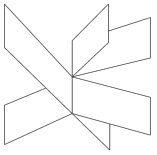


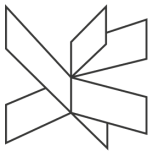
Reduction of Physical Supervision

Herning Vand

Amahdya Delkescamp	**256523**
Mihail Kanchev	**266106**
Dominika Kubicz	**266148**

Supervisor:

Poul Væggemose



VIA University College

herningvand

[Number of characters]

Software Engineering

7th Semester

18th December 2020

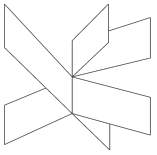


Table of content

Abstract iii

1 Introduction 1

2 Analysis 2

2.1 Requirements 2

2.2 Functional Requirements 2

2.3 Non-Functional Requirements 3

3 Design 4

4 Implementation 5

5 Test 6

5.1 Test Specifications 6

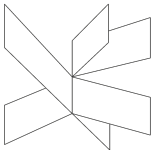
6 Results and Discussion 7

7 Conclusions 8

8 Project future 9

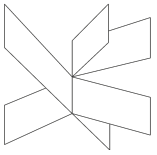
9 Sources of information 10

10 Appendices 1



List of figures and tables

Optional



Abstract

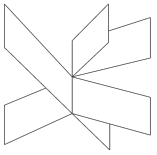
An abstract is a shortened version of the report and should contain all information necessary for the reader to determine:

1. *What are the aim and objectives of the project*
2. *What are the main technical choices*
3. *What are the results*

Frequently, readers of a report will only read the abstract, choosing to read at length those reports that are most interesting to them. For this reason, and because abstracts are frequently made available to engineers by various computer abstracting services, this section should be written carefully and succinctly to have the greatest impact in as few words as possible.

Although it appears as the first section in a paper, most report writers write the abstract section last.

Cf. (Dawson 2009, p.195).



Introduction

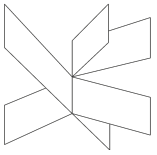
Water and sewage management and treatment, the impact on the environment and maintaining the consistency and reliability of such an important utility have posed a logistical challenge to society for as long as civilization has existed. Technology has increased to accomodate the need to properly manage waste water and countries, such as Denmark, have enjoyed the payoff. However, there are still some challenges that need to be overcome.

Water utility companies such as Herning Vand need to continually monitor the condition of 170 water pumping stations and 14 wastewater treatment plants across Denmark to ensure the consistent water quality and service. While the utility network is normally supervised with the help of a comprehensive SCADA system, there is still a need to check the water pumping stations manually to make sure that all devices in the pipes are working and to avoid potential breakdowns. The management at Herning Vand has noticed that such a method as a preventative measure against breakdowns within the system is not efficient or cost-effective.

Assisting Herning Vand with this problem will benefit not only the company itself but also the customers who rely on Herning Vand's services. This project will contribute to a way to help Herning Vand with efficiency along with reducing the possibility of an interruption in the water utility due to a breakdown that could have been prevented with more responsive monitoring.

(Sources to come)

Due to the complexity of the problem and the time allotted for the project, some limitations had to be set. The product to be completed will only be a proof of concept for the management of Herning Vand to consider and not a deployment ready system. Further this contributed to the delimitations of the project, what we will not consider.



