Homework 1

To: SEng 5801, All Students

CC: NA

From: Heimdahl

Creation Date: 9/29/2022

Re: Homework 1, Use-Cases.

The problem

In this homework we will tackle two problems, the HACS system we discussed in class and a model-rocket launch system.

For both systems, we are asking you to identify the following:

1. The system boundary and the inputs and outputs to the system. You can informally describe the inputs and outputs. We are interested in scoping the system here, not necessarily nailing down all the details of the input and outputs. Thus, the name of the inputs and outputs and a brief informal (natural language) description will do.

2. Develop a use-case diagram for each system.

3. Define two scenarios each for two of your use-cases in each system (giving you four scenarios total for each system). Please pick use-cases that have somewhat interesting scenarios.

The essence of our task

You will develop "use-cases" to help you find requirements. You will be required to turn in a use-case diagram including all the use cases you identify and scenarios describing ***two*** of your use-cases in each system (we are asking for two scenarios per use case giving you four scenarios for each system). Note that these use cases are only to help you find and clarify the requirements, they are not requirements in themselves, and the scenarios do not have to cover all possible cases.

HACS—Homework Collection and Assignment System

Here you will get a set of informal "user requests" of what this system shall do. Note that this is purposely incomplete and vague and that it is your job to sort it all out using use cases.

HACS should aid in the distribution of homework assignments, collection of the assignments, distribution of suggested solutions, and distribution of grades.

The system should:

1. Allow the instructor to distribute the assignment to all students on a distribution list (the class list).
2. Accept solutions from each student. After the solution has been accepted, no other solutions will be accepted from the individual. A confirmation that the assignment has been received must be sent to the student submitting the assignment.
3. After the deadline for the assignment all solutions arriving late should be tagged as being late. The number of days the assignment is late must be recorded.
4. The solution should be automatically distributed to all students at a predefined date and time.
5. All solutions arriving after that time should be singled out and tagged since they will receive a grade of 0.
6. After the solutions have been graded the students will be notified of their grade.
7. The students should have some (secure) mechanism to review the grades for the semester.

This set of user requests will form the basis for your use cases.

Rocket Launch System

Model rocketry can be a dangerous business and one wants to be far away from the launch pad when the rocket takes off. The solid fuel rocket motors are ignited with an "igniter"—a small wire inserted into the rocket motor that ignites the rocket when a current is passed through the igniter. The simplest possible system would have a battery, a simple switch, and wires connected to the igniter. The rocketeer would insert the igniter in the motor, connect the wires to the igniter, and when the switch is closed, igniter ignites, and the rocket takes off. Hopefully, the battery was not connected and the switch open when the wires were connected to the igniter; if not, we have a rocket exploding in the rocketeers face. In addition, long wires (we must be a safe distance away) tend to break so when we close the switch, the rocket does not take off, and we are in the uncomfortable position of trying to figure out if the circuit was broken or the rocket motor is just smoldering waiting for us to go out to the launch pad and then explode in our face. This situation is not good, and we can do better—let us put some computers and software in here…

After some thinking, we have come up with the following rough design. We will have two units, a launch pad unit and a control unit; they will communicate wirelessly using some mechanism.

The launch pad unit contains the battery, has two buttons (test and enable), two lights (red and green), and two connectors for the wires to the igniter. The thinking is that we wire the igniter to the unit (and the circuit is not closed in the unit) and the unit cannot communicate with anyone. When we push the test button, a very low current (not enough to ignite the igniter) is sent through the wires to see if the battery is charged and the circuit closed. If so, the green light will go on. When we are convinced it is all good, we push the enable button and the red light comes on. The enable button enables the communication with the control unit so that we can launch the rocket. We now walk back a safe distance and grab our control unit.

The control unit (in this simplified version) has two buttons (ready and launch) and two lights (red and green). When we push the ready button, the control unit notifies the launch pad unit that we are ready to launch. If the launch pad unit acknowledges the command, the control unit red light is turned on. When we push the launch button, the control unit sends a command to the launch pad unit to launch. If the launch pad unit acknowledges the launch command, the control unit turns on the green light.

After receiving a ready command and a launch command, the pad unit closes the circuit, and the rocket takes off.

As you might have figured out, there is a lot more to this launch system than these happy paths described above. For example, can we abort a launch? What if communication is lost?

Your task

As mentioned in the introduction, for both systems, we are asking you to identify the following:

1. The system boundary and the inputs and outputs to the system. You can informally describe the inputs and outputs. We are interested in scoping the system here, not necessarily nailing down all the details of the input and outputs. Thus, the name of the inputs and outputs and a brief informal (natural language) description will do.

2. Develop a use-case diagram for each system.

3. Define two scenarios each for two of your use-cases in each system (giving you four scenarios total for each system). Please pick use-cases that have somewhat interesting scenarios.

There are some examples of use-case diagrams and scenarios in the slides.

Helpful Hints

Do not invent many unneeded requirements and use-cases. Focus on the core functionality of HACS and the Rocket Launch System, do not add “things that would be nice to have”. “Gold plating” is not a good thing.

Do not try to capture all possible processing scenarios with your use-cases. As mentioned in class, focus on the main processing to clarify what required functions are needed. This is typically enough for the customer to understand the system and for you to get a basic understanding of the customer’s needs.

Asking Questions

If you have questions about how HACS or the Rocket Launch System work (and you should have questions), please first check Canvas discussion (I hope to have that figured out by now) and see if this has been addressed If not, post the question there. If no response, mail Professor Heimdahl and make sure he gets to it. For now, Professor Heimdahl is the customer, he is always right, and will respond on the discussion board.

Deliverables

This is an individual assignment, and you are required to turn in a pdf file with both systems addressed.

Due Dates

HACS and Rocket Launch System use-cases: In Canvas as a pdf file, Saturday, October 8.