NTNU

Norwegian University of
Science and Technology

Optimization of Energy Consumption

Authors

Eilert Mikal Hananger Tunheim

Karin Emilie Pettersen

Mads Greni Arnesen

in colaboration with

Moelven

Bachelor of Engineering in Computer Science (BIDATA),
Faculty of Informationtechnology and Electrical Engineering (IE),
Norwegian University of Science and Technology (NTNU),
20.05.2022

Abstract

Drying wood on an industrial scale is a time-consuming and energy intensive process. In Norway, one of the companies that focuses on wood drying and other wood-related production and services is Moelven. Their current solution for drying wood is based on historical work experience, which often leads to drying the wood more than necessary and presents the opportunity for optimization. In this project, we have developed an application with the goal of optimizing the energy consumption during the wood drying process. The optimization has been achieved by visualizing data from previous drying cycles, and calculating an estimated growth model based on historical data. The application acts as a technology demonstrator in a field where IT is generally underdeveloped. For this reason, the application has been implemented with future improvements in mind, with the framework being flexible and easily maintainable. The results highlight the benefits of analysing stored energy data, which can optimize and streamline the drying process.

For access to our GitLab repository, please follow this link to: Gitlab.com.

Sammendrag

Moelven ASA tørker trevirke på en industriell skala hvor tørkeprosessen er en tids- og energikrevende prosess. Moelvens nåværende løsning er basert på historisk arbeidserfaring, dette fører ofte til at treverket tørkes mer enn nødvendig og åpner opp for optimalisering. Resultatet av prosjektet er en applikasjon som har som mål å optimalisere energiforbruket ved å visualisere data fra tidligere tørkesykluser, og beregne en estimert vekstmodell basert på historisk data. Applikasjonen fungerer som en teknologidemonstrasjon i et felt der IT generelt er underutviklet. En konsekvens av dette er at applikasjonen har blitt implementert med fremtidige forbedringer i tankene, der rammeverket er fleksibelt og enkelt å vedlikeholde. Resultatene fremhever fordelene av å analysere lagrede energidata for å optimalisere og effektivisere tørkeprosessen.

For tilgang til GitLab repository, følg vedlagt link Gitlab.com.

Preface

The research presented in this bachelor thesis was written by Eilert Mikal Hananger Tunheim, Karin Emilie Pettersen and Mads Greni Arnesen, students at the department of Computer Science at NTNU in Gjøvik. First we would like to thank Moelven, and in particular Stefan Djupvik, Ylva Kleiven, Leif Torbjørn Næswold and Lisa Nilsson for an exciting and educational project. In addition, we would also like to thank them for a close cooperation and contribution of feedback during the project period. We appreciate the work experience and learning outcome from this project, and we hope the results can contribute value for Moelven.

We would also like to thank our supervisor Tom Røise for providing encouragement and valuable guidance throughout the project. In addition, we would also thank other lectures and professors at NTNU for providing assistance throughout the project. In this context, we would like to especially thank Ivar Farup for his assistance.

Finally, we would like to thank family and friends for support and proofreading the thesis. We would also like to thank each other for a good collaboration and work effort during the project period.

Table of Contents

List of Figures

Acronyms

ANSI American National Standards Institute. The international standard for querying databases to formulate and perform operations.. 42

API Application Programming Interface. 38

AWS Amazon Web Services. 41

DBMS Database Management System. 38

GCP Google Cloud Platform. 41, 42, 44, 54, 93, 94, 96, 98, 100

GCS Google Cloud Storage. 41, 42

GUI Graphical user interface. 14, 38, 39, 45, 47, 48, 52, 54, 81, 82, 86, 87, 93, 97, 99, 101

IoT Internet of Things. 1

JDBC Java Database Connectivity. 38, 39

JVM Java Virtual Machine. 38

kWh Kilowatt Hours. viii–xii, 2, 4, 31, 33–35, 52, 58–60, 63, 65, 66, 68, 71, 72, 83–85

LTS Long Term Supported. 39

ODBC Open Database Connectivity. 39

OOP Object-Oriented Programming. 38, 39

UI User Interface. 48

VCS Version Control System. 43

Glossary

Big data Large complex data sets that can be hard to process with traditional data-processing software. 1, 41, 42

lean Lean development, agile framework for optimizing development resources and time. 16

lumber typically refers collectively to wood that's been processed for use as a building material—wood that has been milled and cut into boards or planks. *"Timber" vs. "Lumber": How To Cut Out The Confusion Between Them* 2021. 1, 24–29

over-drying Sawn timber being dried too long to make sure that the wood is not moist. 1, 21, 98, 104

RESTful REST architecture based web service. Architectural style that uses HTTP requests to access and use data. The architecture is light weight, highly scalable and maintainable.. 42

saw pattern Same as "Saw set", the dividing pattern of the center block . ix, 26

timber the wood of trees that can or will be used for building material. Timber can refer to living, standing trees themselves or to trees that have been cut down but not yet processed (meaning they have not yet been milled or cut into planks). ibid.. 26, 28, 29, 31

1 Introduction

1.1 Background

The drying process of wood is a time consuming and energy-intensive process. The drying-process decides the quality and material strength of the wood-based products of Moelven. Today, sawn lumber is dried in large industrial kilns with predefined climate programs based on input data for tree species, dimension, humidity goal, and heartwood proportion. For many years, sawmills have been operated by traditions based on previous experience. Therefore, in order to make the drying processes more sustainable, technology has been implemented to improve drying accuracy. The technologies used to digitize these sawmills include IoT, sensors, cutting-edge technology, Big data, and new analysis methods. Moelven dries lumber towards the desired moisture level, intending to improve product quality. Today, Moelven dries lumber longer than required to ensure it is not too moist. Customers are more sensitive to the products being too moist than too dry, which results in them over-drying the products. This is not optimal because over-drying results in reduced capacity for their dryers and increased thermal energy consumption, as well as degraded product quality. In recent years Moelven has collected energy data from a few of their drying kilns. However, this data has not yet been used for analytic purposes. To reduce over-drying, in this work, we propose an application that uses energy data to estimate the duration of a drying cycle, thus decreasing energy consumption. This solution would provide a novel method to create an estimate of the moisture by utilizing energy data. The solution has been developed in collaboration with the partners at Moelven, who provided assistance and guidance within their field. The solution has the potential to be revolutionary by making the drying process more profitable, sustainable, and optimized. Today's solution at Moelven is a monitoring system that does not utilize the stored *energy-data*.

Figure 1: Example image showing the drying kiln used by Moelven in order to dry wood (*Moose River* 2019)

Figure 2: Example image showing the drying kiln used by Moelven in order to dry wood (*FB continuous kiln* 2017)

1.1.1 Monitoring System At Moelven

The sawmills contain sophisticated monitoring systems that include numerous parameters concerning the drying kiln and ongoing cycle. These systems have custom drying programs that tell when the desired moisture of a cycle is reached. The drying programs are based on previous experience and are static in the sense that they do not adapt to each cycle. The final solution of this project is not supposed to be a substitute for their existing monitor program but rather a complementary tool for processing and monitoring energy data, which has never been done at Moelven before. This energy data processing results in new estimations that make it possible to analyze the correlation between energy consumption and moist level and estimate the end time of a drying cycle.

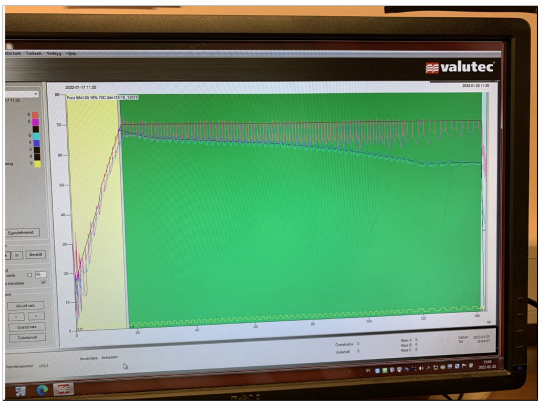


Figure 3: The current monitoring system in use created by Valutec. The x-axis showin the time-axis, and the y-axis showing the kWh usage. The image was given to the project group by Moelven.

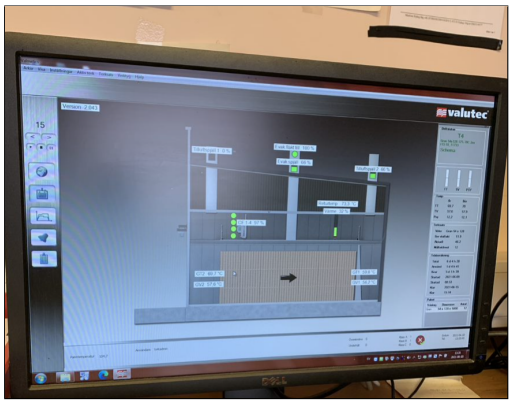


Figure 4: The current monitoring system in use created by Valutec. This figure visualize the drying kiln and what functionality is currently in use. The arrow displays the airflow-direction in order to circulate the air, and different sensors display their data, E.g temperature data.

1.2 Subject Area

The objective of Moelven for the subject area is to optimize their drying process, and in order to achieve this goal, an estimate based on previous data was desirable. The estimates had to be displayed through a graphical interface. Our solution required a substantial amount of database interaction, analysis, data processing, software development, and software security. To be able to produce these estimates, we had to import data, categorize data and use statistics to calculate the estimates. After the estimates had been calculated, they needed to be displayed in an intuitive and understandable manner for the operators at Moelven to see. The visualization of these estimates

required the use of several software development techniques.

1.3 Boundaries

The application developed will be ready for production but not yet fully optimized due to a lack of data. The solution will improve over time and give Moelven better estimates as the time progresses. The graphical interface will be international but not optimized for each language. This interface needs to be simple and intuitive for the operators at Moelven to understand. For this thesis, we will not be deep-diving into the laws of thermodynamics and drying theories but instead get a general perspective to understand the process, the database content, and connections. The application has to work on independent workstations running the program through a cloud-based deployment tool. Another limitation of this project is the limited time at hand. This time limitation decides our development and report course and can affect this project by forcing us to prioritize some processes over others.

1.4 Project Description

The desired goal of this project is to develop an application that can predict and visualize the drying process with the help of energy measurements and other available production data, with the final goal of reducing drying time and energy consumption. Data stored by Moelven regarding the energy consumption of moisture is used in combination with thermodynamics principles and statistics to create the estimate which predicts and visualizes the drying process. This estimate will indirectly tell the moisture level based on previous drying periods. Since drying of sawn lumber is a process that is largely determined by thermodynamic principles, it should be possible to use specific thermal energy as a measure of how much water has been removed from the wood. The application will work as a data analysis tool and will run on cloud-based deployment and retrieve data from their Google Cloud storage environment.

Row	KilnName	DryingStarted	DryingCompleted	Name
1	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
2	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
3	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
4	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
5	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
6	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
7	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
8	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
9	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
10	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
11	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
12	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359
13	5	2021-04-29T06:00:48	0001-01-01T00:00:00	Fura(5272)(2;6) 50x250 3ex 16pkt Vår 18%_10359

Figure 5: Drying cycle data example, extracted from Google Cloud. Each individual row refers to an unique drying cycle. The database contains data regarding start- & stop date/-time, aswell as parameters categorizing each cycle

Row	TimeStamp	VariantValue
1	2021-12-27 06:32:31 UTC	5463496.5
2	2021-12-27 06:32:31 UTC	5463496.5
3	2021-12-27 00:52:31 UTC	5463476.0
4	2021-12-27 00:52:31 UTC	5463476.0
5	2021-12-27 02:02:31 UTC	5463480.0
6	2021-12-27 02:02:31 UTC	5463480.0
7	2021-12-27 03:02:31 UTC	5463484.0
8	2021-12-27 03:02:31 UTC	5463484.0
9	2021-12-27 05:12:31 UTC	5463492.0
10	2021-12-27 05:12:31 UTC	5463492.0
11	2021-12-27 06:12:31 UTC	5463496.0
12	2021-12-27 06:12:31 UTC	5463496.0

Figure 6: Drying cycle data example, extracted from Google Cloud. Each individual row refers to an unique drying cycle. The database contains data regarding timestamp and the corresponding kWh data.

1.5 Target Audience

1.5.1 Application Audience

The application is primarily targeted at the employees at Moelven. This will be used by them to control the drying process and to visualize the data. The employees are assumed to be non-technical end-users. As such, it is essential to design an intuitive and receptive application.

1.5.2 Report & Code Audience

The agenda of the report and code is to inspire and educate the target audience. The report will primarily target the examiner, our supervisor, and fellow students and developers who can use this report as inspiration or learning material for their own projects. The report will also be highly relevant for the employees of Moelven. They will use the report to gain a better understanding of the application and code in order to develop further.

1.6 Our Background

The group consists of three bachelors students in the computer engineering program located in NTNU Gjøvik. All members have the same base knowledge from the mandatory subjects in the education program. However, some electives separate our knowledge. Two of the members have more profound knowledge of mathematics, and one of the members has extensive knowledge of

graphical interfaces.

1.7 Roles

There is a set of essential responsibilities regarding the project. All group members are developers and report writers to gain multiple perspectives and maximize the learning outcome. The roles assigned to each member are intended as a general leader role in their respective fields. In addition, the group as a unit is responsible for operating the environments around this project.

- **Project Supervisor - Lecturer Tom Røise**
  Provide guidance and supervision throughout the project, thanks to his prior experience and knowledge
- **Project Owner - Moelven, Ylva Kleiven, Stefan Djupvik**
  Represent the challenges and share Moelven's vision of the project through scheduled meetings.
- **Project Leader - Eilert Mikal Hananger Tunheim**
  Has the overall responsibility for the project and to manage the project team. Primarily responsible for meeting the deadlines that have been set and the time limits we have set for ourselves.
- **Communications - Karin Emilie Pettersen**
  Responsible for communication, arranging the meetings and keeping in touch with our contacts trough email and Microsoft Teams.
- **Documentation - Mads Greni Arnesen**
  Responsible for documenting and following up all meetings, and structure these notes so it is easy to access and understand for all group members.

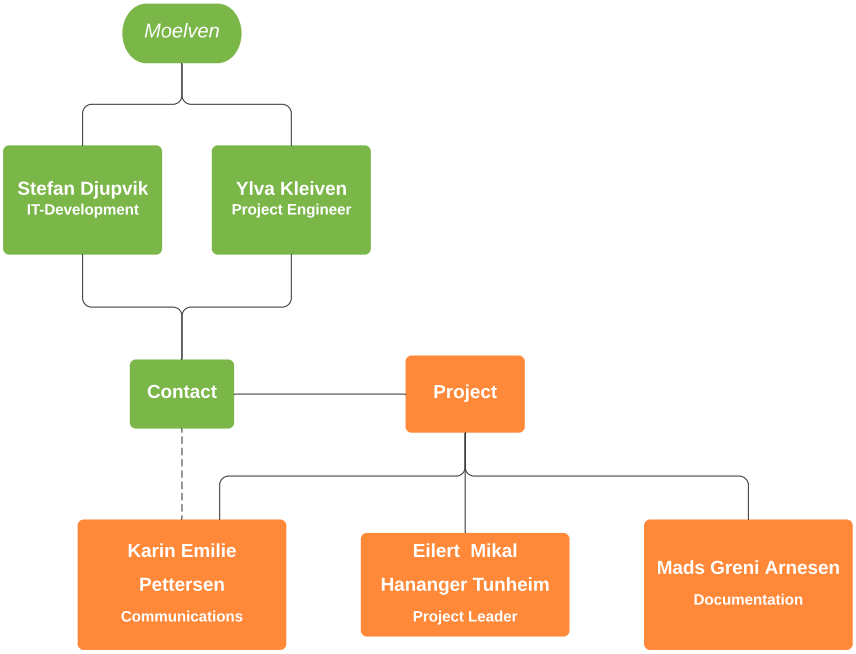


Figure 7: Chart of the individuals involved in the project and their respective roles. Created in Photoshop

1.8 Report Structure

- **Chapter 1: Introduction**
  Introduction of the project, the issues from the client, introduction of the project group and audience.
- **Chapter 2: Requirements**
  Defining the requirements based on descriptions given by client and our goals and framework.
- **Chapter 3: Development Process**
  Review of the development process and how the development was performed.
- **Chapter 4: Theory**
  Explaining key theory concerning tree structure, drying cycles, drying kilns, and statistics. The theory we had to learn to arrive at our solution.
- **Chapter 5: Technologies**
  Explaining the technologies used as tools to reach the requirements and our goals during the project period.
- **Chapter 6: Design** The overall architecture design and front-end design of the dashboard.
- **Chapter 7: Implementation** The implementations conducted based on the requirements and design choices.
- **Chapter 8: Testing** The testing conducted to ensure code quality and functionalities.
- **Chapter 9: Security**
  The security measurements considering safety of the client and the application.
- **Chapter 10: Discussion**
  Discussing the solution and project. What we have accomplished, and improvements. Reflection of the project goals.
- **Chapter 11: Conclusion**
  What are the main conclusions after finishing the work conducted for this bachelor thesis.

2 Requirements

2.1 Functional Requirements

Through the project description, meetings, and close contact with Moelven, we were presented with the requirements for this project and our solution. These requirements changed over the course of the development process but worked as a basic assumption for our application's first sketches and prototypes.

2.1.1 Use Case

From the requirements presented by Moelven, we made a use case diagram. Based on this diagram, we made a set of use cases to understand the functional dependencies of the system and to communicate our solution to Moelven. These cases were used as a basis for the early development of the application. We wanted a balance between functionality and simplicity, considering Moelven wanted a plain and uncomplicated application.

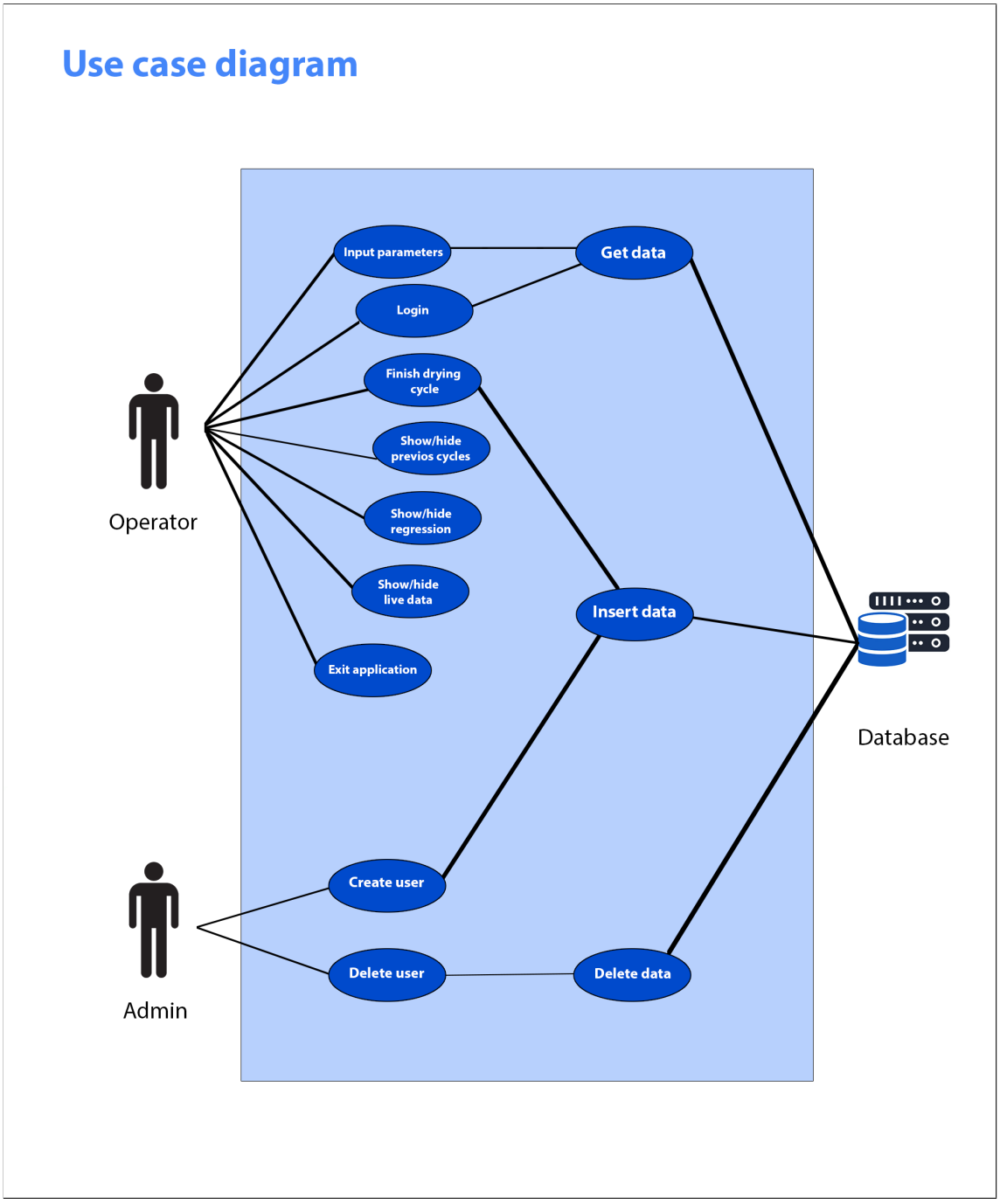


Figure 8: Use case diagram. Created in Photoshop.

Use Cases

Use Case 1 - Input parameters

Prosecutor - Operator

Description - Insertions of input parameters to categorize data and start the estimations

Flow - 1. Operator press "Input Parameters"

1.1cm 2. Input parameters pop-up window appear

3. Insert tree species
4. Insert width
5. Insert saw set
6. Insert desired moisture
7. Press cancel or start
8. If cancel, return to dashboard.
   If continue, the program starts, the data is categorized and
   estimates are created and shown in visualisation window

Figure 9: Use Case 1 - Input parameter

Use Case 2 - Finish drying cycle

Prosecutor - Operator

Description - Finishing the drying cycle by pressing the "Finish" button and inserting manual moisture check

Flow - 1. Operator press finish

2. Manual moisture check pop-up window shows
3. Operator inserts measured moisture
4. Press continue or finish based on the result
5. If continue, the cycle continues.
   If finish, the cycle stops and the moisture check is inserted into the database

Figure 10: Use Case 2 - Finish drying cycle

Use Case 3 - Exit application

Prosecutor - Operator

Description - Exiting the application and canceling the drying cycle

Flow - 1. Press *Exit* button

2. Exits application and stops the drying cycle

Figure 11: Use Case 3 - Cancel drying cycle

Use Case 4 - Show/hide previous cycles

Prosecutor - Operator

Description - Show or hide the previous drying cycles. If checked they are shown, if unchecked they are hidden

Flow - 1. Check or unchecked *Previous Cycles* box

2. Previous data are shown or hidden in the graph visualisation window

Figure 12: Use Case 4 - Show/hide previous cycles

Use Case 5 - Show/hide regression

Prosecutor - Operator

Description - Show or hide the regression. If checked the regression is shown, if unchecked it is hidden

Flow - 1. Check or unchecked *Regression* box

2. The regression are shown or hidden in the graph visualisation window

Figure 13: Use Case 5 - Show/hide regression

Use Case 6 - Show/hide live data

Prosecutor - Operator

Description - Show or hide the ongoing drying cycle. If checked the live data is shown, if unchecked it is hidden

Flow - 1. Check or unchecked *Live Data* box

2. The ongoing drying cycle are shown or hidden in the graph visualisation window

Figure 14: Use Case 6 - Show/hide live data

Use Case 7 - Login

Prosecutor - Operator

Description - Log into the application

Flow - 1. Press *login* button

2. Insert *username*
3. Insert *password*
4. Press *login*
5. If right *password*, user is logged in

   If wrong password, error is displayed
6. *Username* is displayed in application view

Figure 15: Use Case 7 - Login

Use Case 8 - Create user

Prosecutor - Admin

Description - Create new user

Flow - 1. Admin login

2. Press create user
3. Insert *First Name*
4. Insert *Last Name*
5. Insert *Telephone Number*
6. Insert *Password*
7. Insert *Confirm Password*
8. Press *Create*
9. User is created and added into cloud database

Figure 16: Use Case 8 - Create user

Use Case 9 - Delete user

Prosecutor - Admin

Description - Delete existing user

Flow - 1. Admin login

2. Insert *username*

3. Press *Delete*

3. If user exist, user is deleted

   If user does not exist, error is displayed

Figure 17: Use Case 9 - Delete user

2.2 Non-Functional Requirements

2.2.1 GUI

The graphical interface needed to be simple and manageable for the operators at the sawmills to use. It is heavily rooted around ease of use, considering the users will be operators with limited data experience. The dashboard needed to be plain and direct and not create confusion by displaying too much data. The GUI needed to be visually pleasing and use colors according to the guidelines at Moelven. The GUI had to contain:

- Visualization of estimates
- Information about the ongoing drying cycle
- Input window for parameters
- Input window for result moist
- Start and Exit buttons for the application and drying cycle

2.2.2 Performance

Regarding performance, latency was an initial element of uncertainty. Understanding the drying process and close contact with Moelven noted that short response times are not critical for our application or a priority for Moelven. As drying cycles often last for several days, the latency of a few minutes is to be expected and is not critical. However, to maximize the optimization of the application developed, we tried to reduce latency to its lowest possible value.

2.2.3 Security

From initial meetings with Moelven, the group were informed that the data stored in their Google Cloud database is not confidential, as it is classified as non-sensitive. The reason is that the given database is a copy of the original database. However, to maintain cyber security standards and reach for a modern and safe application, specific security standards were still respected during the development of our application.

The most significant security risks we anticipated would comply with our application were:

- SQL Injections - would risk the integrity of the database
- Unauthorized access - would risk the confidentiality of the data

2.2.4 Input Check

Considering the application had to take parameters and categorize data based on these parameters, it was necessary to conduct an input check to ensure that the correct data was parsed. An error message should be displayed if incorrect data is entered.

2.2.5 Platform

The application needed to be able to execute on the stationary computers at the sawmills. Moelven uses Azure cloud-based deployment, and for this reason, the application was deployed through Azure. The reasoning was that Moelven already had experience with Azure. The cloud-based deployment would also make the application platform-independent, which is a big advantage for hassle-free implementation.

2.2.6 Testing

Testing requirements were established in order to conduct testing of high quality. The importance of this was to ensure high code quality and spotting errors.

- Unit testing needed to test individual functions to ensure the correct functionality
- Integration testing needed to ensure the integration of the whole system
- System testing needed to be implemented in order to test end-to-end performance on a complete integrated system
- Acceptance testing needed to be performed in order to ensure that the application fits the target group and contain the required functionality

3 Development Process

To maximize efficiency for this group project, we allotted time slots to the construction of a well-founded development process by choosing the right development model. This created a framework for us with predictability and structure to handle challenges related to the development.

3.1 Development Model

To archive a well-founded development, the choice of development model is crucial. When choosing a model, the group had to unveil the most governing factors for our development cycle. After a group discussion, these factors was established:

- The groups' experience with development models from previous projects.
- The limited time period.
- The functional requirements established by the project description and meetings with Moelven.
- Accommodate for changes in functional requirements

3.1.1 Methodology

Based on the factors specified, the group settled on an agile approach, which was more suitable for the project compared to a plan-driven approach. The reason was the need for a flexible workflow to be able to perform sprints simultaneously. This lean methodology made it possible to accommodate for changes in requirements and revise previous decisions based on new insights. When deciding on a methodology, the group considered two possible options that all the group members were familiar with, Scrum and Kanban. The final decision was a combination of the two options, where the group considered the most convenient characteristics of each option that was used in the project. Scrum was a suitable choice for organizational purposes and structuring the development, while Kanban methodology gave the agile approach the group wanted. Scrum provided sprints with a development focus for each sprint and a valuation at the end of each sprint. The Scrum methodology alone would have structured and organized the project but not made it agile. By combining the Scrum methodology with Kanban, some of the structure from Scrum was sacrificed to maximize efficiency. The Kanban model is based around the Kanban board, which was a great tool to visualize and keep track of the development processes. It also made it easier to prioritize processes based on the importance of the task.

3.1.2 Kanban Board

In order to keep track of processes and their importance, we used Trello (https://trello.com/) to create our Kanban board. The Kanban board consisted of four fields that represent the status of a task:

Todo - Tasks that needed to be done
In Progress - Tasks in progress
For Approval - Review of the process to ensure quality. If the quality of the work was good, it was moved to done. If there were improvements that needed to be done, it was moved back to *In Progress*
Done - Tasks that had been through the approval stage and are considered complete.

The Trello Kanban board contains cards with their respective tasks. The cards created contained a description, an overview of who was working on the task, and a priority level. Each task was assigned a priority level out of five different options, which described the status of the task, as shown in Figure 18).

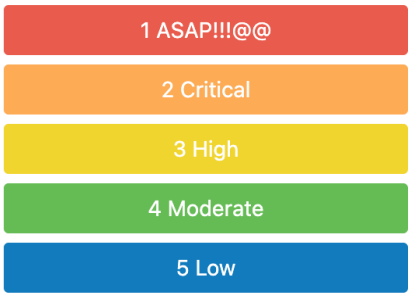


Figure 18: Kanban board, task priority levels sorted from highest to lowest priority. Screenshot is gathered from Trello.

For a project group consisting of three members, there are a lot of different processes (tasks) happening simultaneously. The Kanban board was a great tool to keep track of all processes, but it also had a few disadvantages. When the Kanban board was not updated regularly and correctly with tasks and statuses, it resulted in having to go back to correct the board, which took some time away from development and the tasks themselves. The group was aware that too many tasks in the *In Progress* or *For Approval* field would lead to an accumulation of tasks and therefore obstruct the working process. Too many tasks in the *To-Do* field at a late stage of the development process would force us to prioritize the tasks caused by the limited time of this project and probably have to define the unfinished task as a future improvement. More about this in future improvements, section 11.

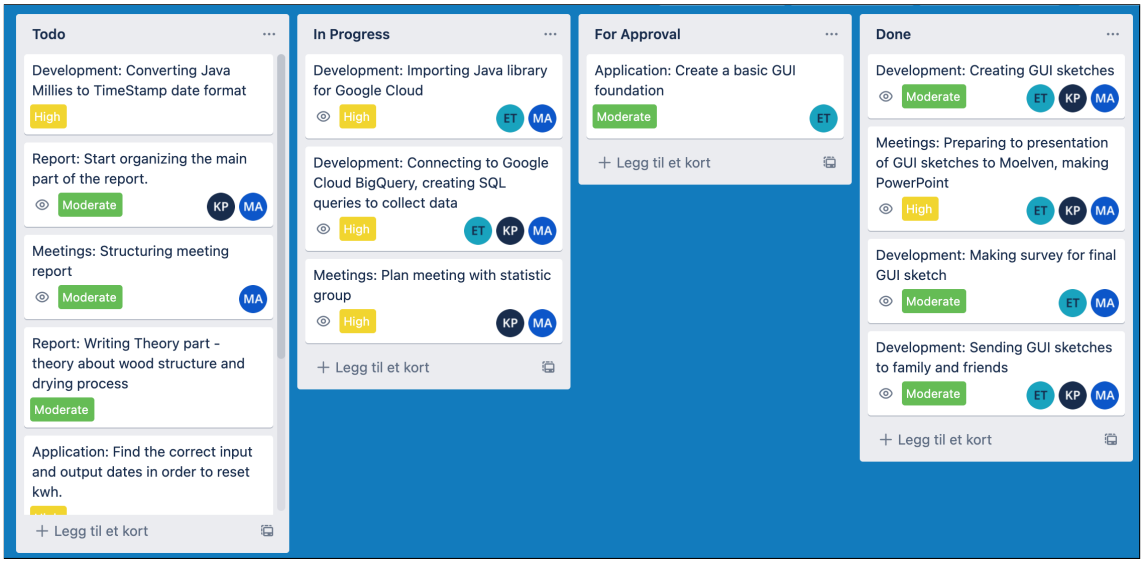


Figure 19: Picture of Kanban board in the early stage of the development phase. Screenshot taken from the kanban table in (*Kanban* 2022)

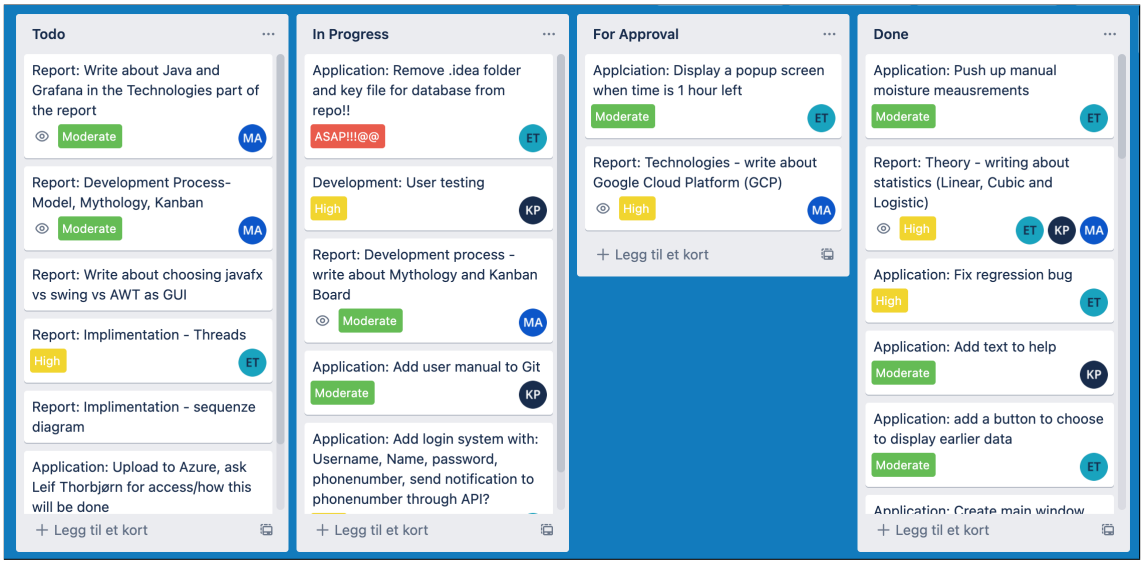


Figure 20: Picture of Kanban board in the late stage of the development phase. Screenshot taken from the kanban table in (*Kanban* 2022)

3.1.3 Iterations

The development process was divided into six milestones:

- **31.January** - Project plan finished
  Writing project plan, conversation with Moelven about database, how to retrieve data from their Google cloud data warehouse. Understanding the data and the drying process.
- **8.February** - Showing sketches/prototypes to Moelven
  Making sketches to clarify what the dashboard needs and how to visualize the estimates.
- **16.Mars** - Visiting a sawmill owned and operated by Moelven to get an insight and a visual representation of how sawmills work. Have a prototype ready for the workers at the sawmills to see.
- **21.April** - First report review by supervisor
- **6.May** - Report ready for a review
- **20.May** - Submission deadline for the whole thesis

With a frequently updated Kanban board and close contact within the group, having detailed sprints was considered unnecessary. The daily group meeting, either at school, through Microsoft Teams, or at the head office at Moelven, contributed to structuring our workflow. Planning and conducting sprints would be challenging and possibly counter-productive, considering the group needed the freedom of moving between iterations in the development process. The group concluded that the sprints would only take time away from development and obstruct the workflow.

3.2 Meetings

The group wanted a development process characterized by close contact with our client, supervisor, lecturers, and within the group. The benefits of the meetings was the assistance provided by the participants, which contributed to ideas for the final solution. The meetings log (See appendix: F) summarizes the meetings, participants and content.

3.2.1 Meetings With Supervisor

Through regular meetings, we kept our supervisor updated on our development process so that he could offer relevant feedback. The initial plan was to have meetings every week, but some periods became biweekly, considering the necessity and outcome value of the meetings. The meetings took place at NTNU in Gjøvik or Microsoft Teams in the form of a conversation where we updated him on our progress and used his experience for guidance. The guidance was provided by him

reviewing our workflow and results, asking thought-provoking questions and presenting dilemmas. The group sent him a draft of the report for him to give feedback and suggestions.

The adequacy of these meetings was getting a third-person view of our work. It was helpful getting some insights from an experienced supervisor. However, we have experienced that it can be challenging for other people to familiarize themselves with our tasks through this project. It can be challenging for a large and complex project to create an understanding of the piece of work in one session. Every project is unique and has diverse proper solutions, which means that our view and his view of a proper solution may collide. We took all of his feedback into consideration, discussed its accuracy, and made changes accordingly.

3.2.2 Meetings With Moelven

As a result of the project's complexity, this thesis required close contact with the client, Moelven. To understand and analyze their data, the group needed to understand the underlying theory of drying. Combining their expertise and knowledge with our engineering skills was necessary to create a suitable solution to the problem. This resulted in frequent meetings with the client throughout the project process. From the early meetings with Moelven through Microsoft Teams, it was determined that working at their head office in Moelven each Tuesday was an advantage for both parties. This contributed to valuable cooperation with Moelven. The frequent meetings with Moelven provided consistent feedback and suggestions for the application according to their desires. Another important tool to maintain communication with the client was the conversation channel, Moelven in Microsoft Teams, which made regular contact and file sharing possible.

3.2.3 Meetings With Statistics Group

In the early phases of the project, a general understanding of related statistics principles and methods to make the solution was necessary. The group was contacted by a statistic group, which was created to help assist students with statistics, as an essential part of their bachelor thesis. A couple of meetings were conducted with this statistic group to understand the statistical principles and methods to solve the problem. The first meeting consisted of explaining our project and its tasks and showing and sending the project-related data. They considered all information and gave some ideas and suggestions for making the estimations. It was hard to make them understand the project description properly. The group used some time understanding the actual requirements for this project, so it was not expected of them to fully understand after one meeting. The group decided not to follow up with the statistics group since it was hard for them to understand the theory and data to offer an excellent statistical solution.

