



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Datum: 04 juni 2010

Opdrachtgever

KNMI
Wilhelminalaan 10
3732 GK De Bilt
030-2206911

Bedrijfsmentoren

Dhr. R.G. Bosboom
Rene.Bosboom@knmi.nl
Dhr. H. van Soest
Henny.van.Soest@knmi.nl

Versie: 1.0

Afstudeerder

Dhr. B.A.S. Wille (60114)
Goudenregenplantsoen 35
2404 EH Alphen aan den Rijn
sybren@willeweb.nl
06-24575129

Examinatoren

Dhr. A.G.P. Pronk
A.G.P.Pronk@hhs.nl
Dhr. J.J. Visser
J.J.Visser@hhs.nl



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Referaat

Sybren Wille, Afstudeeropdracht, Ontwikkeling van een internet applicatie ter behoeve van het weer op zee, KNMI, De Bilt, 04 juni 2010

Dit document is geschreven in het kader van het project "Ontwikkeling van een internet applicatie ter behoeve van het weer op zee". Dit project is in de vorm van een afstudeeropdracht, onderdeel van de Haagse Hogeschool opleiding Technische Informatica. Het afstudeerproject is uitgevoerd bij het KNMI te de Bilt. Gedurende de periode 8 februari tot en met 04 juni 2010.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Voorwoord

Als student Technische Informatica aan de Haagse Hogeschool afdeling Delft, heb ik de afgelopen vier en een halve maand de gelegenheid gehad stage te lopen bij het KNMI te de Bilt.

Deze scriptie is geschreven naar aanleiding van het project waaraan ik de afgelopen periode heb gewerkt.

Graag wil ik hierbij mijn stagebegeleider Rene Bosboom alsmede Henny van Soest bedanken voor de begeleiding tijdens deze periode evenals de gezellige werksfeer.

De Bilt, 04 juni 2010
Sybren Wille



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Samenvatting

Voor het afstuderen is vier en halve maand stage gelopen bij het KNMI. Deze stage had als doel een internetapplicatie te maken van een al gedeeltelijk bestaande applicatie. Het KNMI verzamelt weersinformatie vanaf zee afkomstig van scheepsvaart. De schepen die hieraan mee doen zijn opgeslagen in een systeem evenals de apparatuur en andere zaken die hiermee te maken hebben. Ook zijn er applicaties die de data afkomstig van de schepen controleren en die de kosten van verzenden berekenen. Al deze functies moeten omgezet worden in een internetapplicatie.

Aangezien de bestaande applicaties niet meer voldoen aan de eisen van de gebruiker, moeten de gebruikers eisen opnieuw achterhaald worden. Dit is gedaan doormiddel van een paar interviews die samen het software requirements specification document hebben opgeleverd.

Voor dit project is gekozen voor de ontwikkelmethode RUP. Dankzij deze methode was er genoeg tijd beschikbaar voor het achterhalen van de eisen evenals de overdracht van de werkzaamheden.

Doormiddel van een ontwerpdocument zijn de functionele en niet functionele eisen omgezet in een ontwerp. Bij dit ontwerp is tevens een database ontworpen voor de nieuwe applicatie. Een van de eisen was dat de applicatie ontwikkeld moest worden met PHP en MYSQL.

Doormiddel van de MoSCoW¹ methode zijn de functionele eisen ingedeeld. Hierna konden de must en should haves opgedeeld worden in vier iteraties. Deze iteraties konden achtereenvolgens ontwikkeld worden. Na elke iteratie was een evaluatie moment aanwezig met de toekomstige gebruiker. Dit moment zorgde ervoor dat het uiteindelijke product voldoet aan de eisen van de gebruiker.

Het product dat opgeleverd is, is een internetapplicatie waarin verschillende zaken zijn in te vullen te denken valt hierbij aan:

- Maatschappij
- Schip
- Instrument
- Instrument soort
- Communicatie
- Communicatie soort
- Verzendingen
- Software
- Software soort
- Bezoeken
- Waarnemers

Al deze informatie is in de applicatie in te vullen, te wijzigen evenals te bekijken en te verwijderen. Hiernaast kunnen in de applicatie verschillende bestanden ingelezen en gecontroleerd worden. Zo worden de bestanden met data afkomstig van de schepen gecontroleerd op fouten. Fouten die gevonden worden kunnen in de applicatie gerepareerd worden. Tevens worden nota's in de applicatie ingelezen waardoor het KNMI de kosten van verzending kan bepalen.

¹ Must have, Should have, Could have, Would like to have



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Om de grote verscheidenheid aan informatie die ingelezen wordt op een eenvoudige manier te raadplegen beschikt de applicatie over verschillende rapporten. Deze rapporten laten onder andere de kosten van verzending zien. Maar ook andere zaken zoals welke apparatuur waar aan boord is kunnen opgevraagd worden.

De applicatie beschikt ook nog over een SQL builder. Deze builder stelt de gebruiker in staat zelf query's uit te voeren op de database.

Het product wat opgeleverd is, beschikt over alle must en should have's zoals deze gedefinieerd waren in het software requirements document.



Afstudeerscriptie

Inhoud

Referaat	3
Voorwoord	5
Samenvatting	7
Lijst van figuren	10
Lijst van tabellen	11
Code fragment	11
1. Inleiding	13
2. Organisatie	15
3. Actuele situatie	17
4. Opdrachtoomschrijving	19
4.1 Probleemstelling	19
4.2 Doelstelling	19
4.3 Opdrachtoomschrijving	19
4.4 Op te leveren producten	20
4.5 Projectgrenzen	21
4.6 Randvoorwaarden	22
5. Doel nieuwe systeem	23
6. Analyse	25
7. Inception Fase	27
8. Elaboration fase	29
8.1 Functionele eisen	29
8.2 niet functionele eisen	31
8.3 indeling eisen	33
8.4 Database ontwerp	34
8.5 Ontwerp software	41
8.5.1 Gegevens verzamelen	41
8.5.2 Ingevoerde gegevens interpreteren	47
8.6 Planning	50
8.7 Milestone 2	52
9. Construction fase handmatige invoer	53
9.1 Lay-out	53
9.2 ontwikkeling handmatig invoeren	53
9.3 Evaluatie	54
10. Construction fase inlezen bestanden	55
10.1 Montab	55
10.2 Journaaldiskette	58
10.3 DMG	60
10.3.1 RTWNDB	61
10.3.2 VOS	62
10.4 Evaluatie	62
11. Construction fase rapporten maken	63
11.1 Stratos nota bestand inlezen	63
11.2 Rapporten	64
11.3 SQL builder	64
11.4 Lijsten	66
11.5 Evaluatie	66
12. Construction fase overige zaken	67
12.1 Statistieken	67
13. Transition fase	69
13.1 Data oude systeem naar nieuwe systeem	69



Afstudeerscriptie

13.2 Testen	69
13.3 Overdracht naar programmeur	69
13.4 Overdracht naar gebruikers	70
14. Algemene evaluatie	71
14.1 Opdracht	71
14.1.1 Inception fase	71
14.1.2 Elaboration fase	71
14.1.3 Construction fase 1	71
14.1.4 Construction fase 2	72
14.1.5 Construction fase 3	72
14.1.6 Construction fase 4	72
14.1.7 Transition fase	72
14.2 Proces evaluatie	72
14.3 competenties	73
15. Conclusie	75
15.1 Functionele eisen	75
15.2 niet functionele eisen	77
15.3 Totale conclusie	78
16. Aanbevelingen	79
17. Begrippenlijst	81
18. Bronnenlijst	83
19. Externe bijlage	85
Bijlage 1 – Database ontwerp	87
Bijlage 2 – Use case bestand verwerken	89
Bijlage 3 – Use case gegevens	91
Bijlage 4 – Use case opvragen	93
Bijlage 5 – Klassendiagram handmatig invoer 1	95
Bijlage 6 – Klassendiagram handmatig invoer 2	97
Bijlage 7 – Rapporten	99
Bijlage 8 – Lijsten	101
Bijlage 9 – Invullen	103

Lijst van figuren

C.H.D. Buys Ballot (1817-1890)	15
Actuele situatie	18
Flow diagram nieuwe situatie	23
RUP fases	26
Database versie 1	35
Database versie 2	36
Database versie 3	37
Database versie 4	38
Database versie 5	39
Database versie 6	40
Sequence diagram gegevens inlezen	42
Klassendiagram inlezen bestand	43
Sequence diagram invoeren gegevens	45
Sequence diagram gegevens bekijken	46
Klassendiagram invullen gegevens	47



Afstudeerscriptie

Klassendiagram SQLBuilder	50
Toestandsdiagram Inlezen journaaldiskette	59
situatie DMG.....	61
Database.....	87
Use case bestand verwerken	89
Use case gegevens.....	91
Use case opvragen.....	93
Klassendiagram gegevens invullen.....	95
Klassendiagram Handmatige invoer 2.....	97

Lijst van tabellen

Planning korte termijn	27
Planning lange termijn	27
Planning lange termijn	51

Code fragment

Query company	48
Query company ingevuld	48
MONTAB bestand.....	56
Achterhalen van Pressure waardes.....	56
Verwerking MONTAB gegevens.....	58
Nota naar database	63
Query omzetten naar table	65
Query laden	65
Query Lijsten	66
Query statistieken	67



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

1. Inleiding

Tijdens de afstudeerperiode ben ik werkzaam geweest bij het KNMI. Het KNMI wat onder andere de weersverwachting doet heeft ook een systeem wat de weersverwachting op zee moet verbeteren. Dit systeem werkt doormiddel van de data die binnenkomt van schepen aanwezig op zee. Hierbij is ook een systeem wat bijhoudt welke schepen meedoen en welke apparatuur deze aanboord hebben. Tevens worden in dit systeem bestanden die vanaf de schepen komen gecontroleerd evenals de kosten die de schepen voor het verzenden maken berekend.

Het doel van de afstudeeropdracht is dit systeem om te zetten naar een web based variant. Het voordeel hiervan is dat de applicatie dan ook onderweg bij scheepsbezoeken kan worden geraadpleegd. Omdat het bestaande systeem niet meer aan de eisen van de gebruikers voldoet moeten de eisen opnieuw achterhaald worden. Vervolgens moeten de eisen opgezet worden naar een ontwerp, en moet dit ontwerp ontwikkeld worden.

Het verslag is als volgt opgebouwd, in het hoofdstuk organisatie worden de activiteiten van de organisatie waarbij de afstudeeropdracht wordt uitgevoerd behandeld. Vervolgens volgt een hoofdstuk met de actuele situatie, gevolgd door een hoofdstuk met de opdrachtomschrijving. Hierna wordt het doel van het nieuwe systeem nogmaals duidelijk kenbaar gemaakt. Hierna volgt het hoofdstuk analyse waarin de keuze voor een ontwikkelmethode wordt behandeld. In de hoofdstukken hierna worden de verschillende fases van de gekozen ontwikkelmethode in verschillende hoofdstukken behandeld. Daarna volgen achtereenvolgens de hoofdstukken evaluatie, conclusie, aanbevelingen, begrippenlijst en een bronnenlijst. Het verslag wordt afgesloten met een hoofdstuk waar de externe bijlage te vinden zijn, gevolgd door de normale bijlage.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

2. Organisatie

Het hoofdstuk "organisatie" beschrijft het bedrijf waarbij de afstudeerder zijn opdracht heeft uitgevoerd.

Het KNMI werd bij Koninklijk Besluit op 31 januari 1854 opgericht. Prof. C.H.D. Buys Ballot (1817-1890) was de eerste hoofddirecteur.

Het KNMI is in Nederland vooral bekend van de weersverwachtingen en waarschuwingen, maar als nationaal data- en kenniscentrum voor weer, klimaat en seismologie doet het veel meer.

Het KNMI verstrekt weerinformatie ten behoeve van veiligheid, economie en duurzaam milieu aan het algemeen publiek, de overheid, de luchtvaart en de scheepvaart. Voor langetermijn-ontwikkelingen verricht het KNMI onderzoek naar de veranderingen in het klimaat. Het beschikbaar stellen van bij het KNMI aanwezige kennis, data en informatie is een kernactiviteit



**Figuur 1 C.H.D. Buys Ballot
(1817-1890)**

Het KNMI is vierentwintig uur per dag actief. De combinatie van infrastructuur, techniek, wetenschap en dienstverlening is uniek op het gebied van weer, klimaat en seismologie. Wereldwijd werkt het KNMI intensief samen met andere instituten en onderzoeksinstituten. Bijvoorbeeld met het Ecmwf¹ in Engeland. Het KNMI vertegenwoordigt Nederland in tal van andere internationale organisaties, zoals het WMO², het IPCC³ en het Europese meteorologische satellietnetwerk Eumetsat.

Vanuit de centrale weerkamer in De Bilt verzorgt vierentwintig uur per dag het KNMI de weersverwachtingen en waarschuwingen voor het algemene publiek, de luchtvaart en de maritieme sector. Speciaal voor de nationale luchthaven Schiphol en Rotterdam Airport levert het KNMI meteorologische data en dienstverlening op locatie.

Het KNMI verricht zowel strategisch, als toegepast meteorologisch onderzoek. Het strategisch onderzoek is gericht op het begrijpen van de belangrijkste processen die de ontwikkeling in de toestand van de atmosfeer, de oceaan en de vaste aarde bepalen. Het toegepast onderzoek is vooral gericht op het beantwoorden van vragen uit de maatschappij, en op het ontwikkelen van waarnemings- en modelsystemen, die nodig zijn voor het produceren van de benodigde data.

¹ Europees Centrum voor Weersverwachtingen op de Middellange Termijn

² Wereld Meteorologische Organisatie

³ Intergovernmental Panel on Climate Change



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

3. Actuele situatie

Meteorologische waarnemingen van schepen spelen een vitale rol in de identificatie van weersystemen en het detecteren van klimatologische veranderingen op zee. Bovendien spelen zij een cruciale rol in de weersverwachting en het tijdig waarschuwen voor "gevaarlijk" weer.

De Nederlandse VOS¹ vloot bestaat sinds 1854 toen vissers, op initiatief van het KNMI, begonnen met het maken van waarnemingen op zee. Tegenwoordig bestaat de Nederlandse VOS vloot uit ca. 200, merendeels onder Nederlands vlag varende, koopvaardij schepen en 35 schepen van de Koninklijke Marine. Zij zijn onderdeel van de ca. 5000 schepen tellende internationale VOS vloot die op alle wereldzeeën waarnemingen maakt. Zij zijn daartoe uitgerust met meteorologische instrumenten en in veel gevallen met een door het KNMI verstrekte laptop.

Op het KNMI is ten behoeve van de deelnemende schepen speciale software ontwikkeld:

- TurboWin helpt de waarnemer met het genereren van de juiste code voor de OBS².
- MeteoClassify is een programma dat de waarnemer inzicht geeft in wolken coderingen.
- Ocean Wave Atlas geeft de zeevarenden informatie over de Klimatologie van de oceanen.

Op het KNMI zelf wordt doormiddel van verschillende software de informatie die binnenkomt verwerkt. Deze programma's regelen de volgende zaken:

- Controleren van de binnengekomen bestanden.
- Bijhouden van het aantal waarnemingen per schip.
- Bijhouden van de scheepsbezoeken.
- Administratie van de apparatuur aan boord van een schip.

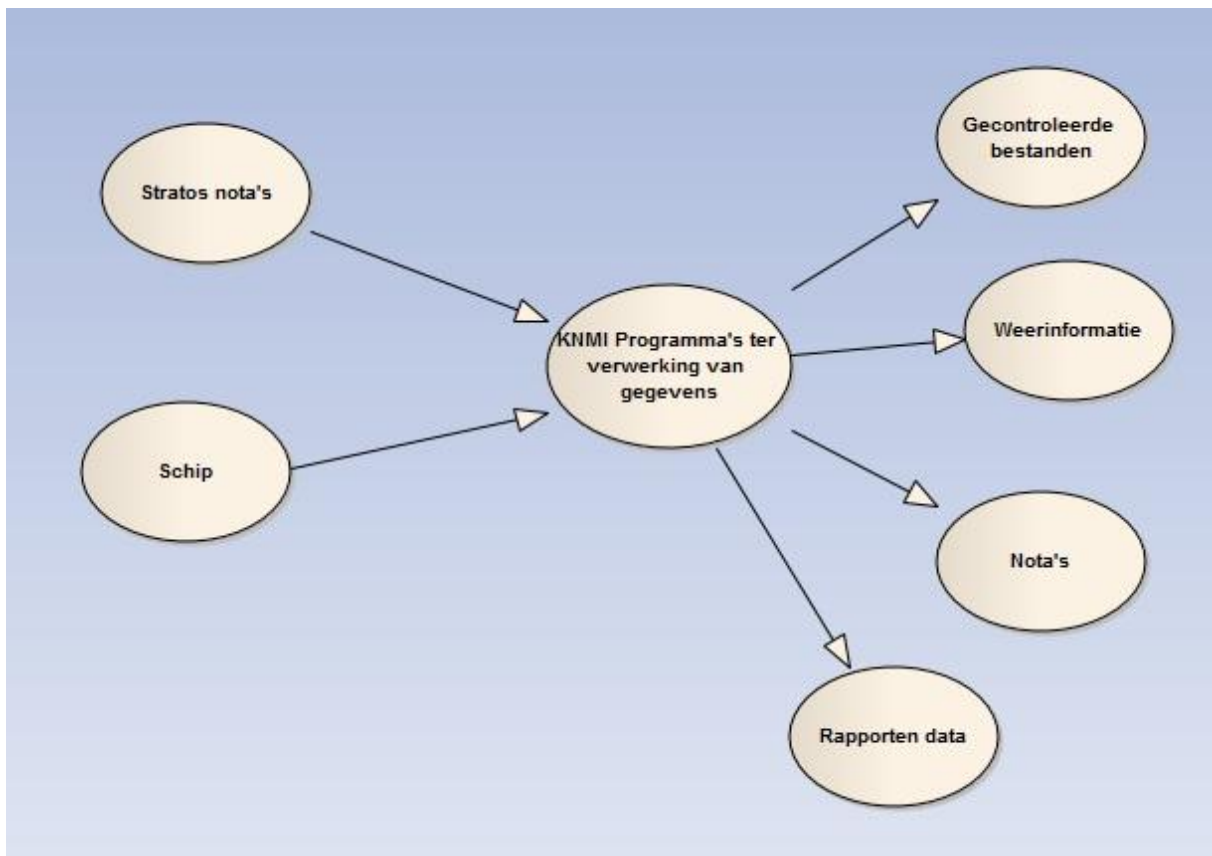
De Port Meteorological Officer (PMO) is de functionaris die invulling geeft aan de taken van PMO-Office. Zijn verantwoordelijkheid is de continuïteit en kwaliteit waarborgen van de scheepswaarnemingen. Hiertoe bezoekt hij regelmatig de participerende VOS (internationaal) wanneer deze de Nederlandse en Belgische havens aandoen. Hierom worden ook bezoeken aan schepen die niet tot de Nederlandse VOS behoren in het systeem opgenomen. **Figuur 2** geeft een schematische weergave van de huidige situatie.

¹ Voluntary Observation SHIP program

² Scheepsofficieren maken op vaste tijden een waarneming ook wel OBS genoemd.



Afstudeerscriptie



Figuur 2 Actuele situatie

Aan de hand van de waarnemingen die deze schepen doen kan het weermodel voor op zee aangepast worden. Aangezien er geen meetapparatuur op zee staat is het gebruik van schepen hiervoor een handige oplossing die ervoor zorgt dat de weermodellen verbeterd kunnen worden.

De instrumenten en sensoren die zich op de schepen bevinden zijn afkomstig van en in onderhoud bij het KNMI.

Van de instrumenten en de scheepsbezoeken wordt administratie bijgehouden om te voldoen aan de kwaliteitseisen van het Kwaliteit Management Systeem (KMS) van het KNMI.



Afstudeerscriptie

4. Opdrachtomschrijving

In dit hoofdstuk volgt een duidelijke beschrijving van de opdracht zodat hierover een goed beeld gevormd kan worden. Hiermee wordt meteen duidelijk welke taken uitgevoerd moeten worden om de opdracht succesvol af te ronden.

Het probleem dat opgelost moet worden wordt gedefinieerd (**Hoofdstuk 4.1**), de doelstelling wordt bepaald (**Hoofdstuk 4.2**), de opdracht wordt duidelijk omschreven en het eindproduct wordt gedefinieerd (**Hoofdstuk 4.3**). Daarnaast worden de op te leveren producten vastgesteld (**Hoofdstuk 4.4**) en worden de projectgrenzen (**Hoofdstuk 4.5**) en randvoorwaarden (**Hoofdstuk 4.6**) bepaald.

4.1 Probleemstelling

De afstudeeropdracht betreft het ontwikkelen van een systeem dat bedoeld is voor het beheer van meteorologische meetinstrumenten en sensoren op de schepen en het monitoren van de data.

Het vervult daarmee een noodzakelijke rol voor de Port Meteorological Officer. Momenteel is er geen goed systeem beschikbaar waardoor een nieuw systeem ontwikkeld moet worden. Bij de ontwikkeling van dit systeem wordt zoveel mogelijk rekening gehouden met de toekomstige ontwikkelingen zoals het aansluiten op internationale VOS systemen.

4.2 Doelstelling

Het doel van de opdracht is het ontwikkelen van een goed systeem voor de opslag, verwerking en het beschikbaar stellen van de gegevens (metadata) van de schepen die behoren tot de VOS vloot. Deze schepen doen op vrijwillige basis mee aan waarnemingen op zee ten behoeve van de verwachtingen en weermodellen. Indien er onverwacht nog tijd beschikbaar is kan tevens nog een onderzoek plaatsvinden naar de mogelijkheden om het systeem internationaal in te zetten. Hierbij zou de database gekoppeld moeten worden aan de database van andere landen.

4.3 Opdrachtomschrijving

De opdracht omvat het ontwikkelen van een nieuw systeem ter opslag, verwerking en beschikbaarheid van de gegevens van de VOS vloot. Hierbij wordt de meteorologisch relevante metadata opgeslagen. Deze data bevat onder andere:

- IMO¹ nummer
- Naam
- Maatschappij
- Telefoonnummer
- Aanwezige meetinstrumenten en sensoren
- Aanwezige software
- Plaats van controle
- Bezocht door

¹ International Maritime Organization



Afstudeerscriptie

- Drukcorrectie
- Scheepsbezoeken

Dit is echter maar een klein deel van de opdracht. Naast de hierboven genoemde data dient er namelijk ook een journaaladministratie bijgehouden te worden waarin onder andere staat welke waarnemers gemeten hebben en het aantal van deze waarnemingen.

Ook gaat de beveiliging van het systeem een rol spelen. Het is namelijk mogelijk om op verschillende websites op te zoeken waar een schip zich op dit moment bevindt. Dit kan gedaan worden door het invullen van de roepletters of naam van het schip. Sommige schepen willen dit echter niet vanwege mogelijke piraterij. Er moet dus een oplossing gezocht worden waaraan het KNMI wel kan zien welk schip het betreft maar een buitenstaander niet.

Er moeten interviews gaan plaatsvinden om de eisen van de toekomstige gebruikers vast te kunnen leggen. Hierdoor wordt mede bepaald welke onderdelen gerealiseerd gaan worden. Het is daarom nog mogelijk dat er andere elementen in de opdracht een rol gaan spelen dan vooraf gepland was.

Indien er nog tijd beschikbaar is kan er nog een onderzoek uitgevoerd worden. Dit onderzoek concentreert zich dan op de haalbaarheid van de applicatie in internationaal verband. Engeland, Duitsland en Frankrijk zijn tevens geïnteresseerd in een systeem waarin de VOS gegevens zitten. De kans bestaat dus dat deze landen een koppeling willen met het systeem.

Een onderzoek naar de haalbaarheid van een koppeling van het systeem met hun database kan dan plaatsvinden. Echter kan al bij de ontwikkeling van het systeem rekening gehouden worden met de mogelijkheid dat andere landen een koppeling willen.

Het systeem moest voor aanvang van de opdracht minimaal over de volgende functionaliteit gaan beschikken:

- Het interactief zoeken middels een grafische user interface naar informatie van schepen. Dit zoeken moet kunnen gebeuren op een specifiek schip of periode. De uitvoer die op het scherm verschijnt, moet uitgeprint of in een bestand gezet kunnen worden.
- Verschillende presentaties van de metadata maken ten behoeve van verschillende gebruikers.
- De betrouwbaarheid van de meetinstrumenten en sensoren kunnen monitoren.
- Aansluiting op de VOS metadata applicatie middels in- en exportfuncties van de data.

4.4 Op te leveren producten

Bij dit project worden verschillende producten gerealiseerd die samen het eindproduct vormen. Deze producten zijn:

- 1) Documentatie systeem
- 2) Code



Afstudeerscriptie

- 3) Software
- 4) Functionele specificaties
- 5) Scriptie
- 6) Handleiding

Ad 1) Documentatie systeem

Documentatie van de gemaakte software. Deze documentatie beschrijft de precieze werking van de software zodat deze later eenvoudig door andere programmeurs aangepast kan worden. De documentatie beschrijft de volledig ontwikkelde software.

Ad 2) Code

De ontwikkelde software wordt ook als code opgeleverd.

Ad 3) Software

De software zelf wordt ook opgeleverd. Inclusief de database die aan de software hangt.

Ad 4) Functionele specificaties

Een document wat de eisen van de toekomstige gebruikers beschrijft. Hierdoor wordt het duidelijk wat men in de software wil en waar men de software in de toekomst kan uitbreiden.

Ad 5) Scriptie

Aangezien het project ook een afstudeerstage is wordt hierover ook een scriptie gemaakt. Deze scriptie concentreert zich op wat er gedaan is in de 17 weken dat het project loopt.

Ad 6) Handleiding

Een handleiding wordt opgeleverd. In deze handleiding wordt de werking van het systeem beschreven voor de programmeur die het geheel in de toekomst moet onderhouden. Tevens wordt een handleiding opgeleverd aan de toekomstige gebruiker over de precieze werking van het systeem.

4.5 Projectgrenzen

Het project concentreren zich op het achterhalen van de functionele specificaties en het ontwikkelen van de software. Door middel van de MoSCoW¹ methode worden de functionele specificaties opgedeeld waardoor goed zichtbaar is welke zaken eerst ontwikkeld moeten worden. Dit wordt gedocumenteerd in de software requirements specification.

De tijdsduur van het project staat tevens vast:

- Start datum 8 februari 2010.
- Eind datum 4 juni 2010.
- 40 uur per week.

¹ Must have, Could have, Should have, Would like to have



Afstudeerscriptie

4.6 Randvoorwaarden

Voor het project gelden de volgende randvoorwaarden:

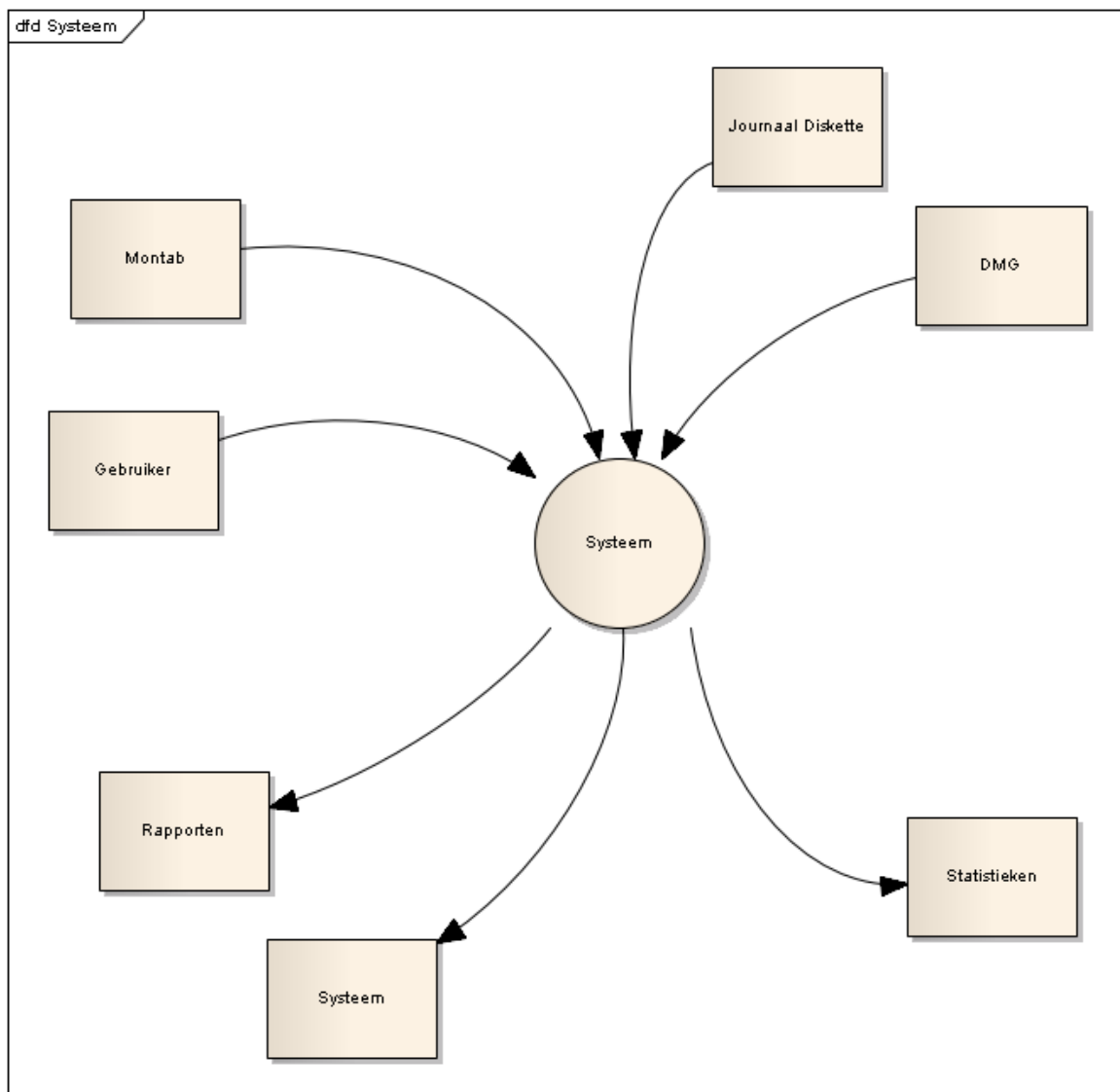
- De software moet op een zodanige manier ontwikkeld worden dat deze eenvoudig te onderhouden is.
- De software moet eenvoudig uitbreidbaar zijn.
- Er moet rekening gehouden worden met een koppeling naar een buitenlandse database.
- De applicatie en daarmee de data moet benaderbaar zijn middels een webbrowser.
- De applicatie dient gebouwd te worden met ondersteunde technieken door de afdeling INFRA-ICT.



Afstudeerscriptie

5. Doel nieuwe systeem

Zoals te lezen is in **Hoofdstuk 3** worden de gegevens die binnenkomen van de schepen verwerkt door verschillende programma's die aanwezig zijn bij het KNMI. Echter zijn deze programma's alleen vanuit het KNMI te benaderen. Men wil graag een deel van deze programma's omzetten naar een web based applicatie. Deze applicatie kan dan ook wanneer men onderweg is via de intranet website benaderd worden. **Figuur 3** geeft doormiddel van een flow diagram de nieuwe situatie weer.



Figuur 3 Flow diagram nieuwe situatie

Het systeem moet alle gegevens bevatten van de Nederlandse schepen behorende bij de VOS vloot. Daarnaast moet het mogelijk zijn verschillende data in te lezen. Hieronder valt het volgende:

- Nota's
- Waarnemingen
- Maand totalen vanuit Engeland



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

- Maand totalen KNMI
- Maand totalen VOS internationaal

Van alle gegevens die ingelezen worden moeten rapporten gemaakt kunnen worden. Deze rapporten geven vervolgens inzicht over de informatie die in het systeem aanwezig is. Hierdoor is het bijvoorbeeld mogelijk te kijken hoeveel waarnemingen een schip doet. Ook kunnen de kosten van het verzenden van deze waarnemingen naar het KNMI bekeken worden.

Naast deze administratieve gegevens die opgeslagen moeten worden dient er ook nog een journaaladministratie bijgehouden te worden. Hierin staat o.a. welke waarnemers gemeten hebben en het aantal van deze waarnemingen. Aan de hand van deze gegevens kan gekeken worden of de desbetreffende waarnemer in aanmerking komt voor een onderscheiding.



Afstudeerscriptie

6. Analyse

Aan het begin van dit project is een analyse uitgevoerd. Nadat globaal duidelijk was wat ontwikkeld ging worden is gekozen voor een ontwikkelmethode.

Ontwikkelmethode

Voor dit project kon uit verschillende ontwikkelmethoden gekozen worden:

- Gefaseerde ontwikkelmethode
- Prototype ontwikkelmethode
- Wegwerpprototype methode
- Extreme Programming
- RUP¹

Om een product op te kunnen leveren wat voldoet aan de eisen van de gebruiker is het van belang de gebruiker te betrekken bij het ontwikkelproces. Door tussentijds werkende stukken van het systeem te laten evalueren door de gebruiker is het makkelijker daadwerkelijk het product te ontwikkelen wat de gebruiker voor ogen heeft. Eis voor de ontwikkelmethode is dus wel dat er tussentijds werkende delen van het systeem aan de toekomstige gebruiker getoond kunnen worden.

Hierdoor is het logisch het project op te delen in verschillende fases. Na elke fase kan een overleg met de gebruiker plaats vinden waarin gekeken kan worden of het ontwikkelde gedeelte voldoet aan de eisen. Eventuele kleine op of aanmerkingen kunnen dan alsnog meegenomen worden in de volgende fase. Ontwikkelmethode die hiervoor geschikt zouden zijn, zijn de volgende:

- Gefaseerde ontwikkelmethode
- Wegwerpprototype methode
- RUP

Hieruit blijkt dat uit de eerder genoemde lijst met ontwikkelmethoden er twee zijn afgevallen die niet voldoen aan de eisen, namelijk: extreme programming en prototype ontwikkelmethodologie. Deze methoden zijn niet geschikt voor dit project omdat ze na een planning niet een algemeen ontwerp of algemene analyse hebben. Het is echter wel gewenst om van te voren een algemeen ontwerp te maken. Alle fases bij elkaar vormen namelijk het product. Een algemeen ontwerp zorgt ervoor dat alle elementen van de verschillende fases goed met elkaar geïntegreerd zijn. Om dit te kunnen realiseren moet dus eerst een algemeen ontwerp en analyse plaats vinden. Hierdoor zijn extreme programming en de prototype ontwikkelmethode niet geschikt voor dit project.

Van de drie ontwikkelmethoden die geschikt zouden kunnen zijn is de wegwerpprototype methode het minst geschikt van de drie. Bij deze methode worden eerst verschillende prototype ontwikkeld van de verschillende delen van het systeem. Het uiteindelijke systeem wordt alsnog in fases ontwikkeld. Dit is niet echt wat goed past bij dit project omdat het tussentijds toevoegen van code hierdoor moeilijker wordt.

De twee methode die overblijven lijken veel op elkaar. RUP maakt gebruik van UML en deelt het project op in vier fases (inception, Construction, elaboration, transition). Bij de gefaseerde ontwikkelmethode gebeurt dit in mindere mate. Aangezien de functionele

¹ Rational Unified Process



Afstudeerscriptie

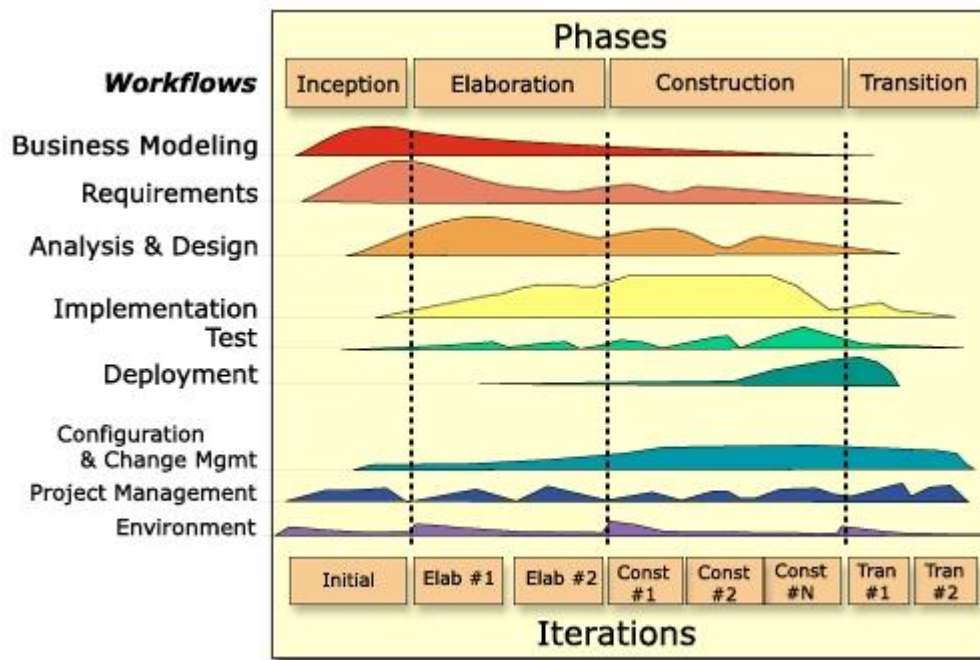
eisen achterhaald moeten worden bij de gebruiker is het noodzakelijk dat er een grote analyse aanwezig is. RUP is beter geschikt hiervoor dan de gefaseerde ontwikkelmethode.

Bij RUP is het mogelijk het idee om te zetten naar een productvisie in de inception fase. Waarna in de elaboration fase de functionele eisen gespecificeerd kunnen worden. Tevens kan een ontwerp gemaakt worden wat teruggekoppeld kan worden naar de gebruiker. Dit kan gedaan worden in meerdere iteraties. Bij de gefaseerde ontwikkelmethode is dit toch minder aanwezig waardoor de voorkeur uitgaat naar RUP. Tevens heeft RUP na de construction fase een transition fase waarin nazorg verleend kan worden en de mogelijkheid en ruimte er is om de verantwoordelijkheden over te dragen. Dit is tevens van belang bij een product wat later door andere mensen onderhouden en uitgebreid gaat worden. Wat betreft de workflows wordt de nadruk gelegd op:

- Requirements
- Analysis & Design
- Implementation
- Test

Figuur 4 geeft een schematische weergave van de RUP fases en de verschillende workflows.

De functionele eisen die de gebruikers aandragen zijn vaak op te delen in verschillende categorieën. Vaak zijn sommige eisen belangrijker dan andere. De MOSCOW¹ methode is hierom geschikt om de functionele eisen op te delen. In combinatie met RUP ontstaat dan ook gelijk meer duidelijkheid welke functionele eisen in welke iteratie ontwikkeld gaan worden.



Figuur 4 RUP fases

¹ Must have, Should have, Could have, Would like to have



Afstudeerscriptie

7. Inception Fase

Tijdens de inceptionfase komt een plan van aanpak tot stand wat de opdracht zoals in **hoofdstuk 4.3** beschrijft. Tevens worden hierin de aanpak projectinrichting en risicofactoren beschreven. Een planning voor de korte termijn werd hierbij ook gerealiseerd zie hiervoor tabel 1.

Week	Activiteit
Week 1 (08-02-10/12-02-10)	Inwerken, PvA maken
Week 2 (15-02-10/19-02-10)	PvA afronden Functioneel ontwerp maken.
Week 3 (22-02-10/26-02-10)	Functioneel ontwerp afronden.
Week 4 (01-03-10/05-03-10)	Modelleren
Week 5 (08-03-10/12-03-10)	Database ontwerpen. Indelen iteraties.

Tabel 1 Planning korte termijn

Een planning voor de lange termijn was op dat moment in de ontwikkeling nog moeilijk te realiseren. Nadat een functioneel ontwerp en de Software requirements gemaakt zijn kan het project pas in goede iteraties opgedeeld worden. Aangezien er gebruik gemaakt wordt van RUP is het wel duidelijk dat er gebruik gemaakt wordt van vier verschillende fases. Deze fases kunnen al wel in een planning gezet worden waardoor een beeld ontstaat van hoe lang de verschillende onderdelen gaan duren. Voor een globale planning zie tabel 2.

Week	Activiteit
Week 1/2 (08-02-10/19-02-10)	Inception Fase
Week 2/5 (15-02-10/12-03-10)	Elaboration Fase
Week 6/15 (15-03-10/21-05-10)	Construction Fase
Week 16/17 (24-05-10/04-06-10)	Transition Fase

Tabel 2 Planning lange termijn

Het plan van aanpak is toegevoegd als **externe bijlage 1**.

Na elke iteratie is er een overleg moment waarin de ontwikkelde iteratie geëvalueerd wordt en er eventueel kleine aanpassingen in de volgende iteraties gemaakt kunnen worden. Dit overleg zorgt ervoor dat gekeken kan worden of de ontwikkelingen nog steeds in de juiste richting plaatsvinden.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

8. Elaboration fase

Om een goed systeem te kunnen maken moeten eerst de functionele eisen achterhaald worden. Door middel van een paar interviews met de toekomstige gebruikers is geprobeerd een beter beeld te krijgen over de functionaliteit van de toekomstige software. De interviews zijn te vinden in **externe bijlage 2**.

8.1 Functionele eisen

Uit de interviews met toekomstige gebruikers zijn de volgende functionele eisen naar voren gekomen:

1) Nieuwe schepen moeten toegevoegd kunnen worden.

De gebruiker moet in staat gesteld worden nieuwe schepen aan het systeem toe te kunnen voegen. Welke informatie hierbij opgeslagen moet worden is te vinden in **externe bijlage 2**.

2) Nieuwe waarnemers moeten toegevoegd kunnen worden.

De gebruiker moet in staat zijn nieuwe waarnemers aan het systeem toe te voegen.

3) Er moet bijgehouden worden hoeveel waarnemingen een waarnemer doet.

Voor elke waarnemer wordt ook bijgehouden hoeveel waarnemingen hij gedaan heeft. Hierbij wordt bijgehouden wanneer hij welke beloning heeft gekregen en hoeveel waarnemingen daarvoor nodig waren.

4) De journaals moeten ingevuld kunnen worden.

De journaals zijn de gegevens die binnenkomen. Hiervan moet geregistreerd worden over welke tijdsduur het gaat. Het programma moet automatisch een journaalnummer toekennen aan een nieuw journaal.

5) Nieuwe maatschappijen moeten ingevoerd kunnen worden.