Delimitations

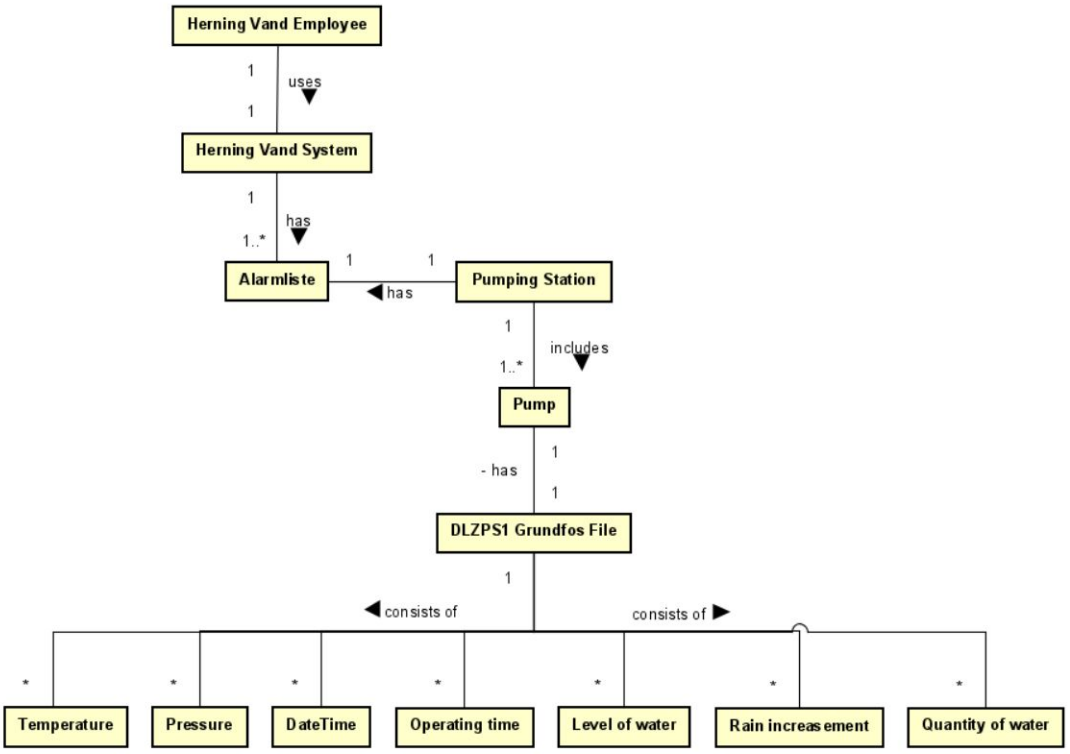
1. We will not consider video recording devices, such as video cameras, as a sensor to be monitored
2. We will not consider the general public as potential users
3. We will not consider the marketability of the system
4. We will not consider an internet-based use for the system
5. We will not consider any literal adaptation or implementation between the system and Herning Vand's current SCADA system
6. We will not consider additional actuators in the system

The proof of concept for Herning Vand will only be made for use by the company and will not be a fully fledged, marketable product. This also means that the system will not be allowed to connect or communicate with the SCADA system used by Herning Vand but should display the possibility for extension. There is also no need to further monitor existing recording devices, such as external cameras. Further, Herning Vand does not need any additional controls in the form of actuators as they already have the systems they want to use in place for such functionality.

The rest of the report to follow will contain details on the process that went into building the proof of concept for Herning Vand. The group worked according to Kanban and Unified Process and as such the sections below will detail work on the system from Inception and Requirements all the way to Transition and Testing.

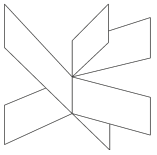
Analysis

Based on the background of the problem, it became clear what the management at Herning Vand was expecting out of this proof of concept. They wanted to see a system that could monitor the data from various sensors within the pumps at the pumping station and be able to alarm workers at Herning Vand when an anomaly is detected in the pipes or equipment. With this, Herning Vand would only dispatch crews when it is necessary instead as a persistent preventative measure. Having such a system eventually included into their SCADA would increase the efficiency of Herning Vand's ability to prevent breakdowns within their utility network and maintain as much uptime as possible.



(Domain Model Draft number 2)

(Further elaboration to come once the final model is agreed upon...)



1.1 Requirements

Employee

There will not be any level of access necessary for the system due to the understanding that the system would exist as a part of Herning Vand's SCADA, therefore security is assumed to be handled elsewhere. This means that all users of the system are assumed to be Employees of Herning Vand. The system is also not being considered as a marketable product and no further customers will be identified.

