STAS – Skidding and Transit Analysis Software

Bachelor thesis 2023

Adrian Chan Bjørnson Ruud, Tormod Fostad Kleiv, Marte Marheim and Liv Marte Olsen.

CALON TechnipFMC

SKIDDING AND TRANSIT ANALYSIS SOFTWARE

BACHELOR THESIS
UNIVERSITY OF SOUTH-EASTH NORWAY

Abstract

For offshore operations, heavy subsea equipment needs to be transported from one location to another by ship. This type of operation is called Skidding and Transit and is used for subsea equipment that is too large or too heavy to be lifted by crane.

Skidding is the operational phase of the activity and is when the equipment needs to be transported from one location to another on a vessel. Transit is when the equipment is not being skidded or operated subsea and is parked in an appropriate location on the vessel.

For skidding and transit, there are several factors that need to be taken into consideration. This includes collection of data of how the vessel, whether it is a ship or an oil rig, moves in correlation to the waves at present location. Mathematical modelling is used to see what forces are working on the equipment, and if the timeframe for operations is inside or outside the weather window, in terms of safety.

For skidding and transit analysis, one either must do these types of calculations manually or use software tools that are expensive and requires training to use. The purpose of this project is to create an application to eliminate the use of commercial software and optimize the presentation of driving factors for the capacity of the equipment.

This application provides a baseline for this type of analysis in the offshore industry. For further development of this software, one needs to explore the possibilities for implementation of analysis based on an irregular wave pattern.

Acknowledgements

Alon would like to express their appreciation and gratitude towards everyone who have contributed to this project.

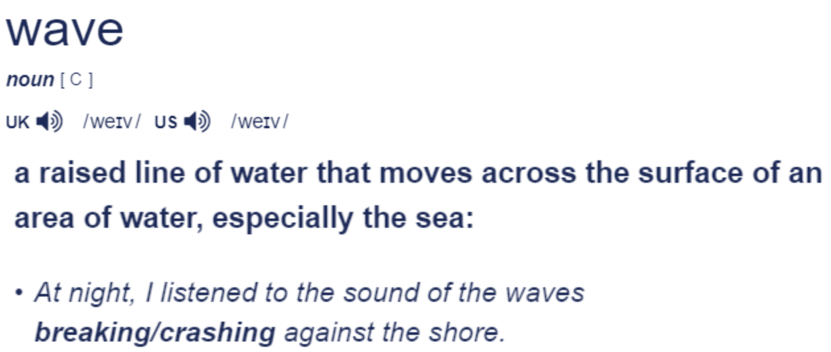
First of all, we would like to thank our external advisors, Mathias Hansen and Halvor Snersrud Gustad, for their support, encouragement, constructive criticism, and challenges they provided throughout the project.

Thank you to Per Thomas Moe, for being our first contact within the firm, TechnipFMC, as one of the task providers.

Additionally, we would also like to thank our internal adviser, Sigmund Gudvangen, for his advice, support, and feedback throughout the project.

Last, but not least, we would like to express our gratitude to the University of South-Eastern Norway for letting us utilize their facilities, components, and equipment to help visualise our project.

Alon – The Tagalog word for wave



wave

noun [C]

UK 🔊 /weɪv/ US 🔊 /weɪv/

a raised line of water that moves across the surface of an area of water, especially the sea:

- *At night, I listened to the sound of the waves* ***breaking/crashing*** *against the shore.*

Table of contents

List of figures 10

List of tables........ 11

1. Introduction 12

1.1 Skidding and transit analysis project 12

1.2 Current situation in the offshore industry 12

1.3 TechnipFMC 12

1.3.1 The department of Structural Analysis Engineering 12

2. Project management........ 13

2.1 Project group........ 13

2.1.1 Group establishment and members. 14

2.2 Roles and Responsibilities 14

2.3 Software tools 16

2.3.1 Tools for documentation and project management........ 16

2.3.2 Software developing tools. 18

2.4 Project contracts. 18

2.4.1 Group contract 18

2.4.2 Work contract between the firm, university, and group 18

2.4.3 Guidance agreement........ 18

2.5 Working hours........ 18

3. Methodology........ 19

3.1 SCRUM 19

3.2 Weekly and daily SCRUM 20

3.2.1 Project planning 20

3.3 Product Backlog 21

3.4 Sprints 21

3.4.1 Sprint planning 21

3.4.2 Sprint reviews 22

3.4.3 Sprint retrospectives 22

3.4.4 Sprint reviews 22

3.5 Stakeholder Analysis 23

3.5.1 Identification of Stakeholders. 23

3.6 User Stories 24

3.Requirements and test........ 25

3.1 Development of project requirements........25
3.2 Requirement iterations........26
3.3 Test........26
4. Risk........28
4.1 Risk management........28
4.2 Identification of project risks........28
4.3 Risk reviews........29
5. Mathematical modelling........30
5.1 The project’s mathematical concepts........30
5.2 Response Amplitude Operator........30
5.2.1 Euler angles........31
5.2.2 Quaternions........32
5.3 Defining all the different symbols and variables........32
5.4 Defining position, velocity and acceleration........33
5.4.1 Vessel position........34
5.4.2 Vessel velocity........35
5.4.3 Vessel acceleration........35
5.4.4 Rotation matrix........36
5.4.5 Point of interest position........37
5.4.6 Point of interest velocity........37
5.4.7 Point of interest acceleration........38
5.4.8 Local acceleration........38
5.5 Bending moment........39
5.6 Centre of Gravity (CoG)........42
5.7 Utilization........44
5.7.1 Pressure capacity calculation........45
6. Software architecture........47
6.1 Implementation overview........49
6.2 Class based software structure........49
6.3 Independent code to class conversion........50
6.4 GUI framework........50
6.5 GUI layout........51
6.6 Database........52
6.6.1 ORM: 1. Iteration........54
6.6.2 ORM: 2. Iteration........55
6.6.3 Database Normalization........55

6.7 Interactive 3D Visualization of Stack 62
6.7.1 Components 64
6.7.2 Connectors 64
6.7.3 Centre of Gravity 65
6.7.3 Filters 65
6.7.4 Libraries 66
6.8 Analysis report in LaTex format 76
7 Proposed further work 77
7.1 Implementation of RAO data 77
7.2 3D graphics 77
7.3 Irregular waves 77
Conclusion 78
References 80
Appendix A: Working hours 84
I. Total working hours for the group 84
II. Work hours for Liv Marte. 85
III. Working hours for Marte. 86
IV. Working hours for Adrian. 87
V. Working hours for Tormod. 88
Appendix B: Sprint reviews 89
I: Sprint review for sprint 1. 89
II: Sprint review for sprint 2 90
III: Sprint review for sprint 3 91
IV: Sprint review for sprint 4 92
V: Sprint review for sprint 5 93
VI: Sprint review for sprint 6 94
VII: Sprint Review for sprint 7 95
VIII: Sprint review for sprint 8. 96
Appendix C: Stakeholder Analysis Diagram 97
Appendix D: User stories 98
Appendix E: Requirements 100
I: Calculations 100
II: Documentation 102
I. 102
III: Database 103
II. 103

IV: Visuals 105
III. 105
V: Safety 108
IV. 108
Appendix F: Test 109
I: Calculations 109
II: Documentation 112
III: Database 114
IV: Visuals 117
V: Safety 120
Appendix G: Risk 121
I. Risk scoring system 121
II. Risk scoring system for probability and severity 122
III: Risk review criteria 123
IV: Risk reviews 132
Appendix I: Tests 139
I: Testing 139
Appendix J: Sequence diagrams 153
Appendix K: ORM 1. Iteration 160
Appendix L: ORM 2. Iteration 161
Appendix M: Database Normalization 162

List of figures

Figure 1: Office logo. 16

Figure 2: Microsoft Teams logo. 16

Figure 3: Discord logo. 16

Figure 4: Jira software logo. 17

Figure 5: Notion logo. 17

Figure 6: Microsoft Visio logo. 17

Figure 7: Draw.io logo. 17

Figure 8: Visual Studio Code. 18

Figure 9: Overleaf logo 18

Figure 10: Scrum process. 19

Figure 11: Graphic depiction of the 6DOF 31

Figure 12: Acceleration output. 39

Figure 13: Bending moment. 41

Figure 14: Bending moment output 42

Figure 15: CoG visualization 44

Figure 16:Use case diagram 47

Figure 17: Sequence diagram - run analysis. 48

Figure 18: MVC diagram for software 48

Figure 1919: Sidebar menu in the GUI. 52

Figure 2020: ORM 1. Iteration 53

Figure 2121: ORM 2. Iteration 55

Figure 2222: Snippet from software 63

Figure 2323: Stack consisting of 5 components with 4 connectors between them. 63

Figure 2424: Showing 3 of the 5 components on the stack. 64

Figure 2525: Showing just the connectors on the stack. 64

Figure 2626: Showing COG for the entire stack. 65

Figure 2727: Showing COG for each component on the stack. 65

Figure 2828: Showing the filters for the 3D figure. 66

List of tables

Table 1: Excerpt from user stories 24
Table 2: Excerpt from calculation requirements. 25
Table 3: Excerpt from calculation tests. 27
Table 4: Excerpt from risk scoring chart. 29
Table 5: All information related to the database. 56
Table 6: Duplicated data 56
Table 7: New table 57
Table 8: Still some data being repeated. 57
Table 9: Vessel table 57
Table 10: Stack table 57
Table 11: Analysis table with a copy of the code from the keys of vessel and stack 57
Table 12: More duplicated data. 58
Table 13: New table moved out of the original 58
Table 14: Still a few duplicates 58
Table 15: Density table 59
Table 16: Component table, with a copy of the key to the density table 59
Table 17: Connector table 59
Table 18: What is left after 1NF. 60
Table 19: Stack configuration table 61
Table 20: Connector configuration table 62

1. Introduction

This chapter provides details about the project, the primary problem, the state of the offshore business, the job provider, and the document structure.

1.1 Skidding and transit analysis project

In the offshore industry, an important aspect when it comes to transportation and relocation of subsea equipment, is skidding and transit operations. These types of operations are impacted by weather data, and how any vessel responds to the wind and sea, and how these forces are transferable to the equipment on deck. The skidding and transit analysis project's main purpose is to create a software which automizes these types of analysis.

1.2 Current situation in the offshore industry

The current situation in the offshore industry for skidding and transit operations, is based on multiple ways to calculate what impacts the equipment aboard any vessel. Either from a company's own *in-house* operations for calculations, or the use of software that is expensive and requires training.

1.3 TechnipFMC

The task provider for this project is the structural analysis department at TechnipFMC (TFMC). TechnipFMC is a technology provider, delivering fully integrated projects, products, and services in the energy industry. TFMC is a publicly traded company with two main business units, subsea and surface technologies.

1.3.1 The department of Structural Analysis Engineering

The task provider for this project is the Structural Analysis Engineering department at TFMC. The departments are located in Lysaker and Kongsberg, with our external advisers based in Lysaker. TFMC provided two engineers as external adviser for this task, Mathias Hansen, and Halvor Snersrud Gustad.

Mathias Hansen is a Specialist Analysis Engineer in the Structural Analysis department. He has been one of our external advisers, and he has been available for questions and advice daily. He is also one of the key stakeholders for this project and has been vital in the development of this project's requirements.

Halvor Snersrud Gustad is an Analysis Engineer in the Structural Analysis department and a PhD – student at NTNU. He has alongside Mathias, been available for questions and advice daily, specifically for the mathematical part of the project, and is one of the key stakeholders in this project.

2. Project management

2.1 Project group.

Liv Marte Olsen
Mechanical Engineer
livmarteo@gmail.com
Project Lead
Budget Lead

Adrian Chan Bjørnson Ruud
Computer Engineer
adrian.ruud@hotmail.com
Product Owner
Technical Lead – Computer Science
Website Architect

Tormod Kleiv
Mechanical Engineer
tfk1805@gmail.com
Scrum Master
Technical Lead – Mechanical

Marte Marheim
Computer Engineer
marte.marheim@gmail.com
Document Lead
Test Lead
Risk Lead

2.1.1 Group establishment and members.

The group formed in the timespan from August to December 2022. The group originally consisted of 5 members, but due to some technicalities, one of the members were unable to participate in this year’s bachelor project.

Computer science and mechanical engineering are the two different fields represented in this multidisciplinary group. With an even gender distribution, an age gap of almost ten years, and a variety of prior education and job experiences, the group believes they possess many essential qualities to make a powerful unit. Which will contribute to seeing the project and issues from different point of views, leading to a solution that is both sustainable in terms of low maintenance and cost-effective software.

2.2 Roles and Responsibilities

Project lead:

The project lead is responsible for high-level management and supervision of the project. This includes not only coordinating the exchange of information between the group and advisers, but also having an overall awareness of the project as a whole in order to manage and oversee the project. The project lead will perform the necessary effort to ensure that the team achieves its end goal rather than making high-level decisions on their own. The project lead will work towards the end result using a systems design and engineering approach in order to anticipate and resolve potential project impediments before they arise.

The team's engagement and motivation are also the project leader's duties, along with maintaining a productive and pleasant work environment.

Product owner:

Scrum has been chosen as the agile project management approach for this project. The product owner is frequently a key stakeholder on a scrum team and sometimes the client. One of the members in our group serves as the project's product owner.

The primary responsibility of the product owner is to ensure that the development team maximizes the benefits of the product they are going to provide. The product owner is also in charge of overseeing the product backlog and ensuring that the development team is aware of it and can understand it. For this project the group will make user stories, which will be presented to TFMC.

The product owner is also responsible for ensuring that the developer team is working on the right task at the right time, which implies that the product owner must prioritize the items in the product backlog based on where the development team is at in the process. Finally, ensure that each task is handled and done to a high standard.

Technical lead – Computer Science:

The primary responsibility of the technical lead for computer science, is to ensure that every aspect that logically fall under the computer science subject area, are handled in an appropriate technical manner. To verify that the technical standards are met, the technical lead is required to perform a variety of duties, such as inspecting code and applications. And in addition, make sure that the technical aspect of the project is fully documented, and that the software architecture is transparent.

Technical lead – Mechanical:

The primary responsibility of the technical lead for mechanical engineering, is to ensure every aspect that falls logically under the mechanical engineering subject area, are handled in the right technical manner. This entails monitoring any prospective designs and calculations made by the developer team and making sure that every area of the project's mechanical part is handled in an appropriate technical manner. To ensure that the technical standards are of the best quality, this requires the technical lead to examine calculations and analyses. The technical lead will also see to it that the project is properly documented from a mechanical standpoint.

Scrum master:

The primary responsibility of the scrum master is to make sure that the development team sticks to the agile framework throughout the duration of the project. The scrum master should instruct the development team and have advanced knowledge of the scrum method's structure. Executing scrum meetings and sprint retrospectives are within the responsibility of the scrum master. The scrum master must also make sure that everyone in the team has a chance to speak their mind.

Document lead:

The document lead is responsible for the project documentation. This entails the responsibility to oversee that the main report's structure is within the frame decided by the group. That all its contents are included, such as all the templates and standards the development team creates. Every group member will develop content for the documentation and contribute to it, but it is the document lead's job to make sure that it is completed within the parameters set by the group.

Test lead:

The test lead's job is to supervise and coordinate all the tests that are run for this project. Testing is an essential component of the project for the team to be able to verify and validate the final product. Since the tests are what provide the framework for the final product, they must not only confirm the project requirements but also have a direct line of traceability back to the requirements. The tests must be validated, and the test lead will oversee that the developer team follows the different criteria established.

Risk lead:

The risk lead's job is to minimize any potential negative impact an event might have on the project, the product, or stakeholders. Under each sprint review, the risk lead must conduct risk-related analysis to ensure this. The goal of the risk analysis is to determine potential risks, their level of severity, and the best way to protect the various project aspects from potential harm. It is the duty of every team member to inform the risk manager of any potential risks they identify while working.

Website architect:

The team's website is created and maintained by the web architect. The web architect is also in charge of regularly updating the website with the project team's work and development.

2.3 Software tools

Listed below are the software tools the group used in terms of project management and task execution.

2.3.1 Tools for documentation and project management

Word and Excel were both used to create the documentation for the group work for this project.

Figure 1: Office logo.

The project is presented through written text and diagrams in Word. Excel has been utilized to calculate and present numerical data as well as to construct tables and diagrams like Gantt charts. Presentations in the project was created using Microsoft Power Point.

Figure 2: Microsoft Teams logo.

For meetings with external advisers and for the days a group member needed to work from home, the group used Microsoft Teams.

Figure 3: Discord logo.

The majority of the group's internal communication has been conducted via Discord. For Alon, the team built its own server, which has served as the primary channel for communication, file sharing, and teamwork.

Figure 4: Jira software logo.

For project management the group used Jira Software from Atlassian. Jira software is an issue tracking product development software, used for agile project management such as Scrum. Jira Software was sponsored by TechnipFMC for the group throughout the project period.

Figure 5: Notion logo.

For personal project management, some of the group members used the software Notion. Notion is a free productivity web application, with tools for task management, to-do lists and more.

Figure 6: Microsoft Visio logo.

Figure 7: Draw.io logo.

For the making of tables and diagrams, the group used two software tools for development, Microsoft Visio and Draw.io. Microsoft Visio has been accessible to the group through the schools IT resources, and Draw.io is a free online diagram software.

2.3.2 Software developing tools.

Figure 8: Visual Studio Code.

The main program for all software work, the group used Visual Studio Code. VS code is a free software for code editing and for building and debugging web applications.

Figure 9: Overleaf logo

Overleaf is a LaTex editing program used for writing, editing, and publishing of scientific documents. Overleaf was used for the creatin of the applications end report.

2.4 Project contracts.

The details of every contract the group signed for this project are included below.

2.4.1 Group contract

In the beginning of the semester, the group developed and signed their own contract regarding this project and how the work would be conducted. The contract clearly stated the rules and guidelines each member would have to follow throughout the project, in terms of working hours, project implementation, work environment, conflict management and general well-being rules.

2.4.2 Work contract between the firm, university, and group

In terms of contracts, the group also needed to sign a contract between the University, the task giving firm and themselves. The contract states what the company's and university's duties are, rules for budgeting, confidentiality and more.

2.4.3 Guidance agreement

The last contract the group signed was a contract between them and their internal advisor. The contract states rules for group composition and work performance, long term leave of absence, conflict management and more.

2.5 Working hours

In the very beginning for this project, the group established a timetable for keeping hours throughout the project. The total and individual working hours can be found in appendix A.

3. Methodology

The project's chosen methodology, the group's implementation and execution of it, the product backlog, sprints, stakeholder analysis, and user stories are all covered in this chapter.

3.1 SCRUM

The chosen model the group used throughout the project period was scrum. Due to our group being small, and scrum is best suited for work teams of a larger scale, the group needed to make some alterations to the working model but kept to the main frame of scrum.

Scrum is an agile development methodology for agile project management teams use to develop, deliver, and maintain complex products. It involves consistently providing customers with value and it serves as a foundation for accomplishing assignments. By utilizing the Scrum framework and incorporating agile ideas into your work and communication, you will soon begin to think more agilely than you did before. [1] [2] [3]

Scrum is utilized by a wide range of teams, but it is most common in the engineering and software development industries. Software is a living, breathing organism, which means that situations, objectives, and requirements all change. Scrum welcomes these changes. With Scrum, a product is developed over the course of several sprint-length iterations as seen in fig. 9. They divide up large, complex assignments into manageable chunks. Projects become easier to manage as a result, and teams can deliver high-quality work more quickly and often while also having greater adaptability. [1] [2] [3]

The open and iterative design of Scrum also aid in resolving many of the persistent issues that frequently arise in waterfall projects. Short iterations provide risk and expense reduction, rapid user feedback, faster time to market, and earlier value recognition. The frequent occurrence of milestones, such as the conclusion of a sprint, gives teams a sense of steady, visual progression. This keeps people motivated and focused, which raises employee satisfaction and engagement levels. [1] [2] [3]

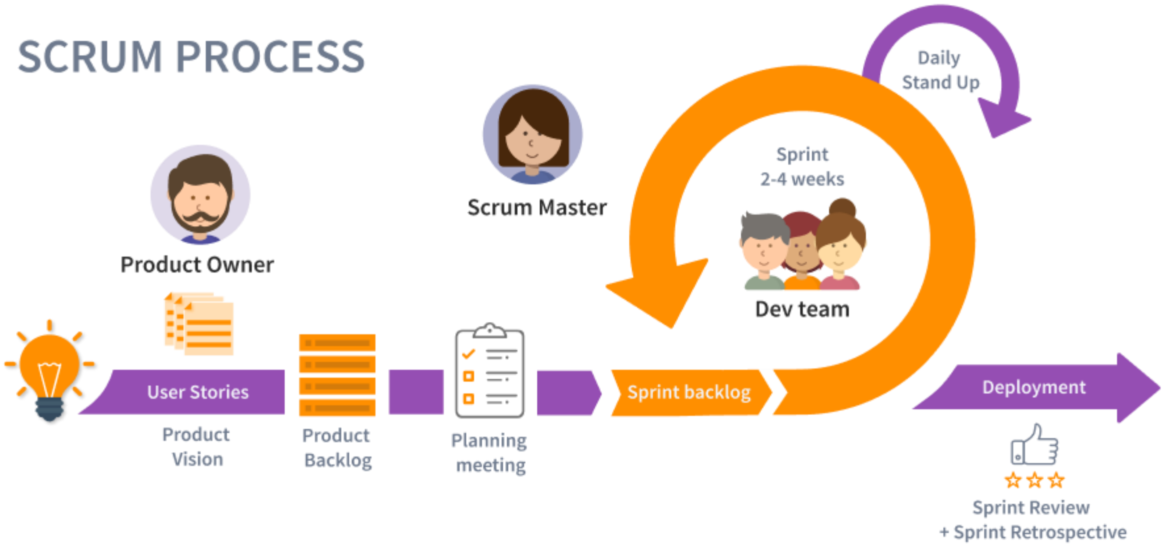


Figure 10: Scrum process.

3.2 Weekly and daily SCRUM

Usually there is a daily stand-up Scrum meeting, where each team member will inform the other members on the status of their work. These meetings are usually extremely short and are a great way for the entire team to have an overview of where all the other members stand. This method of sharing the progression of each task often help eliminating the need to write and read tons of documents or status reports.

These stand-up meetings will focus on revealing each tasks status, what the plan looks like, and discuss any bugs that may have surfaced, and if they need any help. Often the other team members can help you see the solution to problems easier since they can look at the problem with fresh eyes. [2] [3]

Our group, Alon, only consists of four members and two disciplines. We often worked in pairs and sat close by the rest of the team, which means that our communication throughout the project was ongoing almost all the time. For this reason, we chose to have weekly Scrum meetings instead of daily for the time period between 1. January – 31. March. For the first few months of the project, we had three days a week to our bachelor's project. The change to daily scrum meetings, and work 5 days a week, occurred after the Easter break.

Scrum meetings until 31. March:
Show what we have done for the past seven days.
Have we encountered any bugs; do we need help?
What is our plan for the next seven days?

Scrum meetings from 11. April:
Show what we did the previous day.
Have we encountered any bugs; do we need help?
What is our plan for today?

3.2.1 Project planning

Following the first scrum model framework, the team drew out the project in terms of the tasks that required to be completed throughout each sprint. As previously indicated, Jira served as the team's primary tool for project planning. The team used Jira, which TFMC sponsored, and its premium features to plan out the entire project from beginning to end.

Additionally, a set schedule for meetings with internal and external advisers was established. In the beginning, the team and the internal and external advisors had a formal agreement to hold weekly meetings. If there was anything the team was unsure of, the team could simply send an email, call, or text the appropriate advisor with the query in mind.

3.3 Product Backlog

The Product Backlog is an organized list of things required to improve the product. Product Backlog items that can be completed by the Scrum Team in one Sprint are deemed ready for selection at a Sprint Planning session. They usually achieve this level of transparency through refinement procedures. Product backlog refinement is the process of dissecting and further defining items on the product backlog into smaller, more precise ones. Adding information, such as a description, order, and size, is a continuous process since attributes frequently change. [2]

The sizing is the responsibility of the developers who will be performing the work. The client can have an impact on the backlog by assisting the team in understanding and selecting trade-offs [2].

The team's backlog was created and manged using Jira Software for project planning.

3.4 Sprints

In Scrum, the Product Backlog's ideas are implemented during the sprints. A sprint is a set amount of time during which small tasks are managed with the intention of finishing everything scheduled for that specific sprint. Each sprint's duration varies from one team to the next, but it typically lasts between one and four weeks. Longer sprints increase the complexity of what must be done, which makes it hard to determine what is too much for the team. This may result in the sprint goal not being completed, or the goal becoming invalid throughout the process. If the Sprint Goal becomes outdated, the Sprint could be cancelled. The Product Owner is the only person who has the power to terminate the Sprint. [2]

Shorter Sprints can be utilized to generate more learning cycles while keeping costs and risks to a minimum. Each Sprint may be seen as a short project. Sprints aid the team in maintaining an efficient and reliable workflow. All work necessary to complete the goal for that sprint, including sprint planning, daily/weekly scrums, sprint reviews, and sprint retrospectives, happens within the sprint itself. [2]

Our group opted to do sprints that last two weeks. We predicted that many of our tasks would take longer than a week to complete, and we did not want the additional risks that comes with a longer sprint, therefore we decided that two weeks would be suitable for us.

3.4.1 Sprint planning

Sprint planning is a technique that assures improved overall work performance, aids in execution focus, and eliminates surprises. The process should ideally be performed early in the week to avoid disruption of your team's workflow by the weekend. The product owner, scrum master, and team will meet to discuss some critical topics. [2] [4]

Why is this Sprint valuable?

The Product Owner suggests ways to improve the product's worth and usefulness during the current Sprint. The team then works together to create a Sprint Goal that explains to stakeholders why the Sprint is important. Prior to completing Sprint Planning, the Sprint Goal must be determined. [2] [4]

What can be done this Sprint?

The team chooses things out from Product Backlog to add to the current Sprint through consultation with the Product Owner. Choosing the amount that can be accomplished in a Sprint can be difficult. With more experience, we will become more accurate at making these predictions. [2] [4]

How will the chosen work get done?

The team discusses the work necessary to deliver satisfying results for each selected Product Backlog item, and this serves as their Definition of Done. This is typically achieved by dividing tasks from the Product Backlog into shorter work items lasting one day or less. [2] [4]

The answers to these three questions will assist the team achieve its objectives and plan the subsequent two weeks' worth of work. [2] [4]

3.4.2 Sprint reviews

Examining the sprint's results and adjusting for the future are the goals of the sprint review. The Scrum Team discusses progress toward the Product Goal with key stakeholders after presenting the outcomes of their work to them. The Sprint Review is the second-to-last event of the Sprint, and the Product Owner will invite the Scrum Team and important stakeholders to attend them. [5] [6]

The team members will explain which tasks have been marked as "done" and which have not, as well as present their work and respond to any questions. Then the group will highlight the Sprint's positive aspects, its challenges, and how those challenges were overcome. [5] [6]

The Product Owner will then explain the current state of the Product Backlog and the whole team works together to decide what to accomplish next, ensuring that the sprint review is a valuable resource for the next sprint planning. Remember to consider how the market or possible applications of the product may have changed. [5] [6]

3.4.3 Sprint retrospectives

The Sprint Retrospective's goal is to develop strategies for raising quality and efficacy. The Scrum Team evaluated the performance of the previous Sprint in terms of members, interactions, processes, tools, and their definition of doneness. [2] [7]

The team talked about the following in the sprint retrospective:
What was successful about the Sprint?
What might have been done better?
What did we pledge to improve upon throughout the forthcoming Sprint?

The most beneficial adjustments were chosen by the Scrum Team to increase efficiency. The most significant changes were made as quickly as possible, and the Sprint was finished with a retrospective. For a Sprint of one month, it has a timebox of no more than three hours, but the length is generally shorter for shorter Sprints. [2] [7]

3.4.4 Sprint reviews

Throughout the project the team has gone through 9 sprints, where after every ended sprint the team went through a little summary of each sprint. These summaries are called sprint reviews and gives the team an overview of what has been completed and what needs further work. For every review, the biggest question was if the sprint goal was met. There were also subcategories called increments which are considerably smaller tasks that were assigned to one or more members of the team. Initially these increments were assigned a singular sprint but could be moved over to the next spring was the increment not completed.

During sprint reviews the team assessed key result that went well or not so well. Taking those results seriously and learning from them was a big part of the team’s sprint progress. Taking the progress into consideration, another detrimental part of the sprint review was the team members motivation coming into every new sprint. If the motivation was strong among all members, no action would be taken. However, if the motivation among one or more team members was deemed low there would need to be actions taken to attempt to improve the affected members motivation [1] [2]. The complete list of the sprint reviews can be found in appendix B.

3.5 Stakeholder Analysis

An important aspect of project management and execution is stakeholders and the identification of all stakeholders for a project [3].

To ensure that every stakeholder for this project was identified, the team conducted a stakeholder analysis. A stakeholder is: “a person external to the scrum team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the product owner and actively engaged with the scrum team at sprint review” [4] [5].

We quickly realized after completing the analysis that we had several stakeholders and needed to arrange them in a way that was transparent and understandable [6] [7]. To better organize the stakeholders, we listed every stakeholder in a diagram and categorized everything in accordance with their level of involvement in the project and ultimate outcome [3], including whether they were personally involved or would be. We divided our data into primary, secondary, and tertiary categories, and can be found in appendix C.

Primary: for the parties who directly contributed to the creation of the project and finished product. The people who will use the finished product may also be the main stakeholders.
Secondary: for the project's stakeholders, who were either advisors or corporate employees who were in charge of assigning the task.
Tertiary: for interested parties who will in some way be impacted by the finished product or profit from it.

3.5.1 Identification of Stakeholders.

The stakeholders we identified during the analysis was:

The team: The developer team, Alon, who will use the product during the development phase, with the final goal of helping the company simplify their work process and make it so that the company can reduce the use of programs that are expensive and time consuming.

The customers and the key stakeholders: Mathias Hansen and Halvor Snersrud Gustad from TechnipFMC are our key stakeholders. The execution during the project and final product will rely heavily on their feedback and opinions. They are listed as our external advisers in the diagram attached.

Users: The users are the ones who will use the final product to do skidding and transit analysis. The software will be used internally by the TechnipFMC’s analysis department.

Internal adviser: Sigmund Gudvangen was assigned as the group's internal adviser at the start of the project. The internal adviser's suggestions will have a significant impact on the project.

Head of structural analysis department and TechnipFMC:
One of our contacts regarding obtaining the assigned assignment was the department chief, Per Thomas Moe. In addition, he oversees the structural analysis division at TechnipFMC.

Maintainers: The maintainers are the ones who will maintain and upgrade the software.

USN: The University of South - East Norway is the university the group is enrolled in, and where the group’s internal adviser is employed.

Customers of TechnipFMC: For many businesses around the world, the structural analysis department performs skidding and transit analyses. TechnipFMC's clients are classified as a tertiary stakeholder because they provide the data for the analysis and will therefore profit from the finished output.

3.6 User Stories

When entering tasks in the product backlog, the given task is often written as a User Story. A user story will describe a user's experience with a product. The reader will receive a brief narrative, criteria of acceptance, and conditions for when a user story can be considered finished. A user story will inform the reader of the precise demands of the user and the goal of the product [8].

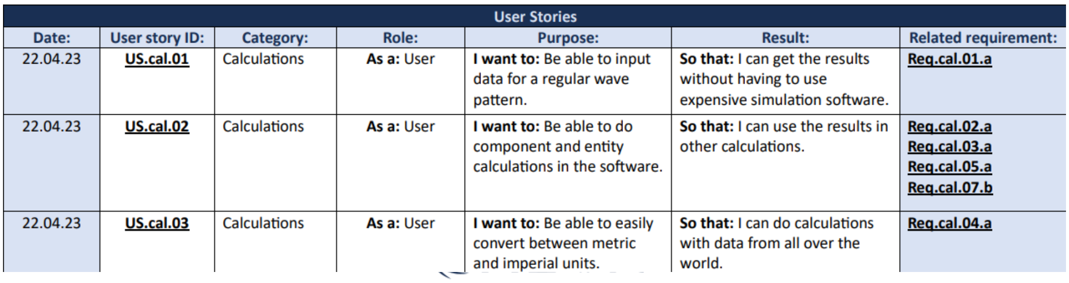
To ensure that the requirements were produced throughout the project with the end user in mind, the team used the stakeholder analysis to create user stories. User stories are crucial to agile development because the team should prioritize meeting the demands of the user first. By creating user stories that provide priority to the stakeholders, the team is putting the end user and significant stakeholders at the centre of all communications that take place during the project. Stakeholders and end users, who are instrumental in the creation of user stories, supply the requirements for the task at hand. The team's adoption of a user-focused methodology will be ensured by the user story, laying the foundation for efficient teamwork and, eventually, a superior product [2] [3].

In the form of user stories, the team opted for a non-technical common language to work from, so that they could comprehend why they are working and what they were trying to accomplish. User stories, which should describe how applications for a certain product would benefit the consumer, must be simple to understand for all team members. User stories must also be traceable back to given requirement and test, and able to define the user, their need, and their purpose [2] [3].

The team used the phrase "As a" to determine whether the user or developer was the intended audience before moving on to the phrase "I want to" to determine the user's requirement/need and finally the phrase "so that" to express the purpose of the user story as seen in table 1 [6] [8].

After completion, the user stories were added to the team's project backlog [4] using the Jira software. This allowed the team to make sure that each user story was worked on at the appropriate time. A user story would be forwarded to the following sprint and tagged as incomplete during the sprint review meeting if it wasn't finished in the allotted time. The complete version for user stories can be found in appendix D.

Table 1: Excerpt from user stories



User Stories						
Date:	**User story ID:**	**Category:**	**Role:**	**Purpose:**	**Result:**	**Related requirement:**
22.04.23	**US.cal.01**	Calculations	**As a:** User	**I want to:** Be able to input data for a regular wave pattern.	**So that:** I can get the results without having to use expensive simulation software.	**Req.cal.01.a**
22.04.23	**US.cal.02**	Calculations	**As a:** User	**I want to:** Be able to do component and entity calculations in the software.	**So that:** I can use the results in other calculations.	**Req.cal.02.a** **Req.cal.03.a** **Req.cal.05.a** **Req.cal.07.b**
22.04.23	**US.cal.03**	Calculations	**As a:** User	**I want to:** Be able to easily convert between metric and imperial units.	**So that:** I can do calculations with data from all over the world.	**Req.cal.04.a**

3.Requirements and test

Information about requirements, tests, and the process the team used to create them can be found in this chapter.

3.1 Development of project requirements

The requirements for this project were developed through collaboration with TFMC. After conduction an initial meeting, where the team and key stakeholders discussed what TFMC wanted as their final product. The team developed requirements, which were presented to the key stakeholders for approval [4] [3].

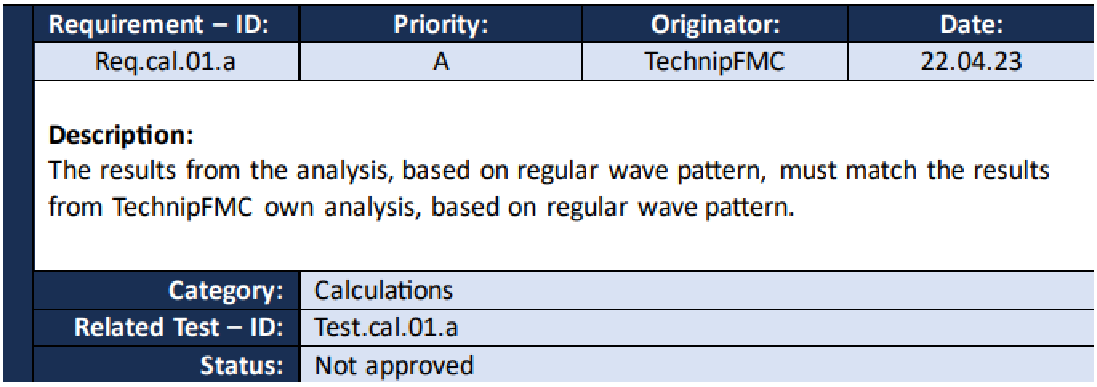
When developing the requirements, the team used the stakeholder's analysis with the vision TFMC had for the final product as a base for development. The team decided that defining the major categories of requirements for the stakeholders and system based on the provided task was the best method to move the project ahead after the stakeholder analysis was completed [9] [10].

The main categories with given ID were:

Mechanical calculations requirements: Req.cal
Document requirements: Req.doc
Visual requirements: Req.vis
Database requirements: Req.db
Safety requirements: Req.safe

With the help of the distinct IDs assigned to each category as seen in table 2, user stories, requirements, and tests could all be linked back to the original category. The requirements the team developed serve as the building blocks for the creation of our software, and each one is accompanied by a related test that confirms and validates its fulfilment. All our requirements are covered by the five categories, and each one has an ID, a priority level, a description, a category, a status, and an ID for a related test. The complete version of the requirements can be found in appendix E.

Table 2: Excerpt from calculation requirements.



Requirement – ID:	Priority:	Originator:	Date:
Req.cal.01.a	A	TechnipFMC	22.04.23
Description: The results from the analysis, based on regular wave pattern, must match the results from TechnipFMC own analysis, based on regular wave pattern.			
Category:	Calculations		
Related Test – ID:	Test.cal.01.a		
Status:	Not approved		

3.2 Requirement iterations

During the project phase, the team needed to conduct several iterations on requirements, user stories, risk, and test, due to change in scope from TFMC and key stakeholders. These iterations were conducted much in the same way as the first iteration, where the team held a meeting with the key stakeholders to discuss what they wanted to alter and to determine whether there were any new requirements for the project and finished product.

The team had to do a total of 4 iterations throughout the project. Were iteration 3 and 4 was the most significant in terms of change to the scope of a larger scale.

For iteration 3, the team was presented with new requirements regarding topside pressure testing. Due to a misunderstanding between the external advisors, the team was presented with terms for requirements, that did not match the criteria for pressure testing in terms of what the team needed to calculate and include in the application. This error was corrected when the team had a meeting with both external advisors, and the mistake was corrected by the remaining external adviser. This ultimately led to the team losing a weeks' worth of work, since the team had to remake requirements, user stories, and tests, including new calculations for topside pressure testing.

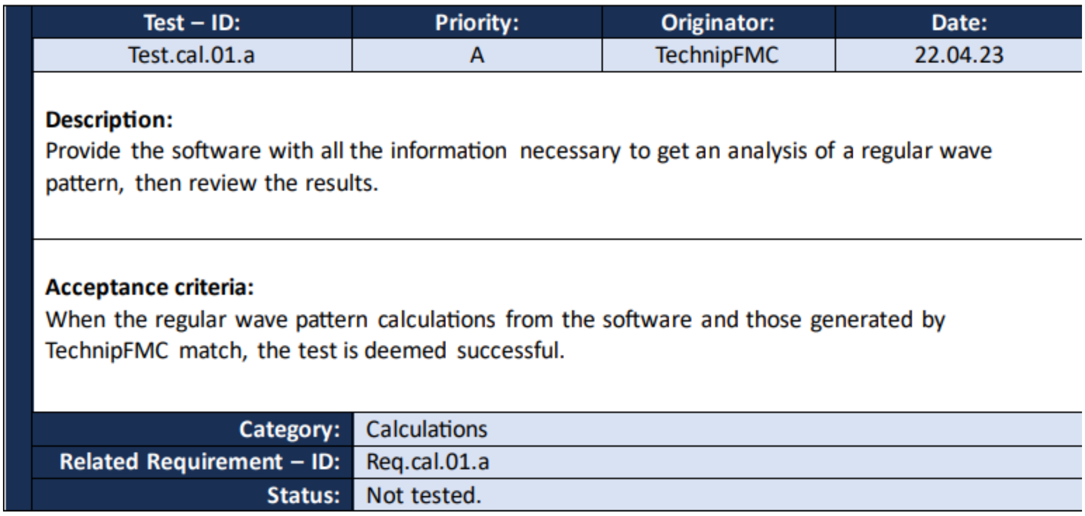
Due to an inaccuracy in the initial calculations provided to the team by TFMC, the team was given new calculations on May 10. The team held an internal meeting to discuss these changes and came to the conclusion that the team needed to concentrate on the last remaining improvements for the software and documentation, due to the closeness to the project's finish date. The mechanical team members will examine the new calculations and create a new, corrected Python code for the calculations and analyses if they are able to, in terms of time and work left to the last presentation.

3.3 Test

The team developed test for each requirement, which were used to validate and make sure that the software was developed in accordance with the specification for the final product.
A requirement must be measurable and testable to verify its accuracy and to be able to identify when the given requirement has been approved [9] [3] [10] [11].

Both the requirements and the test were created in a fairly similar manner as seen in table 3. The team used the requirements, user stories, and stakeholder analysis to determine how to test the requirements and what they required.
Each test was assigned a category, ID, priority, and status to ensure that it could be linked back to the supplied need and user story. The team also included a description of the test and acceptance criteria in order to better inform the reader or tester on how the test should be executed and what criteria was acceptable for completion. Complete version of test can be found in appendix F.

Table 3: Excerpt from calculation tests.



Test – ID:	Priority:	Originator:	Date:
Test.cal.01.a	A	TechnipFMC	22.04.23

Description:
Provide the software with all the information necessary to get an analysis of a regular wave pattern, then review the results.

Acceptance criteria:
When the regular wave pattern calculations from the software and those generated by TechnipFMC match, the test is deemed successful.

Category:	Calculations
Related Requirement – ID:	Req.cal.01.a
Status:	Not tested.

4. Risk

Information on risk and how the team created and managed potential risks during the project is provided in this chapter.

