



Afstudeerscriptie

IMPLEMENTEREN VAN DE 'DATA HIGHWAY' BIJ SMIT LAMNALCO
REMMERSWAAL, M.G.D. (13134140)

Samenvatting

De vloot van Smit Lamnalco telt 180 schepen, 80% van de schepen hebben twee of meerdere Caterpillar motoren. Smit Lamnalco wil de data binnen de motoren gebruiken voor analyses en statistieken. Om dit te realiseren is een applicatie ontworpen en gemaakt om de data te verplaatsen, te transformeren en op te slaan.



Het verzamelen, transformeren en opslaan van motorprestatie data

bij Smit Lamnalco

Door

Michaël Gerardus Diederick Remmerswaal

*Voor het verkrijgen van mijn Bachelor of Science
in Software Engineering.
In het tijdsbestek van 04-02-2019 tot 31-05-2019*

Begeleiders van de Haagse Hogeschool

Ton Biegstraaten, Begeleidend examinerator

Hani Al-Ers, Expert examinerator

Supervisor bij Smit Lamnalco

Sander Woltheus, Marine Technology Manager

I. Samenvatting

Het bedrijf Smit Lamnalco levert maatwerk van onder andere sleepwerkzaamheden, onderhoud en reparaties van olie en gas terminals en FSO en FPSO escorting services. Hiervan is Smit Lamnalco de voornaamste leverancier en doet dit al vele jaren.

De afgelopen twee jaar heeft de afdeling Group Fleet Technology gewerkt aan een gecentraliseerd systeem om data te visualiseren, het Fleet Control Center. Dit Fleet Control Center beslaat onder andere een crew data overzicht, live tracking van schepen en een onderhoudstaken overzicht.

De 180 schepen tellende vloot van Smit Lamnalco beslaat voor een groot deel sleepboten met motoren afkomstig van de fabrikant Caterpillar. Deze motoren genereren, aan de hand van sensoren over de gehele motor, data. Deze data wil Smit Lamnalco gebruiken om inzicht te krijgen in de prestaties en status van de motoren.

Na uitlezen van een CAT motor, aan de hand van onboard diagnostics, wordt een rapport gegenereerd. Dit rapport wordt geëxporteerd in XML formaat. De data moet uit dit rapport worden gehaald en in het datawarehouse, op een SQL Server database, worden opgeslagen.

Om dit te bereiken is gekeken naar de structuur van de XML en is geconstateerd dat deze eerst moet worden getransformeerd. Om dit te realiseren is een XML datastructuur tussen rapport en database ontworpen. Om tot deze tussenstructuur te komen, is een WinForms desktop applicatie geschreven in C#. Deze applicatie transformeert de data naar de ontworpen datastructuur. Daarna wordt het getransformeerde XML document per mail naar een server op het hoofdkantoor gestuurd.

Op het hoofdkantoor draait een Microsoft MessageQueue. Deze queue haalt alle bijlagen uit de mails en verwerkt deze naar een translatie proces binnen Tibco Scribe Insight. Na translatie wordt de data in de database opgeslagen en is klaar om te worden geanalyseerd.

Voor nu is de data alleen beschikbaar voor twee motormodellen van Caterpillar. Voor verdere stappen met dit project is het van belang dat meerdere modellen en fabrikanten worden toegevoegd. Uiteindelijk is het de bedoeling dat alle data uit alle motoren van alle fabrikanten van Smit Lamnalco kan worden opgeslagen.

II. Voorwoord

Het document voor u is de scriptie van de afstudeeropdracht van Mick Remmerswaal, student Software Engineering van de Haagse Hogeschool. Deze afstudeeropdracht is uitgevoerd bij Smit Lamnalco in het tijdsbestek van februari 2019 tot en met mei 2019.

Onder leiding van Sander Woltheus, Emad Zaïd, Matthijs Priester en Hubert Plattel van Smit Lamnalco is de afstudeeropdracht uitgevoerd. Ik wil deze mensen graag bedanken voor de mogelijkheden en alle hulp die zij mij hebben geboden. Graag wil ik ook alle medewerkers van de afdeling bedanken voor het opnemen van mij binnen deze groep.

Tevens wil ik ook mijn dankwoord uitspreken naar Albert Kasbergen van PonPower. Hij heeft het voor mij mogelijk gemaakt om een live demo van een 3516B Caterpillar Marine Engine bij te wonen. Ik heb veel nieuws geleerd over mogelijkheden om data uit motoren te halen tijdens dit bezoek.

Daarnaast wil ik de heren, Gelu Mirzoc, Stefan Chakarov en Anatoliy Dimitrov van de SL Malkoha bedanken. Zij hebben mij geholpen bij het testen van de applicatie tijdens mijn bezoek aan hun schip in Malaga, Spanje.

Ik wil ook mijn mede-afstudeerder Berend-Jan Slobbe bedanken voor het feit dat hij mij in contact heeft gebracht met Sander Woltheus. Samen hebben wij het afstudeertraject met veel plezier afgerond bij Smit Lamnalco.

Als laatste wil ik mijn familie bedanken voor de hulp en steun die zij hebben geboden tijdens het afstudeertraject.

Mick Remmerswaal

Mei 2019, Delft

III. Verklarende woordenlijst

SL	Smit Lamnalco	HTTPS	HyperText Transfer Protocol Secure
FCC	Fleet Control Center	SSL	Secure Sockets Layer
CAT	Caterpillar	UC	Use Case
ET	Electronic Technician	XPath	XML Path Language
OBD	On Board Diagnostics	XSLT	eXtensible Stylesheet Language Transformations
HTML	Hyper Text Markup Language	XQuery	XML Query
XML	eXtensible Markup Language	XBRL	eXtensible Business Reporting Language
PDF	Portable Document Format	LINQ	Language Integrated Query
POA	Plan van Aanpak	MS	Microsoft
XP	eXtreme Programming	MSDN	Microsoft Developer Network
FDD	Feature Driven Development	XSD	XML Schema Definition Language
SQL	Structured Query Language	W3	World Wide Web
SMTP	Simple Mail Transfer Protocol	XSI	XML Schema Instance
P2P	Peer-to-peer	MSMQ	Microsoft Message Queue
FTP	File Transfer Protocol	IP	Integration Proces
HTTP	HyperText Transfer Protocol	DTS	Data Translation Specification
FTPS	File Transfer Protocol Secure		

Inhoud

1. Inleiding	7
2. Introductie	8
2.1 Smit Lamnalco, wat en wie	8
2.2 De huidige situatie	13
2.3 De opdracht	16
3. Plan van aanpak	19
3.1 Planning	20
4. Iteraties, wat is er gedaan	21
4.1 Requirements	21
4.2 Software ontwikkelingsmethodiek	22
4.3 Programmeertaal, C# of Python	26
4.4 Versiebeheer & DevOps	27
4.5 Het verkrijgen van rapporten	28
4.6 Ontwerptool	28
4.7 Desktop of Web	29
4.8 Eerste versie Console App	30
4.9 Onderzoek: Methode van data transfer	33
4.10 Reactie na set back	35
4.11 XML en Datastructuren	37
4.12 Tweede versie Console App	41
4.13 Datastructuur en SQL Server	46
4.14 Derde versie Console App en nieuwe Datastructuur	48
4.15 MS Message Queue en Scribe Insight	54
4.16 Vierde en final versie: WinForm Applicatie	58
4.17 Veldtesten in Malaga	62
5. Conclusie: Wat is uiteindelijk tot stand gekomen	64
6. Evaluatie: Hoe ging het, wat heb ik geleerd	64
7. Reflectie: Wat kan beter, wat had ik anders kunnen doen	67
8. Literatuurlijst	68
9. Figuren lijst	70
10. Bijlagen	72
Bijlage 1. Plan van aanpak	72

Bijlage 2. Business Model Canvas	75
Bijlage 3. Requirements Interview	76
Bijlage 4. Requirements	79
Bijlage 5. Data Mapping	81
Bijlage 6. Pre Research: Collecting Data	84
Bijlage 7. Research Collecting Data	85
Bijlage 8. Use cases	86
Bijlage 9. Diagrammen	92
Bijlage 10. Voorbeeld Document van de Hashim 3	122
Bijlage 11. Data Definition Document	132
Bijlage 12. Ontworpen XML schema's	134
Bijlage 13. SQL Queries	144
Bijlage 14. Tour door de Malkoha	166
Bijlage 15. Verbinding Engines-CAT ET	168
Bijlage 16. Test Cases	176

1. Inleiding

Tegenwoordig is data niet meer weg te denken uit onze huidige maatschappij. Elke persoon genereert het, en elk bedrijf gebruikt het. Van de supermarkt om de hoek tot grote tech bedrijven als Facebook en Alphabet. Er zijn speculaties dat de hoeveelheid data in 2020 meer dan $40 \cdot 10^{12}$ GB omvat en 90% van deze hoeveelheid is in de afgelopen twee jaar gegenereerd.

Zo'n 97% van de bedrijven over de wereld heeft wel eens een investering gedaan in Big Data of Artificial Intelligence. Veelal worden deze investeringen gedaan om kosten te drukken, bijvoorbeeld Netflix bespaard al een miljard op klantbehoud door gebruik te maken van Big Data. Niet alleen kosten kunnen worden gedrukt met Big Data of Artificial Intelligence. Ook het efficiënter laten verlopen van bedrijfsprocessen, is een groot voordeel van het gebruik van data.

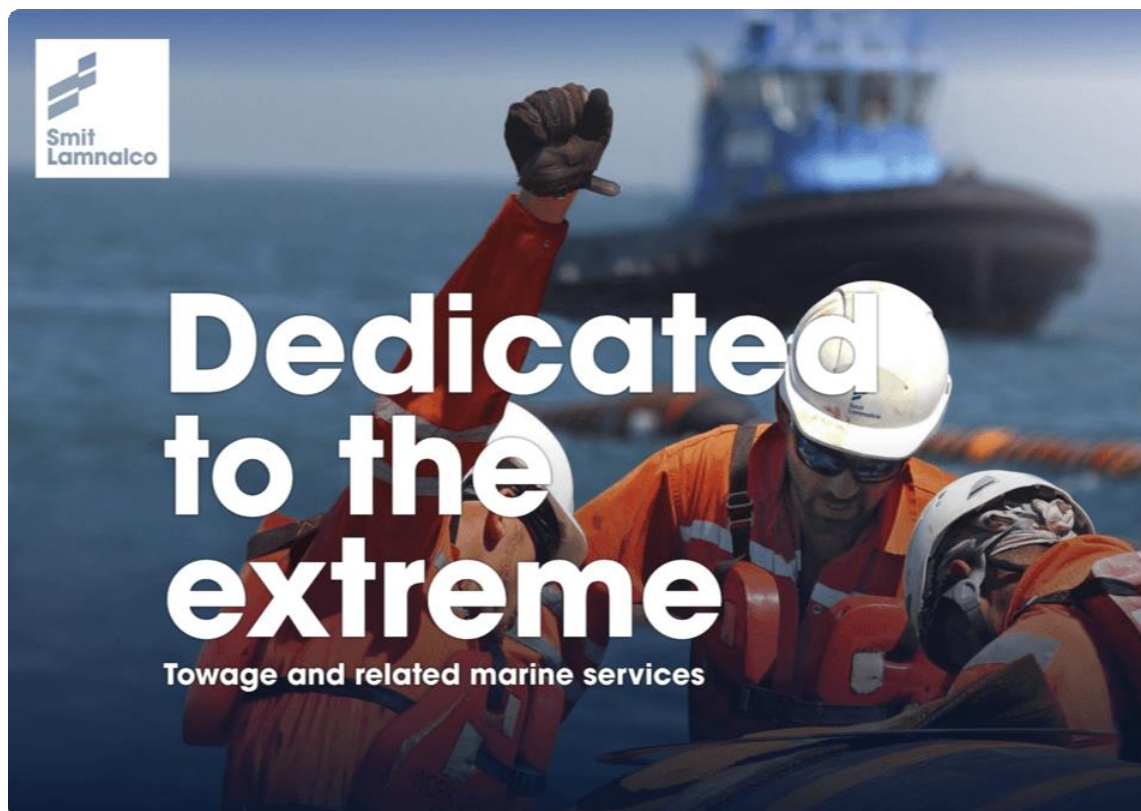
Alleen hoe begint men? Genoeg bedrijven proberen terabytes aan data te verzamelen, zonder daar eerst de goede fundering voor neer te leggen. Onder het gevoel van Fear Of Missing Out en de lovende woorden van zoveel professionals, gaan bedrijven geblinddoekt de datawereld in.

Daarom is het van belang dat men een koel hoofd houdt, als het gaat om een hot topic als Big Data. Smit Lamnalco heeft in de afgelopen twee jaar met het Fleet Control Center de fundering gelegd, voor wat in de toekomst het brein van de operationele kanten van het bedrijf kan worden.

Mijn opdracht; Het uitbreiden van dit Fleet Control Center met motorprestatie data, afkomstig van de wereldwijde vloot van Smit Lamnalco.

2. Introductie

2.1 Smit Lamnalco, wat en wie



Figuur 1 Smit Lamnalco slogan

Smit Lamnalco, *Dedicated to the extreme*, voorziet first class en betrouwbaar maatwerk op elk gebied van maritieme ondersteuning. Smit Lamnalco begon in 1963 als Lamnalco voortvloeiend uit een partnerschap tussen Royal Boskalis en de Rezayat Group.

	Number	ONSHORE				OFFSHORE		
		Oil and LPG	LNG and FSRU	Mining	Port	F(P)SO and SPM	FLNG	SPM
Tug 	102	●	●	●	●			●
OSV 	26					●	●	●
Misc. 	51	●	●	●	●	●	●	●

Figuur 2 Vloot overzicht van Smit Lamnalco

De vloot van Smit Lamnalco beslaat 180 schepen en bestaat uit Harbor & Terminal Tugs, Offshore Support Vessels, Crew & Pilot Boats en Line Handlers.

Harbor & Terminal Tugs



Figuur 3 Smit Lamnalco sleepboot

Voor elke klus worden speciaal ontworpen sleepboten aangeleverd, die precies passen bij wat de klant wil.

Offshore Support Vessels



Figuur 4 Smit Lamnalco Offshore Support Vessel

Speciaal ontworpen met een extra groot deck, voor allerlei ondersteunde taken zoals licht onderhoudswerk en duik operaties

Crew & Pilot Boats



Figuur 5 Smit Lamnalco Crew&Pilot Vessel

Voor het veilig en snel verplaatsen van de crew.

Line Handlers



Figuur 6 Smit Lamnalco Line Handlers

Deze schepen zijn speciaal ontworpen voor het afhandelen van de zware meertrossen en drijvende pijpleidingen.

Al deze schepen verzorgen een zeer groot aantal aan verschillende taken binnen de maritieme industrie. Zo helpt Smit Lamnalco bij het aan- en afmeren van olie en gas tankers en doen ze aan inspecties, reparaties en onderhoud van deze tankers. Hiernaast biedt Smit Lamnalco ondersteuning bij mijnoperaties en duikwerkzaamheden. Ook het onderhouden en repareren van olie & gas terminals wordt gedaan door Smit Lamnalco.

Hiernaast biedt Smit Lamnalco directe noodhulp bij milieu rampen op het water. Naast sleepwerkzaamheden bieden ze maritiem gerelateerde consultancy en het beheer van systemen op zowel schepen als aan wal aan.

Details

	2800 employees
	42 nationalities
	180 vessels
	30 countries

Figuur 7 Detail overzicht van Smit Lamnalco

Onshore duties

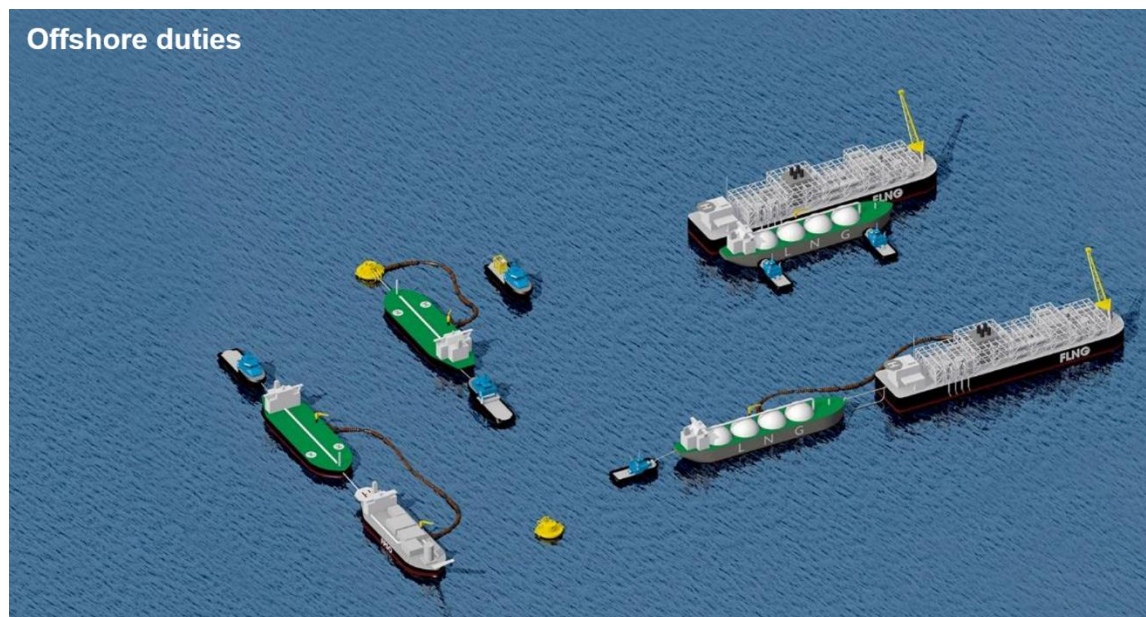


Figuur 8 Onshore duties van Smit Lamnalco, als Af- en Aanmeren, Safety Management en Mining Logistics

De doelgroep van Smit Lamnalco bestaat uit grote leveranciers van olie en gas, havens en terminal bezitters. Het communiceren met deze klanten wordt door de commerciële teams binnen het bedrijf gedaan, waarbij het onderdeel Tendering een belangrijk rol speelt. De Tendering afdeling is verantwoordelijk voor alle contracten en het klantbehoud binnen Smit Lamnalco.

De communicatie met de klanten gebeurt bij elke klant op een andere manier. Met de 42 klanten die Smit Lamnalco heeft, wordt op 42 verschillende manieren gecommuniceerd. Elk contract is tailor-made.

Klanten blijven bij Smit Lamnalco, omdat het een betrouwbare business partner is. Klanten van Smit Lamnalco krijgen bij bedrijfsbezoek dan ook alles te zien. Er is grote transparantie, er worden geen mooie verhaaltjes verkocht en afspraak is afspraak.



Figuur 9 Offshore duties van Smit Lamnalco, als Escorting, Single Point Mooring en Cryogenic Hose Handling

De belangrijkste bedrijfsmiddelen binnen Smit Lamnalco zijn de crew op de schepen en de staff aan wal. Daarnaast vallen in deze categorie ook alle systemen die nodig zijn om het bedrijf op de rails te houden, zoals het Planned Maintenance systeem Marad5 of het Data Management in PowerBI binnen het Fleet Control Center. Uiteraard is de 180 schepen tellende vloot van Smit Lamnalco ook van belang.

Partners van Smit Lamnalco zijn onder andere winch leveranciers als DMT en Rolls Royce, motor fabrikanten als Caterpillar en Wärtsillä, smeerolie en diesel leveranciers als Exxon Mobil en Shell, classificatie bureaus als Veritas en Lloyd's Register en thruster leveranciers als ABB en Schottel.

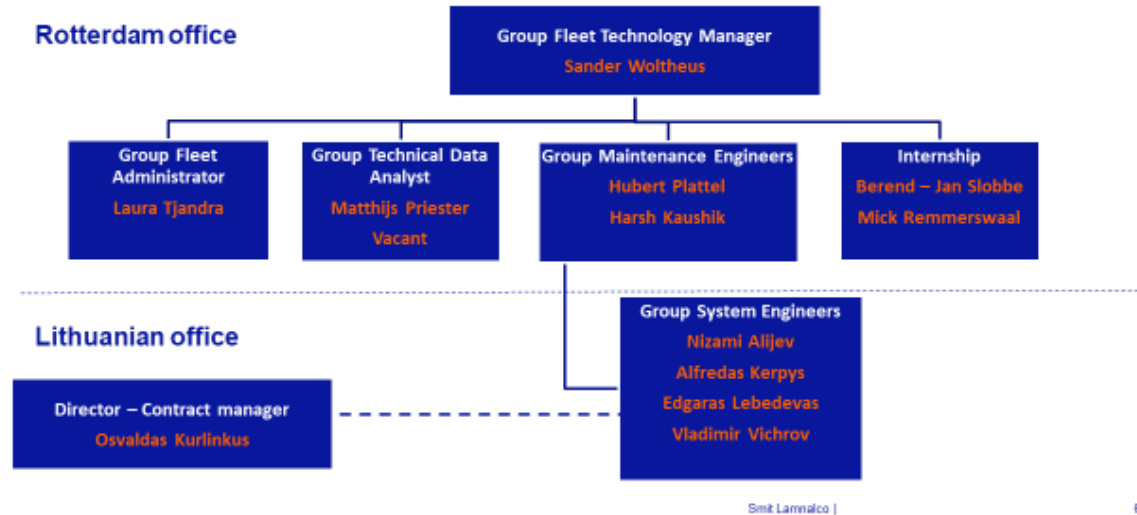
Bij Smit Lamnalco staan veiligheid en betrouwbaarheid centraal. Met veiligheid bouwt men betrouwbaarheid. Zonder veiligheid wordt de kans op incidenten, uitval en schade aan eigendommen groter. Het gevolg is schade aan reputatie en als de reputatie wegvalt, valt de klantenbasis weg.

De organisatie is mechanistisch opgebouwd met een duidelijke structuur. Echter is de cultuur binnen het bedrijf organisch, maar nog wel wat formeel. Alle communicatie gaat namelijk wel volgens afgesproken protocollen en procedures. Zo wordt elke beslissing die gemaakt wordt, uitgerold via een officiële Operational Alert.

Binnen de organisatie bevindt zich de Group Fleet Technology afdeling, hier gebeurt het management ten aanzien van de vloot van Smit Lamnalco. Hierbinnen zal ik werkzaam zijn.



Group Fleet Technology Organization



Figuur 10 Overzicht structuur binnen Fleet Technology afdeling



Figuur 11 Smit Lamnalco tugs aan het werk

2.2 De huidige situatie

2.2.1 Het Fleet Control Center

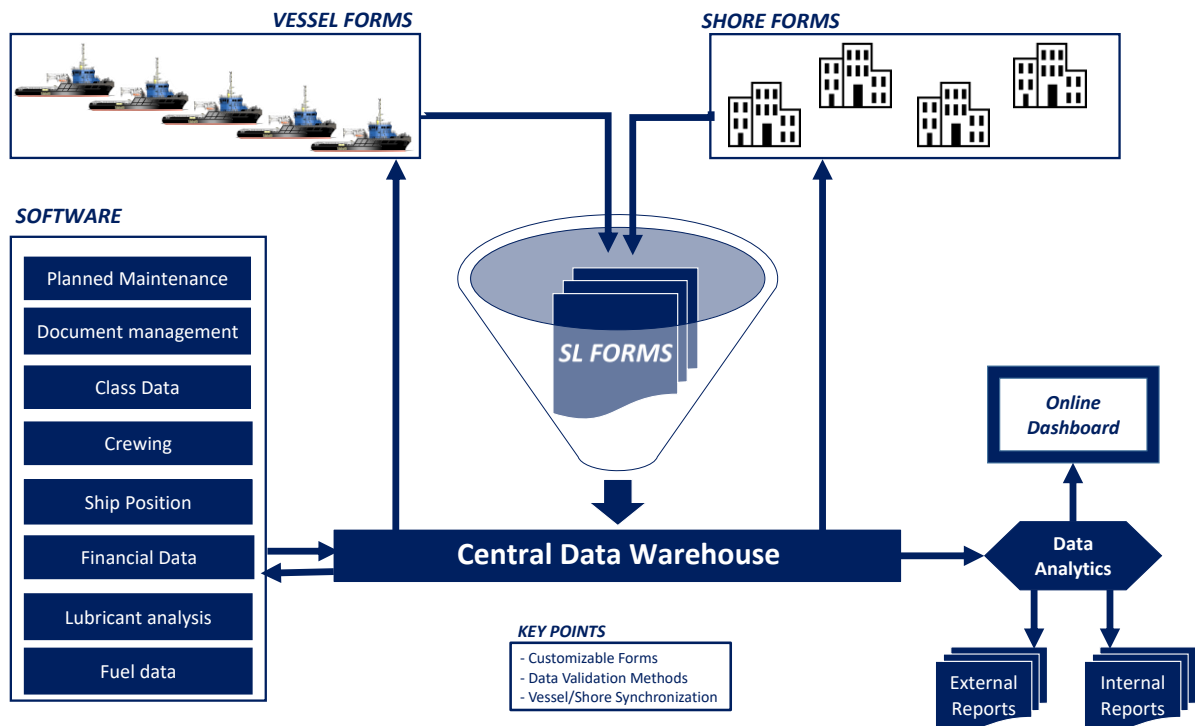
Om inzicht te krijgen in verschillende operationele processen binnen de vloot, is Smit Lamnalco sinds eind 2017 bezig met het visualiseren van deze processen. Om te zorgen dat dit centraal beschikbaar is, is het Fleet Control Center opgericht.



Figuur 12 Het Fleet Control Center

Om dit Control Center te laten functioneren, is een grote hoeveelheid data nodig. Deze data wordt verkregen door het invullen van formulieren. De formulieren worden zowel op de schepen als aan wal ingevuld. Voor het verzorgen van de formulieren is de applicatie *SL Forms* gebouwd. Alles wordt gecentraliseerd in een groot datawarehouse en alle verschillende soorten software, die Smit Lamnalco heeft draaien, haken hier in.

Data Collection, - Analysis and - Reporting



Figuur 13 Overzicht dataflow binnen Smit Lamnalco

De data wordt vervolgens met het programma Power BI gevisualiseerd. Power BI is een product van Microsoft en is een oplossing voor Business Intelligence en bedrijfsanalyses. Met Power BI kunnen gegevens worden gevisualiseerd en kunnen overzichten en rapporten worden gedeeld binnen de organisatie.

Op het Fleet Control Center wordt onder andere via GPS real-time de locatie van elk schip weergegeven. Elke drie maanden worden smeeroilie monster getest en alle data hiervan wordt ook beschikbaar gesteld voor analyse.



Figuur 6 Voorbeeld van het overzicht van de smeeroilie monsters

Naast smeeroilie wordt ook verschillende generale informatie over de motoren die Smit Lamnalco allemaal bezit weergegeven. 80% Daarvan is van de fabrikant Caterpillar.



Figuur 7 Voorbeeld van het overzicht van alle motoren

Ook staat het FCC in verbinding met Marad5, het Planned Maintenance systeem. Dit systeem beslaat alle onderhoudstaken die zijn uitgevoerd, worden uitgevoerd en in de toekomst worden uitgevoerd. Ook alle crew data is beschikbaar in het FCC. De verschillende contracten per schip worden bijgehouden om zo een overzicht te geven van bijvoorbeeld wie waar aanwezig is en hoe lang iemand daar aanwezig was.

2.2.2 De betrokkenen

De grootste stakeholder bij dit project is uiteraard het bedrijf Smit Lamnalco zelf. Het project wordt in hun opdracht uitgevoerd.

Als Project Manager en opdrachtgever is Sander Woltheus, Marine Technology Manager, de belangrijkste stakeholder. Niet alleen is hij verantwoordelijk voor de afdeling, maar is ook de bedenker van het Fleet Control Center en eindverantwoordelijke voor dit project.

De data wordt nu onderhouden door het ICT team met, voor dit project, als belangrijkste stakeholder Emad Zaïd, de Group Application Manager. Hij zorgt dat het Central Data Warehouse optimaal werkt en wordt bijgehouden. Als stakeholder is hij verantwoordelijk voor de datastructuur binnen de database en de koppeling naar andere datasets, zoals Marad5 en de Lube Oil dataset.

Matthijs Priester, de Group Technical Data Analyst, is verantwoordelijk voor de bouw en het onderhoud van de dashboards. Daarmee is hij stakeholder voor het uiteindelijk visualiseren van de Engine Performance data.

Degene verantwoordelijk voor het uiteindelijk doorvoeren van de beslissingen, gemaakt met de Engine Performance data, is Hubert Plattel, Senior Group Maintenance Engineer. Hij is verantwoordelijk voor het bijhouden van alle onderhoudstaken en de uitvoering hiervan.

2.3 De opdracht

De opdracht is het uitbreiden van het Fleet Control Center met motorprestatie data. Dit door middel van het implementeren van een methode om data te verplaatsen, te transformeren, op te slaan en optioneel te visualiseren.

Het visualiseren is optioneel gemaakt, omdat de stakeholders twijfels hebben of er genoeg tijd is om dit te realiseren. Het belangrijkste van deze opdracht omslaat dus de gedeeltes verplaatsen, transformeren en opslaan.

Het opslaan van de data moet gebeuren door middel van het gebruik van de SQL Server database, die on-premise bij Smit Lamnalco staat.

Voor de rapporten zal een duidelijke standaard structuur moeten worden ontworpen, zodat toekomstige rapporten gemakkelijker kunnen worden doorgevoerd naar de database.

2.3.1 Scope

Gekozen is om alleen op motoren van Caterpillar te focussen. Deze keuze is gebaseerd op het feit dat het overgrote deel van de motoren in het bezit van Smit Lamnalco van deze fabrikant is.

De implementatie omvat een applicatie of meerdere, verantwoordelijk voor het verplaatsen, transformeren en opslaan van de motorprestatie data. Visualisatie is optioneel.

Een ontwerp en implementatie zullen worden gemaakt van een applicatie, of meerdere en van een database. Een testdocument is optioneel.

Ook is afgesproken dat de al bestaande SQL Server van Smit Lamnalco gaat worden gebruikt. Dit houdt in dat het te implementeren stuk hierop moet aansluiten.

2.3.2 Caterpillar motoren

De vloot van Smit Lamnalco telt rond de 180 schepen, het overgrote deel van deze schepen is in het bezit van twee of meer motoren van de fabrikant Caterpillar. Deze motoren worden met behulp van het diagnostische programma van Caterpillar zelf, CAT Electronic Technician (ET), uitgelezen.

Dit wordt gedaan door middel van een On Board Diagnostics (OBD) stekker te verbinden met de Electronic Control Module (ECM) van de motor. Alle mogelijke sensoren die een Caterpillar motor bezit, is verbonden met deze ECM.



Figuur 14 Foto van een ECM

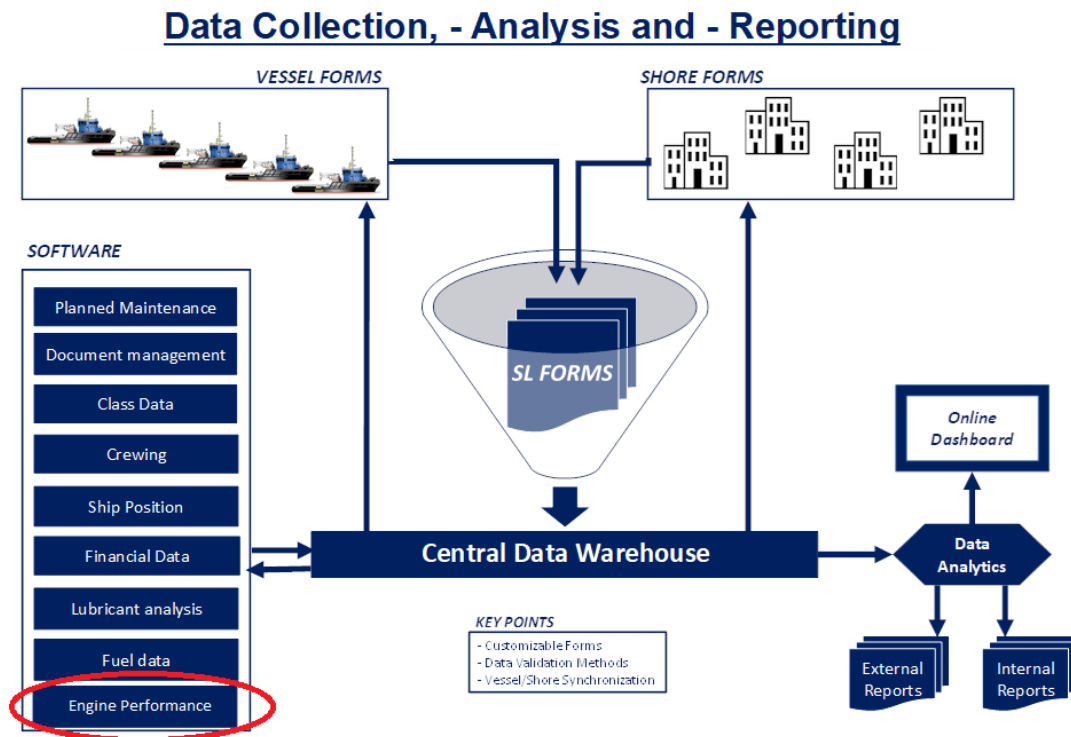


Figuur 15 De connector van Cat

Met behulp van het programma worden de rapporten gegenereerd. Deze rapporten zijn in meerdere formats te verkrijgen, zo zijn ze te verkrijgen in een HTML pagina, een PDF document en een XML bestand.

2.3.3 Toekomstige situatiebeschrijving

Na afronding van de opdracht zal het te implementeren deel gaan uitmaken van de Software die communiceert met het Central Data Warehouse van Smit Lamnalco. Vanuit het te implementeren stuk zal motorprestatie (Engine Performance) data beschikbaar worden gesteld voor analyse. Zie onderstaand overzicht.



Figuur 16 Dataflow na implementatie

Deze analyses kunnen vervolgens helpen bij het maken van management beslissingen. Hiernaast zal door middel van visualisaties in één opslag kunnen worden gezien of zich technische problemen binnen de vloot voordoen.

Ook zal het een verbetering van het onderhoudsproces en de planning tot gevolg hebben. Dit omdat meer inzicht in de prestaties van de motoren kan leiden tot het eerder signaleren van technische problemen. Dit zal zorgen dat efficiënter kan worden opgetreden.

Wanneer het onderhoud wordt verbeterd, zal automatisch een verhoging van kwaliteit plaatsvinden, wat resulteert in minder uitval. Dit kan leiden tot een verhoging van vertrouwen bij de klanten wat de kans op betere contracten met betere betalingsvoorwaarden tot gevolg heeft.

3. Plan van aanpak

Aan het begin van de afstudeeropdracht is een plan van aanpak¹ geschreven met de zojuist benoemde scoping, perspectieven en opdrachtomschrijving. In het plan van aanpak zijn een paar vragen naar voren gekomen. Deze vragen zijn gebaseerd op de opdracht omschrijving en gesprekken met de betrokkenen. De hoofdvraag van de opdracht is:

Wat is de beste methode om motorprestatie data te verzamelen, transporteren, transformeren, op te slaan en te visualiseren en implementeer dit.

Hier zijn een viertal deelvragen uit voortgevloeid.

1. Wat is de beste methode om de data transfer tot stand te brengen?
2. Wat is de beste manier om de data te structureren?
3. Wat zijn de belangrijkste elementen binnen de data?
4. Wat is de beste manier om een applicatie te bouwen?

Na verdere evaluatie zijn de vragen uitgediept.

1. Wat is de beste methode om de data van schip naar wal te brengen?
 - a. Is de beste manier ook de snelste en hoe belangrijk is dat?
 - b. Wat moet er gebeuren als er geen connectie is tussen schip en wal?
 - c. Hoe vaak moet de data transfer plaatsvinden?
2. Wat is de beste manier om de data te structureren?
 - a. Welk van de drie formats, HTML, XML of PDF, sluit het beste aan.
3. Wat zijn de belangrijkste elementen binnen de motorprestatie data?
 - a. Welke data is belangrijk voor de business
 - b. Met welke data kan de business worden verbeterd
4. Wat is de beste manier om een applicatie te bouwen?
 - a. Welke programmeertaal past het beste bij dit project?
 - b. Welk framework sluit het beste aan bij dit project?

¹ Bijlage 1 Plan van Aanpak blz. 72-74

3.1 Planning

Met behulp van de vragen, perspectieven, scoping en opdrachtomschrijving is een grove planning gemaakt. Deze planning is gebaseerd op de 17 weken die zijn gegeven voor het afronden van de afstudeeropdracht vanuit de Haagse Hogeschool. Aan het eind van de periode is rekening gehouden met uitloop van de opdracht.

START	END	TASK
04-02-2019	08-02-2019	POA completed and submitted
11-02-2019	13-02-2019	Discuss example files with Hubert
14-02-2019	14-02-2019	Interview with Hubert and Matthijs on usable data
15-02-2019	18-02-2019	Interviewing stakeholders to figure out requirements
19-02-2019	20-02-2019	Prioritize the requirements
21-02-2019	22-02-2019	Setup research on a way to move data from ship to shore
25-02-2019	01-03-2019	Research a way to move data from ship to shore
04-03-2019	05-03-2019	Setup research on a way for structuring the data
06-03-2019	11-03-2019	Research a way to structure the data
12-03-2019	13-03-2019	Setup research for programming language, framework, databasing, etc.
14-03-2019	15-03-2019	Research programming language, framework, databasing, etc.
18-03-2019	18-03-2019	Make decisions based on the research conducted
19-03-2019	19-03-2019	Discuss decisions with stakeholders
20-03-2019	22-03-2019	Design of application to move data from ship to shore
25-03-2019	29-03-2019	Building application to move data from ship to shore
01-04-2019	03-04-2019	Design of data structure and the application for transforming and storing it in the database
04-04-2019	08-04-2019	Designing database structure
09-04-2019	15-04-2019	Building application for structuring, transforming and storing it in the database
16-04-2019	28-04-2019	Extra time for the building of the applications
16-04-2019	19-04-2019	<i>Optional: designing application for visualization</i>
22-04-2019	26-04-2019	<i>Optional: build application for visualization</i>
29-04-2019	30-04-2019	<i>Optional: Design tests for the applications</i>

4. Iteraties, wat is er gedaan

Na het schrijven van het plan van aanpak en het maken van de planning is besloten om bijna alle documentatie in het Engels te maken, omdat dit de voertaal binnen Smit Lamnalco is.

Om achter de requirements van dit project te komen, is een interview² gehouden met elke stakeholder. Met behulp van een topiclijst zijn de interviewvragen gemaakt.

De topics zijn:

- Het data format
- Belangrijke/kritieke data
- Snelheid en nauwkeurigheid
- Veiligheid van de data
- Connectiviteit
- Leveringsinterval
- De kijk op het project
- Het perfecte eindresultaat

Aan de hand hiervan zijn de volgende vragen opgesteld

1. Wat is, in je eigen woorden, jouw kijk op het project
2. Wat is, in jouw mening, het perfecte eindresultaat?
3. Wat zou de belangrijkste data zijn uit een rapport voor jou?
4. Welk format zou voor jou het interessantste zijn?
5. Wat zou het ideale leveringsinterval zijn?
6. Wat zou er moeten gebeuren wanneer een schip geen connectiviteit heeft?
7. Wat is jouw mening op het gebied van beveiliging van de prestatiedata?

4.1 Requirements

Met de antwoorden is gewerkt aan een requirements document³. Het requirements document is een belangrijk stuk documentatie binnen de software development. De requirements zijn in de vorm van User Stories opgebouwd. Een User Story volgt de vorm van: als 'betrokkene', wil ik 'iets', 'omdat'. Voor het prioriteren is eerst een onderscheidt gemaakt tussen functionele en non-functionele requirements. Vervolgens is met behulp van MoSCoW geprioriteerd. MoSCoW is een vorm van prioriteren op basis van belang, zo staat elk deel van het woord voor een niveau van belang.

- M staat voor Must Have
- S staat voor Should Have
- C staat voor Could Have
- W staat voor Won't Have

De Must Haves zijn de belangrijkste eisen waaraan het project moet voldoen. De Should Haves zijn de voorwaarden die zeer gewenst zijn. De Could Haves zijn eisen die in behandeling kunnen worden genomen als daar genoeg tijd voor is. De Won't Haves zijn te vergelijken met scoping en vertellen wat in een project niet wordt meegenomen.

² Bijlage 3 Requirements Interview blz. 76-79

³ Bijlage 4 Requirements Document blz. 79-80

Het idee van MoSCoW is dat optimaal gebruik kan worden gemaakt van het gelimiteerde tijdsbestek binnen een project. Men bespreekt met de betrokkenen de Must Haves waaraan het project moet voldoen. Het idee is dan ook dat het project officieel faalt als aan het eind van het tijdsbestek niet aan deze voorwaarden is voldaan.

Kritiek op deze methode van prioritering is onder andere het gebrek aan de juiste redenering als het gaat om prioriteren in de categorie Must Haves en Should Haves. Tevens is binnen de MoSCoW methode onvoldoende duidelijkheid over de timing, vooral bij de Won't Haves categorie, komen deze later of helemaal niet aan bod. Als laatste loopt men het risico dat men zich teveel focust op het bouwen van nieuwe features in plaats van het realiseren van technische verbeteringen.

4.1.1 Use cases

Naast het opstellen van user stories en daarmee requirements van de stakeholders, is ook gekeken naar de invloeden op het systeem. De verschillende invloeden op het systeem zijn de User van het systeem, de File system op de pc van de user en de SQL Server database waar de data in moet. Een aantal use cases⁴ is gemaakt, om aan te geven hoe het systeem met deze verschillende actoren omgaat.

4.2 Software ontwikkelingsmethodiek

Een belangrijk onderdeel van software ontwikkelen is de manier waarop software wordt ontwikkeld. Vele verschillende vormen van software ontwikkeling zijn te kiezen en voor elke project is wel een goede te vinden. De meest bekende software ontwikkelingsmethodiek tegenwoordig is Agile, de moderne tegenhanger van de al wat oudere Waterval methode.

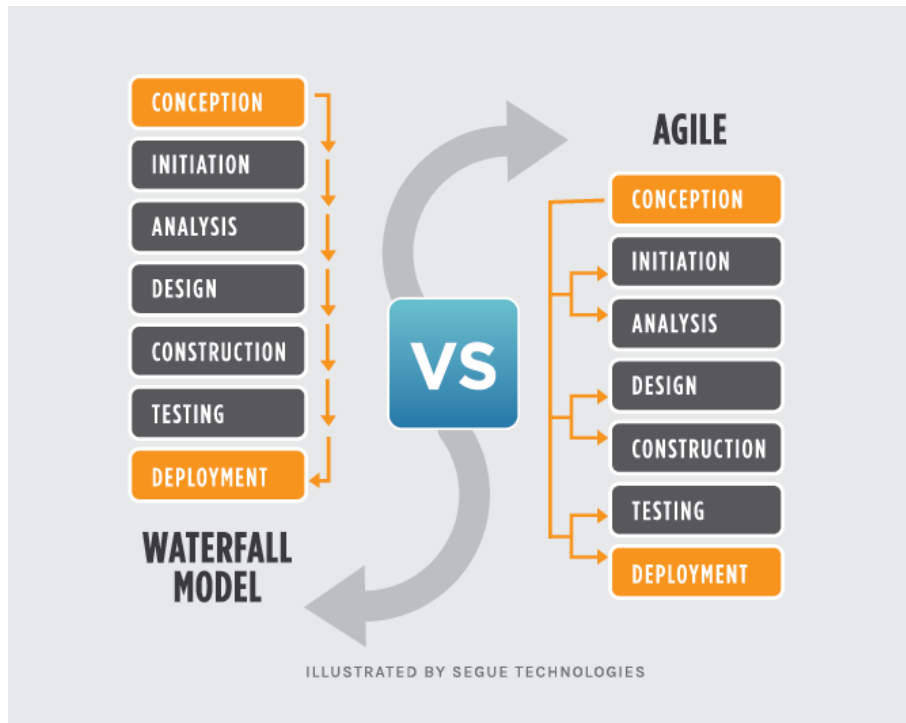
Om een goede keuze te kunnen maken tussen Agile en Waterval is een brede kijk op het project nodig. Het tijdsbestek, teamgrootte, de aard van het project en hoe de stakeholders er tegenover staan zijn belangrijke factoren voor het kiezen van een software ontwikkelingsmethodiek.

Het tijdsbestek is al vastgesteld, dit zijn de 17 weken die de Haagse Hogeschool heeft gegeven voor het afstudeertraject. De teamgrootte is ook vastgesteld, dat is één, alles gebeurt door een en dezelfde persoon, van requirements vergaren en ontwerpen tot coderen en presenteren. De aard van het project is ook vastgesteld. Dit is in gesprekken met de stakeholders duidelijk naar voren gekomen. Dat wat ontwikkeld wordt, is een prototype. De visie van de verschillende stakeholders is in het requirements interview naar voren gekomen en is uitgebreid gedocumenteerd in dat bestand.

⁴ Bijlage 8 Use Cases Iteratie 1 blz. 86-88

4.2.1 Waterval vs Agile

De Waterval methode gaat uit van lineaire software ontwikkeling met aan het begin het vergaren van requirements en het ontwerp en aan het eind het opleveren van de applicatie.



Figuur 17 Waterval vs. Agile

In tegenstelling tot de Waterval methode gaat Agile, wat direct vertaald naar soepel, lenig en vlug, minder rigide om met de manier van ontwikkelen (Lotz, 2018). In plaats van een lineaire aanpak is Agile gebaseerd op een iteratief proces (Lotz, 2018). Deze iteraties worden ook wel 'timeboxes' of 'sprints' genoemd.

Tijdens zo'n iteratie wordt afgesproken wat gedaan gaat worden binnen het tijdsbestek van de iteratie. Als deze elementen niet worden gehaald, dan wordt een nieuwe prioritering gemaakt voor de volgende iteratie. Wanneer een deel wordt opgeleverd, kan het worden gepresenteerd en geëvalueerd bij het team en de consument/stakeholder. Agile is dan ook zeer afhankelijk van een hoge vorm van betrokkenheid van de consument/stakeholder.

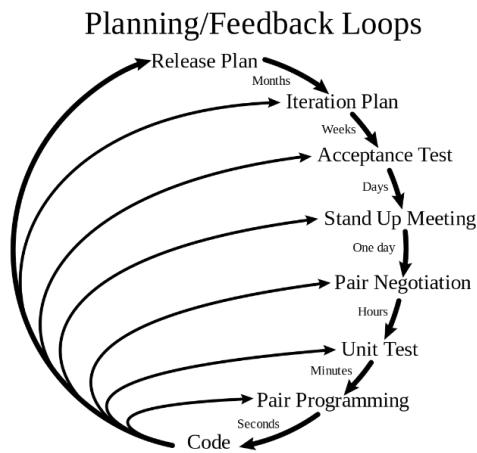
Met het limiet van tijd en teamgrootte in het achterhoofd en omdat het om een prototype gaat, is besloten om zeer regelmatig rond de tafel te zitten met de stakeholders om te presenteren en te evalueren. Dit wordt niet door de Waterval methode ondersteunt en na overweging is besloten de Waterval methode dan ook niet te gebruiken. Daarnaast is het handiger bij kleinere teams om met Agile te werken in plaats van met Waterval (Beng Leau, Khong Loo, Yip Tham & Fun Tan, 2012).

Agile past precies bij de denkwijze van de verschillende stakeholders en is daarom gekozen als algehele software ontwikkelingsmethodiek.

Agile, daarentegen, is een overkoepelende term en houdt verschillende methodieken in. Methodes zoals Extreme Programming, Scrum, Feature Driven Development, Crystal en Kanban.

4.2.2 Extreme Programming(XP)

XP is gebaseerd op een hoge participatie van de twee primaire partijen binnen softwareontwikkeling: de klanten en de developers. Het is een efficiënt en persisterend model, maar heeft wel een groot deel aan discipline. Naast de primaire partijen is ook genoeg participatie nodig van mensen buiten de wereld van informatie technologie (NAYDENOV, 2015).



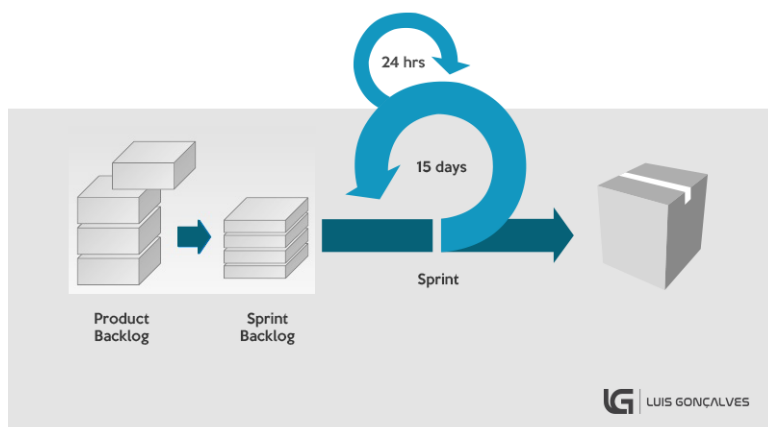
Figuur 18 Overzicht van de XP cyclus

XP streeft naar het makkelijk maken van succesvolle software ontwikkeling in een wereld van veranderende en vage requirements. De belangrijkste karakteristieken van XP zijn korte sprints met kleine release cycli en vervolgens veel feedback daarop, hoge vorm van communicatie en coördinatie, gelimiteerde documentatie en pair programming (Abrahamsson, Salo, Ronkainen & Warsta, 2017).

4.2.3 Scrum

Scrum maakt gebruik van korte iteraties, meestal een maand lang, die sprints worden genoemd. Deze methode promoot transparantie tussen collega's en doet dit door actieve samenwerking tussen development teams, consumenten en stakeholders (NAYDENOV, 2015).

Scrum Process - Overview

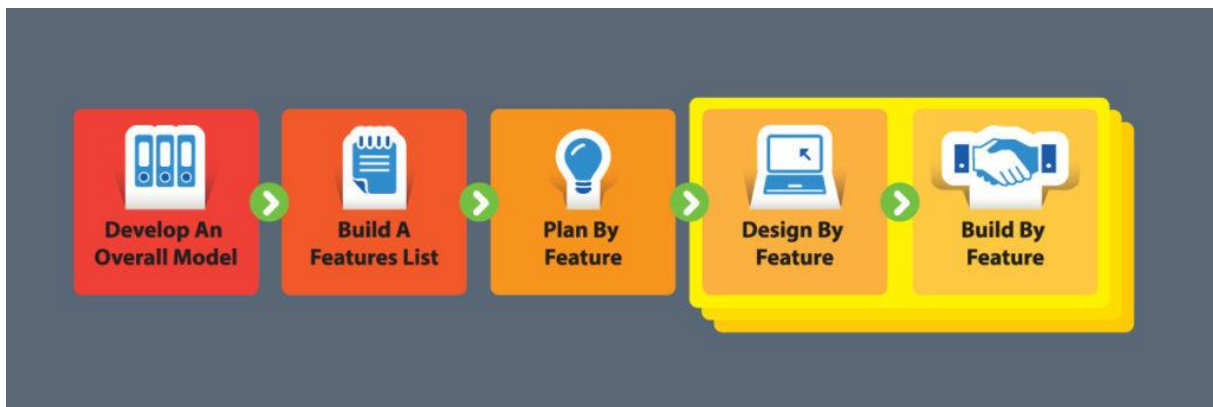


Figuur 19 Overzicht van de SCUM cyclus

De methode is het best te gebruiken voor IT bedrijven die de focus hebben op, met het management hand-in-hand, opleveren van producten en features (Moe & Dingsøyr, 2007). Scrum plant eerst de context en de brede definities van de deliverables en evolueert uiteindelijk deze definities aan de hand van veranderende conditie van de omgeving (Schwaber, 1997).

4.2.4 Feature Driven Development

Feature Driven Development (FDD) maakt net als XP en Scrum gebruik van korte iteratieve cycli, die gepaard gaan met releases van de software die gebouwd is tijdens de cyclus. Elke feature beslaat een klein onderdeel van het gehele project en moet ook zo worden gemodelleerd.



Figuur 20 Overzicht van de stappen van FDD

Het proces van FDD bestaat 5 stappen:

- Stap 1 is het maken van een globaal model.
- Stap 2 is het maken van de feature lijst.
- Stap 3 is het plannen van de features. Dit wordt onder andere gebaseerd op complexiteit, risico en workload.
- Stap 4 is het ontwerpen van de features, die volgens de planning zijn gekozen.
- Stap 5 is het daadwerkelijk ontwikkelen van de feature (Karam, 2017).

4.2.5 De keuze

Uit deze mogelijkheden is de keuze gemaakt om volgens de Extreme Programming methodiek het software development proces uit te voeren. Deze vorm van software development sluit aan bij het karakter van het project, er is namelijk sprake van het ontwikkelen van een prototype. Ook het feit dat de sommige requirements nogal vaag zijn, helpt bij het maken van deze keuze. Daarnaast wordt dit prototype ontwikkeld door één persoon en moet het worden opgeleverd in een zeer kort tijdsbestek.

Wel is gekozen om een paar best practices uit andere vormen van software development mee te pakken. Dit zijn: practices uit Scrum, zoals het definiëren van een back log met daarin verschillende user stories en het gebruik van een Scrumboard, om bij te houden welke elementen actief worden behandeld en welke niet. Daarnaast is gekozen om het project in zo klein mogelijke features op te delen en het iteratief ontwerpen van de applicatie volgens Feature Driven Development.

4.3 Programmeertaal, C# of Python

Na het kiezen van een duidelijke software development methodiek, is het ook van belang om te kijken naar welke programmeertaal gaat worden gebruikt. Doordat het project moet worden uitgevoerd in een kort tijdsbestek, is gekozen om voor bekende talen te gaan. De programmeertalen Python en C# worden door de project uitvoerder het best beheerst en zijn verder uitgewerkt in pro's en con's.

Python heeft het voordeel dat het een hele flexibele taal is. Men kan zowel een Object Georiënteerde, als een Functionele kant opgaan. Het is een high level language met een genereuze standaard library en tal van support voor andere libraries (Raps, 2019).

C# heeft het voordeel dat het een rigide taal is waarvan al heel veel bekend is. Het is Object Georiënteerd en zeer high level. Een zeer uitgebreide documentatie is beschikbaar en wordt permanent bewaakt door de ontwikkelaar van C#, Microsoft. C# ondersteunt ook tal van verschillende libraries die gemanaged worden in hun eigen marktplaats ("Pros/Cons Of The C# Language - C# | Dream.In.Code", 2019).

Python heeft het nadeel dat, door het ontwerp van de taal, fouten kunnen ontstaan die alleen voorkomen tijdens runtime. Daarnaast is de integratie met databases niet native en is de taal, omdat het een interpreted language is, langzamer dan veel populaire tegenhangers (Basel, 2018).

C# heeft het nadeel dat integratie buiten Windows niet mogelijk is, zonder het implementeren van omwegen. Omdat C# onderdeel is van het .NET framework en .NET op Windows moet draaien, is het dus niet mogelijk om de applicatie op andere operating systems te laten draaien, zonder daarvoor omwegen te hanteren (Lysis, 2017).

Uiteindelijk is de keuze gemaakt om verder te gaan met C#. De voornaamste reden voor deze keuze is het feit dat de documentatie ontzettend goed is. Alle vragen die iemand heeft betreffende C#, kunnen worden beantwoord op het Microsoft Developer Network (MSDN). Naast de documentatie is de uitbreidbaarheid van C# met de NuGet Marketplace een erg grote pré. Allerlei verschillende libraries kunnen aan de applicatie worden toegevoegd om zo nog meer functionaliteit mogelijk te maken.

Het gebruik van een Integrated Development Environment (IDE) wordt wijd gezien als een toevoeging in het leven van een developer. Elementen zoals autocompletion, refactoring, code generation, geïntegreerd debuggen, navigeren naar compile of runtime errors, etc. (Skeet, 2008), zijn allemaal grote voordelen. In dit project is gekozen om te programmeren met de IDE Visual Studio 2017 van Microsoft.

Visual Studio 2017 bezit geïntegreerde support voor het programmeren in C# en zorgt voor naadloze toegang tot de NuGet Marketplace. Daarnaast heeft het toegang tot tal van, door Microsoft ontwikkelde, programma's zoals het hele Office pakket en geeft dit de mogelijkheid om op deze programma's te bouwen.

4.4 Versiebeheer & DevOps

De keuze voor een taal en IDE zorgt ervoor dat kan worden begonnen met het programmeren van een applicatie. Om het development proces te vergemakkelijken, is de keuze gemaakt om gebruik te maken van versiebeheer. Versiebeheer vergemakkelijkt het behandelen van een veel voorkomend probleem binnen informatie, namelijk verandering.

Wanneer men met documentatie bezig is en tekst wil toevoegen, aanpassen of verwijderen is dit makkelijk te doen. Echter wanneer men deze acties wil terugdraaien, bestaat een kans dat dat niet mogelijk is. Om dit tegen te gaan worden belangrijke documenten gedupliceerd en veranderd aan de hand van veranderende omstandigheden en context. Veel voorkomende gevallen als 'Kopie van Kopie van Belangrijk Document v4 Final Final.doc' klinken wellicht jammerlijk herkenbaar.

