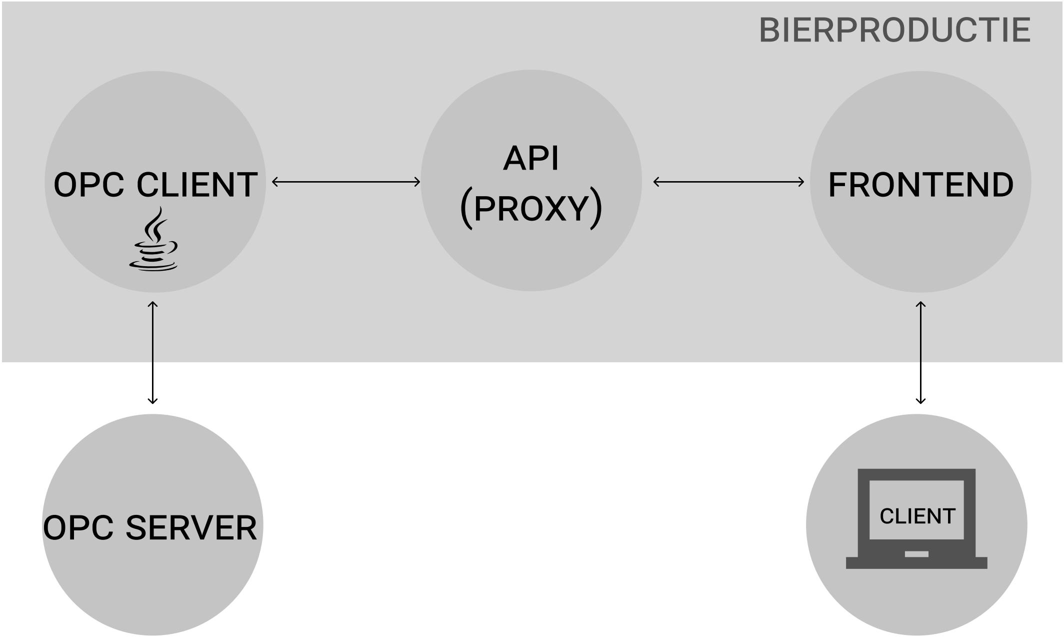
Bierproductie

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

Title:	Bierproductie
Institution:	University of Southern Denmark The Faculty of Engineering, The Mærsk Mc-Kinney Møller Institute Campusvej 55, 5230 Odense M
Education:	Bachelor of Engineering, Software Technology
Semester:	3. Semester
Course Title:	Industrial 4.0 cyber-physical software systems
Internal Course Code:	ST3-PRO
Project Period:	31.08.2020 - 19.12.2020
ECTS:	10 ECTS
Supervisor:	Parisa Niloofar
Project group:	06

Jakob Rasmussen, jakra19@student.sdu.dk

Kenneth Munk

Kenneth M. Christiansen, kechr19@student.sdu.dk

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

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

Simon

Simon Jørgensen, sijo819@student.sdu.dk

Pages: 10
Appendix: 0

By signing this document, each group member confirms that everyone have participated equally to this project, and everyone is thus collectively responsible for the content of the report.

I Summary

II Table of Contents

III Editorial

IV List of Figures

1 Introduction

2 Background

3 Problem analysis

4 Theory & Methods

5 Requirements

5.1 Overall Requirements Specification

5.2 Selected Detailed Requirements

5.2.1 Functional & Non-Functional Requirements

5.2.2 The Physical Setup (The Brewery Machine)

5.2.3 The Simulator

5.3 Use Cases

5.3.1 Actor List

5.3.2 Detailed Use Cases

From project description

5.3.3 Use Case Diagram

6 Analysis

6.1 Use Case analysis

6.1.1 Class Candidates

In order to find potential class candidates, every noun of the detailed Use Cases are found. These are potential candidates, and can be sorted to avoid duplicates and candidates that won't be turned into classes. Naturally, every potential class for the entire system will not be found, as this only reflects use cases. A potential class candidate such as MES (where Start and Stop functionality would otherwise be implemented) will not be reduced to a single class and is therefore not added to the list of class candidates.
The final list of classes, as well as a description of them, can be seen in table 1.

Class Candidate	**Attributes**	**Definition**
Batch	Id, type, product_amount (total, defect, acceptable), amount (time), state (current, history), OEE, production_speed,	A batch refers to a specific batch of products the brewery has made
Product	Id, type, Ingredients,	Product refers to the different options of beer to be produced
Ingredient	Name, id	An ingredient refers to a specific ingredient. Products contain a list of ingredients.

Table 1: Potential class candidates

6.1.2 UML Analysis Diagram

From the verb/noun analysis from the previous chapter, the UML analysis diagram seen in figure 1, can be generated. This diagram shows the classes and attributes found in the requirements from the project description.

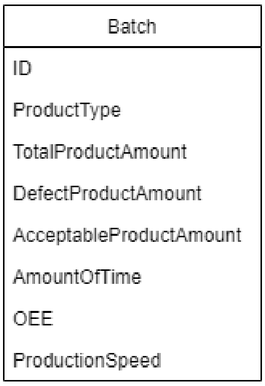


Figure 1: UML Analysis diagram

6.2 Use Case Realisation

6.2.1 Sequence Diagrams

6.2.2 Operation Contracts

6.2.3 Updated UML Class Diagram

7 Architecture

8 Design

9 Implementation

10 Verification & Validation

11 Evaluation

12 conclusion