4.1 Risk management

Risk identification enables the early detection of possible problems and the mitigation of their negative consequences on project parameters like schedule, resources, budget, and product features, risk identification and management of prospective hazards is an essential part of agile work management. Using risk management will help the team identify and evaluate each risk before it worsens, which might potentially lead to crisis circumstances if not adequately addressed at an early stage [12] [13] [14] [15].

Risks may have a variety of traits, such as [13]:

- A potential risk that may or may not have an impact in the future.

- The probability that a future event will have an impact must be greater than zero but lower than 100%.

In order to ensure that they could finish their work in the best way possible, the team conducted an assessment of potential risks through risk identification, analysis of severity, implementation of cause of risk, consequence, risk mitigation, and risk avoidance [14] [13] [12] [3].

4.2 Identification of project risks

As the team began the process of identifying each risk, the group gathered to discuss potential dangers and difficulties, as well as the seriousness of each potential risk. The team also had a meeting with their internal adviser to discuss matters and details that were important but easy to overlook. The team quickly decided that using primary categories—the same strategy we used for stakeholder analysis—was the best course of action moving forward [14] [16] [17] [18].

The main categories:
Group risks: Risks affecting the group as a whole or individual members. This covers issues like a group member quitting or the group breaking up. If a group member gets sick or a family member passes away, and the possibility of conflicts within the group and individual personal issues.

Stakeholder risk:
risks affecting the project's stakeholders. This covers issues such as poor communication with stakeholders, and issues with the final product.

Project risk: Risks concerning the project. This covers issues such as difficulty meeting deadlines, change in scope of the project, and problems with hardware and software used in the project.

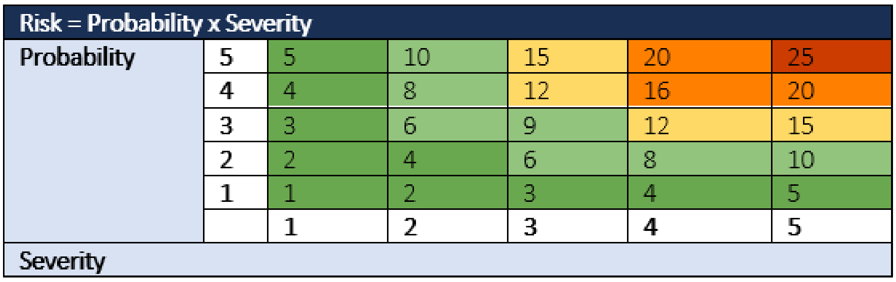
Technical risk: Risks related to the project's technological components. This covers issues such as problems with the mechanical calculations, issues with the software and the creation of the databases, and error in the software code.

The team continued to construct risks in a similar manner to how they had developed requirements and tests.

- ID: To identify each risk and the category to which it belongs.
- Description: an explanation of the risk.
- Cause of risk: criteria for why the risk will occur.
- Probability: The likelihood that the risk will materialize.
- Severity: The risk's level of severity.
- Result: Result of severity and probability combined.
- Consequence: What would happen if the risk occurred.
- Risk mitigation: steps taken to reduce the severity of the risk.
- Avoidance: steps taken to try to avoid the risk from occurring.

The team developed charts that would be used as a scoring tool (table 4), when assessing the different risks and the degree of severity each risk had. The complete version for risk scoring and criteria can be found in appendix G.

Table 4: Excerpt from risk scoring chart.



Risk = Probability x Severity						
Probability	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	2	3	4	5
Severity						

4.3 Risk reviews

Risk analysis requires continuous reviews throughout the duration of the project. The group decided that going forward, each risk assessment would be conducted and finalized right after each sprint review. The risk lead would perform the risk review by going over each risk with the group, to see if any of the previously recorded risks had taken place, or if the risk identification document needed to be updated [13] [3] [14] [16].

Terms for updating:

- Has any of the risks occurred?
- If so, what was the consequence?
- If not, is there any reason for updating probability and severity?
- If risk lead must update probability and severity, why should the risk be updated?
- Is risk mitigation up to date?
- Are there any other measures the team can take to further prevent the risk from occurring?

In addition to taking part in the regular risk assessments that were carried out during the project, each team member had a duty to inform the risk lead of any changes to the project that would adversely affect the project and/or group.

5. Mathematical modelling

This chapter includes details on RAO, and the mathematical modelling in terms of mathematical and mechanical calculations.

5.1 The project’s mathematical concepts

The Skidding and Transit Analysis Software (STAS) is the culmination of many mathematical calculations and ideas. This includes information on the weather, such as wind speed and wave heights, as well as information on how a vessel moves in relation to the waves, how this affects the vessel's acceleration, and how this is transmitted to the large and heavy equipment on board the vessel.

5.2 Response Amplitude Operator

When conducting a skidding and transit analysis, one of the first things to take into account is how a vessel moves in respect to the waves. This type of data can be obtained by something called Response Amplitude Operator (RAO), and RAO is also the type of data that will function as an input into the application, before continuing with the analysis [19] [20] [21].

RAO is based on how a vessel moves in water, with respect to the six degrees of freedom. There are several ways on how one can obtain RAO data, either by conducting model testing in a wave pool or by diffraction analysis software [20]. TFMC provided the group with the RAO data that was used for this project. Since this information is sensitive, the RAO data has been rendered anonymous before being shared with the group. The RAO data is actual data from a previous client.

The six degrees of freedom (6DOF) describes how a rigid body moves with six mechanical degrees of freedom in three-dimensional space as seen in figure 10. Three of which are translational along an axis, and the remaining three are rotational motion [22] [23].

The three translational motions of freedom are:

Surge: The forward and backwards motion along the x – axis. (+ direction forward)

Sway: The movement left and right along the y – axis. (+ direction to port)

Heave: The up and down motion along the z – axis. (+ direction is up)

The three rotational motions of freedom are:

Roll: Tilting motion from side to side along the x – axis. (+ direction is starboard side down)

Pitch: The forwards and backwards tilting motion along the y – axis. (+ direction is bow down)

Yaw: The right and left turning motion along the z – axis. (+ direction is bow to port)

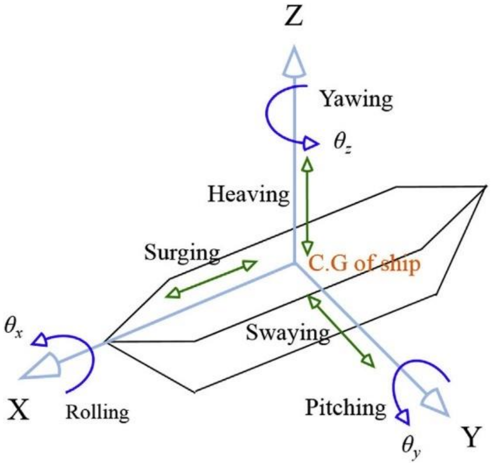


Figure 11: Graphic depiction of the 6DOF

Each RAO or degree of freedom contains data, and in this case, two sets of numbers that describe a vessel's response for a given degree of freedom to a specific direction and a predetermined amount of time. The two sets of numbers are amplitude and phase angle.

The RAO data are different for each vessel. Wave period, also known as frequency, wave direction, and forward speed will all vary. New RAO data must be gathered, and a new analysis must be done for any change in an analysis, such as when a new vessel is present.

5.2.1 Euler angles

In the acceleration calculations shown below, the rotational movement required could be calculated using two distinct methods. Leonard Euler's theorem for rotation of a free body, used in this project, or a different method known as quaternions.

By employing a rotation matrix and the right-hand-rule to construct rotating vectors with a particular angle about the z-, y-, and x-axis, Euler or Eularian angles can determine the motion of a body rotating in free space [22].

The following three rotation matrices are presented as such:

$$R_z(\gamma) = \begin{bmatrix} cos\,(\gamma) & -sin\,(\gamma) & 0 \\ sin\,(\gamma) & cos\,(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

$$R_y(\beta) = \begin{bmatrix} cos\,(\beta) & 0 & sin\,(\beta) \\ 0 & 1 & 0 \\ -sin\,(\beta) & 0 & cos\,(\beta) \end{bmatrix} \quad (2)$$

and

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ (\alpha) & -sin\ (\alpha) \\ 0 & sin\ (\alpha) & cos\ (\alpha) \end{bmatrix} \tag{3}$$

The three matrices can be combined to calculate an $\mathbb{R}^3$matrix, for calculation of rotation in three planes.

$$R_{zyx} \begin{bmatrix} cos(\beta)cos(\gamma) & sin(\alpha)sin(\beta)cos(\gamma) - cos(\alpha)sin(\gamma) & cos(\alpha)sin(\beta)cos(\gamma) + sin(\alpha)sin(\gamma) \\ cos(\beta)sin(\gamma) & sin(\alpha)sin(\beta)sin(\gamma) + cos(\alpha)cos(\gamma) & cos(\alpha)sin(\beta)sin(\gamma) - sin(\alpha)cos(\gamma) \\ -sin(\beta) & sin(\alpha)cos(\beta) & cos\alpha(\alpha)cos(\beta) \end{bmatrix} \tag{4}$$

5.2.2 Quaternions

Quaternions are an alternative method for calculating a body's rotation in free space. There is one issue with Euler angles called Gimbal lock that one can possibly encounter when using Euler angles and may be the deciding factor on whether to utilize Quaternions or Euler angles [24]. Gimbal lock occurs when two of the rotational axes ends up in the same location, and one loses a degree of freedom. Since the rotation angles the subsea equipment experiences are small, it is safe to assume that Gimbal lock will not occur, and it is safe to use Euler angles [24].

With quaternions, the directional vector:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{5}$$

is combined with the corresponding xyz axis, also referred to as ijk when using quaternions, as seen in equation 5.

When it comes to 3D graphics and systems that rotate at greater angles, quaternions are most frequently used. If one were to highlight the benefits of using quaternions in this project, it would be to reduce the number of equations the program needs to solve, to reduce the capacity need of the application. When combining three rotational 3X3 matrix into one, a typical rotation matrix requires twenty-seven multiplications [25] [26], but quaternions only require sixteen multiplications. The difference in the number of calculations would be the reason for considering using quaternions, in terms of the capacity of the application.

5.3 Defining all the different symbols and variables

The mathematical part of the student task is one of the bigger fields of interest when it comes to mechanical engineering work. The final software must include various equations that lead to a result that gives us maximum load capacities, accelerations and bending moments. First, we must clarify what certain symbols mean and what they are used for.

r(t) is the position of the vessel at a given time in a coordinate system. This tells us how the vessel is moving, which is a key factor when it comes to calculating loads and accelerations. r(t) can be considered as the reference point on the vessel.

RAO is the operator that tells us how the vessel responds to a wave. This operator’s unit of measurement is in meters per meter (m/m) or radians per meter (rad/m). This might be a little confusing at first glance, but this tells us how many meters the vessel moves for every meter of corresponding wave and how much the vessel rotates for every meter of wave.

A is the amplitude of the wave measured in meters. The wave amplitude is just a measurement of how tall the wave is. The amplitude is visualized by looking at the middle of the wave and all the way to the top of the wave. Another way to understand the wave amplitude is by dividing the full wave height by two.

ω is the wave frequency and gives information about the number of waves that pass during a certain time frame. We measure the wave frequency in rad/sec.

ϕ is our phase angle measured in radians (which is the angle between the wave and the vessel). This angle also describes how the vessel responds to a wave. This is more of an indicator of how long it takes for the vessel to move accordingly to the wave.

p_0 is the position at the bottom of the stack, where the bending moment is at its peak. This point lies at the coordinates where the x and y values of the bottom component are half their length and the z value on the stack is 0.

(es) is a point similar to p_0, but only the z-value is altered. This point can be described as a point where we want to find information such as acceleration and bending moment. This point will change all the time but is used to describe the current coordinate we are interested in. es is easier explained by saying in which height of the stack we are on.

α, β and γ are the angles measured in radians calculated by the RAO data, amplitude, wave frequency and phase angle from roll, pitch and yaw respectively.

θ is a vector made up by α, β and γ: θ = [α, β, γ]

R(θ) is the rotational matrix with θ being the angle of the vessel compared to the x, y and z axes. The combined rotational matrix R(θ) is given by $R_z(\gamma)R_y(\beta)R_x(\alpha)$, which are all rotational matrices describing rotation around each axis.

5.4 Defining position, velocity and acceleration

When describing the position, velocity, and acceleration of the vessel there are a few things to take into consideration. First, one must find the position, velocity, and acceleration for the vessel itself which is given by the RAO data for the specific vessel and the amplitude of the wave. In addition, the rotation matrix that is built up of movement around the x, y and z axes (described by α, β and γ) is applied together with the position of the stack. The position of the stack is given in x, y and z coordinates in relation to the reference point. For velocity and acceleration, the process becomes more complicated which will be explained further in the subsections for velocity and acceleration [21] [27] [28].

5.4.1 Vessel position

The position of the vessel (a singular point) in a wave is always given by the equation [28]:

$$r(t) = RAO\ Acos(\omega t + \varphi), \tag{6}$$

where the RAO data is what determines how much the vessels position will change in comparison to the wave [27] [21] [28] [29].

Which translates the following equations for the movement along the x, y and z axes:

$$r_x(t) \ = \ RAO_{surge}Acos(\omega t + \varphi_{surge}) \tag{7}$$

$$r_y(t) \ = \ RAO_{sway}Acos\big(\omega t + \varphi_{sway}\big) \tag{8}$$

$$r_z(t) \ = \ RAO_{heave}Acos(\omega t + \varphi_{heave}). \tag{9}$$

And then end up with a position for the reference point (the vessel) for x, y and z given by three coordinates and indicates where the vessel is in comparison to its initial position at t = 0 in [0, 0, 0]:

$$r(t) = [\, r_x(t), \quad r_y(t), \quad r_z(t)]. \tag{10}$$

In addition, we can obtain the angles through a similar operation:

$$\alpha(t) \ = \ RAO_{roll}Acos(\omega t + \varphi_{roll}) \tag{11}$$

$$\beta(t) = \ RAO_{pitch}Acos\big(\omega t + \varphi_{pitch}\big) \tag{12}$$

$$\gamma(t) = \ RAO_{yaw}Acos\big(\omega t + \varphi_{yaw}\big). \tag{13}$$

The combined angle θ, is given as a matrix with the angles of rotation:

$$\theta(t) = [\alpha(t), \quad \beta(t), \quad \gamma(t)]\,. \tag{14}$$

It is worth mentioning that for the rotation angles, the RAO data is often given in deg/m and must be converted to rad/m for use in equations (11), (12) and (13).

It is important to remember that the RAO data that is given for translational movement must be converted to the unit meter per meter (m/m) and for rotational RAO to radians per meter (rad/m) [22] [21] [28] [29] [30].

5.4.2 Vessel velocity

Knowing the equation for the position of the vessel at any given time, we can use the time derivative to get the velocity at the same value for t. The derivative of r(t) is given by [28] [30]:

$$r'(t) = \frac{dr(t)}{dt}$$

$$= -RAO\omega\ A sin(\omega t + \varphi) \qquad (15)$$

Doing this for all axes again we get the following set of equations for each translational velocity [28]:

$$r_x'(t) = -RAO_{surge}\omega\ A sin(\omega t + \varphi_{surge}) \qquad (16)$$

$$r_y'(t) = -RAO_{sway}\omega\ A sin(\omega t + \varphi_{sway}) \qquad (17)$$

$$r_z'(t) = -RAO_{heave}\omega\ A sin(\omega t + \varphi_{heave}) \qquad (18)$$

Combining (16), (17) and (18) into a singular expression that describes the velocity for all x, y and z axes gives [28]:

$$r'(t) = [\,r_x'(t), \quad r_y'(t), \quad r_z'(t)] \qquad (19)$$

Looking at the same set of equation we can extract the angular velocities using the derivatives of α, β and γ [28]:

$$\alpha'(t) = \ -RAO_{roll}\omega A sin(\omega t + \varphi_{roll}) \qquad (20)$$

$$\beta'(t) \ = \ -RAO_{pitch}\omega A sin(\omega t + \varphi_{pitch}) \qquad (21)$$

$$\gamma'(t) \ = \ -RAO_{yaw}\omega A sin(\omega t + \varphi_{yaw}) \qquad (22)$$

For the combined angular velocity of the vessel for (20), (21) and (22) we get following [28]:

$$\theta'(t) = [\alpha'(t)\,, \quad \beta'(t)\,, \quad \gamma'(t)] \qquad (23)$$

5.4.3 Vessel acceleration

For acceleration, we use the second derivative of the position equation instead of once.

$$r''(t) = \frac{d^2r(t)}{dt^2}$$

$$= -RAO\omega^2 Acos(\omega t + \varphi) \qquad (24)$$

Again, using this equation for each translational direction, we conclude that [28] [30]:

$$r_x''(t) = -RAO_{surge}\omega^2\ Acos(\omega t + \varphi_{surge}) \qquad (25)$$

$$r_y''(t) = -RAO_{sway}\omega^2\ Acos(\omega t + \varphi_{sway}) \qquad (26)$$

$$r_z''(t) = -RAO_{heave}\omega^2\ Acos(\omega t + \varphi_{heave}) \qquad (27)$$

The acceleration vector for (25), (26) and (27) ends up being [28]:

$$r''^{(t)} = [r_x''(t), \quad r_y''(t), \quad r_z''(t)] \qquad (28)$$

Lastly, we need angular acceleration which can be described as the second derivatives of α, β and γ. [28]

$$\alpha''(t) = -RAO_{roll}\omega^2 Acos(\omega t + \varphi_{roll}) \qquad (29)$$

$$\beta''(t) = -RAO_{pitch}\omega^2 Acos(\omega t + \varphi_{pitch}) \qquad (30)$$

$$\gamma''(t) = -RAO_{yaw}\omega^2 Acos(\omega t + \varphi_{yaw}) \qquad (31)$$

With the combined angular acceleration of (29), (30) and (31) we get [28]:

$$\theta''(t) = [\alpha''(t)\,, \quad \beta''(t)\,, \quad \gamma''(t)] \qquad (32)$$

5.4.4 Rotation matrix

Looking at the rotation of a vessel, there are three axes for rotation. The combination of all these three gives the total rotation movement of the vessel [20] [28] [22].

The values from the angles α, β and γ that are calculated from the rotation RAO data in combination with the wave data are what we use to describe the orientation of the vessel. The theory for the rotation matrix is given by improper Euler angles, where it is important to compute the rotations in the right order. The reason for it being called improper Euler angles is because in the case where proper Euler angles are being used, the rotation happens around the same axis for the first and last rotation. In our case, the rotation finds place around three different axis, and is therefore referred to as improper Euler angles [22] [20].

The rotation around the z-axis is given by the equations (1), (2) and (3).

The product of all the matrices (1), (2) and (3) in that order gives [22]:

$$R(\theta) = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{33}$$

When calculating the matrix that describes the full orientation of the vessel, we must follow the order of operations. This is done through first multiplying $\mathrm{R_z}(\gamma)$ with $\mathrm{R_y}(\beta)$, and then multiplying the result of $(\mathrm{R_z}(\gamma)\mathrm{R_y}(\beta))$ with $\mathrm{R_x}(\alpha)$ [22] [28].

$$R(\theta) = \begin{bmatrix} cos(\beta)cos(\gamma) & sin(\alpha)sin(\beta)cos(\gamma) - cos(\alpha)sin(\gamma) & cos(\alpha)sin(\beta)cos(\gamma) + sin(\alpha)sin(\gamma) \\ cos(\beta)sin(\gamma) & sin(\alpha)sin(\beta)sin(\gamma) + cos(\alpha)cos(\gamma) & cos(\alpha)sin(\beta)sin(\gamma) - sin(\alpha)cos(\gamma) \\ -sin(\beta) & sin(\alpha)cos(\beta) & cos\alpha(\alpha)cos(\beta) \end{bmatrix} \tag{4}$$

Seeing how the orientation of the vessel is given by R(θ), we also must take into consideration how the rotation matrix components that leads to the orientation of the vessel is before the application of R(θ). That before orientation is described by the transpose of R(θ) expressed by R^T(θ). Considering this is a rotation matrix, the transpose is equal to the inverse considering rotation matrices are orthogonal [22] [28].

For the transpose matrix, considering it is orthogonal, one can swap the rows with the columns to obtain the reverse orientation of the vessel [22].

$$R^T(\theta) = \begin{bmatrix} cos(\beta)\,cos(\gamma) & cos(\beta)\,sin(\gamma) & -\,sin(\beta) \\ sin(\alpha)\,sin(\beta)\,cos(\gamma) - cos(\alpha)\,sin(\gamma) & sin(\alpha)\,sin(\beta)\,sin(\gamma) + cos(\alpha)\,cos(\gamma) & sin(\alpha)\,cos(\beta) \\ cos(\alpha)\,sin(\beta)\,cos(\gamma) + sin(\alpha)\,sin(\gamma) & cos(\alpha)\,sin(\beta)\,sin(\gamma) - sin(\alpha)\,cos(\gamma) & cos(\alpha)\,cos(\beta) \end{bmatrix} \tag{34}$$

5.4.5 Point of interest position

The point of interest is given by p_0+es [28]. If we want to translate this movement to a point that is not the reference point, we will need to include a rotation matrix to describe the position [22]. This is done via the rigid body movement equation:

$$p(t) = r(t) + R(\theta)(p_0 + es). \tag{35}$$

The theory behind this equation is that it takes the position of the ship and adds a rotation around all three axes based on the rotation matrix R with respect to the angle θ.

5.4.6 Point of interest velocity

In a point that is affected by rotation we need some extra work to find the velocity. This is done by the derivative of the position p(t) [28]:

$$p'(t) = r'(t) + (\ \theta'(t) \times\ R(\theta)(p_0 + es)\ ). \tag{36}$$

The first component of this equation is r'(t) which is explained under vessel velocity; however, the second part of this equation is worth mentioning. The angular velocity vector can be visualized easier by thinking about it as the axis of rotation. If something is rotating around the x-axis, the angular velocity vector would also point in the same direction as the x-axis [22].

Knowing this, the best way to understand would be to look at the direction of the θ'(t) vector to be the same direction as the axis of rotation and the direction of R(θ)p_0+es to be the direction of a vector that starts in r(t) and ends up in p_0(t)+es with respect to the angle of rotation. The velocity due to rotation would then be perpendicular to both vectors and the size would be determined by the angular velocity and the distance from r to p_0+es [22] [28].

5.4.7 Point of interest acceleration

Looking at the acceleration for a given point on the vessel we can use the second derivative of the positional equation [28]:

$$p''(t) = r''(t) + \ (\theta''(t) \times R(\theta)(p_0 + es)) + (\theta'(t) \times (\ \theta'(t) \times R(\theta)(p_0 + es))). \qquad (37)$$

Knowing that r''(t) is already the translational acceleration in x, y and z. The next part of the equation: (θ'' (t) × R(θ) (p_0+es)) crosses the angular acceleration with the rotated position of the point we are looking at. This leads to a vector perpendicular to both the angular acceleration and the vector starting in r(t) and ends up in the rotated position of p_0+es. Looking at the direction of this acceleration vector it will always be perpendicular to the velocity vector that is being calculated by the p'(t) equation. This part of the equation is directly influenced by the change of direction of the velocity vector [28].

Lastly, the third part of the equation: (θ'(t) × (θ'(t) × R(θ) (p_0+es))) is directly influenced by the magnitude of the velocity vector. These two are linked in such a way because changes in the magnitude of the acceleration vector also influences the magnitude of the velocity vector [28].

5.4.8 Local acceleration

Once p''(t) is calculated we know the global accelerations in x, y and z direction due to both translational and angular acceleration. What we mean by global acceleration is that the coordinate system being used is the standard one where the z axis is pointing straight down in the negative direction towards the centre of the earth, and up towards the sky in its positive direction. The x and y components of the global coordinate system follow the rules that they both must be perpendicular to the z axis. The direction of the x axis follows the vessel from front to back and the y axis is then just perpendicular to both x and z [28].

To convert from global to local acceleration we can look at it two ways. Either we are rotating the coordinate system to match the orientation of the vessel, or we are rotating the vessel together with the coordinate system to the vessels orientation in a global coordinate system. This is done by implementing the transpose matrix to the global acceleration [28] [22]:

$$p''(t)_{local} = p''(t)R^T(\theta). \qquad (38)$$

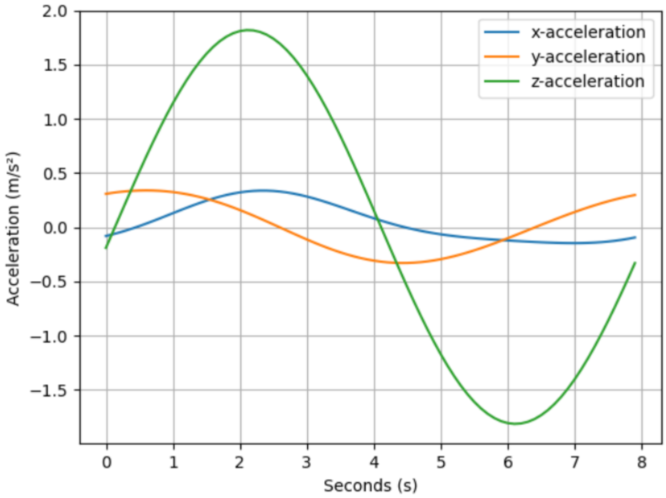


Figure 12: Acceleration output.

Example of the accelerations for local x, y and z calculated by the tool. (HS 30°, 2m wave Amplitude and 8m stack height)

The gravitational acceleration vector g = [0, 0, -9.81], will also be multiplied by the transpose to find the local values for the gravitational acceleration working on the stack. This is done similarly to the conversion of the translational and rotational acceleration:

$$g(t)_{local} = \; g\, R^T(\theta). \tag{39}$$

With both components contributing to the local acceleration, we can use the acceleration components in the newly defined coordinate system. The x and y components will be the accelerations working normal to the stack and will be the components contributing to the leading bending moment.

5.5 Bending moment

Coming from the acceleration components we must transition into bending moment using the combined mass of the stack and the distance from the top of the stack to the height we are finding the bending moment in. That means we start from the top and integrate down towards the bottom. This is done in several steps and depends on how many components the stack is built up by.

The bending moment of the first component (starting from the top of the stack) can be calculated rather easily, however once a new component with different cross section and different density is introduced, the calculation complicates.

The bending moment of the stack is given by an integral [31] [28]:

$$M(x) = \int_{x}^{L-L_n} (s-x)(p''(s,t)_{local} + g_{local})\rho(s)ds. \tag{40}$$

x is the height where we are calculating the bending moment in meters.

L is the total height of the entire stack in meters.

L_n is the sum of heights of all components that are not being included in the calculation. (Example: If we are looking at the top component, all components are being used, which means L_n = 0. However, if we are looking at component number 3 from the top $L_n = L_1 + L_2$. This means we are subtracting the height of the top 2 components.) Also measured in meters.

(s – x) represents the distance from the top of the stack to the point x. We start in s = x and continue until x = 0. s is always the total height of the stack.

$p''(s, t)_{local}$ is the acceleration for the local coordinate system due to both rotation and linear movement. The linear component will always be the same regardless of what the stack looks like. However, while looking at the rotational component we must take the shape of the stack into consideration. The z-value from centre of gravity of the entire stack is where we convert the angular acceleration into tangential acceleration from the formula $a^T = \alpha r$ (α is the angular acceleration) where r is set to the z value of centre of gravity from the stack plus the value of the z-component of p_0. The way this is implemented is to use the angular acceleration and multiply with the z-value of CoG plus the value of p_0 z-value. The angular acceleration is affected in the x direction from yaw and pitch and the y direction is affected by yaw and roll. The z-value of the acceleration is not important when it comes to the bending moment as it is so much thicker in the z direction compared to x and y. This implies that the breaking points will occur due to x and y acceleration.

g_{local} is given by the vector g = [0, 0, -9,81] multiplied by the transposed rotation matrix R^T. The acceleration due to gravity is the same across all heights of the stack. This value is increased in the x and y direction as the ship tilts more in either pitch, roll or both. Looking at yaw movement, it will not affect the gravitational x and y components considering the gravitational vector points straight down along the same axis. That will not change no matter the rotation around z.

ρ(s) is the density function that gives us the mass of the stack for every height, s. This function is made up from the density of the component together with the length and width of the cross section of the component.

The result, M(x), of the equation gives the bending moment measured in Newton meters (Nm). This is a complex calculation considering all the variables that change for each component. The density, width and length can all change when a new component is introduced. In addition, the bending moment from the first component needs to be taken into account while the new dimensions are being applied. The tool converts Nm to kNm for better readability for the user.

Integration of equation (40) gives us [28]:

$$M(x) = \left[\frac{1}{2}(s-x)^2\left(p''_{local} + g_{local}\right)(\rho_x)\right]_{x}^{L-L_n} + C \qquad (41)$$

The C at the end can be considered the bending moment of the already calculated components summed together. It is essential to add them to the newly calculated bending moment for the coming components.

The way the bending moment is calculated through the software can better be visualized through some rough sketches:

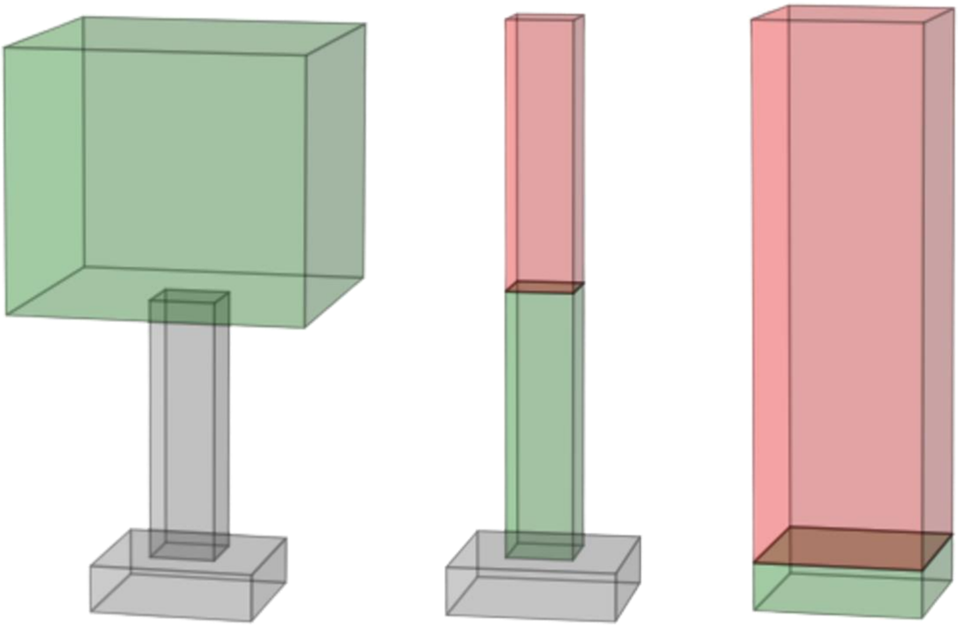


Figure 13: Bending moment.

Visual representation of how bending moment is integrated for height in the calculations.

The idea behind the integration process in our software is that we take the first component and integrate until we reach the height of the first component. We then move on to the next component with an integration from the full height of the first and second component with a new density. Then we subtract the height of the first component with the second components density to get the second component isolated. Lastly we add the bending moment from the first componenet to the newly calculated bending moment for the second component. For the third component, it works the same way as for the second component where we inegrate the full height of the stack with the third density function and subtract the height of the first two components, while keeping the height of the third compoenent. Then we again add the bending moment from the two components we already calculated [28].

Considering this way of calculating the bending moment is through numerical integration, we will find an estimation to the bending moment in each height of the stack. The way one can give more realistic values to the bending moment is to decrease the step between each calculated integral. Using a step of 1m (meter) would not be a good idea considering the purpose of the software is to be able to see the bending moment in each critical point. In addition, the calculation is drastically different when there is a shift in components. Therefore, choosing a step of 1mm (millimeter) would make more sense to give a realistic representation of how the bending moments propagates through the stack [28].

With the x and y componets of the acceleration, the bending moment is calculated in two steps. First in the local x direction with the x-acceleration, and then with the y-acceleration for the direction of the local y. The benefit of doing it this way is that the direction of the bending moment is easier to identify. Through looking at the x and y components negative or positive value one can see which quadrant of the coordinate system the total vector is pointing. With the size of each component its also possible to find the angle and combined size of the bending moment.

Example of the bending moment calculations done by the software:

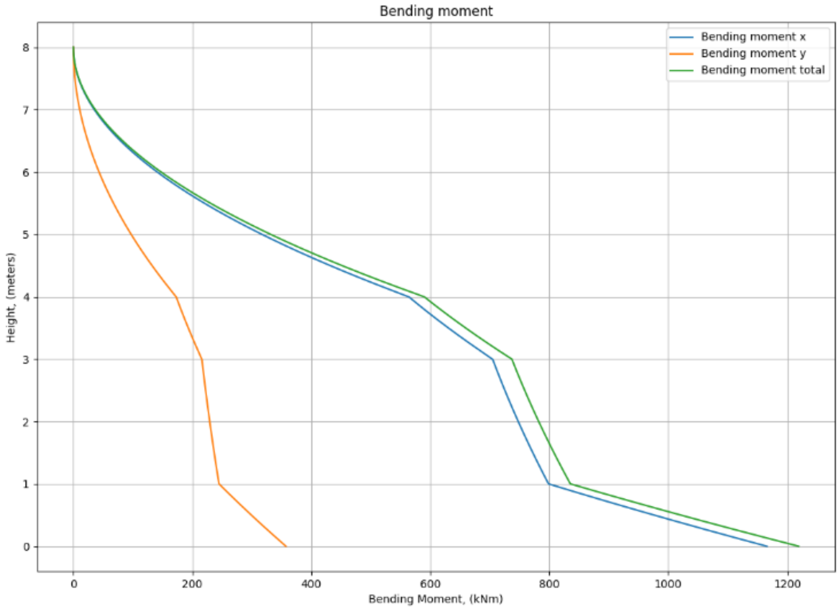


Figure 14: Bending moment output

Output from the tool, showing BM in x- and y-directions for every h of the stack.

Through this graph it is easy to see where a new component starts considering the dents in the graph. This is due to the change of cross section which leads to either a bigger or a lower mass growth per iteration of height. If all the component cross sections would have been the same or if there was only one component, the growth would be more like an exponential graph.

5.6 Centre of Gravity (CoG)

When looking at CoG calculations it is important to look at each component on its own first. Every component has a volume and density which gives us the total mass of the component. In addition to that the placement of the components while stacked on top of each other is crucial. If one or more components are shifted in either x, y or both directions it will lead to an overall shift in the centre of gravity. In this project one can look at each component as uniform in the sense where the CoG lies in the centre of the component unless an exception states otherwise. Every component will also have a length, width and height that will be used for the overall calculation of the total CoG [32].

The overall CoG in our software is calculated with an array with the values [x, y, z]. These values will change depending on the mass and placement of each component. The equations used to find CoG and mass for every component individually are:

$$\bar{x} = \frac{l}{2} \tag{42}$$

$$\bar{y} = \frac{w}{2} \tag{43}$$

$$\bar{z} = \frac{h}{2} \tag{44}$$

$$m = lwh \tag{45}$$

[33]

Once all the individual CoGs and masses are known the next step is the total CoG for the whole stack.

We will first define some of the terms being used in the following equation for CoG.

- $x_{pos\ 1,2\ldots}$, $y_{pos\ 1,2\ldots}$ and $z_{pos\ 1,2\ldots}$ are being used to describe the positions of the components compared to p_0. If we take the second component from the bottom as an example, we look at the CoG of the individual component as the position. If there is no displacement in the x direction the x component of the displacement would be 0. If the components y displacement is 10 cm, the displacement of y would be 0,1 if measured in meters or 3,94 if measured in inches. For the z displacement it is a little more complicated as all components are stacked on top of each other. Let's say there was no displacement of the second component in the z axis we would have to use the height of the first component and add half the height of the second component as the z displacement.
- $m_{1,2\ldots n}$ is used to describe the mass of each component in kg or inches. This is done simply by using the product of the length, width, and height.
- m_{tot} is just the total mass of components all together in the same unit as $m_{1,2\ldots n}$.

The total CoG of the stack is therefore given by three sets of equations:

$$\bar{x}_{total} = \frac{x_{1_{pos}}m_1 + x_{2_{pos}}m_2 \ldots + x_{n_{pos}}m_n}{m_{tot}} \qquad (46)$$

$$\bar{y}_{total} = \frac{y_{1_{pos}}m_1 + y_{2_{pos}}m_2 \ldots + y_{n_{pos}}m_n}{m_{tot}} \qquad (47)$$

$$\bar{\bar{z}}_{total} = \frac{z_{1_{pos}}m_1 + z_{2_{pos}}m_2 \ldots + z_{n_{pos}}m_n}{m_{tot}} \qquad (48)$$

[33]

The final CoG will use these three components of equations as coordinates as follows:

$$CoG_{total} = [\bar{x}_{total}, \quad \bar{y}_{total}, \quad \bar{\bar{z}}_{total}] \qquad (49)$$

Figure 15 shows the CoG of all the individual components. These are the x, y and z coordinates used in the calculation of the total CoG. The coordinates, together with the masses of each component, is what we use to obtain a final value for CoG in the stack. These values are obtained from equation (46), (47) and (48) and then combined into an actual coordinate in equation (49).

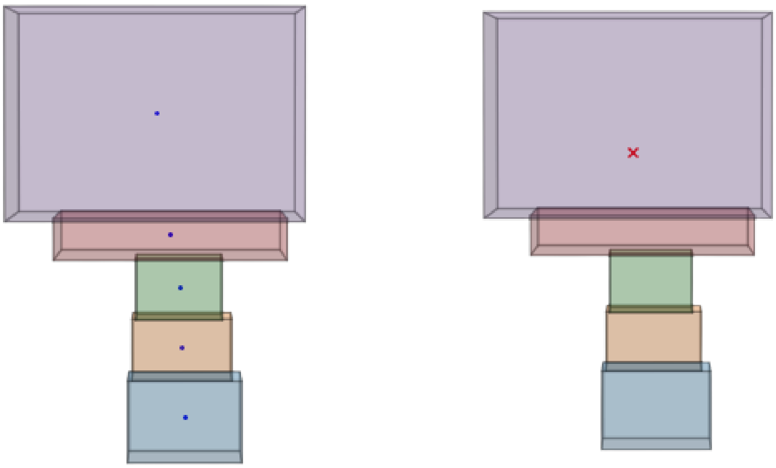


Figure 15: CoG visualization

Individual calulation of CoG with blue dots, and total CoG calculation with a red cross.

5.7 Utilization

Utilization is the comparison of applied load versus the capacity available for the connection between components or cross sections. The factor that divides these two values is factor of safety. Any subsea system is designed using ISO standards, and utilization is calculated using the equations given from the ISO standard [34].

The result from the calculations for capacity and bending moment must be entered into the utilization equations for further assessment of the operation [34] . The calculations we have used for these calculations and analysis is based on ISO standard 13628-7 and the calculations incorporate a safety factor to ensure that skidding and transit operations can occur well below the line of such operations' maximum capacity [34]. There is a maximum permitted capacity of 1 for utilization. This is ultimately merely a numerical method, and it is up to the operator for the skidding and transit operation to determine if the operation is feasible [34].

$U = Maximum\ allowable\ capacity.$

$F_d = Safety\ factor.$

$T_e = Applied\ tensile\ load\ or\ effective\ tension.$

$T_c = Critical\ tensile\ load.$

$M = Applied\ bending\ moment.$

$M_c = Critical\ bending\ moment$

$$U = \frac{1}{F_d}\left(\frac{T_e}{T_c} + \frac{M}{M_c}\right) \tag{50}$$

The factor of safety has its own set of numerical value, based on safety criteria. The criteria given from TFMC are as follows [28] [34]:

$Normal = \frac{2}{3}$

$Extreme: 0.8$

$Accidental: 1$

In the end, one will utilize (56) for the utilization calculation and the consequent safety factor F_d to decide whether the operation is taking place inside or outside the weather window. As a result of material data and outcomes of earlier calculations for accelerations and bending moments, T_e, T_c, M, and M_c are known numerical factors [34].

5.7.1 Pressure capacity calculation

Before a stack can be lowered into the sea through the moonpool, a topside pressure test needs to be conducted. Normally a stack is experiencing a compression force when in transit, but this can change from compression to tension in terms of forces applied to the components. And it is therefore important to know if the change from compression to tension will give a tension level above the yield limit [35], and the material will have a plastic deformation instead of elastic deformation. For pressure test capacity, one must take into consideration the pressure difference between internal and external pressure of the components. This is given by an equation derived from the ISO standard [34]:

$$T_e = T_w - P_{int}A_{int} + P_{ext}A_{ext} \tag{51}$$

$T_e = Effective\ tension.$

$T_w = True\ wall\ tension\ of\ the\ material.$

$P_{int} = Internal\ pressure.$

$P_{ext} = External\ pressure.$

$A_{int} = Internal\ cross - section\ area\ of\ the\ pipe.$

$A_{ext} = External\ cross - section\ area\ of\ the\ pipe.$

T_e is determined and added to the formula used from the ISO standard to determine the overall stress on the pipeline in the components [34]:

$$T = T_e + \frac{\pi}{4}((P_{int} + P_{ext})D_s^2) \qquad (52)$$

$T = total\ tension\ on\ the\ pipeline.$

$T_e = Effective\ tension.$

$P_{int} = Internal\ pressure.$

$P_{ext} = External\ pressure.$

$D_s^2 = Seal\ diamater\ of\ the\ components.$

T_e is the value needed for the utilization equation (56). As mentioned for utilization calculations, these calculations provide a numerical value, and it is up to the operator to determine if the operations can be conducted [34].

