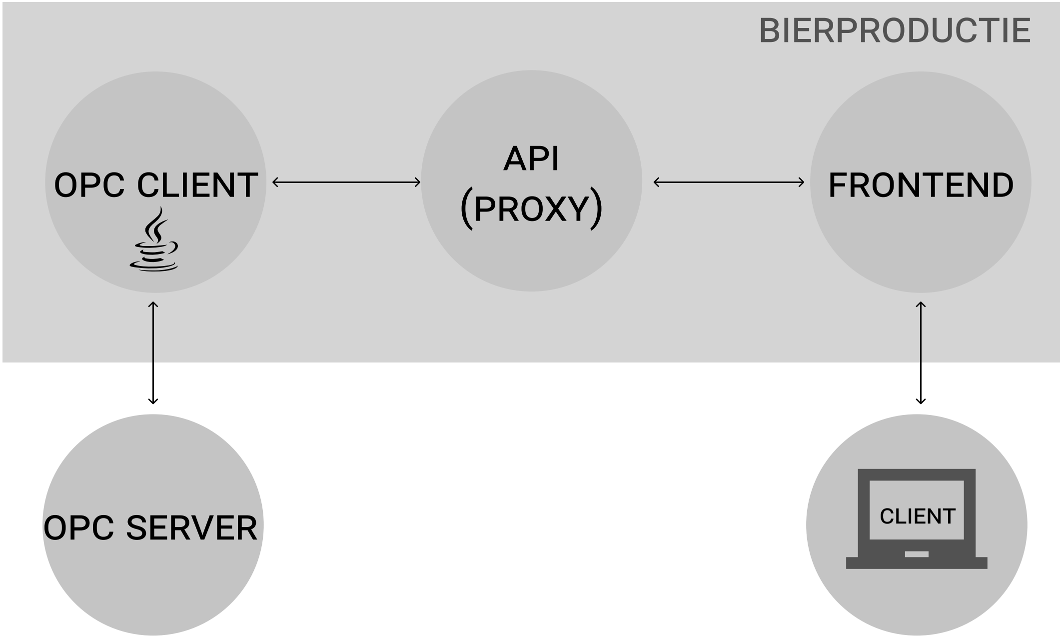
Bierproductie

A management system for brewing machines



Bachelor of Engineering, Software Technology

Semesterproject 3. semester, ST3-PRO

Project Period: 31.08.2020 - 19.12.2020

Hand in date: 19.12.2020

Group 06:

Jakob Rasmussen, jakra19@student.sdu.dk

Kenneth M. Christiansen kechr19@student.sdu.dk

Kevin K. M. Petersen, kepet19@student.sdu.dk

Kristian N. Jakobsen, kjako19@student.sdu.dk

Simon Jørgensen, sijo819@student.sdu.dk

Supervisor: Parisa Niloofar, parni@mmmi.sdu.dk

University of Southern Denmark
The Faculty of Engineering
The Mærsk Mc-Kinney Møller Institute
Campusvej 55, 5230 Odense M

1 Background

The brewery Refslevbæk Bryghus A/S recently won the “best beer of the year” award, which has led to a boost in demand for their products. However, their current production line does not fulfil these demands, and so they have purchased a new, bigger brewing machine. So far, the brewing has involved a lot of manual processes. The brewery now needs a new brewing system developed, that implements a large degree of automation, and enables it to produce beer most efficiently.

2 Motivation

As students, this semester project gives the group great learning experience as well as a good idea of how such a project would progress. During the project, the group will learn how to create and set up an Open Platform Communications Unified Architecture, OPC-UA, client, and how to make it communicate directly to a physical machine. The group will learn to host a web server, from which the machine can be controlled, and access it as a website. The group will also have the chance to learn and use a new scripting language, JavaScript. The gained experience is the main motivation of this project.

3 Aim

This project aims to improve the beer machine by increasing quantity while maintaining quality. To accomplish this the project will offer an interactive web interface for monitoring, controlling and adjusting the brewery machine. This is done by having the web interface interact with a REST API that connects to a client. This client controls the machine through an OPC-UA server connection and stores relevant data in a database.