3.2.4 Meetings With Professor Of Physics

Before the meetings with Moelven, the group had a meeting with a physics professor, *Are Strandlie*, to get a general understanding of the project description. This meeting clarified what to expect for the project and what should be focused on when solving the over-drying problem. The project consisted of some advanced thermodynamic principles that should not be focused on, since it is not our field. We understood that the focus should be on understanding the most important factors of thermodynamics to solve the problem. This thermodynamics theory is explained in chapter 4.

3.2.5 Meetings With Professor Of Computer Science

When implementing the statistics into the code, the group encountered some issues. Therefore we arranged several meetings with *Ivar Farup*, a professor of computer science at NTNU in Gjøvik, in order try to solve the problem, given that non-linear regression is outside our competence and is taught in a separate course that is not part of our course of study (*Advanced Programming course*). He had several ideas, such as cubic and logistic regression, and in the end we settled on logistic regression as our regression model, given it resulted in the growth model that suited best for the estimates.

Given that non-linear regression is outside our competence and is taught in a separate course that is not part of our course of study, this is the *advanced programming* course.

3.3 Time Tracking

To keep track of the working hours and the distribution of hours between different areas of the project, the time tracking and reporting tool called *Toggl Track* was used in the project, please see appendix I for more information. This was a handy tool used to visualize the time spent for different categories and accordingly analyze it for time use improvements. With *Toggl Track* the group created six labels to sort the working hours into:

- Report - Writing main report and other documents.
- Development - Models, sketches, planning.
- Coding - Making the application in Java
- Meetings - All meetings regarding the bachelor
- Project Plan - Working with project plan
- Other - Other bachelor-related work

3.4 Tools

Throughout the project process, the group has used several tools to assist in meeting the project goals.

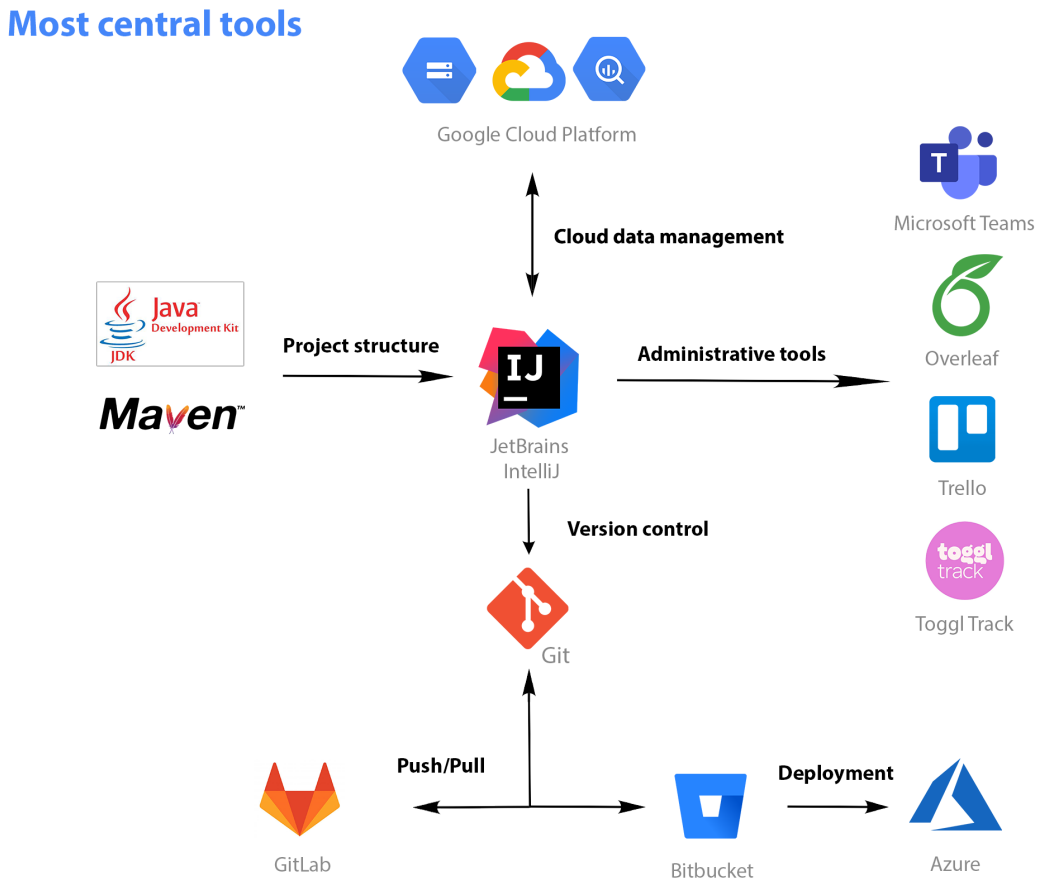


Figure 21: Graph showing the most essential tools used for this work and their relationship and uses. Created in Photoshop.

3.4.1 Development Tools

JetBrains IntelliJ - Development in Java
Maven - Building tool
Java JDK 8 - Development kit

3.4.2 Data Management Tools

Google Cloud BigQuery - Data collecting and analysis
Grafana - Visualisation of data

3.4.3 Administrative Tools

Microsoft Teams

Overleaf - Report structuring and writing

Trello - Kanban board

Toggl Track - Time tracking and reporting

Figma - Prototype sketches

Freeonlinesurveys - Online survey for prototypes

NTNU Open - Reading previous bachelor theses for inspiration

Draw.io - Making models and diagram

3.4.4 Version Control

Git - Version Control

GitLab - Pipline

3.4.5 Deployment

Azure - Cloud based deployment

BitBucket - Git repository management (needed for Azure deployment)

4 Theory

This chapter describes the necessary theory to understand our issue and its relevance to our solution. In particular, it addresses the wood structure, how it affects the drying cycles, how the drying kilns work and why they affect the data and the statistics behind creating the estimations and sorting data.

4.1 What Is The Importance Of Drying Wood?

The product quality is essential for Moelven and contributes to maintaining their reputation and recruiting new customers. Good product quality comes with quality production processes. One of the most important production processes to achieve quality wood-based products is the drying process. Both physical and mechanical properties of wood will improve as a result of the drying process. Moelven dries their lumber because raw lumber is unsuitable for most purposes. All types of wood have a moisture level far above the desired operating humidity in the newly felled state. (Tronstad 2000b).

Dry wood compared with raw have i.a. the following benefits (ibid.):

- Lower weight - Cheaper transport, more profitable for Moelven to handle
- Less shrinkage and swelling in use
- Greater resistance to rot and fungus - crucial for making quality wood based products
- Better impregnability
- Higher strength properties
- Better glueability
- Better adhesion for paints and varnishes
- Lower thermal conductivity
- Better machining properties

To achieve the right humidity and all of its benefits, Moelven is dependant on quality estimates and programs for the drying process. Our solution will be a tool for Moelven to ensure product quality by improving the estimations on drying time and visualizing relevant data. Our estimates and visualizations, combined with their experience, will help create a suitable drying process, which will help create quality wood-based products and reduce energy consumption.

4.1.1 Minimizing Unfortunate Properties

When choosing wood as a building material, we must accept that there will be a certain degree of deformation, discoloration, and cracking. These adverse properties are closely related to the moisture content of the wood. With proper drying, processing, storage, transport, and assembly, most of these unfortunate properties of wood as a material can be reduced. It is important to know the mechanisms that cause wood to shrink, swell, deform, crack and discolor, and not least what measures can be taken to minimized these unfortunate properties.

4.2 The Tree Structure

Wood cannot be imitated by any other building material, reasoned its complicated structure. Each tree species has its own characteristic structure, and within a tree species, no trees are alike. This result in different drying cycles with their own characteristics and unique data. When the estimates where create this was taken in to consideration. The data had to be categorized based on different parameters that all affected the drying cycles in their own way. These parameters include:

- Dimensions - Thickness multiplied by width
- Tree species - Type of tree
- Saw set - The placement and dividing of the lumber center block.
- Desired final moisture - The final moisture level the operator wants to achieve.

Furthermore, it was necessary to remove incorrect data or data with significant differences from the ideal drying cycle. When categorizing and sorting data, the big differences between cycles are removed and only relevant data create the estimates and visualizations.

4.2.1 Where Is The Moisture In The Wood?

Removal of moisture in lumber is a challenge, both in terms of amount of water to be evaporated and to achieve proper and even final moisture. There is a regular variation in the humidity of the stem cross-section. This results in different variation of moisture in planks and boards, cut by the log. The differences in the cross-section of spruce and spine trunks has for example on an average a humidity about 80 percent, while lord heartwood contains 32 - 40 percent and surface wood 120 - 130 percent (Tronstad 2013)p.7 (Tronstad 2000b). The different variation in humidity is seen in the huge variety of data stored in the database.

4.2.2 Saw Pattern

Posting the log is a procedure with the aim of getting the largest possible volume of sawn products out of the log (*Postning* 2016). The outcome is a dividing of the lumber, known as saw pattern. In this sense, dividing refers to center gods and side boards depending on which part of the trunk is sawn. The saw pattern indicates the placement and dividing of the center block. The varying thickness in sawn lumber is a result of the saw pattern (ibid.). This variation in thickness affects the drying times and has to be taken into consideration when creating and adjusting our estimates. We solved this by letting the user of the application enter the type of saw pattern in the input field of the application, expressed in the application as *Saw set*.

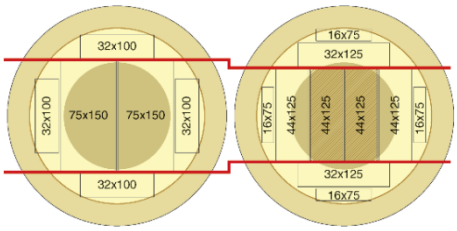


Figure 22: Example of saw pattern (Håkon Toverød 2022).

Figure 23: Heartwood cut accordingly to the saw set. Picture taken by project member during visit to the sawmill.

4.2.3 Raw Wood Contains Free And Bound Water

Wood is built up of a number of cells or hollow fibers, which can be compared to long pipes with varying wall thickness that are glued to each other. Mostly the cells are oriented in a vertical direction and these cells have the task of stiffening the trunk, as well as being transport channels for fluid from the root and up. The horizontally oriented channels are called marginal rays and these cells go in a radial direction from the margin. Among other things, they have as a task to direct nutrients that are formed in the crown inwards from the bast and transport resin, which provides protection of the tree from damage and insect bites. The cavity in the cells can be almost completely filled with water in living trees or in freshly cut timber,which is referred to as *free water* (Tronstad 2000b). The building blocks of the cell walls are saturated with water, and this water is therefore called bound water. The sum of free water and bound water gives an expression of the wood's moisture (ibid.). The surface wood is mostly free water, while heartwood contains mostly bound water.

During the drying process, the removal of free water from the cell cavities occurs first. The removal of moisture increases sharply in the beginning of the drying cycle because of the free water being vaporized, but slowly decrease as the bound water remains in the lumber. When the free water is removed, the humidity is about 30 percent(Tronstad 2000b), and after this there is mostly just bound water in the cell wall. This stage is called the fiber saturation point(ibid.). The fiber at this stage is saturated with bound water, which takes significantly longer to vaporize than the free water. This is a very important stadium in the context of drying, and makes up the majority of the drying cycle. Getting the humidity level from 30 percent (Tronstad 2013)p.7 (bound water) down to the desired level takes time and precision. Today, the drying schemes and the operators experience decides when the drying is finished. Our solution will at the fiber saturation point tell when enough of the bound water has been vaporized and the lumber has reached its optimal humidity level.

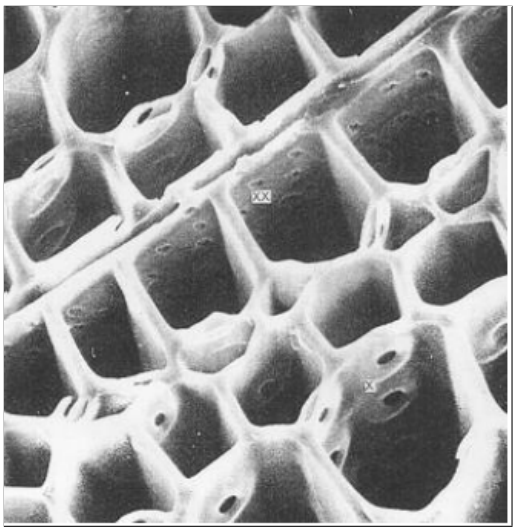


Figure 24: Micro-cut of softwood with cell cavities and pore openings in the wall. The the free water will lie in the cavities and the bound water in the cell walls.

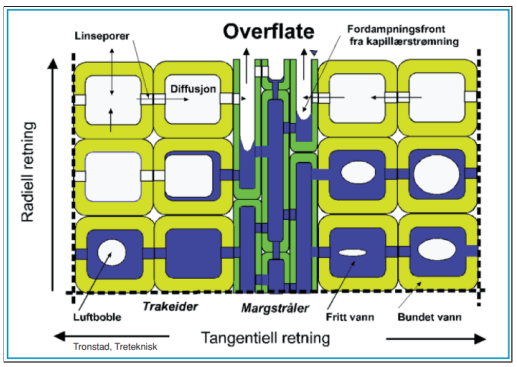


Figure 25: Graph of cut through of some surface cells at the beginning of a drying process

Figure 26: Figures gathered from (Tronstad 2000b)

Figure 4.2.3 is a picture from an early phase of drying where the moisture, in a mixture of capillary transport and diffusion moves to the surface. The drying air transfers heat to the wood to evaporate the water and transports the moist air away from the wood.

4.3 Drying Process

4.3.1 Drying Kilns

Most lumber in Moelven is dried in artificial drying kilns. In the drying kilns, the wood is exposed to a drying temperature usually in the range 60-85°C, a relative humidity of 25-90 percent and an air speed of 2-7 m / s, information gather through meetings, see meeting catalog in appendix F. In modern kilns the lumber is put under pressure to reduce the deformations, especially twisting. It is important to control the drying process to get the correct final moisture adapted to the area of use. The picture below shows a batch kiln.

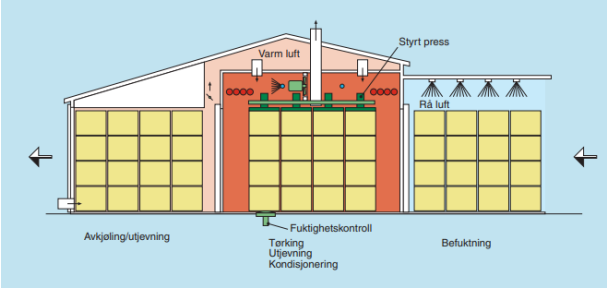


Figure 27: Modern kiln drying with top pressure and climate control before, during and after drying. Figure source: (Tronstad 2000a)

Drying kilns is an enclosed space with controlled climate. In the room there are fans that circulate the air through the stack of lumber. Heat is supplied via heating batteries to speed up the process. The drying kilns makes it possible to dry wood to the desired moisture ratio and control the process completely to minimize the risk of damage(Tronstad 2000a). The controlled environment is crucial for Moelven to achieve the product quality they need. Drying time varies a lot between different types of wood and dimensions, from a few hours up to 3 months for coarse oak planks (*Torkmetoder* 2022). These big differences in drying time makes it important to categorize data to compare relevant data and make estimates out of it. These estimates are not precise if the data used is irrelevant

The drying rate and the drying quality are determined by the air temperature, relative humidity and air velocity passing over the sawn timber. All of these parameters varies allot and affect the drying cycles in various ways. An algorithmic estimate that explains the relation between energy consumed and humidity is therefor not that accurate. In theory this kind of estimate will work, and can estimate when the humidity reaches a certain level, but in practice, other external factors affect the estimate. An estimate based on previous data is therefor more desirable.

4.3.2 Drying Schemes

When building a drying scheme, the objective is to use as few resources as possible over the shortest amount of time, without reducing product quality. This mens finding the balance between drying speed and drought damage. A drying schedule can be divided into four different phases: the heating, drying, equalising and conditioning. How many phases there will be in practice used in addition to the heating phase and the drying phase depends on the end use of the timber and the possibilities of the drying plant (Tronstad 2000a).. After the drying process is finished, the sawn timber is taken to the dry sorting plant for trimming and sorting, with subsequent packaging/binding (ibid.)..

Heating Phase

In the heating phase, the wood must be heated to the temperature which the wood should have at the start of the drying phase. This phase is desirable to be made as short as possible for two reasons: To save drying time and to get as quickly as possible through the temperature ranges where the danger of fungal attack at work is greatest. The rising temperatures in the worm-up phase have to be controlled to prevent damaging the lumber. (ibid.).

Drying Phase

The drying phase takes place by blowing air with a specific course of temperature, relative humidity and speed over the lumber. When choosing a temperature level during the drying phase, one must think about what consequences the level has on the main characteristics such as crack, fungal damage, discoloration and deformations. For crack, fungal damage, deformation and damage insects, it is beneficial to have so as high a temperature as possible. Choice of drying temperature will therefore be a balancing act between these different factors, which in turn will be strongly dependent on the tree species (ibid.).

.

The Conditioning Phase

Reduce surface hardening (the stretching of the outer layer) that has occurred during the drying phase by adding moisture. This is to avoid unfortunate deformations by further splitting of the timber. By adding moisture to the surface of wood that, you will get a swelling of the outer layer of wood. The swelling causes pressure stresses and thus a compression of the outer structure (back to the dimension it would have had by free shrinkage). After the moisture has equalized, this will contribute to the surface hardening effect reduced or completely removed. This process is called conditioning the wood (ibid.).

4.4 Moisture Measurement

The moisture measurements work as a quality assurance for Moelven. It helps them adjust their drying schemes based on the results from the drying process. If there are big differences between the desired moisture and the moisture check the schemes need to be adjusted.

For our solution, these moisture measurements are key to adjust our estimates based on the actual results from the drying process. This makes it possible to determine if the drying cycle was good or not.

4.4.1 Measure Of Moisture And Surface Hardening In The Wood

The moisture in the wood is defined by the weight of the water in the wood in relation to the weight of absolutely dry wood, expressed as a percentage. The measurement method that most satisfies this definition is therefore the drying/weighing method, where the wood is first weighed in the raw state and then dried. The weight difference is therefore the water content, which is then calculated as the moisture content of the wood as a percentage of the dry weight. The drying/weighing method is therefore considered to be the basic and in practice the most accurate measuring methods for wood moisture, and is used as a reference in comparison with other measuring methods.

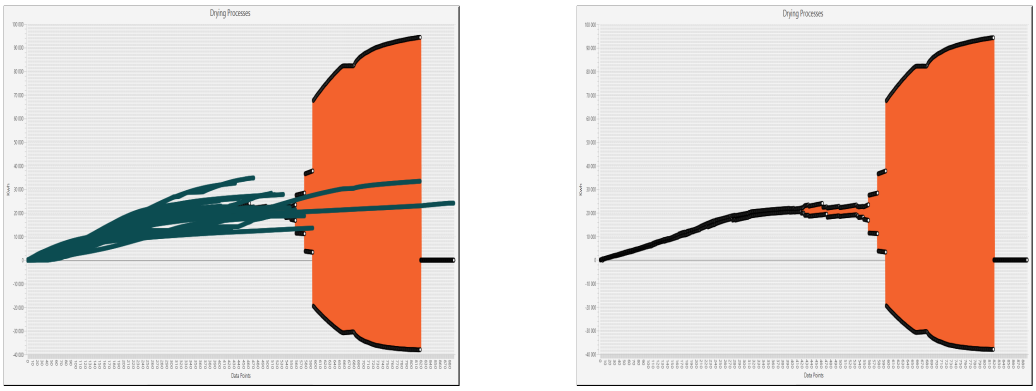
4.4.2 Electrical Resistance Measurement

The electrical resistance of the wood is affected by the moisture of the wood (Tronstad 2013) p.22. This phenomenon is used to measure wood moisture in a simple and fast way. The measurement method based on this principle has therefore become the most widely used so far (ibid.) p.22. This measurement happens in the form of manual checks that is performed when the drying scheme is done and decides if the moisture is at its ideal value. This manual check needs to be implemented into our solution to be inputted into the database. Storing this manual moisture check, opens up the possibility of adjusting the cycles based on these measurements.

4.5 Statistics

Our task was to predict and visualize the drying cycles based on collected data material. To archive this, statistics were used to create the estimates and as quality assurance. Statistics made it possible to find patterns and use these patterns to predict the estimated time it will take for the timber to dry. We used regression to find the function whose curve/graph fits best with the collected data. In addition, we also used confidence intervals to measure how good the estimates of the unknown quantities are. Simply put, a small confidence interval indicates that the estimates are safe, while a large confidence interval indicates that the estimates are more uncertain. To be able to give an estimate that is as concrete and reliable as possible, the application uses a large number of data sets. Considering new energy- data being uploaded into the database in real-time, the application will become more precise as more data is uploaded and processed by regression.

4.5.1 Confidence Interval



(a) 95% confidence interval on approximately 20 previous drying cycles. Previous data in Moelven teal color. Confidence interval in black and orange.

(b) 95% confidence interval on approximately 20 previous drying cycles. Confidence interval in black and orange.

Figure 28: Visualization of how the confidence interval will exponentially become imprecise when data there is a lack of data. Both figures are taken as screenshots from the application where the x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Confidence intervals are used in statistical analysis of data that is associated with uncertainty and variation (Bjørnstad 2018). A confidence interval provides a lower and an upper limit for the size that is estimated. The length of the interval indicates how good the estimate is. A long interval signals greater uncertainty than a short one(ibid.).

By using confidence intervals in our application, we are able to remove imprecise data and improve the calculated estimate on the remaining data for the regression analysis. The drawback of using a confidence interval is that it becomes imprecise if there is a lack of data to base the confidence

interval. This is attempted to be visualized in figure 28a and 28b.

4.5.2 Regression Analysis

Regression analysis is a statistical analysis method that describes the relationship between one or more independent variables and a dependent variable. A regression allows us to confidently determine which factors matter most, which factors can be ignored and how these factors will influence each other. It is essential to comprehend the following terms in order to understand regression analysis fully (Braut 2021).

Dependent Variable: This is the main factor which you try to understand or predict. The dependent variable in our application is moisture. **Independent Variables:** These are the factors which your hypothesis has an impact on your dependent variable. The independent variables in our application are width, height, tree species, and saw set (*What is Regression Analysis and Why Should I Use It?* 2021).

Based on these variables, an expression can be made when the drying cycle has reach the desired moisture value. The regression line which is created, is based on the relationship between the dependent variable and the independent variables.

4.5.3 Linear Growth Model

The first regression model we tested for our statistic calculations was linear regression. The result from this regression was a linear growth model that expressed the estimated optimal line. However, the curvature of the previous drying cycles caused the linear regression to be unsuited for our purpose.

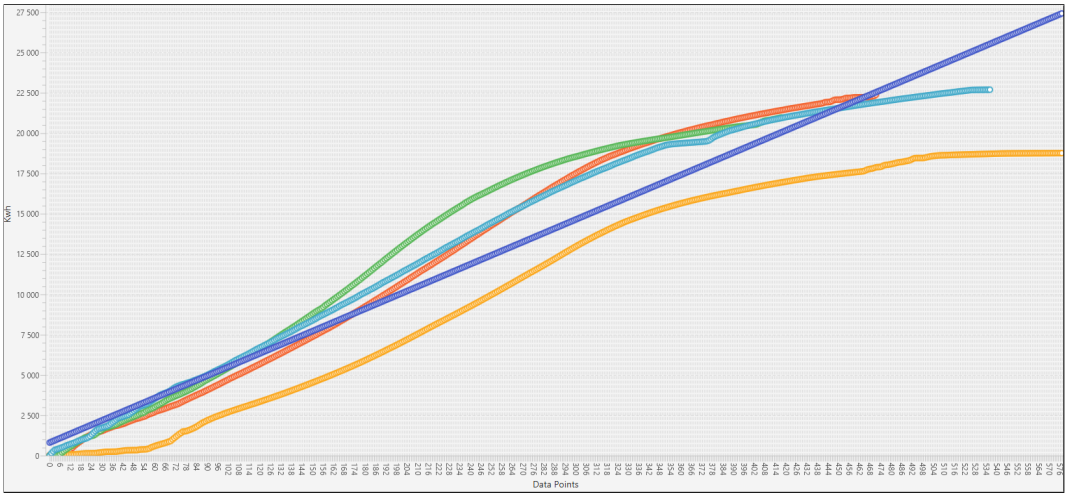


Figure 29: Linear regression(dark blue), historic data(yellow, green, red, light blue). The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Figure 29 shows the linear growth model in blue, and the other colors are the previous drying cycles. The picture was taken from an earlier prototype. Based on the result shown in the figure, we concluded that the graph was approximately linear, especially in the middle stages of the cycle. Towards the end of the cycle, the graphs seemed to converge and become non-linear. The linear growth model could have worked but would have made it hard for operators to spot if the ongoing drying cycles from live data follow the ideal line/area created by the linear regression.

We opted to test other regression alternatives to try matching the curvatures of the cycles. Considering the imprecise estimate caused by linear regression, we decided to try out different types of non-linear regression for our calculations.

4.5.4 Cubic Growth Model

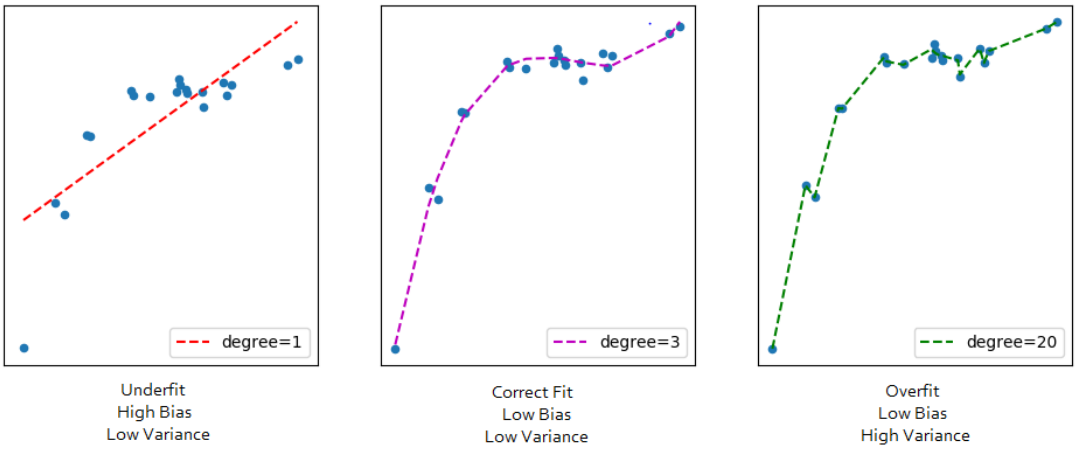


Figure 30: Figures displaying different degrees of polynomial regression and how over or underfitting the graph leads to a bad result, figures gathered from (2022)

Polynomial regression is a non-linear regression model that was considered. The problem with polynomial regression is *the higher the polynomial degree, the worse the growth model becomes*, as can be seen in figure 30. The third-degree regression, also known as cubic regression, was the degree that seemed to fit our data best. If we switched the axis where the original x-axis became the y-axis and the original y-axis became the x-axis, then the graph would take the form of an S-shape, which is approximately what the data looks like.

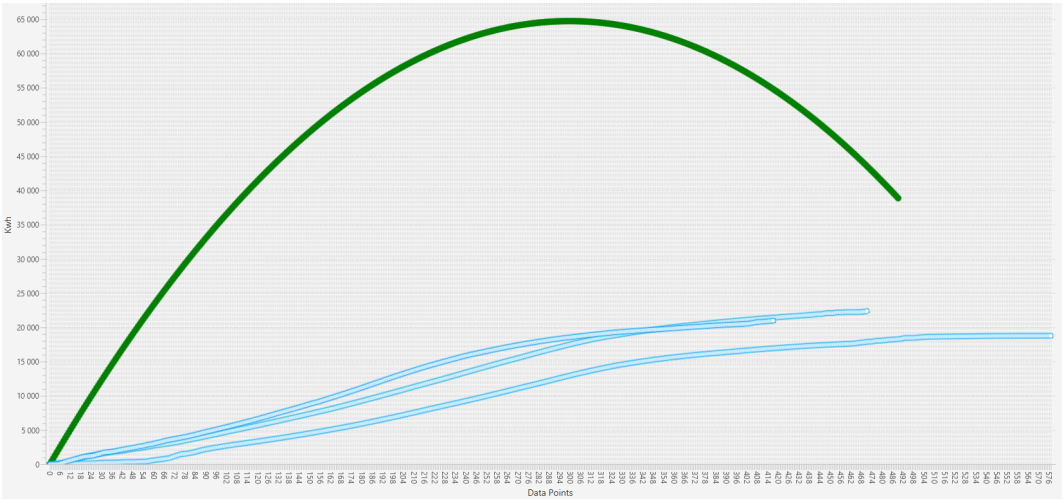


Figure 31: Cubic regression(green) and previous data(blue). The screenshot is taken from the application. The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Figure 31 visualize the attempt at fitting our data to the cubic regression. The regression clearly does not fit the data and gives no clear advantage compared to other non-linear regression models.

4.5.5 Logistic Growth Model

Logistic regression was another S-shaped graph that had the potential to fit the data, with an approximately linear middle part and converging towards the end

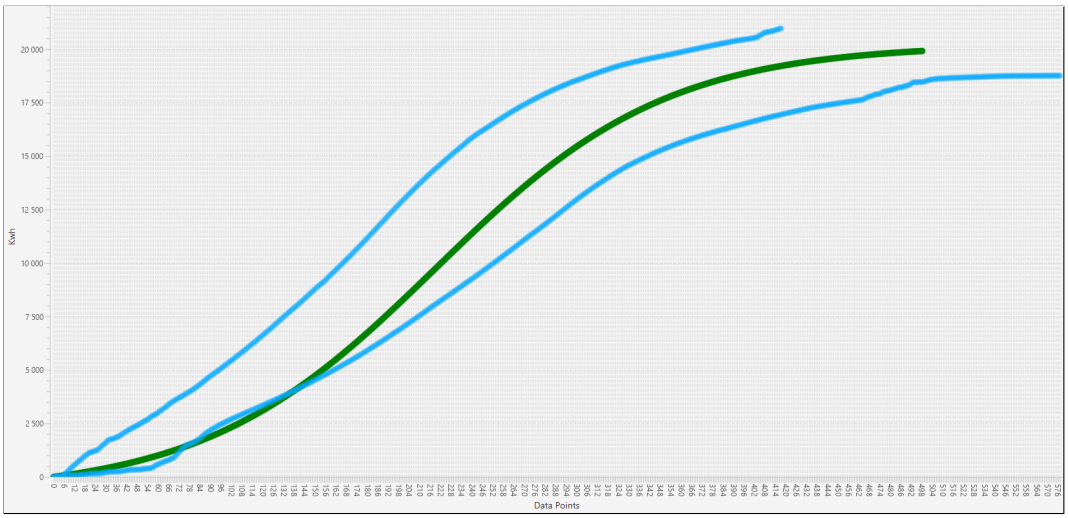


Figure 32: Logistic regression in green and previous data in blue. The screenshot is taken from the application. The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Figure 32 shows the logistic growth model, and it was clear that the logistical model gave the best estimation results and therefore became the obvious choice. The population size at which the rate levels off is called the carrying capacity, or K (Aslak et al. 2010), p.166.

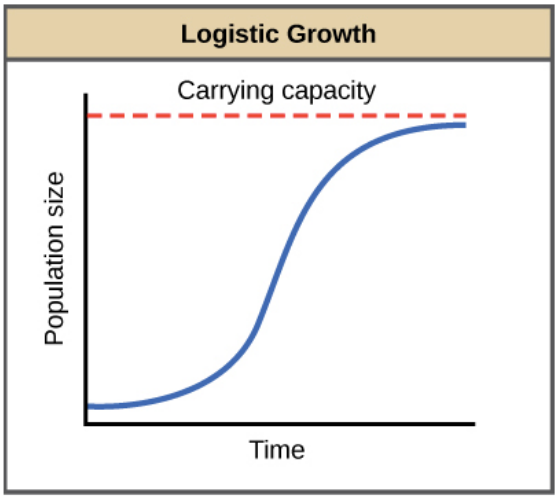


Figure 33: Logistic growth. The images shows a logistic growth where the population expansion decreases and it levels off when the carrying capacity of the environment is reached. This results in a S-shaped curve, (*Exponential & logistic growth* 2022)(*Environmental Limits to Population Growth* 2022).

Since the drying cycle grows in a environment with limited resources, it can be modeled by the logistic model.

$$y'(t) = \alpha y(t)\,(1 - y(t)/\beta)\,, \qquad y(0) = y_0,$$

Figure 34: Formula of the logistic model, formula gather from (Aslak et al. 2010) p.166

In the formula α is the growth factor and β denotes the carrying capacity (Aslak et al. 2010) p.166. In our application we implements the slope α from a library called *SimpleRegression*. As mentioned β is the carrying capacity, and in our application, that is the highest value on the y-axis, which is the highest kWh measured. This formula (ibid.) p.166 gives us the estimated duration of the drying cycle based on the parameters the user has selected. The client wanted the estimate to be as precise as possible. To make the estimate of the duration of the drying cycle more concrete, we implemented a confidence interval. To make the estimate lie between the graphs, we ran a confidence interval on the y-axis so that the regression would lie between the most relevant graphs. Therefore, β is a confidence interval in our application. Further, y_0 is the constant term of the linear regression. To avoid the regression getting mishandled, we need to make sure that the constant does not become negative. Raising the constant term y_0 first to the power of two and then calculating the square root of it after the elevation makes y_0 positive.

5 Technologies

The client had no preferences in the choice of technology or architecture of the application, other than the application being able to run in their cloud-based deployment tool Azure. Thus, was some time spent finding the right technologies to meet the requirements for the application. The technologies were chosen based on the prior knowledge in the group and their ease of applicability to our solution.

5.1 Programming Language

Firstly, the programming language to use to develop the main components and functionalities of the application had to be chosen. The programming language was selected based on the following criteria:

- **Previous experiences:** If the choice of language fell on one that the group was not familiar with, it would take extra time to learn the language. It was also an advantage if Moelven already knew the language.
- **Compatibility across platforms:** Moelven wants the system to be moved to other platforms with minor effort, for example, to the cloud (Azure).
- **Interconnection with the existing technologies in Moleven:** It would be advantageous if the language could be integrated with the existing systems in Moelven.
- **Mathematics:** The programming language needs to be able to perform various statistical calculations, such as regression and confidence intervals.
- **Supported by libraries:** For rapid development of the code, libraries are critical in decreasing the time required for coding. Because of the possibility of reuse, we can use one function call, and a whole task will be performed that would otherwise require code ranging from several to several dozen lines.
- **Database access:** By using a language that has a form of database connectivity which would help in connecting to a database and thereby executing SQL statements against the database.

The different programming languages explored were C, Python, and Java, as the group members were familiar with these program languages. Since a significant amount of work had to be conducted using a statistical part, it was essential to select a language that contained relevant mathematical libraries to use for the regression. In addition, other programming languages could have been used for this application, such as Go or R(Simplilearn 2021). Nevertheless, they were not found relevant enough to learn, as they did not provide significant benefits for the application.

5.1.1 C

The advantages of using C were that Moelven and their employees were well accustomed to the programming language. In addition, the utilization of data structures and algorithms in C has made the program calculations extremely quick and smooth(*10 Most Important Features of C Language* 2022). That being said, the fast processing speeds C offers come with security limitations. This makes C language useful to employ in complex estimations and tasks, which makes C especially beneficial for the statistical part of the assignment. C also has an Open Database Connectivity, which is an API that allows applications to access data in the database management systems (DBMS) using SQL as a standard for accessing the data. The disadvantages would be that the group members have the least knowledge of C compared to the three mentioned above. It also lacks object orientation which means it cannot create subclasses of classes, so it makes it difficult to reuse existing codes (*Advantages And Disadvantages of C Programming Language* 2021). In C, the code would have to be modified for the program to be able to run on different platforms.

5.1.2 Java

Java is a programming language that consistently scores high on most programming indexes (Jonathan and Brad 2022)(*TIOBE Index for May 2022* 2022). The language is Object-Oriented and is a general-purpose programming language that helps create programs and applications on any platform. By being a OOP, it enhances the flexibility and reusability of the code. Another benefit of OOP is that it helps us increase security by binding the data and functions into a single unit and not letting the outside world access it. In addition, Java is also platform-independent, which means that it can run code on any machine that supports the Java Virtual Machine. JVM is also capable of automatic memory management.

Furthermore, Java includes JDBC, which is an application programming interface (API) that defines how a client may access a database. The JDBC is useful considering a large amount of data that needs to be collected and processed from the database. Java is one of the most common programming languages, so learning it and using it is important for future growth and possibilities (*11 Most In-Demand Programming Languages in 2022* 2022). The Java environment is constantly improving with new versions and has grown significantly in recent years. Another benefit is Javas' possibility of multi-treading, which makes it possible to run the GUI and the data analysis on different treads. Java is also the programming language that the team members are most acquainted with. The disadvantage of using Java is that it is a language that is not operated by Moelven today, which means if further development is desirable, it will require additional resources on the part of the employees to learn the language after the bachelor thesis is over. When comparing Java to C, it is not as fast at processing data.

5.1.3 Python

The last alternative explored was Python. Python is a language that is well known for being easy to learn and understand for people with no previous experience in coding. The language has broad support through extensions and frameworks that provide one with a quick opportunity to use the language for otherwise complicated functionality. Python is portable, which means there's no need to change the code to run the program on different platforms like C. Python also has many relevant libraries, especially for mathematical purposes, like NumPy, SciPy, Scikit-learn, Sage, etc. The disadvantage of Python would be that it lacks behind consideration database. Compared to technologies like JDBC and ODBC, Python's database access layer is primitive and underdeveloped. In addition, Python works by interpreting code line by line when executed and is not compiled like C and Java. This results in Python being significantly reduced in performance at processor-intensive algorithms. Furthermore, Python does not have static variable types like Java and C, which opens up more execution problems that otherwise can be detected by a compiler in advance.