6. Software architecture

Information pertaining to implementation methods and technical details of the software is presented in this chapter. For the creation of a software with multiple people on the project, communication is an extremely important part of it. Making sure everyone is one the same page is not always the easiest part so using good modelling techniques to help everyone visualize and work together to reach a common goal is crucial. This is why we have used the modelling techniques below. Some of the diagrams will be listed here and the rest will be shown in the Appendix J.

Our software has four main objectives, these are:

- To read a RAO data file
- Manage Database tables
- Perform Analysis
- Export file

These four objectives are represented by the four use cases for the software (see figure 16 below).

We only have one actor, so all use cases are made with the user in mind. There is also an included objective within the task of performing the analysis, which is to display visual representation of data, more specifically the stack.

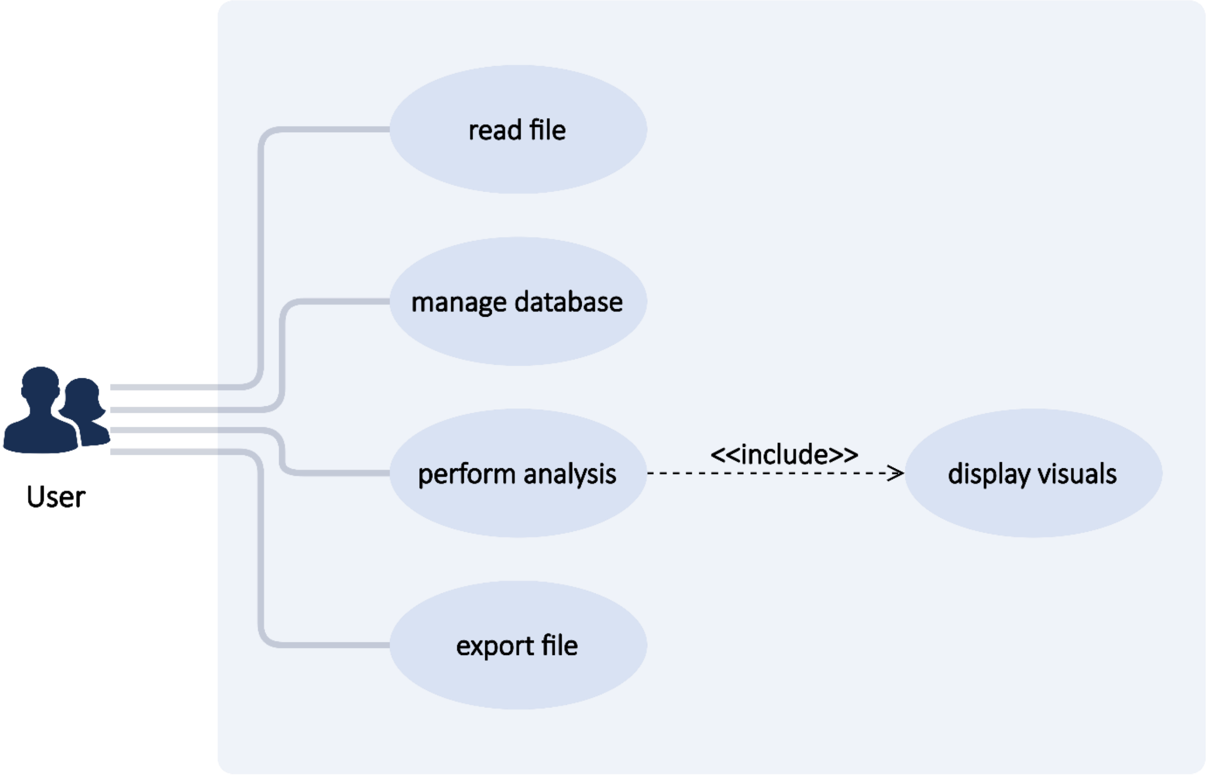


Figure 16:Use case diagram

Eight sequence diagrams were generated from the four use cases above. Sequence diagrams as the name says represents the sequence that objectives are executed in. Below you can see one of our sequence diagrams, run analysis, which shows the order of which the objectives in this specific task should be executed.

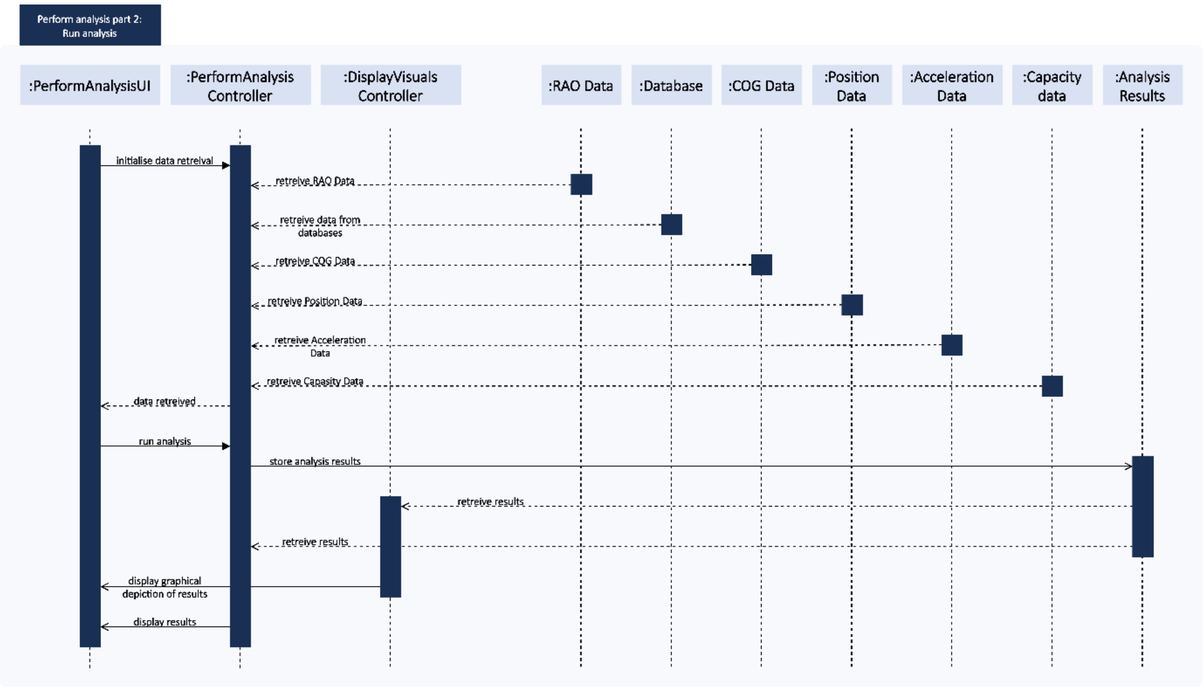


Figure 17: Sequence diagram - run analysis.

The rest of the diagrams are:

- Read file
- Manage database part 1: Show Databases
- Manage database part 2: Detailed view of a database entry
- Manage database part 3: Add to database
- Manage database part 4: Delete from database
- Perform analysis part 1: Input data
- Export file

We have also made a model-view-controller diagram as you can see below, which shows an overview of what controllers interact with the different objects in the sequence diagrams, See figure 18 below. The MVC diagram is mostly to display what part of the software needs to have access to what data.

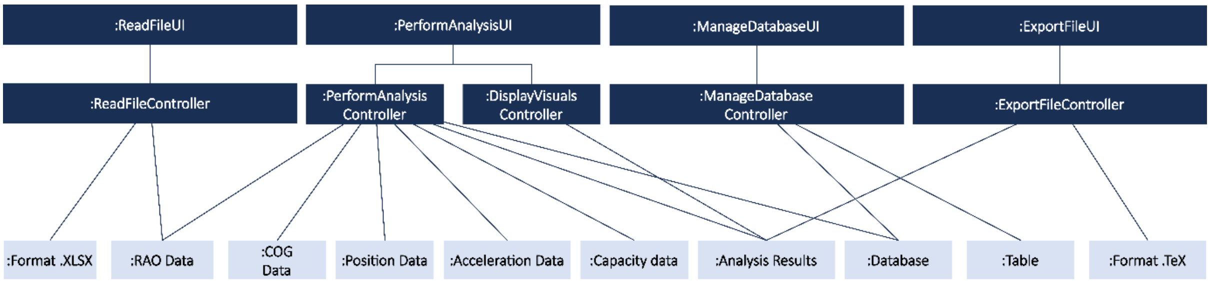


Figure 18: MVC diagram for software

6.1 Implementation overview

The following code is a software implementation of a graphical user interface (GUI) application designed to facilitate various calculations and data management tasks related to the skidding and transit calculations. The project aims to provide a comprehensive tool for analysing and visualizing data, as well as performing custom calculations based on user-defined formulas.

The objective of this software project is to develop an intuitive and user-friendly application that allows users to import and analyse RAO (Response Amplitude Operator) data, manage a database, create custom formulas, calculate utilization parameters, and visualize results. The application is built using the Python programming language and utilizes various libraries such as Tkinter, SQLite, NumPy

The software is implemented using the Tkinter library, which provides a set of tools and widgets for building GUI applications. The code establishes a connection to the SQLite database file named 'Database.db' using the sqlite3 module. It retrieves the names of tables present in the database for further processing.

6.2 Class based software structure

When deciding on the software structure it became clear that there was a clear focus to prioritize choosing a structure that is easy to maintain and possibly edit in the future. Therefore, a mostly class based, or object-oriented approach was selected for the software

A key component of Python's object-oriented programming (OOP), the class-based architecture offers developers a variety of advantages. Classes help developers organize their code, make it more reusable, maintainable, and extensible for their applications.

Modularity and reuse are two of the class-based system's main benefits. Classes make it possible to build modular components and reusable code blocks that are simple to use in various areas of a codebase or in other projects. This encourages code reuse, cuts down on duplication, and increases development effectiveness.

Classes also make abstraction and encapsulation easier. Developers can hide intricate implementation details through abstraction, creating a more straightforward user interface for interacting with objects. Encapsulation groups a class's data and functionality, enhancing data security and integrity while minimizing code.

This proved vital when GUI started to get filled with repeating or otherwise unnecessarily large blocks of code.

The class-based system in Python is built on inheritance which is essential for reducing code clutter. Developers can construct new classes based on existing ones by using inheritance and inheriting their properties and specific variables. This encourages code specialization and reuse without changing the original implementation.

Classes help to organize and maintain code more effectively. Developers can organize comparable functionalities into classes for various components, making it simpler to explore and comprehend the codebase. The ability to localize updates and changes to classes without affecting other areas of the code improves the maintainability of the code.

Collaborative development is facilitated by the class-based system. Developers can work on various components with classes at same time without interfering on each other’s works.

Multiple developers can work on distinct components of a system while simultaneously testing individual parts of their own code.

Classes also offer logical boundaries for testing and debugging. Individual classes may be the focus of unit testing to independently verify their behaviour. By isolating problems to certain classes, it is easier to identify and fix defects, making debugging more manageable.

The class-based structure also encourages the extensibility of the code. Through subclassing or inheritance, developers can add new functionality to existing classes. This enables the codebase to expand and change to meet evolving requirements without requiring substantial changes to the current code.

In conclusion, this project greatly benefits from Python's class-based structure. It improves the organization, reuse, maintainability, and extensibility of the code, making it possible to develop intricate and scalable applications.

6.3 Independent code to class conversion

As software projects grow in complexity, the need for reusability becomes crucial for efficient development and maintenance. In many cases, code that initially works independently needs to be refactored and organized into a class-based structure to enhance reusability. This subchapter explores the process of transitioning from independent pieces of code to a class-based structure, highlighting the modifications required to achieve code reuse.

Independent code refers to individual functions or sections of code that perform specific tasks without any explicit relationship to each other. Such code segments may be functional in isolation, but they lack the ability to be easily reused in other parts of the project. This lack of reusability often leads to redundant code, decreased efficiency, and increased maintenance efforts.

When working with independent code, it is essential to identify common patterns or functionalities that can be extracted and reused across the project. These patterns might involve similar data structures, algorithms, or even user interface components. By recognizing these opportunities, developers can begin the process of refactoring code into a class-based structure to enhance reusability.

One specific example of transitioning independent code to a class-based structure for reusability involved visualizing centre of gravity calculations in 3D, as well as calculations related to bending moment and utilization. Initially, these calculations were implemented as separate functions that were scattered throughout the codebase. While they worked independently, they lacked a cohesive structure and were difficult to reuse in different parts of the project.

6.4 GUI framework

The decision to select either Tkinter or PyQt as the Python GUI frameworks held significant importance for this project. While both frameworks offer different benefits and detriments, only one may be chosen for this project.

One notable advantage of using Tkinter is its inclusion as a built-in library within the standard Python package. This eliminates the need for additional installations and ensures compatibility across different Python setups, making it convenient for the customer who receives the software. Moreover, this native integration with Python allows seamless utilization of Python's extensive standard library and third-party packages.

Tkinter stands out for its simplicity and user-friendly interface. Its API is straightforward and intuitive, making it an ideal choice for both beginners and developers looking to create simple GUI applications efficiently. Compared to PyQt, Tkinter has a gentler learning curve, enabling users to grasp the fundamentals quickly and develop applications with ease.

Cross-platform compatibility is another key benefit offered by Tkinter. It seamlessly functions on various operating systems, such as Windows, macOS, and Linux, without requiring platform-specific modifications. Tkinter achieves this by utilizing native widgets provided by the underlying operating system, resulting in consistent behavior and a native look and feel across different platforms.

Tkinter's lightweight nature and efficient performance make it an attractive option for developers. It has a smaller footprint compared to PyQt, leading to faster load times and reduced memory usage. This is particularly advantageous for applications that prioritize quick startup times or operate in resource-constrained environments such as heavily repeated calculations.

Furthermore, Tkinter benefits from extensive documentation and a large user community. The availability of numerous tutorials, examples, and online resources aids the learning process and provides valuable support when troubleshooting issues.

When developing this application few problems arose however the ones that did were solved due to the number of available resources regarding troubleshooting.

While PyQt offers advanced features, customizable widgets, and a more contemporary appearance, Tkinter excels in scenarios where simplicity, Python integration, cross-platform compatibility, efficiency, and comprehensive documentation are key considerations.

6.5 GUI layout

The GUI interface consists of a toggle menu, a head frame, and a main frame. The toggle menu is a collapsible panel that provides quick access to different sections of the application. It contains buttons for RAO Data, Database, Custom Formula, Utilization, and Visuals. The head frame displays the application title, "Alon Skidding and Transit calculator," and the toggle menu button. The main frame holds the content of the selected section, which is displayed based on the user's interaction with the toggle menu.

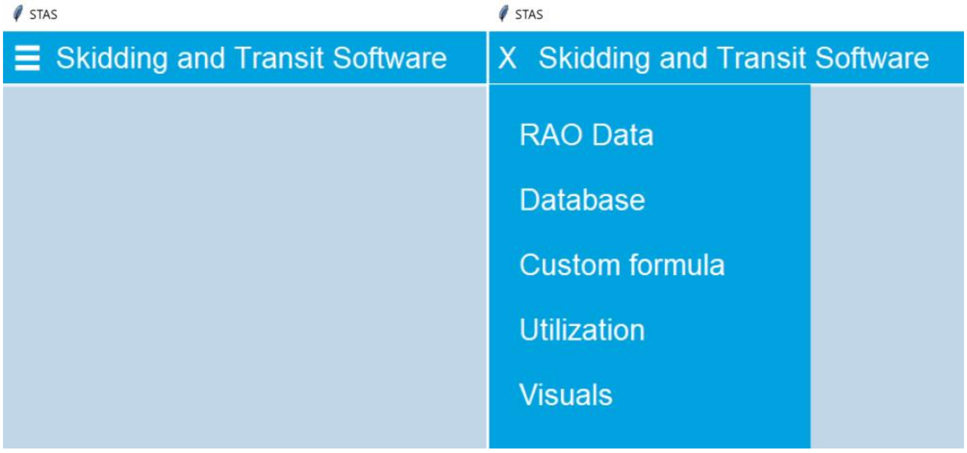

STAS
Skidding and Transit Software
STAS
X Skidding and Transit Software
RAO Data
Database
Custom formula
Utilization
Visuals

Figure 1919: Sidebar menu in the GUI.

The RAO Data section allows users to import Excel files and view the data using the ExcelViewer class. An import button is provided for importing Excel files, and a HelpButton class is used to display the help file in PDF format.

The Database section allows users to manage the database by creating a combobox, also commonly referred to as a dropdown menu, selecting database tables, and editing the database using the DatabaseEditor class. The DatabaseViewer class is used to display the contents of the selected table. While it would be desirable to have the database auto increment with added entries at the current state of the application it is not possible. Later changes to the program should make this a possibility.

The Custom Formula section allows users to define and apply custom formulas. The FormulaApp class is responsible for creating a canvas to display the section's content. A help button is provided to access the corresponding help file.

The Utilization section provides functionality for calculating utilization parameters. The CalculationApp class creates a canvas for displaying the section's content.

The Visuals section allows users to visualize data using the StackConfiguration class. The class displays a stack configuration and provides a help button for accessing the help file.

6.6 Database

A request from our client for their software, was a local database for every user. The purpose of this storage medium would be to contain different types of data that would be used later in the skidding and transit analysis. This would make it possible for our client to be more efficient when running these analysis' since data can be entered into the database once and then be reused later with the click of a button.

With these skidding and transit analysis' there are a lot of different types of data that is included, so a lot of understanding has been needed to recognize what the client needs. This was not a fast process and we had to try a couple of times to get it right.

Alon consists of two computer engineers and two machine engineers, so for us to be able to communicate in the best possible way between the four of us and with our client we decided to make an Object-Role Modelling diagram (*see figure 20 below, a bigger version will be in Appendix K and L*).

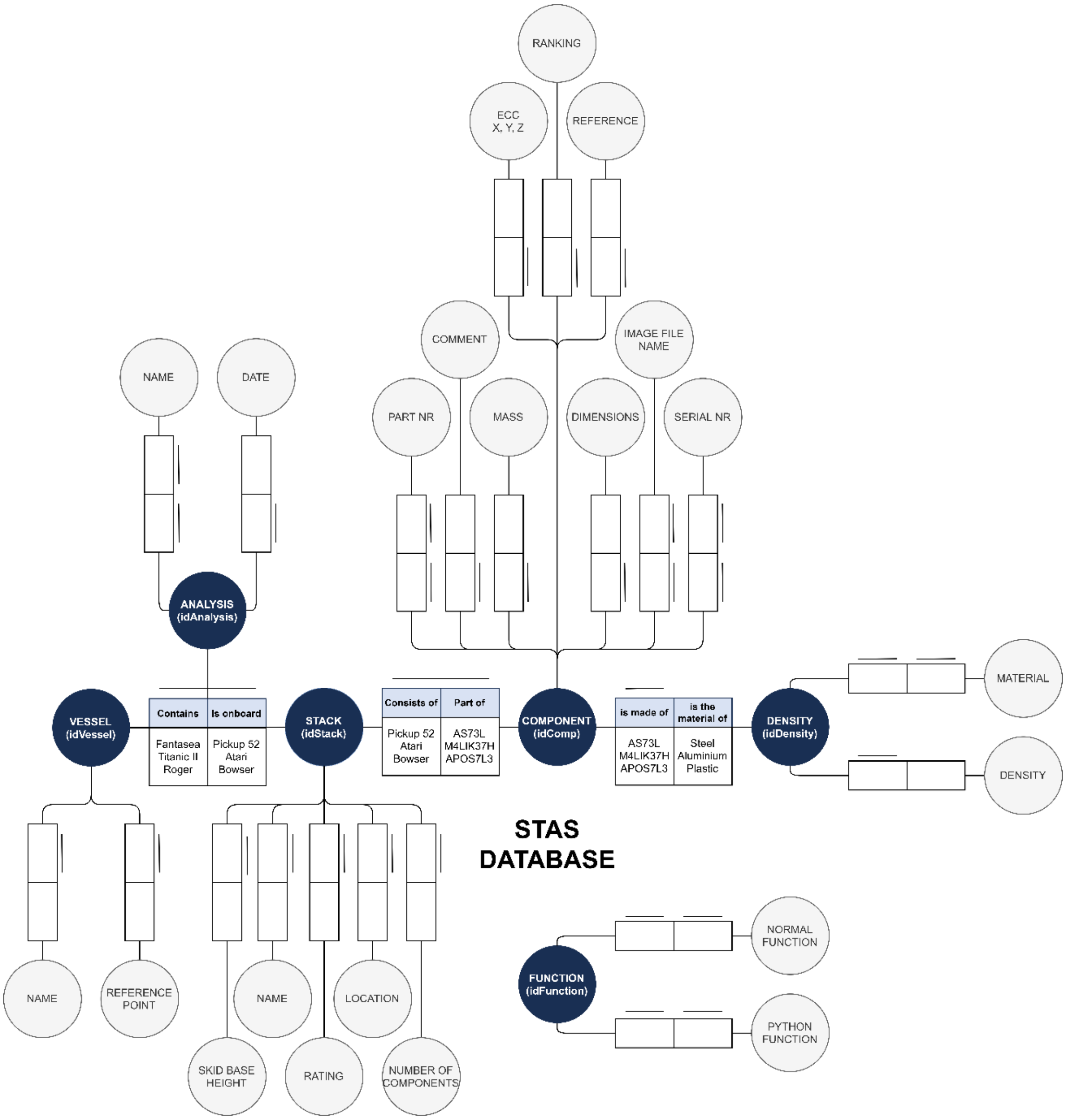


Figure 2020: ORM 1. Iteration

What is ORM?

Object-Role Modelling, or ORM for short, is a method for modelling a system at the conceptual level. The main purpose of this modelling techniques is to make sure information comes through to all parties with clarity and correctness. ORM is very easily readable even without any previous knowledge which makes it a great tool for team and client communication. [36]

To make the model we first found what type of data would be needed to be stored and if some of these had similar enough properties to group them together as entity types. Entity type so many different things but it is something that is similar enough to group together and store the same type of attributes about each data entry. Attributes is types of data we store about an entity, which is a single entry in a table [37] . In our first iteration we found six different entity types:

- Analysis
- Vessel

- Stack
- Component
- Density
- Function

All these entities are a part of the analysis in some way and can be seen on figure 20 where they are represented by dark blue circles. The attributes are represented by light grey circles and there can be a lot of them as you will be able to see.

Between all the circles in the model there are squares, some with text, some with a line above them and some with both. These squares and the lines are the ORM way of representing the relationship between multiple entities, and the text is there to give a little more context about the relationship. There are three different relationships:

One-to-One relationships

A one-to-one relationship in a database means that a row in table 1 can only link to one row in table 2, and a row in table 2 can only link to one row in table 1. [38] This type of relationship is represented with two separate lines above each square.

One-to-Many relationships

When one record in table 1 is connected to one or more entries in table 2, this is known as a one-to-many relationship. One entry in table 2 cannot, however, be connected to more than one record in table 1, and vice versa. [38] This type of relationship is represented with one line above one of the squares, and the side it is on represents the “many” side. In figure 20 above you can see that this is represented between the component type and the density type, meaning that a component can only have one density, but many components can share the same density.

Many-to-Many relationships

When more than one record in one table can be linked to more than one record in another table, this is called a "many-to-many" relationship. [38] This type of relationship is represented with one single line above both squares. In figure 20 above this relationship is present between the stack type and the component type. This means that one stack can consist of multiple components, and a component can be a part of multiple stacks.

6.6.1 ORM: 1. Iteration

The model in figure 20 above was our first iteration of a ORM model of the database. As a team we did find this very useful, because it cleared up multiple misunderstandings that might have been missed without the proper communication tools. It also made it clear for us what we did not have full understanding of yet and we were able to clear this up with TechnipFMC.

After feedback from our client and discussions between us as a group we found out that some parts where missing and some parts needed to be changed (see figure below).

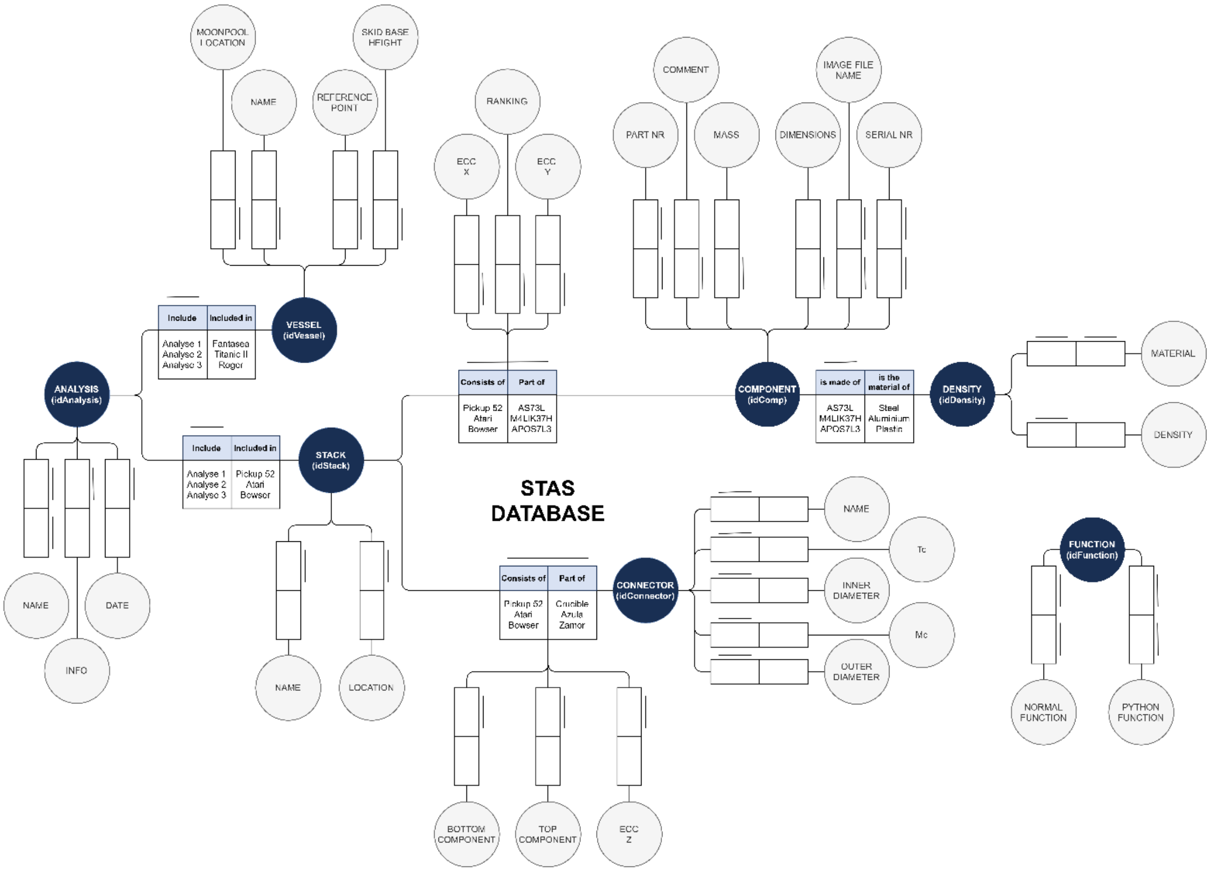


Figure 2121: ORM 2. Iteration

6.6.2 ORM: 2. Iteration

For our second iteration of the model there where some huge changes (see figure 21). Most of them where due to a lack of understanding of what the client needed but with good communication with TechnipFMC we did end up with a model that better represents the vision of the database. A whole new entity type was added, the connector type, which is the part that connects two components together. Every entity type got all the different attributes that would be needed for an analysis and then we could start the process of making the database.

6.6.3 Database Normalization

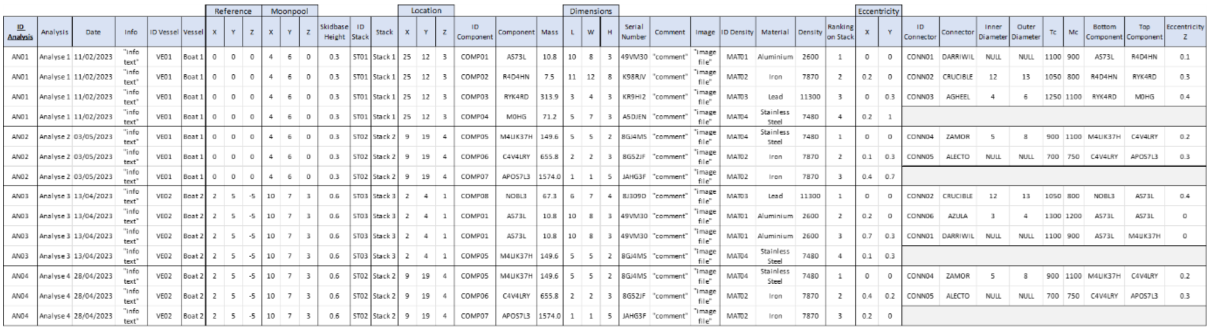
Relational database normalization is a systematic procedure for data management that aims to increase database flexibility by eliminating data redundancy and inconsistencies. By using the database normalization procedure, we make sure that the data is stored in a form that makes inserting, updating, and deleting the data the most efficient. A relational database relationship is frequently referred to as "normalized" if it follows the third normal form. [39] [40]

Normalization is a sequential procedure for organizing data in a consistent manner, which we followed to make our database. [41]

UNF

First, we start out with a database that is not normalized. To start the process of getting it normalized we used Excel to create headers for all the columns of data that would be used. After that we inserted test data, and the results is shown in table 5. ID Analysis has been set as the key for the table.

Table 5: All information related to the database.

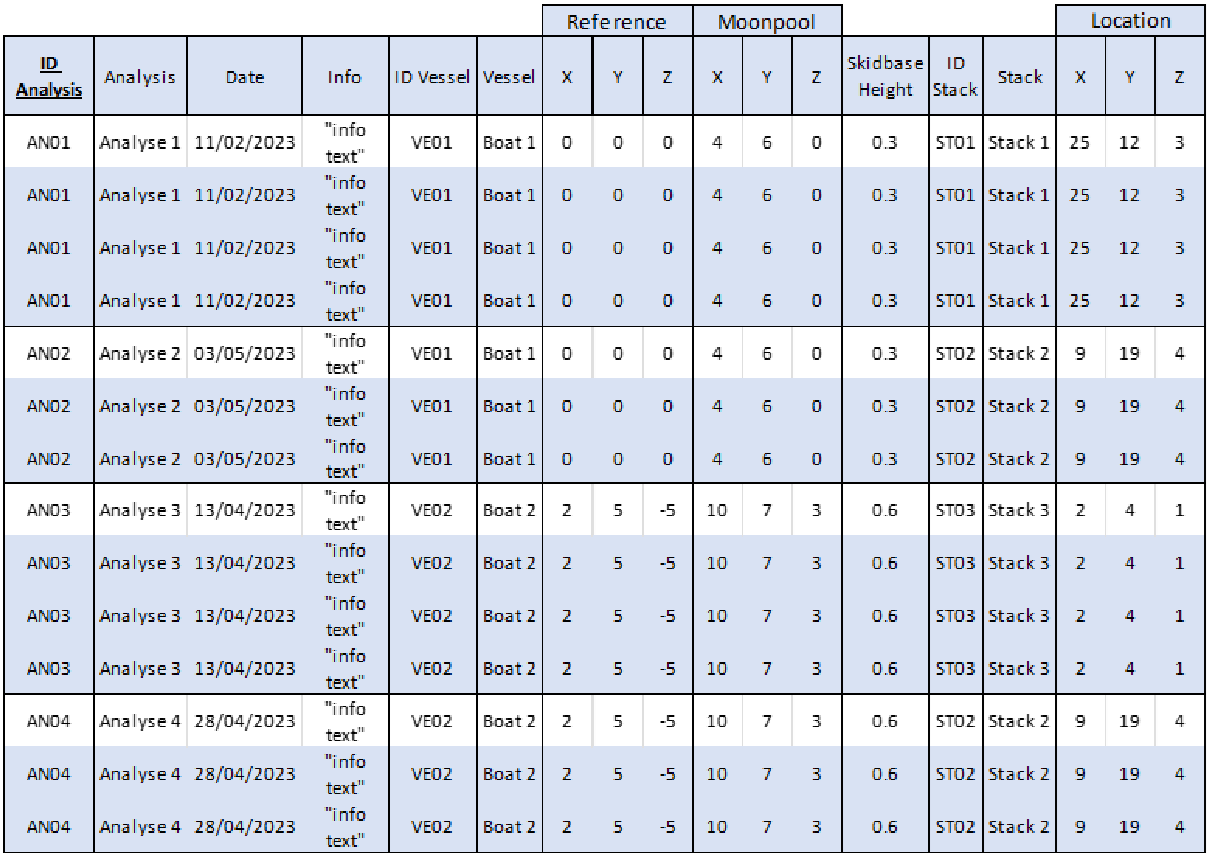


						Reference			Moonpool						Location						Dimensions										Eccentricity										
ID Analysis	Analysis	Date	Info	ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height	ID Stack	Stack	X	Y	Z	ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density	Material	Density	Ranking on Stack	X	Y	ID Connector	Connector	Inner Diameter	Outer Diameter	Tc	Mc	Bottom Component	Top Component	Eccentricity Z
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	1	0	0	CONN01	DARRIWIL	NULL	NULL	1100	900	AS73L	R4D4HN	0.1
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3	COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02	Iron	7870	2	0.2	0	CONN02	CRUCIBLE	12	13	1050	800	R4D4HN	RYK4RD	0.3
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3	COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03	Lead	11300	3	0	0.3	CONN03	AGHEEL	4	6	1250	1100	RYK4RD	M0HG	0.4
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3	COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04	Stainless Steel	7480	4	0.2	1									
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4	COMP05	M4UK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	1	0	0	CONN04	ZAMOR	5	8	900	1100	M4UK37H	C4V4LRY	0.2
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4	COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870	2	0.1	0.3	CONN05	ALECTO	NULL	NULL	700	750	C4V4LRY	APO57L3	0.3
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4	COMP07	APO57L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870	3	0.4	0.7									
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1	COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03	Lead	11300	1	0	0	CONN02	CRUCIBLE	12	13	1050	800	NOBL3	AS73L	0.4
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	2	0.2	0	CONN06	AZULA	3	4	1300	1200	AS73L	AS73L	0
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	3	0.7	0.3	CONN01	DARRIWIL	NULL	NULL	1100	900	AS73L	M4UK37H	0
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1	COMP05	M4UK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	4	0.1	0.3									
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4	COMP05	M4UK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	1	0	0	CONN04	ZAMOR	5	8	900	1100	M4UK37H	C4V4LRY	0.2
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4	COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870	2	0.4	0.2	CONN05	ALECTO	NULL	NULL	700	750	C4V4LRY	APO57L3	0.3
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4	COMP07	APO57L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870	3	0.2	0									

1NF

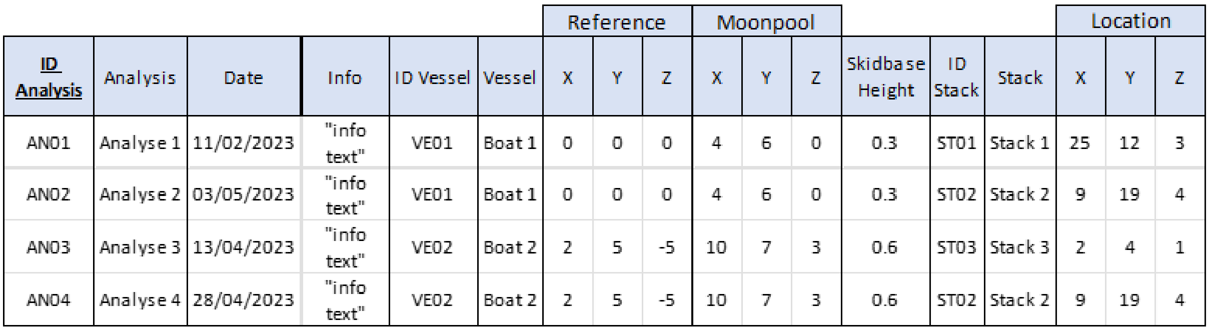
To get to the first form of normalization, we will have to remove duplicated rows of data. Shown below is a great example of duplicated information. This will be moved out to a separate table, but we will save the key.

Table 6: Duplicated data



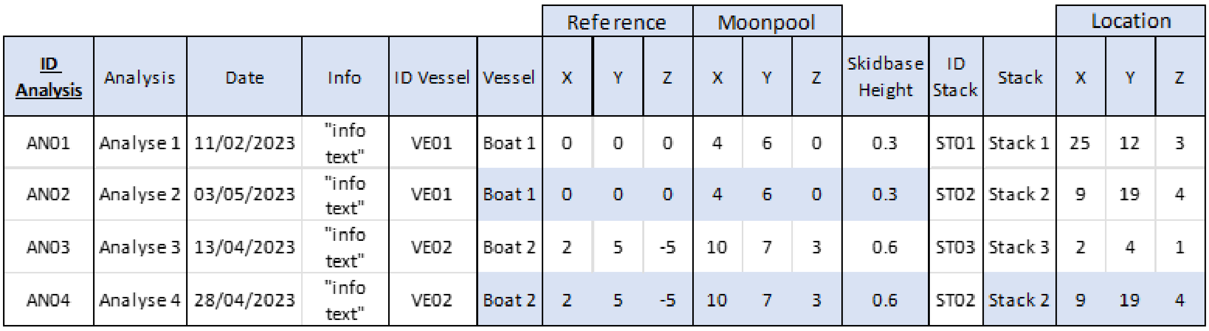
						Reference			Moonpool						Location		
ID Analysis	Analysis	Date	Info	ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height	ID Stack	Stack	X	Y	Z
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4

Table 7: New table



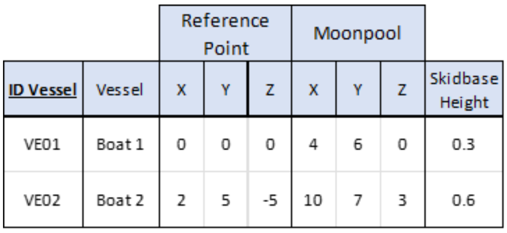
						Reference			Moonpool						Location		
ID Analysis	Analysis	Date	Info	ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height	ID Stack	Stack	X	Y	Z
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4

Table 8: Still some data being repeated.



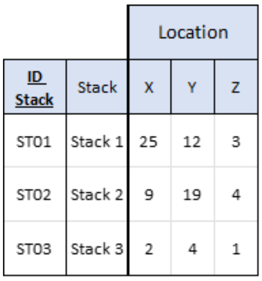
						Reference			Moonpool						Location		
ID Analysis	Analysis	Date	Info	ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height	ID Stack	Stack	X	Y	Z
AN01	Analyse 1	11/02/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST01	Stack 1	25	12	3
AN02	Analyse 2	03/05/2023	"info text"	VE01	Boat 1	0	0	0	4	6	0	0.3	ST02	Stack 2	9	19	4
AN03	Analyse 3	13/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST03	Stack 3	2	4	1
AN04	Analyse 4	28/04/2023	"info text"	VE02	Boat 2	2	5	-5	10	7	3	0.6	ST02	Stack 2	9	19	4

Table 9: Vessel table



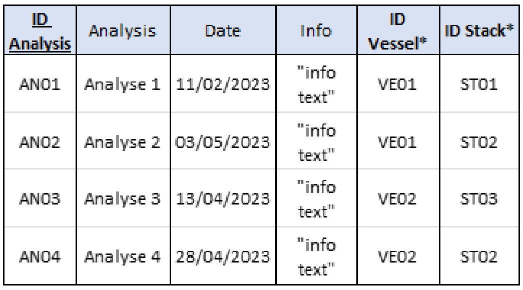
		Reference Point			Moonpool			
ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height
VE01	Boat 1	0	0	0	4	6	0	0.3
VE02	Boat 2	2	5	-5	10	7	3	0.6

Table 10: Stack table



		Location		
ID Stack	Stack	X	Y	Z
ST01	Stack 1	25	12	3
ST02	Stack 2	9	19	4
ST03	Stack 3	2	4	1

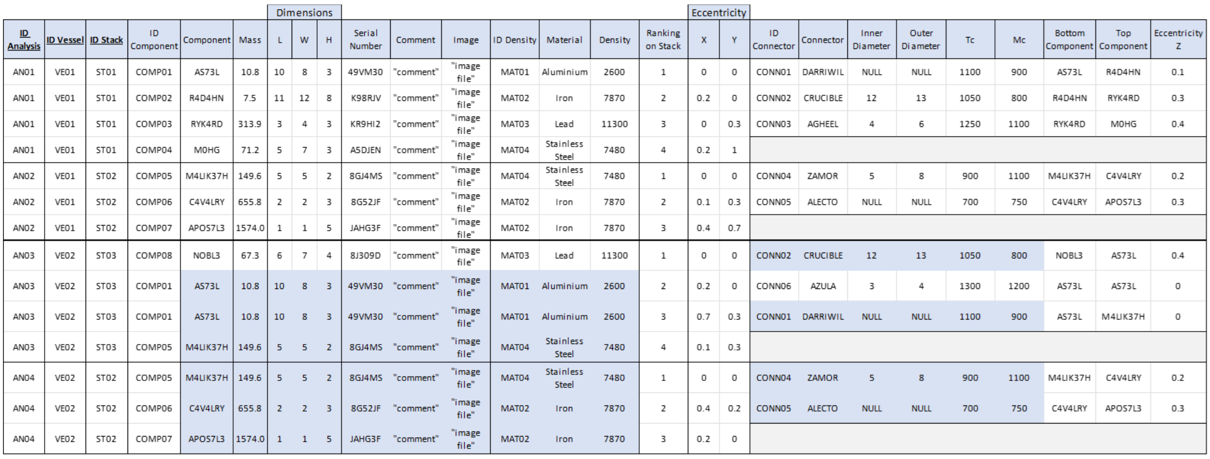
Table 11: Analysis table with a copy of the code from the keys of vessel and stack



ID Analysis	Analysis	Date	Info	ID Vessel*	ID Stack*
AN01	Analyse 1	11/02/2023	"info text"	VE01	ST01
AN02	Analyse 2	03/05/2023	"info text"	VE01	ST02
AN03	Analyse 3	13/04/2023	"info text"	VE02	ST03
AN04	Analyse 4	28/04/2023	"info text"	VE02	ST02

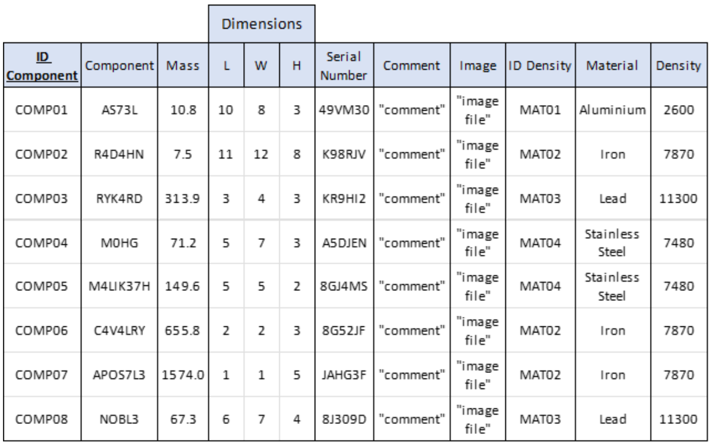
In the tables above *(table 5 to 11)* we have started with identifying the duplicate information, then we have moved this out to a separate table, but we have left a copy of the key. This was done throughout the entire table we got our database to the first normal form. Below is the same procedure done with the rest of the table.

