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.

Design question 2

Suppose you need to design a framework for a simple car racing game with the following requirements:

- The user can do the following actions: go left, go right, increase speed and decrease speed.
- The user gets points for how well it stays on the road, and how fast it can race.

For this question we will only focus on the Car, and the actions on the Car. Suppose we have the following **partial** class diagram of the framework:
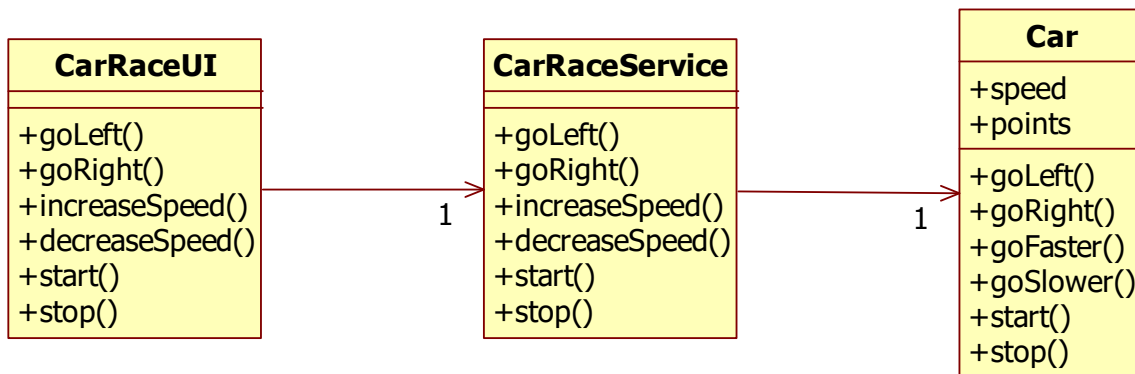
The sequence diagram of a possible scenario looks like this:
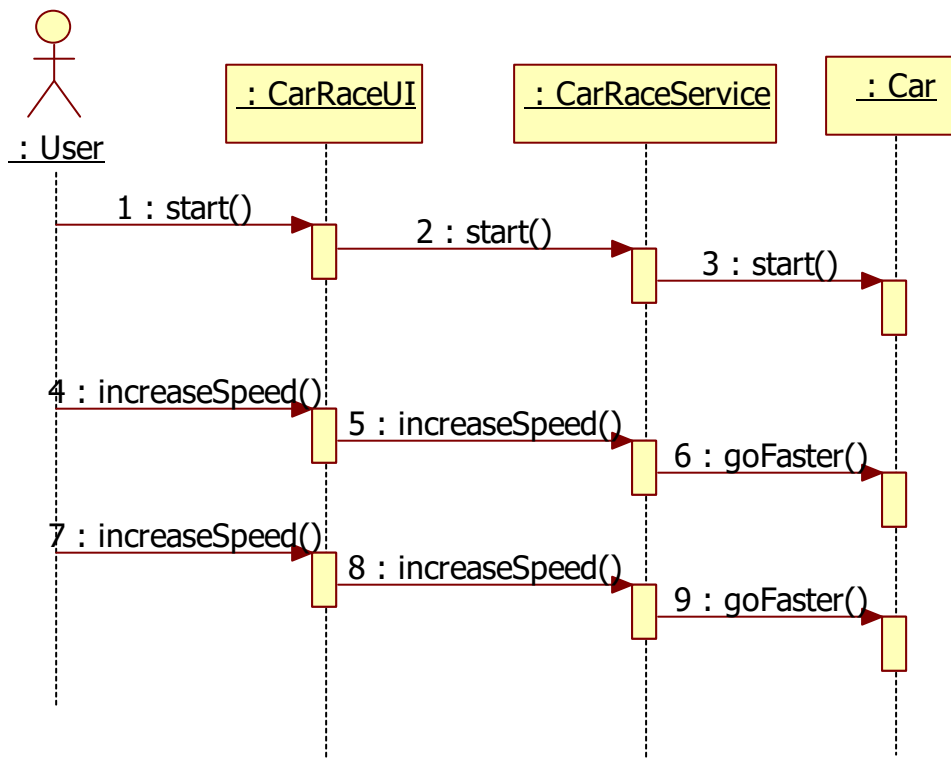
Now suppose we get a new requirements for the framework:

1. The framework should also support undo and redo functionality for the actions
    increaseSpeed() and **decreaseSpeed**(). This means the CarRaceUI class gets 2 more
    buttons: an **undo** button and a **redo** button.

2. The framework should support different car modes. The modes supported by the
    framework are:
a) Rookie Racer: This means you have less than 1000 points, and if you press the
    increaseSpeed button, your speed is increased with 1 mile per hour. If you press the
    decreaseSpeed button, your speed is decreased with 1 mile per hour

b) Racer: This means you have between 1000 and 5000 points, and if you press the
    increaseSpeed button, your speed is increased with 2 miles per hour. If you press the
    decreaseSpeed button, your speed is decreased with 2 miles per hour

c) Pro Racer: This means you have more than 5000 points, and if you press the
    increaseSpeed button, your speed is increased with 3 miles per hour. If you press the
    decreaseSpeed button, your speed is decreased with 3 miles per hour

    If the car has enough points, it will automatically be upgraded to the next node. The
    car can only be upgraded to a higher mode if the goFaster() or goSlower() method is
    called on the Car.

    It should be easy to write racing applications using the framework that support
    different car modes.

3. The framework should also support the following:
    a) When we start the game by pressing the start button on the CarRaceUI, the car
        speed and number of points is loaded from the database. When we stop the game,
        the car speed and number of points is stored in the database.
    b) In the racing game, we want to show all kind of car information (speed, points,
        current user action) on the screen in different window frames. The user can then
        choose which window frames will be shown on the screen. The framework should
        support the following window frames:
        1. A window that shows the current speed of the car
        2. A window that shows the number of points
        3. A window that shows the last user action

    It should be easy to write racing applications using the framework that support
    different window frames, for example a window that shows both the speed and the
    points next to each other.

    **Draw the class diagram of your design. In a sequence diagram, show how your
    design works**