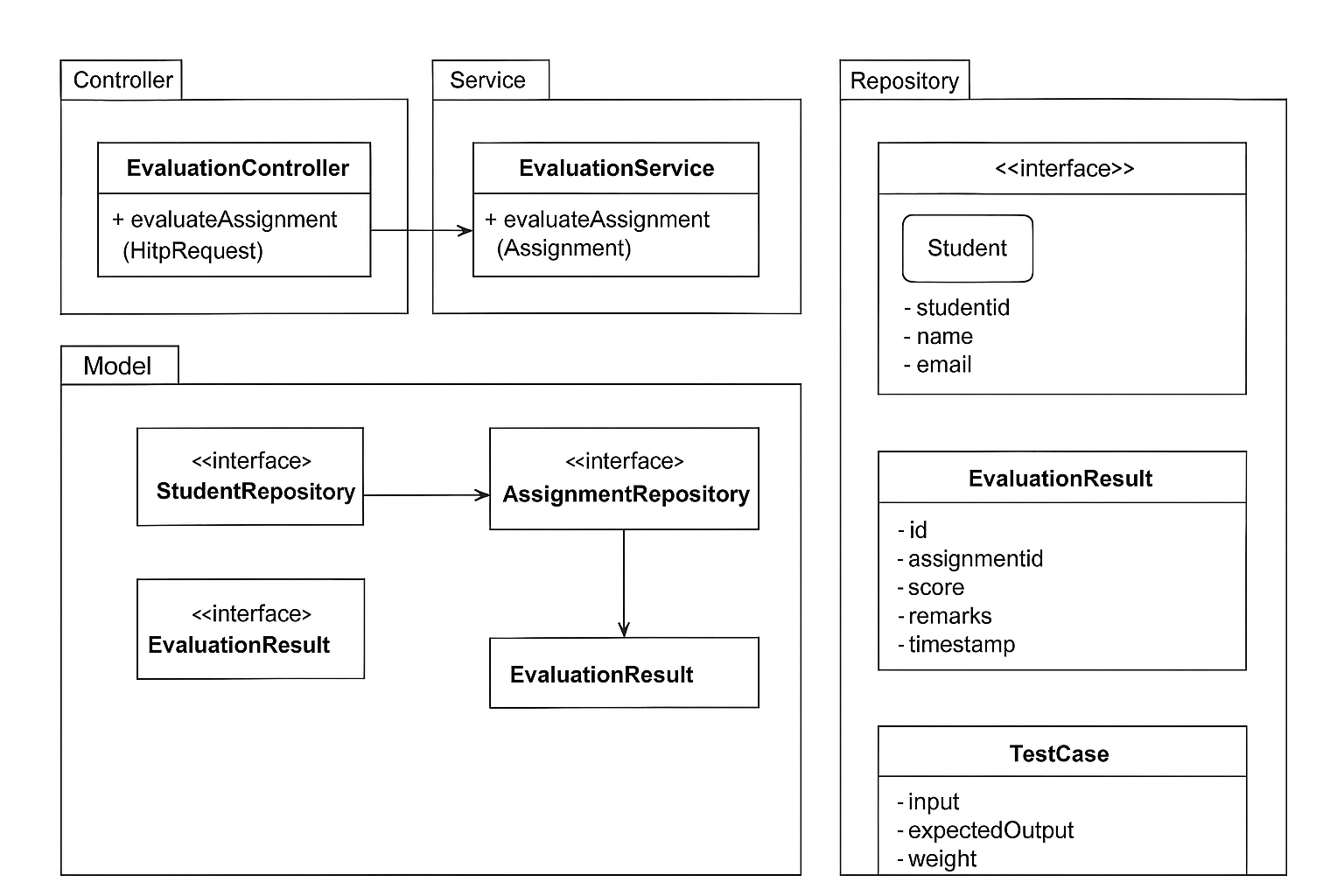
# UML Class Diagram



1**. Controller Layer**

* **EvaluationController**
  + Method: evaluateAssignment (HttpRequest)
  + Entry point for REST API calls.
  + Delegates requests to the EvaluationService.

2**. Service Layer**

* **EvaluationService**
  + Method: evaluateAssignment(Assignment)
  + Handles business logic and orchestrates evaluation.
  + Calls the EvaluationEngine (not explicitly shown here, but part of service logic).
  + May also use LoggingService to log errors.

**3. Repository Layer**

These are interfaces extending Spring Data JPA, used to talk to the database (MySQL):

* StudentRepository
* AssignmentRepository
* EvaluationResultRepository

They provide persistence (CRUD operations) for the corresponding model classes.

4**. Model Layer**

* **Student**
  + Attributes: studentId, name, email
  + Represents the learner.
* **Assignment**
  + Attributes: assignmentId, studentId, title, submissionFilePath, submissionTime, deadline
  + Tracks submissions.
* **EvaluationResult**
  + Attributes: id, assignmentId, score, remarks, timestamp
  + Stores outcome of an evaluation.
* **TestCase**
  + Attributes: input, expectedOutput, weight
  + Defines the criteria for auto-evaluation.

**Relationships**

* Controller → Service: Controller calls service methods.
* Service → Repositories: Service interacts with repositories to store/retrieve data.
* Repositories ↔ Model: Each repository manages one or more model classes.
* EvaluationEngine (inside service) uses TestCase and produces EvaluationResult.