Table 12: More duplicated data.



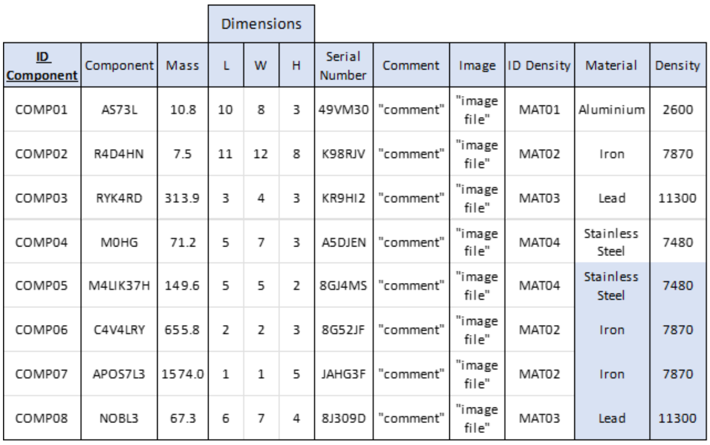
						Dimensions										Eccentricity										
ID Analysis	**ID Vessel**	**ID Stack**	ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density	Material	Density	Ranking on Stack	X	Y	ID Connector	Connector	Inner Diameter	Outer Diameter	Tc	Mc	Bottom Component	Top Component	Eccentricity Z
AN01	VE01	ST01	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	1	0	0	CONN01	DARRIWIL	NULL	NULL	1100	900	AS73L	R4D4HN	0.1
AN01	VE01	ST01	COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02	Iron	7870	2	0.2	0	CONN02	CRUCIBLE	12	13	1050	800	R4D4HN	RYK4RD	0.3
AN01	VE01	ST01	COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03	Lead	11300	3	0	0.3	CONN03	AGHEEL	4	6	1250	1100	RYK4RD	M0HG	0.4
AN01	VE01	ST01	COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04	Stainless Steel	7480	4	0.2	1									
AN02	VE01	ST02	COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	1	0	0	CONN04	ZAMOR	5	8	900	1100	M4LIK37H	C4V4LRY	0.2
AN02	VE01	ST02	COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870	2	0.1	0.3	CONN05	ALECTO	NULL	NULL	700	750	C4V4LRY	APOS7L3	0.3
AN02	VE01	ST02	COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870	3	0.4	0.7									
AN03	VE02	ST03	COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03	Lead	11300	1	0	0	CONN02	CRUCIBLE	12	13	1050	800	NOBL3	AS73L	0.4
AN03	VE02	ST03	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	2	0.2	0	CONN06	AZULA	3	4	1300	1200	AS73L	AS73L	0
AN03	VE02	ST03	COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600	3	0.7	0.3	CONN01	DARRIWIL	NULL	NULL	1100	900	AS73L	M4LIK37H	0
AN03	VE02	ST03	COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	4	0.1	0.3									
AN04	VE02	ST02	COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480	1	0	0	CONN04	ZAMOR	5	8	900	1100	M4LIK37H	C4V4LRY	0.2
AN04	VE02	ST02	COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870	2	0.4	0.2	CONN05	ALECTO	NULL	NULL	700	750	C4V4LRY	APOS7L3	0.3
AN04	VE02	ST02	COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870	3	0.2	0									

Table 13: New table moved out of the original



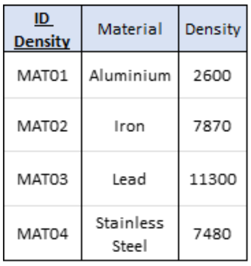
			Dimensions								
ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density	Material	Density
COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600
COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02	Iron	7870
COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03	Lead	11300
COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04	Stainless Steel	7480
COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480
COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870
COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870
COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03	Lead	11300

Table 14: Still a few duplicates



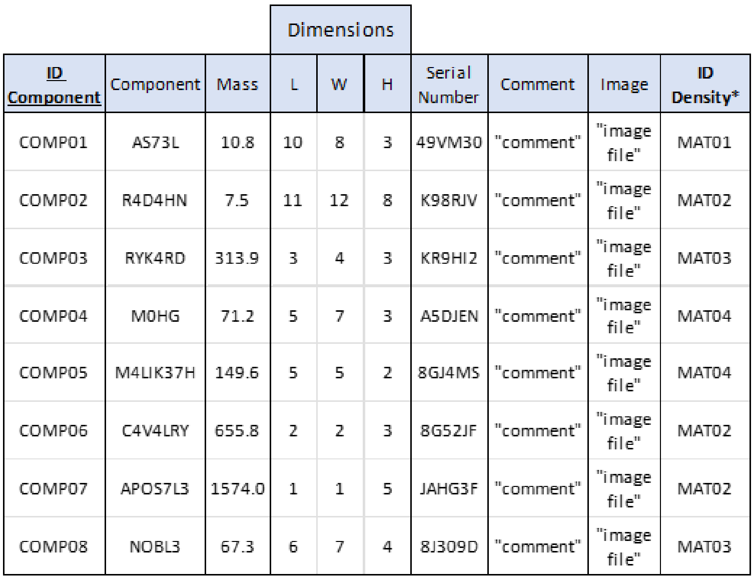
			Dimensions								
ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density	Material	Density
COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01	Aluminium	2600
COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02	Iron	7870
COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03	Lead	11300
COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04	Stainless Steel	7480
COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04	Stainless Steel	7480
COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02	Iron	7870
COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02	Iron	7870
COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03	Lead	11300

Table 15: Density table



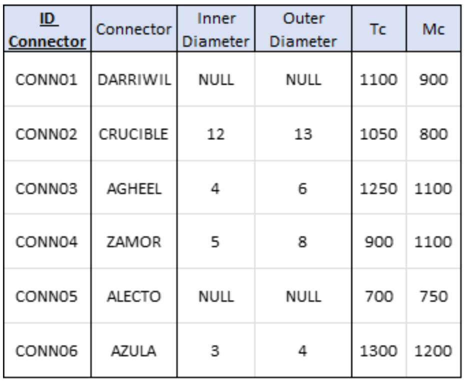
ID Density	Material	Density
MAT01	Aluminium	2600
MAT02	Iron	7870
MAT03	Lead	11300
MAT04	Stainless Steel	7480

Table 16: Component table, with a copy of the key to the density table



			Dimensions						
ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density*
COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01
COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02
COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03
COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04
COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04
COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02
COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02
COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03

Table 17: Connector table

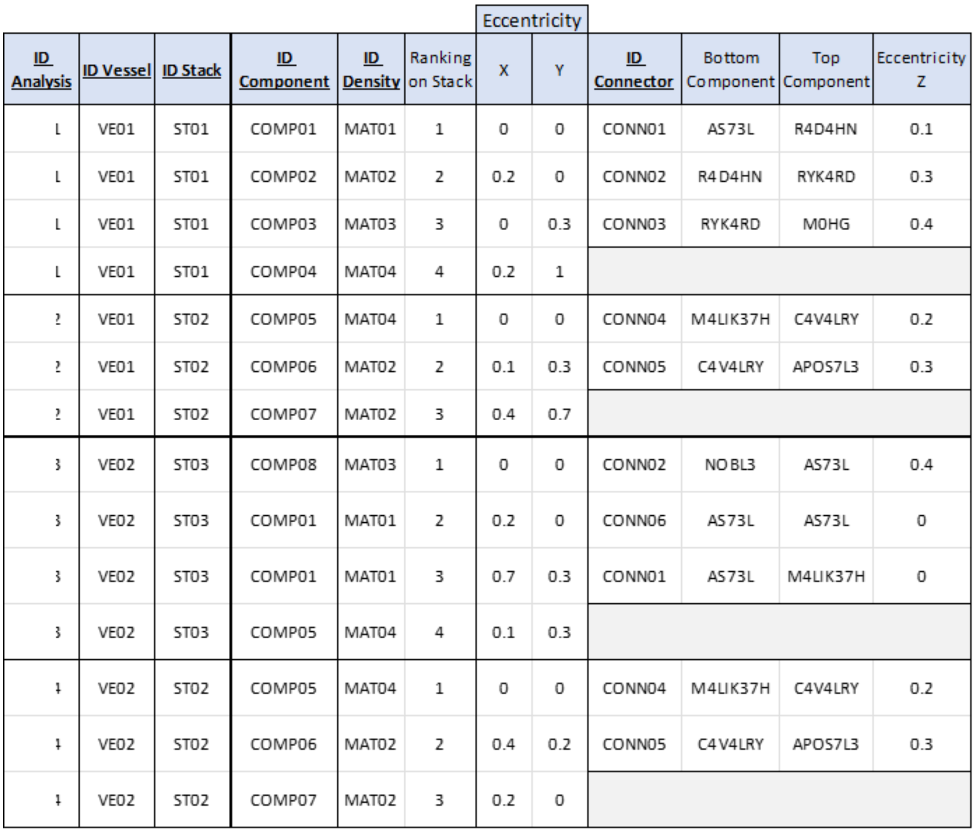


ID Connector	Connector	Inner Diameter	Outer Diameter	Tc	Mc
CONN01	DARRIWIL	NULL	NULL	1100	900
CONN02	CRUCIBLE	12	13	1050	800
CONN03	AGHEEL	4	6	1250	1100
CONN04	ZAMOR	5	8	900	1100
CONN05	ALECTO	NULL	NULL	700	750
CONN06	AZULA	3	4	1300	1200

2NF

Now that we have our database in the first normal form, we want to proceed to get it to the second normal form. To get to the second normal form we need to make sure that everything that is not dependent on the entire key is moved out. Entire key meaning a combination of keys that form a new key. After the last step we are left with this:

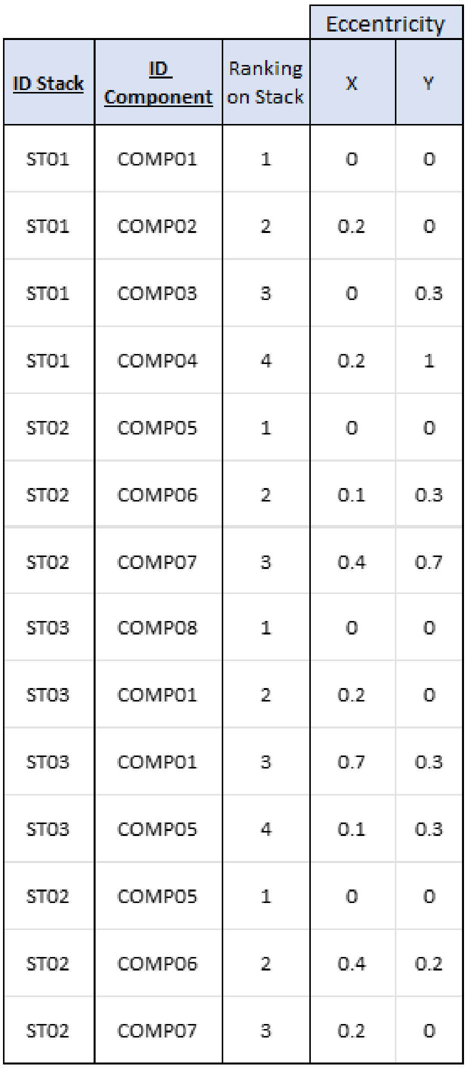
Table 18: What is left after 1NF.



						Eccentricity					
ID Analysis	ID Vessel	ID Stack	ID Component	ID Density	Ranking on Stack	X	Y	ID Connector	Bottom Component	Top Component	Eccentricity Z
1	VE01	ST01	COMP01	MAT01	1	0	0	CONN01	AS73L	R4D4HN	0.1
1	VE01	ST01	COMP02	MAT02	2	0.2	0	CONN02	R4D4HN	RYK4RD	0.3
1	VE01	ST01	COMP03	MAT03	3	0	0.3	CONN03	RYK4RD	M0HG	0.4
1	VE01	ST01	COMP04	MAT04	4	0.2	1				
2	VE01	ST02	COMP05	MAT04	1	0	0	CONN04	M4LIK37H	C4V4LRY	0.2
2	VE01	ST02	COMP06	MAT02	2	0.1	0.3	CONN05	C4V4LRY	APOS7L3	0.3
2	VE01	ST02	COMP07	MAT02	3	0.4	0.7				
3	VE02	ST03	COMP08	MAT03	1	0	0	CONN02	NOBL3	AS73L	0.4
3	VE02	ST03	COMP01	MAT01	2	0.2	0	CONN06	AS73L	AS73L	0
3	VE02	ST03	COMP01	MAT01	3	0.7	0.3	CONN01	AS73L	M4LIK37H	0
3	VE02	ST03	COMP05	MAT04	4	0.1	0.3				
4	VE02	ST02	COMP05	MAT04	1	0	0	CONN04	M4LIK37H	C4V4LRY	0.2
4	VE02	ST02	COMP06	MAT02	2	0.4	0.2	CONN05	C4V4LRY	APOS7L3	0.3
4	VE02	ST02	COMP07	MAT02	3	0.2	0				

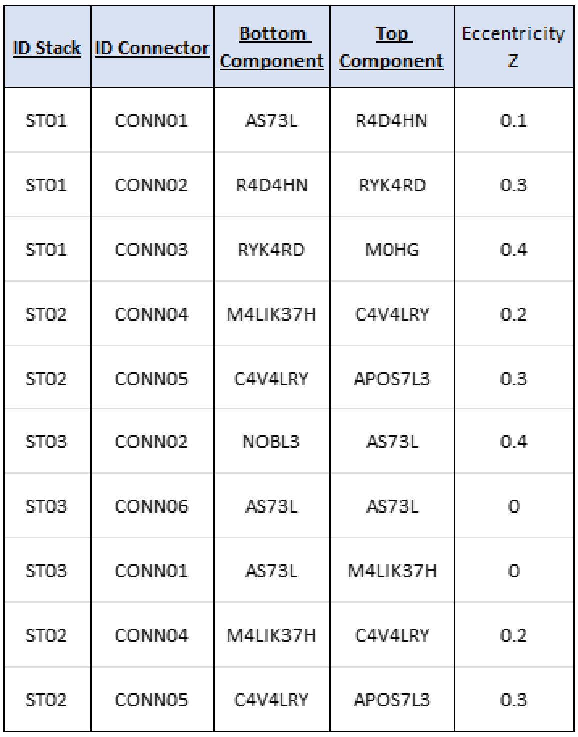
Within the component part of the table, we have the attributes ranking, eccentricity x and eccentricity y that is not dependent on the entire key, but they are dependent on more than one key, ID Stack, and ID Component. This is because a component can be on multiple stacks, and its placement will probably change. We can see the same happening in the connector part of the table but here it is even more keys involved. ID Connector, Bottom Component, Top Component, and ID stack will decide the eccentricity in z for the connector. Bottom component and top component are represented with part nr to make it easier to see what connector is placed where, but in the actual table this will be the id of the component. We moved these out, and we will make foreign keys wherever there is a copy of a key to be able to get more information about the entities just from the keys.

Table 19: Stack configuration table



			Eccentricity	
ID Stack	**ID Component**	Ranking on Stack	X	Y
ST01	COMP01	1	0	0
ST01	COMP02	2	0.2	0
ST01	COMP03	3	0	0.3
ST01	COMP04	4	0.2	1
ST02	COMP05	1	0	0
ST02	COMP06	2	0.1	0.3
ST02	COMP07	3	0.4	0.7
ST03	COMP08	1	0	0
ST03	COMP01	2	0.2	0
ST03	COMP01	3	0.7	0.3
ST03	COMP05	4	0.1	0.3
ST02	COMP05	1	0	0
ST02	COMP06	2	0.4	0.2
ST02	COMP07	3	0.2	0

Table 20: Connector configuration table



ID Stack	ID Connector	Bottom Component	Top Component	Eccentricity Z
ST01	CONN01	AS73L	R4D4HN	0.1
ST01	CONN02	R4D4HN	RYK4RD	0.3
ST01	CONN03	RYK4RD	M0HG	0.4
ST02	CONN04	M4LIK37H	C4V4LRY	0.2
ST02	CONN05	C4V4LRY	APOS7L3	0.3
ST03	CONN02	NOBL3	AS73L	0.4
ST03	CONN06	AS73L	AS73L	0
ST03	CONN01	AS73L	M4LIK37H	0
ST02	CONN04	M4LIK37H	C4V4LRY	0.2
ST02	CONN05	C4V4LRY	APOS7L3	0.3

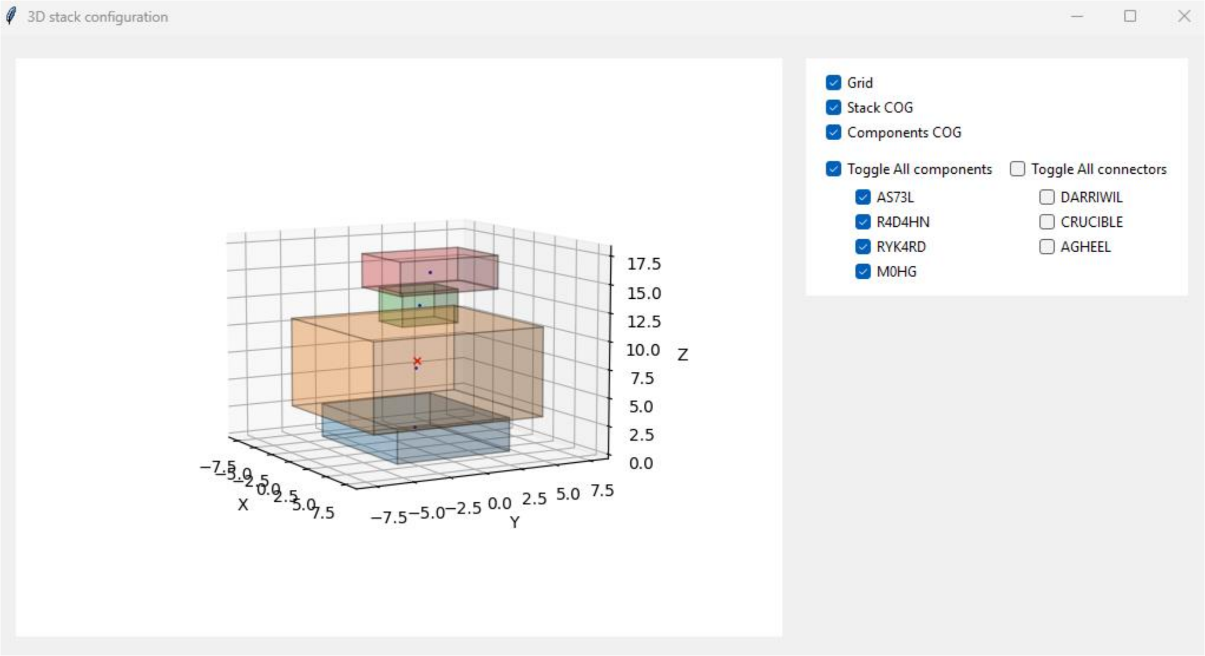
3NF

The last step of database normalization is to check if there is anything that is only dependent on the key, but there is nothing left which means that we reached the third and final normal form. A full view of all the finished tables can be found in Appendix M.

6.7 Interactive 3D Visualization of Stack

In our software we have set aside a separate page for 3D visualization of the stack used in the analysis. This page is meant to make it easier to see the configuration and will make it easy to spot possible mistakes. Detailed explanation of the code is commented in the 3D visualisation code.

The configuration shown in figure 22 below is easily manipulated in the software. With the use of your left mouse button, you can drag it around to see all different sides to of the stack, and the right mouse button will let you get a closer look at all the different parts you want to take a closer look at.



3D stack configuration

☑ Grid

☑ Stack COG

☑ Components COG

☑ Toggle All components ☐ Toggle All connectors

☑ AS73L	☐ DARRIWIL
☑ R4D4HN	☐ CRUCIBLE
☑ RYK4RD	☐ AGHEEL
☑ M0HG	

Figure 2222: Snippet from software

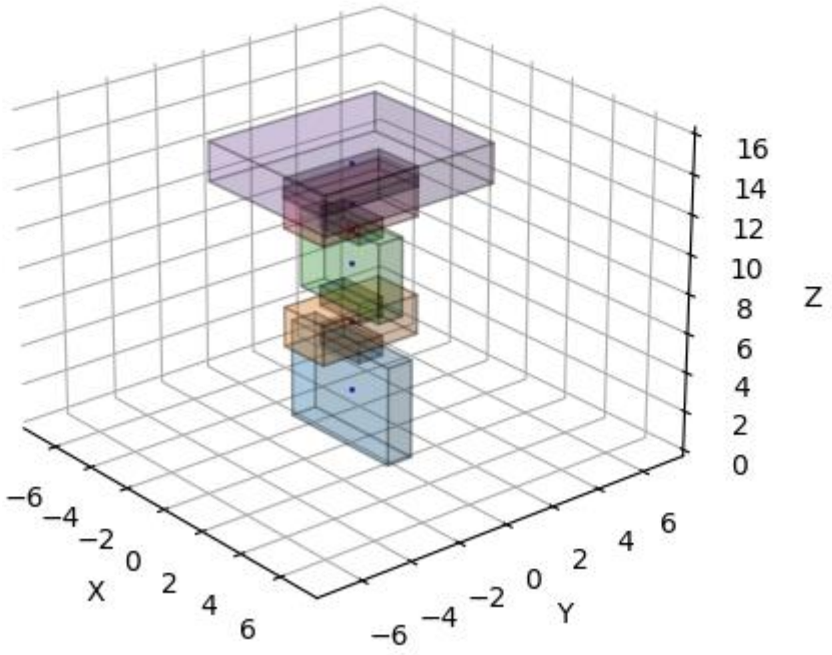


Figure 2323: Stack consisting of 5 components with 4 connectors between them.

6.7.1 Components

The components of the stack is the main building blocks for this visualisation, and all the data about them is contained in a local database on the user's computer. The 3D visualisation will retrieve the data from the database about the component's dimensions, their eccentricity in the x and the y plane and density to make the recreation of the stack as close to real life as possible.

The visualisation window is created by stacking the components on top of each other from the bottom and up. The eccentricity values of x and y plane of the component will determine its offset from the centre of the grid, since not all components are lined perfectly on top of each other.

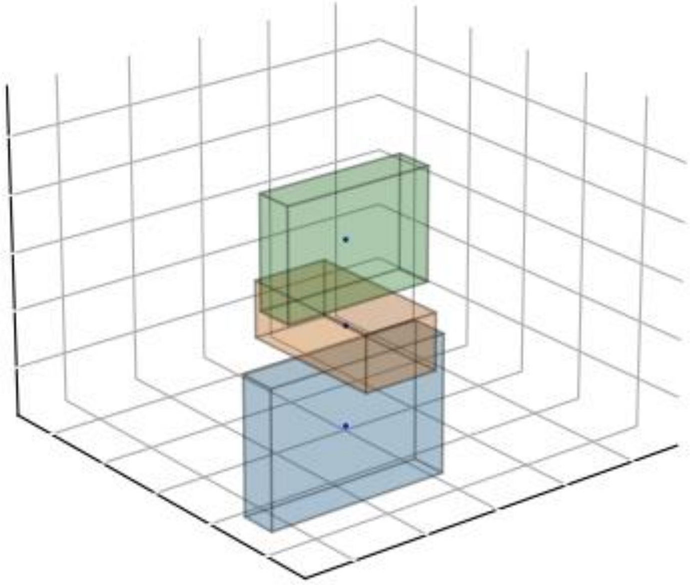


Figure 2424: Showing 3 of the 5 components on the stack.

6.7.2 Connectors

Connectors is also displayed in the 3D figure, but they are there purely for the visuals. You will be able to see all information about them on the software page, but their shape is not made from their data. The sole purpose of the connectors in the figure is to show the user what components the connector is connecting to each other.

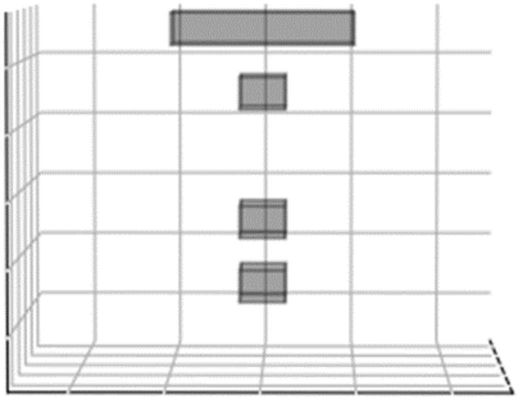


Figure 2525: Showing just the connectors on the stack.

6.7.3 Centre of Gravity

Centre of Gravity (CoG) is also shown in the 3D visualisation window. There are two different markers, blue dots are representing the CoG for each individual component and a red X are representing the CoG of the entire stack as a whole.

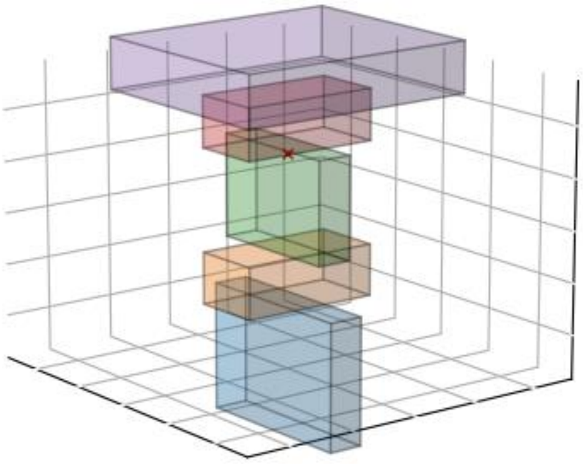


Figure 2626: Showing COG for the entire stack.

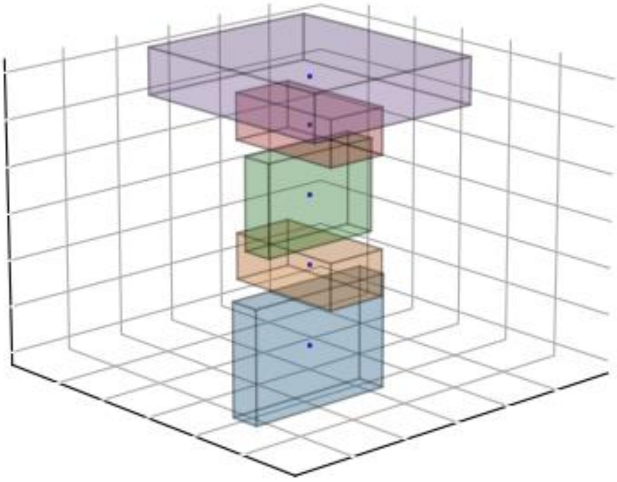
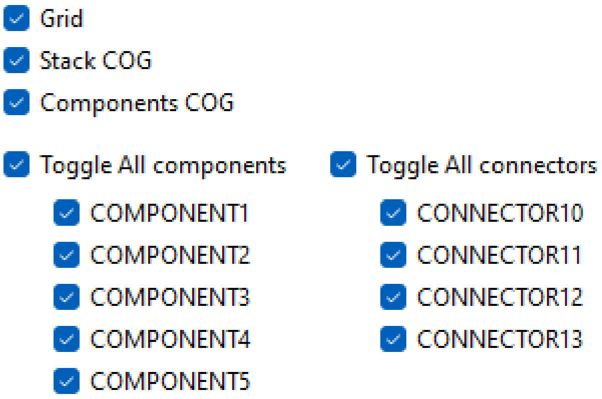


Figure 2727: Showing COG for each component on the stack.

6.7.3 Filters

With a lot of components, connectors, and markers the window can get a little crowded, therefore we have implemented some checkboxes where the user can check of exactly what they choose to see in the figure window. As you can see in figure 28 below there are a couple of choices to make sure the user can see the configuration is as much detail as possible.



☑ Grid
☑ Stack COG
☑ Components COG

☑ Toggle All components
- ☑ COMPONENT1
- ☑ COMPONENT2
- ☑ COMPONENT3
- ☑ COMPONENT4
- ☑ COMPONENT5

☑ Toggle All connectors
- ☑ CONNECTOR10
- ☑ CONNECTOR11
- ☑ CONNECTOR12
- ☑ CONNECTOR13

Figure 2828: Showing the filters for the 3D figure.

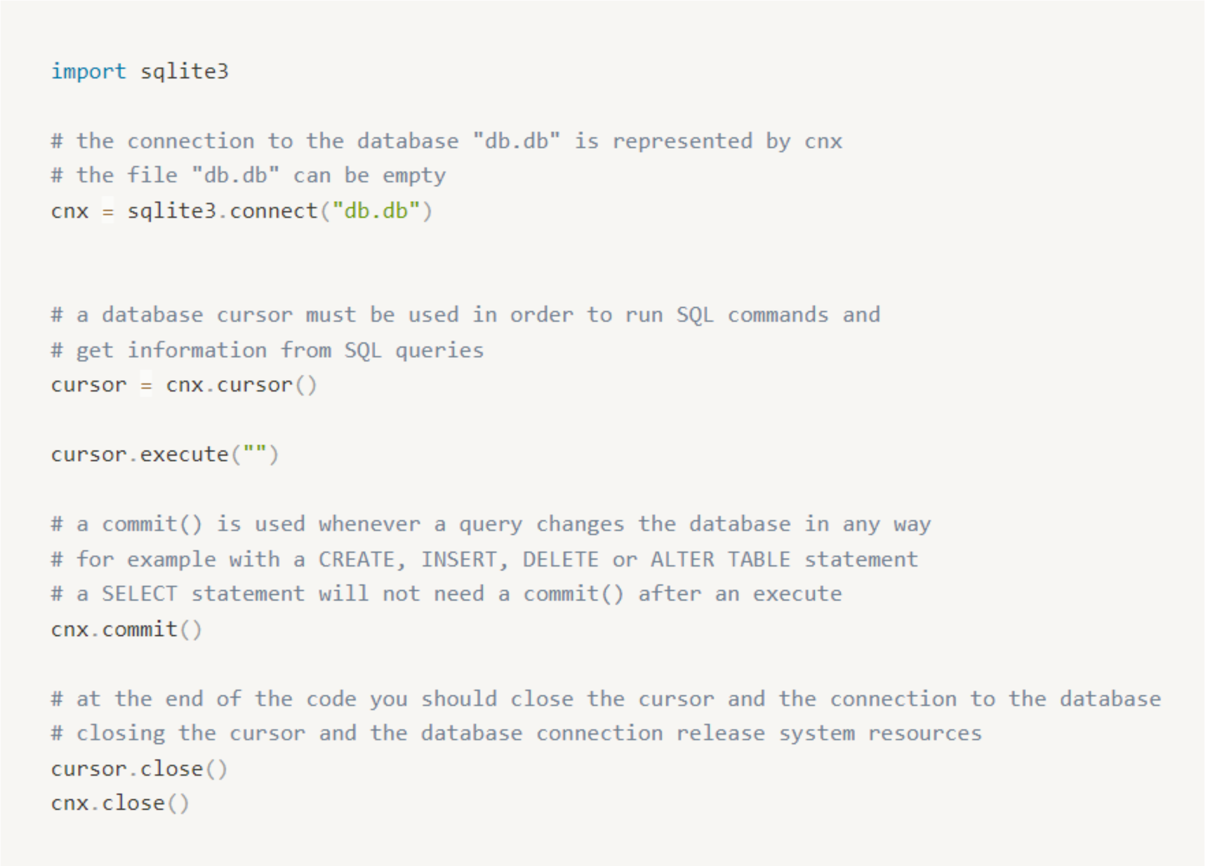
- **Grid:** toggles the grid in the background of the stack configuration.
- **Stack COG:** toggles the X marking the centre of gravity for the entire stack configuration.
- **Component COG:** toggles the dots marking the centre of gravity for each individual component.
- **Toggle all components** and **toggle all connectors** does exactly that.

6.7.4 Libraries

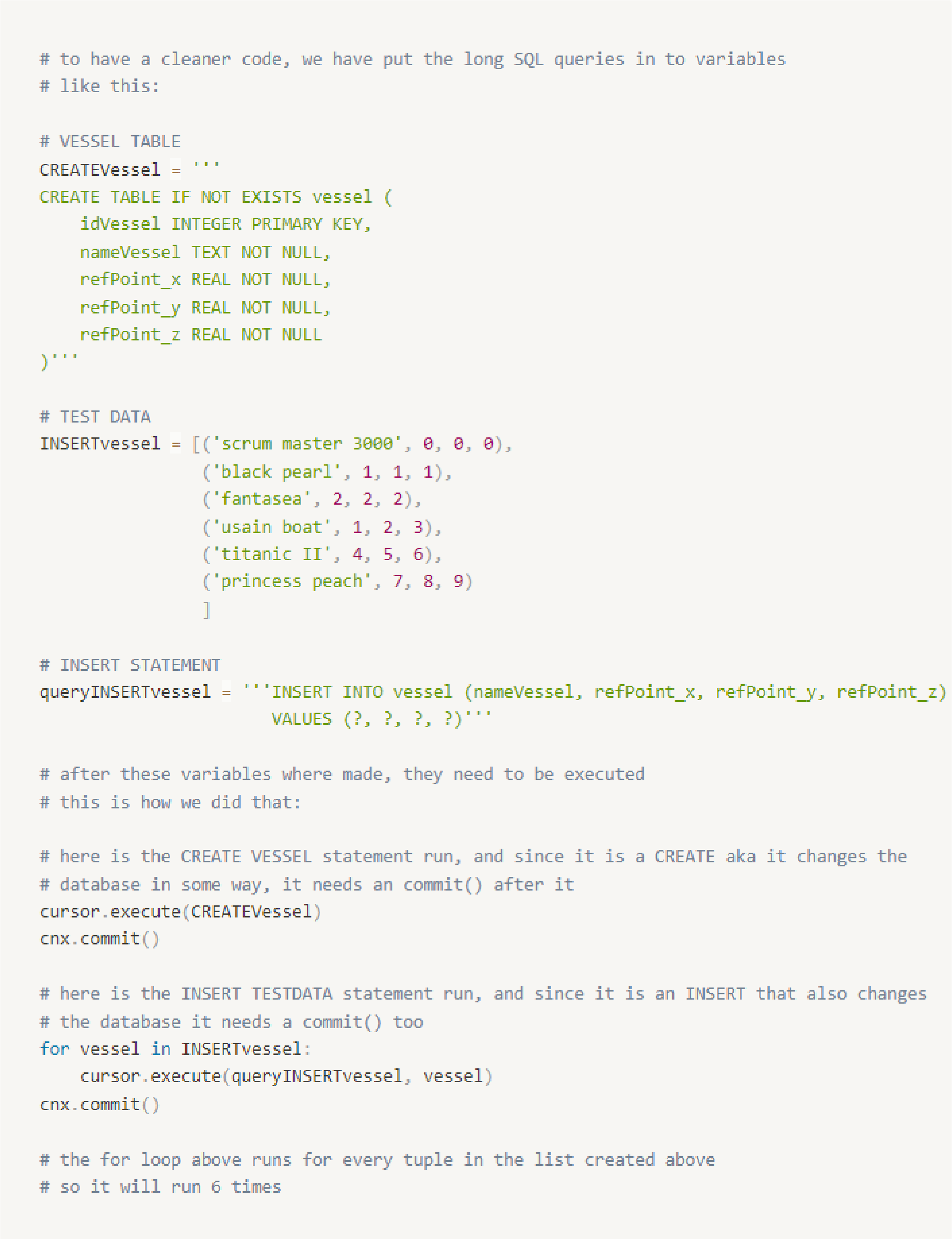
For the 3D Visualisation of the stack there have been used a couple of different libraries. For database connectivity and data retrieval SQLite3 have been used together with Pandas for the visual part of the database. Matplotlib and NumPy has been used for making the figure and calculating data throughout the code.

