Module Software Systems (201500111)
Design Test, 15 December 2016, 8:45–11:45

- Different questions will be graded by different persons. Therefore we ask you **to use a <u>separate sheet</u> for each question** (<u>not</u> the back side of another question).
- You are allowed to consult the manual and a printed version of the slides. It is <u>not</u> allowed to consult other materials – including your own 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.

Outdoor Holiday Tours

Outdoor Holiday Tours (OHT) is a travel company specialized in outdoor group travels. It started as an initiative of a small group of tour guides who organized their own tours, with a bare minimum of administration. But due to enthusiastic posts on social media they get more and more participants, and the program is expanding. This calls for a more professional organization. One of the things OHT needs is a proper booking system. You are asked to help making a design for this system.

OHT organizes hiking tours in different parts of the world. They also do canoe tours, but these take place only in Europe. Accommodation during a tour is in simple guest houses or tents provided by OHT. The length of a tour ranges from a few days to a few weeks. Some tours are more demanding than others. The difficulty of a tour is indicated by a number of footprints ranging from one footprint (very easy) to seven footprints (very challenging). The full program with information about all tours is shown on the website. Some tours are quite popular and are offered multiple times per year. In particular the canoe tours in France enjoy high demand.

Prospective participants can make a booking on the website. (OHT consistently speaks about participants, rather than customers; active participation in the group is key to making a tour successful.)

A booking can be made for one person or for a small group of up to five persons (friends or family going together). A booking through the website is provisional, in the sense that your booking will be deleted if you don’t pay an advance payment in time. The advance payment, a certain percentage of the tour price, should be paid within ten days. If not, the (provisional) booking will be cancelled automatically.

Sometime before the start of the travel, a final payment is due. If the payment hasn’t been made when in time, the participant (for a small group: the contact person of that group) will automatically get a notification, stating that it must be paid within a week. After that week, if no payment has been made, OHT staff will cancel the booking. (This is not done automatically by the system, leaving room for some offline negotiation in special cases. *You don’t have to model this, just assume that OHT staff cancels the booking.*)

When the final payment has been made, OHT will send the participants further documentation about the tour by regular mail.

Unfortunately it can happen that a participant has to cancel a booking. In this case the participant might get a refund, depending on the time of cancellation. Cancelling a booking cannot be done though the website. You have to contact the OHT office, by telephone or email, OHT staff will then enter the cancellation into the system.

Whether you get a refund when you cancel depends on the time of cancellation. If the cancellation is long enough in advance, you get the advance payment back (minus an administrative charge). If you cancel after the final payment, you'll get refunded a percentage that depends on the time left till the start of the tour (0 % for the last few days). The details of this are on the website.

It may also happen – fortunately not very often – that OHT has to cancel a tour (and then cancel all the bookings for that tour). This could happen when the minimum number of participants has not been reached. In such cases, OHT will do its best to offer alternatives or special deals to the participants of the cancelled tour (*not part of the process to be modelled*). The advance payment will be fully refunded, of course. (Such cancellations are always before the final payment is due).

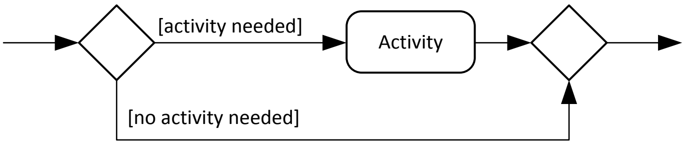
In addition to what was mentioned above, the systems stores the following information.

- A tour has a name; description; number of footprints; maximum ascent per day (only for hiking tours, not for canoe tours); minimum and maximum number of participants; number of days; start date; price per person.
  The price per person of a tour can vary; typically the same tour is a bit more expensive in the peak season.
  When a tour has been cancelled, this is also indicated in the system.
  For canoe tours OHT locally hires canoes. From the canoe rental company the name, address, tel.no, and name of the owner are known. When the same tour is organized multiple times, the canoes are always supplied by the same canoe rental.
- Each tour has a tour guide. For a tour guide is known: name; address; gender; SSN (social security number). It is possible, however, that a tour is in the system (and thus can be booked) while no guide has been assigned yet.
- Each booking has a unique booking number. For each booking it is known how much money has been paid so far.
  When a booking is made, the following information about a participant is stored: name; address; gender; birth date; diet; telephone number; e-mail address. However, if the booking is for more than one person, telephone number and e-mail address are given only for the contact person, not for the other participants in the same booking.
  The attribute *diet* states special dietary requirements of the participant (if any).