4 Solutions

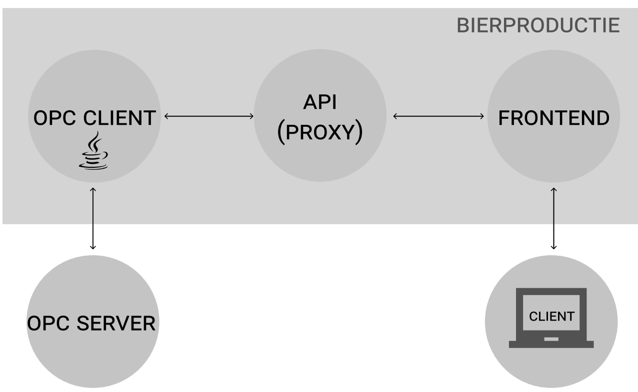


Figure 1: System drawing

The distributed system consists of five sub-systems, where the group are writing three of them.

4.1 OPC-UA Client

The OPC-UA client is responsible for communicating with the brewing machine via the OPC-UA server. This is needed to send and receive data via commands.

4.2 Website

The website user interface is where the client will be able to interact with our system. The website will enable the user to control, monitor and adjust the brewing machine. Live relevant data will also be collected, displayed and can be downloaded as a report.

4.3 REST API

The REST API functions primarily as a gateway in this design. For the front-end to receive data via fetch calls in JavaScript, it needs to communicate with something that uses HTTP and preferably delivers data in JSON.

5 Objectives

The group wants to create a dashboard solution, that gets the necessary information from the brewery machine to display information about the current - and previous batches.
The group wants to host the REST API and Website on a Linux based server on a fully qualified domain name using the container technology Docker which should be configured via Docker Compose. Furthermore, the group aims to set up a continuous integration pipeline for development via GitHub Actions (makes it easy to automate software workflows) and continuous delivery via Watchtower (an automatic deployment tool).

6 Methods

To keep track of the group work, the Scrum framework, with a few buts, is used. Scrum consists of multiple artefacts and ceremonies, and in this project product roadmap, Scrum meetings, product and sprint backlogs and burndown charts will be used. Each sprint will have a duration of two weeks and the issues for the sprint backlog will be chosen at the beginning of each sprint, at the Scrum meeting. The Scrum meeting will take place each Friday and will be a mixture of sprint planning, daily Scrum, and sprint review. To manage Scrum, ZenHub, a management solution that can be integrated with GitHub, is used. On ZenHub the group will create the roadmap, that acts as a schedule for the project and board, which is where the issues will be handled. A burndown chart for each sprint is automatically generated and kept up to date by ZenHub. The board will consist of five columns, as seen below.

Product Backlog	**Sprint Backlog**	**In Progress**	**Review/QA**	**Closed**
Issues	Issues for the given sprint	Issues that is currently being worked on	Issues that are pending (or in) review	Approved issues that have been merged

Table 1: Methods

7 Initial Requirement Analysis

7.1 Problem Statement

In the table 2, the finished problem, the problem statement and related questions are listed.

Problem	The current production line is not effecient enough to keep up with the demand of the beer, while still maintaining a quality product
Problem Statement	How to control and optimise the brewing machine, to maximise the production of high quality beer
Related questions	• How can we optimise the production? • How can we utilise calculus and linear algebra to provide a meaningful overview of the production line, based on statistics? • How can we create a web based frontend for the MES? • How can we separate the different aspects of the system (separation of concerns)

Table 2: Problem statement showcase

7.2 Summary of requirements

The group's proposed solution will adhere to the requirements given by the brewery Refslevbæk Bryghus A/S.
The manufacturing execution system, MES, must be able to control the brewery's production. It must be able to start and stop the production line, as well as monitor the production and collect data from the production line. The data must be stored for further analysis. The MES must be able to keep track of the batches that the new machine is producing, as well as collect various data from the machine that is associated with the current batch number. After a finished batch production, the MES must be able to produce a batch report. The report must contain the following.

