2DV103 – Assignment 1
Requirements Engineering

In this assignment you should work *individually* (NO Teamwork).

If you have questions, post them on the forum. It is ok to help each other in public forums. For any question about the Assignment use **"Assignment 1 Forum"** to get in contact with us.

Please note that all tasks should be documented, that is, a model without text that explains it has little or no value. Do the creative work with pen and paper and stay away from tools. Use tools to prepare documentation after you finished modeling!

Time log

Track the time you spend on this assignment. Create a table where you log time spent on every activity for each task. Include the log in your submission.

Grading

Your submission will receive a grade from A to F where F is Failed. You are allowed to improve your work after the initial submission before the deadline. The grade is final, i.e., you will not get an opportunity to correct/improve after grading.

Your answers should be your own! You are not allowed to copy code, models, or texts (books articles, blogs, wikis) in your answers! Each submission will pass through a plagiarism/clone detection system before correction. If plagiarism is detected, the assignment is failed and a formal investigation will be initiated.

Submissions that arrive after the due date will be downgraded by 25 %-units.

Mauro Caporuscio mauro.caporuscio@lnu.se

Department of Computer Science Lnu.se

Requirements Engineering

The goal of this Assignment is to create a system to manage the front-desk activities of the "Linnaeus Hotel". You have been contracted to replace the existing paper-based system, since your customers believe an automated system will save money and help them to serve guests better. The system will be used to enter reservations as well as to check guests in and out of the hotel.

The hotel contains rooms in which guests can stay. Some hotel rooms adjoin others; that is, there are internal doors between them. Each hotel room is assigned a quality level (e.g. a larger room or a room with a view would be better than a smaller room without a view). Each room also has a certain number and type of beds, a room number, and a smoking/non-smoking status. Each quality level has a maximum daily rate, although the rate that a guest pays may be less.

When a hotel guest wishes to make a reservation, the hotel clerk asks him or her which nights he or she wants to stay and the type of room he or she wants. The system must verify if room(s) are available on those nights before allowing a reservation to be made. To avoid the potential guest getting anxious while waiting for the response about room availability, the verification procedure must take less than 2 seconds to complete.

The hotel needs to record basic information about each guest, such as his or her name, address, telephone number, credit card, passport number, etc. A reservation can be canceled at any time but some fees (a percentage of the room price) may be charged if the cancelation is done too late.

When a guest checks in, a room is allocated to him or her until he or she checks out. When the customer requests a specific room, this can be allocated in advance at the discretion of the manager. The system must keep track of the guest's account, and print his or her bill. To avoid guests standing in a line while waiting for the checkout, the whole procedure must take in average less than 60 seconds to complete.

Task 1 – Domain Analysis Document

Perform a domain analysis about "hotel reservations". This will help you to resolve certain ambiguities that might be present in the above statement of requirements.

Task 2 – Requirements Document

Develop a full requirements definition for the above problem. Among the techniques you should consider employing are the following: 1) take a look at existing front-desk systems; 2) write scenarios; 3) derive use cases; 4) perform use case analysis, to determine who the actors are and what tasks they must perform; 5) narrow the problem statement, excluding features that will not be needed in the first release.

Mauro Caporuscio mauro.caporuscio@lnu.se
Department of Computer Science Lnu.se