Vanuit het systeem moet het mogelijk zijn om een nieuwe maatschappij toe te voegen. Voor een nieuwe maatschappij wordt ook een eerste en tweede contactpersoon opgeslagen met de middelen waarmee deze bereikt kunnen worden.

6) De versie en uitgiftdatum van de software op schepen moet bijgehouden worden.

Voor elk schip moet bijgehouden worden welke software aan boord is. Hierbij is de soort software en de versie van die software van belang.

7) Verzendingen naar schepen moeten bijgehouden worden.

Materiële zaken die naar schepen verzonden worden moeten worden opgeslagen in het systeem. Het is belangrijk dat de datum van verzending en via wie het verzonden is in het systeem opgeslagen worden.

8) Verschillende soorten instrumenten moeten bijgehouden worden.

Elk instrument wat geplaatst wordt moet worden geregistreerd. Het is hierbij belangrijk dat minimaal het soort instrument en het KNMI-nr opgeslagen worden.

9) De plaatsingsdatum van instrumenten op het schip moet bijgehouden worden.

De plaatsingsdatum van instrumenten moet opgeslagen worden. Dit houdt in dat wanneer een instrument, wat zich in de instrumenten database bevindt, wordt geplaatst



Afstudeerscriptie

dit aangegeven wordt in de database. De datum van plaatsing en of het een primaire druksensor betreft is hierbij van belang.

10) Buitenlandse schepen moeten ingevoerd kunnen worden.

Ook buitenlandse schepen moeten in het systeem opgenomen kunnen worden. Hierbij moeten echter andere gegevens worden opgeslagen dan bij Nederlandse schepen. Er gelden hierbij andere minimum verplichte velden welke te vinden zijn in **Externe bijlage 2**.

11) Bezoeken van Nederlandse schepen moeten kunnen worden geregistreerd.

De bezoeken die bij Nederlandse schepen plaatsvinden moeten in het systeem worden opgeslagen. Het is hierbij belangrijk dat gegevens als de correctie, bijstelling en de data van het bezoek worden opgeslagen.

12) Bezoeken aan buitenlandse schepen moeten worden geregistreerd.

Ook bezoek aan buitenlandse schepen moet worden geregistreerd. Hier hoeft echter minder informatie bij worden opgeslagen.

13) De ingelezen data moet gecontroleerd worden op fouten.

Data die ingelezen wordt moet door het systeem worden gecontroleerd op fouten. Vooral fouten in de datum/tijd groep zijn belangrijk. Aangezien het invoeren van de data op de schepen nog mensenwerk is, is het mogelijk dat bij het invoeren van de datum en tijd een fout wordt gemaakt. In het bestand moet de datum en tijd chronologisch zijn.

14) Fouten die gevonden worden moeten door de gebruiker in het programma hersteld kunnen worden.

Fouten die het systeem vindt moeten worden aangegeven. Hierna moet de gebruiker in staat zijn om deze fouten te herstellen. Het is helaas niet mogelijk dit automatisch te doen omdat de goede waarde van veel factoren afhankelijk is en men dit toch bij menselijk handelen wil houden. De verbeterde data moet weer worden opgeslagen op het opslag medium.

15) De data moet in verschillende rapporten gezet worden en kunnen worden uitgeprint.

Van de data die in de database staat moeten verschillende rapporten gemaakt kunnen worden. Het is de bedoeling dat voor de meeste rapporten de gebruiker kan instellen over welke periode de rapporten moeten gaan. De rapporten moeten ook uitgeprint kunnen worden.

16) Verwerken MONTAB (Monthly table)

Het aantal scheepswaarnemingen die in Engeland ingewonnen zijn worden maandelijks in een email gezet en naar de PMO's gestuurd. Dit worden de MONTAB gegevens genoemd en deze gegevens moeten verwerkt worden. De gegevens moeten opgeslagen worden in de database.

17) Verwerken Journaal diskette

De journaaldiskette moet ingelezen en verwerkt worden. Echter moet dit op een zodanige manier gedaan worden dat het programma naast het inlezen van een diskette ook overweg kan met bijvoorbeeld een USB-stick of het lezen van een netwerk schijf. De data die ingelezen wordt moet worden gecontroleerd op fouten. De fouten die gevonden worden moeten worden aangegeven in het programma waarna de gebruiker deze kan wijzigen en opslaan.



Afstudeerscriptie

18) Verwerken Nota's

De nota's dienen ook nog verwerkt te worden. Deze worden ingelezen via een .txt bestand en moeten daarna in de database geplaatst worden. Dit moet grotendeels automatisch gebeuren. Van deze nota's moeten ook rapporten gemaakt worden.

19) Rapporten over schepen die langer dan een bepaalde datum geleden bezocht zijn moeten uitgedraaid kunnen worden.

Er moeten rapporten gemaakt worden van de schepen die langer geleden dan een door de gebruiker ingevoerde datum bezocht zijn.

20) De stratos gegevens moeten ingelezen worden waarna er nota's van gemaakt moeten worden.

Via stratos komen de gegevens binnen over de kosten die zijn gemaakt met het verzenden van de data via de satelliet. Deze gegevens komen ook binnen voor de Duitse schepen aangezien die geen eigen station hebben. Hierdoor moet ook voor de Duitsers berekend worden wat de kosten zijn zodat een rekening voor hun opgesteld kan worden.

21) DMG bestanden moeten verwerkt kunnen worden.

De DMG bestanden die op een netwerkschijf geplaatst worden moeten in het systeem ingevoerd kunnen worden.

22) Koppeling met buitenlands systeem.

Men ziet de mogelijkheid om het systeem te koppelen aan buitenlandse databases. Hierdoor kan het systeem internationaal ingezet worden. Hoe dit precies moet gaan werken is nog onduidelijk waardoor voordat dit onderdeel geïmplementeerd kan worden eerst een onderzoek hiernaar gedaan moet worden.

8.2 niet functionele eisen

Naast de functionele eisen zijn ook de niet functionele naar aanleiding van deze interviews vastgelegd. Deze eisen zien er als volgt uit:

1) Het systeem moet stabiel zijn.

Het systeem moet een fout kunnen verwerken en niet crashen wanneer bijvoorbeeld een corrupt bestand wordt ingelezen. Hiervoor moet goede foutafhandeling verwezenlijkt worden.

2) Foutafhandeling

Het is de bedoeling dat wanneer het systeem niet de gewenste informatie krijgt hierover een foutmelding geeft. Het systeem mag hierbij niet blijven hangen. De applicatie geeft hierbij "warning messages" en blijft wel draaien. Indien het systeem in een situatie komt waarbij herstart moet worden, wordt een "Fatal error message" gegeven en het programma afgesloten.

3) Het systeem is web based.

Het systeem moet via een browser draaien. Het voordeel hiervan is dat het systeem platform onafhankelijk is. Tevens kan, indien men dit wenst, het systeem vanuit huis benaderen via de browser.



Afstudeerscriptie

4) Het systeem moet uiteindelijk op een LAMP omgeving gaan draaien.

Het is de bedoeling dat het systeem uiteindelijk op een LAMP¹ omgeving komt te draaien. Hier moet tijdens de ontwikkeling van het systeem rekening mee gehouden worden.

5) Buitenlandse en Nederlandse schepen moeten gescheiden blijven.

Schepen uit het buitenland en schepen uit de Nederlandse vloot moeten gescheiden blijven. Deze schepen moeten dus een aparte tabel hebben en niet met elkaar verweven zijn.

6) De software moet zelfstandig taken uitvoeren.

De software moet in staat zijn zelfstandig taken uit te voeren zoals het controleren op fouten. Hier hoeft de gebruiker niet eerst toestemming voor te geven.

7) De gegevens uit de bestaande database moeten overgezet worden.

De gegevens die nu in een database staan moeten beschikbaar worden in de nieuwe database. Hier moet dus rekening mee gehouden worden bij het ontwerp van de nieuwe database.

8) Het systeem moet gebruik maken van PHP en MySQL

Doordat het systeem op een LAMP server gaat draaien wordt het product ontwikkeld in PHP. De database wordt hierom ontwikkeld met MySQL.

9) Het programma is Engelstalig

De taal waarin naar de gebruiker gecommuniceerd wordt is Engels. De code wordt ook geprogrammeerd door middel van Engelse naamgeving en het toevoegen van commentaar in het Engels.

10) Het systeem moet wereldwijd beschikbaar zijn.

Het systeem moet in de toekomst wereldwijd toegankelijk zijn. Hierdoor moet het afgeschermd worden met een gebruikersnaam en wachtwoord combinatie. In eerste instantie is dit onderdeel minder urgent aangezien het programma voorlopig intern gaat draaien.

11) Performance

Het programma moet binnen enkele seconden (1- 5 seconden) kunnen reageren op de actie van de gebruiker. Het is echter wel toegestaan dat de gebruiker een paar minuten moet wachten bij zware acties. Denk hierbij aan het inlezen en verwerken van een diskette. Afhankelijk van de grootte van het bestand neemt het inlezen een tijdsduur in beslag die boven de 30 seconden ligt en soms zelfs enkele minuten kan duren. Voor de gebruiker moet in elk geval duidelijk zijn dat het systeem bezig is.

12) Systeem grootte

De database van het systeem wordt redelijk groot (denk hierbij aan meerdere gigabytes) worden aangezien gegevens eeuwig bewaard blijven. Het is dus belangrijk dat er nauwkeurig gezocht kan worden waardoor de performance goed blijft.

8.3 indeling eisen

Door middel van de MOSCOW methode is het nu mogelijk de functionele eisen op te delen. Hierbij worden aan de belangrijkste eisen en aan de eisen die de basis van een

¹ Linux, Apache, PHP, MySQL



Afstudeerscriptie

goed functionerend systeem vormen een hoge prioriteit gegeven. De indeling ziet er als volgt uit:

Must have

- Nieuwe schepen toevoegen.
- Nieuwe waarnemers toevoegen.
- Maatschappij toevoegen.
- Buitenlandse schepen toevoegen.
- Nederlandse bezoeken toevoegen.
- Buitenlandse bezoeken toevoegen.
- Journaals invoeren.
- Verzendingen naar schepen bijhouden.
- Soorten instrumenten bijhouden.
- Journaal diskette verwerken.
- Communicatiemiddelen en soorten bijhouden per schip.
- MONTAB gegevens verwerken.
- Nota gegevens verwerken.
- Gevonden fouten moeten door gebruiker kunnen worden hersteld.
- DMG bestand verwerken.
- VOS (AOW) bestand verwerken.
- Stratos Nota's verwerken.

Deze eisen zijn dus in elk geval in het programma aanwezig. Deze eisen zijn in de Must have terecht gekomen omdat veel van deze eisen het invullen van basisinformatie bevat. Deze informatie vormt de basis van het systeem. Zonder deze informatie heeft het systeem geen nut waardoor deze gegevens aanwezig moeten zijn. Deze basisinformatie bevat niet alleen het invullen van formulieren maar ook het inlezen van sommige belangrijke bestanden.

Should have

- Plaatsingsdatum instrumenten bijhouden.
- Versie en uitgifte datum op schepen bijhouden.
- Aantal waarnemingen bijhouden.
- Rapporten over laatste bezoekdatum moeten uitgedraaid kunnen worden.
- Rapporten over bezoeken en bezochte schepen over opgegeven periode.
- Data moet in rapporten gezet worden.
- Ingelezen data moet door het systeem gecontroleerd worden op fouten.

Deze functionele eisen worden geïmplementeerd nadat de *must have's* geïmplementeerd zijn. Deze eisen staan in de should have omdat ze in mindere mate belangrijk zijn zoals het bijhouden van plaatsing- en uitgifte datum. Ook staan sommige van deze eisen in de should have omdat ze de data uit de must have beter verwerken of hiervan een presentatie maken. Dit is uiteraard wel belangrijk voor het systeem, maar minder belangrijk dan de realisatie van de invoer van de gegevens. Immers als de gegevens niet in het systeem staan kan er ook geen rapport van gemaakt worden. Hierom is er gekozen om deze elementen bij de should have's te zetten.

Could have

De volgende functionele eisen zijn ingedeeld bij de could have's:

- Koppeling met buitenlands systeem.

Indien er nog tijd beschikbaar is na het implementeren van de *should have's* wordt aandacht aan deze eisen worden besteed. Deze eis is in de Could have terecht gekomen



Afstudeerscriptie

omdat bij aanvang al duidelijk was dat aan deze eis alleen aandacht besteed wordt indien er nog voldoende tijd beschikbaar is.

Would like to have

Er zijn geen functionele eisen in de Would like to have's.

De functionele eisen, niet functionele eisen, indeling en een algemene indeling van de rapporten vormen samen het software requirements document. In dit document worden de functionele eisen ook doormiddel van use-cases diagrammen gevisualiseerd. Deze use cases diagrammen zijn te vinden bij **bijlage 2,3,4**. Dit document is teruggekoppeld aan de toekomstige gebruiker. Deze hebben het document geëvalueerd en waar nodig commentaar gegeven. Dit commentaar is vervolgens na overleg verwerkt in het document. Hierdoor is een document ontstaan wat de functionele en niet functionele eisen van de toekomstige gebruikers beschrijft zoals deze niet alleen door de ontwikkelaar maar ook door de opdrachtgever/gebruiker gezien worden.

8.4 Database ontwerp

Een van de belangrijkste onderdelen van het systeem is de database. Hierin worden alle gegevens opgeslagen. Het is daarom ook van essentieel belang dat hiervan zo snel mogelijk een ontwerp klaar is. Er is geprobeerd om via Meteo France een ontwerp van hun database te krijgen. Hierdoor zou het makkelijker worden om later de zaken internationaal te regelen. Helaas mocht alleen de ontwikkelde applicatie ingezien worden en was er geen mogelijkheid tot inzage van het database ontwerp. Hierdoor is gekozen om het ontwerp volledig zelf te ontwikkelen voor de Nederlandse VOS vloot. Landen kunnen deze database dan later zelf omzetten naar hun eigen vloot.

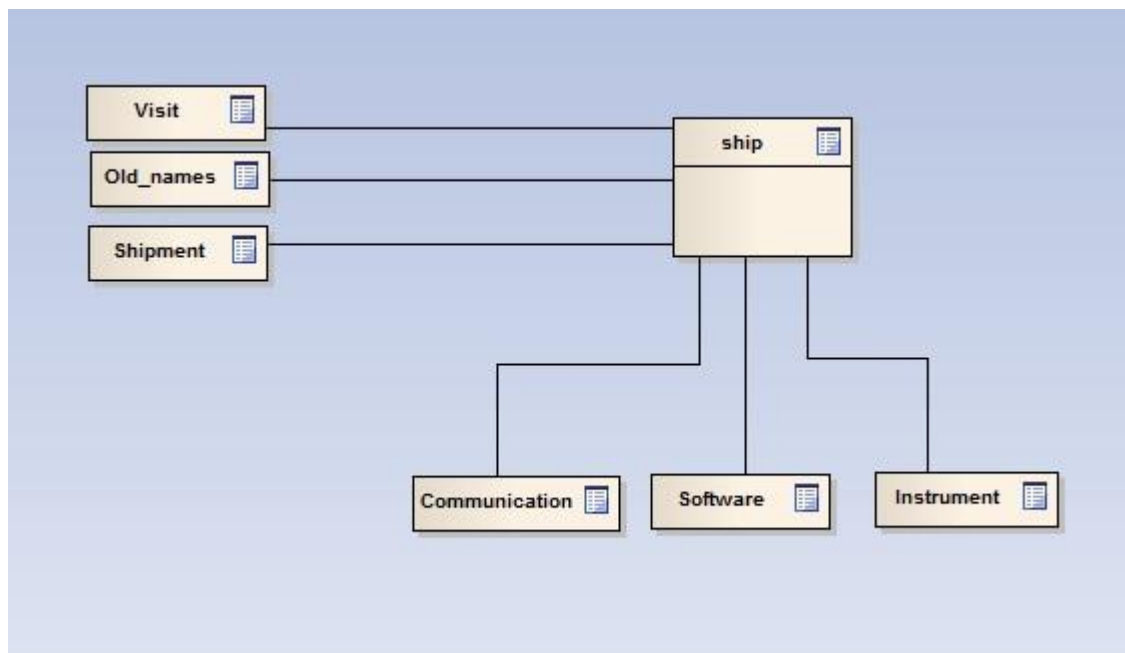
Het schip zelf vormt de belangrijkste schakel in het systeem. Alles kan tot op zekere mate aan het schip zelf gekoppeld worden. Dit komt doordat de waarnemingen op de schepen gedaan worden. Alle informatie wordt dus op het schip verzameld en is daarom aan het schip te linken. Er is daarom ook gekozen om het schip als middelpunt te nemen van de database. Op de schepen zijn verschillende zaken aanwezig:

- Software
- Instrumenten
- Communicatie middelen
- Waarnemers

Om de verschillende elementen aan het schip te linken moet er een uniek element aan elk schip zijn dat als primaire sleutel kan dienen. Navraag heeft opgeleverd dat elk schip een IMO nummer heeft. Dit is een uniek nummer en kan als primaire sleutel dienen. Door het IMO nummer ook bij de aanwezige zaken te zetten kunnen deze gekoppeld worden aan het schip. Hierdoor ontstaat de database zoals in **figuur 5**.



Afstudeerscriptie

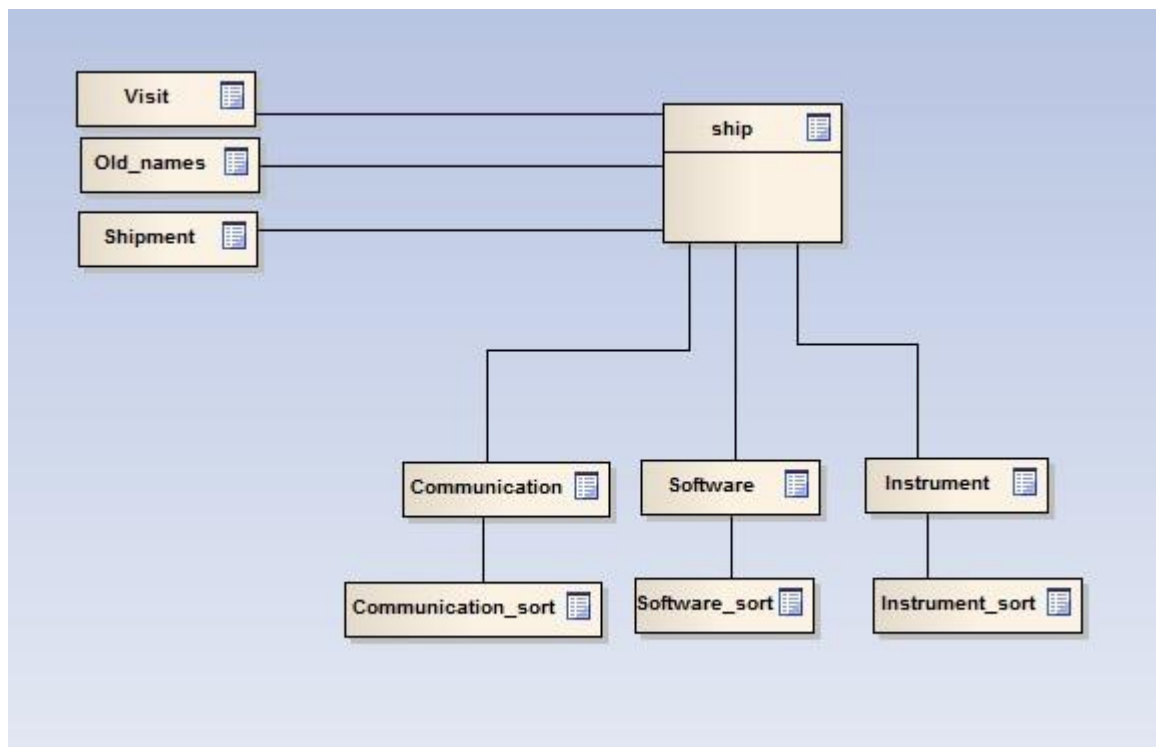


Figuur 5 Database versie 1

Aangezien op meerde schepen dezelfde soort software, instrumenten en communicatie middelen aanwezig zijn is het handig om de soorten hiervan in een aparte tabel op te slaan. Door een ID aan een soort toe te kennen en dit te verbinden met wat aan het schip zou moeten zitten kan makkelijk een soort gekozen worden. Veranderd onverwacht de naam van bijvoorbeeld een applicatie dan hoeft dit maar op één plaats gewijzigd te worden waarna bij alle schepen de juiste informatie aanwezig is. Zaken die uniek voor elk schip zijn zoals, installatie datum, identifier en calibratie waarden hoeven dan alleen nog maar per schip ingevuld te worden. Deze constructie levert de in **Figuur 6** getoonde database op.



Afstudeerscriptie

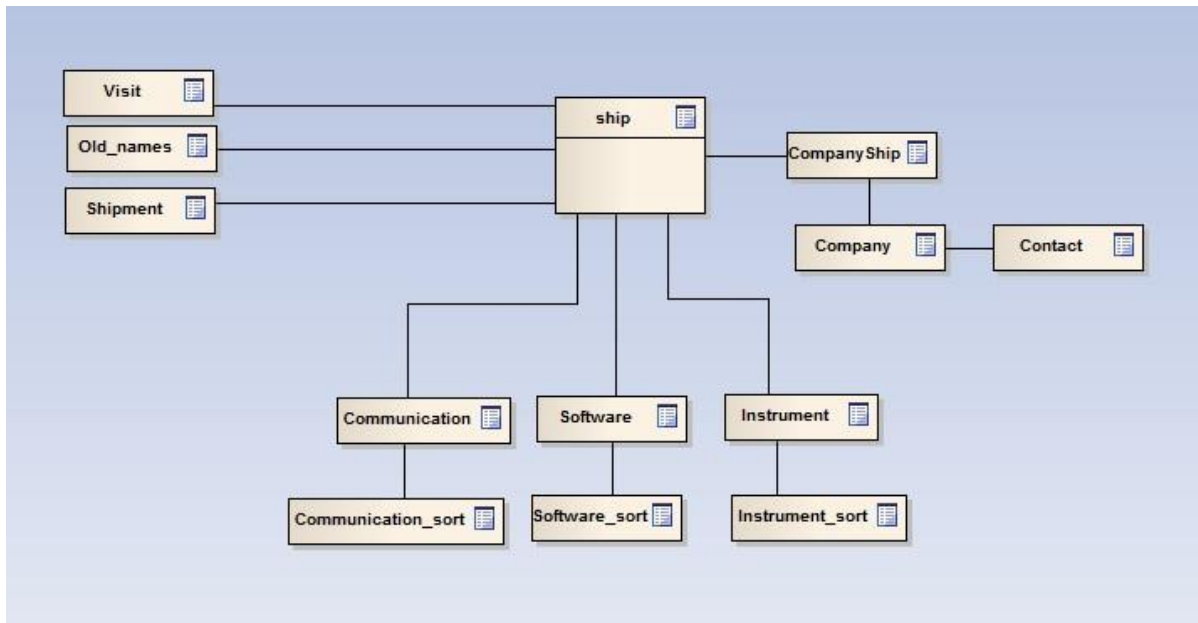


Figuur 6 Database versie 2

Elk schip heeft een maatschappij. Het is niet ondenkbaar dat een schip van maatschappij wisselt. Een tabel die bijhoudt welk schip bij welke maatschappij zit (en gezeten heeft) is dus handig. In deze tabel wordt het IMO nummer van het schip gekoppeld aan het *companyID* van de maatschappij. Een maatschappij kan meerdere contactpersonen hebben. Hierom is gekozen de contactpersonen in een aparte tabel te zetten en deze via het *companyID* te koppelen aan de maatschappij. Hierdoor ontstaat de in **figuur 7** afgebeelde situatie.



Afstudeerscriptie

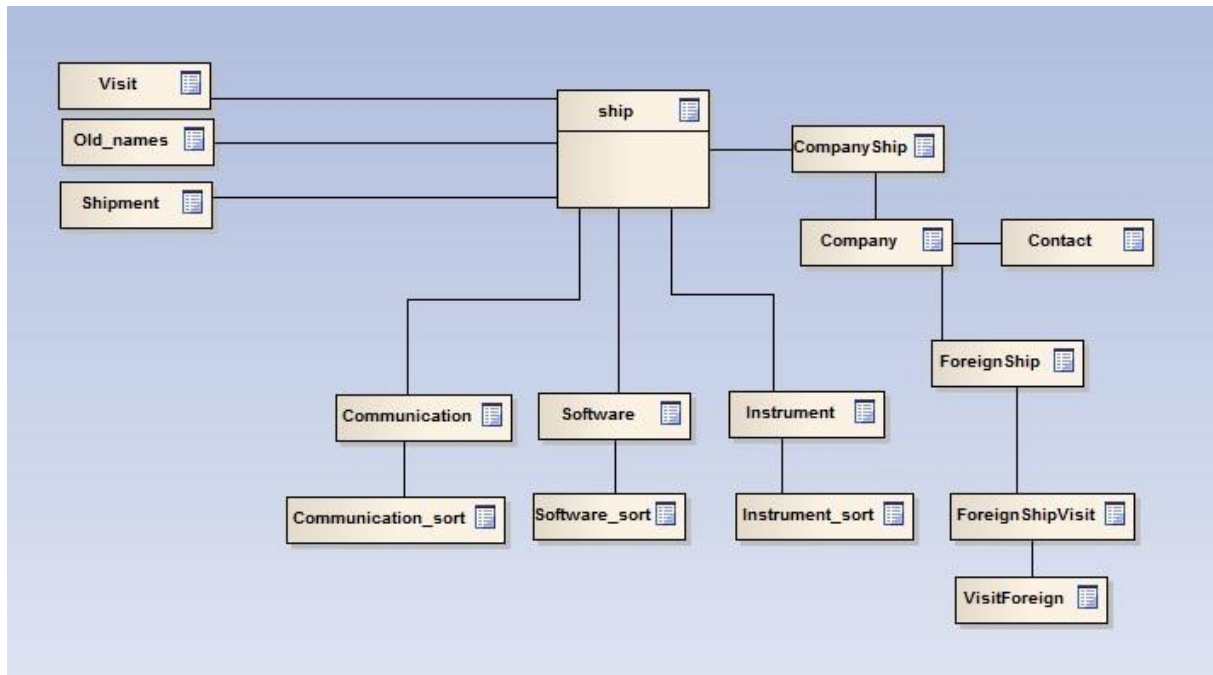


Figuur 7 Database versie 3

Er is voor een iets wat ongewone oplossing gekozen aangezien er in de database nog een tabel is met schepen. Deze tabel bevat de buitenlandse schepen. Welke ook uniek zijn door middel van hun IMO nummer. De eindgebruikers hadden als wens dat de buitenlandse schepen fysiek gescheiden bleven van de Nederlandse schepen in de database. Er werd door de opdrachtgever verzocht hiervoor twee aparte tabellen te maken wat dus ook gebeurd is. Buitenlandse schepen worden op dezelfde manier aan een maatschappij gekoppeld als Nederlandse schepen. Dit gaat tevens naar dezelfde tabel *company* toe aangezien het mogelijk is dat een maatschappij schepen heeft die voor Nederland werken en schepen die voor andere landen werken. Bij buitenlandse schepen wordt alleen de link met een maatschappij bijgehouden en bezoeken van het KNMI aan een dergelijk schip. Overige zaken zoals de instrumenten, software en communicatie middelen aan boord worden niet opgeslagen in de database voor dit soort schepen. Een link met deze tabellen is daarom ook niet noodzakelijk. Hierdoor ontstaat de in **figuur 8** getoonde situatie.



Afstudeerscriptie

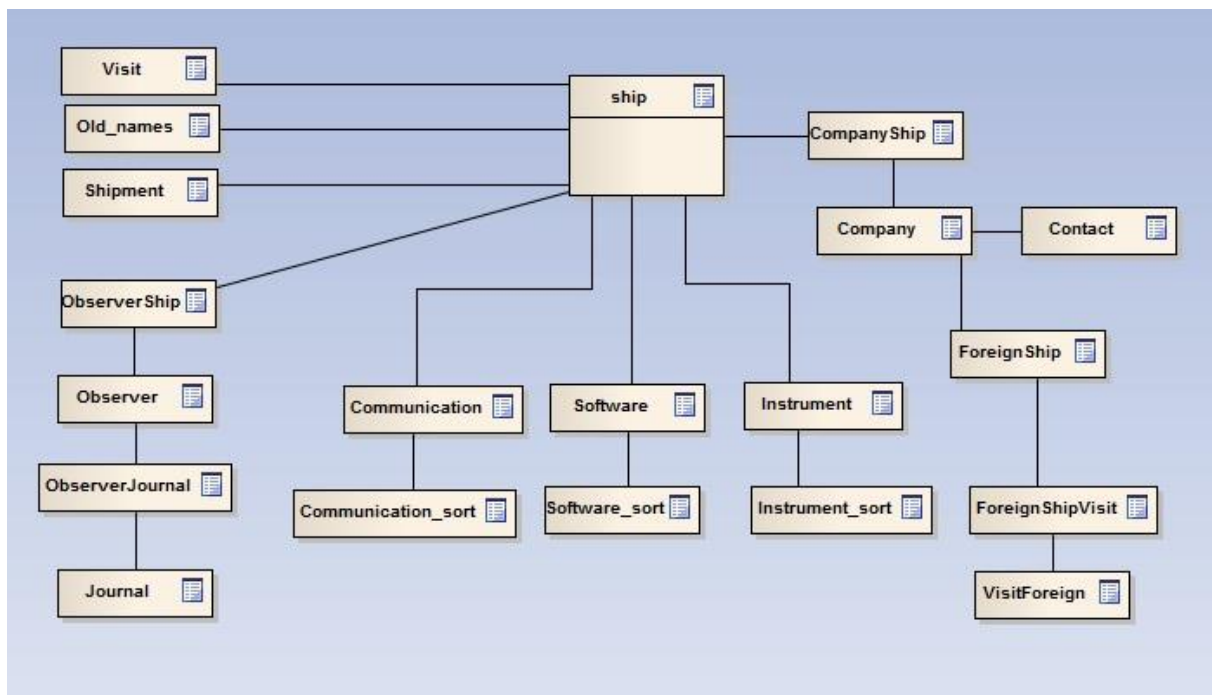


Figuur 8 Database versie 4

Een schip heeft verschillende waarnemers. Een waarnemer kan voor verschillende schepen werken. Via een aparte tabel *observerShip* wordt de waarnemer aan het schip gekoppeld. Tevens worden journaals bijgehouden. Deze journaals bevatten de tijdsduur van de gegevens die binnenkomen bij het KNMI. Aangezien de data van een journaal door verschillende waarnemers tot stand is gekomen worden de waarnemers via een tabel *observerjournaal* aan elk journal gekoppeld waaraan zij en bijdrage hebben geleverd. Hierdoor ontstaat de database zoals getoond in **figuur 9**.



Afstudeerscriptie

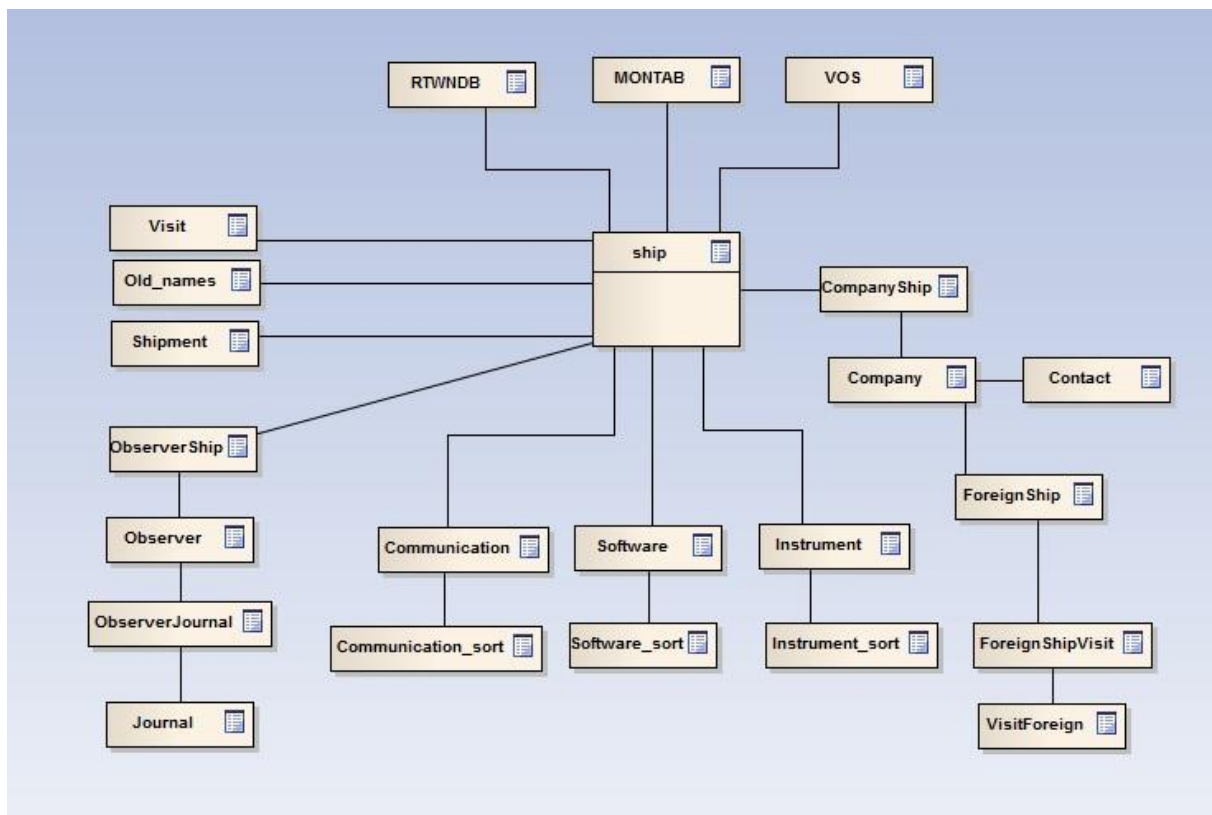


Figuur 9 Database versie 5

Tevens zijn er nog een aantal zaken aanwezig die door middel van files ingelezen worden. Al deze zaken kunnen teruggevoerd worden tot het schip door middel van het IMO nummer wat aanwezig is in deze files. Indien er geen IMO nummer aanwezig is wordt een Inmarsat-c nummer gebruikt. Inmarsat-c is een communicatie middel. Via de tabel communicatie kan zo achterhaald worden welk Inmarsat-c nummer bij welk IMO nummer hoort. De database komt er dan als in **figuur 10** getoond uit te zien



Afstudeerscriptie



Figuur 10 Database versie 6

Het is mogelijk dat een schip na van loop van tijd van naam veranderd. Een aparte tabel waarin de namen van de schepen worden opgeslagen is daarom handig. Hierbij wordt via het IMO nummer, de naam, roepletters, startdatum en einddatum van een naam doorgegeven. Door de actuele naam een einddatum '0' mee te geven kunnen we makkelijk de actuele naam achterhalen. Voor een uitgebreid diagram zie **bijlage 1**.

De database is omgezet in een normaalvorm. Er is express niet voor een al te hoge normaalvorm gekozen omdat hierdoor de performance van de database en dus de snelheid van de applicatie te ver achteruit gaat.

Eerste normaalvorm

Een relatie is in de eerste normaalvorm als hierbij alle attributen van deze relatie enkelvoudig zijn. Er komen dus geen repeating groups voor in de relatie. Tevens is elk attribuut atomair en bevat dus een enkele waarde. De database die ontwikkeld is staat in de eerste normaalvorm in de database komen geen repeating groups voor, en elk attribuut is atomair.

Tweede normaalvorm

Een relatie is in de tweede normaalvorm indien de relatie in de eerste normaalvorm is en ieder niet sleutel attribuut volledig functioneel afhankelijk is van iedere kandidaat sleutel. De database die ontwikkeld is staat ook in de tweede normaalvorm aangezien voldaan wordt aan bovenstaande voorwaarden.



Afstudeerscriptie

8.5 Ontwerp software

De basisfunctionaliteit van de software ligt op 2 vlakken:

- Gegevens verzamelen
- Ingevulde gegevens interpreteren

8.5.1 Gegevens verzamelen

Het verzamelen van de gegevens is onder te verdelen in twee verschillende subcategorieën:

- Inlezen bestanden.
- Handmatig invullen.

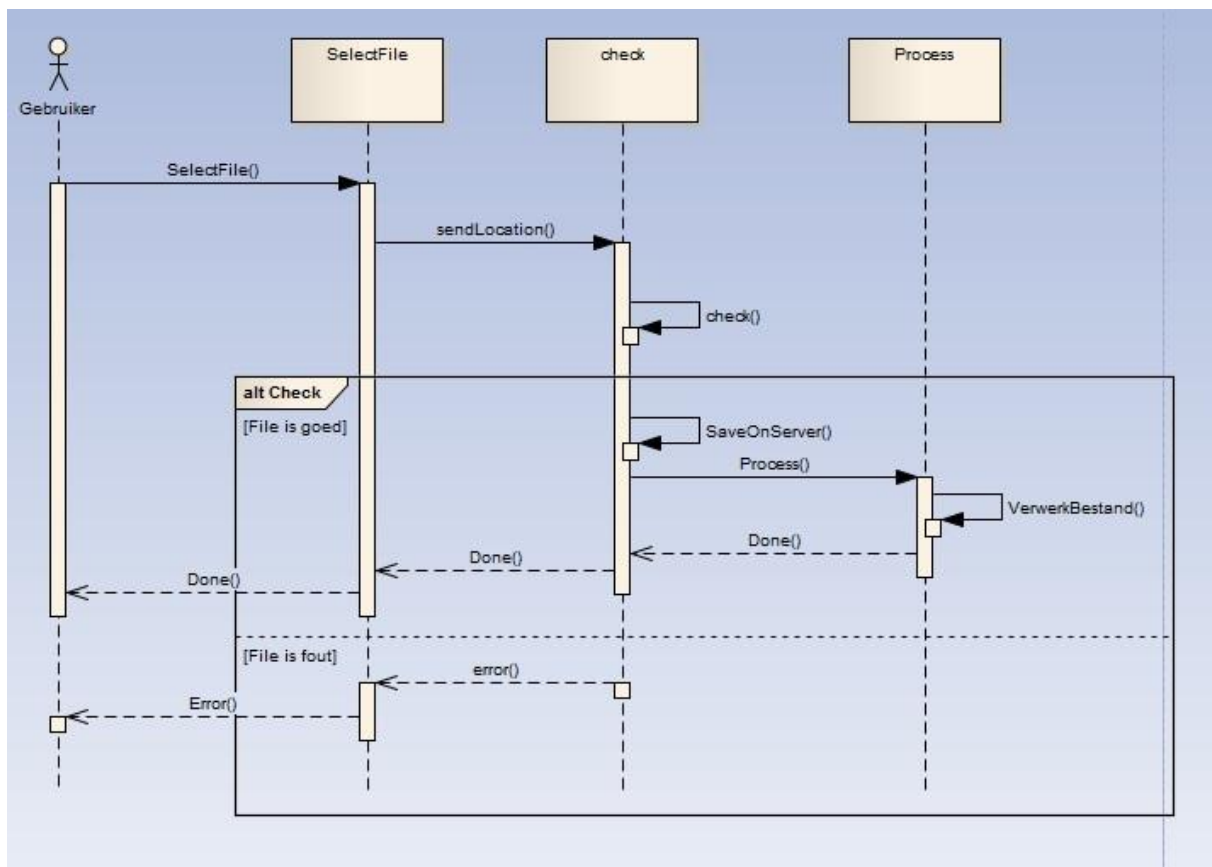
Inlezen bestanden

Voor de applicatie moeten verschillende bestanden met data ingelezen worden.

- Journaaldiskette
- MONTAB
- NOTA
- DMG
- RTWNDB
- Duitse schepen

Als eerste wordt een bestand wat verwerkt moet worden geselecteerd. Dit wordt gedaan door de functie `selectFile()`. Deze functie geeft de locatie van het bestand door aan de functie `check()`. Deze functie controleert aan de naam van het bestand of de extensie klopt. Indien dit zo is wordt het bestand opgeslagen op de server. Zodra het bestand op de server geplaatst is wordt door de functie `check()` de functie `process()` aangeroepen welke het bestand verwerkt. **Figuur 11** geeft hiervan een versimpelde weergave doormiddel van een Sequence diagram.

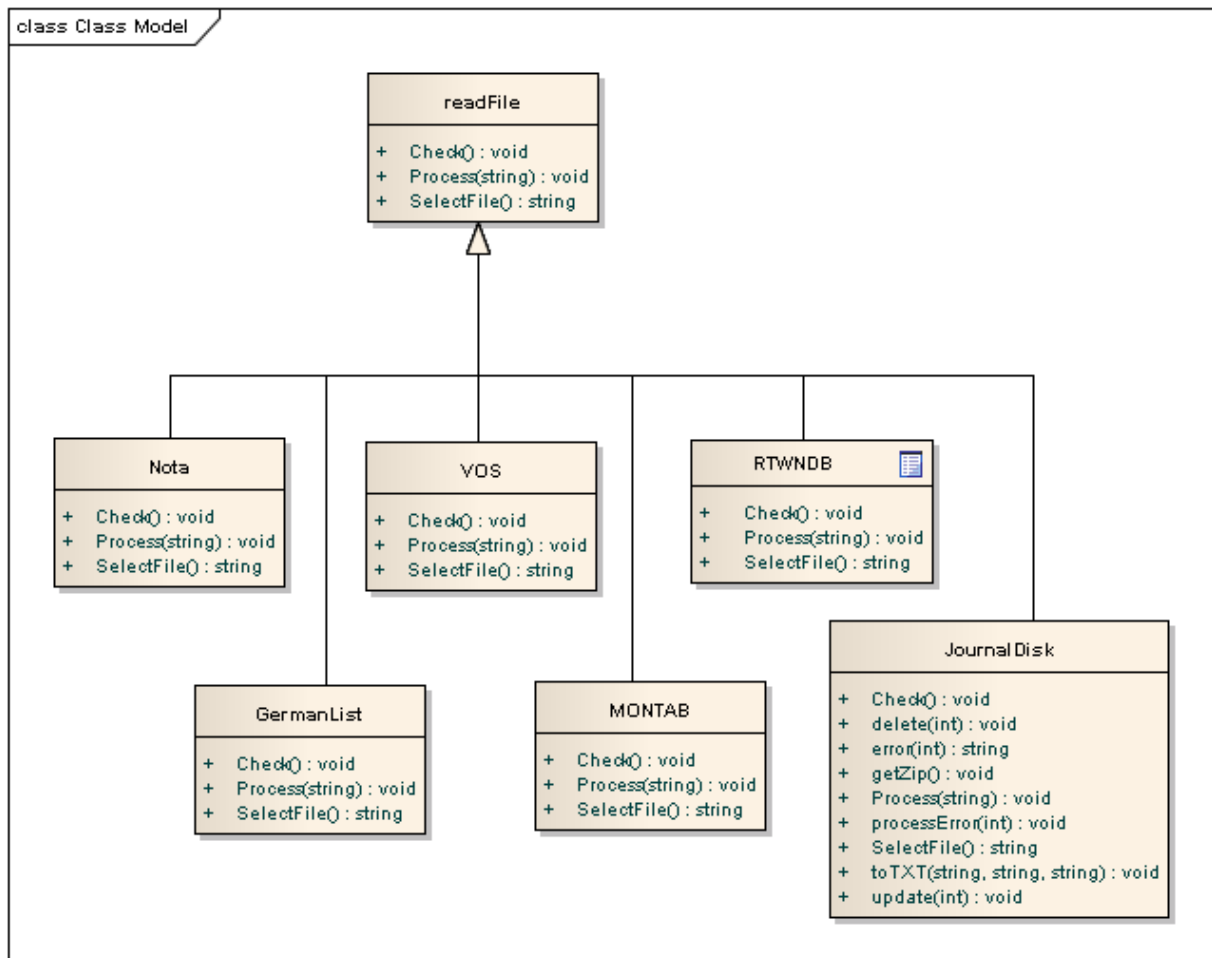
Afstudeerscriptie



Figuur 11 Sequence diagram gegevens inlezen

Aangezien elk bestand een aparte manier van verwerken heeft is voor elke bestand een aparte klasse gerealiseerd. Deze klassen beschikken allemaal over dezelfde functies namelijk , selectFile(), check() en process(). Tevens is er nog een 'algemene' klasse readFile welke ook deze functies bevat. Door middel van inheritance worden de andere klassen geextended aan deze klasse. Hierdoor beschikken alle klasse over dezelfde functionaliteit. Aangezien het selecteren en checken van de bestanden steeds op dezelfde manier gebeurd worden deze functies niet aangemaakt bij de subklassen. Hierdoor wordt bij een aanroep de code gebruikt die staat bij de klasse readFile. Echter maken we wel bij elke klasse de functie process() waardoor wel de juiste process functie aangeroepen wordt. **Figuur 12** geeft hiervan een betere weergave.