SQLite3

SQLite is a SQL database engine that works within a software and does therefor not need a server or any configuration. Instead, SQLite reads and writes straight to and from regular disk files, which is exactly what our client wants. Despite its power, SQLite is very little, with a library size of less than 750KiB, which will help with speed. It goes through extensive testing before each release, establishing a good reputation for its dependability and everyone is welcome to use the public-domain SQLite source code for any use. [42] [43]

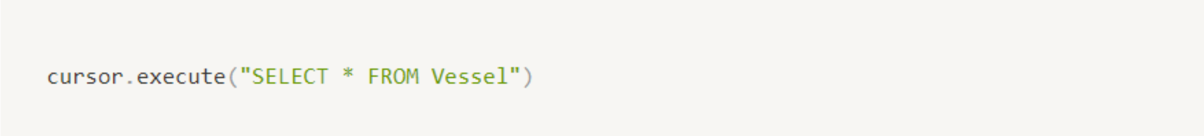


```
import sqlite3

# the connection to the database "db.db" is represented by cnx
# the file "db.db" can be empty
cnx = sqlite3.connect("db.db")


# a database cursor must be used in order to run SQL commands and
# get information from SQL queries
cursor = cnx.cursor()

cursor.execute("")

# a commit() is used whenever a query changes the database in any way
# for example with a CREATE, INSERT, DELETE or ALTER TABLE statement
# a SELECT statement will not need a commit() after an execute
cnx.commit()

# at the end of the code you should close the cursor and the connection to the database
# closing the cursor and the database connection release system resources
cursor.close()
cnx.close()
```



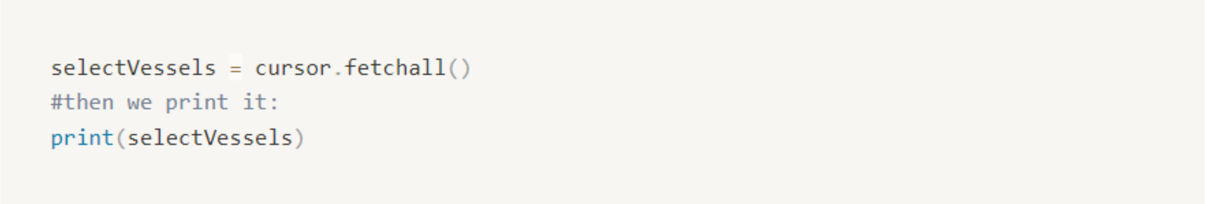
```
# to have a cleaner code, we have put the long SQL queries in to variables
# like this:

# VESSEL TABLE
CREATEVessel = '''
CREATE TABLE IF NOT EXISTS vessel (
    idVessel INTEGER PRIMARY KEY,
    nameVessel TEXT NOT NULL,
    refPoint_x REAL NOT NULL,
    refPoint_y REAL NOT NULL,
    refPoint_z REAL NOT NULL
)'''

# TEST DATA
INSERTvessel = [('scrum master 3000', 0, 0, 0),
                ('black pearl', 1, 1, 1),
                ('fantasea', 2, 2, 2),
                ('usain boat', 1, 2, 3),
                ('titanic II', 4, 5, 6),
                ('princess peach', 7, 8, 9)
                ]

# INSERT STATEMENT
queryINSERTvessel = '''INSERT INTO vessel (nameVessel, refPoint_x, refPoint_y, refPoint_z)
                       VALUES (?, ?, ?, ?)'''

# after these variables where made, they need to be executed
# this is how we did that:

# here is the CREATE VESSEL statement run, and since it is a CREATE aka it changes the
# database in some way, it needs an commit() after it
cursor.execute(CREATEVessel)
cnx.commit()

# here is the INSERT TESTDATA statement run, and since it is an INSERT that also changes
# the database it needs a commit() too
for vessel in INSERTvessel:
    cursor.execute(queryINSERTvessel, vessel)
cnx.commit()

# the for loop above runs for every tuple in the list created above
# so it will run 6 times
```

To make sure the test data is correctly committed to the database we used SELECT statements to visualise the data. Let’s check the vessel table:



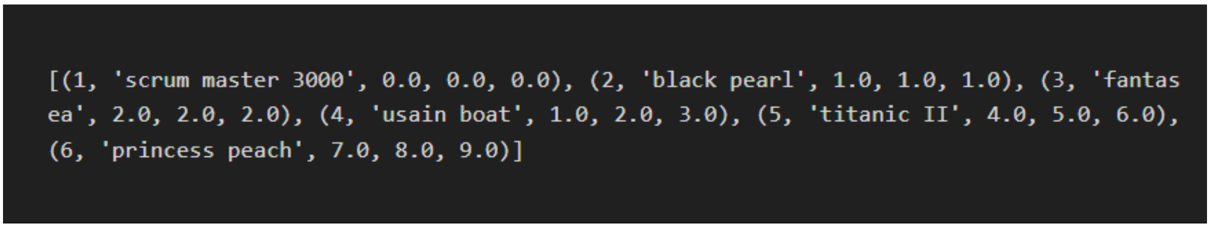
```
cursor.execute("SELECT * FROM Vessel")
```

For the data that is being selected to be viewed it needs to be stored in a variable. fetchall() takes the data that was retrieved by the SELECT statement in the execute and makes it possible for us to store it like this:



```
selectVessels = cursor.fetchall()
#then we print it:
print(selectVessels)
```

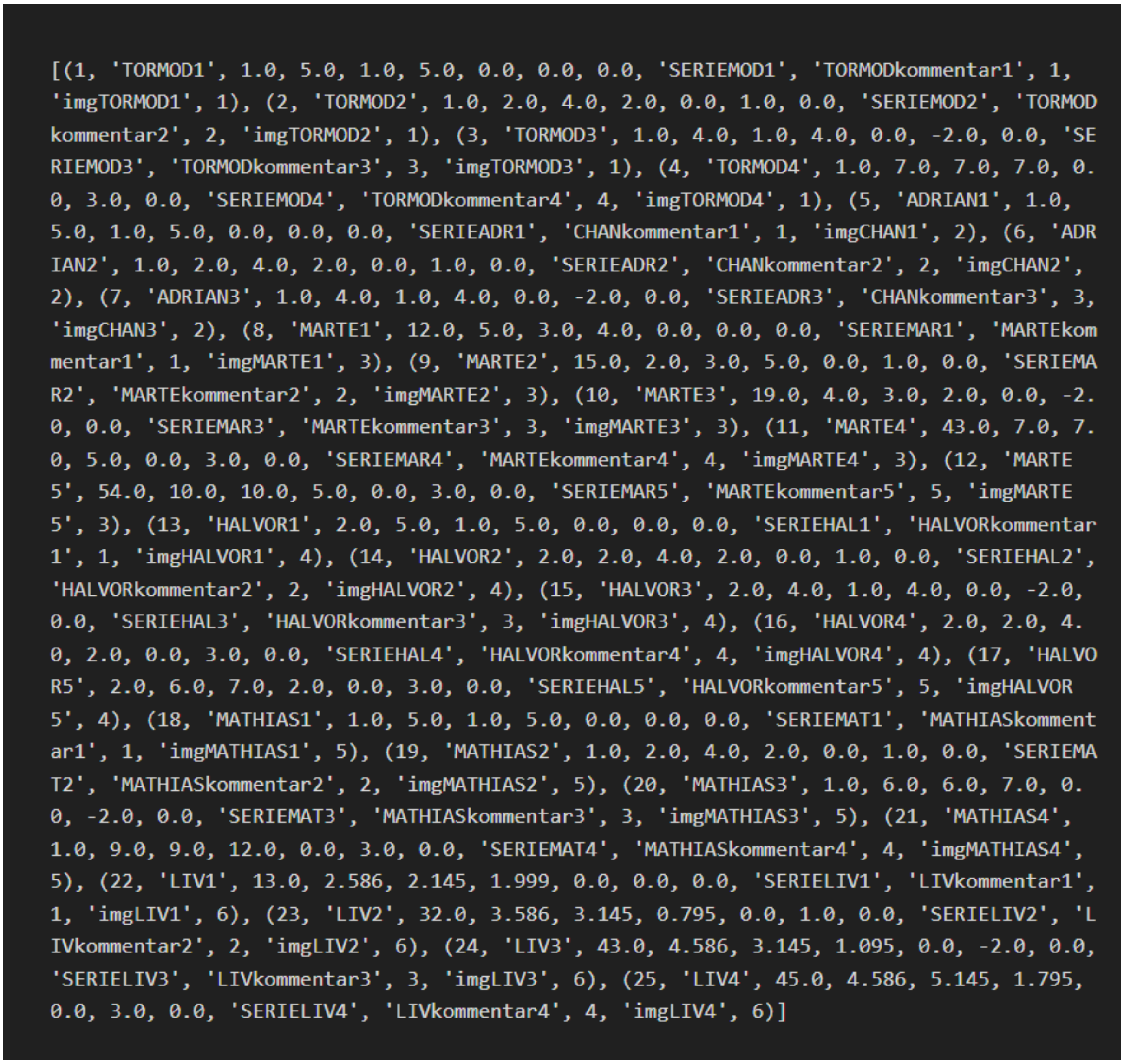
The result in the terminal will look like this:



```
[(1, 'scrum master 3000', 0.0, 0.0, 0.0), (2, 'black pearl', 1.0, 1.0, 1.0), (3, 'fantas
ea', 2.0, 2.0, 2.0), (4, 'usain boat', 1.0, 2.0, 3.0), (5, 'titanic II', 4.0, 5.0, 6.0),
(6, 'princess peach', 7.0, 8.0, 9.0)]
```

The output is okey when there is very little data in the table, but with bigger tables with more data this will be a horrible way to check for errors or even just to display.

To demonstrate, this would be the output from our component table:

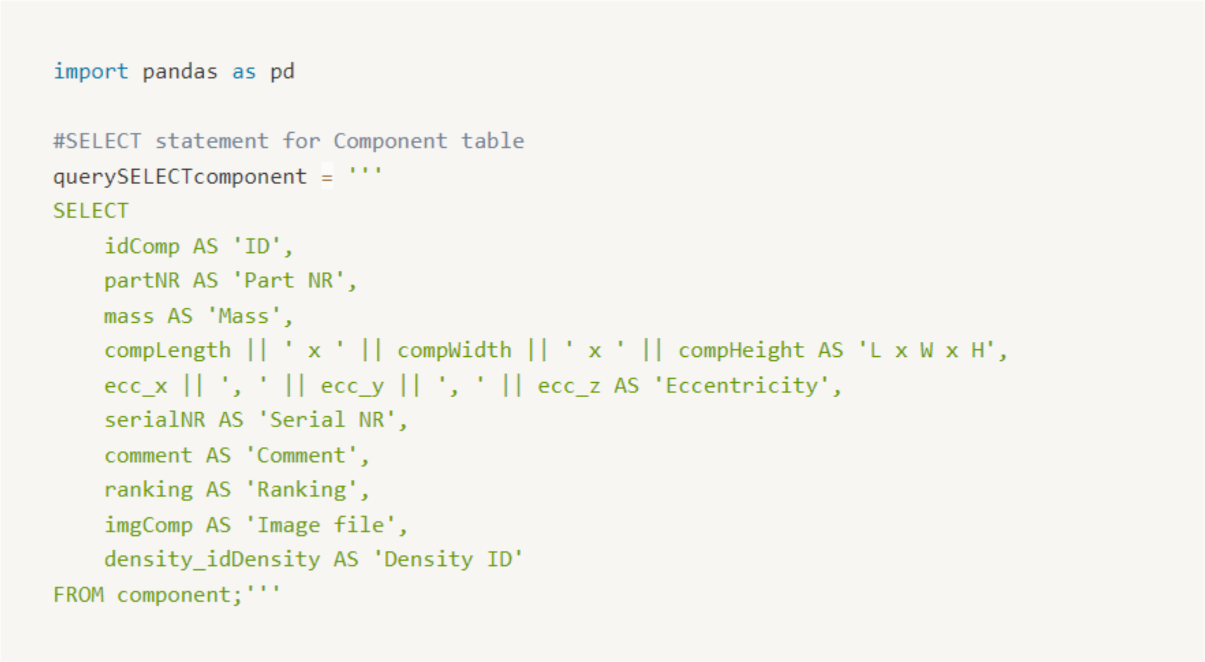


```
[(1, 'TORMOD1', 1.0, 5.0, 1.0, 5.0, 0.0, 0.0, 0.0, 'SERIEMOD1', 'TORMODkommentar1', 1,
'imgTORMOD1', 1), (2, 'TORMOD2', 1.0, 2.0, 4.0, 2.0, 0.0, 1.0, 0.0, 'SERIEMOD2', 'TORMOD
kommentar2', 2, 'imgTORMOD2', 1), (3, 'TORMOD3', 1.0, 4.0, 1.0, 4.0, 0.0, -2.0, 0.0, 'SE
RIEMOD3', 'TORMODkommentar3', 3, 'imgTORMOD3', 1), (4, 'TORMOD4', 1.0, 7.0, 7.0, 7.0, 0.
0, 3.0, 0.0, 'SERIEMOD4', 'TORMODkommentar4', 4, 'imgTORMOD4', 1), (5, 'ADRIAN1', 1.0,
5.0, 1.0, 5.0, 0.0, 0.0, 0.0, 'SERIEADR1', 'CHANkommentar1', 1, 'imgCHAN1', 2), (6, 'ADR
IAN2', 1.0, 2.0, 4.0, 2.0, 0.0, 1.0, 0.0, 'SERIEADR2', 'CHANkommentar2', 2, 'imgCHAN2',
2), (7, 'ADRIAN3', 1.0, 4.0, 1.0, 4.0, 0.0, -2.0, 0.0, 'SERIEADR3', 'CHANkommentar3', 3,
'imgCHAN3', 2), (8, 'MARTE1', 12.0, 5.0, 3.0, 4.0, 0.0, 0.0, 0.0, 'SERIEMAR1', 'MARTEkom
mentar1', 1, 'imgMARTE1', 3), (9, 'MARTE2', 15.0, 2.0, 3.0, 5.0, 0.0, 1.0, 0.0, 'SERIEMA
R2', 'MARTEkommentar2', 2, 'imgMARTE2', 3), (10, 'MARTE3', 19.0, 4.0, 3.0, 2.0, 0.0, -2.
0, 0.0, 'SERIEMAR3', 'MARTEkommentar3', 3, 'imgMARTE3', 3), (11, 'MARTE4', 43.0, 7.0, 7.
0, 5.0, 0.0, 3.0, 0.0, 'SERIEMAR4', 'MARTEkommentar4', 4, 'imgMARTE4', 3), (12, 'MARTE
5', 54.0, 10.0, 10.0, 5.0, 0.0, 3.0, 0.0, 'SERIEMAR5', 'MARTEkommentar5', 5, 'imgMARTE
5', 3), (13, 'HALVOR1', 2.0, 5.0, 1.0, 5.0, 0.0, 0.0, 0.0, 'SERIEHAL1', 'HALVORkommentar
1', 1, 'imgHALVOR1', 4), (14, 'HALVOR2', 2.0, 2.0, 4.0, 2.0, 0.0, 1.0, 0.0, 'SERIEHAL2',
'HALVORkommentar2', 2, 'imgHALVOR2', 4), (15, 'HALVOR3', 2.0, 4.0, 1.0, 4.0, 0.0, -2.0,
0.0, 'SERIEHAL3', 'HALVORkommentar3', 3, 'imgHALVOR3', 4), (16, 'HALVOR4', 2.0, 2.0, 4.
0, 2.0, 0.0, 3.0, 0.0, 'SERIEHAL4', 'HALVORkommentar4', 4, 'imgHALVOR4', 4), (17, 'HALVO
R5', 2.0, 6.0, 7.0, 2.0, 0.0, 3.0, 0.0, 'SERIEHAL5', 'HALVORkommentar5', 5, 'imgHALVOR
5', 4), (18, 'MATHIAS1', 1.0, 5.0, 1.0, 5.0, 0.0, 0.0, 0.0, 'SERIEMAT1', 'MATHIASkomment
ar1', 1, 'imgMATHIAS1', 5), (19, 'MATHIAS2', 1.0, 2.0, 4.0, 2.0, 0.0, 1.0, 0.0, 'SERIEMA
T2', 'MATHIASkommentar2', 2, 'imgMATHIAS2', 5), (20, 'MATHIAS3', 1.0, 6.0, 6.0, 7.0, 0.
0, -2.0, 0.0, 'SERIEMAT3', 'MATHIASkommentar3', 3, 'imgMATHIAS3', 5), (21, 'MATHIAS4',
1.0, 9.0, 9.0, 12.0, 0.0, 3.0, 0.0, 'SERIEMAT4', 'MATHIASkommentar4', 4, 'imgMATHIAS4',
5), (22, 'LIV1', 13.0, 2.586, 2.145, 1.999, 0.0, 0.0, 0.0, 'SERIELIV1', 'LIVkommentar1',
1, 'imgLIV1', 6), (23, 'LIV2', 32.0, 3.586, 3.145, 0.795, 0.0, 1.0, 0.0, 'SERIELIV2', 'L
IVkommentar2', 2, 'imgLIV2', 6), (24, 'LIV3', 43.0, 4.586, 3.145, 1.095, 0.0, -2.0, 0.0,
'SERIELIV3', 'LIVkommentar3', 3, 'imgLIV3', 6), (25, 'LIV4', 45.0, 4.586, 5.145, 1.795,
0.0, 3.0, 0.0, 'SERIELIV4', 'LIVkommentar4', 4, 'imgLIV4', 6)]
```

This is extremely hard to read, so to get the results from the SELECT statements we are using in a more organised matter we imported pandas.

Pandas

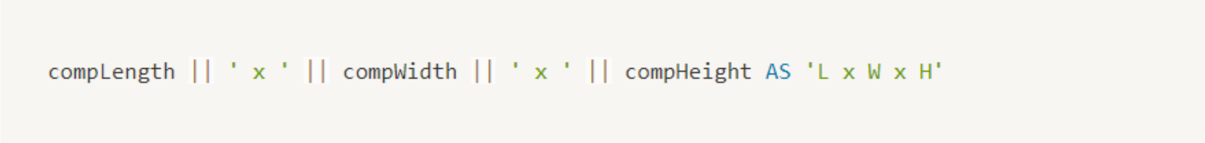
```
pip install pandas
```



```
import pandas as pd

#SELECT statement for Component table
querySELECTcomponent = '''
SELECT
    idComp AS 'ID',
    partNR AS 'Part NR',
    mass AS 'Mass',
    compLength || ' x ' || compWidth || ' x ' || compHeight AS 'L x W x H',
    ecc_x || ', ' || ecc_y || ', ' || ecc_z AS 'Eccentricity',
    serialNR AS 'Serial NR',
    comment AS 'Comment',
    ranking AS 'Ranking',
    imgComp AS 'Image file',
    density_idDensity AS 'Density ID'
FROM component;'''
```

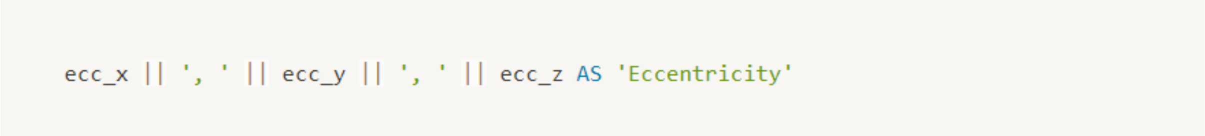
The column names in the table are not always the best headers to display since they often are shorted. Using the SQL “AS 'new column name'” will make the table even easier to understand.

We have also used the:



```
compLength || ' x ' || compWidth || ' x ' || compHeight AS 'L x W x H'
```

And:



```
ecc_x || ', ' || ecc_y || ', ' || ecc_z AS 'Eccentricity'
```

to combine multiple columns into one new column only for display purposes. This does not change anything about the table.

Now instead of just printing out the result of an execute running that query we use pandas.

```
# TABLE: COMPONENT
tableComponent = pd.read_sql_query(querySELECTcomponent, cnx)
dfComponent = pd.DataFrame(tableComponent, columns=
    ['ID',
     'Part NR',
     'Mass',
     'L x W x H',
     'Eccentricity',
     'Serial NR',
     'Comment',
     'Ranking',
     'Image file',
     'Density ID'])
dfComponent.set_index('ID', inplace=True)
print(dfComponent)
```

```
tableComponent = pd.read_sql_query(querySELECTcomponent, cnx)
```

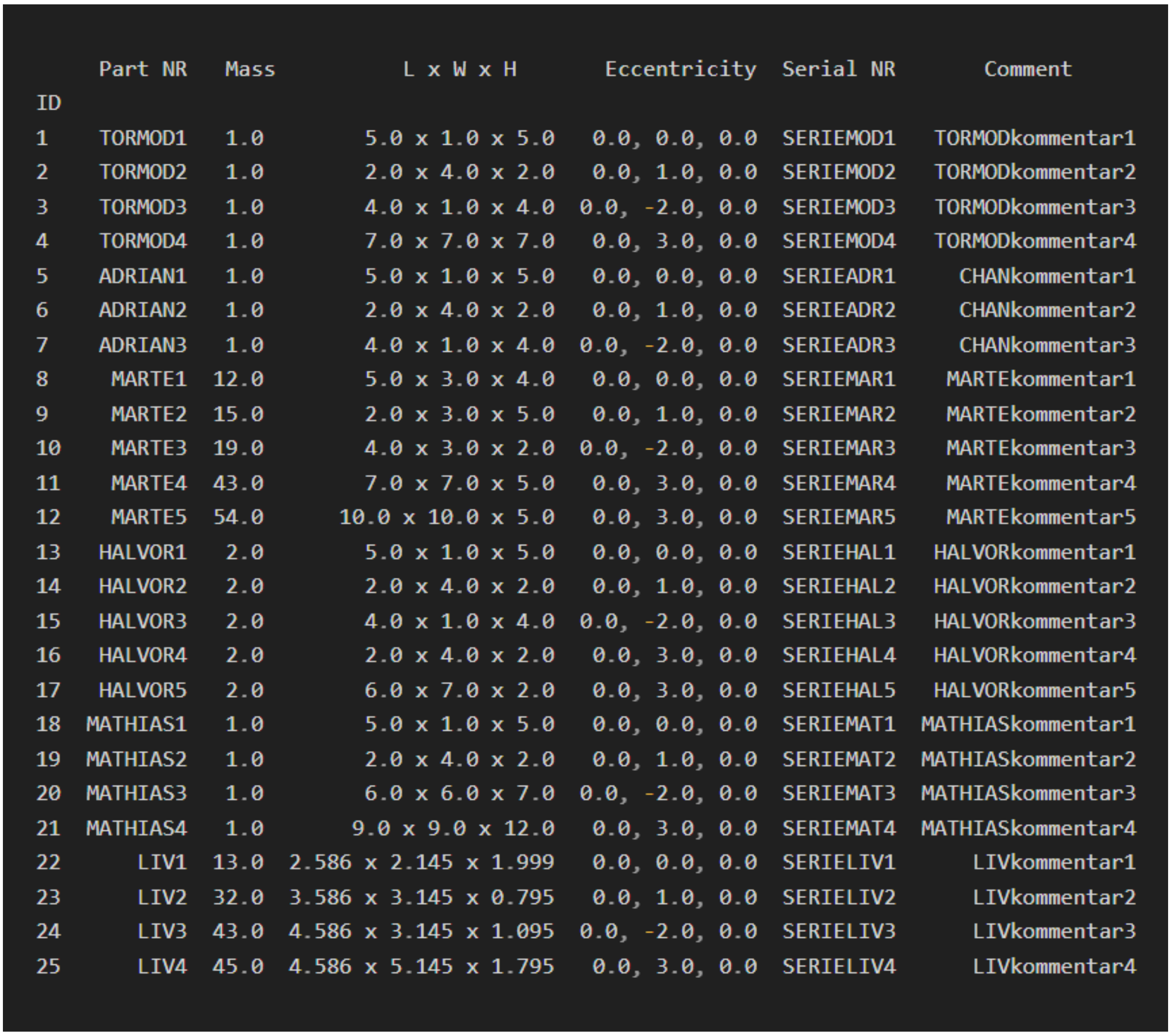
This first line in this code is the way to tell pandas to read the result of the query "querySELECTcomponent" from the database we have connected to with "cnx" earlier.

The next line creates a data frame that is stored in the variable dfComponent. It uses the result from the previous line as the data for the data frame and the columns parameter matches the ones from the SELECT query. This will be used as column headers.

```
dfComponent.set_index('ID', inplace=True)
```

The second to last line of this code lets pandas know that the 'ID' column is the index for each row in the table. It doesn't need to add a new one as an index, just use the one that is already there.

The last thing we do is print the data frame we created, and this is the result from the terminal:



```
    Part NR  Mass              L x W x H   Eccentricity  Serial NR           Comment
ID
1   TORMOD1   1.0        5.0 x 1.0 x 5.0  0.0, 0.0, 0.0  SERIEMOD1  TORMODkommentar1
2   TORMOD2   1.0        2.0 x 4.0 x 2.0  0.0, 1.0, 0.0  SERIEMOD2  TORMODkommentar2
3   TORMOD3   1.0        4.0 x 1.0 x 4.0 0.0, -2.0, 0.0  SERIEMOD3  TORMODkommentar3
4   TORMOD4   1.0        7.0 x 7.0 x 7.0  0.0, 3.0, 0.0  SERIEMOD4  TORMODkommentar4
5   ADRIAN1   1.0        5.0 x 1.0 x 5.0  0.0, 0.0, 0.0  SERIEADR1    CHANkommentar1
6   ADRIAN2   1.0        2.0 x 4.0 x 2.0  0.0, 1.0, 0.0  SERIEADR2    CHANkommentar2
7   ADRIAN3   1.0        4.0 x 1.0 x 4.0 0.0, -2.0, 0.0  SERIEADR3    CHANkommentar3
8    MARTE1  12.0        5.0 x 3.0 x 4.0  0.0, 0.0, 0.0  SERIEMAR1   MARTEkommentar1
9    MARTE2  15.0        2.0 x 3.0 x 5.0  0.0, 1.0, 0.0  SERIEMAR2   MARTEkommentar2
10   MARTE3  19.0        4.0 x 3.0 x 2.0 0.0, -2.0, 0.0  SERIEMAR3   MARTEkommentar3
11   MARTE4  43.0        7.0 x 7.0 x 5.0  0.0, 3.0, 0.0  SERIEMAR4   MARTEkommentar4
12   MARTE5  54.0      10.0 x 10.0 x 5.0  0.0, 3.0, 0.0  SERIEMAR5   MARTEkommentar5
13  HALVOR1   2.0        5.0 x 1.0 x 5.0  0.0, 0.0, 0.0  SERIEHAL1  HALVORkommentar1
14  HALVOR2   2.0        2.0 x 4.0 x 2.0  0.0, 1.0, 0.0  SERIEHAL2  HALVORkommentar2
15  HALVOR3   2.0        4.0 x 1.0 x 4.0 0.0, -2.0, 0.0  SERIEHAL3  HALVORkommentar3
16  HALVOR4   2.0        2.0 x 4.0 x 2.0  0.0, 3.0, 0.0  SERIEHAL4  HALVORkommentar4
17  HALVOR5   2.0        6.0 x 7.0 x 2.0  0.0, 3.0, 0.0  SERIEHAL5  HALVORkommentar5
18 MATHIAS1   1.0        5.0 x 1.0 x 5.0  0.0, 0.0, 0.0  SERIEMAT1 MATHIASkommentar1
19 MATHIAS2   1.0        2.0 x 4.0 x 2.0  0.0, 1.0, 0.0  SERIEMAT2 MATHIASkommentar2
20 MATHIAS3   1.0        6.0 x 6.0 x 7.0 0.0, -2.0, 0.0  SERIEMAT3 MATHIASkommentar3
21 MATHIAS4   1.0       9.0 x 9.0 x 12.0  0.0, 3.0, 0.0  SERIEMAT4 MATHIASkommentar4
22     LIV1  13.0  2.586 x 2.145 x 1.999  0.0, 0.0, 0.0  SERIELIV1     LIVkommentar1
23     LIV2  32.0  3.586 x 3.145 x 0.795  0.0, 1.0, 0.0  SERIELIV2     LIVkommentar2
24     LIV3  43.0  4.586 x 3.145 x 1.095 0.0, -2.0, 0.0  SERIELIV3     LIVkommentar3
25     LIV4  45.0  4.586 x 5.145 x 1.795  0.0, 3.0, 0.0  SERIELIV4     LIVkommentar4
```

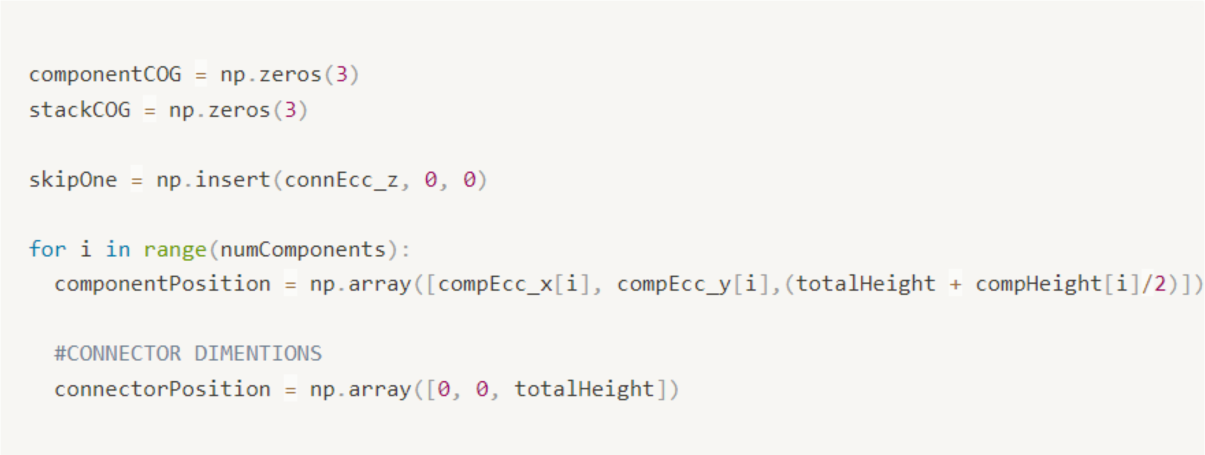
this is a lot easier to read and makes detecting errors so much faster.

NumPy

In the code below you can see NumPy being used to create separate arrays containing only the same datatypes. Each variable is storing different column data for each row from the query “sqlComponent”. This will make it a lot easier to write the code for the software.

```
#QUERY TO RETREIVE INFORMATION ABOUT COMPONENTS
sqlComponent = f'''
SELECT
    component.idComp AS 'ID',
    component.partNR AS 'Part NR',
    component.compLength AS 'L',
    component.compWidth AS 'W',
    component.compHeight AS 'H',
    component.ecc_x AS 'ecc X',
    component.ecc_y AS 'ecc Y',
    component.ranking AS 'Ranking',
    component.imgComp AS 'Image file',
    density.nameMaterial AS 'Material',
    density.componentDensity AS 'Density'
FROM analysis
    JOIN vesselConfig ON analysis.vesselConfig_idConfig = vesselConfig.idVesselConfig
    JOIN vessel ON vesselConfig.vessel_idVessel = vessel.idVessel
    JOIN stack ON vesselConfig.stack_idStack = stack.idStack
    JOIN stackConfig ON stack.idStack = stackConfig.stack_idStack
    JOIN component ON stackConfig.comp_idComp = component.idComp
    JOIN density ON component.density_idDensity = density.idDensity
        WHERE analysis.idAnalysis = {analysis} ORDER BY component.ranking;'''
cursor.execute(sqlComponent)
tableComponents = cursor.fetchall()

#SAVE COMPONENT DATA
idComp =        [row[0] for row in tableComponents]
partNR =        [row[1] for row in tableComponents]
compLength =    np.array([row[2] for row in tableComponents])
compWidth =     np.array([row[3] for row in tableComponents])
compHeight =    np.array([row[4] for row in tableComponents])
compEcc_x =     np.array([row[5] for row in tableComponents])
compEcc_y =     np.array([row[6] for row in tableComponents])
compRanking =   [row[7] for row in tableComponents]
imgComp =       [row[8] for row in tableComponents]
compMaterial =  [row[9] for row in tableComponents]
compDensity =   np.array([row[10] for row in tableComponents])
```

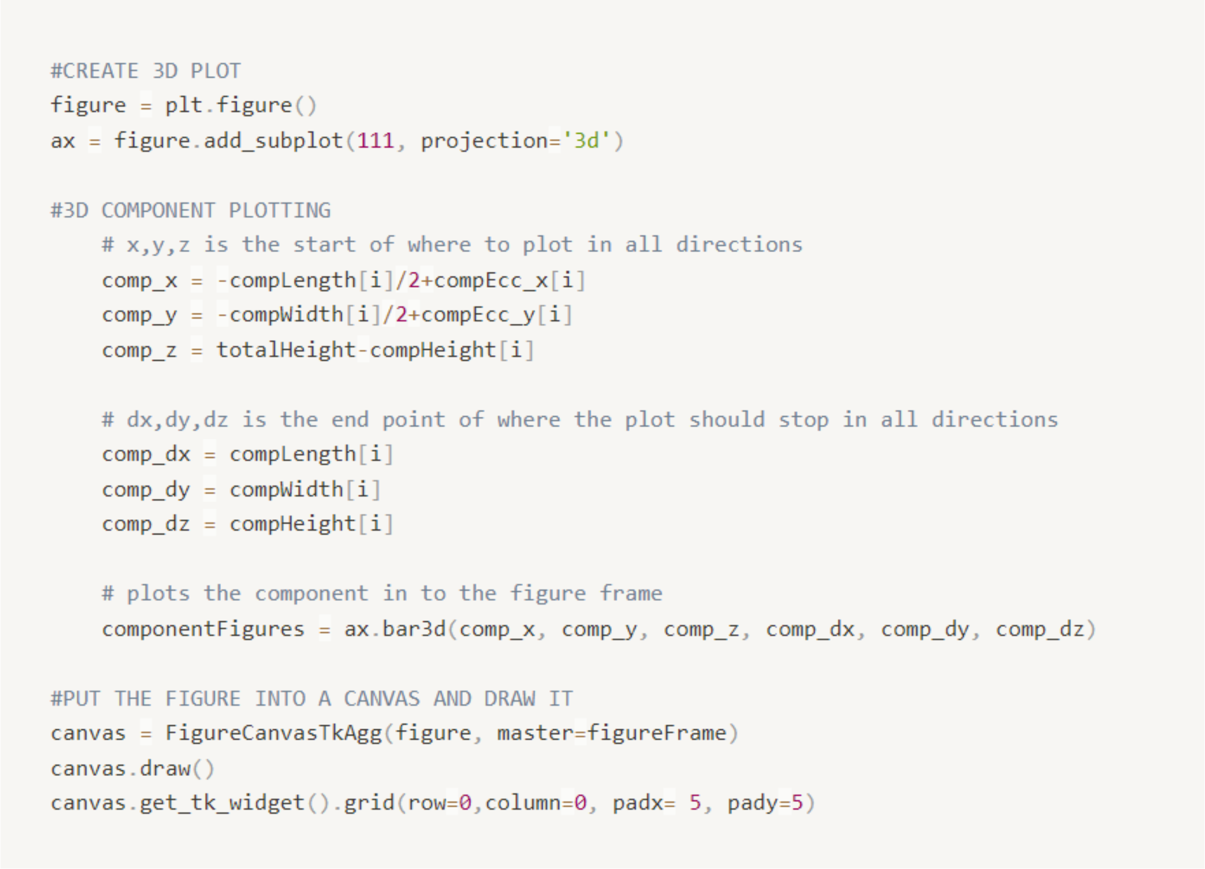
The lines below are used in the main code for the stack configuration which in the end will let you see a 3D visualisation of the stack you are doing the analysis on. Here you can see NumPy being used multiple times.



```
componentCOG = np.zeros(3)
stackCOG = np.zeros(3)

skipOne = np.insert(connEcc_z, 0, 0)

for i in range(numComponents):
  componentPosition = np.array([compEcc_x[i], compEcc_y[i],(totalHeight + compHeight[i]/2)])

  #CONNECTOR DIMENTIONS
  connectorPosition = np.array([0, 0, totalHeight])
```

Matplotlib

The code below shows how the matplotlib library is used to plot the components on top of each other. First the figure is created, then the components are plotted and then they are put into the figure at the bottom.



```
#CREATE 3D PLOT
figure = plt.figure()
ax = figure.add_subplot(111, projection='3d')

#3D COMPONENT PLOTTING
    # x,y,z is the start of where to plot in all directions
    comp_x = -compLength[i]/2+compEcc_x[i]
    comp_y = -compWidth[i]/2+compEcc_y[i]
    comp_z = totalHeight-compHeight[i]

    # dx,dy,dz is the end point of where the plot should stop in all directions
    comp_dx = compLength[i]
    comp_dy = compWidth[i]
    comp_dz = compHeight[i]

    # plots the component in to the figure frame
    componentFigures = ax.bar3d(comp_x, comp_y, comp_z, comp_dx, comp_dy, comp_dz)

#PUT THE FIGURE INTO A CANVAS AND DRAW IT
canvas = FigureCanvasTkAgg(figure, master=figureFrame)
canvas.draw()
canvas.get_tk_widget().grid(row=0,column=0, padx= 5, pady=5)
```

6.8 Analysis report in LaTex format

One of TFMC's criteria was that the program needed to deliver an end report in either PDF or LaTex format. LaTex was the preferred format for TFMC because they frequently use this type of format. This provided a challenge, since this meant that the LaTex code needed to be integrated into the application's primary code.

The first stage in creating the application's final report involved utilizing a LaTex compiler called Overleaf, to code a report that matched one provided by TFMC, to make sure that the report would adhere to TFMC standards.

The code was incorporated into the main program after the initial stage of coding completed and the report was finalized in accordance with TFMC standards. It was possible to integrate a LaTex library into the main application using Visual Studio Code before integrating the end report's code directly into the main application. This allows program users the option to modify and include new parts as needed.

7 Proposed further work

Since the time frame for a project like this bachelor thesis is a major constraint, there were several areas of interest the team did not have time to explore. Therefor there is a number of subjects one can elaborate on more in the future. The following subjects could be included or expanded upon with the continuation of this project in future bachelor projects.

7.1 Implementation of RAO data

RAO data is implemented by having the application read an excel file with the given data from a customer. Early on in the project, it was mentioned that the customer may provide RAO data in a variety of file formats, including text, CSV, and Excel. It would be beneficial to look at how one might develop an extension for the application that automates the import of RAO data from all file formats, not only the excel format.

7.2 3D graphics

The project's specifications called for a front and side view of the components; in the end, this was accomplished using a straightforward 3D orientation of boxes that indicated the components' size and centre of gravity (COG). Given that each component has a unique geometry, it would be fascinating to construct a true 3D graphical representation of the components. This will result in foregoing one of the demands from TFMC in terms of an application that did not require a lot of processing power from a computer.

7.3 Irregular waves

The team used regular waves for this project, which are waves with constant amplitude and frequency. Waves that vary in amplitude and frequency are referred to as irregular waves. In contrast to irregular waves, which would provide a much more realistic study of the forces and impacts a vessel experiences, regular waves are a conservative manner of examining these forces and impacts. It would continue to fulfil the biggest objective of TFMC, which is to phase out pricey analytical software, with the addition of irregular water waves.

Conclusion

This year's bachelor project has shown that it is achievable to create an application for skidding and transit analysis, with the goal to collect data for evaluation on when to perform a skidding and transit operation depending on sea roughness, and the impact it has on a vessel and equipment onboard.

The team's mechanical members were in charge of developing the mathematical modelling for the project, which serves as the foundation for the project in terms of having the data necessary to conduct an analysis. In conjunction to the mathematical modelling, the mechanical team has also been tasked with developing some of the computer-based sections.

The initial and primary criteria stated that the analysis for a regular wave pattern had to match the outcome of an earlier analysis that TFMC performed for one of their clients. The reason behind this is because the same RAO data, made anonymous, was provided for this project. When working with the requirements with the key stakeholders, it was decided that this was the best form in terms of stating a requirement that would have a viable test. The mechanical team members met this requirement by constructing the mathematical modelling piece by piece, in terms of outlining the equations line by line, and then create python scripts for the mathematical modelling. This was then given over to the computer part of the team, for implementation into the application. This mathematical model contains calculations on acceleration for a vessel, based on the RAO data provided, forces in terms of bending moment the equipment onboard a vessel is impacted by. And finally, pressure test calculations for capacity calculation for topside pressure testing, and utilization calculation.

The requirement was fulfilled in terms of developing a mathematical model for acceleration and force, however at the very end of the project it was discovered that one of the initial equations provided by TFMC contained a mistake, therefore a new equation was offered to the team. This indicates that the calculations might not be entirely accurate. After the project's completion, the mechanical members will examine the new, revised framework for the mathematical modelling. To guarantee that the analysis is accurate, one of the key stakeholders will need to fix the equation in the application if the mechanical team members do not have the time to conduct this correction.

One of the other requirements that was included in the mathematical modelling, stated that the visual representation of the CoG must give a summery of the stack as a whole. This was solved by giving a 3D-representation of CoG on the stack together with the coordinate values. The mechanical team created a framework for the calculations and produced the first iteration of a python script that gave a clear view of the CoG. The script was then passed over to the computer part for a modularization of the script to operate with as many or few components as the situation required. In addition, a depiction of connectors and information about the individual components was added for more clarity.

Since neither of the team's mechanical members had any significant prior experience with developing a large-scale mathematical model, this project presented a significant challenge for them. In correlation to this, they also had to collaborate with a discipline outside of their own in order to comprehend what the other team members were working on and understand what they needed the mechanical team to create. To enable a simple integration of the mathematical model into the application, they also needed to make the mathematical model understandable for the computer team. Learning higher-level theoretical mathematics and building a whole mathematical model from scratch entailed a steep learning curve. However, they also gained a lot of knowledge about working with a different discipline and on a larger project.

Regarding the code overall, the transition from independent code to a class-based structure enabled efficient reuse of the centre of gravity, bending moment, and utilization calculations. This approach promoted code modularity, enhanced maintainability, and facilitated future enhancements or extensions to the calculations without impacting other parts of the project.

The software delivered is a comprehensive database accessed by a graphical user interface. With the provided tools in the software the user is able to do skidding and transit analysis. While the calculations are systematically added into the software the calculations relating to bending moment are accessed via compiling the given python file. Among the files delivered there is a template for the skidding and transit analysis. The functionality of adding and appending information onto the LaTex file via the software is an easy process but will require improvement.

References

[1] scrum.org, “What is a Sprint Review?,” [Online]. Available: https://www.scrum.org/resources/what-is-a-sprint-review. [Accessed 31 January 2023].

[2] A. Srivastava, S. Bhardwaj and S. Saraswat, “SCRUM model for agile methodology,” *2017 International Conference on Computing,Communication and Automation,* Desember 2017.

[3] G. M. Bonnema, K. T. Veenvliet and J. F. Broenink, Systems design and engineering, Boca Raton, Florida: CRC Press, 2016.

[4] Scrum.org, “Scrum Glossary,” Scrum.org, [Online]. Available: https://www.scrum.org/Resources/Scrum-Glossary. [Accessed 07 March 2023].

[5] M. Larusdottir, Å. Cajander, P. Gregory, G. Cockton, D. Salah, K. Kuusinen and G. Nauwerck, “Stakeholder Involvement in Agile Software Development,” *Nordic Conference on Human-Computer Interaction: Game-Changing Design, New York,* 2016.

[6] K. Schwaber and J. Sutherland, “The Scrum Guide,” November 2020. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

[7] K. Power, “Stakeholder Identification in Agile Software Product Development Orginizations: A model for understanding who and what really counts,” *2012 Agile Conference, Nashville,* 2012.

[8] Å. Cajander, J. Gulliksen and M. Larusdottir, “Existing but not explicit - The user perspective in scrum projects in practice,” *IFIP Conference on Human-Computer Interaction,* September 2013.

[9] D. M. Buede, The Engineering Design of Systems, Hoboken, New Jersey: John Wiley & Sons Inc., 2009.

[10] C. Shamieh, Systems Engineering for Dummies, Hoboken, New Jersey: Wiley Publishing Inc., 2011.

[11] D. W. Oliver, T. P. Kelliher and J. G. Keegan, Engineering Complex Systems with Models and Objects, New York: McGraw-Hill, 1997.

[12] The MITRE Corporation, “Mitre.org,” 2014. [Online]. Available: https://www.mitre.org/sites/default/files/2022-09/MITRE-SEG.pdf. [Accessed March 2023].

[13] E. S. Chia, “Risk Assessment Framework for Project Management,” *2006 IEEE International Engineering Managment Conference,* 30 July 2007.

[14] N. Lavanya and T. Malarvizhi, “Risk analysis and management - A vital key to effective project management,” [Online]. Available: https://www.pmi.org/learning/library/risk-analysis-project-management-7070#. [Accessed March 2023].

[15] Association for Project Management, “Project risk analysis and management,” [Online]. Available: https://www.apm.org.uk/media/10466/pram_web.pdf. [Accessed March 2023].

[16] S. W. O'Connor, “How to Create a Risk Management Plan for Your Project,” Northeastern University Graduate Programs, 18 September 2020. [Online]. Available: https://www.northeastern.edu/graduate/blog/risk-management-plan-risk-register/. [Accessed March 2023].

