Day1: Requirements

**E-Learning Platform with Auto Evaluation**

### **1. Project Objective**

The goal is to create an E-Learning Platform where students can upload assignments and receive automatic evaluation. The platform will use Java for backend logic and AWS DynamoDB for storing scores and logs. The final project will implement modern development practices including testing, CI/CD, and behavior-driven development.

### **2. Functional Requirements**

* Students should be able to upload assignment files.
* The system should process the uploaded files and run predefined test cases.
* The system should calculate and store evaluation scores.
* The system should keep logs of evaluation details and errors.
* The platform should be scalable, using DynamoDB to store structured and unstructured data.

### **3. Non-Functional Requirements**

* The system should be secure and handle file uploads safely.
* Data should be consistent and reliable across sessions.
* The platform should be easily extendable and maintainable.
* Automated testing and deployment pipelines should be used.

### **4. Actors**

* **Student** – uploads assignments and views evaluation results.
* **System** – processes uploads, evaluates assignments, stores results and logs.

### **5. Use Cases**

1. **Student uploads assignment**
   1. Input: Assignment file, student ID
   2. Output: Confirmation of upload
2. **System evaluates assignment**
   1. Input: Uploaded assignment
   2. Output: Evaluation score, feedback
3. **System stores results**
   1. Input: Evaluation data
   2. Output: Stored score in DynamoDB
4. **System logs activities**
   1. Input: Logs from evaluation process
   2. Output: Stored logs in DynamoDB

6. **UML Diagram**

* **Class Diagram**

@startuml

class Student {

- studentId : String

- name : String

- email : String

}

class Assignment {

- assignmentId : String

- title : String

- description : String

- submissionDate : Date

}

class EvaluationEngine {

+ runTestCases() : int

+ calculateScore() : int

+ generateReport() : String

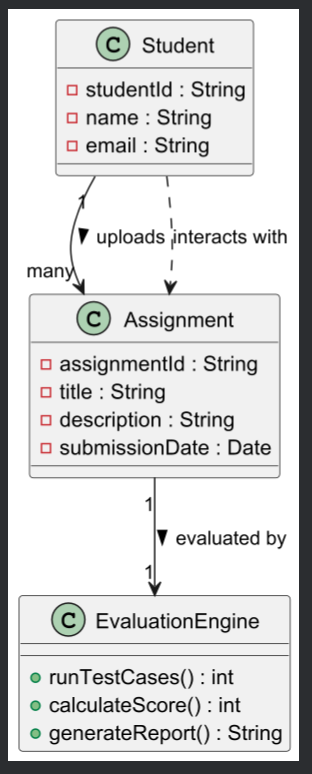
}

Student "1" --> "many" Assignment : uploads >

Assignment "1" --> "1" EvaluationEngine : evaluated by >

Student ..> Assignment : interacts with

@enduml

****

* **Use Case Diagram**

@startuml

left to right direction

actor Student

actor System

Student --> (Upload Assignment)

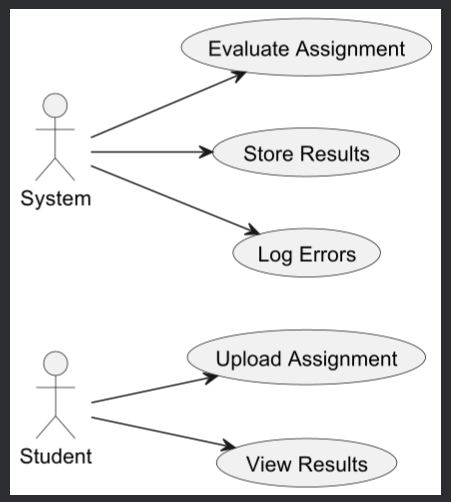
Student --> (View Results)

System --> (Evaluate Assignment)

System --> (Store Results)

System --> (Log Errors)

@enduml



7. **Initial Java Code Templates**

**Student.java:**

package org.example;

public class Student {

private String studentId;

private String name;

private String email;

public Student(String studentId, String name, String email) {

this.studentId = studentId;

this.name = name;

this.email = email;

}

public String getStudentId() {

return studentId;

}

public String getName() {

return name;

}

public String getEmail() {

return email;

}

public void setStudentId(String studentId) {

this.studentId = studentId;

}

public void setName(String name) {

this.name = name;

}

public void setEmail(String email) {

this.email = email;

}

@Override

public String toString() {

return "Student{" +

"studentId='" + studentId + '\'' +

", name='" + name + '\'' +

", email='" + email + '\'' +

'}';

}

}

**Assignment.java:**

package org.example;

import java.util.Date;

public class Assignment {

private String assignmentId;

private String title;

private String description;

private Date submissionDate;

public Assignment(String assignmentId, String title, String description, Date submissionDate) {

this.assignmentId = assignmentId;

this.title = title;

this.description = description;

this.submissionDate = submissionDate;

}

public String getAssignmentId() {

return assignmentId;

}

public String getTitle() {

return title;

}

public String getDescription() {

return description;

}

public Date getSubmissionDate() {

return submissionDate;

}

public void setAssignmentId(String assignmentId) {

this.assignmentId = assignmentId;

}

public void setTitle(String title) {

this.title = title;

