Software Systems Design Test
13 December 2018, 8:45–11:45

Program: Technical Computer Science / Business & IT, University of Twente
Module: 201700117 Software Systems
Module Coordinator: Luís Ferreira Pires
Responsible teacher: Klaas Sikkel

- Different questions will be graded by different persons. Therefore we ask you **to use a separate sheet for each question** (not the back side of another question).
- You are allowed to bring the lecture slides, but no other documentation or personal notes.
- Diagrams can be drawn with pen or pencil as you like.
- When you are ready, please hand in only the answers to the questions. You can take the test with you.

Questions 1–4 relate to the following case description.

Hotel Administration

The Marigold Hotel is a large hotel located in the city. It does attract tourists, who want to visit the city, but increasingly it is also used by business travellers. The hotel wants to simplify procedures for companies who regularly have employees staying in the Marigold. This requires some adaptations in the software.

You are requested to make a design for the software system for the Marigold Hotel.

Bookings for the Marigold hotel can be made online. When making a booking, you have to indicate the day of arrival, day of departure, and what kind of room you want. The hotel has single and double rooms with different luxury standards. E.g. a double room "executive style" has more space and better furnishing, but is also more expensive than a "standard" double room.
Employees from companies that have a contract with the hotel have to indicate their company when they make a booking. Private customers (i.e., not from a company) have to give a valid credit card number. (At the end of the stay you can pay in any way you like, but if you don't show up, the fee for one night will be debited to your credit card.)

A customer who arrives at the hotel goes to the reception desk to check in. A receptionist then checks the customer in in the system. This involves assigning a room and programming a key. For example: when a customer books a standard single room, it is left unspecified which room exactly it is going to be. When the customer checks in, the receptionist chooses one of the available standard single rooms, possibly asking the customer for preferences (e.g. better view at the front side vs. quieter room at the back side). The key is a plastic card with a magnetic stripe that can be programmed to allow entrance to a specific room for a specific period.

Customers can also arrive at the hotel without having made a booking in advance. In that case, they inquire at the reception whether there would still be a suitable room. If so, the receptionist registers the customer and checks the customer in. If there is no suitable room, the customer has to look for another hotel.

Customers staying in the hotel can make use of various services. If you have a drink at the bar, you can ask the barman to put it on the bill, rather than paying directly in cash. Also you can order food and drink to be brought up to your room; the hotel can provide laundry services; etc. Any hotel staff (reception, kitchen, ...) can enter services into the system when they are provided, so that it can be paid for when the guest checks out.

For checking out, a customer goes to the reception. The receptionist enters into the system that the customer checked out. For private customers, the receptionist prints a bill and the customer pays the bill (the payment method does not matter). If any services were provided, these will have been added to the bill. For company customers, the bill will be sent to the company. However, if any services were provided, these will be billed to the customer (not to company); a bill for these is printed and paid at check-out.

At the last day of each month an invoice is sent to company contact persons with the bills for their employees for that month.
The company contact person then pays the invoice. (*Cases where company contacts have to reminded that they need to pay are very rare and need not be included in the model.*)

Last but not least, the Marigold Hotel has a generous cancellation policy. Until 18:00 on the day of arrival it is possible to cancel, without any costs. A booking can be cancelled in two ways: online or by calling the hotel reception. After 18:00, cancellations are no longer possible.
If a customer has not checked in by midnight, it will be considered a *no-show*. At midnight, the hotel system will automatically undo the booking and charge the price for one night to the customer's credit card. (For company customers: include it in the invoice at the end of the month).

Question 1 (Activity Diagram) [2.8 points]

Draw an activity diagram with the activities related to stay in the hotel – from booking to payment.

Hint: Do not distinguish between different kinds of hotel staff for the Activity Diagram.

Question 2 (Use Cases) [1.7 points]

2a (Actor list) [0.4 points]

Make an actor list for the system.

2b (Use case diagram) [1.3 points]

Draw a use case diagram for system.

- You do not have to «include» additional use cases for functions that are part of a more general use case.
  For example: when the receptionist handles a customer who checks in, the receptionist should, among other things, allocate a room – but there is no need to write an extra use for this, as shown in Figure 1.

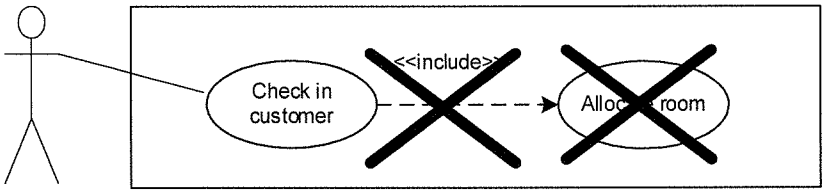

Figure 1: There is no need to include extra use cases for standard parts of a more general use case

- However, if a use case includes a function that is sometimes, but not always executed, depending on the circumstances, then please model this using «extend».
- Payment involves other intermediate systems. However, for the sake of simplicity it can be modelled as an interaction from the paying person with the hotel system.