Afstudeerscriptie



Figuur 12 Klassendiagram inlezen bestand

Door nu bijvoorbeeld journaaldisk->selectFile() aan te roepen wordt eerst de functie selectFile() die gedefinieerd staat in readFile samen met de functie check() uitgevoerd. Hierna wordt de functie process() die gedefinieerd staat bij de klasse journaaldisk uitgevoerd. Indien er rtwnodb->selectFile() wordt aangeroepen gebeurt hetzelfde als bij journaaldisk alleen wordt nu de functie process() die bij RTWNDB gedefinieerd staat aangeroepen.

Voordeel hiervan is dat de functies selectFile() en check() maar één keer geschreven hoeven te worden. Tevens is het doorvoeren van wijzigingen nu makkelijker en kan dit op één centrale plaats gebeuren.

Handmatig invullen

In de applicatie is het ook mogelijk verschillende zaken handmatig in te voeren door middel van formulieren. Het gaat hierbij om de volgende elementen:

- Nederlands bezoek
- Buitenlands bezoek
- Software
- Observer



Afstudeerscriptie

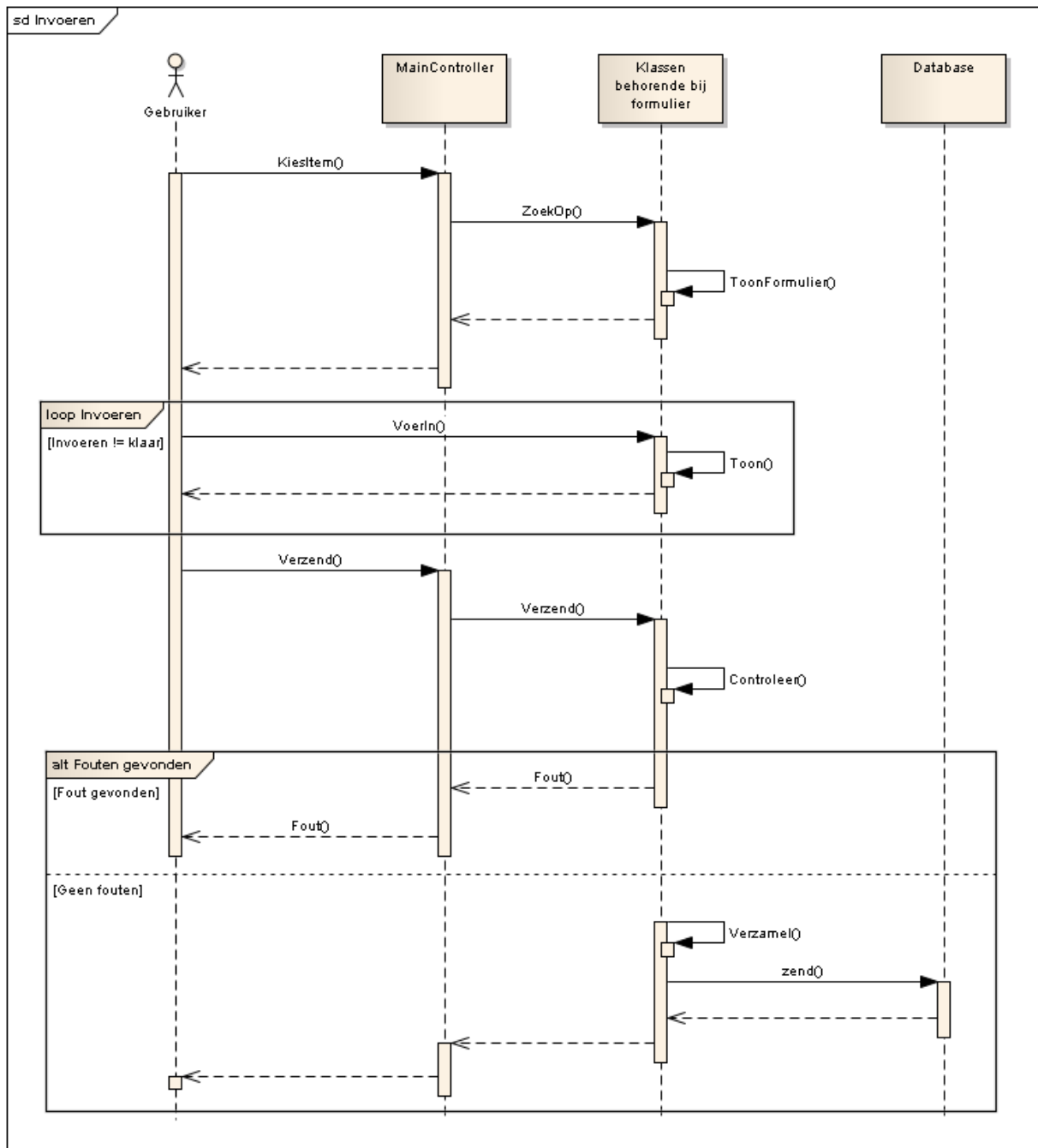
- Nederlands schip
- Buitenlands schip
- Scheepsnamen
- Instrumenten
- Journaals
- Verzendingen
- Maatschappijen
- Communicatie

Deze gegevens die ingevoerd worden moeten ook gevalideerd worden. Het is hierdoor handig om voor elk element (software, observer, schip, enz.) wat ingevuld moet worden een aparte klasse te maken. Deze klasse beschikt telkens over dezelfde functionaliteit alleen gebaseerd op wat voor gegevens opgehaald moeten worden. De klasse bevat hierom ook het formulier waarin de gegevens ingevuld moeten worden.

Om het formulier te tonen beschikken al deze klassen over een functie `getForm($i)` deze functie geeft het formulier wat ingevuld moet worden terug. De waarde die met `$i` wordt meegegeven bepaald of het formulier getoond moet worden voor het invullen van gegevens of dat alleen de gegevens bekeken mogen worden.

Het formulier roept de functie `process()` aan voor de verwerking van de ingevulde gegevens. Deze functie haalt alle gegevens uit het formulier en stuurt deze naar verschillende functies die één element van deze gegevens controleren op fouten. Indien er een fout gevonden wordt, wordt een error verzonden. Tevens wordt het dan voor de functie `process()` onmogelijk gemaakt de gegevens naar de database te verplaatsen. Hierdoor moet de gebruiker dus eerst de gegevens wijzigen om de data te kunnen opslaan. **Figuur 13** geeft dit process schematisch vereenvoudigd weer doormiddel van een sequencediagram.

Afstudeerscriptie



Figuur 13 Sequence diagram invoeren gegevens

Er is gekozen om aparte functies te maken voor het controleren van de data omdat alle data andere checks heeft. Het is hierdoor makkelijk om per element te zeggen waarop gecontroleerd moet worden. Tevens kan hier aangegeven worden of het invullen van het betreffende veld wel überhaupt verplicht is.

Tevens beschikt elke klasse over een functie `setData($parameter1, $parameter2, enz.)` deze functie vult de meegegeven parameters in, in de query die bij de functie staat en plaats hiermee de data op de juiste plaatsen in de database.

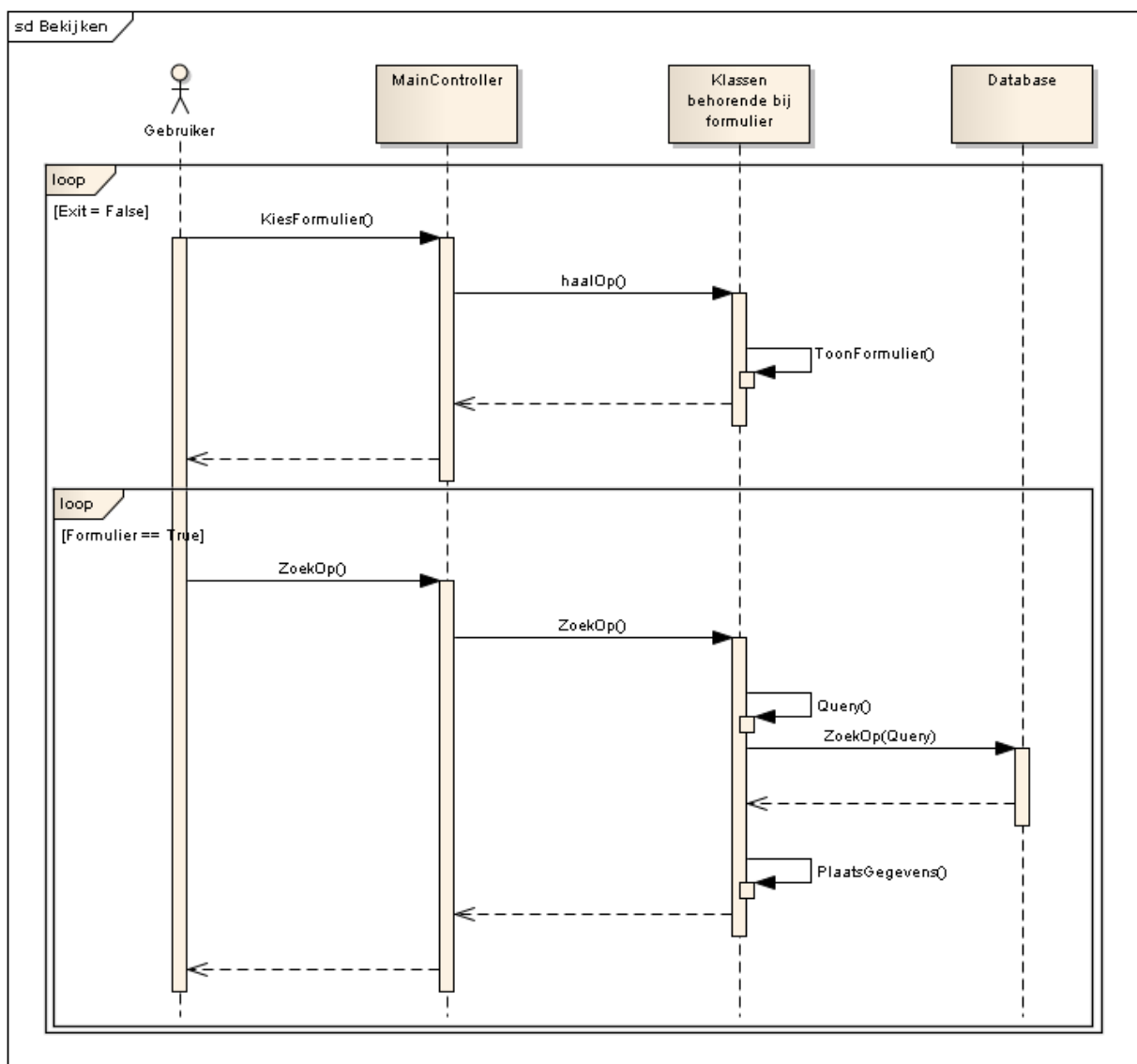


Afstudeerscriptie

De functie `changeData($parameter1, $parameter2, enz.)` doet hetzelfde met het enige verschil dat hierbij bestaande data wordt geüpdate in plaats van nieuwe data in de database wordt geplaatst.

De functie `deleteData($id)` verwijdert de data behorend bij de het desbetreffende formulier met het opgegeven ID uit de database.

De functie `getData($id)` haalt de data behorende bij het ID en het formulier op uit de database. Deze data wordt geplaatst in de bijbehorende private variabelen. Indien hierna de functie `getForm()` aangeroepen wordt, wordt de opgehaalde data in het formulier getoond. Het formulier toont namelijk de private variabelen op de juiste plaats in het formulier. Dit process wordt schematisch weergegeven in **Figuur 14**.

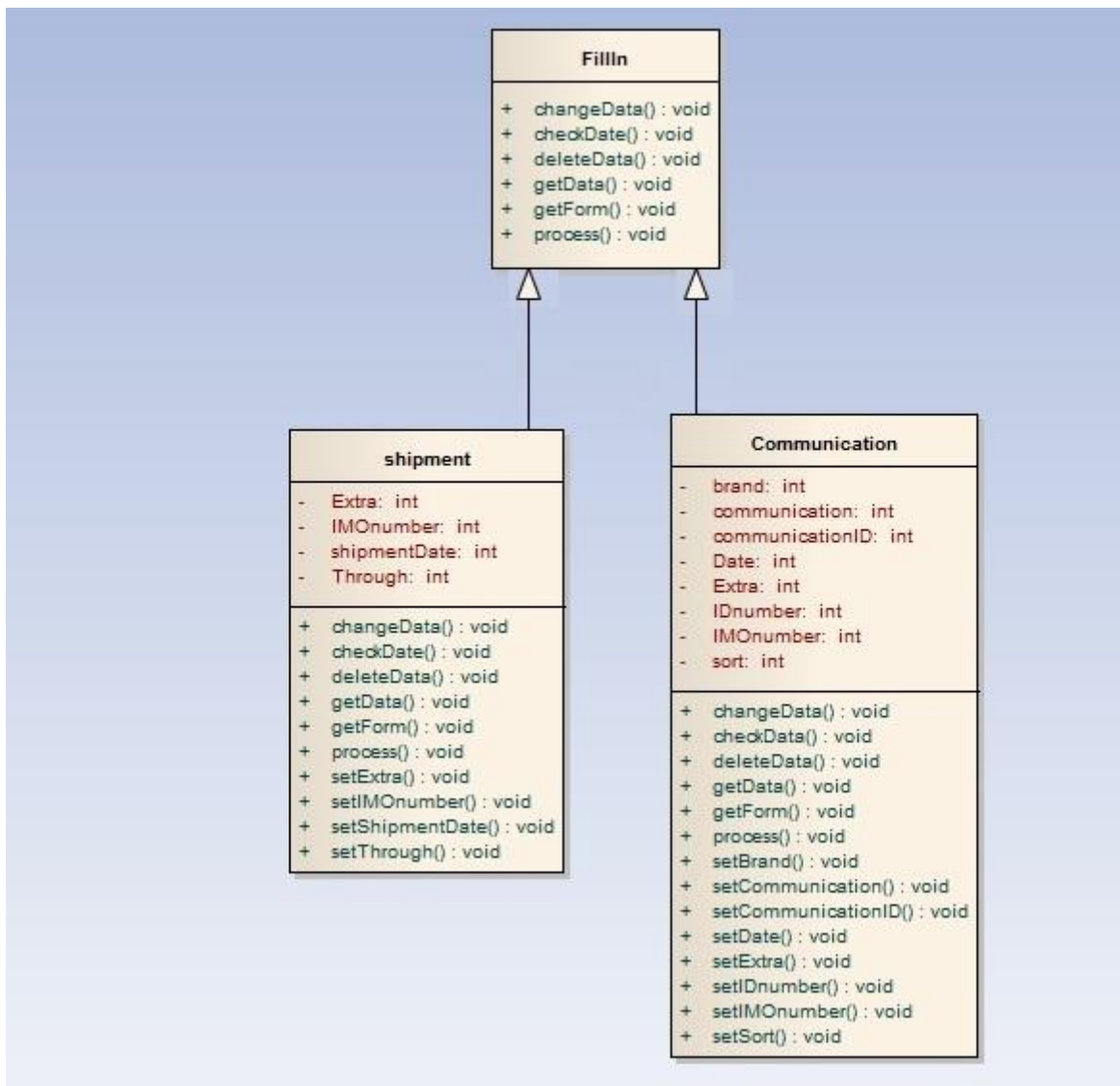


Figuur 14 Sequence diagram gegevens bekijken

Meer sequencediagrammen over de verschillende processen zijn te vinden in **Externe bijlage 3**. Aangezien voor elke klasse weer grotendeels dezelfde functies gebruikt worden, worden deze door middel van inheritance vanuit de klasse `fillIn` afgeleid. Zo

Afstudeerscriptie

moet er bijvoorbeeld vaak een datum gecontroleerd worden. Aangezien het hierbij altijd om de basisfunctionaliteit gaat (is de datum in het verleden, bestaat deze wel) is hiervoor in de klasse fillIn een aparte functie gemaakt. Deze functie wordt door middel van inheritance doorgegeven aan alle klasse. Hierdoor hoeft indien dit mechanisme moet worden aangepast dit maar op één plek te gebeuren. **Figuur 15** geeft een vereenvoudigde weergave van het principe. Voor een groot overzicht van alle klassen kunnen **Bijlage 5 en 6** geraadpleegd worden. **Bijlage 9** geeft weer voor welke zaken een formulier beschikbaar is.



Figuur 15 Klassendiagram invullen gegevens

8.5.2 Ingevulde gegevens interpreteren

Gegevens die in het systeem worden geplaatst moeten een interpretatie krijgen en vervolgens teruggekoppeld worden naar de gebruiker. Dit gebeurt op verschillende manieren, namelijk door middel van:



Afstudeerscriptie

- Rapporten
- Statistieken
- Lijsten
- SQL Builder

Rapporten

Rapporten bevatten samengevoegde informatie uit de database. Een rapport bestaat uit één of meerdere query's. Deze query's vormen samen het rapport. Dit rapport is te downloaden als .doc of .odt

De klasse rapport bevat een functie `getForm()` welke een formulier terugstuurt. In dit formulier kan aangegeven worden welk rapport gemaakt moet worden. Ook kunnen hier waardes ingevoerd worden bijvoorbeeld alleen voor de schepen met beginletter A tot C. Het formulier roept de functie `process()` aan welke de ingevoerde data interpreteert. In de functie `process()` wordt de functie `getQuery()` aangeroepen. Deze functie haalt de gewenste Query die apart is opgeslagen op en stuurt deze als string terug naar de functie `process()`. In deze query zijn herkenningpunten aangebracht zoals `#from`. De functie `process()` zoekt deze punten op en vervangt deze door de waardes die in het formulier zijn ingevoerd.

Voorbeeld:

*Men wil de maatschappijen opvragen die beginnen met een naam die ligt van A t/m C. De Query in **code fragment 1** wordt hiervoor opgehaald.*

```
SELECT Name, Subname, Visiting_address, zip_code, visiting_site
FROM company
WHERE Name
BETWEEN #from AND #until
```

Code fragment 1 Query company

*In het formulier is ingevoerd dat alleen de namen tussen A en C meegenomen moeten worden. De functie `process` zoekt vervolgens naar `#from` en `#until` en vervangt deze door respectievelijk A en D. Hierdoor wordt de Query veranderd in de query uit **code fragment 2**.*

```
SELECT Name, Subname, Visiting_address, zip_code, visiting_site
FROM company
WHERE Name
BETWEEN 'A' AND 'D'
```

Code fragment 2 Query company ingevuld

Wanneer dit proces klaar is wordt de query doorgestuurd naar de Output van de klasse `SQLBuilder`. Deze functie verwerkt de query en geeft een tabel met de gewenste data terug.

Deze data wordt vervolgens door de functie `process()` in een document gezet en ter download aangeboden.

Zoals in bovenstaand voorbeeld is te zien wordt wanneer de namen van A t/m C meegenomen moet worden in de query *Between 'A' AND 'D'* neergezet. Dit komt omdat bij mysql geldt van A tot D en niet tot en met. Hierdoor moet de letter in de query één verder in het alfabet zijn dan in het formulier is ingevuld.



Afstudeerscriptie

Zodra de namen van C t/m Z opgevraagd worden. Wordt de query op een andere manier aangepast. Er kan namelijk geen letter na de Z neergezet worden waardoor de query veranderd wordt in NOT BETWEEN 'A' AND 'C' hierdoor wordt in plaats van wat meegenomen moet worden gezegd wat juist niet meegenomen moet worden.

Statistieken

Statistieken bestaan net als rapporten uit één of meerdere query's. De klasse statistiek bestaat ook weer uit een functie getForm() welke het formulier terugstuurt. De functie process() verwerkt de data uit het formulier. Vervolgens wordt de functie Output() van de klasse SQLBuilder aangeroepen. Deze functie stuurt vervolgens een tabel met de output terug welke getoond wordt.

Lijsten

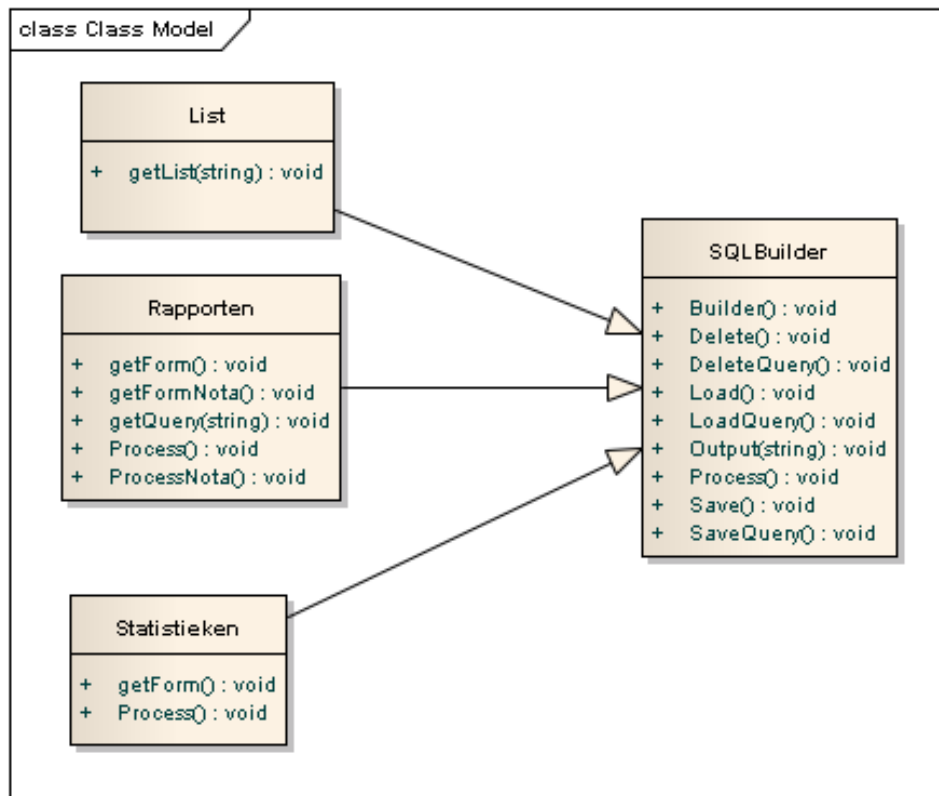
Lijsten zijn een directe weergave van de volledige inhoud van een tabel. De klasse List bevat een functie getList welke de gewenste lijst ophaalt doormiddel van het versturen van een query naar de functie Output() van de klasse SQLBuilder.

SQL Builder

De SQL builder stelt de gebruiker in staat zelf Query's uit te kunnen voeren op de database. De functie Builder() roept het formulier op waarin de query gemaakt kan worden. De functie process() verwerkt de query vervolgens. De query wordt hierbij doorgestuurd naar de functie Output() deze voert de query uit en maakt vervolgens een tabel met kop welke teruggestuurd wordt. De functies save(), load() en delete() zorgen ervoor dat een query ook opgeslagen, geladen en verwijderd kan worden.

Figuur 16 geeft een duidelijker weergave van deze klassen.

Afstudeerscriptie



Figuur 16 Klassendiagram SQLBuilder

Het klassendiagram is samen met de sequensendiagrammen opgenomen in het ontwerpdocument. Dit document beschrijft de werking van het systeem. Het ontwerpdocument is opgenomen als **Externe bijlage 3**.

8.6 Planning

Na aanleiding van het software requirements specification document is het mogelijk een betere planning te geven voor de werkzaamheden. Bij deze planning ontstaan de verschillende iteraties bij de constructionfase mede door de indeling van de functionele eisen. Belangrijke eisen die bij de Must have's staan moeten eerst geïmplementeerd worden en vinden daarom ook in de eerste iteraties plaats. De planning ziet er als volgt uit:

Week	Activiteiten
Week 6 (15-03-10/19-03-10)	Afmaken ontwerp
Week 7 (22-03-10/26-03-10)	Iteratie 1
Week 8 (29-03-10/02-04-10)	Iteratie 1
Week 9 (05-04-10/09-04-10)	Iteratie 2
Week 10 (12-04-10/16-04-10)	Iteratie 2
Week 11 (19-04-10/23-04-10)	Iteratie 2
Week 12 (26-04-10/30-04-10)	Iteratie 3
Week 13 (03-05-10/07-05-10)	Iteratie 3
Week 14 (10-05-10/14-05-10)	Iteratie 3
Week 15 (17-05-10/21-05-10)	Iteratie 4



Afstudeerscriptie

Week 16 (24-05-10/28-05-10)	Iteratie 4
Week 17 (31-05-10/04-06-10)	Testen/afronden

Tabel 3 Planning lange termijn

Als eerste worden de must haves van de functionele specificaties uitgevoerd waarna over gegaan wordt op de should haves. Indien er meer tijd beschikbaar is dan gepland worden de could haves uitgevoerd. De iteraties bevat de volgende elementen:

Iteratie 1:

Deze iteratie begint met het ontwikkelen van de ontworpen database. De database vormt een belangrijk onderdeel van het systeem en moet dus zo vroeg mogelijk ontwikkeld worden.

- Nieuwe schepen toevoegen.
- Nieuwe waarnemers toevoegen.
- Maatschappij toevoegen.
- Buitenlandse schepen toevoegen.
- Nederlandse bezoeken toevoegen.
- Buitenlandse bezoeken toevoegen.
- Journaals invoeren.
- Verzendingen naar schepen bijhouden.
- Soorten instrumenten bijhouden.
- Instrumenten aan boord van schepen bijhouden
- Communicatiemiddelen schepen bijhouden

Iteratie 2:

- Journaal diskette verwerken.
- MONTAB gegevens verwerken.
- Gevonden fouten moeten door gebruiker kunnen worden hersteld.
- Plaatsingsdatum instrumenten bijhouden.
- Versie en uitgifte datum op schepen bijhouden.
- DMG bestand verwerken.

Iteratie 3:

- Stratos notabestanden inlezen.
- Nota's verwerken.
- Data moet in rapporten gezet worden.
- Ingelezen data moet door het systeem gecontroleerd worden op fouten.

Iteratie 4:

- Rapporten over laatste bezoekdatum en aantal actieve schepen moeten uitgedraaid kunnen worden.
- Aantal waarnemingen bijhouden.

Het is mogelijk dat na elke iteratie een paar kleine wijzigingen plaatsvinden in de planning. Na elke iteratie wordt met de toekomstige gebruikers gekeken of de



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

gerealiseerde functionaliteit naar wens is. Mogelijk commentaar/ aanvullingen worden dan meegenomen in de volgende iteratie.

8.7 Milestone 2

Nadat het software requirements specification document en het ontwerpdocument voltooid zijn is er overleg geweest met de gebruikers. Deze hebben de documenten in kunnen zien en hierop hun commentaar kunnen geven. Dit commentaar is vervolgens verwerkt in de documenten.

Uit de 2^e milestone is op te maken dat de eisen van de toekomstige gebruikers goed zijn vastgelegd en hierbij een passend ontwerp bij is gevonden.



Afstudeerscriptie

9. Construction fase handmatige invoer

Construction fase één omvat voornamelijk de ontwikkeling van de elementen die handmatige invoer vereisen. Deze invoer wordt gerealiseerd doormiddel van formulieren

9.1 Lay-out

Voor de website moet ook een lay-out ontwikkeld worden. Er is gekozen om de HTML van de lay-out los te houden van de programmacode. Hierdoor kan makkelijk centraal de lay-out aangepast worden. De pagina *begin.html* bevat de HTML code. In deze code staat een menu waarmee de verschillende items te raadplegen zijn. Op de plaats waar de output van de verschillende PHP functies moet komen staat in de HTML %body%. De links van de verschillende items gaan naar verschillende PHP pagina's die vervolgens doormiddel van de parameters die bij de link meegegeven worden de juiste functie aanroepen. Indien de functie bijvoorbeeld een formulier teruggeeft als string, wordt deze string opgeslagen in de parameter \$input. Onder aan de pagina wordt in begin.html gezocht op %body%. Dit wordt vervangen door de inhoud van \$input. Hierdoor wordt op de juiste plaats het formulier getoond in de HTML. De lay-out is opgebouwd doormiddel van CSS.

9.2 ontwikkeling handmatig invoeren

De eerste iteratie omvat de ontwikkeling van voornamelijk de handmatige invoer. Deze invoer gebeurt via verschillende formulieren. Zaken die handmatig ingevoerd moeten kunnen worden in het systeem zijn:

- Instrumenten
- Instrument soorten
- Verzendingen
- Communicatie
- Communicatie soorten
- Software
- Software soorten
- Schepen
- Namen van schepen
- Waarnemers
- Maatschappijen
- Contactpersonen
- Journaals.

Elk van deze formulieren die tevens een eigen tabel hebben in de database krijgen een eigen klassen toegewezen. Deze klassen bevat het formulier wat de gegevens moet verzamelen. Tevens zijn functies opgenomen die de gegevens moeten checken. De klassen zorgt er tevens voor dat de gegevens op de goede plaats in de database komen te staan. Het is ook mogelijk om de gegevens te wijzigen. De klassen haalt dan de juiste gegevens op en laat deze in het formulier zien. De gebruiker krijgt dan de mogelijkheid de gegevens aan te passen. Waarna de gegevens opnieuw gecontroleerd worden en de database geüpdate wordt. Gegevens verwijderen is ook mogelijk door middel van de functie delete() die elke klassen heeft.



Afstudeerscriptie

Bij klassen zoals instrument, verzending en communicatie moet in het formulier het IMO-nummer ingevuld worden. Door het IMO-nummer kan de data dan namelijk gekoppeld worden aan het schip in de database. Het IMO-nummer is echter een moeilijk te onthouden nummer. Tevens is het mogelijk om een tikfout te maken. Hierom is ervoor gekozen om de gebruiker niet het IMO-nummer maar de naam van het schip te kiezen. In de code wordt dan de scheepsnaam aan het IMO-nummer gekoppeld en het IMO-nummer doorgegeven aan de juiste kolom in de database. Voor de gebruiker is dit echter een gebruiksvriendelijkere manier die tevens minder fout gevoelig is.

Elk formulier laadt in de invoervelden de waarde die opgeslagen staan bij de private variabelen. Indien er nieuwe gegevens ingevoerd moeten worden zijn deze variabelen leeg en krijgt men een leeg formulier te zien. Echter wanneer men een item wil wijzigen worden eerst de gegevens uit de database gehaald en in de private variabelen gestopt. Hierna wordt pas het formulier getoond. Het formulier wat nu getoond wordt laat de gegevens die in de database staan nu in het formulier zien. Dit is ook mogelijk wanneer men alleen gegevens wil bekijken zonder deze te wijzigen. Hiervoor geeft men een parameter mee aan `getForm` die aangeeft dat het formulier readonly getoond moet worden. Het is dan dus niet meer mogelijk de gegevens nog te wijzigen.

De checkfuncties die de data die ingevoerd wordt moeten controleren hebben de mogelijkheid een foutmelding te geven. Deze foutmelding wordt weggeschreven in de array `error`. Deze array wordt in een kolom naast het formulier getoond. Hierdoor komt de error direct naast het item waarop dit van toepassing is te staan. Indien ergens een foute waarde is ingevoerd wordt de variabele `check` 1 gemaakt. Dit zorgt ervoor dat de data niet echt verwerkt wordt. De data mag namelijk alleen, zoals in **externe bijlage 2** naar voren is gekomen, naar de database geschreven worden zolang alle data correct is. Hiervoor moet `check` 0 zijn. Wanneer het geheel dus een fout bevat wordt de data niet weggeschreven maar dient de gebruiker de fout eerst te herstellen.

De variabele die bij `getForm()` wordt meegegeven bepaald mede hoe het formulier er uit komt te zien. Indien `readonly` wordt meegegeven is het formulier alleen zichtbaar en ontbreekt de submit knop. Indien `change` wordt meegegeven bevat de submit knop de tekst `change values`. In de overige gevallen wordt gewoon submit weergegeven.

9.3 Evaluatie

Evaluatie met de toekomstige gebruikers heeft aangetoond dat de basisfunctionaliteit die gerealiseerd is overeenkomt met wat de gebruiker in gedachten had. Men wil graag wel de mogelijkheid om per tabel eenvoudig te zien wat erin staat. Dit onderdeel kan eenvoudig gerealiseerd worden bij het ontwikkelen van de rapporten. Aangezien hier een functie in zit die bij het opsturen van een Query een tabel terug geeft. Hier kan voor elke tabel een query gemaakt worden die alles teruggeeft van die tabel.



Afstudeerscriptie

10. Construction fase inlezen bestanden

Construction fase iteratie twee concentreert zich voornamelijk op het inlezen van verschillende bestanden. Hieronder vallen:

- Journaaldiskette
- MONTAB
- DMG bestanden

Tevens vallen de laatste zaken die handmatige invoer vereisen onder deze fase.

10.1 Montab

Montab is een afkorting voor montly table. Dit bestand bevat per schip de gegevens van een maand. Elke maand wordt dit bestand opgestuurd naar het KNMI en dit moet ingelezen worden in het systeem. In dit bestand zijn van elk (Nederlands) schip de volgende zaken te vinden:

- PRESSURE (hPa)
- TEMPERATURE (deg C)
- WIND SPEED (m s⁻¹)
- WIND DIRECTION (deg)
- RELATIVE HUMIDITY (%)
- SEA SURFACE TEMPERATURE (deg C)

Deze elementen staan niet gegroepeerd per schip, maar zijn gegroepeerd per onderwerp. Een schip komt daardoor meerdere keren voor in het bestand. Een (verkort) bestand zou er bijvoorbeeld uit kunnen zien als in **Code fragment 3**.

UK MET. OFFICE MONITORING STATISTICS (Mrt 2010) PRESSURE (hPa)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	AALSMEERGRACHT	PCAM	19	0	0.1	0.7
NL	ACHTERGRACHT	PCCL	7	0	-1.1	0.8
NL	ADMIRALENGRACHT	PCDE	22	0	-1.2	0.5
TEMPERATURE (deg C)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	AALSMEERGRACHT	PCAM	19	0	0.3	1.5
NL	ACHTERGRACHT	PCCL	7	0	0.7	2.1
NL	ADMIRALENGRACHT	PCDE	22	0	-0.2	0.7
WIND SPEED (m s ⁻¹)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	AALSMEERGRACHT	PCAM	19	0	-1.0	2.4
NL	ACHTERGRACHT	PCCL	7	0	3.7	1.1
NL	ADMIRALENGRACHT	PCDE	22	0	0.7	3.2



Afstudeerscriptie

WIND DIRECTION (deg)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	ADMIRALENGRACHT	PCDE	18	0	6.0	32.5
NL	AFRICABORG	PHMH	20	0	7.4	8.6
NL	AGULHAS STREAM	PJKV	58	0	3.5	22.2
RELATIVE HUMIDITY (%)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	AALSMEERGRACHT	PCAM	17	0	0.6	6.8
NL	ACHTERGRACHT	PCCL	7	0	4.9	12.2
NL	ADMIRALENGRACHT	PCDE	22	0	-2.8	7.5
SEA SURFACE TEMPERATURE (deg C)						
CTRY	SHIP NAME	CALL-SIGN	TOTAL	% GROSS ERROR	BIAS	STANDARD DEVN
NL	ACHTERGRACHT	PCCL	5	0	0.7	0.9
NL	ADMIRALENGRACHT	PCDE	42	0	0.4	1.4
NL	AFRICABORG	PHMH	34	0	1.3	0.9

Code fragment 3 MONTAB bestand

De afstand tussen de verschillende elementen in het bestand zijn bekend. Zo loopt bijvoorbeeld de CTRY van positie 0 t/m 3 en de Ship name van 4 t/m 15. Om er achter te komen waar de verschillende waarden voor staan is het zaak op zoek te gaan naar de namen die boven elk kopje staan. Als een bekende kopnaam is gevonden is het duidelijk dat de waarden daaronder betrekking hebben op die naam. Totdat een andere kopnaam gevonden wordt. Door middel van een if statement wordt gekeken onder welke kop de elementen die ingelezen worden vallen.

```
else if((substr($buffer, 61, 8)) == "PRESSURE")
{
    //The data we now read have something to do with pressure
    $type = "pres";
}
```

Code fragment 4 Achterhalen van Pressure waardes

In **Code fragment 4** is te zien dat wanneer de kop "PRESSURE" gevonden wordt \$type de waarde "pres" krijgt. Dit wordt gedaan voor alle mogelijke koppen. Voor elke regel wordt dus gekeken of het een kop betreft. Indien er op een regel geen kop gevonden wordt betekent dit dus dat we een regel met data gevonden hebben. Deze regel behoort automatisch bij de kop die in \$type staat. Aangezien dit de laatste kop is die door het programma is gevonden.

Als eerste worden de waardes uit het bestand gehaald het betreft hier het Callsign, het aantal waarnemingen, de afwijking en de Stdev.

Vervolgens wordt door middel van een Query gekeken of het Callsign gevonden kan worden in de database. Hierdoor krijgen we de beschikking over het bijbehorende IMO_number wat de primaire sleutel gaat vormen.



Afstudeerscriptie

Hierna komen we in een switch statement die opgebouwd is rond de \$type. \$type bepaald in welke case we komen. Hiermee wordt dus bepaald of de waarde afwijking die opgehaald is bijvoorbeeld in pressure_afwijking of in wind_afwijking geplaatst moet worden.

Aangezien de waarden in de database niet per type maar per schip worden opgeslagen wordt voordat de data in de database geplaatst wordt gekeken of er al een record met dezelfde maand van het desbetreffende schip bestaat. Indien dit het geval is wordt er geen record gemaakt, maar wordt het bestaande record geüpdate. Reden hiervoor is dat wanneer bijvoorbeeld eerst de pressure is opgehaald, en hiermee een nieuwe regel in de database wordt aangemaakt. Er geen nieuwe regel wordt aangemaakt wanneer de wind wordt opgehaald, de data van wind komt dan op dezelfde regel in de database te staan bij de kolomen voor wind. **Code fragment 5** geeft de code van deze functie weer.

```
switch($type)
{
    case "pres":
        if($data == false)
        {
            //make query and send to database
            $sql_query = "INSERT INTO
`montab`(`IMO_number`,`Reference_date`,`Pressure_number_observation`,`Press
ure_deviation`,`Pressure_stddev`)" ;

            $sql_query .= "VALUES ('" . $IMOnumber .
"', '" . $peildatum . "', '" . $Aantalwaarn . "', '" . $Afwijking . "', '" . $Stdev .
"');" ;

            mysql_query($sql_query);
        }
        else
        {
            $sql_query = "UPDATE `montab`
SET `Pressure_number_observation`='" . $Aantalwaarn . "',
`Pressure_deviation`='" . $Afwijking . "',
`Pressure_stddev`='" . $Stdev . "'
WHERE `IMO_number`='" . $IMOnumber . "' AND
`Reference_date`='" . $peildatum . "'";

            mysql_query($sql_query);
        }
        break;

    case "wind":
        if($data == false)
        {
            //make query and send to database
            $sql_query = "INSERT INTO
`montab`(`IMO_number`,`Reference_date`,`Wind_deviation`,`Wind_stddev`,`Wind_
number_observation`)" ;

            $sql_query .= "VALUES ('" . $IMOnumber .
"', '" . $peildatum . "', '" . $Afwijking . "', '" . $Stdev . "', '" . $Aantalwaarn .
```



Afstudeerscriptie

```
"');";

mysql_query($sql_query);
}
else
{
    $sql_query = "UPDATE `montab`
SET `Wind_deviation`='" . $Afwijking. "',
`Wind_stddev`='" . $Stdev. "',
`Wind_number_observation`='" . $Aantalwaarn. "'
WHERE `IMO_number`='" . $IMOnumber . "' AND
`Reference_date`='" . $peildatum. "'";

    mysql_query($sql_query);
}
break;

case "no":
    // do nothing
    break;
}
```

Code fragment 5 Verwerking MONTAB gegevens

10.2 Journaaldiskette

De schepen behorende bij de VOS vloot verzamelen een aantal maal per dag de gegevens van de instrumenten aan boord. Deze gegevens worden vervolgens ingevuld in een programma aan boord van het schip. Dit programma slaat de gegevens via een vaste indeling op in een .txt bestand. Om de paar maanden worden deze bestanden verstuurd naar het KNMI. Dit opsturen kan op 2 manieren gebeuren: via email of een diskette.

Ongeacht welke methode wordt gebruikt komen de gegevens bij het KNMI aan. In het .txt bestand staan de gegevens van de waarnemingen. Aangezien de gegevens handmatig ingevoerd worden is er een mogelijkheid dat er foutieve informatie is ingevuld. Hierbij moet gedacht worden aan fouten in de datum/tijd. De datum en tijd moeten in het bestand oplopen. Wanneer dit niet zo is, betekent dit dat er een fout in het bestand aanwezig is.

