

# **Advanced Software Development**

## **Midterm Exam April 2020**

*Solution*

## Question 1 [ 50 points ] {60 minutes}

Suppose you have to design the controller software for an automatic gate. The gate controller application has the following requirements:

We have a remote control with only one button.

If we press the button on the remote control, and the gate is open, then the gate should be closed.

If we press the button on the remote control, and the gate is closed, then the gate should be opened.

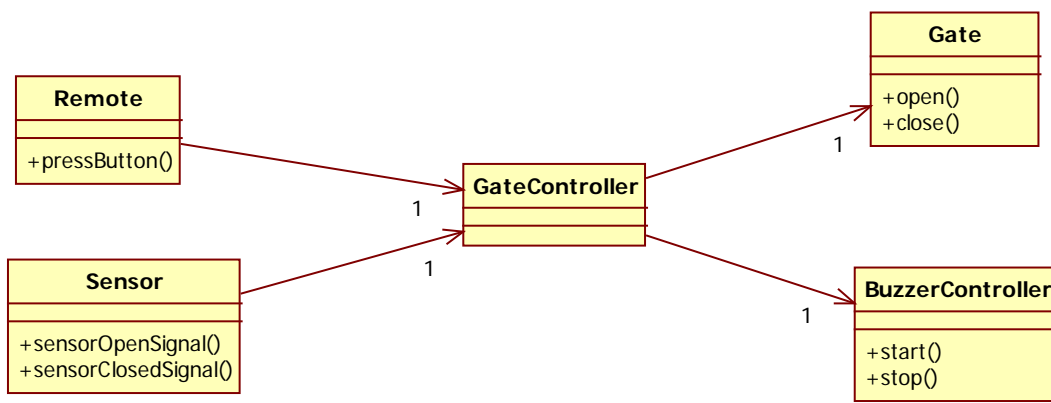
If we press the button on the remote control, and the gate is opening, then the gate should be closed.

If we press the button on the remote control, and the gate is closing, then the gate should be opened.

We also have a gate sensor that signals our application if the gate is completely closed or opened.

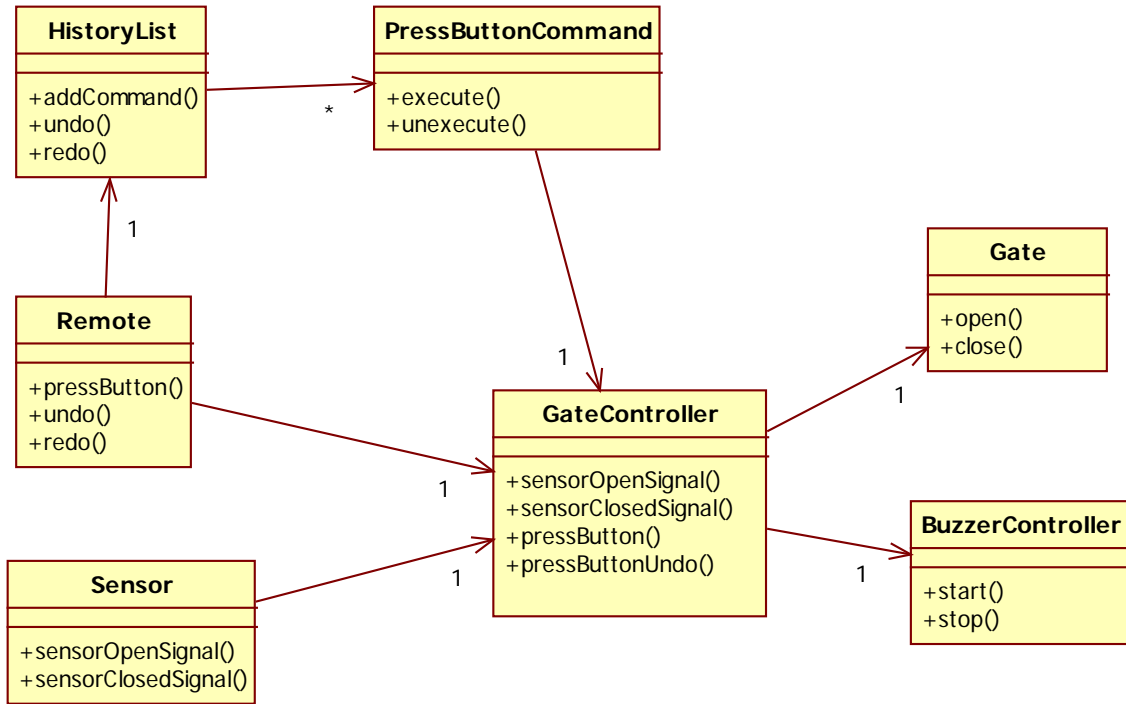
Because we want to make our automatic gate as safe as possible, we also have an audio buzzer that makes noise when the gate is opening or closing. When the gate is completely open or closed, the buzzer is idle.

