Software Modeling (401016): Final Exam

Instructions:

- Carefully read the text before staring the exam
- No books or reference material is allowed
- No calculator, mobile phones or other electronic devices

Part of this exam is based on the following case study:

E-house is an engineering company specialized in the development of domotic systems. They want to develop a control panel that allows users to manage and personalize their living environment. Specifically, the control panel allows users to manage the lights, the temperature, the security alarm, the hi-fi system, and the curtains independently for each room in the house. The control panel allows users to directly regulate all the aspects mentioned above or to set a timer that regulates them automatically at specific hours of the day. Users can modify the settings of the domotic system directly accessing the control panel or remotely through an application on their mobile phone that can issue commands to the control panel.

In addition:

- The control panel always keeps trace of the behavior of the users in terms of energy consumption (it records when the users turn on the heating system, when they wake up or leave the house, etc.). This information is used to suggest concrete actions to reduce or optimize their energy consumption.
- The display of the control panel always shows the current wheatear forecasts and the latest news collected in real-time from the Web.

Note: this problem description may be ambiguous and incomplete. In answering the questions, you are free to complete it (if needed) and briefly motivate your assumptions.

Theory Questions

1. Use the five decision points learnt in the course to illustrate the properties of the life cycle model "V-model". Is it heavyweight, lightweight or hybrid? Why? **[1 point]**
2. Explain the concepts of coupling and cohesion. Discuss if and why we prefer systems with high (or low) coupling and high (or low) cohesion. **[1 point]**
3. LOC (lines of code) is a measure of complexity for software systems. Explain trough an example why LOC is not always an effective measure of complexity. **[1 point]**

4. Explain the role of service providers, requesters and service registry. Discuss where in the above case study may make sense to have a service provider and requester.
   [1 point]
5. Explain the difference between functional and non-functional requirements. Provide an example for each.
   [0.5 point]

Questions related to the case study

6. For the case study, specify the functional requirements with an UML use case diagram. Use additional text to describe your model and your assumptions where needed.
   [2 points]
7. Complement your specification with a UML sequence diagram that models the scenario in which a user is setting a timer to broadcast some music in a specific time the morning in the sleeping room.
   [1 point]
8. Choose a UML diagram to represent the design for the above case study. Use additional text to document your design. Motivate the choice of diagram and ensure constancy with your former specification provided in questions 1 and 2.
   [1.5 point]
9. The following code excerpt sorts the rooms in the house by their temperature. Compute the LOC and the McCabe complexity measures (for the latter consider p=1).
   [1 point]

```
public void sortRoomsByTemperature( Room[] rooms)
{
    int i, j, first, temp;
    for ( i = rooms.length-1; i>0; i--){

          first = 0;
          for(j = 1; j <= i; j++){
                if(rooms[j].getTemp()<rooms[first].getTemp()){
                      first = j;
                }
          }
          temp = rooms[first];
          rooms[first] = rooms[i];
          rooms[i] = temp;
    }
}
```