Question 3 (Class Diagram) [3.2 points]

Draw a class diagram for the proposed system. *Use generalization when different classes share attributes or associations.*

In addition to the case description above, please take the following information into account:

- A has a room number and one or two beds. A room is further characterized by a luxury standard. All rooms with the same luxury standard have the same price per night. For example: for a double room "executive style" this is currently €265. Occasionally something in a room breaks down. If it cannot be repaired immediately, it is possible to indicate in the system that a room is unavailable until a particular date.
- For a booking, in addition to what was specified above, the following information is known: time[1] of the booking; check-in time; check-out time; (if applicable:) time of cancellation (for a no-show this is midnight after the booked date of arrival).
- When a service for a customer is registered, the following data are registered: description of the service; price; employee number of the hotel staff member who enters the service; time at which this is done.
- For private customers the following information is known: surname; first name; telephone number; credit card number.
- For company customers the following information is known: surname; first name; telephone number; company which will pay the bill. For the company a name and address needs to be known, as well as a company contact person. Perhaps surprisingly, one contact person can act on behalf of different companies, e.g., when related companies delegated part of their administration to a shared administrative office. For a company contact person the following information is known: surname; first name; telephone number; e-mail address.

Question 4 (Sequence Diagram) [1.1 points]

Receptionists answer various kinds of telephone calls. Related to the functions described in this case study, these are cancellations and requests for service. *(People could phone with all kinds of other questions, which we will ignore here.)* From a system perspective these could be modelled by means of different sequence diagrams, one for handling cancellations and one for handling service requests. However, from the perspective of a receptionist it is logical to model it by means of a single sequence diagram: if they pick up the phone it is not yet clear what kind of phonecall it is going to be.

If a customer phones the reception for a service request, there are two ways in which the receptionist can hande it. Either: the receptionist enters the requested service into the system and sees to it that it will be delivered. Or: the receptionist only acts as an intermediate and passes the request to other hotel staff (e.g. by phoning the kitchen). In that case the receptionist does not have to enter anything into the system, the other staff will take care of it as they deliver the service.

It is possible that a customer phones with different service requests at one time, e.g. asking for laundry services and also ordering food to be delivered.

Make a sequence diagram for handling a phone call by the receptionist. You don't have to include a control object.

[1] It can be assumed that an attribute of type "Time" will include both the time and the date.

Question 5 (Software Metrics) [1.2 points]

5a. Lack of Cohesion in Methods [0.4 points]

The metric LCOM2 is defined as follows:
For a class with a attributes $A_1 \ldots A_a$ and m methods $M_1 \ldots M_m$,
let mA_k be the number of methods that access A_k,
with avg(mA) the average of mA_k for k = 1 ... a,
then LCOM2 = (m − avg(mA)) / (m − 1) .

Determine LCOM2 for the following class. Show how you compute your answer.

```
public class Position {
      private int h_pos;
      private int v_pos;

      public Position(int h, int v) {
            h_pos = h;
            v_pos = v;
      }
      public void goUp() {
            v_pos = v_pos + 1;
      }
      public void goDown() {
            v_pos = v_pos - 1;
      }

// continued in the right column
```

```
//continued from left column

   public void goLeft() {
         h_pos = h_pos - 1;
   }
   public void goRight() {
         h_pos = h_pos + 1;
   }
   public int getHorizontal() {
         return h_pos;
   }
   public int getVertical() {
         return v_pos;
   }
}
```

5b. Cyclomatic Complexity [0.8 points]

Both methods below compute the same result in a slightly different manner. Draw a flow graph and give the cyclomatic complexity for each method. Show how you compute your answer.
N.B.: Math.*signum*(i) returns 1.0 if i > 0; −1.0 if i < 0 ; 0.0 if i = 0.

```
public String sign1(int i) {
     String result;
     if (i < 0) {
           result = "negative";
     } else if (i > 0) {
           result = "positive";
     } else {
           result = "zero";
     }
     return result;
 }
```

```
public String sign2(int i) {
     String result;
     switch ((int) Math.signum(i)) {
     case -1: result = "negative";
        break;
     case +1: result = "positive";
        break;
     default: result = "zero";
     }
     return result;
}
```