[17] T. M. Willams, “Using a risk register to integrate risk managment in project definition,” *International Journal of Project Management,* pp. 17-22, February 1994.

[18] C. George, “The Essence of Risk Identification in Project Risk Management,” *International Journal of Science and Research,* February 2020.

[19] C. s. U. t. calculators, “CalcLata Suite,” [Online]. Available: https://www.calqlata.com/productpages/00081-help.html. [Accessed January 2023].

[20] A. H. Gjeraker, “Response Amplitude Operator Estimation and Wave Modeling Sensitivity,” NTNU: Norwegian University of Science and Technology. Faculty of Engineering, Department of Marine Technology., Trondheim, 2021.

[21] Ø. Rabliås, “Numerical and Experimental Studies of Manuvering in Regular and Irregular waves.,” NTNU: Norwegian University of Science and Technology. Faculty of Engineering. Department of Marine Technology., Trondheim, 2022.

[22] H. Goldstein, J. L. Safko and C. P. Poole, Jr., Classical Mechanics, Harlow: Pearson Education Limited, 2014.

[23] D. W. Baker and W. Haynes, Engineering Statics: Open and Interative, Colorado: Daniel W. Baker and William Haynes, 2020.

[24] E. G. Hemingway and O. M. O'Reilly, “Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments,” *Multibody System Dynamics. Vol. 44 ,* pp. 31-56, 2018.

[25] Y.-B. Jia, “Quaternions and Rotations,” Stanford University, Stanford, 2013.

[26] R. Mustad, “Quaternions with applications,” NTNU: Norwegian University of Science and Technology, Trondheim, 2019.

[27] R. Shankar, Fundamentals of Physics: Mechanics, Relativity and Thermodynamics, Yale University Press, 2014.

[28] H. Snersrud Gustad, “Skidding on Vessel: An introduction to skidding operations,” TechnipFMC, 2023.

[29] H. H. a. I. Hallseth, Svingninger og bølger, vol. II, NKI Forlaget, 1996.

[30] K. K. a. V. Gulbrandsen, Matematikk for ingeniører, Gyldendal Akademisk, 2016.

[31] C. H. Edwards and D. E. Penney, Calculus, Pearson Education International, 2002.

[32] E. Gregersen, “Britannica,” [Online]. Available: https://www.britannica.com/science/centre-of-gravity. [Accessed Januray 2023].

[33] B. Ram, Engineering mathematics, 2nd ed., vol. I, Pearson India, 2012.

[34] Dansk Standard, “Olie og naturgasindustrien - Design og drift af undersøiske produktionsystemer,” Dansk Standard, København, 2007.

[35] S. Kalpakjian, R. S. Schmid and K. S. V. Sekar, Manufacturing Engineering and Technology, Singapore: Pearson Education South Asia Pte Ltd, 2014.

[36] T. Halpin, “ORM/NIAM Object-Role Modeling,” 1998. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-03526-9_4#citeas. [Accessed 20 05 2023].

[37] IBM, “Key concepts: Entity, attribute, and entity type,” 25 10 2022. [Online]. Available: https://www.ibm.com/docs/en/imdm/12.0?topic=concepts-key-entity-attribute-entity-type. [Accessed 20 05 2023].

[38] dbForge Team, “Relationships in SQL – Complete Guide With Examples,” 26 10 2021. [Online]. Available: https://blog.devart.com/types-of-relationships-in-sql-server-database.html. [Accessed 20 05 2023].

[39] D. Mendoza and N. Piedra, “TutNorBD: Assistant for teaching and learning process of rlational database normalization up to 3NF from a universal table,” 19 03 2021. [Online]. Available: https://ieeexplore-ieee-org.ezproxy2.usn.no/stamp/stamp.jsp?tp=&arnumber=9381184. [Accessed 20 05 2023].

[40] N. Mendjoge, A. R. Joshi and M. Narvekar, “Intelligent tutoring system for Database Normalization,” 23 02 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy2.usn.no/document/7860013. [Accessed 20 05 2023].

[41] K. Kumar and D. S. K. Azad, “Database Normalization Design Pattern,” 11 01 2018. [Online]. Available: https://ieeexplore-ieee-org.ezproxy2.usn.no/document/8251067. [Accessed 20 05 2023].

[42] SQLite, “About SQLite,” SQLite, [Online]. Available: https://sqlite.org/about.html. [Accessed 20 05 2023].

[43] SQLite, “What Is SQLite?,” SQLite, [Online]. Available: https://sqlite.org/index.html. [Accessed 20 05 2023].

[44] Atlassian, “A guide to scrum: what it is and how it works,” [Online]. Available: https://www.atlassian.com/agile/scrum. [Accessed 31 January 2023].

[45] Tuleap, “Understanding Agile Scrum in 10 minutes,” [Online]. Available: https://www.tuleap.org/agile/agile-scrum-in-10-minutes. [Accessed 31 January 2023].

[46] Atlassian, “Sprint planning,” [Online]. Available: https://www.atlassian.com/agile/scrum/sprint-planning. [Accessed 31 January 2023].

[47] Atlassian, “Agile sprint reviews,” [Online]. Available: https://www.atlassian.com/agile/scrum/sprint-reviews. [Accessed 31 January 2023].

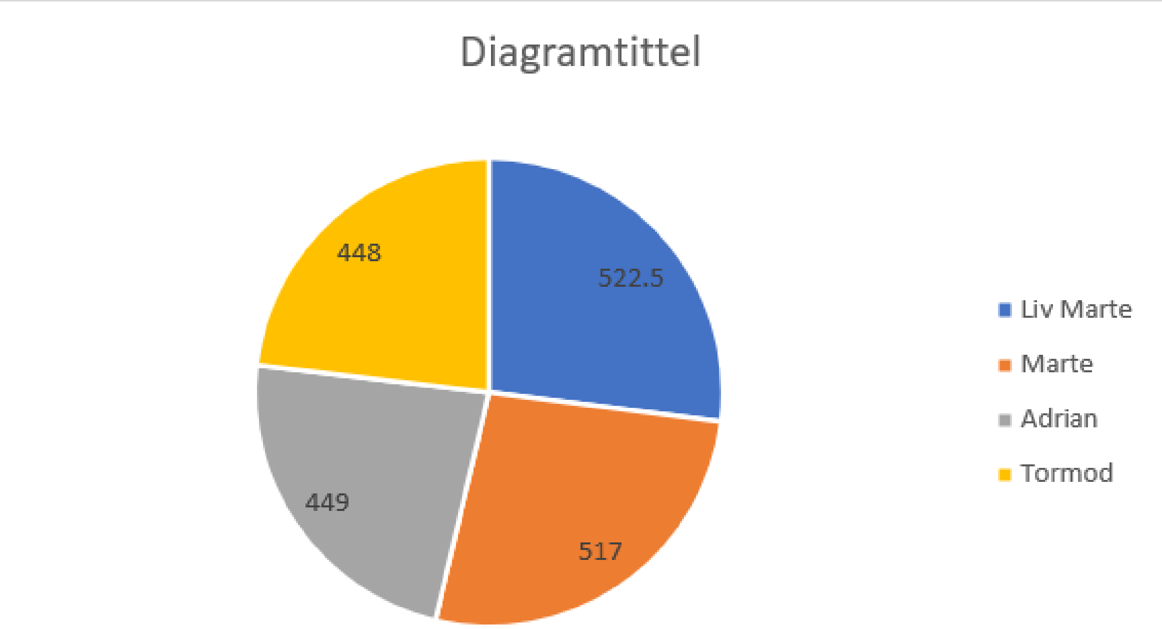
[48] scrum.org, “What is a Sprint Retrospective?,” [Online]. Available: https://www.scrum.org/resources/what-is-a-sprint-retrospective. [Accessed 31 January 2023].

[49] H. Goldstein, C. Poole, Jr. and J. Safko, Classical Mechanics, Harlow: Pearson Education Limited, 2014.

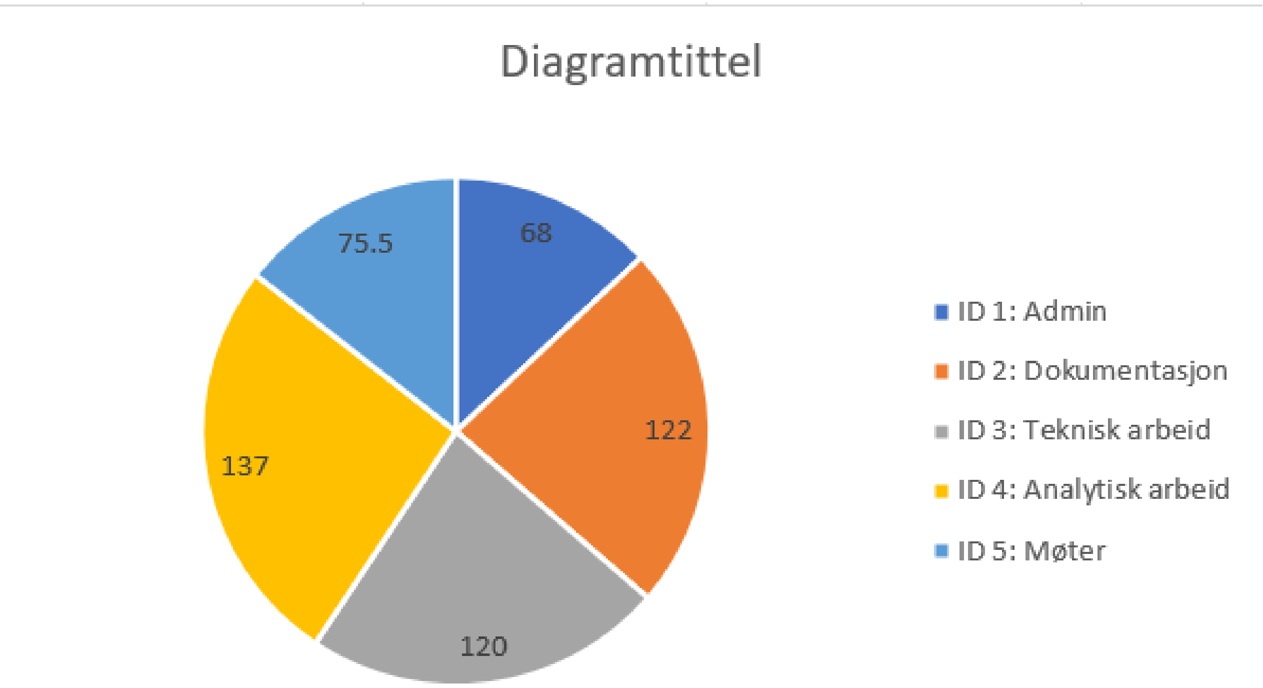
[50] J. Wittenburg, Kinematics : Theory and Applications, Berlin: Imprit: Springer, 2016.

Appendix A: Working hours

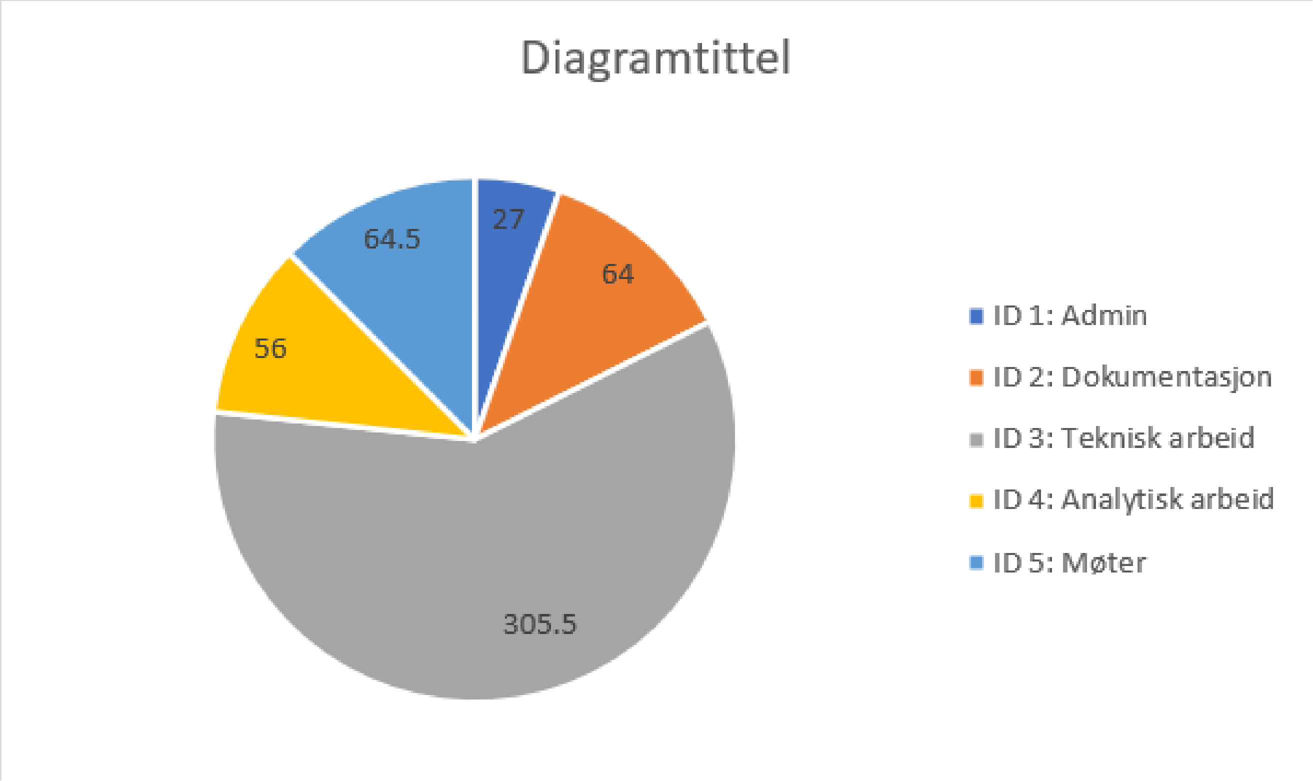
I. Total working hours for the group



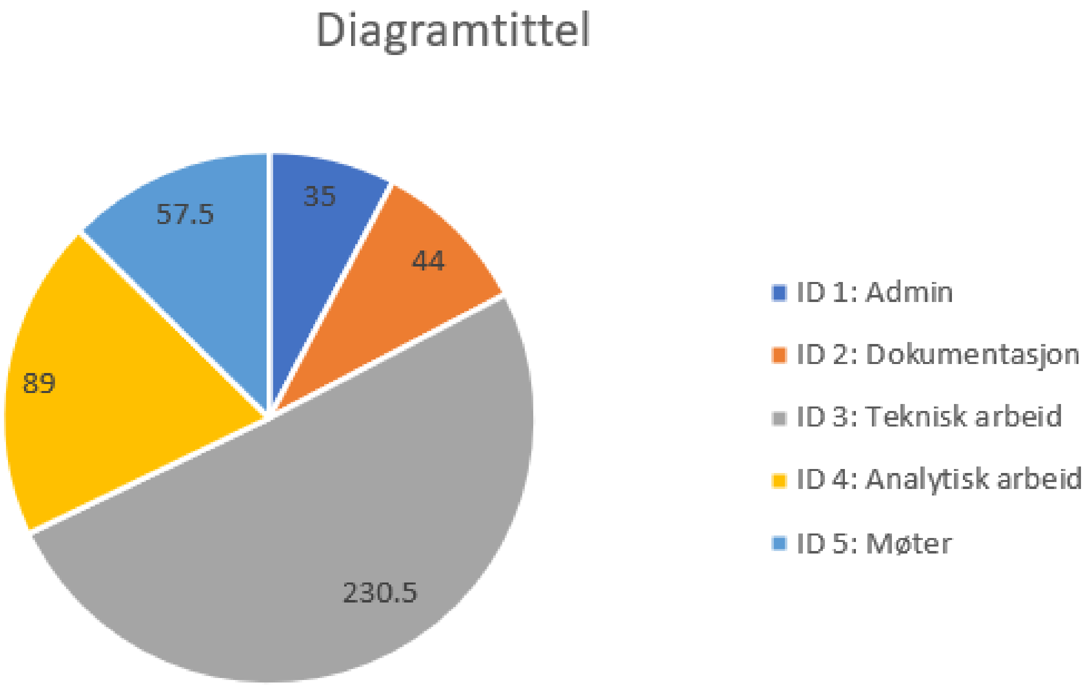
II. Work hours for Liv Marte.



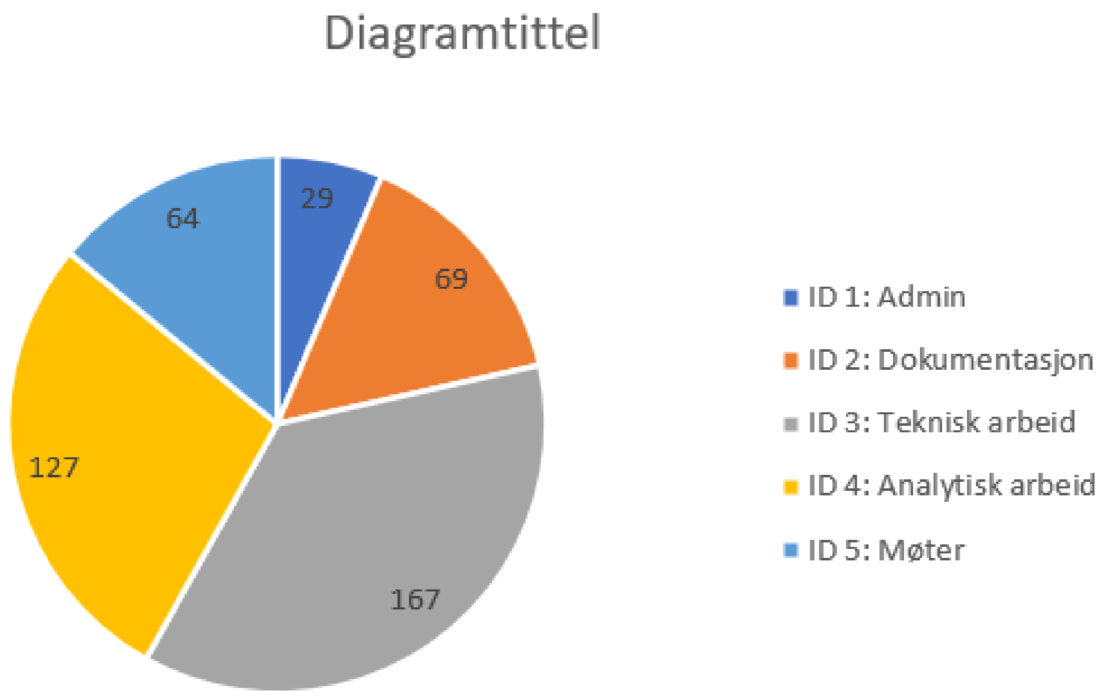
III. Working hours for Marte.



IV. Working hours for Adrian.

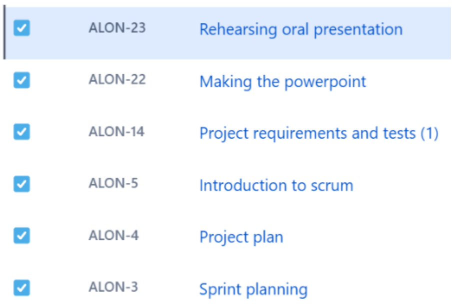
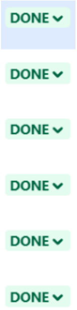


V. Working hours for Tormod.

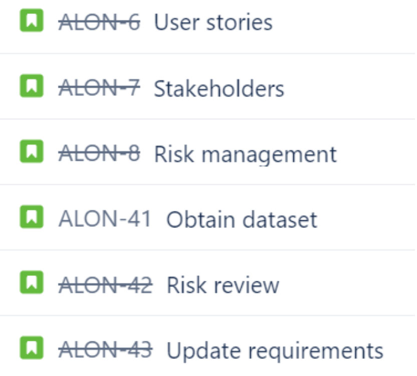
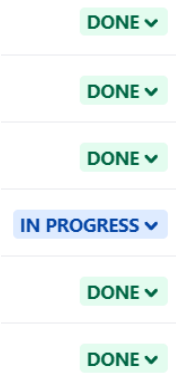


Appendix B: Sprint reviews

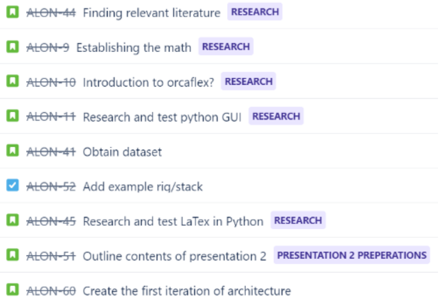
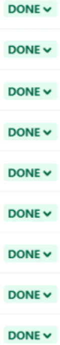
I: Sprint review for sprint 1.

Description:	Status:	Responsibility / role:
Overview	Completed	All team members
Sprint goal (Ready for presentation 1)	Sprint goal is met.	
Increments	“Definition of done” met	
What went well and not so well?	Everything went well	
Presentation (sprint goal)	Satisfied all expectations	
Motivation coming into sprint 2	Motivation on top	All team members
Planning of sprint 2	Completed.	
 ☑ ALON-23 Rehearsing oral presentation ☑ ALON-22 Making the powerpoint ☑ ALON-14 Project requirements and tests (1) ☑ ALON-5 Introduction to scrum ☑ ALON-4 Project plan ☑ ALON-3 Sprint planning	 DONE DONE DONE DONE DONE DONE	

II: Sprint review for sprint 2

Description:	Sprint status:	Responsibility / role:
How is everyone doing?	Great!	All team members
Overview	Completed.	
Sprint goal (All administrative prework done?)	All administrative prework is done!	
Increments	Everything finished except “obtaining dataset”.	
What went well and not so well?	Communication with stakeholders has proven a little harder than usual.	
Motivation coming into sprint 3?	Motivation is still going strong.	
Planning sprint 3	Whether orcaflex will be used or not will be decided within sprint 3.	Mechanical.
	LaTex integration with python. (PyLaTex) Python GUI	Computer.
 ~~ALON-6~~ User stories ~~ALON-7~~ Stakeholders ~~ALON-8~~ Risk management ALON-41 Obtain dataset ~~ALON-42~~ Risk review ~~ALON-43~~ Update requirements	 DONE DONE DONE IN PROGRESS DONE DONE	

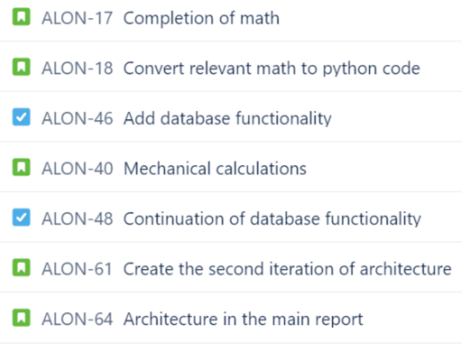
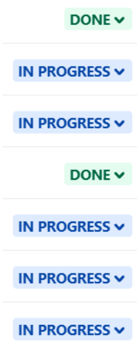
III: Sprint review for sprint 3

Description:	Sprint status:	Responsibility / role:
Overview Sprint goal Increments What went well and not so well? Presentation status? (Sprint goal)	Completed. Goal is reached. All met. Software architecture is deemed difficult. Everything is read. Pastries are ready, coffee ordered, and the streaming gear is ready for use.	All team members
Motivation coming into sprint 4. Planning of sprint 4	Motivation going forward is strong. Math lecture for an external supervisor to ensure our understanding of the field of interest. A new test where employees are testing our software will be added.	All team members
 ~~ALON-44~~ Finding relevant literature RESEARCH ~~ALON-9~~ Establishing the math RESEARCH ~~ALON-10~~ Introduction to orcaflex? RESEARCH ~~ALON-11~~ Research and test python GUI RESEARCH ~~ALON-41~~ Obtain dataset ☑ ~~ALON-52~~ Add example riq/stack ~~ALON-45~~ Research and test LaTex in Python RESEARCH ~~ALON-51~~ Outline contents of presentation 2 PRESENTATION 2 PREPERATIONS ~~ALON-60~~ Create the first iteration of architecture	 DONE DONE DONE DONE DONE DONE DONE DONE DONE	

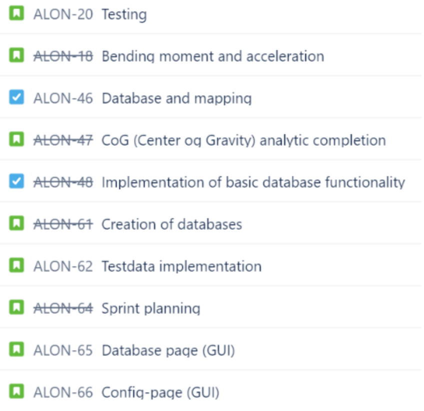
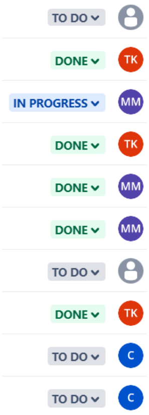
IV: Sprint review for sprint 4

Description:	Sprint status:	Responsibility / role:
Overview	Completed.	
Sprint goal	Goal is reached.	
Increments	Two increments pulled over to coming sprint.	
What went well and not so well?	Teambuilding went surprisingly well. Better visualization of how the software will look like. Great feedback on presentation.	
Motivation coming into sprint 5.	We are looking strong coming into sprint 5.	
	Orcaflex issue is deleted.	
Planning of sprint 5		

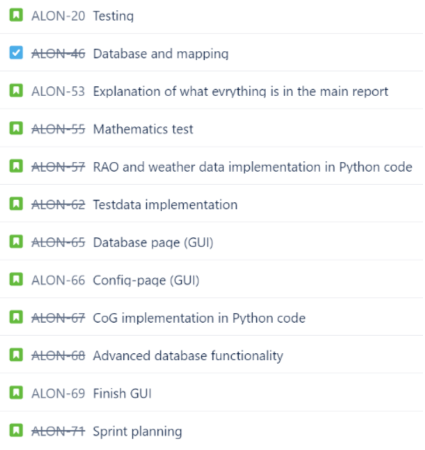
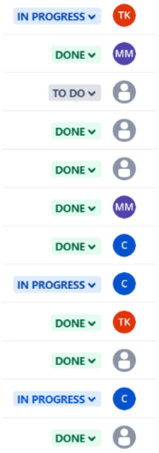
V: Sprint review for sprint 5

Description:	Sprint status:	Responsibility / role:
Overview Sprint goal Increments What went well and not so well?	Completed. Incompleted. Pulled over five increments. Teambuilding went well. Planning did not go so well for this sprint. Half of the sprint was gone due to exams.	
Motivation coming into sprint 6. Planning of sprint 6.	Motivation is still going strong. A little tired, but strong motivation. Full sprint planning for the remaining sprints is scheduled for today.	
 ALON-17 Completion of math ALON-18 Convert relevant math to python code ☑ ALON-46 Add database functionality ALON-40 Mechanical calculations ☑ ALON-48 Continuation of database functionality ALON-61 Create the second iteration of architecture ALON-64 Architecture in the main report	 DONE IN PROGRESS IN PROGRESS DONE IN PROGRESS IN PROGRESS IN PROGRESS	

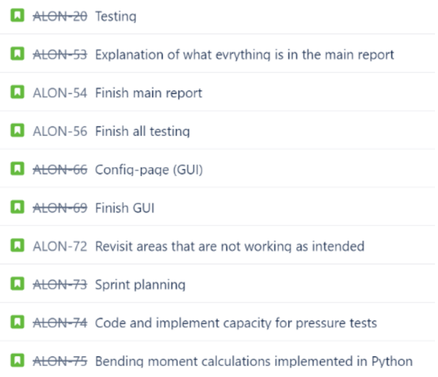
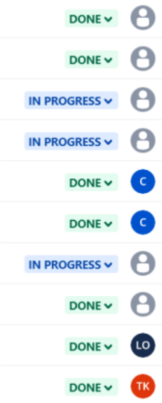
VI: Sprint review for sprint 6

Description:	Sprint status:	Responsibility / role:
Overview	Completed.	
Sprint goal (Finish all analytical mathematical work)	Completed.	The whole team.
Increments	5 increments completed.	
What went well and not so well?	TK Inter documentation had good documentation. Discussion regarding graphical depiction.	
Motivation coming into sprint 7	Motivated to start sprint 7 and the tasks it has to offer.	
Planning of sprint 7	Completed.	
 ALON-20 Testing ~~ALON-18~~ Bending moment and acceleration ☑ ALON-46 Database and mapping ~~ALON-47~~ CoG (Center og Gravity) analytic completion ☑ ~~ALON-48~~ Implementation of basic database functionality ~~ALON-61~~ Creation of databases ALON-62 Testdata implementation ~~ALON-64~~ Sprint planning ALON-65 Database page (GUI) ALON-66 Config-page (GUI)	 TO DO DONE TK IN PROGRESS MM DONE TK DONE MM DONE MM TO DO DONE TK TO DO C TO DO C	

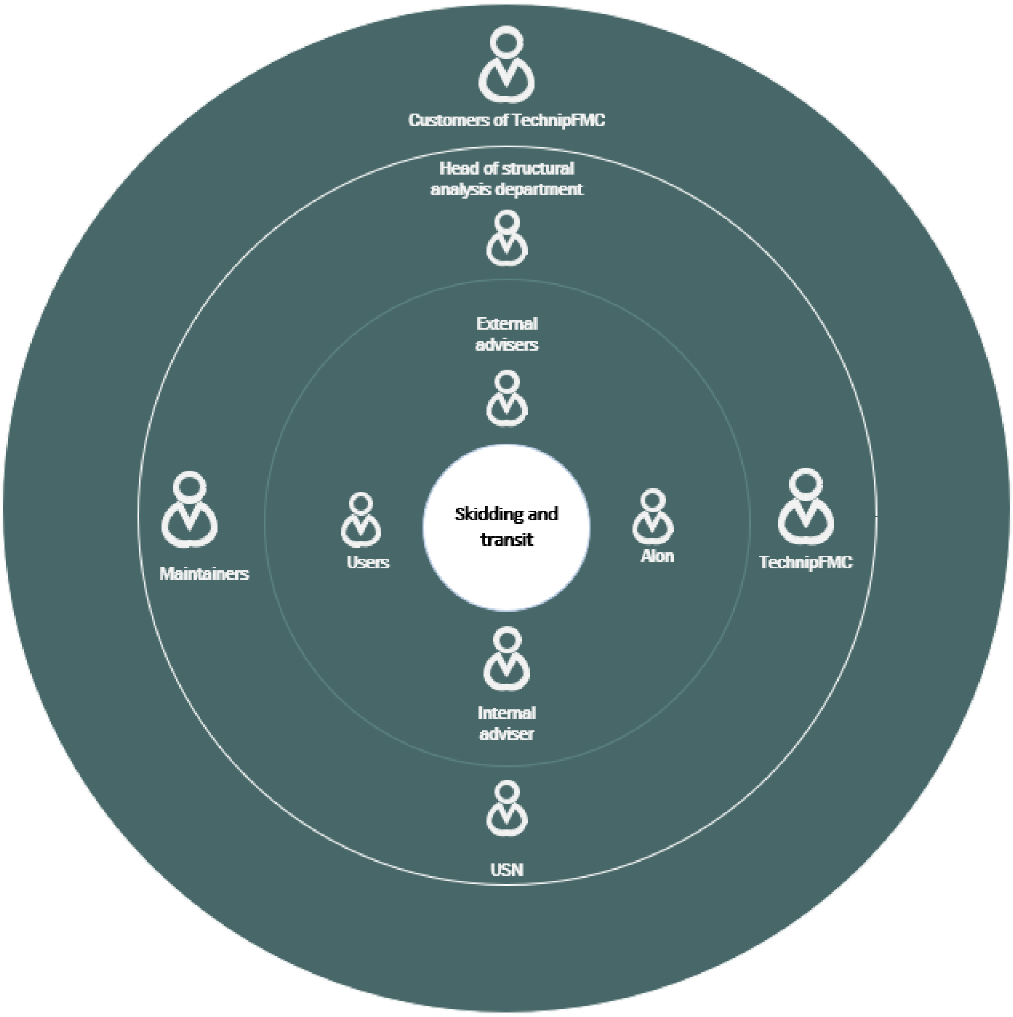
VII: Sprint Review for sprint 7

Description:	Sprint status:	Responsibility / role:
Overview Sprint goal (Finish all analytical mathematical work) Increments What went well and not so well?	Completed. Not met. Goal will be transferred to an increment for next sprint. 4 increments put over to next sprint. Great teamwork. Database work finished and went well. The tests we ran had better results than expected.	All members Adrian and Marte Liv Marte and Tormod
Motivation coming into sprint 8. Planning of sprint 8.	Motivation is good. Ready for the next sprint. Completed.	Everybody.
 ALON-20 Testing ☑ ~~ALON-46~~ Database and mapping ALON-53 Explanation of what evrything is in the main report ~~ALON-55~~ Mathematics test ~~ALON-57~~ RAO and weather data implementation in Python code ~~ALON-62~~ Testdata implementation ~~ALON-65~~ Database page (GUI) ALON-66 Config-page (GUI) ~~ALON-67~~ CoG implementation in Python code ~~ALON-68~~ Advanced database functionality ALON-69 Finish GUI ~~ALON-71~~ Sprint planning	 IN PROGRESS TK DONE MM TO DO DONE DONE DONE MM DONE C IN PROGRESS C DONE TK DONE IN PROGRESS C DONE	

VIII: Sprint review for sprint 8.

Overview Sprint goal (Finish software and main report) Increments What went well and not so well?	Completed. Goal not met. (Focus on finishing the software before the document) Three increments pulled over to the next sprint. Beta test with external adviser went well.	Everyone. Adrian and Liv Marte
Motivation coming into sprint 9. Planning of sprint 9.	Motivation is good. Ready for the next sprint. Completed.	
 ~~ALON-20~~ Testing ~~ALON-53~~ Explanation of what evrything is in the main report ALON-54 Finish main report ALON-56 Finish all testing ~~ALON-66~~ Config-page (GUI) ~~ALON-69~~ Finish GUI ALON-72 Revisit areas that are not working as intended ~~ALON-73~~ Sprint planning ~~ALON-74~~ Code and implement capacity for pressure tests ~~ALON-75~~ Bending moment calculations implemented in Python	 DONE DONE IN PROGRESS IN PROGRESS DONE C DONE C IN PROGRESS DONE DONE LO DONE TK	

Appendix C: Stakeholder Analysis Diagram



Appendix D: User stories

User Stories						
Date:	**User story ID:**	**Category:**	**Role:**	**Purpose:**	**Result:**	**Related requirement:**
22.04.23	**US.cal.01**	Calculations	**As a:** User	**I want to:** Be able to input data for a regular wave pattern.	**So that:** I can get the results without having to use expensive simulation software.	**Req.cal.01.a**
22.04.23	**US.cal.02**	Calculations	**As a:** User	**I want to:** Be able to do component and entity calculations in the software.	**So that:** I can use the results in other calculations.	**Req.cal.02.a** **Req.cal.03.a** **Req.cal.05.a** **Req.cal.07.c**
22.04.23	**US.cal.03**	Calculations	**As a:** User	**I want to:** Be able to easily convert between metric and imperial units.	**So that:** I can do calculations with data from all over the world.	**Req.cal.04.a**
22.04.23	**US.cal.04**	Calculations	**As a:** User	**I want to:** Be able to insert my own utilization calculations.	**So that:** I have the possibility to change between different ISO standards.	**Req.cal.06.b**
22.04.23	**US.doc.01**	Documentation	**As a:** Developer	**I want to:** Add accessibility aid for the user.	**So that:** The user of the software can seek guidance if they encounter anything in the software they don’t understand how to use.	**Req.doc.01.a**
22.04.23	**US.doc.02**	Documentation	**As a:** User	**I want to:** Be offered guidance for the use of the software in case I encounter a problem.	**So that:** I can understand how to operate the software.	**Req.doc.04.c**
22.04.23	**US.doc.03**	Documentation	**As a:** User	**I want to:** Have the option to export the results of the analysis.	**So that:** I have a way to document and save the output.	**Req.doc.02.a**

22.04.23	**US.doc.04**	Documentation	**As a:** User	**I want to:** Be able to import various file formats for the software to read.	**So that:** There is less work moving essential information from data to the software.	**Req.doc.03.a**
22.04.23	**US.db.01**	Database	**As a:** Developer	**I want to:** Set up a database for the purpose of adding and deleting information.	**So that:** The user will save time and avoid potential mistakes with less manual input for each calculation.	**Req.db.01.a**
22.04.23	**US.db.02**	Database	**As a:** User	**I want to:** Be able to add and remove information about different items from a database.	**So that:** Even if I use the same item in several calculations, I only have to enter it into the software once.	**Req.db.02.a** **Req.db.03.a** **Req.db.04.a** **Req.db.05.b** **Req.db.06.c**
22.04.23	**US.vis.01**	Visuals	**As a:** User	**I want to:** Be given a graphical depiction of the various critical points, forces and movement data calculated by the software.	**So that:** I'll be able to understand the calculations better.	**Req.vis.01.a** **Req.vis.02.a** **Req.vis.03.a** **Req.vis.04.a** **Req.vis.05.a** **Req.vis.06.a** **Req.vis.07.a** **Req.vis.08.c**
22.04.23	**US.safe.01**	Safety	**As a:** User	**I want to:** Be able to display a visual preview of the data I intend to add to a database, before doing so.	**So that:** I can be sure that what I am adding to the database is correct.	**Req.safe.02.c**
22.04.23	**US.safe.02**	Safety	**As a:** Developer	**I want to:** View the input from the user in an organized manner after each stage of an analysis.	**So that:** They can check for error before submitting the data.	**Req.safe.01.a**

Appendix E: Requirements

I: Calculations

Requirement – ID:	**Priority:**	**Originator:**	**Date:**
Req.cal.01.a	A	TechnipFMC	22.04.23
Description: The results from the analysis, based on regular wave pattern, must match the results from TechnipFMC own analysis, based on regular wave pattern.			
Category:	Calculations		
Related Test – ID:	Test.cal.01.a		
Status:	Not approved		

Requirement – ID:	**Priority:**	**Originator:**	**Date:**
Req.cal.02.a	A	TechnipFMC	22.04.23
Description: The software must be able to calculate component capacities and account for variables.			
Category:	Calculations		
Related Test – ID:	Test.cal.02.a		
Status:	Not approved.		

Requirement – ID:	**Priority:**	**Originator:**	**Date:**
Req.cal.03.a	A	TechnipFMC	22.04.23
Description: The software must be able to determine the COG of all components as one entity.			
Category:	Calculations		
Related Test – ID:	Test.cal.03.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.cal.04.a	A	TechnipFMC	22.04.23

Description:
The software must have the possibility to convert between metric and imperial units.

Category:	Calculations
Related Test – ID:	Test.cal.04.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.cal.05.a	A	TechnipFMC	22.04.23

Description:
The software must provide data on spare capacity for pressure testing on stack, while in transit.

Category:	Calculations
Related Test – ID:	Test.cal.05.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.cal.06.b	B	TechnipFMC	22.04.23

Description:
The software should allow you to insert your own utilization calculations.

Category:	Calculations
Related Test – ID:	Test.cal.06.b
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.cal.07.c	C	TechnipFMC	22.04.23

Description:
The software should have the ability to add forces working on the different attachment points when the stack is in transit.

Category:	Calculations
Related Test – ID:	Test.cal.07.c
Status:	Not approved

II: Documentation

Requirement – ID:	Priority:	Originator:	Date:
Req.doc.01.a	A	TechnipFMC	22.04.23
Description: The software must provide an accessibility aid for easy navigation of the software.			
Category:	Documentation		
Related Test – ID:	Test.doc.01.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.doc.02.a	A	TechnipFMC	22.04.23
Description: The software must have a feature that lets you save the results of the analysis as a file in LaTex format.			
Category:	Documentation		
Related Test – ID:	Test.doc.02.a		
Status:	Not approved.		

Requirement – ID:	Priority:	Originator:	Date:
Req.doc.03.a	A	Alon	22.04.23
Description: The software must have a standardized .xlsx file format for RAO input.			
Category:	Documentation		
Related Test – ID:	Test.doc.03.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.doc.04.c	C	TechnipFMC	22.04.23
Description: The software should have a user manual that describes the software's features and functions in detail.			
Category:	Documentation		
Related Test – ID:	Test.doc.04.c		
Status:	Not approved		

III: Database

Requirement – ID:	Priority:	Originator:	Date:
Req.db.01.a	A	TechnipFMC	22.04.23

Description:
The software must have access to a database for local storage.

Category:	Database
Related Test – ID:	Test.db.01.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.db.02.a	A	TechnipFMC	22.04.23

Description:
The software must have a feature for adding new components to the database.

Category:	Database
Related Test – ID:	Test.db.02.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.db.03.a	A	TechnipFMC	22.04.23

Description:
The software must have a feature for adding new vessels to the database.

Category:	Database
Related Test – ID:	Test.db.03.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.db.04.a	A	TechnipFMC	22.04.23

Description:
The software must have a feature for storing the analysis in the database.

Category:	Database
Related Test – ID:	Test.db.04.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.db.05.b	B	Alon	22.04.23

Description:
The software should have a feature for adding new material densities to the database.

Category:	Database
Related Test – ID:	Test.db.05.b
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.db.06.c	C	Alon	22.04.23

Description:
The software must have a feature for adding ISO standard for utilization calculations to the database.

Category:	Database
Related Test – ID:	Test.db.06.c
Status:	Not approved

IV: Visuals

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.01.a	A	TechnipFMC	22.04.23

Description:
Based on the skidding calculations, the software must show a picture of the critical points on the components.