Deze fouten moeten door de software gevonden worden. Echter het 'repareren' van deze fouten moet handmatig gebeuren. Het is voor software namelijk erg moeilijk om in sommige gevallen te achterhalen wat precies de fout is. De eis van de gebruiker is dat de fouten handmatig gerepareerd worden. De volgende fouten kunnen bijvoorbeeld voorkomen:

- *De datum loopt af in plaats van op.*
De datum 12-03-2010 13:00 wordt opgevolgd door 12-02-2010 18:00. De apparatuur geeft hierbij aan dat 12-02-2010 18:00 fout is omdat deze waarde lager is dan de voorgaande datum. Echter kijken we nog een datum terug dan zien we de datum: 12-02-2010 06:00. Hieruit kunnen we opmaken dat niet 12-02-2010 18:00 fout is maar dat bij 12-03-2010 13:00 per ongeluk de verkeerde maand is ingevoerd. Echter is in andere gevallen het wel mogelijk dat 12-02-2010 18:00 verkeerd is.



Afstudeerscriptie

- *De datum is hetzelfde.*
De datum 12-03-2010 13:00 wordt opgevolgd door 12-03-2010 13:00. Dit kan 2 dingen betekenen of de tijd is verkeerd ingevoerd. Of de gegevens zijn door twee bemanningsleden ingevoerd waardoor een dubbel record ontstaat.

In sommige gevallen is niet meer te achterhalen wat de waarden wel moeten zijn. In deze gevallen wordt het record net als bij een dubbelrecord verwijderd. Het is namelijk beter om minder records te hebben dan foutieve records.

Programma oplossing

Het bestand met de records wordt ingelezen. Alle records krijgen een (oplopend) uniek ID toegekend. Elk record wordt met het ID tijdelijk in de database opgeslagen op de volgende manier:

ID	Record
----	--------

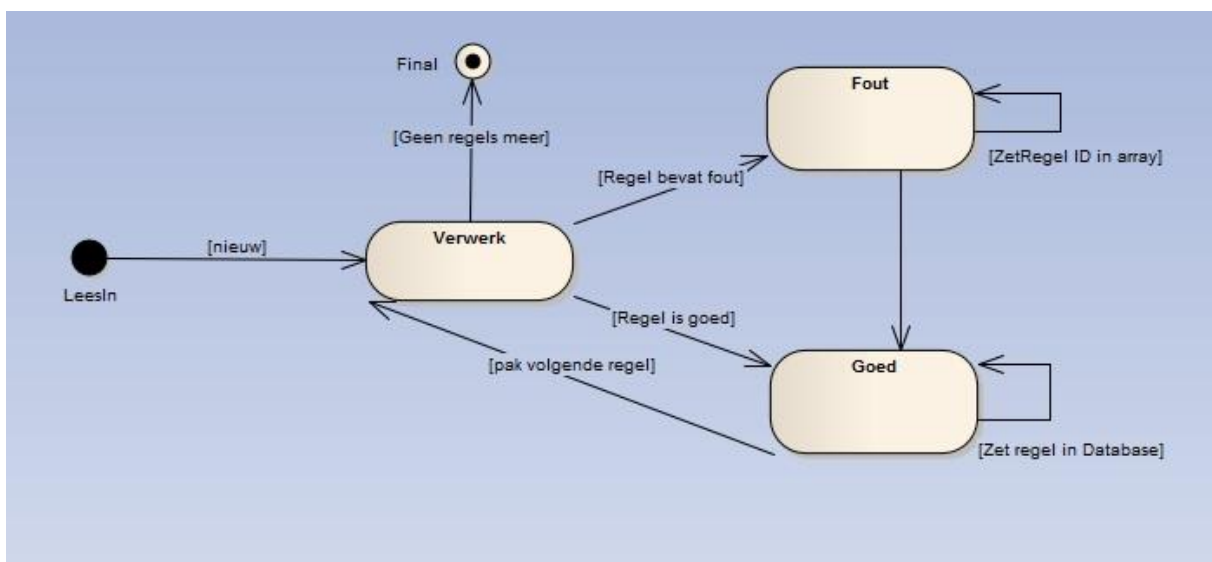
Tijdens het inlezen wordt tevens gecheckt of de datum/tijd chronologisch oploopt. Elke regel (record) begint met de volgende reeks getallen: 32009122306. Het eerste getal is hierin niet belangrijk echter het 2^e t/m het 5^e getal betreft het jaar waarna de maand, dag en het uur volgen. Door deze reeks als een getal op te slaan kunnen we de datum/tijd op een simpele manier controleren. Aangezien de datum/tijd chronologisch moet oplopen moet het volgende getal altijd steeds groter zijn.

Voorbeeld.

Datum is 23-12-2009 06 uur het getal is dan 2009122306

Datum is 23-12-2009 12 uur het getal is dan 2009122312

Indien het getal niet groter is dan het voorgaande getal wordt het ID dat wordt meegegeven aan het record in de database ook opgeslagen in een array. **Figuur 17** geeft in een toestandsdiagram het proces van inlezen schematisch weer.



Figuur 17 Toestandsdiagram Inlezen journaaldiskette

Wanneer alle records zijn ingelezen bevat het array alle ID's van de records die mogelijk een fout bevatten. De gebruiker kan nu per fout klikken op bewerken. Een nieuw browserscherm wordt geopend waarin het record met het foutieve ID wordt



Afstudeerscriptie

geladen. Tevens worden boven dit record de records met ID-1 en ID-2 en onder het record de records met ID+1 en ID+2 geladen. De gebruiker heeft nu de mogelijkheid om de fout op te zoeken in het record en te verbeteren, Aangezien de fout ook een record voor of een record na het door de computer aangegeven record kan zitten worden deze ook weergegeven. Indien het record verwijderd moet worden kan dit ook gedaan worden door de gebruiker. In dit geval wordt het record vervangen door "DELETE"

Zodra de gebruiker denkt alle fouten te hebben verbeterd drukt men op check opnieuw. Hiermee worden alle records uit de database weer in een .txt bestand opgeslagen. Alle record waarin "DELETE" staat worden overgeslagen. Hiermee wordt een bestand gemaakt wat gelijk is aan het bestand wat normaal wordt ingelezen. Indien dit bestand klaar is wordt dit gelijk weer doorgegeven aan de functie die de gegevens checkt. Waarna in feite dus het proces opnieuw begint.

Indien het proces klaar is blijft het bestand op de server staan. Een paar keer per jaar worden alle bestanden verzameld en door een al bestaand programma tot een .txt file per schip verwerkt. De gebruiker hoeft om al de bestanden dan te downloaden alleen op "download journals" te drukken waarna alle .txt bestanden in een .zip ter download wordt aangeboden. De directory waar deze bestanden stonden op de server wordt tevens leeg gemaakt. Het .zip bestand blijft wel in een aparte directory op de server opgeslagen zodat deze altijd te downloaden is.

10.3 DMG

De gegevens kunnen op twee manieren naar het KNMI gezonden worden: direct via mail vanaf het schip of vanaf het schip via satelliet. De gegevens die binnenkomen via de mail zijn de gegevens die in het VOS bestand vermeld staan. Deze gegevens zijn gratis voor het KNMI. Het betreft hier namelijk schepen die een internetverbinding aan boord hebben en het KNMI hoeft hier niks voor te betalen.

Gegevens die via satelliet verzonden worden naar het KNMI moeten betaald worden door het KNMI. Deze gegevens worden via ontvangststation Burum verzonden. Dit zijn de nota bestanden.

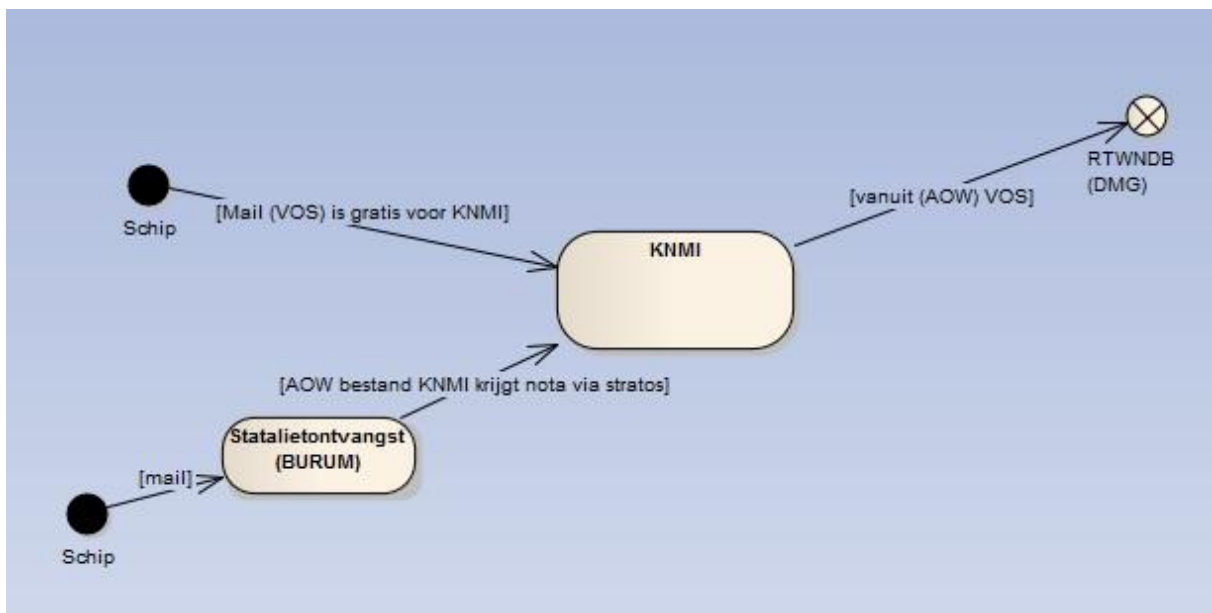
Al deze gegevens komen binnen bij het KNMI. Vervolgens wordt door het KNMI één bestand gemaakt waarin alle binnenkomen gegevens staan. Dit is het RTWNDB bestand. In feite kan men dus zeggen dat wanneer men de gegevens van VOS en de nota's samenvoegt het RTWNDB bestand krijgt.

Dit is niet helemaal waar aangezien het door bepaalde software mogelijk is dat een bericht wel bij Burum aankomt en niet verzonden wordt naar het KNMI. Het bericht staat dan wel in de nota maar aangezien het niet is aangekomen bij het KNMI staat het niet in RTWNDB. Het RTWNDB bestand is dus niet per definitie gelijk aan het VOS bestand + het nota bestand. Echter is door deze gegevens te vergelijken het dus wel mogelijk te kijken waar gegevens verloren zijn gegaan.

DMG omvat het bestand van VOS en het RTWNDB bestand. Deze twee bestanden worden door het KNMI samengesteld en moeten in het systeem verwerkt worden. **Figuur 18** geeft een schematische weergave van het geheel.



Afstudeerscriptie



Figuur 18 situatie DMG

10.3.1 RTWNDB

Het RTWNDB bestand bestaat uit een Excel bestand waarin in aparte kolommen, Roepletters, CCCC, Maand, Aantal en vertraging worden opgeslagen. Het inlezen van dit bestand gebeurt via een .csv bestand. De gebruiker wordt geacht het Excel bestand om te zetten naar een ';' gescheiden .csv bestand. Hierna is het mogelijk het bestand in te lezen. De process() functie gaat weer elke regel langs. In dit geval wordt er op elke regel gezocht naar de ';' . Vanaf positie 0 tot de positie van de eerste ';' hebben we te maken met de Roepletters. Vervolgens verwijderen we alle karakters t/m de eerste ';' en wordt er weer opzoek gegaan naar een ';' . Dit moet in het geval van een RTWNDB bestand dan de CCCC waarde zijn. Dit proces herhaalt zich met maand en aantal. Wat na aantal overblijft, is dan automatisch de vertraging. Vervolgens worden deze waarde via een Query in de database tabel RTWNDB geplaatst.

Het is van belang dat het bestand wat ingelezen wordt een vaste bestandsnaam bevat. Deze bestandsnaam moet als volgt opgebouwd zijn: Rtwndb_Sea_maand_20091105. Hierbij wordt 20091105 steeds veranderd door de datum die voor het bestand van toepassing is. Hierdoor kan de functie process() achterhalen voor welke maand de gegevens ingelezen worden. De functie process() verwijdert alle gegevens met deze maand die al in de database staan eerst voordat de nieuwe data ingelezen wordt. Ook kan deze functie kijken of de datum die in het bestand staat klopt met de datum in de bestandsnaam. Indien dit op een regel niet zo is geeft het programma een melding over de betreffende regel. De gebruiker kan dan kiezen het bestand te bewerken en opnieuw in te lezen of niks te doen waarmee de desbetreffende regel niet in de database is opgenomen.

Het RTWNDB bestand wordt tevens als back-up op de server opgeslagen.



Afstudeerscriptie

10.3.2 VOS

Het VOS bestand bestaat net als het RTWNDB bestand uit een Excel bestand met aparte kolommen. In het geval van het VOS bestand bevatten deze kolommen achtereenvolgens: Inmarsat_C, roepletters, extra, maand, aantal en vertraging. Deze gegevens worden via een zelfde principe ingelezen door middel van een ';' gescheiden .csv bestand. De gegevens worden in de VOS tabel in de database geplaatst.

Voor het VOS bestand geldt net als voor het RTWNDB bestand dat de bestandsnaam volgens een vaste lay-out opgebouwd moet worden. In dit geval namelijk: Vos_Sea_maand_20091005. Hiermee kan net als bij RTWNDB gekeken worden of de datum klopt en alles wat al van deze maand in de database staat eerst verwijderd worden.

Het VOS bestand wordt tevens als back-up op de server opgeslagen.

10.4 Evaluatie

Een evaluatie met de toekomstige gebruikers toont aan dat de ontwikkelingen van de software nog steeds in de richting die gewenst is plaatsvinden.

Men heeft echter nog wel een paar extra zaken die men eventueel in het programma zou willen zien. De meeste van deze ideeën zijn niet in de nog beschikbare tijd te realiseren. Echter één aanvullende eis is goed toe te voegen in de nog te ontwikkelen software. Het betreft hier een SQL builder. Het idee is dat de gebruiker zelf ook query's kan uitvoeren op de database en zo tabellen naar voren kan halen. Dit idee kan mogelijk ingepast worden met de ontwikkeling van de functie die rapporten moet maken.

Vooruitlopend op de volgende fase zijn nog wel een paar vragen die nadere uitleg vereisen. Deze vragen konden beantwoord worden.

Herkenning Duitse schepen

Er is een aparte lijst beschikbaar met hierin de Duitse schepen. Dit is een Excel bestand waarin de scheepsnamen samen met het callsign, het imo_number en inmarsat_c nummer. Dit bestand wordt om de zoveel tijd naar het KNMI opgestuurd. In het bestand staan alle Duitse schepen. Door een extra tabel aan de database toe te voegen waarin de Duitse schepen zijn opgeslagen. Is het ook mogelijk de nota's voor Duitsland te maken.



Afstudeerscriptie

11. Construction fase rapporten maken

Construction fase iteratie drie concentreert zich op het inlezen en verwerken van de nota bestanden. Tevens worden van alle data die in de database staat rapporten gemaakt. De wens van een SQL builder kwam tevens naar voren bij de gebruikers en is mogelijkserwijs in dit tijdsbestek nog toe te passen. Hierdoor heeft deze fase wel iets langer geduurd dan gepland.

11.1 Stratos nota bestand inlezen.

Het stratos nota bestand bevat alle communicatie die via een satelliet naar het KNMI is gestuurd. Uiteraard moet er voor het gebruik van een satelliet betaald worden. Het KNMI betaald aan de schepen de kosten voor deze communicatie.

Dit nota bestand is een .txt bestand waarbij de waardes weer op vaste posities staan. Er kan verschillende informatie op een regel staan. De eerste twee waardes op de regel bepalen welke informatie volgt. Zo bevat alles wat met 02 begint elke communicatie. Alles wat met 03 begint is een totaal hiervan per schip. Voor het programma is alleen alles met 02 belangrijk. De gegevens met 03 kunnen we naderhand via een query in de applicatie weer naar voren halen.

Aangezien er veel zaken in aparte kolommen moet worden opgeslagen, is er een XML bestand waarin de beginpositie en lengte staat.

De code is zo opgebouwd dat indien er onverwachts meer data opgeslagen moet worden dit mogelijk is zonder ingrijpende veranderingen in de code. In principe zou het toevoegen van een kolom in de database en het opgeven van de positie en lengte in de XML voldoende moeten zijn. **Code fragment 6** geeft dit nog is weer.

```
if($waarde[1] == "02")
{
    $l = 8;
    $sql_query = "INSERT INTO `Nota` VALUES ('" . $waarde[1] .
    "','" . $waarde[2] . "','" . $waarde[5] . "-" . $waarde[4] .
    "-" . $waarde[3] . "','" . $waarde[6] . ":" . $waarde[7] . "''";

    while($l != $id)
    {
        $sql_query = $sql_query . "','" . $waarde[$l] . "''";
        $l++;
    }
    $sql_query = $sql_query . "));";
    mysql_query($sql_query);
}
```

Code fragment 6 Nota naar database

Door middel van de while lus is het mogelijk de query via de XML uit te breiden met kolommen. Bij het inlezen van de XML wordt het aantal elementen wat ingelezen wordt geteld, waardoor men tevens weet hoeveel kolommen er zijn. Zaak is hierbij wel dat in de XML dezelfde volgorde wordt aangehouden als bij de database omdat anders de verkeerde data in de verkeerde kolom terecht komt.



Afstudeerscriptie

De reden dat de eerste 7 elementen niet zijn opgenomen in de while lus komt doordat hier de datum en tijd worden opgeslagen deze wordt in Nederlandse notatie opgeslagen terwijl hij in Amerikaanse notatie in het bestand staat.

11.2 Rapporten

Zeer belangrijk zijn de rapporten die gemaakt moeten worden. Men wil graag rapporten hebben van verschillende zaken. Er is gekozen om behalve de rapporten in .doc beschikbaar te stellen deze ook in .odt beschikbaar te maken. Dit is gedaan omdat de overheid zoveel mogelijk over wil gaan op opensource en daarmee vanzelfsprekend van programma's als Microsoft Office af wil. Echter aangezien Microsoft office Word vrij algemeen gebruikt wordt en het niet ondenkbaar is dat de gebruiker thuis alleen dit pakket heeft blijft .doc ook beschikbaar.

Een rapport is opgebouwd uit het resultaat van één of meerdere query's. Alle query's zijn apart opgeslagen op de server als .sql bestanden en dus afzonderlijk van de code aan te passen of uit te breiden. De functie `getQuery($name)` haalt de gewenste query van de server op en geeft deze terug als een string.

Het ophalen van de gegevens uit de database was in eerste instantie toegeschreven aan de functie `getData($sql_query)` in de klasse `rapport`. Echter aan het eind van iteratie twee is de wens uitgesproken ook zelf query's te kunnen maken en uitvoeren in het systeem. Hierdoor is een aparte klasse `SQLBuilder` ontstaan waarnaar deze functie is verplaatst. De werking van deze functie wordt daarom uitgelegd bij deze klasse.

Zodra de query uitgevoerd is wordt de data in een bestand geplaatst (.doc of .odt). Hiervoor wordt simpel een nieuw document aangemaakt met de gewenste extensie en de data in HTML weggeschreven. Hierdoor ontstaat een document met de gewenste gegevens wat de gebruiker kan opslaan en netjes als a4 kan uitprinten.

In **bijlage 7** is te vinden welke rapporten gemaakt kunnen worden.

11.3 SQL builder

De SQL builder is ontstaan na een aanvullende eis van de gebruiker. De basisfunctionaliteit van deze klasse was vrij eenvoudig te realiseren. Zou eerst de klasse `rapport` een functie krijgen die een query verstuurt en een tabel van de data terug krijgt. Nu is deze functie bij de SQL builder geplaatst.

In de SQL builder is het mogelijk dat de gebruiker zelf een query maakt en hiervan het resultaat te zien krijgt. De gebruiker voert een query naar wens in en welke vervolgens naar de functie `Process()` gestuurd wordt. Deze functie slaat de query tijdelijk op in `TEMP.sql` zodat de query nog snel achterhaalt kan worden. De Query wordt vervolgens verzonden naar de functie `Output($sql_query)`. Deze functie voert de query uit en maakt hiervan netjes een tabel. De code in **code fragment 7** laat dit zien.

```
public function Output($sql_query)
{
    $result = mysql_query($sql_query);
```




Afstudeerscriptie

```
$return = "<table border= 2><tr>";

for ($i = 0; $i < mysql_num_fields($result); ++$i)
{
    $field = mysql_field_name($result, $i);
    $return .= "<td style='background-color:
#ccc;'>".$field."</td>";
}
$return .= "</tr>";

while ($row = mysql_fetch_row($result))
{
    $return .= "<tr>";
    for ($i = 0; $i < mysql_num_fields($result); ++$i)
    {
        $return .= "<td>".$row[$i]."</td>";
    }
    $return .= "</tr>";
}
$return .= "</table>";

return $return;
}
```

Code fragment 7 Query omzetten naar table

In de for lus wordt de kop van de tabel gemaakt. Door middel van `mysql_field_name($result, $i)` is te achterhalen hoe elke kolom moet heten. In de while lus wordt vervolgens telkens een rij toegevoegd aan de tabel. Als laatste wordt het geheel teruggezonden naar de desgewenst functie.

Indien het resultaat van de query niet is wat men wenst keert men terug naar de query builder. Deze builder kijkt eerst in het document TEMP.sql en laat de query die hierin is opgeslagen zien. Dit is de laatste query die is uitgevoerd. Door middel van een try en catch wordt gekeken of dit bestand aanwezig is **code fragment 8** laat dit zien.

```
try
{
    if ($handle)
    {
        $Query = fread($handle, filesize($file));
        fclose($handle);
    }
    else
    {
        $Query = "";
        throw new Exception('There is no old query available');
    }
}
catch (Exception $e)
{
    $return .= "<br><font color='red'>".$e->getMessage()."</font><br>";
}
```

Code fragment 8 Query laden



Afstudeerscriptie

Hierdoor hoeft de gebruiker de laatste query niet volledig opnieuw in te typen wat veel werk bespaard. Kanttekening hierbij is natuurlijk wel dat een andere gebruiker in de tussentijd geen query gemaakt heeft omdat anders deze query in TEMP.sql staat. Echter wordt de applicatie niet door veel mensen gebruikt en werken deze mensen samen dan is de kans dat dit probleem optreedt zo goed als 0% is.

Indien een query ontwikkeld is die men in een later tijdstip nogmaals wil uitvoeren is het mogelijk deze op te slaan. De laatste query staat nog in TEMP.sql en wordt dus weer getoond. Door nu een naam op te geven wordt de query onder naam.sql opgeslagen in de database. Tevens wordt deze naam toegevoegd aan het XML bestand wat al deze namen bevat. Via een dropdown box kunnen opgeslagen query's dan weer geladen worden en opnieuw worden uitgevoerd.

Uiteraard is het ook mogelijk een query te verwijderen het bestand naam.sql wordt dan verwijderd van de server en het XML bestand wordt geüpdate.

11.4 Lijsten

Ook is de wens uitgesproken om van elke tabel eenvoudig alle inhoud naar voren te kunnen halen. Door middel van de SQL builder is dit nu eenvoudig te realiseren. Een klasse List regelt hiervoor de uitvoer. Aan de functie getList(\$list) wordt de tabel waarvan de data opgehaald moet worden meegegeven. In deze functie wordt op de in **code fragment 9** getoonde manier de query opgebouwd.

```
"SELECT * FROM `".$list."`"
```

Code fragment 9 Query Lijsten

Deze Query wordt verzonden naar de Query Builder welke een tabel met de inhoud van de database tabel teruggeeft.

In **bijlage 8** is te vinden welke lijsten getoond kunnen worden.

11.5 Evaluatie

De gebruikers zijn zeer tevreden met de rapporten die gerealiseerd zijn. Deze zijn zeer nuttig voor evaluatie en om mee te nemen wanneer een schip bezocht wordt. Men is ook blij dat de mogelijkheid van een SQL builder aanwezig is. Hierdoor kunnen zelf makkelijk query's gemaakt worden. De nota's worden ook goed ingelezen. Er zijn geen aanvullende eisen waardoor zonder wijzigingen doorgegaan kan worden met de volgende fase.



Afstudeerscriptie

12. Construction fase overige zaken

Construction fase iteratie vier concentreert zich op het afmaken van de should haves. Minder belangrijke zaken uit de should haves zijn in deze iteratie gepland. Dit is gedaan zodat bij eventuele uitloop de minst belangrijke zaken af konden vallen.

12.1 Statistieken

De behoefte naar statistieken over de laatste bezoekdatum aan een schip, is tijdens het achterhalen van de eisen naar voren gekomen. De klasse statistiek heeft een functie ter beschikking welke een formulier aan de gebruiker toont. In dit formulier kan gekozen worden tussen welke datums het laatste bezoek moet liggen. Zodra de gebruiker de datums heeft ingevoerd wordt de functie process() aangeroepen. Deze functie vult de datums op de juiste plaatsen in, in de query getoond in **code fragment 10**.

```
SELECT visit.IMO_number, old_names.Name, visit.Date
FROM visit
LEFT JOIN old_names
ON old_names.IMO_number = visit.IMO_number
WHERE old_names.End_date = '0'
AND visit.Date BETWEEN #from AND #untilwith
```

Code fragment 10 Query statistieken

De query wordt vervolgens weer naar de SQLbuilder gezonden die een table terugstuurt. Deze tabel wordt getoond aan de gebruiker.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

13. Transition fase

Tijdens de transition fase vindt de overdracht van de verantwoordelijkheden plaats. Het product wordt hierbij over gedragen aan de programmeur die het later gaat onderhouden. Tevens wordt het resultaat gepresenteerd aan de gebruikers.

13.1 Data oude systeem naar nieuwe systeem

Aangezien er al een oude database bestaat, moet deze data in de nieuwe database beschikbaar komen. Hiervoor moet de data uit de oude database naar de nieuwe getransporteerd worden. Aangezien de oude database anders is opgebouwd dan de nieuwe database vergt dit enige tijd. Het omzetten van de maatschappijen met hun contactpersonen kon vrij eenvoudig gebeuren. Dit is gedaan doormiddel van een paar query's op de oude database uit te voeren waarna een bestand gemaakt kon worden wat geïmporteerd kon worden in de nieuwe database.

Het grootste probleem ligt echter bij de overige data. Deze kan helaas op dit moment niet direct geïmporteerd worden in het nieuwe programma. Het oude programma maakt gebruik van de scheepsnaam als primaire sleutel. Echter het nieuwe programma maakt gebruik van het IMO-nummer als primaire sleutel. In het oude programma zijn echter bij veel schepen geen IMO-nummer toegevoegd. Voor het nieuwe programma was voor een andere primaire sleutel gekozen omdat de scheepsnaam kan veranderen en niet uniek is. Dit zorgde daarom voor problemen bij dat programma. Echter doordat niet over de nieuwe, wel, unieke primaire sleutel aanwezig is kan deze data niet zomaar geïmporteerd worden. Eerst moet bij elk schip het IMO-nummer opgezocht worden. Deze data is daarom nog niet geïmporteerd in het programma.

13.2 Testen

Aan het begin van de transition fase is het volledige systeem nog aan wat testen onderworpen. Bij deze testen is gekeken naar zaken als data invoer en output. Ook is gekeken of de software op verschillende browsers draait. Hierbij zijn Internet Explorer 7, Firefox 3.6 en Google Chrome gebruikt. Voor een uitgebreid overzicht van de tests wordt verwezen naar **externe bijlage 6**.

Aangeraden is om de komende maanden het oude systeem nog naast het nieuwe systeem te draaien. Eventuele fouten en verschillen tussen de twee systemen kunnen op deze manier nog aan het licht komen zonder veel schade aan te richten.

13.3 Overdracht naar programmeur

Het programma is overgedragen naar de programmeur/opdrachtgever die het programma in de toekomst gaat onderhouden en uitbreiden. Het programma is op de pc van de programmeur geïnstalleerd. Tevens is uitleg gegeven over de werking van het systeem. Ter verduidelijking is ook nog een ontwikkelaars handleiding overhandigd. Deze handleiding (**externe bijlage 5**) verduidelijkt de werking van het programma. Hierdoor is het mogelijk om op een later tijdstip de precieze werking van een functie nogmaals op te zoeken.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

13.4 Overdracht naar gebruikers

Het programma is ook gepresenteerd aan de gebruikers. Dit is gedaan doormiddel van een presentatie die het eindresultaat aan de gebruiker toont. Tevens wordt hiermee de werking van het systeem getoond zodat de gebruikers hiermee snel aan de slag kunnen. Ter verduidelijking van de werking van het systeem is ook nog een gebruikershandleiding opgeleverd. Deze handleiding (**externe bijlage 4**) verduidelijkt de werking van het systeem voor de gebruiker.



Afstudeerscriptie

14. Algemene evaluatie

In dit hoofdstuk volgt een evaluatie van de hele afstudeerperiode. Als eerste wordt in **hoofdstuk 14.1** de opdracht besproken. Hierna wordt het proces geëvalueerd (**hoofdstuk 14.2**). Om vervolgens te kijken of de vooraf geselecteerde competenties behaald zijn.

14.1 Opdracht

In deze paragraaf wordt de opdracht per fase geëvalueerd. Hierbij wordt kort gekeken hoe deze fase is verlopen.

14.1.1 Inception fase

Tijdens de start van de opdracht moest eerst bekeken worden wat de opdracht precies inhield en hoe het systeem precies werkt. Dit heeft ongeveer een week gekost waarna een duidelijker beeld gevormd kon worden van het systeem. Een aantal bezoeken aan schepen waar waarnemingen worden gedaan heeft hierbij geholpen het hele proces duidelijk te krijgen van invoer tot verwerking.

14.1.2 Elaboration fase

Het uitzoeken van de functionele en niet functionele eisen bleek een hele klus te zijn. Vaak krijgt men maar halve antwoorden omdat men zelf dagelijks met de materie werkt en dingen vanzelfsprekend beschouwt. Het achterhalen van de eisen en het afstemmen van de eisen tussen de verschillende gebruikers heeft daarom ook veel tijd gekost.

Hierbij moest uiteraard ook gekeken worden of de eisen niet te persoonlijk waren en alleen 'leuk' voor de gebruiker maar niet noodzakelijk voor het systeem. Het uitzoeken vergde tevens nog enige tijd.

Nadat de eisen vastgesteld waren moesten deze ook nog ingedeeld worden met de MOSCOW methode. Aangezien elke gebruiker zijn eigen gedeelte van het programma het belangrijks vindt wil men dit graag in de must have's hebben. Het zoeken naar een goede manier op de eisen in te delen vergde enige tijd, maar uiteindelijk is een logische indeling gevonden waarmee alle toekomstige gebruikers tevreden waren.

Het maken van een eerste ontwerp van de database was niet al te makkelijk. Een ER diagram kon na een paar dagen gerealiseerd worden. Maar de exacte inhoud van elke tabel vergde wel een groot onderzoek. Grote delen hiervan waren al onderzocht bij het achterhalen van de eisen. Echter wat voor elk bestand opgeslagen moest worden was nog vrij onduidelijk.

Het ontwikkelen van een klassendiagram kostte enige tijd. Mede doordat gekeken moest worden hoe alle gegevens geordend verwerkt konden worden. Aangezien het om zeer veel verschillende data gaat moest hiervoor een goede oplossing komen die tevens makkelijk uitbreidbaar was. Dit is goed gelukt.

14.1.3 Construction fase 1

De ontwikkeling van deze fase ging vrij snel. Echter vergde het wel wat meer tijd dan gedacht. Mede doordat er nog al veel verschillende klassen gemaakt moesten worden



Afstudeerscriptie

met verschillende formulieren. Er moest hierbij telkens goed gekeken worden wat op welke manier opgeslagen moest worden. Naar aanleiding van het gemaakte ontwerp was het realiseren van deze code in verhouding tot de rest niet heel moeilijk maar wel veel werk. Dit kwam vooral door de hoeveelheid data die verwerkt moest worden. Het eindresultaat voldeed aan de verwachtingen van de opdrachtgevers.

14.1.4 Construction fase 2

Het kostte veel tijd om de precieze bestanden te krijgen die ingelezen moesten worden. Vervolgens moest ook nog gekeken worden waar wat stond opgeslagen in het bestand en hoe dit naar de database gebracht kon worden. Elk bestand is anders opgebouwd waardoor niet makkelijk voor een oplossing van inlezen gekozen kon worden. Men was bereid om Excel bestanden om te zetten in .csv om deze in te kunnen lezen in het programma.

14.1.5 Construction fase 3

Voordat deze fase ontwikkeld kon worden moest nog enig onderzoek plaatsvinden aan de nota's. Er bestond nog veel onduidelijkheid over hoe dit precies werkt. En wat de RTWNCB en VOS voor invloed hadden bij de nota's. Hiervoor zijn nog enige aanvullende gesprekken gevoerd. Tevens was er nog documentatie van stratos beschikbaar. Dit is het bedrijf dat de nota bestanden opstuurt. In deze documentatie was te vinden hoe het bestand precies is opgebouwd en waar alles precies voor dient. Na het doorlezen van deze documentatie kon begonnen worden met de realisatie van deze fase. Door deze gesprekken ontstond gelijk meer duidelijkheid over hoe de rapporten die betrekking hadden op de nota's precies moesten worden opgebouwd.

14.1.6 Construction fase 4

Deze fase verliep voorspoedig. Er zijn geen problemen opgetreden en de iteraties kon in de beschikbare tijd uitgevoerd worden. Deze iteratie betrof ook meer het afmaken van lopende zaken. Hierbij hoefde niet veel nieuwe klassen gemaakt te worden en kon veelal gebruik gemaakt worden van de al geïmplementeerde code.

14.1.7 Transition fase

Tijdens deze fase is het product overgedragen aan de opdrachtgever. Enigste probleem wat optrad was dat de data niet makkelijk naar het nieuwe systeem overgezet kon worden, door het ontbreken van het IMO-nummer. Dit moet dus eerst opgelost worden voordat het nieuwe programma volledig gevuld kan worden met de data. Dit wordt de komende tijd gedaan waarna het systeem operationeel is.

14.2 Proces evaluatie

In deze paragraaf wordt het doorlopen proces geëvalueerd.

- *Ontwikkelmethode*

De gekozen ontwikkelmethode RUP bleek goed geschikt voor dit project. Mede door de grote elaboration fase kon goed achterhaald worden wat de eisen van de gebruiker precies waren. Dit bleek zeer veel tijd te kosten. Dankzij de milestones en de opdeling van de construction fase in verschillende iteraties was het mogelijk



Afstudeerscriptie

vaak te evalueren met de toekomstige gebruikers. Hierdoor is een product ontstaan wat voldoet aan de eisen die gesteld zijn.

- *Plan van aanpak*
Het plan van aanpak was een handig eerste hulpmiddel om duidelijk te maken op welke manier zaken gedaan gingen worden. Door een globale planning kon tevens ingeschat worden wat er gerealiseerd kon worden in de tijd die beschikbaar was voor het project.
- *Software requirements specification document*
Dit document zorgde ervoor dat de eisen van de toekomstige gebruikers goed opgeschreven konden worden. Een terugkoppeling met de toekomstige gebruikers zorgde er tevens voor dat functionele eisen juist waren geïnterpreteerd. Het maken van dit document kostte veel tijd maar gaf veel inzicht in de zaken die ontwikkeld moesten worden. De opdeling met de MOSCOW methode zorgde er mede voor dat de belangrijkste zaken eerst ontwikkeld werden. Door de MOSCOW methode was het ook makkelijker de zaken goed op te delen in iteraties.
- *Ontwerpdocument*
Het ontwerpdocument heeft ervoor gezorgd dat de te ontwikkelen software via een goede en makkelijk uitbreidbare manier te realiseren was. Het maken van het ontwerp kostte uiteraard enige tijd maar hierdoor kon wel makkelijker overgegaan worden op het programmeren.
- *Ontwikkeling*
De ontwikkeling verliep over het algemeen goed. Hier en daar waren nog wat onduidelijkheden over waar iets precies voor diende en hoe zaken geïnterpreteerd moesten worden. Hier kon na overleg vaak wel mee verder gegaan worden. Over het algemeen is er geen probleem ontstaan met de planning. Sommige zaken liepen uit maar anderen gingen sneller dan gepland. Hierdoor bleef de uiteindelijke duur van de ontwikkeling hetzelfde. Doormiddel van de verschillende iteraties was het mogelijk tussentijds kleine wijzigingen toe te voegen. Hierdoor kon mede nog de mogelijkheid toegevoegd worden om de gebruiker zelf query's te laten maken.

14.3 competenties

Aan het begin van deze afstudeerperiode zijn een paar competenties vastgesteld. Hieronder worden deze geëvalueerd:

C11 Ontwerpen van een mens-machine interface

Er is tijdens de ontwikkeling rekening gehouden met de doelgroep. Hierdoor is ook een SQL builder ontwikkeld die de gebruiker in staat stelt zelf query's te ontwikkelen en uit te voeren. Dit past binnen de doelgroep waarvoor het programma is opgedeeld. Tevens is de invoer van velden logisch en eenduidig waardoor geen ook mensen met minder kennis snel met de software aan de gang kunnen gaan.



Afstudeerscriptie

D16 Realiseren van software.

De code die gerealiseerd is, is netjes opgebouwd en tevens op allerlei fronten eenvoudig uitbreidbaar. De software voldoet aan de eisen die vooraf gesteld zijn en voldoet aan de eisen van de gebruikers. Tevens is de software goed gedocumenteerd.

H5 Professioneel werken: Zelfstandig werken.

Een zelfstandige werkwijze is aangetoond tijdens de duur van het project. Er is zelfstandig een plan van aanpak gerealiseerd. Tevens zijn er gesprekken met de toekomstige gebruiker gevoerd waarna zelfstandig de functionele eisen uit deze gesprekken zijn opgesteld.

Ook is zelfstandig een indeling gemaakt via de MOSCOW methode. Voor de rest is de ontwikkeling zelfstandig uitgevoerd. En zijn er gesprekken ter evaluatie van de tussentijdse 'oplevermomenten' geweest.

H6 Professioneel werken: Resultaatgericht werken.

Er is gewerkt richting een goed eindproduct. Vanaf het begin was het doel een product op te leveren wat minstens voldoet aan de must en should have's die waren ingedeeld via de MOSCOW methode. Dit is gerealiseerd.

H7 Professioneel werken: Methodisch werken.

Er is gewerkt via de methode RUP. Deze methode zorgt er mede voor dat het eindproduct zo dicht mogelijk bij de wensen van de gebruiker ligt als nu het geval is. Door methodisch werken is het mede gelukt de belangrijkste zaken van het systeem als eerste te ontwikkelen.

J11 Je plek vinden in een (inter)nationale organisatie.

Er is snel een plek gevonden binnen de organisatie. Mede hierdoor was het mogelijk snel en goed de eisen van de toekomstige gebruikers naar voren te halen.



Afstudeerscriptie

15. Conclusie

In dit hoofdstuk wordt gekeken of de functionele en niet functionele eisen gerealiseerd zijn in het eindproduct. Indien de eis niet gerealiseerd is wordt een reden hiervoor gegeven.

15.1 Functionele eisen

In deze paragraaf worden de functionele eisen zoals gedefinieerd in **externe bijlage 2** geëvalueerd.

1) *Nieuwe schepen moeten toegevoegd kunnen worden.*

Het is in de software mogelijk om schepen toe te voegen aan de database. Ingevoerde schepen kunnen tevens bekeken worden, verwijderd worden en worden gewijzigd.

2) *Nieuwe waarnemers moeten toegevoegd kunnen worden.*

Het is in de software mogelijk om waarnemers toe te voegen aan de database. Ingevoerde waarnemers kunnen tevens bekeken worden, verwijderd worden en worden gewijzigd.

3) *Er moet bijgehouden worden hoeveel waarnemingen een waarnemer doet.*

Het aantal waarnemingen van een waarnemer kan ingevoerd worden.

4) *De journaals moeten ingevuld kunnen worden.*

Journaals kunnen toegevoegd worden. Het programma genereert hierbij automatisch een journaalnummer. Bij het journaal wordt het aantal waarnemingen en de periode van de waarnemingen opgeslagen.

5) *Nieuwe maatschappijen moeten ingevoerd kunnen worden.*

Het is mogelijk nieuwe maatschappijen aan het systeem toe te voegen. Maatschappijen kunnen ook gewijzigd, bekeken en verwijderd worden. Bij elke maatschappij kan een onbeperkt aantal contactpersonen toegevoegd worden.

6) *De versie en uitgiftedatum van de software op schepen moet bijgehouden worden.*

Bij elk schip kan ingevoerd worden welke software aan boord is en wat de plaatsingsdatum hiervan is. De soorten software kunnen apart van de schepen toegevoegd en gewijzigd worden. Een fout in een software naam kan dus op een plaats voor alle schepen tegelijk veranderd worden.

7) *Verzendingen naar schepen moeten bijgehouden worden.*

Verzendingen naar de schepen kunnen bijgehouden worden. Tevens kunnen bestaande verzendingen gewijzigd, bekeken en verwijderd worden.

8) *Verschillende soorten instrumenten moeten bijgehouden worden.*

Instrumenten aan boord kunnen bijgehouden worden. De instrument soorten worden apart bijgehouden en kunnen zo dus makkelijk gewijzigd worden.

9) *De plaatsingsdatum van instrumenten op het schip moet bijgehouden worden.*

De plaatsingsdatum van de instrumenten kan ook bijgehouden worden.



Afstudeerscriptie

10) Buitenlandse schepen moeten ingevoerd kunnen worden.

Buitenlandse schepen kunnen ingevoerd, bekeken, gewijzigd en verwijderd worden. Zoals gewenst was worden ze fysiek gescheiden opgeslagen van de Nederlandse schepen.

11) Bezoeken van Nederlandse schepen moeten kunnen worden geregistreerd.

Bezoeken aan Nederlandse schepen kunnen geregistreerd, bekeken, gewijzigd en verwijderd worden.

12) Bezoeken aan buitenlandse schepen moeten worden geregistreerd.

Bezoeken aan buitenlandse schepen kunnen ook opgeslagen worden. Deze worden apart van de Nederlandse bezoeken in de database opgeslagen omdat er andere informatie moet worden opgeslagen. De bezoeken kunnen ook bekeken, gewijzigd en verwijderd worden.

