

Lab 1 - Object-Oriented Software Development / ... with Design Patterns TOMK18 / TOUK18 - H22

Use Cases, Sequence Diagrams

last presentation opportunity: Thursday, 15 September 2022

submission deadline: Friday, 16 September 2022, 23:59

Jönköping University, School of Engineering

Tuesday, 30 August 2022

General instructions

In this lab we deal with the process of requirements gathering. A very common tool used to detect requirements are *use cases*. Your main task is to write use cases for a flight booking system. You will during this process also apply your UML knowledge (on *use case context diagrams* and *sequence diagrams*). However, the main challenge is to write *good* use cases. It is therefore **indispensable** that you read chapter 6 on *Use Cases* in our book *Applying UML and Patterns*. Another reason to read chapter 6 is that we will follow the terms and notions from chapter 6 without further explanation. Further, in order to produce something *good*, recall that you may need to redo and refine things, go back and rework parts, before you proceed or obtain your final result.

1 The system to develop

A flight booking system shall be developed. We give here a rough description of the system, containing a few requirements. It is your task to detect and document many more requirements writing appropriate use cases.

In order not to complicate things too much, we only deal with one airline and only flights that this airline is operating itself. The system shall be usable by the customer through a web-interface, or by the customer service personal (that might be on the phone with customers). Otherwise, to put it short, the customer shall be able to book/purchase flights via the system.

You may well think of an existing (online) flight booking system that you know, in order to come up with use cases. You may also confer chapter 6, in particular the guidelines from 6.15 and 6.16 (3rd edition only).

2 Your task

In order to simplify your task a little, and to make it easier for you to convince your lab assistant that you did your work and understand this task, we break it down into the following incremental steps.

1. Identify *actors*, at least four. Identify *goals* for each actor.
2. Identify *use cases*, at least four.
3. For each above use case, classify the actors into the roles *primary*, *supporting*, *off stage*. Possibly revise your actors and/or use cases (note that the same actor could have different roles in the context of different use cases).
4. Create a *use case context diagram*¹.
5. Write the use cases from step 2 in *brief* or *casual* format.
6. Pick a use case (from step 2) and develop it into a *fully dressed* one, following the format from *Alistair Cockburn* (see lecture slides). It shall contain at least three scenarios.
7. Pick a scenario (from any use case from step 2) that involves at least four system events/messages and at least three objects/actors. Create a sequence diagram for this scenario¹.

When you go through these steps, remember that you might need to iterate/go back to previous steps, in order to adjust to new knowledge or insights you obtained in further down steps.

3 Deliverables and presentation

You shall deliver *one* pdf document that contains the results of steps 1 to 6. You need to orally present your solution to your lab assistant. Be prepared to answer any question regarding your solutions. If you want to draw by hand, this is accepted, as long as it is **well-readable** and submitted within the pdf (scann it).

4 Hints

There is room for creativity in this lab. But keep it realistic: extremely fancy or unreasonable systems might not pass. Be also careful with minimalist solutions. Are they of *good* quality?

¹You may use a tool for this, there are numerous free tools available on the web.