Een stap verder is, wanneer tussen personen documenten moeten worden gedeeld. Wellicht wil men, naast het maken en terugdraaien van aanpassingen, graag weten wie de aanpassing heeft gedaan en waarom en wanneer dit is gebeurd. Een versie beheer systeem kan dit allemaal oplossen (McQuaid, 2014).

De keuze voor versiebeheer kwam neer op het kiezen tussen Azure Repo's, GitLab of GitHub. Deze drie mogelijkheden zijn bekend bij de projectuitvoerder en daarom gekozen om nader te onderzoeken. Alle drie de mogelijkheden zijn gebaseerd op Git. GitLab en Github zeggen het al. Ook Azure Repo's, het versie beheer systeem dat onderdeel uitmaakt van Azure DevOps, het oude Visual Studio Teamservices (bij andere bekend als Team Foundation Services), is gebaseerd op Git.

Git is een open-source gedistribueerd versie beheer systeem. Gedistribueerd houdt in dat de codebase, inclusief de volledige historie van de codebase, wordt weerspiegeld op elke computer van elke developer. Dit in tegenstelling tot een gecentraliseerd versie beheer systeem, waarbij constant een connectie moet zijn met een centrale server waar de codebase is opgeslagen (Chacon, 2009).

GitLab, GitHub en Azure Repo's zijn in grote maten hetzelfde omdat ze alle drie gebaseerd zijn op Git. Groot verschil bij GitLab en Azure in tegenstelling tot GitHub, is de implementatie van DevOps toepassingen. Een belangrijk voordeel, omdat blijkt dat de implementatie van DevOps in het ontwikkelingsproces leidt tot hogere kwaliteit software (Mohamed, 2015)

DevOps is het samenbrengen van Development en Operations. Het wordt gedefinieerd als een methode

- die het mogelijk maakt om beter te communiceren en samen te werken binnen teams van developers en operators,
- om een brug te slaan tussen developers en operators,
- die het makkelijk maakt om continue deployment en continue integratie op te zetten en
- die de zorgen van kwaliteitsborging bij operations en verschillende development praktijken combineert om de performance te verhogen (Jabbari, bin Ali, Petersen & Tanveer, 2016).

De keuze om voor DevOps te gaan, is dan ook gebaseerd op voorgaande paragrafen. Vervolgens is gekozen voor Azure DevOps om dit te realiseren. Daarmee is ook de keuze gemaakt voor Azure Repo's als versie beheer systeem. De keuze voor Azure in plaats van Gitlab is gebaseerd op de naadloze integratie met Visual Studio en C# en het feit dat Gitlab zelf zegt dat .NET beter kan worden gedaan met Azure DevOps (Gitlab, 2018).

In Azure DevOps is een Scrumboard gedefinieerd met de user stories. Aan elke user story is een aantal features toegevoegd om deze user story te vervullen. In elke iteratie zullen, aan de hand van veranderde context en requirements, features worden toegevoegd of verwijderd.

4.5 Het verkrijgen van rapporten

Om de data te verkrijgen worden rapporten gegenereerd uit de motoren van Caterpillar. Deze rapporten worden gegenereerd via CAT Electronic Technician(ET) en kunnen worden verkregen in drie verschillende formats: HTML, PDF en XML. Eén van de requirements van de maintenance engineer, was dat de rapporten in Excel formaat konden worden aangeleverd. Om dit te realiseren is contact gezocht met verschillende CAT dealers over de hele wereld.

Na veel telefonisch en mail contact, met onder andere CAT Michigan, PonPower in Rotterdam en SDK Diesel Support in Werkendam, is geconstateerd dat ten eerste ten aanzien van CAT ET zeer weinig beschikbaar is, alleen wat standaard manuals. Ten tweede dat CAT HQ internationaal gezien praktisch niet te bereiken is. Getracht is om contact te krijgen, maar elke keer werd doorverbonden naar de lokale CAT dealer PonPower. Ten derde dat niemand überhaupt problemen met CAT ET heeft. Een van de ICT medewerkers van SDK Diesel Support vertelde, dat wanneer hij problemen heeft met de software, een nieuwe licentie aanvraagt en ook daadwerkelijk krijgt. Vervolgens wordt verder niet gekeken naar het onderliggende probleem. Dit is besproken met de stakeholders en vervolgens is besloten om deze requirement te laten vervallen.

Door de goede relatie tussen Smit Lamnalco en PonPower is contact ontstaan met Albert Kasbergen, Contractmanager bij PonPower. Samen met Albert is een afspraak gemaakt om langs te gaan bij PonPower om te kijken naar een live draaitest van een 3516B Caterpillar motor. Deze meeting was opgezet om een kleine oriëntatie mogelijk te maken in de wereld van de maritieme industrie.

Met een IT collega van Albert is tijdens het bezoek besproken op welke manier de data wordt gegenereerd en hoe dit kan worden veranderd of verbeterd. De communicatie in de motor wordt gedaan via de J1939 standaard van de Society of Automotive Engineering (SAE).

Deze standaard is de communicatie bus binnen voertuigen en wordt aanbevolen om te gebruiken in communicatie en diagnostiek tussen voertuig componenten. De J1939 standaard maakt gebruik van de ISO 11992 fysieke laag, de Controller Area Network (CAN). CAN is ideaal voor applicaties die een hoog aantal aan, zeer betrouwbare, korte berichten in een klein tijdsbestek vereisen. Daarnaast is de foutbestendigheid van CAN een erg groot voordeel. Het is namelijk zo dat, mocht een node binnen het netwerk uitvallen, deze automatisch komt te vervallen om te zorgen dat niet het hele netwerk wordt meegenomen in de uitval (Corrigan, 2002).

Uiteindelijk is geconstateerd dat teveel moet gebeuren om direct op de J1939 standaard te acteren en dat dit niet alleen ver buiten de scope van deze opdracht valt, maar ook deels buiten de opleiding Software Engineering. Samen met de stakeholders is besloten om dit achterwege te laten en door te gaan met CAT ET en de bijbehorende drie formats.

4.6 Ontwerptool

Tijdens het opstellen van het requirements document en oriëntatie van de rapporten, is gezocht naar een goede tool voor het maken van ontwerpdigrammen. Ontwerpdigrammen zijn goede representaties van de ideeën van de software engineer en kunnen een goed beeld geven voor de buitenwereld van het verloop van het project. Na een tijd zoeken is LucidChart op het internet gevonden.

LucidChart ondersteunt meerdere vormen van diagrammen en tekeningen, van Flowcharts tot Business proces mapping, van UML diagrammen tot Entity Relationship diagrammen, van Wireframe en Iphone Mockups tot volledige vloerplannen voor architecten (Fox, 2017).

Met LucidChart zijn tijdens de ontwerpstadia verschillende diagrammen gemaakt om de ideeën die zijn opgekomen te visualiseren. Er is een Data Transfer diagram⁵ gemaakt om zowel de plaats van het te implementeren deel te visualiseren alsmede de datastroom binnen Smit Lamnalco. Na het visualiseren van de datastroom, is van de opgestelde Use cases een Use case Diagram⁶ gemaakt.

Na het maken van het Use case diagram, zijn verschillende Sequence diagrammen⁷ gemaakt. Deze diagrammen geven weer wat er gebeurt bij de verschillende invloeden van de actoren op het systeem.

4.7 Desktop of Web

“Wat voor een applicatie is het handigst om te gaan maken?”, is vaak als vraag naar voren gekomen. In de requirements staat dat de werknemers geen educatie nodig moeten hebben om de applicatie te gebruiken. Dit was een belangrijke factor om aan te houden. Het moet een applicatie worden die goed is te overzien en intuïtief en makkelijk te hanteren.

De keuze voor het platform van de applicatie was tussen Desktop of Web. De grootste drawback was echter de connectie op het schip. Er wordt door Smit Lamnalco op elk schip een connectie beschikbaar gesteld via KVH Industries. KVH Industries is één van de meest toonaangevende fabrikanten van oplossing voor hoge snelheid internet, televisie en spraakdiensten ter wereld. Dit doen zij via satelliet zowel op land als ter zee (KVH Industries, 2019).

Deze connectie kost Smit Lamnalco een aanzienlijk bedrag per schip, terwijl de connectie maximaal 128 kB per seconde en maar een maxima van 2 GB per maand bedraagt. Om deze connectie te gebruiken voor een web applicatie leek niet verstandig. Daarom is besloten om geen web applicatie te ontwikkelen.

Eén van de voordelen van een desktop applicatie is dat deze makkelijk te vervoeren is. Een .exe-bestand delen kan via een usb of een ander file transfer systeem. Ook kan een desktop applicatie offline worden gebruikt, wat, zeker met de gelimiteerde connecties aan boord, zeer gunstig uitkomt.

Voor het kiezen van een framework geeft Microsoft drie mogelijkheden: .NET Framework, .NET Core en Xamarin. Dit project gaat niet over het ontwikkelen van een mobiele applicatie, het gebruik van Xamarin is daarmee overbodig. Omdat er geen sprake is van cross-platform mogelijkheid is het niet nodig om voor .NET Core te gaan (MSDN, 2018).

Daarmee is gekozen om een Console Applicatie te ontwikkelen die draait op het .NET Framework 4.6.1.

⁵ Bijlage 9 Data Transfer diagram Iteratie 1 blz. 97

⁶ Bijlage 9 Usecase diagram Iteratie 1 blz. 93

⁷ Bijlage 9 Sequence diagrammen Iteratie 1 blz. 106-111

4.8 Eerste versie Console App

4.8.1 Ontwerp

Na de keuze om door te gaan met een Console Applicatie, is een bijbehorend Klasse diagram⁸ gemaakt. Het idee was om een algemene Controller de meeste logica af te laten handelen. De user krijgt een console prompt voor zich en loopt daarmee het proces van document transformatie door.

Een Loader zorgt voor het inladen van de documenten vanuit het file systeem van de user, een Transformer voor het transformeren van de documenten en een Saver voor de uiteindelijke doorvoer naar de SQL Server database.

Om de database te gebruiken, is een connectie daarmee van belang. Om deze connectie af te handelen, wordt hier een aparte klasse voor opgericht: de Connector. De Connector gaat gebruik maken van de Data.SqlClient library van Microsoft.

4.8.2 Implementatie

De implementatie van de library helpt bij het opzetten van een connectie met een SQL Server database. De connectie wordt opgebouwd door gebruik te maken van een *connectionString*. In de *connectionString* wordt een aantal argumenten meegegeven, zodat de *SqlClient* een *SqlConnection* kan openen. In de *connectionString* worden de *Data Source*, *Initial Catalog*, *User ID* en *Password* meegegeven. Deze variabelen staan respectievelijk voor de server naam, de database naam en de inloggegevens voor de database.

```
1. connetionString="Data Source=ServerName;  
2. Initial Catalog=DatabaseName;  
3. User ID=UserName; Password=Password"
```

Na dit te hebben doorgegeven aan een *SqlConnection* wordt deze met de *.Open* methode geopend.

```
1. cnn = new SqlConnection(connetionString);  
2. cnn.Open();
```

Om een SQL statement uit te voeren wordt de klasse *SqlCommand* gebruikt. Deze geeft men een SQL statement mee in de vorm van een string en de connectie van het type *SqlConnection*.

```
1. sql = "Your SQL Statement Here , like Select * from product";  
2. command = new SqlCommand(sql, connection);
```

⁸ Bijlage 9 Klasse diagram Iteratie 1 blz. 99

De *SqlCommand* wordt met een *.ExecuteReader* methode uitgevoerd, dit geeft vervolgens een *SqlDataReader* terug. Vervolgens kan de data worden ingezien met een *.Read* methode. Wanneer geen query wordt uitgevoerd, maar een Create, Update of Delete (CRUD) statement, wordt in plaats van een *.ExecuteReader* een *.ExecuteNonQuery* methode uitgevoerd.

```
1. //Statement for Select commands
2. using(SqlDataReader reader = command.ExecuteReader())
3. {
4.     while (reader.Read())
5.     {
6.         Console.WriteLine(String.Format("{0}, {1}",
7.             reader[0], reader[1]));
8.     }
9. }
10. //Statement for CRUD commands
11. SqlCommand command = new SqlCommand(queryString, connection);
12.     command.Connection.Open();
13.     command.ExecuteNonQuery();
```

Naast implementatie van de Connector en de connectie met de database, is geprobeerd een goede implementatie te vinden voor de Loader. Bedacht werd om via een, door de user uitgekozen, folder de documenten in te laden. De Loader zou dan automatisch deze folder uitlezen om zo alle mogelijke documenten in te laden en vervolgens te transformeren.

Op het internet is gezocht hoe dit te implementeren is. Uiteindelijk is de *FileSystemWatcher* gevonden (MSDN, 2002). Na deskresearch werd duidelijk dat dit niet betrouwbaar werkt en hebben meerdere mensen veel problemen ondervonden na het implementeren van de *FileSystemWatcher* (Asmul, 2014). Ondanks de vele afraders is toch besloten om een kleine implementatie te maken van de *FileSystemWatcher* en te kijken hoe dit in zijn werk gaat.

Een aparte console applicatie is gemaakt om de *FileSystemWatcher* te testen. Een methode is ontwikkeld die de gehele setup van de *FileSystemWatcher* afhandelt.

```
1. private static FileSystemWatcher SetupFSW()
2. {
3.     FileSystemWatcher fsw = new FileSystemWatcher();
4.     fsw.NotifyFilter = NotifyFilters.FileName;
5.     fsw.Created += OnChanged;
6.
7.     return fsw;
8. }
```

Een event wordt toegevoegd aan de *Created* event lijst van de *FileSystemWatcher*. Wanneer documenten in de folder worden gekopieerd of gesleept, wordt een *Created* event afgevuurd. De *FileSystemWatcher* itereert vervolgens over alle methodes die aan de *Created* event lijst zijn toegevoegd.

```
1. private static void OnChanged(object sender, FileSystemEventArgs
2. args) => Console.Out.WriteLine("changed");
```

De applicatie werkt als volgt: Als de applicatie start, wordt een *FileSystemWatcher* object aangemaakt. Vervolgens wordt aan de user gevraagd om een folder. Bestaat deze, dan wordt deze toegevoegd aan het *.Path* van het *FileSystemWatcher* object. Uiteindelijk wordt de *FileSystemWatcher* 'aangezet', door aan *EnableRaisingEvents* de waarde *true* te erkennen.

```
1. FileSystemWatcher fileWatcher = SetupFSW();
2.
3. Console.Out.WriteLine("What folder should be checked");
4. string path = Console.ReadLine();
5.
6. while (!Directory.Exists(path))
7. {
8.     Console.Out.WriteLine("The path does not exist \n "
9.                             + "Try again");
10.    path = Console.ReadLine();
11. }
12.
13. fileWatcher.Path = path;
14.
15. fileWatcher.EnableRaisingEvents = true;
16. Console.Out.WriteLine("Started checking folder on path: " +
17.                          path);
```

Na het starten van de applicatie wordt een console prompt geopend. Wanneer de user dan een file in de directory aanmaakt, kopieert of sleept, zal het woord 'changed' verschijnen in de console.

Uiteindelijk is in overleg met de stakeholders gekozen om dit niet te implementeren, omdat de kans bestond dat het niet robuust genoeg was. Het risico wat hiermee ontstond, was reden genoeg om voor sommige stakeholders af te haken. Besloten is om verder te zoeken naar een alternatief.

Voor de implementatie van de Transformer klasse is bedacht om te kijken welk format handig zou zijn om te bewerken. Gekeken is of vanuit het PDF format kan worden gewerkt. Vervolgens is op het internet gezocht naar een library om deze functionaliteit niet eigenhandig te hoeven ontwikkelen.

Na een tijd zoeken is de iTextSharp library gevonden. Deze PDF library geeft de mogelijkheid om PDF documenten te maken, aan te passen, te inspecteren en te onderhouden.

Het is bijvoorbeeld mogelijk om vanuit een database of XML bestand een document of rapport te genereren in PDF. Maar belangrijker voor dit project was de mogelijkheid om de data uit een PDF te scrapen en om te vormen naar een simpel tekst bestand. Hier kan vervolgens verdere logica op los worden gelaten.

Om deze library te gebruiken is deze van de NuGet Marketplace gehaald en gerefereerd in het project. Vervolgens wordt met twee *using* statements tegen de compiler gezegd dat de library wordt gerefereerd en dat voor sommige functionaliteit naar deze library wordt verwezen.

```
1. using iTextSharp.text.pdf;
2. using iTextSharp.text.pdf.parser;
```

Om te kijken of dit een goed uitgangspunt is, is een methode ontwikkeld die zorgt dat een PDF bestand wordt geprint naar de console. Het te transformeren document wordt in de vorm van een *filePath* meegegeven aan deze methode.

Met dit *filePath* wordt een *PdfReader* object gebouwd, dit object vormt de brug tussen document en applicatie. Wanneer de applicatie tekst vanuit een PDF wil trekken, is een vertalingsstrategie nodig. Hiervoor geeft iTextSharp de keuze tussen de *SimpleTextExtractionStrategy*, de *FilteredTextEventListener*, de *GlyphTextEventListener* en de *LocationTextExtractionStrategy*.

Omdat men noch op events noch op locatie gebaseerd wil extracten, is gekozen voor de *SimpleExtractionStrategy*.

Over de pagina's van het PDF document wordt geïtereerd en van elke pagina wordt de tekst met een *PdfTextExtractor* onttrokken. De tekst wordt vervolgens opgeslagen in een lokale array, waarmee uiteindelijk de tekst wordt weergegeven.

```
1.     public void Extract(string filePath)
2.     {
3.         PdfReader reader = new PdfReader(filePath)
4.         ITextExtractionStrategy Strategy = new
SimpleTextExtractionStrategy();
5.         string[] lines = { };
6.         for (int x = 1; x <= reader.NumberOfPages; x++)
7.         {
8.             string page = "";
9.             page = PdfTextExtractor.GetTextFromPage(reader, x,
Strategy);
10.            lines = page.Split('\n');
11.        }
12.        foreach(string line in lines)
13.        {
14.            Console.WriteLine(line);
15.        }
16.        reader.Close();
17.    }
18. }
```

Deze methode geeft netjes de data lijn voor lijn weer en maakt de data klaar voor verdere extractie.

Naast een Transformer en een Loader klasse, is ook een Saver klasse nodig die verantwoordelijk is voor het opslaan naar de database, nadat de Connector de connectie heeft gelegd.

Omdat de datastructuur binnen de database nog niet bekend is, is besloten om implementatie van de Saver klasse te verplaatsen naar een latere iteratie.

4.9 Onderzoek: Methode van data transfer

Een van de vragen uit het plan van aanpak was, 'Wat is de beste methode om de data transfer tot stand te brengen?' Om hier achter te komen, is een onderzoek⁹ gedaan naar verschillende manieren van data transfer. Omdat geen tijd was om een grote hoeveelheid manieren te onderzoeken, is eerst een vooronderzoek¹⁰ gedaan om een afbakening toe te passen.

Om het onderzoek zo kort mogelijk te houden, is niet gekeken naar implementaties van bepaalde protocollen, maar naar de protocollen zelf. Het vooronderzoek ging tussen drie mogelijke kandidaten, een Mail server, een op Resilio gebaseerd systeem of een volledig nieuwe applicatie, gebaseerd op een ander file transfer protocol.

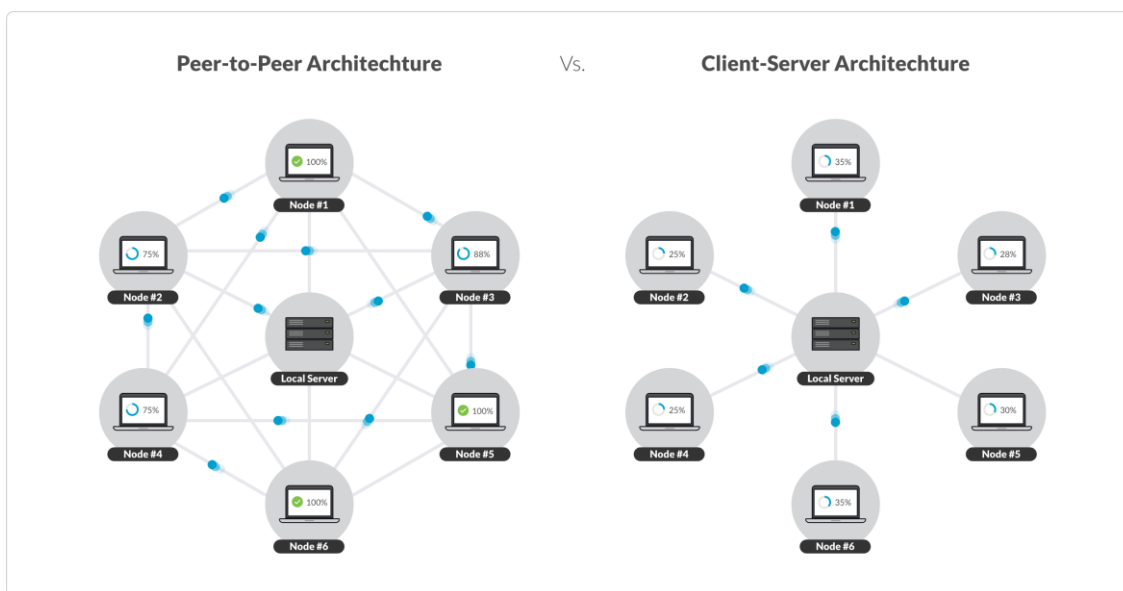
⁹ Bijlage 7 Onderzoek: Transfer mogelijkheden blz. 85-86

¹⁰ Bijlage 6 Vooronderzoek: Transfer mogelijkheden blz. 84-85



Figuur 21 Voorbeelden van mail applicaties

De Mail server kan in de vorm van een Outlook of Gmail of zelfs een geïntegreerde Simple Mail Transfer Protocol (SMTP) server.



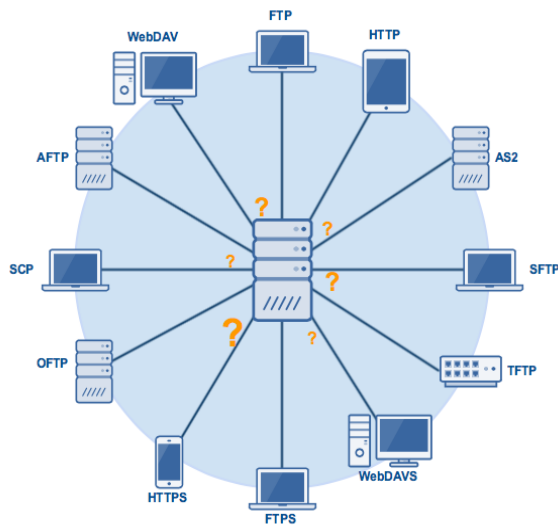
Figuur 22 Resilio P2P netwerk vs Client Server netwerk

Resilio is gebaseerd op het peer-to-peer (P2P) netwerk protocol, wat zorgt voor een schaalbaar real-time synchroniserend file systeem, voor wanneer men de Cloud niet wil gebruiken.

In tegenstelling tot bijvoorbeeld een Microsoft Distributed File System Replication, wat gebaseerd is op het Client-Server model, gebruikt Resilio het P2P netwerk. Dit houdt in dat alle Nodes in het systeem met elkaar communiceren in plaats van een gecentraliseerde server die alle communicatie afhandelt (Konstadin, 2018).

Smit Lamnalco heeft een systeem gebaseerd op Resilio genaamd SLDoX. Dit is het eigen centrale file systeem waar elk schip mee verbonden is en waar alle rapporten, certificaten, maintenance jobs, contracten, etc. in zijn opgeslagen.

Bij het maken van een nieuwe applicatie kan gebruik worden gemaakt van vele verschillende soorten transfer protocollen. Zo heb je onder andere het File Transfer Protocol (FTP), één van de oudste transfer protocollen en het HyperText Transfer Protocol (HTTP), de standaard die gaat over het internet.



Figuur 23 Transfer protocollen

Omdat zowel FTP als HTTP intrinsieke problemen met veiligheid hebben, hebben deze een Secure versie namelijk, FTPS en HTTPS. Deze implementaties maken gebruik van de Secure Socket Layer (SSL).

Veel andere transfer protocollen zijn niet geschikt voor dit project en slaan vooral op speciale toepassingen: zoals Trivial File Transfer protocol (TFTP), die vooral wordt gebruikt voor network management en het Applicability Statement 2 (AS2) protocol, die wordt gebruikt voor Electronic Data Interchange applicaties (Villanueva, 2015).

Eén van de Must Have requirements van het project, is dat de user niet iets nieuws hoeft te leren voor het gebruik maken van de applicatie. Dit neemt meteen al een hele hoop aan opties weg. Na gesprekken met de stakeholders, is afgesproken dat mag worden uitgegaan van het feit dat elke user weet hoe deze met een Mail server of Resilio moet omgaan.

Het vooronderzoek concludeerde dan ook geen nieuwe applicatie te bouwen voor het verzamelen van de data, maar om gebruik te maken van ofwel een op Resilio gebaseerde add-on op SLDox of een Mail server.

Voor verder onderzoek is geprobeerd om twee applicaties te bouwen. Eerst is geprobeerd om een, op SMTP gebaseerde, mail server te ontwikkelen waarmee verschillende documenten kunnen worden verstuurd.

Helaas is tijdens de ontwikkeling geconstateerd dat de SMTP-poort van het netwerk van Smit Lamnalco niet toegankelijk is. Na deze constatering is een gesprek aangevraagd met de stakeholders.

Tijdens dit gesprek is geconstateerd dat de focus verkeerd ligt en dat aan de verkeerde onderdelen wordt gewerkt. De stakeholders willen dat de focus wordt verlegd van het verzamelen en verplaatsen van data, naar het transformeren en opslaan van data.

4.10 Reactie na set back

Na het gesprek is meteen gekeken naar wat de stakeholders veranderd willen zien. Duidelijk kwam naar voren dat de verantwoordelijkheden die aan de applicatie zijn gegeven, niet in lijn lagen met wat de stakeholders ervoor bedacht hadden. Hier is dan ook veel aan veranderd. Met deze verandering van verantwoordelijkheid, veranderde de manier van invloed op het systeem ook. Dit had vervolgens invloed op de use cases.

Naast de invloed op de use cases had het wegvallen van bepaalde verantwoordelijkheid ook invloed op het klasse diagram en het data transfer document. Deze zijn beide aangepast. De planning is

omgegooid en besloten is om veel meer tijd te steken in het proces van transformatie. Daarnaast is de scoping waar de applicatie allemaal verantwoordelijk voor is, veranderd.

4.10.1 Veranderde verantwoordelijkheid applicatie

De applicatie is niet meer verantwoordelijk voor het verzamelen van data van de schepen en niet meer verantwoordelijk voor het opslaan in de database. Samen met de Application Manager Emad Zaïd is afgesproken dat met tools gaat worden gewerkt voor het inladen van de data in de SQL Server database. Deze tools zijn Microsoft MessageQueue en Tibco Scribe Insight. Maar voordat dat mogelijk was, was het belangrijker om eerst de applicatie te maken en er voor te zorgen dat überhaupt data aanwezig was om op te slaan.

Doordat eerder is besloten om de FileSystemWatcher van Microsoft niet te gebruiken maar een alternatief te zoeken, is in overleg met de de Application Manager Emad Zaïd besloten om het proces te veranderen. Het idee is nu om tijdens het transformatie proces de documenten uit een lokale folder in de AppData folder te halen.

4.10.2 Nieuwe Use case diagram en Use cases

De veranderde verantwoordelijkheid van de applicatie heeft tot gevolg dat de use cases¹¹ zijn veranderd. In plaats van drie actoren zijn er nog maar twee overgebleven, de User en de File System. Dit is dan ook weergegeven in het nieuwe Use case diagram¹².

4.10.3 Nieuw scope

De applicatie was verantwoordelijk voor het opslaan van data. Door de verandering in requirements is dat niet meer zo. Dit word niet meer in de applicatie geïmplementeerd. Wel wordt het verplaatsen en transformeren geïmplementeerd.

Duidelijk is aangegeven dat gewerkt gaat worden met de tools Microsoft MessageQueue en Tibco Scribe Insight. Deze worden in de scope meegenomen.

4.10.4 Nieuw data format

Waar eerst bedacht was dat het PDF format van de rapporten een goed uitgangspunt leek te zijn, bleek na nadere verdieping, teveel hardcoding nodig was. Door verder te zoeken in de verschillende rapporten, is gebleken dat elke motor gebaseerd is op een ander motormodel van Caterpillar. Dit leidt tot heel verschillende rapporten, wat tot gevolg heeft veel opzoek werk binnen getransformeerde rapporten. Uiteindelijk is rond de tafel gezeten met de Application Manager Emad Zaïd en is besloten om door te gaan met het XML format van de rapporten.

Met het kantoor in Jordanië is mailcontact gezocht en zijn van zes verschillende schepen rapporten ontvangen. Deze zes schepen hebben ieder twee main engines, in totaal zijn twaalf rapporten ontvangen. Na het bekijken van de rapporten is geconstateerd dat van twee verschillende modellen rapporten zijn ontvangen; het 3516B model en het C18ACERT model. De verschillen in de modellen zijn ondergebracht in een Excel worksheet¹³ om een goed overzicht te creëren voor de stakeholders.

¹¹ Bijlage 8 Use cases Iteratie 2 blz. 88-89

¹² Bijlage 9 Use case diagram Iteratie 2 blz. 94

¹³ Bijlage 5 Data Mapping blz. 81-83

4.11 XML en Datastructuren

Na het contact met het kantoor in Jordanië en het gesprek met de Application Manager Emad Zaïd is onderzoek gedaan naar het XML format van de rapporten. Hieruit is gebleken dat de documenten niet zijn opgebouwd voor logica maar voor presentatie. Dit is waarschijnlijk gedaan door middel van XSLT gebruik in CAT ET.

eXtensible Stylesheet Language Transformations (XSLT) wordt gebruikt voor het omzetten van de informatie van een XML document naar een ander format of een ander gestructureerd XML document.

4.11.1 XML, wat en hoe

Reguliere XML opbouw bestaat uit een tag en een value, de tag geeft veelal aan wat het is wat de value weergeeft. Het idee van XML is gebaseerd op self-describing syntax, je ziet waar het over gaat. Voorbeeld om in een XML structuur de leeftijd van iemand aan te geven is:

```
1. <leeftijd>25</leeftijd>
```

Dit heet een XML-node of element. XML is gebaseerd op een boom structuur, het heeft net als een boom wortels, takken en bladeren. Om de relaties weer te geven, wordt de ouder-kind relatie benoemd. Als een XML-node onder een andere node valt heet deze de child-node (kind), andersom krijgt deze de naam parent-node (ouder).

De root, wortel, is de eerst node in een document:

```
1. <persoon></persoon>
```

Hieraan kunnen dan verschillende branches, takken, worden toegevoegd:

```
1. <persoon>
2.   <leeftijd></leeftijd>
3.   <naam></naam>
4. </persoon>
```

De leaves, bladeren, van de boomstructuur zijn de values binnen de branches:

```
1. <persoon>
2.   <leeftijd>25</leeftijd>
3.   <naam>Mick</naam>
4. </persoon>
```

In dit voorbeeld is het duidelijk dat het om een persoon gaat met de naam Mick en de leeftijd 25. Om extra informatie te geven over branches, gebruikt men attributen. Om aan te geven dat leeftijd uitgedrukt wordt in jaren, voegt men een *meeteenheid* attribuut toe met de value *jaren*:

```
1. <persoon>
2.   <leeftijd meeteenheid="jaren">25</leeftijd>
3.   <naam>Mick</naam>
4. </persoon>
```

Met XML kan een heel duidelijke structuur van allerlei verschillende elementen worden gemaakt en is het door de self-describing syntax duidelijk wat wordt bedoeld.

De self-describing syntax ontbrak compleet in de documenten geëxporteerd uit CAT Electronic Technician. Wat hierboven besproken is, is meestal de normale gang van zaken bij XML, maar bij deze rapporten ziet dat er zo uit:

```

1. <Block>
2.   <Table Width="100%" Align="Left" FontSize="12">
3.     <Headings>
4.       <Heading Bold="yes" Align="Left" Underline="no">Parameter</Heading>
5.       <Heading Bold="yes" Align="Left" Underline="no">Value</Heading>
6.     </Headings>
7.     <Rows>
8.       <Row/>
9.       <Row>
10.        <Value Bold="no" Align="Left" Underline="no">Engine Serial Number</Value>
11.        <Value Bold="no" Align="Left" Underline="no">TTJ00555</Value>
12.      </Row>
13.    </Rows>
14.  </Table>
15.</Block>

```

De data is beschikbaar alleen de logica ontbreekt hier. Met dit format is geen duidelijke self-describing syntax meer te vinden en dat maakt het ontcijferen des te lastiger.

4.11.2 XPath, de weg er naar toe

Om door een XML boomstructuur heen te itereren is XPath bedacht. XPath maakt gebruik van zogenoemde 'pad expressies', deze 'pad expressies' maken gebruik van verschillende tekens om door de boomstructuur heen te lopen:

Expressie	Beschrijving
naam	Selecteer alle elementen met de naam "naam"
/	Selecteer absoluut vanaf de root node
//	Selecteer alle elementen in het document, ongeacht waar deze zich bevinden
@	Selecteer een attribuut

Wat, in het geval van het voorbeeld persoon, gebeurd is het volgende:

Expressie	Resultaat
persoon	Resulteert in alle 'persoon' nodes
/persoon	Resulteert in het root element 'persoon', verschil tussen met en zonder / is dat met altijd een absoluut pad naar een element weergeeft
persoon/naam	Resulteert in alle 'naam' elementen welke vallen onder 'persoon'
//leeftijd	Resulteert in alle 'leeftijd' elementen ongeacht waar deze zich bevinden
//@meeteenheid	Resulteert in alle attributen met de naam 'meeteenheid'

Om aan te geven hoe onlogisch de structuur binnen de CAT ET rapporten is, is de uitleg van XPath nodig. Voorbeeld van een rapport om bij een Engine Serial Number van een CAT Engine Model 3516B te komen is als volgt.

Product Status Report	
Parameter	Value
Engine Serial Number	TTJ00555

`/XMLData/Reports/Report/Data/Block[2]/Table/Rows/Row[2]/Value[2]/text()`

Om de documenten te bekijken en om Xpath te genereren, wordt de tool Altova XMLSpy gebruikt. Altova is één van de best verkopende XML editors op de markt. Het is mogelijk om zowel XML als XPath, XQuery, XSLT en XBRL te genereren en te valideren (Altova, 2005).

4.11.3 Nieuwe structuur met XML

Om dit logischerwijs in de database te krijgen, is besloten om een tussenproduct te maken, een nieuw XML document met de self-describing syntax herintroduceerd. Hiervoor is een datastructuur gemaakt in overleg met de Application Manager Emad Zaïd en de Senior Maintenance System Engineer Hubert Plattel. Met de Senior Maintenance System Engineer Hubert Plattel is besproken welke data het belangrijkste was om op te slaan.

In het gesprek is het voorbeeld document¹⁴ gebruikt als leidraad voor de datastructuur van het tussenproduct. Het document bestaat uit verschillende kopjes waar zich tabellen met data onder bevinden. Omdat op een motor meerdere Electrical Control Modules zitten, zijn meerdere headers beschikbaar. In het geval van het voorbeeld document is naast de 3515B engine op de Portside (links) kant header, ook een back-up header aanwezig.

Het was van belang dat alleen relevante data ten aanzien van de motor zelf wordt opgenomen. Alle andere data mag worden genegeerd. Uiteindelijk is in overleg een keuze gemaakt welke data relevant genoeg wordt bevonden. Deze data is opgenomen in een Data Definition Document¹⁵.

Aan de hand van het Data Definition Document is een datastructuur ontworpen in XML. In deze datastructuur was het van belang dat de self-describing syntax weer werd herintroduceerd, zoals deze te vinden is in de meeste XML documenten. Om te beginnen is een goede representatieve root-node bedacht.

```
1. <EngineReport></EngineReport>
```

Daarna is bedacht om alle algemene info zoals datum, serial number, locatie etc. eronder als kind elementen te plaatsen.

```
1. <EngineReport>
2.   <DateTime>20-04-2001T13:42:38</DateTime>
3.   <SerialNumber>TTJ00555</SerialNumber>
4.   <EquipmentID>SNMR T10 PORT</EquipmentID>
5.   <VesselName>Hashim 3</VesselName>
6.   <EngineLoc>Port</EngineLoc>
7.   <EngineType>ME</EngineType>
8.   <EngineModel>3516B</EngineModel>
9. </EngineReport>
```

Na de algemene data te hebben gemodelleerd, zijn de verschillende logs gemodelleerd. Te zien in het Data Definition Document zijn vier verschillende soorten logs aanwezig, Active Event, Logged Event, Active Diagnostic en Logged Diagnostic.

¹⁴ Bijlage 10 Voorbeeld Document blz. 122-132

¹⁵ Bijlage 11 Data Definition document blz. 132-133

Deze zijn op de volgende manier gemodelleerd.

```
1. <EngineReport>
2.   <Log codeType="LoggedDiagnostic">
3.     <Code>No Logged Diagnostics Codes</Code>
4.     <Description></Description>
5.     <Occurrence></Occurrence>
6.     <First></First>
7.     <Last></Last>
8.   </Log>
9.   <Log codeType="ActiveDiagnostic">
10.    <Code>No Active Codes</Code>
11.    <Description></Description>
12.  </Log>
13. </EngineReport>
```

Elke Log wordt vergezeld met een *codeType*-attribuut wat aangeeft wat voor Log het is. Dit om verwarring tegen te gaan bij de user. Na de Logs zijn de Current Totals gemodelleerd op ongeveer dezelfde wijze als de Logs, vergezeld met een *type*-attribuut en een *unit*-attribuut. Weer ter verduidelijking voor de user.

```
1. <EngineReport>
2.   <CurrentTotals type="Total Fuel" unit="L">
3.     <Value>434092</Value>
4.   </CurrentTotals>
5.   <CurrentTotals type="Total Time" unit="H">
6.     <Value>7560</Value>
7.   </CurrentTotals>
8. </EngineReport>
```

Bij de Life Time: Time vs ***¹⁶ is ook gekozen om een representatief attribuut toe te voegen. In dit geval betekende dat, bij Engine Load, dat zo werd gemodelleerd.

```
1. <EngineReport>
2.   <EngineData type="EngineLoad">
3.     <Range>11-20</Range>
4.     <Hours>25</Hours>
5.     <Percent>25</Percent>
6.   </EngineData>
7.   <EngineData type="EngineLoad">
8.     <Range>0-10</Range>
9.     <Hours>20</Hours>
10.    <Percent>20</Percent>
11.  </EngineData>
12. </EngineReport>
```

De Range geeft aan wat de omvang van de data is. Hours geeft het aantal uur binnen die omvang aan. Percent geeft het percentage wat in die omvang is besteed aan, ten opzichte van het totaal aantal gedraaide uren.

¹⁶ Omdat er meerdere Life Time headers zijn, maar hetzelfde gemodelleerd zijn, zijn ze hier gerepresenteerd met ***

Ook de Monitoring System en de Configuration zijn op dezelfde manier gemodelleerd.

```
1.  <EngineReport>
2.    <Monitoring type="High Engine Oil Temperature">
3.      <Severity>1</Severity>
4.      <State>On</State>
5.      <TripPoint>110C</TripPoint>
6.      <Delay>5</Delay>
7.    </Monitoring>
8.    <Monitoring type="High Engine Oil Temperature">
9.      <Severity>2</Severity>
10.     <State>On</State>
11.     <TripPoint>115C</TripPoint>
12.     <Delay>30</Delay>
13.   </Monitoring>
14.   <Configuration type="Low Idle Speed" unit="rpm">
15.     <Value>540</Value>
16.   </Configuration>
17.   <Configuration type="Crank Terminate RP" type="rpm">
18.     <Value>600</Value>
19.   </Configuration>
20. </EngineReport>
```

Hiermee is de self-describing syntax herintroduceerd en is het een stuk duidelijker te begrijpen.

4.12 Tweede versie Console App

Tijdens de afgelopen iteratie is geconstateerd dat het gebruik van de Extreme Programming methodiek is afgezwakt en dat meer volgens Feature Driven Development is gewerkt. Vervolgens is besloten om van de Extreme Programming methodiek af te stappen en door te gaan met Feature Driven Development.

In Azure DevOps zijn de nieuwe features toegevoegd en aan de hand van de features zijn verdere iteraties gepland. Om te zorgen dat een tweede set back niet mogelijk is, zijn vaste gesprekken met de Group Fleet Technology Manager Sander Woltheus gepland. In het eerste gesprek heeft een evaluatie van afgelopen iteratie plaatsgevonden.

Na de evaluatie is begonnen aan het development proces van de transformatie applicatie.

4.12.1 Ontwerp

Met de nieuwe datastructuur en de nieuwe verantwoordelijkheden is het klasse diagram¹⁷ aangepast. De Saver en Connector klassen zijn niet meer nodig en verdwijnen uit het klasse diagram. Ook zijn nieuwe Sequence¹⁸ diagrammen gemaakt voor de veranderde use cases.

¹⁷ Bijlage 9 Klasse diagram Iteratie 2 blz. 100

¹⁸ Bijlage 9 Sequence diagrammen Iteratie 2 blz. 112-114

4.12.2 Implementatie

Eerst is aandacht besteed aan het implementeren van de weg naar de datastructuur toe. Gekeken is of een makkelijke manier bestond om XML documenten uit te lezen in C#. Na het doorspitten van de Microsoft Developer Network voor verschillende methodes om XML documenten uit te lezen, is een overzicht van verschillende methodes gevonden (MSDN, 2015). Na het bekijken van elke keuze, is gekozen om de .NET Language-Integrated Query (LINQ) to XML (System.XML.Linq) library te gebruiken. Deze keuze is gebaseerd op het feit dat het MSDN zelf aangeeft dat dit het handigste is om mee te werken.

Naast de algemene XML library is ook een XPath library nodig. Dit om verschillende XPath expressies te kunnen gebruiken, om zo het document te kunnen bevragen. Hiervoor bood de standaard XPath library van Microsoft zelf de oplossing voor. Beiden staan zeer uitgebreid gedocumenteerd op het Microsoft Developer Network.

Om als voorbeeld te dienen, worden de stappen besproken die leiden tot het maken van een XML document met alleen een Active Diagnostics log. De rest wordt weggelaten omdat de stappen hetzelfde zijn, alleen de parameters verschillen.

Om te zorgen voor de self-describing syntax en de vooraf ontworpen datastructuur, is via een reverse engineering's perspectief gewerkt. Eerst wordt een leeg document van het type *XDocument* gemaakt.

```
1. XDocument doc = new XDocument;
```

Na het maken van dit lege document wordt de rootnode toegevoegd. Dit gebeurt op dezelfde manier net als met alle andere nodes. In de datastructuur is besloten dat de rootnode *EngineReport* heet en om dit element toe te voegen wordt de *Add* methode gebruikt.

```
1. doc.Add(new XElement("EngineReport"));
```

Binnen de Active Diagnostic log zijn twee elementen beschikbaar, de Code en de Description. Tegelijkertijd word de soort log weergegeven aan de hand van een attribuut.

```
1. doc.Element("EngineReport").Add(new XElement("Log",
2.     new XElement("Code", "No Active Codes"),
3.     new XElement("Description", ""),
4.     new XAttribute("codeType",
5.         "ActiveDiagnostic")));
```

Het probleem hier echter is, dat alles nu hardcoded is. Dit zorgt voor veel te veel type werk en wordt algemeen gezien als zeer bad-practice.

Om hardcoding tegen te gaan, wordt bijgehouden hoeveel Active Diagnostic logs aanwezig zijn. Om hier achter te komen, is bedacht om een lijst bij te houden waarin alle Active Diagnostic logs komen te staan. Vervolgens itereert men door de lijst en kan men elke log apart opslaan.

```
1. for (int i = 0; i < ActiveDiagnostics.Count; i++)
```

Het is echter zo dat niet elke record in de lijst een aparte log bevat, maar dat per twee records elk een aparte log bestaat. Daarom wordt de Active Diagnostic lijst gehalveerd en wordt nog een for-loop geïntroduceerd.

```
1. for (int i = 0; i < ActiveDiagnostics.Count / 2; i++)
2. {
3.     for (int j = i * 2; j < 2 + (i * 2); j++)
4.     {
5.         LogInfo[j % 2] = ActiveDiagnostics[j];
6.     }
7. }
```

Deze geneste for-loop bouwt een array van steeds twee elementen op en uiteindelijk wordt met deze array het Active Diagnostic log toegevoegd aan het document.

```
1. doc.Element("EngineReport").Add(new XElement("Log",
2.     new XElement("Code", LogInfo[0]),
3.     new XElement("Description", LogInfo[1]),
4.     new XAttribute("codeType",
        "ActiveDiagnostic")));
```

De data binnen de Active Diagnostic lijst wordt gevonden door, met behulp van de XML.Linq library, het document te doorzoeken. Om het document klaar te maken om te doorzoeken, wordt deze omgevormd tot een werkbaar document. Dit wordt gedaan door het document te veranderen in een *XpathDocument* object. Hiervan wordt een *XPathNavigator* object gemaakt. De navigator zorgt voor de mogelijkheid om door het document heen te navigeren.

```
1. private XPathNavigator _nav;
2. private XPathDocument _currentDocument;
3.
4. _currentDocument = new XPathDocument(fileName);
5. _nav = _currentDocument.CreateNavigator();
```

Alle data in de documenten vallen onder het XML element *Block*. Om te weten te komen hoeveel *Blocks* aanwezig zijn, wordt door het document geïtereerd met de volgende XPath expressie.

```
1. strExpression = "/XMLData/Reports/Report/Data/Block";
```

Deze expressie wordt uitgevoerd door de navigator via een *.Select* methode. Een *.Count* methode geeft vervolgens de hoeveelheid *Blocks* terug.

```
1. numberOfBlocks = _nav.Select(strExpression).Count;
```

Vervolgens wordt door deze hoeveelheid *Blocks* geïtereerd en wordt tijdens het itereren gekeken of de tekst van de huidige node gelijk is aan dat waar men naar zoekt, in dit geval dus de *Active Diagnostic*. Om de tekst van een node te krijgen, is een nieuwe XPath expressie nodig.

```
1. for (int i = 0; i < numberOfBlocks; i++)
2. {
3.     strExpression = "/XMLData/Reports/Report/Data/Block[" + i +
4.         "]/TextValue/text()";
5. }
```

XPath geeft de mogelijkheid om, net als bij een array, een index te plaatsen tussen blokhaken. Zo wordt de iteratie over het aantal *Blocks* mogelijk gemaakt. Binnen XPath is ook een aantal functies beschikbaar die men kan gebruiken. De *text()* functie is hier een voorbeeld van, deze functie geeft de value in de node terug. Met de navigator wordt de expressie uitgevoerd. Omdat het expliciet gaat om een enkele node, wordt de methode *SelectSingleNode* toegepast. Deze node wordt opgeslagen in een variabele voor verdere toepassing.

```
1. var temp = navi.SelectSingleNode(strExpression);
```

Binnen de variabele zit de tekst van *Block* node, de value wordt vergeleken met dat wat we zoeken: *Active Diagnostics*. Omdat de *text()* methode een *null* retourneert bij lege nodes, worden deze eerst afgehandeld. Daarna wordt de daadwerkelijk value aangesproken met een *.Value* methode.

Omdat de value meer dan alleen de tekst *Active Diagnostics* bevat, is de *.StartsWith* methode gebruikt in plaats van de *.Equals* methode.

```
1. if (temp != null)
2. {
3.     actual = temp.Value;
4.     if (actual.StartsWith("Active Diagnostic"))
5.     {
6.         index = i;
7.         break;
8.     }
9. }
10.
11. return index;
```

Nadat de index is gevonden, wordt gestopt met zoeken en wordt dit getal geretourneerd. Nu bekend is waar de data zich bevindt in het document, wordt met deze info opnieuw een XPath expressie gemaakt. Omdat de *Block*, welke correspondeert met de zojuist geretourneerde index, alleen de titelheader van de data bevat, moet één *Block* verder worden gegaan. Dit omdat onder de titelheader zich de daadwerkelijke tabel met data bevindt. De XPath expressie wordt dan als volgt.

```
1. strExpression = "/XMLData/Reports/Report/Data/Block[" +
2.     (Index + 1) + "]/Table/Rows/Row/Value";
```

De navigator doet weer een *.Select* statement en daarmee wordt een *XPathNodeIterator* geretourneerd.

```
1. XPathNodeIterator nodes = _nav.Select(strExpression);
2. return nodes;
```

Om de data uit de tabel te halen wordt een lijst gemaakt van strings. Hier wordt van elke node de value aan toegevoegd. Vervolgens wordt doorgegaan naar de volgende node. Omdat de *XPathNodeIterator* op een *null* positie begint, moet eerst een stap worden gemaakt met de *.MoveNext* methode

```
1. List<string> StringNodes = new List<string>();
2.
3. for (int i = 0; i < nodes.Count; i++)
4. {
5.     nodes.MoveNext();
6.
7.     XPathNavigator nodesNavigator = nodes.Current;
8.
9.     XPathNodeIterator nodesText =
10.         nodesNavigator.SelectDescendants(XPathNodeType.Text,
11.                                         false);
12.
13.     StringNodes.Add(nodesText.Current.Value);
14. }
15. return StringNodes;
```

Nu de lijst is gevormd, kan deze worden gebruikt in het voorbeeld van het bouwen van het element Log in het *XDocument*. Dit is gedaan voor alle mogelijke soorten data gemodelleerd in de datastructuur.

Het beschikbaar maken van de documenten wordt gedaan via de AppData folder. Om dit proces te implementeren is de locatie van de AppData folder nodig. Deze is te verkrijgen met het volgende statement.

```
1. Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
```

Door gebruik te maken van deze folder wordt de afhandeling van laden en opslaan gestandaardiseerd over het gehele spectrum en bij elke gebruiker. Dit zorgt voor betere portability en een makkelijkere interface.

```
1. public Loader()
2. {
3.     _docLocations = Path.Combine(Environment.GetFolderPath(
4.         Environment.SpecialFolder.ApplicationData),
5.         "SL-DataHighway");
6. }
7. public void Load()
8. {
9.     if (Directory.Exists(_docLocations))
10.    {
11.        _repo = new DirectoryInfo(Path.Combine(
12.            _docLocations, "Source"));
13.        Files = _repo.GetFiles();
14.    }
15.    else
16.    {
17.        Directory.CreateDirectory(Path.Combine(
18.            _docLocations, "Source"));
19.    }
20. }
```

De code hierboven wordt runtime uitgevoerd door de applicatie. Mocht geen folder aanwezig zijn, dan wordt deze aangemaakt door de applicatie.

Andere requirements van de Application Manager waren logging en error handling. De error handling is geïmplementeerd door op verschillende belangrijke plekken de code te voorzien van try-catch clauses. Een try-catch clause geeft aan de compiler door, dat alles wat in het try-blok staat uitgevoerd moet worden en zodra een error zich voordoet het catch-blok moet worden uitgevoerd.

```
1. try
2. {
3.     ConstructDocument();
4. }
5. catch (Exception error)
6. {
7.     Console.WriteLine(error.Message);
8. }
```

Naast implementatie van error handling is gewerkt aan een logging systeem. Met het logging systeem kan worden terug gevonden waar iets fout is gegaan met de transformatie of met andere stappen in de applicatie.

Omdat elke klasse gebruikt maakt van de logger, maar het onoverzichtelijk wordt als elke klasse een dependency heeft van een logger object, is de design keuze gemaakt om de logger klasse static te maken. De gedacht om een Singleton te gebruiken was ook opgekomen, maar waarom een object maken, als je deze niet nodig hebt. Daarom de keuze voor Static.

Static houdt grofweg in dat er geen object van een klasse gemaakt hoeft te worden om gebruik te maken van de methodes ervan. Dit houdt in dat op elke mogelijke plek een log message kan worden gemaakt.

```
1. public static void Log(string message)
2. {
3.     try
4.     {
5.         _logLocation =
6.             ConfigurationManager.AppSettings["LogFile"];
7.         //Log the message + time stamp
8.
9.         using (StreamWriter writer = new
10.             StreamWriter(_logLocation), true)
11.         {
12.             writer.WriteLine(message + " " + DateTime.Now);
13.             writer.Close();
14.         }
15. }
```

Het nadeel van het gebruiken van een static klasse is dat het alleen gebruik kan maken van static objecten. Mocht het zo zijn dat de logger acteert met andere objecten, dan moeten deze objecten static zijn. Dit nadeel is gelukkig niet van toepassing op deze implementatie, omdat de logger geen gebruik maakt van andere objecten.

4.13 Datastructuur en SQL Server

Na de implementatie van de algemene functionaliteiten van de applicatie en deze goed zijn bevonden in een review, is de manier van opslaan besproken. Zoals vooraf afgesproken, moet dit in een SQL Server database gebeuren. SQL Server is een relationele database en maakt gebruik van tabellen met regels en kolommen.

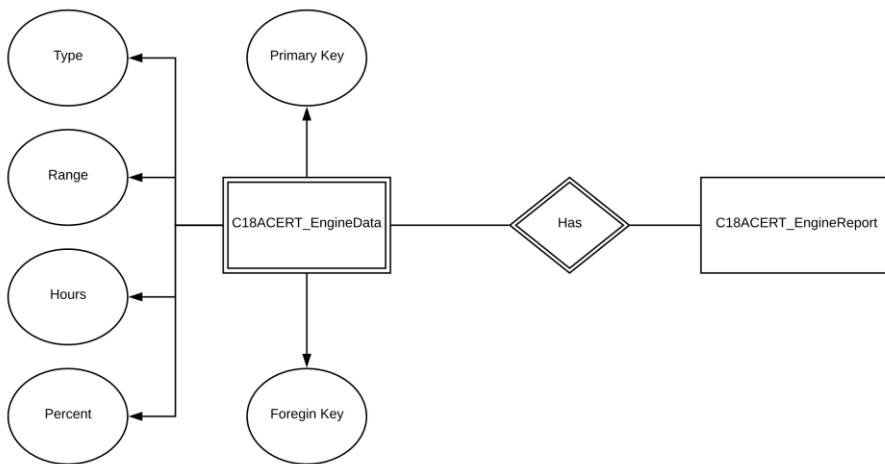
Een relationele database heeft een vaste structuur. Deze structuur moet van te voren worden ontworpen voordat de database kan worden aangemaakt. De data behoort zich ook aan deze structuur te houden. Wanneer dit niet gebeurt zullen gaten vallen in de database of zal de data ronduit worden geweigerd.

Om tot een ontwerp te komen en dit makkelijk te visualiseren, is eerst gebruik gemaakt van Microsoft Access. De user interface van Access is wat gebruiksvriendelijker en intuïtiever dan die van SQL Server. Met het gebruik van MS Access zijn de eerste ontwerpen gemaakt van de database.

Altova XMLSpy heeft een ingebouwde functie om de gemaakte XML datastructuur om te vormen naar een MS Access database. Met behulp van deze functie zijn verschillende database ontwerpen gemaakt. Samen met de Application Manager Emad Zaïd en de Data Analyst Matthijs Priester zijn deze ontwerpen besproken.

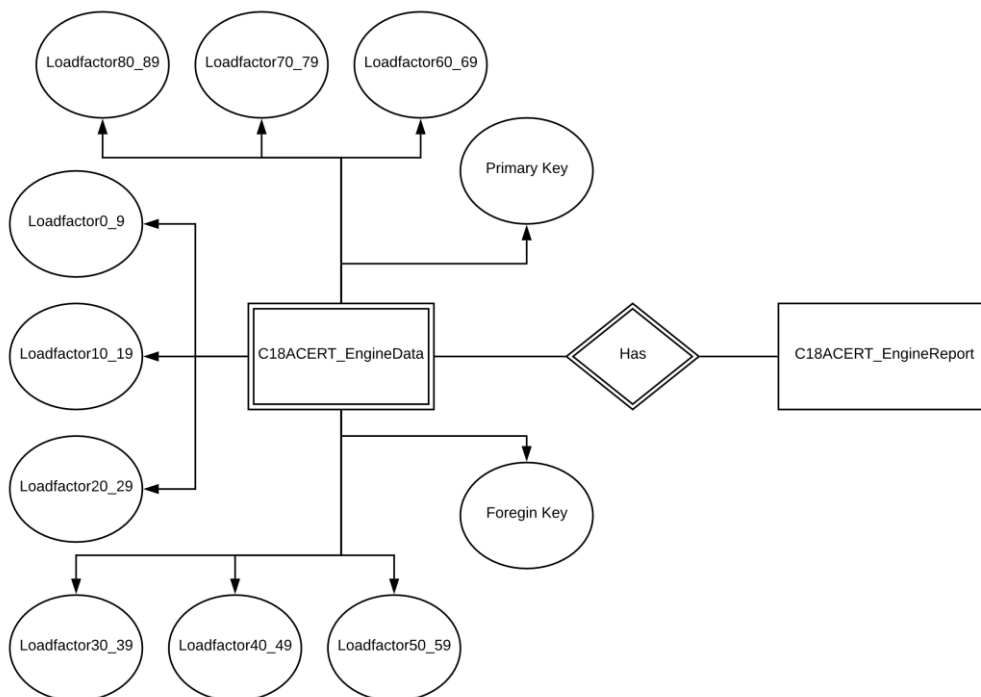
Tussen beiden ontstond wat onenigheid, omdat Emad de data zo genormaliseerd mogelijk wilde hebben, terwijl Matthijs het liefst een gedraaid ontwerp had.