Category:	Visuals
Related Test – ID:	Test.vis.01.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.02.a	A	TechnipFMC	22.04.23

Description:
Based on the skidding calculations, the software must show a picture of the forces acting on the components.

Category:	Visuals
Related Test – ID:	Test.vis.02.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.03.a	A	TechnipFMC	22.04.23

Description:
The software must show a graphical depiction of the 2D – renderings of the components.

Category:	Visuals
Related Test – ID:	Test.vis.03.a
Status:	Not approved

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.04.a	A	TechnipFMC	22.04.23
Description: The 2D – renderings must have a front view.			
Category:	Visuals		
Related Test – ID:	Test.vis.04.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.05.a	A	TechnipFMC	22.04.23
Description: The 2D – renderings must have a side view.			
Category:	Visuals		
Related Test – ID:	Test.vis.05.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.06.a	A	TechnipFMC	22.04.23
Description: The software must show a depiction of critical points on the components, based on the transit calculations.			
Category:	Visuals		
Related Test – ID:	Test.vis.01.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.07.a	A	TechnipFMC	22.04.23
Description: The software must show a graphical depiction of forces working on the components, based on the transit calculations.			
Category:	Visuals		
Related Test – ID:	Test.vis.02.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.vis.08.c	C	TechnipFMC	22.04.23

Description:
When entering movement information into the software, the software should give a graphic visual of how the boat moves in correlation to the waves.

Category:	Visuals
Related Test – ID:	Test.vis.06.c
Status:	Not approved

V: Safety

Requirement – ID:	Priority:	Originator:	Date:
Req.safe.01.a	A	Alon	22.04.23
Description: After each stage of the analysis, a summary message must be displayed.			
Category:	Safety		
Related Test – ID:	Test.safe.01.a		
Status:	Not approved		

Requirement – ID:	Priority:	Originator:	Date:
Req.safe.02.c	C	TechnipFMC	22.04.23
Description: When entering input into the database, the software should give a preview of the input data before adding it to the database.			
Category:	Visuals		
Related Test – ID:	Test.safe.02.c		
Status:	Not approved		

Appendix F: Test

I: Calculations

Test – ID:	Priority:	Originator:	Date:
Test.cal.01.a	A	TechnipFMC	22.04.23
Description: Provide the software with all the information necessary to get an analysis of a regular wave pattern, then review the results.			
Acceptance criteria: When the regular wave pattern calculations from the software and those generated by TechnipFMC match, the test is deemed successful.			
Category:	Calculations		
Related Requirement – ID:	Req.cal.01.a		
Status:	Not tested.		

Test – ID:	Priority:	Originator:	Date:
Test.cal.02.a	A	TechnipFMC	22.04.23
Description: Provide the software with all the information necessary to determine the component capacity, then review the results.			
Acceptance criteria: When the capacity calculations from the software and those generated by TechnipFMC match, the test is deemed successful.			
Category:	Calculations		
Related Requirement – ID:	Req.cal.02.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.cal.03.a	A	TechnipFMC	22.03.23
Description: Give the software all the information necessary to calculate the COG of all components combined, then evaluate the results.			
Acceptance criteria: When the calculations for COG from the software match those generated by TechnipFMC, the test is deemed successful.			
Category:	Calculations		
Related Requirement – ID:	Req.cal.03.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.cal.04.a	A	TechnipFMC	22.04.23
Description: Run several tests using various imperial and metric values, then compare the results.			
Acceptance criteria: When the software successfully converts data from the various types of units, the test is approved.			
Category:	Calculations		
Related Requirement – ID:	Req.cal.04.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.cal.05.a	A	TechnipFMC	22.04.23
Description: Enter input into the software, to run a calculation on capacity for pressure testing of stack, when in transit.			
Acceptance criteria: When data is entered into the software for a pressure test capacity, and the software provides details for pressure test capacity, the test is approved.			
Category:	Calculations		
Related Requirement – ID:	Req.cal.05.a		
Status:	Not tested		

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.cal.06.b</td><td>B</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
As part of the input, enter your own calculations and evaluate the outcome.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is deemed successful when the software enables you to input your own calculations and then incorporate those calculations in the final result.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Calculations</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.cal.06.b</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.cal.07.c</td><td>C</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Provide the software with all the details required to add the forces working on the attachment points while the stack is in transit and evaluate the outcome.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is deemed successful when the forces working on the attachment points are added to/displayed on the stack.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Calculations</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.cal.07.c</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

II: Documentation

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.doc.01.a</td><td>A</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Verify whether the accessibility tool makes it easier for the user to utilize the software.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is deemed successful when a user with an engineering background can use the software with only the aid from the accessibility tool.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Documentation</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.doc.01.a</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.doc.02.a</td><td>A</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Export the analysis results.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is deemed successful when the software can produce a LaTex file with all the required data appropriately arranged.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Documentation</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.doc.02.a</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

Test – ID:	Priority:	Originator:	Date:
Test.doc.03.a	A	TechnipFMC	22.04.23

Description:
Import a .xlsx file into the software.

Acceptance criteria:
The test is approved when the software can retrieve data from a .xlsx file with RAO data. The RAO data imported must match the original file data.

Category:	Documentation
Related Requirement – ID:	Req.doc.03.a
Status:	Not tested

Test – ID:	Priority:	Originator:	Date:
Test.doc.04.c	C	TechnipFMC	22.04.23

Description:
Verify whether the software can be utilized with the help of a user manual.

Acceptance criteria:
The test is deemed successful when a user with an engineering background can operate the software with just the aid of the user manual.

Category:	Documentation
Related Requirement – ID:	Req.doc.04.c
Status:	Not tested

III: Database

Test – ID:	Priority:	Originator:	Date:
Test.db.01.a	A	TechnipFMC	22.04.23
Description: Test the connection to the database.			
Acceptance criteria: The test is approved when access is granted and information from the database is displayed for the user.			
Category:	Database		
Related Requirement – ID:	Req.db.01.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.db.02.a	A	TechnipFMC	22.04.23
Description: Test the database by adding component data.			
Acceptance criteria: The test is approved when the component data is stored in the component database.			
Category:	Database		
Related Requirement – ID:	Req.db.02.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.db.03.a	A	TechnipFMC	22.04.23

Description:
Test the database by adding vessel data.

Acceptance criteria:
The test is approved when the vessel data is stored in the vessel database.

Category:	Database
Related Requirement – ID:	Req.db.03.a
Status:	Not tested

Test – ID:	Priority:	Originator:	Date:
Test.db.04.a	A	TechnipFMC	22.04.23

Description:
Test the database by adding the analysis to the database.

Acceptance criteria:
The test is approved when the analysis is stored in the analysis database.

Category:	Database
Related Requirement – ID:	Req.db.04.a
Status:	Not tested

Test – ID:	Priority:	Originator:	Date:
Test.db.05.b	B	Alon	22.04.23

Description:
Test the database by adding density data.

Acceptance criteria:
The test is approved when the density data is stored in the density database.

Category:	Database
Related Requirement – ID:	Req.db.05.b
Status:	Not tested

Test – ID:	Priority:	Originator:	Date:
Test.db.06.c	C	Alon	22.04.23

Description:
Test the database by adding ISO standards to the database.

Acceptance criteria:
The test is approved when the function is stored in the database.

Category:	Database
Related Requirement – ID:	Req.db.06.c
Status:	Not tested

IV: Visuals

Test – ID:	Priority:	Originator:	Date:
Test.vis.01.a	A	TechnipFMC	22.04.23
Description: Give the software the data related to displaying the critical points of a component and check the results.			
Acceptance criteria: The test is approved when the software can display a depiction of the critical points of components correctly.			
Category:	Visuals		
Related Requirement – ID:	Req.vis.01.a, Req.vis.06.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.vis.02.a	A	TechnipFMC	22.04.23
Description: Give the software the data related to displaying the forces working on a component and check the results.			
Acceptance criteria: The test is approved when the software can display a depiction of the forces working on the components.			
Category:	Visuals		
Related Requirement – ID:	Req.vis.02.a, Req.vis.07.a		
Status:	Not tested		

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.vis.03.a</td><td>A</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Give the software the data related to displaying 2D – renderings of components and check the results.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is approved when the software can display 2D – renderings of components</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Visuals</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.vis.03.a</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.vis.04.a</td><td>A</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Give the software the data related to displaying any 2D – renderings and check the results.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is approved when the software can display a 2D – rendering of the front view of any given component.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Visuals</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.vis.04.a</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

<table>
<tr><th>Test – ID:</th><th>Priority:</th><th>Originator:</th><th>Date:</th></tr>
<tr><td>Test.vis.05.a</td><td>A</td><td>TechnipFMC</td><td>22.04.23</td></tr>
<tr><td colspan="4"><b>Description:</b>
Give the software the data related to displaying any 2D – renderings and check the results.</td></tr>
<tr><td colspan="4"><b>Acceptance criteria:</b>
The test is approved when the software can display a 2D – rendering of the side view of any given component.</td></tr>
<tr><td><b>Category:</b></td><td colspan="3">Visuals</td></tr>
<tr><td><b>Related Requirement – ID:</b></td><td colspan="3">Req.vis.05.a</td></tr>
<tr><td><b>Status:</b></td><td colspan="3">Not tested</td></tr>
</table>

Test – ID:	Priority:	Originator:	Date:
Test.vis.06.c	C	TechnipFMC	22.04.23

Description:
Check the outcome after providing the software with the movement data from the waves.

Acceptance criteria:
When the boat’s motion and the waves properly adapt to the inputs supplied, the test is deemed successful.

Category:	Visuals
Related Requirement – ID:	Req.vis.08.c
Status:	Not tested

V: Safety

Test – ID:	Priority:	Originator:	Date:
Test.safe.01.a	A	Alon	22.04.23
Description: Input data in the different stages in the software, one page at a time, and check the results.			
Acceptance criteria: If a summary message shows up and displays the input data correctly after each page, the test Is approved.			
Category:	Safety		
Related Requirement – ID:	Req.safe.01.a		
Status:	Not tested		

Test – ID:	Priority:	Originator:	Date:
Test.safe.02.c	C	TechnipFMC	22.04.23
Description: Fill out the database's required fields and press "preview".			
Acceptance criteria: The test is approved when the software can display the information correctly when "preview" is pressed.			
Category:	Visuals		
Related Requirement – ID:	Req.safe.02.c		
Status:	Not tested		

Appendix G: Risk

I. Risk scoring system

Risk = Probability x Severity						
Probability	**5**	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	**2**	**3**	**4**	**5**
Severity						

Risk Scoring System				
1 - 5	6 - 10	11 - 15	16 - 20	21 - 25
Risk = Probability x Severity				

Very high risk	Risk-reducing measures must be implemented immediately
High risk	Risk-reducing measures must be implemented early in the project
Medium risk	Risk-reducing measures must be considered
Low risk	Risk-reducing measures may not be required
Very low risk	Risk-reducing measures are not required

II. Risk scoring system for probability and severity

Definition of the degree of probability		
Degree of probability	**Probability**	**Definition**
1	Very low	Happens very rarely
2	Low	Happens rarely
3	Medium	Happens sometimes
4	High	Happens often
5	Very high	Happens very often

Definition of the degree of Severity		
Degree of severity	**Severity**	**Definition**
1	Insignificant	The project moves forward as usual
2	Less significant	Despite a slight delay, the project's outcome won't suffer much
3	Significant	The project pauses. To proceed, action must be taken.
4	Serious	The project stops. To continue, critical solutions are required.
5	Catastrophic	The project unravels. To proceed, all measures must be implemented.

III: Risk review criteria

Risk Register								
1. Group Risks								
Risk ID	**Risk Description**	**Cause of Risk**	**P**	**S**	**R**	**Consequence**	**Risk Mitigation**	**Avoidance**
1.1	Minor illness within the group	Exposure to sickness	5	2	10	Slightly more difficult to collaborate when we are not in the same room	Join group on Microsoft Teams if needed and work from home	Wash hands and maintain proper hygiene. If you feel sick stay at home to avoid infecting the rest of the team members
1.2	A group member dies	A lethal encounter	1	5	5	The remaining group members will get a bigger workload	We will have to accept it and take care of each other	Avoid activity that could lead to one's own demise
1.3	Severe disagreement or arguments within the group	Misunderstanding and miscommunication	2	4	8	Bad work environment and reduced motivation	The group must conduct the rules stated in the group contract	Maintain a healthy relationship with good communication
1.4	The group split up	Severe argument or disagreement	1	5	5	The project will not be completed	The group must conduct the rules stated in the group contract	Maintain a healthy relationship with good communication
1.5	One of the group members loses someone in their family	A lethal encounter	3	3	9	The member in question will be severely demotivated	The remaining group members must ensure a warm and including work environment	Unavoidable
1.6	Group member(s) keep failing to arrive on time for meetings	Bad planning or no motivation	3	3	9	Other team members may lose motivation, resulting in less work being completed	The group leader should attempt to resolve the problem with the member(s) in question	Make sure everyone is aware of the weekly schedule and understands how critical it is to be on time
1.7	Group member(s) not contributing as much as expected	Missing motivation or conflicts in the group	2	5	10	The remainder of the team will have to share the extra workload on them,	The group leader should attempt to resolve the problem with the	Do teambuilding exercises on a regular basis to keep motivation high and group

						resulting in irritated team members	member(s) in question	chemistry strong
1.8	Group member(s) are not following deadlines	Laciness or having too much work on their hand	3	4	12	More work for the rest of the group, leading to loss of motivation	The group leader will approach the individual and attempt to push them in the right direction or delegate some of their workload	Try to split the amount of work as evenly as possible and try to keep the team motivated
1.9	Loss of group member(s)	Illness, private reasons, or conflicts in the group	1	5	5	The rest of the group will have to split the work amongst them	The team will have to do a lot of planning for the rest of the process and maybe alter the project if it seems impossible to finish	Try to resolve issues before they become critical
1.10	Performance affecting illness within the group	Exposure to sickness	3	4	12	Plenty of extra work for the individuals who are well. Some of the tasks might not be finished by their deadline	The remaining team members must complete their own job in addition to taking on as many of the ill member's obligations as they can	Wash hands and maintain proper hygiene. If you feel sick stay at home to avoid infecting the rest of the team members
1.11	Home with sick child	Exposure to sickness	3	3	12	Some tasks might not be finished in time	Work from home whenever possible or reassign duties to the other team members If necessary	Wash hands and maintain proper hygiene.
1.12	Personal problems	No cause	2	4	8	Can affect motivation and some tasks might not be finished in time	The affected group member(s) must inform the team leader and seek assistance	Unavoidable

							from the other team members if some tasks seem unachievable	
1.13	A family incident impacting performance	No cause	2	4	8	Can affect motivation and some tasks might not be finished in time	The affected group member(s) must inform the team leader and seek assistance from the other team members if some tasks seem unachievable	Unavoidable

Risk Register								
2. Stakeholder Risks								
Risk ID	**Risk Description**	**Cause of Risk**	**P**	**S**	**R**	**Consequence**	**Risk Mitigation**	**Avoidance**
2.1	The final product does not meet the expectations of the stakeholder	The amount of work put into the project is not enough or the task itself is too difficult	2	5	10	Stakeholders' satisfaction will be low, and grades will fall	Hold a meeting where the reasons for the product state are explained	Maintain constant communication with stakeholders and ensure that you both understand what the other expects
2.2	The software is too difficult to use	Lack of guidance or the software is built too complicated	1	4	4	The software will not be more efficient than existing market solutions	Research ways to simplify the software	Use frequent testing to find weaknesses and come up with better solutions
2.3	The final product is not finished	Poor planning and insufficient effort from team members	2	5	10	The software will prove to have no worth to our client	Speak with all stakeholders and inform them that it will not be completed on time	Put in the required effort to catch up if we discover that we are behind on some parts of the project
2.4	Lack of communication between group and stakeholder	Lack of regular updates to stakeholders	1	5	5	The team might not properly comprehend the client's desires	Have a meeting with client to clear up any uncertainties and create a better plan for communication	Hold regular meetings to make sure the needs, wants and wishes are understood by both parts
2.5	Employer cancelling the project	Not needing the product anymore, bankruptcy or not happy with our progress	1	5	5	A new project must be started, which can have an impact on grades	Contact administration at USN and update them on the situation. Must look for a new project	Show the customer respect and communicate with them often throughout the process. Work hard to meet their needs
2.6	Unresponsive client	Busy client	2	5	10	We don't get information or answers in time, which may delay us	Request missing information or answers by email again	Clearly specify the severity of getting information in time

Risk Register								
3. Project Risks								
Risk ID	**Risk Description**	**Cause of Risk**	**P**	**S**	**R**	**Consequence**	**Risk Mitigation**	**Avoidance**
3.1	Part of the project will not be finished in time	Lack of work or too much work	3	3	9	We will have an unfinished product which will not satisfy the client	Speak with all stakeholders and inform them that it will not be completed on time	Manage workload and ensure that no requirements are beyond scope
3.2	Group room becomes unavailable	The group room becomes inhabitable or too damaged to use	1	2	2	Not having a fixed place to gather at all times	The team leader will locate a new place for the group to meet, and in the meantime, the group will need to make use of other physical and digital resources	Maintain the group room with care. Candles and other fire hazards are not allowed
3.3	Hardware becomes unusable	Hardware malfunction	2	3	6	Anything saved locally to the machine may become unusable and could set back progress of the project	The group member(s) in question will have to find a backup for lost functionality	Use all hardware properly and store all documentation on our shared storage disk
3.4	Software becomes unusable	Software malfunction	2	2	4	The software in question may not be used in the final product	The group will have to find an alternative to the malfunctioning software	All group members must make sure to understand proper usage of the software
3.5	Not being able to reach sprint goals on time	A lack of understanding of how long each task would take or the emergence of unknown risks	4	2	8	The goal is not achieved, and the work must be carried over to the following sprint	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling behind on anything else	Make sure that the group are working efficiently and collaboratively to achieve a shared objective
3.6	Lack of documented work from group member(s)	Laciness and lack of motivation	2	5	10	Lower grades if not taken care of before delivering	Group member(s) will be given assignments by the document lead to	All the project's documentation will be closely monitored by

							complete the necessary documentation	the project's document lead
3.7	Having to move tasks to the next or other sprints	A lack of understanding of how long each task would take or the emergence of unknown risks	5	2	10	The work must be carried over to the following sprint	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling behind on anything else	Ensure that everyone is contributing effectively, and assist one another if necessary
3.8	Accidentally deleting documentation	No backup uploaded to our shared storage	1	5	5	Loss of documentation and what was lost must be redone	Look for earlier iteration of the documentation first. Redo the document as fast as possible while it is fresh in memory	Always store all documentation on our shared storage disk
3.9	Significant project changes	Requirements that are not well articulated and/or poor client communication	3	5	15	Previous work being useless and a lot of work to get the new changes in place	Documentation will need to be changed and updated with the reason for the changes	Maintain regular communication with the customer and ensure that the requirements are as specific as possible and that everyone is fully aware of them
3.10	Having unidentified risk(s)	Lack of knowledge	3	3	9	Having to move tasks further down the line when these risks arise	Update risk register with the new risk and have a meeting on how to solve the problem	Update the risk register regularly at the end of each sprint
3.11	Documentation being detected by plagiarism control	Laciness, not using references correctly	2	5	10	Bad grades or failing if not corrected before delivering	Speak to the member that wrote the documentation in question and educate them on proper reference usage. Update the document	Check for plagiarism on all documentation throughout the project
3.12	Documentation lacks	Laciness or lack of knowledge	2	5	10	Failing plagiarism checks	Speak to the member that wrote the	Ensure that references are constantly up

	necessary references						documentation in question and educate them on proper reference usage. Update the document	to date by updating them whenever new information is added to the documentation
3.13	Our chosen project model does not suit our team	Not sticking to scrum or lack of knowledge on how to use it	2	4	8	Teamwork becomes more difficult and end product will suffer	Scrum master will arrange a meeting to go over the framework and the benefits of using it	Scrum master will keep the team educated on scrum through the project and making sure all members work within the framework

Risk Register

4. Technical Risks

Risk ID	Risk Description	Cause of Risk	P	S	R	Consequence	Risk Mitigation	Avoidance
4.1	The mechanical calculations are too complicated	Lack of knowledge on how to simplify calculations	2	3	6	Using a lot of time to find a solution	Have a meeting discussing the problem and try to figure out a possible solution	Have a wide understanding in the area that is being worked on before deciding on a solution
4.2	The software runs slower than what is expected	long calculations or non optimal memory usage	3	4	12	Unhappy client	Go back and redo the calculations in a way that lowers the program's execution time	Research ways to optimize calculation execution in python
4.3	Calculating COG turns out to be a problem	The structures that are being calculated are too difficult	2	5	10	The output from the software will be wrong	Locate the issue and do research to find a solution	Have a wide understanding in the area that is being worked on before starting on a solution
4.4	Some features are missing from the software	Too many requirements	4	2	8	The software will not be optimal	Look at which features are there and discuss whether the right priorities were made or not	Have regular communication with client regarding the level of priority of each feature
4.5	Database integration is not available	Parts of the database does not work properly	2	4	8	The database functionality will not be included	We will try to fix the problem and if that fails we will have to find an alternative to using a database	Research ways to implement databases into the software early
4.6	Animations does not work properly	Lack of knowledge	2	4	8	Animations will not be included	The program will have to display or simulate movement in a different manner	Research ways to make animations in the software early
4.7	Input formats are not possible to use for calculation	The given input files are not in a format that is readable to the program	2	5	10	The input files will need to be reformatted or the user will be required to type in manual input	Check if there is any room for changes in the formatting of the input files	Have good communication with the client and update them when problems arise to discuss them and find solutions
4.8	Faults in the software code	Lack of knowledge	3	1	3	Spending more time than expected on	The faulty part of the code will need to be	Always test your code, and don't write huge

					🟩	that section of code	debugged and redesigned	sections of it before you do

IV: Risk reviews

Risk Review – Sprint Nr 2				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
1.1	**Old:** Illness within the group **New:** 1.1: Minor illness within the group 1.10: Performance affecting illness within the group	Performed mitigation strategy. **Changes:** Severity decreased from 3 to 2. Risk splits into 2 separate risks with different degrees of sickness. Risk updated with new information	Exposure to sickness	**Old:** Work from home if possible **New:** Join group on Microsoft Teams if needed and work from home
2.6	Unresponsive client **(New risk)**	New risk. All associated information regarding this risk has been added to risk register. Performed mitigation strategy.	Busy client	Request missing information or answers by email again
3.10	Having unidentified risk(s)	Performed mitigation strategy.	Lack of knowledge	Update risk register with the new risk and have a meeting on how to solve the problem

Risk Review – Sprint Nr 3				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
1.11	Home with sick child (New risk)	New risk. All associated information regarding this risk has been added to risk register. Performed mitigation strategy.	Exposure to sickness	Work from home whenever possible or reassign duties to the other team members If necessary
1.12	Personal problems (New risk)	New risk. All associated information regarding this risk has been added to risk register. Performed mitigation strategy.	No cause	The affected group member(s) must will inform the team leader and seek assistance from the other team members if some tasks seem unachievable
3.10	Having unidentified risk(s)	Performed mitigation strategy.	Lack of knowledge	Update risk register with the new risk and have a meeting on how to solve the problem

<u>Some changes were made during sprint 3 in the risk register, but this risk did not occur:</u>

Old version:

Risk ID	Risk Description	Cause of Risk	P	S	R	Consequence	Risk Mitigation	Avoidance
4.8	Faults in the software code	Lack of knowledge	5	3	15	Spending more time than expected on that section of code	The faulty part of the code will need to be debugged and redesigned	Always test your code, and don't write huge sections of it before you do

New iteration:

Risk ID	Risk Description	Cause of Risk	P	S	R	Consequence	Risk Mitigation	Avoidance
4.8	Faults in the software code	Lack of knowledge	3	1	3	Spending more time than expected on that section of code	The faulty part of the code will need to be debugged and redesigned	Always test your code, and don't write huge sections of it before you do

Changes:

- Decreased Probability from 5 to 3
- Decreased Severity from 3 to 1

Reason for changes:

In the previous iteration, we considered all programming errors, down to the slightest typos. We have determined that since this would undoubtedly happen, we won't count all insignificant coding errors as risks but the bigger once will still stay as a risk. As a risk only refers to uncertain events, it doesn't seem appropriate to label minor code flaws as a risk since we can be positive that they will occur.

We have also lowered the severity of this risk. As we are still at the beginning of the project and coding errors will continue to happen as we create the software, we have decided to go with severity 1 for the time being. This does not imply that it will remain this low for the duration of the project; rather, it may increase as a result of software failures becoming a greater concern as delivery draws nearer.

Risk Review – Sprint Nr 4				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
1.10	Performance affecting illness within the group	Performed mitigation strategy.	Exposure to sickness	The remaining team members must complete their own job in addition to taking on as many of the ill member's obligations as they can
3.7	Having to move tasks to the next or other sprints	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling behind on anything else

Risk Review – Sprint Nr 5				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
3.5	Not being able to reach sprint goals on time	Performed mitigation strategy. **Changes:** Severity decreased from 4 to 2.	A lack of understanding of how long each task would take or the emergence of unknown risks	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling behind on anything else
3.7	Having to move tasks to the next or other sprints	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	Ensure that everyone is contributing effectively, and assist one another if necessary

Risk Review – Sprint Nr 6				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
3.7	Having to move tasks to the next or other sprints	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	Ensure that everyone is contributing effectively, and assist one another if necessary
3.9	Significant project changes	Performed mitigation strategy.	Requirements that are not well articulated and/or poor client communication	Documentation will need to be changed and updated with the reason for the changes

Risk Review – Sprint Nr 7				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
1.1	Minor illness within the group	Performed mitigation strategy.	Exposure to sickness	Join group on Microsoft Teams if needed and work from home
1.13	A family incident impacting performance **(New risk)**	New risk. All associated information regarding this risk has been added to risk register. Performed mitigation strategy.	No cause	The affected group member(s) must inform the team leader and seek assistance from the other team members if some tasks seem unachievable
3.5	Not being able to reach sprint goals on time	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling

			of unknown risks	behind on anything else
3.7	Having to move tasks to the next or other sprints	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	Ensure that everyone is contributing effectively, and assist one another if necessary
3.9	Significant project changes	Performed mitigation strategy.	Requirements that are not well articulated and/or poor client communication	Documentation will need to be changed and updated with the reason for the changes
3.10	Having unidentified risk(s)	Performed mitigation strategy.	Requirements that are not well articulated, and poor client communication	Update risk register with the new risk and have a meeting on how to solve the problem

Risk Review – Sprint Nr 8				
Risk ID	Risk Description	Status	Cause of Risk	Mitigation
1.8	Group member(s) are not following deadlines	Performed mitigation strategy. Due to project changes that required rework, one deadline was missed.	Laciness or having too much work on their hand	The group leader will approach the individual and attempt to push them in the right direction or delegate some of their workload
3.5	Not being able to reach sprint goals on time	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	The group will assign tasks to all team members and work as efficient as possible to complete the remaining chores without falling behind on anything else
3.7	Having to move tasks to the next or other sprints	Performed mitigation strategy.	A lack of understanding of how long each task would take or the emergence of unknown risks	Ensure that everyone is contributing effectively, and assist one another if necessary
3.9	Significant project changes	Performed mitigation strategy.	Requirements that are not well articulated and/or poor client communication	Documentation will need to be changed and updated with the reason for the changes

Appendix I: Tests

I: Testing

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.01.a	Req.cal.01.a	US.cal.01	Tormod Fostad Kleiv, Liv Marte Olsen	Mathias Hansen	A
User Story	As a **user**, I want to be able to input data for a regular wave pattern, so that I can get the results without having to use expensive simulation software.				
Requirement Description	The results from the analysis, based on regular wave pattern, must match the results from TechnipFMC own analysis, based on regular wave pattern.				
Test Description	Provide the software with all the information necessary to get an analysis of a regular wave pattern, then review the results.				
Condition for Test Approval	When the regular wave pattern calculations from the software and those generated by TechnipFMC match, the test is deemed successful.				
26.04.2023	APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.02.a	Req.cal.02.a	US.cal.02	Tormod Fostad Kleiv	Tormod Fostad Kleiv	A
User Story	As a **user**, I want to be able to do component and entity calculations in the software, so that I can use the results in other calculations.				
Requirement Description	The software must be able to calculate component capacities and account for variables.				
Test Description	Provide the software with all the information necessary to determine the component capacity, then review the results.				
Condition for Test Approval	When the capacity calculations from the software and those generated by TechnipFMC match, the test is deemed successful.				
02.05.2023	APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.03.a	Req.cal.03.a	US.cal.02	Tormod Fostad Kleiv	Tormod Fostad Kleiv	A
User Story	As a **user**, I want to be able to do component and entity calculations in the software, so that I can use the results in other calculations.				
Requirement Description	The software must be able to determine the COG of all components as one entity.				
Test Description	Give the software all the information necessary to calculate the COG of all components combined, then evaluate the results.				
Condition for Test Approval	When the calculations for COG from the software match those generated by TechnipFMC, the test is deemed successful.				
20.04.2023	APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.04.a	Req.cal.04.a	US.cal.03	Liv Marte Olsen	-	A
User Story	As a **user**, I want to be able to easily convert between metric and imperial units, so that I can do calculations with data from all over the world.				
Requirement Description	The software must have the possibility to convert between metric and imperial units.				
Test Description	Run several tests using various imperial and metric values, then compare the results.				
Condition for Test Approval	When the software successfully converts data from the various types of units, the test is approved.				
20.05.2023	PARTLY APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.05.a	Req.cal.05.a	US.cal.02	Liv Marte Olsen	Liv Marte Olsen	A
User Story	As a **user** I want to be able to do component and entity calculations in the software so that I can use the results in other calculations.				
Requirement Description	The software must provide data on spare capacity for pressure testing on stack, while in transit.				
Test Description	Enter input into the software, to run a calculation on capacity for pressure testing of stack, when in transit.				
Condition for Test Approval	When data is entered into the software for a pressure test, and the results match those of TechnipFMC, the test is approved.				
02.05.2023	APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.06.b	Req.cal.06.b	US.cal.02	Adrian Chan Bjørnson Ruud	Adrian Chan Bjørnson Ruud	B
User Story	As a **user**, I want to be able to do component and entity calculations in the software, so that I can use the results in other calculations.				
Requirement Description	The software should allow you to insert your own utilization calculations.				
Test Description	As part of the input, enter your own calculations and evaluate the outcome.				
Condition for Test Approval	The test is deemed successful when the software enables you to input your own calculations and then incorporate those calculations in the final result.				
04.05.2023	APPROVED				

Calculation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.cal.07.c	Req.cal.07.c	US.cal.02	-	-	C
User Story	As a **user** I want to be able to do component and entity calculations in the software so that I can use the results in other calculations.				
Requirement Description	The software should have the ability to add forces working on the different attachment points when the stack is in transit.				
Test Description	Provide the software with all the details required to add the forces working on the attachment points while the stack is in transit and evaluate the outcome.				
Condition for Test Approval	The test is deemed successful when the forces working on the attachment points are added to/displayed on the stack.				
	NOT TESTED				

Dokumentation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.doc.01.a	Req.doc.01.a	US.doc.01	Adrian Chan Bjørnson Ruud	Adrian Chan Bjørnson Ruud	A
User Story	As a **developer**, I want to add accessibility aid for the user, so that the user of the software can seek guidance if they encounter anything in the software they don't understand how to use.				
Requirement Description	The software must provide an accessibility aid for easy navigation of the software.				
Test Description	Verify whether the accessibility tool makes it easier for the user to utilize the software.				
Condition for Test Approval	The test is deemed successful when a user with an engineering background can use the software with only the aid from the accessibility tool.				
12.05.2023	APPROVED				

Dokumentation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.doc.02.a	Req.doc.02.a	US.doc.03	Liv Marte Olsen, Adrian Chan Bjørnson Ruud	Liv Marte Olsen, Adrian Chan Bjørnson Ruud	A
User Story	As a **user**, I want to have the option to export the results of the analysis, so that I have a way to document and save the output.				
Requirement Description	The software must have a feature that lets you save the results of the analysis as a file in Latex format.				
Test Description	Export the analysis results.				
Condition for Test Approval	The test is deemed successful when the software can produce a Latex file with all the required data appropriately arranged.				
03.05.2023	APPROVED				

Dokumentation Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.doc.03.a	Req.doc.03.a	US.doc.04	Adrian Chan Bjørnson Ruud	Adrian Chan Bjørnson Ruud	A
User Story	As a **user**, I want to be able to import various file formats for the software to read, so that there is less work moving essential information from data to the software.				
Requirement Description	The software must have a standardized .xlsx file format for RAO input.				
Test Description	Import a .xlsx file into the software.				
Condition for Test Approval	The test is approved when the software can retrieve data from a .xlsx file with RAO data. The RAO data imported must match the original file data.				
28.03.2023	APPROVED				

<table>
<tr><th colspan="6">Dokumentation Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.doc.04.c</td><td>Req.doc.04.c</td><td>US.doc.02</td><td>Adrian Chan Bjørnson Ruud</td><td>Adrian Chan Bjørnson Ruud</td><td>C</td></tr>
<tr><td>User Story</td><td colspan="5">As a **user**, I want to be offered guidance for the use of the software in case I encounter a problem, so that I can understand how to operate the software.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">The software should have a user manual that describes the software's features and functions in detail.</td></tr>
<tr><td>Test Description</td><td colspan="5">Verify whether the software can be utilized with the help of a user manual.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">The test is deemed successful when a user with an engineering background can operate the software with just the aid of the user manual.</td></tr>
<tr><td>12.05.2023</td><td colspan="5">APPROVED</td></tr>
</table>

<table>
<tr><th colspan="6">Database Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.db.01.a</td><td>Req.db.01.a</td><td>US.db.01</td><td>Marte Marheim</td><td>Marte Marheim</td><td>A</td></tr>
<tr><td>User Story</td><td colspan="5">As a **developer**, I want to set up a database for the purpose of adding and deleting information, so that the user will save time and avoid potential mistakes with less manual input for each calculation.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">The software must have access to a database for local storage.</td></tr>
<tr><td>Test Description</td><td colspan="5">Test the connection to the database.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">The test is approved when access is granted and information from the database is displayed for the user.</td></tr>
<tr><td>12.04.2023</td><td colspan="5">APPROVED</td></tr>
</table>

Database Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.db.02.a	Req.db.02.a	US.db.02	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be able to add and remove information about different items from a database, so that even if I use the same item in several calculations, I only have to enter it into the software once.				
Requirement Description	The software must have a feature for adding new components to the database.				
Test Description	Test the database by adding component data.				
Condition for Test Approval	The test is approved when the component data is stored in the component database.				
17.04.2023	APPROVED				

Database Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.db.03.a	Req.db.03.a	US.db.02	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be able to add and remove information about different items from a database, so that even if I use the same item in several calculations, I only have to enter it into the software once.				
Requirement Description	The software must have a feature for adding new vessels to the database.				
Test Description	Test the database by adding vessel data.				
Condition for Test Approval	The test is approved when the vessel data is stored in the vessel database.				
17.04.2023	APPROVED				

Database Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.db.04.a	Req.db.04.a	US.db.02	-	-	A
User Story	As a **user**, I want to be able to add and remove information about different items from a database, so that even if I use the same item in several calculations, I only have to enter it into the software once.				
Requirement Description	The software must have a feature for storing the analysis in the database.				
Test Description	Test the database by adding the analysis to the database.				
Condition for Test Approval	The test is approved when the analysis is stored in the analysis database.				
	NOT TESTED				

Database Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.db.05.b	Req.db.05.b	US.db.02	Marte Marheim	Marte Marheim	B
User Story	As a **user**, I want to be able to add and remove information about different items from a database, so that even if I use the same item in several calculations, I only have to enter it into the software once.				
Requirement Description	The software should have a feature for adding new material densities to the database.				
Test Description	Test the database by adding density data.				
Condition for Test Approval	The test is approved when the density data is stored in the density database.				
21.04.2023	APPROVED				

<table>
<tr><th colspan="6">Database Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.db.06.c</td><td>Req.db.06.c</td><td>US.db.02</td><td>Marte Marheim, Adrian Chan Bjørnson Ruud</td><td>Marte Marheim, Adrian Chan Bjørnson Ruud</td><td>C</td></tr>
<tr><td>User Story</td><td colspan="5">As a **user**, I want to be able to add and remove information about different items from a database, so that even if I use the same item in several calculations, I only have to enter it into the software once.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">The software must have a feature for adding ISO standard for utilization calculations to the database.</td></tr>
<tr><td>Test Description</td><td colspan="5">Test the database by adding ISO standards to the database.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">The test is approved when the function is stored in the database.</td></tr>
<tr><td>21.04.2023</td><td colspan="5">APPROVED</td></tr>
</table>

<table>
<tr><th colspan="6">Visual Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.vis.01.a</td><td>Req.vis.01.a</td><td>US.vis.01</td><td>Marte Marheim</td><td>Marte Marheim</td><td>A</td></tr>
<tr><td>User Story</td><td colspan="5">As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">Based on the skidding calculations, the software must show a picture of the critical points on the components.</td></tr>
<tr><td>Test Description</td><td colspan="5">Give the software the data related to displaying the critical points of a component and check the results.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">The test is approved when the software can display a depiction of the critical points of components correctly.</td></tr>
<tr><td>13.05.2023</td><td colspan="5">APPROVED</td></tr>
</table>

Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.02.a	Req.vis.02.a	US.vis.01	-	-	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	Based on the skidding calculations, the software must show a picture of the forces acting on the components.				
Test Description	Give the software the data related to displaying the forces working on a component and check the results.				
Condition for Test Approval	The test is approved when the software can display a depiction of the forces working on the components.				
	NOT TESTED				

Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.03.a	Req.vis.03.a	US.vis.01	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	The software must show a graphical depiction of the 2D – renderings of the components.				
Test Description	Give the software the data related to displaying 2D – renderings of components and check the results.				
Condition for Test Approval	The test is approved when the software can display 2D – renderings of components.				
05.05.2023	APPROVED				

Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.04.a	Req.vis.04.a	US.vis.01	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	The 2D – renderings must have a front view.				
Test Description	Give the software the data related to displaying any 2D – renderings and check the results.				
Condition for Test Approval	The test is approved when the software can display a 2D – rendering of the front view of any given component.				
05.05.2023	APPROVED				

Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.05.a	Req.vis.05.a	US.vis.01	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	The 2D – renderings must have a side view.				
Test Description	Give the software the data related to displaying any 2D – renderings and check the results.				
Condition for Test Approval	The test is approved when the software can display a 2D – rendering of the side view of any given component.				
05.05.2023	APPROVED				

Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.01.a	Req.vis.06.a	US.vis.01	Marte Marheim	Marte Marheim	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	The software must show a depiction of critical points on the components, based on the transit calculations.				
Test Description	Give the software the data related to displaying the critical points of a component and check the results.				
Condition for Test Approval	The test is approved when the software can display a depiction of the critical points of components correctly.				
13.05.2023	APPROVED				

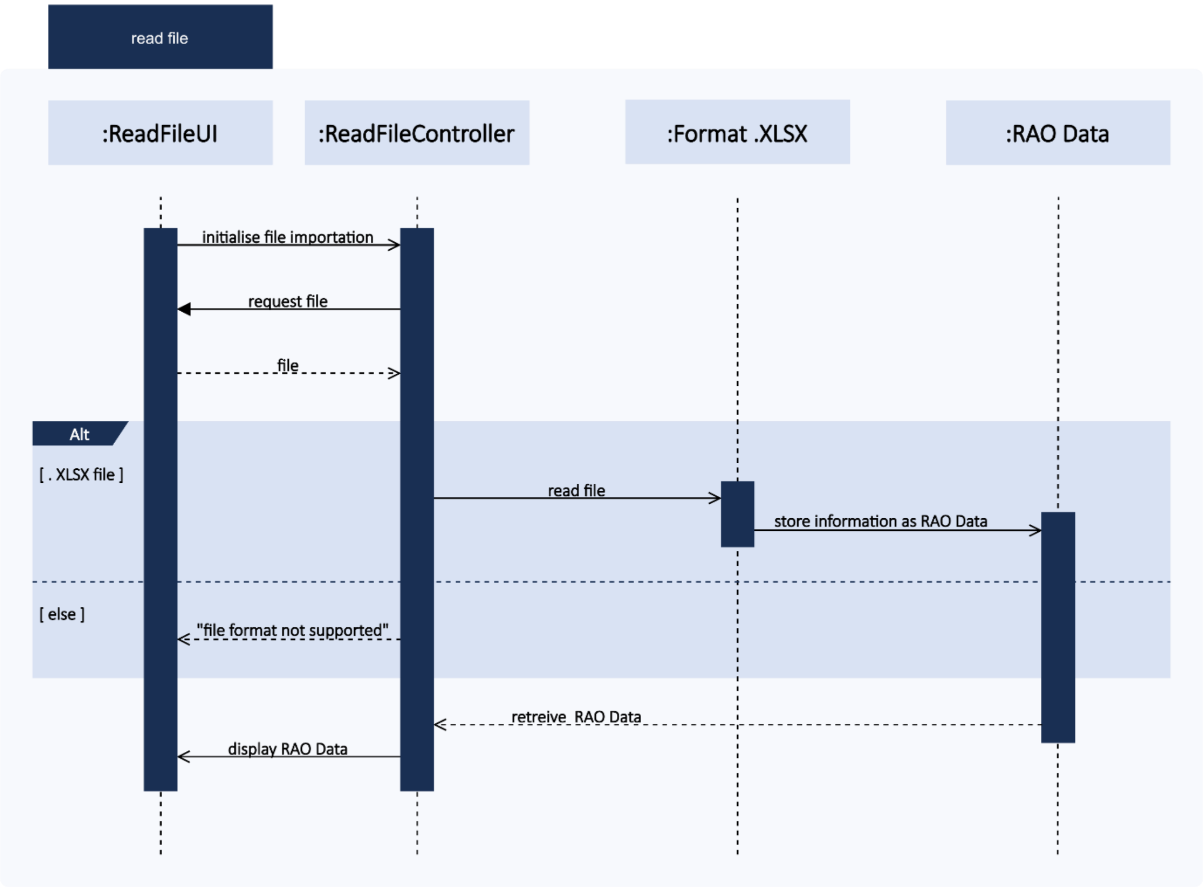
Visual Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.vis.02.a	Req.vis.07.a	US.vis.01	-	-	A
User Story	As a **user**, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I'll be able to understand the calculations better.				
Requirement Description	Based on the skidding calculations, the software must show a picture of the forces acting on the components.				
Test Description	Give the software the data related to displaying the forces working on a component and check the results.				
Condition for Test Approval	The test is approved when the software can display a depiction of the forces working on the components.				
	NOT TESTED				