Your first design might look as follows:



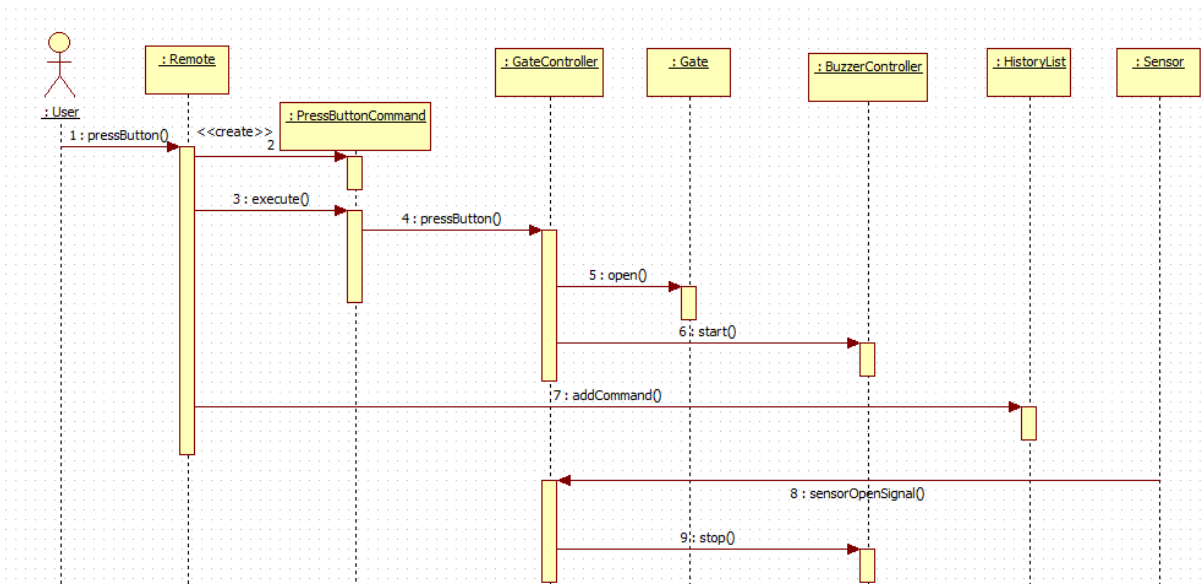
- a. [10 points] Now you receive a new requirement from marketing, that the remote control now gets 3 buttons. One button to open and close the door, and an undo and a redo button.