Vooral op de Engine Data gaf dit interessante resultaten. Onderstaand ontwerp werd gemaakt voor de Engine Data tabel, volgens de ideeën van Matthijs.



Figuur 24 Engine Data tabel ontwerp v. Matthijs

Het ontwerp volgens de ideeën van Emad is, na het normaliseren van de data, als onderstaand gemodelleerd. Omdat door de grote hoeveelheid attributen het ontwerp niet meer leesbaar is gekozen om het niet volledig te ontwerpen.



Figuur 25 Engine Data tabel ontwerp v. Emad

Uiteindelijk is gekozen voor een genormaliseerd ontwerp welke bestaat uit zeven verschillende tabellen: Engine Report, Configuration, Monitoring, Logs, Current Totals, Engine Data en Injector Codes. Deze tabellen worden aan elkaar gerelateerd door de, bij Engine Report, aanwezige primary key. Deze primary key wordt in alle andere tabellen middels een foreign key relatie gerefereerd.

Na een aantal keer proberen inserten, is een design fout geconstateerd. De verschillende modellen van de motoren, C18ACERT en 3516B, kunnen niet onder dezelfde tabellen worden geïnsert. Dit komt omdat de data binnen de rapporten niet met elkaar overeenkomt. Vervolgens is bedacht om voor elk motormodel een aparte prefix te maken en deze bij de tabel naam te voegen. De tabellen hebben hiermee het format: 3516B_Configuration en C18ACERT_Configuration.

Dit heeft helaas het effect dat de database met elk toegevoegde motormodel in veelvoud groeit. Op de short-term is dit geen probleem, omdat aangegeven is dat nog teveel moet worden gedaan om data van andere modellen en motor fabrikanten uit te lezen. Maar op de long-term kan dit voor zowel onderhouds- als schaalbaarheidsproblemen zorgen.

Het ontwerp¹⁹ naar idee van Application Manager Emad Zaïd is uiteindelijk gebruikt om de database op te zetten. Een automatisch incrementele primary key in de EngineReport tabel is gebruikt en deze is vervolgens in de andere tabellen gerefereerd.

4.14 Derde versie Console App en nieuwe Datastructuur

Omdat on-premise tools gaan worden gebruikt voor het inladen in de database, is in een gesprek met de stakeholders gevraagd hoe de data moet worden verplaatst van schip naar wal. In dit gesprek is het onderzoek, waarin verschillende mogelijkheden van data transfer zijn belicht, aangehaald.

Tijdens het bespreken van het onderzoek is besloten om de Mail client te gebruiken om de getransformeerde documenten te verplaatsen van schip naar wal. In het onderzoek is de conclusie getrokken dat het alternatief SLDox/Resilio beter was. Maar deze mogelijkheid werd door de meeste stakeholders als onverstandig gezien. Naast het feit dat veel tijd zou worden besteed om implementatie uit te zoeken, zou teveel hulp nodig zijn van andere collega's en dit kon de afdeling zich op dit moment niet permitteren.

Bedacht is om de Mail client te integreren in de applicatie. Dit zorgt voor een gestroomlijnd proces voor de user. Omdat iedere pc gebruikt door Smit Lamnalco de Outlook client bevat, is gekozen om met de Outlook mail client te werken.

Omdat de Outlook client in de applicatie wordt geïntegreerd, is een nieuw systeem van invloed op de applicatie. Hierbij hoort een nieuwe use case, UC8²⁰. Een nieuw use case diagram²¹ is gemaakt om de invloed weer te geven.

Tijdens deze iteratie is veel aandacht gespendeerd aan het denormaliseren van de data. In de database begon de data een onoverzichtelijke massa te worden. Om dit tegen te gaan is besloten om delen van de data te denormaliseren. Zo is besloten om het overgrote deel van de tabellen terug te brengen naar de eerder besproken vorm volgens het ontwerp voor de Data Analyst Matthijs Priester. Dit had als gevolg dat meerdere records per rapport voorkomen bij zeven van de negen tabellen. Alleen de Configuration en de Current Totals tabel houden een 1 op 1 relatie met de Engine Report tabel. De andere tabellen hebben een 1 op veel relatie.

De verandering in de database had gevolgen voor de XML datastructuur. De stakeholders wilden dat de integratie tussen tussenstructuur en database zo goed mogelijk verliep. Om te zorgen dat de integratie soepel verloopt is data integriteit van belang. Om data integriteit af te handelen is samen met de Application Manager Emad Zaïd bedacht XML schema's te gebruiken.

Om deze schema's te gebruiken, moeten ze eerst worden gemaakt. Vervolgens moet elk XML document worden gevalideerd aan de hand van dit schema.

In Azure DevOps zijn de nieuwe features, Validator en Mail client, ondergebracht en zijn ze verdiept in onderlinge onderdelen. Aan de hand van het Scrumboard zijn de features gepland.

¹⁹ Bijlage 9 Entity Relationship Diagram Versie 1 blz. 104

²⁰ Bijlage 8 Use case Iteratie 3 blz. 90

²¹ Bijlage 9 Use case diagram Iteratie 3 blz. 95

4.14.1 Ontwerp

Na het maken van allerlei verschillende beslissingen, zijn aanpassing gemaakt aan het database ontwerp en aan het klasse diagram. In het klasse diagram²² zijn twee klassen toegevoegd, de MailClient klasse en de Validator klasse.

Omdat de Controller Klasse overbodig was geworden, is besloten deze te verwijderen. Alle logica is verplaatst naar de TransformMenu klasse. Deze is nu verantwoordelijk voor het maken van een Loader en een Transformer en slaat hier de brug tussen.

Omdat bij de start van de applicatie automatisch folders voor opslaan en inladen worden aangemaakt, is de SettingsMenu klasse ook overbodig geworden. Deze is daarom verwijderd uit het ontwerp.

In het databaseontwerp²³ zijn ook een aantal veranderingen doorgevoerd. Omdat simpelweg teveel attributen aanwezig zijn om te modelleren, is besloten om dit niet te doen. Wel zijn de relaties geüpdatet, zoals de 1-op-1 relatie tussen EngineReport en CurrentTotals en tussen EngineReport en Configuration. Om uniciteit af te handelen is een GUID in het database ontwerp geïntroduceerd.

Met de bijgekomen Use cases zijn nieuwe Sequence²⁴ diagrammen gemaakt.

4.14.2 Implementatie

Als eerste zijn de schema's voor de documenten geïmplementeerd. Dit tezamen met een validator om te zorgen dat de documenten de structuur in de schema's aanhouden. Daarna zijn aan de hand van deze schema's processen opgebouwd in Microsoft MessageQueue en Tibco Scribe om te zorgen dat de data wordt opgeslagen in de SQL Server database.

Om te zorgen voor integratie met andere datasets is een extra ID toegevoegd aan de bestanden. Om uniciteit van de documenten af te handelen is een GUID opgenomen in de XML documenten en database.

4.14.2.1 XSD's, Schema's voor XML

XML Schema's zijn een template voor een XML document. Ze geven aan welke elementen wel en niet in een XML document mogen voor komen, hoe vaak de elementen mogen voor komen en waar de elementen mogen voor komen. Voor beide motormodellen, de 3516B en de C18ACERT, is een XML schema gemaakt.

Een XML schema wordt geschreven in de XML Schema Definition Language (XSD) en gebruikt andere vormen van self-describing syntax. Een zogeheten namespace wordt gebruikt om aan te geven waar extra informatie kan worden gevonden over de gebruikte elementen. Naast aangeven waar meer informatie kan worden gevonden, is de voornaamste reden van een namespace het zorgen voor conflict vermindering. In sommige XML documenten kan naar meerdere dezelfde element worden verwezen. Om dan aan te geven over welk element het gaat, wordt de namespace toegevoegd als prefix voor deze elementen.

²² Bijlage 9 Klasse diagram Iteratie 3 blz. 101

²³ Bijlage 9 Entity Relationship Diagram Versie 2 blz. 105

²⁴ Bijlage 9 Sequence diagrammen Iteratie 3 blz. 115-116

In het geval van een XML Schema wordt gerefereerd naar de World Wide Web(W3) webpagina over XML schema's. Deze namespace wordt gekoppeld aan de prefix 'xs' en deze ziet men terug in alle tags van het schema.

Om aan te geven hoe de XML documenten zich moeten gedragen, worden in het XSD op verschillende manieren restricties opgelegd. In dit voorbeeld is aangegeven hoe het element *ActiveDiagnostic* zich moet gedragen in een XML document.

```
1. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
2.   <xs:element name="ActiveDiagnostic">
3.     <xs:complexType>
4.       <xs:sequence>
5.         <xs:element ref="Code" />
6.         <xs:element ref="Description" />
7.       </xs:sequence>
8.     </xs:complexType>
9.   </xs:element>
10.  <xs:element name="Code" type="xs:string" />
11.  <xs:element name="Description" type="xs:string" />
```

Eerst wordt aangegeven dat een element met naam *ActiveDiagnostic* bestaat. Dit element wordt het type *complexType* toegewezen. Een *complexType* geeft weer dat het element zelf meerdere elementen of/en attributen bevat. Vervolgens wordt aangegeven dat zich een *sequence* voordoet binnen het element. De *sequence* bevat zelf twee elementen met naam *Code* en *Description*. In plaats van een naam wordt een referentie gegeven. In dit geval wordt met de *ref* verwezen naar de twee onderstaande elementen van het type *xs:string*. Het refereren naar andere elementen zorgt voor een overzichtelijker document en makkelijker op te zoeken elementen. Het aangegeven type zorgt voor een extra laag aan data integriteit op het element niveau, waar het schema zelf op document niveau zorgt voor data integriteit.

4.14.2.2 Validatie van de schema's

Om deze schema's²⁵ te implementeren is de *Validator* klasse geïmplementeerd, deze *Validator* klasse maakt ook gebruik van System.XML library. Om een XML document te valideren zijn twee onderdelen nodig, het te valideren document en het schema om tegen te valideren.

Met een switch-statement op het *engineModel* wordt gedicteerd welk schema het document moet aanhouden. Omdat de schemalocatie een prefix nodig heeft van XMLSchema-instance (xsi), is het nodig om deze ook toe te voegen aan het document.

```
1. private XNamespace _xsi = XNamespace.Get(
2.     "http://www.w3.org/2001/XMLSchema-instance");
3.
4. switch (engineModel)
5. {
6.     case "3516B":
7.         _schemaPath = Path.Combine(Path.GetDirectoryPath(
8.             Assembly.GetExecutingAssembly().Location),
9.             @"SL_Schemas\SL_3516B_Schema.xsd");
10.        _xsd = XNamespace.Get(_schemaPath);
11.        document.Root.Add(new XAttribute(XNamespace.Xmlns +
12.            "xsi", _xsi.NamespaceName),
13.            new XAttribute(_xsi +
14.                "noNamespaceSchemaLocation", _xsd));
```

²⁵ Bijlage 12 XML schema's blz. 134-144

Het schema wordt aan een *schemaSet* toegevoegd. Deze set wordt vervolgens gebruikt om het document te valideren. De *.Validate* functie is een ingebouwde functie, onderdeel van de *XDocument* klasse van de *System.XML.linq* library. Mocht een probleem zich voor doen, zal de message van de error op het scherm worden weergegeven en zal de validatie mislukken.

```
1. _schemaSet.Add("", _schemaPath);
2. document.Validate(_schemaSet, (o, e) =>
3. {
4.     Console.WriteLine("{0}", e.Message);
5.     _validated = false;
6. });
7.
8. return _validated;
```

4.14.2.3 Integratie met andere datasets

Halverwege de iteratie is gevraagd om integratie met andere datasets mogelijk te maken. Samen met de Data Analyst Matthijs Priester is gekeken naar andere datasets en de identifiers daarbinnen. De dataset van Marad5, de Maintenance Job database, en LOAMS, de Lube Oil dataset, gebruiken beiden de *EffectsID* als identifier. Elk schip bezit een *EffectsID* en dit zorgt voor een goede identifier om integratie mogelijk te maken.

Samen met Matthijs is een XML document, *Vessels.xml*, gemaakt. Hier komt elk schip in voor met naam, *EffectsID* en de regio waar het schip opereert.

```
1. <region name="Africa">
2.   <vessel>
3.     <name>SL AFRICA</name>
4.     <ID>122</ID>
5.   </vessel>
6. </region>
```

Om te zorgen voor het toevoegen van het *EffectsID* aan het XML document, is een methode geschreven. Deze methode geeft in de console de regio's vanuit *Vessels.xml* weer. De user geeft vervolgens aan in welke regio de user opereert. Dit vermindert het aantal te kiezen schepen van 180 naar gemiddeld 25. Vervolgens kiest de user het schip en wordt het bijbehorende *EffectsID* opgeslagen.

Dit betekent wel dat een vorm van human error is geïntroduceerd in de applicatie. Om een foutieve keuze tegen te gaan, wordt een keer extra gevraagd aan de user of dit daadwerkelijk het schip is waar de user op opereert. In gesprek met de stakeholders is besloten deze vorm van human error te accepteren.

4.14.2.4 Mailsysteem

Op internet is gezocht welke mogelijkheden beschikbaar zijn om de Outlook Mail client te implementeren. Na wat deskresearch op het Microsoft Developer Network en enkele andere websites. Is de *Microsoft.Office.Interop.Outlook* library gevonden.

Eerst is ontwerpen hoe de Outlook client door de user moet worden gebruikt. Na het transformeren van de documenten zullen alle getransformeerde documenten per mail worden verstuurd. Om human error te elimineren, is bedacht om een vooraf ingevulde mail te genereren, deze te presenteren aan de user en uiteindelijk wel door de user te laten versturen.

Vanuit de AppData folder worden alle getransformeerde documenten in een array gestopt.

```
1. DirectoryInfo targetLocation = new DirectoryInfo(  
2.     Path.Combine(locations, "Target"));  
3.  
4. FileInfo[] attachmentFiles = targetLocation.GetFiles();
```

Om te zorgen dat, al verzonden, documenten niet bij een volgende mail weer worden verzonden, worden deze verplaatst naar een folder met datum van verzending. Dit gebeurt nadat ze aan de mail zijn toegevoegd.

```
1. string mailBatch = targetLocation.FullName + @"\MailBatch_" +  
2.     DateTime.Now.ToString("dd-MM-yyyy");  
3. Directory.CreateDirectory(mailBatch);  
4.  
5. foreach (var attachmentFile in attachmentFiles)  
6. {  
7.     oMailItem.Attachments.Add(attachmentFile.FullName);  
8.     File.Move(attachmentFile.FullName, mailBatch +  
9.         @"\\" + attachmentFile.Name);  
10. }
```

In de werkelijkheid heeft de Outlook client voor veel problemen gezorgd. De problemen kwamen vooral voor als de user al een Outlook proces open had staan en bestond een bug tussen de connectie met de applicatie en de Outlook server. De open Outlook proces problemen zijn opgelost door, net als bij de open folders bij de start van de applicatie, alle processen met de naam Outlook te elimineren.

Na veel testen van het programma is geconstateerd dat het niet verstandig is om een al bestaand proces te elimineren. In verband met usability van de applicatie is een aanpassing gemaakt. In plaats van een bestaand proces te elimineren en een nieuw proces aan te maken, wordt een openstaande proces gebruikt. Pas wanneer deze niet aanwezig is, wordt een nieuwe aangemaakt.

```
1. Outlook.Application outlookApp;  
2.  
3. try  
4. {  
5.     outlookApp = (Outlook.Application)Marshal.GetActiveObject(  
6.         "Outlook.Application");  
7. }  
8. catch (System.Exception e)  
9. {  
10.     outlookApp = new Outlook.Application();  
11. }
```

Om bij dit bestaande proces te komen wordt de *Marshal* klasse, onderdeel van de Office.InteropServices library, gebruikt. De *Marshal* klasse geeft toegang tot een collectie van methodes voor het afhandelen van niet gemanagede code. Deze *Marshal* klasse geeft de mogelijkheid om te zoeken naar een actief Outlook object en deze te retourneren aan de applicatie.

Hiermee was het probleem van meerdere Outlook processen verholpen. De synchronisatie problemen deden zich echter nog steeds voor. Deze problemen ontstonden wanneer geen aanspraak meer werd gemaakt op een Outlook object.

Zodra geen aanspraak meer op een object wordt gemaakt, wordt deze door de garbage collector van Visual Studio weggegooid. Tijdens het weggooien wordt het Outlook proces afgesloten en dit zorgt



voor te weinig tijd om te synchroniseren met de Outlook server. Dit had vervolgens het effect dat alle te versturen emails bleven hangen in de Outlook Outbox of dat de mail helemaal niet werd opgebouwd.

Figuur 26 Outlook RPC-server error

De oplossing voor dit probleem is gemaakt, door de Outlook applicatie daadwerkelijk te openen. Het ontwerp was eerst om het verzenden van de emails op de achtergrond te doen. Dit gaf de synchronisatie problemen.

```
1. Namespace ns = null;
2.
3. ns = outlookApp.GetNamespace("MAPI");
4. Folder inbox = (Folder)ns.GetDefaultFolder(
5.     OlDefaultFolders.olFolderInbox);
6. Explorer ex = inbox.GetExplorer();
7.
8. ex.Display();
```

Om dit te implementeren wordt een *Namespace* geopend vanuit de bestaande Outlook applicatie. Hiervan wordt de Inbox folder aangeroepen. Deze wordt voor de user gepresenteerd met de *.Display* methode. Na deze te hebben gepresenteerd, wordt een extra synchronisatie methode aangeroepen. Dit om zeker te zijn dat de emails daadwerkelijk verzonden worden en niet toch in de Outbox blijven hangen.

```
1. ns = application.GetNamespace("MAPI");
2.
3. foreach (SyncObject syncObj in ns.SyncObjects)
4. {
5.     if (syncObj != null)
6.     {
7.         syncObj.Start();
8.         Marshal.ReleaseComObject(syncObj);
9.     }
10. }
```

In deze methode wordt dezelfde *Namespace* aangeroepen vanuit de applicatie. Hiervan worden verschillende Send/Recieve groepen, in de vorm van *SyncObjects*, opgevraagd. Door de *SyncObjects* wordt geïtereerd en met *.Start* wordt aangegeven dat het Send and Recieve proces moet worden gestart. Het *syncObject* wordt vervolgens losgelaten, zodat het door ander objecten kan worden gebruikt.

4.15 MS Message Queue en Scribe Insight

Na alle problemen te hebben opgelost met de Outlook client en deze in een review te hebben besproken, is het proces van opslaan in de SQL Server database behandeld.

Om de laatste stap te automatiseren is met de Application Manager Emad Zaïd een pipeline ontworpen van email attachment tot database tabel. De eerste stap in deze pipeline is het automatische scrapen van email attachments. Dit is gedaan door middel van het implementeren van een Scribe Integration Proces met een Microsoft Message Queue (MSMQ). In de queue kan worden aangegeven wat de queue triggert en wat vervolgens moet gebeuren.

In MSMQ wordt aangegeven waar deze op acteert, in dit geval een mailserver. De connectie is aan de hand van een server IP-adres, inloggegevens en een emailpoort aangegeven.

Step 4 Publisher/Bridge Properties

Connection Settings

Email Server: 10.31.2.101

Account: ScribeQueue

Password: xxxxxxxx

Mail Port: 110 Connection Timeout (seconds): 60

Authentication Mode

☒ Simple

☐ APOP ☐ Auto Degradation

Message Settings

Extract XML from

☒ Email Attachment

☐ Email Body

Set Message Label using

☐ Attachment Name

☐ Email Subject

☒ Other text

SLEngineReport

Step 5 Activation

Figuur 27 Eerste Queue

Aangegeven wordt waar de XML zich bevindt en voor logging doeleinden wordt een Message Label toegevoegd.

Step 3 Event Settings

Queue Event Settings

Message Identification

The combinations of these four values uniquely identify XML messages for processing by this Integration Process.

Message label *

Scribe label *

XML root node SLENG3516B

XML schema *

* = match any value

Open Browse

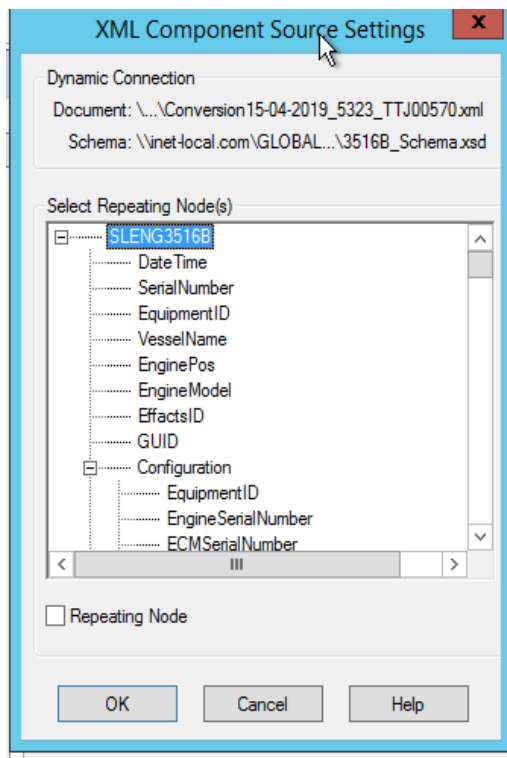
Figuur 28 Filteren op rootnode

Met bovenstaande queue wordt aangegeven dat alle mogelijke XML email attachments moeten worden meegenomen.

Omdat er twee verschillende modellen zijn en deze in verschillende processen moeten worden verwerkt, wordt op motormodel gefilterd.

Dit wordt gedaan door een value aan de Event Settings XML root node te erkennen. Op de value, in dit geval SLENG3516B, wordt vervolgens gefiltreerd.

Na het inrichten van een Integration Proces met een Microsoft MessageQueue, is de volgende stap het juist toepassen van een Data Translation Specification (DTS) in Scribe Insight.

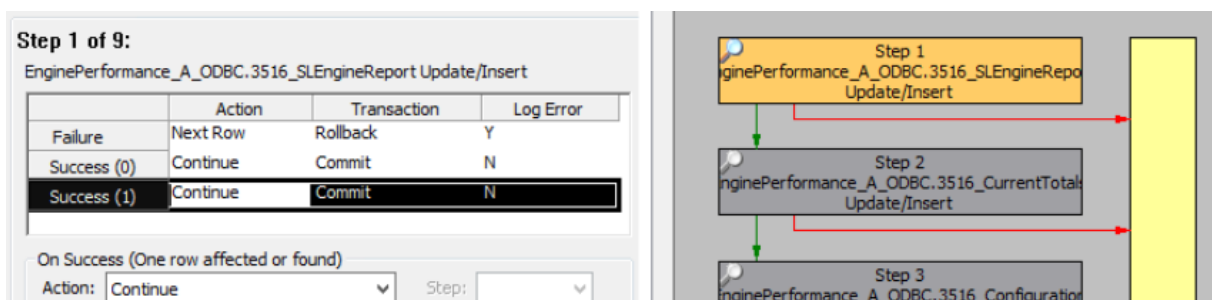


Door het toewijzen van een XML Schema aan een DTS als *Source*, wordt een overzicht gecreëerd van elementen gevonden in het schema.

Wanneer zich herhalende XML elementen voordoen, kan men dit aangeven in dit overzicht. Het programma zal dan het aantal herhalende elementen tellen en voor elke een keer herhalen.

Na het aangeven van het schema, geeft men de database als *Target* aan.

Figuur 29 Schema toewijzen aan de DTS



Figuur 30 Database als Target

Omdat het automatiseren van de relaties in de database moeilijk te realiseren was, is gekozen om een algemene Message ID te maken in de vorm van een GUID. Dit GUID wordt in de applicatie gegenereerd en zorgt voor uniciteit binnen de database.

```
1. doc.Element(rootNode).Add(new XElement("GUID", Guid.NewGuid()));
```

GUID staat voor Globally Unique Identifier en de kans op duplicerende data is 1 op de 2^{128} ($34 \cdot 10^{36}$). Deze GUID wordt gebruikt als primary key in de database.

Naast het GUID is nog een aanpassing gemaakt op de XML datastructuur. Na een aantal inserts, is geconstateerd dat sommige onderdelen van de data herhaaldelijk worden geïnsert. Dit had te maken met de repeating nodes, aangegeven aan het begin van het maken van een DTS. Om dit tegen te gaan worden alle herhalende elementen binnen de schema's voorzien van een ID, van 1 tot n. Daarnaast zijn aparte queue's gemaakt per herhalend element.

Figuur 31 Opnieuw de queue in

Zoals hierboven te zien is wordt, na succesvol processen, de message opnieuw de queue in gestuurd. Dit wordt gedaan met een ander label, 3516BMONITORING. Bij het volgende proces wordt gefilterd op dit label en in het uit te voeren DTS wordt aangegeven dat, in dit geval, de Monitoring header het herhalende element is.

Uiteindelijk wordt in elke DTS, na het bepalen van source en target, aangegeven hoe de mapping moet worden uitgevoerd.

Link	Ref	Field Name	Type/Length
✓	S2	DateTime	<SLENG3516B> datetime
✓	S3	SerialNumber	<SLENG3516B> nchar
✓	S4	EquipmentID	<SLENG3516B> nchar
✓	S5	VesselName	<SLENG3516B> nchar
✓	S6	EnginePos	<SLENG3516B> nchar
✓	S7	EngineModel	<SLENG3516B> nchar
✓	S8	EffectsID	<SLENG3516B> smallint
✓	S9	GUID	<SLENG3516B> nchar
✓	S11	EquipmentID	<SLENG3516B, Config> nchar
✓	S12	EngineSerialNumber	<SLENG3516B, Cc> nchar
✓	S13	ECMSerialNumber	<SLENG3516B, Conf> nchar
✓	S14	PersonalityModulePartNumber	<SLENG3516B, Conf> nchar
✓	S15	PersonalityModuleReleaseDate	<SLENG3516B, Conf> nchar
✓	S16	RatingNumber	<SLENG3516B, Config> tinyint
✓	S17	RatedPower	<SLENG3516B, Config> nchar
✓	S18	RatedPeakTorque	<SLENG3516B, Conf> nchar
✓	S19	RatedEngineSpeed	<SLENG3516B, Conf> nchar
✓	S20	EngineRotation	<SLENG3516B, Config> nchar
✓	S21	EngineLocation	<SLENG3516B, Config> nchar
✓	S22	FuelRatioControlOffset	<SLENG3516B, Conf> tinyint
✓	S23	RatedFuelPosition	<SLENG3516B, Conf> nchar
✓	S24	LowIdleSpeed	<SLENG3516B, Config> nchar

Link	Index	Step Name	Step Field	Type/Length
✓		EnginePerformance	EffectsID	nchar(255)
✓		EnginePerformance	SerialNumber	nchar(255)
✓		EnginePerformance	EquipmentID	nchar(255)
✓		EnginePerformance	VesselName	nchar(255)
✓		EnginePerformance	EngineType	nchar(255)
✓		EnginePerformance	EngineModel	nchar(255)
✓		EnginePerformance	EnginePos	nchar(255)
✓		EnginePerformance	DateTime	datetime(8)
✓		EnginePerformance	GUID	char(16)
✓		EnginePerformance	modifiedOn	datetime(8)
✓		EnginePerformance	TotalTime	nchar(255)
✓		EnginePerformance	TotalFuel	nchar(255)
✓		EnginePerformance	TotalIdleTime	nchar(255)
✓		EnginePerformance	TotalIdleFuel	nchar(255)
✓		EnginePerformance	TotalMaxFuel	nchar(255)
✓		EnginePerformance	LifetimeTotalEngineRevolutions	nchar(255)
✓		EnginePerformance	PercentageIdleTime	nchar(255)
✓		EnginePerformance	TotalOperatingHours	nchar(255)
✓		EnginePerformance	AverageRPM	nchar(255)
✓		EnginePerformance	AverageFuelRate	nchar(255)
✓		EnginePerformance	EngineStarts	nchar(255)
✓		EnginePerformance	OverallLoadFactor	nchar(255)

Figuur 32 Mapping van XML links en Database rechts

In de mapping worden twee datalinks gedefinieerd, LookUp en Data. LookUp wordt gebruikt voor matching voor alle zoek, update en delete stappen. Data wordt gebruikt voor het koppelen van Source naar Target velden.

Na integratie met Scribe is een bug gevonden in de transformatie van de documenten en hoe Scribe met de aangeleverde documenten omgaat. Tijdens het testen van de pipeline is gebruik gemaakt van, met Altova XMLSpy, bewerkte documenten. Deze documenten kwamen altijd door de pipeline heen. Het moment dat werd gewerkt met, door de applicatie gegenereerde en niet met Altova XMLSpy bewerkte, documenten, ging de pipeline kapot.

Na veel onderzoek op internet en het vergelijken van de documenten, is uiteindelijk het probleem gevonden. Samen met de Application Manager Emad Zaïd is naar de hexadecimale values van een bewerk document en van een niet bewerk document gekeken. Het blijkt dat wanneer een *.Save* methode wordt aangeroepen op een *XDocument* object, dit document automatisch een UTF-8 Byte Order Mark (BOM) krijgt toegevoegd. Wanneer de documenten vervolgens worden bewerkt met Altova XMLSpy dwingt deze de UTF-8 encoding zonder BOM af.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EF	BB	BF	3C	3F	78	6D	6C	20	76	65	72	73	69	6F	□	□	□	<	?	x	m	l		v	e	r	s	i	o
6E	3D	22	31	2E	30	22	20	65	6E	63	6F	64	69	6E	n	=	"	1	.	0	"		e	n	c	o	d	i	n
67	3D	22	75	74	66	2D	38	22	3F	3E	0D	0A	3C	53	g	=	"	u	t	f	-	8	"	?	>			<	S
4C	45	4E	47	43	31	38	41	43	45	52	54	3E	0D	0A	L	E	N	G	C	1	8	A	C	E	R	T	>		
20	20	3C	44	61	74	65	54	69	6D	65	3E	32	30	31			<	D	a	t	e	T	i	m	e	>	2	0	1
39	2D	30	33	2D	30	35	54	31	30	3A	30	30	3A	30	9	-	0	3	-	0	5	T	1	0	:	0	0	:	0
30	3C	2F	44	61	74	65	54	69	6D	65	3E	0D	0A	20	0	<	/	D	a	t	e	T	i	m	e	>			

Figuur 33 Document met BOM op 1, 2 en 3

Dit hield in dat Scribe de UTF-8 encoding niet meer herkende en compleet dichtsloeg wanneer het documenten de BOM toevoeging van **EF BB BF** kreeg. Om dit op te lossen is de methode van opslaan veranderd.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3C	3F	78	6D	6C	20	76	65	72	73	69	6F	6E	3D	22	<	?	x	m	l		v	e	r	s	i	o	n	=	"
31	2E	30	22	20	65	6E	63	6F	64	69	6E	67	3D	22	1	.	0	"		e	n	c	o	d	i	n	g	=	"
75	74	66	2D	38	22	3F	3E	0D	0A	3C	53	4C	45	4E	u	t	f	-	8	"	?	>		<	S	L	E	N	
47	43	31	38	41	43	45	52	54	3E	0D	0A	20	20	3C	G	C	1	8	A	C	E	R	T	>				<	
44	61	74	65	54	69	6D	65	3E	32	30	31	39	2D	30	D	a	t	e	T	i	m	e	>	2	0	1	9	-	0
33	2D	30	35	54	31	30	3A	30	30	3A	30	30	3C	2F	3	-	0	5	T	1	0	:	0	0	:	0	0	<	/
44	61	74	65	54	69	6D	65	3E	0D	0A	20	20	3C	53	D	a	t	e	T	i	m	e	>				<	S	

Figuur 34 Document zonder BOM op 1, 2 en 3

Een methode is gemaakt om te zorgen dat het document alleen een UTF-8 encoding toegevoegd krijgt en geen BOM krijgt toegewezen. In plaats van alleen de *.Save* methode aan te roepen, wordt gebruik gemaakt van een *StreamWriter* en een *FileStream*. De *StreamWriter* implementeert de *TextWriter* klasse maar kan een bepaalde encoding afdwingen. Een *FileStream* object wordt gebruikt voor alle acties ten aanzien van file handling.

```

1. using (var stream = File.Create(savingFileName))
2. {
3.     WriteXmlToStream(stream, doc);
4. }
5.
6. private void WriteXmlToStream(FileStream stream, XDocument document)
7. {
8.     using (var writer = new StreamWriter(stream,
9.                                     new UTF8Encoding(false)))
10.    {
11.        document.Save(writer);
12.    }
13. }

```

Om zeker te zijn dat bij het opslaan van documenten vanuit de applicatie geen problemen meer voor komen, wordt aan de *StreamWriter* een nieuwe *UTF8Encoding* meegegeven met de *UTF8Identifier* op *false*. Wanneer deze *true* meekrijgt zal altijd een Byte Order Mark aan het document worden meegegeven.

Na het oplossen van de problemen zijn custom queries geschreven voor de Data Analist Matthijs Priester. Deze queries²⁶ zijn samen met hem gemaakt, om te zorgen dat het makkelijker wordt om analyses uit te voeren.

De integratie met Scribe is hiermee hersteld en alle problemen zijn verholpen. Na evaluatie is door de stakeholder gevraagd om de applicatie om te vormen met een Graphical User Interface (GUI), een flinke toevoeging aan de requirements. In deze evaluatie is verteld dat dit tijd gaat kosten. In overleg is de keuze gemaakt om het maken van visualisaties voor de data achterwege te laten en dit in te ruilen voor het maken van een GUI.

4.16 Vierde en final versie: WinForm Applicatie

Om van console naar grafisch te gaan, geeft Visual Studio twee verschillende mogelijkheden. Dit zijn een Windows Forms (WinForms) applicatie en een Windows Presentation Foundation (WPF) applicatie. Winforms is een ouder desktop applicatie format en WPF een wat nieuwer format. Waar WinForms gebruik maakt van een drag en drop design van de GUI, maakt WPF gebruik van de taal eXtensible Application Markup Language (XAML) voor het design. UI elementen bij WinForms zijn minder scalable dan bij WPF en de performance van WPF, in vergelijking met WinForms, is een stuk beter (EDUCBA, 2019).

Uiteindelijk zijn beide mogelijkheden bekeken en geprobeerd, om uiteindelijk tot de conclusie te komen dat een nieuwe taal leren, XAML, niet haalbaar is binnen het resterende tijdsbestek. Daarnaast is het in de scope van het project niet nodig om gebruik te maken van betere features, zoals de betere scalability en performance, van WPF.

Met de shift naar grafisch zijn extra features gekomen. Deze zijn verdiept in Azure DevOps en op het Scrumboard gepland.

4.16.1 Nieuwe requirements, nieuwe Use cases

Een paar nieuwe requirements waren

- dat men de schepen kon selecteren, net zoals men dat kon in de console applicatie,
- dat men niet zomaar door kon gaan voordat aan eerdere stappen werd voldaan en
- dat een search bar aanwezig was om het makkelijker te maken voor de user om een schip te kiezen.

Met de nieuwe requirements zijn nieuwe Use cases²⁷ opgesteld. Deze zijn opgenomen in een vernieuwd Use case diagram²⁸.

²⁶ Bijlage 13 SQL Queries blz. 144-165

²⁷ Bijlage 8 Use cases Iteratie 4 blz. 90-92

²⁸ Bijlage 9 Use case diagram Iteratie 4 blz. 96

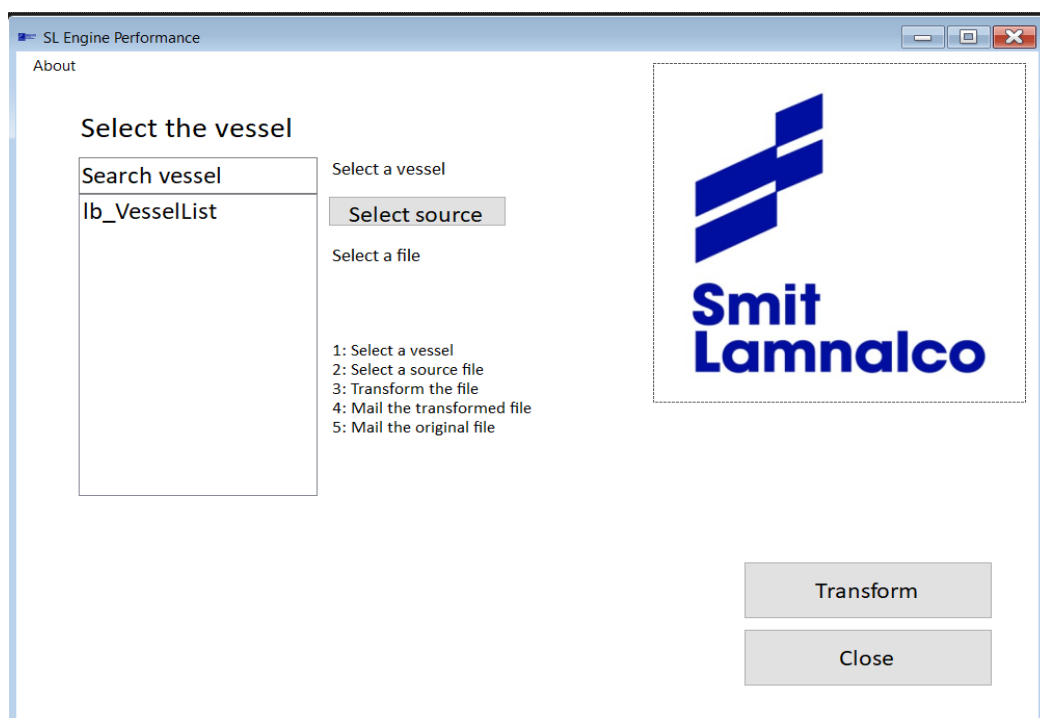
4.16.2 Ontwerp

Door de ommezwaai in het project is in het Klasse diagram²⁹ de Forms klasse geïntroduceerd waar GUI's op worden gebaseerd en van de nieuwe en veranderde Use cases zijn Sequence diagrammen³⁰ gemaakt.

4.16.3 UI design met Wireframes

Met LucidChart zijn Wireframes³¹ gemaakt om te kunnen laten zien wat ontworpen is, qua UI design. Deze zijn gepresenteerd en goed bevonden na hier en daar een aanpassing. Eén van de aanpassingen besloeg het maken van een klein stappenplan, met wat een user moet doen om tot het gewenste resultaat te komen.

Na het maken van de designs, is in Visual Studio de Winform Applicatie gemaakt met het gewenste design.



Figuur 35 Uitkomst WinForms design

4.16.4 Implementatie

De requirements vroegen om restricties op de uit te voeren stappen. Deze zijn op de verschillende labels en buttons gelegd. Zo zijn de 'Select a vessel' label, de 'Select a file' label, de 'Select source' button en de 'Transform' button pas te zien nadat voorgaande stappen zijn voltooid. 'Select a vessel' geeft, wanneer in de 'lb_VesselList' een schip geselecteerd is, de naam van het schip weer. Zodra dit is gedaan, wordt de button 'Select source' beschikbaar. Het geselecteerde document wordt met

²⁹ Bijlage 9 Klasse diagram Iteratie 4 blz. 102

³⁰ Bijlage 9 Sequence diagrammen Iteratie 4 blz. 117-120

³¹ Bijlage 9 Wireframes blz. 121

volledige naam weergegeven in de label 'Select a file'. Zodra de source is geselecteerd, wordt de button 'Transform' actief en kan de user hier op klikken.

De lijst met schepen wordt, net als bij de Console Applicatie, gegenereerd vanuit de eerder gemaakte config file Vessels.xml. Het enige wat veranderd is, is dat in plaats van de mogelijkheid om het aantal schepen te verminderen door per regio te filteren, het nu mogelijk is om direct naar het desbetreffende schip te zoeken met een search bar.

Deze search bar is gerealiseerd door een *TextChanged* event te implementeren. De volledige lijst van schepen wordt doorzocht, om te kijken of de letters die in de searchbar zijn getypt in de naam van het schip zitten. Dit betekent echter niet dat per letter wordt gekeken. Het volledige 'woord' wat getypt is in de search bar, wordt in behandeling genomen. Wanneer een user bijvoorbeeld 'abc' in typt, wordt gekeken of een schip in de lijst voorkomt waarin 'abc' voorkomt, niet of een 'a', 'b' of 'c' in het schip voorkomt.

```
1. private void tb_VesselSearch_TextChanged(object sender,
2.                                     EventArgs e)
3. {
4.     var listItems = new List<string>(_vesselCollection);
5.
6.     lb_VesselList.Items.Clear();
7.
8.     foreach (string item in listItems)
9.     {
10.         if(item.ToUpper().Contains(tb_VesselSearch.Text.ToUpper()))
11.         {
12.             lb_VesselList.Items.Add(item);
13.         }
14.     }
15. }
```

Om zeker te zijn dat alles goed verloopt, wordt alle input eerst omgevormd tot hoofdletters met een *.ToUpper* methode.

4.16.5 Refactoring en Strategy Pattern

Na het houden van een codereview zijn wat problemen binnen de code aangepakt. Zo is, in het nieuwe ontwerp³², de Validator klasse herschreven naar één gebaseerd op het Strategy Pattern en zijn aanpassingen in de Transformer klasse doorgevoerd.

Het Strategy Pattern is één van de Behavioral Design Patterns. Het pattern wordt gedefinieerd als: het gebruik van een familie van algoritmes, elk in een eigen klasse en hun objecten verwisselbaar. De kern van het Strategy Pattern is de verwisselbaarheid van de verschillende strategieën, geïmplementeerd in hun eigen klasse. Dit houdt in dat een algemene interface moet worden gemodelleerd, waaraan elke strategie zijn functionaliteit af leent.

³² Bijlage 9 Klasse diagram Iteratie 5 blz. 103

Implementatie is gerealiseerd met een interface *IVValidationStrategy*.

```
1. public interface IValidationStrategy
2. {
3.     bool ExecuteStrategy(XDocument doc);
4. }
```

Van deze interface worden concrete strategieën gemaakt aan de hand van het motormodel.

```
1. public class _3516BStrategy : IValidationStrategy
```

Bij gebruik wordt in de *Transformer* klasse een *Validator* object aangemaakt. Deze bepaald aan de hand van *EngineModel* de strategy.

```
1. Validator validator = new Validator();
2. validator.SetValidationStrategy(EngineModel, savingFileName);
3. bool validated = validator.ExecuteValidationStrategy(doc);
```

De *Validator* is nu niet meer officieel verantwoordelijk voor de validatie, maar alleen voor het selecteren van de *_validationStrategy*.

```
1. public void SetValidationStrategy(string engineModel, string docName)
2. {
3.     if (engineModel.Equals("C18ACERT"))
4.     {
5.         _validationStrategy = new C18ACERTStrategy();
6.     }
7.     else if (engineModel.Equals("3516B"))
8.     {
9.         _validationStrategy = new _3516BStrategy();
10.    }
11. }
```

Voor het uitvoeren van de *_validationStrategy* hebben beide strategieën hun eigen implementatie van de *.ExecuteStrategy()* methode. Dit houdt in dat runtime een strategy kan worden toegewezen, wat zorgt voor extra flexibiliteit van de applicatie.

```
1. public bool ExecuteValidationStrategy(XDocument document)
2. {
3.     _validated = _validationStrategy.ExecuteStrategy(document);
4.
5.     return _validated;
6. }
```

Naast het implementeren van het Strategy Pattern is de refactor methode Extract Method toegepast op een deel van de *Transformer* klasse. Deze actie heeft tot een vermindering van code geleid, wat op zijn beurt weer leidt tot verhoogde leesbaarheid, maintainability en usability van de code. Waar voorheen een aantal methodes nodig was om data te verkrijgen die onderstaande vorm volgde:

```
1. private List<string> GetActiveDiagnostics(string serialNum)
2. {
3.     List<string> ActiveDiagnostics = new List<string>();
4.
5.     ActiveDiagnostics =
6.         GetNodes(GetDataValues(GetNumberOfBlocks(),
7.                                 "Active Diagnostic", serialNum));
8.
9.     return ActiveDiagnostics;
10. }
```

Is nu een algemene *GetData()* methode gemaakt, waarin gespecificeerd kan worden, aan de hand van *dataType*, welke data moet worden verkregen.

```
1. private List<string> GetData(string serialNum, string dataType)
2. {
3.     List<string> dataList = new List<string>();
4.
5.     dataList = GetNodes(GetDataValues(GetNumberOfBlocks(),
6.                                         dataType, serialNum));
7.
8.     return dataList;
9. }
```

4.16.6 Overdracht en afsluiting development

Na het omvormen van de applicatie, is de development cyclus afgesloten met een laatste review van de gemaakte applicatie. De applicatie is door alle stakeholders goed bevonden en de Azure Repository, waar de gehele codebase zich bevindt, is overgedragen aan de opdrachtgever en de andere betrokkenen.

4.17 Veldtesten in Malaga

Na de overdracht van de applicatie is een trip naar Malaga, Spanje georganiseerd om daar de applicatie te veldtesten. Smit Lamnalco heeft een laptop met de laatste versie van CAT Electronic Technician aangeschaft en deze is meegenomen op de trip. Van 13 mei tot en met 15 mei is in Malaga verbleven en is, onder leiding van Technical Superintendent's Gelu Mirzoc en Stefan Chakarov en Electrical Technical Officer Anatoliy Dimitrov, het proces van Engine tot email getest.

4.17.1 UI Testing

Nadat de rapporten gegenereerd zijn, is het transformatie proces gestart. Om zeker te weten dat het transformatie proces voldoet aan de eisen gesteld in de requirements en de gemaakte use cases, zijn UI Acceptance tests³³ gemaakt.

Elke TestCase is gebaseerd op één van de zeven nog staande use cases. Elke test case is gebaseerd op een bepaald test scenario en een bijbehorend test proces. Men checkt of de verwachte uitkomst overeenkomt met de echte uitkomst en geeft aan of de test slaagt of faalt.

4.17.2 Testing met Azure Test Plan

Alle UI tests zijn na bedenken in Azure DevOps opgesteld. Alle tests worden ondergebracht in een Test suite. De Test suite hangt aan een taak die is gemaakt of aan een iteratie die is gespecificeerd.

³³ Bijlage 16 Test cases blz. 176-178

Test suite: 48 : Test UI (Suite ID: 51)

Tests Charts

Run

Outcome	Order	ID ↑	Title
Active	1	50	TC01 Start
Active	2	52	TC02.1 Stop
Active	3	53	TC2.2 Stop
Active	4	54	TC3.1 Search for Vessel
Active	5	55	TC3.2 Search for Vessel
Active	6	56	TC4 Select Vessel
Active	7	57	TC5 Select Document
Active	...	8	TC6 Transform
Active	9	59	TC7 Send Mail

Figuur 36 Overzicht van de UI test suite in Azure DevOps

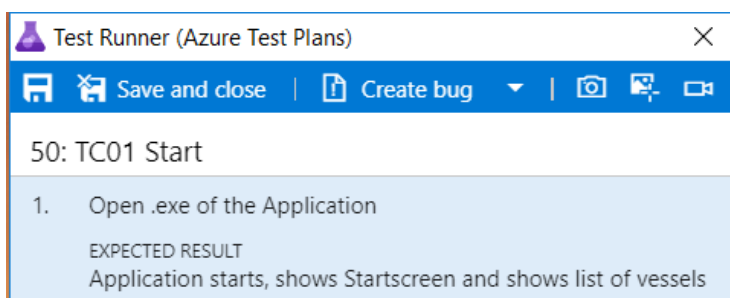
Steps

Steps Action Expected result

1.	Open .exe of the Application	Application starts, shows Startscreen and shows list of vessels
----	------------------------------	---

Click or type here to add a step

Figuur 37 Overzicht van Test case 1 in Azure DevOps



Figuur 38 Overzicht van Test case 1 in de Test Runner

In Azure Test Plan kunnen test cases gemakkelijk worden aangemaakt. Het werkt hetzelfde als wanneer de user met de hand een tabel invult. De user geeft de testcase een naam, een beschrijving, geeft stappen aan en hangt aan de stappen de verwachte uitkomst.

De test kan vervolgens automatisch worden uitgevoerd in een Azure Pipeline. Deze tests worden dan bij een nieuwe versie van de software automatisch gerund. Omdat hier gewerkt wordt met UI tests en de UI niet automatisch te runnen is, zonder daarvoor een complex systeem te bouwen, worden de tests manueel uitgevoerd.

Om de UI tests manueel uit te voeren heeft Azure Test Plan een handige applicatie bedacht, de Azure Test Runner. De Azure Test Runner is een desktop applicatie waarmee de user manueel tests kan uitvoeren. Het geeft een overzicht van de test weer met alle stappen en hun verwachte uitkomst.

Men kan een screenshot, image action log of screencapture maken van de acties die zijn gedaan. Wanneer de test stappen slagen, kan de user dat met een rechtermuisklik aangeven, faalt de test dan is dit ook aan te geven met een rechtermuisklik. Bij falen kan de user een bug aanmaken en Azure vult de stappen die ondernomen zijn zelf in. De bug wordt vervolgens aan het te testen onderdeel vastgemaakt, zodat het hele team weet dat iets moet veranderen aan de code.

Met de Test Runner zijn alle test gerund en geslaagd. Alle onderdelen voldoen aan de eisen die gesteld staan in de use cases.

5. Conclusie: Wat is uiteindelijk tot stand gekomen

Na een proces van 17 weken is een eind gekomen aan de afstudeerperiode. Tijdens deze periode is een applicatie gemaakt met als kerntaak het transformeren van een onoverzichtelijke XML structuur naar een overzichtelijke XML structuur. De onoverzichtelijke XML komt uit rapporten van Caterpillar motoren. Deze rapporten worden gegenereerd met Caterpillar Electronic Technician en geven de prestatiedata van de motor weer.

Deze applicatie is gemaakt door het volgen van de Extreme Programming en Feature Driven Development ontwikkelmethodieken. Met behulp van een aantal Scrum technieken is een aantal aanpassingen gemaakt op de ontwikkelmethodieken.

De applicatie is gebaseerd op het WinForms framework van Microsoft, die onder het .NET framework 4.6.1 valt. De applicatie is geschreven in de taal C# en maakt veelvuldig gebruik van de System.XML library van Microsoft voor de transformatie van de XML bestanden. Voor het verplaatsen van de bestanden is in de applicatie gebruik gemaakt van een ingebouwde Mail client. De Mail client maakt gebruik van de Microsoft.Office.Interop.Outlook library.

De getransformeerde XML documenten worden via een mail naar een vooraf afgesproken mail adres verzonden. Bij binnenkomst draait op de server een Scribe Integration Proces met een Microsoft MessageQueue. Dit proces neemt automatisch alle XML documenten uit de bijlagen mee in een queue. Tijdens dit proces wordt gefilterd op motormodel en wordt aan de hand van het motormodel bepaald welke Data Translation Specification in Scribe Insight wordt uitgevoerd.

Na het uitvoeren van de Data Translation Specification, wordt de data in een, speciaal per motormodel ontworpen, SQL Server database opgeslagen. In de toekomst kan de opgeslagen motorprestatie data worden gebruikt om overzichten en visualisaties te creëren voor het management.

6. Evaluatie: Hoe ging het, wat heb ik geleerd

Ik heb zeer veel geleerd tijdens deze periode. Ik heb leren werken in een multinational, wat in tegenstelling tot mijn derdejaars stage, bij een klein en informeel bedrijf was. Daarmee kwam dat ik mij moest aanpassen aan de structuren binnen het bedrijf. Waar ik in mijn derde jaar, overal zomaar om de hoek kon komen zetten, heb ik echt tijd moeten inplannen en afspraken moeten maken met mensen die ik wilde spreken.

Ik dacht in het begin van het project dat het werken binnen een multinational niet een heel grote invloed zou hebben op mijn functioneren. Echter, dat is een verkeerde aanname gebleken. Ik heb leren werken op hoog niveau, wat terug te zien is in het eindproduct. Alle betrokkenen zijn namelijk aangenaam verrast en ontzettend blij met het product. Ze hadden nooit gedacht dat ik zo ver zou komen. Wat ik heb neergezet is goed en dat geeft mij veel vertrouwen.

Ik heb me voor de tweede keer verdiept in XML en heb meer geleerd dan de vorige keer. Ik heb leren werken met moeilijkere facetten van XML zoals XPath en XML Schema's³⁴.

³⁴ XML Schema's gemaakt blz. [49-50](#)

Ik heb op hoog niveau problemen geanalyseerd en opgesteld, ik heb een bedrijfsanalyse uitgevoerd en heb mij verdiept in alle facetten van mijn project en daarbij de doelstelling van het bedrijf voor ogen gehouden. Daarnaast heb ik met Sander het Business Model Canvas³⁵ uitgediept om meer over het bedrijf en de processen te weten te komen.

Ik heb op hoog niveau leren werken aan mijn communicatie met verschillende mensen van verschillende achtergronden, zeker omdat de voertaal van het bedrijf Engels is. Ik heb stakeholders geïnterviewd³⁶ en daarmee requirements geëliciteerd. Deze heb ik vervolgens geprioriteerd en heb hiervan bij bijna elke iteratie nieuwe use cases³⁷ van gemaakt.

Ik heb bij elke iteratie gedacht over verschillende oplossingen en heb de, naar mijn inzicht, beste uitgekozen en ontworpen. Ik heb met de keuze voor C# als programmeertaal een object georiënteerde taal uitgekozen³⁸ en dat is terug te zien in het ontwerp. Er zijn meerdere keuzes gemaakt voor het invoeren van Design Patterns³⁹, of juist niet⁴⁰. Bij elke iteratie is een klasse diagram gemaakt wat op het eind altijd netjes overeen kwam met de werkelijkheid.

Ik heb ontwerpen⁴¹ gemaakt om het voor mijzelf en de stakeholders duidelijk te maken, wat ik ging implementeren en hoe ik dit ging realiseren. Ik heb meerdere Sequence diagrammen over hoe buitenstaande systemen en users invloed uitoefenen op de applicatie, Data Transfer diagrammen om aan te geven hoe de data verloopt wanneer mijn applicatie wordt geïmplementeerd en UI wireframes om de GUI van de applicatie te presenteren, gemaakt.

Naast het ontwerpen van software heb ik ook een databasemodel⁴² gemaakt. Ik heb bij elke stap gedacht aan wat de invloed van het systeem zou hebben op de tussen structuur en wat dat uiteindelijk voor invloed had op de database. Met dit in het achterhoofd, is steeds terug gekomen op het datamodel en heb ik aanpassingen gemaakt die op dat moment nodig waren, al was dat normaliseren⁴³ of juist weer terug.

Naast ontwerpen heb ik ook daadwerkelijk software⁴⁴ ontwikkeld en daarmee een product gemaakt. Er is zoals eerder verteld in C# gewerkt in Visual Studio. Hiermee is aan de hand van het WinForms⁴⁵ framework een desktop applicatie gemaakt, welke rust op het .NET framework 4.6.1. Ik heb een volledige Outlook client geïntegreerd binnen mijn applicatie en heb alle transformaties gedaan met de XML library van Microsoft.

Naast de software is volgens ontwerp ook een database⁴⁶ opgesteld. Deze is bij elke iteratie veranderd en opnieuw gerealiseerd. Er zijn custom views gemaakt om het voor de data analyst makkelijker te maken. Daarnaast is een volledige data pipeline⁴⁷ opgebouwd met Microsoft

³⁵ Bijlage Business Model Canvas blz. [76](#)

³⁶ Hoofdstuk Requirements blz. [21-22](#)

³⁷ Use Cases bij Iteraties blz. [87-93](#)

³⁸ Keuze programmeertaal blz. [26](#)

³⁹ Invoeren Design Pattern blz. [62](#)

⁴⁰ Niet invoeren Design Pattern blz. [45](#)

⁴¹ Hoofdstukken ontwerpen blz. [30](#) & [41](#) & [49](#) & [59](#) & [60](#)

⁴² Database modellen blz. [105-106](#) & [46](#) & [48](#)

⁴³ Keuze normaliseren of niet blz. [46-48](#)

⁴⁴ Hoofdstukken implementatie blz. [30-33](#) & [42-46](#) & [49-53](#) & [59-60](#) & [60-62](#)

⁴⁵ Keuze Winforms blz. [58](#)

⁴⁶ SQL Queries blz. [58](#) & [134-154](#)

⁴⁷ Ontworpen data Pipeline blz. [53-58](#)

MessageQueue en heb ik in Scribe Insight Data Translation Specifications geschreven, uniek voor elk soort motormodel.

Aan het eind zijn testcases⁴⁸ opgebouwd aan de hand van de use cases die over waren na het omvormen van de applicatie. Deze tests zijn met behulp van Azure DevOps gerund⁴⁹ en hebben aangetoond dat de applicatie voldoet aan de eisen die gesteld zijn.