5.1.4 Choice Of Programming Language

After analyzing all the advantages and disadvantages of the programming language, the final decision came to select Java. Java covered all the criteria set for the language, except that it is not a familiar language to Moelven, which may affect the maintenance of the application after the bachelor. C could have been a good alternative, but considering it not being OOP, the collaboration across the platform, and the group's experience with C, it was chosen against. Database access was an essential reason why Python was not selected.

The group chose to use Java 8 instead of the newer Java versions. This was justified by prior experience with the performance and Java 8 still being the most popular version (*Java 8 still dominates, but Java 17 wave is coming* n.d.) since it is a LTS version. Java 8 had all the functionality needed, and it was found more effortless than using Java 14 or 17.

5.1.5 JavaFX

JavaFX is an open-source internal GUI library for Java, which offers support for numerous operating systems. After examining the options, the decision came to use JavaFX as it provided the key factors of freedom needed to customize the GUI. More on the decision making in section 7.3.1. The issues around visualizing the estimates in the GUI were complex and specific. Therefore it was essential to build the GUI from the ground up. Making the coordinate system modified to plot the estimates and previous cycles. A possible solution to this problem was using Scene Builder, but this tool could not process the energy data. Instead of figuring out this problem in Scene Builder,

the group decided to use JavaFX with CSS.

5.2 Sensor Data

Moelven has 15 sawmills across Norway and Sweden which use sensor technology in their drying kilns. These sensors collect data that help Moelven monitor and analyze the drying process. The sensor data is collected and uploaded to their database system in real-time. In recent years, Moelven has encountered digitization of their sawmills. A part of this digitization has been gathering energy data from their drying kilns. This energy data is stored in their database system but has not been processed and monitored before. *Årjäng*, *Valåsen* and *Notnäs* are sawmills where data of energy consumption is collected. Our solution processes the stored energy data and the real-time uploaded energy data from the sawmills. The processed data is then used to create estimates and visualizations for analytic purposes.(*Våre produksjonsenheter* 2022)

5.2.1 Sensor Technology

At the sawmills, there are two types of sensors located at the drying kilns:

- **Water flow sensor** (Model: *LRF 2000s GEN III Clamp On* (*Flödesmätare Ultraljud för vatten och alla andra typer av vätskor och flytande gaser.* 2022)) - The flow sensor is based on ultrasound, and the principle is to measure the speed of sound in the flowing water between two points.
- **Temperature sensor** (Model: *PT 100 Clamp ON*) - Measures the water temperature on its way in and on the way out.

To collect energy data, Moelven needs to calculate thermal energy usage using sensor data as a factor. The volume of hot water from the flow sensor and the temperature difference from the temperature sensor is fed into a calculating unit, which uses a formula to calculate the energy usage based on the two parameters. The result of the formula is consumed thermal energy (kW/h).

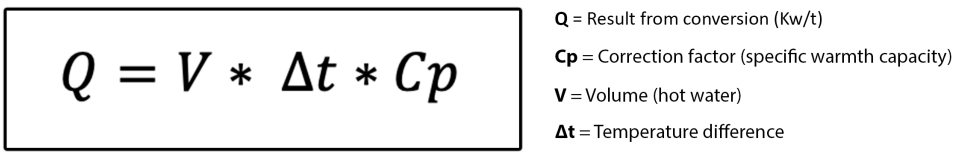


Figure 35: Formula for conversion to energy. Created in Microsoft Word. (Haugen 2019)

5.3 Google Cloud

Google Cloud includes numerous services that aim toward accelerating an organization's abilities to become digitized. Google Cloud consists of the Google Cloud platform, which offers services like computing, storage and database, networking, Big data and machine learning. The GCP is a modern and fast-growing cloud-based platform with a global interconnected infrastructure. The GCP has many strengths that make it an attractive cloud solution:

- Offers great documentation with walkouts of implementation of or services.
- Fast, consistent and scalable performance because of advanced software defining networking at a global level.
- Offers security with encryption, recovery plans, monitoring logs and identity access management tools (IAM)

There are not many disadvantages in using GCP, besides the fact it is not the most used platform and therefore offers more minor services compared to other leading Cloud platforms like AWS.

For this thesis, the storage and database services offered by the GCP is used. GCP is the data management and storage service of Moelven, GCP works as a uniform cloud-based data processing solution for the sawmills. The usage of GCP has been a part of the digitization of Moelven. Sensor data from the different production areas are directly uploaded into Google Cloud and later handled by Moelven. To help manage and analyze the data, Moelven uses a Google Cloud engine called BigQuery. The built-in features, serverless solution, and BigQuery make it possible to run big queries in seconds with no infrastructure management. The BigQuery data sets are saved in Google Cloud Storage. The connection between Java and GCS is essential for the application, considering all the data needs to be processed by the application, which is located in the GCS. The bridge to collect data is BigQuery and its capability of querying using SQL statements. The Google Cloud documentation became a necessary tool to understand this connection.

Figure 36: The Google Cloud platform. Figure gathered from (Perez 2016).

5.3.1 Google Cloud Storage

GCS is a RESTful online service for file storage and access to data from the GCP infrastructure. The data and analysis done in BigQuery can be saved in the GCS.

5.3.2 BigQuery

BigQuery is a serverless data warehouse that serves as a Big data analysis tool on the GCP. It supports the usage of ANSI SQL querying and returns the query data as a JSON format. BigQuery is a great tool for managing data (create/delete tables, functions, views) from the GCS. As mentioned, BigQuery was essential to manage the database and create a bridge between Google Cloud and our application running in Java.

5.4 Grafana

Grafana is an analytic and monitoring system for databases. Moelven already uses this tool to visualize and monitor their stored data. Grafana provided an understanding and visualization of the correlations in their database for the group. Grafana supports a JSON key file upload from Google Cloud, making it possible to use Grafana's query function to visualize queries. The visualization of these queries became crucial for us to understand how drying periods work and how to connect the correct energy data to the proper periods.

5.5 Overleaf

Overleaf was used in the project as a collaborative writing tool because it provides easy sharing, simultaneous editing, real-time preview, and its capabilities of finding and displaying LaTeX errors. It is a valuable tool for writing scientific documents, hence its beneficial use for referring to figures, chapters, tables, and appendixes throughout the report, which creates a common thread. The only disadvantage of using a LaTeX-based report generator was the error handling since the error handling took some time away from the development and writing of the report.

5.6 Git

The version control for the application was Git. Git was chosen due to several of its benefits in collaboration with code development in teams. Since the project has multiple developers working in parallel, Git was necessary as version control to ensure there were no code conflicts between the developers in the team (*What is Git? What benefits does Git offer?* 2022). One of the convinces of using Git in the project was its branch capabilities (Sridhar 2018). This gave each member their

independent branch, ensuring that the main branch always included production-quality code (*Why Git for your organization* 2022). In addition, Git is also usable. It keeps track of various changes made to each iteration, code changes can be easily committed, and the effortless comparison and merging of version branches. Additionally, familiarity with the usage of Git through working with this VCS in previous projects was also a deciding factor in the choice of VCS.

Furthermore, GitLab was used as a hosting repository provider to develop the application, while BitBucket was used to publish the final code of the application to the production server (*How to Choose Code Repository for Your Project* 2021). BitBucket is used by Moelven as their open-source platform and are in many ways similar to GitLab. BitBucket is flexible, considering its access to a wider range of version control systems (*7 Reasons Bitbucket is Better than GitHub (the Ultimate Smackdown)* 2022), such as Google Cloud Platform, which Moelven uses.

6 Design

6.1 Backend Architecture

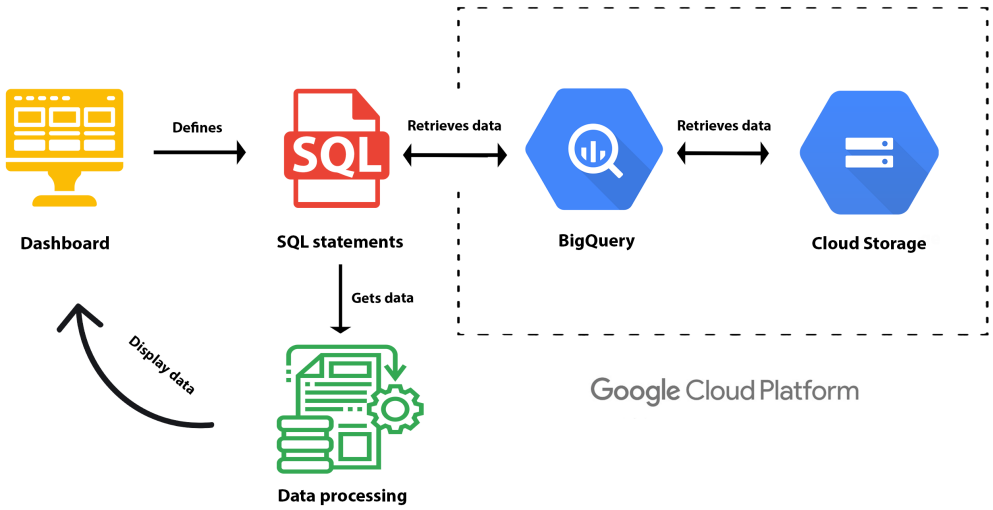


Figure 37: Architecture visualizing the dataflow through the program and interacting with components. Figure created in Photoshop.

The architecture of our application is built around collecting and processing data from the GCP. The group learned the Google Cloud structure by reading BigQuery and Cloud Storage documentation. We found that all of the relevant data for the data processing is located in the Cloud Storage unit of Google Cloud.

The application communicates with Google Cloud by feeding SQL statements into BigQuery and retrieving data from the Cloud Storage unit. The dashboard defines the SQL statements by letting the prosecutor insert parameters that get inserted into the SQL statement. The Statements are then handled by BigQuery, making it possible to collect data from the Google Cloud storage. The data coming out of the Google Cloud environment is then processed. This data processing consists of calculating estimates through logistic regression and categorizing data by inserted parameters and confidence intervals. The data processing results are visualized in a coordinate system inside the dashboard.

6.1.1 Modular Processes Design

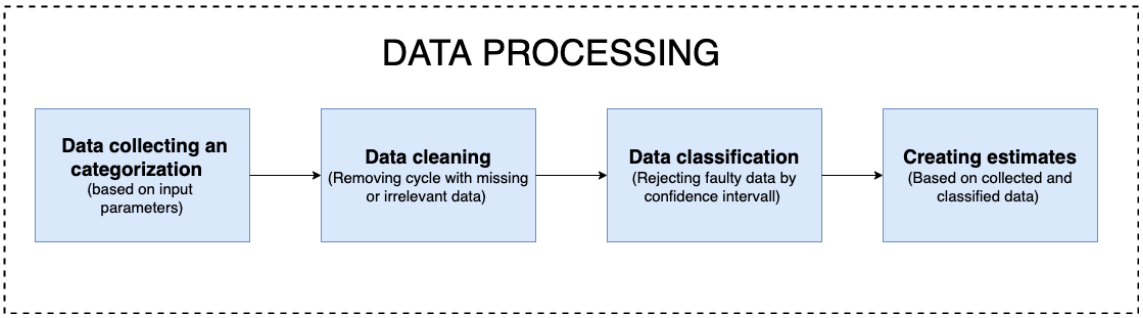


Figure 38: Module Process visualizing the data processing steps (created in Draw.io)

The data processing modules consist of several steps that result in a quality estimate. These modules happen in the order visualized above.

- **Data collection and categorization** - Data is collected from BigQuery and categorized based on the parameters inserted by the prosecutor. This makes sure that the estimate created and previous data collected agree with the type of timber that is being dried.
- **Data cleaning** - Cycles with missing data or irrelevant data are removed. This module makes sure that cycles with incomplete or missing data are not influencing the estimate.
- **Data classification** - A confidence interval is created that rejects data that is too distant from the mean. The classification makes sure large discrepancies do not affect the estimates.
- **Creating estimates** - The estimates are created using the collected, categorized, cleaned, and classified data (from the previous modules). A linear and logistic regression calculation takes the data and creates the estimate.

The result from the different modules is represented by visualizations in the GUI Dashboard. The live data is plotted as it is collected and categorized in the first module. The cycles based on previous data are plotted after the second module has cleaned the data for missing or irrelevant data inside the cycles. The estimation is plotted after the fourth and last module, creating the estimates by calculating a logistic growth model. These connections can be seen in figure 39 below.

6.1.2 Modular Process Design And Their Connection To GUI

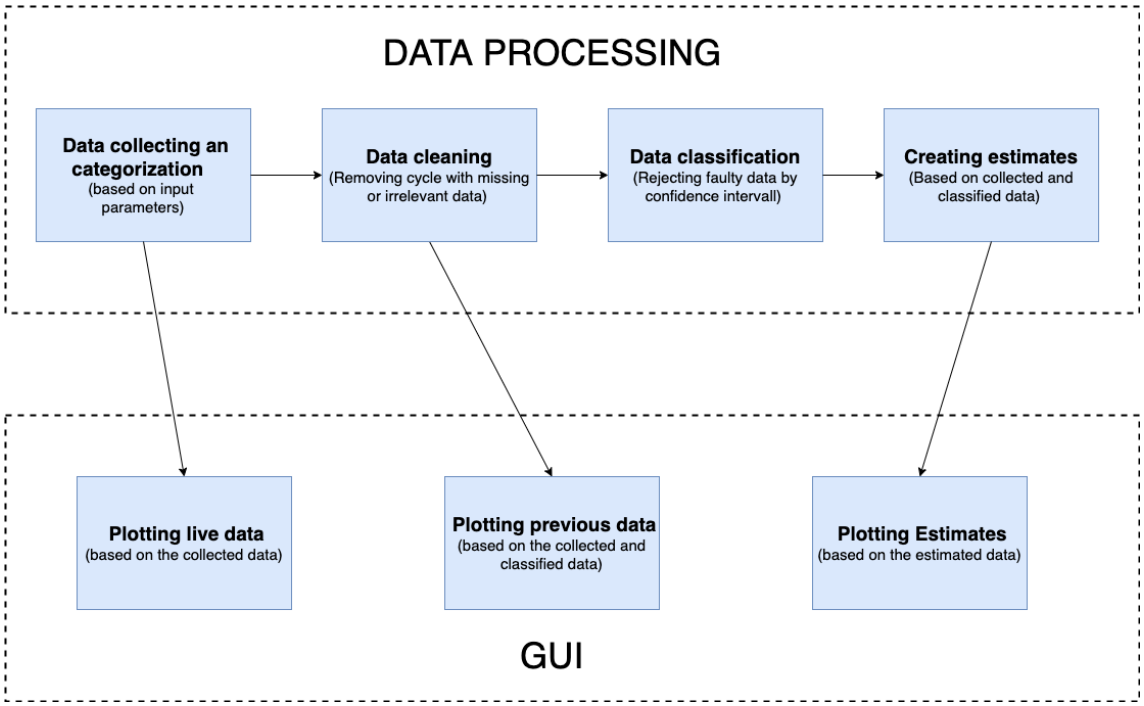


Figure 39: Module Diagram visualizing the different steps the data goes through and when they get sent to the GUI, (created in Draw.io).

6.2 GUI

6.2.1 Dashboard

The primary instance of the graphical interface for the application is the dashboard. The dashboard is one of the most important components, considering all data and visualizations are presented through this dashboard. The application would be useless if the dashboard were not understandable for the operators with limited data experience.

6.2.2 Sketches

In the early stages of the development process, sketches of the application were created. These sketches provided a visualization of possible solutions for the application, with its functionality and information. There was some time spent making detailed sketches to ensure that the dashboard fulfilled the visualization of Moelven. These sketches became useful when trying to reach for the requirements for both UI and UX.

Figma was the interface design tool used to create the sketches. Some of the first sketches was also made in photoshop. The sketches in Figma were beneficial for mapping the functionality needed and

exchanging the ideas of the application with Moelven. The illustrations were presented to Moelven in a meeting (F). The sketches made it easier for Moelven to give feedback on our solution and point us in the right direction early on. Each sketch became increasingly accurate for our solution and suitable for our client's desires. Through meetings with Moelven, we learned that operators with limited experience in data programs would operate and overlook our application. This means that the interface has to be user-friendly and intuitive. The final solution has to be simple but robust, which means the underlying code and processing of the program are advanced, but the way it is presented is easy to understand.

Sketches presented to Moelven

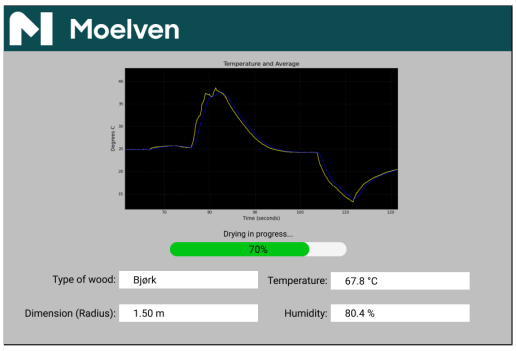


Figure 40: First Sketch

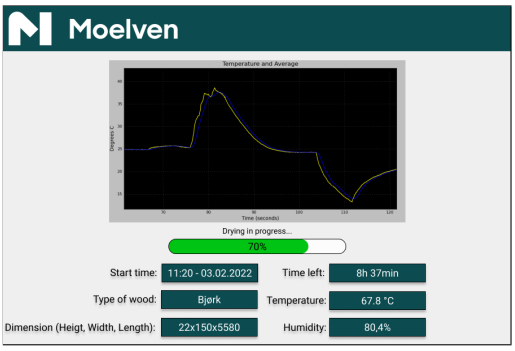


Figure 41: Second sketch

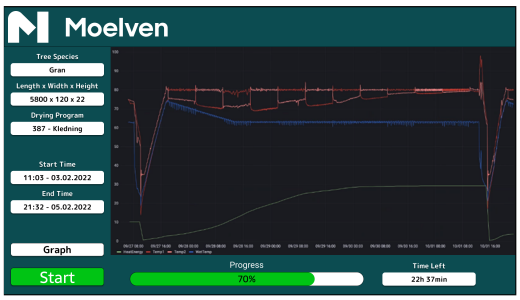


Figure 42: Third sketch

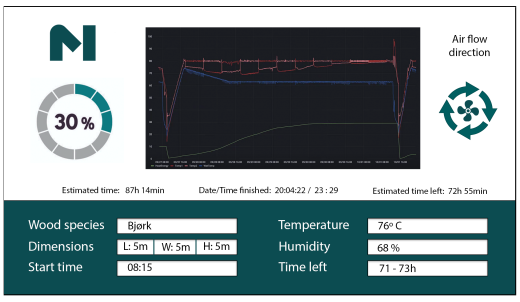


Figure 43: Fourth sketch

Figure 44: These four figures display the sketches we had. They were created in figma and photoshop

6.2.3 Necessary Information

To reach for simplicity and ease of use for the GUI, it was important only to display necessary information.

Moelven informed us that too much information displayed will only confuse the operators. The information we needed to display was connected to the input parameters inserted, the calculated,

and the previous drying cycles. We had this in mind when making and improving the sketches. The presentation of our sketches to Moelven enlightened us that the fourth sketch 43 had some unnecessary information. These wore *Start time*, *Humidity* and *Temperature*. We also made another fourth sketch option, with information displayed about the accuracy of the estimates and a correlation between an algorithmic estimate and a data-based estimate. This became overly complicated and unnecessary compared to our client's desires and, therefore, quickly rejected.

6.2.4 Coordinate View

The most important section of the dashboard is the graph visualization. We established early on that Moelven wanted a graphical view of the processed data and decided to have a main section with a coordinate system in our UI. Live cycle data, estimates made by regression, and previous drying cycles were plotted in the coordinate system. By plotting both live data together with estimates and prior data, the operators can use data estimates and experience to decide when the ongoing drying cycle is finished.

6.2.5 Progress Bar

We started off believing a linear progress bar would be the best solution but quickly realized that we instead wanted a circular progress visualization. The main reasoning for this was to leave more space for the other sections of the dashboard. From our experience, the linear bar would occupy more space. The GUI essentially consists of one main page, which is the dashboard, while the remaining space is reserved for other elements needed to be displayed. Our objective was to obtain a structured and spacious display instead of being compact and stacked with information. Another reason for choosing a circular progression was that it felt more visually pleasing, and it made it possible to put the percentage progression inside the circle, which was convenient.

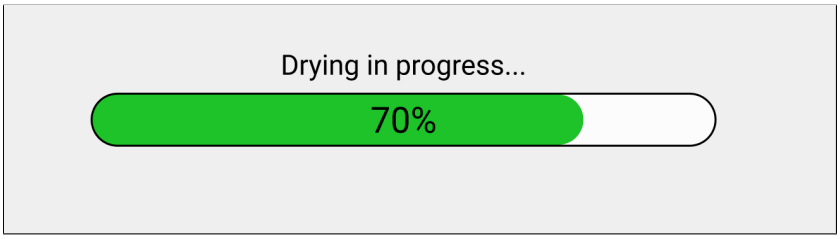


Figure 45: Linear progress bar, created in figma.

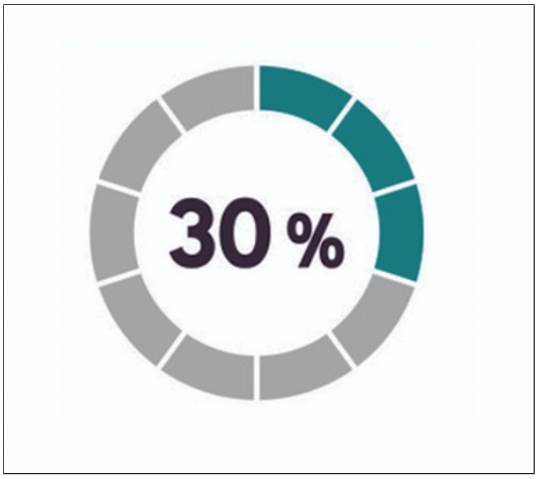


Figure 46: Circular progress visualization, created in figma.

6.2.6 Estimated End Time

To clarify for the operators, when a drying cycle is estimated to be finished, a count-down timer that shows how much time there is estimated to be left was created. The visual graph representations and the estimated time left, and the operator's experience conclude when the timber has reached desired humidity, and the drying cycle is finished. We're also added.

The initial thought was to have a square that got created by horizontal and vertical lines that visualized in what area the desired humidity is reached.

6.2.7 Input Parameters

Before starting the application and data processing, it is necessary for the operator to insert parameters that specify the cycle. The group concluded with having a pop-up window centered in the dashboard that is displayed after pressing the *Start* button. The input parameter window needed to be simple and visible for the operators to understand and insert the parameters correctly. To ensure that the parameters are inserted correctly, a drop-down menu with parameter options for each parameter and error handling was desired. If the parameters are not inserted correctly, an error message should appear that informs the operators of the issue. An operator may insert as many parameters as required; all parameter boxes need to be filled.

6.2.8 Manual Check

A similar concept was designed for the manual check as input parameters. A pop-up window should appear when the drying cycle is finished, and a manual moisture check should be performed by the operator. The value from the manual review decides if the cycle is completed or if it needs a longer time. If finished, the moisture is inserted into the database. These moisture values are

stored because of the ability to adjust later the estimates based on the actual moist results (more about this in section) 11.1).

6.2.9 Logos

During initial conception, the Moelven logo was planned to be shown in the application. For this to be possible, Moelven's guidelines for usage of their logo needed to be followed. Moelven wanted its logo to illuminate the core of the business and has established some rules and guidelines for its logo usage. The logo needs to be presented on a calm background and a good contrast between the background and logo. If the logo is not shown in the original Teal color, it can be shown in black or white. Through their brand center, they have a profile manual that establishes the wrong usage of their logo. We used this as a basis for the use of their logo in our UI:

- The logo should never be used in colors other than those in the logo variants
- The logo symbol should never have colors other than those in the logo variants
- The logo must not be rotated
- The logo should never be scaled in such a way that the proportions are not one to one
- The logo should never have graphic effects like a shadow
- Outline variant should not be used
- The logo should not be used with a slogan, signature, or company name close to it

Figure 47: Right use of Moelven logo, the teal colored version (downloaded from profile manual (*Moelvens logo* 2022))

6.2.10 Colors

Moelven uses their color to express their identity. It is essential for them to leave a teal impression. Therefore, the group wanted to use their teal color called *Moelven teal* and only the colors shown

in their profile manual.

Their profile handbook says:

Colors are important identity carriers, and fidelity to the brand palette should always be strived for. This means that all produced graphic material must follow the guidelines for using colors. (*Farger* 2022)

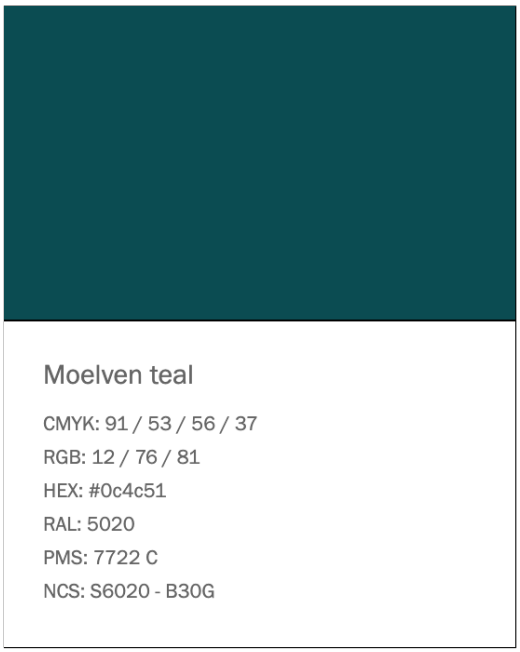


Figure 48: Moelven teal color from their profile manual

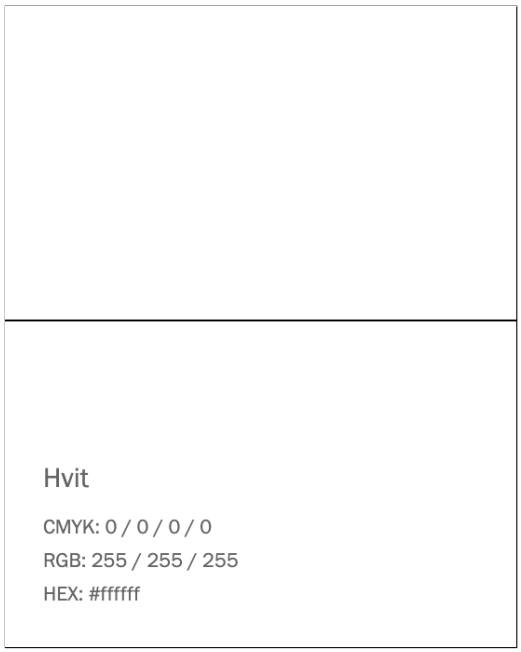


Figure 49: Moelven white color from their profile manual

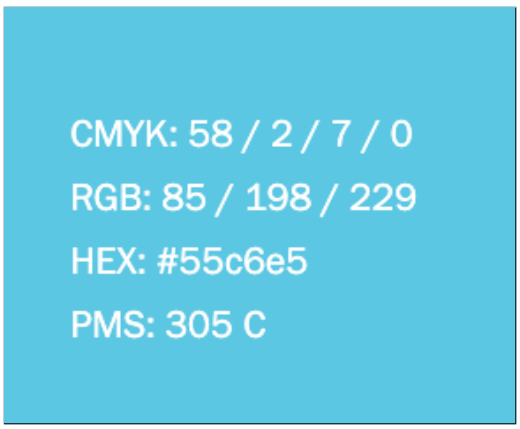


Figure 50: Moelven blue color from their profile manual

Figure 51: Moelven red color from their profile manual

Figure 52: Figures displaying different color palettes. Figures gathered from, (*Farger* 2022).

6.3 Final Sketch

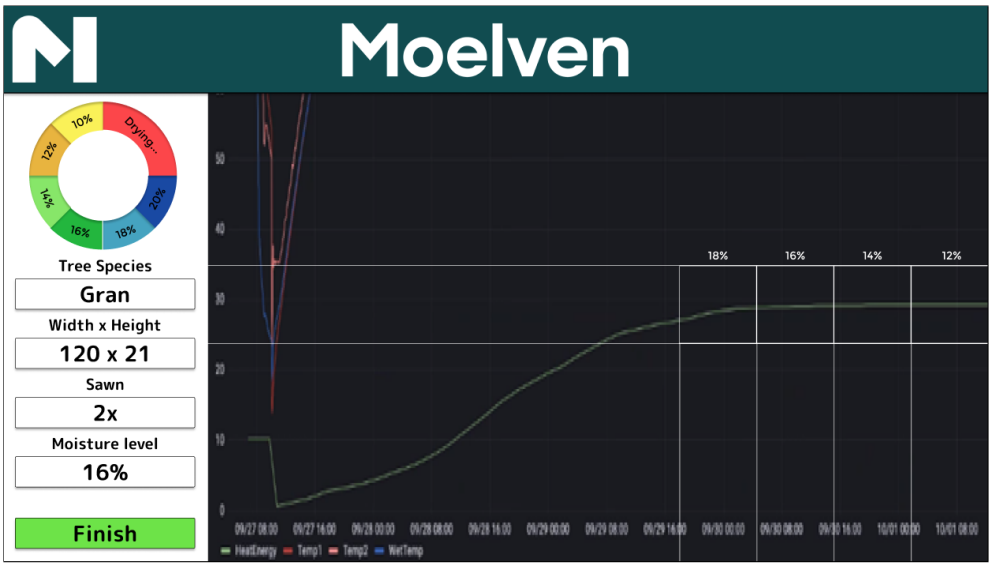


Figure 53: Final sketch of the dashboard where the sketch is created in Figma. The x-axis(horizontal) represents the time axis and have the unit: index point. The y-axis(vertical) represents the kWh.

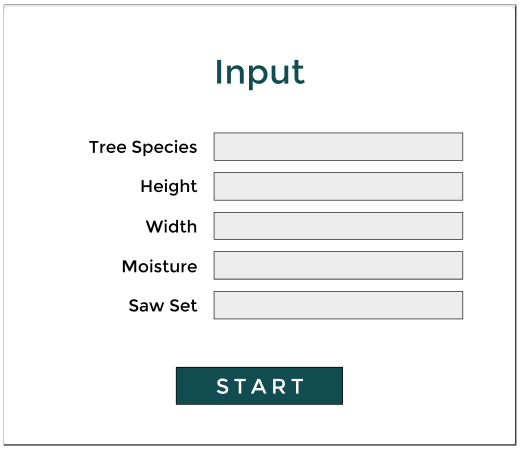


Figure 54: Final sketch for input parameters window, created in Figma.

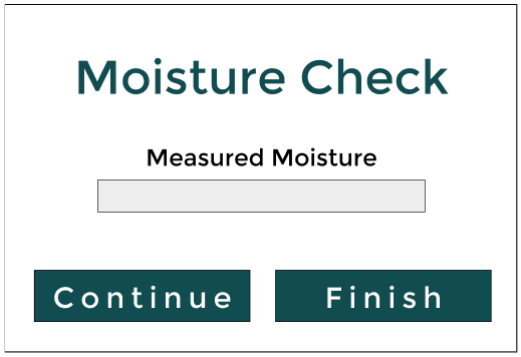


Figure 55: Final sketch for manual moisture input window, created in Figma.

A final sketch was made based on feedback from Moelven, the survey, and our opinions. The final sketch was used as inspiration when creating the actual GUI prototypes and the final application. The group did a survey based on the last sketch to see if the layout and colors were pleasing to everyday people with limited data experience. Link to online survey: (*Survey* 2022)

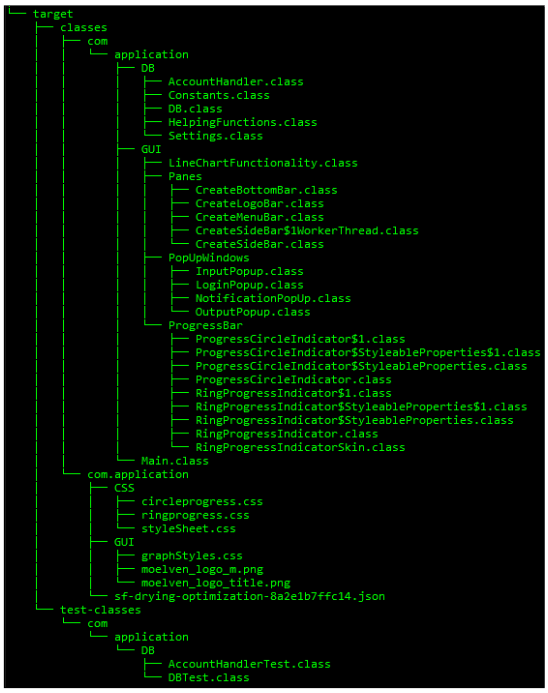
7 Implementation

This chapter describes the requirements, theory, and design choices related to developing the application, which are discussed in earlier chapters. The chapter is structured to follow the data flow in a linear fashion. It begins with the backend connecting to the database. Further, it sends the data over to the frontend to display the data. After that, the backend and frontend work together simultaneously using threads.

7.1 Overview Of The Application

The application follows the standard Maven directory layout. The advantage of following this standard layout is if a programmer is already familiar with this layout, then they will have a easier time navigating the layout (*Introduction to the Standard Directory Layout* 2022). This, in turn, reduces complexity and increases efficiency for a programmer.

(a) Figure displaying the first part of the file structure of the application



(b) Figure displaying the second part of the file structure of the application

Figure 56: The file structure of the application. Both figures are screenshots taken from the command line.

The top layer contains only two subdirectories, src(source) and target, along with some text-based files such as .gitignore, README.txt, and pom.xml. The target subdirectory contains all the output files from the build of the application. The src subdirectory contains all the programming files. The src subdirectory separates Java files, resources, and testing into separate subdirectories.

Java classes are again separated into different functionalities, such as Database-related or GUI-related functions. Resources contain files in order to style the dashboard. This could be images or CSS files. Lastly, testing is separated into a separate directory.

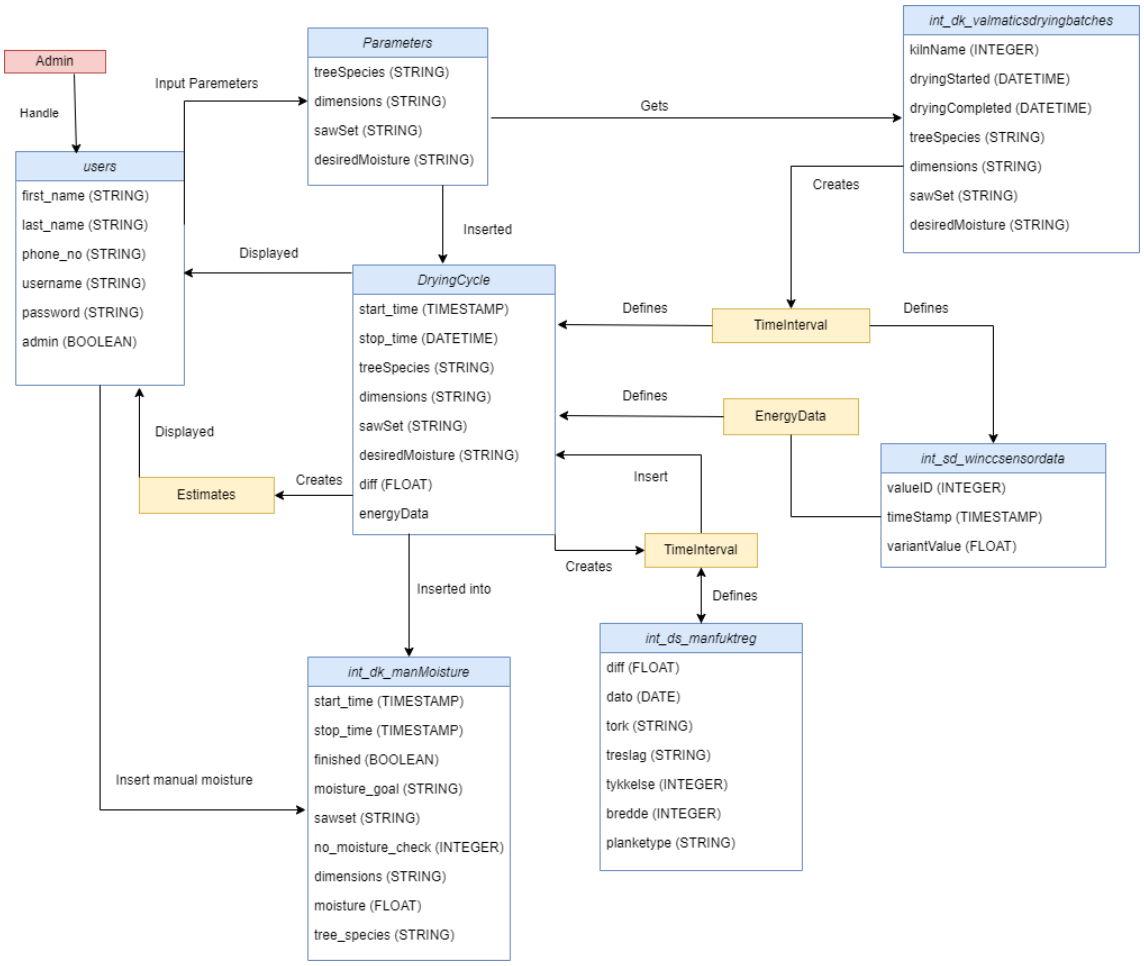


Figure 57: Class entity diagram visualizing the data flow through the frontend, backend and the database (made in Draw.io)

7.2 Backend

Backend refers to the part of a software program that the user does not see and focuses on the processes that are done in the background (*Backend* 2020). Relevant for this application, backend refers to connecting to the database, creating SQL statements, categorizing data, processing and cleaning data, statistics, and lastly, the login system.

7.2.1 Database Connection

All the sensor data needed for this project is stored in the Google Cloud BigQuery system. Therefore the first step was to connect Google Cloud Platform (GCP) to the application. The connection

was established through these four steps (*BigQuery Connection API Client Libraries* 2022):

- Firstly, a private unique key-file was generated;
- Secondly, a connection to the Google Cloud BigQuery API was established;
- Thirdly, a queryjob was built and run in order to send or receive data from the database;
- Lastly, if data were returned from the queryjob, then this data was processed.