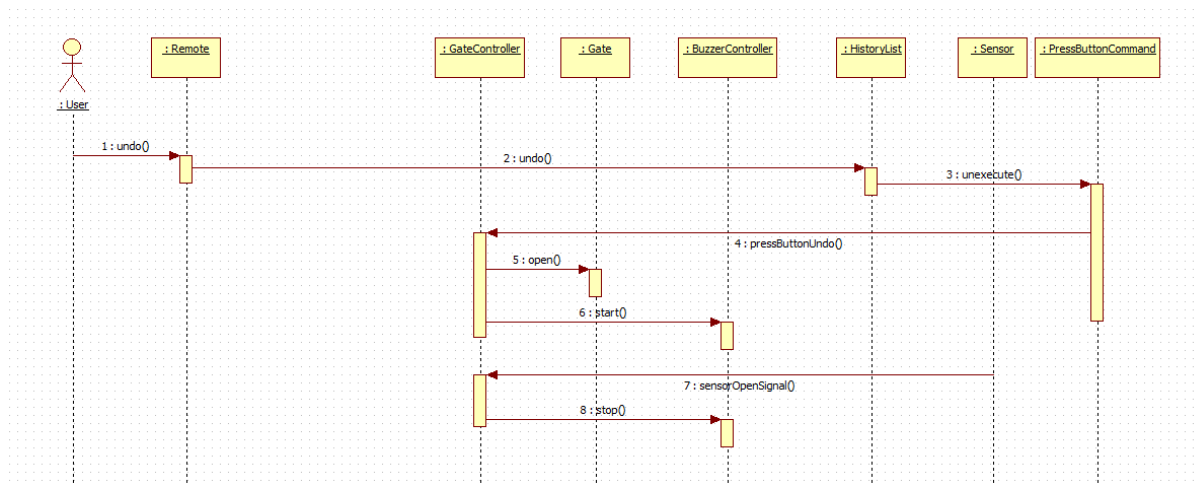
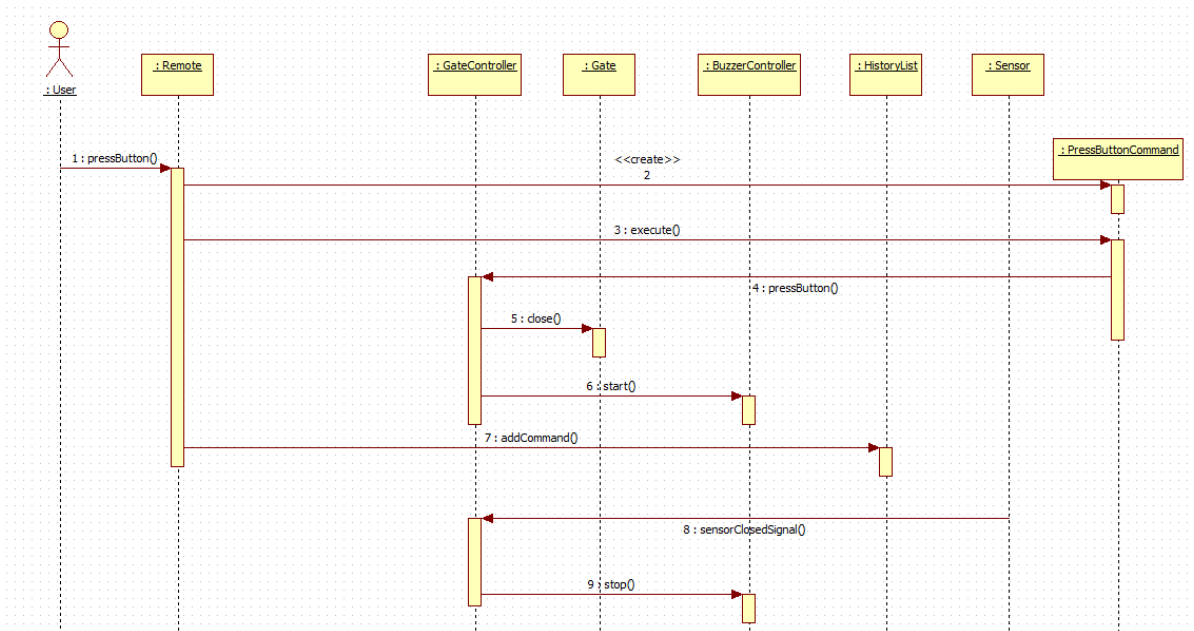
Draw the class diagram of your new design. Make sure that your diagram shows all the important UML parts, and that it implements the appropriate design principles we studied in this course