1.2 Functional Requirements

1. Forecast potential breakdowns
2. Store and display continuous sensor signals
3. Consume data in .csv format
4. Expose a CRUD based API

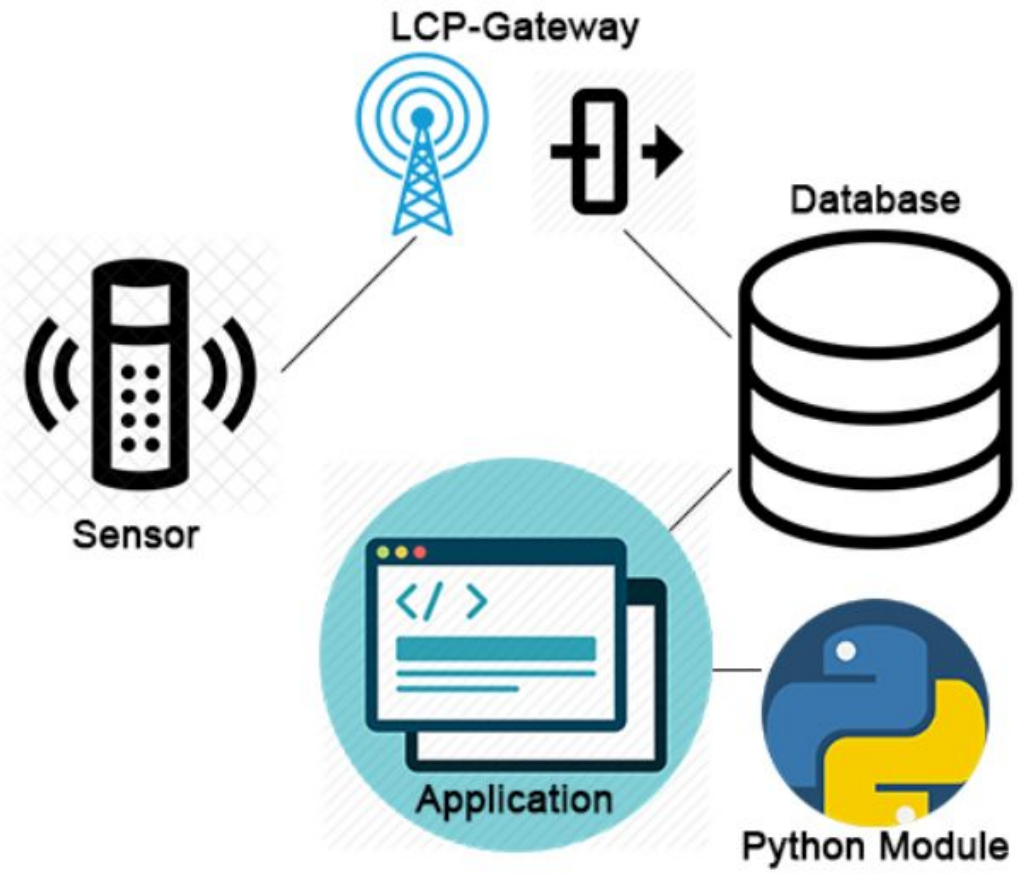
1.3 Non-Functional Requirements

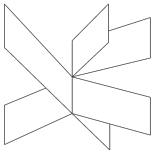
1. Has a single user interface panel
2. Is deployed locally
3. Does not access the internet (needs elaboration)

Design

The purpose of the design section is to outline HOW the system is structured; i.e. to transform the artefacts of the analysis into a model that can be implemented. The design section is relevant for the programmer, whereas the analysis is relevant for the stakeholder.
Elements that may be relevant in this section:

- Architecture: Find architecture patterns here (Leszek Maciaszek 2004, chap.9).





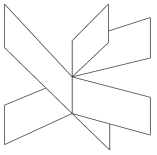
- Technologies: Describe technologies used, also alternative technologies. Argue for choice of technology according to the project aim.

ASP.NET Core Blazor

Python Module

- Design Patterns: Describe which design patterns (GoF (Gamma et al. 2002) etc.) you are using and why.
- Class Diagrams
- Interaction Diagrams
- UI design choices
- Data models, persistence, etc.

You must explain all diagrams in the report. These diagrams including descriptions are the blueprints for the implementation.
Hint: One way to figure out which objects/classes are needed in the design is to apply the General Responsibility Assignment Software Patterns/principles (GRASP) (Larman 2004, chap.17).
Hint: Consider how to design your system to make it testable.



