Hotel Project
Description

1 First of all

This is a very different problem from what you are used to at Chalmers. Usually you obtain a problem and are then asked to give a solution. Here the problem is a lot more complex; as a group you have to define the problems, make the requirements and models for the solution and then implement them. This might sound simple, but it is far from that. Some of you will find this extremely frustrating, even to the extent that you don't know what to do.

Well, this is what you will have to deal with when you have finished your studies. You will not get your problems clearly defined and served on a silver platter in industry. The problem to solve will not be clear for you before well into the course. You will only get there by working hard and redoing the design a number of times.

In summary, you have to:

- Find the problems to be solved
- Argue that these are the problems
- For each problem, find the best possible solution consistent with the other solutions
- Justify your solutions

2 Project Overview

The aim of this project is to design a hotel management system for running a hotel business. The system should be as flexible as possible so that it can be used for different hotels. You have to find out which procedures hotels have and based on that information, you should create a system which makes it efficient. You need to find out how a hotel system works. Use the Internet, use your own experience, or talk directly with people in the hotel business. The more diverse the sources of your information are, the better the resulting system will be, and, possibly, your grade. To be more precise, your system shall not support the finding of a hotel, but the running of a hotel! Essential features which your system has to support are: making a booking, checking the availability of rooms, and the check in and check out procedures. The system does not need to support the running of the economics of the hotel, such as salaries, but it needs to be able to calculate what a customer has to pay for their stay.

The overall project has an analysis phase – defining the scope and the requirements of your system, a design phase – designing the software system, and an implementation part. After the analysis and design phases, your work up to that point will be assessed by another project group – the so-called half-time review. This is not in order to pass/fail you, but to give you feedback and encourage improvements! The project will be graded based on your final report and the implemented system. Additionally, you will have to produce an individual report which is used for individual grading; more details in section 5.

3 Supervisor and Hotel Manager

The course assistants and the course responsible will play two roles in the project. The first role is as a supervisor, explaining the issues related to modeling and how these models can be applied in a general setting. The second role is as a hotel manager, the person who has the money (course points) and wants as good a system as possible. We, the project managers, have been travelling to many of the corners of the world, for business or holidays. So, we have some understanding on how a hotel works, in particular the booking part. We are more than happy to share this domain knowledge with you throughout the course. As all other customers, project managers, or humans for that part, we have our own opinions and views on things. This means that you should not base your entire system purely on our knowledge and input. It is crucial that you retrieve information from other sources, like internet booking systems, staff at real hotels, and, very importantly, through discussions among each other!

4 Scope and Correctness

There are hundreds of hotel booking systems out there. Try to think from a business perspective: Which innovative features could you offer in order to distinguish your system from existing ones? How can you stick out? **We welcome and encourage creative solutions.** At the same time, the core functionality of the hotel business has to be there!

We do not know nor can we know all the solutions of this project, because there is not a single solution, but several. The benefits of a solution depend on what you want to achieve with your system. This also means that we often cannot tell you whether your solution is 'right' or 'wrong'. What we can do is to guide you to make sure that your solution does not become too simple or too complex, and discuss the advantages and disadvantages of it. For example, a hotel system which can only handle one room would be far too simple and would be considered a failed project, no matter how well the analysis and the design of the system is.

5 Individual Part

In the second week, you will assign a responsible for each part of your system's functionality. The responsible has to make sure that their functionality is up-to-date, reasonable, and finally gets implemented (of course taking into consideration that not every feature can make it into the final system). In the end of the project, you will be required to hand in an individual report describing this process. Tell us how your features were designed, how they changed throughout the process, which decisions had to be made and your role in all of this. If a feature did not make it into the implemented system, tell us why. Also tell us why a feature ended up not working. Note that being responsible for a feature does not mean that you have to implement and do everything yourself, it is still a group effort!

The rationale for this task is not to give you extra work, but to introduce you to a typical role in industry: the role of **Function Owner**!

6 How to work

There are some critical points to consider if you want to finish the course with a smile on your lips.

6.1 Read the literature - and find your own literature

This is a project course and many students seem to assume that it is enough to read most of the lecture slides. They are wrong. First of all, lectures can't cover all aspects of the material. More importantly, the more you read the more certain you will be when you carry out the assignments. The time you spend reading will make a huge impact on the time you spend arguing and discussing how to do things within the group. A sound theoretical understanding of the models you are using will also mean less time wasted on re-doing your assignments after the supervisor has discarded them.

Additionally, the literature which is most suited for you is the literature that covers exactly the things which you need to understand in more detail. This depends a lot on the individual and we cannot provide you with a recommendation that fits everyone. We provided you with literature recommendations covering most aspects of the course material. However, try to find your own literature (e.g. on www.lib.chalmers.se) which fits you best!

6.2 Hand in your assignments on time

We will read your assignments and comment on your work - if you hand them in on time. Make sure that the supervisor gets your report in time and that all parts are included in the report. Otherwise you will miss out on the opportunity of getting feedback on your work.

6.3 Justify

There should always be a justification to your solution. And a discussion of what you have excluded from your solution and why. Never draw a model without explaining what it represents. Diagrams should be given names and be referred to in the textual description of your system so that it is clear what process/structure it represents and why.

6.4 Work as a group

It is tempting to divide the job between youselves to save time and effort. This is alright. But make sure that everyone in the group knows and understands your models. You will otherwise get stuck during the final examination when you are asked precise questions about the report and the implementation. In total you should spend anything up until a total of 200 hours on the course.

One tip from us: It is crucial to work together a lot during the frst part of the project.

6.5 Iterate

Work in an iterative style and everything will become easier. Go back and make sure that your old models are consistent with your new models. Especially in the beginning, rather write down 'wrong' or unclear things and fix them later, than discussing a single aspect for hours and end up running out of time. Change old models as you get a better grip of what is happening and how things interact. When you implement you should always start with a small part of the system and by testing and debugging make sure that it works before taking on a new part. A working system is preferred over one with several nearly completed processes/classes.