

Lab 10

Design question 1

During this DE course we used Sakai as a Learning Management System (LMS).

Suppose you have to design a framework for a LMS with the following requirements:

Requirements for the LMS framework:

- The LMS framework supports multiple courses.
- A course can be taken by multiple students
- The professor should be able to add assignments to a course. An assignment has a title, openDate, dueDate, instructions and can contain multiple attachments.
- Students can see the assignments, and submit their solution for every assignment. The system keeps track of the submission date. Students can submit multiple attachments to their solution.
- The professor can grade the solutions from the students. The LMS framework should support different grading options like letter grade, point grade or pass/fail.
- When the professor sets the grade for a solution, then the student will receive an email about this. An attachment is stored on the local file system.

Requirements for the application using the LMS framework:

- The application using this LMS framework (called MIU LMS) wants the possibility that an assignment contains multiple sub-assignments. A sub-assignment has the same openDate and dueDate as its parent assignment, but has its own title and instruction. A sub-assignment can also have multiple attachments.
- When the professor sets the grade for a solution, then the student will receive either an email, a WhatsApp message or a SMS messages about this, depending on the preferences of the student.
- It should be easy for LMS applications using this framework to add their own grading option, for example a checkmark.

Draw the class diagram of your design. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework (based on the requirements for the framework, and the framework best practices we studied in this course) . You only need to show the domain model of the point awards framework and application, you do not need to worry about GUI classes, service classes, database classes, etc.