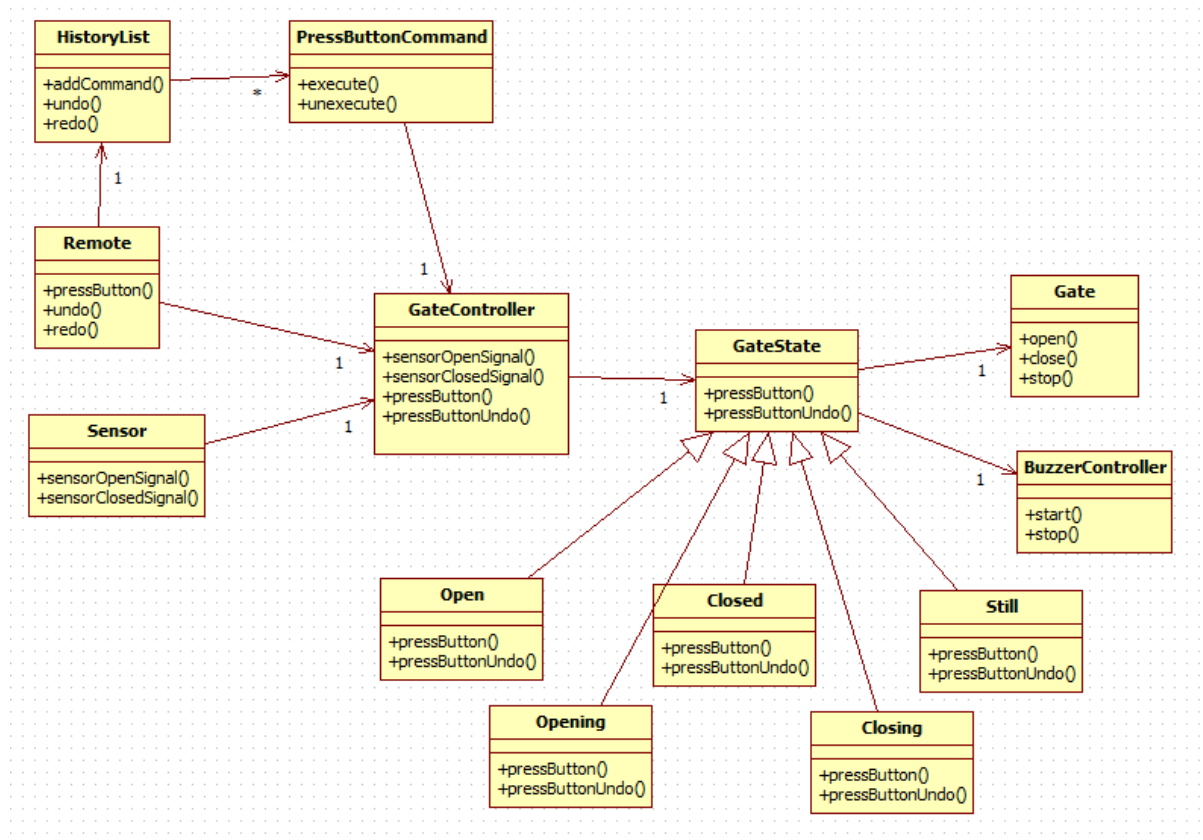
b. [10 points] Draw the sequence diagram that shows clearly how your design works. The initial state of the gate is closed, so assume the gate is closed when the sequence diagram starts. Show the sequence diagram of the following user actions:

1. The user presses the button that opens the door
2. When the door is completely open, the user presses the button again to close the door
3. When the door is completely closed, the user presses the undo button.





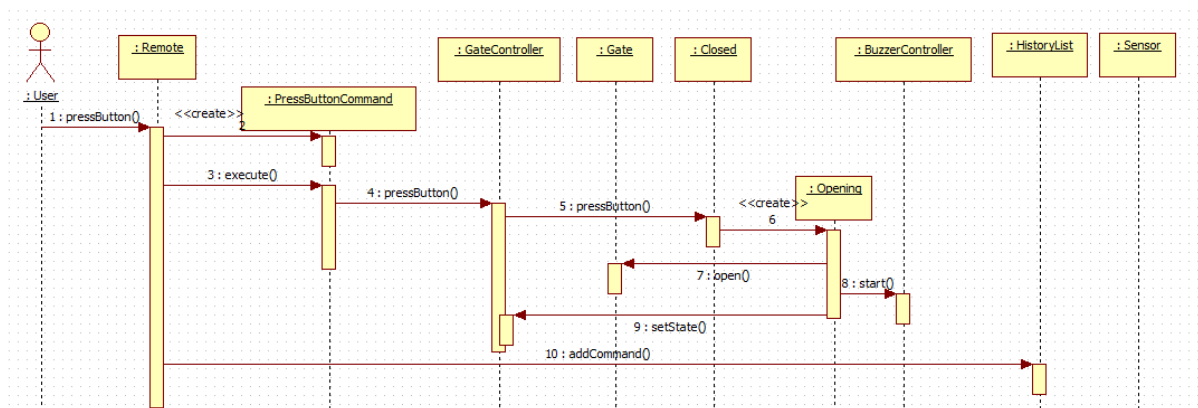
- c. [10 points] Now you receive another new requirement from marketing. When the gate is busy with opening or closing, and you press the button on the remote, then the gate should stop moving and stand completely still. The buzzer should also go idle if the gate stands completely still. In this state, when you press the button again, it should continue with the action it was doing, either opening or closing the gate. You notice now that the implementation of the GateController becomes ugly with a lot of conditional code. Draw the class diagram that shows how you would redesign the class diagram given in part a. Make sure that the undo and redo actions still work fine. Make sure that your diagram shows all the important UML parts, and that it implements the appropriate design principles we studied in this course



d. [10 points] Draw the sequence diagram that shows clearly how your new design works. The initial state of the gate is closed, so assume the gate is closed when the sequence diagram starts. Show the sequence diagram of the following user actions:

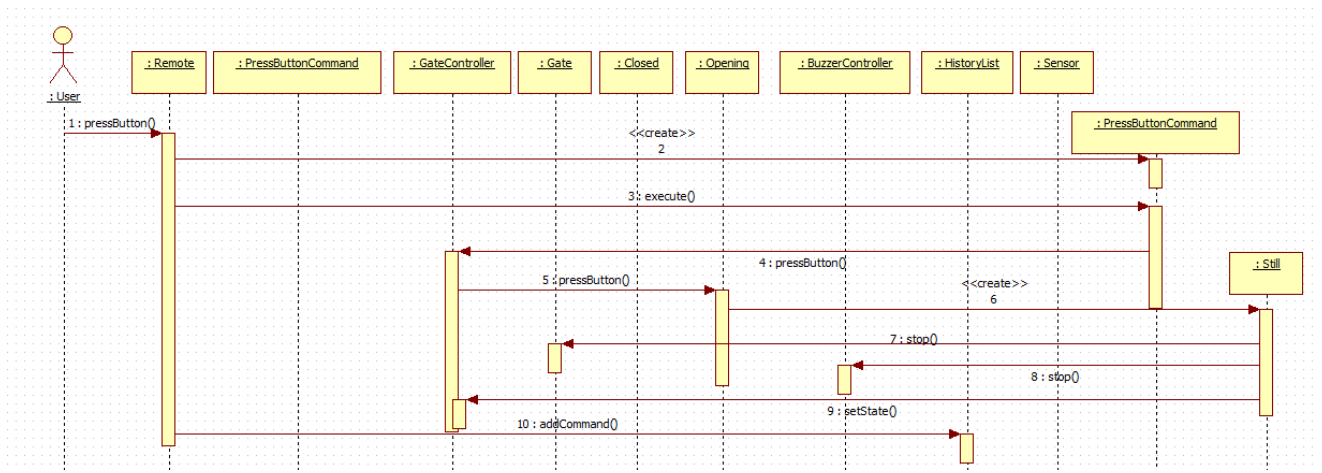
1. The user presses the button that opens the door
2. When the door is half open, the user presses the button again so that the door will stand still.
3. The user presses the button again

e.

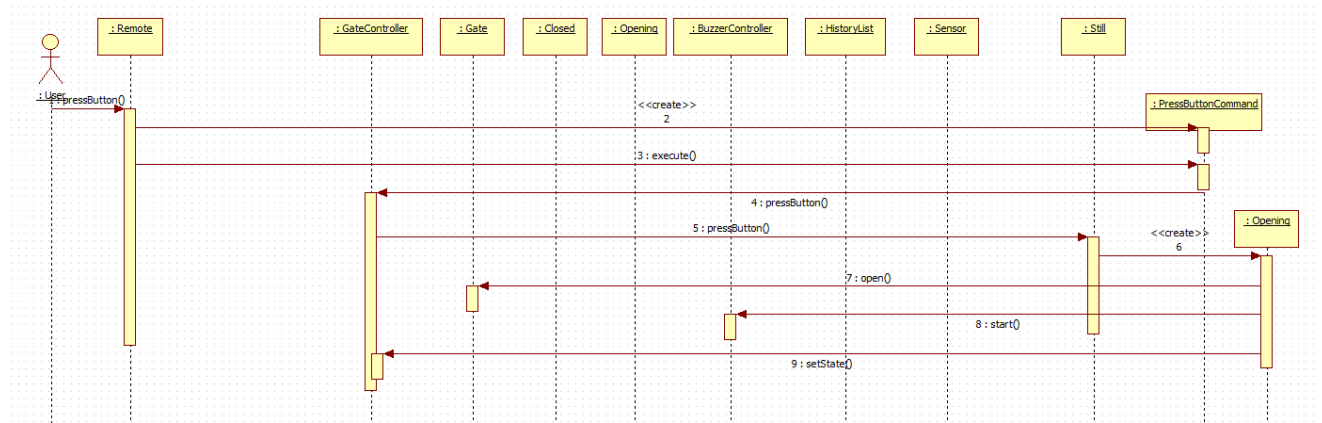


f.

g.

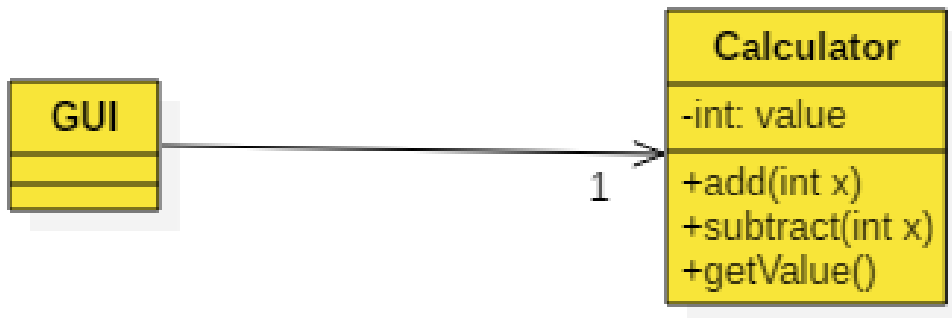


h.