De meeste keuzes die ik heb gemaakt, zijn gebaseerd geweest op kennis van het internet. En zoals altijd moet je kritisch omgaan met gegevens van het internet. Daarom ben ik ook op zoek gegaan naar bronnen daarbuiten. Ook al is Google Scholar⁵⁰ onderdeel van het internet, beschouw ik de data daarop niet als ‘zomaar data op het internet’. Er is een aantal wetenschappelijke artikelen gevonden en deze hebben mij bij sommige keuzes geholpen om de juiste te maken.

Ik heb geprobeerd binnen de Extreme Programming⁵¹ methodiek te werken. Later is gebleken dat dit een verkeerde keuze is geweest. Uiteindelijk is daarom gekozen om het ontwikkel proces om te vormen naar Feature Driven Development⁵². Aan zowel de Extreme Programming, als de Feature Driven Development methodiek zijn Scrum en Scrumboard toegevoegd. Daarnaast heb ik alle features opgeslagen en uitgewerkt in Azure DevOps en daar zijn alle iteraties mee gemaakt.

In Azure DevOps is gebruik gemaakt van Azure Repo's als versiebeheer, Azure Board's om te dienen als Scrumboard en Azure Testplans om UI tests mee te maken en uit te voeren.

Binnen Smit Lamnalco wordt heel informeel omgegaan met stagairs en dat is terug te zien in het contact tussen stakeholders en stagiairs. Ik werd in het begin veel los gelaten om mijn eigen dingen te doen en dat heeft gezorgd voor een gebrek aan communicatie. Door gebrek aan communicatie heb ik de neiging om dingen aan te nemen die voor mij als vanzelfsprekend klinken. Dit heeft gevolgen gehad op het development proces⁵³.

Ik heb na deze set back dan ook afgesproken om aan het eind van elke iteratie een review te houden met verschillende stakeholders. Hier werd dan uitgebreid gesproken, niet alleen over mijn applicatie maar ook over mij en mijn plannen in de toekomst.

Ik heb veel over mijzelf geleerd tijdens dit afstudeertraject. Over de manier hoe ik communiceer binnen een team zonder andere developers, hoe ik ben tegenover mensen met een andere cultuur en hoe ik om ga met een hiërarchische structuur in een bedrijf.

Daarnaast heb ik ontzettend veel geleerd over verschillende bedrijfsvoering van de IT. Ik heb nog nooit zo'n groot Business Intelligence systeem als dat van Smit Lamnalco gezien. De gesprekken met Matthijs hebben mij veel geleerd en inzicht gegeven over de BI kant van de IT. In gesprekken met Emad heb ik veel geleerd over de SysAdmin en Databasing kant van de IT.

Ondertussen bereid ik me voor op mijn Master in Computer Science and Advance Data Analytics aan de Universiteit Leiden. Ik heb veel geleerd over hoe ik bepaalde dingen wel en niet moet aanpakken en ik kijk vol zelfvertrouwen naar de toekomst.

⁴⁸ Testcases ontworpen blz. [63](#) & [165-167](#)

⁴⁹ Testcases gerund blz. [63-64](#)

⁵⁰ Wetenschappelijke Artikelen in Literatuurlijst blz. [68-69](#)

⁵¹ Methodische gewerkt via XP blz. [22-25](#)

⁵² Verandering naar FDD blz. [41](#)

⁵³ Hoofdstuk Reactie na set back blz. [35](#)

7. Reflectie: Wat kan beter, wat had ik anders kunnen doen

Er kan altijd wel wat verbeterd worden en mijn applicatie en proces zijn geen uitzondering. Naast het UI testen zou het mooi zijn als ik ook wat unit tests had gemaakt.

Door aan het begin van het proces te weinig te communiceren met de betrokkenen zijn verkeerde aannames gemaakt. Dit had ik beter kunnen aanpakken door eerder vaste besprekingen te plannen.

Er is helaas een afweging gemaakt om geen visualisaties te maken, maar te focussen op het omvormen naar een applicatie met een GUI. Dit had met meer tijd, beide kunnen worden gemaakt.

Doordat ik getest heb in Malaga op een Auxilliary engine en niet op een Main Engine, heb ik helaas getest op een niet geïmplementeerde motor. Deze heb ik tijdens het test proces moeten implementeren. Ik had graag op een 3516B getest of op een C18ACERT.

Omdat er een kort tijdsbestek beschikbaar was, is er met bekende dingen gewerkt. Een taal als C# heb ik gebruikt mede omdat ik deze taal beheers. Er was met meer tijd wellicht een alternatief gevonden dat beter paste bij het project.

Doordat ik in mijn eentje werkte, zijn er keuzes gemaakt, zoals bijvoorbeeld om via XP te werken, die wellicht niet zouden zijn gemaakt met een groter team. Hiermee krijg je een ontzettend eenzijdige kijk op het project wat van invloed is op de kwaliteit van het project.

Sommige onderdelen hebben veel tijd gekost, waar eerder vanaf kon worden gestapt. Een onderdeel als bijvoorbeeld het proberen om een PDF te extracten, kon wellicht eerder worden geschrapt. Hiermee zou extra tijd vrijkomen, waarmee visualisatie konden worden gebouwd.

Omdat ik in mijn eentje heb gewerkt aan alle facetten van de softwareontwikkeling, heb ik ook zelf mijn eigen code getest. Dit geeft automatisch een bias ten aanzien van de kwaliteit van de software. Dit kan worden verbeterd door, iemand buiten de ontwikkeling om te laten testen.

8. Literatuurlijst

1. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. Retrieved from <https://arxiv.org/abs/1709.08439>
2. Airbrake. (2017). *XP Feedback loop* [Image]. Retrieved from <https://airbrake.io/blog/sdlc/extreme-programming>
3. Altova. (2005). XML Editor: XMLSpy | Altova. Retrieved from <https://www.altova.com/xmlspy-xml-editor>
4. Asmul, S. (2014). FileSystemWatcher vs polling to watch for file changes. Retrieved from <https://stackoverflow.com/questions/239988/filesystemwatcher-vs-polling-to-watch-for-file-changes>
5. Basel, K. (2018). Python Pros and Cons: What are The Benefits and Downsides of the Programming Language. Retrieved from <https://www.netguru.com/blog/python-pros-and-cons-what-are-the-benefits-and-downsides-of-the-programming-language>
6. Beng Leau, Y., Khong Loo, W., Yip Tham, W., & Fun Tan, S. (2012). *Software Development Life Cycle AGILE vs Traditional Approaches* [Ebook] (37th ed., pp. 165-166). Singapore: International Conference on Information and Network Technology. Retrieved from https://www.researchgate.net/profile/Leau_Yu_Beng/publication/268334807_Software_Development_Life_Cycle_AGILE_vs_Traditional_Approaches/links/546989b40cf2f5eb1804f3d1/Software-Development-Life-Cycle-AGILE-vs-Traditional-Approaches.pdf
7. bigsan. (2011). iTextSharp library. Retrieved from <https://www.nuget.org/packages/iTextSharp/>
8. Chacon, S. (2009). *Pro Git* (2nd ed., pp. 11-12). New York: Apress.
9. Cockburn, A. (2004). *Crystal clear a human-powered methodology for small teams* (1st ed., pp. 2-4,). Addison-Wesley Professional.
10. Corrigan, S. (2002). *Introduction to the Controller Area Network (CAN)* [Ebook] (1st ed., p. 16). Texas Instruments. Retrieved from <http://shavali.yolasite.com/resources/Embedded/protocols/can.pdf>
11. DevOps. (2019). Retrieved from <https://en.wikipedia.org/wiki/DevOps>
12. EDUCBA. (2019). Difference between Winforms vs WPF [Blog]. Retrieved from <https://www.educba.com/winforms-vs-wpf/>
13. Evolution4all. (2019). *Overview of the SCRUM Process* [Image]. Retrieved from <https://luis-goncalves.com/what-is-scrum-methodology/>
14. Fox, B. (2017). LucidChart — One Of The Best Diagramming & Wireframing Apps | Noplag Review. Retrieved from <https://medium.com/@braydenfox/lucidchart-one-of-the-best-diagramming-wireframing-apps-noplag-review-dc234f16c85a>
15. Git. (2019). Retrieved from <https://en.wikipedia.org/wiki/Git>
16. GitLab. (2019). Retrieved from <https://about.gitlab.com/devops-tools/azure-devops-vs-gitlab.html>
17. Grieg, J. (2018). Microsoft Outlook rolling out end-to-end encryption to protect business email [Blog]. Retrieved from <https://www.techrepublic.com/article/microsoft-outlook-rolling-out-end-to-end-encryption-to-protect-business-email/>
18. Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps?: A Systematic Mapping Study on Definitions and Practices. *XP '16 Workshops Proceedings Of The Scientific Workshop Proceedings Of XP2016*, 12(1), 5. doi: 10.1145/2962695.2962707
19. Karam, L. (2017). FDD; ITS PROCESSES & COMPARISON TO OTHER AGILE METHODOLOGIES [Blog]. Retrieved from <https://apiumhub.com/tech-blog-barcelona/feature-driven-development/>
20. Konstatin. (2018). A DFSR Alternative: Fast & Resilient P2P Replication With Connect [Blog]. Retrieved from <https://www.resilio.com/blog/dfs-r-alternative>
21. KVH Industries. (2019). About Us. Retrieved from <https://www.kvh.com/About-Us.aspx>
22. Lotz, M. (2018). Waterfall vs. Agile: Which Methodology is Right for Your Project?. Retrieved from <https://www.seguetech.com/waterfall-vs-agile-methodology/>
23. Lysis. (2017). Pros and Cons of Using C# as Your Backend Programming Language. Retrieved from <https://agilites.com/pros-and-cons-of-using-c-as-your-backend-programming-language.html>
24. McQuaid, M. (2014). Why Use Version Control?. Retrieved from <https://mikemcquaid.com/2014/01/18/why-use-version-control/>

25. Microsoft Visio: software voor maken van diagrammen. (2019). Retrieved from <https://products.office.com/nl-nl/visio/flowchart-software?SilentAuth=1&wa=wsignin1.0>
26. Moe, N., & Dingsøyr, T. (2007). Scrum and Team Effectiveness: Theory and Practice. Retrieved from https://www.researchgate.net/profile/Torgeir_Dingsoyr/publication/221592867_Scrum_and_Team_Effectiveness_Theory_and_Practice/links/0046353ce2d281516b000000.pdf
27. Mohamed, S. (2015). DevOps shifting software engineering strategy Value based perspective. *IOSR Journal Of Computer Engineering*, 17(2), 56. doi: 10.9790/0661-17245157
28. MoSCoW-methode. (2019). Retrieved from <https://nl.wikipedia.org/wiki/MoSCoW-methode>
29. MSDN. (2002). FileSystemWatcher Class (System.IO). Retrieved from <https://docs.microsoft.com/en-us/dotnet/api/system.io.filesystemwatcher?view=netframework-4.8>
30. MSDN. (2006). ConfigurationUserLevel Enum (System.Configuration). Retrieved from <https://docs.microsoft.com/en-us/dotnet/api/system.configuration.configurationuserlevel?view=netframework-4.8>
31. MSDN. (2015). XML Processing Options. Retrieved from <https://docs.microsoft.com/en-us/dotnet/standard/data/xml/xml-processing-options>
32. NAYDENOV, P. (2015). Different Agile Methodologies: Find Which One Fits Best Your Needs [Blog]. Retrieved from <https://kanbanize.com/blog/right-agile-methodology-for-your-project/>
33. MSDN. (2018). Choose between .NET Core and .NET Framework for server apps. Retrieved from <https://docs.microsoft.com/en-us/dotnet/standard/choosing-core-framework-server>
34. Normandeau, K. (2018). How Netflix Uses Big Data to Drive Success [Blog]. Retrieved from <https://insidebigdata.com/2018/01/20/netflix-uses-big-data-drive-success/>
35. Openxcell. (2017). 5 Process Steps of FDD [Image]. Retrieved from <https://www.openxcell.com/9-things-must-know-fdd-feature-driven-development>
36. Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development - IEEE Conference Publication. Retrieved from http://www.ist.tugraz.at/_attach/Publish/Aksoftwaretechnologie1/9_requirements_engineering_and_agile_software_development.pdf
37. Petrov, C. (2019). Big Data Statistics 2019. Retrieved from <https://techjury.net/stats-about/big-data-statistics/>
38. Pros/Cons Of The C# Language - C# | Dream.In.Code. (2019). Retrieved from <https://www.dreamincode.net/forums/topic/29142-proscons-of-the-c#-language/>
39. Raps, M. (2019). What are the important features of Python?. Retrieved from <http://net-informations.com/python/intro/features.htm>
40. Schwaber, K. (1997). *SCRUM Development Process* (p. 9). London: Springer-Verlag London Limited.
41. Scribe Software Inc. (2018). Scribe Insight – A Data Integration Software Platform | Scribe Software. Retrieved from <https://www.scribESOFT.com/products/scribe-insight/>
42. Skeet, J. (2008). Why should I use an IDE?. Retrieved from <https://stackoverflow.com/questions/208193/why-should-i-use-an-ide>
43. Steigerwald, R., & Luqi. (1990). Rapid Software Prototyping. *Dspace Repository*, 1(1), 470. doi: 0073-1129-1/92
44. Thomas, P. (2019). GitLab vs GitHub: What Are The Key Differences?. Retrieved from <https://usersnap.com/blog/gitlab-github/>
45. Univertis. (2019). *Crystal Methodologies* [Image]. Retrieved from <https://univertis.com/general-en/introduction-to-agile-software-development/>
46. Villanueva, J. (2015). Managed File Transfer and Network Solutions [Blog]. Retrieved from <https://www.jscape.com/blog/12-file-transfer-protocols-businesses>
47. Wikipedia. (2019). Advanced Encryption Standard. Retrieved from https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
48. Wikipedia. (2019). Transport Layer Security. Retrieved from https://nl.wikipedia.org/wiki/Transport_Layer_Security
49. Wikipedia. (2019). X.509. Retrieved from <https://en.wikipedia.org/wiki/X.509>

9. Figuren lijst

FIGUUR 1 SMIT LAMNALCO SLOGAN.....	8
FIGUUR 2 VLOOT OVERZICHT VAN SMIT LAMNALCO.....	8
FIGUUR 3 SMIT LAMNACLO SLEEPBOOT	9
FIGUUR 4 SMIT LAMNALCO OFFSHORE SUPPORT VESSEL.....	9
FIGUUR 5 SMIT LAMNALCO CREW&PILOT VESSEL	10
FIGUUR 6 SMIT LAMNALCO LINE HANDLERS	10
FIGUUR 7 DETAIL OVERZICHT VAN SMIT LAMNALCO	10
FIGUUR 8 ONSHORE DUTIES VAN SMIT LAMNALCO, ALS AF- EN AANMEREN, SAFETY MANAGEMENT EN MINING LOGISTICS	10
FIGUUR 9 OFFSHORE DUTIES VAN SMIT LAMNALCO, ALS ESCORTING, SINGLE POINT MOORING EN CRYOGENIC HOSE HANDLING	11
FIGUUR 10 OVERZICHT STRUCTUUR BINNEN FLEET TECHNOLOGY AFDELING.....	12
FIGUUR 11 SMIT LAMNALCO TUGS AAN HET WERK	12
FIGUUR 12 HET FLEET CONTROL CENTER.....	13
FIGUUR 13 OVERZICHT DATAFLOW BINNEN SMIT LAMNALCO.....	14
FIGUUR 14 FOTO VAN EEN ECM	17
FIGUUR 15 DE CONNECTOR VAN CAT	17
FIGUUR 16 DATAFLOW NA IMPLEMENTATIE	18
FIGUUR 17 WATERVAL VS. AGILE	23
FIGUUR 18 OVERZICHT VAN DE XP CYCLUS.....	24
FIGUUR 19 OVERZICHT VAN DE SCUM CYCLUS.....	24
FIGUUR 20 OVERZICHT VAN DE STAPPEN VAN FDD	25
FIGUUR 21 VOORBEELDEN VAN MAIL APPLICATIES.....	34
FIGUUR 22 RESILIO P2P NETWERK VS CLIENT SERVER NETWERK.....	34
FIGUUR 23 TRANSFER PROTOCOLLEN	35
FIGUUR 24 ENGINE DATA TABEL ONTWERP V. MATTHIJS.....	47
FIGUUR 25 ENGINE DATA TABEL ONTWERP V. EMAD	47
FIGUUR 26 OUTLOOK RPC-SERVER ERROR.....	53
FIGUUR 27 EERSTE QUEUE	54
FIGUUR 28 FILTEREN OP ROOTNODE	54
FIGUUR 29 SCHEMA TOEWIJZEN AAN DE DTS.....	55
FIGUUR 30 DATABASE ALS TARGET	55
FIGUUR 31 OPNIEUW DE QUEUE IN.....	56
FIGUUR 32 MAPPING VAN XML LINKS EN DATABASE RECHTS.....	56
FIGUUR 33 DOCUMENT MET BOM OP 1, 2 EN 3	57
FIGUUR 34 DOCUMENT ZONDER BOM OP 1, 2 EN 3	57
FIGUUR 35 UITKOMST WINFORMS DESIGN	59
FIGUUR 36 OVERZICHT VAN DE UI TEST SUITE IN AZURE DEVOPS	63
FIGUUR 37 OVERZICHT VAN TEST CASE 1 IN AZURE DEVOPS.....	63
FIGUUR 38 OVERZICHT VAN TEST CASE 1 IN DE TEST RUNNER	63
FIGUUR 39 CAT MOTOREN AANBOORD DE SL MALKOHA	167
FIGUUR 40 BENODIGDHEDEN VOOR HET UITLEZEN VAN CAT MOTOREN	168
FIGUUR 41 OBD PLUG ZONDER KAP	168
FIGUUR 42 OBD PLUG MET KAP	168
FIGUUR 43 OBD KABEL IN PLUG	169

FIGUUR 44 CONNECTOR EN OBD KABEL.....	169
FIGUUR 45 PLUG DE OBD KABEL IN DE PLUG.....	169
FIGUUR 46 CONNECTOR STAAT AAN.....	170
FIGUUR 47 HOOFDSCHERM CAT ET	170
FIGUUR 48 J1939 LAMPJE FLIKKERT	171
FIGUUR 49 CAT ET MAAKT VERBINDING	171
FIGUUR 50 CONNECT MET DE MOTOR	171
FIGUUR 51 SELECTEER DE ECM	172
FIGUUR 52 OVERZICHT NA SELECTIE VAN ECM	172
FIGUUR 53 GENEREREN VAN EEN RAPPORT	173
FIGUUR 54 OVERZICHT VAN DE KEUZES VOOR HET RAPPORT.....	173
FIGUUR 55 GENEREER HET RAPPORT	174
FIGUUR 56 WACHT TOT HET GENEREREN KLAAR IS.....	174
FIGUUR 57 NIEUWE KOPIE OPSLAAN	175
FIGUUR 58 HET GEMAAKTE RAPPORT	175
FIGUUR 59 DICONNECT VAN DE MOTOR	175
FIGUUR 60 OPSLAAN ALS XML	175
FIGUUR 61 OPSLAAN ALS PDF	175
FIGUUR 62 SLUIT HET RAPPORT	175

10. Bijlagen

Bijlage 1. Plan van aanpak

Introduction

Smit Lamnalco is a big company made up of a partnership between Boskalis and Rezayat Group. Smit Lamnalco manages the fleet of 170 tugs located all around the world. Each ship has a set engines that generates data. This data must be structured, transformed and stored to eventually make useful visualizations, with the aim to support or improve various other business processes.

Stakeholders

Name	Function	Stake
Sander Woltheus	Group Fleet Technology Manager	Overall manager of this project, responsible
Hubert Plattel	Group Maintenance System Engineer	The Manager of the motor data
Emad Zaid	Group Technical Application Manager	Person which is in the process of with the current system
Matthijs Priester	Group Technical Data Analyst	Person which visualizes the data and perform analyses on data

Problem exploration

Problem domain

The more than 180 tugboats of Smit Lamnalco generate a lot of data from their engines. These engines are made by different manufacturers. The main manufacturers, and those that have the focus, are Caterpillar, Niigata, Wärtsilä, Yanmar and Bergen Diesel. These manufacturers have several ways to generate their data.

The towing business is tough so it is important that all operations are done economic, efficient and safe. For this, Smit Lamnalco wants to get a better understanding of the performance of the engines as monitoring and understanding the performances of all engines are vital. To assist in this, it is needed to have the operational data of the engines available ashore. First, this data must be standardized, second the usable data must be derived from the whole and third the data must be saved and visualized.

Research questions

There are a number of questions that must be answered within this project:

1. What is the best way to get data from ship to shore?
 - (a). Is the best also the quickest way and how important is that?
 - (b). What to do when there is no connectivity with a ship?
 - (c). How often should the data be delivered?
2. What is the best way to structure data, with the least possible data loss?
 - (a). What format is better for the data to be in?
3. Which elements are useful within the data and which are not?
 - (a). Where is important for the business?
 - (b). With which data can business processes be improved?
4. What is the best way to build the application?
 - (a). Which programming language is the most useful?
 - (b). What is the proper framework on which the application will be built?

Scope

Focus is on the engines of the manufacturer Caterpillar. Other engines are not included in this project

Focus is on the 'Data Highway', this means that the focus is on obtaining, processing and storing data.

Visualization is optional.

There will be a design and an implementation of the software and database.

A test document is optional.

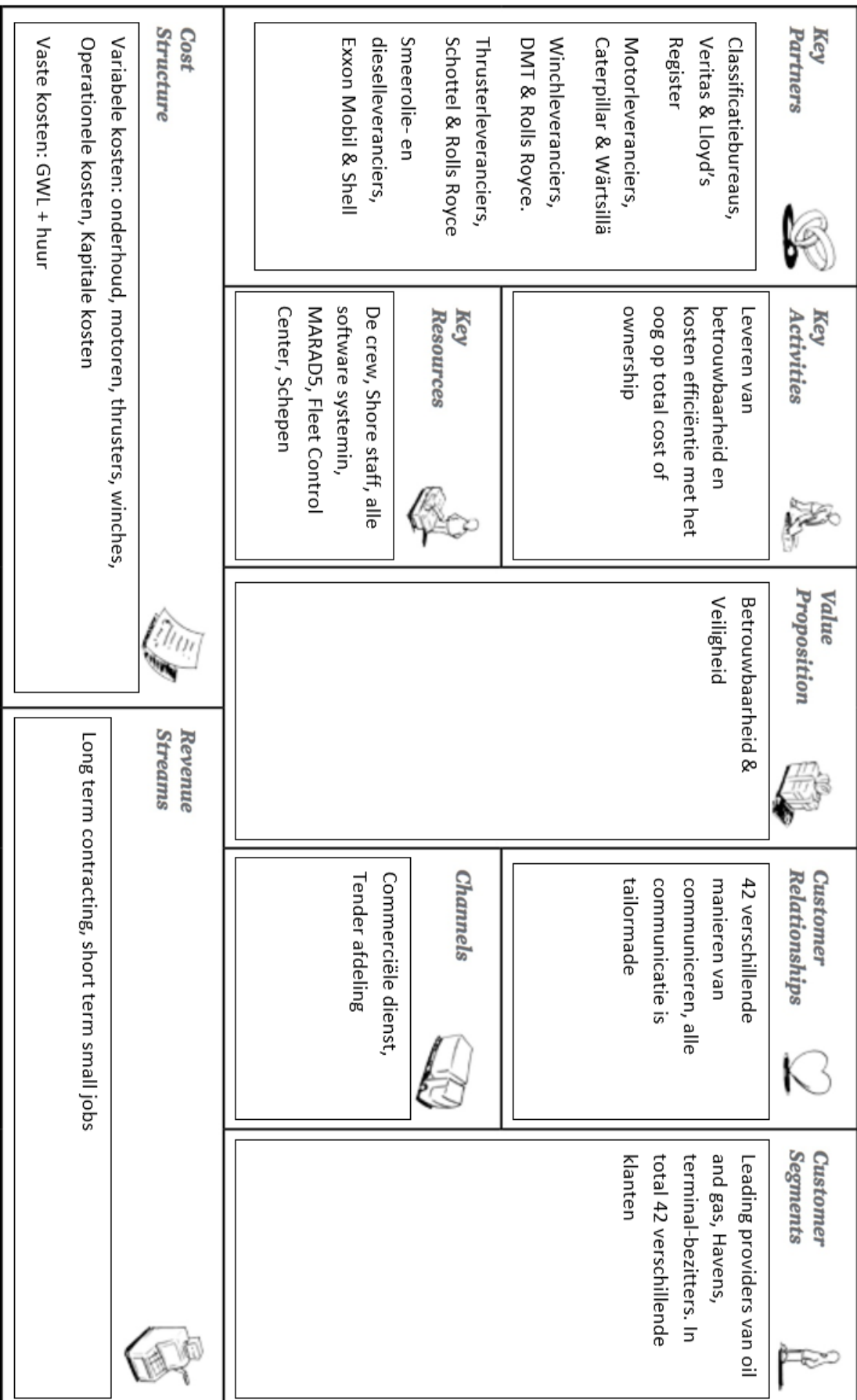
In this project the existing database and data warehouse will be used.

They do not have to again be designed and implemented.

Planning

Start	End	Task
04-02-2019	08-02-2019	POA completed and submitted
11-02-2019	13-02-2019	Discuss example files with Hubert
14-02-2019	14-02-2019	Interview with Hubert and Matthijs on usable data
15-02-2019	18-02-2019	Interviewing stakeholders to figure out requirements
19-02-2019	20-02-2019	Prioritize the requirements
21-02-2019	22-02-2019	Setup research on a way to move data from ship to shore
25-02-2019	01-03-2019	Research a way to move data from ship to shore
04-03-2019	05-03-2019	Setup research on a way for structuring the data
06-03-2019	11-03-2019	Research a way to structure the data
12-03-2019	13-03-2019	Setup research for programming language, framework, databasing, etc.
14-03-2019	15-03-2019	Research programming language, framework, databasing, etc.
18-03-2019	18-03-2019	Make decisions based on the research conducted
19-03-2019	19-03-2019	Discuss decisions with stakeholders
20-03-2019	22-03-2019	Design of application to move data from ship to shore
25-03-2019	29-03-2019	Building application to move data from ship to shore
01-04-2019	03-04-2019	Design of data structure and the application for transforming and storing it in the database
04-04-2019	08-04-2019	Designing database structure
09-04-2019	15-04-2019	Building application for structuring, transforming and storing it in the database
16-04-2019	28-04-2019	Extra time for the building of the applications
16-04-2019	19-04-2019	<i>Optional: designing application for visualization</i>
22-04-2019	26-04-2019	<i>Optional: build application for visualization</i>
29-04-2019	30-04-2019	<i>Optional: Design tests for the applications</i>

Bijlage 2. Business Model Canvas



Bijlage 3. Requirements Interview

Topic list:

- Data format
- Important data
- His view on the project
- His perfect end result
- Speed or accuracy
- Connectivity issues and security
- Delivery period

People:

- H = Hubert Plattel
- E = Emad Zaid
- M = Matthijs Priester
- S = Sander Woltheus

Question list:

1. Q: How do you, in your own words, see the project?
 - a. H: Doel in feite zou zij dat je de performance van de hoofdmotoren inzichtelijk kunnen maken, van hoe draait de motor, de een verbruikt vaker dan de ander loopt vaker in me kaar bijvoorbeeld je wil daar een beeld van maken wat er aan zit te komen. Elke keer als een boot docked is het een verassing wat er aan de hand is, en daar willen we wat aan doen, momenteel in marad staan alle jobs die worden uitgevoerd. Over de performance staat niks, er staat elke 1500 uur een performance job, geeft een rapportje over bijv. temperaturen, verbruik, alarmen, etc. als wat elektronisch versteld is of vervangen kan je terugvinden. Zoals nu komt er iemand van Caterpillar aan boord en die geeft dan een rapport. Maar we willen daar zelf naar kijken. In het verleden hebben ze bijvoorbeeld gezegd dat er wat moest gebeuren terwijl dat wel meeviel, dus we willen zelf inzicht. In kaart brengen wie en waar liggen de laptops met retrieval software. De data is niet vrijgegeven, iemand van Caterpillar zou wellicht meer kunnen zien dat iemand van ons.
 - b. M: wij hebben alle Caterpillar aan boor den wij willen weten wat de condities daarvan is, sensoren die alles meten zo snel mogelijk verzamelen om dat allemaal op te slaan we zijn daar nog niet mee bezig maar we willen dat binnen een half jaar/ jaar hebben. We zij meer met de foundation bezig, maar we willen die data aan wal krijgen en daar moet een systeem voor komen, hoe dat maakt niet uit, maar zo lang het maar hierheen komt en zo min mogelijk moeilijk voor de crew, misschien en tooltje voor maar niet teveel uitleg en werk, als dat kan met de systemen die we hebben. We hebben er al genoeg dus hoe minder hoe beter, bij ons gestructureerd die duidelijk gemapped zijn aan bestaande systemen, dus niet dat ik zelf moet zoeken met tabel uit Marad, wellicht een mapping table die je handmatig moet invoeren. Zodra het in de database zit is het voor mij genoeg, zou leuk zijn voor de crew als zij zelf zien wat de data doet. Als je data naar de wal stuurt, maak ik er een rapportje van zou mooi zijn als je dat kan visualiseren voor de crew zodat zij ook zien wat er gebeurd. Zou fijn zijn als ze interactief konden kijken hoe ze het hebben gedaan bijv. ten opzichten van vorige maand. We willen uiteindelijk terugkoppeling naar de schepen zelf.

- c. E: I think its converting different versions of data into structured data for analyzation in the office
 - d. S: ik zie dit project als een added value in het technical maintenance van de hele vloot binnen Smit Lamnalco en we gaan dit gebruiken als performance management tool
2. Q: What would be, in your opinion, the perfect end result?
- a. H: Zou mooi zijn zoals het gebeurt met lube oil, steeds de performance terug komt, bijvoorbeeld een alert, zoals temp gaat omhoog, of vermogen bij die belasting klopt niet, daar alerts van, zodat je niet alle datasheets door hoeft. Probleem is kwantiteit 170 schepen 2 motoren is erg veel om diep in te duiken. Dus alerts en dan bij die alerts dieper erin.
 - b. M: Ik ben super tevreden als wij in een sql database periodiek data krijgen met alle links tussen data die al bestaat. Werkzaam zijn voor de crew, opletten dat zij er mee moeten kunnen werken
 - c. E: a unified data set of the engines data of every different manufacturer, for 5 engines it is ambitious, so at least 2 engines. Roadmap is for done then, for the other 3. Description of the way to desynchronize the data/getting the data here.
 - d. S: Hier in het FCC een overzicht die in het groen oranje rood weergeeft welke engine goed is of niet. Afspraak is het vangen van data, wordt weggeschreven in het central data warehouse en klaargemaakt voor visualisatie. Frontpage zoals nu met de Lube Oil Samples.
3. Q: What would be the important data from each report for you?
- a. Q: From Caterpillar, Niigata, Wärtsilä, Yanmar, Bergen Diesel?
 - i. H: temperatuur, brandstof verbruik, koel water, smeerolie, belastingen, wellicht beter om te standaardiseren. Terug naar de basis het zijn allemaal dieselmotoren. We willen niet teveel te gedetailleerd willen weten. We komen van ver en we willen de basis. Focus is om dingen aan te zien komen zodat we niet afhankelijk zijn van fabrikant.
 - ii. M: Weet niet of alle 5 gaat lukken, we hebben eerste contact gelegd met Caterpillar, niet met niigata wartsila, of Bergen diesel. Eerst contact leggen om sensoren uit te lezen of dat überhaupt mogelijk is. Caterpillar heeft de mogelijkheden om uit te lezen, wellicht dat andere dat nog niet hebben. Mappen wie wel wie niet, geeft doorslag of we wellicht meer schepen er aan gaan hangen. Beetje leuk maken voor management.
 - 1. Load profiles: Hoeveel uren draai je op een bepaalde load. Vermogen en RPM
 - 2. Hoe vaak een lager draait, checken of je kan vervangen predictive maintenance
 - 3. Is meer aan Hubert
 - iii. E: more important to get proper error handling, logging, care more about the it side than the content.
 - iv. S: per region weten welk % boot groen oranje rood, Welke boot is in de problemen, welke motor is in de problemen. Details pas achteraf, management overview eerst creëren. Opgebouwd uit verschillende lagen van diepte.
4. Q: What format would you prefer that the data comes in?
- a. PDF, XML, XSLT, DOC, EXCEL, CSV?

- i. H: Het moet in feite in makkelijk overzichtelijk worden, dus excel zou mooi zijn. Zodat je makkelijk bewerkingen kan doen.
 - ii. M: Niet geïnteresseerd in
 - iii. E: just make sure it is structured
 - iv. S: Moet je met Matthijs en Emad bespreken
- 5. Q: What should be the ideal delivery period?
 - a. Quarterly, monthly, weekly?
 - i. H: in de nieuwe maintenance strategie staan jobpies van elke 1500 draaiuren. Voor sommige is dat 3 maanden andere boten een jaar. In principe is onderhoud draaiuren gerelateerd. Zou mooi om elke 1500 uur een rapport te krijgen. In de logs staan alarmen, load profiles, geeft een goed beeld van de afgelopen periode. Elke 1500 uur minstens een rapport. Elke maand zou mooi zijn.
 - ii. M: Voorlopig maandelijks genoeg, zodra het automatisch gaat wil ik het vaker.
 - iii. E: is a business requirement, not up to us
 - iv. S: Eigenlijk minimaal 1 keer per half jaar, per kwartaal is goed genoeg. Uiteindelijk als we dit automatiseren, wordt dit vaker gedaan.
- 6. Q: What, in your opinion, should be done when a ship has no connectivity, but should have sent its data?
 - a. H: net zoals nu bij Marad, moet door kunnen gaan. Achteraf sturen is geen probleem. Überhaupt blij als het komt, al is het week of meer achteraf.
 - b. M: Meetsystemen die we hebben zijn er op ingesteld, mail houdt het vast totdat er weer verbinding is, residual is hetzelfde. Als je niet kan sturen dat kan dat gebeuren, liever niet maar he.
 - c. E: service should be able to store the data, ping the service to get connectivity, make sure you log everything from source to shore. Error handling is very important
 - d. S: Als het 1 keer per kwartaal is, is dat geen probleem want 1 keer maand connecten met Marad5. Per week checken we of er connectiviteit is.
- 7. What are your thoughts on security of the performance data?
 - a. H: volgens mij is dat niet zo geheimzinnig, dus niet te veel rekening mee houden.
 - b. M: goeie om er rekening mee te houden, maar het is niet super spannende data om te stelen. Nice have, niet uit de weg gaan om te enforcement. Zelfde security als SLDOX, of als een mailtje.
 - c. E: Not really business critical, not personal bound, not GDPR, so not that important
 - d. S: Niet direct business kritisch, maar net zoals de rest. We willen niet dat dit lekt naar concurrenten. Hele FCC gebeuren niet naar buiten lekken.
- 8. Remarks?
 - a. H: Wat is er totaal beschikbaar, volgens mij kan CAT meer zien dan een klant. Als we weten wat er beschikbaar is, kunnen we ons gaan specificeren en focussen.
 - b. M: Dingen die lang kunnen duren zsm oppakken, zoals contact leggen met Wartsila of Bergen Diesel. Regelmatig met je begeleider zitten. Wellicht meetingjes plannen elke week.
 - c. E: engine connectivity, makes sense to use the same pc to auto get the data and send the data to shore, for automation one config, make a selfrunning exe, so no install is required. TSI manages more than 1 ship, dropdown for ship-ID, original file need to be saved as well.

- d. S: Veel vooruit plannen, 12-03 of 13-03 of 15-03 meeting met Examiner. Hou zelf de vaart erin, schroom niet om te bellen/mailen. Laat het op tijd weten of het allemaal lukt. Kwaliteit is koning. Hubert en Harsh gaan er nauw mee werken, dus veel met hen communiceren. Communiceren met Hubert en Laura over het gebruik van SLDox

Bijlage 4. Requirements

Functional

Must have

1. As an application manager, I want to have a unified standard form for the data structure, so that storage is easier and faster.
2. As a data analyst, I want to get the data in the existing database, so that I can use this as I am already doing.
3. As a data analyst, I want to have the engine data linked to all other instances of ship data, so that I can take this in my analyses.
4. As an application manager, I want to have very extensive logging and error handling, so when something goes wrong it is easy to see how and why.
5. As an application manager and technical manager, I want to have at least the Caterpillar engines covered, because this is the largest group of engines we have.

Should have

1. As a technical manager, I want to see a general overview per region, per ship, per engine, so that I can see what is going on faster.
2. As a maintenance engineer, I want to get alerts when there is a problem with an engine, so that I can see what is going on faster.
3. As a maintenance engineer, I want to get when trends indicate engine problems, so that I can see what is going on faster.
4. As a maintenance engineer, I want to have insight in the basic engine performance, so that I can see what is going on faster.

Could have

1. As an application manager, I want a system on board a vessel to detect the connectivity on a scheduled basis and sync when possible, so that I have an overview of which ship is in connection range.
2. As a general stakeholder, I want the data to stay put when there is a connectivity issue and send when reconnected, so that every document is sure to be send.
3. As a general stakeholder, I want the data to be of mail level security, so that it will not be intercepted.

Non-functional

Must have

1. As a general stakeholder, I want the system to be intuitive, so that crew members don't have to learn how to use yet another system aboard a ship.
2. As a maintenance engineer, I want to know what data is possible to get from an engine

Should have

1. As a maintenance engineer, I want the data format to be Excel, so that I can create easy overviews.

2. As a maintenance engineer, I want to know which engine manufacturers can send us engine data, so that an overview can be made for the future.

Could have

1. As a maintenance engineer, I want a report at least every 1500 engine working hours or once every six months, so I get a good overview of the engine health.
2. As a technical manager, I want a report every quarter, so I get a good overview of the engine health.

Bijlage 5. Data Mapping

	Alawaset ME Port	ALawaset ME Startboard	Hashim3 ME Port	Hashim3 ME Starboard
Product Status Report	1	1	1	1
Extra Parameters	1	1	1	1
Logged Diagnostic Codes	1	1	1	1
Logged Event Codes	1	1	1	1
Active Diagnostic Codes	1	1	1	1
Active Event Codes	1	1		1
Current Totals	1	1	1	1
Configuration	1	1	1	1
Lifetime: Time vs Engine Coolant Temperature	1	1		
Lifetime: Time vs Inlet Air Temperature	1	1		
Lifetime: Time vs Boost Pressure	1	1		
Lifetime: Time vs Engine Speed And Engine Load Factor	1	1	1	1
Lifetime: Total Occurrences vs Engine Speed	1	1		
Lifetime: Time vs Engine Speed	1	1	1	1
Lifetime: Time vs Engine Load Factor	1	1	1	1
Lifetime: Time vs Right Exhaust Temperature			1	1
Lifetime: Time vs Left Exhaust Temperature			1	1
Lifetime: Time vs Unknown			2	2
Lifetime: Time vs Fuel Pressure	1	1		
Lifetime: Time vs Turbo Inlet Temperature And Inlet Air Temperature	1	1		
Injector Trim Calibration	1	1		
Monitoring System	1	1		
Injector Codes Calibration				
3516B Port Backup				
Logged Diagnostic Codes Backup			1	1
Logged Event Codes Backup			1	1
Active Diagnostic Codes Backup			1	1
Current Totals Backup			1	1
Configuration Backup			1	1
PL1000 #2			1	1
Current Totals PL1000 #2			1	1
Configuration PL1000 #2			1	1
Engine Room MPD			1	1
Configuration - Engine Room MPD			1	1
Total	19	19	27	27

Product Status Report	Petra3 ME Port	Petra3 ME Starboard	Tareq 3 ME Port	Tareq 3 ME Starboard
Extra Parameters	1	1	1	1
Logged Diagnostic Codes	1	1	1	1
Logged Event Codes	1	1	1	1
Active Diagnostic Codes	1	1	1	1
Active Event Codes				
Current Totals	1	1	1	1
Configuration	1	1	1	1
Lifetime: Time vs Engine Coolant Temperature				
Lifetime: Time vs Inlet Air Temperature				
Lifetime: Time vs Boost Pressure				
Lifetime: Time vs Engine Speed And Engine Load Factor	1	1	1	1
Lifetime: Total Occurrences vs Engine Speed				
Lifetime: Time vs Engine Speed	1	1	1	1
Lifetime: Time vs Engine Load Factor	1	1	1	1
Lifetime: Time vs Right Exhaust Temperature	1	1	1	1
Lifetime: Time vs Left Exhaust Temperature	1	1	1	1
Lifetime: Time vs Unknown	2	2	2	2
Lifetime: Time vs Fuel Pressure				
Lifetime: Time vs Turbo Inlet Temperature And Inlet Air Temperature				
Injector Trim Calibration				
Monitoring System	1	1	1	1
Injector Codes Calibration	1	1	1	1
3516B Port Backup	1	1	1	1
Logged Diagnostic Codes Backup	1	1	1	1
Logged Event Codes Backup	1	1	1	1
Active Diagnostic Codes Backup	1	1	1	1
Current Totals Backup	1	1	1	1
Configuration Backup	1	1	1	1
PL1000 #2	1	1	1	1
Current Totals PL1000 #2	1	1	1	1
Configuration PL1000 #2	1	1	1	1
Engine Room MPD	1	1	1	1
Configuration - Engine Room MPD	1	1	1	1
Total	27	27	27	27

	Aldaem M/E Port	Aldaem M/E Starboard	Alsanad M/E Port	Alsanad M/E Starboard	TTJ00598 Port	TTJ00598 Starboard	Percentage
Product Status Report	1	1	1	1	1	1	100%
Extra Parameters	1	1	1	1	1	1	100%
Logged Diagnostic Codes	1	1	1	1	1	1	100%
Logged Event Codes	1	1	1	1	1	1	100%
Active Diagnostic Codes	1	1	1	1	1	1	100%
Active Event Codes	1	1	1	1	1	1	43%
Current Totals	1	1	1	1	1	1	100%
Configuration	1	1	1	1	1	1	100%
Lifetime: Time vs Engine Coolant Temperature	1	1	1	1	1	1	43%
Lifetime: Time vs Inlet Air Temperature	1	1	1	1	1	1	43%
Lifetime: Time vs Boost Pressure	1	1	1	1	1	1	43%
Lifetime: Time vs Engine Speed And Engine Load Factor	1	1	1	1	1	1	100%
Lifetime: Total Occurrences vs Engine Speed	1	1	1	1	1	1	43%
Lifetime: Time vs Engine Speed	1	1	1	1	1	1	100%
Lifetime: Time vs Engine Load Factor	1	1	1	1	1	1	100%
Lifetime: Time vs Right Exhaust Temperature					1	1	57%
Lifetime: Time vs Left Exhaust Temperature					1	1	57%
Lifetime: Time vs Unknown					2	2	114%
Lifetime: Time vs Fuel Pressure	1	1	1	1	1	1	43%
Lifetime: Time vs Turbo Inlet Temperature And Inlet Air Temperature	1	1	1	1	1	1	43%
Injector Trim Calibration	1	1	1	1	1	1	43%
Monitoring System	1	1	1	1	1	1	100%
Injector Codes Calibration					1	1	57%
3516B Port Backup					1	1	57%
Logged Diagnostic Codes Backup					1	1	57%
Logged Event Codes Backup					1	1	57%
Active Diagnostic Codes Backup					1	1	57%
Current Totals Backup					1	1	57%
Configuration Backup					1	1	57%
PL1000 #2					1	1	57%
Current Totals PL1000 #2					1	1	57%
Configuration PL1000 #2					1	1	57%
Engine Room MPD					1	1	57%
Configuration - Engine Room MPD					1	1	57%
Total	19	19	19	19	27	27	

Bijlage 6. Pre Research: Collecting Data

Introduction

A simple pre research consisting of asking question within Smit Lamnalco and some searching on the internet.

Interviews

Interviews have been held with Matthijs Priester, Hubert Plattel, Emad Zaid and Sander Woltheus.

The following options have been discussed SLDox/Resilio, Mail Service, new application.

What are the options?

There are three options up for debate.

One: the SLDox/Resilio solution, a solution which uses synchronized directories to deposit documents from all around the world to a centralized document store.

Two: a Mail service, using a standard mail address e.g. engineperformance@smitlamnalco.com, where chief engineers or super intendents can mail the engine reports to.

Three: a complete new system, using another form of file transfer system e.g. FTPs, MQTT, HTTPS.

The speed of the connection is not an issue, as told by Emad Zaid. This is because it is not of critical importance due to the data not being real time.

Findings

No way of knowing if one of the three would be easier to use over the other two, as there is simply no time to question 30 Super intendents all over the world within a week, but according to Hubert and Matthijs, I can assume that every TSI and Chief engineer knows how to use a mail service or SLDox. So both these option would require no extra education for this matter.

A new app would require education on how to use that new system.

A new app would require time to make, something there is not a lot of.

A new app would require research based on implementation possibilities, which are uncertain and could require days, even weeks, to collect.

The standing systems may not be feasible without a workaround.

The standing systems of SLDox and Mail may not be scalable enough to keep them in the future, there may eventually be need of another system.

SLDox will be the main document store point.

Conclusion

In the interviews it is said that the solution to this problem must be easy to use and require no to little extra education, the new system will not guarantee this. The current solutions may not be scalable, but this is not a critical problem as of now. Therefore the conclusion is to use one of the two standing structures.

Bijlage 7. Research Collecting Data

Introduction

This research is a qualitative exploratory research in order to research the possibilities on how to get data from the ships to the shore and make a conclusion as which one is better.

Collection methods

The researcher is going to collect data with an interview and using existing data found on the internet.

The interview is based on individual face-to-face questioning. The interview is mixed with a topic list and open interview questions.

Population

The population will be limited to the stakeholders of the project: Hubert Plattel, Matthijs Priester, Emad Zaid and Sander Woltheus.

Pre research conclusion

A small pre research concluded that there are two options to choose from, Resilio/SLDox or a Mailservice. This is based on the fact that they are both easy to use and are already in place. Both options in the future could be done on the ship itself so there is no more need for in between persons handling the documents. Both options can send the files within the discussed timespan of 1500 working hours or every quarter. Both options are known with the crew as well with the TSI's, which will mean no extra education on using something new. Speed is a non-issue with both of the options as this is not a critical aspect of the project.

Research questions

Main research question

What would be the best method of collecting engine performance data from a vessel?

Secondary research questions

- How will loss of connectivity be handled by both systems?
- In what format should the data arrive?
- How will security be handled by both systems

Findings

How will loss of connectivity be handled by both systems?

Both options have the ability to wait when not connected. The Mailservice onboard will have the document in the outbox until there is a connection. Resilio/SLDox has the ability to wait for all the directories to synchronize with each other.

In what format should the data arrive? XML, Excel, PDF, XSLT?

Because of the importance of data integrity and structure of the data, it is important to know what is the best format the data should arrive in. The reports are coming in as either HTML, XSLT or PDF. CAT themselves are not able to give this data in any other format.

In contact with Albert Kasbergen from PonPower, SDK Diesel Support B.V., CAT and MichiganCAT findings were that it is not possible for CAT Electronic Technician to deviate from the formats they have.

How will security be handled by both systems?

Although in the interview it is said that security is not of critical importance.

Outlook uses Windows Authentication before the client is accessed, when sending mails it uses end-to-end encryption so that emails will be send secured.

SLdox /Resilio makes use of Secure Sockets Layer(SSL) which ensures that files transferred using their peer to peer network are secure. SLdox/Resilio uses AES-128 encryption when the data is in transit.

Conclusion

The researcher concludes that SLdox will be the more suitable solution to the problems ahead, this due to the fact that, although a mail service solution is as good as a SLdox solution, SLdox will be the main document storing application and is already structured.

Bijlage 8. Use cases

Iteratie 1

User

Case 1

ID	UC1
Use Case	User – Start
Beschrijving	De user start het programma.
Primaire actor	User
Secundaire actor	n.v.t.
Pre-conditie	Het programma is nog niet gestart.
Main flow	1. Actor opent de .exe van het programma 2. De applicatie start 3. De applicatie laat het start scherm zien
Post-conditie	Applicatie is geopend en het start scherm wordt getoond.
Alternatieve flow	n.v.t.

Case 2

ID	UC2
Use Case	User - Stop
Beschrijving	De user stopt het programma.
Primaire actor	User
Secundaire actor	n.v.t.
Pre-conditie	Het programma is gestart.
Main flow	1. Actor sluit programma 2. Applicatie stopt met werken 3. Applicatie sluit zichzelf
Post-conditie	Applicatie is gesloten.
Alternatieve flow	n.v.t.

Case 3

ID	UC3
Use Case	User - Transform
Beschrijving	De user transformeert het document.
Primaire actor	User
Secundaire actor	SQL Server
Pre-conditie	Programma is gestart en documenten zijn beschikbaar voor transformatie.
Main flow	1. User start transformatie proces 2. Systeem transformeert het document 3. Include UC4: Save Document
Post-conditie	Transformatie proces is gelukt, document is opgeslagen.
Alternatieve flow	n.v.t.

SQL Server

Case 4

ID	UC4
Use Case	SQL Server – Connect
Beschrijving	Er wordt een connectie vast gesteld met een SQL Server database.
Primaire actor	SQL Server
Secundaire actor	n.v.t.
Pre-conditie	Er is een SQL Server Database beschikbaar.
Main flow	1. Connectie data wordt verzameld 2. Connectie wordt opgebouwd 3. Connectie wordt geretourneerd
Post-conditie	Connectie bestaat en is klaar voor gebruik.
Alternatieve flow	n.v.t.

Case 5

ID	UC5
Use Case	SQL Server – Save Document
Beschrijving	Tijdens het Transform proces wordt de data opgeslagen in een SQL Server.
Primaire actor	SQL Server
Secundaire actor	User
Pre-conditie	Transformatie proces is afgerond, connectie is gemaakt met de SQL Server.
Main flow	1. Insert statement wordt opgebouwd 2. Insert statement wordt over de connectie naar de database gestuurd
Post-conditie	Data staat in de database opgeslagen.
Alternatieve flow	n.v.t.

File System

Case 6

ID	UC6
Use Case	File System - Load Documents
Beschrijving	Documenten worden in het systeem geladen vanuit een folder.
Primaire actor	File System
Secundaire actor	n.v.t.
Pre-conditie	Programma is gestart.
Main flow	1. Documenten worden geselecteerd 2. Documenten worden ingeladen in het systeem 3. Documenten worden beschikbaar gesteld binnen het systeem
Post-conditie	Documenten zijn beschikbaar voor transformatie.
Alternatieve flow	n.v.t.

Iteratie 2

Herzien use case 1

ID	UC1
Use Case	User – Start
Beschrijving	De user start het programma.
Primaire actor	User
Secundaire actor	File System
Pre-conditie	Het programma is nog niet gestart.
Main flow	1. User opent de .exe van het programma 2. De applicatie start 3. Include UC7: Initialise 4. De applicatie laat het start scherm zien
Post-conditie	Applicatie is geopend en het start scherm wordt getoond en de mappen in de AppData zijn aanwezig.
Alternatieve flow	n.v.t.

Herzien use case 3

ID	UC3
Use Case	User - Transform
Beschrijving	De user transformeert het document.
Primaire actor	User
Secundaire actor	File System
Pre-conditie	Programma is gestart.
Main flow	1. User start transformatie proces 2. Include U6: Load Documents 3. Systeem transformeert het documents 4. Include UC5: Save Document
Post-conditie	Transformatie proces is gelukt, documenten zijn opgeslagen.
Alternatieve flow	AF3.A Document wordt niet goed getransformeerd, het systeem stopt met het huidige document. Laat de user weten dat er wat mis is gegaan. Gaat verder met een nieuw document in stap 3.

Herzien use case 5

ID	UC5
Use Case	File System – Save Document
Beschrijving	Getransformeerde documenten worden opgeslagen in een folder in de AppData map.
Primaire actor	File System
Secundaire actor	User
Pre-conditie	Programma is gestart en transformatie is gestart.
Main flow	1. Applicatie selecteert de opslag folder op de user's pc 2. Documenten worden opgeslagen
Post-conditie	Documenten zijn opgeslagen in de AppData folder op de user's pc.
Alternatieve flow	n.v.t.

Herzien use case 6

ID	UC6
Use Case	File System - Load Documents
Beschrijving	Documenten worden in het systeem geladen vanuit een folder in de AppData Map.
Primaire actor	File System
Secundaire actor	User
Pre-conditie	Programma is gestart en transformatie is gestart
Main flow	1. Systeem selecteert de inlaad folder op de User's pc 2. Systeem laad alle XML files in het systeem 3. Systeem presenteert documenten.
Post-conditie	Documenten zijn beschikbaar voor transformatie.
Alternatieve flow	AF2.A Er zijn geen documenten beschikbaar, het systeem stopt en geeft aan dat er geen documenten aanwezig zijn.

Nieuwe use case 7

ID	UC7
Use Case	File System - Initialise
Beschrijving	De folders welke nodig zijn worden aan de start van het document aangemaakt.
Primaire actor	File System
Secundaire actor	User
Pre-conditie	Programma is gestart
Main flow	1. Document map AppData wordt opgezocht op de User's pc 2. Map voor inladen wordt aangemaakt 3. Map voor opslaan wordt aangemaakt 4. Map locaties worden opgeslagen voor verder gebruik
Post-conditie	Twee mappen zijn aanwezig in de AppData map van de User's pc.
Alternatieve flow	AF2.A Map bestaat al, stap 2 wordt overgeslagen. Systeem gaat verder met stap 3. AF3.A Map bestaat al, stap 3 wordt overgeslagen. Systeem gaat verder met stap 4.

Iteratie 3

Nieuwe use case 8

ID	UC8
Use Case	User – Send Mail
Beschrijving	De user gebruikt een mail client om emails te sturen na de transformatie
Primaire actor	Mail Client
Secundaire actor	User
Pre-conditie	De transformatie is gelukt
Main flow	1. De applicatie geeft een vooraf gevulde mail weer 2. De user geeft aan de mail te versturen 3. De mail client verstuurd de mail
Post-conditie	De mail is verzonden vanuit de desbetreffende mail client
Alternatieve flow	AF2.A De user cancelt de mail, stap 3 wordt overgeslagen

Herzien use case 3

ID	UC3
Use Case	User – Transform
Beschrijving	De user zet het transformatie proces in en de documenten wordt getransformeerd
Primaire actor	User
Secundaire actor	Mail client, File System
Pre-conditie	Het programma is nog niet gestart
Main flow	1. De user geeft aan het transformatieproces te willen starten 2. Include UC6: Load Document 3. De applicatie transformeert het document 4. Include UC5: Save Document 5. Include UC8: Send Mail
Post-conditie	Applicatie heeft het transformatieproces goed doorlopen en er is een mail verzonden met de documenten
Alternatieve flow	AF2.A Document is niet goed getransformeerd, de applicatie geeft aan dat er iets is misgegaan, slaat stap 3 over en wacht op input van de user

Iteratie 4

Nieuwe use case 9

ID	UC9
Use Case	User – Search for Vessel
Beschrijving	De user gebruikt de search bar om naar een schip te zoeken
Primaire actor	User
Secundaire actor	n.v.t.
Pre-conditie	Het programma is gestart
Main flow	1. User typt de naam of deel van de naam van een schip in 2. De lijst met schepen veranderd aan de hand van de input 3. De applicatie laat de lijst met schepen zien, welke overeenkomen met de input
Post-conditie	De applicatie laat vessels, aan de hand van de input, zien in een schip lijst
Alternatieve flow	AF3.A Er zijn geen schepen die overheen komen, de lijst is leeg en het systeem wacht op verdere input

Nieuwe use case 10

ID	UC10
Use Case	User – Select Vessel
Beschrijving	De user selecteert een schip uit een lijst met vessels
Primaire actor	User
Secundaire actor	n.v.t.
Pre-conditie	Het programma is gestart, er zijn schepen beschikbaar
Main flow	1. De user selecteert een schip uit de schepen lijst 2. De applicatie onthoudt het schip en laat de naam zien aan de user 3. De applicatie activeert de volgende stap
Post-conditie	De naam van het schip wordt weergegeven en de volgende stap is geactiveerd
Alternatieve flow	n.v.t.