13) De ingelezen data moet gecontroleerd worden op fouten.

Data die wordt ingelezen wordt gecontroleerd op fouten. Zodra er een fout in de data staat wordt dit kenbaar gemaakt aan de gebruiker. De gebruiker krijgt dan de kans de data te herstellen en opnieuw in te lezen.

14) Fouten die gevonden worden moeten door de gebruiker in het programma hersteld kunnen worden.

De gebruiker kan de fouten die optreden tijdens het inlezen van de journaaldiskettes in het programma herstellen. De gebruiker heeft de optie de regel aan te passen of zelfs volledig te verwijderen uit het programma.

15) De data moet in verschillende rapporten gezet worden en kunnen worden uitgeprint.

Er zijn verschillende rapporten beschikbaar waarin de data gepresenteerd wordt aan de gebruiker.

16) Verwerken MONTAB (Monthly table)

De MONTAB gegevens worden gecontroleerd en ingelezen in de database.

17) Verwerken Journaal diskette

De journaaldiskette kan ingelezen worden. Tevens wordt gecontroleerd op fouten en kan de gebruiker deze herstellen.

18) Verwerken Nota's

Nota bestanden kunnen worden ingelezen. Tijdens het inlezen worden deze bestanden gecontroleerd op fouten.

19) Rapporten over schepen die langer dan een bepaalde datum geleden bezocht zijn moeten uitgedraaid kunnen worden.

Er kunnen rapporten over de laatste bezoekdatum worden uitgeprint. Deze rapporten zijn te vinden onder statistieken.

20) De stratos gegevens moeten ingelezen worden waarna er nota's van gemaakt moeten worden.

De stratos bestanden kunnen worden ingelezen. Hierna kan doormiddel van rapporten hier een nota van gemaakt worden.



Afstudeerscriptie

21) *DMG bestanden moeten verwerkt kunnen worden.*

DMG bestanden kunnen worden ingelezen.

22) *Koppeling met buitenlands systeem.*

Er is geen koppeling met een buitenlands systeem gerealiseerd. Voor een onderzoek was geen tijd beschikbaar. Tevens was inzicht in de database van Frankrijk niet binnen de projectduur mogelijk.

15.2 niet functionele eisen

In deze paragraaf worden de niet functionele eisen zoals gedefinieerd in **externe bijlage 2** geëvalueerd.

1) *Het systeem moet stabiel zijn.*

Het systeem is stabiel en crasht niet zodra een onbekend bestand wordt ingelezen.

2) *Foutafhandeling*

Indien een bestand wordt ingelezen wat niet verwacht wordt, wordt dit aan de gebruiker kenbaar gemaakt. Foutieve informatie in de formulieren wordt ook aan de gebruiker kenbaar gemaakt.

3) *Het systeem is web based.*

Het systeem draait via een internet browser.

4) *Het systeem moet uiteindelijk op een LAMP omgeving gaan draaien.*

Het systeem is op een WAMP omgeving ontwikkeld en kan makkelijk naar een LAMP omgeving overgezet worden.

5) *Buitenlandse en Nederlandse schepen moeten gescheiden blijven.*

Nederlandse en buitenlandse schepen zijn fysiek in de database gescheiden.

6) *De software moet zelfstandig taken uitvoeren.*

Het systeem controleert zelf alle data die ingevoerd wordt op fouten. Indien een fout gevonden is wordt dit kenbaar gemaakt aan de gebruiker.

7) *De gegevens uit de bestaande database moeten overgezet worden.*

Sommige gegevens uit de bestaande database zoals de maatschappijen zijn al overgezet. Andere informatie volgt later omdat voor deze informatie niet overal het IMO-nummer bekend is. Het IMO-nummer is de nieuwe primaire sleutel en dus voor alle data vereist.

8) *Het systeem moet gebruik maken van PHP en MySQL*

Het systeem is ontwikkeld met PHP en MYSQL.

9) *Het programma is Engelstalig*

Het programma is volledig Engelstalig.

10) *Het systeem moet wereldwijd beschikbaar zijn.*

Het systeem kan toegevoegd worden aan de intranet omgeving van het KNMI. Hierdoor kan iedereen met een gebruikersnaam en wachtwoord overal ter wereld de applicatie bereiken.



Afstudeerscriptie

11) Performance

Het programma voldoet aan de performance eisen en reageert binnen 3 seconden op een actie van de gebruiker. Het inlezen van bestanden kan afhankelijk van de grootte langer dan 30 seconden duren. Echter vormt dit geen probleem.

12) Systeem grootte

Het systeem kan overweg met grootte hoeveelheden informatie. De database is ontworpen om makkelijk informatie te vinden.

15.3 Totale conclusie

Het product wat opgeleverd is voldoet aan de eisen van de opdrachtgever zoals deze gedefinieerd zijn in **externe bijlage 2**. Er is een product opgeleverd wat alle must en should have's zoals gedefinieerd in **externe bijlage 2** geïmplementeerd heeft.

Doormiddel van tussentijds overleg is een product opgeleverd wat dicht bij de eisen van de opdrachtgever ligt.

Het software requirements document en het ontwerpdocument hebben mede bijgedragen aan de goede implementatie van de eisen. Hierdoor konden de eisen goed vastgelegd en uitgewerkt worden.

De ontwikkelmethode RUP bleek de juiste methode voor dit project te zijn. Doormiddel van de inception, elaboration, construction en transition fase kon het project goed opgestart, doorlopen en afgerond worden. Doormiddel van verschillende iteraties in de constructionfase was het tevens nog mogelijk tussentijds wijzigingen toe te voegen. Zo is ook nog de SQL builder toegevoegd welke origineel niet gepland was.



Afstudeerscriptie

16. Aanbevelingen

De software die ontwikkeld is kan nog op een aantal punten uitgebreid worden. Een mogelijkheid is om de verschillende journaaldiskettes die ingelezen worden niet meer door een apart programma tot een bestand te laten maken, maar dit door de internetapplicatie te laten doen.

Een verdere ontwikkeling naar een internationaal systeem is eveneens een mogelijkheid om dieper op in te gaan. Momenteel blijkt er nog een lange weg te gaan naar een internationaal systeem. Maar systemen die met elkaar gegevens uitwisselen vormen een goede tussenoplossing. Een onderzoek naar de mogelijkheden hiervan is daarom zeker nuttig om aandacht aan te besteden.

Uiteraard kunnen er nog veel meer rapporten gemaakt worden dan nu het geval is. Met de rapporten kan informatie aan het oppervlak komen die in eerste instantie niet direct zichtbaar is.

Er kan tevens meer gedaan worden met de journaaldiskettes die ingelezen worden. Deze worden momenteel alleen gevalideerd en met de data zelf wordt weinig gedaan. Er zijn natuurlijk mogelijkheden te bedenken waarbij met deze data wel dingen gedaan kunnen worden.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

17. Begrippenlijst

CSS – **C**ascading **S**tyl**S**heet. Wordt gebruikt voor de opmaak van HTML.

Database - Een database, gegevensbank of databank is een digitaal opgeslagen archief, ingericht met het oog op flexibele raadpleging en gebruik.

HTML - Hypertext Markup Language (afgekort HTML) is een opmaaktaal voor de specificatie van documenten, voornamelijk bedoeld voor het World Wide Web.

Inheritance - In Nederlands overerving is een term die gebruikt wordt bij de methodiek van het object-georiënteerd programmeren. Het betekent dat een klasse variabelen (eigenschappen) en - afhankelijk van de definitie van de programmeertaal voor methodes - functies en procedures erft van een superklasse.

LAMP - Een acroniem voor een set van gratis vrije softwarepakketten, die vaak samen gebruikt worden om dynamische websites te draaien:

- Linux, het besturingssysteem.
- Apache, de webserver.
- MySQL, het database management systeem (of databaseserver).
- PHP, Perl en/of Python.

MoSCoW - De **MoSCoW-methode** is een manier voor het stellen van prioriteiten.

- **Must have** - deze eis *moet* in het eindresultaat verwerkt zijn.
- **Should have** - deze eis is zeer gewenst, maar een vergelijkbare eigenschap is ook goed.
- **Could have** - deze eis mag aan bod komen indien er nog tijd aanwezig is.
- **Would like to have** - deze eis komt zeker niet tijdens de projectduur aan bod. Maar kan in de toekomst interessant zijn om geïmplementeerd te worden.

MySQL - MySQL is een open source relationele databasemanagementsysteem (RDBMS), dat gebruikmaakt van SQL

PHP - Een scripttaal die bedoeld is om op webserver dynamische webpagina's te creëren.

Query - Met een query wordt in de informatica een opdracht bedoeld die aan een database wordt gegeven om een bepaalde actie uit te voeren, die ook potentieel gegevens terug geeft.

RUP – **R**ational **u**nified **p**rocess. Betreft een ontwikkelmethode om een productvisie om te zetten in een product.

SQL - (Structured Query Language) is een ANSI/ISO-standaardtaal voor een relationeel 'database management systeem' (DBMS).



Afstudeerscriptie

UML - De Unified Modeling Language, is een modelmatige taal om objectgeoriënteerde analyses en ontwerpen voor een informatiesysteem te kunnen maken.

WAMP - Een acroniem voor een combinatie van softwarepakketten waarop een dynamische website kan draaien:

- Windows (besturingssysteem)
- Apache (webserver)
- MySQL (databaseserver)
- PHP (scripttaal) of Perl of Python

XML - Extensible Markup Language is een standaard van het World Wide Web Consortium voor de syntaxis van formele markuptalen waarmee men gestructureerde gegevens kan weergeven in de vorm van platte tekst. Deze representatie is zowel machineleesbaar als leesbaar voor de mens.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

18. Bronnenlijst

www.php.net - informatie over verschillende php functies

<http://www.phpfreakz.nl/> - het forum voor fouten die al eerder voorgekomen zijn.

<http://dev.mysql.com/> - voor alles wat met mysql te maken heeft.

www.wikipedia.nl – voor begrippenlijst.

Dictaat databases, Henk van den Bosch – Dictaat over het ontwikkelen van databases.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

19. Externe bijlage

De externe bijlage zijn te vinden op bijgevoegde CD. Hieronder is een overzicht te vinden welke bijlage bij welk document hoort.

Externe bijlage 1 – Plan van aanpak.

Externe bijlage 2 – Software requirements specification document.

Externe bijlage 3 – Ontwerpdokument.

Externe bijlage 4 – Gebruikershandleiding.

Externe bijlage 5 – Ontwikkelaarhandleiding.

Externe bijlage 6 – Testdocument.

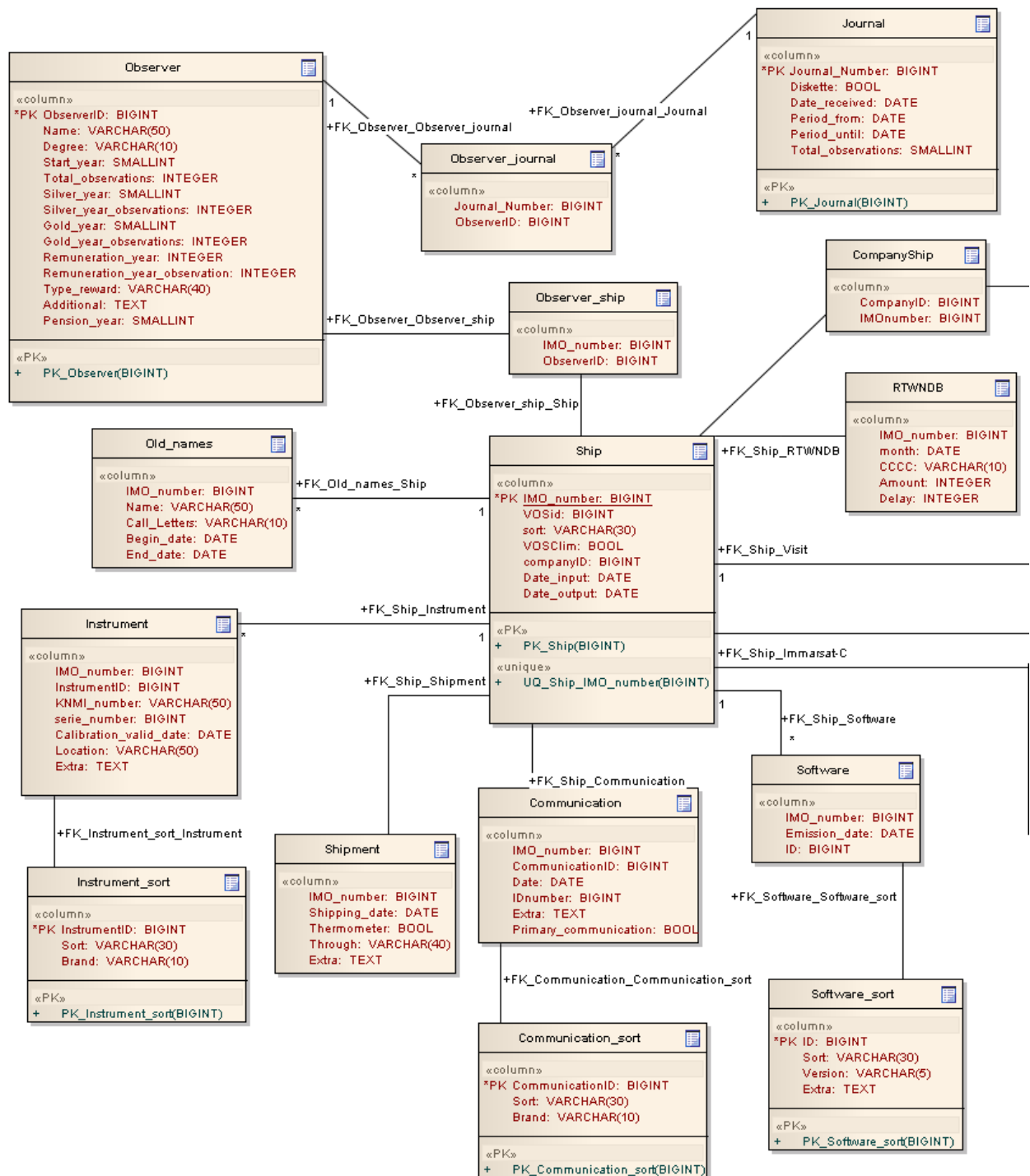


Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Bijlage 1 – Database ontwerp

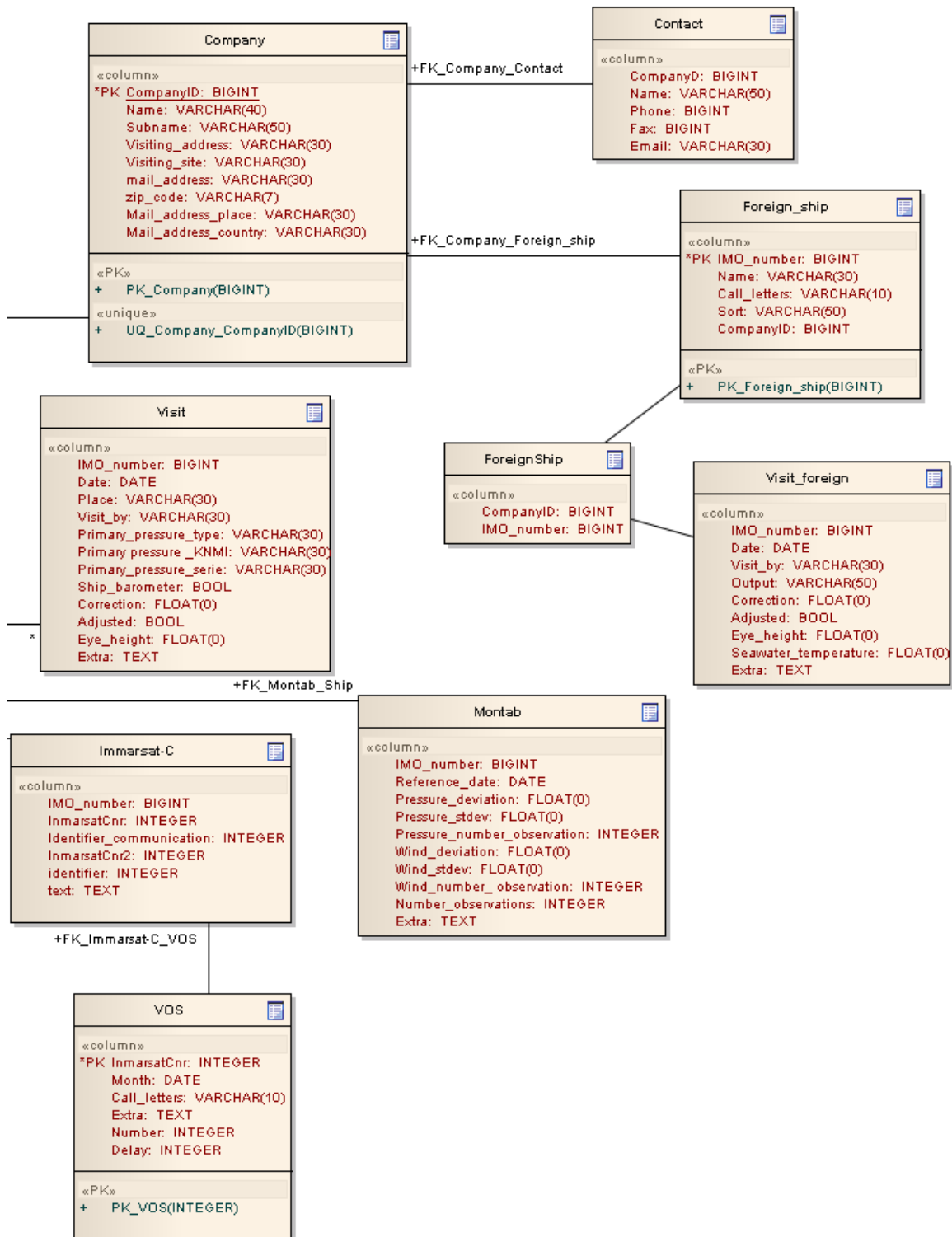
```
class Database
```



Figuur 19 Database

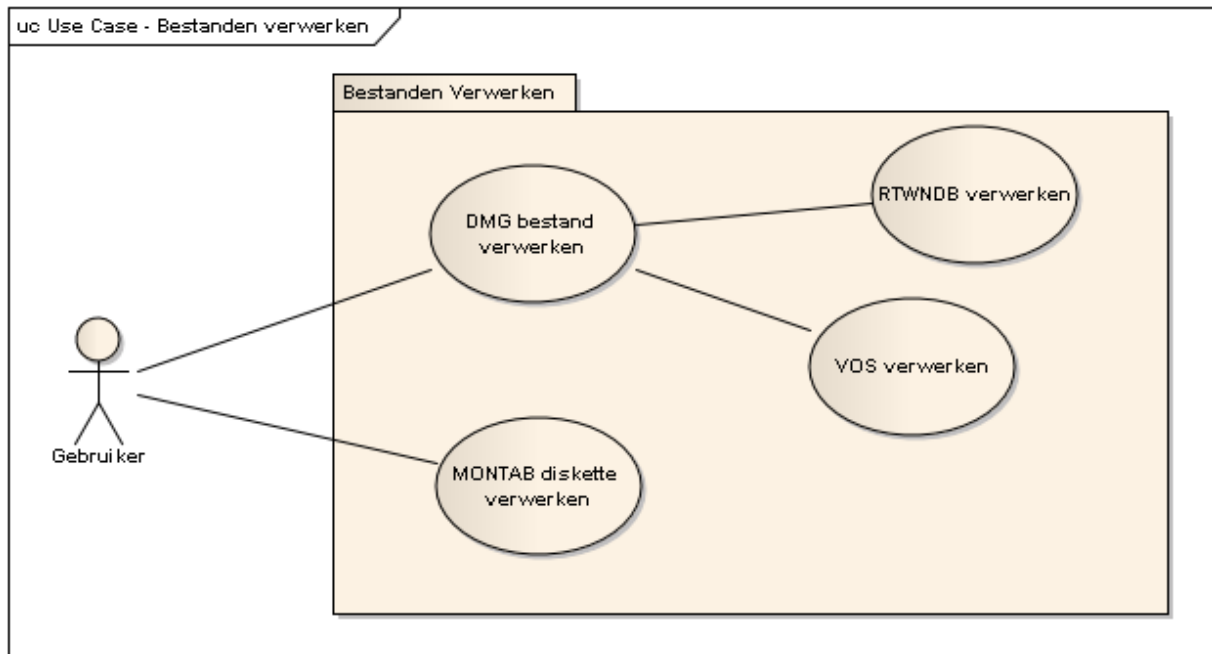


Afstudeerscriptie



Afstudeerscriptie

Bijlage 2 – Use case bestand verwerken



Figuur 20 Use case bestand verwerken

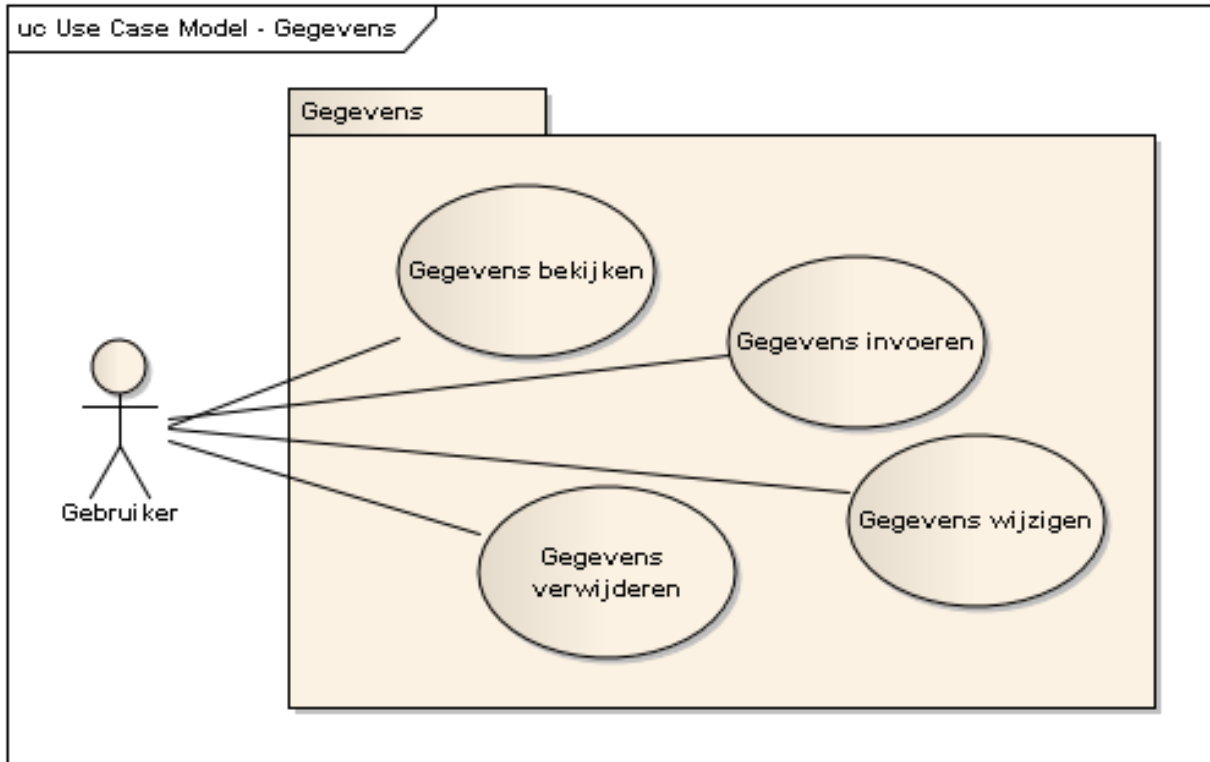


Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Afstudeerscriptie

Bijlage 3 – Use case gegevens



Figuur 21 Use case gegevens

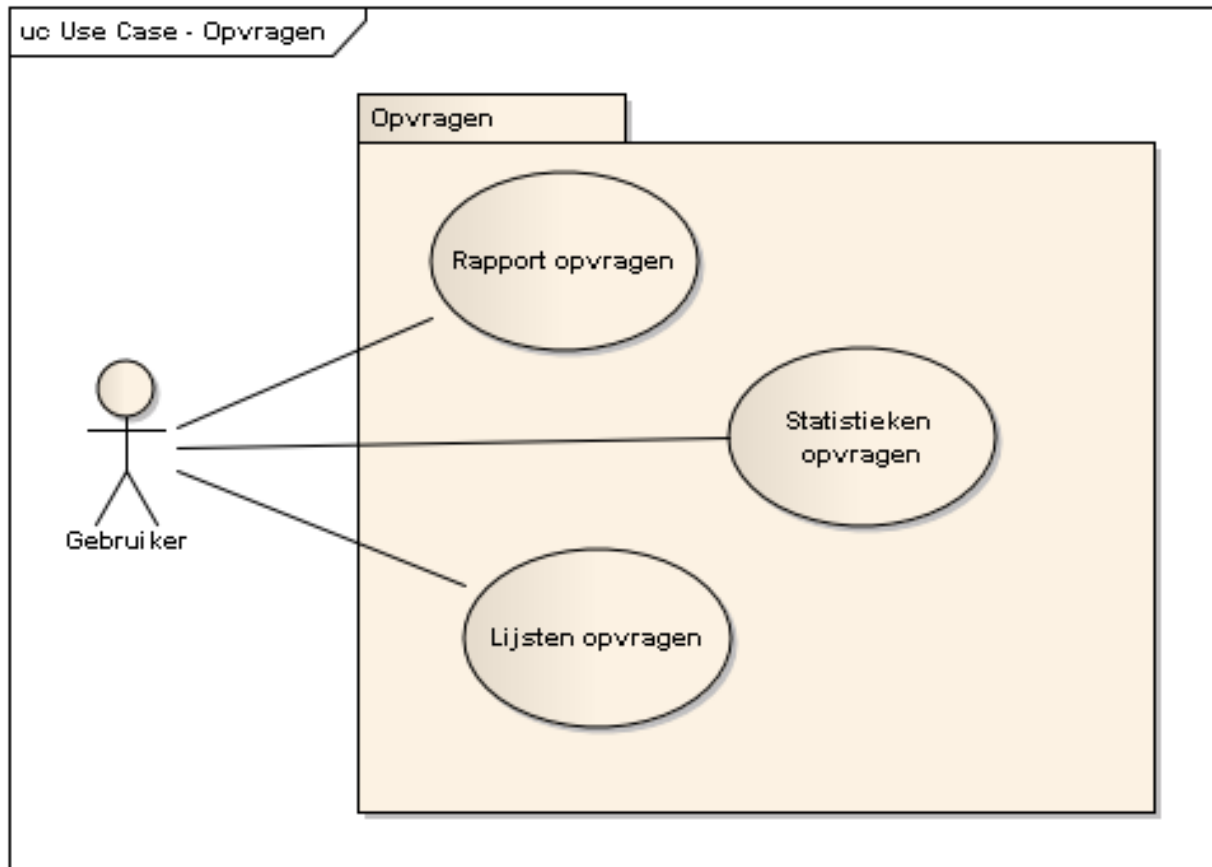


Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie

Afstudeerscriptie

Bijlage 4 – Use case opvragen



Figuur 22 Use case opvragen



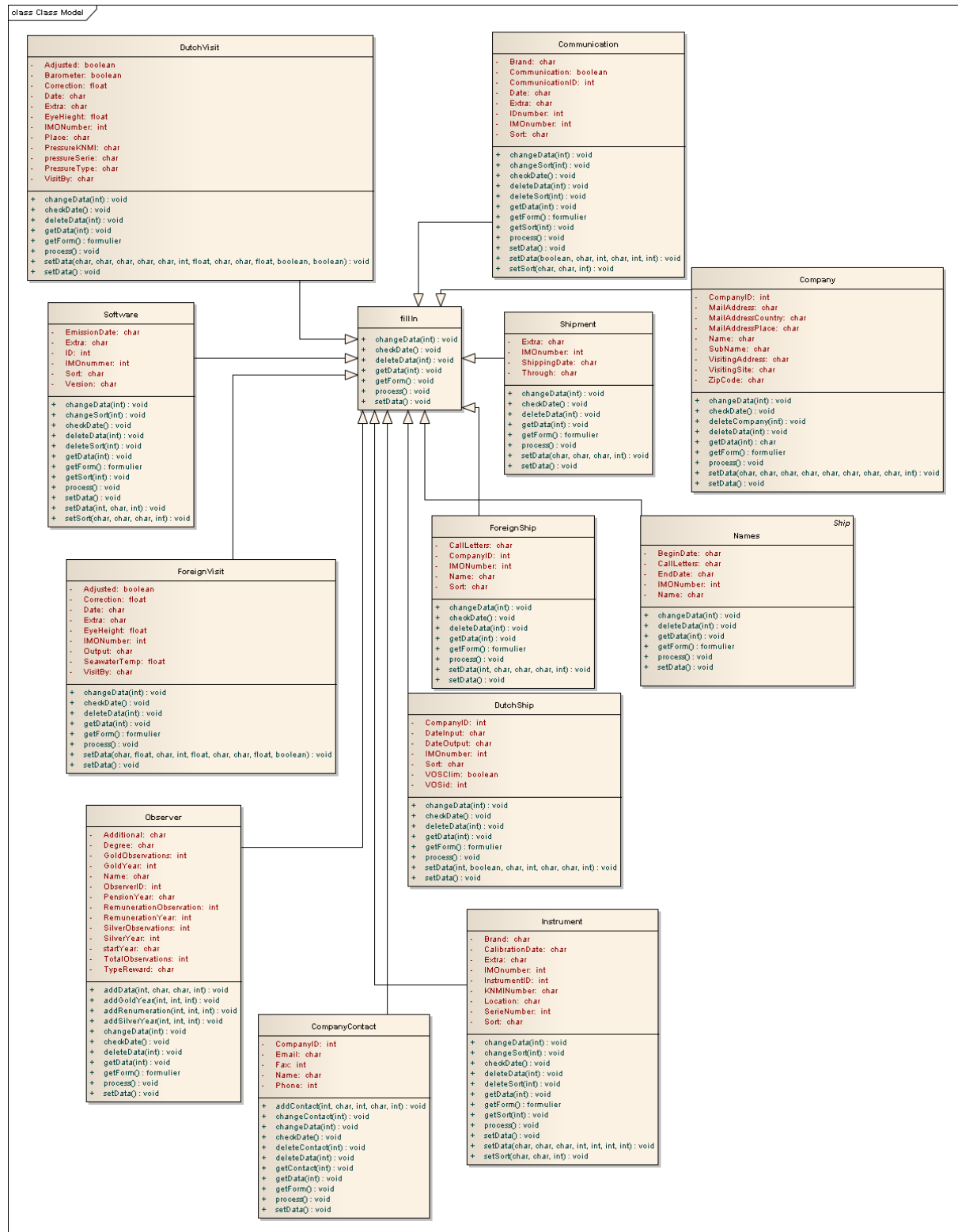
Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Bijlage 5 – Klassendiagram handmatig invoer 1



Figuur 23 Klassendiagram gegevens invullen



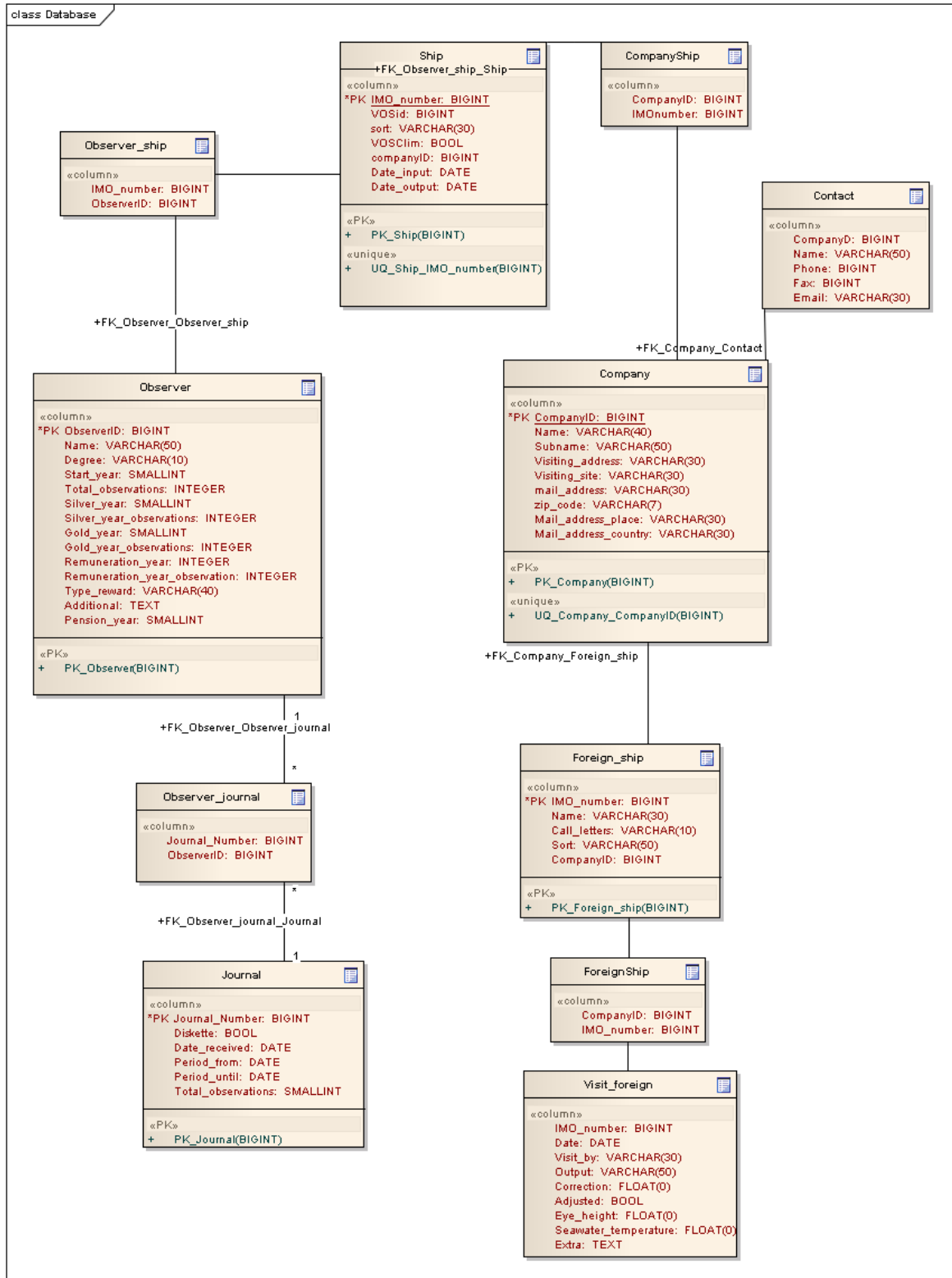
Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Bijlage 6 – Klassendiagram handmatig invoer 2



Figuur 24 Klassendiagram Handmatige invoer 2



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Bijlage 7 – Rapporten

De volgende rapporten zijn geïmplementeerd in de software:

- Ships
- Visit
- Shipments
- Software
- Communication
- Instrument
- Foreign ship
- Visit foreign ship
- Nota NL
- Nota buitenland
- Nota totaal
- Nota overig
- Nota Duitsland (alternatief)
- RTW ontvangen totalen
- Nederlandse schepen via alle stat
- Nederlandse schepen via EHDB
- Rpt AOW Nederlandse schepen
- Rpt AOW verworpen berichten
- Buitenlandse schepen via alle stat
- Buitenlandse schepen via EHDB
- AOW alle schepen
- Nederlandse schepen (vos id) via alle stat
- Nederlandse schepen (vos id) via EHDB
- Rpt AOQ Nederlandse schepen (vos id)
- Company
- Instrument sort
- Journal
- MONTAB
- Observer
- Observerjournal



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Bijlage 8 – Lijsten

De volgende lijsten kunnen opgevraagd worden:

- Instrument
- Instrument sort
- Shipment
- Communication
- Communication sort
- Software
- Software sort
- Visit
- Company
- Ship
- Observer
- Visit foreign
- Foreign ship
- Journal



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Afstudeerscriptie

Bijlage 9 – Invullen

De volgende zaken kunnen in het systeem ingevoerd worden:

- Company
- Company contact
- Company ship
- Company foreign ship
- Instrument
- Instrument sort
- Shipment
- Communication
- Communication sort
- Software
- Software sort
- Visit
- Ship
- Observer
- Foreign visit
- Foreign ship
- Journal
- Ship name
- Observer company
- Observer journal



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Afstudeerscriptie



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Plan van aanpak

Naam: B.A.S. Wille
Studentnummer: 60114
Datum: 18-02-2010
Versie: 1.0



Plan van aanpak

Samenvatting

Het KNMI is hét nationale instituut voor weer, klimaat en seismologie. Het KNMI verstrekt weerinformatie ten behoeve van veiligheid, economie en duurzaam milieu aan het algemeen publiek, de overheid, de luchtvaart en de scheepvaart.

De afstudeeropdracht betreft het ontwikkelen van een systeem dat bedoeld is voor het beheer van meteorologische meetinstrumenten en sensoren op de schepen en het monitoren van de data.

Het doel van de opdracht is het ontwikkelen van een goed systeem voor de opslag, verwerking en het beschikbaar stellen van de gegevens (metadata) van de schepen die behoren tot de VOS¹ vloot. Deze schepen doen op vrijwillige basis mee aan waarnemingen op zee ten behoeve van de verwachtingen en weermodellen. Indien er nog tijd beschikbaar is vindt er een onderzoek plaats naar de mogelijkheden om het systeem internationaal in te zetten. Er zou dan een koppeling moeten komen met de databases van andere landen.

De opdracht wordt uitgevoerd met RUP². Om de functionele specificaties te achterhalen worden er interviews afgenomen met de eindgebruikers. Uit deze interviews wordt via de MoSCoW³ methode bekeken welke elementen als eerste in het uiteindelijke product geïmplementeerd gaan worden.

Tijdens dit project zal *Enterprise Architect* gebruikt worden om de modellen te maken. Tevens zal gebruik gemaakt worden van een WAMP⁴ omgeving om de programmatuur te ontwikkelen. De test- en operationele omgeving draait op een LAMP⁵ omgeving.

Een project brengt altijd risico's met zich mee. Zo is het mogelijk dat het uiteindelijke product niet is wat de eindgebruiker in gedachten had. Om deze problemen te voorkomen moeten maatregelen genomen worden. Dit kan gedaan worden door tijdens het ontwikkelproces de gebruiker steeds te laten evalueren of het product nog aan zijn eisen voldoet.

¹ Voluntary Observation SHIP program

² Rational Unified Process

³ Must have, Could have, Should have, Would like to have

⁴ Windows, Apache, MySQL, PHP

⁵ Linux, Apache, MySQL, PHP



Plan van aanpak

Inhoudsopgave

Hoofdstuk 1: Achtergrond.....	5
1.1 Bedrijfsbeschrijving.....	5
1.2 Aanleiding tot het project	5
Hoofdstuk 2 : Opdrachtschrijving	6
2.1 Probleemstelling	6
2.2 Doelstelling	6
2.3 Opdrachtschrijving	6
2.4 Op te leveren producten	8
2.5 Projectgrenzen	8
2.6 Randvoorwaarden.....	9
Hoofdstuk 3: Aanpak	10
3.1 Aanpak van de opdracht.....	10
3.2 Werkzaamheden.....	10
3.3 Verwachte resultaat	11
3.4 Planning	11
Hoofdstuk 4: Projectinrichting	14
4.1 Faciliteiten	14
4.2 Kwaliteitsborging	14
Hoofdstuk 5: Risicofactoren	15
5.1 Risicofactoren	15
5.2 Maatregelen	15



Plan van aanpak

Inleiding

Dit plan van aanpak heeft als doel inzicht te geven in de aanpak van de project opdracht. Over deze opdracht zal uitgebreide informatie geven worden. Op deze manier wordt het duidelijk hoe de opdracht aangepakt gaat worden. Tevens is het ook voor "buitenstaanders" mogelijk om een beeld te vormen over de opdracht en de aanpak ervan.

Begonnen wordt met een stuk achtergrond. Hierin wordt het bedrijf en de opdracht globaal beschreven. (**Hoofdstuk 1**)

Bovendien wordt de probleemstelling en de doelstellingen van de opdracht behandeld. Hierdoor zal een beter beeld gevormd kunnen worden van de opdracht (**Hoofdstuk 2**)

Om een idee te krijgen hoeveel tijd het project in beslag neemt wordt er een planning gemaakt, deze biedt een overzicht van alle activiteiten en tevens een indicatie wanneer deze aanvangen en eindigen. (**Hoofdstuk 3**)

Ook wordt er aandacht besteed aan de kwaliteit, waarborging en de faciliteiten die gebruikt gaan worden. (**Hoofdstuk 4**)

Een project brengt ook risico's met zich mee. Om deze risico's zo veel mogelijk te vermijden moet er van te voren een manier bedacht worden om deze te ondervangen. (**Hoofdstuk 5**)



Plan van aanpak

Hoofdstuk 1: Achtergrond

Om een beeld te vormen van het opdrachtgevende bedrijf en het project wordt er achtergrondinformatie gegeven. In dit hoofdstuk wordt het soort bedrijf en het doel van het bedrijf beschreven (**Hoofdstuk 1.1**). Ook is de aanleiding beschreven voor het starten van dit project (**Hoofdstuk 1.2**).

1.1 Bedrijfsbeschrijving

Het KNMI¹ is hét nationale instituut voor weer, klimaat en seismologie. Het KNMI verstrekt weerinformatie ten behoeve van veiligheid, economie en duurzaam milieu aan het algemeen publiek, de overheid, de luchtvaart en de scheepvaart. Voor lange termijnontwikkelingen verricht het KNMI onderzoek naar de veranderingen in het klimaat. Het beschikbaar stellen van bij het KNMI aanwezige kennis, data en informatie is een kernactiviteit. Uiteraard kan dit slechts voor het KNMI voor zover dit past binnen de geldende internationale overeenkomsten.

Het instituut is een agentschap van het Ministerie van Verkeer & Waterstaat.

Het KNMI is vierentwintig uur per dag actief. De combinatie van infrastructuur, techniek, wetenschap en dienstverlening is uniek op het gebied van weer, klimaat en seismologie. Wereldwijd werkt het KNMI intensief samen met andere instituten en onderzoeksinstellingen. Bijvoorbeeld met het Europees Centrum voor Weersverwachtingen op de Middellange Termijn (ECMWF) in Engeland. Het KNMI vertegenwoordigt Nederland in tal van andere internationale organisaties, zoals de Wereld Meteorologische Organisatie (WMO), het Intergovernmental Panel on Climate Change (IPCC) en het Europese meteorologische satellietnetwerk Eumetsat.

