**Your Name** <your-ju-id@student.ju.se>,  
**Your Name** <your-ju-id@student.ju.se>

A Project Work in *Client-Server Communication*

Jönköping University 2019

Your platform title here

*In this template, all italic text should be removed and replaced with your own text (which should not be italic); the italic text is just a placeholder letting you know what to write there.*

*On the cover page, change to your own platform title, your own names and your own JU email address.*

*You have a lot of freedom when it comes to writing this report. You do not have to use any part of this template, but the report you write should in the end somehow (in a good way) provide the same information as indicated in this template. Most students trying to do it in their own way usually fail, so if you try that, be sure to know what you are doing!*

*This page should be removed.*

Table of Contents

[Introduction 3](#_Toc18577585)

[Architecture 4](#_Toc18577586)

[Database 5](#_Toc18577587)

[REST API 6](#_Toc18577588)

[Backend Application 7](#_Toc18577589)

[Frontend Application 8](#_Toc18577590)

# Introduction

*Introduce your platform. Write text that* ***indirectly*** *answers questions like:*

* *Why does the platform exist/what is the problem it solves?*
* *How does the platform solve the problem?*
* *Why is your platform solution better than existing solutions (if any exists)?*
* *How will end users use the platform?*
* *...*

*After having read this chapter, those that have never heard of the platform before should a have a good understanding of what it. If they would like to learn how it has been implemented, they just need to continue reading the rest of the report.*

*If possibly, please use pictures/figures of some kind.*

*This chapter can be seen as a pitch text: imagine the reader is an investor, and you should convince the reader that your platform solves an existing problem in an excellent way and that it is worth investing money in it.*

# Architecture

*Give an overview of the platform. Which components does it consist of (backend application, frontend applications, frontend devices, databases, etc.)? Visualize this using a figure and show how the different parts make use of/communicate with each other.*

*For each component, provide a sub-chapter with more information about how that component works.*

*After having read this chapter, the reader should have a broad (but shallow) understanding of the entire platform.*

# Database

*Describe the resources in detail. What attributes do they consist of? Showing an ER diagram might be a good way to visualize the resources?*

*After having read this chapter, the reader should know how the data stored on the platform is structured and stored. If the reader is a new programmer that should start working on the platform, she should now know what she needs to know if she wants to change the resources or add more type of resources (e.g. know how to add a new table to the database with a relation to an existing table in the database).*

# REST API

*This chapter should contain the specification for the REST API on your platform. It should contain enough information so a new programmer can start using the REST API without reading through the code on the backend application. Be sure to mention all the details one need to know to use the REST API.*

# Backend Application

*This chapter should in detail describe how your backend application has been implemented. Which programming language have you used? Which framework have you used? How has the source code been structured? In multiple files? Which libraries/packages are you using? Etc.*

# Frontend Application

*This chapter should describe how your frontend application has been implemented in detail. Visualize the GUI in figures. In the beginning the figures can be simple drawings made with pen and paper, and in the end, you can also have screenshots of the application. Also mention the structure of your code, such as which Vue components you have in the "views" folder and which Vue components you have in the "components" folder, etc.*