Nieuwe use case 11

ID	UC11
Use Case	User – Select Document
Beschrijving	De user selecteert het document dat de user wil transformeren
Primaire actor	User
Secundaire actor	File System
Pre-conditie	De user heeft een schip geselecteerd
Main flow	1. De user geeft aan dat deze een document wil selecteren 2. De applicatie geeft een overzicht van de te selecteren documenten 3. De user selecteert een document 4. Include UC6: Load Document 5. De applicatie geeft de naam van het document weer 6. De applicatie activeert de volgende stap
Post-conditie	Applicatie geeft de naam van het document weer, de volgende stap is geactiveerd
Alternatieve flow	AF3.A Het geselecteerde document is geen XML bestand, de applicatie laat weten aan de user dat deze een XML bestand moet selecteren en gaat terug naar stap 3

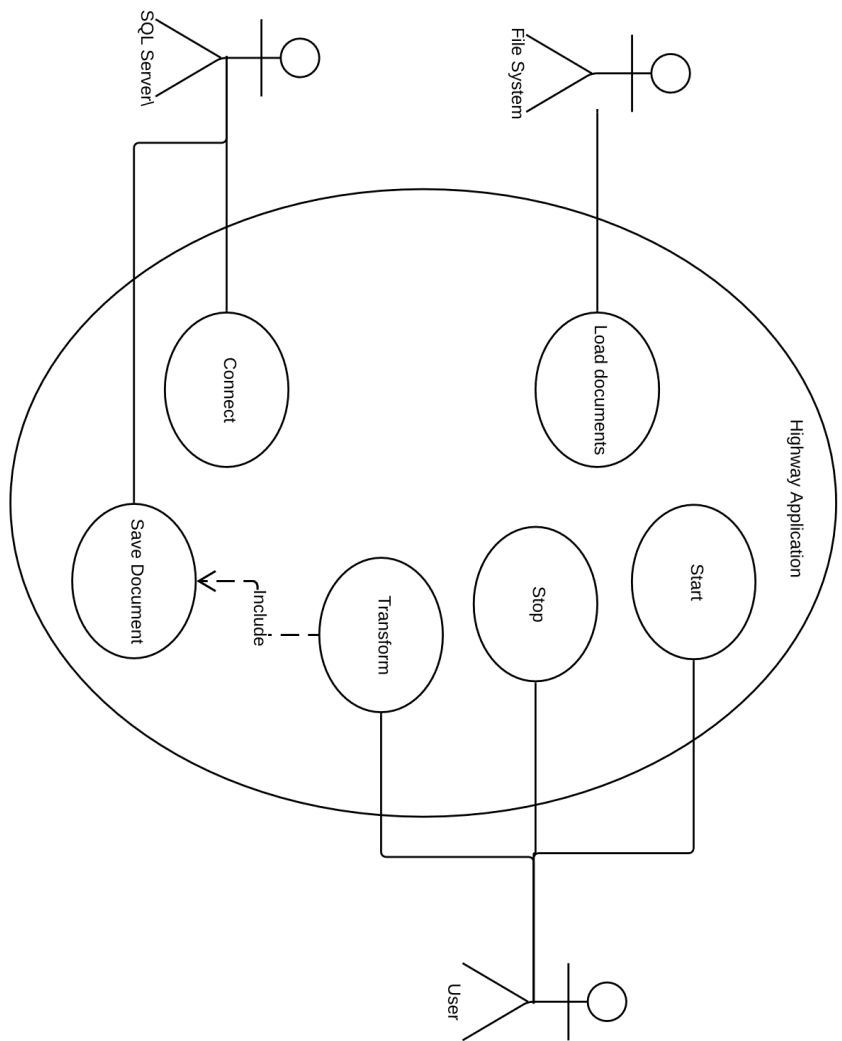
Herzien use case 1

ID	UC1
Use Case	User – Start
Beschrijving	De user start het programma.
Primaire actor	User
Secundaire actor	n.v.t.
Pre-conditie	Het programma is nog niet gestart.
Main flow	1. User opent de .exe van het programma 2. De applicatie start 3. De applicatie laat het start scherm zien 4. De applicatie laat een lijst met schepen zien
Post-conditie	Applicatie is geopend en het start scherm, met een lijst met vessels, wordt getoond.
Alternatieve flow	n.v.t.

Herzien use case 6

ID	UC6
Use Case	File System - Load Document
Beschrijving	Document dat geselecteerd is door de user wordt in het systeem geladen
Primaire actor	File System
Secundaire actor	User
Pre-conditie	Programma is gestart en er is een document geselecteerd
Main flow	1. Systeem lokaliseert het geselecteerde bestand in de File System 2. Systeem laad dit document in
Post-conditie	Documenten zijn beschikbaar voor transformatie.
Alternatieve flow	n.v.t.

Bijlage 9. Diagrammen



Usecase Diagram - OLD1

Mick Remmerswaal | May 29, 2019



**Smit
Lamnalco**

Description

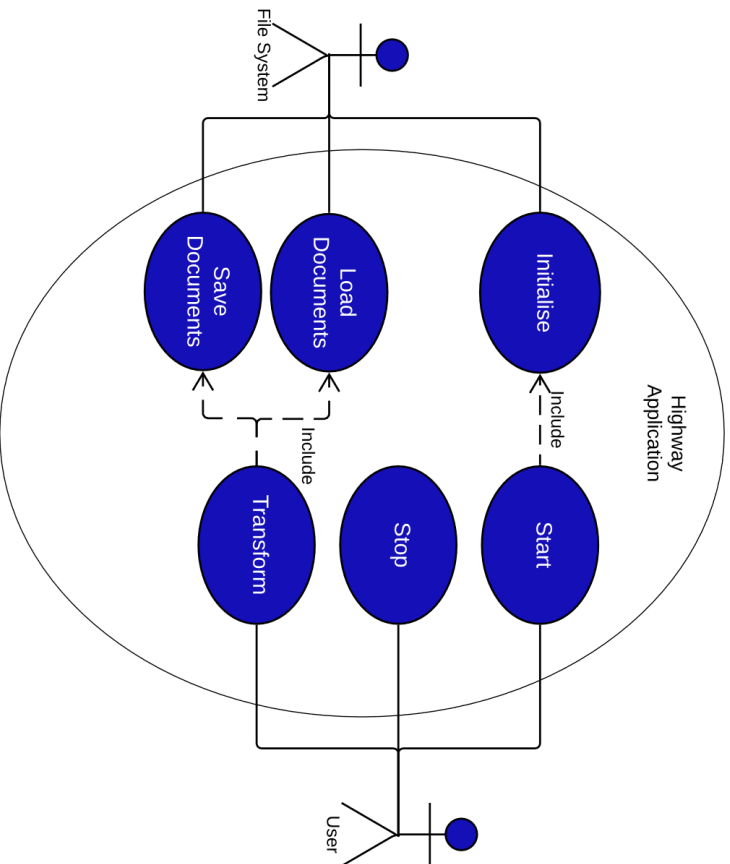
Usecase Diagram
SL Data Highway

The system communicates with
three outside actors
- The user of the system
- An instance of SQL Server
- A Document repository

All actors have different
influences on the system as
shown in the diagram

Usecase Diagram - OLD2

Mick Remmerswaal | May 29, 2019



**Smit
Lamnalco**

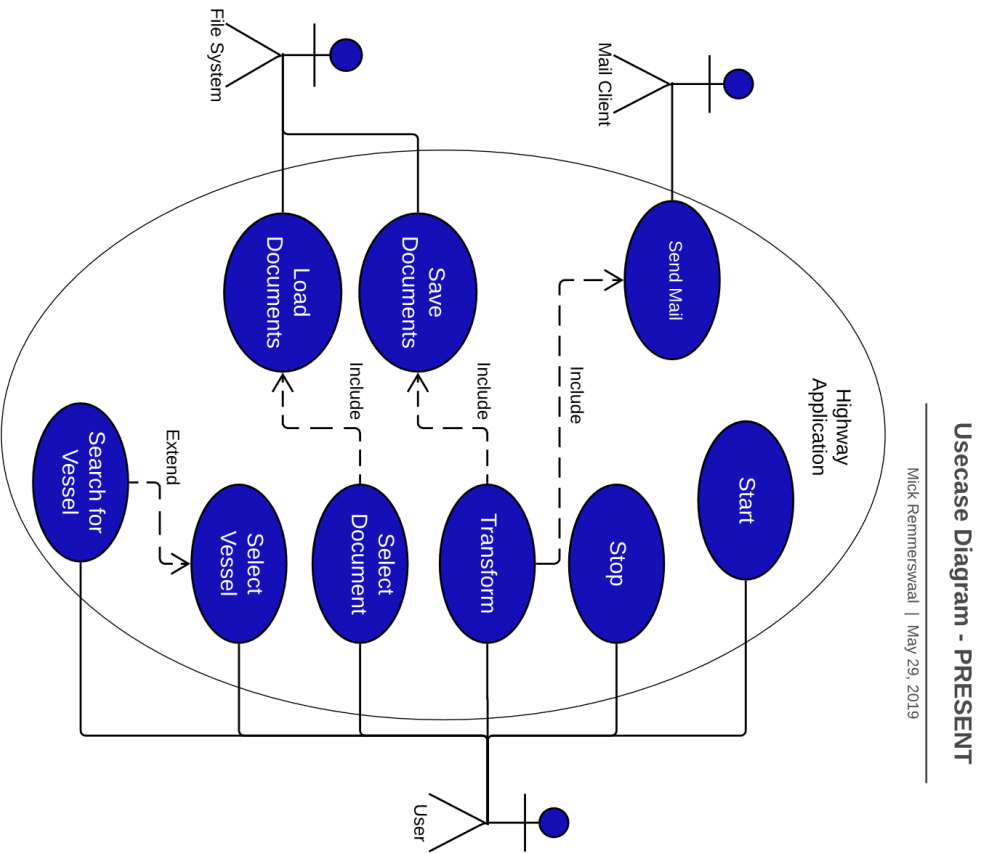
Description

Usecase Diagram
SL Data Highway

The user has different influences on the system as shown in the diagram

As well the document system of the target PC is of influence on the system





**Smit
Lamnalco**

Description

Usecase Diagram
SL Data Highway

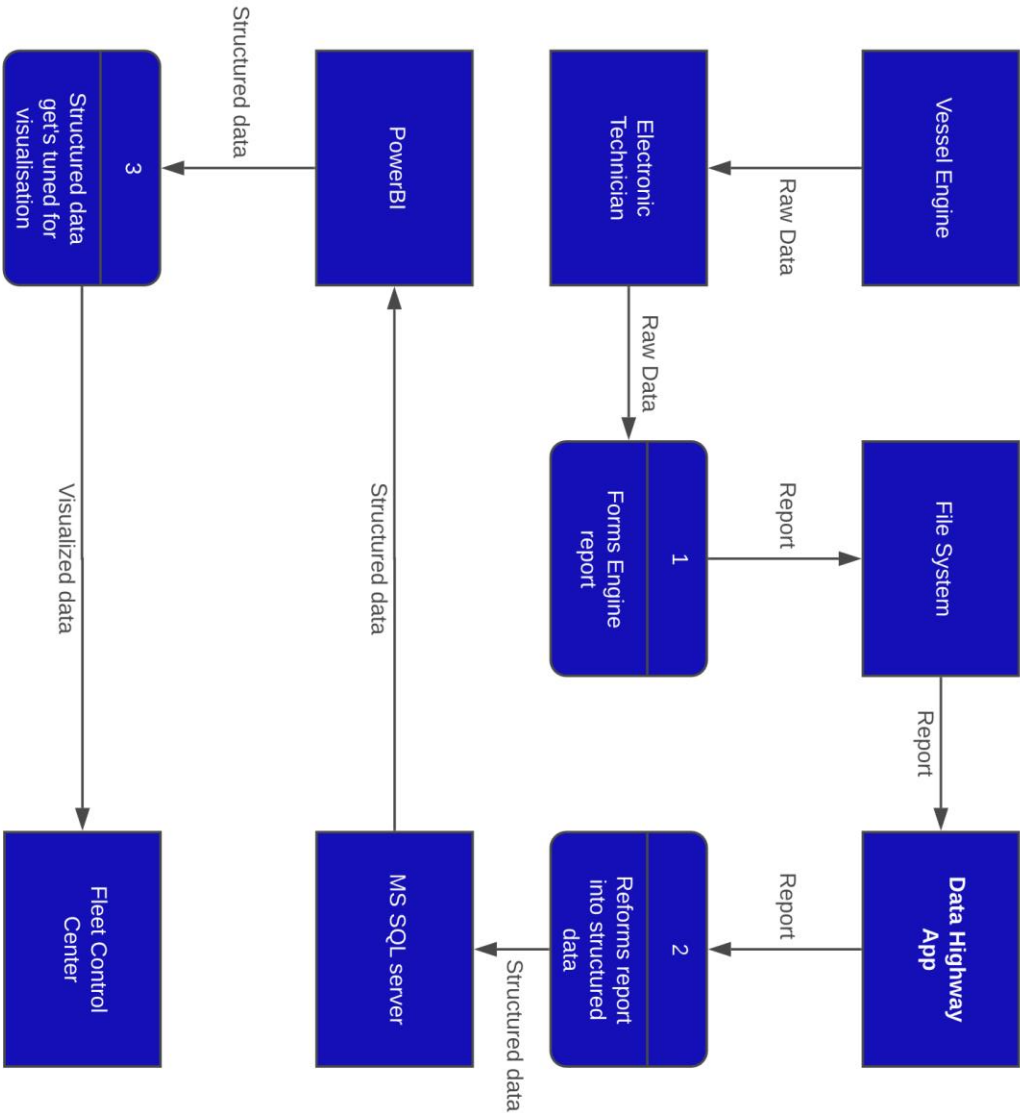
The system communicates with the user through several different methods

The actor has different influences on the system as shown in the different Use Cases

With the redesign to a GUI application some influences have changed

Data Transfer Design - OLD

Mick Remmerswaal | May 29, 2019



Description

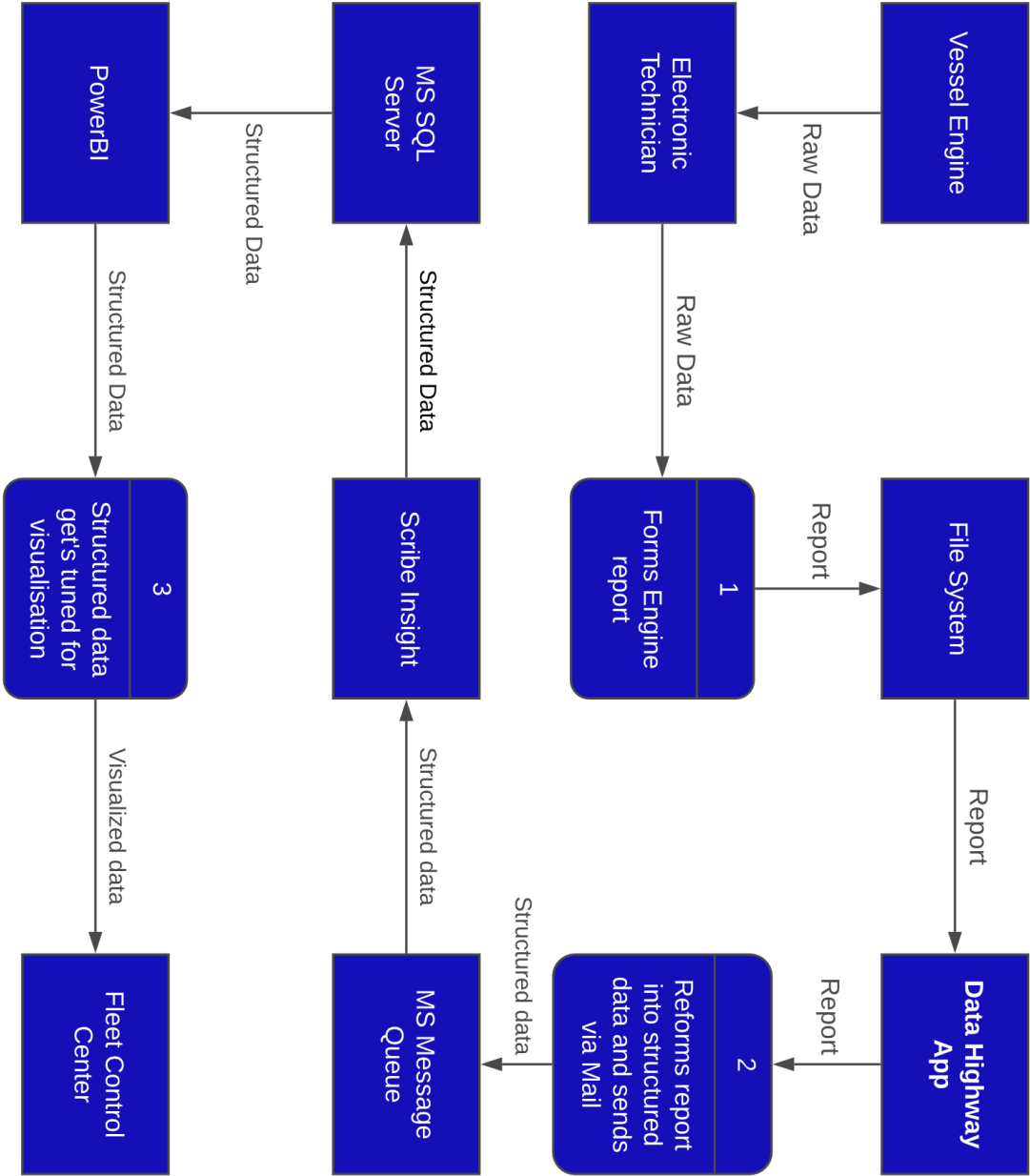
Data Transfer Design
SL Data Highway
Overview of the SL Data Highway as a whole



Smit
Lamnalco

Data Transfer Diagram - PRESENT

Mick Remmerswaal | May 29, 2019



Iteratie 2

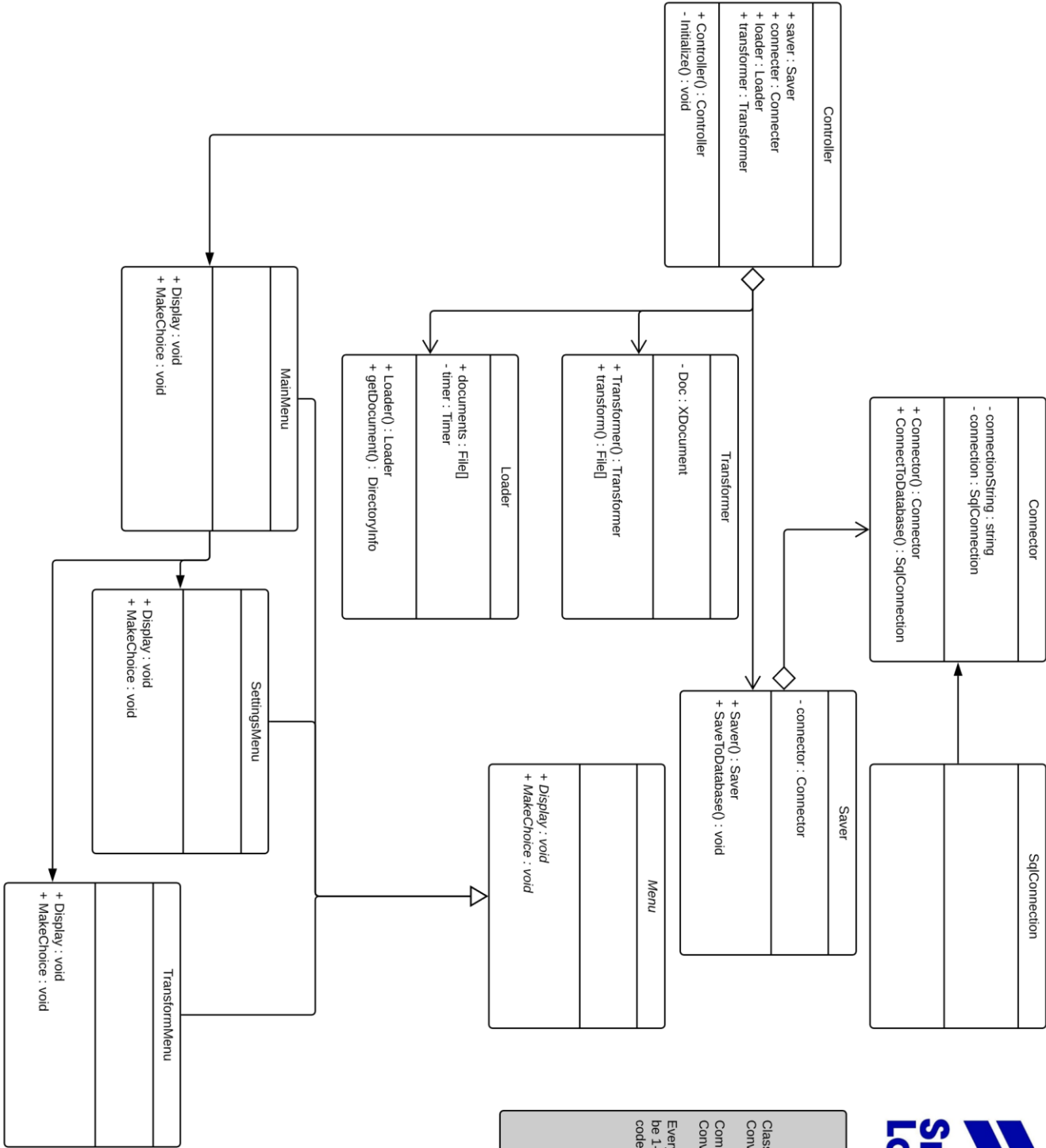


**Smit
Lamnalco**

Description

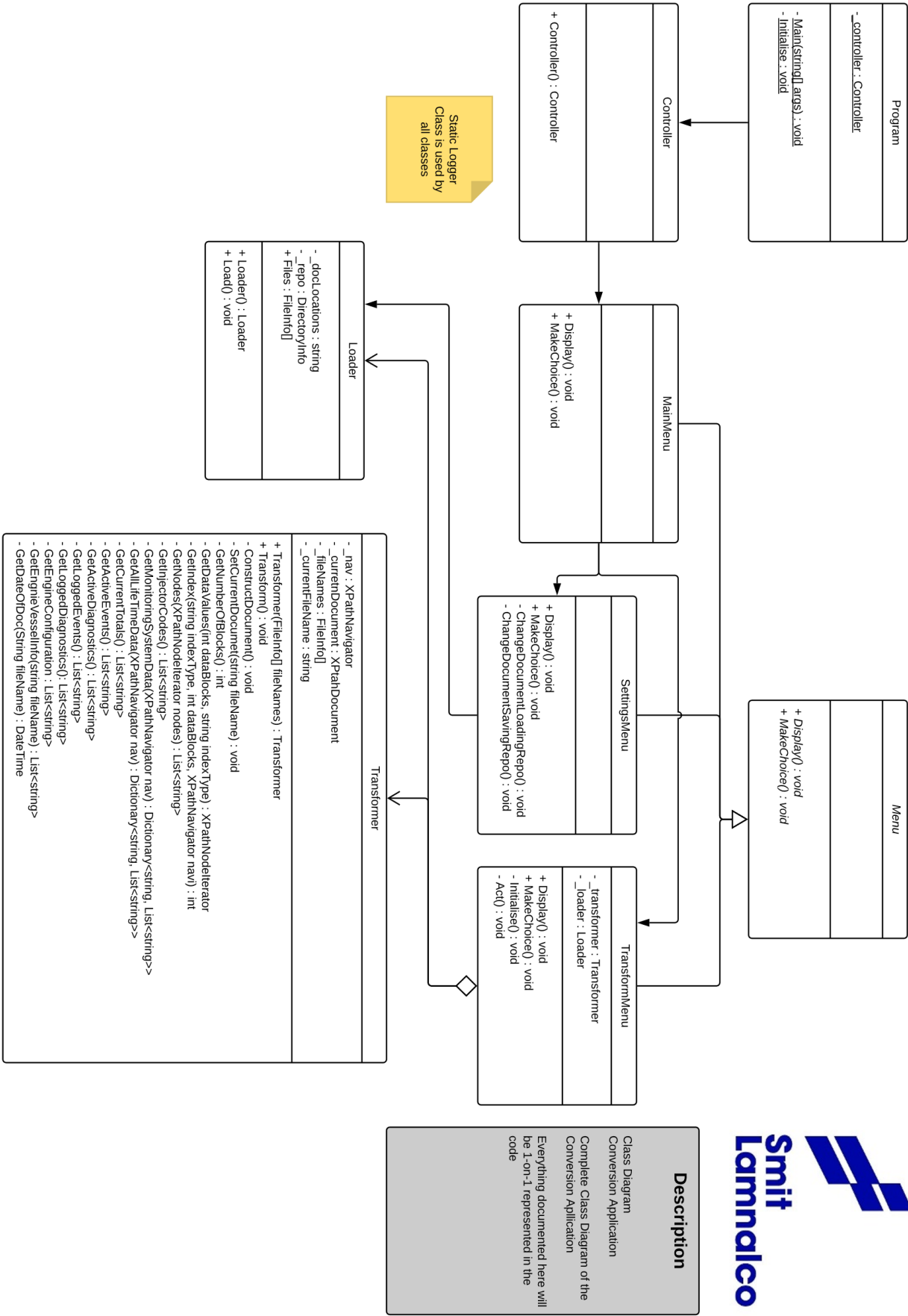
Data Transfer Diagram
This diagram reflects the flow of data within the whole of the project.
Starts at the vessel level and ends at the control center.

Class Diagrams
Iteratie 1



Description

Class Diagram
Conversion Application
Complete Class Diagram of the
Conversion Application
Everything documented here will
be 1-on-1 represented in the
code



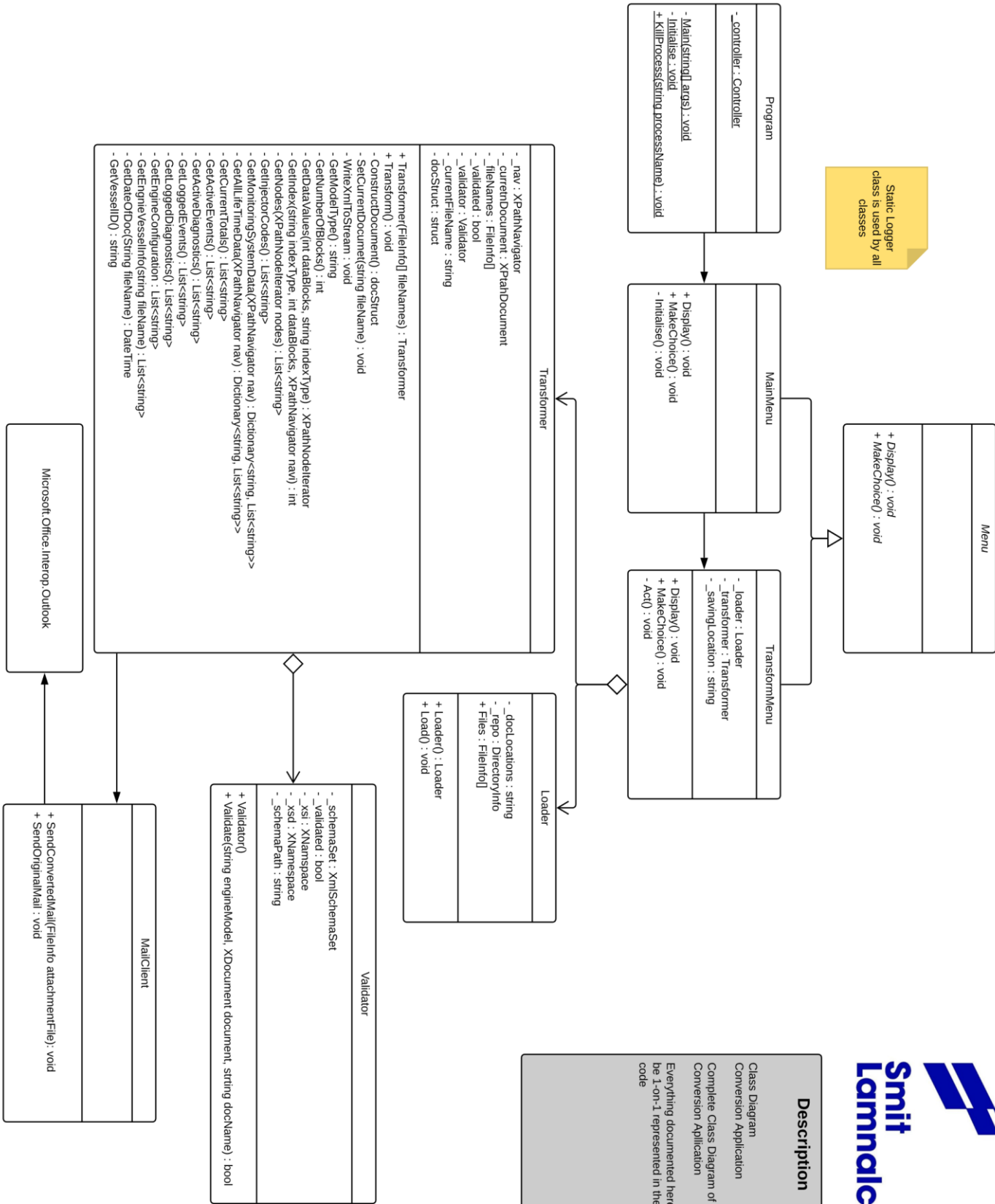


**Smit
Lamnalco**

Description

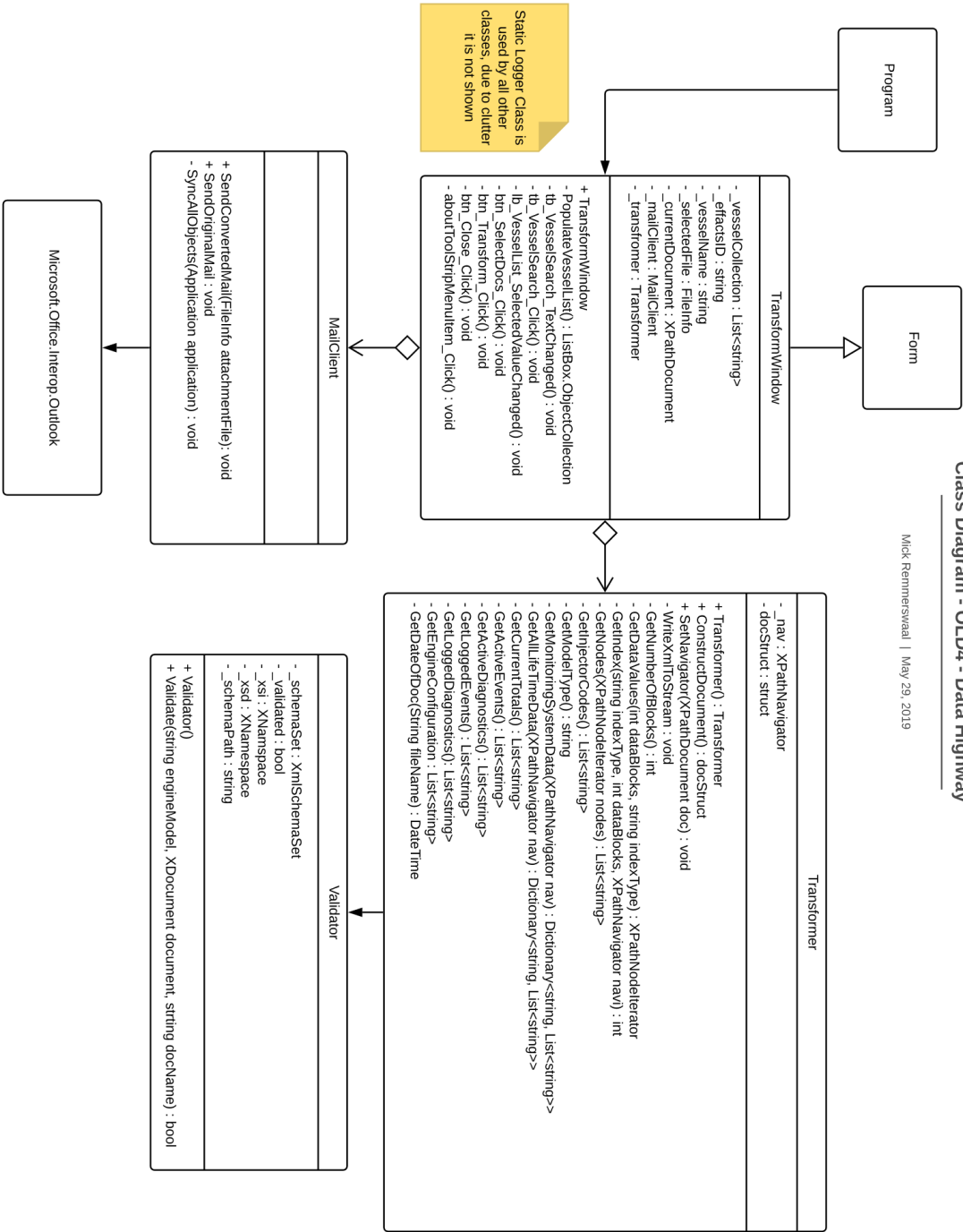
Class Diagram
Conversion Application
Complete Class Diagram of the
Conversion Application
Everything documented here will
be 1-on-1 represented in the
code

Static Logger
class is used by all
classes



Class Diagram - OLD4 - Data Highway

Mick Remmeswaal | May 29, 2019



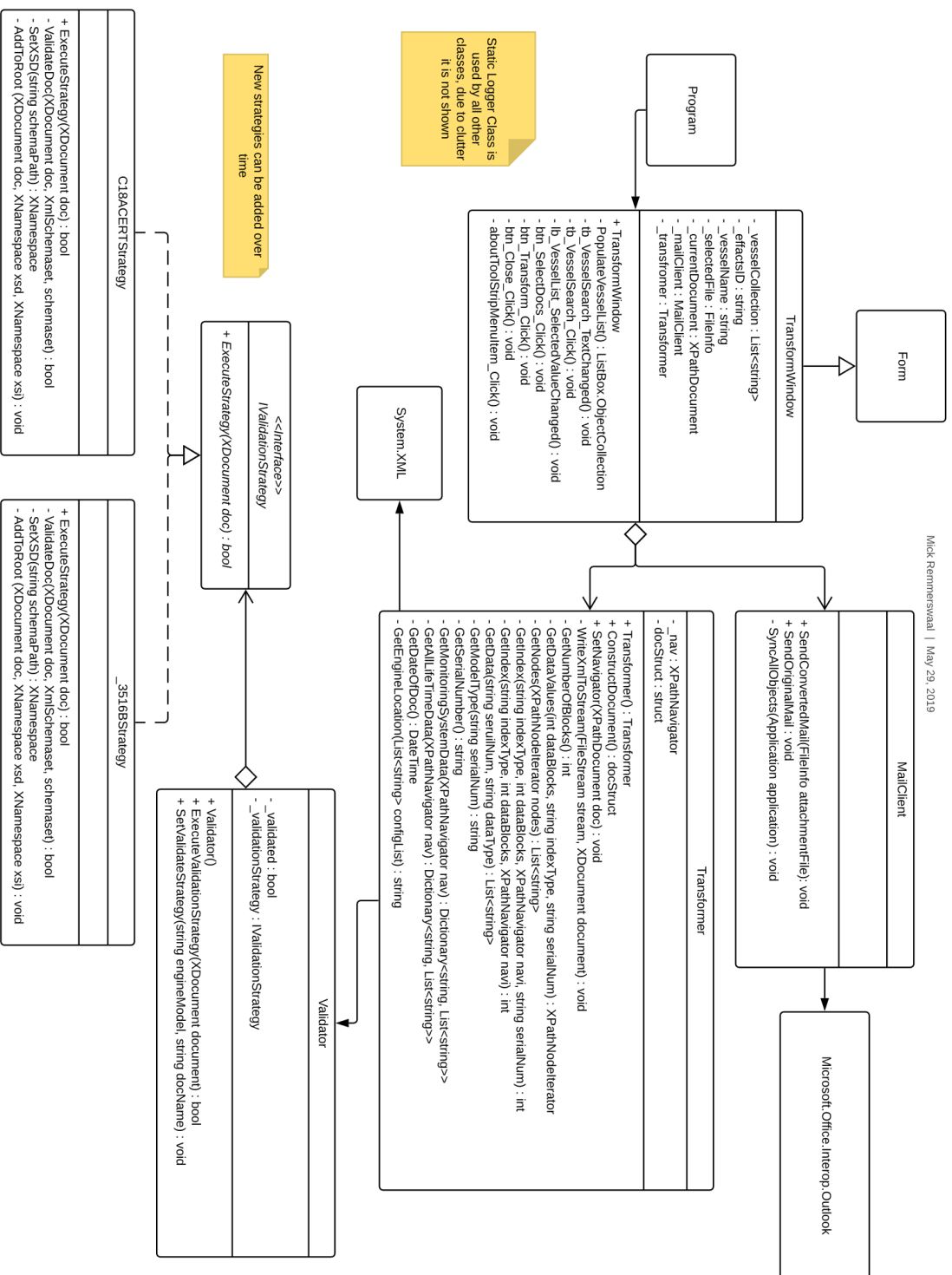
Smit
Lamnalco

Description

Class Diagram
Conversion Application
Complete Class Diagram of the
Conversion Application
Everything documented here will
be 1-on-1 represented in the
code

Class Diagram - PRESENT - Data

Mick Remmerswaal | May 29, 2019



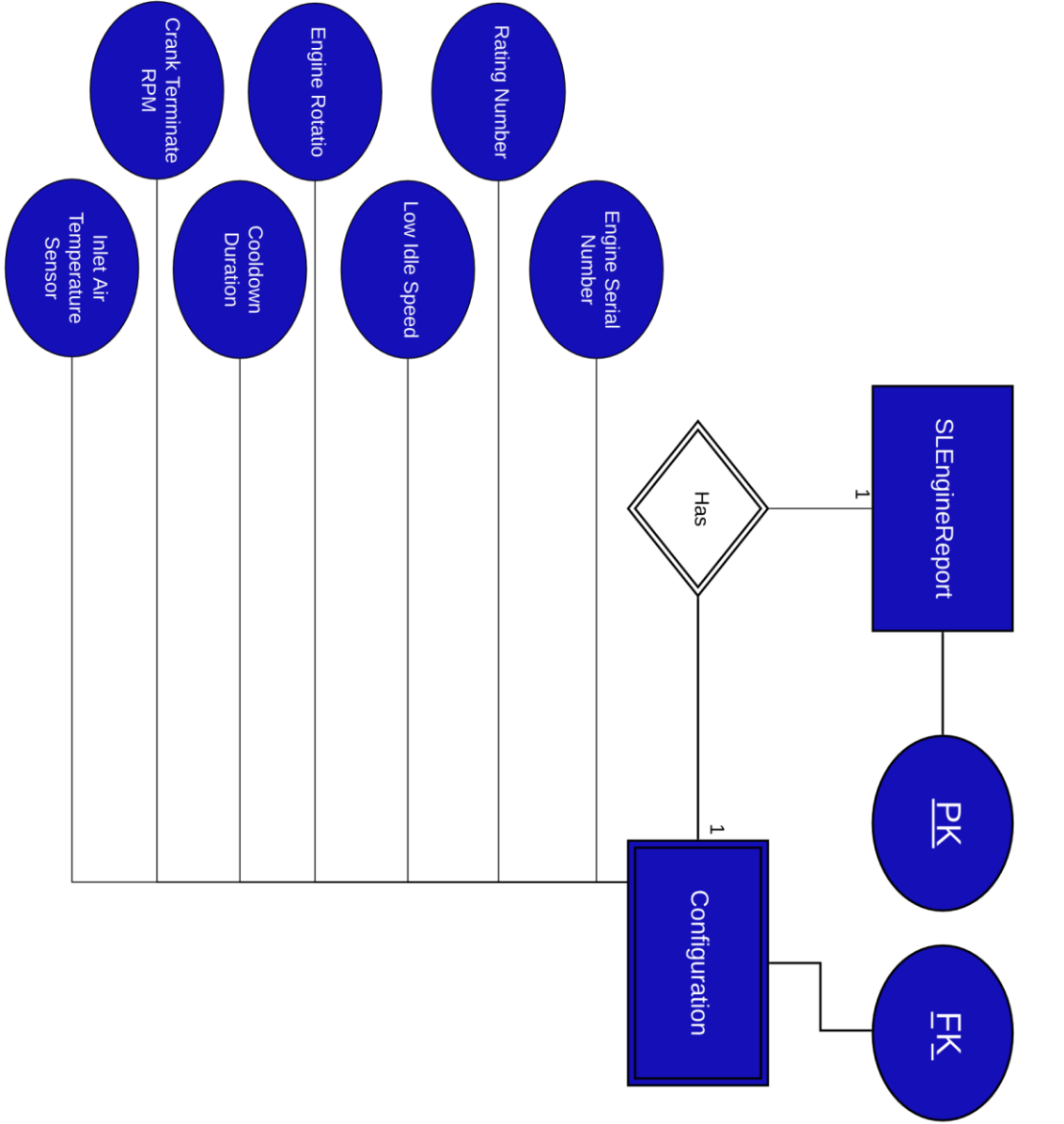
Description

Class Diagram
Conversion Application
Complete Class Diagram of the
Conversion Application
Everything documented here will
be 1-on-1 represented in the
code

Iteratie 5

Entity Relationship Diagram - OLD

Mick Remmerswaal | May 29, 2019



**Smit
Lamnalco**

Description

Entity Relationship Diagram
Database Diagram

This diagram is a simple
overview of how the database
structure is implemented

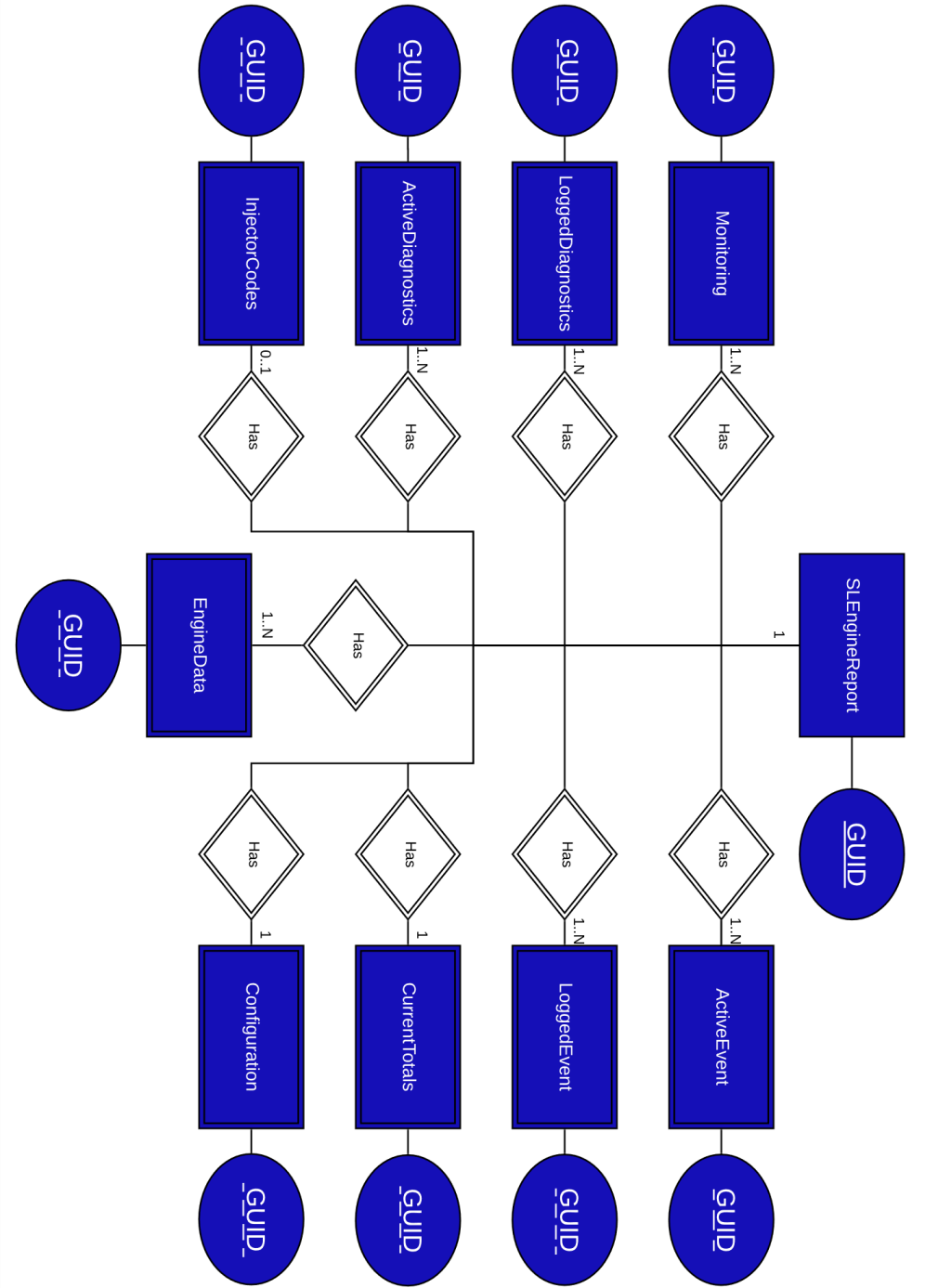
As there are way too many
attributes, they are not shown in
this design

Entity Relationship Diagram

Versie 1

Entity Relationship Diagram - PRESENT

Mick Remmerswaal | May 29, 2019



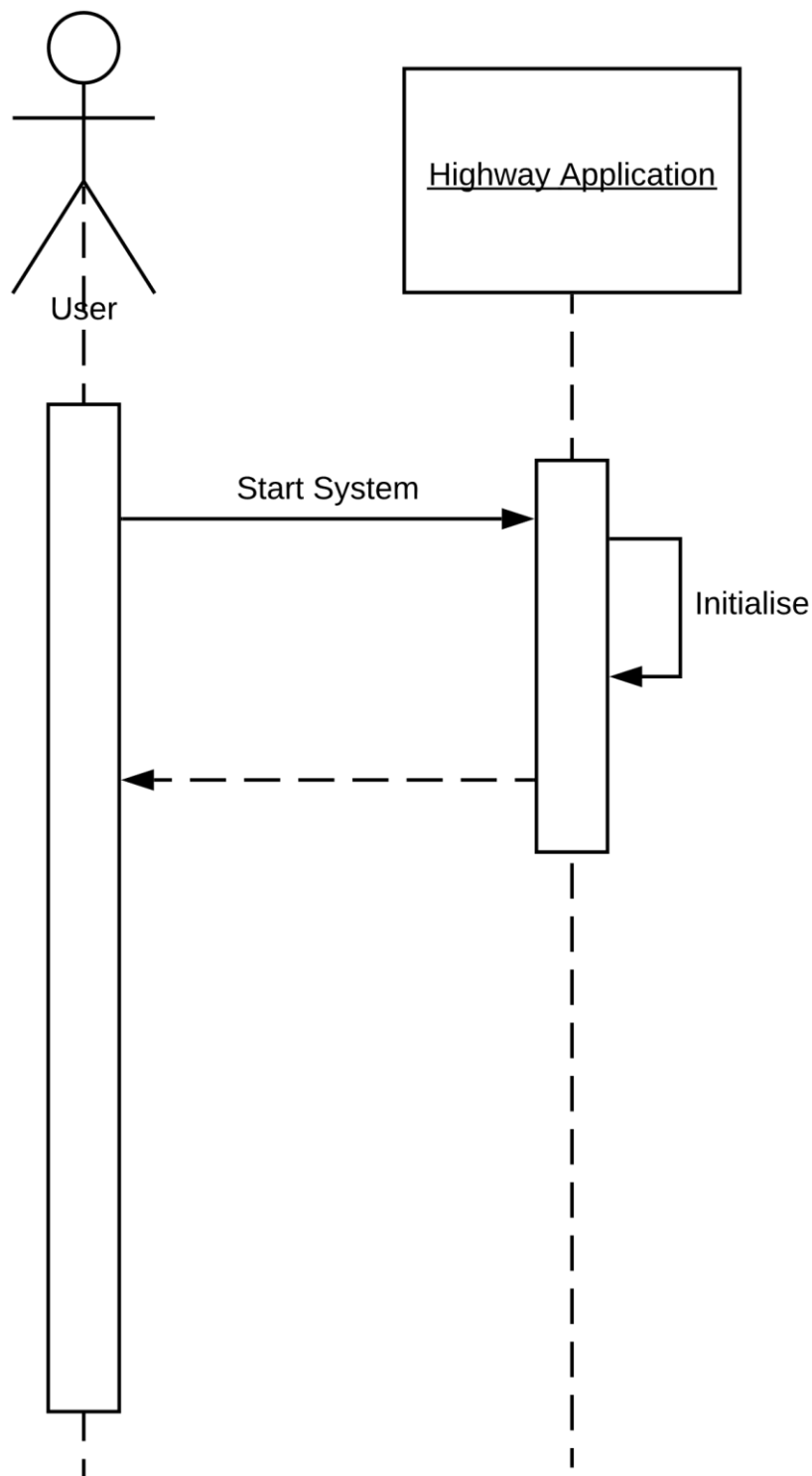
**Smit
Lamnalco**

Description

Entity Relationship Diagram
Database Diagram
This diagram is a simple
overview of how the database
structure is implemented
As there are way too many
attributes, they are not shown in
this design

