Project Report

**Data Storage Paradigms, IV1351**

Dennis Hadzialic denhad@kth.se

11/01/2022

# Tips for Report Writing

## REMOVE THIS SECTION BEFORE SUBMITTING THE REPORT.

*The target audience has exactly the same skills as the author, except they do not know anything at all about the specific application described in the report.*

Consider the following:

## The report must be *centered around the requirements*. Which are they (Introduction), how did you work to meet them (Method), what is the solution that meets them (Result), and how can you be sure they are met (Discussion). This is the IMRaD method.

* **The report must show that you have done the work yourself and that you have understood what you have done. Both of these goals are met by carefully explaining your application (both database and program).**
* Is spelling and grammar correct? Is spoken language avoided?
* Does the report have a good structure with sections, subsections and paragraphs?
* Is the solution clearly explained? Will the reader understand the application? What would you yourself want to know if you read about the application, is that included in the report?
* Is the solution analyzed and evaluated? Are important properties of the application explained? Should there have been more extensive evaluation?
* Is the text clarified with images and/or other figures, and with links to the code in your Git repository? Remember that all figures (images, tables, graphs, code listings, etc) shall be numbered and have a short explaining text.

# Introduction

## This section tells *what* are you going to do.

Explain the task and the requirements on the solution. It’s important to clearly state the requirements. *Also specify which other student you worked with when solving the tasks, or if you worked alone.* Write one single introduction covering all tasks; do not split it into one subsection per task.

# Literature Study

This section must prove that you collected sufficient knowledge before starting devel- opment, instead of just hacking away without knowing how to complete a task. State what you have read and briefly summarize what you have learned. It is your choice if you include literature study for all tasks in the same section, or if you divide this section into one subsection per task.

# Method

## This section tells *how* you solved the task.

Explain how you worked when solving the tasks and how you evaluated that your solution met the requirements. Mention diagram editor(s), IDE(s), DBMS(s) and other tools you used. This section *must be split into four subsections*, one per task. *Do not explain your solution and do not refer to code*, that belongs to the *Result* section.

//START

I used [IntelliJ](https://www.jetbrains.com/idea/) and [DataGrip](https://www.jetbrains.com/datagrip) which are both IDEs from [Jet Brains](https://www.jetbrains.com/). The reason I used IntelliJ instead of NetBeans or something similar is mostly that I am familiar with the IDE and the perks of auto-generating a diagram of the Database on the fly while modifying it and debugging. DataGrip is also a tool that made creating SQL queries a breeze, viewing results and modifying tables/columns very quickly. As for DBMS, we used PostgreSQL, which is recommended to use throughout the whole course.

When solving the following problem/assignment I started by reading thoroughly the requirements one by one. Because our group wanted to complete the higher-grade task, we needed to implement MVC and Layer patterns and implement them correctly. There was a lecture on [Database Applications](https://canvas.kth.se/courses/27118/pages/project#:~:text=at%20the%20page-,Database%20Applications,-.%20The%20following%20must) which made this part much easier because I thought that our application would be similar from the architectural standpoint. The last part for making this a higher-grade rated assignment was to make the code easy to understand and not have repeating code. This is achieved by analyzing a task before starting to write the code. I started by illustrating with pen and paper (iPad in my case) how the different classes will interact with each other and try to see if there is a task that is repeating itself then I know I must make it a function that can be called upon multiple times. And to make the code easy to understand I try always to avoid making a function too long. Course IV1350 helped me understand how to divide code into smaller chunks. This is to make it easier to edit later, make it reusable and make it easier for a non-author to understand. The method should only do what the inspector of the code would think the method-name does. So if a method is called *sortInstruments()* it should sort instruments, if the method does something besides that, then have I failed to make the code understandable and logical.

# Result

**This section explains *the result* of what you did.**

Present the solution. Explain your code and prove that it meets the requirements. It’s very important to *state each requirement that is met* and explain *how you met it*. It’s also important to include links to your code in your Git repository, and to also include diagrams, see Figure 1, and other figures to illustrate your reasoning. Also remember that these figures must be referenced in the text. This section *must be split into four subsections*, one per task.

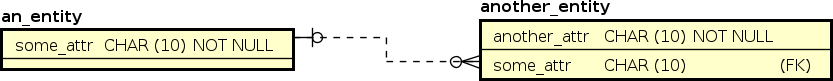


Figure 1: A sample diagram, included to illustrate caption (this text), numbering and reference in text.

# Discussion

## This section *analysis* the result presented in the previous section.

Summarize the requirements and *clearly state which of them you have met*. What lessons have you learned and what problems did you face? How were the problems solved? Should you have done something differently? This section *must be split into four sub- sections*, one per task.

# Comments About the Course

## This section is optional.

Any comment(s) related to this course offering or to coming offerings is much appre- ciated. *Please also tell approximately how much time you spent on the assignment*, including lectures and labs. This is of great help for course evaluation.