Firstly, the key-file had to be generated and handled. The key-file is a unique private file that grants both read- and write access to the *sf-drying-optimization* database. The file was generated through Goggle Cloud BigQuery's website, downloaded, and stored locally on the team members' computers. The file was placed in the .gitignore file in order to prevent uploading to our git repository on GitLab. However, later on, in the development process, the file was moved around in the folder structure and ultimately ended up being uploaded to the repository. This caused a security risk. If the file was obtained by an unauthorized person, they could potentially cause major damage by adding misinformation or altering or even deleting data. It was, therefore, essential that the key-file was unavailable for anyone other than the group members regard to a security perspective.

Secondly, a connection was established using the Google Cloud BigQuery API, which takes SQL statements as input and outputs the results back to the application (*BigQuery API Client Libraries* 2022). The choice of importing the library through maven was made in order to reduce security risks and ensure ease of installation. Compared to downloading the library from a website with a potentially compromised library, this could turn out to be a security risk. The maven dependencies are defined as follows in the pom.xml file:

```
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>com.google.cloud</groupId>
            <artifactId>libraries-bom</artifactId>
            <version>24.3.0</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

<dependencies>
    <dependency>
        <groupId>com.google.cloud</groupId>
        <artifactId>google-cloud-bigquery</artifactId>
    </dependency>
    <dependency>
```

```
        <groupId>com.google.cloud</groupId>
        <artifactId>google-cloud-storage</artifactId>
    </dependency>
<dependencies>
```

After the library was imported, the key-file was loaded in the *getCredentials()* function and tried to gain access to the database:

```
/**
 * Retrieves the credentials file and grants access to the database
 *
 * @return the credentials
 * @throws Exception for potential errors
 */
private static GoogleCredentials getCredentials(String KEY_FILE_NAME) throws
    Exception {
    File credentialsPath = new File("./src/main/resources/com.application/"+
    KEY_FILE_NAME);

    GoogleCredentials credentials;
    try (FileInputStream serviceAccountStream = new FileInputStream(credentialsPath
    )) {
        credentials = ServiceAccountCredentials.fromStream(serviceAccountStream);
    }
    return credentials;
}
```

If access was granted, the next sequence of events was to establish a connection with the database by building query jobs and passing along the SQL statements to the database. This connection was established by creating a builder using the *getbuilder()* function:

```
/**
 * Creates a bigquery builder. Here we set the project ID and get the `BigQuery`
    service object.
 * this is the interface to our BigQuery instance that we use to execute jobs on.
 *
 * @return a builder
 * @throws Exception returns potential error
 */
private static BigQuery getBuilder() throws Exception {
    return BigQueryOptions.newBuilder().
            setCredentials(getCredentials()).
            setProjectId(PROJECT_ID)
            .build().getService();
}
```

After the builder were built, a job and queryjob was created using the functions *getJob()* and *createQueryJob()*:

```java
/**
 * Creates a job for the database
 *
 * @param queryConfig query configuration information
 * @return a job
 * @throws Exception returns potential error
 */
private static Job getJob(JobConfiguration queryConfig) throws Exception {
    // Step 3: Run the job on BigQuery
    // create a `Job` instance from the job configuration using the BigQuery
    service
    // the job starts executing once the `create` method executes
    Job queryJob = getBuilder().create(JobInfo.newBuilder(queryConfig).build());
    queryJob = queryJob.waitFor();

    // the waitFor method blocks until the job completes
    // and returns `null` if the job doesn't exist anymore
    if (queryJob == null) {
        throw new Exception("job no longer exists");
    }
    // once the job is done, check if any error occured
    if (queryJob.getStatus().getError() != null) {
        throw new Exception(queryJob.getStatus().getError().toString());
    }
    return queryJob;
}
```

```java
/**
 * This function creates a query job that uses the query statement
 * in order to retrieve information from the database
 *
 * @param sqlStatement input for the query statement
 * @return returns the queryjob with the results
 * @throws Exception Throws exception in case of error
 */
static TableResult createQueryJob(String sqlStatement) throws Exception {

    // Creates a job configuration
    Job queryJob = getJob(QueryJobConfiguration.newBuilder(sqlStatement).build());

    // Retrieves the results from the queryjob
    return queryJob.getQueryResults();
}
```

After the query job was built, it was ready to pass SQL statements to the database.

Thirdly, the data needed to be sent and received to and from the database, and in order to do this, the database structure needed to be understood. The database consists of data collected from multiple sawmills, where some sawmills collect data in similar ways while others have unique data

collection systems. The original idea was to follow good practice and write dynamic code with high cohesion and loose coupling where multiple sawmills could share the same functions and SQL statements. However, as a consequence of the different database schemes, this turned out to be difficult to achieve. The inconsistency in the data collection systems meant there were differences in the different database schemes for each sawmill, therefore forcing the application to effectively create custom SQL statements for each sawmill. Had the database structure been consistent, then it would be easy to reuse the already existing functions for new sawmills. Moelven decided to give access to data from *Valåsen* and *Årjang* because these sawmills were the most developed with regard to data collection and had a similar data collection system. This made it easier to write dynamic code with high cohesion and loose coupling. More detail about the SQL statements used is described in section 7.2.4.

After the key-file had been generated, the connection with the database had been established, and the SQL statement was sent. If the information was expected to be returned, then the response needed to be handled and iterated through the data. This is explained in further detail in section 7.2.5.

7.2.2 Defining Drying Periods

The data collected by Moelven is stored in the database. There are mainly two tables the application gathers data from, and these are the *int_dk_Valmetics* and the *int_sd_winccsensordata* tables. *int_dk_Valmetics* contains a number of different attributes, such as: Tree species, dimensions, sawset, moisture goal, timestamp-in, timestamp-out, and KilnId. These are the parameters used for categorizing the data based on the input parameters given by the user. *int_sd_winccsensordata* on the other hand contains timestamp and the kWh value. However, the kWh value is always increasing and is therefore never reset. This makes defining each drying period difficult.

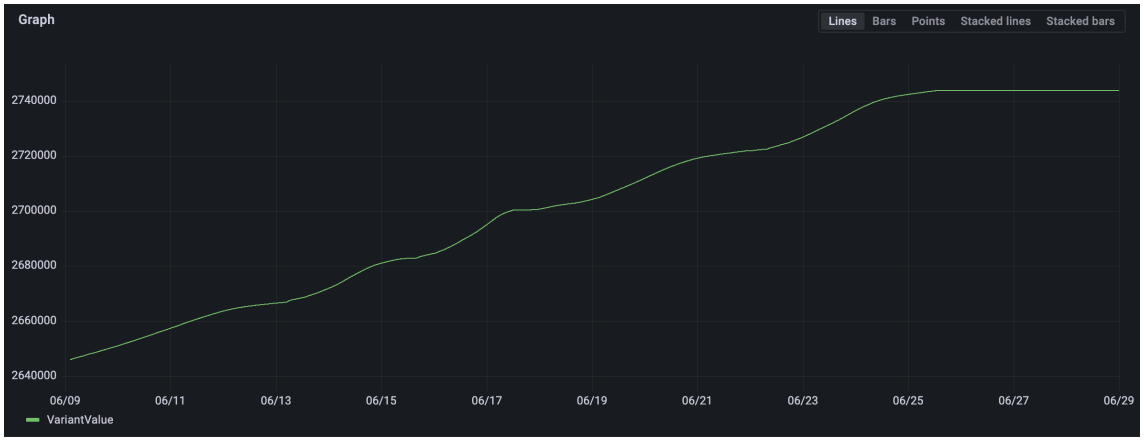


Figure 58: Visualization of increasing energy variable using Grafana (VariantValue). The x-axis(horizontal) represents the time axis and have the unit: date. The y-axis(vertical) represents the kWh.

Figure 58 visualizes the increasing kWh value over multiple drying periods. In order to create an estimate, the data needed to be split up into each separate drying period, as visualized in figure 59.

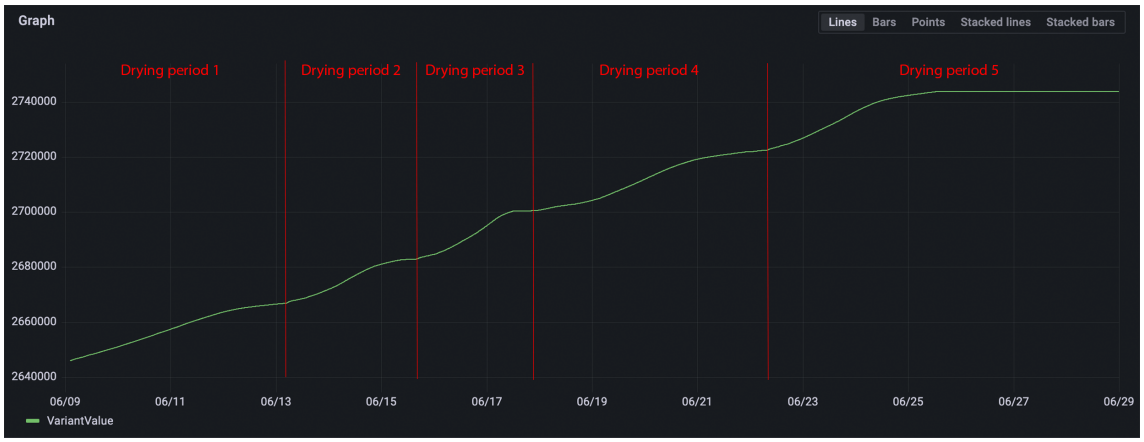


Figure 59: Visualization of increasing energy variable with Drying periods using Grafana (VariantValue with time intervals). The x-axis(horizontal) represents the time axis and have the unit: date. The y-axis(vertical) represents the kWh.

Our first solution was to calculate when the graph leveled out and use this for our start and end time for each drying period. However, this would be too imprecise for the regression. After gaining a better understanding of the database, we found the timestamp-in and timestamp-out attributes in the *int_dk_Valmetics* table. If we could link these timestamps with the timestamp attribute in the *int_sd_winccsensordata* table, then we could find each separate drying cycle.

An optimal database structure would be to link the tables using primary and foreign keys, as it would allow synchronization of the tables. Each row in one table would then contain a unique attribute that corresponds to a unique row in the other table. However, this was not the case for our database, and therefore it was necessary to find a way to synchronize the tables manually. The only unique data type contained in both tables was the Timestamp data type, and therefore this was the only way to connect the two tables.

When implementing this solution the group came across a problem where the *int_dk_Valmetics* table from one sawmill would contain data for Timestamp-in and Timestamp-out data, where as another sawmill would not record this data. We chose to use a different Time-in and Time-out attribute that were a little out of sync with the actual time interval. However, the time difference was around 5-10 seconds, which is negligible compared to a whole drying period that can last multiple days.

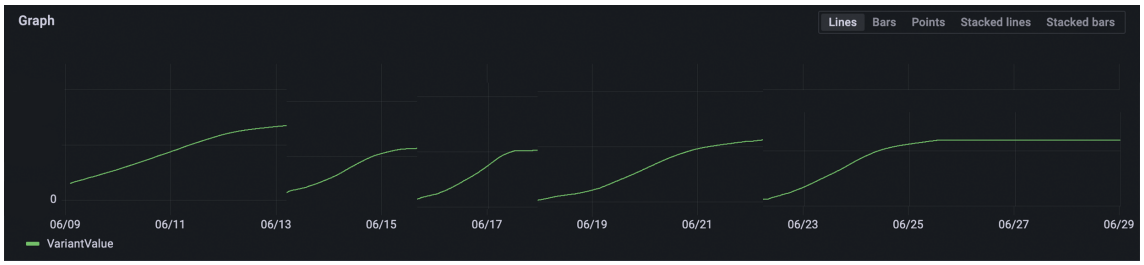


Figure 60: Visualization of energy variable after asserting time interval's and resetting after each period using Grafana (VariantValue). The x-axis(horizontal) represents the time axis and have the unit: date. The y-axis(vertical) represents the kWh.

Figure 60 shows how the data is split into separate drying periods using the timestamp-in and timestamp-out attributes from *int_dk_Valmetics* by searching for kWh data in the *int_sd_winccsensordata* table between the timestamps.

7.2.3 Categorizing Of Data

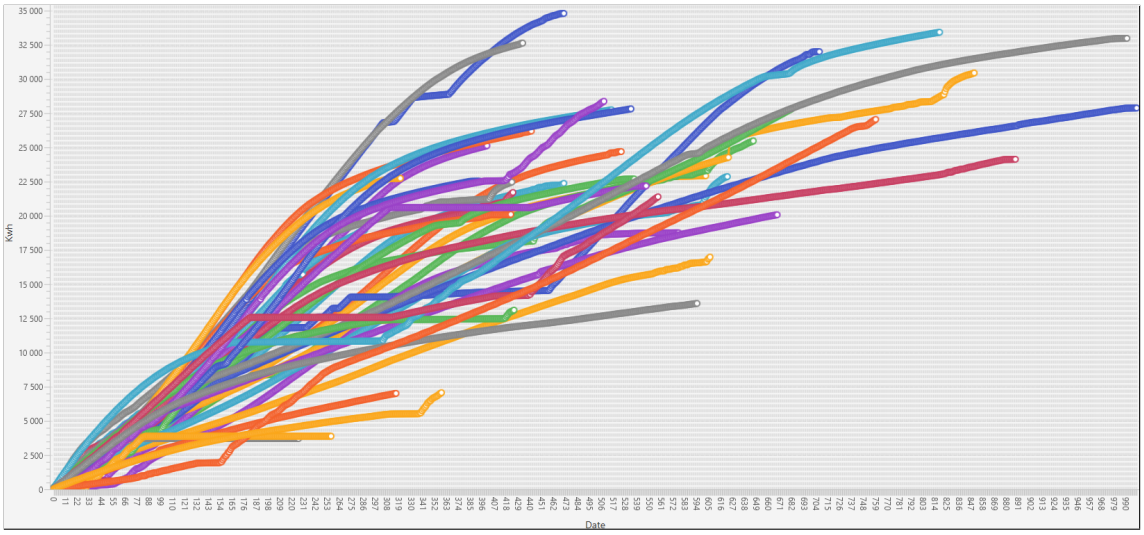


Figure 61: Screenshot of the application displaying 42 different drying periods. The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Drying periods can produce a great level of variety in the data depending on multiple factors. This is visualized in figure 61. When calculating the estimate, the data must be as accurate as possible in order to maximize the probability of predicting when a drying period is completed. Therefore, it was decided to categorize the data based on the most important factors. The most important factors consist of: Tree species, dimensions, sawset, and moisture goal. When searching for data, the application will take these factors into consideration by basing the search on input parameters given by the drying operator.

If there are no data for the input parameters given, then the estimate has no data to be based upon. To combat this issue, a priority system was created where the parameter with the least amount of impact on the estimate will be deleted, then tried to search for data again. If there is still no data, then another parameter will be deleted, and the data search will recommence. This process will continue until data is found or all the parameters are deleted. This priority system will ensure that data will be found. The categorization will increase the accuracy of the estimate thanks to only taking relevant drying periods into consideration. However, the more parameters are deleted, the worse the quality of data gets. Thankfully, Moelven will continue to collect data in the future, and at a point, the application won't have to delete parameters in order to find data anymore. This means that the accuracy of the estimate will increase over time.

7.2.4 Database Connection SQL Statements

In order to retrieve the data from Google Cloud, the group had to run queries and handle the query results. As described in detail earlier in section 7.2.1, a query job takes an SQL statement as input and forwards it to the database.

SQL injections are a major security risk, as discussed later in section 9.2. Prevention has been made in order to reduce the likelihood of an SQL injection taking place. Input parameters take a maximum of 10 characters and do not allow for the keyword "UNION" as an input in order to prevent an SQL injection from happening.

After the input parameters have been checked for SQL injections, the SQL statements are run through:

```
SELECT MAX(Name) as DryingSchedule, MAX(KilnName) as Kiln_ID, DryingStarted, MAX(
    CalculatedStop) as DryingCompleted
FROM `sf-drying-optimization.124.int_dk_valmaticsdryingbatches_v2`
WHERE KilnName = 5
AND DryingStarted BETWEEN "1990-01-01 00:00:00" AND "2022-05-12 16:56:07"
AND CalculatedStop BETWEEN "1990-01-01 00:00:00" AND "2022-05-12 16:56:07"
AND CalculatedStart BETWEEN "1990-01-01 00:00:00" AND "2022-05-12 16:56:07"
AND LOWER(Name) LIKE LOWER("%Gran%")
AND LOWER(Name) LIKE LOWER("%47x200%")
AND LOWER(Name) LIKE LOWER("%2ex%")
AND LOWER(Name) LIKE LOWER("%16\\%%")
AND LOWER(Name) NOT LIKE LOWER("%test%")
Group by DryingStarted
Order by DryingStarted DESC
LIMIT 1000
```

The SQL statement gathers data from the *int_dk_Valmetics* table and tries to find all drying periods matching the given input parameters. In this case, the input parameters are as follows: Gran, 47x200, 2ex, and 16%.

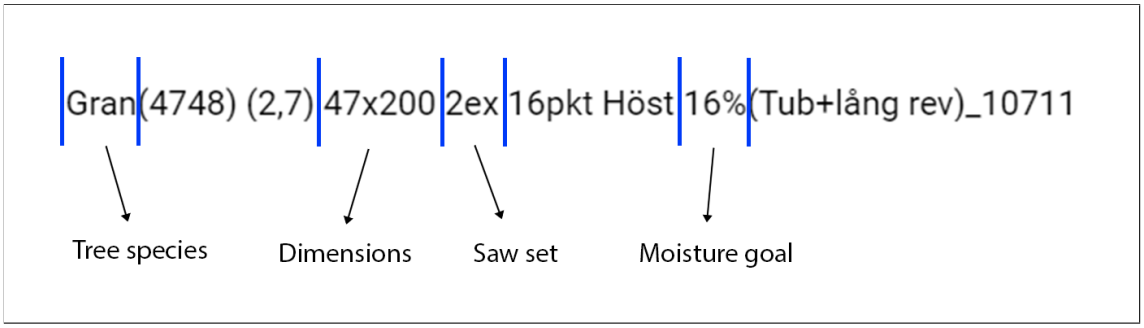


Figure 62: Dividing "Name" attribute into multiple input parameters. Created in Photoshop

Optimally, the user input parameters would be stored in separate attributes, but as visualized in figure 62, the *Name* parameter contains all the user input parameters. In order to retrieve the information, the SQL statement checks if the *Name* parameter contains a part that equals the user input parameter.

In addition to the user inputs, the SQL statement search for data between dates, from a hardcoded value "1990-01-01 00:00:00" and "today's" date. In addition to the parameters, the *kilnName* or *kilnID* specify which drying chamber to search for data. A limit is also set for good practice in order to limit the maximum number of drying cycles allowed to be returned. A limit of 1000 was chosen in order to gather as many drying periods as possible. Nevertheless, when 1000 drying cycles are reached, the application should have enough data to calculate a good estimate. If the drying cycles would exceed the limit, then it would be important that the cycles were retrieved in descending order, which is why the SQL statements are ordered by *DryingStarted* DESC. This way, the most up-to-date information would be retrieved first, and when reaching the limit of 1000, the least relevant information would be lost. If the order was in ascending order, the most relevant and most up-to-date information would be lost instead.

After the input parameters have been checked for SQL injections, the SQL statements are run through:

Row	DryingSchedule	Kiln_ID	DryingStarted	DryingCompleted
1	Gran(3996) (2,7) 47x200 2ex 16pkt Vinter 16%_11133	5	2022-01-28T02:47:27	2022-01-31T23:51:50
2	Gran(4748) (2,7) 47x200 2ex 16pkt Höst 16%(Tub+lång rev)_10757	5	2021-10-10T11:01:34	2021-10-13T16:37:31
3	Gran(4748) (2,7) 47x200 2ex 16pkt Höst 16%(Tub+lång rev)_10711	5	2021-09-30T18:35:21	2021-10-04T01:19:19
4	Gran(4748) (2,7) 47x200 2ex 16pkt Höst 16%(Tub+lång rev)_10582	5	2021-08-28T10:34:38	2021-08-31T14:57:39
5	Gran(4357) (3,4) 47x200 2ex 16pkt Vår 16%(sim 17%+lång rev)_10484	5	2021-06-16T21:32:20	2021-06-19T22:57:33

Figure 63: Results of the SQL statement defined above gathering data from the *int_dk_Valmetics* table showing drying started- and drying completed timestamps for each unique drying cycle. Screenshot taken from Google Cloud.

Figure 63 shows the result of the SQL statement above, which will give the drying started and

drying completed timestamps for each unique drying cycle found in the *int_dk_Valmetics* table. These timestamps can be used to search for kWh values in the *int_sd_winccsensordata* table using this SQL statement:

```
SELECT DISTINCT `Timestamp`, `VariantValue`
FROM `sf-drying-optimization.124.int_sd_winccsensordata`
WHERE Timestamp BETWEEN "2022-01-29 06:37:47" AND "2022-02-02 08:07:06"
AND ValueID = 56
AND VariantValue <> 0
ORDER BY Timestamp ASC
```

The SQL statement above retrieves kWh data from the attribute "VarientValue" for one drying cycle in the *int_sd_winccsensordata* table. The SQL statement takes started and drying completed timestamps from the *int_dk_Valmetics* table as input parameters and searches for kWh data between these timestamps. ValueID determine which drying chamber to search for data in, and VariantValue ⟨⟩ 0 determine that values of 0 are not allowed. Timestamp is ordered in ascending order. This gives the lowest kWh value first so that a baseline can be set and kWh can be reset to start at 0 in the code.

Row	Timestamp	VariantValue
1	2022-01-29 06:39:14 UTC	26058702.0
2	2022-01-29 06:49:14 UTC	26058702.0
3	2022-01-29 06:59:14 UTC	26058702.0
4	2022-01-29 07:09:14 UTC	26058702.0
5	2022-01-29 07:19:14 UTC	26058702.0
6	2022-01-29 07:29:14 UTC	26058702.0

Results per page: 50 1 – 50 of 585

Figure 64: Results of the SQL statement defined above, gathering kWh data from the "VarientValue" attribute in the *int_sd_winccsensordata* table. Screenshot taken from Google Cloud.

Figure 64 displays the results of the sql statement retrieving kWh data from the *int_sd_winccsensordata* table. The query returns 585 rows of data points to display in the application. After all the data is received from the database, the data needs to be processed by the application.

7.2.5 Processing Response Data

When the application recessives data from the database, it is given in a *TableResult* datatype that can be iterated through. The getZeroPointDate() function retrieves data from the *int_dk_Valmetics* table.

```
/**
 * This function retrieves the StartedDrying and DryingCompleted dates from the
    database.
 * These parameters are then passed to the iterateKwhValues() functio,
 * in order to find Kwh values for each drying cycle.
 *
```

```
 * @return Returns a treemap that sorts the Start- and End time for each drying
    period incrementally
 * @throws Exception Throws exception if an error occurs
 */
private static Map<String, String> getZeroPointDate(int locationID, String
    extraUserInput, String nameParameter,
                                                    String kilinName, int kilinID,
    String startDryingTime,
                                                    String stopDryingTime, String
    valmeticsTableName, int limit)
                                                    throws Exception {

    ....
    // Purposely omitted code, already discussed, SQL statement
    ....

    // Iterating through the results
    for (FieldValueList row : result.iterateAll()) {

        // Retrieving the data
        // DryingStarted:
        if(!row.get("DryingStarted").isNull()){

            // Check if response is given in millis
            try{
                long doubleValue = row.get("DryingStarted").getTimestampValue();
                long InTidTorkLong = doubleValue/1000;
                // Formating the data from long to a string in the correct date
    format
                formatedInTidTork = HelpingFunctions.getDateFormat().format(
    InTidTorkLong);
            } catch(NumberFormatException e){
                //not long value, must be of type string
                if(row.get("DryingStarted").getValue().toString().contains("T")){
                    // stores the value
                    String value = (String) row.get("DryingStarted").getValue();
                    // Splits the string based on 'T'
                    String[] splitValue = value.split("T");
                    // Combines the values into a new format
                    formatedInTidTork = splitValue[0]+" "+splitValue[1];
                }
                else formatedInTidTork = row.get("DryingStarted").getValue().
    toString();
            }
        }

        // DryingCompleted:
        // Check if response is given in millis
        try{
```

```
48              long doubleValue = row.get("DryingCompleted").getTimestampValue();
49              long utTidTorkLong = doubleValue/1000;
50              // Formating the data from long to a string in the correct date format
51              formatedUtTidTork = HelpingFunctions.getDateFormat().format(
    utTidTorkLong);
52          } catch(NumberFormatException e) {
53              //not long value, must be of type string
54              if(row.get("DryingCompleted").getValue().toString().contains("T")){
55                  // stores the value
56                  String value = (String) row.get("DryingCompleted").getValue();
57                  // Splits the string based on 'T'
58                  String[] splitValue = value.split("T");
59                  // Combines the values into a new format
60                  formatedUtTidTork = splitValue[0] + " " + splitValue[1];
61              }
62              else formatedUtTidTork = row.get("DryingCompleted").getValue().toString
    ();
63          }
64
65          // Checks if intidtork or outtidtork is empty, if so they are ignored and
    not added to the list
66          if (!formatedInTidTork.isEmpty() && !formatedUtTidTork.isEmpty()){
67              // Adds the data to the dates map
68              dates.put(formatedInTidTork,formatedUtTidTork);
69          }
70      }
71
72      // Returns a treemap that sorts the dates incrementally
73      return new new TreeMap<>(dates);
74  }
```

The data received from the *int_dk_Valmetics* table are the start and end times of the drying cycles in the form of a Timestamp datatype. More information about this will be given later. The timestamps are further forwarded to iteratekWhValues() function:

```
1  /**
2   * Iterates through all the Kwh values and storing them in a map
3   *
4   * @param data a map to store all the data
5   * @param result TableResult to iterate through
6   * @param kwhNameParameter Name of the Kwh name parameter in the database
7   * @param kwhTimestampNameParameter Name of the timestamp parameter in the database
8   */
9  static void iterateKwhValues(Map<String, Number> data, TableResult result, String
       kwhNameParameter, String kwhTimestampNameParameter) {
10     // A baseline to base all the next values of to get a zero point
11     int baseline = 0;
12
13     for (FieldValueList row : result.iterateAll()) {
```

```
        // Sets the baseline in order to reset the kWh counter
        if (baseline == 0) {
            baseline = row.get("" + kwhNameParameter + "").getNumericValue().
    intValue();
        }

        // kWh value
        int variantValue = row.get("" + kwhNameParameter + "").getNumericValue().
    intValue() - baseline; //-baseline

        // Retrieving the wanted data
        long timeStamp = row.get("" + kwhTimestampNameParameter + "").
    getTimestampValue() / 1000;
        // Riktig format, men i string
        String formatedTimeStamp = getDateFormat().format(timeStamp);

        // Checks for negative values
        if (variantValue > 0) {
            // Adding the data to a list in order to sort through later
            data.put(formatedTimeStamp, variantValue);
        }
    }
}
```

The iteratekwhValues() function iterates through the TableResult and finds rows that corresponds to the rows in the SQL statements. If no baseline is set, then the first row with the "kwhNameParameter" parameter will get set with .getNumericValue().intValue().

After the baseline is set, the VarientValue is retrieved and the baseline is subtracted in order to start the kWh values at 0, as described in section 7.2.2.

When retrieving Timestamp dates from Google Cloud to Java environment, it is formatted as the current instance of the clock in milliseconds. This format is called "Java millis", and is a part of the Java DateTime API. The millisecond instant is measured from 1970-01-01 00:00:00Z (UTC) to the current date/time. To convert these millisecond back to an understandable date format, we had to make a new SimpleDateFormat("yyyy-MM-dd HH:mm:ss"). However, some tables stored the date/time data in a DateTime datatype that follows the ISO 8601 standard represented as YYYY-MM-DDTHH-MM-SS (**Iso˙8601˙standard**). In order to format the data type to the previously implemented datatypes, we needed to separate the DateTime response based on the "T" character contained in the ISO 8601 standard. The formatting was done using the Joda time library (*Why Joda Time?* 2022).

As shown in the code bellow:

```
String[] splitValue = value.split("T");
// Combines the values into a new format
formatedInTidTork = splitValue[0]+" "+splitValue[1];
```

If VarientValue have a positive value over 0, then the formated timestamp and VarientValue is added to the list and stored.

7.2.6 Statistics

The application uses a combination of confidence interval and different types of regression in order to produce a logistic growth model.

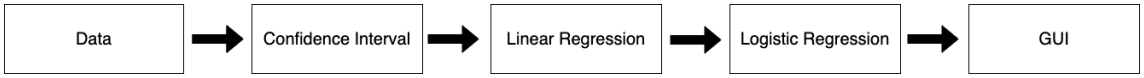
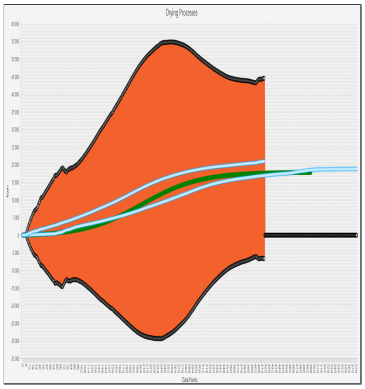
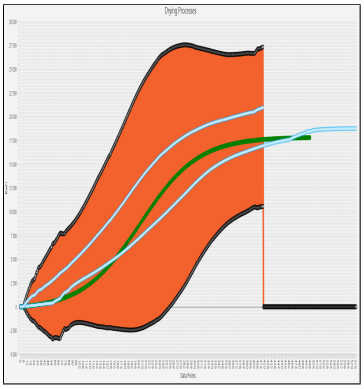


Figure 65: Visualisation of how the data flows through the different statistics functions (made in Draw.io)

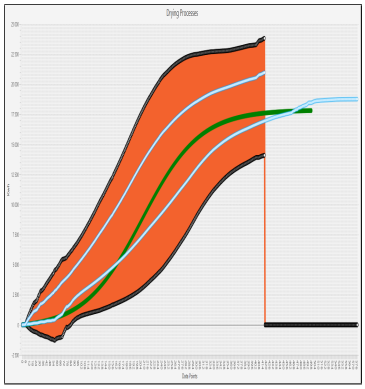
The application receives the data from the database and processes it via a confidence interval in order to exclude irrelevant data before sending the data to the regression models. This way, the growth model is as accurate as possible.



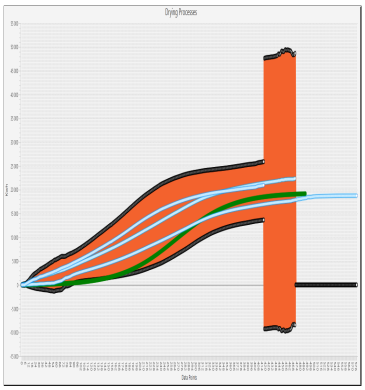
(a) 95% confidence interval, 2 graphs



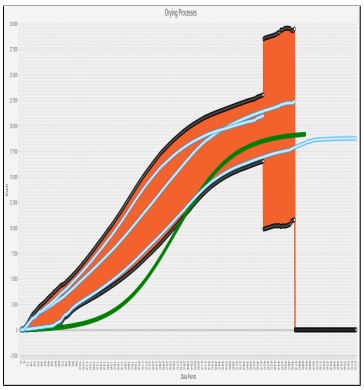
(b) 85% confidence interval, 2 graphs



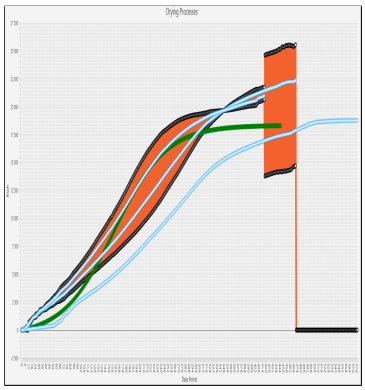
(c) 75% confidence interval, 2 graphs



(d) 95% confidence interval, 3 graphs



(e) 85% confidence interval, 3 graphs



(f) 75% confidence interval, 3 graphs

Figure 66: Visualization of different levels of confidence interval. All figures are screenshots from the application, where the x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Figure 66 displays different levels of confidence interval for graph two and three. The visualization is meant to visualize how the interval changes for different levels. When data falls outside the interval, the data is disregarded and is not taken into account when calculating the regression growth model. This can be seen in figure 66f.

Currently, there is an amount of data in the database which is limited, and as discussed in section 4.5.1, the confidence interval becomes exponentially worse with little data. This means that in the current state of the application, a confidence level of 75% could be considered satisfactory. However, as more data is added to the database, a 95% level would be preferable.

The confidence interval is calculated at each point along the x-axis and takes into account up to multiple points along the y-axis at each given x-axis point. This means that at a given x-axis point, there can only be one point. However, at each x-axis point, there could be several points along the y-axis. The confidence interval takes into account the number of graphs and calculates the relevance of each graph, and if the graph is irrelevant, the graph data.

The data is sent as Map to the statistics() function, which calls upon the calcMeanCI() function:

```
/**
 * This function calculates the confidence interval and removes any irrelevant data
 *
 * @param multiMap a map containing all the data
 * @param CIShadow if this variable is true, then the lower and upper limits
 *                 of the confidence interval is stored instead of the drying process data,
 *                 Intended for graphing the shadow of the confidence interval
 * @return a map containing only relevant data inside the confidence interval
 */
private static Map<Integer, ArrayList<Double>> statistics(Map<Integer, ArrayList<Double>> multiMap, boolean CIShadow){

    for (Map.Entry<Integer, ArrayList<Double>> entry : multiMap.entrySet()) {
            SummaryStatistics stats = new SummaryStatistics();
            for (double val : entry.getValue()) {
                stats.addValue(val);
            }

            // Calculate 95% confidence interval
            double ci = calcMeanCI(stats, Settings.CONFIDENCE_INTERVAL);
            double lower = stats.getMean() - ci;
            double upper = stats.getMean() + ci;

            // Deletes entries if they are out of bounds with the confidence interval
            entry.getValue().removeIf(value -> Double.compare(value, lower) < 0 || Double.compare(value, upper) > 0);

            if(CIShadow){
                ArrayList<Double> lowerUpperBounds = new ArrayList<>();
                lowerUpperBounds.add(lower);
                lowerUpperBounds.add(upper);
                multiMap.replace(entry.getKey(), lowerUpperBounds);
            }
    }
    return multiMap;
```

```
/**
 * This function creates a T distribution, critical value and the confidence interval
 *
 * @param stats containing the data
 * @param level the level of confidence interval wanted. E.g 95%
 * @return
 */
private static double calcMeanCI(SummaryStatistics stats, double level) {
    try {
```

```
        // Create T Distribution with N-1 degrees of freedom
        TDistribution tDist = new TDistribution(stats.getN() - 1);
        // Calculate critical value
        double critVal = tDist.inverseCumulativeProbability(1.0 - (1 - level) / 2);
        // Calculate confidence interval
        return critVal * stats.getStandardDeviation() / Math.sqrt(stats.getN());
    } catch (MathIllegalArgumentException e) {
        return Double.NaN;
    }
}
```

The map returned has excluded irrelevant data and are ready to be sent to the linear regression:

```
SimpleRegression simpleRegression = new SimpleRegression();
simpleRegression.addData(data);
```

The linear regression is calculated using the org.apache.commons.math3.stat.regression.SimpleRegression library.

From the linear regression, two variables are further used; these are the slope and the intercept calculated. These variables are used in the logistic regression in order to estimate the growth model. The following function getNonLinearRegression() calculates the logistic regression:

```
/**
 * This function calculates a logistic regression based on the input parameters
 *
 * @param confidenceIntervalData a map containing data points after the confidence
    interval have taken place
 * @param y0 are the intercept value from linear regression
 * @param alpha are the slope value from linear regression
 * @param j are the index along the x-axis (time)
 * @param n total number of indexes along the x-axis (time)
 * @return returns the predicted value at point j
 */
public static double getNonLinearRegression(Map<Integer, ArrayList<Double>>
    confidenceIntervalData, double y0, double alpha, double j, double n) {

    double beta = getDataPointsYAxis()+y0;
    return (((beta * y0))/(y0 +((beta- y0)*Math.exp(-alpha*j/n/((alpha/10))))))-y0;
}
```

The getNonLinearRegression() function takes a number of different input parameters and predicts a value for each point along the x-axis(timeline). Once all the data points are predicted, they can be put in a series and plotted as a graph on the dashboard.

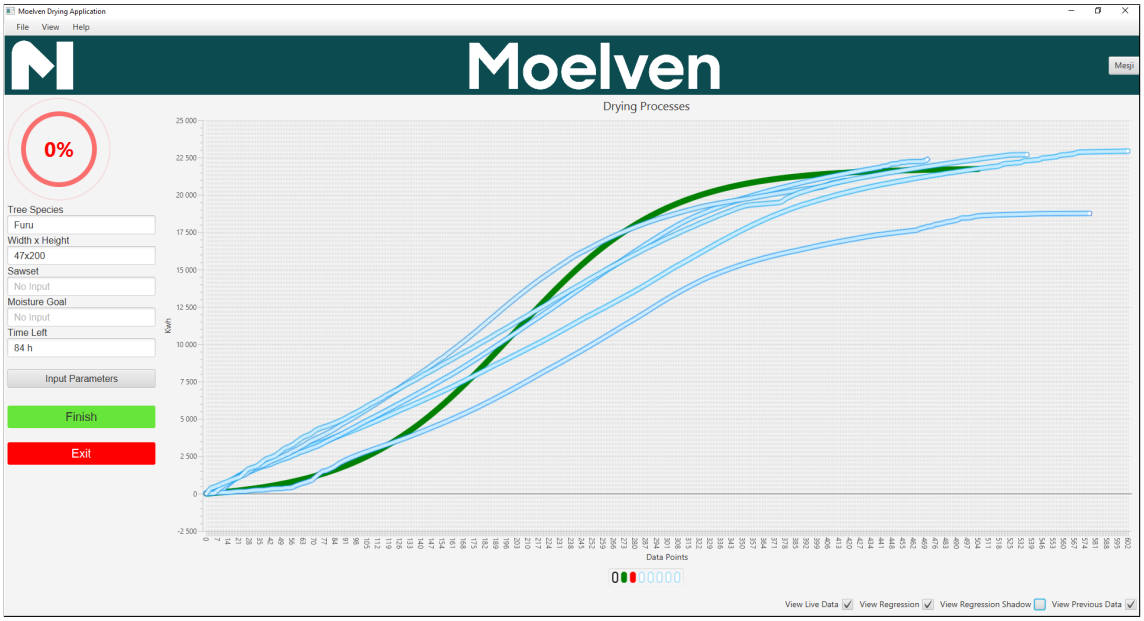


Figure 67: Screenshot of the application graphing the logistic growth model(green), and previous data(blue). The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Figure 67 is a screenshot of the dashboard in the application displaying previous data in blue alongside the logistic growth model in green. The model seems to fit the data well. As can be seen towards the end of the drying cycles, the lowest graph is considerably further away compared to the other three that are grouped together.