1.2 Aanleiding tot het project

Het KNMI wint, naast de gegevens van 35 automatische waarnemstations, ook scheepswaarnemingen in die op vrijwillige basis door de officieren aan boord verricht worden. Deze schepen worden de VOS² vloot genoemd en omvat ongeveer 200 schepen.

Momenteel is er geen goed systeem aanwezig dat het beheer van de meteorologische meetinstrumenten en sensoren op de schepen en het monitoren van de data afhandelt. Hierdoor worden de gegevens die beschikbaar zijn niet volledig benut.

¹ Koninklijk Nederlands Meteorologisch Instituut

² Voluntary Observation SHIP program



Plan van aanpak

Hoofdstuk 2 : Opdrachtomschrijving

In dit hoofdstuk volgt een duidelijke beschrijving van de opdracht zodat hierover een goed beeld gevormd kan worden. Het wordt zo ook duidelijker welke taken uitgevoerd moeten worden.

Het probleem dat opgelost moet worden wordt gedefinieerd (**Hoofdstuk 2.1**), de doelstelling wordt bepaald (**Hoofdstuk 2.2**), de opdracht wordt duidelijk omschreven en het eindproduct wordt gedefinieerd (**Hoofdstuk 2.3**). Daarnaast worden de op te leveren producten vastgesteld (**Hoofdstuk 2.4**) en worden de projectgrenzen (**Hoofdstuk 2.5**) en randvoorwaarden (**Hoofdstuk 2.6**) bepaald.

2.1 Probleemstelling

De afstudeeropdracht betreft het ontwikkelen van een systeem dat bedoeld is voor het beheer van meteorologische meetinstrumenten en sensoren op de schepen en het monitoren van de data.

Het vervult daarmee een noodzakelijke rol voor de Port Meteorological Officer. Momenteel is er geen goed systeem beschikbaar waardoor een nieuw systeem ontwikkeld zal moeten worden. Bij de ontwikkeling van dit systeem wordt zoveel mogelijk rekening gehouden met de toekomstige ontwikkelingen zoals het aansluiten op internationale VOS systemen.

2.2 Doelstelling

Het doel van de opdracht is het ontwikkelen van een goed systeem voor de opslag, verwerking en het beschikbaar stellen van de gegevens (metadata) van de schepen die behoren tot de VOS. Deze schepen doen op vrijwillige basis mee aan waarnemingen op zee ten behoeve van de verwachtingen en weermodellen. Indien er nog tijd beschikbaar is kan een onderzoek plaatsvinden naar de mogelijkheden om het systeem internationaal in te zetten. Er zou dan een koppeling moeten komen met de databases van andere landen.

2.3 Opdrachtomschrijving

De opdracht omvat het ontwikkelen van een nieuw systeem ter opslag, verwerking en beschikbaarheid van de gegevens van de VOS. Hierbij wordt de meteorologisch relevante metadata opgeslagen. Deze data bevat onder andere:

- IMO¹ nummer
- Positie
- Naam
- Maatschappij
- Telefoonnummer
- Aanwezige meetinstrumenten en sensoren

¹ International Maritime Organization



Plan van aanpak

- Aanwezige software
- Plaats van controle
- Bezocht door
- Drukcorrectie
- Scheepsbezoeken

Dit is echter maar een klein deel van de opdracht. Naast de hierboven genoemde data dient er namelijk ook een journaaladministratie bijgehouden te worden waarin onder andere staat welke waarnemers gemeten hebben en het aantal van deze waarnemingen.

Ook gaat de beveiliging van het systeem een rol spelen. Het is namelijk mogelijk om op verschillende websites op te zoeken waar een schip zich op dit moment bevindt. Sommige schepen willen dit echter niet vanwege mogelijke piraterij. Hier zal dus een oplossing voor gevonden moeten worden.

Er moeten interviews gaan plaatsvinden om de eisen van de toekomstige gebruikers vast te kunnen leggen. Hierdoor zal mede bepaald worden welke onderdelen gerealiseerd gaan worden. Het is daarom nog mogelijk dat er andere elementen in de opdracht een rol gaan spelen.

Indien er nog tijd beschikbaar is kan er nog een onderzoek uitgevoerd worden. Dit onderzoek concentreert zich dan op de haalbaarheid van de applicatie in internationaal verband. Engeland, Duitsland en Frankrijk zijn tevens geïnteresseerd in een systeem waarin de VOS gegevens zitten. De kans bestaat dus dat deze landen een koppeling willen met het systeem.

Een onderzoek naar de haalbaarheid van een koppeling van het systeem met hun database kan dan plaatsvinden. Echter kan al bij de ontwikkeling van het systeem rekening gehouden worden met de mogelijkheid dat andere landen een koppeling willen.

Het systeem moet uiteindelijk over de volgende functionaliteit beschikken:

- Het interactief zoeken middels een grafische user interface naar informatie van schepen. Dit zoeken moet kunnen gebeuren op een specifiek schip of periode. De uitvoer die op het scherm verschijnt moet uitgeprint of in een bestand gezet kunnen worden.
- Verschillende presentaties van de metadata maken ten behoeve van verschillende gebruikers.
- De betrouwbaarheid van de meetinstrumenten en sensoren kunnen monitoren.
- Aansluiting op de VOS metadata applicatie middels in- en exportfuncties van de data.
- Met behulp van Google Maps moet de positie van de schepen op aarde bekeken kunnen worden. Bij klikken op de naam van het schip wordt automatisch de naam van het schip, IMO nummer en maatschappij getoond.

Dit kan mogelijk nog wijzigen nadat er interviews met alle toekomstige gebruikers hebben plaatsgevonden, maar geeft toch een goede indicatie over de basisfunctionaliteit die het systeem zal bevatten.



Plan van aanpak

2.4 Op te leveren producten

Bij dit project worden verschillende producten gerealiseerd die samen het eindproduct vormen.

Deze producten zijn:

- 1) Documentatie systeem
- 2) Code Systeem
- 3) Software
- 4) Functionele Specificaties
- 5) Scriptie

Ad 1) Documentatie systeem

Documentatie van de gemaakte software. Deze documentatie beschrijft de precieze werking van de software zodat deze later eenvoudig door andere programmeurs aangepast kan worden. De documentatie beschrijft de volledig ontwikkelde software.

Ad 2) Code

De ontwikkelde software zal ook als code opgeleverd worden.

Ad 3) Software

De software zelf zal opgeleverd worden. Inclusief de database die aan de software hangt.

Ad 4) Functionele specificaties

Een document wat de eisen van de toekomstige gebruikers beschrijft. Hierdoor wordt het duidelijk wat men in de software wil en waar men de software in de toekomst kan uitbreiden.

Ad 5) Scriptie

Aangezien het project ook een afstudeerstage is zal er een scriptie over gemaakt worden. In deze scriptie zal precies duidelijk worden wat er gedaan is in de 17 weken dat het project loopt.

2.5 Projectgrenzen

Het project zal zich concentreren op het achterhalen van de functionele specificaties en het programmeren van de software. Doormiddel van de MoSCoW¹ methode worden de functionele specificaties opgedeeld waardoor goed zichtbaar is welke zaken eerst ontwikkeld moeten worden. Dit wordt gedocumenteerd in de software requirements specification.

Indien er nog tijd beschikbaar is in de 17 weken dat het project zal draaien kan er nog een onderzoek uitgevoerd worden. Een onderzoek naar de haalbaarheid van de toepassing in internationaal verband behoort hierbij tot de mogelijkheid. Indien er geen tijd meer beschikbaar is komt dit deel te vervallen.

¹ Must have, Could have, Should have, Would like to have



Plan van aanpak

2.6 Randvoorwaarden

Voor het project gelden de volgende randvoorwaarden:

- De software moet op een zodanige manier ontwikkeld worden dat deze eenvoudig te onderhouden is.
- De software moet eenvoudig uitbreidbaar zijn.
- Er moet rekening gehouden worden met een koppeling naar een buitenlandse database.
- De applicatie en daarmee de data moet benaderbaar zijn middels een webbrowser.
- Er dient ingelogd te worden met usercode/password.
- De applicatie dient gebouwd te worden met ondersteunde technieken door de afdeling INFRA-ICT.



Plan van aanpak

Hoofdstuk 3: Aanpak

Om een goed resultaat op te leveren moet er eerst nagegaan worden welke stappen ondernomen moeten worden en in welke volgorde. In dit hoofdstuk wordt de aanpak van de opdracht beschreven (**Hoofdstuk 3.1**). Wordt gekeken naar werkzaamheden die verricht moeten worden (**Hoofdstuk 3.2**), het verwachte resultaat (**Hoofdstuk 3.3**) en de planning (**Hoofdstuk 3.4**).

3.1 Aanpak van de opdracht

Om de eisen aan het eindproduct vast te leggen moet met de eindgebruikers gesproken worden. Door middel van interviews moeten de eisen van de toekomstige gebruikers naar voren komen.

Een functioneel ontwerp zal duidelijk moeten gaan maken wat de eisen van de gebruiker zijn. Hierna zal via de MoSCoW methode bekeken worden wat de belangrijkste eisen zijn.

Bij deze opdracht zal gebruik gemaakt gaan worden van de ontwikkelmethode RUP. Het product zal hierdoor in verschillende iteraties gemaakt gaan worden. Elke iteratie voegt nieuwe functionaliteit toe aan het eindproduct. Hierdoor is het mogelijk om tussentijds een demo van de tot dan toe gerealiseerde software te tonen waardoor eventuele eisen van toekomstige gebruikers aangepast kunnen worden.

Tevens zal een ontwerpdocument gerealiseerd worden welke het nieuwe ontwerp van de database en software uitlegt.

3.2 Werkzaamheden

De werkzaamheden bestaan als eerste uit het achterhalen van de wensen van de toekomstige gebruiker.

Aan de hand van de interviews die plaatsvinden wordt een functioneel ontwerp gemaakt. In dit functionele ontwerp wordt duidelijk hoe de software er uit moet gaan zien. Hierna worden met de MoSCoW methode de eisen van de gebruikers onderverdeeld. Hierdoor ontstaat een goede indeling waarin de belangrijkste eisen altijd uitgevoerd gaan worden. Dit gaat beschreven worden in het software requirements specificatie document.

Nadat het functioneel ontwerp voltooid is en de eisen zijn onderverdeeld via de MoSCoW methode wordt het product gerealiseerd. Hiervoor zullen de werkzaamheden opgedeeld worden in verschillende iteraties. Elk van deze iteraties zal nieuwe functionaliteit toe gaan voegen.

Om dit te kunnen verwezenlijken zal een nieuwe database ontwikkeld moeten worden. Echter moet de oude database omgezet worden in de nieuwe waar tevens rekening mee gehouden moet worden.



Plan van aanpak

3.3 Verwachte resultaat

Het verwachte resultaat is een product wat goed gedocumenteerd wordt zodat in de toekomst ook het product door andere medewerkers aangepast kan worden. Tevens wordt een werkend stuk software verwacht die voldoet aan de eisen van de gebruikers. Hierbij zullen in elk geval de "Must Haves" uit de MoSCoW methode geïmplementeerd zijn. Als eindproduct wordt een internet applicatie die voldoet aan de belangrijkste functionele specificaties van de gebruiker verwacht.

3.4 Planning

Het project wordt opgedeeld in verschillende iteraties. Hoe deze iteraties er op dit moment uit gaan zien is nog niet bekend. Dit zal bekend worden nadat er interviews gedaan zijn met de gebruikers. Hierdoor zal een duidelijker beeld naar voren komen van de werkzaamheden. Dit wordt vastgelegd in een functioneel ontwerp waarna de iteraties uiteengezet gaan worden.

Een gedetailleerde planning voor de korte termijn ziet er als volgt uit:

Planning korte termijn

Week	Activiteit
Week 1 (08-02-10/12-02-10)	Inwerken, PvA maken
Week 2 (15-02-10/19-02-10)	PvA afronden Functioneel ontwerp maken.
Week 3 (22-02-10/26-02-10)	Functioneel ontwerp afronden.
Week 4 (01-03-10/05-03-10)	Modelleren
Week 5 (08-03-10/12-03-10)	Database ontwerpen. Indelen iteraties.

Een planning voor de lange termijn is op dit moment nog moeilijk te realiseren. Nadat een functioneel ontwerp en de Software requirements gemaakt zijn kan het project pas in goede iteraties opgedeeld worden. Aangezien er gebruik gemaakt wordt van RUP is het wel duidelijk dat er gebruik gemaakt wordt van vier verschillende fases. Deze fases kunnen al wel in een planning gezet worden waardoor een beeld ontstaat van hoe lang de verschillende onderdelen gaan duren. Een globale planning voor de lange termijn ziet er daarom nu als volgt uit:

Planning lange termijn

Week	Activiteit
Week 1 (08-02-10/12-02-10)	Inception Fase
Week 2 (15-02-10/19-02-10)	
Week 3 (22-02-10/26-02-10)	Elaboration Fase
Week 4 (01-03-10/05-03-10)	
Week 5 (08-03-10/12-03-10)	
Week 6 (15-03-10/19-03-10)	
Week 7 (22-03-10/26-03-10)	Construction Fase
Week 8 (29-03-10/02-04-10)	



Plan van aanpak

Week 9 (05-04-10/09-04-10)	
Week 10 (12-04-10/16-04-10)	
Week 11 (19-04-10/23-04-10)	
Week 12 (26-04-10/30-04-10)	
Week 13 (03-05-10/07-05-10)	
Week 14 (10-05-10/14-05-10)	
Week 15 (17-05-10/21-05-10)	
Week 16 (24-05-10/28-05-10)	
Week 17 (31-05-10/04-06-10)	Transition Fase

Bij RUP wordt er gebruikt gemaakt van vier verschillende fases. In elk van deze fases worden andere activiteiten gedaan.

De fases zijn:

- 1) Inception
- 2) Elaboration
- 3) Construction
- 4) Transition

Ad 1) Inception (aanvang)

De haalbaarheid van het project, de begrenzingen en de inhoud worden bepaald. Het oorspronkelijke idee wordt omgezet in een productvisie. Hierbij worden de belangrijkste risico's geïdentificeerd en ingeschat. Dit plan van aanpak vormt een onderdeel van de inception fase.

Ad 2) Elaboration (detaillering)

Het merendeel van de functionele specificaties wordt gespecificeerd. Ook wordt de systeemarchitectuur ontworpen. Er worden use-case's gemaakt om de functionele specificaties te visualiseren. De database wordt in deze fase ontworpen en de Software requirements worden gespecificeerd.

Ad 3) Construction (bouw)

Het product wordt ontwikkeld van de architectuur tot een compleet systeem. Dit zal gedaan worden in verschillende iteraties. Na elke iteratie wordt met de toekomstige gebruiker gekeken of het product zich in de goede richting ontwikkeld. Commentaar wordt meegenomen in de volgende iteratie.

Ad 4) Transition (afronding)

Het project wordt afgerond in deze fase. Er wordt aandacht besteed aan de nazorg en overdracht van de verantwoordelijkheden. Deze fase wordt afgesloten met een voltooide scriptie over de opgedane kennis.

Deze vier verschillende fases zullen in verschillende iteraties opgedeeld worden. Hierbij zullen voornamelijk fase twee (Elaboration) en fase drie (Construction) verschillende iteraties bevatten.

Aan het eind van elke fase vindt een milestone plaats. Bij deze milestones wordt gekeken naar wat gerealiseerd is en of er inderdaad doorgegaan kan worden naar de volgende fases.

De milestones staan gepland voor de volgende weken:



Plan van aanpak

Milestone	Week
Milestone 1	Week 2 (15-02-10/19-02-10)
Milestone 2	Eind week 5 (08-03-10/12-03-10)
Milestone 3	Eind week 15 (17-05-10/21-05-10)
Milestone 4	Eind week 17 (31-05-10/04-06-10) (oplevering product)

De precieze invulling van de iteraties bij de construction fase en het aantal iteraties wordt bekeken in de elaboration fase. De elaboration fase gaat waarschijnlijk drie iteraties bevatten:

- 1) Functionele specificaties, Software requirements en use-case's
- 2) Ontwerpen database
- 3) Modelleren software

Nadat de functionele specificaties voltooid zijn is het duidelijk welke elementen precies ontwikkeld gaan worden waardoor voor de construction fase de iteraties gepland kunnen worden.



Plan van aanpak

Hoofdstuk 4: Projectinrichting

Om een project goed te kunnen voltooien is het noodzakelijk dat er bepaalde faciliteiten beschikbaar zijn (**Hoofdstuk 4.1**). Om de kwaliteit van het eindproduct te kunnen waarborgen moeten ook verschillende handelingen worden verricht (**Hoofdstuk 4.2**).

4.1 Faciliteiten

De modellen worden gerealiseerd in *Enterprise Architect*. Om de modellen te realiseren zal UML gehanteerd worden.

Om de software te ontwikkelen zal gebruik gemaakt worden van een WAMP¹ server waarop de software in een testomgeving kan draaien. Voor het schrijven van de software wordt gebruik gemaakt van Notepad++ of een vergelijkbaar programma.

4.2 Kwaliteitsborging

Om de kwaliteit van het eindproduct te waarborgen zal na elke iteratie met verschillende gebruikers bekeken worden of het product zich in de juiste richting ontwikkeld. Hierdoor ontstaat tussentijds de mogelijkheid het product verder naar de gebruiker toe te ontwikkelen. Het wordt hierdoor makkelijker om een product te ontwikkelen wat de gebruiker wenst.

Na elke fase vind een milestone plaats waarin gekeken wordt of verder gegaan kan worden naar de volgende fase. Dit zijn momenten waarin de kwaliteit van het product tot nu toe geëvalueerd kan worden.

¹ Windows, Apache, MySQL, PHP



Plan van aanpak

Hoofdstuk 5: Risicofactoren

Elk project brengt risico's met zich mee. Door van te voren vast te stellen wat de mogelijke risico's zijn (**Hoofdstuk 5.1**) is het mogelijk al maatregelen te nemen om deze te voorkomen (**Hoofdstuk 5.2**).

5.1 Risicofactoren

Een project brengt altijd risicofactoren met zich mee.

Risicofactoren die bij dit project een rol kunnen spelen zijn:

- De eisen van de gebruikers liggen te ver uit elkaar.
- Het product voldoet niet aan de eisen van de gebruikers.
- In een later stadium komen nog belangrijke eisen naar boven

5.2 Maatregelen

Om de risicofactoren van te voren zo klein mogelijk te houden moeten er maatregelen worden getroffen. De maatregelen zijn als volgt:

Eisen van gebruikers liggen te ver uit elkaar

Om dit probleem te elimineren zal goed gekeken worden naar wat daadwerkelijk van belang is voor het product. Doormiddel van de MoSCoW methode worden de eisen van gebruikers in verschillende categorieën ingedeeld. Het is van groot belang dat de eisen van de gebruiker zo snel mogelijk op papier komen omdat dit de basis vormt van het product.

Het product voldoet niet aan de eisen van de gebruikers

Om het product aan de eisen van de gebruiker te laten voldoen zal de gebruiker betrokken moeten worden bij het product. Doordat gebruik gemaakt wordt van RUP en er verschillende iteraties zullen plaatsvinden is het mogelijk de gebruiker tussentijds een demo te geven. De wijzigingen die aan de hand hiervan doorgevoerd moeten worden kunnen dan meegenomen worden in de volgende iteratie.

In een later stadium komen nog belangrijke eisen naar boven

Het is altijd mogelijk dat in een laat stadium van de ontwikkeling nieuwe belangrijke eisen naar boven komen. Het is daarom belangrijk dat bij het bepalen van de functionele specificaties en de Software requirements duidelijk gekeken wordt naar de basisfunctionaliteit die gerealiseerd moet worden. Indien eenmaal de functionele en de Software requirements zijn goedgekeurd wordt hier in de meeste gevallen niks meer aan toegevoegd. Afhankelijk van de mate van prioriteit/invloed kan besloten worden de eis alsnog te implementeren echter behoort dit tot de uitzonderingen.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Software requirements specification

Naam: B.A.S. Wille
Studentnummer: 60114
Datum: 16-03-2010
Versie: 1.0



Software Requirements Specification

Samenvatting

Het systeem wat ontwikkeld gaat worden moet in staat zijn verschillende data op te slaan. Deze data moet door middel van verschillende rapporten gepresenteerd kunnen worden. Het moet hierbij mogelijk zijn de ingevoerde data aan te passen of te verwijderen.

Zaken die in het systeem ingevoerd moeten kunnen worden zijn:

- Schepen
- Waarnemers
- Journaals
- Maatschappijen
- Software
- Instrumenten
- Nota's
- Bezoeken aan schepen

Schepen kunnen tevens van naam veranderen waardoor een tabel bijgehouden moet worden met de namen die een schip heeft gehad.

Tevens wordt er verschillende data ingelezen vanuit tekstfiles. Data die ingelezen wordt via .txt files moet gecontroleerd worden op fouten. De gebruiker moet in staat zijn om de fouten in de files handmatig aan te passen. Het programma mag wel een suggestie geven voor verbetering maar niet de fout verbeteren. Uiteraard geeft het programma wel aan waar de fout zich bevindt.

Het programma wat ontwikkeld gaat worden moet op een LAMP¹ omgeving gaan draaien. Het wordt daarom ontwikkeld met behulp van PHP en MySQL.

Het programma moet eigenhandig taken uitvoeren zoals het valideren van de invoer en het maken van rapporten. Het systeem moet ook in staat zijn gegevens van schepen die voor het buitenland bezocht zijn op te slaan. Buitenlandse en Nederlandse schepen moeten echter wel fysiek gescheiden blijven. Twee verschillende tabellen zijn dus noodzakelijk.

Data die nu in de oude niet goed functionerende database staat moet overgezet worden naar de nieuwe database. In de toekomst moet het mogelijk zijn om een internationale database aan het systeem te koppelen. Om dit te kunnen realiseren moet een onderzoek naar de haalbaarheid hiervan plaatsvinden. Dit onderzoek zal plaatsvinden indien er nog tijd beschikbaar is. Echter zal bij het ontwerpen van de programmatuur wel al rekening worden gehouden met de mogelijkheid hiervan.

¹ Linux, Apache, MySQL, PHP



Software Requirements Specification

Inhoudsopgave

Inleiding	4
Hoofdstuk 1: Context	5
Hoofdstuk 2: Functionele eisen	6
2.1 Functionele specificaties	6
Hoofdstuk 3: Niet functionele eisen	9
3.1 Niet functionele eisen	9
Hoofdstuk 4: Indeling via MoSCoW	11
4.1 De MoSCoW methode	11
4.2: Indeling	11
4.2.1 <i>Must have</i>	11
4.2.2 <i>Should have</i>	12
4.2.3 <i>Could have</i>	12
4.2.4 <i>Would like to have</i>	12
Hoofdstuk 5: Gegevens	13
5.1 Waarnemer	13
5.2 Journaal	13
5.3 Maatschappij	14
5.4 Schepen	14
5.5 Software	15
5.6 Verzending	16
5.7 Instrument	16
5.8 Instrument t.b.v. schip	16
5.9 Communicatie	16
5.10 Buitenlandse schepen	17
5.11 Bezoek (NED)	17
5.12 Bezoek (buit)	18
5.13 Rapporten	18
5.14 Lijsten	19
5.15 Statistieken	19
Hoofdstuk 6: Use cases	20
6.1 Use case inloggen	20
6.2 Use case gegevens	21
6.3 Use case bestanden verwerken	22
6.4 Use case opvragen	24
6.5 Use case nota	26
Hoofdstuk 7: Planning	28
Bijlage 1: Interview met Peter Westenbrink	30
Bijlage 2: Interview met Rene Rozeboom	31
Bijlage 3: Vergadering	33
Bijlage 4: Rapporten	35
Bijlage 5: Rapporten DMG	43



Software Requirements Specification

Inleiding

In dit document wordt beschreven hoe het nieuw te realiseren systeem moet gaan functioneren. Door middel van het houden van interviews zijn de functionele specificaties duidelijk geworden.

Al de functionele specificaties zijn opgenomen in dit document (**Hoofdstuk 2**). Tevens wordt er nog aandacht besteed aan de niet functionele eisen (**Hoofdstuk 3**). Vervolgens worden deze specificaties geïnterpreteerd en onderverdeeld volgens de MoSCoW¹ methode (**Hoofdstuk 4**).

Ook worden de systeem requirements beschreven (**Hoofdstuk 5**) welke meer informatie geven over de inhoud die opgeslagen gaat worden.

Aan de hand van de functionele eisen worden use cases gemaakt die een beter beeld over de interactie tussen gebruiker en systeem moeten weergeven (**Hoofdstuk 6**).

Tevens is er aandacht voor een vernieuwde gedetailleerde lange termijnplanning met hierin de invulling van de iteraties (**Hoofdstuk 7**).

¹ Must have, Should have, Could have, Would like to have.



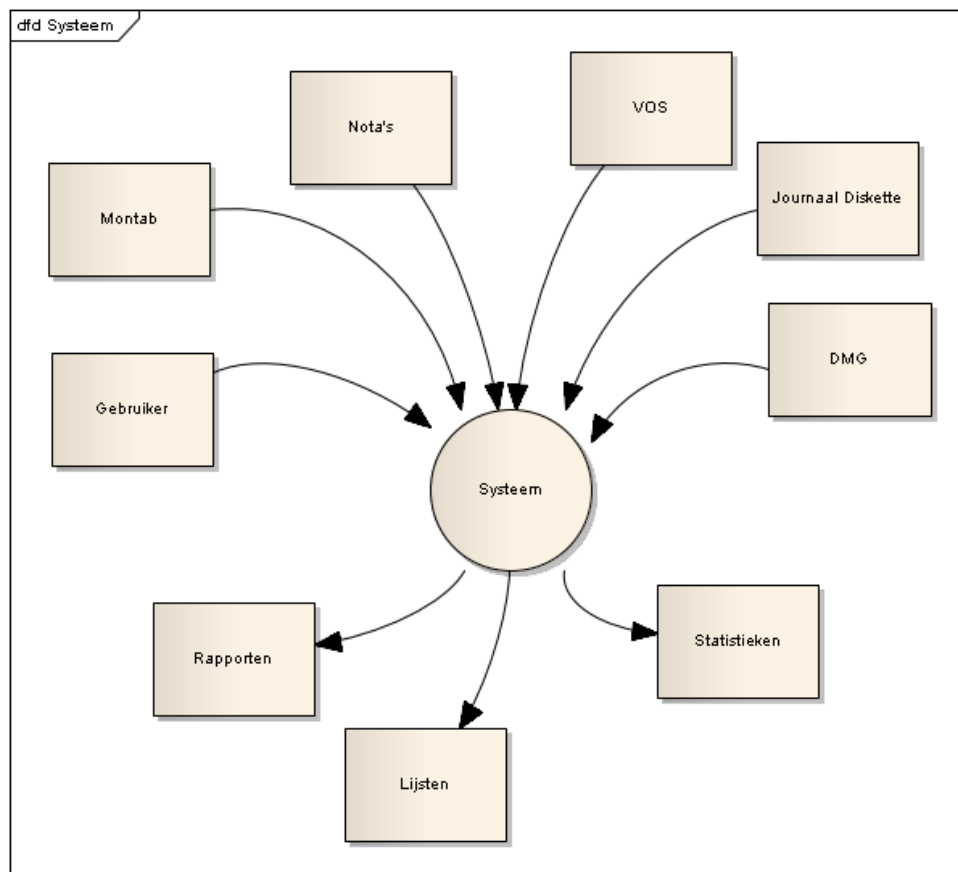
Software Requirements Specification

Hoofdstuk 1: Context

Dit project concentreert zich op het maken van een webapplicatie voor het onderhouden van informatie van de VOS¹ vloot. In dit systeem zullen zaken opgeslagen worden die noodzakelijk zijn om statistieken te kunnen bijhouden.

In eerste instantie zal het systeem zich concentreren op Nederlands gebruik. Het is mogelijk dat in de toekomst het systeem internationaal ingezet gaat worden. Hiervoor moet dan een koppeling komen met een andere database. Waar rekening gehouden kan worden met mogelijke internationale inzetbaarheid wordt dit al gedaan.

Het systeem gaat gegevens inlezen, verwerken en presenteren. Het moet mogelijk zijn om de data op te vragen in verschillende rapporten.



Figuur 1 Data Flow Diagram

¹ Voluntary Observation SHIP program



Software Requirements Specification

Hoofdstuk 2: Functionele eisen

Dit hoofdstuk behandelt de functionele eisen die belangrijk zijn voor dit project. De functionele eisen zijn opgesteld naar aanleiding van een aantal interviews (**Bijlage 1 en 2**) en een vergadering (**Bijlage 3**) die hebben plaatsgevonden.

2.1 Functionele specificaties

De functionele specificaties die achterhaald zijn voor dit systeem:

1) Nieuwe schepen moeten toegevoegd kunnen worden.

De gebruiker moet in staat gesteld worden nieuwe schepen aan het systeem toe te kunnen voegen. Welke informatie hierbij opgeslagen moet worden is te vinden in **Hoofdstuk 5**.

2) Nieuwe waarnemers moeten toegevoegd kunnen worden.

De gebruiker moet in staat zijn nieuwe waarnemers aan het systeem toe te voegen.

3) Er moet bijgehouden worden hoeveel waarnemingen een waarnemer doet.

Voor elke waarnemer wordt ook bijgehouden hoeveel waarnemingen (per journaal en totaal) hij gedaan heeft. Hierbij wordt bijgehouden wanneer hij welke beloning heeft gekregen en hoeveel waarnemingen daarvoor nodig waren.

4) De journaals moeten ingevuld kunnen worden.

De journaals bevatten dezelfde gegevens die via het GTS worden ontvangen, aangevuld met informatie over waarnemers en quality controllers

Hiervan moet geregistreerd worden over welke tijdsduur het gaat en hoeveel waarnemingen het bevat. Het programma moet automatisch een journaalnummer toekennen aan een nieuw journaal.

5) Nieuwe maatschappijen moeten ingevoerd kunnen worden.

Vanuit het systeem moet het mogelijk zijn om een nieuwe maatschappij toe te voegen. Voor een nieuwe maatschappij wordt ook een eerste en tweede contactpersoon opgeslagen met de middelen waarmee deze bereikt kunnen worden.

6) De versie en uitgiftedatum van de software op schepen moet bijgehouden worden.

Voor elk schip moet bijgehouden worden welke software aan boord is. Hierbij is de soort software en de versie van die software van belang.

7) Verzendingen naar schepen moeten bijgehouden worden.

Materiële zaken die naar schepen verzonden worden moeten worden opgeslagen in het systeem. Het is belangrijk dat de datum van verzending en via wie het verzonden is in het systeem opgeslagen worden.

8) Verschillende soorten instrumenten moeten bijgehouden worden.

Elk instrument wat geplaatst wordt moet worden geregistreerd. Het is hierbij belangrijk dat minimaal het soort instrument, de verblijfplaats (om in te vullen indien niet aan boord van een schip, maar bijv. onderweg), het KNMI-nr, serienummer en extra informatie (memoveld) opgeslagen worden.

9) De plaatsingsdatum van instrumenten op het schip moet bijgehouden worden.



Software Requirements Specification

De plaatsingsdatum van instrumenten moet opgeslagen worden. Dit houdt in dat wanneer een instrument, wat zich in de instrumenten database bevindt, wordt geplaatst dit aangegeven wordt in de database. De datum van plaatsing en of het een primaire druksensor betreft is hierbij van belang.

10) Buitenlandse schepen moeten ingevoerd kunnen worden.

Ook buitenlandse schepen moeten in het systeem opgenomen kunnen worden. Hierbij moeten echter andere gegevens worden opgeslagen dan bij Nederlandse schepen. Er gelden hierbij andere minimum verplichte velden welke te vinden zijn in **Hoofdstuk 5**.

11) Bezoeken van Nederlandse schepen moeten kunnen worden geregistreerd.

De bezoeken die bij Nederlandse schepen plaatsvinden moeten in het systeem worden opgeslagen. Het is hierbij belangrijk dat gegevens als de correctie, bijstelling en de data van het bezoek worden opgeslagen.

12) Bezoeken aan buitenlandse schepen moeten worden geregistreerd.

Ook bezoek aan buitenlandse schepen moet worden geregistreerd. Hier hoeft echter minder informatie bij worden opgeslagen.

13) De ingelezen data van de journaalbestanden moet gecontroleerd worden op fouten.

Data die ingelezen wordt moet door het systeem worden gecontroleerd op fouten. Vooral fouten in de datum/tijd groep zijn belangrijk. Aangezien het invoeren van de data op de schepen nog mensenwerk is, is het mogelijk dat bij het invoeren van de datum en tijd een fout wordt gemaakt. In het bestand moet de datum en tijd chronologisch zijn.

14) Fouten die gevonden worden in de journaalbestanden moeten door de gebruiker m.b.v. het programma hersteld kunnen worden.

Fouten die het systeem vindt moeten worden aangegeven. Hierna moet de gebruiker in staat zijn om deze fouten te herstellen. Het zal helaas niet mogelijk zijn dit automatisch te doen omdat de goede waarde van veel factoren afhankelijk is en men dit toch bij menselijk handelen wil houden. De verbeterde data moet weer worden opgeslagen op het opslag medium.

15) De data moet in verschillende rapporten gezet worden en kunnen worden uitgeprint.

Van de data die in de database staat moeten verschillende rapporten gemaakt kunnen worden. Het is de bedoeling dat voor de meeste rapporten de gebruiker kan instellen over welke periode de rapporten moeten gaan. De rapporten moeten ook uitgeprint kunnen worden.

16) Verwerken MONTAB (Monthly table)

Statistische gegevens (kwaliteit en kwantiteit) over in Engeland ingewonnen Nederlandse scheepswaarnemingen worden maandelijks in een email gezet en naar de PMO's gestuurd. Dit worden de MONTAB gegevens genoemd en zullen verwerkt moeten worden. De gegevens m.b.t. luchtdruk en wind moeten opgeslagen worden in de database.

17) Verwerken Journaal diskette

De journaal diskette moet ingelezen en verwerkt worden. Echter moet dit op een zodanige manier gedaan worden dat het programma naast het inlezen van een diskette ook overweg kan met bijvoorbeeld een USB-stick of het lezen van een netwerk schijf. De data die ingelezen wordt moet worden gecontroleerd op fouten. De fouten die gevonden worden moeten worden aangegeven in het programma waarna de gebruiker deze kan wijzigen en opslaan.



Software Requirements Specification

18) Verwerken Nota's

De nota's dienen ook nog verwerkt te worden. Deze worden ingelezen via een .txt bestand en moeten daarna in de database geplaatst worden. Dit moet grotendeels automatisch gebeuren. Van deze nota's zullen ook rapporten gemaakt moeten worden. Voor meer informatie over deze rapporten zie **Hoofdstuk 5**.

19) Rapporten over schepen die langer dan een bepaalde datum geleden bezocht zijn moeten uitgedraaid kunnen worden.

Er moeten rapporten gemaakt worden van de schepen die langer geleden dan een door de gebruiker ingevoerde datum bezocht zijn.

Ook moeten er rapporten gemaakt worden over het aantal bezochte schepen en het aantal bezoeken. Hierbij is het van belang dat schepen meerdere keren bezocht kunnen zijn. Het onderscheid tussen bezoeken en bezochte schepen moet in het rapport terug zijn te vinden.

20) De nota's van satellietprovider "Stratos" komen als tekstbestand binnen.

Deze gegevens moeten worden ingelezen t.b.v. statistische berekeningen..

Deze gegevens worden gebruikt om (m.b.v koppeling met de Inmarsatnummers van de nederlandse schepen) het aandeel van de kosten van Nederlandse schepen te vergelijken met die van buitenlandse schepen. I.v.m. internationale afspraken over compensatie, krijgt het KNMI van Duitsland de kosten (gegenereerd door haar schepen) terugbetaald. Om het aandeel van "duitse kosten" te berekenen moet er een tabel komen waarin de inmarsatnummers ,namen en roepletters worden bijgehouden.

Middels rapporten moeten deze kosten berekend worden. Het KNMI stuurt elk jaar een rekening naar Duitsland. De duitse meteo dienst stuurt maandelijks een excel bestand welke de inhoud van de duitse tabel in het systeem moet vervangen. Dit om een up to date berekening mogelijk te maken. De rapporten moeten onderscheid maken tussen echte berichten, verzonden door het schip en zgn. "ontvangstbevestigingen". Deze zijn in de nota van elkaar te onderscheiden door te kijken of er sprake is van een e-mail adres als bestemming (obsvos@knmi.nl) of een inmarsat-c nr. Een inmarsatnr. als bevestiging betekent een ontvangstbevestiging verstuurd door Stratos naar het schip.

21) DMG bestanden moeten verwerkt kunnen worden.

De DMG bestanden die op een netwerkschijf geplaatst worden moeten in het systeem ingevoerd kunnen worden.

22 VOS bestanden moeten verwerkt kunnen worden.

De VOS bestanden (maandlogs uit het AOW systeem) die op een netwerkschijf geplaatst worden moeten in het systeem ingevoerd kunnen worden. Het AOW systeem is het systeem dat de waarnemingen van Stratos (satellietprovider) ontvangt en verwerkt. Met deze VOS bestanden kunnen de volgende waardevolle berekeningen worden uitgevoerd:

- Vergelijking met de notabestanden (aantal waarnemingen per schip hetzelfde?)
- Hoeveel waarnemingen (en van welke schepen) zijn niet via Stratos verstuurd maar rechtstreeks via de eigen mail van betreffende schepen (afzender is mail-adres i.p.v. 9-cijferig inmarsat nummer).

23) Koppeling met buitenlands systeem.

Men ziet de mogelijkheid om het systeem te koppelen aan buitenlandse databases. Hierdoor kan het systeem internationaal ingezet worden. Hoe dit precies moet gaan werken is nog onduidelijk waardoor voordat dit onderdeel geïmplementeerd kan worden eerst een onderzoek hierna gedaan moet worden.



Software Requirements Specification

Hoofdstuk 3: Niet functionele eisen

Naast functionele eisen zijn er bij een software product ook niet functionele eisen. Deze eisen beschrijven zaken als welke taal en aan welke performance het systeem moet voldoen. In dit hoofdstuk worden de niet functionele eisen behandeld.

3.1 Niet functionele eisen

1) Het systeem moet stabiel zijn.

Het systeem moet een fout kunnen verwerken en niet gelijk crashen. Hiervoor moet goede foutafhandeling verwezenlijkt worden.

2) Foutafhandeling

Het is de bedoeling dat wanneer het systeem niet de gewenste informatie krijgt hierover een foutmelding geeft. Het systeem mag hierbij niet blijven hangen. De applicatie geeft hierbij "warning messages" en blijft wel draaien. Indien het systeem in een situatie komt waarbij herstart moet worden, wordt een "Fatal error message" gegeven en het programma afgesloten.

3 het systeem is web based.

Het systeem moet via een browser draaien. Het voordeel hiervan is dat het systeem platform onafhankelijk is. Tevens kan, indien men dit wenst, het systeem vanuit huis benaderen via de browser.

4 Het systeem moet uiteindelijk op een LAMP omgeving gaan draaien.

Het is de bedoeling dat het systeem uiteindelijk op een LAMP omgeving komt te draaien. Hier moet tijdens de ontwikkeling van het systeem rekening mee gehouden worden.

5 Buitenlandse en Nederlandse schepen moeten gescheiden blijven.

Schepen uit het buitenland en schepen uit de Nederlandse vloot moeten gescheiden blijven. Deze schepen moeten dus een aparte tabel hebben en niet met elkaar verweven zijn.

6 De software moet zelfstandig taken uitvoeren.

De software moet in staat zijn zelfstandig taken uit te voeren zoals het controleren op fouten. Hier hoeft de gebruiker niet eerst toestemming voor te geven.

7 de gegevens uit de bestaande database moeten overgezet worden.

De gegevens die nu in een database staan moeten beschikbaar worden in de nieuwe database. Hier moet dus rekening mee gehouden worden bij het ontwerp van de nieuwe database.

8) Het systeem moet gebruik maken van PHP en MySQL

Doordat het systeem op een LAMP server gaat draaien zal het product ontwikkeld moeten worden in PHP. De database zal hierdoor ontwikkeld gaan worden met MySQL.

9) Het programma is Engelstalig

De taal waarin naar de gebruiker gecommuniceerd zal worden is Engels. De code zal ook geprogrammeerd worden door middel van Engelse naamgeving en het toevoegen van commentaar in het Engels.



Software Requirements Specification

10) Het systeem moet wereldwijd beschikbaar zijn.

Het systeem moet in de toekomst wereldwijd toegankelijk zijn. Hierdoor moet het afgeschermd worden met een gebruikersnaam en wachtwoord combinatie. In eerste instantie is dit onderdeel minder urgent aangezien het programma voorlopig intern gaat draaien.

11) Performance

Het programma moet redelijk vlot kunnen reageren op de actie van de gebruiker. Het is echter wel toegestaan dat de gebruiker een paar minuten moet wachten bij zware acties. Denk hierbij aan het inlezen en verwerken van een diskette. Voor de gebruiker moet in elk geval duidelijk zijn dat het systeem bezig is.

12) Systeem grootte

De database van het systeem zal redelijk groot worden aangezien gegevens eeuwig bewaard blijven. Het is dus belangrijk dat er nauwkeurig gezocht kan worden waardoor de performance goed blijven.



Software Requirements Specification

Hoofdstuk 4: Indeling via MoSCoW

Doormiddel van de MoSCoW methode (**Hoofdstuk 4.1**) worden de functionele specificaties opgedeeld in verschillende categorieën. Hierdoor wordt het makkelijker om de belangrijkste specificaties als eerste te implementeren.

4.1 De MoSCoW methode

De MoSCoW methode is uitermate geschikt om de functionele specificaties op te delen in verschillende categorieën. Deze methode kent de volgende categorieën:

1. **Must** have
2. **Should** have
3. **Could** have
4. **Would like to** have

Ad 1) Must have

Deze eisen moeten in het eindresultaat verwerkt zitten.

Ad 2) Should have

Deze eisen zijn zeer gewenst maar een eventuele vergelijkbare eigenschap is ook goed genoeg voor het eindresultaat.

Ad 3) Could have

Deze eis mag alleen aan bod komen indien er tijd genoeg aanwezig is en mag geen negatief effect hebben op de al geïmplementeerde functionaliteit.

Ad 4) Would like to have

Deze eis komt zeker niet in de projectduur aan bod maar kan voor in de toekomst interessant zijn om geïmplementeerd te worden.

Belangrijke eis bij deze methode is dat men er wel vanuit gaat dat minimaal de Must have's geïmplementeerd zijn aan het einde van het project. Tevens zijn de should have's of een vergelijkbare interpretatie daarvan gewenst.