- This Batch ID
- Product type
- Amount of products (total, defect and acceptable)
- Amount of time used in the different states
- Logging of temperature over the production time
- Logging of humidity over the production time

The MES/SCADA (Supervisory control and data acquisition) system must be able to monitor the production and display live relevant data from the machine. The documentation of the system must contain an illustration that defines the different components in the setup, in relation to the ISA88ˆ1213 Part 1 Physical Hierarchy model. The system must have a visualisation that can be accessed and used to display production data. The system must be able to collect the necessary data from the machine and calculate the overall equipment effectiveness, OEEˆ131516, of the machine. The OEE must be available to be displayed by the system. The system must be able to estimate the error function associated with the different products. The system must be able to find the optimal production speed for each product type, based on an error simulation and the appertaining graph upon which the error simulation is built.

7.3 List of requirements

Below is a list of the above requirements. These requirements have been prioritised using the MoSCoW method, where M is for Must have, S is for Should have, C is for could have, and W is for Won't have.

ID	Name	Description	Prio
R01	Control production line	Control the brewery's production	M
R02	Control production line	Start/stop production line	M
R03	Monitor production	Monitor data from the production line	M
R04	Monitor production	Store the collected data for further analysis	M
R05	Administer batches	Keep track of produced batches (batch ID)	M
R06	Store batch info	Collect various data associated with current batch number from the machine	M
R07	Batch report	Produce a batch report (PDF/dashboard style format)	M
R08	Live data	Monitor and display live relevant data from the machine	M
R09	Documentation	Documentation must contain an illustration that defines the different components in the setup in relation to the ISA88ˆ1213 Part 1 Physical Hierarchy model	M
R10	Visualisation	Visualisation that can be accessed and used to display the production data	M
R11	OEE	Collect necessary data from the machine and calculate the OEE. OEE must be available to be displayed by the system	M
R12	Estimate error function	Estimate the error function associated with the products	S
R13	Optimal Production speed	Estimate the optimal production speed for each product type	M

Table 3: List of requirements

7.4 Use Case Diagram

As seen below, the user (primary actor) of the system can interact with the MES. The green ellipses represent the different use cases in the system, e.g. Start/Stop production line, which enables the user to start or stop the production.

On the left side of the diagram, the SCADA actor is shown, which is a secondary actor in Bierproductie. The SCADA actor is responsible for making the beer production machines endpoints available for the MES.

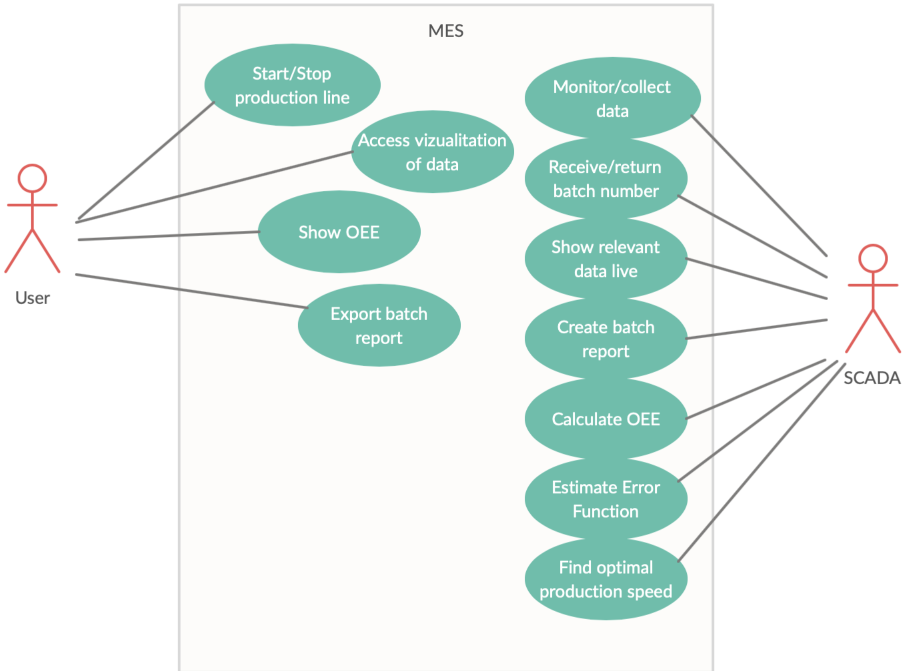


Figure 2: Use Case Diagram

7.5 Supplementary Requirements

FURPS is a model for classifying functional and non-functional requirements. The acronym stands for functionality, usability, reliability, performance, and supportability. The group have used FURPS to classify the following:

FURPS	**Demands**
Functionality	
Usability	Documentation on usage of the REST API
Reliability	On server reboot, the application will automatically restart
Performance	Max response time (API: 400 ms)
Supportability	Minimum browser versions

Table 4: Supplementary Requirements

8 Stakeholders

For this project, there is no stakeholder, as the group does not have the opportunity to communicate with Reflevsbæk. This is a problem when developing using Scrum, since this role is usually filled by a Product owner (abbreviated to PO from here on out). Instead, the group has elected Kristian Nymann Jakobsen as an internal PO. The PO should under normal circumstances be one from the customers firm, and their job is to choose which features (user stories) that they would like developed next.

9 Project Organisation

In this project, work is distributed from planning meetings. These planning meetings take place at the beginning of each sprint, where the workload is estimated and distributed across the team members. Kristian Nymann Jakobsen was chosen to be the Product Owner, so he oversees the "what", with a focus on value, time to market etc. Kenneth Munk Christiansen was chosen to be the Scrum Master, so he ensures that the Scrum framework is understood and is used correctly throughout the project. Like the rest of the group, they are also a part of the Development Team. For easy version control and collaboration, GitHub will be used. GitHub is a hosting service with a centralised repository, built around git, a version control system that allows each team member to download the code, modify it and upload the new version. For each issue is created, a corresponding branch will automatically be made. When development is deemed finished, the developer opens a merge request, where the code in questions will be subject to review. The goal of this process is to improve the quality and readability of the source code.

10 Project Plan

To ensure that it is feasible to complete the project within the designated time frame, a project plan has been made with inspiration from the project plan supplied by the supervisor.