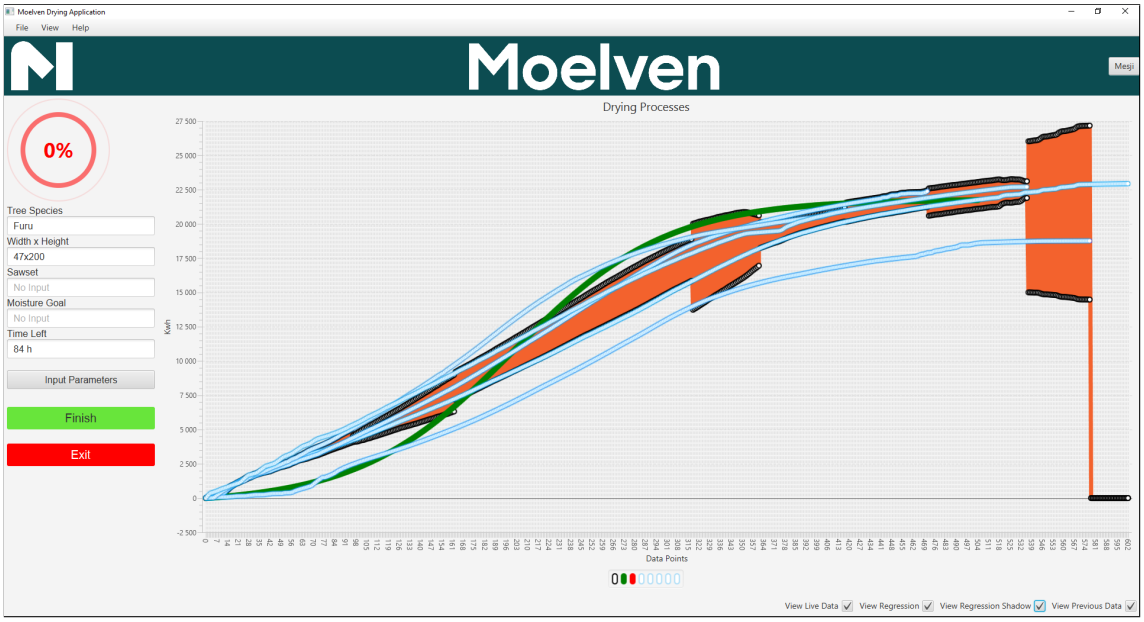


Figure 68: Screenshot of the application graphing the logistic growth model(green), previous data(blue) and a 80% confidence interval(black/orange). The x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

The confidence interval have mostly excluded the lower graph for the logistic regression as can be seen in figure 68, and therefor the logistic growth model fits better among the other three graphs that are grouped together around the same area.

7.2.7 Login System

A login system is implemented with the principle of least privileged, as described in further detail in section **??**, and for future development regarding notifications. The login system consists of a number of pre-made accounts. These accounts are split into normal accounts and Admin accounts. Admin accounts are authorized to add or delete accounts through the application, as well as view the database for a list of all stored accounts. Normal accounts are made for the drying operators. These accounts must be created by an Admin. All accounts contain information regarding a person, namely: First name, last name, phone number, username, password, and a boolean variable if the account is Admin or not.

With regards to security and the principle of least privileged, unauthorized personnel should not be able to start a drying cycle and thus are required to login in order to use the application. This login requirement allows for the opportunity to store information about the drying operator, such as a phone number. If a phone number is registered, then the application could send an SMS to the phone number as a notification when the drying cycle is close to or completely finished.

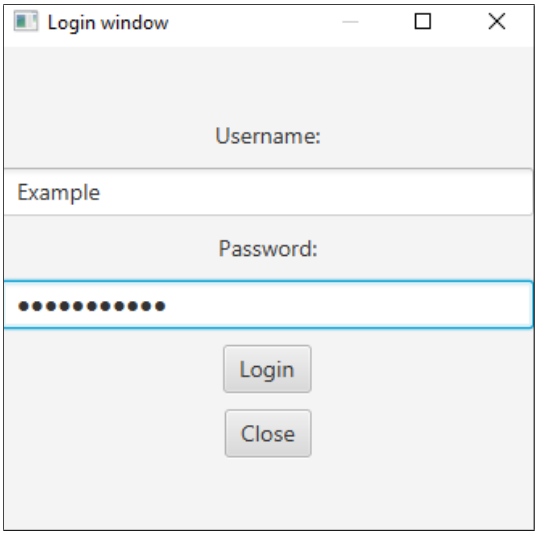


Figure 69: An example of the login popup window. Screenshot taken from the application.

The login popup window takes a username and a password as input parameters, and the user has the option to press login or close the window. The username is passed alongside the password to a SQL statement that checks if the user is stored in the database. The username is handled as a normal String because this information is not sensitive. However, the password is sensitive and secret information, as described in further detail in section 9.3.1.

```
/**
 * This function hash a password based on the SHA-512 algorithm
 *
 * @param password input parameter for password
 * @return a 64 character long string containing the hashed password
 */
private static String hashPassword(String password){
    try {
        MessageDigest messageDigest = MessageDigest.getInstance("SHA-512");

        assert messageDigest != null;
        messageDigest.update(password.getBytes(StandardCharsets.UTF_8));

        byte[] hashedPassword = messageDigest.digest();

        StringBuilder hashedPasswordString = new StringBuilder();

        for (byte b: hashedPassword) {
            hashedPasswordString.append(String.format("%02x",b));
        }

        return hashedPasswordString.toString();

    } catch (Exception e) {
        e.printStackTrace();
```

```
    }
    return null;
}
```

The password input is directly sent to the hashPassword() function that uses the SHA-512 algorithm in order to keep the confidential integrity of the information. The password is stored as the hashed value in the database as a 64-bit long string.

```
/**
 * This function retrieves data regarding the user given by the user input
 *
 * @param username input parameter given by the user
 * @param password input parameter given by the user
 * @return a TableResult consisting of information about the user
 * @throws Exception throws an error if something goes wrong
 */
public static TableResult logIn(String username, String password) throws Exception
    {

    // Sqlstatement
    final String sqlStatement = "SELECT Username, Admin, Phone_no, First_name,
    Last_name " +
            "FROM " + PROJECT_ID + "." + LOCATION_ID + "." + USERS_TABLE_NAME + " "
     +
            "WHERE Username = " + '"' + username + '"' + " " +
            "AND Password = " + '"' + password+ '"';

    System.out.println(sqlStatement);

    // Retrieves the results from the queryjob
    return HelpingFunctions.createQueryJob(sqlStatement);
}
```

The logIn() function retrieves data regarding the account that correlates to the username and password given by the user inputs. The logIn() function passes along the response from the database to the getUser() function:

```
/**
 * This function retrieves information from the login() function
 * and iterates through the response
 *
 * @param username input parameter given by the user
 * @param password input parameter given by the user
 * @throws Exception throws an error if something goes wrong
 */
public static void getAccountInformation(String username, String password) throws
    Exception {

    TableResult result = logIn(username,password);
```

```
12
13      if(result.getTotalRows() != 0) {
14          for (FieldValueList row : result.iterateAll()) {
15
16              if (row.get("Username").getValue().equals(username)) {
17                  CreateLogoBar.getLogin().setText(username);
18                  setUserName(username);
19
20                  if (!row.get("Phone_no").isNull()) {
21                      setPhoneNo(row.get("Phone_no").getStringValue());
22                  }
23                  if (!row.get("First_name").isNull()) {
24                      setFirstName(row.get("First_name").getStringValue());
25                  }
26                  if (!row.get("Last_name").isNull()) {
27                      setLastName(row.get("Last_name").getStringValue());
28                  }
29                  if (!row.get("Admin").isNull()) {
30                      setIsAdmin(row.get("Admin").getBooleanValue());
31                  }
32              }
33          }
34      } else {
35          NotificationPopUp.displayNotificationWindow("Wrong username or password!");
36          getPasswordTextField().clear();
37      }
38  }
```

The getUser() function receives a TableResult from the logIn() function in line 11. Line 13 checks if any information is received. If no data is found, then an error message is returned to the user informing that the username or password is wrong, or the user does not exist in the database. If data is found, then the TableResult is iterated through and sets the information in their corresponding variables.

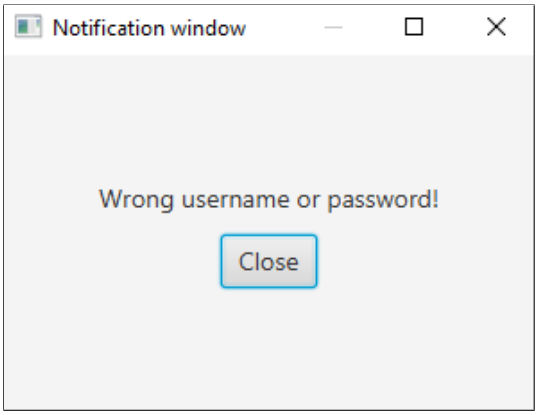


Figure 70: Notification given to the user if the username and password does not match any found in the database. Screenshot taken from the application.

If no match is found in the database, an error message will be returned to the user as shown in figure 70. If the login is successful, then the login button will be renamed to the username of the account logged in. One of the attributes returned from the database is, as mentioned, a boolean variable to determine if the account is Admin or not. These variables are stored and checked by the following code snippet:

```
// Login button
getLogin().setOnAction(event -> {
    if(getLogin().getText().equals("Login")){
        LoginPopup.login();
    } else {
        if(isIsAdmin()){
            LoginPopup.adminPopup();
        } else {
            LoginPopup.userPopup();
        }
    }
});
```

This code checks the status of the login button. If the button displays "Login", then the login popup window will be opened. If the returned boolean variable is true, then the admin window will open. Lastly, if none of these are true, then a normal user is logged in, and the normal account popup window is displayed.

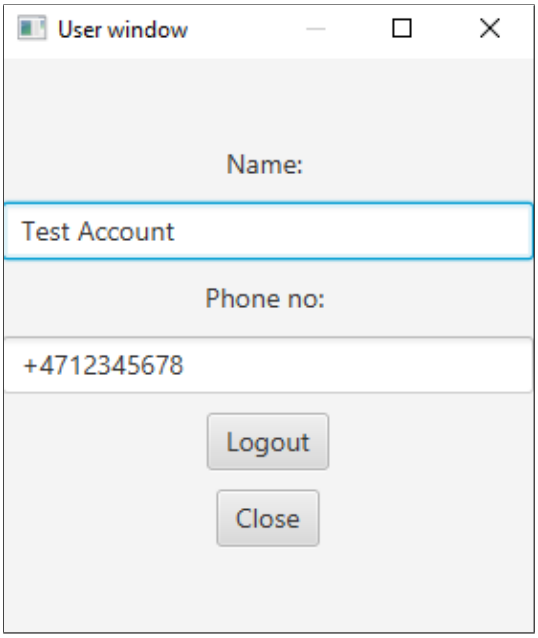


Figure 71: An example of a normal account logged in. Screenshot taken from the application.

If a normal non-admin account is logged in, then the login button is renamed to the username of the account. If the button is pressed after the user has logged in, a new popup window will appear as shown in figure 71 containing information regarding the account: First name, last name, and

telephone number, as well as the option to close the window or logout.

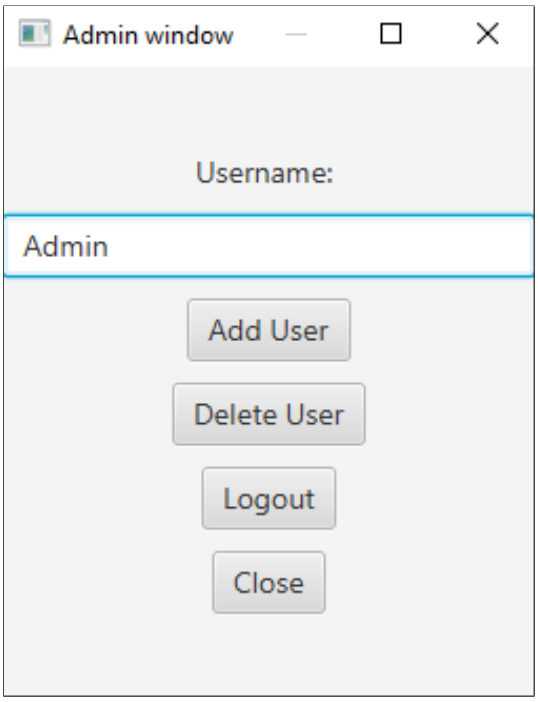
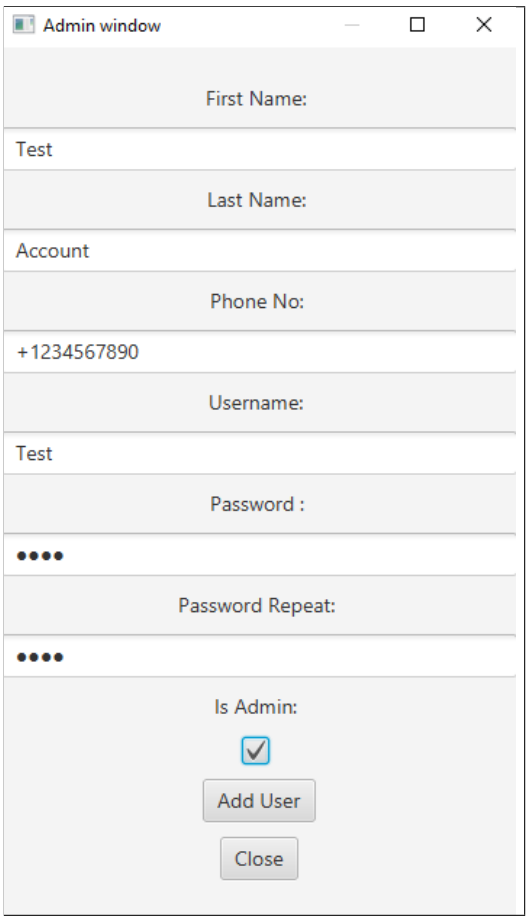
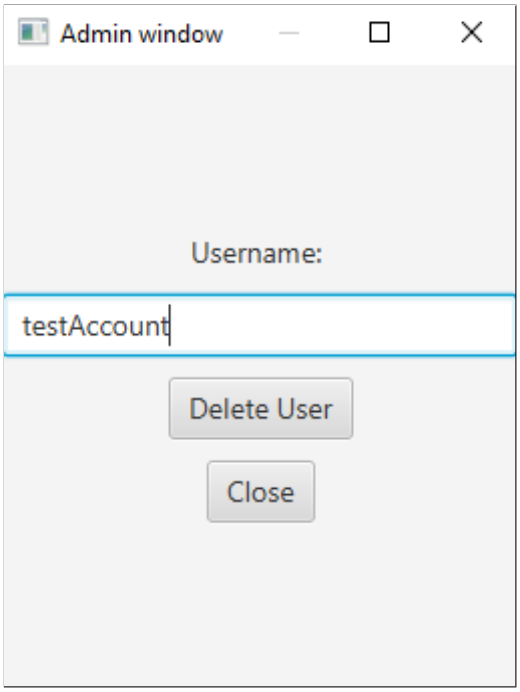


Figure 72: An example of a admin account logged in. Screenshot taken from the application.

If the account is admin, then a special popup window will open that only admin accounts have access to. This window contains a number of functionalities such as adding or deleting an account, and close the window or logout.



(a) Popup window for adding a new account



(b) Popup window for deleting an account

Figure 73: Popup windows displayed when pressing "Add User" and "Delete User" in figure 72. Both screenshots taken in the application.

When adding a new account, a new popup window will open with input parameters for all the required information, as can be seen in figure 73a.

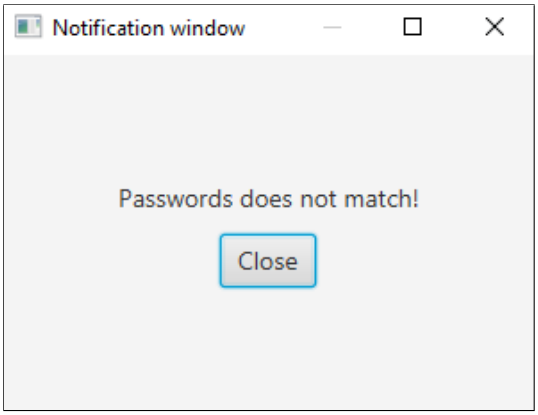


Figure 74: Notification given to the user if the two passwords does not match. Screenshot taken from the application

The password is a sensitive bit of information and is therefore blacked out when typing it into the application. This makes it challenging to know if the information given is correct or not. Since the user can't see the password, it has to be checked by inputting the same password twice. If the two passwords do not match, one of the passwords has been mistyped. Figure 74 shows the error message returned to the user informing that the passwords do not match.

All parameters are handled as normal strings except the password, which is hashed as previously discussed. The information is sent to a SQL statement and forwarded to the database through the addUser() function:

```
1 private static void addUser(String firstName, String lastName, String phoneNo,
      String username, String password, boolean isAdmin) throws Exception {
2         // Sqlstatement
3         final String sqlStatement =
4             "INSERT INTO " + PROJECT_ID + "." + LOCATION_ID + "." +
      USERS_TABLE_NAME + "(First_name, Last_name, Phone_no, Username, Password, Admin
      ) " +
5                     "VALUES("+'"'+firstName+'"'+","+'"'+lastName+'"'+","+'"'+
      phoneNo+'"'+","+'"'+username+'"'+","+'"'+password+'"'+","+ isAdmin+") ";
6
7     HelpingFunctions.createQueryJob(sqlStatement);
8 }
```

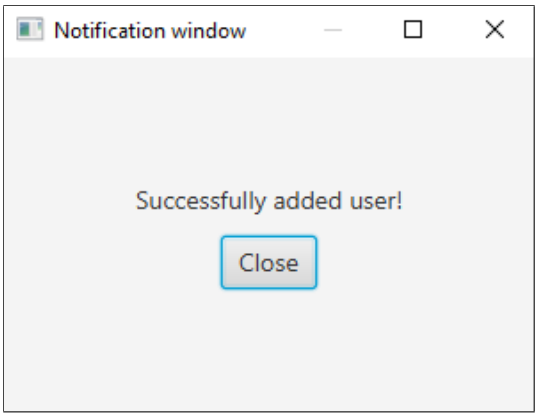
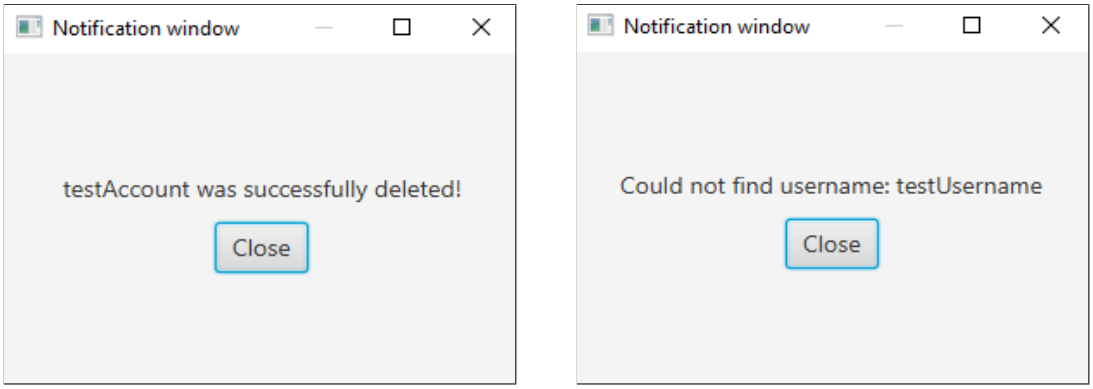


Figure 75: Notification given to the user if the user is added to the database. Screen shot taken from the application.

When deleting a user, a new popup window is displayed as shown in figure 73b. The username of the desired deleted account is inputted into the text field. When pressing the "Delete User" button, the application first checks if the username given by the user input exists in the database. By checking if the account exists before actually deleting it, the application can return a message to the user that the deletion was successful.



(a) Notification displayed to the user if the account is successfully deleted

(b) Notification displayed to the user if the account could not be found in the database

Figure 76: Notification windows displayed if a user is deleted or not found. Both screenshots taken from the application.

The disadvantage of this solution is that the application has no way of knowing if the deletion of SQL worked successfully. Another solution could be to add an additional check by running the getAccount() function again to see if the account was deleted successfully or not. However, this seemed redundant and excessively heavy in the number of SQL queries, and therefore it was decided against.

```
private static TableResult getAccount(String username) throws Exception {

    // Sqlstatement
    final String sqlStatement = "SELECT Username " +
            "FROM " + PROJECT_ID + "." + LOCATION_ID + "." + USERS_TABLE_NAME + " "
     +
            "WHERE Username = " + '"' + username + '"';

    // Retrieves the results from the queryjob
    return HelpingFunctions.createQueryJob(sqlStatement);
}
```

The getAccount() function checks the database if the username input exists in the database. The function is called upon in the deleteUser() function:

```
private static boolean deleteUser(String username) throws Exception {

    if(getAccount(username).getTotalRows() != 0){
        // Sqlstatement
        final String sqlStatement = "DELETE FROM "+ PROJECT_ID + "." + LOCATION_ID
    + "." + USERS_TABLE_NAME +" WHERE Username = "+'"'+username+'"';

        HelpingFunctions.createQueryJob(sqlStatement);
        return true;
    } else {
```

```
10          return false;
11      }
12  }
```

The getAccount() function returns a TableResult, and an if statement checks the number of rows contained in the response. If no rows are found, then that means that the username does not exist in the database. If the user is found, another SQL statement is sent that actually deletes the account.

7.3 Frontend

Frontend development, also known as client-side development, refers to development regarding what the users sees and can interact with, in contrast with backend development (*What Is a Front-End Developer?* 2022). With regards to the application, this involves styling the application with colors, sizes, placements among other factors that satisfy the design requirements discussed in chapter 6.

7.3.1 Why JavaFX?

Swing, AWT, JavaFX are all a part of Java JDK and are used to create GUI with JavaFX being one of the latest entries in this list (Singh 2018).

Abstract Window Toolkit, also known as AWT, is the most known out of the three and is also the oldest. AWT contain few features and includes no inbuilt libraries/packages and is a manual labor intensive library to use. Swing on the other hand is most of the time a lightweight, more sophisticated library with a number of inbuilt libraries/packages. Swing builds and runs on top of AWT itsels, and is a clear improvement over AWT. However, this would only last until JavaFX was added to the JDK.

JavaFX is a modern state of the art library to create GUI applications, as mentioned in section 5.1.5. The main benefit of JavaFX compared to Swing or AWT is the support for customization using Cascading Style Sheets, better known as CSS files. By using CSS files for styling, it is easy to change the color of a background or the text of a label. The main benefit of using JavaFX for the application is the flexibility and relatively ease of implementation compare to Swing or AWT.

7.3.2 Hierarchy Structure

The JavaFX application is divided hierarchically into three main components: Stage, Scene and nodes (*JavaFX Application Structure* 2021). The stage in a JavaFX application is similar to the Frame in a Swing Application, it contains all the JavaFX objects to be displayed by the scene.

The stage is put inside a scene, and in order to display the scene the method show() must be called upon the *PrimaryStage* object, additional stages could be added as well.

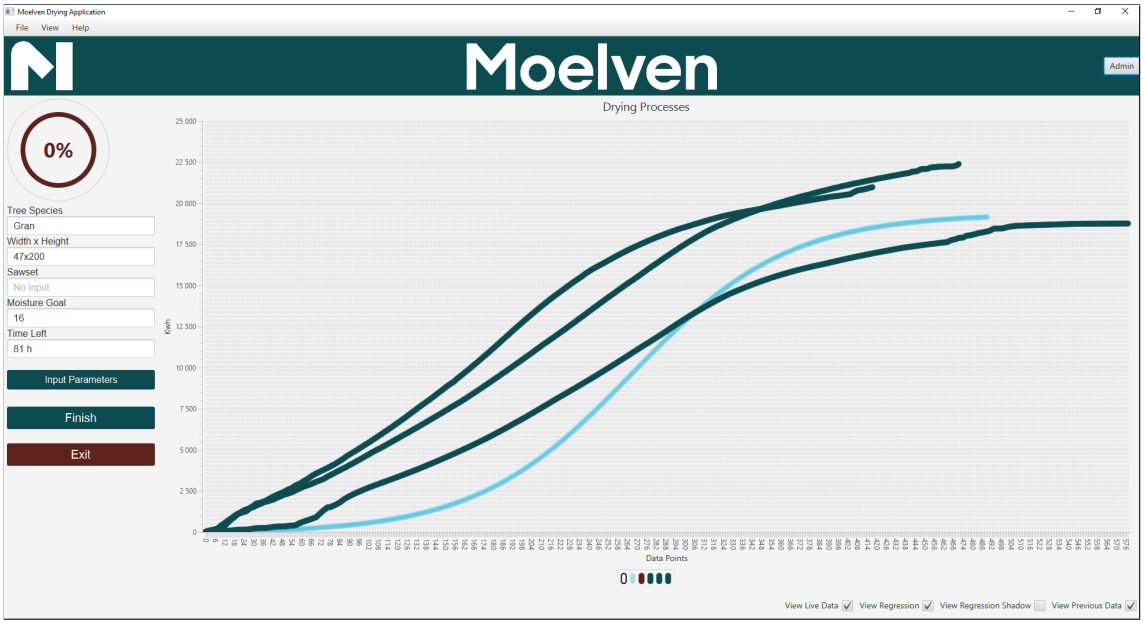
The *PrimaryStage* follows a hierarchy structure and allows for multiple components to be used, these components could be: GridPane, VBox, HBox, etc. A gridpane is especially useful for it's alignment features that automatically keeps the alignment when the window is scaled. A gridpane lays out its children within a flexible grid of rows and columns (*Class GridPane* 2015).

7.3.3 Scene Builder

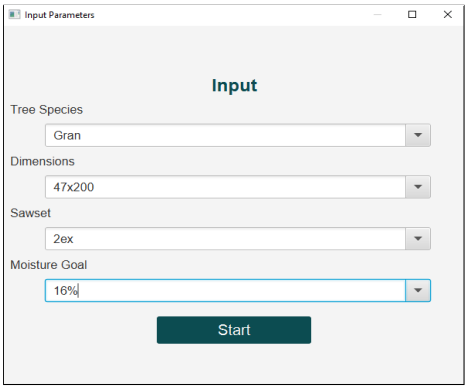
At the start of the project period, the decision of using JavaFX was made. The tool *Scene Builder* was later found to design the GUI. This program created an FXML file that the application could import and create the GUI from. This made it easy to design the GUI in a 2D program, compared to writing code.

However, some issues arose when trying to send data from the database to the graph in the FXML file. Given that sending data from the database to the GUI is a major part of our project, the JavaFX code had to be written manually instead of using the scene builder. Using the scene builder tool would have been beneficial and more efficient than writing the GUI manually. However, creating the GUI from scratch gave increased flexibility and full control over the GUI.

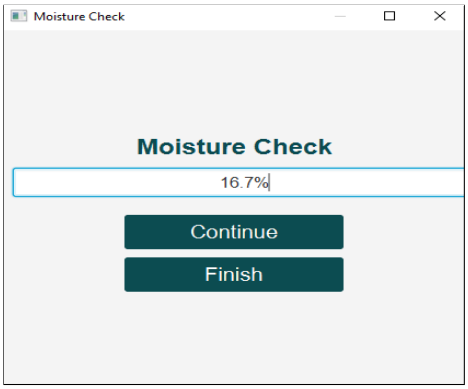
7.3.4 Dashboard



(a) The final design of the dashboard in the application, where the x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.



(b) The final design of the input parameters window



(c) The final design of the output parameter window

Figure 77: The final design of the application. All three screenshots taken from the application.

The dashboard was implemented based on the requirements discussed in section 2.2.1, Moelvens profile manual requirements for color-pallets as discussed in section 6.2.10, and on the final sketch in section 6.3. Figure 77 displays the final version of the application, all previous shown figures in the report were taken at an earlier version of the application and the styling of the dashboard was done at a later stage in the development process. The dashboard consists of a menubar, logobar, sidebar, bottombar and a main a graph window. The main component is the graph window that takes up approximately 80% of the screen.

The menubar contains three submenues: File, view and help. The file submenu currently only

contains an exit button to close the application, but additional features could easily be implemented in the future. The view submenu contains the same functionality as the check boxes at the bottombar, in order to toggle on and off different graphs. The help submenu contains a help section with basic information about the application, and a about us section that informs the user about Moelven and basic information regarding the application. The logobar contains Moelven's logos.

The sidebar contains a progression bar imported from a library written by *Andrea Vacondio* (Sanket 2019a) in order to track the progress of the drying cycle. The library is copyrighted with the APACHE-2.0 LICENCE, for more information, see appendix 11.1. In addition to the progress bar, the sidebar contains the input parameters given by the user, a count down timer that displays the remanding time of the drying cycle that is based on the regression, starting a search, the finish button that opens a window where manual moisture values can be inputted, and an exit button to close the application. As mentioned, the bottombar contains check boxes to toggle on and off which graphs to display.

The graph is the main aspect of the application, and is therefore correspondingly large compared to the other components. The graph is a *linechart* object that contains a number of series, where each series is a plotted graph. Each series contains data points along the x-axis and y-axis. The y-axis represents a value in the unit of kWh. The x-axis is the time axis divided up into index points. The original idea was to use hours as units for the timeline. However, an unexpected issue occurred that caused the graph to look *"choppy"*. The decision was made to reject hours as the unit of choice and choose index points instead. By choosing index points as the unit for the x-axis, the graph would look more smooth.

The linechart can display four different types of series. These are:

- Live data,
- Previous data,
- Logistic regression,
- Confidence interval.

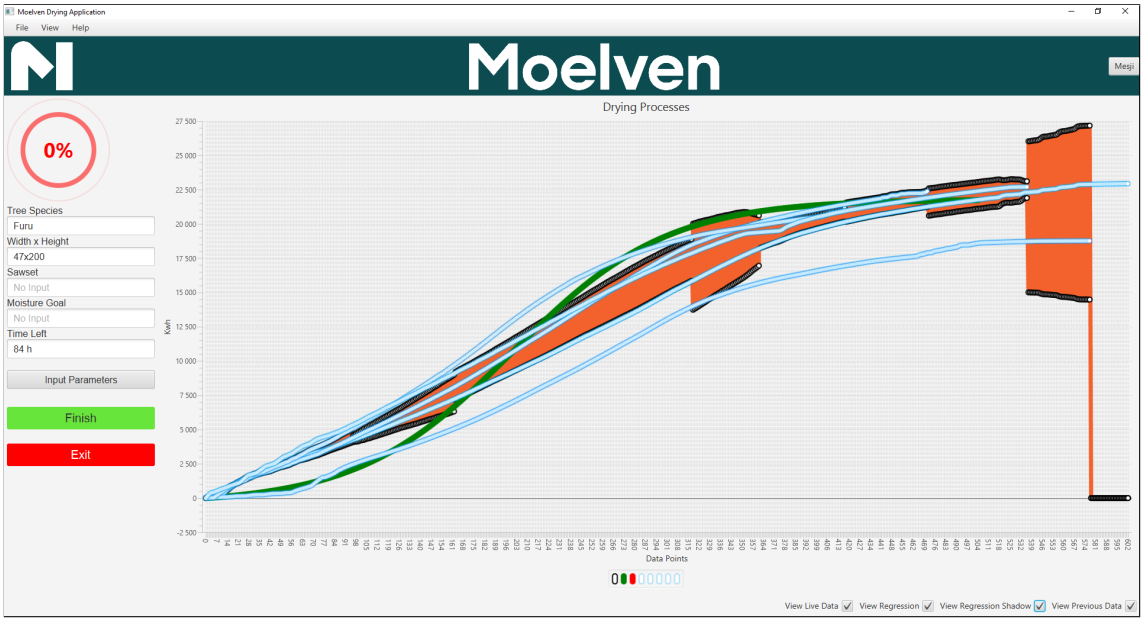


Figure 78: Screenshot of the application graphing the live data(red, however the drying cycle at 0%, therefore not visible), the logistic growth model(green), previous data(blue) and a 80% confidence interval(black/orange). Where the x-axis(horizontal) represents the time axis and have the unit: index points, where each index point represents 10 minutes. The y-axis(vertical) represents the kWh.

Live data are searched for when the application starts. The search will take place at a constant speed which can be changed as preferred, but at default this constant is set to every 60 seconds. The database are checked for new data from the time the drying cycle was started until the current time. If any new data is found, then the data are placed in a *Live data series* and plotted in the graph if the check box is checked. The live data are set to the color red.

Previous data gathers all the data from each drying cycle and creates a unique series for each cycle. This way separate graphs are plotted. These are set to the color blue.

Logistic regression is predicted at each index point along the x-axis, until the regression meets the last index point, then this data is put into a regression series and plotted in the graph if the check box is checked. The regression series are set to the color green.

Confidence interval puts the lower and upper limits along the index points of the x-axis, and puts the lower and upper values in a series. This series is then plotted if the check box is checked. The confidence interval is set to the color black/orange.

7.3.5 Styling

As mentioned in section 7.3.1, the main advantage of JavaFX is the support for CSS files in order to style the application. Here is an example of styling the start button:

```
startButton.setId("inputButtonStart");
```

```
#inputButtonStart {
    -fx-pref-width: 200;
    -fx-translate-y: 10;
    -fx-pref-height: 25;
    -fx-font-size: 20;
    -fx-font-family: Arial;
    -fx-background-color: rgba(12, 76, 81, 1);
    -fx-text-fill: white;
    -fx-alignment: bottom-center;
}
```

The use of CSS files for styling meant it was easy to change factors such as size, font, colors and alignment. The implementation of the styling follows the requirements and design choices already discussed in the design section, chapter 6. Another solution could be to hard code the styling variables in the code, however, this can quickly become hard to navigate.

7.4 Threads

Searching for previous data is a time consuming process that can take upwards of several minutes. Originally the application ran on just one thread, and when the search for data started, the GUI would freeze until the search was completed. This is because the thread can only handle one process at a time, either run the GUI or search for data in the backend. This was not user friendly and thus something had to be done in order to fix the issue.

The solution was to split up the workload into multiple threads that could run multiple processes simultaneously.

```
// Gather previous data
Thread thread = new Thread(() -> {
    try {
        // Retrieves the data from the database from setInputParameters()
        Map<Integer, Map<String, Number>> data = setInputParameters();
        // Graphs the data in the GUI after this thread is finished running
        Platform.runLater(() -> {
            try {
                loadSingleSeries(data);
                setLoadedData(true);
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
```

```
18 }
19 );
20 thread.setDaemon(false);
21 thread.start();
```

This code snippet creates a new thread and tries to load the data simultaneously alongside the GUI thread running. The thread is set to non-daemon because, the Java Virtual Machine exits when the only threads running are all daemon threads (*setDaemon* 2020). If the thread is not a deamon, it can be stopped without stopping the GUI thread, because the GUI thread is set to a daemon thread. If a search needs to be restarted, then the previous thread can be stopped and a new take it's place without stopping the GUI thread.

Threads have a problem when it comes to synchronization. It is not possible to return a value from one thread over to another thread, therefore issues appeared when trying to graph the data in the GUI thread that was search for by another thread. The application threw an error stating that the thread was not on the same thread as the GUI thread. Through many hours of googling and several meetings with professors, the solution was to stop the thread after the search for data was finished and to graph the value once the data was returned over to the GUI thread. This was done using the Platform.runlater() function which would run after the thread was finished running (*Class Platform* 2015).

By implementing threads into the application, the workload was divided into multiple processes which increased performance, added the feature to research for data and made the application more user friendly.

8 Testing

The goal of creating and running tests is to find errors and bugs early on during the development process, as the cost of bugs and errors become exponentially larger the further into the development phase a project gets (Sanket 2019b).

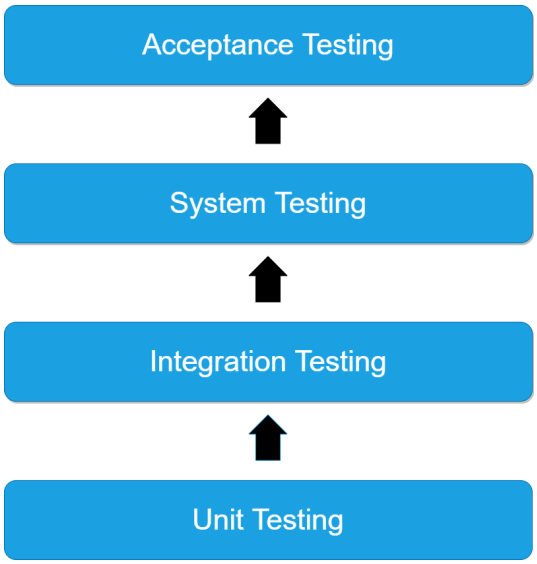


Figure 79: Visualization of the testing hierarchy that the testing model follows (Created in Draw.io)

Testing follows a structure visualized by figure 79 where unit tests lays the foundation, followed closely by integration testing, then system testing, and lastly acceptance/user testing as mentioned in section 2.2.6.