User - Start - Sequence - OLD1

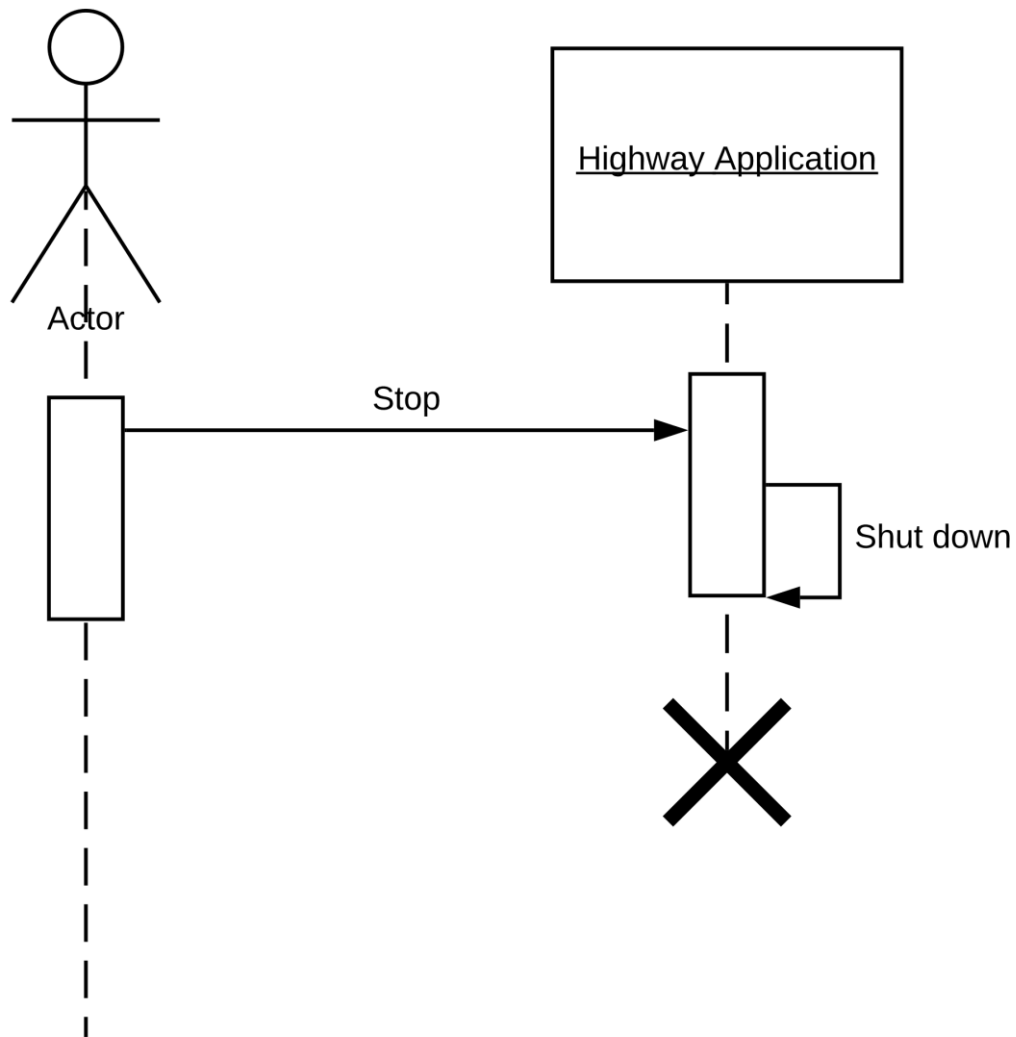
Mick Remmerswaal | May 29, 2019



Stop

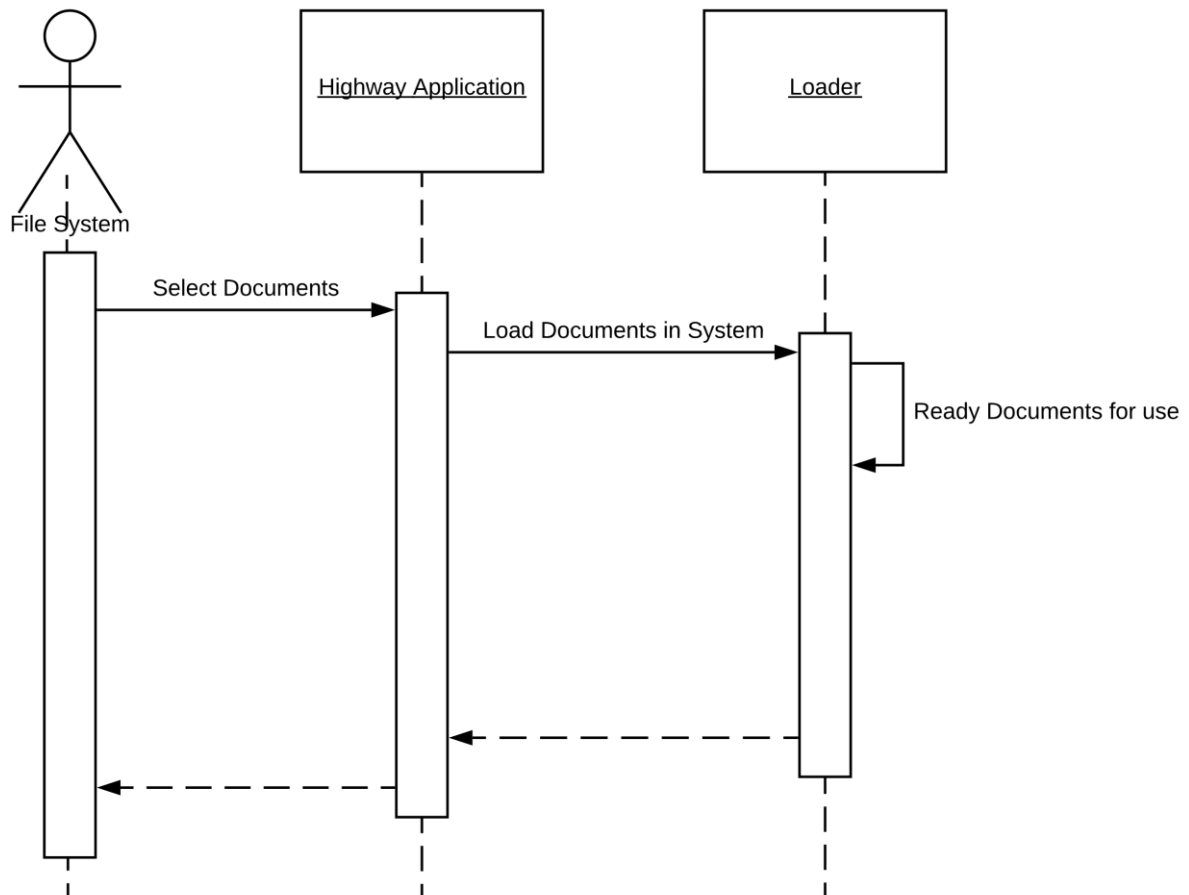
User - Stop - Sequence - OLD1

Mick Remmerswaal | May 29, 2019



File System - Load Document Sequence - OLD1

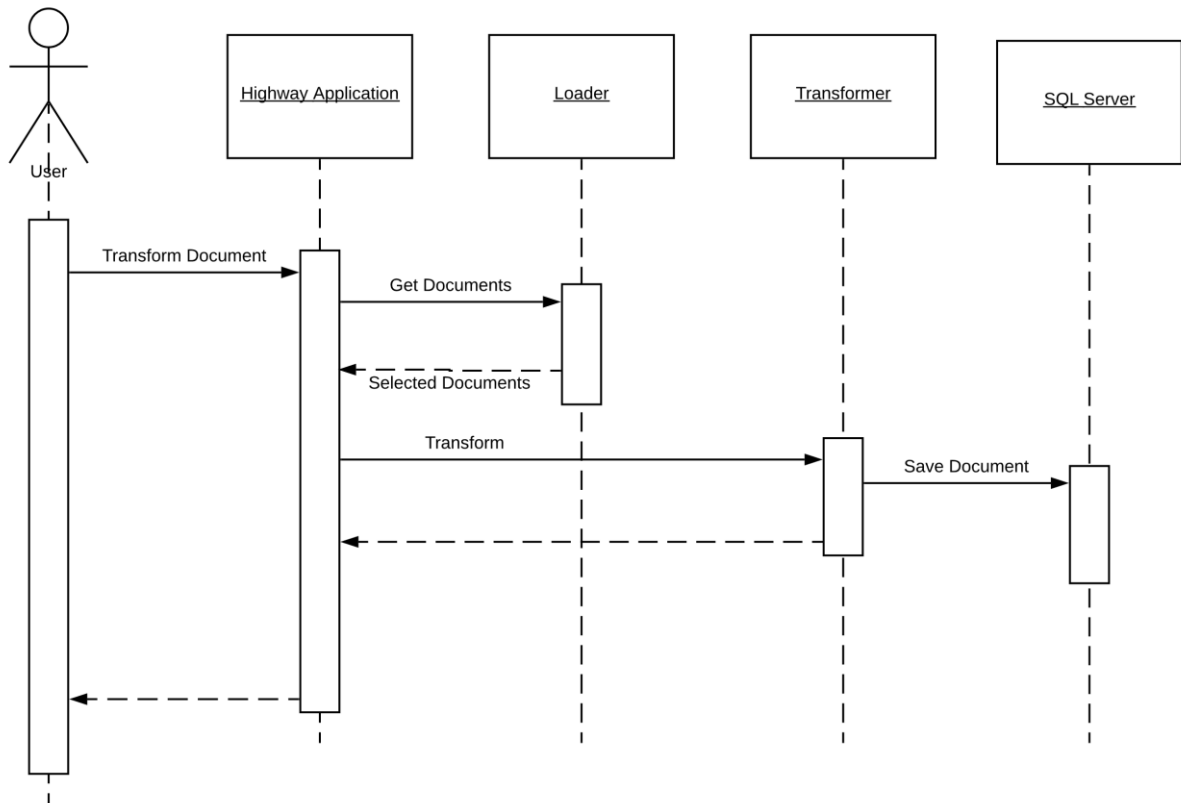
Mick Remmerswaal | May 29, 2019



Transform

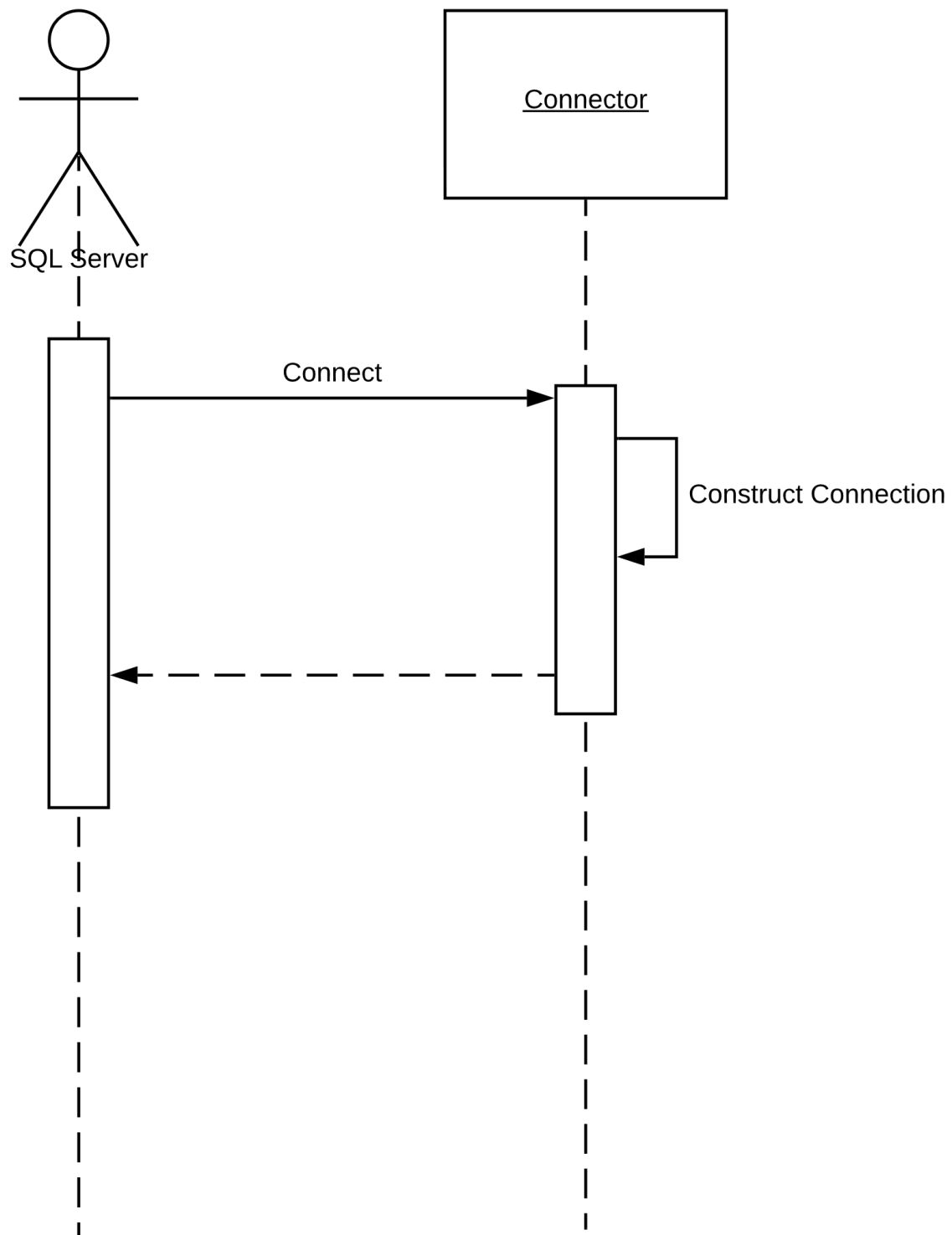
User - Transform - Sequence - OLD1

Mick Remmerswaal | May 29, 2019



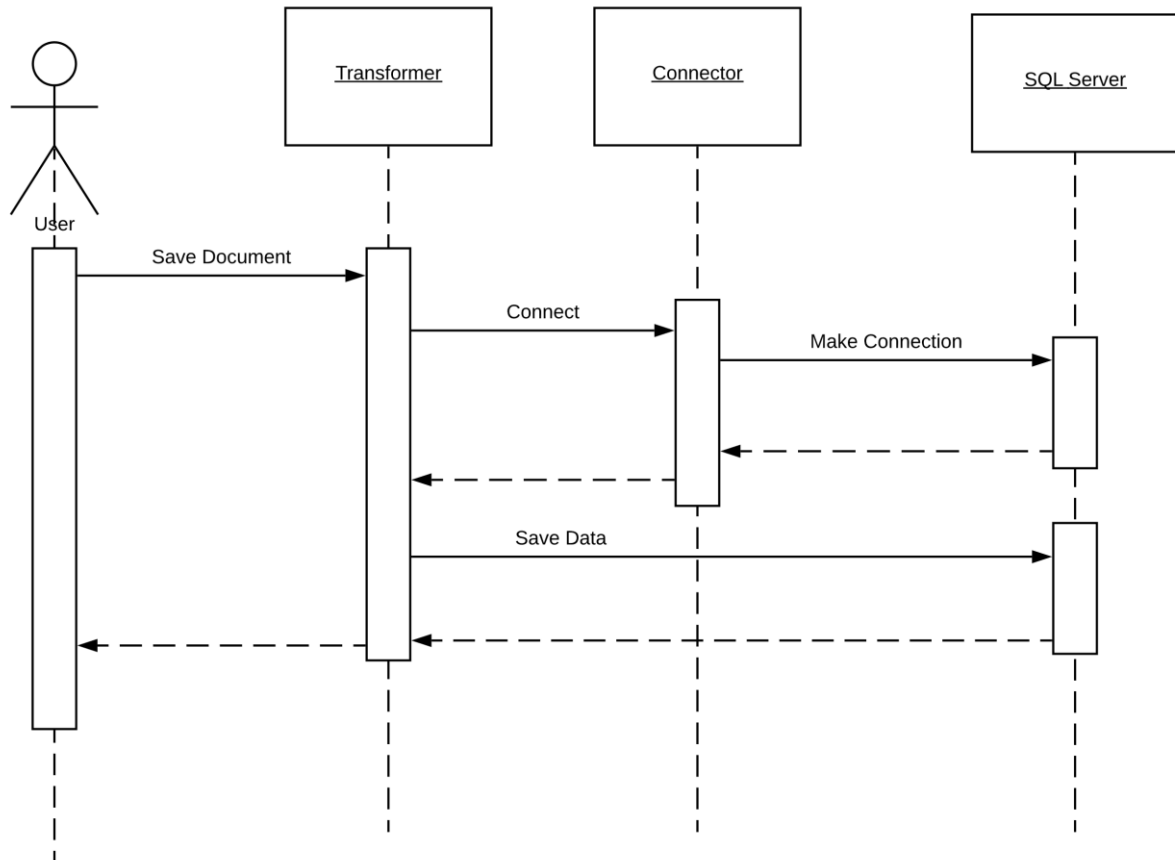
SQL Server - Connect - Sequence - OLD1

Mick Remmerswaal | May 29, 2019



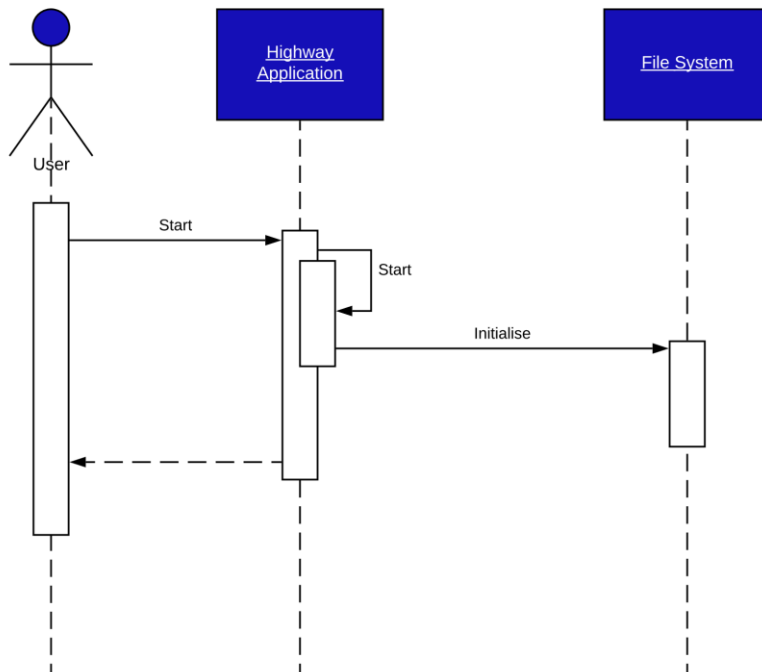
User - Save Document - Sequence - OLD1

Mick Remmerswaal | May 29, 2019



Iteratie 2

Start



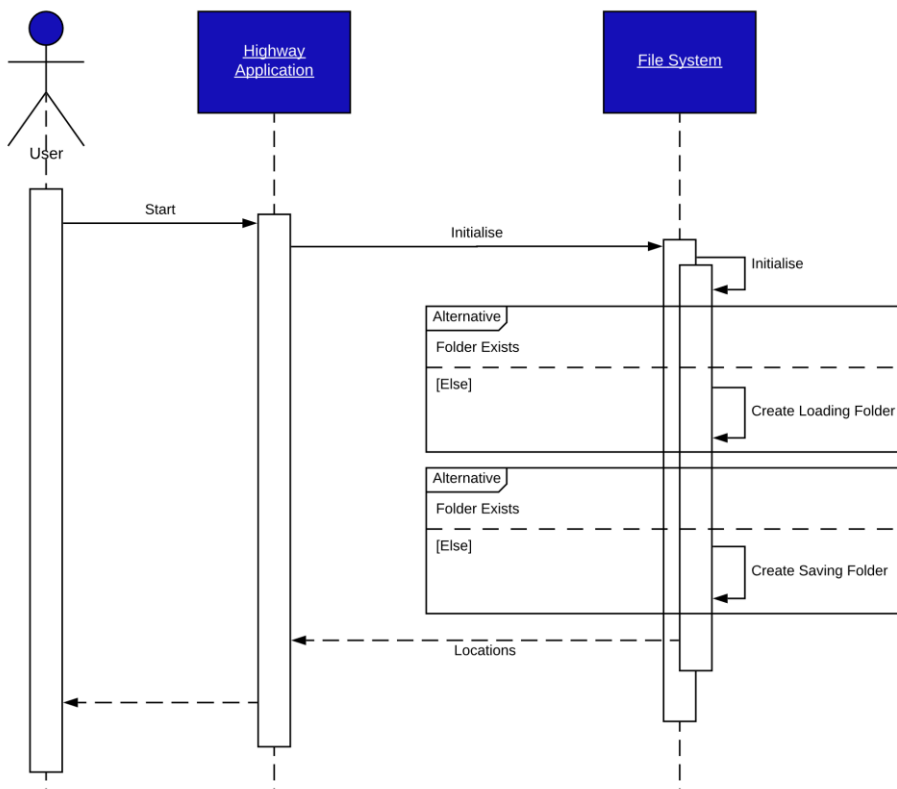
Description

Sequence Diagram
SL Data Highway

The user starts the application

The application starts itself and starts the initialise proces in the file system

Initialise



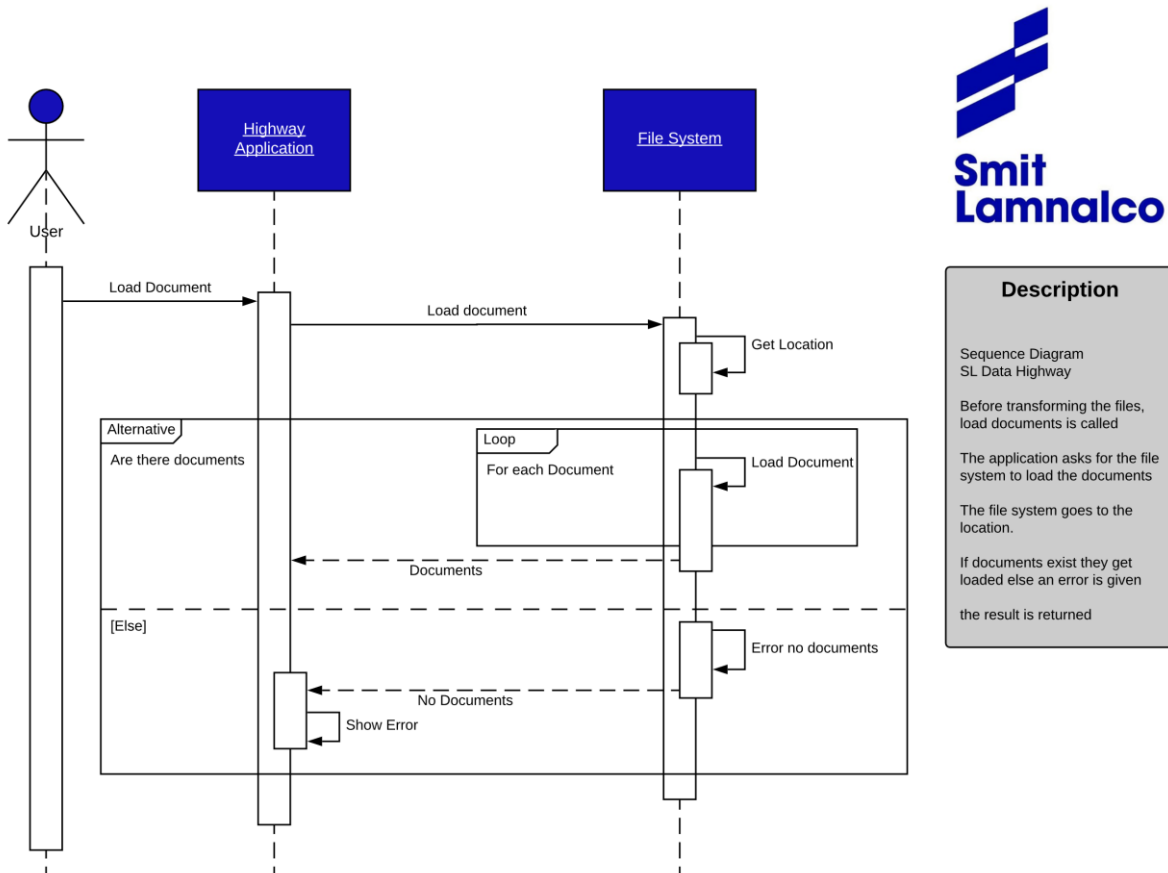
Description

Sequence Diagram
SL Data Highway

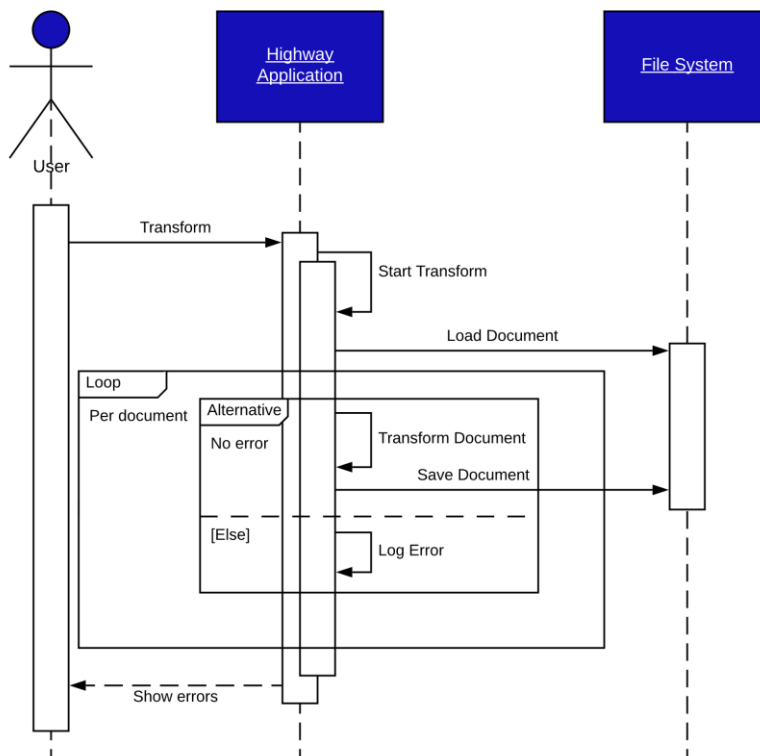
After starting the application, initialise is called on the File System

The file system creates two folders, one for loading and one for saving

Load Documents



Transform



Description

Sequence Diagram
SL Data Highway

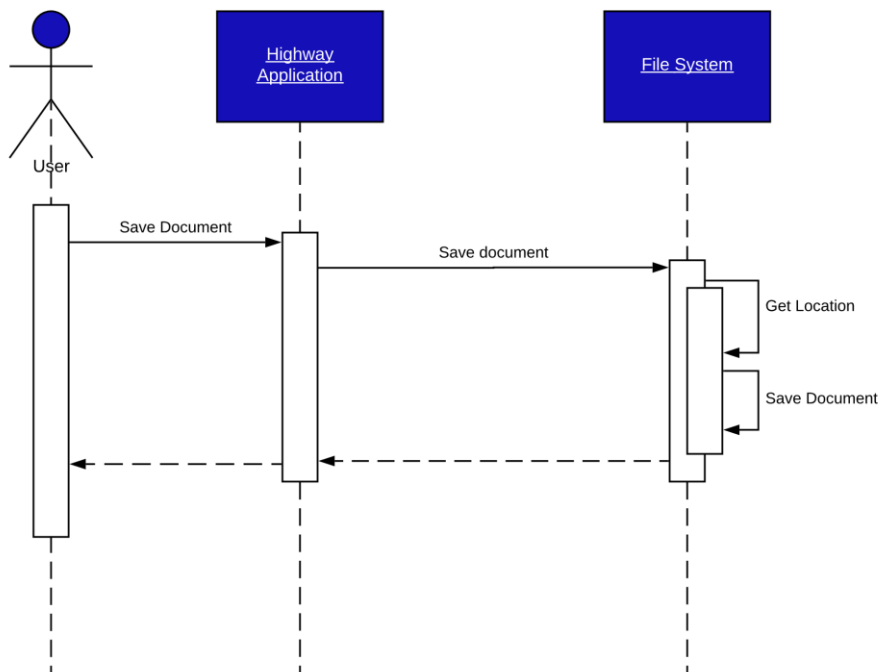
The user starts the transform process

The application starts the transform process and loads the documents

The system loops through the documents while transforming and saving them

Shows error if something went wrong

Save Documents



Description

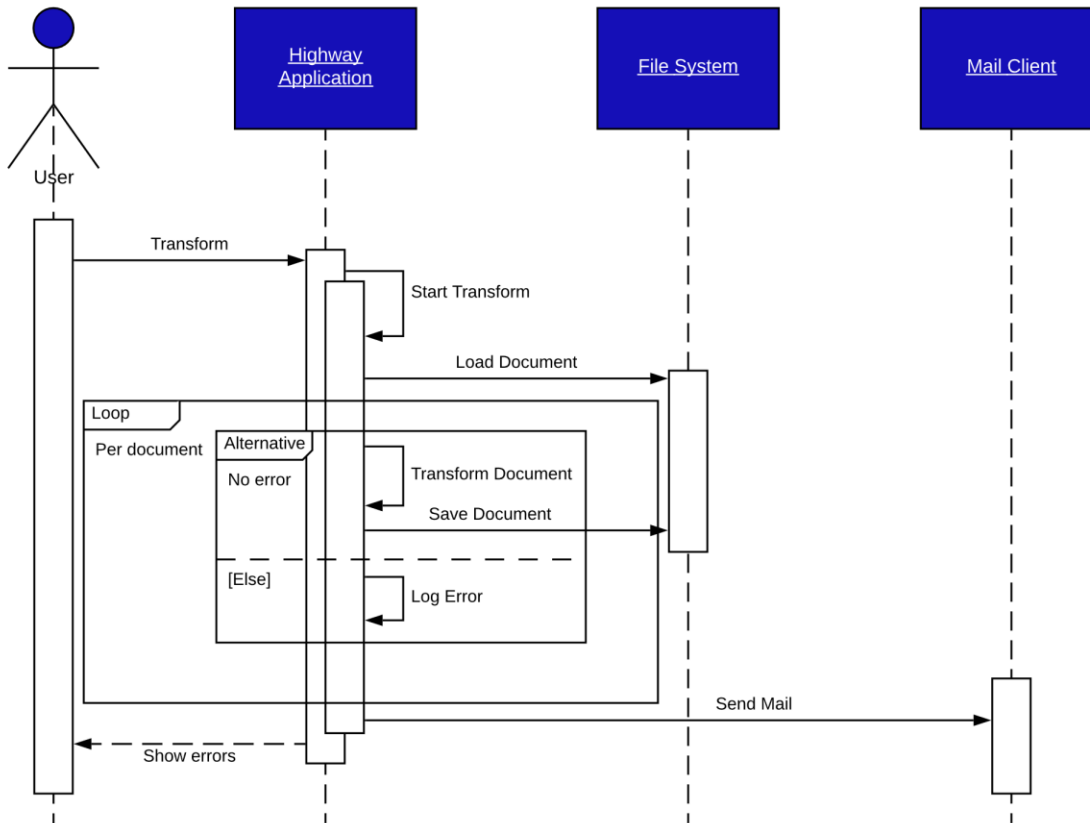
Sequence Diagram
SL Data Highway

Afer transform the application saves the document

The file system gets the saving location

The document gets saved to this location

Iteratie 3
Transform



Description

Sequence Diagram
SL Data Highway

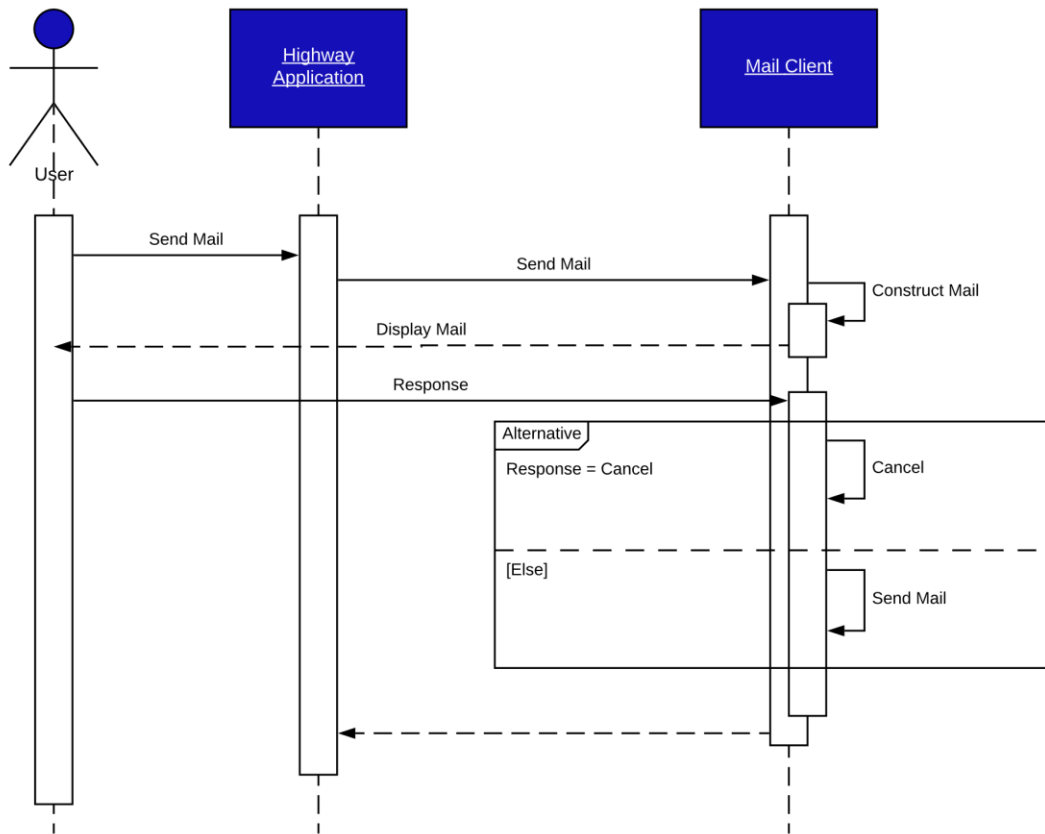
The user starts the transform process

The application starts the transform process and loads the documents

The system loops through the documents while transforming and saving them

Shows error if something went wrong

Send Mail



Description

Sequence Diagram
SL Data Highway

After Transform is called, Send Mail is called

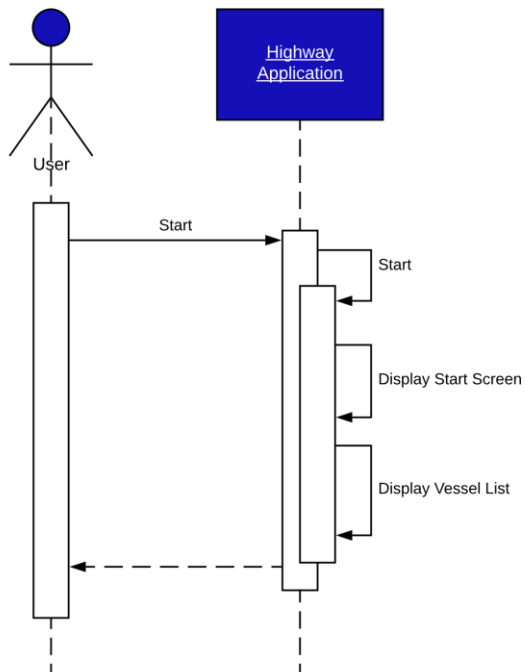
The Mail Client constructs a mail

The Mail Client displays the mail

According to the user's choice the mail is sent or not

Iteratie 4

Start



Description

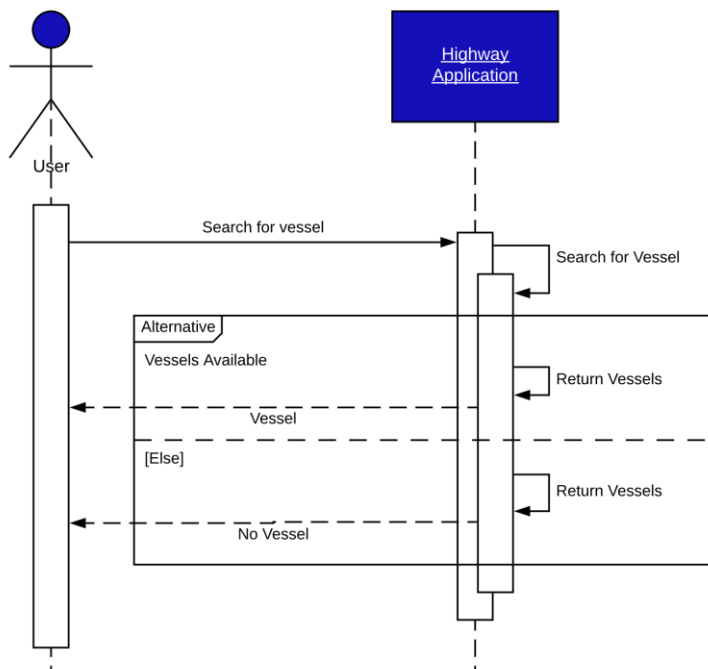
Sequence Diagram
SL Data Highway

The user starts the application

The application starts and displays a startscreen.

The start screen displays a vessel list

Search Vessel



Description

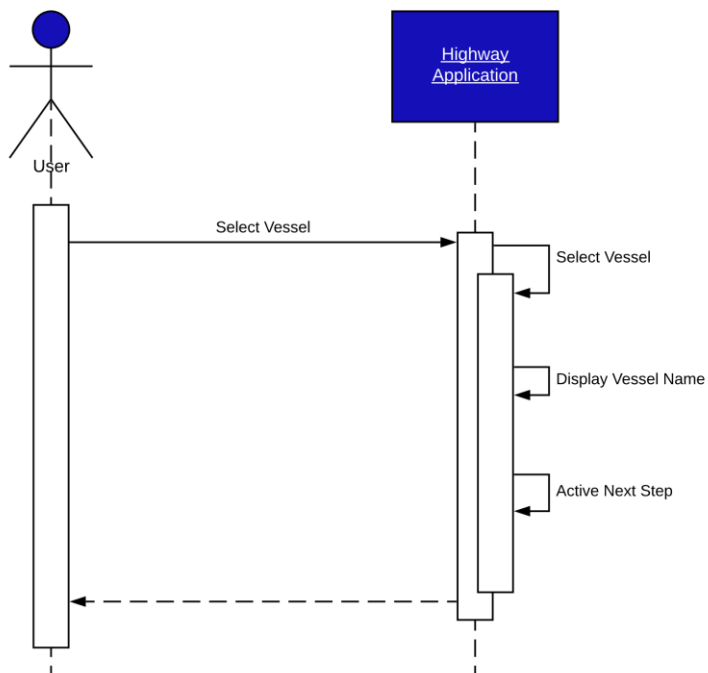
Sequence Diagram
SL Data Highway

The user starts the application

The application starts and displays a startscreen.

The start screen displays a vessel list

Select Vessel



Description

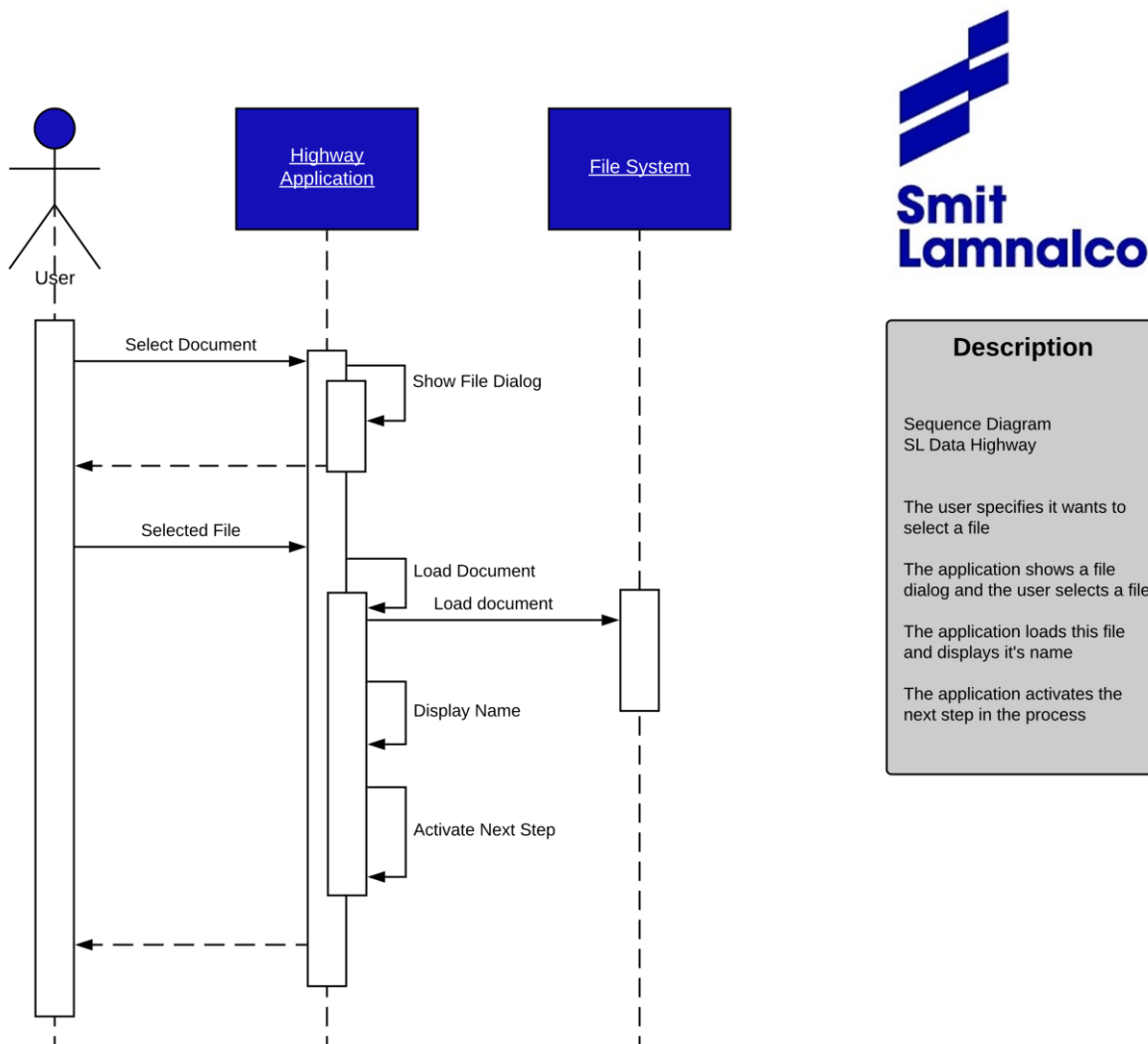
Sequence Diagram
SL Data Highway

The user starts the application

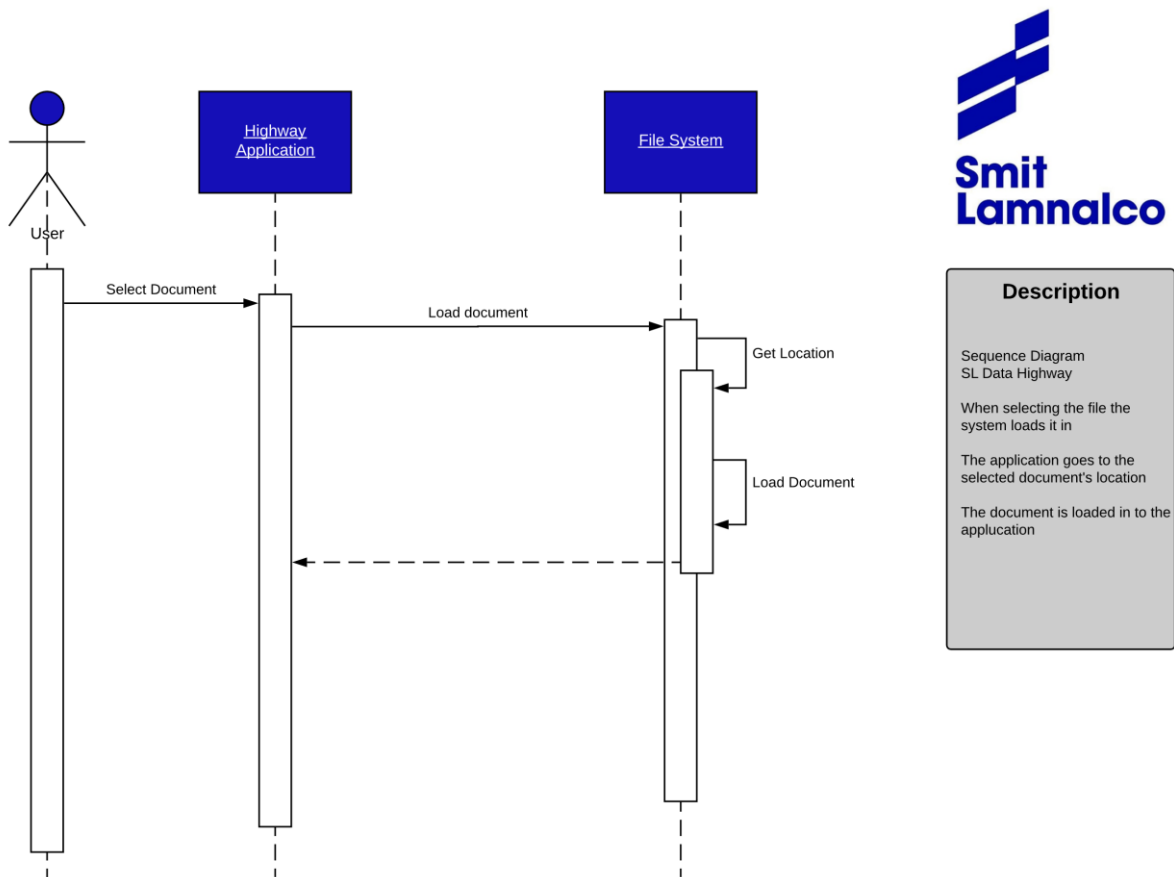
The application starts and displays a startscreen.

The start screen displays a vessel list

Select Document

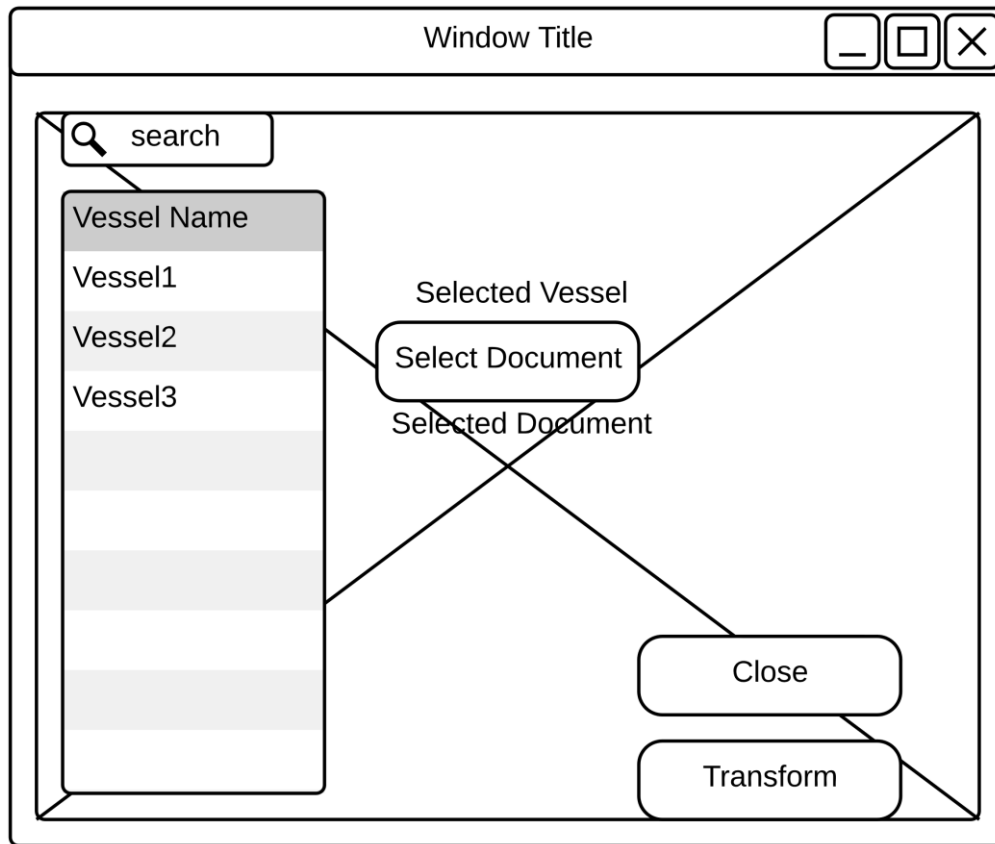


Load Document

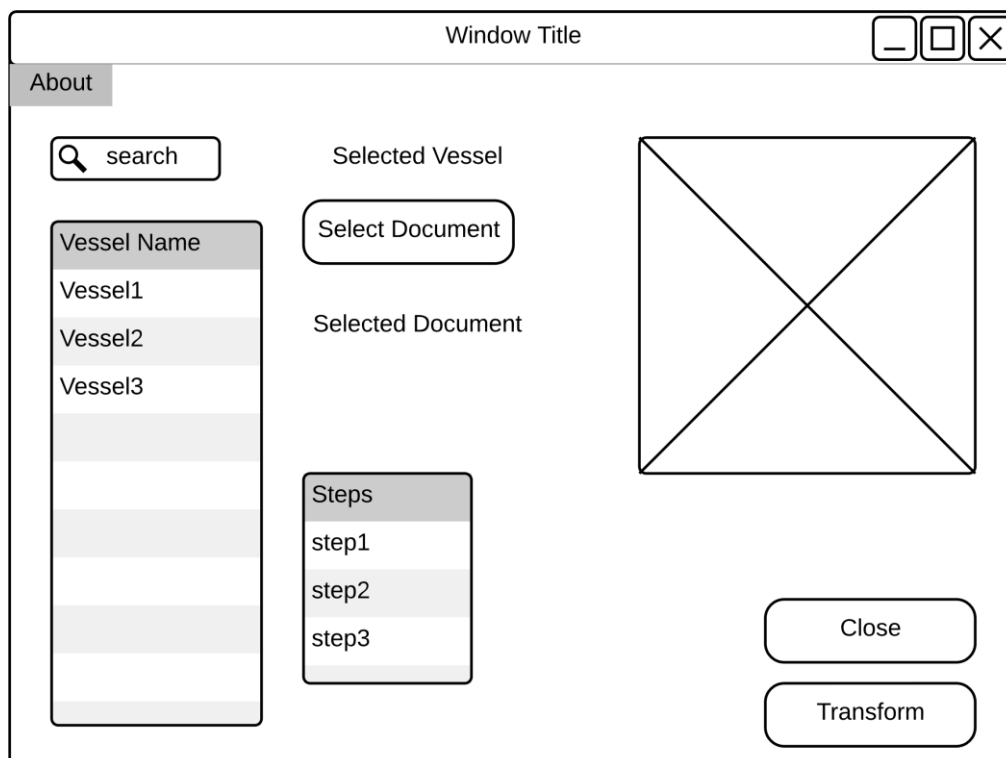


UI Design Wireframe

Versie 1



Versie 2



Cat Electronic Technician 2015A v1.0

Product Status Report

3/6/2019 1:42 PM

Product Status Report

Parameter	Value
Engine Serial Number	TTJ00555
Equipment ID	NOT PROGRAMMED
Comments	

3516B Port Backup (TTJ00555)

Parameter	Value
Equipment ID	NOT PROGRAMMED
Engine Serial Number	TTJ00555
ECM Serial Number	0274B025JP
Personality Module Part Number	4326858-00
Personality Module Release Date	JUN13
Personality Module Description	3516 Marine Certified

Logged Diagnostic Codes [Diagnostic Clock = 7560 hours] - 3516B Port Backup (TTJ00555)

Code	Description	Occ.	First	Last
No Logged Diagnostic Codes				

Logged Event Codes [Diagnostic Clock = 7560 hours] - 3516B Port Backup (TTJ00555)

Code	Description	Occ.	First	Last
No Logged Events				

Active Diagnostic Codes - 3516B Port Backup (TTJ00555)

Code	Description
No Active Codes	

Current Totals - 3516B Port Backup (TTJ00555)

Description	Value	Unit
Total Time	7560	hours

Configuration - 3516B Port Backup (TTJ00555)

Description	Value	Unit
Equipment ID	NOT PROGRAMMED	
Engine Serial Number	TTJ00555	
ECM Serial Number	0274B025JP	

Personality Module Part Number	4326858-00	
Personality Module Release Date	JUN13	
Engine Rotation	Invalid Data	
Engine Location	Port	
Low Idle Speed	650	rpm
Air Shutoff	Unavailable	
Engine Speed Droop	Unavailable Parameter	%
Air Intake Shutoff Detection Installation Status	Not Installed	
Secondary Throttle Enable Status	Disabled	
Number of Synchronized Engines Configuration	1	
Governor Gain Factor #1	7864	
Governor Minimum Stability Factor #1	79	
Governor Maximum Stability Factor #1	734	
FLS	0	
FTS	0	
Total Tattletale	7	

PL1000E #2

Parameter	Value
ECM Part Number	3461903-01
ECM Serial Number	3463B037TE
Personality Module Part Number	3764410-00
Personality Module Release Date	MAR11
Personality Module Description	PL1000E Communication ECU

Current Totals - PL1000E #2

Description	Value	Unit
ECM Lifetime Clock	34150	hours

Configuration - PL1000E #2

Description	Value	Unit
CAT Data Link Module Identifier Configuration	PL1000 #2	
Ethernet Media Access Control (MAC) Address	00:0A:75:04:84:37	
Internet Protocol Host Name	PL1000E	
Internet Protocol (IPv4) Address	192.168.1.1	
Internet Protocol (IPv4) Network Mask	255.255.255.0	
Network Default Gateway Internet Protocol (IPv4) Address	0.0.0.0	
Web Server Administrator Password	admin	
Dynamic Host Configuration Protocol (DHCP) Enable Status	Disabled	
Modbus #1 Port Selection	Serial Port #1	
ModBus #1 Baud Rate	9600	
ModBus #1 Echo/ No Echo	No Echo	
ModBus #1 Flow Control	None	
ModBus #1 Data Size	8	

ModBus #1 Parity	No Parity	
Modbus #2 Port Selection	Serial Port #2	
ModBus #2 Baud Rate	9600	
ModBus #2 Echo/ No Echo	No Echo	
ModBus #2 Flow Control	None	
ModBus #2 Data Size	8	
ModBus #2 Parity	No Parity	
Modbus #3 Port Selection	Serial Port #4	
ModBus #3 Baud Rate	9600	
ModBus #3 Echo/ No Echo	No Echo	
ModBus #3 Flow Control	None	
ModBus #3 Data Size	8	
ModBus #3 Parity	No Parity	
RS-485 Mode	2-wire	
Modbus TCP Data Link Address	254	

Engine Room MPD - Port Engine

Parameter	Value
ECM Part Number	307-754205
ECM Serial Number	0154G005MP
Software Group Part Number	3942605-00
Software Group Release Date	JAN 12
Software Group Description	Mini MPD/MPD

Configuration - Engine Room MPD - Port Engine

Description	Value	Unit
Display Location	Engine Room	
Engine Location	Port	

3516B Port (TTJ00555)

Parameter	Value
Equipment ID	SNMR T10 PORT
Engine Serial Number	TTJ00555
ECM Serial Number	0274B077JP
Personality Module Part Number	4326857-00
Personality Module Release Date	JUN13
Personality Module Description	3516 Marine Certified

Logged Diagnostic Codes [Diagnostic Clock = 7560 hours] - 3516B Port (TTJ00555)

Code	Description	Occ.	First	Last
No Logged Diagnostic Codes				

Logged Event Codes [Diagnostic Clock = 7560 hours] - 3516B Port (TTJ00555)

Code	Description	Occ.	First	Last
No Logged Events				

Active Diagnostic Codes - 3516B Port (TTJ00555)

Code	Description
No Active Codes	

Current Totals - 3516B Port (TTJ00555)

Description	Value	Unit
Total Time	7560	hours
Total Fuel	434092	L
Total Idle Time	5099:45	hours
Total Idle Fuel	245537	L
Total Max Fuel	299104	L
Lifetime Total Engine Revolutions	345696476	rev
Percentage Idle Time	67.45	%
Total Operating Hours	7561	hours
Average RPM	762.12	rpm
Average Fuel Rate	57.42	L/h
Engine Starts	1999	
Overall Load Factor	100	%
Definite Period Average Percent Engine Load Factor	6.7	%
Total Engine Rated Speed Operating Hours	5	hours
Percentage Rated Engine Speed Time	0.07	%

Configuration - 3516B Port (TTJ00555)

Description	Value	Unit
Equipment ID	SNMR T10 PORT	
Engine Serial Number	TTJ00555	
ECM Serial Number	0274B077JP	
Personality Module Part Number	4326857-00	
Personality Module Release Date	JUN13	
Rating Number	2	
Rated Power	2525 kW at 1800 rpm	
Rated Peak Torque	13400 N-m at 1800 rpm	
Rated Engine Speed	1800 rpm	
Engine Rotation	Invalid Data	
Engine Location	Port	
Fuel Ratio Control Offset	0	
Rated Fuel Position	28.200	mm
Low Idle Speed	650	rpm
High Idle Speed	1944	rpm
Fuel Correction Factor	0	%
Engine Cooling System Config	SCAC (Separate Circuit Aftercooled)	
Cold Cylinder Cutout	Enabled	
Cooldown Speed	600	rpm
Cooldown Duration	5	min

Engine Pre-Lube Duration	20	sec
Crank Duration	10	sec
Maximum Number of Crank Cycles	3	
Crank Terminate RPM	400	rpm
Secondary Throttle Enable Status	Enabled	
Number of Synchronized Engines Configuration	1	
Torque Limit Enable Status	Disabled	
Maximum Engine Torque Limit	16458	N-m
Minimum Engine Torque Limit	10081	N-m
Coolant Warning Engine Load Enable Threshold	40	%
Programmable Droop Feature Installation Status	Not Installed	
Engine Speed Droop	Unavailable Parameter	%
Transmission Oil Temperature Configuration	Not Installed	
Transmission Oil Pressure Sensor Configuration	Not Installed	
Fuel Level Sensor Configuration	Not Installed	
Coolant Level Sensor Configuration	Not Installed	
Air Intake Shutoff Detection Installation Status	Not Installed	
Aftercooler Coolant Level Sensor Installation Status	Not Installed	
Inlet Air Temperature Sensor	Installed	
Engine Oil Temperature Sensor Installation Status	Installed	
Engine Oil Level Sensor Installation Status	Not Installed	
Governor Gain Factor #1	7864	
Governor Minimum Stability Factor #1	79	
Governor Maximum Stability Factor #1	734	
Engine Oil Renewal System Installation Status	Not Installed	
Engine Oil Renewal System Enable Status	Disabled	
Oil Renewal Rate	0	
FLS	0	
FTS	0	
Total Tattletale	18	
Prog Eng Load Fac	Off	

Lifetime: Time vs Engine Speed - 3516B Port (TTJ00555)

Engine Speed(rpm)	hours	%
<400	2.15	0.03
400-499	0.50	0.01
500-599	64.35	0.85
600-699	5120.85	67.59
700-799	326.60	4.31
800-899	339.65	4.48
900-999	1019.70	13.46

1000-1099	158.00	2.09
1100-1199	179.50	2.37
1200-1299	140.00	1.85
1300-1399	160.10	2.11
1400-1499	25.30	0.33
1500-1599	11.10	0.15
1600-1699	12.00	0.16
1700-1799	8.95	0.12
1800-1899	7.35	0.10
1900-1999	0.05	0.00
>2000	0.00	0.00

Lifetime: Time vs Engine Load Factor - 3516B Port (TTJ00555)

Engine Load Factor(%)	hours	%
<5	5691.65	75.13
5-9	220.45	2.91
10-14	117.50	1.55
15-19	721.55	9.52
20-24	188.25	2.48
25-29	84.35	1.11
30-34	93.10	1.23
35-39	92.30	1.22
40-44	133.40	1.76
45-49	112.65	1.49
50-54	56.35	0.74
55-59	31.95	0.42
60-64	12.40	0.16
65-69	3.85	0.05
70-74	1.85	0.02
75-79	1.45	0.02
80-84	3.50	0.05
85-89	3.40	0.04
90-94	0.20	0.00
95-99	0.20	0.00
>100	5.75	0.08

Lifetime: Time vs Right Exhaust Temperature - 3516B Port (TTJ00555)

Right Exhaust Temperature(Deg C)	hours	%
<450	7401.00	97.69
450-464	23.35	0.31
465-479	18.25	0.24
480-494	19.95	0.26
495-509	20.35	0.27
510-524	14.05	0.19
525-539	16.50	0.22
540-554	8.05	0.11
555-569	5.55	0.07

570-584	8.95	0.12
585-599	10.45	0.14
600-614	10.80	0.14
615-629	3.10	0.04
630-644	1.90	0.03
645-659	0.85	0.01
660-674	5.25	0.07
675-689	2.75	0.04
690-704	3.15	0.04
705-719	0.55	0.01
720-734	1.10	0.01
735-749	0.10	0.00
750-764	0.00	0.00
>765	0.00	0.00

Lifetime: Time vs Left Exhaust Temperature - 3516B Port (TTJ00555)

Left Exhaust Temperature(Deg C)	hours	%
<450	7303.75	96.41
450-464	48.70	0.64
465-479	57.25	0.76
480-494	21.05	0.28
495-509	18.55	0.24
510-524	20.90	0.28
525-539	20.20	0.27
540-554	18.60	0.25
555-569	12.65	0.17
570-584	7.65	0.10
585-599	10.80	0.14
600-614	11.05	0.15
615-629	8.50	0.11
630-644	2.15	0.03
645-659	1.30	0.02
660-674	3.75	0.05
675-689	2.25	0.03
690-704	2.10	0.03
705-719	3.30	0.04
720-734	0.40	0.01
735-749	1.05	0.01
750-764	0.10	0.00
>765	0.00	0.00

Lifetime: Time vs Unknown - 3516B Port (TTJ00555)

Unknown()	hours	%
<0	0.00	0.00
0-4	0.00	0.00
5-9	0.00	0.00
10-14	0.00	0.00

15-19	0.70	0.01
20-24	2.85	0.04
25-29	30.15	0.40
30-34	155.00	2.05
35-39	469.65	6.20
40-44	633.35	8.36
45-49	3492.50	46.10
50-54	2580.05	34.05
55-59	207.90	2.74
60-64	4.20	0.06
65-69	0.00	0.00
70-74	0.00	0.00
75-79	0.00	0.00
80-84	0.00	0.00
85-89	0.00	0.00
90-94	0.00	0.00
95-99	0.00	0.00
100-104	0.00	0.00
105-109	0.00	0.00
110-114	0.00	0.00
115-119	0.00	0.00
120-124	0.00	0.00
125-129	0.00	0.00
>130	0.00	0.00

Lifetime: Time vs Unknown - 3516B Port (TTJ00555)

Unknown()	hours	%
<0	0.00	0.00
0-4	0.00	0.00
5-9	0.00	0.00
10-14	0.00	0.00
15-19	0.25	0.00
20-24	1.50	0.02
25-29	10.70	0.14
30-34	89.55	1.18
35-39	411.55	5.43
40-44	634.10	8.37
45-49	3446.90	45.50
50-54	2740.00	36.17
55-59	240.65	3.18
60-64	1.15	0.02
65-69	0.00	0.00
70-74	0.00	0.00
75-79	0.00	0.00
80-84	0.00	0.00
85-89	0.00	0.00
90-94	0.00	0.00
95-99	0.00	0.00

100-104	0.00	0.00
105-109	0.00	0.00
110-114	0.00	0.00
115-119	0.00	0.00
120-124	0.00	0.00
125-129	0.00	0.00
>130	0.00	0.00

Lifetime: Time vs Engine Speed And Engine Load Factor - 3516B Port (TTJ00555)

rpm	<600	600-799	800-999	1000-1199	1200-1399	1400-1599	1600-1799	>1800	Total
%									
<50	65.30	5429.45	1358.45	335.25	234.35	9.90	2.75	1.05	7436.50
50-60	0.00	0.25	0.35	1.80	62.75	18.85	4.15	0.05	88.20
60-70	0.00	0.15	0.20	0.25	2.50	7.20	5.80	0.00	16.10
70-80	0.00	0.05	0.15	0.05	0.30	0.25	1.65	0.75	3.20
80-90	0.00	0.00	0.05	0.00	0.05	0.00	1.80	4.85	6.75
>90	0.00	0.00	0.00	0.00	0.05	0.05	4.75	0.60	5.45
Total	65.30	5429.90	1359.20	337.35	300.00	36.25	20.90	7.30	7556.20

Monitoring System - 3516B Port (TTJ00555)

Description	State	Trip Point	Delay Time
<u>Altitude (atmospheric pressure)</u>			
Moderate Severity (2)	On	93.0 kPa	0 sec
<u>Engine Overspeed</u>			
Least Severe (1)	On	2070 rpm	0 sec
Most Severe (3)	On	2070 rpm	0 sec
<u>High Aftercooler Coolant Temperature</u>			
Least Severe (1)	On	70 Deg C	5 sec
Moderate Severity (2)	On	90 Deg C	5 sec
Most Severe (3)	On	102 Deg C	5 sec
<u>High Air Filter Restriction Pressure</u>			
Least Severe (1)	On	7.0 kPa	5 sec
Moderate Severity (2)	On	7.0 kPa	5 sec
<u>High Boost Pressure</u>			
Least Severe (1)	Off	300.0 kPa	0 sec
Moderate Severity (2)	Off	300.0 kPa	0 sec
<u>High Crankcase Pressure</u>			
Least Severe (1)	On	2.0 kPa	3 sec
Moderate Severity (2)	On	3.0 kPa	10 sec
Most Severe (3)	On	3.5 kPa	3 sec
<u>High Engine Coolant Temperature</u>			
Least Severe (1)	On	102 Deg C	5 sec
Moderate Severity (2)	On	105 Deg C	30 sec

Most Severe (3)	On	107 Deg C	5 sec
High Engine Inlet Air Temperature			
Least Severe (1)	On	132.0 Deg C	5 sec
Moderate Severity (2)	Off	137.0 Deg C	5 sec
Most Severe (3)	Off	137.0 Deg C	5 sec
High Engine Oil Filter Restriction Pressure			
Least Severe (1)	On	138 kPa	5 sec
High Engine Oil Temperature			
Least Severe (1)	On	90 Deg C	5 sec
Moderate Severity (2)	Off	110 Deg C	30 sec
Most Severe (3)	Off	110 Deg C	5 sec
High Exhaust Temperature			
Least Severe (1)	On	756 Deg C	5 sec
Moderate Severity (2)	On	756 Deg C	5 sec
High Fuel Filter Restriction Pressure			
Least Severe (1)	On	105 kPa	5 sec
High Transmission Oil Temperature			
Least Severe (1)	Off	99 Deg C	5 sec
Low Aftercooler Coolant Level			
Least Severe (1)	Off	None	30 sec
Low Coolant Level			
Least Severe (1)	Off	None	30 sec
Low Engine Coolant Temperature			
Least Severe (1)	Off	63 Deg C	60 sec
Low Engine Oil Level			
Least Severe (1)	Off	None	30 sec
Low Engine Oil Pressure			
Least Severe (1)	On	None	4 sec
Most Severe (3)	On	None	9 sec
Low System Voltage			
Least Severe (1)	On	20 Volts	10 sec
Low Transmission Oil Pressure			
Least Severe (1)	Off	1500 kPa	5 sec

Injector Codes Calibration - 3516B Port (TTJ00555)

Injector	Code
Injector 1	1432
Injector 2	3552
Injector 3	5420
Injector 4	1684

Injector 5	1240
Injector 6	5517
Injector 7	6502
Injector 8	6300
Injector 9	4305
Injector 10	9805
Injector 11	1135
Injector 12	5407
Injector 13	5617
Injector 14	2129
Injector 15	5509
Injector 16	4962

Bijlage 11. Data Definition Document

Data Definition Document

Graduation project of 04-02-2019 to 31-05-2019

Smit Lamnalco, Rotterdam

Mick Remmerswaal 13134140

Data type	Definition
Date	The date the document was generated with Caterpillar Electronic Technician.
Configuration	Contains the complete engine configuration, e.g. Engine Serial Number, Equipment ID, Engine Location, Engine Pre-Lube Duration etc.
Logged Diagnostic Codes	The logged codes during the making of the report, shows how many occurred and when the first and last one occurred. Diagnostic codes display Electrical problems.
Logged Event Codes	The logged codes during the making of the report, shows how many occurred and when the first and last one occurred. Event codes display Mechanical problems.
Active Diagnostic Codes	The active codes during the making of the report. Diagnostic codes display Electrical problems
Active Event Codes	The active codes during the making of the report. Event codes display Mechanical problems.
Current Totals	The total currently taken from the engine, e.g. the current total running hours & the current total liters of fuel burned
Time vs ***	This contains the histogram data of the engine when comparing, e.g. Engine Speed over Time or Load Factor over Time or Exhaust Temperature over Time
Monitoring System	Contains data which shows when alerts are fired, e.g. a level (1) High Engine Coolant Temperature alert is fired, when the temperature rises above the 102 Deg C for 5 seconds, a level (2) High Engine Coolant Temperature alert is fired, when temperature rises above the 105 Deg C for 30 seconds, etc.
Injector Codes	Contains the code ID's of the injectors of the engine.