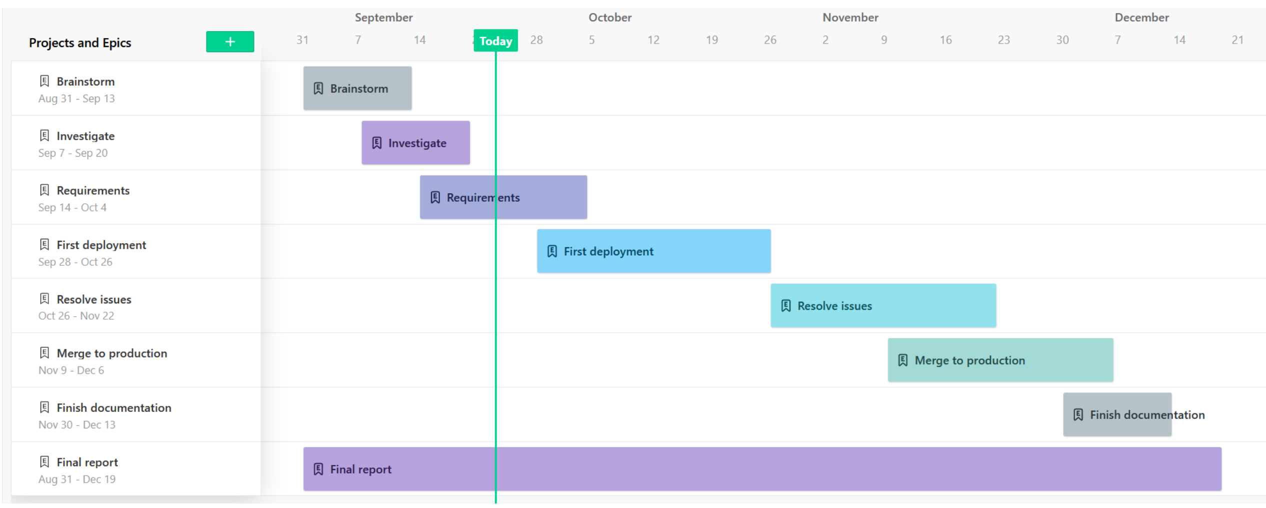


Figure 3: Project Plan

In this, it is shown when the different milestones must be completed to succeed with the project. As mentioned above, sprints with a duration of two weeks will be used throughout this project. Each sprint consists of issues from the product backlog and is chosen at the beginning of each sprint, as described in section Methods. To keep track of these sprints, a burndown chart will be made. This ensures the group that each sprint is completed, and if not, it is easy to see where it went wrong, and the remaining issues can be transferred to the next sprint.

11 Risk

The group foresees three risks with varying impact levels.

Risks	**Assessment**	**Impact**	**Handling**	**Critical**
Covid-19	High	High	With regulations concerning social distancing and the risk of being infected with Covid-19, most work will take place from home. To avoid complications, expectations to finish delegated work to each group member will be raised. Each member is also expected to follow the lectures. The supervisor will be available for contact via e-mail, if questions should arise.	Yes
Schedule	High	High	If the group falls behind schedule, the delegated work tasks will be evaluated.	Yes
Lost team member	Medium	Low	Since the group has already lost a member, the group is currently undermanned. To stay on top of the semester project, expectations to finish delegated work to each member will be raised.	Yes

Table 5: Risks

12 Milestones

12.1 About milestones

This is our milestones and activities. *Activities* will include scrum planning and meeting. It will also show some key activities from the university, that is mandatory and at last, it will also show when the project needs to be handed in. *Milestones* are the overall structure of the project. Milestones show the different phases the group will go through.

Phase	Activities	Start Date	End Date	week
Brainstorm	Scrum planning (1)	28-08-20		35
	kickoff	01-09-20		36
Investigate	Scrum meeting	04-09-20		37
Requirements	Scrum planning (2)	11-09-20		38
	Scrum meeting	18-09-20		39
First deployment	Scrum planning (3)	25-09-20		40
	Team leader meeting	25-09-20		
	Project draft	02-10-20		
	Peer feedback start	05-10-20	09-10-20	41
	Scrum planning (4)	09-10-20		
	Final project foundation	18-10-20		42
	Scrum planning (5)	23-10-20		43
	Team leader meeting	09-10-20		
	Project week	18-10-20	24-10-20	
Resolve issues	Scrum meeting	30-10-20		44
	Code review	30-10-20		
	Scrum planning (6)	06-11-20		45
	Scrum meeting	13-11-20		46
Merge to production	Scrum planning (7)	20-11-20		47
	Scrum meeting	27-11-20		48
Finish documentation	Scrum planning (8)	04-12-20		49
	Scrum meeting	11-12-20		50
	Scrum meeting	18-11-20		51
	Final report	02-12-20		

Table 6: Milestones

13 Tentativ outline

The estimated outline of the project report is as follows;

I Cover(title, information, signatures)

II Summary

III Preface

IV Table of Contents

V Editorial(contribution table)

VI List of figures

13.1 Main report

1 Introduction

(a) Motivation?

(b) Problem formulation

2. Background
   (a) Review of relevant literature and other background information
3. Problem analysis
   (a) Use-case analysis
   (b) Use-case realisation
4. Theory and Methods
   (a) Theory
   (b) Methods
5. Requirements
   (a) Overall requirements specification
   (b) Selected detailed requirements
   (c) Detailed use-cases
   (d) Functional and non-function requirements
   (e) The physical setup (brewing machine)
   (f) Description of the simulator (mathematical?)
6. Architecture (System architecture)
7. Design
   (a) Description of specific parts of the system (what's important and interesting)
8. Implementation (description of technically complicated parts of the system)
9. Verification and validation (verify, simulate and test that the implemented system fulfils the requirements)
10. Evaluation (Evaluation of the developed product from a user/customer point of view)
11. Conclusion
12. References
13. Appendix (all technical details that are not essential to understanding the report)