# 

2DV603 - Performance Evaluation

By Johan Andersson (ja222um), David Larsson (dl222is) & Fredrik Norrman (fn222ep)

###### Version 1

# 1 Introduction

This document will include our evaluation of the software, motivation for why we think the software is running well and some measurements as proof.

The measurements were made based of the quality requirements as this is the reference we received from the stakeholders about how their ideal performance should be. And the measurements show how well the application is running as a whole as well as all operations run just as fast as the ones being displayed.

# 2 Evaluation

While developing the application, we had a lot of focus on that the performance should be as high as possible. And the application have been thoroughly tested and refactored during the implementation to make sure that the users experience whilst using the application is as smooth as possible.

## 2.1 Motivation

The applications actions as mentioned, runs more or less instantly. Meaning that extensive measuring and calculation of the performance will not prove anything and will most likely only cause confusion, as there is not much to measure. Simply by running the application itself is proof of good performance.

The application will only be used by a small number of people since it is a front-desk application meaning that it will not be public for anyone to use. There will be a select few front-desks at two different hotels that will be using it. So the performance will not diminish by the select people using it at the same time.

The guidelines we received from the stakeholders for how the performance should be is met by far. The rest of how the application should run, performance wise was not specified. Although we used the given guidelines for checking for room availability across the whole application. That a task should not take more than 2 seconds to complete. And no task in the application is even close to reaching that number.

# 3 Measurements

## 3.1 Room availability verification procedure

### 3.1.1 Requirement

*“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.”*

### 3.1.2 Results

Extensive measurements regarding the room availability verification procedure resulted in the average time of around **400 milliseconds** - less than half a second (0.4 seconds). The execution time is the same regardless of preferences picked (e.g. hotel choice, room quality choice etc.).

### 3.1.3 Verdict

As the average time resulting from the measurements is basically only **a fifth** (**20 %**) of the required time, we consider the requirement more than fulfilled.

## 3.2 Check in/out process

### 3.2.1 Requirement

*“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.”*

### 3.2.2 Results

Extensive measurements regarding the process to check in and out guests resulted in the total average time of **900 milliseconds**. Divided into sub-processes, it takes on average around **550 milliseconds** to retrieve the information about the reservation to be checked, and on average around **350 milliseconds** to check it in / out - leading to the aforementioned total average time.

These measurements are applied to both the check in, and the check out, processes. And they yielded the same average times.

### 3.2.3 Verdict

The above disclosed average time resulting from the measurements is around **1.5 %** of the in the requirement mentioned time. Or in rational form - **3 / 200** (three out of two hundred, or three two hundredths).

Considering this is extremely much faster than the specified limit, we regard this requirement more than satisfyingly fulfilled.