}

public void setDescription(String description) {

this.description = description;

}

public void setSubmissionDate(Date submissionDate) {

this.submissionDate = submissionDate;

}

@Override

public String toString() {

return "Assignment{" +

"assignmentId='" + assignmentId + '\'' +

", title='" + title + '\'' +

", description='" + description + '\'' +

", submissionDate=" + submissionDate +

'}';

}

}

**EvaluationEngine.java:**

package org.example;

public class EvaluationEngine {

public int runTestCases(Assignment assignment) {

System.out.println("Running test cases for assignment: " + assignment.getTitle());

// Simulated result

return 80;

}

public int calculateScore(int testResult) {

System.out.println("Calculating score based on test result.");

return testResult;

}

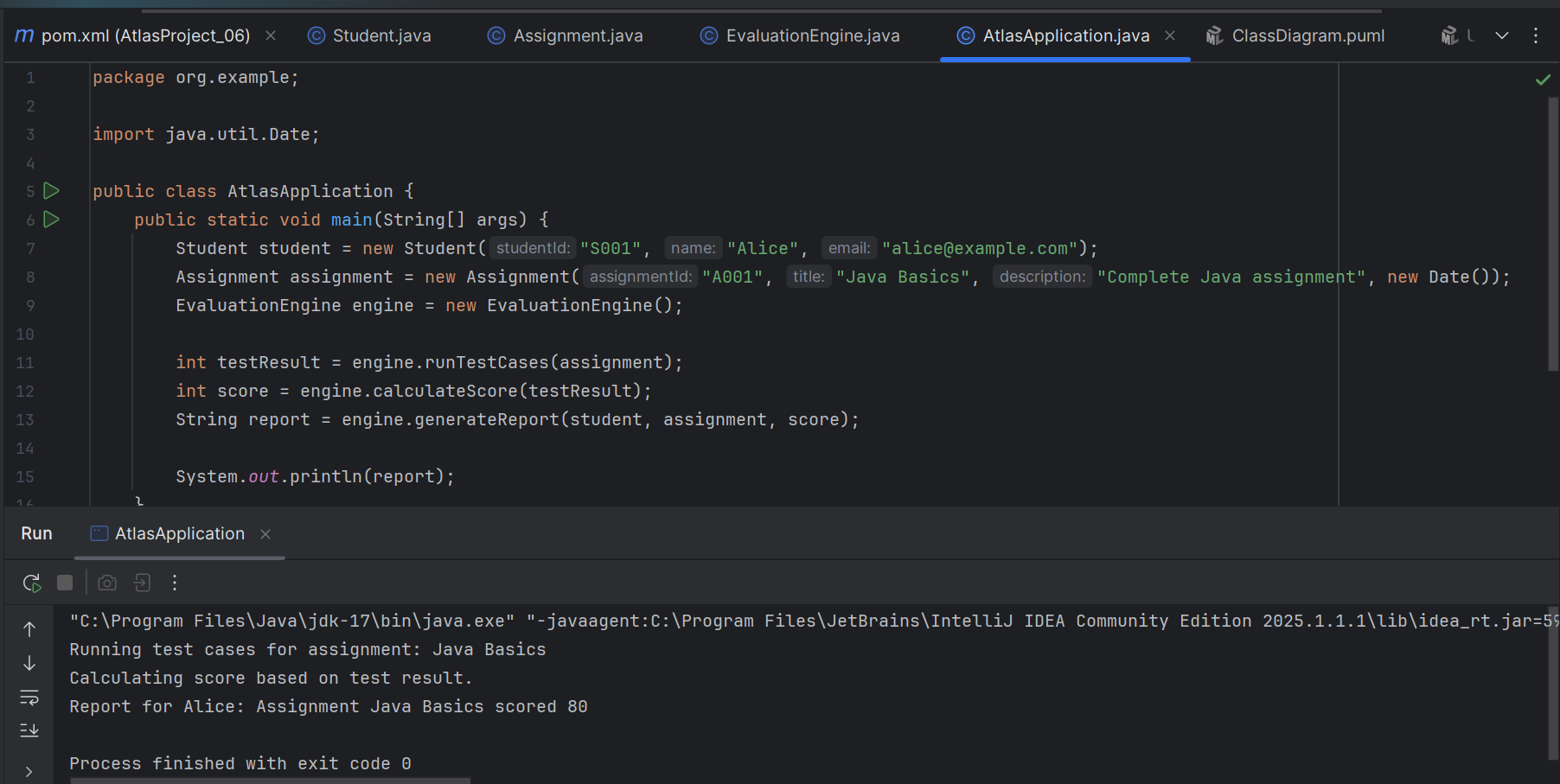
public String generateReport(Student student, Assignment assignment, int score) {

return "Report for " + student.getName() + ": Assignment " + assignment.getTitle() + " scored " + score;

}

}

**AtlasApplication.java:**



**Day 1 – Conclusion:**

The project setup for the E-Learning Platform with Auto Evaluation has been completed successfully. The folder structure, package organization, UML diagrams, and initial Java templates have been created and tested in IntelliJ IDEA. The main application AtlasApplication.java executed as expected, demonstrating interaction between the Student, Assignment, and EvaluationEngine classes. The project is now ready for version control and further development in the upcoming phases.