## Question 2 [ 30 points ] {40 minutes}

Suppose we have the following simple calculator application:



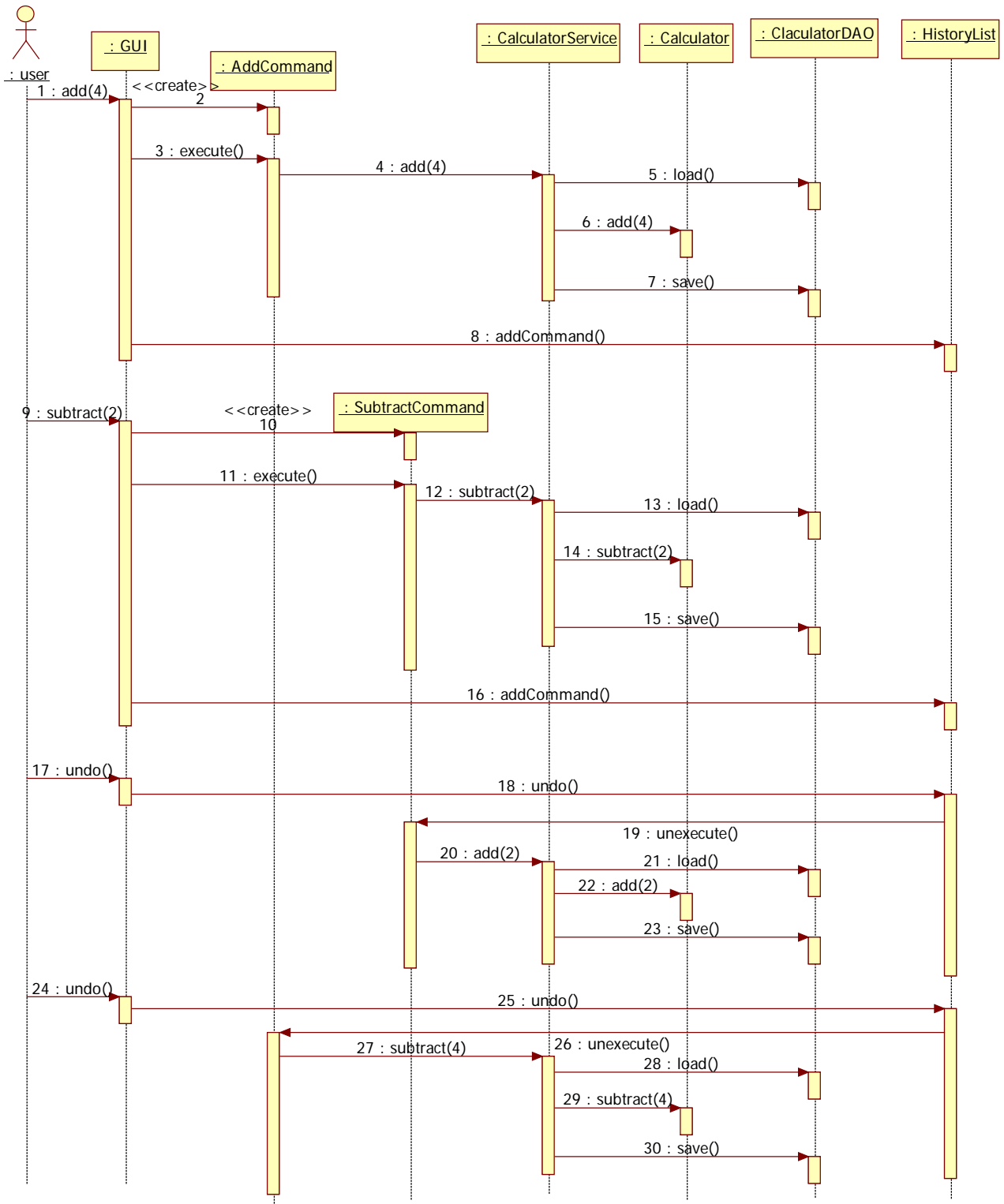
When we start the application, a new Calculator is created with value=0. Then we can perform the add and subtract operations on the calculator. The GUI will show the current value of the Calculator. Now we get 2 new requirements:

1. We need to support undo/redo functionality for the calculator
2. For every add() or subtract() action, we first need to retrieve the calculator value from the database, then do the add or subtract action on the calculator, and finally store the calculator value in the database. This way the database always has the latest value of the calculator.