8.1 Separation Of Concerns

The application has been implemented with the design principle Separation of concerns in mind. Therefore, the test classes are separated from the main source code, so the test classes are developed, executed and maintained independently from the production code. For the test implementations we have used the package structure src/test/... directory for test classes, instead of src/main/... directory. Thus, improving the maintainability and readability of the test code. In addition, a separate test database have been implemented in order to separate test data from real data (Bansal 2021).

8.2 Unit Testing

Unit testing is a type of software testing where individual components of an application are tested (Hamilton 2022b). By creating Unit tests, we can understand where the errors are located piece

of the code in a precise manner. Without this type of testing, finding errors at an application level would result in extra time spent in finding the location of errors. Unit testing is an agile method, which is updated as new features are added. By running Unit testing on our code, we are able to determine if the new code has broken an old feature. The reason for choosing Unit Testing is because it segregate the code and fix errors, which results in developing a code that is more reliable and bug-free.

```
@Test
void addUser() throws Exception {
    AccountHandler.deleteUser("testAccount");
    AccountHandler.addUser("test","test","+4712345678","testAccount","test",false);
    Assertions.assertEquals(1, AccountHandler.getAccount("testAccount").
    getTotalRows());
}
```

```
@Test
void deleteUser() throws Exception {
    AccountHandler.addUser("test","test","+4712345678","deleteTestAccount","test",
    false);
    AccountHandler.deleteUser("deleteTestAccount");
    Assertions.assertEquals(0, AccountHandler.getAccount("deleteTestAccount").
    getTotalRows());
}
```

Above is an example of how the functions for adding and deleting a user is tested. Firstly, if there are any data from a previous test, this is handled before doing anything in the test. After previous data is handled then the test can proceed with the intention of the test. Assertions.assertEquals() is called in order to check the response. When adding a user, it is expected to retrieve one row regarding the username added. When deleting a user, the opposite is expected where zero total rows are expected.

8.3 Integration Testing

Further integration testing has been performed on the application. This is the second level of the software testing and it is the process that happens after Unit testing. In this level of testing, individual or unit components of the software are tested together. The purpose of doing integration testing on the application is to expose defects at the time of interaction between integrated units or components (Awati 2022). In contrast to Unit testing where modules are used for testing purposes, integration testing combines these modules and test these, since the application is developed with a number of application modules that are coded by the different individuals in the group (*Integration testing* 2022) . It is therefore important to do integration testing to look for correctness of communications between the modules (Hamilton 2022a).

```
class DBTest {
```

```
2
3       DBTest(){
4           Constants.TREE_SPECIES = "Gran";
5           Constants.DIMENSIONS = "47x200";
6           Constants.SAWSET = "2ex";
7           Constants.MOISTURE_GOAL = "16";
8           Constants.CURRENT_DATE = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss")
    .format(LocalDateTime.now());
9           Constants.START_TIME = Constants.CURRENT_DATE;
10          Constants.STOP_TIME = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss").
    format(LocalDateTime.now());
11      }
12
13      @Test
14      void setInputParameters() throws Exception {
15          Assertions.assertEquals(2, DB.setInputParameters().size());
16      }
17
18      @Test
19      void getCurrentDrying() throws Exception {
20          Assertions.assertEquals(0, DB.getCurrentDrying().size());
21      }
22  }
```

Above is an example of how integration testing is done. The first test retrieves data regarding the parameters given in the test.

8.4 System Testing

System testing is performed on a completely integrated system and is the third level of the software testing hierarchy. During system testing, the tests are carried out on the system as a whole in order to test how the integrated system respond. System testing is often performed by a separate independent testing team, rather than the development team, in order to test the quality of the software (pp_pankaj 2019). More details regarding system testing in section 10.1.5.

8.5 Acceptance Testing

Acceptance testing covers the fourth and last step in the testing hierarchy model. Acceptance testing evaluates whether the application complies with the specification requirements and also evaluate if the application is ready for deployment (*Acceptance testing* 2022). More details regarding acceptance testing is found in section 10.1.5.

9 Security

This section describes the security aspect of the application based on the security requirements written in section 10.1.4. In order to secure the data, the focus was set on using the key security concepts of Confidentiality, Integrity and Availability related to technology, implementation and design. The following sections are arranged in descending order, from security risks with the potential for causing the most amount of damage, to security risks with less impact for the solution.

9.1 Access To JSON Key-File

The JSON key-file used to grant read and write access to Moelven is located in the source code of the proposed application. The key-file makes it possible to modify the database by creating or deleting tables, and inserting, deleting or changing table-data. This can potentially be very destructive for the application, considering that it is based on the historical data stored in the database. If this previous data is altered, it would affect the estimates and visualisations. As mentioned in 10.1.4, the data shared by Moelven for the application has not been classified as sensitive. The BigQuery database used is also a copy of the original database. Therefore, a compromise in the integrity by intruders accessing key-file, would not affect Moelven as much as the proposed application. The potential problem for Moelven would be losing the inputted manual moisture that our application inserts into the BigQuery database. The JSON key-file is only located internally on the group members computers, and not located in the repository. The reason for this was to prevent us having to upload the key-file to GitLab and BitBucket.

The group experienced an episode where the key-file was located in the repository, and the repository was uploaded to GitLab publicly. The collective decision was therefore to delete the old key file, and get a new one that we had internally. More information regarding this in section 7.2.1.

9.2 SQL Injections

SQL injection was considered by the group as one of the biggest security risks for this application, and could compromise the integrity of the database. The modifications an intruder can do to the database used in this work with injections can be completely destructible for the proposed application. Therefore, it is crucial to prevent all kinds of injections. The most common SQL injection is the *UNION SELECT* injection, which lets the user execute additional *SELECT* queries to collect additional data and append it to the original query. This makes it possible to collect data that is not intended to be collected by the original SQL-statement.

```
1 SELECT a, b FROM table1 UNION SELECT c, d FROM table2
```

As mentioned, the database used in this work is just a copy of the original at Moelven. This means

that the security risks concerning the database is more of a treat for our application than for the database system at Moelven. The measurements taken to avoid SQL injections is mentioned in section 7.2.4

9.3 Unauthorized Access

The internal databases, systems and network of Moelven is critically important, therefore it is essential that internal sensitive information is only granted to those entities that are authenticated. Authentication is a security concept which assures the claim of an entity (E.g a person) by verifying identity information (Paul 2013) p.11. The application has a Single sign-on(SSO) solution for authentication (*How Does Single Sign-On Work?* 2022). The user uses one set of login credentials, which includes a username and password management for both user and administrators. This authentication is mostly relevant for this application as the SSO-solution enables users to access the application more rapidly since they do not have to keep track of different sets of credentials. This is a beneficial authentication for the application considering that it should be used effectively at a sawmill. Further, the application has an admin account which grants the application the ability to control additional features such as creating and deleting accounts in the application. An admin makes it possible to prevent measures as well as provide quick solutions to breaches. By having an admin, the data visibility of the application is taken care of (Hurley 2020). This increases the security of the application. Another benefit of having an admin is that administrators can quickly relinquish login privileges across the application when a user leaves the organization. This safeguards the personnel safety of the application by reducing the risk of insider trading, since a former employer is not able to access the application in order to perform actions that may cause harm or losses to Moelven (*Grunnprinsipper for personellsikkerhet* 2020).

9.3.1 Password Hashing

Hashing has been used to protect the integrity of the application by securing the passwords of the users of the application (Stickney 2021). Hashing changes the plain text of the passwords to a hash value based on the hashing function (Hoffman 2022) (Stickney 2022). SHA-512 algorithm has been used for hashing of the passwords in the application. SHA-512 is implemented in the application over other algorithms such as Message Digest (MD5) or SHA-256. MD5 was not selected since SHA-512 provides a more adequate and cryptographically secure functionality(Santos 2015) (Khaishagi 2019). SHA-256 was considered as an algorithm for hashing the passwords. Although, since SHA-256 uses 32-byte words, it makes it less resistant to attacks compared to SHA-512 which uses 64-byte words (*SHA-2* 2022). In addition, SHA-512 is also on the list of approved Hash functions by NIST based on the Federal Information Processing Standard: FIPS 180-4, Secure Hash Standard (*Hash Functions* 2022). Another reason for SHA-512 was its speed. For data lookup and files organization, a fast hashing algorithm is desirable, since it allows to search and

find data quickly.

9.3.2 Salting

Hashing can be a good security measure, yet it is not optimal to use hashing alone, with regards to a security perspective. It is therefore important to add another safeguard to the stored passwords. Salting adds random data to the hashes, which makes it more difficult for an attacker to crack a password's hash. If password hashes are not salted, an attacker can perform a dictionary attack (Arias 2021). In a dictionary attack, an attacker can easily compare the hashes of a stolen passwords table with every hash on the list, and if a match is found the password then can be deduced (ibid.).

9.4 Usage Of Libraries

The usage of both internal and external libraries was essential for creating the application. The data processing and calculations achieved would not be possible without the use of libraries. As a general rule in regards to security measurements, as few external libraries ias possible should be used and all unused libraries should be deleted. Importing libraries can be a threat, which made the group strive for usage of trustworthy libraries created by big corporations. The wrong use of libraries could lead to vulnerability regarding the integrity of the application and database.

9.4.1 Internal Libraries

The group considered the internal Java libraries to be safe, and no major threat for the integrity of our system. General security measurements were still applied, by deleting the unused libraries, but did not research each library for the possible threats. The JavaFX library was imported and used to build the GUI, more about the usage of JavaFX in section 5.1.5.

9.4.2 External Libraries

In the application, a few external libraries are used, which were essential for the final solution. Since these libraries were imported to handle the GCP database connection, calculate the estimates using logistic regression and cleaning the data sets using confident interval. By using these external libraries, it was important to think about the risk it could potentially cause the application. Caution when importing these libraries was used, and attention to not using libraries from unknown or unreliable sources.

- **Joda-time** (org.joda.time) - formatting time variable, from millis to Timestamp format.
- **Common-math3** (org.apache.commons.math3) - calculation of linear regression and confidence interval.

- **Junit** - library for unit testing in Java

The Google cloud libraries created the connection between Java and GCP, which made it possible to run SQL-statements, collect data and insert data to the database:

- **Google Cloud** (com.google.cloud) - Getting access to the GCP
- **BigQuery library** (com.google.cloud.bigquery) - connection to BigQuery and its functions

9.5 Copy Of The Database

The database which the application extracts data from is a copy of the database of Moelven. This means that they have a backup of the data they have given us. In this way Moelven is protected against data loss both intentional and unintentional, protected from virus infection and theft of information from the hackers of our application and has ease of management (*What is Backup? Benefits and Evolutions.* 2022). Ease of management creates consistencies in the processes for backing up data and information when restoring lost data, which can be stressful and time-sensitive (Becker 2022). Further, a backup and recovery solution is beneficial to having a strong security when is comes to protecting and saving data.

9.6 Awareness Of Security Weaknesses For Future Development

Several security concepts have been implemented in the design and technology of the application to secure especially Confidentiality, Integrity and Availability. There are still some weaknesses in the application that future developers need to be aware of. The application does not have multi-factor authentication, which is an authentication method where users are required to provide two or more verification factors before granting access to the application (Radtke 2022). The application has fewer potential entry points for hackers since it only has SSO, therefore by implementing MFA, the likelihood of cyber-attacks will decrease (ibid.) (*What is Multi-Factor Authentication (MFA) and How Does it Work?* 2022). Since our application is only to be used as a tool at the sawmills, efficiency has been a higher priority than top security at the moment. We have implemented the most necessary security the application will need to begin with, considering that not all workers will have their mobile phone with them around the sawmill. Therefore a two factor or multi-factor authentication with the use of mobile verification is not necessary yet. In order to improve security in the future, this is something that can be easily implemented in the code. Furthermore, the hashing of the passwords to the users does not have the safeguard of salting. This was something that was tried to be implemented in the code, but unfortunately the result of using salting resulted in different result every time. In retrospect, it could have been possible to use a hash algorithm which included salting such as Bcrypt and Scrypt (Gibbs 2016), since salting is preferred to secure

user password. These were chosen against since they do not support Java, only some Java libraries support them like Spring Security (Millington 2022). As mentioned above, a fast Hash algorithm was prioritized over a more secure Hash algorithm since the application is used to search and find data quickly, but for future development this can be implemented to make it more secure.

In the application, safety aspects have been prioritized as they were assessed as most necessary to secure Moelven, considering the time aspect for developing the application. Thus, the weaknesses mentioned above should be addressed in future development of the application.

10 Discussion

The purpose of this research was to develop an application that can predict and visualize the drying process with the help of energy measurements and other available production data, with the final goal of reducing drying time and energy consumption. The original idea was to make estimations to create an algorithmic formula that would take input parameters and calculate the drying duration. However, this solution was not reasonable due to many unpredictable factors, which would contribute to uncertainties in the estimate. Thus, another solution to the problem was to use a combination of regression analysis and confidence interval on the stored production data to create a more reliable estimate. This became the final solution to the project.

10.1 Result

The result of the development process and research conducted in this project is a functional application that processes data, visualize data, and predicts the drying process to reduce energy consumption. A combination of utilized energy data and the operator's experience will be able to predict a drying cycle easier. The group is pleased with the product and the accomplishment of the functional requirements set in the requirements section. 2

10.1.1 Data Processing

The back-end structure of the application consists of several modules that process the data to create as relevant and precise estimations as possible, as discussed in the section. 6.1.1. The data processing implemented resulted in visualizations that can guide the operators. We found that combining statistic formulas with data processing was perfect for the processing modules of our application (7.2.6). A confidence interval (4.5.1) was created to clean the data, and a logistic regression (4.5.5) was used to develop the estimates based on this cleaned data. The group discovered that logistic regression was the most suitable considering the S-curve of the drying cycles. The level of the confidence intervals was decided by testing a lot of different intervals to find the most suitable, like shown in figure (66).

Our data processing resulted in:

- Categorize and collect data from GCP
- Define each drying cycle, 7.2.2
- Creating estimates by linear and logistic regression
- Visualizing previous cycles, logistic growth model and confidence interval

10.1.2 Visualizations

The result of the data processing is visualizations that are displayed through a coordinate system.(6.2.4) in the dashboard of the application. These visualizations in the dashboard (7.3.4 are an essential aspect of the GUI, considering the approach of visualizing the estimates to show the application's predictions of the drying cycle.

10.1.3 Inspiration

The final application will inspire Moelven to continue the work of analyzing and processing the energy data they store. The energy data has never been used for analytic purposes before. We hope that the result of our development and research will inspire and intrigue Moelven to use similar methods to predict drying cycles.

10.1.4 Security

Security is a significant field concerning computer engineering. Through the security principles, as discussed in section , we implemented what is considered to be a safe and reliable application. A secure login system has been implemented using a username and password for authentication. See section 7.2.7. Basic SQL injections prevention such as: Checking the number of characters in the input and checking if keyword "UNION" are contained in the input parameters in section 9.2. Confidentiality is maintained by hashing the password and storing the hashed password in the database instead of the raw string directly 9.3.1. The change of key file ensured the integrity of the database 7.2.1. And lastly, libraries were chosen carefully to use as few libraries as possible and delete old libraries not to encounter any unnecessary security risk, as discussed in section 9.4.

10.1.5 Testing

Regarding testing, our application has implemented broad unit and integration tests to assure quality code and functionality. We are satisfied with the tests written that test for both individual functions, as discussed in the unit section 8.2, and modules in integration testing, as mentioned in section 8.3. When it comes to launching the application in Azure, Moelven has implemented an internal pipeline of tests that represent our system testing.

10.1.6 Sustainable Solution

An overall goal for Moelven is for both the company and its products to be climate-positive. They will continue to build a sustainable future in wood by safeguarding natural resources, as well as focusing on people, local values, and climate-smart products and services (*Framtiden bygges i*

tre 2021). This is their contribution which helps achieve the UN's sustainability goals together (*Moelvens bærekraftsmål* 2022). As engineers, it is fundamental to develop products for the future. Then it is helpful to think sustainable. Thus, we felt it was important that the product we developed for Moelven should meet this goal, as it is relevant and essential to us as engineers to create sustainable solutions. To achieve this goal, Moelven has chosen to focus on five of the UN's climate goals where they have the greatest authority(ibid.). Based on these climate goals, we have created a product that is found relevant to two of the five UN climate goals they have chosen to focus on. The first is climate goal 13: Climate action (*Take Action for the Sustainable Development Goals* 2022). Our solution contributes to this climate goal as it can help reduce energy consumption for Moelven. By providing a clearer visualization of when the lumber has finished drying. They can use less energy since they do not have to do as many manual checks as they have had in the past, which makes them release a lot of energy, as a different cause that they have to use more power to make up for the energy that is released. In addition, they have also previously ended up over-drying their products, which means that they have spent more energy than necessary on the drying process. Since they have not been quite sure if the lumber has finished drying, our solution can contribute to energy savings in these areas since it will provide a more accurate estimate of when lumber has been dried. The other goal is 15: Life on land (ibid.). To this goal, our application can contribute with resource optimization since the application strives to reduce over-drying, which increase the product quality.

A sustainable decision made considering the running of the application was to decrease the frequency between each search for new data, explained in section 7.3.4. To plot an ongoing cycle, the application needed to check regularly for new sensor data inserted into BigQuery. Considering each SQL statement consumes energy, it was desirable to decrease the frequency between each search to decrease the overall consumption. On the other hand, a modern application with low latency was also desired. The group settled on a frequency of one minute between each search, which was considered a balance between sustainability and latency.

10.2 Limitations

10.2.1 Data

During the development process, insights into the data sets, sensors, and database structure at Moelven were received. The group learned that Moelven had encountered digitization of their sawmills, which includes storing energy data and integrating their old data and live sensor data with GCP. During this digitization, they have encountered downtimes. This included downtime they were aware of and downtime they were unaware of, which was noticed while processing their data. The periods of rest affected the energy data in their database, and for our solution, it meant fewer data to make statistic calculations and quality estimations. Because of the downtime and

missing data, the result of our final products is a solution for the future. The implementations were done with the end in mind, where the calculated estimations become more precise as more data is uploaded into their database. This new data is included in the calculations and visualizations as it is uploaded. Our solution works in theory but has not been thoroughly tested in practice because of the missing data.

During the development process, Moelven introduced a new data set from a new sawmill called Notnäs. The idea was to implement the data-set to our solution to increase the data amount. However, the database structure and tables were too different from the other two sawmills. Notnäs had missing parameters, which would not work with our solution. Our solution categorizes based on essential parameters to create suitable estimates, but Notnäs did not have these parameters, which would have made inaccurate estimates because of irrelevant data. The group's decision was therefore not to include Notnäs.

The way the data was structured and the table connections in the database made it difficult to see the solution straight away. The group understood fast that some alternative solutions were needed for connecting the correct energy data to the proper cycle. As mentioned in 7.2.3, time intervals were created that defined what energy data belonged to which cycle. This is not the optimal way of doing this, but it was the only way to collect data from a specific cycle.

10.2.2 Logistic Regression

The group concluded by using the logistic regression model (4.5.5) as the growth model because it was the regression that most accurately fitted the shapes of the cycles. The logistic growth model was the more accurate option available, but it is not entirely optimal for the estimations. The drying cycles vary greatly (can be seen in figure 61), which also varies the accuracy of the logistic model. A better solution would have been a cubic model. Unfortunately, this was challenging to implement due to the reasons discussed and explained in section 4.5.4.

10.2.3 Security

The security requirements established (see section 10.1.4) were based on prior knowledge, experience and our desire to develop a modern and future oriented application. Despite the security measurements implemented, there are still improvements to be made regarding the security of the application, as mentioned in section 9.6. Improvements such as prepared statements and checking inside the function instead of the user input in the GUI would make our application safer. Multi factor authentication would have made the authentication process safer, though also slower. Since the database content is not considered sensitive, the group did not want to sacrifice the applications feel over its safety find it unnecessary. However, from a security point of view, it would be beneficial with multi factor authentication. Salting should be used in addition to hashing when handling the

password. This was attempted, however, not successful and therefor a potential improvement for the future.

10.2.4 Testing

Although extensive testing was implemented, acceptance testing was not done to the fullest extent. Results from the survey (See (*Survey* 2022)) and continuous feedback from Moelven serves as our acceptance testing. However, the application did not get tested as a final product. Therefore, potential errors or weaknesses in the system might not be detected.

10.3 Evaluation Of Group Work

10.3.1 Workflow

The group spent some time at the beginning of the project to find a suitable workflow and separation of work. In the beginning, everyone started working on the development of the application. We soon found out that it was more beneficial than we were divided into different parts of the issue. As a result, the group work was divided into three main fields.

- Coding/development of application
- Theoretical work concerning statistics and drying process
- GCP, database connections and understanding data

The dividing of work became important, reasoned that a lot of theoretical knowledge had to be acquired to develop a suitable application. The separation of work made it possible to pass knowledge between group members and then work together to find solutions and come up with ideas to use in the application based on the diverse knowledge. The understanding of the drying process and data had to be linked to the statistics, and the statistics needed to be implemented in the application. It was essential to understand which parts of the statistics we could use to find the best estimate based on the data in the database. Further, a person also had to be more familiar with the different attributes stored in the databases from the various sawmills which we extract data from. In separating workflow, we saved time on what it would take everyone to get acquainted with the different parts.

10.3.2 Sickness

Due to sickness, the project's workflow had to be changed from its initial iterations. At some points, the group was reduced to only two members, which divided work tasks quickly became

such that one worked with the code and the other wrote the report. In this way, we maintained a flow in developing the application and the information. This led to some tough weeks for the remaining members since the work was more on the two group members. The reduction led to postponed several of the deadlines we had set early in the project development, such as completing the report by the 6th of May. Due to a reduction in group members, the issue of postponing the bachelor thesis was discussed at some point. Still, an agreement between the remaining group members and the supervisor was accomplished. Reduction in the group may have affected the workflow and meant that we could not do things as planned in the beginning. Another possible solution could have been to develop and complete the application and then write the report. We felt that the best solution was to have one person on each part. In this way, it was possible to write about the development process of the application at the same time as we developed it.

10.3.3 Communication

The excellent communication with Moelven through meetings has been key for this thesis. Regular contact between the group and the client was instructive and created good cooperation where they regularly expressed their wishes, requirements, and concerns. Moelven seemed interested and engaged, which inspired the group. Aside from communication with the client, contact with the supervisor and lecturers/professors at NTNU were also satisfactory. When facing critical challenges, the group sought help from qualified individuals. More about the meeting that made these excellent connections possible in section 3.2 or the meeting log F.

10.3.4 The Group

The success of this thesis was mainly caused by excellent communication, effort, and collaboration inside the group. Frequent meetings kept a steady workflow throughout the thesis. All group members could adapt to changes in requirements based on wishes from Moelven and other unforeseen events. All group members appeared intuitive, academic, and creative, which was necessary considering our engineering capabilities were required to arrive at our solution.

10.3.5 Good Development Process

When looking back at the development period, the final result is found to be satisfying due to a suitable development methodology and a flexible development team. A functional application has been implemented that answers the task description. The group is satisfied with the development process and implementation of the project's user-centered design. Implementation of the GUI was aided through multiple design iterations and the survey. The application's design started with complex ideas and advanced features, but this would not fit our target group. Thus simplicity and ease of use were prioritized instead, as discussed in section 6.2

10.3.6 Theoretical Knowledge

A primary task during the thesis was to gain the theoretical knowledge necessary to engineer an intuitive solution. Therefore, the theory part became an essential part of the thesis and report. This theory was gained through meeting with Moelven, books provided by Moelven, "focus på the website" recommended by Moelven, research via the internet, and meetings with Professor *Ivar Farup*. The group feels that the result achieved was satisfactory, and a great deal of theoretical and organizational knowledge was acquired during the process.

10.4 Opportunities & Choices

10.4.1 Programming Language

The choice of using Java as our primary programming language was a successful choice that played on the developers' strengths. Combined with the use of JavaFX, included support for CSS files for styling leads to flexibility with regards to quickly making changes to the application's aesthetics. However, a better solution could be to use Python, given the broader support for mathematical libraries such as non-linear regression libraries. We tried to cross-feed data from our java code over to a python script, but it was challenging to communicate data between the Python Script and Java. Thus we ended up coding the mathematical formula ourselves.

10.4.2 Writing In English

The bachelor thesis is written in English, which was the group's collective decision. English was chosen because Moelven is a Scandinavian supplier. The code is written in English and based on the learning outcome of writing in English. By writing in English, the bachelor is more workable for Moelven since it can be understood by their businesses outside Norway, which is essential considering the results we have achieved in our data analysis of the drying process. These results can be used to create value and inspire to help bring light to a field that has been worked on little in the past. The results can highlight the benefits of using stored data to create applications on the sawmills, which can optimize and streamline their processes. Further, all of the code for the application was written in English, which made us choose to coordinate the language in both places. Finally, we thought that writing in English was an opportunity to improve our skills in English and thus get an even more significant learning benefit during the period. Afterward, we have seen that writing in English has also had some disadvantages. This is because a lot of time has been spent working with the report's spelling to get the best possible sentence structure to convey the report academically.

10.4.3 Rejected Functionality

During the development process, there was a functionality that was rejected. An example was a functionality aimed to connect the username to the manual moisture check inserted into the database to be able to see the operator at the station and responsible for the cycle. The group decided to reject this functionality because of confidentiality concerns. Connection moisture check and the user was no requirement from Moelven, and we did not want to risk the confidentiality of the operators since the connection was not necessary.

11 Conclusion

Drying wood is a process that nowadays still uses antiquated and inefficient methods. The Norwegian company Moelven wanted to optimize the energy consumption of drying wood. Given the problem regarding over-drying, there was room for optimization. The application developed in this project attempts to optimize energy consumption by shortening the drying duration. The estimates were calculated by using logistic regression, which created a logistic growth model. A confidence interval was implemented to remove irrelevant data in order to calculate the most optimal estimates.

By visualizing historical data and calculating an estimated growth model, the drying operator can better understand how the drying cycle is progressing. This way, the operator can combine previous work experience with concrete data to reduce the drying duration and correspondingly optimize the energy consumption. Optimizing energy consumption contributes to creating a sustainable solution for a climate-positive and cost-efficient future. In addition, the material quality of the products is improved, and the application demonstrates the potential that analysis of energy data can bring to their field of work. As a technology demonstrator, the application serves as a foundation.

11.1 Future Development

While the resulting application developed satisfies the requirements and objectives set at the beginning of the work, several additional functionalities were suggested during the development process by the group. However, due to time limitations, these were not prioritized. Nonetheless, future development could take these into account.

- **Evaluate the accuracy of the solution** A critical part of developing something new is to evaluate if the solution is an improvement compared to the current solution. For future development, the application should be tested using sample data in order to evaluate the accuracy of the application.
- **Adjustable estimate** Computer learning could allow for the creation of an adaptive estimate that could adjust based on the movement of the current life cycle. The group had no previous experience with computer learning, and as such, this was not implemented. Adjustable estimates would be the best solution, albeit complicated and time-consuming.
- **Update drop down menu** Updating the drop-down menu when a new cycle type is being inputted. This would make it possible to create a new type of cycle by inputting a non-existing combination of input parameters.
- **Show manual moisture with the graph** Displaying the result moist together with its corresponding cycle would make it easier for the operators to see the previous results and,

based on that, decide when the ongoing cycle is done.

Bibliography

(2022). URL: https://miro.medium.com/max/1400/1*zOI_ztYqnzyWRkBffeOsRQ.png (visited on 12th May 2022).

"Timber" vs. "Lumber": How To Cut Out The Confusion Between Them (Sept. 2021). URL: https://www.dictionary.com/e/timber-vs-lumber/ (visited on 12th Mar. 2022).

10 Most Important Features of C Language (May 2022). URL: https://www.simplilearn.com/tutorials/c-tutorial/features-of-c-language (visited on 29th Mar. 2022).

11 Most In-Demand Programming Languages in 2022 (2022). URL: https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/ (visited on 29th Mar. 2022).

7 Reasons Bitbucket is Better than GitHub (the Ultimate Smackdown) (2022). URL: https://www.idalko.com/bitbucket-vs-github/ (visited on 27th Apr. 2022).

Acceptance testing (2022). URL: https://www.javatpoint.com/acceptance-testing (visited on 19th May 2022).

Advantages And Disadvantages of C Programming Language (Oct. 2021). URL: https://unstop.com/blog/advantages-and-disadvantages-of-c-programming-language (visited on 29th Mar. 2022).

Arias, Dan (Feb. 2021). *Adding Salt to Hashing: A Better Way to Store Passwords*. URL: https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/ (visited on 10th May 2022).

Aslak, Tveito. et al. (2010). *Elements of Scientific Computing*. 7th. Springer. URL: https://link.springer.com/content/pdf/10.1007/978-3-642-11299-7.pdf (visited on 20th Apr. 2022).

Awati, Rahul (2022). *Integration testing or integration and testing (I&T)*. URL: https://www.techtarget.com/searchsoftwarequality/definition/integration-testing (visited on 12th May 2022).

Backend (Apr. 2020). URL: https://techterms.com/definition/backend (visited on 11th May 2022).

Bansal, Anshul (Nov. 2021). *Best Practices For Unit Testing In Java*. URL: https://www.baeldung.com/java-unit-testing-best-practices (visited on 12th May 2022).

Becker, Davina (2022). *6 Advantages of Data Backup and Recovery*. URL: https://www.peerspot.com/articles/6-advantages-of-data-backup-and-recovery (visited on 10th May 2022).

BigQuery API Client Libraries (May 2022). URL: https://cloud.google.com/bigquery/docs/reference/libraries (visited on 18th Apr. 2022).

BigQuery Connection API Client Libraries (2022). URL: https://cloud.google.com/bigquery/docs/reference/bigqueryconnection (visited on 11th Feb. 2022).

Bjørnstad, Jan (June 2018). *Konfidensintervall*. URL: https://snl.no/konfidensintervall (visited on 27th Apr. 2022).

Braut, Geir Sverre (Dec. 2021). *Regresjonsanalyse*. URL: https://snl.no/regresjonsanalyse (visited on 26th Apr. 2022).

Class GridPane (2015). URL: https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html (visited on 9th May 2022).

Class Platform (2015). URL: https://docs.oracle.com/javase/8/javafx/api/javafx/application/Platform.html (visited on 28th Apr. 2022).

Environmental Limits to Population Growth (2022). URL: https://openstax.org/books/biology/pages/45-3-environmental-limits-to-population-growth (visited on 28th Apr. 2022).

Exponential & logistic growth (2022). URL: https://www.khanacademy.org/science/ap-biology/ecology-ap/population-ecology-ap/a/exponential-logistic-growth (visited on 27th Apr. 2022).

Farger (2022). URL: https://brand.moelven.no/brandcenter/no/moelven/component/default/66510 (visited on 23rd Apr. 2022).

Farger (2022). URL: https://brand.moelven.no/brandcenter/no/moelven/component/default/66510 (visited on 19th May 2022).

FB continuous kiln (Oct. 2017). URL: https://news.cision.com/valutec/i/fb-continuous-kiln,c2261652 (visited on 19th May 2022).

Flödesmätare Ultraljud för vatten och alla andra typer av vätskor och flytande gaser. (2022). URL: http://www.processcenter.se/produkter/flodesmatare/flodesmatare-ultraljud/flodesmatare-lrf-2000s-ultraljud/ (visited on 30th Mar. 2022).

Framtiden bygges i tre (2021). URL: https://www.moelven.com/globalassets/konsern/finansielle-rapporter/barekraftsrapporter/2021-Baerekraftsrapport.pdf (visited on 13th May 2022).

Gibbs, Samuel (Dec. 2016). *Passwords and hacking: the jargon of hashing, salting and SHA-2 explained*. URL: https://www.theguardian.com/technology/2016/dec/15/passwords-hacking-hashing-salting-sha-2 (visited on 12th May 2022).

Grunnprinsipper for personellsikkerhet (Aug. 2020). URL: https://nsm.no/regelverk-og-hjelp/rad-og-anbefalinger/grunnprinsipper-for-personellsikkerhet/introduksjon/ (visited on 5th May 2022).

Håkon Toverød, Audun Øvrum (2022). *Uttak av kjerne furuved*. URL: https://www.treteknisk.no/resources/filer/publikasjoner/fokus-pa-tre/Fokus-nr-29.pdf (visited on 8th Mar. 2022).

Hamilton, Thomas (Apr. 2022a). *Integration Testing: What is, Types, Top Down & Bottom Up Example*. URL: https://www.guru99.com/integration-testing.html (visited on 12th May 2022).

— (Apr. 2022b). *Java Versions and Features*. URL: https://www.guru99.com/unit-testing-guide.html (visited on 14th May 2022).

Hash Functions (June 2022). URL: https://csrc.nist.gov/projects/hash-functions (visited on 3rd May 2022).

Haugen, Marie (2019). *Kostnader ved individuell måling av varme og kjøling*. Norges vassdrags- og energidirektorat. URL: https://publikasjoner.nve.no/eksternrapport/2019/eksternrapport2019_48.pdf (visited on 7th Apr. 2022).

Hoffman, Barbara (2022). *How Do Passwords Work?* URL: https://delinea.com/blog/how-do-passwords-work (visited on 3rd May 2022).

How Does Single Sign-On Work? (2022). URL: https://www.onelogin.com/learn/how-single-sign-on-works (visited on 7th May 2022).

How to Choose Code Repository for Your Project (Dec. 2021). URL: https://huspi.com/blog-open/software-code-repositories/ (visited on 27th Apr. 2022).

Hurley, John (July 2020). *The Importance of Administrator Accounts: Visibility & Control*. URL: https://www.smartfile.com/blog/importance-administrator-accounts-visibility-control/ (visited on 12th May 2022).

Integration testing (2022). URL: javatpoint.com/integration-testing (visited on 12th May 2022).

Introduction to the Standard Directory Layout (May 2022). URL: https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html (visited on 26th Apr. 2022).

Java 8 still dominates, but Java 17 wave is coming (n.d.).

JavaFX Application Structure (2021). URL: https://www.javatpoint.com/javafx-application-structure (visited on 17th Apr. 2022).

Jonathan, Sandals and Mitchell Brad (Mar. 2022). *Top Programming Languages of 2022*. URL: https://www.codingdojo.com/blog/top-programming-languages (visited on 29th Mar. 2022).

Kanban (2022). URL: https://trello.com/b/9dxLNIDH/bachelor-project (visited on 19th May 2022).

Khaishagi, Zaid (Jan. 2019). *Cryptography: Explaining SHA-512*. URL: https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1 (visited on 6th May 2022).

Millington, Sam (Jan. 2022). *Hashing a Password in Java*. URL: https://www.baeldung.com/java-password-hashing (visited on 12th May 2022).

Moelvens bærekraftsmål (2022). URL: https://www.moelven.com/no/miljo-og-barekraft/moelvens-barekraftsmal/ (visited on 13th May 2022).

Moelvens logo (2022). URL: https://brand.moelven.no/brandcenter/no/moelven/component/default/56408 (visited on 19th May 2022).

Moose River (Oct. 2019). URL: https://www.valutec.ca/references/moose-river/ (visited on 19th May 2022).

Paul, Mano (2013). *Official (ISC)2 Guide to the CSSLP CBK*. 2th. Auerbach Publications. URL: https://cloudflare-ipfs.com/ipfs/bafykbzacebsmj3lhjrmbqkmy7fx2wak4spfguy6fqms3xyscz4qualo3yqq7m?filename=Mano%20Paul%20-%20Official%20(ISC)2%20Guide%20to%20the%20CSSLP%20CBK-Auerbach%20Publications%20(2013).pdf (visited on 26th Apr. 2022).

Perez, Ross (May 2016). *Announcing Tableau Server on Google Compute Engine*. URL: https://www.tableau.com/about/blog/2016/5/announcing-tableau-server-google-compute-engine-54147 (visited on 12th May 2022).

Postning (Dec. 2016). URL: https://www.traguiden.se/om-tra/materialet-tra/sagverksprocessen/sagprocessen/postning/ (visited on 16th Apr. 2022).

pp_pankaj (Aug. 2019). *System Testing*. URL: https://www.geeksforgeeks.org/system-testing/ (visited on 19th May 2022).

Radtke, Maggie (2022). *Improving Security with SSO and MFA*. URL: https://www.huntress.com/blog/improving-security-with-sso-and-mfa (visited on 12th May 2022).

Sanket (Jan. 2019a). *fx-progress-circle*. URL: https://github.com/torakiki/fx-progress-circle (visited on 24th Feb. 2022).

— (Jan. 2019b). *The exponential cost of fixing bugs*. URL: https://deepsource.io/blog/exponential-cost-of-fixing-bugs/ (visited on 12th May 2022).

Santos, Omar (June 2015). *SHA512 Checksums for All Cisco Software*. URL: https://blogs.cisco.com/security/sha512-checksums-for-all-cisco-software (visited on 3rd May 2022).

setDaemon (2020). URL: https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html#setDaemon(boolean) (visited on 5th May 2022).

SHA-2 (June 2022). URL: https://en.wikipedia.org/wiki/SHA-2#Cryptanalysis_and_validation (visited on 3rd May 2022).

Simplilearn (Oct. 2021). *What is R: Overview, its Applications and what is R used for?* URL: https://www.simplilearn.com/what-is-r-article (visited on 3rd May 2022).

Singh, Japkeerat (Sept. 2018). *What is JavaFX and how is it different from Swing and AWT?* URL: https://medium.com/@japkeerat21/what-is-javafx-and-how-is-it-different-from-swing-and-awt-54de995e4869 (visited on 23rd Apr. 2022).

Sridhar, Aditya (Aug. 2018). *An introduction to Git: what it is, and how to use it.* URL: https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/ (visited on 27th Apr. 2022).