4.2: Indeling

De functionele eisen uit **Hoofdstuk 2** zijn geïnterpreteerd en omgezet volgens de MoSCoW methode. Hierbij krijgen de requirements met de hoogste prioriteit voorrang. Sommige eisen met minder prioriteit krijgen toch voorrang omdat deze sterk samenhangen met requirements die wel een hoge prioriteit hebben.

4.2.1 Must have

De volgende functionele eisen zijn ingedeeld bij de must have's:

- Nieuwe schepen toevoegen.
- Nieuwe waarnemers toevoegen.
- Maatschappij toevoegen.
- Buitenlandse schepen toevoegen.
- Nederlandse bezoeken toevoegen.
- Buitenlandse bezoeken toevoegen.



Software Requirements Specification

- Journaals invoeren.
- Verzendingen naar schepen bijhouden.
- Soorten instrumenten bijhouden.
- Journaal diskette verwerken.
- Communicatiemiddelen en soorten bijhouden per schip.
- MONTAB gegevens verwerken.
- Nota gegevens verwerken.
- Gevonden fouten moeten door gebruiker kunnen worden hersteld.
- DMG bestand verwerken.
- VOS (AOW) bestand verwerken.
- Stratos Nota's verwerken.

Deze eisen zullen dus in elk geval in het programma aanwezig zijn.

4.2.2 Should have

De volgende functionele eisen zijn ingedeeld bij de *should have*s:

- Plaatsingsdatum instrumenten bijhouden.
- Versie en uitgifte datum op schepen bijhouden.
- Aantal waarnemingen bijhouden.
- Rapporten over laatste bezoekdatum moeten uitgedraaid kunnen worden.
- Rapporten over bezoeken en bezochte schepen over opgegeven periode.
- Data moet in rapporten gezet worden.
- Ingelezen data moet door het systeem gecontroleerd worden op fouten.

Deze functionele eisen zullen geïmplementeerd worden nadat de *must have*s geïmplementeerd zijn.

4.2.3 Could have

De volgende functionele eisen zijn ingedeeld bij de *could have*s:

- Koppeling met buitenlands systeem.

Indien er nog tijd beschikbaar is na het implementeren van de *should have*s zal aandacht aan deze eisen worden besteed.

4.2.4 Would like to have

Er zijn geen functionele eisen in de *Would like to have*s.



Software Requirements Specification

Hoofdstuk 5: Gegevens

In dit hoofdstuk worden de gegevens die voor het systeem van belang zijn beschreven. De elementen die in elk geval dan wel niet verplicht ingevoerd moeten worden, worden vermeld. Er wordt nog geen aandacht besteed aan hoe de gegevens gekoppeld gaan worden. Dit zal gedaan worden in het Technische ontwerp document. Tevens kunnen er later meer gegevens bijkomen indien dit gewenst blijkt te zijn bij het ontwerpen van de database.

Ook wordt er aandacht besteed aan de rapporten die gemaakt moeten worden en welke gegevens hierbij opgeslagen moeten worden. De functies die verder nog aanwezig moeten zijn in het systeem worden nauwkeuriger uitgelegd.

5.1 Waarnemer

Van de waarnemer zullen de volgende gegevens opgeslagen moeten worden. Niet alle gegevens zijn verplicht om in te vullen.

Gegevens	Verplicht	Betekenis
Naam	Ja	Naam waarnemer.
Rang	Ja	De rang waartoe men behoort.
Begin jaar	Ja	Jaar wanneer men begonnen is.
Maatschappij	Ja	De Maatschappij waarbij men in dienst is
Totaal aantal waarnemingen	Nee	Alle waarnemingen die men gedaan heeft.
Zilver jaar	Nee	Het jaar waarin men een zilveren oorkonde heeft gehad.
Zilver jaar aantal waarnemingen	Nee	Het aantal waarnemingen in het zilveren jaar.
Goud jaar	Nee	Het jaar waarin men een gouden oorkonde heeft gehad.
Goud jaar aantal waarnemingen	Nee	Het aantal waarnemingen in het gouden jaar.
Beloning jaar	Nee	Het jaar waarin men een beloning heeft gehad.
Beloning jaar aantal waarnemingen	Nee	Het aantal waarnemingen in het Beloningsjaar.
Soort beloning	Nee	Het soort beloning
Extra	Nee	Mogelijke aanvullende gegevens.
Pensioen/HIS jaar	Nee	Het jaar wanneer men met pensioen/HIS gaat.

5.2 Journaal

Bij de journaal registratie moet het mogelijk zijn om de volgende informatie op te kunnen slaan. Deze zijn allemaal verplicht. Opmerking: Waarnemer en journaal worden gebruikt om de bijdrage van waarnemers per journaal bij te houden.



Software Requirements Specification

Gegevens	Verplicht	Betekenis
Journaalnummer	Ja	Automatisch gegenereerd ID
Diskette	Ja	Is er een diskette aanwezig.
Datum ontvangst	Ja	Datum van ontvangst
Periode van	Ja	Begindatum waarnemingen
Periode tot	Ja	Einddatum waarnemingen
Totaal journaal	Ja	Totaal waarnemingen tussen begin en einddatum.

5.3 Maatschappij

Bij de maatschappij dient de volgende informatie ingevuld te kunnen worden. Niet alle informatie is verplicht voor een goede werking van het systeem.

Gegevens	Verplicht	Betekenis
Index	Ja	ID nummer
Naam	Ja	Naam maatschappij
Subnaam	Nee	Mogelijke subnaam
Bezoekadres	Nee	Adres voor bezoek
Bezoekadres Postcode	Nee	Postcode voor bezoek
Bezoekadres Plaats	Nee	Plaats voor bezoek
Postadres	Nee	Postadres
Postadres postcode	Nee	Postcode voor postadres
Postadres Plaats	Nee	Plaats van postadres
Postadres Land	Nee	Land behorend bij postadres
Contactpersoon #1 naam	Nee	Naam contactpersoon 1
Contactpersoon #1 telefoon	Nee	Telefoonnummer contactpersoon 1
Contactpersoon #1 fax	Nee	Fax contactpersoon 1
Contactpersoon #1 email	Nee	E-mailadres contactpersoon 1
Contactpersoon #2 naam	Nee	Naam contactpersoon 2
Contactpersoon #2 telefoon	Nee	Telefoonnummer contactpersoon 2
Contactpersoon #2 fax	Nee	Fax contactpersoon 2
Contactpersoon #2 email	Nee	E-mailadres contactpersoon 2

5.4 Schepen

Van de schepen dient in elk geval de volgende informatie opgeslagen te worden. Niet alles is verplicht voor een goede werking van het systeem.

Gegevens	Verplicht	Betekenis
Naam Schip	Ja	De naam van het schip
Roepletters	Ja	De roepletters behorende bij het schip
VOS id	Nee	Het Vos ID is de mogelijke fake callsign.
Soort Schip	Ja	Het soort schip.
VOSclim	Nee	Betreft het een VOSclim schip.
Maatschappij	Ja	Maatschappij van het schip
IMO-nummer	Ja	Het imonummer ook wel bekend als "chassisnummer"



Software Requirements Specification

Laatste bezoek	Nee	Datum laatste bezoek
Luchtdruk (hpa)	Nee	De luchtdruk die gemeten is bij het bezoek.
Plaats controle	Nee	De plaats waar gecontroleerd is
Bezoek door	Nee	Wie het schip bezocht heeft
Datum invoer	Ja	Datum van invoer.
Afvoer	Nee	Datum waarop het schip gestopt is met metingen.
Primaire Druksensor	Nee	Wat is het knmi-nummer van de evt. primaire druksensor (selecteren uit alle aan het schip gekoppelde instrumenten) Dit instrument wordt vergeleken met de controlebarometer!
KNMI-nr	Nee	Het bijbehorende KNMI nummer.
Serie-nr	Nee	Het serienummer van de druksensor
Scheepsbarometer	Nee	Is er een scheepsbarometer aanwezig.
Scheepsbarograaf	Nee	Is er een scheepsbarograaf aanwezig.
Correctie	Nee	Wat is de correctie.
Bijgesteld	Nee	Hoeveel is hij bijgesteld
Ooghoogte	Nee	Wat is de ooghoogte
Zeewaterthermometer	Nee	Is er een elektronische zeewaterthermometer aanwezig.
Anemometer	Nee	Is er een anemometer
Hoogte (meters)	Nee	Wat is de hoogte van de anemometer.
Locatie	Nee	Wat is de locatie van de anemometer
Extra	Nee	Mogelijke extra informatie.

Als het IMO-nummer de primary key gaat worden zal dit echter wel een verplicht veld worden.

5.5 Software

Van de software moeten de volgende gegevens kunnen worden opgeslagen. Niet alle gegevens zijn noodzakelijk voor een goede werking van het systeem.

Gegevens	Verplicht	Betekenis
Uitgifte datum	Ja	Datum waarop de software is uitgegeven.
Soort software	Ja	De soort software.
Versie	Ja	Het versienummer wat bij de desbetreffende release hoort.
Extra	Nee	Extra aanvullende informatie.



Software Requirements Specification

5.6 Verzending

Om de verzendingen bij te kunnen houden moeten de volgende gegevens ingevoerd kunnen worden. Niet alle gegevens zijn verplicht.

Gegevens	Verplicht	Betekenis
Verzenddatum	Ja	De datum waarop de spullen zijn verzonden
Via	Ja	Via wie is het verzonden
Thermometers	Nee	Wat is er verstuurd
Extra	Nee	Extra gegevens zoals wat er verzonden is.

5.7 Instrument

Bij de instrumenten moeten de volgende gegevens bijgehouden kunnen worden. Niet alle gegevens zijn verplicht voor een goede werking van het systeem.

Gegevens	Verplicht	Betekenis
Soort	Ja	Het soort instrument.
Merk	Nee	Het merk van het instrument.
KNMI-nr	Ja	Bijbehorend KNMI nummer (uniek).
Serie-nr	Nee	Het bijbehorende serienummer.
IJking geldig tot	Nee	De datum tot wanneer de ijking van het instrument geldig is.
Verblijfplaats	Nee	Actuele locatie van het instrument.
Extra	Nee	Mogelijke extra informatie.

5.8 Instrument t.b.v. schip

Indien een instrument geplaatst is op een schip moeten de volgende gegevens opgeslagen worden. Deze zijn allemaal verplicht

Gegevens	Verplicht	Betekenis
Geplaatst	Ja	Datum waarop het instrument geplaatst is.
Primaire druksensor	Ja	Betreft het de primaire druksensor?

5.9 Communicatie

Voor de communicatie dienen de volgende gegevens opgeslagen te kunnen worden. Niet alle gegevens zijn verplicht om in te vullen.

Gegevens	Verplicht	Betekenis
Datum	Ja	Datum waarop deze geplaatst is
Soort	Ja	Het soort communicatie (fax, immarsat-c)
Merk	Nee	Het merk van de communicatie



Software Requirements Specification

ID nummer ¹	Ja	Bijbehorend ID nummer
Extra	Nee	Mogelijke extra informatie
Primaire communicatie	Nee	Betreft het de primaire communicatie?

5.10 Buitenlandse schepen

Voor de buitenlandse schepen moeten verschillende gegevens ingevoerd kunnen worden. Niet alle gegevens zijn hierbij verplicht om in te vullen.

Gegevens	Verplicht	Betekenis
Naam schip	Ja	De naam van het schip.
Roepletters	Ja	De bijbehorende roepletters.
Soort schip	Ja	Het soort schip.
Maatschappij	Nee	Tot welke maatschappij het schip behoort.
IMO nummer	Nee	Het IMO nummer van het schip (uniek)
Laatste bezoek	Nee	Datum laatste bezoek door KNMI
Plaats controle	Nee	De plaats van dit bezoek.
Bezoek door	Nee	Door wie het bezocht is.
Datum invoer	Nee	Datum dat de gegevens hiervan ingevoerd zijn.
Afvoer	Nee	Datum waarop het weer afgevoerd wordt.
Correctie	Nee	Correctie die gedaan is.
Bijgesteld	Nee	Bijstellingen die gedaan zijn.
Ooghoogte	Nee	Ooghoogte.
Zeewater temperatuur	Nee	De zeewatertemperatuur bij dit bezoek.
Extra	Nee	Mogelijke extra gegevens.

5.11 Bezoek (NED)

Wanneer een bezoek heeft plaatsgevonden dienen de volgende gegevens opgeslagen te kunnen worden. Al deze gegevens zijn niet verplicht om in te vullen. Uiteraard is het wel gewenst zoveel mogelijk gegevens in te vullen die bekend zijn.

Gegevens	Verplicht	Betekenis
Laatste bezoek	Nee	Datum laatste bezoek.
Plaats controle	Nee	Plaats van bezoek
Bezoek door	Nee	Naam van persoon die het schip bezocht heeft.
Primaire druksensor soort	Nee	De soort primaire druksensor die aanwezig is.
Primaire druksensor knmi nr	Nee	Het knmi nummer van deze sensor
Primaire druksensor serie nr	Nee	Het serie nummer van deze sensor
Scheepsbarometer	Nee	Is er een scheepsbarometer

¹ Dit kan een nummer (Inmarsat C nr) of een emailadres zijn.



Software Requirements Specification

		aanwezig? (wordt automatisch op ja gesteld indien er geen knmi barometer als primaire sensor in geselecteerd).
Correctie	Nee	De correctie die gevonden is.
Bijgesteld	Nee	De bijstelling die gedaan is
Ooghoogte	Nee	Gemiddelde hoogte barometer boven zeeniveau.
Extra	Nee	Mogelijke extra informatie

5.12 Bezoek (buit)

Wanneer een bezoek wordt gebracht aan een buitenlands schip dienen andere zaken opgeslagen te worden dan bij een Nederlands schip. Bij een buitenlands schip is niks verplicht om in te vullen.

Gegevens	Verplicht	Betekenis
Laatste bezoek	Nee	Datum laatste bezoek
Plaats controle	Nee	Plaats van dit bezoek
Bezoek door	Nee	Naam van persoon die het schip bezocht heeft.
Correctie	Nee	Correctie die gevonden is
Bijgesteld	Nee	Bijgesteld ja/nee
Ooghoogte	Nee	Gemiddelde hoogte barometer boven zeeniveau.
Extra	Nee	Mogelijke extra informatie.

5.13 Rapporten

Er moeten ook een hoop rapporten gemaakt worden. Rapporten die gemaakt moeten worden zijn:

- 1) Schepen
- 2) Nederlands bezoek
- 3) Verzending
- 4) Software
- 5) Communicatie
- 6) Instrumenten
- 7) Instrumenten (alleen)
- 8) Naamschip nieuw en oud
- 9) Maatschappij
- 10) Buit_schepen
- 11) Buit_bezoek
- 12) Waarnemer
- 13) Journaal
- 14) Waarnemers per journaal
- 15) MONTAB
- 16) Phenomena (*niet zo belangrijk*)
- 17) Nota Nederland
- 18) Nota Buitenland



Software Requirements Specification

- 19) Nota alternatief (is Duitsland)
- 20) Nota overig
- 21) Nota totaal
- 22) Nota ned-pd

Meer informatie over de inhoud van de rapporten is te vinden in bijlage 4.

5.14 Lijsten

Ook dienen er lijsten gemaakt te worden. De lijsten zijn een directe weergave van de gegevens uit de database. Voor elke tabel moet dus een lijst gemaakt kunnen worden. Deze lijsten tonen de actuele inhoud van de database.

5.15 Statistieken

Er moeten ook statistieken geproduceerd worden. De statistieken geven weer wanneer een schip voor het laatst bezocht is en hoeveel schepen er bezocht zijn. De gebruiker moet kunnen instellen over welke periode de betreffende bezoeken/bezochte schepen weergegeven en berekend moet worden. Tevens moet worden geteld hoeveel actieve schepen er op het moment zijn (soortschip<>"HIS").

DMG

In de DMG moeten de RTWNDB en VOS gegevens verwerkt worden. Tevens moeten de volgende (extra) rapporten gemaakt kunnen worden:

- 1) RTW ontvangen totaal.
- 2) Nederlandse schepen via alle ontvangende Meteorodiensten
- 3) Nederlandse schepen via EHDB.
- 4) AOW Nederlandse schepen.
- 5) AOW verworpen berichten.
- 6) Buitenlandse schepen via alle ontvangende Meteorodiensten
- 7) Buitenlandse schepen via EHDB.
- 8) AOW alle schepen.

De inhoud van deze rapporten wordt beschreven in bijlage 5.

Phenomena

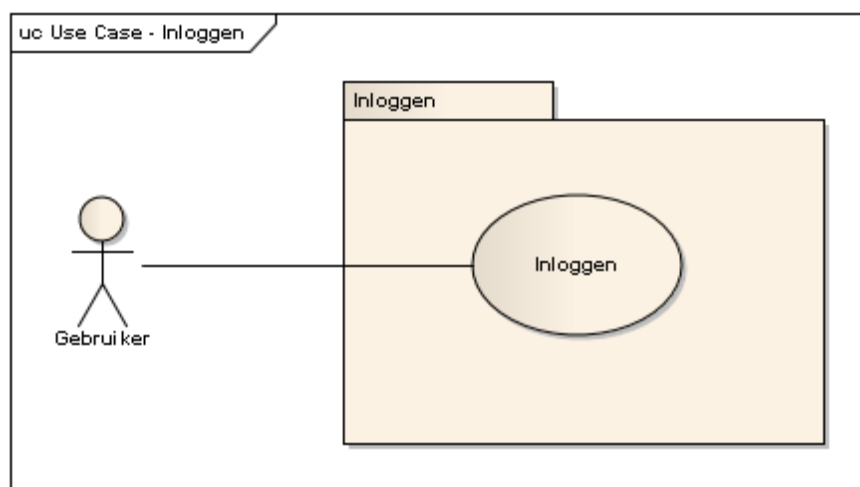
Phenomena is momenteel minder belangrijk. Dit onderdeel beschrijft de bijzonderheden die aan boord gebeuren. Denk hierbij aan spotten van een walvis, het zien van een vogel of het zien van een regenboog. Met deze data wordt niks gedaan en deze is dus ook niet/minder belangrijk voor het systeem.

Software Requirements Specification

Hoofdstuk 6: Use cases

In dit hoofdstuk worden de use cases beschreven die ontwikkeld zijn aan de hand van de functionele specificaties. Door middel van de use cases wordt duidelijk wat voor interactie de gebruiker met het systeem heeft. De use cases zijn opgedeeld in verschillende categorieën om het overzicht te behouden.

6.1 Use case inloggen

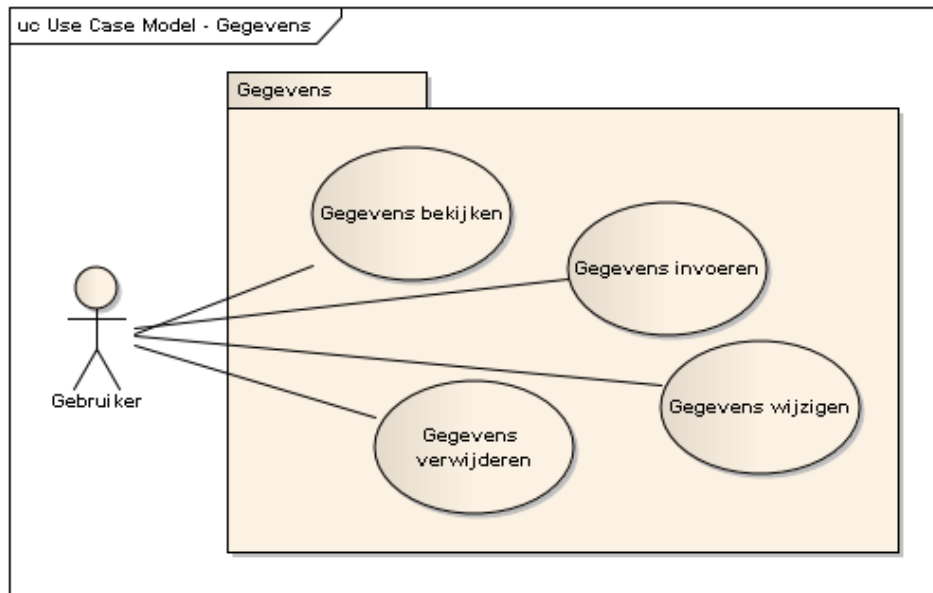


Figuur 2 Use case - inloggen

Naam	Inloggen
Aannamen	Systeem is niet geopend. Men is nog niet ingelogd
Beschrijving	(1) Gebruiker geeft zijn gebruikersnaam en wachtwoord op. (2) Systeem controleert gegevens. Gegevens fout dan treedt uitzondering [gebruiker niet bekend] op. Anders systeem opent openingsscherm
Uitzonderingen	[gebruiker niet bekend] Een melding wordt gegeven dat het systeem niet start omdat de gebruiker niet bekend is.
Resultaat	De gebruiker is ingelogd en het openingsscherm voor ingelogde gebruikers is zichtbaar.

Software Requirements Specification

6.2 Use case gegevens



Figuur 3 Use case - Gegevens

Naam	Gegevens invoeren
Aannamen	Gebruiker is ingelogd en ziet het hoofdmenu.
Beschrijving	(1) Gebruiker geeft aan gegevens in te willen voeren. (2) de gebruiker geeft aan welke gegevens hij wil invoeren. (3) De gegevens worden goed ingevoerd. Zo niet uitzondering [verkeerde invoer] treedt op. (4) De gebruiker bevestigt invoer.
Uitzonderingen	[verkeerde invoer] De gebruiker wordt erop geattendeerd dat er foutieve informatie in het formulier is ingevoerd.
Resultaat	De gegevens zijn in de database opgeslagen.

Naam	Gegevens wijzigen
Aannamen	Gebruiker is ingelogd en ziet het hoofdmenu
Beschrijving	(1) De gebruiker navigeert naar de gewenste informatie. (2) De gebruiker geeft aan deze informatie te willen wijzigen. (3) Het invoerscherm komt naar voren met de data die al ingevoerd is in de velden. (4) De gebruiker kan de data die gewijzigd mag worden wijzigen. (5) De gebruiker geeft aan de gegevens opnieuw op te willen slaan. (6) Het systeem geeft een bevestiging
Uitzondering	Geen.



Software Requirements Specification

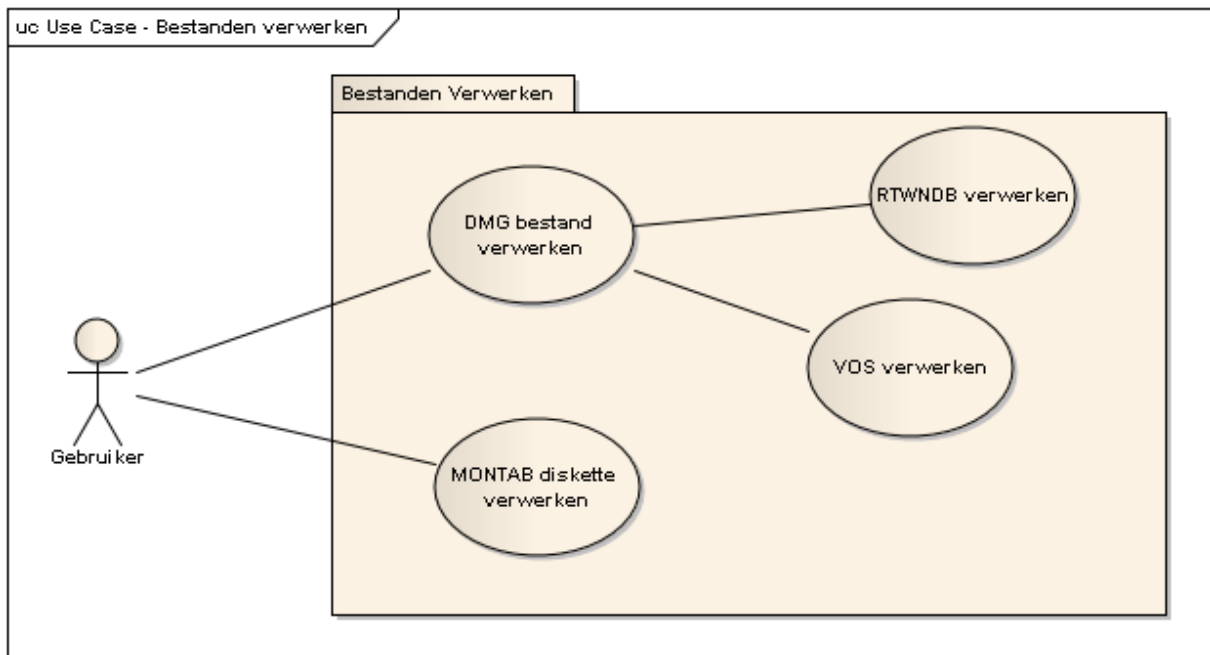
Resultaat	De gegevens zijn in de database veranderd.
-----------	--

Naam	Gegevens bekijken
Aannamen	Gebruiker is ingelogd en ziet het openingsscherm.
Beschrijving	(1) De gebruiker geeft aan gegevens te willen bekijken. (2) De gebruiker kiest de gewenste gegevens. (3) Het systeem toont de gegevens. Indien de gegevens niet beschikbaar zijn treedt uitzondering [niet beschikbaar] op.
Uitzondering	[niet beschikbaar] Het systeem geeft een melding dat de gegevens die opgevraagd worden niet beschikbaar zijn.
Resultaat	De gebruiker heeft inzage in de gegevens.

Naam	Gegevens verwijderen
Aannamen	De gebruiker is ingelogd en heeft het hoofdmenu voor zich.
Beschrijving	(1) De gebruiker gaat naar het gewenste formulier en zoekt hierin het record waarvan de gegevens verwijderd moeten worden. (2) De gebruiker drukt op verwijderen. (3) Het systeem verwijdert de data uit de database. (4) De gebruiker krijgt een bevestiging.
Uitzonderingen	Nee.
Resultaat	De gegevens zijn uit de database verwijderd.

6.3 Use case bestanden verwerken

Software Requirements Specification



Figuur 4 Use case - bestanden verwerken

Naam	DMG bestand verwerken
Aannamen	Gebruiker is ingelogd en ziet het hoofdmenu
Beschrijving	(1) gebruiker geeft aan gegevens in te willen lezen. (2) De gebruiker wijst het bestand aan. Indien dit bestand niet gevonden wordt treedt uitzondering [niet gevonden] in werking. (3) De gebruiker kiest voor het verwerken van een RTWNDB of VOS bestand. (4) Het systeem leest de file in. Bij onbekende gegevens treedt uitzondering [onbekend] in werking. Indien de gegevens ingelezen zijn krijgt de gebruiker een bevestiging.
Uitzonderingen	[niet gevonden] de gebruiker krijgt een melding dat het systeem het gewenste bestand niet kan vinden. [onbekend] Het systeem geeft een melding dat er onbekende gegevens in de file staan die niet geïnterpreteerd kunnen worden.
Resultaat	DMG gegevens staan in database opgeslagen.

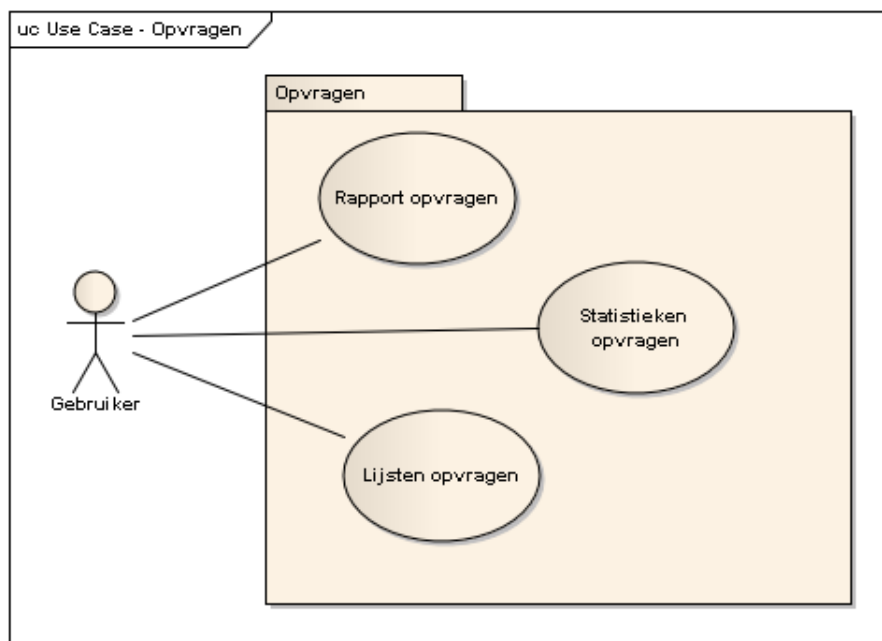
Naam	MONTAB diskette verwerken
Aannamen	Gebruiker is ingelogd en ziet het hoofdmenu.
Beschrijving	(1) Gebruiker geeft aan een diskette te willen verwerken. (2) Gebruiker kiest de file (van diskette) die verwerkt moet worden. (3) het systeem verwerkt de



Software Requirements Specification

	diskette. Indien de file niet gelezen kan worden treedt uitzondering [onleesbaar] op. (4) Het systeem verwerkt de file en controleert deze op fouten. Indien er een fout in het bestand zit wordt dit kenbaar gemaakt aan de gebruiker. (5) De gebruiker krijgt de gelegenheid fouten te herstellen (6) het bestand wordt opnieuw opgeslagen en indien er fouten aanwezig waren wordt weer begonnen met stap (3). (7) De gebruiker krijgt een melding dat het verwerken geslaagd is.
Uitzonderingen	[onleesbaar] De gebruiker krijgt een melding dat de file onleesbaar is.
Resultaat	De file is verwerkt en fouten zijn uit het bestand gehaald.

6.4 Use case opvragen



Figuur 5 Use case - Opvragen

Naam	Lijsten opvragen
Aannamen	De gebruiker is ingelogd en heeft het hoofdmenu voor zich.
Beschrijving	(1) De gebruiker kiest in het scherm om een lijst op te vragen. (2) De gebruiker kiest een tabel uit de database en bevestigt zijn invoer. (3) Het systeem haalt alle gegevens uit die tabel op en toont deze aan de gebruiker.
Uitzonderingen	Geen



Software Requirements Specification

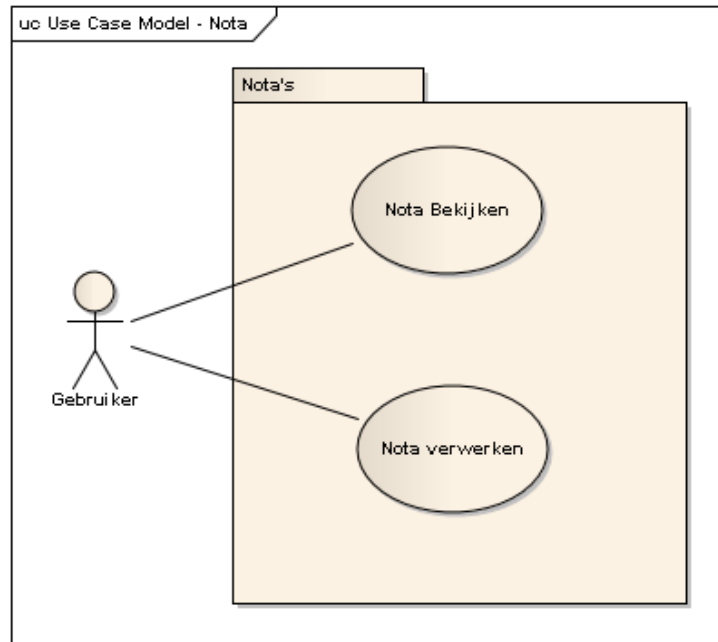
Resultaat	Men heeft een overzicht van de inhoud van een tabel.
-----------	--

Naam	Rapport opvragen
Aannamen	Gebruiker is ingelogd en ziet het hoofdmenu.
Beschrijving	(1) Gebruiker geeft aan een rapport te willen maken. (2) de gebruiker geeft aan welk rapport gemaakt moet worden over welke tijdsduur en waarop de gegevens gesorteerd moeten worden. Is de datum verkeerd dat treedt uitzondering [datum] op. (3) De gebruiker geeft aan klaar te zijn met het invullen van de vereiste gegevens
Uitzonderingen	[datum] Het programma laat weten dat over de gekozen data nog geen informatie beschikbaar is.
Resultaat	Het programma heeft een rapport gegenereerd met de gewenste gegevens.

Naam	Statistieken bekijken
Aannamen	De gebruiker is ingelogd en heeft het hoofdmenu voor zich.
Beschrijving	(1) De gebruiker geeft aan statistieken te willen inzien. (2) De gebruiker vult in hoe lang geleden een schip minimaal bezocht moet zijn om opgenomen te worden (begin en einddatum) Indien een datum ingevoerd wordt in de toekomst, treedt uitzondering [datum] op. (3) De gebruiker bevestigt zijn invoer. (4) Het systeem haalt de data uit de database.
Uitzonderingen	[datum] De gebruiker krijgt een melding dat de gekozen datum in de toekomst ligt en een andere datum gekozen moet worden.
Resultaat	Een rapport met de gewenste gegevens wordt getoond.

Software Requirements Specification

6.5 Use case nota



Figuur 5 Use case - Nota

Naam	Nota bekijken
Aannamen	De gebruiker is ingelogd en heeft het hoofdmenu voor zich.
Beschrijving	(1) De gebruiker kiest een van de beschikbare nota types. (2) De gebruiker bevestigt of verandert de dollar koers. (3) Het systeem haalt de informatie uit de database op en presenteert deze in een rapport
Uitzonderingen	Geen.
Resultaat	De gewenste nota wordt aan de gebruiker getoond.

Naam	Nota verwerken
Aannamen	De gebruiker is ingelogd en heeft het hoofdmenu voor zich.
Beschrijving	(1) De gebruiker geeft aan een nota in te willen voeren. (2) De gebruiker kiest het bestand met gegevens wat verwerkt moet worden. (3) Het systeem interpreteert de naam van het bestand en vraagt aan de gebruiker of de gekozen datum goed is. Zo nee uitzondering [datum] treedt op. (4) Het systeem leest de gegevens in. Indien er een fout optreedt treedt uitzondering [fout] op. (5) De gebruiker krijgt een bevestiging.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Software Requirements Specification

Uitzonderingen	[datum] De gebruiker krijgt een melding dat het systeem geen datum kan bepalen. [fout] De gebruiker krijgt een melding dat in het bestand zich data bevindt die niet goed geïnterpreteerd kan worden.
Resultaat	De nota is in het systeem ingevoerd.



Software Requirements Specification

Hoofdstuk 7: Planning

Dit hoofdstuk geeft een gedetailleerd overzicht van de lange termijn planning. Deze planning is nu mogelijk doordat de functionele specificaties en de software requirements bekend zijn. Dit hoofdstuk behandelt en benoemt de iteraties die doorlopen gaan worden.

Week	Activiteiten
Week 6 (15-03-10/19-03-10)	Afmaken ontwerp
Week 7 (22-03-10/26-03-10)	Iteratie 1
Week 8 (29-03-10/02-04-10)	Iteratie 1
Week 9 (05-04-10/09-04-10)	Iteratie 2
Week 10 (12-04-10/16-04-10)	Iteratie 2
Week 11 (19-04-10/23-04-10)	Iteratie 2
Week 12 (26-04-10/30-04-10)	Iteratie 3
Week 13 (03-05-10/07-05-10)	Iteratie 3
Week 14 (10-05-10/14-05-10)	Iteratie 3
Week 15 (17-05-10/21-05-10)	Iteratie 4
Week 16 (24-05-10/28-05-10)	Iteratie 4
Week 17 (31-05-10/04-06-10)	Testen/afronden

Als eerste worden de must haves van de functionele specificaties uitgevoerd waarna over gegaan zal worden op de should haves. Indien er meer tijd beschikbaar is dan gepland worden de could haves uitgevoerd. De iteraties zullen de volgende elementen bevatten:

Iteratie 1:

Deze iteratie begint met het ontwikkelen van de ontworpen database. De database vormt een belangrijk onderdeel van het systeem en moet dus zo vroeg mogelijk ontwikkeld worden.

- Nieuwe schepen toevoegen.
- Nieuwe waarnemers toevoegen.
- Maatschappij toevoegen.
- Buitenlandse schepen toevoegen.
- Nederlandse bezoeken toevoegen.
- Buitenlandse bezoeken toevoegen.
- Journaals invoeren.
- Verzendingen naar schepen bijhouden.
- Soorten instrumenten bijhouden.
- Instrumenten aan boord van schepen bijhouden
- Communicatiemiddelen schepen bijhouden

Iteratie 2:

- Journaal diskette verwerken.
- MONTAB gegevens verwerken.
- Gevonden fouten moeten door gebruiker kunnen worden hersteld.
- Plaatsingsdatum instrumenten bijhouden.
- Versie en uitgifte datum op schepen bijhouden.
- DMG bestand verwerken.



Software Requirements Specification

Iteratie 3:

- Stratos notabestanden inlezen.
- Nota's verwerken.
- Data moet in rapporten gezet worden.
- Ingelezen data moet door het systeem gecontroleerd worden op fouten.

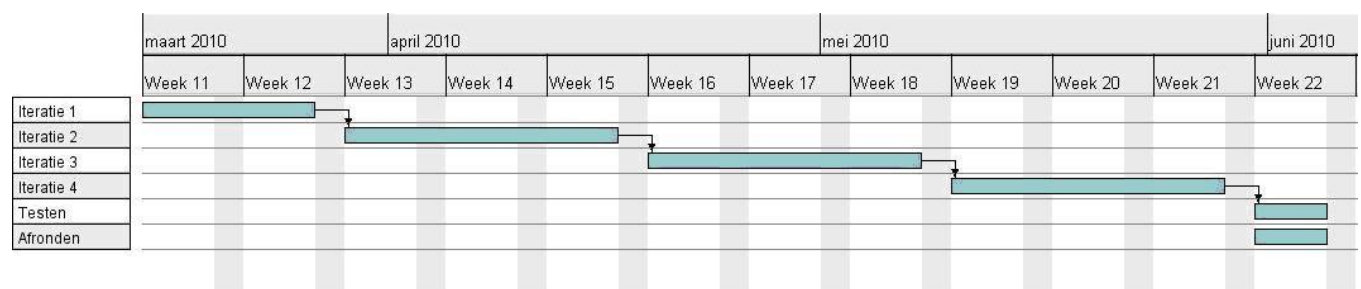
Iteratie 4:

- Rapporten over laatste bezoeksdatum en aantal actieve schepen moeten uitgedraaid kunnen worden.
- Aantal waarnemingen bijhouden.

Het is mogelijk dat na elke iteratie een paar kleine wijzigingen plaats vinden in de planning. Na elke iteratie wordt met de toekomstige gebruikers gekeken of de gerealiseerde functionaliteit naar wens is. Mogelijk commentaar/ aanvullingen worden dan meegenomen in de volgende iteratie.

Gantt visualisatie

Doormiddel van een gantt grafiek wordt de planning nogmaals visueel weergegeven.



Figuur 6 Gantt grafiek planning



Software Requirements Specification

Bijlage 1: Interview met Peter Westenbrink

Relatie met systeem

De data die binnenkomt wordt gecontroleerd op fouten en vervolgens gecorrigeerd. Als de data vrij van fouten is wordt deze toegevoegd aan de database. Vervolgens wordt eens in de zoveel tijd gekeken welke mensen het meest gelogd hebben en in aanmerking komen voor een medaille. De data wordt ook in verschillende rapporten uitgeprint en handmatig vergeleken.

Uitkomst

Tijdens dit interview kwam naar voren dat er veel rapporten gegenereerd moeten worden uit de database.

Er is informatie nodig voor het uitzoeken van de personen die een medaille krijgen uitgereikt en tevens voor de mensen die een oorkonde krijgen. Al deze informatie staat opgeslagen in de database en dient dus in rapporten omgezet te worden.

De data die ingelezen wordt, moet ook gecontroleerd worden op fouten. Een groot stuk fout afhandeling dient dus ook onderdeel van het systeem te gaan worden. Zo moet het systeem dubbele records verwijderen en controleren of de data chronologisch oploopt betreffende datum tijd.

Het is echter niet mogelijk om automatisch de fouten records te verbeteren naar de juiste waarde. De waarde van het record is afhankelijk van de omliggende records en een menselijke interpretatie hiervan. Dit kan dus beter door mensen dan door een PC gedaan worden en men wil dit dan ook handmatig blijven doen.

Tevens wordt er data ingelezen die bevat hoeveel er betaald moet worden voor het versturen van de data. Deze gegevens moeten ook in de database ingelezen worden.

De kapiteins krijgen het totaal aantal waarnemingen op hun naam voor de medaille registratie. Dit gebeurt dus ook als de kapitein zelf amper registreert en de bemanning dit wel veel doet. In de waarnemingen log kunnen meerdere kapiteins voorkomen aangezien over een grotere tijdsduur niet altijd dezelfde kapitein aan boord is. Het aantal waarnemingen wordt dan eerlijk verdeeld over de kapiteins. Onafhankelijk van hoeveel elke kapitein heeft gelogd wordt dus aan elke kapitein evenveel log registraties toegekend.

Er zijn tevens een hoop handelingen die met de hand gebeuren die meer geautomatiseerd kunnen worden. Dit kan een hoop werk schelen. Zo wordt nu met de hand in de file opgezocht waar de fouten zitten. Een file kan duizenden regels bevatten. Het opzoeken van fouten zou daarom meer geautomatiseerd kunnen worden. In de zin van dat de computer zelf naar de regel gaat die fout is. Zo kan de computer aangeven waar de fout zit en eventueel indien dit mogelijk is een suggestie geven.

De data wordt nu ingelezen doormiddel van het uitlezen van floppy's. Men kan echter wel zeggen dat de floppy's binnen aanzienbare tijd zullen verdwijnen een meer universele manier van inlezen die niet alleen van floppy's kan inlezen is dus gewenst. De gecorrigeerde data moet weer terug geschreven worden op het gewenste medium.



Software Requirements Specification

Bijlage 2: Interview met Rene Rozeboom

Relatie met systeem

Rapporten maken en uitprinten en gegevens in voeren in de database. Het systeem wat nu bestaat is niet in staat goed rapporten te maken. Hierdoor worden veel rapporten nu alsnog met de hand doormiddel van kopiëren/plakken gemaakt.

Uitkomst

Het systeem moet de mogelijkheid krijgen om data goed in te kunnen voeren. Bij schepen is het niet ongebruikelijk dat de naam veranderd. Hierdoor moet ook een tabel met oude namen van het schip bijgehouden worden.