Bijlage 12. Ontworpen XML schema's

3516B

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" <xs:element
   name="ActiveDiagnostic">
3.   <xs:complexType>
4.     <xs:sequence>
5.       <xs:element ref="Code" />
6.       <xs:element ref="Description" />
7.     </xs:sequence>
8.     <xs:attribute ref="Id" use="required" />
9.   </xs:complexType>
10. </xs:element>
11. <xs:element name="ActiveEvent">
12.   <xs:complexType>
13.     <xs:sequence>
14.       <xs:element ref="Code" />
15.       <xs:element ref="Description" />
16.     </xs:sequence>
17.     <xs:attribute ref="Id" use="required" />
18.   </xs:complexType>
19. </xs:element>
20. <xs:element
   name="AftercoolerCoolantLevelSensorInstallationStatus"
   type="xs:string" />
21. <xs:element name="AirIntakeShutoffDetectionInstallationStatus"
   type="xs:string" />
22. <xs:element name="AverageFuelRate" type="xs:string" />
23. <xs:element name="AverageRPM" type="xs:string" />
24. <xs:element name="Code" type="xs:string" />
25. <xs:element name="ColdCylinderCutout" type="xs:string" />
26. <xs:element name="Configuration">
27.   <xs:complexType>
28.     <xs:sequence>
29.       <xs:element ref="EquipmentID" />
30.       <xs:element ref="EngineSerialNumber" />
31.       <xs:element ref="ECMSerialNumber" />
32.       <xs:element ref="PersonalityModulePartNumber" />
33.       <xs:element ref="PersonalityModuleReleaseDate" />
34.       <xs:element ref="RatingNumber" />
35.       <xs:element ref="RatedPower" />
36.       <xs:element ref="RatedPeakTorque" />
37.       <xs:element ref="RatedEngineSpeed" />
38.       <xs:element ref="EngineRotation" />
39.       <xs:element ref="EngineLocation" />
40.       <xs:element ref="FuelRatioControlOffset" />
41.       <xs:element ref="RatedFuelPosition" />
42.       <xs:element ref="LowIdleSpeed" />
43.       <xs:element ref="HighIdleSpeed" />
44.       <xs:element ref="FuelCorrectionFactor" />
45.       <xs:element ref="EngineCoolingSystemConfig" />
46.       <xs:element ref="ColdCylinderCutout" />
47.       <xs:element ref="CooldownSpeed" />
48.       <xs:element ref="CooldownDuration" />
49.       <xs:element ref="EnginePreLubeDuration" />
50.       <xs:element ref="CrankDuration" />
51.       <xs:element ref="MaximumNumberOfCrankCycles" />
52.       <xs:element ref="CrankTerminateRPM" />
53.       <xs:element ref="SecondaryThrottleEnableStatus" />
```

```

54.         <xs:element ref="NumberOfSynchronizedEnginesConfiguration"
/>
55.         <xs:element ref="TorqueLimitEnableStatus" />
56.         <xs:element ref="MaximumEngineTorqueLimit" />
57.         <xs:element ref="MinimumEngineTorqueLimit" />
58.         <xs:element ref="CoolantWarningEngineLoadEnableThreshold"
/>
59.         <xs:element
ref="ProgrammableDroopFeatureInstallationStatus" />
60.         <xs:element ref="EngineSpeedDroop" />
61.         <xs:element ref="TransmissionOilTemperatureConfiguration"
/>
62.         <xs:element
ref="TransmissionOilPressureSensorConfiguration" />
63.         <xs:element ref="FuelLevelSensorConfiguration" />
64.         <xs:element ref="CoolantLevelSensorConfiguration" />
65.         <xs:element
ref="AirIntakeShutoffDetectionInstallationStatus" />
66.         <xs:element
ref="AftercoolerCoolantLevelSensorInstallationStatus" />
67.         <xs:element ref="InletAirTemperatureSensor" />
68.         <xs:element
ref="EngineOilTemperatureSensorInstallationStatus" />
69.         <xs:element ref="EngineOilLevelSensorInstallationStatus"
/>
70.         <xs:element ref="GovernorGainFactor1" />
71.         <xs:element ref="GovernorMinimumStabilityFactor1" />
72.         <xs:element ref="GovernorMaximumStabilityFactor1" />
73.         <xs:element ref="EngineOilRenewalSystemInstallationStatus"
/>
74.         <xs:element ref="EngineOilRenewalSystemEnableStatus" />
<xs:element ref="OilRenewalRate" />
75.         <xs:element ref="FLS" />
76.         <xs:element ref="FTS" />
77.         <xs:element ref="TotalTattletale" />
78.         <xs:element ref="ProgEngLoadFac" />
79.     </xs:sequence>
80. </xs:complexType>
81. </xs:element>
82. <xs:element name="CoolantLevelSensorConfiguration"
type="xs:string" />
83. <xs:element name="CoolantWarningEngineLoadEnableThreshold"
type="xs:string" />
84. <xs:element name="CooldownDuration" type="xs:string" />
85. <xs:element name="CooldownSpeed" type="xs:string" />
86. <xs:element name="CrankDuration" type="xs:string" />
87. <xs:element name="CrankTerminateRPM" type="xs:string" />
88. <xs:element name="CurrentTotals">
89.     <xs:complexType>
90.         <xs:sequence>
91.             <xs:element ref="TotalTime" />
92.             <xs:element ref="TotalFuel" />
93.             <xs:element ref="TotalIdleTime" />
94.             <xs:element ref="TotalIdleFuel" />
95.             <xs:element ref="TotalMaxFuel" />
96.             <xs:element ref="LifetimeTotalEngineRevolutions" />
97.             <xs:element ref="PercentageIdleTime" />
98.             <xs:element ref="TotalOperatingHours" />
99.             <xs:element ref="AverageRPM" />
100.            <xs:element ref="AverageFuelRate" />
101.            <xs:element ref="EngineStarts" />

```

```

102.         <xs:element ref="OverallLoadFactor" />
103.         <xs:element
104.             ref="DefinitePeriodAveragePercentEngineLoadFactor" />
105.         <xs:element ref="TotalEngineRatedSpeedOperatingHours" />
106.         <xs:element ref="PercentageRatedEngineSpeedTime" />
107.     </xs:sequence>
108. </xs:complexType>
109. </xs:element>
110. <xs:element name="DateTime" type="xs:dateTime" />
111. <xs:element name="DefinitePeriodAveragePercentEngineLoadFactor"
112.     type="xs:string" />
113. <xs:element name="Delay" type="xs:string" />
114. <xs:element name="Description" type="xs:string" default="_" />
115. <xs:element name="ECMSerialNumber" type="xs:string" />
116. <xs:element name="EffectsID" type="xs:short" />
117. <xs:element name="EngineCoolingSystemConfig" type="xs:string" />
118. <xs:element name="EngineData">
119.     <xs:complexType>
120.         <xs:sequence>
121.             <xs:element ref="Hours" />
122.         </xs:sequence>
123.         <xs:attribute ref="Description" use="required" />
124.         <xs:attribute ref="BinRange" use="required" />
125.         <xs:attribute ref="Id" use="required" />
126.     </xs:complexType>
127. </xs:element>
128. <xs:element name="EngineDataHeader">
129.     <xs:complexType>
130.         <xs:sequence>
131.             <xs:element ref="EngineData" maxOccurs="unbounded" />
132.         </xs:sequence>
133.     </xs:complexType>
134. </xs:element>
135. <xs:element name="EngineLocation" type="xs:string" />
136. <xs:element name="EngineModel" type="xs:string" />
137. <xs:element name="EngineOilLevelSensorInstallationStatus"
138.     type="xs:string" />
139. <xs:element name="EngineOilRenewalSystemEnableStatus"
140.     type="xs:string" />
141. <xs:element name="EngineOilRenewalSystemInstallationStatus"
142.     type="xs:string" />
143. <xs:element name="EngineOilTemperatureSensorInstallationStatus"
144.     type="xs:string" />
145. <xs:element name="EnginePos" type="xs:string" />
146. <xs:element name="EnginePreLubeDuration" type="xs:string" />
147. <xs:element name="EngineRotation" type="xs:string" />
148. <xs:element name="EngineSerialNumber" type="xs:string" />
149. <xs:element name="EngineSpeedDroop" type="xs:string" />
150. <xs:element name="EngineStarts" type="xs:short" />
151. <xs:element name="EquipmentID" type="xs:string" />
152. <xs:element name="FLS" type="xs:byte" />
153. <xs:element name="FTS" type="xs:byte" />
154. <xs:element name="First" type="xs:int" />
155. <xs:element name="FuelCorrectionFactor" type="xs:string" />
156. <xs:element name="FuelLevelSensorConfiguration" type="xs:string" />
157. </xs:sequence>
158. </xs:complexType>
159. </xs:element>
160. <xs:element name="FuelRatioControlOffset" type="xs:byte" />
161. <xs:element name="GUID" type="xs:string" />
162. <xs:element name="GovernorGainFactor1" type="xs:short" />
163. <xs:element name="GovernorMaximumStabilityFactor1"
164.     type="xs:short" />

```

```

155.     <xs:element name="GovernorMinimumStabilityFactor1"
        type="xs:byte" />
156.     <xs:element name="HighIdleSpeed" type="xs:string" />
157.     <xs:element name="Hours" type="xs:decimal" />
158.     <xs:element name="Injector1" type="xs:short" />
159.     <xs:element name="Injector10" type="xs:short" />
160.     <xs:element name="Injector11" type="xs:short" />
161.     <xs:element name="Injector12" type="xs:short" />
162.     <xs:element name="Injector13" type="xs:short" />
163.     <xs:element name="Injector14" type="xs:short" />
164.     <xs:element name="Injector15" type="xs:short" />
165.     <xs:element name="Injector16" type="xs:short" />
166.     <xs:element name="Injector2" type="xs:short" />
167.     <xs:element name="Injector3" type="xs:short" />
168.     <xs:element name="Injector4" type="xs:short" />
169.     <xs:element name="Injector5" type="xs:short" />
170.     <xs:element name="Injector6" type="xs:short" />
171.     <xs:element name="Injector7" type="xs:short" />
172.     <xs:element name="Injector8" type="xs:short" />
173.     <xs:element name="Injector9" type="xs:short" />
174.     <xs:element name="InjectorCodes">
175.         <xs:complexType>
176.             <xs:sequence>
177.                 <xs:element ref="Injector1" />
178.                 <xs:element ref="Injector2" />
179.                 <xs:element ref="Injector3" />
180.                 <xs:element ref="Injector4" />
181.                 <xs:element ref="Injector5" />
182.                 <xs:element ref="Injector6" />
183.                 <xs:element ref="Injector7" />
184.                 <xs:element ref="Injector8" />
185.                 <xs:element ref="Injector9" />
186.                 <xs:element ref="Injector10" />
187.                 <xs:element ref="Injector11" />
188.                 <xs:element ref="Injector12" />
189.                 <xs:element ref="Injector13" />
190.                 <xs:element ref="Injector14" />
191.                 <xs:element ref="Injector15" />
192.                 <xs:element ref="Injector16" />
193.             </xs:sequence>
194.         </xs:complexType>
195.     </xs:element>
196.     <xs:element name="InletAirTemperatureSensor" type="xs:string" />
197.     <xs:element name="Last" type="xs:int" default="0" />
198.     <xs:element name="LifetimeTotalEngineRevolutions"
        type="xs:string" />
199.     <xs:element name="LogHeader">
200.         <xs:complexType>
201.             <xs:sequence>
202.                 <xs:element ref="LoggedDiagnostic" minOccurs="0"
                    maxOccurs="unbounded" />
203.                 <xs:element ref="LoggedEvent" minOccurs="0"
                    maxOccurs="unbounded" />
204.                 <xs:element ref="ActiveDiagnostic" minOccurs="0"
                    maxOccurs="unbounded" />
205.                 <xs:element ref="ActiveEvent" minOccurs="0"
                    maxOccurs="unbounded" />
206.             </xs:sequence>
207.         </xs:complexType>
208.     </xs:element>
209.     <xs:element name="LoggedDiagnostic">

```

```

210.     <xs:complexType>
211.         <xs:sequence>
212.             <xs:element ref="Code" />
213.             <xs:element ref="Description" />
214.             <xs:element ref="Occurence" />
215.             <xs:element ref="First" />
216.             <xs:element ref="Last" />
217.         </xs:sequence>
218.         <xs:attribute ref="Id" use="required" />
219.     </xs:complexType>
220. </xs:element>
221. <xs:element name="LoggedEvent">
222.     <xs:complexType>
223.         <xs:sequence>
224.             <xs:element ref="Code" />
225.             <xs:element ref="Description" />
226.             <xs:element ref="Occurence" />
227.             <xs:element ref="First" />
228.             <xs:element ref="Last" />
229.         </xs:sequence>
230.         <xs:attribute ref="Id" use="required" />
231.     </xs:complexType>
232. </xs:element>
233. <xs:element name="LowIdleSpeed" type="xs:string" />
234. <xs:element name="MaximumEngineTorqueLimit" type="xs:string" />
235. <xs:element name="MaximumNumberOfCrankCycles" type="xs:byte" />
236. <xs:element name="MinimumEngineTorqueLimit" type="xs:string" />
237. <xs:element name="Monitoring">
238.     <xs:complexType>
239.         <xs:sequence>
240.             <xs:element ref="State" />
241.             <xs:element ref="Threshold" />
242.             <xs:element ref="Delay" />
243.         </xs:sequence>
244.         <xs:attribute ref="Description" use="required" />
245.         <xs:attribute ref="Severity" use="required" />
246.         <xs:attribute ref="Id" use="required" />
247.     </xs:complexType>
248. </xs:element>
249. <xs:element name="MonitoringHeader">
250.     <xs:complexType>
251.         <xs:sequence>
252.             <xs:element ref="Monitoring" maxOccurs="unbounded" />
253.         </xs:sequence>
254.     </xs:complexType>
255. </xs:element>
256. <xs:element name="NumberOfSynchronizedEnginesConfiguration"
    type="xs:byte" />
257. <xs:element name="Occurence" type="xs:int" default="0" />
258. <xs:element name="OilRenewalRate" type="xs:byte" />
259. <xs:element name="OverallLoadFactor" type="xs:string" />
260. <xs:element name="PercentageIdleTime" type="xs:string" />
261. <xs:element name="PercentageRatedEngineSpeedTime"
    type="xs:string" />
262. <xs:element name="PersonalityModulePartNumber" type="xs:string"
    />
263. <xs:element name="PersonalityModuleReleaseDate" type="xs:string"
    />
264. <xs:element name="ProgEngLoadFac" type="xs:string" />
265. <xs:element name="ProgrammableDroopFeatureInstallationStatus"
    type="xs:string" />

```

```

266.     <xs:element name="RatedEngineSpeed" type="xs:string" />
267.     <xs:element name="RatedFuelPosition" type="xs:string" />
268.     <xs:element name="RatedPeakTorque" type="xs:string" />
269.     <xs:element name="RatedPower" type="xs:string" />
270.     <xs:element name="RatingNumber" type="xs:byte" />
271.     <xs:element name="SLENG3516B">
272.         <xs:complexType>
273.             <xs:sequence>
274.                 <xs:element ref="DateTime" />
275.                 <xs:element ref="SerialNumber" />
276.                 <xs:element ref="EquipmentID" />
277.                 <xs:element ref="VesselName" />
278.                 <xs:element ref="EnginePos" />
279.                 <xs:element ref="EngineModel" />
280.                 <xs:element ref="EffectsID" />
281.                 <xs:element ref="GUID" />
282.                 <xs:element ref="Configuration" />
283.                 <xs:element ref="LogHeader" />
284.                 <xs:element ref="CurrentTotals" />
285.                 <xs:element ref="EngineDataHeader" />
286.                 <xs:element ref="MonitoringHeader" />
287.                 <xs:element ref="InjectorCodes" />
288.             </xs:sequence>
289.         </xs:complexType>
290.     </xs:element>
291.     <xs:element name="SecondaryThrottleEnableStatus"
292.         type="xs:string" />
292.     <xs:element name="SerialNumber" type="xs:string" />
293.     <xs:element name="State" type="xs:string" />
294.     <xs:element name="Threshold" type="xs:string" />
295.     <xs:element name="TorqueLimitEnableStatus" type="xs:string" />
296.     <xs:element name="TotalEngineRatedSpeedOperatingHours"
297.         type="xs:string" />
297.     <xs:element name="TotalFuel" type="xs:string" />
298.     <xs:element name="TotalIdleFuel" type="xs:string" />
299.     <xs:element name="TotalIdleTime" type="xs:string" />
300.     <xs:element name="TotalMaxFuel" type="xs:string" />
301.     <xs:element name="TotalOperatingHours" type="xs:string" />
302.     <xs:element name="TotalTattletale" type="xs:byte" />
303.     <xs:element name="TotalTime" type="xs:string" />
304.     <xs:element name="TransmissionOilPressureSensorConfiguration"
305.         type="xs:string" />
305.     <xs:element name="TransmissionOilTemperatureConfiguration"
306.         type="xs:string" />
306.     <xs:element name="VesselName" type="xs:string" />
307.     <xs:attribute name="BinRange" type="xs:string" />
308.     <xs:attribute name="Description" type="xs:string" />
309.     <xs:attribute name="Severity" type="xs:byte" />
310.     <xs:attribute name="Id" type="xs:int" />
311. </xs:schema>

```

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3.   <xs:element name="ActiveDiagnostic">
4.     <xs:complexType>
5.       <xs:sequence>
6.         <xs:element ref="Code" />
7.         <xs:element ref="Description" />
8.       </xs:sequence>
9.       <xs:attribute ref="Id" use="required" />
10.     </xs:complexType>
11.   </xs:element>
12.   <xs:element name="ActiveEvent">
13.     <xs:complexType>
14.       <xs:sequence>
15.         <xs:element ref="Code" />
16.         <xs:element ref="Description" />
17.       </xs:sequence>
18.       <xs:attribute ref="Id" use="required" />
19.     </xs:complexType>
20.   </xs:element>
21.   <xs:element
    name="AftercoolerCoolantLevelSensorInstallationStatus"
    type="xs:string" />
22.   <xs:element name="AverageFuelRate" type="xs:string" />
23.   <xs:element name="AverageLoadFactor" type="xs:string" />
24.   <xs:element name="AverageRPM" type="xs:string" />
25.   <xs:element name="BollardTestTimeRemaining" type="xs:string" />
26.   <xs:element name="Code" type="xs:string" />
27.   <xs:element name="ColdCylinderCutout" type="xs:string" />
28.   <xs:element name="Configuration">
29.     <xs:complexType>
30.       <xs:sequence>
31.         <xs:element ref="EquipmentID" />
32.         <xs:element ref="EngineSerialNumber" />
33.         <xs:element ref="ECMSerialNumber" />
34.         <xs:element ref="SoftwareGroupPartNumber" />
35.         <xs:element ref="SoftwareGroupReleaseDate" />
36.         <xs:element ref="SoftwareGroupDescription" />
37.         <xs:element ref="RatingNumber" />
38.         <xs:element ref="RatedPower" />
39.         <xs:element ref="RatedPeakTorque" />
40.         <xs:element ref="TopEngineSpeedRange" />
41.         <xs:element ref="TestSpec" />
42.         <xs:element ref="EngineLocation" />
43.         <xs:element ref="NumberofSynchronizedEnginesConfiguration"
    />
44.       <xs:element ref="CrankTerminateRPM" />
45.       <xs:element ref="LowIdleSpeed" />
46.       <xs:element ref="HighIdleSpeed" />
47.       <xs:element ref="MaximumTrollingEngineSpeed" />
48.       <xs:element ref="SecondaryThrottleEnableStatus" />
49.       <xs:element ref="TertiaryDesiredSpeedInputConfiguration" />
50.       <xs:element ref="TachometerFrequencyConfiguration" />
51.       <xs:element ref="ColdCylinderCutout" />
52.       <xs:element ref="BollardTestTimeRemaining" />
53.       <xs:element ref="TransmissionOilTemperatureSensor" />
54.       <xs:element ref="TransmissionOilPressureSensor" />
55.       <xs:element ref="FuelLevelSensorInstallationStatus" />

```

```

56.         <xs:element ref="CoolantLevelSensor" />
57.         <xs:element
    ref="AftercoolerCoolantLevelSensorInstallationStatus" />
58.         <xs:element ref="EngineOilLevelSensorInstallationStatus" />
59.         <xs:element
    ref="EngineCoolantPressureSensorInstallationStatus" />
60.         <xs:element
    ref="UnfilteredEngineOilPressureSensorInstallationStatus" />
61.         <xs:element
    ref="UnfilteredFuelPressureSensorInstallationStatus" />
62.         <xs:element
    ref="ExhaustTemperatureSensorInstallationStatus" />
63.         <xs:element
    ref="EngineOilTemperatureSensorInstallationStatus" />
64.         <xs:element
    ref="HighExhaustTemperatureWarningSystemEnableStatus" />
65.         <xs:element ref="EngineOilCapacity" />
66.         <xs:element ref="FuelCorrectionFactor" />
67.         <xs:element ref="MaintenanceIndicatorMode" />
68.         <xs:element ref="PM1Interval" />
69.         <xs:element ref="FLS" />
70.         <xs:element ref="FTS" />
71.         <xs:element ref="CustomerPassword1" />
72.         <xs:element ref="CustomerPassword2" />
73.         <xs:element ref="TotalTattletale" />
74.         <xs:element ref="MonSysOvrInstal" minOccurs="0" />
75.     </xs:sequence>
76. </xs:complexType>
77. </xs:element>
78. <xs:element name="CoolantLevelSensor" type="xs:string" />
79. <xs:element name="CrankTerminateRPM" type="xs:string" />
80. <xs:element name="CurrentTotals">
81.     <xs:complexType>
82.         <xs:sequence>
83.             <xs:element ref="TotalIdleTime" />
84.             <xs:element ref="TotalFuel" />
85.             <xs:element ref="TotalIdleFuel" />
86.             <xs:element ref="TotalMaxFuel" />
87.             <xs:element ref="AverageLoadFactor" />
88.             <xs:element ref="TotalOperatingHours" />
89.             <xs:element ref="EngineStarts" />
90.             <xs:element ref="LifetimeTotalEngineRevolutions" />
91.             <xs:element ref="StartsHour" />
92.             <xs:element ref="PercentageIdleTime" />
93.             <xs:element ref="AverageRPM" />
94.             <xs:element ref="AverageFuelRate" />
95.             <xs:element ref="OverallLoadFactor" />
96.         </xs:sequence>
97.     </xs:complexType>
98. </xs:element>
99. <xs:element name="CustomerPassword1" type="xs:string" />
100. <xs:element name="CustomerPassword2" type="xs:string" />
101. <xs:element name="DateTime" type="xs:dateTime" />
102. <xs:element name="Delay" type="xs:string" />
103. <xs:element name="Description" type="xs:string" />
104. <xs:element name="ECMSerialNumber" type="xs:string" />
105. <xs:element name="EffectsID" type="xs:short" />
106. <xs:element name="EngineCoolantPressureSensorInstallationStatus"
    type="xs:string" />
107. <xs:element name="EngineData">
108.     <xs:complexType>

```

```

109.         <xs:sequence>
110.             <xs:element ref="Hours" />
111.         </xs:sequence>
112.         <xs:attribute ref="Description" use="required" />
113.         <xs:attribute ref="BinRange" use="required" />
114.         <xs:attribute ref="Id" use="required" />
115.     </xs:complexType>
116. </xs:element>
117. <xs:element name="EngineDataHeader">
118.     <xs:complexType>
119.         <xs:sequence>
120.             <xs:element ref="EngineData" maxOccurs="unbounded" />
121.         </xs:sequence>
122.     </xs:complexType>
123. </xs:element>
124. <xs:element name="EngineLocation" type="xs:string" />
125. <xs:element name="EngineOilCapacity" type="xs:string" />
126. <xs:element name="EngineOilLevelSensorInstallationStatus"
127.     type="xs:string" />
128. <xs:element name="EngineOilTemperatureSensorInstallationStatus"
129.     type="xs:string" />
130. <xs:element name="EnginePos" type="xs:string" />
131. <xs:element name="EngineSerialNumber" type="xs:string" />
132. <xs:element name="EngineStarts" type="xs:short" />
133. <xs:element name="EquipmentID" type="xs:string" />
134. <xs:element name="ExhaustTemperatureSensorInstallationStatus"
135.     type="xs:string" />
136. <xs:element name="FLS" type="xs:byte" />
137. <xs:element name="FTS" type="xs:byte" />
138. <xs:element name="First" type="xs:string" />
139. <xs:element name="FuelCorrectionFactor" type="xs:string" />
140. <xs:element name="FuelLevelSensorInstallationStatus"
141.     type="xs:string" />
142. <xs:element
143.     name="HighExhaustTemperatureWarningSystemEnableStatus"
144.     type="xs:string" />
145. <xs:element name="GUID" type="xs:string" />
146. <xs:element name="HighIdleSpeed" type="xs:string" />
147. <xs:element name="Hours" type="xs:decimal" />
148. <xs:element name="Last" type="xs:string" />
149. <xs:element name="LifetimeTotalEngineRevolutions"
150.     type="xs:string" />
151. <xs:element name="LogHeader">
152.     <xs:complexType>
153.         <xs:sequence>
154.             <xs:element ref="LoggedDiagnostic" minOccurs="0"
155.                 maxOccurs="unbounded" />
156.             <xs:element ref="LoggedEvent" minOccurs="0"
157.                 maxOccurs="unbounded" />
158.             <xs:element ref="ActiveDiagnostic" minOccurs="0"
159.                 maxOccurs="unbounded" />
160.             <xs:element ref="ActiveEvent" minOccurs="0"
161.                 maxOccurs="unbounded" />
162.         </xs:sequence>
163.     </xs:complexType>
164. </xs:element>
165. <xs:element name="LoggedDiagnostic">
166.     <xs:complexType>
167.         <xs:sequence>
168.             <xs:element ref="Code" />
169.             <xs:element ref="Description" />

```

```

159.         <xs:element ref="Occurence" />
160.         <xs:element ref="First" />
161.         <xs:element ref="Last" />
162.     </xs:sequence>
163.     <xs:attribute ref="Id" use="required" />
164. </xs:complexType>
165. </xs:element>
166. <xs:element name="LoggedEvent">
167.     <xs:complexType>
168.         <xs:sequence>
169.             <xs:element ref="Code" />
170.             <xs:element ref="Description" />
171.             <xs:element ref="Occurence" />
172.             <xs:element ref="First" />
173.             <xs:element ref="Last" />
174.         </xs:sequence>
175.         <xs:attribute ref="Id" use="required" />
176.     </xs:complexType>
177. </xs:element>
178. <xs:element name="LowIdleSpeed" type="xs:string" />
179. <xs:element name="MaintenanceIndicatorMode" type="xs:string" />
180. <xs:element name="MaximumTrollingEngineSpeed" type="xs:string" />
181. <xs:element name="EngineModel" type="xs:string" />
182. <xs:element name="MonSysOvrInstal" type="xs:byte" />
183. <xs:element name="Monitoring">
184.     <xs:complexType>
185.         <xs:sequence>
186.             <xs:element ref="State" />
187.             <xs:element ref="Threshold" />
188.             <xs:element ref="Delay" />
189.         </xs:sequence>
190.         <xs:attribute ref="Description" use="required" />
191.         <xs:attribute ref="Severity" use="required" />
192.         <xs:attribute ref="Id" use="required" />
193.     </xs:complexType>
194. </xs:element>
195. <xs:element name="MonitoringHeader">
196.     <xs:complexType>
197.         <xs:sequence>
198.             <xs:element ref="Monitoring" maxOccurs="unbounded" />
199.         </xs:sequence>
200.     </xs:complexType>
201. </xs:element>
202. <xs:element name="NumberofSynchronizedEnginesConfiguration"
    type="xs:byte" />
203. <xs:element name="Occurence" type="xs:string" />
204. <xs:element name="OverallLoadFactor" type="xs:string" />
205. <xs:element name="PM1Interval" type="xs:string" />
206. <xs:element name="PercentageIdleTime" type="xs:string" />
207. <xs:element name="RatedPeakTorque" type="xs:string" />
208. <xs:element name="RatedPower" type="xs:string" />
209. <xs:element name="RatingNumber" type="xs:byte" />
210. <xs:element name="SLENGC18ACERT">
211.     <xs:complexType>
212.         <xs:sequence>
213.             <xs:element ref="DateTime" />
214.             <xs:element ref="SerialNumber" />
215.             <xs:element ref="EquipmentID" />
216.             <xs:element ref="VesselName" />
217.             <xs:element ref="EnginePos" />
218.             <xs:element ref="EngineModel" />

```

```

219.         <xs:element ref="EffectsID" />
220.         <xs:element ref="GUID" />
221.         <xs:element ref="Configuration" />
222.         <xs:element ref="LogHeader" />
223.         <xs:element ref="CurrentTotals" />
224.         <xs:element ref="EngineDataHeader" />
225.         <xs:element ref="MonitoringHeader" />
226.     </xs:sequence>
227. </xs:complexType>
228. </xs:element>
229. <xs:element name="SecondaryThrottleEnableStatus" type="xs:string"
/>
230. <xs:element name="SerialNumber" type="xs:string" />
231. <xs:element name="SoftwareGroupDescription" type="xs:string" />
232. <xs:element name="SoftwareGroupPartNumber" type="xs:string" />
233. <xs:element name="SoftwareGroupReleaseDate" type="xs:string" />
234. <xs:element name="StartsHour" type="xs:decimal" />
235. <xs:element name="State" type="xs:string" />
236. <xs:element name="TachometerFrequencyConfiguration"
type="xs:decimal" />
237. <xs:element name="TertiaryDesiredSpeedInputConfiguration"
type="xs:string" />
238. <xs:element name="TestSpec" type="xs:string" />
239. <xs:element name="Threshold" type="xs:string" />
240. <xs:element name="TopEngineSpeedRange" type="xs:string" />
241. <xs:element name="TotalFuel" type="xs:string" />
242. <xs:element name="TotalIdleFuel" type="xs:string" />
243. <xs:element name="TotalIdleTime" type="xs:string" />
244. <xs:element name="TotalMaxFuel" type="xs:string" />
245. <xs:element name="TotalOperatingHours" type="xs:string" />
246. <xs:element name="TotalTattletale" type="xs:byte" />
247. <xs:element name="TransmissionOilPressureSensor" type="xs:string"
/>
248. <xs:element name="TransmissionOilTemperatureSensor"
type="xs:string" />
249. <xs:element
name="UnfilteredEngineOilPressureSensorInstallationStatus"
type="xs:string" />
250. <xs:element name="UnfilteredFuelPressureSensorInstallationStatus"
type="xs:string" />
251. <xs:element name="VesselName" type="xs:string" />
252. <xs:attribute name="BinRange" type="xs:string" />
253. <xs:attribute name="Description" type="xs:string" />
254. <xs:attribute name="Severity" type="xs:byte" />
255. <xs:attribute name="Id" type="xs:int" />
256. </xs:schema>

```

Bijlage 13. SQL Queries

Gemaakte tabellen

```

1. USE [SLDASH_A]
2. GO
3. /***** Object: Table [dbo].[3516 ActiveDiagnostics]      Script
   Date: 4/16/2019 4:08:40 PM *****/
4. SET ANSI_NULLS ON
5. GO

```

```

6. SET QUOTED_IDENTIFIER ON
7. GO
8. CREATE TABLE [dbo].[3516_ActiveDiagnostics] (
9.     [Code] [nvarchar](255) NULL,
10.    [Description] [nvarchar](255) NULL,
11.    [GUID] [uniqueidentifier] NULL,
12.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_3516_ActiveDiagnostics_modifiedOn] DEFAULT (getdate()),
13.    [id] [int] NULL
14. ) ON [PRIMARY]
15.
16. GO
17. /***** Object: Table [dbo].[3516_ActiveEvent]      Script
    Date: 4/16/2019 4:08:40 PM *****/
18. SET ANSI_NULLS ON
19. GO
20. SET QUOTED_IDENTIFIER ON
21. GO
22. CREATE TABLE [dbo].[3516_ActiveEvent] (
23.     [Code] [nvarchar](255) NULL,
24.     [Description] [nvarchar](255) NULL,
25.     [GUID] [uniqueidentifier] NULL,
26.     [modifiedOn] [datetime] NULL,
27.     [id] [int] NULL
28. ) ON [PRIMARY]
29.
30. GO
31. /***** Object: Table [dbo].[3516_Configuration]      Script
    Date: 4/16/2019 4:08:40 PM *****/
32. SET ANSI_NULLS ON
33. GO
34. SET QUOTED_IDENTIFIER ON
35. GO
36. CREATE TABLE [dbo].[3516_Configuration] (
37.     [EquipmentID] [nvarchar](255) NULL,
38.     [EngineSerialNumber] [nvarchar](255) NULL,
39.     [ECMSerialNumber] [nvarchar](255) NULL,
40.     [PersonalityModulePartNumber] [nvarchar](255) NULL,
41.     [PersonalityModuleReleaseDate] [nvarchar](255) NULL,
42.     [RatingNumber] [nvarchar](255) NULL,
43.     [RatedPower] [nvarchar](255) NULL,
44.     [RatedPeakTorque] [nvarchar](255) NULL,
45.     [RatedEngineSpeed] [nvarchar](255) NULL,
46.     [EngineRotation] [nvarchar](255) NULL,
47.     [FuelRatioControlOffset] [nvarchar](255) NULL,
48.     [RatedFuelPosition] [nvarchar](255) NULL,
49.     [LowIdleSpeed] [nvarchar](255) NULL,
50.     [HighIdleSpeed] [nvarchar](255) NULL,
51.     [FuelCorrectionFactor] [nvarchar](255) NULL,
52.     [EngineCoolingSystemConfig] [nvarchar](255) NULL,
53.     [ColdCylinderCutout] [nvarchar](255) NULL,
54.     [CooldownSpeed] [nvarchar](255) NULL,
55.     [CooldownDuration] [nvarchar](255) NULL,
56.     [EnginePreLubeDuration] [nvarchar](255) NULL,
57.     [CrankDuration] [nvarchar](255) NULL,

```

```

58.         [MaximumNumberOfCrankCycles] [nvarchar](255) NULL,
59.         [CrankTerminateRPM] [nvarchar](255) NULL,
60.         [SecondaryThrottleEnableStatus] [nvarchar](255) NULL,
61.         [NumberOfSynchronizedEnginesConfiguration]
        [nvarchar](255) NULL,
62.         [TorqueLimitEnableStatus] [nvarchar](255) NULL,
63.         [MaximumEngineTorqueLimit] [nvarchar](255) NULL,
64.         [MinimumEngineTorqueLimit] [nvarchar](255) NULL,
65.         [CoolantWarningEngineLoadEnableThreshold]
        [nvarchar](255) NULL,
66.         [ProgrammableDroopFeatureInstallationStatus]
        [nvarchar](255) NULL,
67.         [EngineSpeedDroop] [nvarchar](255) NULL,
68.         [TransmissionOilTemperatureConfiguration]
        [nvarchar](255) NULL,
69.         [TransmissionOilPressureSensorConfiguration]
        [nvarchar](255) NULL,
70.         [FuelLevelSensorConfiguration] [nvarchar](255) NULL,
71.         [CoolantLevelSensorConfiguration] [nvarchar](255)
        NULL,
72.         [AirIntakeShutoffDetectionInstallationStatus]
        [nvarchar](255) NULL,
73.         [AftercoolerCoolantLevelSensorInstallationStatus]
        [nvarchar](255) NULL,
74.         [InletAirTemperatureSensor] [nvarchar](255) NULL,
75.         [EngineOilTemperatureSensorInstallationStatus]
        [nvarchar](255) NULL,
76.         [EngineOilLevelSensorInstallationStatus]
        [nvarchar](255) NULL,
77.         [GovernorGainFactor1] [nvarchar](255) NULL,
78.         [GovernorMinimumStabilityFactor1] [nvarchar](255)
        NULL,
79.         [GovernorMaximumStabilityFactor1] [nvarchar](255)
        NULL,
80.         [EngineOilRenewalSystemInstallationStatus]
        [nvarchar](255) NULL,
81.         [EngineOilRenewalSystemEnableStatus] [nvarchar](255)
        NULL,
82.         [OilRenewalRate] [nvarchar](255) NULL,
83.         [FLS] [nvarchar](255) NULL,
84.         [FTS] [nvarchar](255) NULL,
85.         [TotalTattletale] [nvarchar](255) NULL,
86.         [ProgEngLoadFac] [nvarchar](255) NULL,
87.         [GUID] [uniqueidentifier] NULL,
88.         [modifiedOn] [datetime] NULL CONSTRAINT
        [DF_3516_Configuration_modifiedOn] DEFAULT (getdate())
89. ) ON [PRIMARY]
90.
91. GO
92. /***** Object: Table [dbo].[3516_CurrentTotals] Script
    Date: 4/16/2019 4:08:40 PM *****/
93. SET ANSI_NULLS ON
94. GO
95. SET QUOTED_IDENTIFIER ON
96. GO

```

```

97. CREATE TABLE [dbo].[3516_CurrentTotals] (
98.     [TotalTime] [nvarchar](255) NULL,
99.     [TotalFuel] [nvarchar](255) NULL,
100.    [TotalIdleTime] [nvarchar](255) NULL,
101.    [TotalIdleFuel] [nvarchar](255) NULL,
102.    [TotalMaxFuel] [nvarchar](255) NULL,
103.    [LifetimeTotalEngineRevolutions] [nvarchar](255) NULL,
104.    [PercentageIdleTime] [nvarchar](255) NULL,
105.    [TotalOperatingHours] [nvarchar](255) NULL,
106.    [AverageRPM] [nvarchar](255) NULL,
107.    [AverageFuelRate] [nvarchar](255) NULL,
108.    [EngineStarts] [nvarchar](255) NULL,
109.    [OverallLoadFactor] [nvarchar](255) NULL,
110.    [DefinitePeriodAveragePercentEngineLoadFactor]
    [nvarchar](255) NULL,
111.    [TotalEngineRatedSpeedOperatingHours] [nvarchar](255)
    NULL,
112.    [PercentageRatedEngineSpeedTime] [nvarchar](255) NULL,
113.    [GUID] [uniqueidentifier] NULL,
114.    [modifiedOn] [datetime] NULL
115.) ON [PRIMARY]
116.
117.GO
118./****** Object:  Table [dbo].[3516_EngineData]      Script
    Date: 4/16/2019 4:08:40 PM *****/
119.SET ANSI_NULLS ON
120.GO
121.SET QUOTED_IDENTIFIER ON
122.GO
123.CREATE TABLE [dbo].[3516_EngineData] (
124.    [Description] [nvarchar](255) NULL,
125.    [BinRange] [nvarchar](255) NULL,
126.    [Hours] [float] NULL,
127.    [GUID] [uniqueidentifier] NULL,
128.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_3516_EngineData_modifiedOn] DEFAULT (getdate()),
129.    [id] [int] NULL
130.) ON [PRIMARY]
131.
132.GO
133./****** Object:  Table [dbo].[3516_InjectorCodes]    Script
    Date: 4/16/2019 4:08:40 PM *****/
134.SET ANSI_NULLS ON
135.GO
136.SET QUOTED_IDENTIFIER ON
137.GO
138.CREATE TABLE [dbo].[3516_InjectorCodes] (
139.    [Injector1] [nvarchar](255) NULL,
140.    [Injector2] [nvarchar](255) NULL,
141.    [Injector3] [nvarchar](255) NULL,
142.    [Injector4] [nvarchar](255) NULL,
143.    [Injector5] [nvarchar](255) NULL,
144.    [Injector6] [nvarchar](255) NULL,
145.    [Injector7] [nvarchar](255) NULL,
146.    [Injector8] [nvarchar](255) NULL,

```

```

147.         [Injector9] [nvarchar](255) NULL,
148.         [Injector10] [nvarchar](255) NULL,
149.         [Injector11] [nvarchar](255) NULL,
150.         [Injector12] [nvarchar](255) NULL,
151.         [Injector13] [nvarchar](255) NULL,
152.         [Injector14] [nvarchar](255) NULL,
153.         [Injector15] [nvarchar](255) NULL,
154.         [Injector16] [nvarchar](255) NULL,
155.         [GUID] [uniqueidentifier] NULL,
156.         [modifiedOn] [datetime] NULL
157.) ON [PRIMARY]
158.
159.GO
160./***** Object: Table [dbo].[3516 LoggedDiagnostics]
    Script Date: 4/16/2019 4:08:40 PM *****/
161.SET ANSI_NULLS ON
162.GO
163.SET QUOTED_IDENTIFIER ON
164.GO
165.CREATE TABLE [dbo].[3516_LoggedDiagnostics] (
166.         [Code] [nvarchar](255) NULL,
167.         [Description] [nvarchar](255) NULL,
168.         [Occurence] [nvarchar](255) NULL,
169.         [First] [nvarchar](255) NULL,
170.         [Last] [nvarchar](255) NULL,
171.         [GUID] [uniqueidentifier] NULL,
172.         [modifiedOn] [datetime] NULL,
173.         [id] [int] NULL
174.) ON [PRIMARY]
175.
176.GO
177./***** Object: Table [dbo].[3516_LoggedEvent]      Script
    Date: 4/16/2019 4:08:40 PM *****/
178.SET ANSI_NULLS ON
179.GO
180.SET QUOTED_IDENTIFIER ON
181.GO
182.CREATE TABLE [dbo].[3516_LoggedEvent] (
183.         [Code] [nvarchar](255) NULL,
184.         [Description] [nvarchar](255) NULL,
185.         [Occurence] [nvarchar](255) NULL,
186.         [First] [nvarchar](255) NULL,
187.         [Last] [nvarchar](255) NULL,
188.         [GUID] [uniqueidentifier] NULL,
189.         [modifiedOn] [datetime] NULL,
190.         [id] [int] NULL
191.) ON [PRIMARY]
192.
193.GO
194./***** Object: Table [dbo].[3516_Monitoring]      Script
    Date: 4/16/2019 4:08:40 PM *****/
195.SET ANSI_NULLS ON
196.GO
197.SET QUOTED_IDENTIFIER ON
198.GO

```

```

199.CREATE TABLE [dbo].[3516_Monitoring] (
200.    [Description] [nvarchar](255) NULL,
201.    [Severity] [int] NULL,
202.    [State] [nvarchar](255) NULL,
203.    [Threshold] [nvarchar](255) NULL,
204.    [Delay] [nvarchar](255) NULL,
205.    [GUID] [uniqueidentifier] NULL,
206.    [modifiedOn] [datetime] NULL,
207.    [id] [int] NULL
208.) ON [PRIMARY]
209.
210.GO
211./***** Object: Table [dbo].[3516_SLEngineReport]      Script
    Date: 4/16/2019 4:08:40 PM *****/
212.SET ANSI_NULLS ON
213.GO
214.SET QUOTED_IDENTIFIER ON
215.GO
216.CREATE TABLE [dbo].[3516_SLEngineReport] (
217.    [EffactsID] [nvarchar](255) NULL,
218.    [SerialNumber] [nvarchar](255) NULL,
219.    [EquipmentID] [nvarchar](255) NULL,
220.    [VesselName] [nvarchar](255) NULL,
221.    [EngineType] [nvarchar](255) NULL,
222.    [EngineModel] [nvarchar](255) NULL,
223.    [EnginePos] [nvarchar](255) NULL,
224.    [DateTime] [datetime] NULL,
225.    [GUID] [uniqueidentifier] NULL,
226.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_3516_SLEngineReport_ModifiedDateTime] DEFAULT (getdate())
227.) ON [PRIMARY]
228.
229.GO
230./***** Object: Table [dbo].[C18ACERT ActiveDiagnostics]
    Script Date: 4/16/2019 4:08:40 PM *****/
231.SET ANSI_NULLS ON
232.GO
233.SET QUOTED_IDENTIFIER ON
234.GO
235.CREATE TABLE [dbo].[C18ACERT_ActiveDiagnostics] (
236.    [Code] [nvarchar](255) NULL,
237.    [Description] [nvarchar](255) NULL,
238.    [GUID] [uniqueidentifier] NULL,
239.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_C18ACERT_ActiveDiagnostic_modifiedOn] DEFAULT (getdate()),
240.    [id] [int] NULL
241.) ON [PRIMARY]
242.
243.GO
244./***** Object: Table [dbo].[C18ACERT_ActiveEvent]      Script
    Date: 4/16/2019 4:08:40 PM *****/
245.SET ANSI_NULLS ON
246.GO
247.SET QUOTED_IDENTIFIER ON
248.GO

```

```

249.CREATE TABLE [dbo].[C18ACERT_ActiveEvent] (
250.    [Code] [nvarchar](255) NULL,
251.    [Description] [nvarchar](255) NULL,
252.    [GUID] [uniqueidentifier] NULL,
253.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_C18ACERT_ActiveEvent_modifiedOn] DEFAULT (getdate()),
254.    [id] [int] NULL
255.) ON [PRIMARY]
256.
257.GO
258./****** Object:  Table [dbo].[C18ACERT_Configuration]
    Script Date: 4/16/2019 4:08:40 PM *****/
259.SET ANSI_NULLS ON
260.GO
261.SET QUOTED_IDENTIFIER ON
262.GO
263.CREATE TABLE [dbo].[C18ACERT_Configuration] (
264.    [EquipmentID] [nvarchar](255) NULL,
265.    [EngineSerialNumber] [nvarchar](255) NULL,
266.    [ECMSerialNumber] [nvarchar](255) NULL,
267.    [PersonalityModulePartNumber] [nvarchar](255) NULL,
268.    [PersonalityModuleReleaseDate] [nvarchar](255) NULL,
269.    [RatingNumber] [nvarchar](255) NULL,
270.    [RatedPower] [nvarchar](255) NULL,
271.    [RatedPeakTorque] [nvarchar](255) NULL,
272.    [RatedEngineSpeed] [nvarchar](255) NULL,
273.    [EngineRotation] [nvarchar](255) NULL,
274.    [FuelRatioControlOffset] [nvarchar](255) NULL,
275.    [RatedFuelPosition] [nvarchar](255) NULL,
276.    [LowIdleSpeed] [nvarchar](255) NULL,
277.    [HighIdleSpeed] [nvarchar](255) NULL,
278.    [FuelCorrectionFactor] [nvarchar](255) NULL,
279.    [EngineCoolingSystemConfig] [nvarchar](255) NULL,
280.    [ColdCylinderCutout] [nvarchar](255) NULL,
281.    [CooldownSpeed] [nvarchar](255) NULL,
282.    [CooldownDuration] [nvarchar](255) NULL,
283.    [EnginePreLubeDuration] [nvarchar](255) NULL,
284.    [CrankDuration] [nvarchar](255) NULL,
285.    [MaximumNumberOfCrankCycles] [nvarchar](255) NULL,
286.    [CrankTerminateRPM] [nvarchar](255) NULL,
287.    [SecondaryThrottleEnableStatus] [nvarchar](255) NULL,
288.    [NumberOfSynchronizedEnginesConfiguration]
    [nvarchar](255) NULL,
289.    [TorqueLimitEnableStatus] [nvarchar](255) NULL,
290.    [MaximumEngineTorqueLimit] [nvarchar](255) NULL,
291.    [MinimumEngineTorqueLimit] [nvarchar](255) NULL,
292.    [CoolantWarningEngineLoadEnableThreshold]
    [nvarchar](255) NULL,
293.    [ProgrammableDroopFeatureInstallationStatus]
    [nvarchar](255) NULL,
294.    [EngineSpeedDroop] [nvarchar](255) NULL,
295.    [TransmissionOilTemperatureConfiguration]
    [nvarchar](255) NULL,
296.    [TransmissionOilPressureSensorConfiguration]
    [nvarchar](255) NULL,

```

```

297.         [FuelLevelSensorConfiguration] [nvarchar](255) NULL,
298.         [CoolantLevelSensorConfiguration] [nvarchar](255)
        NULL,
299.         [AirIntakeShutoffDetectionInstallationStatus]
        [nvarchar](255) NULL,
300.         [AftercoolerCoolantLevelSensorInstallationStatus]
        [nvarchar](255) NULL,
301.         [InletAirTemperatureSensor] [nvarchar](255) NULL,
302.         [EngineOilTemperatureSensorInstallationStatus]
        [nvarchar](255) NULL,
303.         [EngineOilLevelSensorInstallationStatus]
        [nvarchar](255) NULL,
304.         [GovernorGainFactor1] [nvarchar](255) NULL,
305.         [GovernorMinimumStabilityFactor1] [nvarchar](255)
        NULL,
306.         [GovernorMaximumStabilityFactor1] [nvarchar](255)
        NULL,
307.         [EngineOilRenewalSystemInstallationStatus]
        [nvarchar](255) NULL,
308.         [EngineOilRenewalSystemEnableStatus] [nvarchar](255)
        NULL,
309.         [OilRenewalRate] [nvarchar](255) NULL,
310.         [FLS] [nvarchar](255) NULL,
311.         [FTS] [nvarchar](255) NULL,
312.         [TotalTattletale] [nvarchar](255) NULL,
313.         [ProgEngLoadFac] [nvarchar](255) NULL,
314.         [GUID] [uniqueidentifier] NULL,
315.         [modifiedOn] [datetime] NULL CONSTRAINT
        [DF_C18ACERT Configuration modifiedOn] DEFAULT (getdate()),
316.         [SoftwareGroupPartNumber] [nvarchar](255) NULL,
317.         [SoftwareGroupReleaseDate] [nvarchar](255) NULL,
318.         [SoftwareGroupDescription] [nvarchar](255) NULL,
319.         [TopEngineSpeedRange] [nvarchar](255) NULL,
320.         [TestSpec] [nvarchar](255) NULL,
321.         [EngineLocation] [nvarchar](255) NULL,
322.         [MaximumTrollingEngineSpeed] [nvarchar](255) NULL,
323.         [TertiaryDesiredSpeedInputConfiguration]
        [nvarchar](255) NULL,
324.         [TachometerFrequencyConfiguration] [nvarchar](255)
        NULL,
325.         [BollardTestTimeRemaining] [nvarchar](255) NULL,
326.         [TransmissionOilTemperatureSensor] [nvarchar](255)
        NULL,
327.         [TransmissionOilPressureSensor] [nvarchar](255) NULL,
328.         [FuelLevelSensorInstallationStatus] [nvarchar](255)
        NULL,
329.         [CoolantLevelSensor] [nvarchar](255) NULL,
330.         [EngineCoolantPressureSensorInstallationStatus]
        [nvarchar](255) NULL,
331.         [UnfilteredEngineOilPressureSensorInstallationStatus]
        [nvarchar](255) NULL,
332.         [UnfilteredFuelPressureSensorInstallationStatus]
        [nvarchar](255) NULL,
333.         [ExhaustTemperatureSensorInstallationStatus]
        [nvarchar](255) NULL,

```

```

334.         [HighExhaustTemperatureWarningSystemEnableStatus]
          [nvarchar](255) NULL,
335.         [EngineOilCapacity] [nvarchar](255) NULL,
336.         [MaintenanceIndicatorMode] [nvarchar](255) NULL,
337.         [PM1Interval] [nvarchar](255) NULL,
338.         [CustomerPassword1] [nvarchar](255) NULL,
339.         [CustomerPassword2] [nvarchar](255) NULL,
340.         [MonSysOvrInstal] [nvarchar](255) NULL
341.) ON [PRIMARY]
342.
343.GO
344./****** Object:  Table [dbo].[C18ACERT_CurrentTotals]
      Script Date: 4/16/2019 4:08:40 PM *****/
345.SET ANSI_NULLS ON
346.GO
347.SET QUOTED_IDENTIFIER ON
348.GO
349.CREATE TABLE [dbo].[C18ACERT_CurrentTotals](
350.         [TotalTime] [nvarchar](255) NULL,
351.         [TotalFuel] [nvarchar](255) NULL,
352.         [TotalIdleTime] [nvarchar](255) NULL,
353.         [TotalIdleFuel] [nvarchar](255) NULL,
354.         [TotalMaxFuel] [nvarchar](255) NULL,
355.         [LifetimeTotalEngineRevolutions] [nvarchar](255) NULL,
356.         [PercentageIdleTime] [nvarchar](255) NULL,
357.         [TotalOperatingHours] [nvarchar](255) NULL,
358.         [AverageRPM] [nvarchar](255) NULL,
359.         [AverageFuelRate] [nvarchar](255) NULL,
360.         [EngineStarts] [nvarchar](255) NULL,
361.         [OverallLoadFactor] [nvarchar](255) NULL,
362.         [DefinitePeriodAveragePercentEngineLoadFactor]
          [nvarchar](255) NULL,
363.         [TotalEngineRatedSpeedOperatingHours] [nvarchar](255)
          NULL,
364.         [PercentageRatedEngineSpeedTime] [nvarchar](255) NULL,
365.         [GUID] [uniqueidentifier] NULL,
366.         [modifiedOn] [datetime] NULL CONSTRAINT
          [DF_C18ACERT_CurretnTotals modifiedOn] DEFAULT (getdate()),
367.         [AverageLoadFactor] [nvarchar](255) NULL,
368.         [StartsHour] [nvarchar](255) NULL
369.) ON [PRIMARY]
370.
371.GO
372./****** Object:  Table [dbo].[C18ACERT_EngineData]      Script
      Date: 4/16/2019 4:08:40 PM *****/
373.SET ANSI_NULLS ON
374.GO
375.SET QUOTED_IDENTIFIER ON
376.GO
377.CREATE TABLE [dbo].[C18ACERT_EngineData](
378.         [Description] [nvarchar](255) NULL,
379.         [BinRange] [nvarchar](255) NULL,
380.         [Hours] [float] NULL,
381.         [GUID] [uniqueidentifier] NULL,