Question 1 (Activity Diagram) [2.4 points]

Draw an activity diagram for the booking (and possibly cancellation) of a tour with OHT.

Tip: You might wonder whether there is a notion of an *optional* activity in activity diagrams, i.e. an activity that may or may not take place, depending on the circumstances. No special construct has been defined for this; it can easily be modelled by means of a branch + merge, as follows:



Question 2 (Use Cases) [1.6 points]

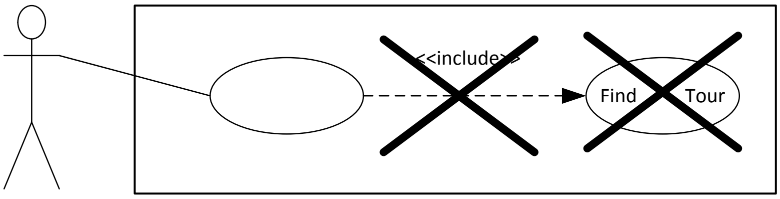
2a (Glossary) [0.4 points]

Make a small glossary with 3 entries for the booking system for OHT. Include those terms that you consider most appropriate, e.g., terms that have a specific meaning in this context.

2b (Use case diagram) [1.2 points]

Draw a use case diagram for the booking system for OHT.

- You do not have to include use cases for activities that don't relate to bookings, such as adding new tours to the program, appointing tour guides, etc.
- Also, you do not have to include auxiliary use cases to look up things in the system, like "Find Booking" (as in the Theatre Tickets case) or "Find Tour", as illustrated below



Question 3 (Class Diagram) [3.3 points]

Draw a class diagram for the proposed system. If different classes share several attributes, please model this by means of a generalization.

Question 4 (Sequence Diagram) [1.2 points]

Draw a sequence diagram for making a booking.
To limit the amount of drawing, you do not have to include a control object (and, as a consequence, you don't have to draw return messages).

Making a booking involves the following steps:

- Selecting a tour
- Creating a new booking for that tour
- Adding the participants with the requested data for each participant
- Confirming the booking when it has been completed.

Please note:

- If a booking is made for multiple participants, contact details are needed only for the first participant (the contact person), not for the other participants.
- Some participants have special dietary requirements due to an allergy or according to their religious background. Where applicable, this can be added.

Question 5 (Software Metrics) [1.5 points]

5a (Coupling and cohesion) [0.5 points]

LCOM as presented in the slides uses two counts, P and Q:

- P is the number of method pairs (including constructors) that do not share any accessed variable
- Q is the number of method pairs (including constructors) that share at least one accessed variable
- If P>Q, then LCOM1 = P-Q, otherwise LCOM1 = 0

Consider a class that has 4 fields of different types, each with its own get-method, and no other methods. Moreover, there is a constructor that provides each field with an initial value.

1. What is the LCOM value for this class, according to the definition above?

2. If you split the class in two separate classes, each with two fields and the corresponding get-methods and a constructor that initializes both fields, then what are the resulting LCOM values?

5b (Cyclometric complexity) [1 point]

Consider the following two Java methods, which implement slight variations on a test whether a List of Integer numbers is ordered.

```
public boolean isOrdered1(List<Integer> list) {
       boolean result = true;
       int i = 0;
       while (i < list.size()-1 && result) {
              result = list.get(i) <= list.get(i+1);
              i = i+1;
       }
       return result;
}

public boolean isOrdered2(List<Integer> list) {
       int i = 0;
       while (i < list.size()-1) {
              if (list.get(i) > list.get(i+1)) {
                     return false;
              }
              i = i+1;
       }
       return true;
}
```

1. Draw the flow graphs, taking into account that the &&-operator in the **while**-condition in isOrdered1 only evaluates its right-hand side operand if the left-hand side operand yields **true**.

2. Compute and compare the cyclomatic complexity of these two methods. Explain your computation; do not just give your answer.

3. Explain the difference you see in the cyclomatic complexity of isOrdered1 versus isOrdered2, or explain why their complexity is the the same if there is no difference.