It is your task to modify the design so that it supports the 2 new requirements.

- a. Draw the sequence diagram of the following scenario:
  1. We start the calculator application
  2. The user adds the value 4 to the calculator
  3. The user subtracts the value 2 from the calculator
  4. The user performs the undo action
  5. The user performs the undo action again
- b. Write the complete Java code of one Command class. You can choose yourself which Command class.

## Solution:





```
public class AddCommand implements ICommand{
    private CalculatorService calculatorService;
    private int value;

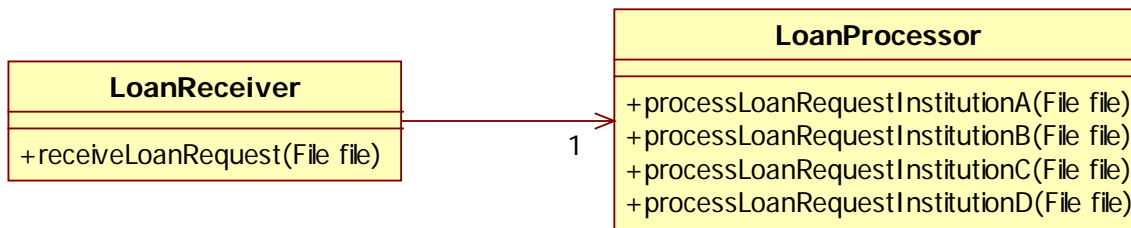
    public AddCommand(CalculatorService calculatorService, int value) {
        this.calculatorService = calculatorService;
        this.value = value;
    }

    public void execute() {
        calculatorService.add(value);
    }

    public void unExecute() {
        calculatorService.subtract(value);
    }
}
```

### Question 3 [ 15 points ] {20 minutes}

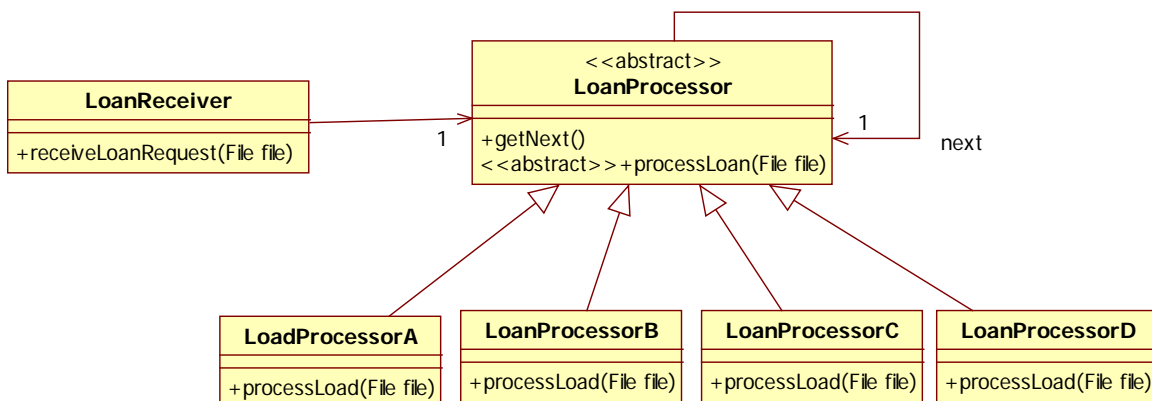
Suppose we have an existing loan application with the following design:

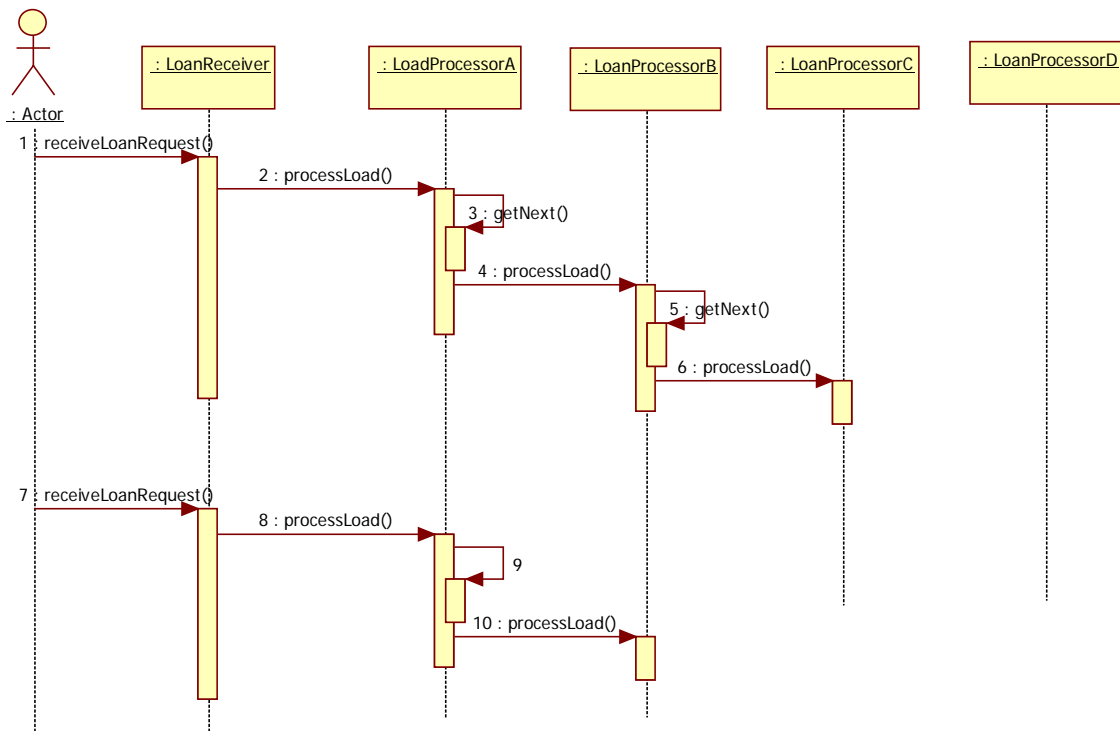


The loan application receives requests from different financial institutions, creates a loan offer and send this load offer back to the institution that submitted the request. The basic problem is that the loan offer request we receive from every institution is different. They all send a file, but they use different file formats, different data formats and the data is structured in a different way for every institution. The current application can handle only 4 institutions (A to D).

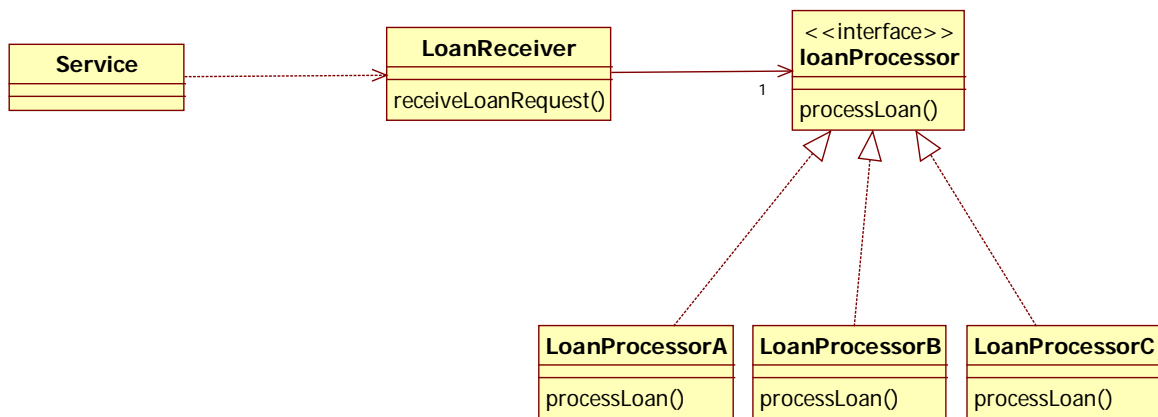
It is your job to create a better design than the existing application. Your design should support processing loan offers from many financial institutions and it should be easy to add support for processing loan offers from new financial institutions.

- a) Draw the class diagram of your design
- b) Draw a sequence diagram that shows how your design works. Show the scenario of the following actions:
  1. Receive a loan request from institution D
  2. Receive a loan request from institution B





Strategy does not solve the problem



**Question 4 [ 5 points ] {10 minutes}**

Describe how we can relate the **Observer** pattern to one or more principles of SCI. Your answer should be about half a page, but should not exceed one page (handwritten). The number of points you get for this questions depends how well you explain the relationship between the **Observer** pattern and one or more principles of SCI.