<table>
<tr><th colspan="6">Visual Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.vis.06.c</td><td>Req.vis.08.c</td><td>US.vis.01</td><td>-</td><td>-</td><td>C</td></tr>
<tr><td>User Story</td><td colspan="5">As a <b>user</b>, I want to be given a graphical depiction of the various critical points, forces and movement data calculated by the software, so that I’ll be able to understand the calculations better.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">When entering movement information into the software, the software should give a graphic visual of how the boat moves in correlation to the waves.</td></tr>
<tr><td>Test Description</td><td colspan="5">Check the outcome after providing the software with the movement data from the waves.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">When the boat’s motion and the waves properly adapt to the inputs supplied, the test is deemed successful.</td></tr>
<tr><td></td><td colspan="5">NOT TESTED</td></tr>
</table>

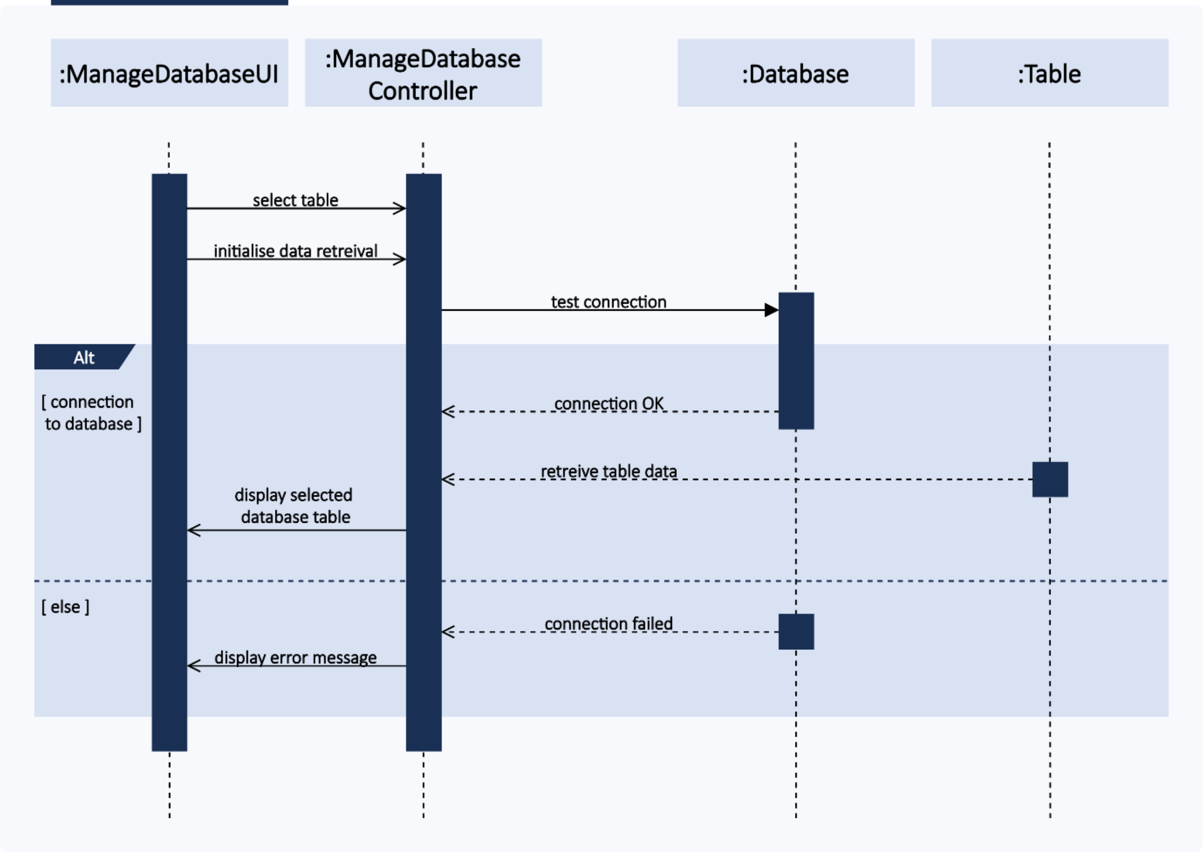
<table>
<tr><th colspan="6">Safety Tests</th></tr>
<tr><th>Test ID</th><th>Requirement ID</th><th>User Story ID</th><th>Test Performed By</th><th>Test Approved By</th><th>Priority</th></tr>
<tr><td>Test.safe.01.a</td><td>Req.safe.01.a</td><td>US.safe.02</td><td>-</td><td>-</td><td>A</td></tr>
<tr><td>User Story</td><td colspan="5">As a <b>developer</b>, I want to view the input from the user in an organized manner after each stage of an analysis, so that they can check for error before submitting the data.</td></tr>
<tr><td>Requirement Description</td><td colspan="5">After each stage of the analysis, a summary message must be displayed.</td></tr>
<tr><td>Test Description</td><td colspan="5">Input data in the different stages in the software, one page at a time, and check the results.</td></tr>
<tr><td>Condition for Test Approval</td><td colspan="5">If a summary message shows up and displays the input data correctly after each page, the test Is approved.</td></tr>
<tr><td></td><td colspan="5">DELETED</td></tr>
</table>

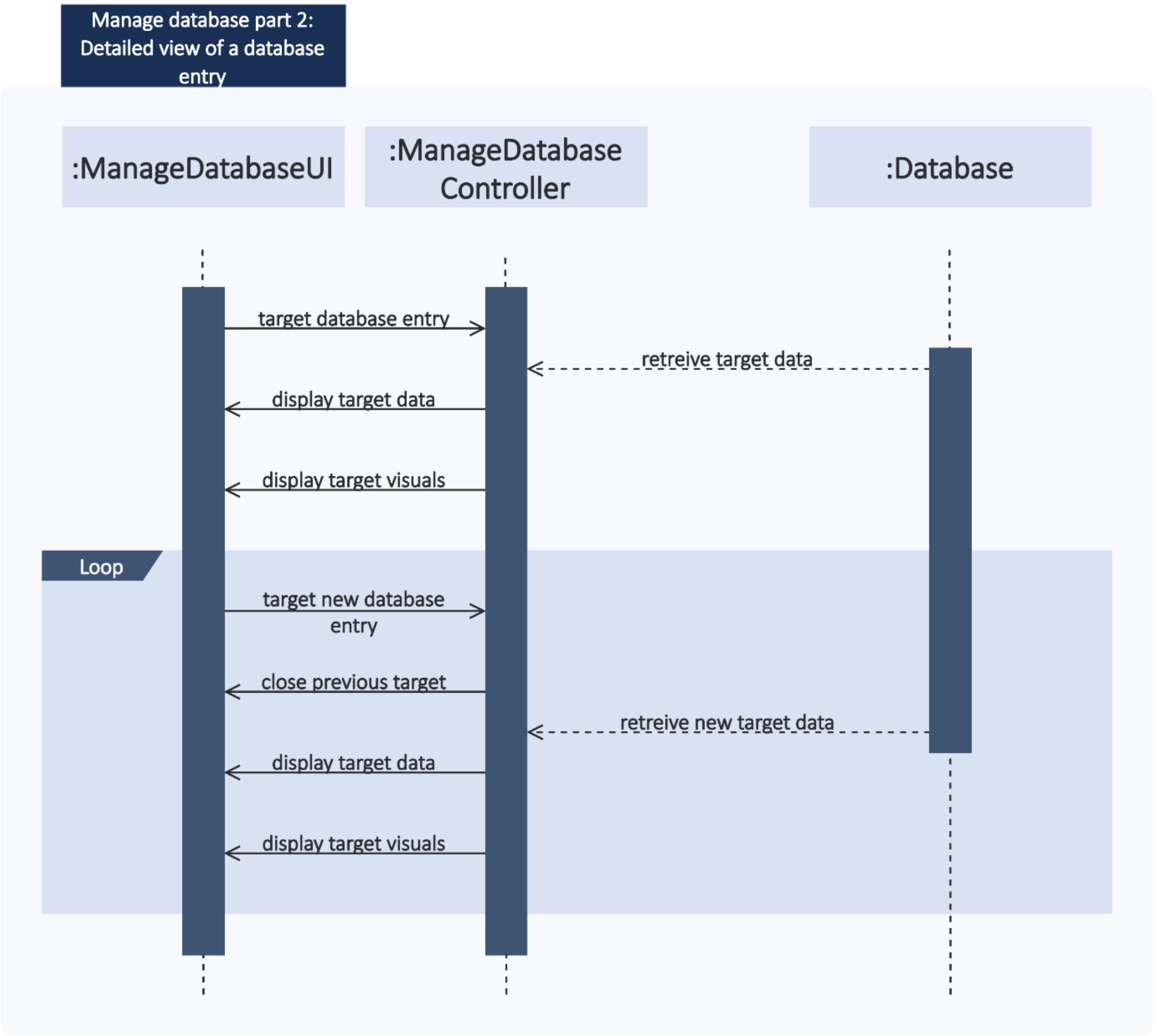
Safety Tests					
Test ID	Requirement ID	User Story ID	Test Performed By	Test Approved By	Priority
Test.safe.02.c	Req.safe.02.c	US.safe.01	-	-	C
User Story	As a **user**, I want to be able to display a visual preview of the data I intend to add to a database, before doing so, so that I can be sure that what I am adding to the database is correct.				
Requirement Description	When entering input into the database, the software should give a preview of the input data before adding it to the database.				
Test Description	Fill out the database's required fields and press "preview".				
Condition for Test Approval	The test is approved when the software can display the information correctly when "preview" is pressed.				
	DELETED				

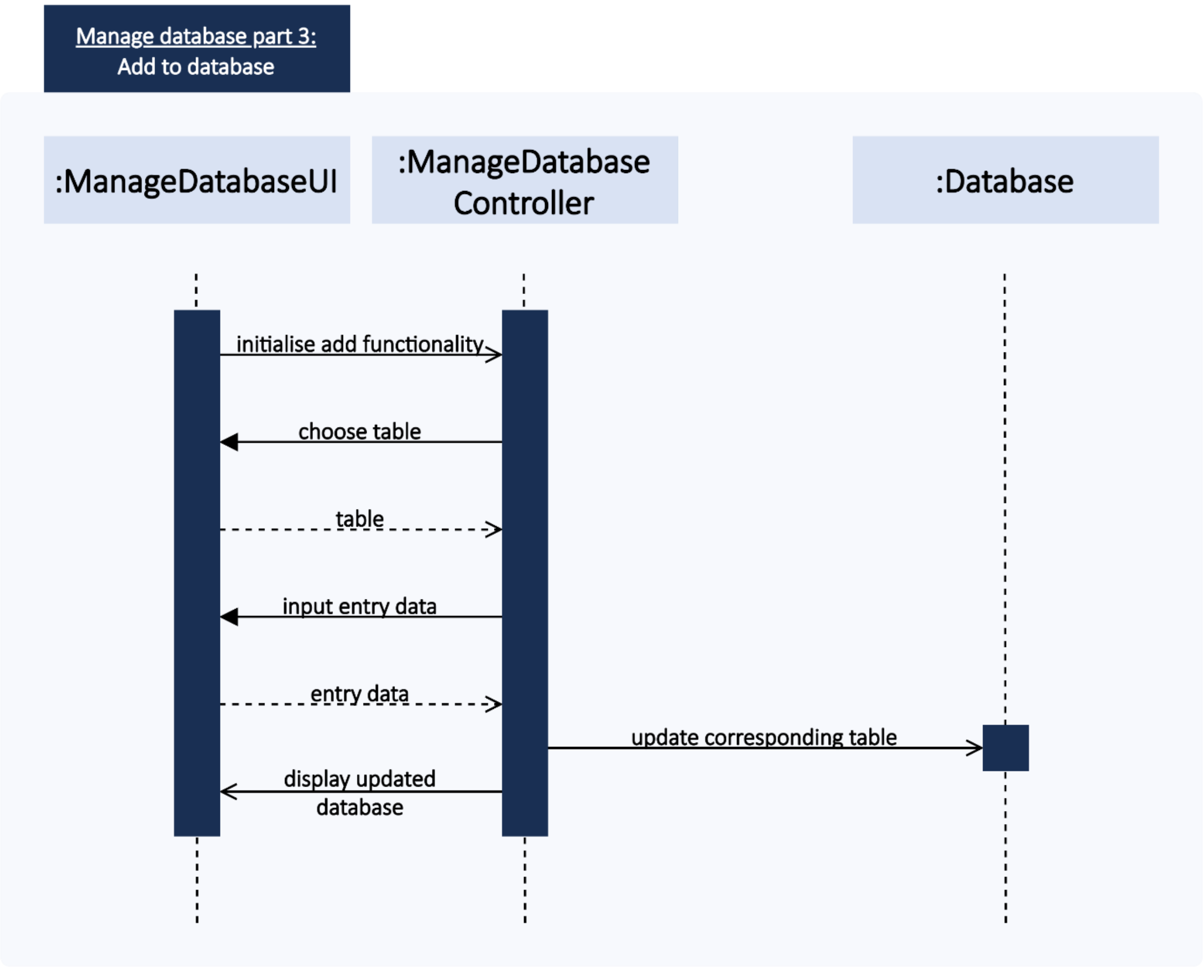
Appendix J: Sequence diagrams



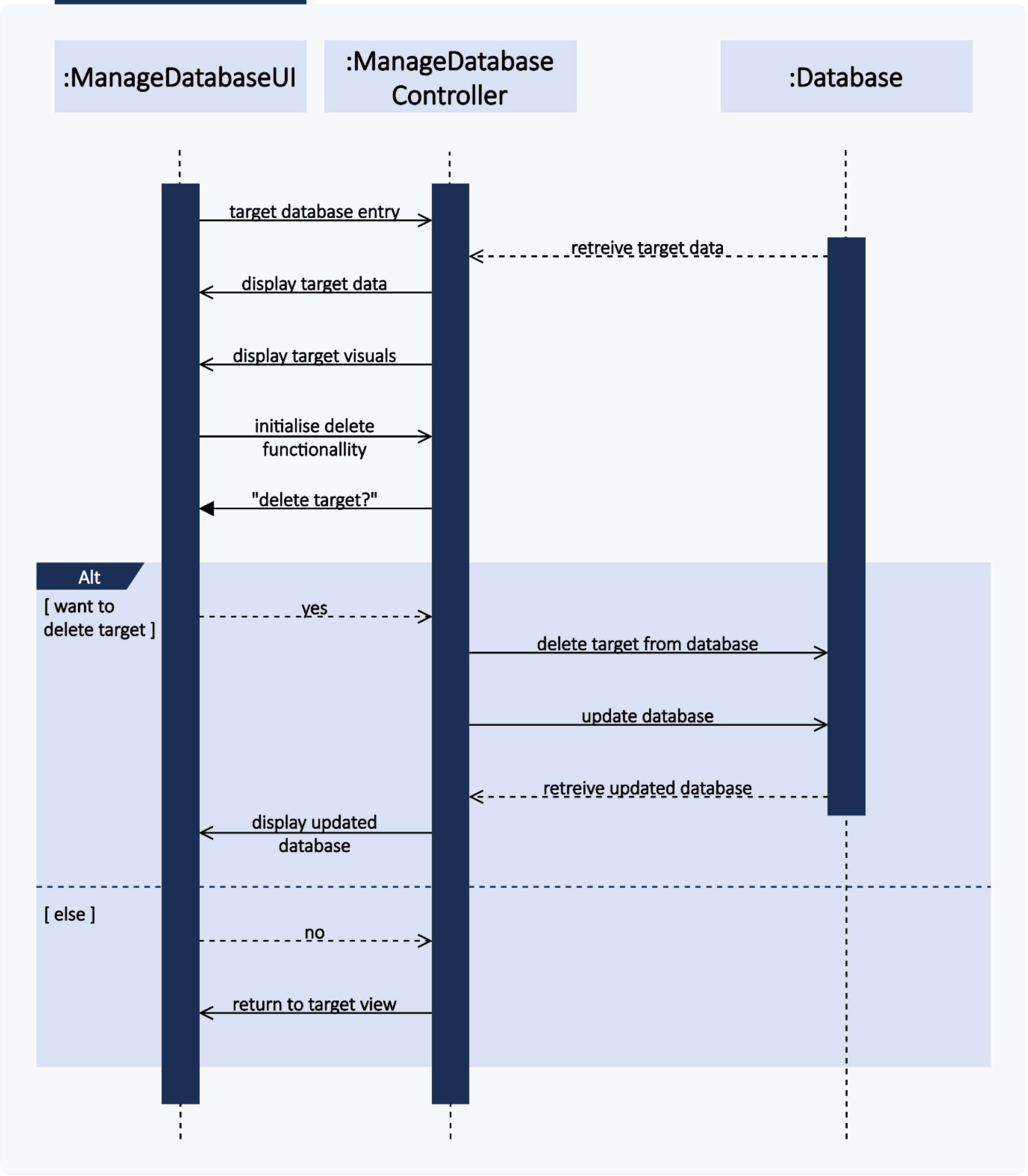
Manage database part 1:
Show Databases



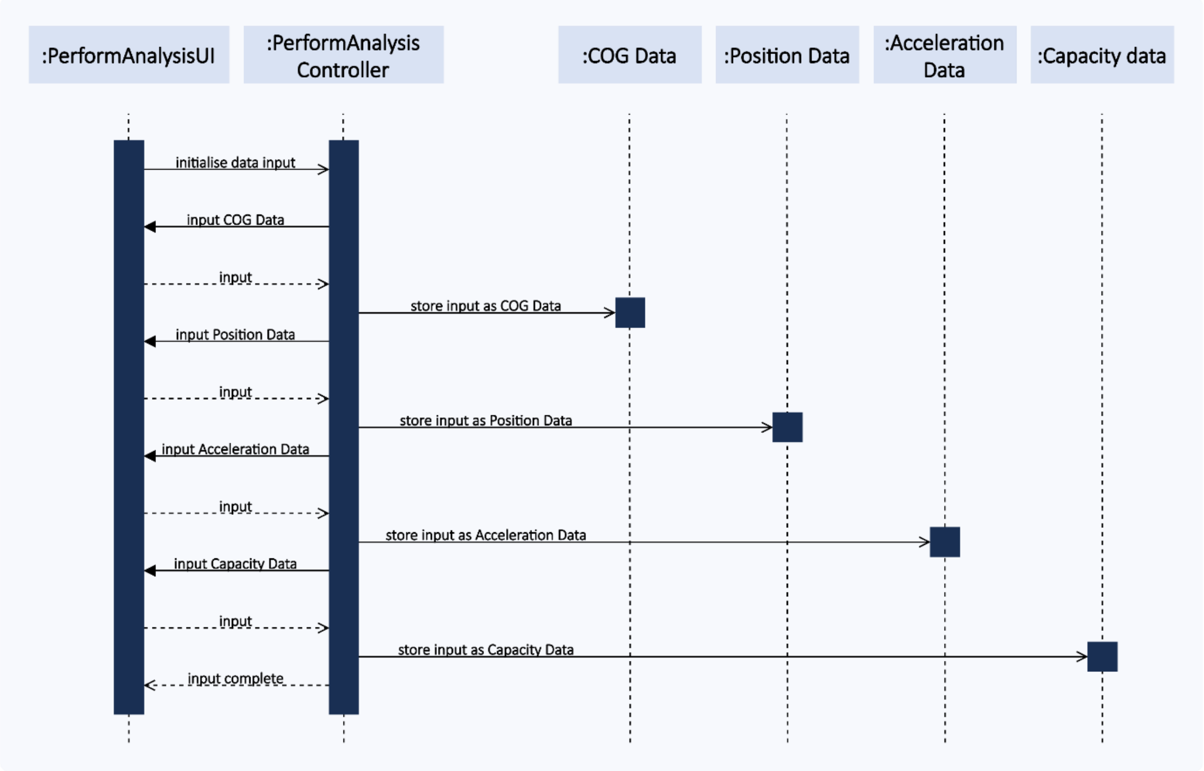

Manage database part 2: Detailed view of a database entry
:ManageDatabaseUI
:ManageDatabase Controller
:Database
target database entry
retreive target data
display target data
display target visuals
Loop
target new database entry
close previous target
retreive new target data
display target data
display target visuals


Manage database part 3:
Add to database
:ManageDatabaseUI
:ManageDatabase Controller
:Database
initialise add functionality
choose table
table
input entry data
entry data
update corresponding table
display updated database

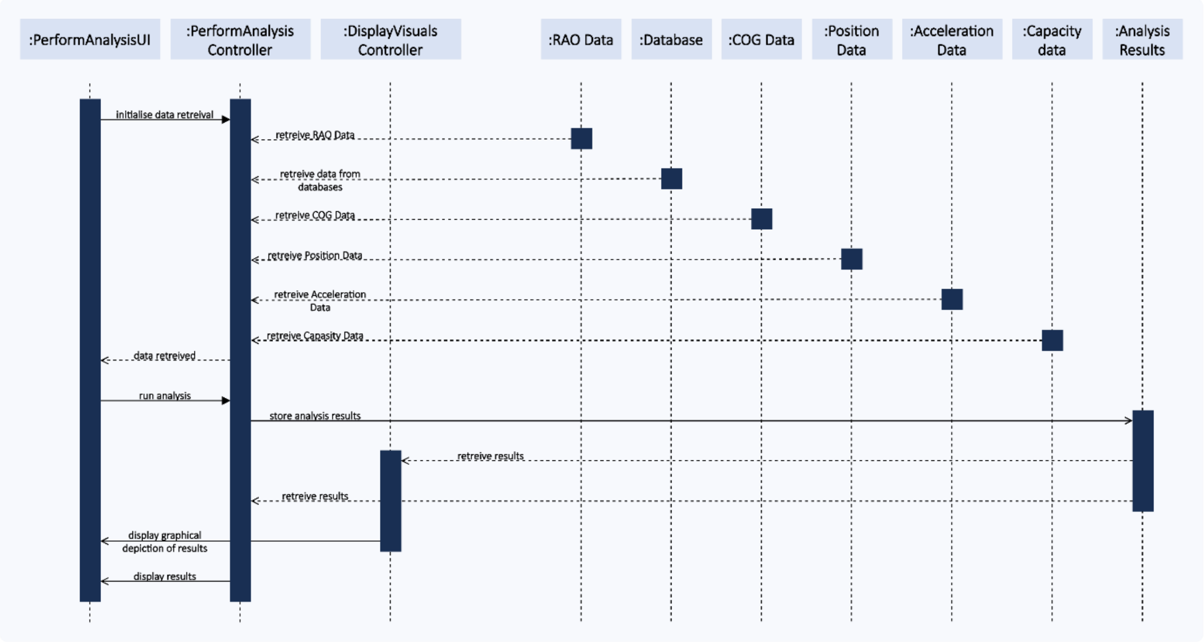
Manage database part 4: Delete from database



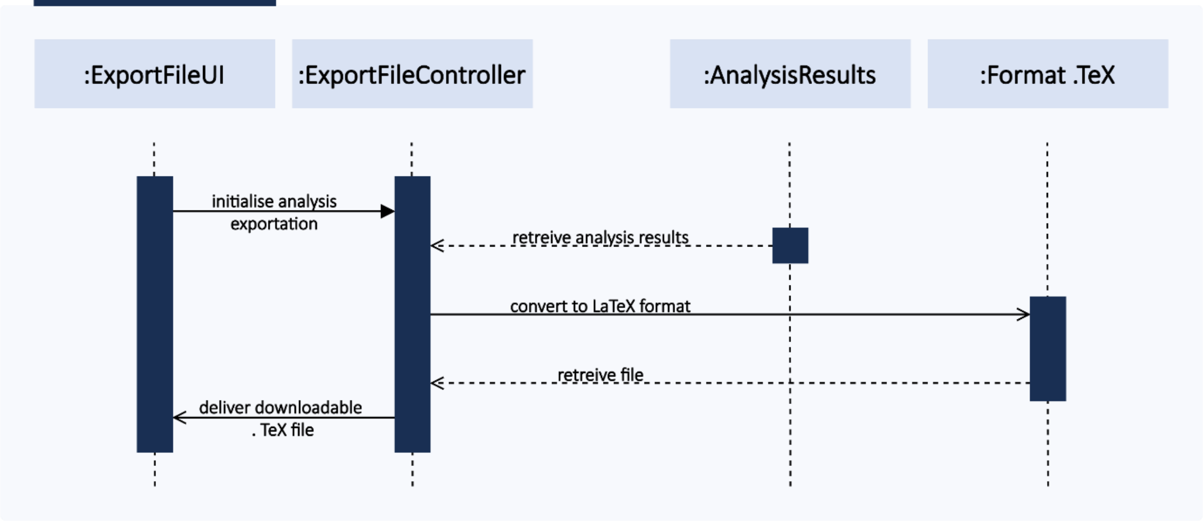
Perform analysis part 1:
Input data



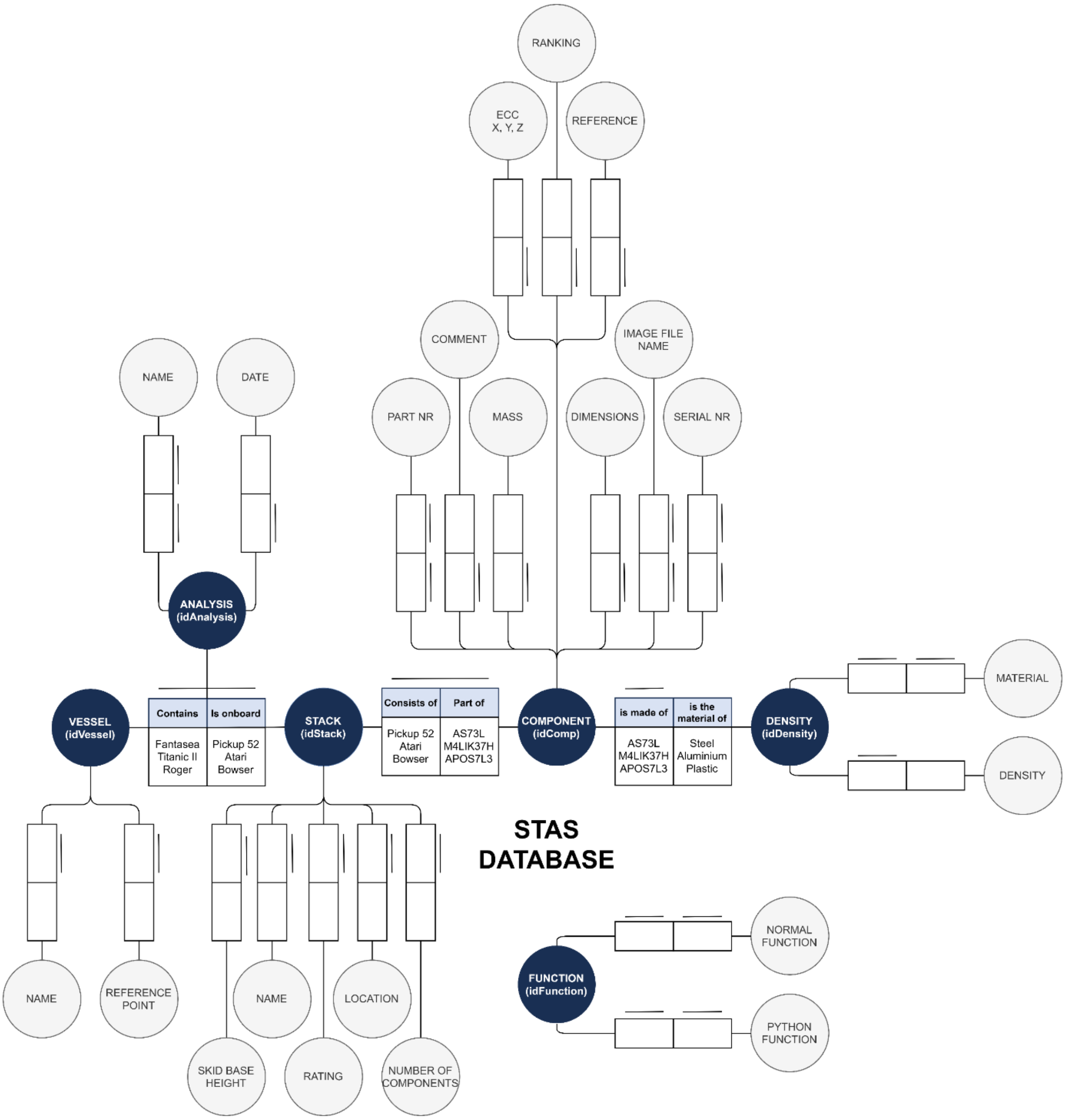
Perform analysis part 2:
Run analysis



Export file


:ExportFileUI
:ExportFileController
:AnalysisResults
:Format .TeX
initialise analysis exportation
retreive analysis results
convert to LaTeX format
retreive file
deliver downloadable . TeX file

Appendix K: ORM 1. Iteration

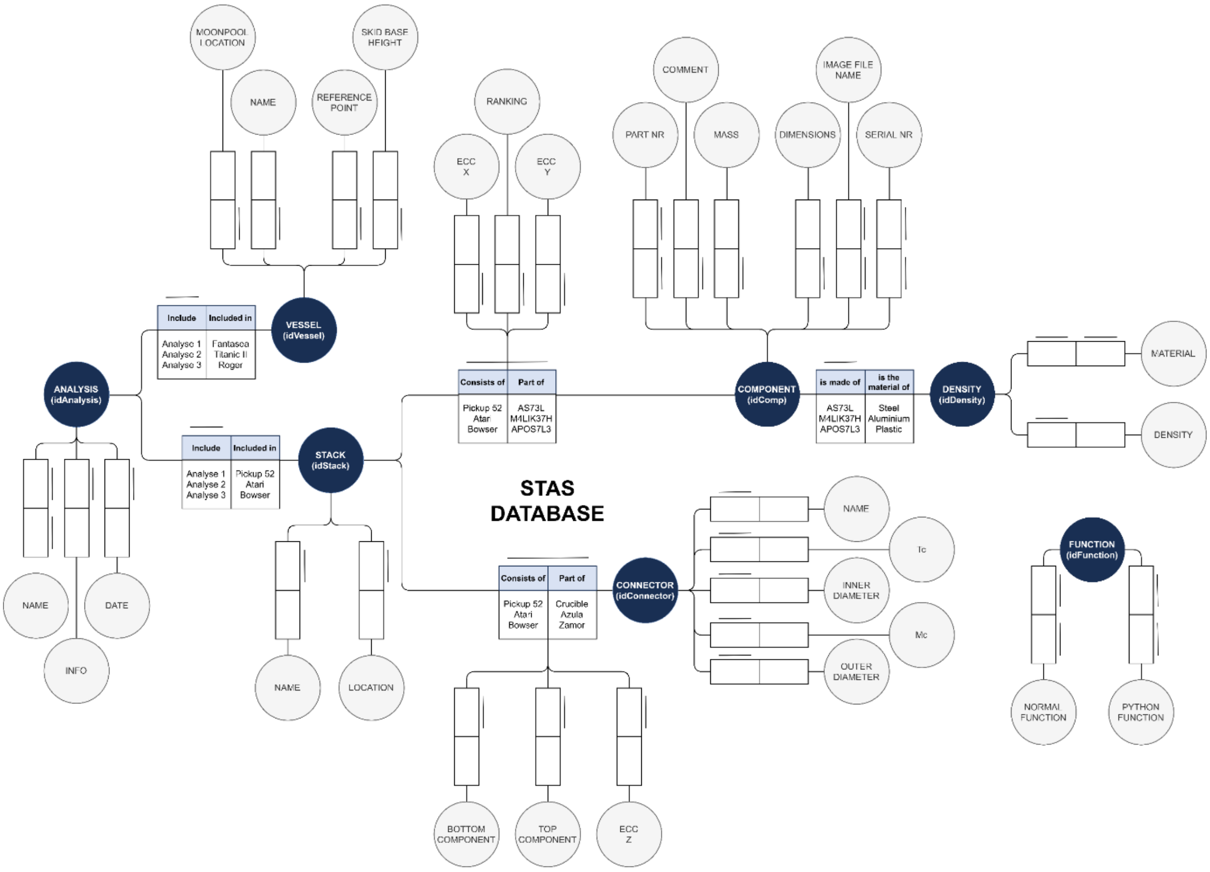


Contains	Is onboard
Fantasea Titanic II Roger	Pickup 52 Atari Bowser

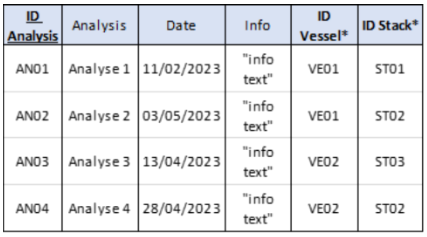
Consists of	Part of
Pickup 52 Atari Bowser	AS73L M4LIK37H APOS7L3

is made of	is the material of
AS73L M4LIK37H APOS7L3	Steel Aluminium Plastic

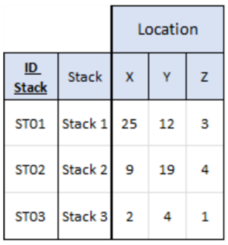
Appendix L: ORM 2. Iteration



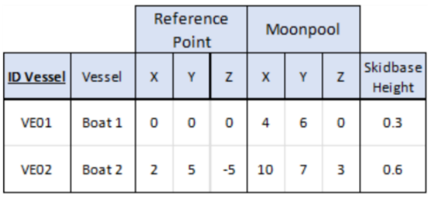
Appendix M: Database Normalization



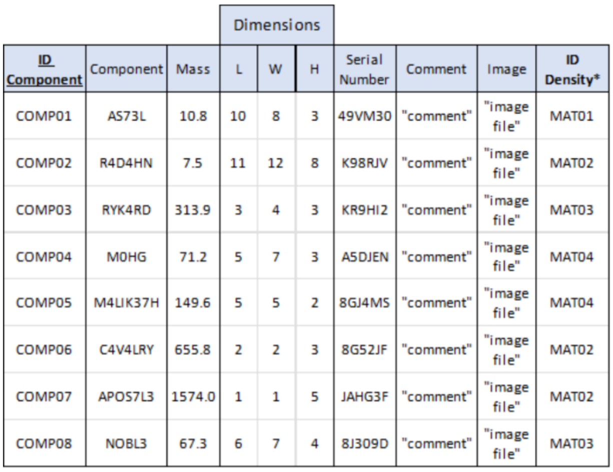
ID Analysis	Analysis	Date	Info	ID Vessel*	ID Stack*
AN01	Analyse 1	11/02/2023	"info text"	VE01	ST01
AN02	Analyse 2	03/05/2023	"info text"	VE01	ST02
AN03	Analyse 3	13/04/2023	"info text"	VE02	ST03
AN04	Analyse 4	28/04/2023	"info text"	VE02	ST02



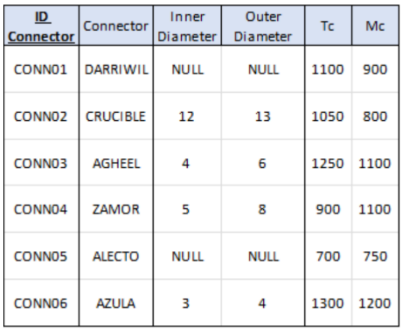
		Location		
ID Stack	Stack	X	Y	Z
ST01	Stack 1	25	12	3
ST02	Stack 2	9	19	4
ST03	Stack 3	2	4	1



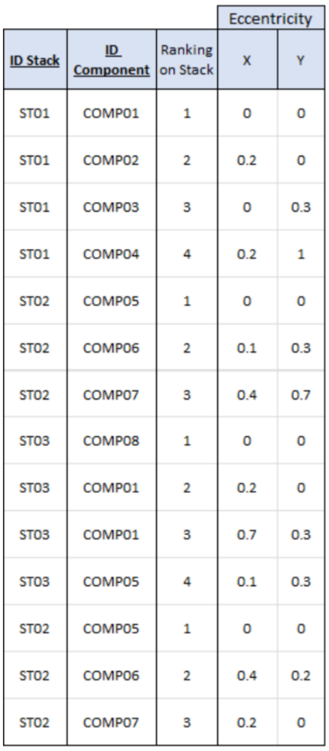
		Reference Point			Moonpool			
ID Vessel	Vessel	X	Y	Z	X	Y	Z	Skidbase Height
VE01	Boat 1	0	0	0	4	6	0	0.3
VE02	Boat 2	2	5	-5	10	7	3	0.6



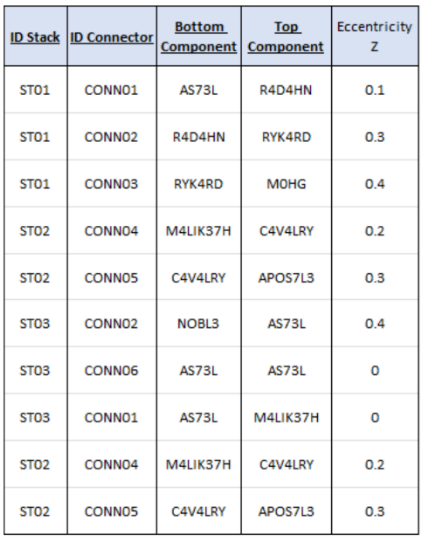
			Dimensions						
ID Component	Component	Mass	L	W	H	Serial Number	Comment	Image	ID Density*
COMP01	AS73L	10.8	10	8	3	49VM30	"comment"	"image file"	MAT01
COMP02	R4D4HN	7.5	11	12	8	K98RJV	"comment"	"image file"	MAT02
COMP03	RYK4RD	313.9	3	4	3	KR9HI2	"comment"	"image file"	MAT03
COMP04	M0HG	71.2	5	7	3	A5DJEN	"comment"	"image file"	MAT04
COMP05	M4LIK37H	149.6	5	5	2	8GJ4MS	"comment"	"image file"	MAT04
COMP06	C4V4LRY	655.8	2	2	3	8G52JF	"comment"	"image file"	MAT02
COMP07	APOS7L3	1574.0	1	1	5	JAHG3F	"comment"	"image file"	MAT02
COMP08	NOBL3	67.3	6	7	4	8J309D	"comment"	"image file"	MAT03



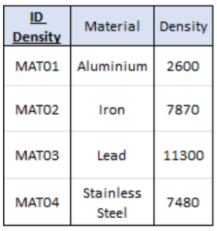
ID Connector	Connector	Inner Diameter	Outer Diameter	Tc	Mc
CONN01	DARRIWIL	NULL	NULL	1100	900
CONN02	CRUCIBLE	12	13	1050	800
CONN03	AGHEEL	4	6	1250	1100
CONN04	ZAMOR	5	8	900	1100
CONN05	ALECTO	NULL	NULL	700	750
CONN06	AZULA	3	4	1300	1200



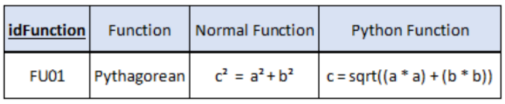
			Eccentricity	
ID Stack	ID Component	Ranking on Stack	X	Y
ST01	COMP01	1	0	0
ST01	COMP02	2	0.2	0
ST01	COMP03	3	0	0.3
ST01	COMP04	4	0.2	1
ST02	COMP05	1	0	0
ST02	COMP06	2	0.1	0.3
ST02	COMP07	3	0.4	0.7
ST03	COMP08	1	0	0
ST03	COMP01	2	0.2	0
ST03	COMP01	3	0.7	0.3
ST03	COMP05	4	0.1	0.3
ST02	COMP05	1	0	0
ST02	COMP06	2	0.4	0.2
ST02	COMP07	3	0.2	0



ID Stack	ID Connector	Bottom Component	Top Component	Eccentricity Z
ST01	CONN01	AS73L	R4D4HN	0.1
ST01	CONN02	R4D4HN	RYK4RD	0.3
ST01	CONN03	RYK4RD	M0HG	0.4
ST02	CONN04	M4LIK37H	C4V4LRY	0.2
ST02	CONN05	C4V4LRY	APOS7L3	0.3
ST03	CONN02	NOBL3	AS73L	0.4
ST03	CONN06	AS73L	AS73L	0
ST03	CONN01	AS73L	M4LIK37H	0
ST02	CONN04	M4LIK37H	C4V4LRY	0.2
ST02	CONN05	C4V4LRY	APOS7L3	0.3



ID Density	Material	Density
MAT01	Aluminium	2600
MAT02	Iron	7870
MAT03	Lead	11300
MAT04	Stainless Steel	7480



idFunction	Function	Normal Function	Python Function
FU01	Pythagorean	$c^2 = a^2 + b^2$	c = sqrt((a * a) + (b * b))