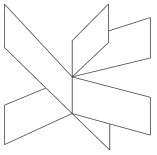
Implementation

The purpose of the implementation section is to explain interesting code snippets. An idea is to explain the complete path through your system from UI to database etc.

Remember that your implementation must be consistent with your design (Larman 2004, chap.20).

Which standard libraries are used? How are design patterns implemented, etc.

Hint: Implement your code in a testable manner.



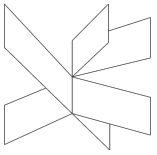
Test

The purpose of the test section is to document the result of your testing; to verify if the content of the requirements section has been fulfilled. How is the system tested, which strategy has been used; e.g. White Box (Unit Test), Black Box, etc.

1.4 Test Specifications

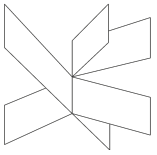
For functional requirements, test specifications must be listed. These test specifications can be described as soon as the functional requirements have been completed (Use Cases including descriptions).

IEEE can be used as a template for test specification (IEEE Computer Society 2008). VIA Library can give you access to this standard.



Results and Discussion

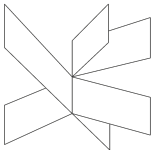
The purpose of the results and discussion section is to present the outcome and achieved results of the project.



Conclusions

The purpose of the conclusion section is to compile the results from each section in the report. What is the conclusion? Did the project fulfil the requirements? Etc.

You can only comment on report contents, no new topics or content can be introduced in this section.



Project future

Reflect on your project from a technical viewpoint and describe what you would change if you could.

Suggest how the project could be improved or made ready for production. Discuss scalability, suggest possible spin offs, what is needed, missing, etc.?

Sources of information

Note: Use the standard reference method: Harvard Anglia. A very good reference tool is Mendeley (Mendeley.com 2016), **ask VIA Library if you need help.**

Banger, D., 2014. A Basic Non-Functional Requirements Checklist « Thoughts from the Systems front line.... Available at: https://dalbanger.wordpress.com/2014/01/08/a-basic-non-functional-requirements-checklist/ [Accessed January 31, 2017].

Business Analyst Learnings, 2013. MoSCoW : Requirements Prioritization Technique — Business Analyst Learnings. , pp.1–5. Available at: https://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization [Accessed January 31, 2017].

Dawson, C.W., 2009. *Projects in Computing and Information Systems*, Available at: http://www.sentimentaltoday.net/National_Academy_Press/0321263553.Addison.Wesley.Publishing.Company.Projects.in.Computing.and.Information.Systems.A.Students.Guide.Jun.2005.pdf.

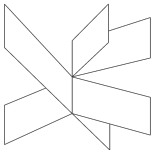
Gamma, E. et al., 2002. *Design Patterns – Elements of Reusable Object-Oriented Software*, Available at: http://books.google.com/books?id=JPOaP7cyk6wC&pg=PA78&dq=intitle:Design+Patterns+Elements+of+Reusable+Object+Oriented+Software&hl=&cd=3&source=gbs_api%5Cnpapers2://publication/uuid/944613AA-7124-44A4-B86F-C7B2123344F3.

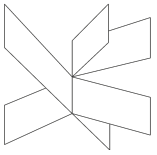
IEEE Computer Society, 2008. *IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation*,

Larman, C., 2004. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*,

Mendeley.com, 2016. Homepage | Mendeley. Available at: https://www.mendeley.com/ [Accessed February 2, 2017].

YourCoach, S.M.A.R.T. goal setting | SMART | Coaching tools | YourCoach Gent. Available at: http://www.yourcoach.be/en/coaching-tools/smart-goal-setting.php [Accessed August 19, 2017].





Appendices

The purpose of your appendices is to provide extra information to the expert reader.

List the appendices in order of mention.

Examples of appendices

- Project Description
- User Guide
- Source code – source documentation
- Diagrams
- Data sheets
- Etc.

Appendix A Project Description

Insert the original Project Description here