Stickney, Jacob (July 2021). *Hashing & Integrity — The "I" in the CIA Triad*. URL: https://jacob-e-stickney.medium.com/hashing-integrity-the-i-in-the-cia-triad-98b722b6fe39 (visited on 3rd May 2022).

— (May 2022). *Difference Between Hashing and Encryption*. URL: https://www.clickssl.net/blog/difference-between-hashing-vs-encryption (visited on 6th May 2022).

Survey (2022). URL: https://freeonlinesurveys.com/s/WDIbQRPZ?fbclid=IwAR3clp0za4sPo-I9jtl2yZ_sFsnHRtL8j-XRqDYYYlx2lSHQbRyS4o73MkA#/4364412 (visited on 19th May 2022).

Take Action for the Sustainable Development Goals (2022). URL: https://www.un.org/sustainabledevelopment/sustainable-development-goals/ (visited on 14th May 2022).

TIOBE Index for May 2022 (2022). URL: https://www.tiobe.com/tiobe-index/ (visited on 29th Mar. 2022).

Torkmetoder (2022). URL: https://www.tracentrum.se/sv/publikationer/tratorkning/torkmetoder/ (visited on 6th Apr. 2022).

Tronstad, Sverre (Dec. 2000a). *Tørking av gran og furu*. URL: https://www.treteknisk.no/resources/filer/publikasjoner/fokus-pa-tre/Fokus-nr-39.pdf (visited on 14th Apr. 2022).

— (Mar. 2000b). *Trefuktighet-tørking*. URL: https://www.treteknisk.no/resources/filer/publikasjoner/fokus-pa-tre/Fokus-nr-38.pdf (visited on 25th Apr. 2022).

— (2013). *Tre og fuktighet*. 4th. Treteknisk.

Våre produksjonsenheter (2022). URL: https://www.moelven.com/no/om-moelven/divisjon-timber/mer-om-vare-sagbruk/?gclid=CjwKCAjw9qiTBhBbEiwAp-GE0e-kd8-%20PgMl7xYmFAj5rZXave3FUUycOFKabNV7ImBwE (visited on 28th Mar. 2022).

What Is a Front-End Developer? (May 2022). URL: https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html (visited on 21st Apr. 2022).

What is Backup? Benefits and Evolutions. (May 2022). URL: https://www.digitaldistribution.co.th/news-articles/what-is-backup/ (visited on 12th May 2022).

What is Git? What benefits does Git offer? (2022). URL: https://guide.quickscrum.com/git-guide/ (visited on 27th Apr. 2022).

What is Multi-Factor Authentication (MFA) and How Does it Work? (2022). URL: https://www.onelogin.com/learn/what-is-mfa (visited on 12th May 2022).

What is Regression Analysis and Why Should I Use It? (June 2021). URL: https://www.alchemer.com/resources/blog/regression-analysis/ (visited on 26th Apr. 2022).

Why Git for your organization (2022). URL: https://www.atlassian.com/git/tutorials/why-git (visited on 27th Apr. 2022).

Why Joda Time? (Mar. 2022). URL: https://www.joda.org/joda-time/ (visited on 12th May 2022).

Appendix

A Project Description

Optimalisering av energiforbruk

Moelven er et skandinavisk industrikonsern, etablert i 1899 med hovedkontor i Moelv. Alle produksjonsenhetene er lokalisert i Skandinavia, som også er hovedmarkedet. Moelvens visjon er å være det naturlige valget for folk som skal bygge og bo skandinavisk. Moelven-konsernet omfatter i alt 33 produksjonsselskaper fordelt på 41 produksjonssteder i Norge og Sverige. I tillegg har vi salgsapparat i Norge, Sverige, Danmark, Storbritannia, Tyskland og Kina. Vi har ca. 3200 ansatte som hver dag jobber for å gi deg gode rom.

Oppgaven

Mye trelast tørkes i dag industrielt i store kammertørker med ferdig definerte klimaprogrammer basert på blant annet treslag, tømmerdimensjon, trelastdimensjon, krav til fuktighet og andre kvalitetsparametere. Generelt er kunder mer følsomme for at produktene er for fuktige enn for tørre. Dette fører til at trelasten ofte tørkes noe lenger ned i fuktighet enn den egentlig trenger. Overtørking resulterer i redusert kapasitet for våre tørker, økt termisk- og energiforbruk, samt forringet kvalitet på produktet. Det er prøvd forskjellige måter å måle fukt underveis i tørkeprosessen for å få bedre treff i sluttfukt, men i dag finnes ikke et fullgodt alternativ som gir gode nok målinger samtidig som det er enkelt håndterbart i en industriell prosess.

Oppgavens mål

I og med en økt grad av digitalisering på Moelvens sagbruk begynner vi nå å få data som beskriver våre prosesser godt. Blant annet har vi data for energiforbruk på tørkene. Siden tørking av trelast er en prosess som i stor grad styres av termodynamiske prinsipper, burde det være mulig å bruke spesifikk termisk energi (kWh/m3) som et mål på hvor mye vann som er fjernet fra trevirket og da også indirekte hva fuktnivået ligger på.

Ønsket med prosjektet er å kunne utvikle en applikasjon som kan predikere og visualisere tørkeprosessen bedre ved hjelp av energimålinger og annen tilgjengelig produksjonsdata, med et mål om å kunne redusere tørketid og energiforbruk.

Kontaktpersoner

- Ylva Kleiven, prosjektingeniør, Moelven Timber, epost: ylva.kleiven@moelven.com
- Stefan Djupvik, leder IT Utvikling, Moelven Industrier, epost: stefan.djupvik@moelven.com

B Project Agreement

NTNU
Kunnskap for en bedre verden

Group contract

Bachelor thesis

Expectations:

Attendance:

Each group member should attend all meetings and project related lectures. A minimum of 6 hours each day or 30 hours a week is required by each student. Although more hours than this is expected by each group member. If a group member is not able to attend project related activity, then they should notify the rest of the group as soon as possible.

Responsibility:

Each group member has a responsibility to follow the lectures, engage in discussions and is expected to contribute to value to the group. Each group member is responsible for arriving on time to scheduled meetings, rested and prepared for the day. It is expected that each group member shows up with a positive attitude and dare to speak out about both positive and negative subjects within the group. We are a team, and each teammate is expected to help and cheer each other on.

Workload:

We will try to divide the workload as equal we can. Tasks will be divided between members at the first meeting of every day/week, or when needed. Every team member is expected to complete their tasks to the best of their ability.

Communications:

Most communication will take place on Facebook's Messenger app for general communications and to organize meetings.

Leader role:

None of the group members have a typical "leader role", instead we are all working in all fields regarding the project. This is to gain different levels of perspectives for problem solving. However, in each field we have a designated chief that has the main responsibility for their respected field. These fields are designated into:

- Lead deveopler, report – Eilert
- Communications, report– Karin
- Documentation, meetings – Mads

Time schedule:

As a minimum requirement, each group member is expected to work 30 hours a week, ie 6 hours 5 times a week. We plan to work from 8:15 AM to 4:00 PM from Mon-Fri. We will have a lunch break from 12:00 PM – 12:30 PM. In addition, everyone must remember to list the number of hours they work themselves in our hour overview in Toggl track.

Responsibility of work assignments:

We have decided that everyone will work with all the work tasks of the bachelor, whether it concerns programming, report writing, meeting reports, etc. Thus, all the work tasks will have different points of view and perspectives from the different group members. Which can contribute to different solutions. But in order to avoid clutter or possible uncertainty about how the work task is to be solved, there must always be a main person responsible for the various work tasks one can consult with.

Sickness:

We reckon that it is unlikely that some of the group members will become ill over a longer period. But if this should happen, we think we will find a solution to it then, because then we will hopefully know more about the extent of the sickness, and what is possible for the person to do. The fact that someone in the group should catch a cold or get corona may be a possibility. Should this happen, meetings will be held in teams, and much of the communication zone will then take place digitally.

Other issues:

If repeatedly bigger issues were to occur, we will find a solution to the problem when the time comes. If we are not able to find a solution to the problem, we will ask for aid from our supervisor: Tom Røise

How to deal with unmet expectations or other problems:

To deal with unmet expectations we will run a group meeting and address the feedback and in which parts of the assignment the requirements were not met. Further on we will, as a group, find out what measures we must take to re-do and re-deliver the assignment to pass.

Most of the unexpected problems we face during the project period will be addressed during the team meetings, and together the team will find an appropriate way to approach the issue. Communication with- and guidance from the teaching staff is important to achieve the wanted achievements.

If a team member fails to attend at least 80% of the meetings or repeatedly does not complete his given tasks, then we will address this with the member it concerns and give them a chance to improve. If the issue does not improve then we will discuss the situation with the associate professor and/or student assistant to find a solution to the problem.

Process for peer feedback:

Peer feedback can be useful as a perspective change, however, must be taken with a grain of salt as the feedback might be unverified. We should discuss the feedback with the whole group to gain different viewpoints and find the best approach to the problem at hand.

Group member	E-mail	Telephone	Date	Signature
Karin Emilie Pettersen	Karinep@stud.ntnu.no	909 31 758	12.01.22	Karin E. Pettersen
Eilert Mikal Hananger Tunheim	emtunhei@stud.ntnu.no	959 27 498	12.01.22	Eilert Tunheim
Mads Greni Arnesen	Madsga@stud.ntnu.no	976 71 740	12.01.22	Mads Arnesen

C Project Plan

Project plan - Energy consumption optimization

Authors:

Eilert Mikal Hanager Tunheim,

Karin Emilie Pettersen,

Mads Greni Arnesen

In collaboration with:

Spring, 2021

Table of Contents

List of Figures

1 Goals and limitations

1.1 Background

Moelven dries wood in huge industrial drying chambers. These chambers are surrounded by pipes containing heated water in order to dissipate heat. Logs are supplied to Moelven, and these logs contain bark. The bark is removed and burned to heat the flowing water. A number of fans are used to circulate the air to distribute heat. The dissipated warmth combined with the circulating air, results in an efficient drying environment.

Moelven's current solution uses predefined drying programs. These programs are based on a number of input parameters that estimates the drying duration. These parameters are: tree species, weight, moist level requirements, tree dimensions: length, width, height. The moist levels in wood determines the quality and material strength of the material which plays a critical role for later use in construction and other products. In general customers are more sensitive to moist wood compared to too dry. This usually results in the wood being dried more than necessary which this report will refer to as, over-drying. Over-drying results in reduced capacity for the drying chambers, increase energy consumption, reduced quality in the products and lastly increase expenses.

Moelven Industrier ASA is Scandinavia's leading supplier of wood based construction products [4]. Moelven deliver products and solutions to commercial and industrial customers, as well as to construction and contractor customers in the project market. Most of their companies work with wood as a raw material. This is because wood is a renewable resource, and sustainability is crucial to their competitiveness. Hence their products and system solutions provide flexibility and reuse, and are an active contribution to their common global environmental responsibility.

1.2 Project Goals

The objective of this project is to create an improved estimate for the drying duration. Additionally, visualize this estimate and sensor data through a user friendly interface. The project goals are divided into three categories. Learning goals define the knowledge desired to be achieved from creating the application and writing the report. Result goals represent what the final product should be and what it should look like. Lastly, effect goals specify how our product can be useful for Moelven.

1.2.1 Effect Goals

The application will be used to substitute a solution that, due to modern standards, can be seen as old-fashioned. The goal is to optimize the drying process and create a better estimate than today's solution in order to save time and consequentially energy and money. By lowering the drying duration, less bark is used as fuel and Moelven can sell this saved bark for a profit instead. Lastly, the application should be open for further expansion.

1.2.2 Result Goals

The final solution should be an overall improvement to the already existing solution. Based on the given attributes, the following result goals were established:

- Use live input data from sensors to estimate when the drying process is finished;
- Create an user friendly interface that can visualize real time data and display progression;
- Well written- and organized report.

1.2.3 Learning Goals

The learning goals for this project is to acquire knowledge through theoretical and practical problem solving. Among these include:

- How to professionally communicate ideas and file sharing;
- How to apply theoretical knowledge into practice;
- How to work as a team and efficiently divide workload, schedule meetings and document the process during the project duration;
- Improve programming skills in:
  - Languages: Java, Python;
  - Error handling: Exceptions;
  - Testing: Unit tests, test automation;
  - Code documentation: Javadoc.

1.3 Limitations

The starting date for project is 10.Jan.2022 and the deadline 20.May.2022. However, an internal deadline is set two weeks prior to the submission deadline, to take into consideration possible unexpected occurrences.

The technology-limitations:

- The accuracy of the instruments: sensors, machinery;
- The application must function on a independent workstation;
- The application must be able to run on Windows;

1.4 Tools

The resources used during this project are going to evolve as the project develops. This thesis will include a diversity of digital tools and software. Below is a list of what resources we have used and envisage using.

Name	**Type**	**Use case**
Toggl Track	Time tracking tool	Tracking hours and dividing into categories
Overleaf	Cloud-based LaTeX editor	Project Rapport
NTNU Open	Document Archive for NTNU	Reading at other bachelor theses for inspiration
Intellij	IDE for back-end coding	Back-end coding in Java
Azure	Cloud-based hosting environment	Deployment of the application
PyCharm	Python IDE	Development application
Figma	Design tool	Prototype Creation
Draw IO	Diagram creation	UML-, Flow-, Sequence diagrams
Trello	Kanban table	Assigning- and prioritize tasks
One drive	Cloud based file sharing	File sharing
Gitlab	Version control	Branch merging and version control

Table 1: List of tools expected to be used during the project

2 Scope

2.1 Subject Field

Moelven wants to improve their current solution of drying trees. By incorporating live sensor data when estimating the drying duration, they can create a better estimate compared to the current solution. The solution should offer the employees a user friendly graphical interface that categorizes the trees after dimension, show the progression and show the time left of the drying. Users of the application should be given a notification when the drying process is completed.

2.2 Production Flow

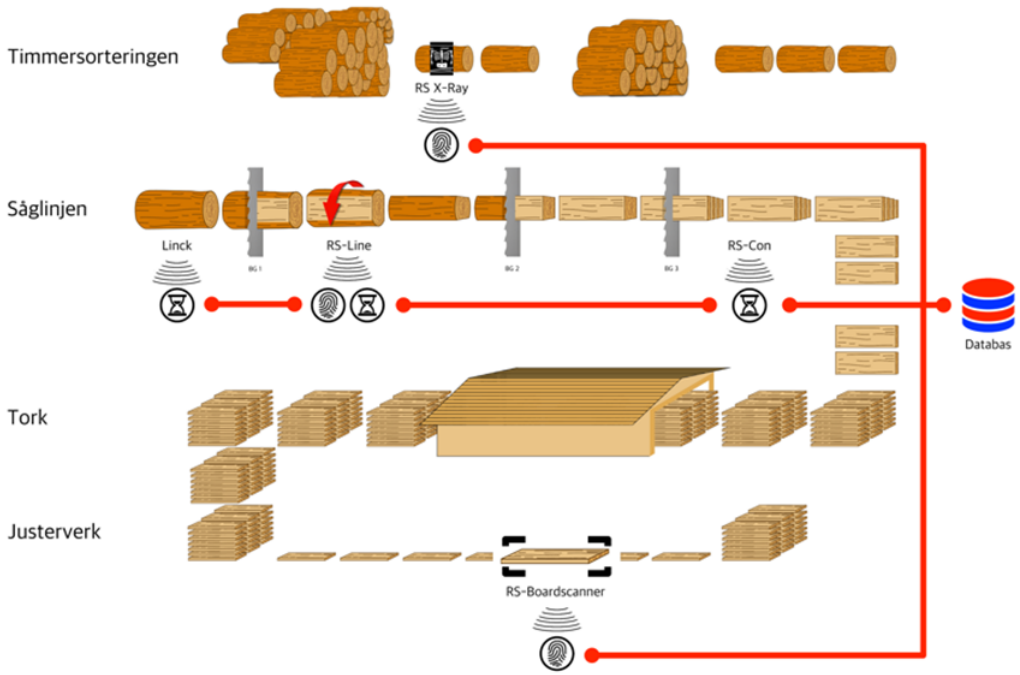


Figure 1: Production flow

The production flow at Moelven begins with timber being delivered to the warehouse in Valåsen. Here the logs are measured using a 3D / X-ray scanner. Furthermore, the logs are categorized and cut, in order to benefit the most out of the logs. Next, they are sorted in timber compartments based on wood species and dimensions. Further, the barking process begins. The compartment is emptied and they are placed on the timber inlet of the barking machine. Here the bark is milled away, and the timber is sent on towards the saw line. Thereafter, the timber is measured again by Microtech 3D scanners, which take place before the timber arrives at the saw. This is a control system determines how the log should be optimized. This means how much is to be reduced (milled away), which way the log is to be rotated and how it should be sawn.

Further, three saws will be used to cut the timber. Saw one is carried out after the optimization of the logs, here the side edges of the log are sorted to the edging, while the rest of the log continues on the saw line and is measured. Then the process continues to saw two, where the sawing process from saw one is repeated, this results in a center block. The side planks are sorted for the edging. After that the center block is sawn into several equal pieces by saw three, which

in turn are measured by rsCon. Furthermore, this ends up in the raw sorting for center pieces. The side planks end up on the edging where they are sawn to optimized size, then sent on to raw sorting for side planks.

On the raw sorting, the planks are dropped into compartments based on dimensions. Then one compartment is emptied so that each package from the raw sorting only contains planks with the same dimensions. Further, the package information is registered. The parcel registration system is integrated with siPal parcel logistics, which keeps track of the parcel's position between raw sorting and adjustment. Further the drying process takes place where sensor data is collected. When a package is dried, it is stored again, before it is split on the adjuster. Here, each individual plank is scanned before the planks are again sorted into compartments based on quality, type of wood and dimensions. Lastly, the packaging takes place. A compartment is emptied, the planks are parceled and sent to customer.

2.3 Prototype

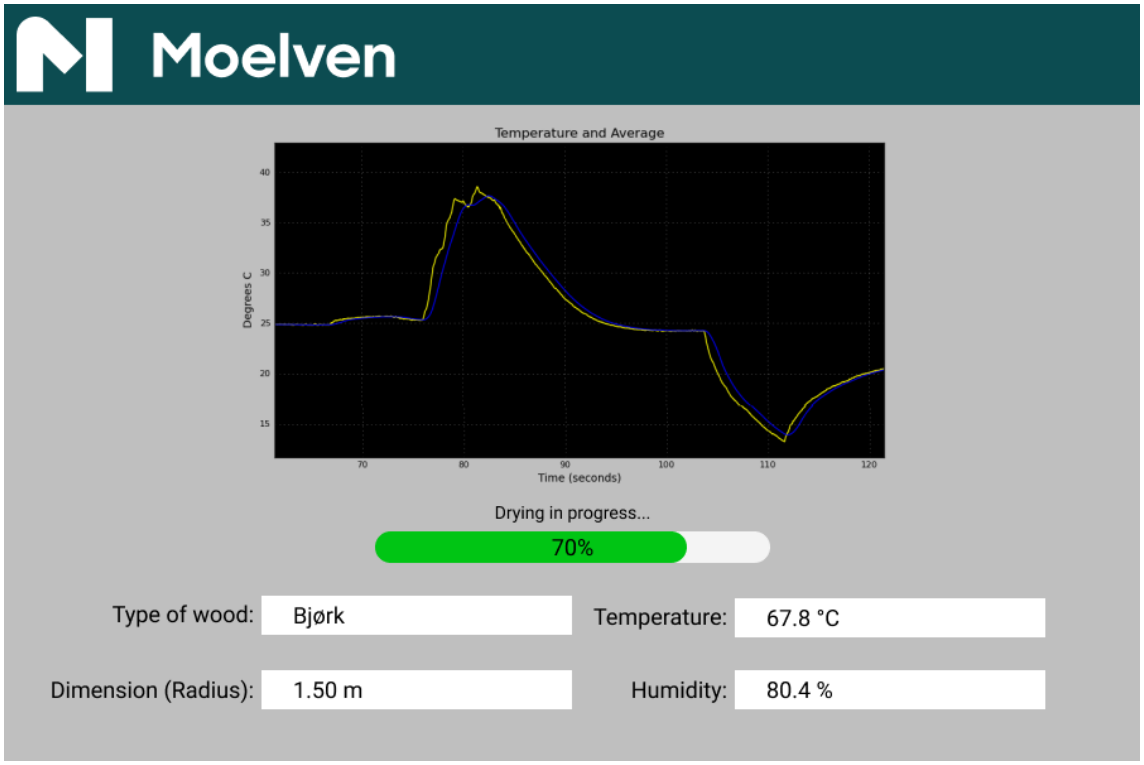


Figure 2: Wireframe prototype

Figure 2 is an attempt to visualize the team's vision of what the graphical user interface(GUI) could look like. The prototype was created using Figma [2] and contains a number of displays for individual sensors. We imagine a graph that shows the temperature over time in the drying chamber as well as progress bar that displays the progress of the drying process. In addition, displays for the most relevant information such as; Type of wood, dimension of the wood, current temperature in the chamber, and lastly the humidity in the chamber.

Interviews with Moelven's employees will be done in order to create an application that fits the employees requirements and wishes. The GUI is expected to change drastically later in the development process, this prototype is meant as a start and foundation for the final product.

2.4 Task Description

During this project, an application will be created that visualize and estimate the drying duration. The target group consist of Moelven's employees and will combine computer analysis with their work experience in relation to drying. In theory, this should create the most accurate estimate for determining when the drying process is finished. The employees might not have extensive IT-knowledge, for that reason it is important to create a simple and user friendly interface. Input parameters should be named intuitively to understand and interact correctly with the application. The number of input parameters should be kept to a minimum to avoid confusion. Additionally, the input parameters needs to be spaced out in a clean and symmetrical structure.

Before the drying process begins, the chamber need to warm up, this is called the warm-up phase. During the warm-up phase, water is sprayed to assure that the drying does not start before the chamber is at it's operational temperature. The application need to take this phase into account, since it is not part of the drying process. The drying duration starts when the chamber reach the operational temperature.

When the drying process is finished, the moisture level is tested, if the level is too high the drying process is started again which can result in over-drying. The application will make it easier for workers not to disturb the drying process. Our application will give them more concrete estimates than before, using application calculations. The employees will receive two estimates of moisture in the application. One estimate is calculated based on previous data of the drying, while the other estimate will be calculated based on sensor data and thermo-physical formulas. For further development, it will be possible to make an estimate that tells the accuracy of the two estimates on when the drying process is completed. In addition AI and machine learning can be used to calculate the estimates better, based on machine learning and the workers competency.

When the drying-process is finished based on the estimates, the application will send out a notification. The notification can be sent to the employees phone, in addition to a clear message on the interface screen. The notification tell the employees to do a manual check of the moisture level. The algorithm need to take in to consideration the incorrect measurements that may occur. A margin of error has to decide what data is valid, and what should be rejected. Using statistics, the algorithm should calculate the credibility of the estimate.

2.5 Task Boundaries

There will not be done any work on Moelven's existing software. However, the applications will be created using sample data provided by Moelven to calibrate a model. This will make the application as compliant as possible and available for further development. The thesis will not extensively cover thermal physics, it will only cover the necessary terminologies to easier understand the estimations and be able to create the application. Considering, all the sensors being installed, no hardware work will be needed to complete this project.

3 Organization Of The Project

3.1 Organization Chart

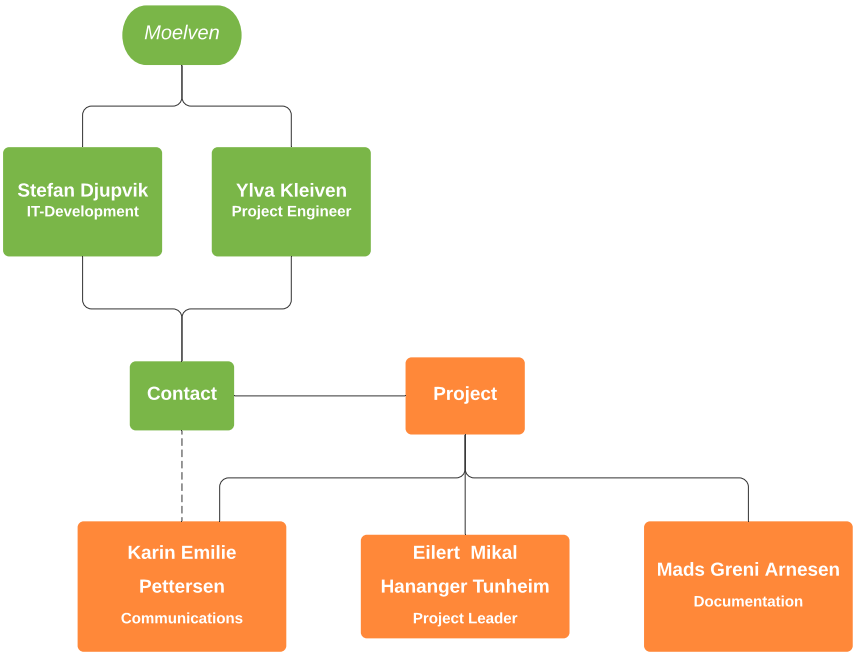


Figure 3: Chart of the individuals involved in the project and their respective roles

3.2 Roles And Responsibilities

There is a set of important responsibilities regarding the project. All group members are developers and report writers in order to gain multiple perspectives and maximize the learning outcome. The roles assigned to each member are intended as a general leader role in their respective fields. In addition, the group as a unit is responsible for operating the environments around this project.

- **Project Supervisor - Prof. Tom Røise**
  Provide guidance and supervision throughout the duration of the project, thanks to his prior experience and knowledge
- **Project Owner - Moelven, Ylva Kleiven, Stefan Djupvik**
  Represent the challenges and share Moelven's vision of the project through scheduled meetings.
- **Project Leader - Eilert Mikael Hananger Tunheim**
  Has the overall responsibility for the project and to manage the project team. Primarily responsible for meeting the deadlines that has been set and the time limits we have set for ourselves.

- **Communications - Karin Emilie Pettersen**
  Responsible for communication, arranging the meetings and keeping in touch with our contacts trough email and Microsoft Teams.
- **Documentation - Mads Greni Arnesen**
  Responsible for documenting and following up all meetings, and structure these notations so it is easy to access and understand for all group members.

3.3 Rules & Routines

A list of rules and routines is created that defines how the group should work during the project period, see *Rules & Routines* in Appendix. These rules and routines is created to ensure efficient work and to establish clear boundaries in case of conflict.

In addition to the rules & routines, a "Plan for the day" is expected to be created every work day to ensure a clear understanding of what to do, who are going to do it and what the objective for the day is.

4 Planning, Reporting And Following Up

4.1 Development Methodology

Software development methodology is used in order to ensure efficient work, distribute tasks and set deadlines for the project-team. In general, it defines a set of norms between a group of people that tells them how they are going to work and communicate. Specifically, it refers to a structured set of activities that include many different processes such as: specification, design & implementation, validation, and evolution. Most methodologies can be categorized into one of three categories; linear, iterative or a agile model.

4.2 Choosing A Development Model

When choosing a development methodology model it is important to have a few things in mind;

- The University's requirements for the project;
- Requirements and wishes asked by the client;
- Development team preferences;
- How the requirements and specifications is defined and if they can be changed later.

Waterfall is a linear development model with a predefined sequence of phases [5]. Historically, this model was primarily used in the early days of software development due to it's simplicity and it was easy to understand, particularly for new developers. The waterfall model is sequential where each phase is fed into the next phase. As a consequence, testing would only occur at the end of the development, unfortunately, some issues might be harder to fix at a later stage. In addition, this means that the client only sees the final product compared to iterations where potential feedback could be given. The waterfall model can be successful if the requirements is clearly defined at the start of the process. However, given it's linear feature, complex- and unclear requirements given by Moelven, the waterfall model is undesirable for this project.

Alternatively, an iterative development model is more forgiving towards vague requirements, compared to a linear model such as waterfall. An iterative model is a repeating cycle, also know as iterations, with fixed phases. The model focuses on an initial implementation with gradual improvements over multiple iterations [3]. The iterations continue until the client is satisfied with the product. Although this model is more suitable, it is strict considering deadlines. Given the lack of work experience within the project team, it is difficulty to predict the time needed in order to complete a given task, this is what makes an iterative model undesirable.

The last general category for development methodology is an agile development model. In contradiction to iterative- and linear, agile models have a different approach where the focus is directed towards satisfying the end user rather than emphasizing documentation [5]. There where two agile models that was attractive for the team, Kanban and Scrum.

The Kanban model is a model where tasks are assigned to all group members using a to-do list, also know as a Kanban board. The Kanban board follows an "It's done, when it's done" philosophy compared to other methodologies which are usually time-based. This gives the team the time they need in order to complete a task.

Scrum on the other hand breaks down tasks into shorter "sprints" with a duration of 1 to 4 weeks [6]. Scrum is also role based where roles are assigned to the group members. Scrum consist of different activities which differentiates Scrum from other agile development methodologies.

4.3 Model Choice

A combination of Kanban and Scrum will be used as a development model for the project. The Kanban board was attractive for the team because of "It's done, when it's done" philosophy. As mentioned, the team lacks work experience and therefore it is hard to estimate the amount of time each task will take. This philosophy will give the team flexibility in regard to tasks and how much time each task will allocate. In order to efficiently assign tasks it is important to minimize the scope of each task and create as "easy" tasks as possible. If this is not possible, to split up a bigger task into smaller segments should be done. In addition, a limit of 3 tasks per group member should be allowed in the "In Progress" column. Additionally, "For Approval" should be limited to 20 tasks to insure not too many tasks will need to be approved at the end of the day. From Scrum, the group wanted to assign a general leader-role for each group member to be responsible for. Hence, the role based feature from Scrum was more attractive for the group as discussed in the section 3.2 - "Roles and responsibilities". In conclusion, the Kanban model was chosen for assigning tasks, and Scrum was chosen as an organizational structure. This combination gives us the desired features from both models.

Kanban Board

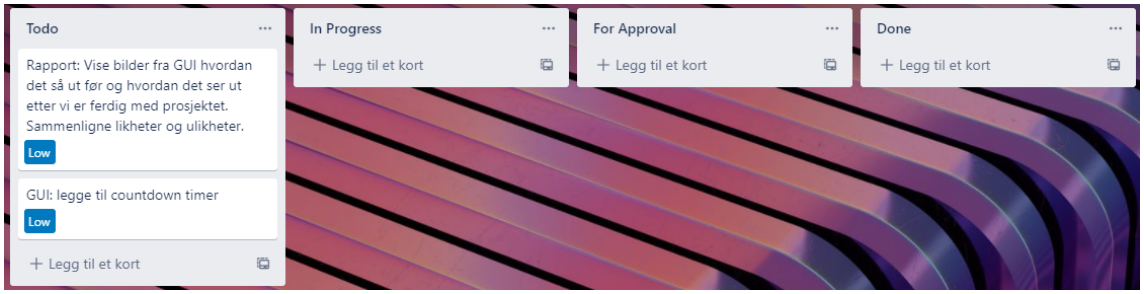


Figure 4: Kanban Table

The Kanban board is created using Trello [1] and consist of four fields. "Todo" uses a priority system of 5 levels (Low, Moderate, High, Critical, ASAP!!@@). The "Todo" list is sorted from highest to lowest priority level, that way the team can easily spot what task needs to be prioritized. When a developer chooses a task, it is moved to "In progress". When the task is done it is moved to "For approval" where the rest of the developers have to review it. The review process will be done at the end of the day to ensure efficient workflow and not overwhelm each other with reviews all day long. After the review process, the task is moved to "Done" and considered completed.

4.4 Supervision Meetings

Every Tuesday a supervision meeting with Moelven will take place. During this meeting the group will inform Moelven around the progress of the project, additionally gain feedback on the current iteration. Generally, questions regarding the project can be asked.

Every Thursday a supervision meeting with the projects supervisor, Tom Røise, will take place. Much like the meeting with Moelven, general progress will be given by the group and question can be asked.

5 Quality Assurance

5.1 Documentation And Source Code

The application will be created using Java and Python therefore JetBrains s essential tools like IntelliJ IDEA and PyCharm will be used for developing. JetBrains provide compleation, refactoring tools and extensive plugin support.

Since the application will be coded in Java, Javadoc will be used as a tool to generate the Java code documentation in the HTML format from Java source code.

Further for the backend part of the application we will be using Java and Python, while for the frontend part of the project we will be using CSS, Javascript and HTML.

5.2 Risk Analysis

Risk analysis has been conducted where a risk-index has been estimated for each issue. This helps the team to determine the degree of severity and what to be aware of during the project. See *Risk Analysis* in Appendix for risk analysis table.

5.3 Tools

5.3.1 JetBrains

JetBrains provide essential tools for software development. They ensure a smooth experience when building, shipping code, planning code work and collaborating. JetBrains have a development for every language and platform. Therefore they will provide the project with necessary tools for the given languages the application will need, such as Java and Python. Intellij will be used for Java programming, while PyCharm will be used for Python programming.

5.3.2 Git and GitHub

Git is an open source version control system which is designed to handle small to larger projects with speed and efficiency. Git allows the ability to have multiple branches that can be entirely independent of each other. It will be used to store revisions of the projects in an online repository, it will also manage and keep track of source code history. GitHub on the other hand is a cloud-based hosting service which will manage the Git repositories of the project. Together these will be used to store related documents as code and final report of the bachelor project.

5.3.3 Azure

Azure is a public cloud computing platform. Which include solutions such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) which can be used for services such as analytics, virtual computing, storage, networking, and much more. Azure is Moelven's hosting environment, which is where the application will be running. Everything of their database is in Azure.

5.3.4 Figma

Figma is a vector graphics editing and protoype tool which is manily web-based, with additional offline features enabled by desktop applications for macOS and Windows. The program will be used to create different prototypes of the application to give our client a better understanding of how the GUI will be. This will give the client an opportunity to provide feedback and requests about what features that are important for them.

5.3.5 Draw.io

Draw.io is a free Google Drive(TM) that is a enable charting program that allows one to create different diagrams online, such as flow diagrams, UML, sequence diagrams etc. It will be used to create UML and sequence diagrams for the application.

5.3.6 Toggl Track

Toggl Track is a time tracking application which is used to track the groups daily activities. This gives detailed insights and an opportunity to optimize the projects workflow by identifying areas such as project plan, codeing, meetings and others to see where to improve.

5.3.7 Overleaf

Overleaf is a cloud based LaTeX editor for typesetting documents, which makes it well suited for scientific and technical documents. It uses syntax based language by defining the layout of the final document. In this project Overleaf will be used to write documents related to the final report.

5.3.8 Trello

Trello is developed by Trello Enterprise, a subsidiary of Atlassian, and is a web based tool usefull for creating Kanban tables. Trello offers a felxible tool so that users can customize Kanban tables according to their needs.

5.3.9 One Drive

One Drive is a cloud based file sharing system, that makes it easy for each group member to access project related files. It also serves as a back up, in case files are lost on local drives.

5.4 Development Routines

Gitlab will be used for version control by utilising Git. For merging of branches, IntelliJ have a an in build Git tool that will be utilized for merging. If merge conflicts occur, the developer that tries to push will be responsible for handling the merge conflict. If there are merge conflicts that can not be resolved, the project leader for the application, Eilert M.H. Tunheim, will have the final say on how to resolve the conflict as best as possible.

6 Schedule

6.1 Gantt Scheme

The project is planned as described in the ghantt chart bellow, see figure 5:

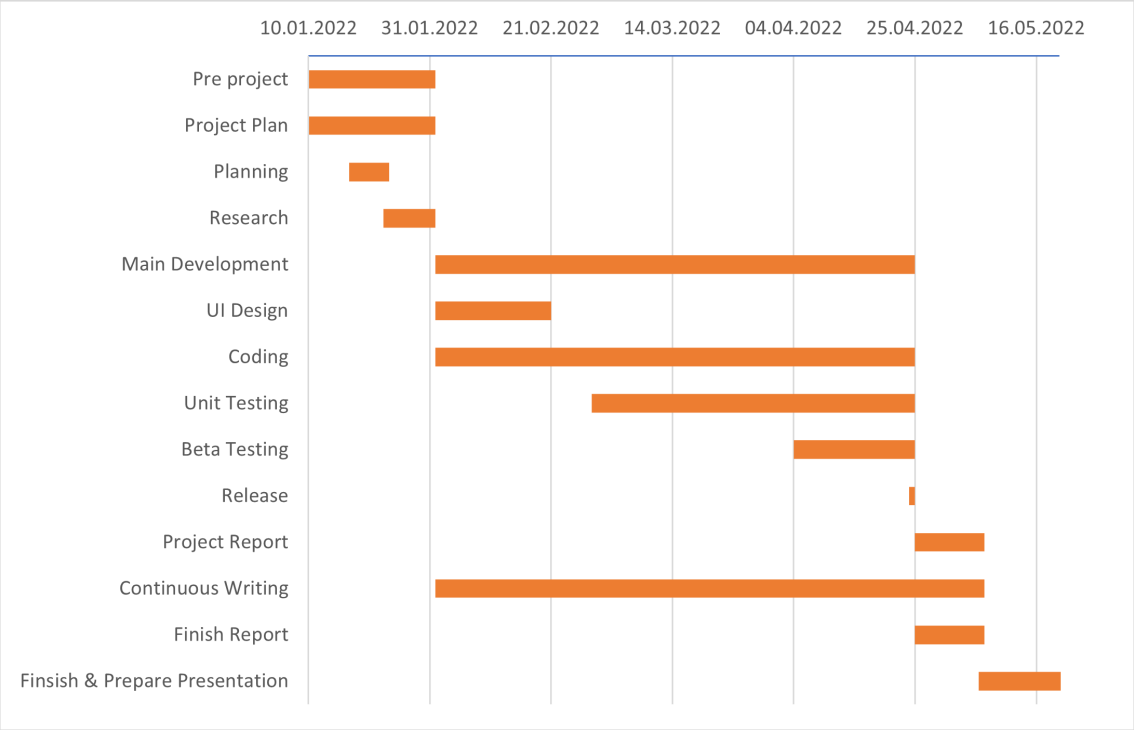


Figure 5: Ghantt Chart

The main development towards the final product consist of a linear path with regular milestones along the way.