```

```

382.         [modifiedOn] [datetime] NULL CONSTRAINT
        [DF_C18ACERT_EngineData_modifiedOn] DEFAULT (getdate()),
383.         [id] [int] NULL
384.) ON [PRIMARY]
385.
386.GO
387./****** Object: Table [dbo].[C18ACERT_InjectorCodes]
        Script Date: 4/16/2019 4:08:40 PM *****/
388.SET ANSI_NULLS ON
389.GO
390.SET QUOTED_IDENTIFIER ON
391.GO
392.CREATE TABLE [dbo].[C18ACERT_InjectorCodes] (
393.         [Injector1] [nvarchar](255) NULL,
394.         [Injector2] [nvarchar](255) NULL,
395.         [Injector3] [nvarchar](255) NULL,
396.         [Injector4] [nvarchar](255) NULL,
397.         [Injector5] [nvarchar](255) NULL,
398.         [Injector6] [nvarchar](255) NULL,
399.         [Injector7] [nvarchar](255) NULL,
400.         [Injector8] [nvarchar](255) NULL,
401.         [Injector9] [nvarchar](255) NULL,
402.         [Injector10] [nvarchar](255) NULL,
403.         [Injector11] [nvarchar](255) NULL,
404.         [Injector12] [nvarchar](255) NULL,
405.         [Injector13] [nvarchar](255) NULL,
406.         [Injector14] [nvarchar](255) NULL,
407.         [Injector15] [nvarchar](255) NULL,
408.         [Injector16] [nvarchar](255) NULL,
409.         [GUID] [uniqueidentifier] NULL,
410.         [modifiedOn] [datetime] NULL
411.) ON [PRIMARY]
412.
413.GO
414./****** Object: Table [dbo].[C18ACERT_LoggedDiagnostics]
        Script Date: 4/16/2019 4:08:40 PM *****/
415.SET ANSI_NULLS ON
416.GO
417.SET QUOTED_IDENTIFIER ON
418.GO
419.CREATE TABLE [dbo].[C18ACERT_LoggedDiagnostics] (
420.         [Code] [nvarchar](255) NULL,
421.         [Description] [nvarchar](255) NULL,
422.         [Occurence] [nvarchar](255) NULL,
423.         [First] [nvarchar](255) NULL,
424.         [Last] [nvarchar](255) NULL,
425.         [GUID] [uniqueidentifier] NULL,
426.         [modifiedOn] [datetime] NULL CONSTRAINT
        [DF_C18ACERT_LoggedDiagnostic_modifiedOn] DEFAULT (getdate()),
427.         [id] [int] NULL
428.) ON [PRIMARY]
429.
430.GO
431./****** Object: Table [dbo].[C18ACERT_LoggedEvent]      Script
        Date: 4/16/2019 4:08:40 PM *****/

```

```

432.SET ANSI_NULLS ON
433.GO
434.SET QUOTED_IDENTIFIER ON
435.GO
436.CREATE TABLE [dbo].[C18ACERT_LoggedEvent] (
437.    [Code] [nvarchar](255) NULL,
438.    [Description] [nvarchar](255) NULL,
439.    [Occurence] [nvarchar](255) NULL,
440.    [First] [nvarchar](255) NULL,
441.    [Last] [nvarchar](255) NULL,
442.    [GUID] [uniqueidentifier] NULL,
443.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_C18ACERT_LoggedEvent_modifiedOn] DEFAULT (getdate()),
444.    [id] [int] NULL
445.) ON [PRIMARY]
446.
447.GO
448./****** Object:  Table [dbo].[C18ACERT_Monitoring]      Script
    Date: 4/16/2019 4:08:40 PM *****/
449.SET ANSI_NULLS ON
450.GO
451.SET QUOTED_IDENTIFIER ON
452.GO
453.CREATE TABLE [dbo].[C18ACERT_Monitoring] (
454.    [Description] [nvarchar](255) NULL,
455.    [Severity] [int] NULL,
456.    [State] [nvarchar](255) NULL,
457.    [Threshold] [nvarchar](255) NULL,
458.    [Delay] [nvarchar](255) NULL,
459.    [GUID] [uniqueidentifier] NULL,
460.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_C18ACERT_Monitoring_modifiedOn] DEFAULT (getdate()),
461.    [id] [int] NULL
462.) ON [PRIMARY]
463.
464.GO
465./****** Object:  Table [dbo].[C18ACERT_SLEngineReport]
    Script Date: 4/16/2019 4:08:40 PM *****/
466.SET ANSI_NULLS ON
467.GO
468.SET QUOTED_IDENTIFIER ON
469.GO
470.CREATE TABLE [dbo].[C18ACERT_SLEngineReport] (
471.    [EffectsID] [nvarchar](255) NULL,
472.    [SerialNumber] [nvarchar](255) NULL,
473.    [EquipmentID] [nvarchar](255) NULL,
474.    [VesselName] [nvarchar](255) NULL,
475.    [EngineType] [nvarchar](255) NULL,
476.    [EngineModel] [nvarchar](255) NULL,
477.    [EnginePos] [nvarchar](255) NULL,
478.    [DateTime] [datetime] NULL,
479.    [GUID] [uniqueidentifier] NULL,
480.    [modifiedOn] [datetime] NULL CONSTRAINT
    [DF_C18ACERT_SLEngineReport_ModifiedDateTime] DEFAULT
    (getdate())

```

```

481.) ON [PRIMARY]
482.
483.GO
484./****** Object:  Table [dbo].[test4]      Script Date:
      4/16/2019 4:08:40 PM *****/
485.SET ANSI_NULLS ON
486.GO
487.SET QUOTED_IDENTIFIER ON
488.GO
489.CREATE TABLE [dbo].[test4](
490.    [ID] [numeric](18, 0) NOT NULL,
491.    [Name] [nchar](10) NULL,
492.    CONSTRAINT [PK_test4] PRIMARY KEY CLUSTERED
493. (
494.    [ID] ASC
495.)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
      IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
      ON) ON [PRIMARY]
496.) ON [PRIMARY]
497.
498.GO
499./****** Object:  View [dbo].[Union_Configuration]      Script
      Date: 4/16/2019 4:08:40 PM *****/
500.SET ANSI_NULLS ON
501.GO
502.SET QUOTED_IDENTIFIER ON
503.GO
504.CREATE VIEW [dbo].[Union_Configuration]
505.AS
506.
507. /***** Script for SelectTopNRows command from SSMS
      *****/
508.SELECT TOP 1000 [EquipmentID]
509.    , [EngineSerialNumber]
510.    , [ECMSerialNumber]
511.    , [PersonalityModulePartNumber]
512.    , [PersonalityModuleReleaseDate]
513.    , [RatingNumber]
514.    , [RatedPower]
515.    , [RatedPeakTorque]
516.    , [RatedEngineSpeed]
517.    , [EngineRotation]
518.    , [FuelRatioControlOffset]
519.    , [RatedFuelPosition]
520.    , [LowIdleSpeed]
521.    , [HighIdleSpeed]
522.    , [FuelCorrectionFactor]
523.    , [EngineCoolingSystemConfig]
524.    , [ColdCylinderCutout]
525.    , [CooldownSpeed]
526.    , [CooldownDuration]
527.    , [EnginePreLubeDuration]
528.    , [CrankDuration]
529.    , [MaximumNumberofCrankCycles]
530.    , [CrankTerminateRPM]

```

```

531.      , [SecondaryThrottleEnableStatus]
532.      , [NumberOfSynchronizedEnginesConfiguration]
533.      , [TorqueLimitEnableStatus]
534.      , [MaximumEngineTorqueLimit]
535.      , [MinimumEngineTorqueLimit]
536.      , [CoolantWarningEngineLoadEnableThreshold]
537.      , [ProgrammableDroopFeatureInstallationStatus]
538.      , [EngineSpeedDroop]
539.      , [TransmissionOilTemperatureConfiguration]
540.      , [TransmissionOilPressureSensorConfiguration]
541.      , [FuelLevelSensorConfiguration]
542.      , [CoolantLevelSensorConfiguration]
543.      , [AirIntakeShutoffDetectionInstallationStatus]
544.      , [AftercoolerCoolantLevelSensorInstallationStatus]
545.      , [InletAirTemperatureSensor]
546.      , [EngineOilTemperatureSensorInstallationStatus]
547.      , [EngineOilLevelSensorInstallationStatus]
548.      , [GovernorGainFactor1]
549.      , [GovernorMinimumStabilityFactor1]
550.      , [GovernorMaximumStabilityFactor1]
551.      , [EngineOilRenewalSystemInstallationStatus]
552.      , [EngineOilRenewalSystemEnableStatus]
553.      , [OilRenewalRate]
554.      , [FLS]
555.      , [FTS]
556.      , [TotalTattletale]
557.      , [ProgEngLoadFac]
558.      , [GUID]
559.      , [modifiedOn]
560.      , [SoftwareGroupPartNumber]
561.      , [SoftwareGroupReleaseDate]
562.      , [SoftwareGroupDescription]
563.      , [TopEngineSpeedRange]
564.      , [TestSpec]
565.      , [EngineLocation]
566.      , [MaximumTrollingEngineSpeed]
567.      , [TertiaryDesiredSpeedInputConfiguration]
568.      , [TachometerFrequencyConfiguration]
569.      , [BollardTestTimeRemaining]
570.      , [TransmissionOilTemperatureSensor]
571.      , [TransmissionOilPressureSensor]
572.      , [FuelLevelSensorInstallationStatus]
573.      , [CoolantLevelSensor]
574.      , [EngineCoolantPressureSensorInstallationStatus]
575.      , [UnfilteredEngineOilPressureSensorInstallationStatus]
576.      , [UnfilteredFuelPressureSensorInstallationStatus]
577.      , [ExhaustTemperatureSensorInstallationStatus]
578.      , [HighExhaustTemperatureWarningSystemEnableStatus]
579.      , [EngineOilCapacity]
580.      , [MaintenanceIndicatorMode]
581.      , [PM1Interval]
582.      , [CustomerPassword1]
583.      , [CustomerPassword2]
584.      , [MonSysOvrInstal]
585. FROM [SLDASH_A].[dbo].[C18ACERT_Configuration]

```

```

586.
587.  union all
588.
589.  /***** Script for SelectTopNRows command from SSMS
        *****/
590. SELECT TOP 1000 [EquipmentID]
591.             , [EngineSerialNumber]
592.             , [ECMSerialNumber]
593.             , [PersonalityModulePartNumber]
594.             , [PersonalityModuleReleaseDate]
595.             , [RatingNumber]
596.             , [RatedPower]
597.             , [RatedPeakTorque]
598.             , [RatedEngineSpeed]
599.             , [EngineRotation]
600.             , [FuelRatioControlOffset]
601.             , [RatedFuelPosition]
602.             , [LowIdleSpeed]
603.             , [HighIdleSpeed]
604.             , [FuelCorrectionFactor]
605.             , [EngineCoolingSystemConfig]
606.             , [ColdCylinderCutout]
607.             , [CooldownSpeed]
608.             , [CooldownDuration]
609.             , [EnginePreLubeDuration]
610.             , [CrankDuration]
611.             , [MaximumNumberOfCrankCycles]
612.             , [CrankTerminateRPM]
613.             , [SecondaryThrottleEnableStatus]
614.             , [NumberOfSynchronizedEnginesConfiguration]
615.             , [TorqueLimitEnableStatus]
616.             , [MaximumEngineTorqueLimit]
617.             , [MinimumEngineTorqueLimit]
618.             , [CoolantWarningEngineLoadEnableThreshold]
619.             , [ProgrammableDroopFeatureInstallationStatus]
620.             , [EngineSpeedDroop]
621.             , [TransmissionOilTemperatureConfiguration]
622.             , [TransmissionOilPressureSensorConfiguration]
623.             , [FuelLevelSensorConfiguration]
624.             , [CoolantLevelSensorConfiguration]
625.             , [AirIntakeShutoffDetectionInstallationStatus]
626.             , [AftercoolerCoolantLevelSensorInstallationStatus]
627.             , [InletAirTemperatureSensor]
628.             , [EngineOilTemperatureSensorInstallationStatus]
629.             , [EngineOilLevelSensorInstallationStatus]
630.             , [GovernorGainFactor1]
631.             , [GovernorMinimumStabilityFactor1]
632.             , [GovernorMaximumStabilityFactor1]
633.             , [EngineOilRenewalSystemInstallationStatus]
634.             , [EngineOilRenewalSystemEnableStatus]
635.             , [OilRenewalRate]
636.             , [FLS]
637.             , [FTS]
638.             , [TotalTattletale]
639.             , [ProgEngLoadFac]

```

```

640.         , [GUID]
641.         , [modifiedOn],
642.         '0' as [SoftwareGroupPartNumber],
643.         '0' as [SoftwareGroupReleaseDate],
644.         '0' as [SoftwareGroupDescription],
645.         '0' as [TopEngineSpeedRange],
646.         '0' as [TestSpec]
647.         , '0' as [EngineLocation]
648.         , '0' as [MaximumTrollingEngineSpeed]
649.         , '0' as [TertiaryDesiredSpeedInputConfiguration]
650.         , '0' as [TachometerFrequencyConfiguration]
651.         , '0' as [BollardTestTimeRemaining]
652.         , '0' as [TransmissionOilTemperatureSensor]
653.         , '0' as [TransmissionOilPressureSensor]
654.         , '0' as [FuelLevelSensorInstallationStatus]
655.         , '0' as [CoolantLevelSensor]
656.         , '0' as [EngineCoolantPressureSensorInstallationStatus]
657.         , '0'
        as [UnfilteredEngineOilPressureSensorInstallationStatus]
658.         , '0' as [UnfilteredFuelPressureSensorInstallationStatus]
659.         , '0' as [ExhaustTemperatureSensorInstallationStatus]
660.         , '0'
        as [HighExhaustTemperatureWarningSystemEnableStatus]
661.         , '0' as [EngineOilCapacity]
662.         , '0' as [MaintenanceIndicatorMode]
663.         , '0' as [PM1Interval]
664.         , '0' as [CustomerPassword1]
665.         , '0' as [CustomerPassword2]
666.         , '0' as [MonSysOvrInstal]
667. FROM [SLDASH_A].[dbo].[3516_Configuration]
668.
669.
670.
671.
672. GO
673. /***** Object:  View [dbo].[Union_SLEngineReport]      Script
        Date: 4/16/2019 4:08:40 PM *****/
674. SET ANSI_NULLS ON
675. GO
676. SET QUOTED_IDENTIFIER ON
677. GO
678. CREATE VIEW [dbo].[Union_SLEngineReport]
679. AS
680. /***** Script for SelectTopNRows command from SSMS *****/
681. SELECT TOP 1000 [EffectsID]
682.         , [SerialNumber]
683.         , [EquipmentID]
684.         , [VesselName]
685.         , [EngineType]
686.         , [EngineModel]
687.         , [EnginePos]
688.         , [DateTime]
689.         , [GUID]
690.         , [modifiedOn]
691. FROM [SLDASH_A].[dbo].[3516_SLEngineReport]

```

```

692. UNION ALL
693. SELECT TOP 1000 [EffectsID]
694.      , [SerialNumber]
695.      , [EquipmentID]
696.      , [VesselName]
697.      , [EngineType]
698.      , [EngineModel]
699.      , [EnginePos]
700.      , [DateTime]
701.      , [GUID]
702.      , [modifiedOn]
703. FROM [SLDASH_A].[dbo].[C18ACERT_SLEngineReport]
704.
705. GO
706. /***** Object: View [dbo].[SL_Engines_Configurations]
       Script Date: 4/16/2019 4:08:40 PM *****/
707. SET ANSI_NULLS ON
708. GO
709. SET QUOTED_IDENTIFIER ON
710. GO
711. CREATE VIEW [dbo].[SL_Engines_Configurations]
712. AS
713. SELECT      dbo.Union_SLEngineReport.EffectsID,
714.      dbo.Union_SLEngineReport.VesselName,
715.      dbo.Union_SLEngineReport.EngineType,
716.      dbo.Union_SLEngineReport.EngineModel,
717.      dbo.Union_SLEngineReport.DateTime,
718.      dbo.Union_Configuration.*
719. FROM      dbo.Union_SLEngineReport INNER JOIN
720.      dbo.Union_Configuration ON
721.      dbo.Union_SLEngineReport.GUID = dbo.Union_Configuration.GUID
722.
723. GO
724. ALTER TABLE [dbo].[C18ACERT_InjectorCodes] ADD CONSTRAINT
725.      [DF_C18ACERT_InjectorCodes_modifiedOn] DEFAULT (getdate()) FOR
726.      [modifiedOn]
727. GO

```

Gemaakte views

```

1. USE [SL_ENGINE_PERFORMANCE_T]
2. GO
3. /***** Object: View [dbo].[Union_ActiveDiagnostics] Script
   Date: 16-4-2019 17:07:34 *****/
4. SET ANSI_NULLS ON
5. GO
6. SET QUOTED_IDENTIFIER ON
7. GO
8. CREATE VIEW [dbo].[Union_ActiveDiagnostics]
9. AS
10. /***** Script for SelectTopNRows command from SSMS *****/
11. SELECT [Code], [Description], [GUID], [modifiedOn], [id]
12. FROM
13.      [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_ActiveDiagnostics]
14. UNION ALL

```

```

13. /***** Script for SelectTopNRows command from SSMS *****/
    SELECT [Code], [Description], [GUID], [modifiedOn], [id]
14. FROM
    [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_ActiveDiagnostics]
15. GO
16. /***** Object: View [dbo].[Union_ActiveEvent]      Script
    Date: 16-4-2019 17:07:34 *****/
17. SET ANSI_NULLS ON
18. GO
19. SET QUOTED_IDENTIFIER ON
20. GO
21. CREATE VIEW [dbo].[Union_ActiveEvent]
22. AS
23. /***** Script for SelectTopNRows command from SSMS *****/
    SELECT [Code], [Description], [GUID], [modifiedOn], [id]
24. FROM
    [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_ActiveEvent]
25. UNION ALL
26. /***** Script for SelectTopNRows command from SSMS *****/
    SELECT [Code], [Description], [GUID], [modifiedOn], [id]
27. FROM
    [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_ActiveEvent]
28. GO
29. /***** Object: View [dbo].[Union_Configuration]      Script
    Date: 16-4-2019 17:07:34 *****/
30. SET ANSI_NULLS ON
31. GO
32. SET QUOTED_IDENTIFIER ON
33. GO
34. CREATE VIEW [dbo].[Union_Configuration]
35. AS
36. /***** Script for SelectTopNRows command from SSMS *****/
    SELECT [EquipmentID], [EngineSerialNumber], [ECMSerialNumber],
    [PersonalityModulePartNumber], [PersonalityModuleReleaseDate],
    [RatingNumber],
37.         [RatedPower], [RatedPeakTorque],
    [RatedEngineSpeed], [EngineRotation], [FuelRatioControlOffset],
    [RatedFuelPosition], [LowIdleSpeed], [HighIdleSpeed],
    [FuelCorrectionFactor], [EngineCoolingSystemConfig],
38.         [ColdCylinderCutout],
    [CooldownSpeed], [CooldownDuration], [EnginePreLubeDuration],
    [CrankDuration], [MaximumNumberofCrankCycles],
    [CrankTerminateRPM], [SecondaryThrottleEnableStatus],
39.         [NumberofSynchronizedEnginesConfiguration],
    [TorqueLimitEnableStatus], [MaximumEngineTorqueLimit],
    [MinimumEngineTorqueLimit],
    [CoolantWarningEngineLoadEnableThreshold],
40.         [ProgrammableDroopFeatureInstallationStatus],
    [EngineSpeedDroop], [TransmissionOilTemperatureConfiguration],
    [TransmissionOilPressureSensorConfiguration],
    [FuelLevelSensorConfiguration],
41.         [CoolantLevelSensorConfiguration],
    [AirIntakeShutoffDetectionInstallationStatus],

```

```

[AftercoolerCoolantLevelSensorInstallationStatus],
[InletAirTemperatureSensor],
[EngineOilTemperatureSensorInstallationStatus],
42. [EngineOilLevelSensorInstallationStatus],
[GovernorGainFactor1], [GovernorMinimumStabilityFactor1],
[GovernorMaximumStabilityFactor1],
[EngineOilRenewalSystemInstallationStatus],
[EngineOilRenewalSystemEnableStatus],
43. [OilRenewalRate], [FLS], [FTS],
[TotalTattletale], [ProgEngLoadFac], [GUID], [modifiedOn],
[SoftwareGroupPartNumber], [SoftwareGroupReleaseDate],
[SoftwareGroupDescription], [TopEngineSpeedRange], [TestSpec],
44. [EngineLocation],
[MaximumTrollingEngineSpeed],
[TertiaryDesiredSpeedInputConfiguration],
[TachometerFrequencyConfiguration], [BollardTestTimeRemaining],
[TransmissionOilTemperatureSensor],
45. [TransmissionOilPressureSensor],
[FuelLevelSensorInstallationStatus], [CoolantLevelSensor],
[EngineCoolantPressureSensorInstallationStatus],
[UnfilteredEngineOilPressureSensorInstallationStatus],
46. [UnfilteredFuelPressureSensorInstallationStatus],
[ExhaustTemperatureSensorInstallationStatus],
[HighExhaustTemperatureWarningSystemEnableStatus],
[EngineOilCapacity], [MaintenanceIndicatorMode], [PM1Interval],
47. [CustomerPassword1],
[CustomerPassword2], [MonSysOvrInstal]
48. FROM [dbo].[C18ACERT_Configuration]
49. UNION ALL
50. /***** Script for SelectTopNRows command from SSMS *****/
SELECT [EquipmentID], [EngineSerialNumber], [ECMSerialNumber],
[PersonalityModulePartNumber], [PersonalityModuleReleaseDate],
[RatingNumber],
51. [RatedPower], [RatedPeakTorque],
[RatedEngineSpeed], [EngineRotation], [FuelRatioControlOffset],
[RatedFuelPosition], [LowIdleSpeed], [HighIdleSpeed],
[FuelCorrectionFactor], [EngineCoolingSystemConfig],
52. [ColdCylinderCutout],
[CooldownSpeed], [CooldownDuration], [EnginePreLubeDuration],
[CrankDuration], [MaximumNumberOfCrankCycles],
[CrankTerminateRPM], [SecondaryThrottleEnableStatus],
53. [NumberOfSynchronizedEnginesConfiguration],
[TorqueLimitEnableStatus], [MaximumEngineTorqueLimit],
[MinimumEngineTorqueLimit],
[CoolantWarningEngineLoadEnableThreshold],
54. [ProgrammableDroopFeatureInstallationStatus],
[EngineSpeedDroop], [TransmissionOilTemperatureConfiguration],
[TransmissionOilPressureSensorConfiguration],
[FuelLevelSensorConfiguration],
55. [CoolantLevelSensorConfiguration],
[AirIntakeShutoffDetectionInstallationStatus],
[AftercoolerCoolantLevelSensorInstallationStatus],

```

```

[InletAirTemperatureSensor],
[EngineOilTemperatureSensorInstallationStatus],
56. [EngineOilLevelSensorInstallationStatus],
[GovernorGainFactor1], [GovernorMinimumStabilityFactor1],
[GovernorMaximumStabilityFactor1],
[EngineOilRenewalSystemInstallationStatus],
[EngineOilRenewalSystemEnableStatus],
57. [OilRenewalRate], [FLS], [FTS],
[TotalTattletale], [ProgEngLoadFac], [GUID], [modifiedOn], '0'
AS [SoftwareGroupPartNumber], '0' AS
[SoftwareGroupReleaseDate], '0' AS [SoftwareGroupDescription],
'0' AS [TopEngineSpeedRange],
58. '0' AS [TestSpec], '0' AS
[EngineLocation], '0' AS [MaximumTrollingEngineSpeed], '0' AS
[TertiaryDesiredSpeedInputConfiguration], '0' AS
[TachometerFrequencyConfiguration], '0' AS
[BollardTestTimeRemaining],
59. '0' AS
[TransmissionOilTemperatureSensor], '0' AS
[TransmissionOilPressureSensor], '0' AS
[FuelLevelSensorInstallationStatus], '0' AS
[CoolantLevelSensor], '0' AS
[EngineCoolantPressureSensorInstallationStatus],
60. '0' AS
[UnfilteredEngineOilPressureSensorInstallationStatus], '0' AS
[UnfilteredFuelPressureSensorInstallationStatus], '0' AS
[ExhaustTemperatureSensorInstallationStatus],
61. '0' AS
[HighExhaustTemperatureWarningSystemEnableStatus], '0' AS
[EngineOilCapacity], '0' AS [MaintenanceIndicatorMode], '0' AS
[PM1Interval], '0' AS [CustomerPassword1], '0' AS
[CustomerPassword2],
62. '0' AS [MonSysOvrInstal]
63. FROM [dbo].[3516_Configuration]
64. GO
65. /***** Object: View [dbo].[Union_CurrentTotals] Script
Date: 16-4-2019 17:07:34 *****/
66. SET ANSI_NULLS ON
67. GO
68. SET QUOTED_IDENTIFIER ON
69. GO
70. CREATE VIEW [dbo].[Union_CurrentTotals]
71. AS
72. /***** Script for SelectTopNRows command from SSMS *****/
73. SELECT [TotalTime]
74. , [TotalFuel]
75. , [TotalIdleTime]
76. , [TotalIdleFuel]
77. , [TotalMaxFuel]
78. , [LifetimeTotalEngineRevolutions]
79. , [PercentageIdleTime]
80. , [TotalOperatingHours]
81. , [AverageRPM]
82. , [AverageFuelRate]
83. , [EngineStarts]

```

```

84.         , [OverallLoadFactor]
85.         , [DefinitePeriodAveragePercentEngineLoadFactor]
86.         , [TotalEngineRatedSpeedOperatingHours]
87.         , [PercentageRatedEngineSpeedTime]
88.         , [GUID]
89.         , [modifiedOn]
90.         , '0' as [AverageLoadFactor]
91.         , '0' as [StartsHour]
92.     FROM [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_CurrentTotals]
93.
94.     Union All
95.
96.     /***** Script for SelectTopNRows command from SSMS
*****/
97. SELECT [TotalTime]
98.         , [TotalFuel]
99.         , [TotalIdleTime]
100.        , [TotalIdleFuel]
101.        , [TotalMaxFuel]
102.        , [LifetimeTotalEngineRevolutions]
103.        , [PercentageIdleTime]
104.        , [TotalOperatingHours]
105.        , [AverageRPM]
106.        , [AverageFuelRate]
107.        , [EngineStarts]
108.        , [OverallLoadFactor]
109.        , [DefinitePeriodAveragePercentEngineLoadFactor]
110.        , [TotalEngineRatedSpeedOperatingHours]
111.        , [PercentageRatedEngineSpeedTime]
112.        , [GUID]
113.        , [modifiedOn]
114.        , [AverageLoadFactor]
115.        , [StartsHour]
116.     FROM
        [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_CurrentTotals]
117. GO
118. /***** Object: View [dbo].[Union_EngineData] Script
        Date: 16-4-2019 17:07:34 *****/
119. SET ANSI_NULLS ON
120. GO
121. SET QUOTED_IDENTIFIER ON
122. GO
123. CREATE VIEW [dbo].[Union_EngineData]
124. AS
125. /***** Script for SelectTopNRows command from SSMS *****/
126. SELECT [Description]
127.         , [BinRange]
128.         , [Hours]
129.         , [GUID]
130.         , [modifiedOn]
131.         , [id]
132.     FROM [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_EngineData]
133.
134.     Union All
135.

```

```

136.  /***** Script for SelectTopNRows command from SSMS
      *****/
137. SELECT [Description]
138.        , [BinRange]
139.        , [Hours]
140.        , [GUID]
141.        , [modifiedOn]
142.        , [id]
143. FROM [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_EngineData]
144. GO
145. /***** Object: View [dbo].[Union_LoggedDiagnostics]
      Script Date: 16-4-2019 17:07:34 *****/
146. SET ANSI_NULLS ON
147. GO
148. SET QUOTED_IDENTIFIER ON
149. GO
150. CREATE VIEW [dbo].[Union_LoggedDiagnostics]
151. AS
152. /***** Script for SelectTopNRows command from SSMS *****/
      SELECT [Code], [Description], [Occurence], [First], [Last],
      [GUID], [modifiedOn], [id]
153. FROM
      [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_LoggedDiagnostics]
154. UNION ALL
155. /***** Script for SelectTopNRows command from SSMS *****/
      SELECT [Code], [Description], [Occurence], [First], [Last],
      [GUID], [modifiedOn], [id]
156. FROM
      [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_LoggedDiagnostics]
157. GO
158. /***** Object: View [dbo].[Union_LoggedEvent]      Script
      Date: 16-4-2019 17:07:34 *****/
159. SET ANSI_NULLS ON
160. GO
161. SET QUOTED_IDENTIFIER ON
162. GO
163. CREATE VIEW [dbo].[Union_LoggedEvent]
164. AS
165. /***** Script for SelectTopNRows command from SSMS *****/
      SELECT [Code], [Description], [Occurence], [First], [Last],
      [GUID], [modifiedOn], [id]
166. FROM
      [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_LoggedEvent]
167. UNION ALL
168. /***** Script for SelectTopNRows command from SSMS *****/
      SELECT [Code], [Description], [Occurence], [First], [Last],
      [GUID], [modifiedOn], [id]
169. FROM
      [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_LoggedEvent]
170. GO
171. /***** Object: View [dbo].[Union_Monitoring]      Script
      Date: 16-4-2019 17:07:34 *****/
172. SET ANSI_NULLS ON
173. GO
174. SET QUOTED_IDENTIFIER ON

```

```

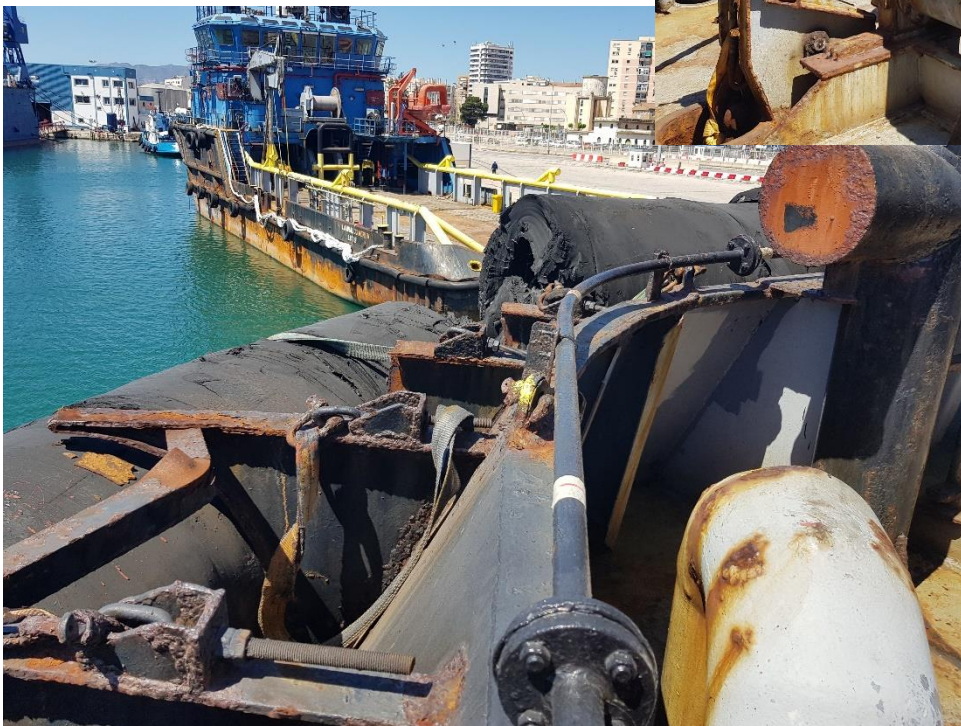
175.GO
176.CREATE VIEW [dbo].[Union_Monitoring]
177.AS
178./***** Script for SelectTopNRows command from SSMS *****/
    SELECT [Description], [Severity], [State], [Threshold],
    [Delay], [GUID], [modifiedOn], [id]
179.FROM
    [SL_ENGINE_PERFORMANCE_T].[dbo].[3516_Monitoring]
180.UNION ALL
181./***** Script for SelectTopNRows command from SSMS *****/
    SELECT [Description], [Severity], [State], [Threshold],
    [Delay], [GUID], [modifiedOn], [id]
182.FROM
    [SL_ENGINE_PERFORMANCE_T].[dbo].[C18ACERT_Monitoring]
183.GO
184./***** Object: View [dbo].[Union_SLEngineReport] Script
    Date: 16-4-2019 17:07:34 *****/
185.SET ANSI_NULLS ON
186.GO
187.SET QUOTED_IDENTIFIER ON
188.GO
189.CREATE VIEW [dbo].[Union_SLEngineReport]
190.AS
191./***** Script for SelectTopNRows command from SSMS *****/
    SELECT [EffactsID], [SerialNumber], [EquipmentID],
    [VesselName], [EngineType], [EngineModel], [EnginePos],
    [DateTime], [GUID], [modifiedOn]
192.FROM [dbo].[3516_SLEngineReport]
193.UNION ALL
194.SELECT [EffactsID], [SerialNumber], [EquipmentID],
    [VesselName], [EngineType], [EngineModel], [EnginePos],
    [DateTime], [GUID], [modifiedOn]
195.FROM [dbo].[C18ACERT_SLEngineReport]
196.GO

```

Bijlage 14. Tour door de Malkoha



Aangekomen op het schip Lamnalco Malkoha, is er eerst een tour rond het schip gemaakt. Duidelijk werd waarom dit schip al een aantal maanden aan wal ligt. Overal is wel wat kapot of verroest, Sander Woltheus heeft ook voor het vertrek verteld dat, “Dit schip is precies wat we willen voorkomen met jouw stuk.”



Na de tour rond het schip is de Engine Room betreden, hierin staan alle motoren welke nodig zijn om het schip van zowel voortstuwing als stroom te voorzien. In de Engine Room zijn er een aantal foto's gemaakt van de twee diesel generatoren waarvan readings gingen worden genomen.



Figuur 39 CAT motoren aanboord de SL Malkoha

Beide motoren zijn van het model C18ACERT, alleen omdat het hier gaat om auxiliary engines en niet om main engines, krijgen deze in CAT Electronic Technician het type C18GenSet toegewezen. Helaas had dit tot gevolg dat de hele 'Data Highway' van Engine tot Database niet gebruikt kon worden. Dit omdat in de database voorlopig alleen rekening gehouden wordt met de modellen 3516B en C18ACERT main engines.

Bijlage 15. Verbinding Engines-CAT ET

Na het bekijken van de engines is er connectie gelegd met allebei en zijn readings genomen met CAT ET. Benodigdheden voor het maken van readings zijn het volgende.



Figuur 40 Benodigdheden voor het uitlezen van CAT motoren

- 1 Een USB naar VGA kabel
- 2 Een Laptop met CAT ET
- 3 Een OBD naar VGA kabel
- 4 Een CAT Connector

Zoek naar een plug als onderstaand en haal het kapje er af



Figuur 42 OBD plug met kap

Figuur 41 OBD Plug zonder kap



Figuur 45 Plug de OBD kabel in de plug



Figuur 43 OBD Kabel in plug

Plug de OBD kabel in de plug, maar maak het jezelf niet te moeilijk.

Connect de USB naar VGA kabel in de onderste VGA connectie van de CAT Connector, dit kan alvorens en na het inpluggen in de OBD plug.



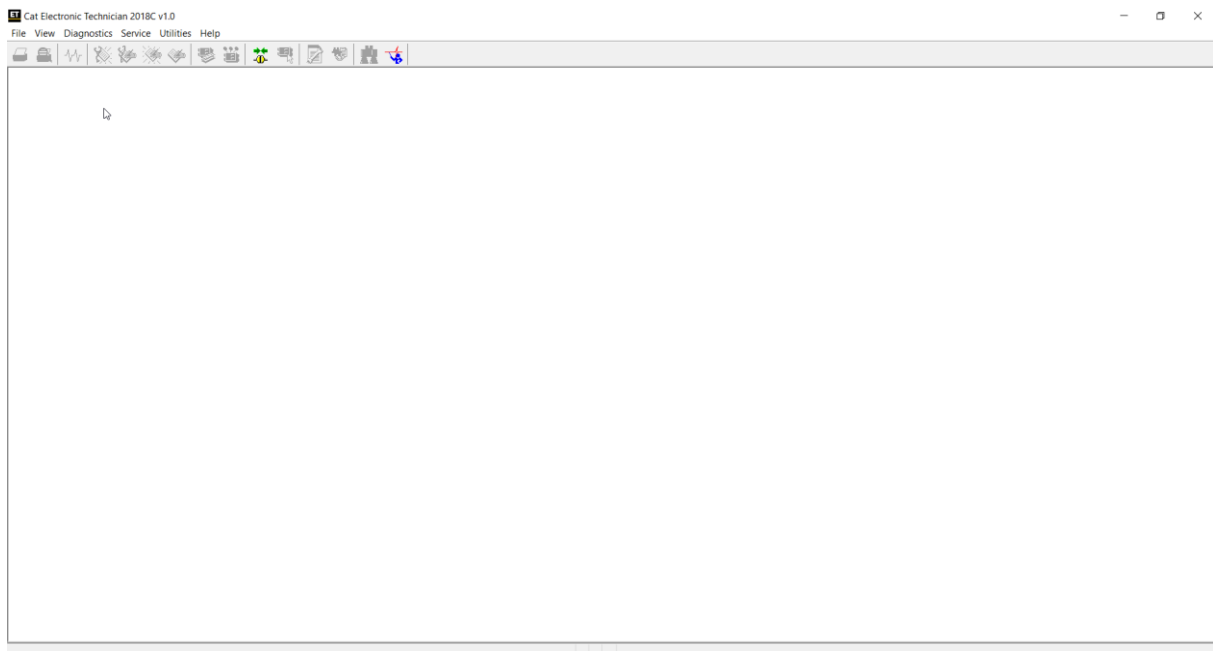
Figuur 44 Connector en OBD kabel



Wanneer de CAT Connector 'Power' lampje rood gloeit is connectie gelukt. Stop de usb kabel in de pc met CAT ET en start CAT Electronic Technician op.

Figuur 46 Connector staat aan

Het start scherm wordt getoond en de connectie met de Engine kan worden gestart.



Figuur 47 Hoofdscherm CAT ET

ET Cat Electronic Technician 2018C v1.0

File View Diagnostics Service Utilities Help

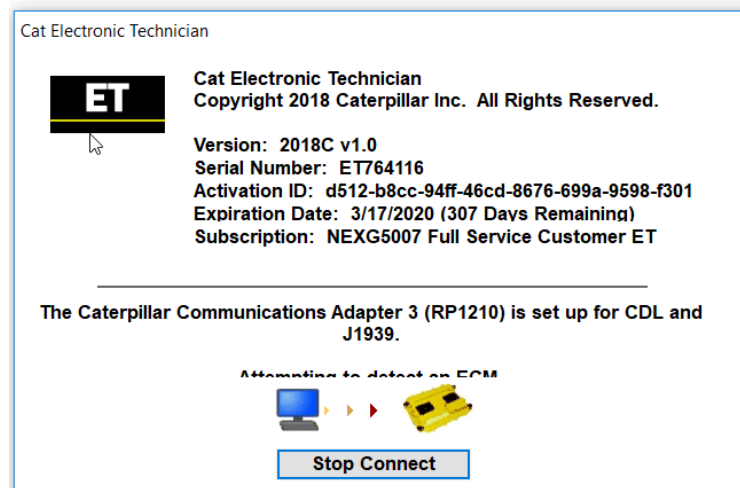


Figuur 50 Connect met de motor

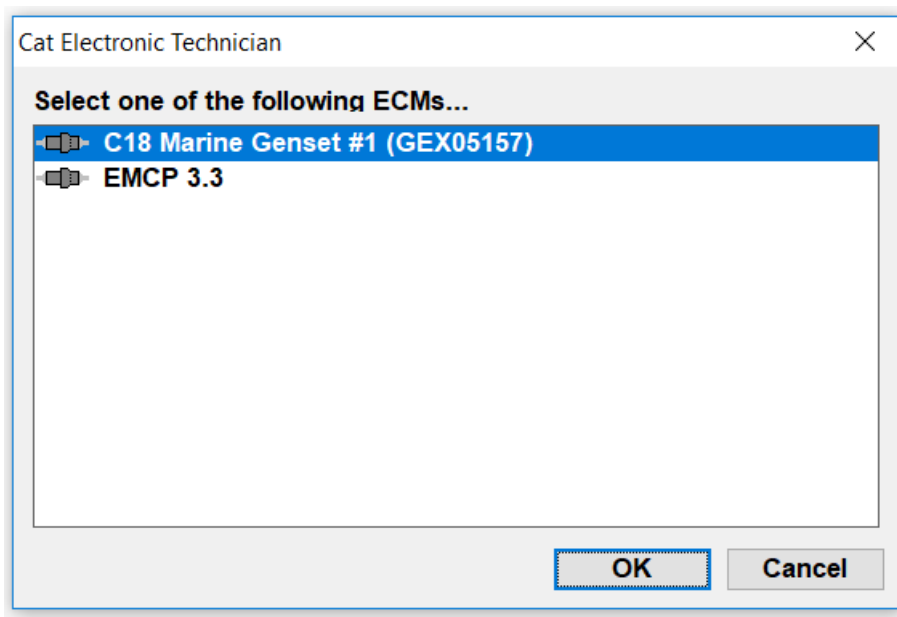


Figuur 48 J1939 lampje flikkert

Wanneer men op 'Connect' drukt wordt de connectie met de Electronic Control Modules (ECM's) van de Engine gestart. Tijdens het connecten zal het 'J1939' lampje op de connector fllikkeren.



Figuur 49 CAT ET maakt verbinding



Figuur 51 Selecteer de ECM

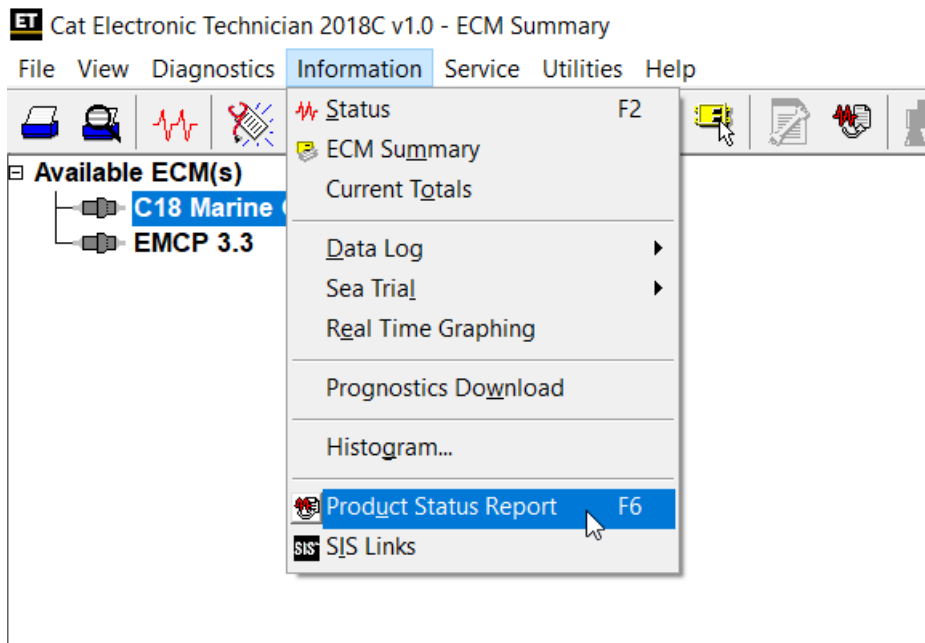
Wacht tot het connect pop up scherm verdwenen is en selecteer de juiste ECM tussen de lijst.

Na het selecteren van de juist ECM, krijg je een overzicht te zien van algemene data welke van de verschillende EMC's.

Description	Value
C18 Marine Genset #1 (GEX05157)	
Equipment ID	GEN 1
Engine Serial Number	GEX05157
ECM Serial Number	29356108JV
Software Group Part Number	4730568-00
Software Group Release Date	MAR15
Software Group Description	C18_TIER2_A4E4v2
EMCP 3.3	
ECM Part Number	3007647-03
ECM Serial Number	3513B028TB
Software Group Part Number	3222285-00
Software Group Release Date	NOV2007
Software Group Description	L2 11_0 Standard -- English, 2.2 Prod, 7Nov2007

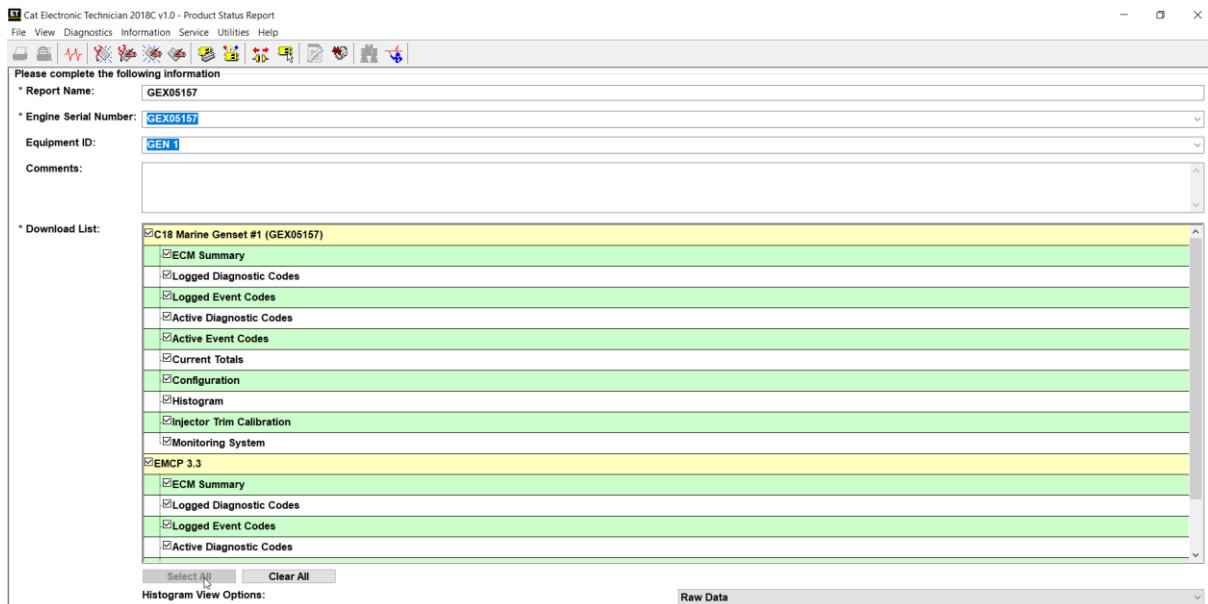
Figuur 52 Overzicht na selectie van ECM

Om een rapport te generen gaat men naar 'Information' en vervolgens naar 'Product Status Report'



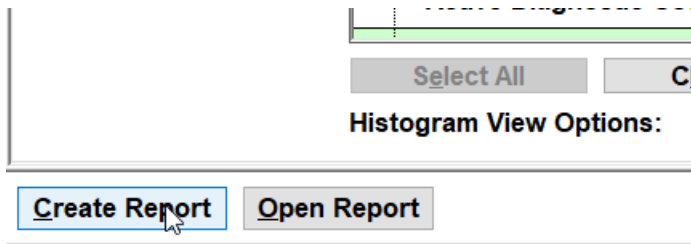
Figuur 53 Genereren van een rapport

Na het klikken op 'Product Status Report' wordt men geleid naar een ander scherm waarin verschillende headers staan. Deze headers vormen de verschillende soorten data te vinden in de rapporten.



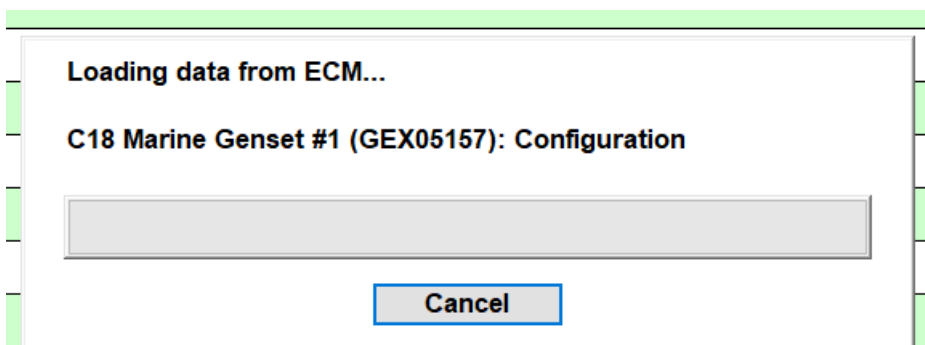
Figuur 54 Overzicht van de keuzes voor het rapport

Om een zo duidelijk en volledig mogelijk rapport te genereren is het belangrijk om op de 'Select All' knop te klikken. Dit zorgt dat alle data welke mogelijk is, ook daadwerkelijk uit de ECM's wordt getrokken.



Links onder de 'Select All' knop bevindt zich de 'Create Rpeort' knop. Klik hierop om een rapport van de zojuist geselecteerde data te generen.

Figuur 55 Genereer het rapport



Men krijgt een pop up scherm te zien, wacht geduldig tot deze verdwijnt en men wordt gegroet men het gemaakte rapport.

Figuur 56 Wacht tot het genereren klaar is

Cat Electronic Technician 2018C v1.0
Product Status Report
5/15/2019 2:38 PM

- C18 Marine Genset #1 (GEX05157)
- EMCP 3.3

Product Status Report

Parameter	Value
Engine Serial Number	GEX05157
Equipment ID	GEN 1
Comments	

C18 Marine Genset #1 (GEX05157)

Parameter	Value
Equipment ID	GEN 1
Engine Serial Number	GEX05157
ECM Serial Number	29356108JV
Software Group Part Number	4730568-00
Software Group Release Date	MAR15
Software Group Description	C18_TIER2_A4E4v2

Figuur 58 Het gemaakte rapport

Het rapport is gemaakt en klaar voor export. Klik op de 'Save Additional Copy' en sla het rapport op als zowel een XML als PDF bestand. Het XML bestand kan worden gebruikt om te transformeren en het PDF bestand als goed overzicht.

Occ.	First	Last

Save Additional Copy

Figuur 57 Nieuwe kopie opslaan

Save As

Save in: Product Status Reports

Name	Date modified	Type
CYG00246_13-05-2019_PSRPT_2019-05-13_1...	5/13/2019 2:19 PM	XML Document
GEX05157_PSRPT_2019-05-13_16.11.01	5/13/2019 4:11 PM	XML Document
GEX05157_PSRPT_2019-05-15_14.38.20	5/15/2019 2:38 PM	XML Document

File name: GEX05157_PSRPT_2019-05-15_14.42.31

Save as type: XML Files (*.xml)

Save **Cancel**

Save As

Save in: Product Status Reports

Name	Date modified	Type
CYG00246_13-05-2019_PSRPT_2019-05-13_1...	5/13/2019 2:19 PM	PDF File
GEX05157_PSRPT_2019-05-13_16.11.01	5/13/2019 4:11 PM	PDF File
GEX05157_PSRPT_2019-05-15_14.38.20	5/15/2019 2:38 PM	PDF File

File name: GEX05157_PSRPT_2019-05-15_14.42.01

Save as type: PDF Files (*.pdf)

Save **Cancel**

Figuur 60 Opslaan als XML

Figuur 61 Opslaan als PDF

Cat Electronic Technician 2018C v1.0 - Product Status Report

File View Diagnostics Information Service Utilities Help

Please complete the following information

* Report Name: **GEX05157**

Disconnect (F8)

Figuur 59 Disconnect van de motor

Sluit het rapport en disconnect van de engine.

Logged Diagnostic Codes [Diagnostic Code]

No Logged Diagnostic Codes

Close Report

Figuur 62 Sluit het rapport

Bijlage 16. Test Cases

TestCase 1 - Start

Test	TC01
Use case	UC1 Start
Test scenario	Check of het starten van de applicatie lukt.
Pre-conditie	Het programma is nog niet gestart
Testproces	1. Open de .exe van de applicatie
Verwachte Uitkomst	1. Applicatie start 2. Applicatie laat startscherm zien 3. Applicatie laat lijst met schepen zien
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 2 - Stop

Test	TC02.1
Use case	UC2 Stop
Test scenario	Check het proces bij het stoppen van de applicatie via button
Pre-conditie	Het programma is gestart
Testproces	1. Druk op de 'Close' knop
Verwachte Uitkomst	1. Applicatie sluit af
Uitkomst	1. Als verwacht
Pass/Fail	Pass

Test	TC02.2
Use case	UC2 Stop
Test scenario	Check het proces bij het stoppen van de applicatie via X knop
Pre-conditie	Het programma is gestart
Testproces	1. Druk op de X knop rechtsboven
Verwachte Uitkomst	1. Applicatie sluit af
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 3 – Search for Vessel

Test	TC03.1
Use case	UC8 Search for Vessel
Test scenario	Check zoekresultaat met valide zoekopdracht
Pre-conditie	Programma is gestart
Testproces	1. Klik op search bar 2. Type in Zoekopdracht
Test Data	Zoekopdracht = 'alko'
Verwachte Uitkomst	1. De search bar wordt leeg gemaakt 2. De schepen lijst verandert bij het intikken van elke letter 3. De schepen lijst geeft het schip 'LAMNALCO MALKOHA' weer
Uitkomst	1. Als verwacht
Pass/Fail	Pass

Test	TC03.2
Use case	UC8 Search for Vessel
Test scenario	Check zoekresultaat met invalide zoekopdracht
Pre-conditie	Programma is gestart
Testproces	1. Klik op search bar 2. Type in Zoekopdracht
Test data	Zoekopdracht = 'ppppp'
Verwachte Uitkomst	1. De search bar wordt leeg gemaakt 2. De schepen lijst verandert bij het intikken van elke letter 3. De schepen lijst geeft geen schip weer
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 4 – Select Vessel

Test	TC04
Use case	UC9 Select Vessel
Test scenario	Check resultaat bij het selecteren van een schip
Pre-conditie	Programma is gestart
Testproces	1. Klik op het Geselecteerhet schip
Test Data	Geselecteerhet schip = 'LAMNALCO MALKOHA'
Verwachte Uitkomst	1. Applicatie laat de naam van het Geselecteerhet schip zien 2. de knop 'Select source' wordt actief
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 5 – Select Document

Test	TC05
Use case	UC10 Select Document
Test scenario	Check resultaat na het drukken op knop 'Select source'
Pre-conditie	Er is een schip geselecteerd
Testproces	1. Druk op 'Select source' knop 2. Druk op 'Geselecteerd Document' 3. Druk op 'Openen' knop
Test Data	Geselecteerd Document = 'CYG00246_13-05-2019_PSRPT_2019-05-13_14.23.00.xml'
Verwachte Uitkomst	1. Applicatie opent een File Dialog window 2. Applicatie laat de naam van 'Geselecteerde Document' zien 3. 'Transform' knop wordt geactiveerd
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 6 – Transform

Test	TC06
Use case	UC3 Transform
Test scenario	Check resultaat na het drukken op knop 'Transform'
Pre-conditie	Er is een document geselecteerd
Testproces	1. Druk op 'Transform' knop 2. Wacht op het pop-up scherm, 'Save An XML File' 3. Ga naar 'Save Locatie' 4. Vul naam 'Save Naam' in 5. Klik op opslaan
Test Data	Save Locatie = 'Desktop' Save Naam = 'testingFile'
Verwachte Uitkomst	1. Applicatie laat Folder Dialog zien 2. File 'Save Naam'.xml staat op de Desktop 3. Er wordt een email scherm geopend met 'Save Naam'.xml als attachment
Uitkomst	1. Als verwacht
Pass/Fail	Pass

TestCase 7 – Send Mail

Test	TC07
Use case	UC8 Send Mail
Test scenario	Check resultaat na de eerste email pop-up
Pre-conditie	Transformatie proces is succesvol afgerond en er is een email pop up scherm getoond
Testproces	1. Druk op 'Send' knop 2. Wacht op volgende email pop up 3. Druk op 'Send' knop
Verwachte Uitkomst	1. Outlook is geopend 2. Er staat twee extra emails in de Verzonden Berichten map
Uitkomst	1. Als verwacht
Pass/Fail	Pass