6.2 Milestones

When working on a large scale project like this, it is important to split the task into smaller segments. Some of the smaller milestones for the project consist of:

- Setup all environments: Overleaf, Gitlab repository, onedrive filesharing;
- Creating a user friendly GUI;
- High quality, functioning and tested code, along with corresponding documentation;
- Beta test with probable bugs that needs to be fixed;
- Finish the report and deliver the application.

Bibliography

[1] *8 Best Software Development Methodologies*. Jan. 2022. URL: https://trello.com/home.

[2] *Figma*. Jan. 2022. URL: https://www.figma.com.

[3] *Iterative Model: What Is It And When Should You Use It?* Jan. 2022. URL: https://airbrake.io/blog/sdlc/iterative-model.

[4] Moelven. *Om Moelven-konsernet*. Jan. 2022. URL: https://www.moelven.com/no/om-moelven/.

[5] Aliona Nikolaieva. *8 Best Software Development Methodologies*. Jan. 2022. URL: https://www.uptech.team/blog/software-development-methodologies.

[6] *What is Scrum?* Jan. 2022. URL: https://www.digite.com/agile/scrum-methodology/.

D Rules & Routines

Rules & Routines

- Be supportive and polite - Treat each other with respect and encourage one another. Healthy factual debates is expected with contribution from all group members. If an agreement cannot be agreed upon, the project leader(Eilert Mikal Hananger Tunheim) have the final say, unless the other group members find the matter of great importance. In this scenario, the supervisor(Tom Røise) shall be informed on the matter and the supervisor will have the final say in order to find a solution to the issue.
- Group members are expected to deliver assigned tasks on time. If a group member consistently fails to deliver on time with no valid excuse, the member is given a warning by the other group members. If multiple warnings are given, the supervisor will be contacted and hopefully find a solution to the problem. If the issue continues, with consent from the supervisor and the rest of the group, the group member can be expelled from the group project.
- Group members are expected to to attend at least 80% of the meetings
- If issues occur that result in diverges from the planned project related activity, each group member is expected to notify the rest of the group as soon as possible in order to reschedule.
- Financial expenses not covered by NTNU or Moelven shall be divided equally among the group members.
- Standards and good practices for quality- code and report writing is expected.
- At a joint work session, focus must be pointed towards project related matter. Time shall not be spent on personal interests that are not project related, this will have to wait until a break.
- Work days: Monday through Friday.
- Work hours: 8:15 am through 4:00 pm, with regular 15 min breaks and 30 min lunch break at 12:00 PM. Each group member is responsible for tracking their own time using Toggle **Toggle**.
- A plan for the day will be made at the start of each work session in order to gain an understanding for the agenda of the day

- All commits to the repository should be completed working and tested code, including documentation such as comments.

E Risk Analysis

				Low: 1-3	Medium:4-6	High: 7-9	
				Risk			
Nr	Sub-task	Danger/cause	Possible consequence	Probability	Consequence	Riskindex	Risk-reducing
1		Functionality surpasses knowledge and time	Spends too much time on something that causes other items not to be completed. Thus, we may not reach the goal of the end product.	6	3	4,5	Plan in advance and follow scrum and kanban board.
2		Implementation of code not completed by the deadline	Prioritize a minimal solution.	4	9	6,5	Ask Moelven's IT-department for aid
3		Loss of data	Can not continue working with the work as data is lost.	2	9	5,5	Daily backups by each group member.
4		Develops an application that is too complex	Client ends up not using it.	5	6	5,5	Begin with a minimal solution and then further develop it. Also
5	Product	Cloud-based platforms temporarily	Work and testing are put on hold.	1	8	4,5	Estabilish local tools to work on the
6		Absence due to sickn	More to do for the others, as the group becomes one person less.	7	6	6,5	Good hygine, avoid
7	Project	Disagreements in the group regarding the application's design	Time is lost due to disagreement. Disagreement can cause bad mood inside the group.	3	3	3	Be open minded and understandable to others
8		Report not completed by deadline	Poor evaluation of the report.	1	7	4	Set aside enough time on different parts of the report.
9		The application becomes too complicated for the user	The application is not used.	3	6	4,5	Keep it simple and minimize complexity.
10		Client platforms are down	Work and testing are put on hold.	3	8	5,5	Estabilish local tools to work on the application.
11		Corrupt data	Lose all our data.	1	6	3,5	Multiple backups.
12	Company	Sensors give significant incorrect measurements	This will lead to wrong data on when the timber is finished drying. Which can cause money loss, loss of customer reputdation or physical damage.	4	9	6,5	Correct callibration.

F Meeting summary

Meeting with Supervisor

13.01.2022

Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

First meeting with supervisor Tom Røise. He had some input on the time usage and time tracking:

- Not a requirement to work 6 our every day, just a pointer
- Do not track when we are not working, not justify being off work
- Toggl Track: around five categories
- We do not have to track every minute, rough overview

Talked about how to write the project plan and academic writing. Tom also gave us the following input:

- Readable format and size, no requirements
- Chose the best reference style (Vancouver is recommended)
- Do not write “I” or “We”, but rather “The group” or “The groups evaluation”
- Result goals: Goals for what we are going to deliver
- Effect goals: Goals for improvements at Moelven
- Learning goals: Goals for what we want to learn
- Talking about Moelven and if they have secrets regarding the thesis, sensitive information.
- Consequence and probability analysis: Uncertainties that can occur and affect the project and the time usage
- Analyze different risks, what consequences may they have. Elaborate the consequences concerning our project uniquely, not to general.
- What uncertainties at Moelven can affect our project

Meeting with Professor of Physics

14.01.2022

Location: Teams

Attending:

Group Members	Participants
Karin Emilie Pettersen	Are Strandlie
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

Had a meeting with physics lecturer Are Stradlie to clarify the project description and understand what to expect. Are there any physics that needs to be implemented in our solution, Thermodynamics?

Are had some input concerning the solution:

- Work in your field, the drying process and thermodynamic principles are tough, find a level that suits you
- Thermodynamics is outside our field
- Use the data on energy and moisture to find connections
- Plot as data and make estimates (XY coordinates)
- Focus should be creating good visualizations
- Few measurements during the process to find a clear connection between energy and moist

Meeting with Moelven

18.01.2022
Location: Teams

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Stefan Djupvik
Mads Greni Arnesen	

Introduction, who are we, who are they. Arranging a weekly meeting at Moelven. We learned a lot from the meeting (drying process, understanding data), and we agreed upon meetings in Moelven on Tuesday every week. They offered to have a meeting where they teach us the basics behind drying of trees (some thermodynamics and drying theory).

They clarified the project description:

- No sensitive data, wait and see if there is some sensitive data that is needed
- Allowed to publish our thesis

- Find connections between moisture and energy consumption
- Create estimates, visual
- Estimated time left
- Reduce over-drying and energy consumption
- Simple but robust functionality

Moelven also clarified the data we had been given:

- Fløde = Flow (speed of the hot water flowing trough the pipes)
- ELM = Electricity meter
- VMM = Thermal energy
- Diff = Difference between temperature inn and out of the kiln
- Kwh/m3, thermal energy consumption connected to m3
- ValueId (sensor)
- VarientValue – Sensor data/measurement
- L1/L2/L3 – The different fans

Meeting with Supervisor

19.01.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

Status update with our supervisor. More feedback and input on the project plan:

- Scope: Description, what must be done
- Field of study: Moelven is drying timber, what challenges is there, who are the users of the application. Working with the drying process
- Background: Why they contact NTNU for thesis, who are Moelven?

Meeting with Moelven

25.01.2022
Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Stefan Djupvik
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

First time meeting client face to face. The group got a tour at Moelven and met allot of their employees. Discussed the solution bit more. We got a small introduction to their platforms (Google Cloud and Azure).

Meeting with Supervisor

27.01.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

Status update and more feedback and input on project plan:

- Placement of “List of figures”
- Attach group contract to project plan
- For the thesis: Show previous GUI and final GUI
- Development routines: use repo, mechanism for merging and branching
- Automate tests, how it is to be done

Meeting with Moelven

01.02.2022
Location: Moelven head office (Moelv)/Teams

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Lisa Nilsson
Mads Greni Arnesen	

Drying Course

We got a course about the processes at the sawmills at Moelven. Further we got an introduction to how the drying process works, the importance and benefits of drying, and factors that affect the drying time.

Key factors that affect the drying process:

- Dimensions – Height multiplied by with
- Tree species – Type of wood
- Saw set – The placement and dividing of the lumber center block
- Desired final moisture – The final moisture level the operators wants to achieve

Some important points we learned:

- Target group: 50+ years
- Heat, circulation and humidity control are needed to dry
- Quality bent planks and cracks on drying go wrong. A major cause of wood damage is improper drying.
- Water weighs a lot, heavy for transport
- The heating phase is different, the heating has nothing to do with drying, since it varies greatly. Can create a system that starts only when the temperature hits 80 degrees and the drying begins.
- Strength properties are much better in dried wood. Withstands loads without water.
- Most importantly, what was the moisture, get an indication of when moisture must be measured. What is the measurement on?
- Intervals are better to use on how much of the energy is assumed used

Meeting with Supervisor

03.02.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

We received feedback on the project plan and were informed of what changes need to be made to the report.

Improvements on the plan:

- Work a little more with the goals of the project
- The task description must be completed more
- Write more about the combination of Kanban and Scrum
- Find a x-factor for the application
- More specifically about the drying process

Ideas for the development of the application:

- Create solutions that are open to adjustments, no hardcoding.
- Different estimates based on historic data and a formula
- Work with different estimation models and visualization models, find the best estimate.

Meeting with Moelven

08.02.2022
Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Leif Torbjørn Næsvold
Mads Greni Arnesen	

(Prototypes)

Presented the prototypes to Moelven and got feedback on them. We came up with some possible solutions after discussing the feedback we got. The GUI must be simpler, and do not need to have that much information. Made an Overleaf document for the report, searched for a good report template.

Target group:

- Too much information (Fan direction not necessary)
- Keep it simple

Energy data is what they have, the rest is manual checking. Most of their stored data is from Valåasen and Årjäng. Create a model that gets better the more data it gets.

What the application needs:

- Tree species
- Dimensions
- Saw sett (2x, 3x, 4x inner and outer)
- Desired moisture level
- Graphs that retrieve previous data.
- Circular progress bar with countdown from 18%
- Use statistics to calculated which intervall one lies in
- Do not relate to season

Data we need from the database:

- Kwh
- Finish moisture
- Width and height
- Tree species
- Saw sett

Explanation of important variables:
124 – Valåsen - Has the best stored data
174 – Årjäng
Int - Integration
Rs9060 - Measure on plank before drying, thickness width, etc.
6 - Gran
7 – Furu
Int_gs_ds – Data both before and after drying.

Meeting with Supervisor

10.02.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

Explained the feedback we got from Moelven on the prototypes to Tom Røise. Discussed with supervisor how we could implement the statistic to the application. We came up with a suggestion to contact a group that could help a little extra with the statistics part of the task. To see if they had any input or ideas on how we could implement statistics in our application.

Meeting with Moelven

15.02.2022
Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Lisa Nilsson
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

In the meeting with Lisa Nilsson, we got a better understanding of all the data in the database. She explained the different variables and showed and gave an small introduction to what they were and what the are used for. This gave us the possibility to ask more specifically about the different data. We also presented the newest prototype to Moelven with the changes from the suggestions last time.

Meeting with Supervisor

17.02.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Mads Greni Arnesen	

We gave the supervisor a summary of what we have done since last week. Among other things, we have gained access to more data in the database of the Moelven, connected the database to IntelliJ. In addition, we talked about that we had set up a meeting with the statistics group and that we would meet them over teams later that day. Thus, we discussed a bit with the supervisor different questions we had prepared for them and if he had any suggestions for something more, we should address them.

Meeting with Statistics Group

17.02.2022

Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Janne Cathrin Hetle Aspheim
Eilert Mikal Hananger Tunheim	Ole-Magnus Høiback
Mads Greni Arnesen	Hilde Hefte Haug

In this meeting, we told the statistics group about what our thesis is about. Then we talked about how we had intended to solve it, and ideas and wishes about how we intend to implement this. We talked about which topics in statistics we think may be relevant such as confidence interval and regression. We also ask them about what kind of immediate thoughts they had after hearing about our thesis, and what topics in statistics they think might be relevant, so that we could look a little more at these. Their first thought was regression. We agreed to arrange a new meeting so that they could have some time to look at the task and produce some ideas and suggestions.

Meeting with Moelven

22.02.2022

Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Stefan Djupvik
Eilert Mikal Hananger Tunheim	Ylva Kleiven
Mads Greni Arnesen	

At the meeting we told them that we had been in contact with a statistics group and got some ideas from them about the task. Consulted with them if it was possible to send the statistics group a csv-file of the data in the database. We got approval from them to send them a csv-file. We also gave them an update on how the development of the application is going.

Meeting with Moelven

01.03.2022
Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Leif Torbjørn Næsvold
Eilert Mikal Hananger Tunheim	Stefan Djupvik
Mads Greni Arnesen	

In the meeting with Moelven, we understood that we still had some ambiguities regarding the various variables in the database. After a while we found out that there had been some misunderstandings regarding the various drying chambers and after this clarification things suddenly made more sense.

Meeting with Supervisor

03.03.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

We gave the supervisor an update on what we have done on the application since the last meeting. We explained a bit about the problems we have had regarding understanding the data in the database and the misunderstandings that have occurred. Furthermore, we talked about that we had sent out a csv-file with approval from Moelven to the statistics group, so that they could get a little better insight into the data we are working with.

Meeting with Statistics Group

03.03.2022
Location: Teams

Attending:

Group Members	Participants
Karin Emilie Pettersen	Janne Cathrin Hetle Aspheim
Mads Greni Arnesen	Ole-Magnus Høiback
	Hilde Hefte Haug

In a meeting with the statistics group, we discussed the various data they had received from us, talked a bit about what they thought was the smartest thing. Their idea was to use our various input parameters as explanatory variables in regression analysis. We agreed on meeting again at 24.03 since we needed a few weeks to work just a little for ourselves.

Meeting with Moelven

08.03.2022
Location: Moelven head office (Moelv)

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Leif Torbjørn Næsvold
Mads Greni Arnesen	Lisa Nilsson

In the meeting, we talked about a trip to the Årjäng to get a better understanding of how the drying process works. We talked to them about the time of the trip, in addition to which we gave a small summary of what we have been working on since the last time and whether they had any input. We also talked about having a mini protype from the to try out at the saw mill and the opportunity to talk with the user of the application to get their feedback.

Meeting with Moelven

16.03.2022
Location: Årjäng Sawmill

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Lisa Nilsson
Mads Greni Arnesen	

Trip to Årjäng

We went on a trip to Årjäng in Sweden with Moelven. We got a tour of the sawmill and got a better understanding of what the data in the database corresponds to at the sawmill. We also got a better understanding of the drying process, and how today's solution works. We showed Moelven the prototype and we got a good response. The only problem we need to figure out is how to deal with the data correctly.

Meeting with Moelven

24.03.2022
Location: Moelven head office

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Lisa Nilsson
Mads Greni Arnesen	Leif Torbjørn Næsvold

At the meeting with Moelven, we figured out that we need to redo how we reset the Kwh in the code, now that we have gotten a better understanding of what the data corresponds to. We understood better the database and how to handle the data processing. We talked about finding a few cycles with corresponding energy data. We would retrieve 63 drying periods with in- and out dates, but the database could only find kwh data for 11 of the 63 drying periods. We tried both at Valåsen(124) and Årjäng(174), but at Årjäng it was even worse with only 14 drying periods and only 2 of these had kwh data. We need to figure out a solution for the small amount of data we receive.

Meeting with Supervisor

24.03.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

In a meeting with the supervisor, we gave an update on what had been done since the last time. We talked about the visit in the year and what we had learned from it. We told about the response we had received in the application zone. showed guides what the application zone looks like now. Furthermore, we talked about how we intend to implement the confidence interval and the regression. In addition, we agreed to submit a draft of the report before Easter to the supervisor.

Meeting with Moelven

28.03.2022
Location: Moelven head office

Attending:

Group Members	Participants
Karin Emilie Pettersen	Ylva Kleiven
Eilert Mikal Hananger Tunheim	Lisa Nilsson
Mads Greni Arnesen	Leif Torbjørn Næsvold

We had many meetings with different people on the Moelven since we had found some downtime in the database, approximately six months. These were something they had not been aware of, so they had to figure the reason for the down time of their data. In addition, we had to try to produce a solution to what we were going to do. As this affects our solution since it gets less data to work with. One suggestion was to use data from a third sawmill so that the application received more data. We also talked to moelven if they had any input to a problem we had with the threads.

Meeting with Supervisor

21.04.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	

In the meeting, we received feedback on the draft of the report we had submitted to the supervisor. We went through the report together and discussed the various parts. We found that there is still a part of the report that needs to be written. Thus, it begins to be important to complete the application so that we can all write on the report. We also discussed the challenges around the fact that we are only two of us in the group due to illness and how the workflow now works in the group

Meeting with Professor

22.04.2022
Location: Teams

Attending:

Group Members	Participants
Eilert Mikal Hananger Tunheim	Ivar Farup

Eilert had a meeting with Ivar since we could not solve the problem with the threads together with Moelven. We thus spoke to the supervisor and were advised to consult with Ivar Farup regarding the problem. Ivar Farup also helped with a problem with had regarding linear regression. Turns out the array format for the linear regression was wrong. The thread could not update directly to the GUI thread, but the GUI thread had an observable list that was always listing.

Meeting with Moelven

26.04.2022
Location: Moelven head office

Attending:

Group Members	Participants
Karin Emilie Pettersen	Leif Torbjørn Næsvold
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

At the meeting with Moelven, we gave an update on what we had done since the last time. We showed them the solution to the thread problem, and we further discussed how to do it with non-linear regression in the application. In addition, we talked about the solution to their downtime problem.

Meeting with Supervisor

27.04.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

In a meeting with Tom, we showed and discussed different models we had for our application. These were:

- Class and entity model
- Backend architecture model
- Use case model, with the following use cases
- Tools visualization
- Sketches

We also had a run-through for our supervisor of our application.

Meeting with Professor

27.04.2022

Location: NTNU Gjøvik

Attending:

Group Members	Participants
Eilert Mikal Hananger Tunheim	Ivar Farup

Had two meetings with Ivar to figure out the non-linear regression. He suggested using the linear regression allready implemented and using the intercept and slope to fit a logisitc regression model. Takes the form of an “S” shaped graph which is close to what we have.

Meeting with Supervisor

04.05.2022

Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Eilert Mikal Hananger Tunheim	
Mads Greni Arnesen	

In the meeting with the supervisor, we talked about how important it was to reflect more on the choice of programming language. What advantages and disadvantages did they have,

what did we consider, etc. We also talked about how important it is to get context for the whole report and constantly show back to things like requirements etc.

Meeting with Supervisor

12.05.2022
Location: NTNU Gjøvik

Attending:

Group Members	Participants
Karin Emilie Pettersen	Tom Røise
Mads Greni Arnesen	

At the last meeting we had a quick review of the most important elements to keep in mind now towards the end. We went through a to-do list that has been shared with us and talked a little more about some of our models.

G Group Log

Date	Plan of the day	Karin	Eilert	Mads
10.01.22	Setting up various environments, timesheet document, meeting report document, set up overleaf and Git repo.	Planned meetings, setting up meeting report	Made overleaf document, and Git repo	Made timesheet document, meeting report and overleaf document
11.01.22	Setting up Toggl Track, look at previous thesis for inspiration, start at the project plan. Make project and group contract. Creating formal rules.	Reading other bachelors for inspiration. Start working on project plan. Making Group contract	Setting up Toggl track, reading trough previous bachelor thesis. Making Group contract	Setting up OneDrive, Toggl Track, looking at previous bachelor Thesis. Started project plan. Making Group contract
12.01.22	Reading previous exams, working on project plan, group contract and creating individual logs.	Finish reading trough previous bachelors for inspiration. Start working on project plan	Finish reading trough previous bachelors for inspiration. Start working on project plan	Finish reading trough previous bachelors for inspiration. Start working on project plan
13.01.22	Work on project plan: Introduction, Goals and Framework and Extent of project Meeting with Tom	Project plan: writing introduction and background	Continue writing on project plan	Continue writing on project plan. Finishing and assigning group contract.
14.01.22	Working on project plan. Meeting with Are Strandlie	Finished introduction and background. Reading about physics. Started on scope part of project plan.	Writing goal section in project plan. Scanning contract document	Writing on goals in project plan. Prepare for meeting with Are
17.01.22	Project plan: Group Rules, Scope, Organization of project	Writing project plan- scope and task boundaries. Preparing for meeting with Moelven.	Project plan: creating group rules, defining questions for Moelven	Project plan: Organization chart, roles and responsibilities. Reading trough and correcting
18.01.22	Project plan: Scope, Subject Field, Organization of project Meeting with Moelven.	Project plan: Subject field, task boundaries and task description. Reading about Kanban and Scrum	Writing project plan, understanding data after meeting	Project plan: Organization of project, development model
19.01.22	Decide development model and tool, finish working with scope. Meeting with supervisor.	Project plan: Completing Subject field, task boundaries and task description.	Project Plan: Development model	Project Plan: Finish working on scope. Discussed development model.
24.01.22	Project plan: reading trough and rearranging	Documentation, source code and tools	Rearranging and cleaning project plan.	Correcting Project plan
25.01.22	Project plan: Documentation, tools, mythology Meeting with Moelven.	Tour at Moelven Documentation in project plan	Tour at Moelven Finished development mythology in project plan.	Tour at Moelven Making tools chart, writing about tools
26.01.22	Working on Project plan	Project plan: Production flow, risk management, tools	Creating a GUI wireframe and Gantt Diagram	
27.01.22	Working on Project plan Meeting with supervisor.	Production flow, risk management	Project plan: Writing about prototypes and milestones	
30.01.22	Completing project plan and delivering	Reviewing and delivering project plan	Reviewing and delivering project plan	

01.02.22	Drying course at Moelven	Drying course at Moelven, discussing the information	Drying course at Moelven, discussing the information	Drying course at Moelven, discussing the information
03.02.22	Creating sketches Looking into the DB Meeting with Supervisor	Creating prototype sketched	Creating prototype sketched	Creating prototype sketched
04.02.22	Creating sketches, preparing presentation for Moelven. Looking at feedback from supervisor on Project plan	Creating prototype sketched, looking at feedback from supervisor	Creating prototype sketched, looking at feedback from supervisor	Creating prototype sketched, looking at feedback from supervisor
07.02.22	Finish presentation for Moelven. Creating a survey for the GUI sketch	Finished sketches	Creating survey for prototypes and presentation	Finished prototype sketches and made presentation to Moelven
08.02.22	Presentation of sketches to Moelven. Start the development process.	Presented sketches for Moelven, discussed our solution and got requirements for input parameters	Presented sketches for Moelven, discussed our solution and got requirements for input parameters	Presented sketches for Moelven, discussed our solution and got requirements for input parameters
10.02.22	Redesigning sketches, making a final sketch. Meeting with Tom. Start organizing main report.	Changing prototypes and looking into DB	Redesigning prototype, started main report	Adjusting prototypes, started looking into the DB
11.02.22	Understanding the DB, and how to visualize data	Finishing and reviewing survey. Set up meeting with Statistics group	Looking at the DB with Grafana, and DB connection	Finishing Survey, started looking into DB connection, visualizing using Grafana
14.02.22	Working on GUI Find a way to reset kwh/m3 Finding each drying period.	Looking into the statistics of the project, understanding data	Creating initial IntelliJ project, connecting Git to GitLab	Understanding data, defining a drying cycle
15.02.22	Asking Moelven about data. Setting up IntelliJ, connect to Git. Need statistics to solve our problem.	Getting data explained, looking into different data	Looking at how to connect Google Cloud DB	Getting access to their Google Cloud database
17.02.22	Google Cloud Database connection Meeting with Supervisor.	Trying to connect to cloud DB at Moelven	Creating GUI to display simple data	Looking into connecting Google Cloud with Java
18.02.22	Connecting BigQuery using JSON key-file. Creating early version GUI.	Creating Simple GUI	Creating GUI to display simple data	
22.02.22	Collecting data from BigQuery tables. Creating GUI using JavaFX	Finishing input field in scene builder, gaining csv file for statistic group	Working at GUI, getting contact with BigQuery	Finishing input field in scene builder, connecting to BigQuery
24.02.22	Collecting data by making SQL statements.	Testing SQL statements to collect data	Making SQL statements to define drying periods	Creating SQL statements and testing them in Grafana
28.02.22	Making SQL statement for collecting data. Making sort class for categorization.	Making sort class to categorize data		Creating SQL statements and testing them in Grafana
01.03.22	Understanding of data Meeting with Moelven	Explanation of data by Moelven		Explanation of data by Moelven
03.03.22	Sending data to Statistics group. Working on sort class and statements Meeting with Supervisor.	Received and sent csv file to Statistics group, worked on sorting class	Rewriting GUI code using JavaFX	Creating necessary statement, Git bugs. Sending csv file.

04.03.22	Working on sort class. Formatting data, and developing GUI	Finishing sort class	Continue working with GUI, database connection and data processing	Formatting collected time from google cloud, Millis to Time date format
07.03.22	Structure the main report after new input. Working on GUI and comparing datatypes.	Discussing better work flow, structuring overleaf after input	Try resetting kwh, trouble with the comparing dates	
08.03.22	Start working on the main report. Resetting energy data to plot the right data. Working with datasets and categorizing data.	Writing on main report, introduction	Try resetting kwh, trouble with the comparing dates	Structuring Report Formatting Millis to Time date format
10.03.22	Working on main report: Introduction Working on GUI Comparing dates to reset kwh/m3	Writing on main report: background, project description, subject area	GUI data handling, plotting	Comparing dates, writing on main report, Target audience
11.03.22	Working on main report: Introduction Working on GUI and data processing. Visualization of cycles	Writing on main report: boundaries, project description, subject area	GUI: Sidebar. Data processing: resetting kwh, splitting into cycles.	
16.03.22	Trip to Årjäng to see a sawmill and understand the drying process. Showing prototype to workers	Trip to Årjäng	Trip to Årjäng	
17.03.22	Working on main report: Introduction	Completed introduction part of report		Completing introduction in main report, start writing about requirements: tools
24.03.22	Reading about statistics Meeting with Tom and Moelven. Redo how we reset kwh/m3.	Gaining knowledge about statistics: regression analysis, confidence interval	Handling data processing, collecting drying cycles	Main report: Finishing writing tools and making figure. Start writing Technologies part
25.03.22	Working on main report: Theory - Statistics Technologies	Statistics part of main report, why is it important for us?	Making constant class, and replacing hard coded variables with constants	Main report: Technologies - Google Cloud Platform
28.03.22	Working on main report: Theory - Statistics, technologies Working on GUI	Main report: Theory about confidence interval and regression analysis	GUI, pop up window, input parameters	Main report: Technologies - Grafana and Google Cloud
29.03.22	Working on main report: Theory - Three structure, technologies Working on GUI	Main report theory Moisture in wood, free and bound water	GUI, pop up window, input parameters, connecting data	Main report: Technologies -intro, programming language, Overleaf
30.03.22	Working on main report: Theory - Drying process, technologies Data categorization and GUI	Main report theory Drying process, removal of moisture	Plotting graphs, priority of input parameters, switch between graphs	Main report: Technologies -sensor data and programming language
31.03.22	Working on main report: Theory - Drying process and Drying Kiln, Technologies Development Process	Main report theory Drying process, and drying kilns and schemas	Treads, run GUI simultaneously with data processing	Main report: Technologies -sensor technologies and sensor data.

	Running separate processes			
01.04.22	Working on main report: Theory part, Development process Working on GUI and Treads	Main report theory Drying process, and drying kilns and schemas, sending draft to Ylva	Redoing input windows and drop-down menus. Case sensitivity.	Main report: Development process - Meetings
04.04.22	Working on main report: Finishing Theory part, development process Data processing	Main report: corrected theory part after feedback. Finishing working on drying phase	Connecting both sawmills, might add a third sawmill to expand the data	Main report: Development process - Methodology, Kanban board
05.04.22	Working on main report: Theory, Development process Meeting with Moelven Fixing treads	Main report: Theory: measurements of energy and moisture	Start looking at treads and how to fix them, cannot run parallel processes (GUI and data processing)	Main report: Updating requirements section, time tracking and iterations
06.04.22	Working on main report	Main report: Theory: measurements of energy and moisture		Main report: quality assuring technologies and requirements part.
07.04.22	Working on main report Sending report draft to Supervisor	Reading trough report before sending first draft to Tom		Main report: quality assuring technologies and introduction. Started looking into design part
08.04.22	Main report - Design Data processing/categorizing		Adding chambers for more data, fixing moisture parameter. Converting days to hours on x-axis.	Main report: Design part - start adding sketches an writing about them
09.04.22	GUI plotting cycles		GUI, converting to hours, plotting of cycles	
19.04.22	Implement confidence interval	Discussing how to implement confidence interval and regression. Revisiting the statistics part, more specific writing	Application: Calculating confidence interval	
20.04.22	Implement confidence interval, Regression Library Statistics main report	Main report Writing statistics part, adding formulas and visualizations	Application: Deleting unwanted interval using the confidence interval, finding regression libraries	
21.04.22	Implementing regression analysis	Main report, feedback on drying theory, looking at requirements part	Feedback main report, continue working on regression and treads	
22.04.22	Meeting with Professor Ivar Farup. Fixing regression and treads.	Main report, feedback on drying theory, looking at requirements part	Fixing both treads and linear regression, pulling manual moist measurement	
25.04.22	Tread handling, Nonlinear regression	Main report, fixing drying theory feedback	Collecting manual moisture, shutting down treads with exit button	Main report: Design part, backend architecture and sketches
26.04.22	Meeting at Moelven Nonlinear regression, finding libraries	fixing drying theory feedback	Finding libraries for regression, trying to use phyton	Main report: Design part, necessary

				information, coordinate view
27.04.22	Meeting with Professor Ivar Farup. Implementing nonlinear regression	fixing statistics and technologies feedback	Implementing the logistic Model, Meeting with Ivar	Main report: Design part, backend architecture making figures
28.04.22	Nonlinear regression, GUI Working on main report	fixing technologies feedback and writing on requirements	GUI: CSS file for styling the graph visualizations. Working on non-linear regression	Development: Making Class and entity diagram
29.04.22	Working on main report	Main report: Rewriting the programming language (Java, Phyton) part.		Main report: Inserting Use Case diagram and writing functional requirements
02.05.22	Working on main report	Main report: Fixing the programming language and technologies part.	Application: Time left sidebar, connecting progress bar and live data	Main report: Writing Requirements part, non-functional
03.05.22	Working on main report	Main report: Fixing the programming language and technologies part.	Presenting application, feedback. Suggestions for improvement.	Main report: Rewriting Google Cloud part in technologies,
04.05.22	Working on main report	Fixing error in overleaf, references. Statistics part	Application: Non linear cubic regression of third degree, goin back to logistic	Main report: Introduction, todays solution, adding pictures, writing report structure
05.05.22	Working on main report	Fixing error in overleaf, references. Statistics part	Application: Adding checkboxes for different views. Adding notifications	Main report: Notes on discussion, Linear growth model
06.05.22	Working on main report	Main report: Adjusting and categorizing theory chapter and statistics	Changing key-file, handeling SQL injections	Main report: Adjusting and categorizing theory chapter and statistics
07.05.22	Working on main report	Main report: Drying phases, adjusting importance of drying wood	Reading report, making adjustments, login system, hashing	Main report: Security: key file, libraries. adjusting importance of drying wood
08.05.22	Working on main report	Main report: Logistic growth model,	Adding admin user, add and delete users. Hashing password, inserting into cloud DB	Report: Design - module diagrams, and connection to GUI
09.05.22	Working on main report	Main report: criteria for programming language	Main report: Implementation	Main report: module diagrams, and connection to GUI,
10.05.22	Working on main report Testing	Unit testing, user testing.	Main report: Implementation and statistics part	Main report: Adding points and visualizations to statistics part

11.05.22	Working on main report Testing	Unit testing, user testing. Testing part report	Main report: Implementation Unit testing, user testing	Main report: Security part, SQL injections, Usage of libraries
12.05.22	Working on main report	Main report: Writing security part awareness of security weakness	Main report: Implementation Unit testing, user testing, integration testing	Main report: Security part, SQL injections, Usage of libraries
13.05.22	Working on main report	Main report: Writing authentication, security part, hashing	Main report: Enhancing Implementation part, and started on the testing part	Main report Finishing design part, logos and colors,
14.05.22	Working on main report	Main report: Writing security part, hashing and salting	Main report: Enhancing Implementation part, and writing testing part	
15.05.22	Working on main report GUI, styling	Main report: Writing confidence interval and rejecting faulty data	Writing testing part, changing GUI colors	Main report: Finishing security part, organizing
16.05.22	Working on main report	Main report: Writing discussion part in report	Writing Discussion and conclusion	Main report: Finishing the security part, writing testing and making figure
17.05.22	Working on main report	Writing discussion part in report	Finishing implementation and writing discussion	Main report: Writing discussion part
18.05.22	Finishing main report	Writing discussion and organizing Appendix	Rewriting some parts of implementation and writing conclusion	Organizing Appendix and finishing discussion part
19.05.22	Reading trough, final adjustments	Reading and making changes to final report	Reading final report and making some changes	Reading and adjusting, adding last appendix
20.05.22	Delivering Bachelor Thesis	Delivering	Delivering	Delivering

H Progress Bar Library Licence

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include

works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such

litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall

be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of

your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

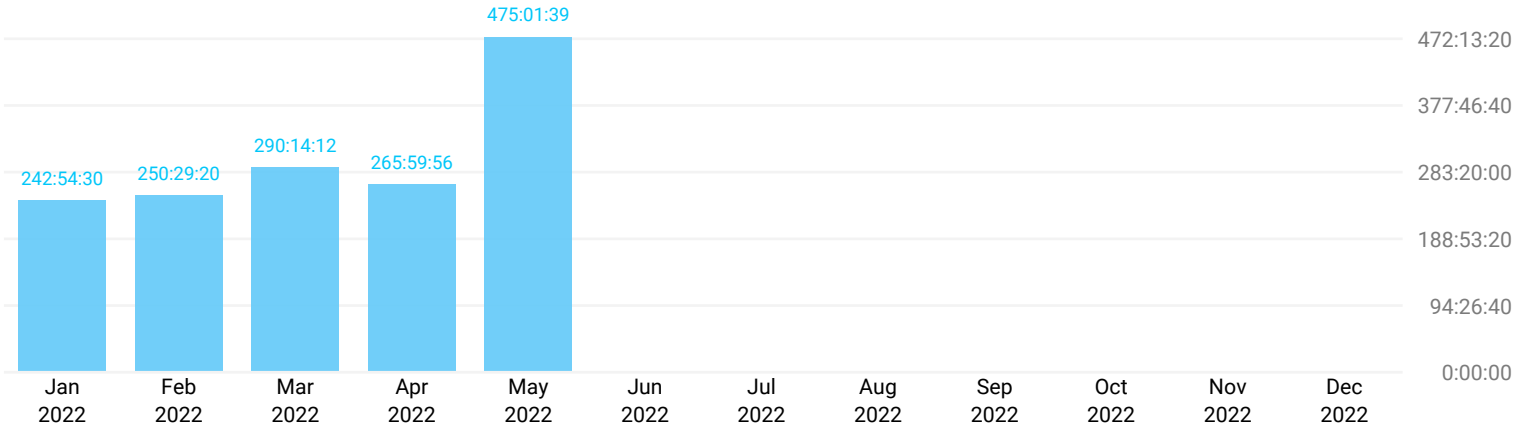
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

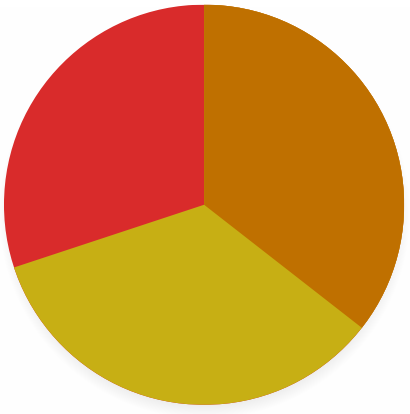
I Toggl Track Summary

Summary Report

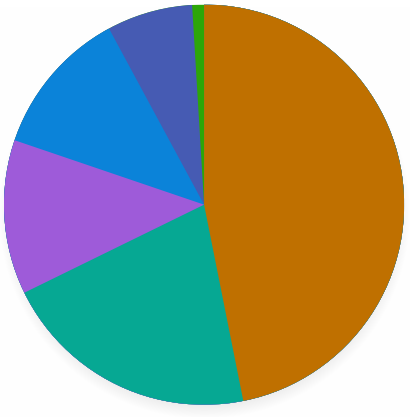
01/01/2022 – 12/31/2022

TOTAL HOURS: 1524:39:37





USER	DURATION
KA Karinep	540:45:56
EM Emtunhei	525:39:22
MA Mads Arnesen	458:14:19



TIME ENTRY	DURATION
Report	716:04:58
Coding	318:58:24
Project plan	191:16:34
Development	181:38:12
Meeting	104:13:33
Other	12:27:56

USER - TIME ENTRY	DURATION
EM Emtunhei	525:39:22
Coding	250:15:38
Development	49:11:04
Meeting	45:21:14
Other	4:10:47
Project plan	74:30:35
Report	102:10:04
KA Karinep	540:45:56
Coding	50:42:46
Development	45:49:55
Meeting	34:42:41
Other	4:09:42
Project plan	73:17:33
Report	332:03:19
MA Mads Arnesen	458:14:19
Coding	18:00:00
Development	86:37:13

USER - TIME ENTRY	DURATION
Meeting	24:09:38
Other	4:07:27
Project plan	43:28:26
Report	281:51:35