Er moeten rapporten uitgeprint kunnen worden die verschillende informatie bevatten. De gebruiker moet kunnen selecteren over welke tijdsduur de gegevens worden gepresenteerd.

De buitenlandse schepen moeten niet tussen de Nederlandse schepen komen in de database. Men wil dit fysiek gescheiden houden. En wil dus het liefst een aparte tabel voor Nederlandse en voor buitenlandse schepen.

Duitsland heeft geen eigen satelliet ontvangst station die de gegevens van schepen ontvangt. Dit gebeurt daarom ook via Nederland. Hierdoor is het belangrijk om te weten welke buitenlandse schepen hun data naar Nederland verzenden en wat hiervan de kosten zijn. Daarnaast, specifiek voor de Duitsers is er een extra tabel toegevoegd met gegevens van Duitse schepen. Deze wordt m.b.v. informatie in excel formaat regelmatig up to date gehouden. M.b.v. deze alternatieve lijst kan specifiek het aandeel in de kosten, gegenereerd door Duitse schepen worden berekend. Dit kan dan in rekening worden gebracht aan de Duitsers.

Bij het berekenen van de nota's wordt ook het bedrag van de "message is delivered" mail meegenomen. Men wil echter dat schepen dit uitzetten omdat het veel geld kost. Het is daarom ook handig om dit los te koppelen van het uiteindelijke bedrag zodat gezien kan worden wat een mogelijke besparing is. Het werkelijk aantal gestuurde meteorologische berichten kan zo worden vergeleken met de aantallen uit VOS (AOW). Het verschil in aantallen wordt dan veroorzaakt door:

- Verworpen berichten (fout gecodeerd en daardoor verworpen door AOW)
- "Message delivered confirmation" berichten
- Door schepen rechtstreeks via eigen mail naar AOW verstuurd berichten (staan niet op de nota van Stratos omdat de schepen zelf voor deze berichten betalen).

Rapporten van VOS (AOW) kan ook onderscheid maken tussen via Stratos ontvangen berichten en rechtstreeks van schepen ontvangen berichten. Zo is er door combinatie tussen DMG (RTWNDB)-data, Nota data -en VOS (AOW) data veel te vertellen/controleren.

Men zou graag een rapport zien waarin de Nederlandse inspecties van de VOS vloot staan vermeld. Hierbij wil men graag weten hoeveel schepen bezocht zijn en hoeveel bezoeker er totaal zijn geweest (een schip kan meerdere keren bezocht worden). Tevens wil men graag weten hoeveel buitenlandse schepen op verzoek zijn bezocht.



Software Requirements Specification

Schepen kunnen regelmatig van naam veranderen. Hiervoor moeten de verschillende namen opgeslagen worden. De vraag is nog hoe de naamsverandering in het systeem doorgevoerd moeten worden. Immers bezoeken zijn gepleegd bij het schip onder de oude naam. Het is denkbaar dat men dit wil blijven weten. Echter instrumenten onder de oude naam zitten nu gewoon op het schip bij de nieuwe naam. Hoe dit precies opgelost moet worden moet nog bekeken worden.

Via Montab worden de e-mails met (van Nederlandse schepen afkomstige) scheepswaarnemingen ingevoerd die de Engelsen binnen hebben gekregen.

Men zou graag wat dynamische queries zien in het systeem. Men ziet dit als volgt: Er kan een rapport opgesteld worden waarin men zelf kan kiezen tussen welke tijdsspanne de gegevens moeten worden weergegeven. Tevens kan men in sommige gevallen nog zeggen op welke kolom gesorteerd moet worden.

Statistieken moeten kunnen worden omgezet in een rapport. De gebruiker moet het liefst zelfstandig de tijd in kunnen stellen waarover dit rapport gaat. Statistieken gaan over hoelang geleden een schip bezocht is. Een interface met alles na *datum* of tussen twee data in lijken dan voor de hand liggende keuzes.

De vervangingsdatum van instrumenten en de geldigheid van de ijktermijnen is belangrijk om bij te houden. Een tabel waarin alle verschillende instrumenten staan is ook gewenst.

Gegevens moeten ingevoerd, gewijzigd en bekeken kunnen worden. Als men gegevens bekijkt mag het niet mogelijk zijn ze te wijzigen.

Het moet mogelijk zijn door de verschillende bezoekrapporten van één bepaald schip te bladeren.

Om nota's te verwerken moet steeds de dollarkoers ingevuld worden, waarna dit goed omgerekend wordt. Het zou handig zijn als de laatste dollarkoers in het systeem blijft staan zodat niet telkens dezelfde koers ingevuld moet worden als meerdere gegevens op een dag ingevuld worden. Zelfs als men een nota van een bepaalde maand heeft verwerkt, moet men namelijk voor het verwerken van die nota t.b.v. de alternatieve (duitse) lijst de dollarkoers opnieuw ingeven. Dit terwijl het om dezelfde maand gaat en de dollarkoers dus zeker niet is verandert hierbij.

Momenteel worden niet alle gegevens op de juiste manier aan elkaar gekoppeld. Dit moet in de nieuwe database gaan veranderen.



Software Requirements Specification

Bijlage 3: Vergadering

Om ideeën voor het nieuwe systeem te krijgen is er een vergadering geweest met de belanghebbende. Deze vergadering heeft zo snel mogelijk plaats gevonden op donderdag 18-02-2010. Uit deze vergadering zijn de volgende zaken naar voren gekomen:

Het systeem wat nu gebruikt wordt is niet volledig, zeer ongebruiksvriendelijk, bevat veel fouten en is onbetrouwbaar.

Het nieuwe systeem moet taken die nu nog door de gebruiker handmatige gedaan worden over nemen en automatiseren. Echter blijft er voor sommige foutcorrectie mensen werk nodig.

Er zijn verschillende gebruikers met verschillende wensen/eisen aan het systeem. Er zal dus goed gekeken moeten worden naar wat er in het systeem komt te zitten en of dit haalbaar is.

Het systeem is op te delen in verschillende stukken. Die enigszins overlappen maar grotendeels naast elkaar informatie uit de database halen.

Het systeem dat ontwikkeld moet gaan worden zal zeer veel data moeten bijhouden en verwerken. Elke gebruiker moet weer andere zaken met het systeem doen.

Het systeem wat ontwikkeld moet worden is zeer groot het idee is daarom ook om het op te splitsen in verschillende stukken.

Er is nu een slechte database die als primaire sleutel een zgn. unieknr gebruikt. Schepen worden regelmatig omgevlagd om economische redenen. Dit heeft tot gevolg dat op zijn minst de roepletters (Call Sign) verandert. Tevens kunnen schepen worden "verhuurd" aan andere maatschappijen waardoor ze (al dan niet tijdelijk) een andere naam krijgen. Een VOSID is geïntroduceerd om schepen onidentificeerbaar te maken om economische of veiligheidsredenen (kapingen). Sommige schepen willen een VOSID, maar lang niet alle. Als alle schepen een VOSID zouden willen zou deze ook gebruikt kunnen worden als primaire sleutel omdat een VOSID door het KNMI wordt uitgegeven en dus niet hoeft te veranderen. Aangezien roepletters veranderen en een VOSID niet door alle schepen gebruikt worden blijft het IMOnummer als enige een geschikte sleutel. Het gebruik van het unieknr zoals dat nu het geval is heeft het nadeel dat er geen schepen mogen worden verwijderd, aangezien dit unieknummer word berekend a.d.h. van het aantal records!

Er moet goed rekening worden gehouden met wat de gebruiker graag wenst en wat echt noodzakelijk is. Gebruikers hebben namelijk eisen en wensen en hier moet een duidelijk onderscheid tussen gemaakt worden om het systeem in eerste instantie zinvol te maken.

Er is afgesproken dat de verschillende belanghebbenden eerst onderling tot overeenstemming proberen te komen zodat tijdens de interviews niet door verschillende mensen tegenover gestelde dingen gezegd gaan worden. Dit zorgt ervoor dat het makkelijker wordt om de functionele specificaties te definiëren.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Software Requirements Specification

Bepaalde zaken zijn in het systeem minder belangrijk en kunnen daarom later uitgevoerd worden. Over de internationalisering van het geheel wordt nog nadacht.



Software Requirements Specification

Bijlage 4: Rapporten

Schepen

Een rapport van de schepen moet uitgedraaid kunnen worden. Hierbij moet men kunnen kiezen of men wil sorteren op: Naam, roepletters of maatschappij. Tevens moet het mogelijk zijn om bijvoorbeeld alles op te vragen wat ligt tussen de beginletters A en D. De informatie die verder aanwezig moet zijn in het rapport is:

Naam	Roepletters	Maatschappij	IMO-nummer	Inmarsat-Cnr	Laatste bezoek
AALSMEERGRACHT	PCAM	SPL	9044748	430681210	8-9-09
ACHTERGRACHT	PCCL	SPL	8821802	424460110	7-10-09

Nederlands bezoek

Een rapport over de bezoeken van Nederlandse schepen. Hierbij moet men kunnen sorteren op Naam, Roepletters, bezoekdatum of maatschappij. Tevens moet het mogelijk zijn om bijvoorbeeld alles op te vragen wat ligt tussen de beginletters A en D. De informatie die verder aanwezig moet zijn in het rapport is:

Naam	Roepletters	Maatschappij	Soort Schip	Laatste Bezoek	Correctie	bijgeleid
AALSMEERGRACHT	PCAM	SPL	SEL	8-9-09	-0,7	Ja
AALSMEERGRACHT	PCAM	SPL	SEL	1-3-07	-0,7	Ja
ACHTERGRACHT	PCCL	SPL	SEL	7-10-09	0	Nee

Verzending

Dit rapport bevat de informatie over de verzendingen die naar schepen zijn gedaan. Hierbij kiest de gebruiker het soort schip en een andere kolom uit de tabel waarop gesorteerd moet worden. Tevens moet het mogelijk zijn om bijvoorbeeld alles op te vragen wat ligt tussen de beginletters A en D. De informatie die aanwezig moet zijn:

Naam	Roepletters	Maatschappij	Soort schip	Verzenddatum	Aantal thermometers
AALSMEERGRACHT	PCAM	SPL	SEL	15-11-2005	0
AALSMEERGRACHT	PCAM	SPL	SEL	9-9-2004	2
ACHTERGRACHT	PCCL	SPL	SEL	28-4-2008	2

Software



Software Requirements Specification

In dit rapport wordt de aanwezige software aan boord van het schip getoond. Er gelden dezelfde selectie criteria als bij *verzending*. De informatie die aanwezig moet zijn:

Naam	Roepletters	Maatschap	Soort schip	Uitgifte datum	Software	Versienr
AALSMEERGRACHT	PCAM	SPL	SEL	24-4-95	Turbo1	3.2
AALSMEERGRACHT	PCAM	SPL	SEL	1-10-94	Turbo1	3.0
ACHTERGRACHT	PCCL	SPL	SEL	14-3-96	Turbo1	3.3

Communicatie

Dit rapport toont de communicatie met het schip. Er gelden weer dezelfde selectie criteria als bij *verzending*. De informatie die aanwezig moet zijn:

Naam	Roepletters	Maatschap	Soort schip	Datum	Soort	ID-Nummer
AALSMEERGRACHT	PCAM	SPL	SEL	24-4-95	Inmarsat-A	1301505
AALSMEERGRACHT	PCAM	SPL	SEL	1-10-94	Inmarsat-C	43068121
ACHTERGRACHT	PCCL	SPL	SEL	14-3-96	Inmarsat-A	1300373

Instrumenten

Hier worden de aanwezige instrumenten aan boord van een schip getoond. Dezelfde criteria als bij *verzending* zijn van kracht. De informatie die aanwezig moet zijn:

Naam	Roepletters	Maatschap	Soort schip	Datum	Soort	KNMI-nr	Ijkdatum geldig tot
AALSMEERGRACHT	PCAM	SPL	SEL	8-9-2009	Barometer	01-04-004-	01-02-2011
AALSMEERGRACHT	PCAM	SPL	SEL	8-9-2009	Barograaf	01-04-001-	05-04-2010
ACHTERGRACHT	PCCL	SPL	SEL	7-10-2009	Barometer	01-04-004-	01-02-2010

Instrumenten (alleen)

Dit rapport toont per instrument op welke locatie zich dit bevindt. De informatie die aanwezig moet zijn per instrument:



Software Requirements Specification

Type	Merk	KNMI-nr	Serie-nr	Verblijfplaats	Extra	Jikdatum geldig tot
BAROGRAAF	Vaisala	01-04-001-012	Onbekend	JWS	Micro	01-02-2011
BAROGRAAF	FU	01-04-001-014	64890	Onbekend		05-04-2010
BAROMETER	FU	01-04-004-003	onbekend	kabinet		01-02-2010

Naamschip (oud/nieuw)

Hierin worden alle namen die een schip ooit gehad heeft vermeld. Dezelfde selectie criteria als bij *verzending* gelden. Informatie die aanwezig moet zijn:

Naam	Roepleetters	Oude namen	Oude roepleetters	Datum	Maatschappij
ARAMAC	PGDR	NEDLLOYD HOUTMAN	PGDR	25-9-1998	NLL
BERGE NORD	LADY5	BERGE NORTH	LADY5	3-3-2005	NEU
GLOBAL CARRIER	V2SP	TAIPAN SCAN	V2SP	3-5-2005	UNI

Maatschappij

Dit rapport toont de informatie over de maatschappijen die in het systeem aanwezig zijn. De informatie die aanwezig moet zijn:

Naam	Maatschappij	Adres	Postcode	Plaats
Anthony Veder Rederijzaken b.v.	ANT	Postbus 1159	3000 BD	Rotterdam
Maersk agencies	APM	Coloradoweg 50, kap. Kamer	3199 LA	Maasvlakte
Alpha Ship Nederland b.v.	ASN	Sint jansstraat 2	9712 JN	Groningen

Buitenlandse schepen



Software Requirements Specification

Dit rapport bevat informatie van de buitenlandse schepen. Dezelfde selectie criteria als bij *verzending* gelden. De informatie die beschikbaar moet zijn:

Naam	Roepletters	Maatschappij	Soort schip	Land	Laatste Bezoek
ACADIA FOREST	D5DI	NFA	SEL		8-1-1993
AKADEMIC GORBUNOV	UZMH	BSC	SEL	RUSLAND	13-4-1994
AL AWDAH	9KWA	KOTC	SEL	UK/US	30-6-1998

Buitenlands bezoek

Dit rapport bevat de informatie over buitenlandse bezoeken. Dezelfde selectie criteria als bij *verzending* zijn weer van kracht. De informatie die beschikbaar moet zijn:

Naam	Roepletters	Maatschappij	Soort schip	Laatste bezoek	Correctie	Bijgesteld
AL SAMIDOON	9KKF	KOTC	SEL	5-7-1999	0	Nee
AL SAMIDOON	9KKF	KOTC	SEL	15-6-1998	-0,1	Nee
AMANDA		V2EX	SEL	25-8-2008	-1	Ja

Waarnemer

Hier worden alle waarnemers gepresenteerd. Hierbij kan het soort bemanning gekozen worden. Men kan kiezen welke letters aanwezig moeten zijn. Ook moet het mogelijk zijn per jaar op te vragen hoeveel iemand gelogd heeft. Hierbij moet de volgende informatie beschikbaar zijn:

Naam	Rang	Maatschappij	Beginjaar	Aantal waarnemingen
Bakker J.B.	KAPITEIN	ANT	2004	390
Boer A.W.	KAPITEIN	ANT	1982	2099
Bos A.	KAPITEIN	ANT	1986	1574

Men moet deze gegevens kunnen opvragen voor:

- Kapitein
- Stuurman
- R/O



Software Requirements Specification

- LL
- ALL (Kapitein)
- ALL
- Goud
- Zilver
- Beloning

Gesorteerd op:

- Naam
- Maatschappij
- Aantal waarnemingen

Journaal

Hierbij moet de volgende informatie beschikbaar zijn:

Naam schip	Journaalnummer	Maatschappij	Soort schip	Periode van	Periode tot	Totaal journaals
AL SAMIDOON	19960713	ANT	SEL	13-7-96	18-2-98	345
AMANDA	20011302	ANT	SEL	13-2-01	15-2-02	342
AL AWDAH	19980412	ANT	SEL	12-4-98	14-3-99	12

Gesorteerd op:

- Naam
- Maatschappij
- Journaal nummer

Men moet kunnen kiezen over wat voor soort schip het gaat.

Waarnemers per journaal

In dit rapport worden de waarnemers per journaal getoond. Informatie die beschikbaar moet zijn:

Naam	Journaalnummer	Maatschappij	Soort schip	Periode van	Periode tot	Aantal waarnemingen
Bakker J.B.	19960713	ANT	SEL	13-7-96	18-2-98	345
Boer A.W.	20011302	ANT	SEL	13-2-01	15-2-02	342
Bos A.	19980412	ANT	SEL	12-4-98	14-3-99	12



Software Requirements Specification

Gesorteerd op:

- Naam
- Maatschappij
- Journaal nummer

MONTAB

Informatie die beschikbaar moet zijn in het MONTAB rapport:

Naam	Maand	Roepletters	Afwijking	Druk st dev	Aantal	Afwijking	Wind st dev	Aantal
AALSMEERGRACHT	9-2009	PCAM	-1,5	2,5	30	-2,2	2,1	30
AALSMEERGRACHT	8-2009	PCAM	0	0	0	0	0	0
ACHERGRACHT	9-2009	PCCL	-0,3	0,6	71	-0,1	1,9	71

Gesorteerd op:

- Naam schip
- Maatschappij

Phenomena

Uitprinten van de verschillende phenomena. De phenomena is momenteel minder belangrijk voor het systeem. Het rapport moet de verschillende bijzonderheden melden die aan boord van een schip zijn gezien.

Nota Nederland

Informatie die beschikbaar moet zijn:

Naam schip	Inmersat-C	Periode van	Periode tot	Aantal	USD	Euro
AALSMEERGRACHT	430681210	13-7-96	18-2-98	45	1.07	1.00

Berichten en ontvangstbevestigingen moeten te onderscheiden zijn en apart berekend worden.

Nota Buitenland

Informatie die beschikbaar moet zijn:



Software Requirements Specification

Naam schip	Inmarsat-C	Periode van	Periode tot	Aantal	USD	Euro
AMANDA	430531110	13-7-96	18-2-98	45	1.36	1.00

Berichten en ontvangstbevestigingen moeten te onderscheiden zijn en apart berekend worden.

Nota alternatief

Informatie die beschikbaar moet zijn:

Call Sign	Inmarsat -C	Periode van	Periode tot	Nummer	USD	Euro
FU	430681210	13-7-96	18-2-98	45	1.07	1.00

Berichten en ontvangstbevestigingen moeten te onderscheiden zijn en apart berekend worden.

Nota overig

Informatie die beschikbaar moet zijn:

Naam schip	Inmarsat-C	Periode van	Periode tot	Nummer	USD	Euro
AALSMEERGRACHT	430681210	13-7-96	18-2-98	45	1.07	1.00

Berichten en ontvangstbevestigingen moeten te onderscheiden zijn en apart berekend worden.

Nota totaal

Een totaal overzicht.

Nota ned-pd

Informatie die beschikbaar moet zijn:

Naam schip	Inmarsat -C	Periode van	Periode tot	Nummer	USD	Euro
------------	-------------	-------------	-------------	--------	-----	------



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Software Requirements Specification

AALSMEERGRACHT	430681210	13-7-96	18-2-98	45	1.07	1.00
----------------	-----------	---------	---------	----	------	------

Berichten en ontvangstbevestigingen moeten te onderscheiden zijn en apart berekend worden.



Software Requirements Specification

Bijlage 5: Rapporten DMG

RTW ontvangen totaal

In dit rapport worden de volgende elementen getoond:

Maand	CCCC	Aantal
07-2009	AMMC	108
07-2009	EGRR	108
07-2009	EHDB	3098
	Totaal	3314

Totaal aantal waarnemingen via EHDB: 8356

Totaal aantal waarnemingen via alle stations: 334063

Nederlandse schepen via alle statistieken & Nederlandse schepen via EHDB

In dit rapport worden de volgende elementen getoond:

Naam	Roepletters	Ontvangen van	Maand	Aantal	Vertraging	VOS-ID
AALSMEERGRACHT	PCAM	EHDB	07-2009	13	16	
			Totaal	13		
ACHTERGRACHT	PCCL	AMMC	07-2009	45	31	
ACHTERGRACHT	PCCL	AMMC	07-2009	13	31	
			Totaal	58		

AOW Nederlandse schepen

In dit rapport worden de volgende elementen getoond:



Software Requirements Specification

Naam	Roepletters	Inmarsat-C	Maatschappij	Aantal	VOS-ID
AALSMEERGRACHT	PCAM	424601910@	Spliethoff beheer b.v.	13	
AFRICABORG	PHMH	424503112@	Wagenborg Scheepvaart b.v.	17	
ALBLASGRACHT	PCIG	424538910@	Spliethoff beheer b.v.	4	

Dit rapport moet ook dezelfde gegevens kunnen weergeven voor alleen de data die via stratos is binnengekomen en via scheepsmail berichten ontvangen is.

AOW verworpen berichten

In dit rapport worden de volgende elementen getoond:

Naam schip	Roepletters	Inmarsat-C	Extra	Aantal
		421120139@	VERWORPEN	1
		421128084@	VERWORPEN	11
		421133576@	VERWORPEN	1

Dit rapport moet ook dezelfde gegevens kunnen weergeven voor alleen de data die via stratos is binnengekomen en via scheepsmail berichten ontvangen is.

Buitenlandse schepen via alle statistieken & Buitenlandse schepen via EHDB

In dit rapport worden de volgende elementen getoond:

Naam	Roepletters	Ontvangen van	Maand totaal	Vertraging
ARAMAC	PCAM	EHDB	13	16
			13	
GLOBAL CARRIER	PCCL	AMMC	45	31
	PCCL	AMMC	13	31
		Totaal	58	

AOW alle schepen

In dit rapport worden de volgende elementen getoond:



Software Requirements Specification

Roepletters	Inmarsat-C	Aantal
ZQCP3	423501720@	4
ZQDI9	423502010@	5
ZSAF	460101059@	3

Totaal:12

Dit rapport moet ook dezelfde gegevens kunnen weergeven voor alleen de data die via stratos is binnengekomen en via scheepsmail berichten ontvangen is.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp document

Naam: B.A.S. Wille
Studentnummer: 60114
Datum: 24-03-2010
Versie: 1.0



Ontwerp Document

Samenvatting

Na aanleiding van de functionele specificaties is een ontwerp gemaakt. Dit ontwerp bevat de elementen die in het systeem moeten zitten om aan de functionele specificaties te voldoen.

Als eerste is een database ontwikkeld. Deze database slaat de gegevens op van:

- Schepen
- Journaals
- Waarnemers
- Instrumenten
- Verzendingen
- Software
- Maatschappijen
- Bezoeken
- RTWNDB
- Inmarsat-C
- VOS
- Montab
- Contact personen
- Buitenlandse schepen
- Buitenlandse bezoeken

Naar aanleiding van de use-cases die gemaakt zijn bij de software requirements zijn sequeunce diagrammen gemaakt. Deze diagrammen beschrijven de precieze werking van het systeem.

Tevens is een klassendiagram opgenomen die de klassen beschrijft die gemaakt worden. Het systeem is zo opgebouwd dat er makkelijk dingen toegevoegd of gewijzigd kunnen worden zonder dat dit ingrijpende gevolgen heeft voor de rest van het systeem.

Er zal een aparte HTML file ontwikkeld worden voor de lay-out. Hierdoor kan de lay-out onafhankelijk van de functionaliteit gewijzigd worden. Hierbij zal tevens gebruik gemaakt worden van een CSS bestand.

Voor instellingen opslag zal gebruik gemaakt worden van XML die via de PHP DOM parser ingelezen zal worden.



Ontwerp Document

Inleiding

Dit document beschrijft het ontwerp van de software aan de hand van de eisen die zijn vastgelegd in het Software Requirements document.

In dit document zullen de mockups (**Hoofdstuk 1**) beschreven worden die ene globale indruk zullen geven over de lay-out van het product.

Het database ontwerp wordt besproken. Waarbij voor elke kolom tevens het datatype wordt vermeld (**Hoofdstuk 2**).

Ook wordt aandacht besteed aan de sequence diagrams die een diepere indruk zullen geven over hoe het systeem precies werkt. (**Hoofdstuk 3**)

Tevens zal het klassendiagram beschreven worden die een betere indruk van het systeem zal weergeven. (**Hoofdstuk 4**)

Er is ook aandacht voor het eventuele gebruik van XML, een HTML engine en CSS.



Ontwerp Document

Inhoudsopgave

Samenvatting	2
Inleiding	3
Inhoudsopgave	4
Hoofdstuk 1: mockup	6
Hoofdstuk 2: database ontwerp	8
2.1 ER model	8
2.2 Conceptuele schema	9
2.3 datatypes	9
2.3.1 <i>Journal</i>	9
2.3.2 <i>Observer journal</i>	10
2.3.3 <i>Observer</i>	10
2.3.4 <i>Old Names</i>	10
2.3.5 <i>Instrument</i>	10
2.3.6 <i>Instrument Sort</i>	11
2.3.7 <i>Shipment</i>	11
2.3.8 <i>Communication</i>	11
2.3.9 <i>Communication Sort</i>	11
2.3.10 <i>Software</i>	12
2.3.11 <i>Software sort</i>	12
2.3.12 <i>CompanyShip</i>	12
2.3.13 <i>Company</i>	12
2.3.14 <i>Contact</i>	12
2.3.15 <i>Visit</i>	13
2.3.16 <i>RTWNDB</i>	13
2.3.17 <i>Inmarsat-C</i>	13
2.3.18 <i>VOS</i>	13
2.3.19 <i>Ship</i>	14
2.3.20 <i>Observer Ship</i>	14
2.3.21 <i>Foreign_ ship</i>	14
2.3.22 <i>VisitForeign ship</i>	14
2.3.23 <i>Visit Foreign</i>	14
2.3.24 <i>Montab</i>	15
2.4 Normaliseren	15
2.4.1 <i>Eerste normaalvorm</i>	15
2.4.2 <i>Tweede normaalvorm</i>	16
Hoofdstuk 3: sequence diagrams	17
3.1 inloggen	17
3.2 Gegevens bekijken	18
3.3 Gegevens invoeren	19
3.4 Gegevens verwijderen	20
3.5 Gegevens Wijzigen	21
3.6 RTWNDB bestanden verwerken	22
3.7 Montab bestanden verwerken	23
3.8 VOS verwerken	24
3.9 Opvragen lijsten	24
3.10 Opvragen Rapporten	25
3.11 Opvragen Statistieken	26
3.12 Nota bekijken	27
3.13 Nota verwerken	28
Hoofdstuk 4: klassendiagram	29



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp Document

Hoofdstuk 5: HTML layout	30
Hoofdstuk 6: CSS	31
Hoofdstuk 7: XML	32
Bijlage 1: Conceptueel schema.....	33
Bijlage 2: klassendiagram.....	35

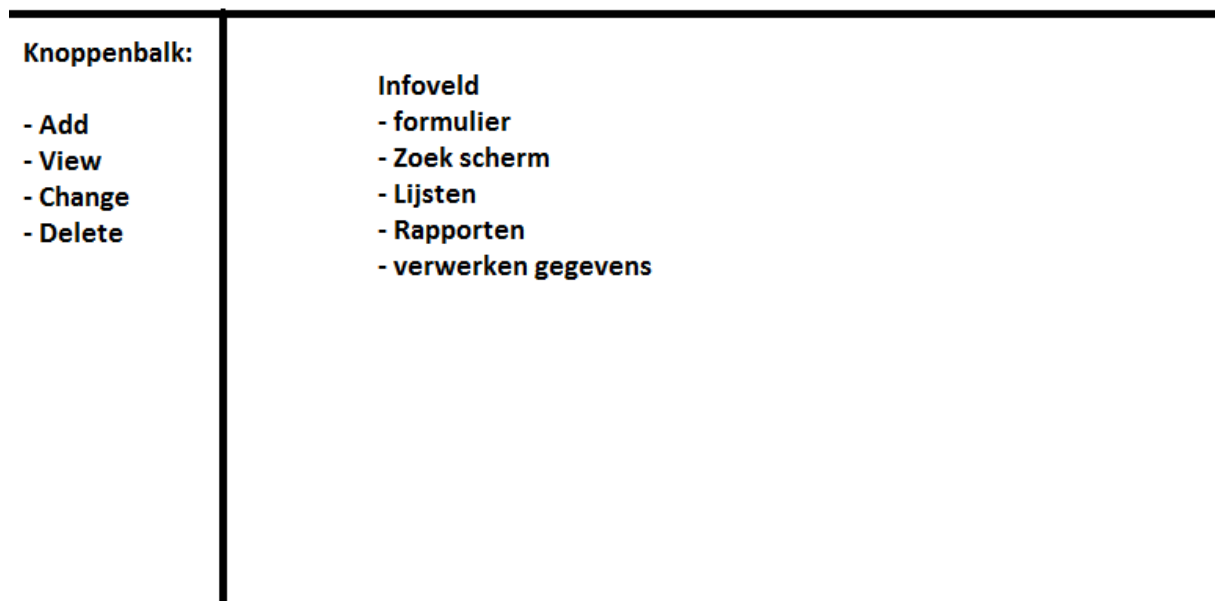


Ontwerp Document

Hoofdstuk 1: mockup

In dit hoofdstuk wordt een eerste indruk van de lay-out van de website gegeven. Hierdoor kan de gebruiker alvast zien hoe het geheel er zeer globaal uit gaat zien.

Banner



Figuur 1 Lay-out

In bovenstaand figuur 1 wordt een globale weergave gegeven van de lay-out van de website. Bij de banner zal een afbeelding komen die de functionaliteit van de applicatie zal verwoorden. Links bevindt zich de knoppenbalk. De gebruiker kan kiezen uit het menu Add, View, change en delete. Hierbij kan de gebruikers de gegevens bijwerken, bekijken, veranderen en verwijderen van:

- Instrument
- Instrument soort
- Verzending
- Communicatie
- Communicatie soort
- Software
- Software soort
- Bezoek
- Maatschappij
- Schip

Ook zal in dit menu de mogelijkheid komen om lijsten en rapporten op te vragen. Tevens wordt de mogelijkheid gegeven om de bestanden in te lezen.

In het informatie veld wordt de informatie getoond die de gebruiker selecteert uit het menu. Kiest de gebruiker bijvoorbeeld om een instrument te bekijken dan wordt hier eerst een zoek veld getoond. En vervolgens het gevonden instrument. Indien men



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp Document

bijvoorbeeld een schip wil invoeren wordt hier het formulier getoond waarin de gegevens kunnen worden ingevoerd die nodig zijn bij een schip.

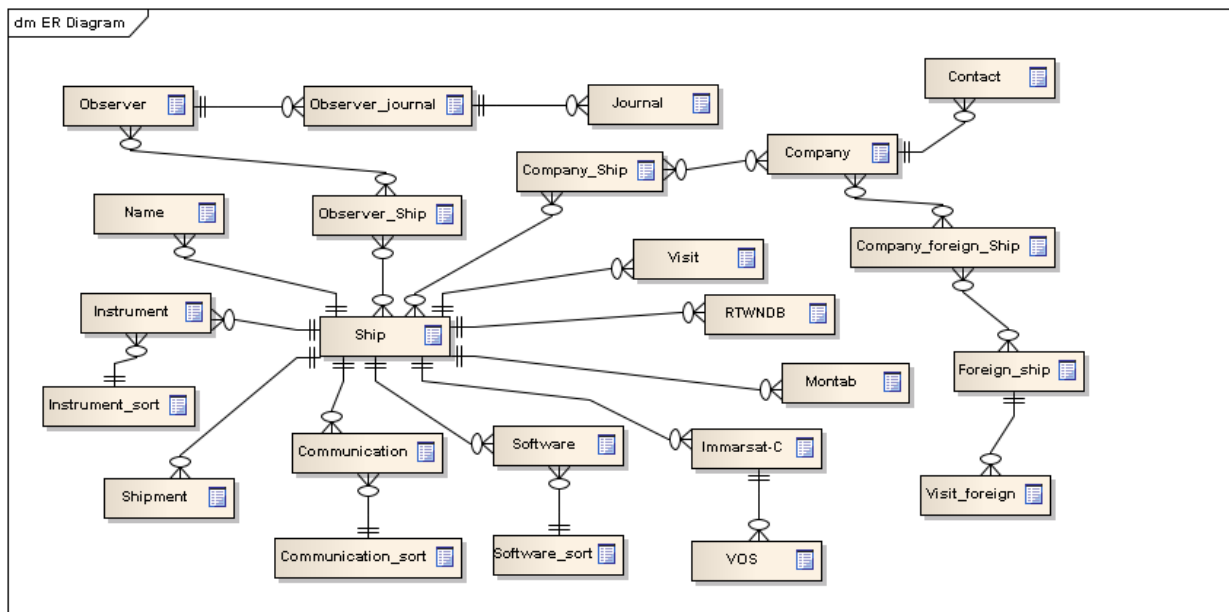
De website zal volledig Engelstalig zijn waardoor een mogelijke latere samenwerking met het buitenland makkelijker gerealiseerd kan worden. Hierbij zal de taal dan geen barrière meer vormen. Omdat de website volledig Engelstalig is, is besloten het commentaar bij de code ook Engelstalig te houden.



Ontwerp Document

Hoofdstuk 2: database ontwerp

2.1 ER model



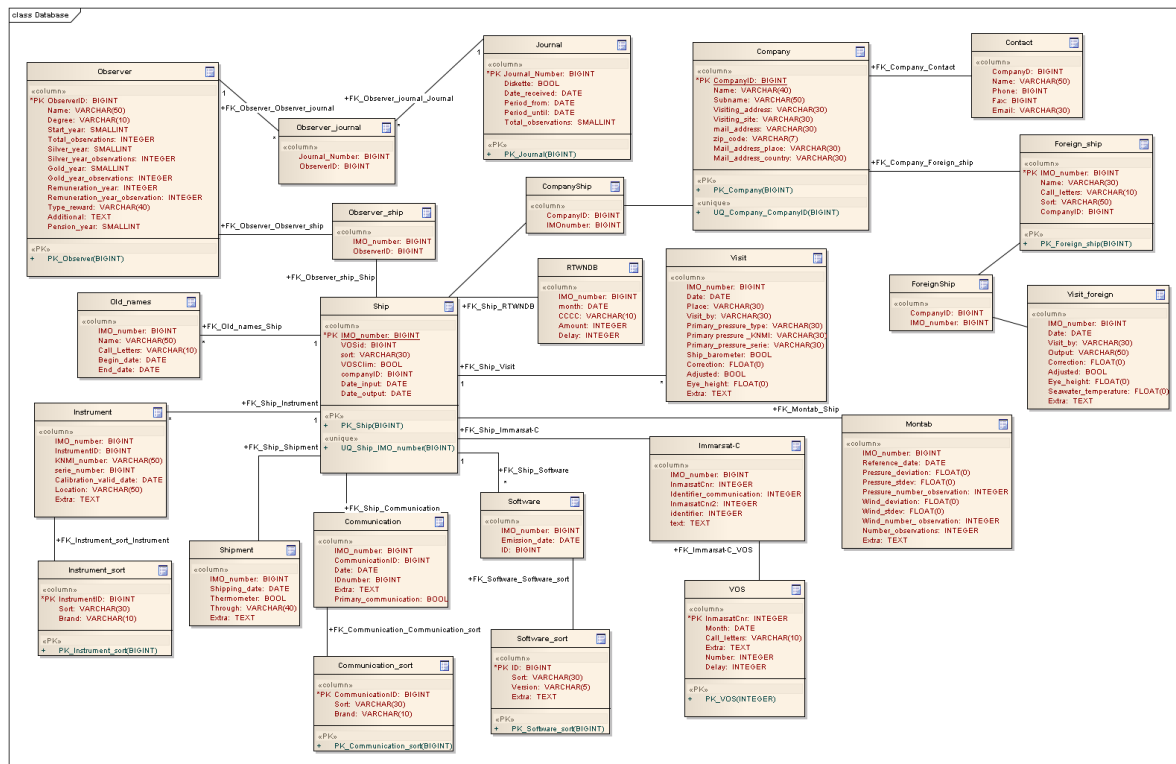
Figuur 2 ER model

Het ER model in figuur 2 geeft weer hoe de relaties tussen de verschillende tabellen in de database zijn. Zoals te zien is kan een schip over meerder zaken uit een tabel beschikken. Zo kan een schip meerdere instrumenten aan boord hebben. Echter een instrument kan zich maar op één schip tegelijk bevinden. Het soort instrument kan wel hetzelfde zijn bij verschillende instrumenten waardoor een soort instrument bij meerdere instrumenten kan voorkomen. Echter is het weer niet mogelijk dat een instrument meerdere soorten instrumenten beschikt.



Ontwerp Document

2.2 Conceptuele schema



Figuur 3 Conceptueel schema

Bovenstaand conceptueel schema toont die inhoud van de database (zie **bijlage 1** voor een duidelijkere weergave). Hierin worden de namen van de kolommen in de tabellen vermeld. In hoofdstuk 2.3 is per kolom en tabel te zien welk datatype gekozen is. In bovenstaand figuur worden de Primary key en foreign key per tabel kenbaar gemaakt.

2.3 datatypes

De verschillende velden hebben allen hun eigen data type. In deze paragraaf worden de datatype per tabel schematisch weergegeven.

2.3.1 Journal

Deze tabel bevat de informatie die nodig is voor de journaals.

Naam	Type
Journal number	BIGINT
Diskette	BOOL
Date received	DATE
Period from	DATE
Period until	DATE
Total observations	INT



Ontwerp Document

2.3.2 Observer journal

Deze tabel verbindt de informatie die hoort bij het journaal met de observer die het gedaan heeft.

Naam	Type
Journal number	BIGINT
ObserverID	BIGINT

2.3.3 Observer

Hierin worden de observers gemeld.

Naam	Type
IMO number	BIGINT
ObserverID	BIGINT
Name	VARCHAR
Degree	VARCHAR
Start year	SMALLINT
Total observations	INT
Silver year	SMALLINT
Silver year observations	INT
Gold year	SMALLINT
Gold year observations	INT
Remuneration year	SMALLINT
Remuneration year observations	INT
Type reward	VARCHAR
Extra	MEDIUMTEXT
Pension/HIS year	SMALLINT

2.3.4 Old Names

Deze tabel houdt de oude namen van schepen bij. Hierbij wordt ook de tijd vermeld dat de naam is gebruikt.

Naam	Type
IMO number	BIGINT
Name	VARCHAR
Call Letters	VARCHAR
BeginDate	Date
EndDate	Date

2.3.5 Instrument

Deze tabel bevat de instrumenten aan boord van de schepen. Tevens worden de bijbehorende nummers vermeld.

Naam	Type
IMO number	BIGINT
InstrumentID	BIGINT



Ontwerp Document

KNMI number	VARCHAR
Serie number	BIGINT
Calibration valid date	DATE
Location	VARCHAR
Extra	MEDIUMTEXT

2.3.6 Instrument Sort

Deze tabel bevat de instrumenten soorten die mogelijk zijn aan boord van de schepen.

Naam	Type
InstrumentID	BIGINT
Sort	VARCHAR
Brand	VARCHAR

2.3.7 Shipment

In deze tabel worden de verzendingen naar een schip bijgehouden.

Naam	Type
IMO number	BIGINT
Send date	DATE
Thermometer	BOOL
Thorough	VARCHAR
Extra	MEDIUMTEXT

2.3.8 Communication

In deze tabel wordt de communicatie die met een schip mogelijk is vermeld.

Naam	Type
IMO number	BIGINT
Date	DATE
CommunicationID	BIGINT
ID number	BIGINT
Extra	MEDIUMTEXT
Primary communication	BOOL

2.3.9 Communication Sort

In deze tabel wordt de communicatie typen die met een schip mogelijk zijn vermeld.

Naam	Type
CommunicationID	BIGINT
Sort	VARCHAR
Brand	VARCHAR



Ontwerp Document

2.3.10 Software

De software aan boord van de schepen worden in deze tabel vermeld.

Naam	Type
IMO number	BIGINT
Emmision date	DATE
SoftwareID	BIGINT

2.3.11 Software sort

De softwaresoort aan boord van de schepen worden in deze tabel vermeld.

Naam	Type
SoftwareID	BIGINT
Sort	VARCHAR
Version	VARCHAR
Extra	MEDIUMTEXT

2.3.12 CompanyShip

Deze tabel verbind de maatschappij aan haar schepen..

Naam	Type
CompanyID	BIGINT
IMONumber	BIGINT

2.3.13 Company

In deze tabel worden de verschillende maatschappijen vermeld.

Naam	Type
CompanyID	BIGINT
Name	VARCHAR
Subname	VARCHAR
Visiting address	VARCHAR
Visiting site	VARCHAR
Mail address	VARCHAR
Mail address place	VARCHAR
Mail address country	VARCHAR
Zip code	VARVARCHAR

2.3.14 Contact

Hierin worden de verschillende contactpersonen per maatschappij vemeld.

Naam	Type
CompanyID	BIGINT
Name	VARCHAR
Phone	BIGINT
Fax	BIGINT



Ontwerp Document

Email	VARCHAR
-------	---------

2.3.15 Visit

Hierin worden de bezoeken aan een schip vermeld.

Naam	Type
IMO number	BIGINT
Date	DATE
Place	VARCHAR
Visit By	VARCHAR
Primary pressure type	VARCHAR
Primary pressure knmi nr	VARCHAR
Primary pressure serie nr	VARCHAR
Ship barometer	BOOL
Correction	FLOAT
Adjusted	BOOL
Eye Height	FLOAT
Extra	MEDIUMTEXT

2.3.16 RTWNDB

Hierin worden de gegevens van de RTWNDB opgeslagen. Het IMO number verbindt deze gegevens aan een schip.

Naam	Type
Month	DATE
IMO number	BIGINT
CCCC	VARCHAR
Amount	INT
Delay	INT

2.3.17 Inmarsat-C

In deze tabel worden de inmarsat-c gegevens opgeslagen.

Naam	Type
IMO number	BIGINT
InmarsatCnr	INT
Identifier Communication	INT
InmarsatCnr2	INT
Identifier	INT
Text	TEXT

2.3.18 VOS

In deze tabel worden de VOS gegevens opgeslagen.

Naam	Type
Month	DATE



Ontwerp Document

Call Letters	VARCHAR
InmarsatCnr	INT
Extra	MEDIUMTEXT
Number	INT
Delay	INT

2.3.19 Ship

Deze tabel bevat de informatie van een schip.

Naam	Type
IMO number	BIGINT
VOSid	BIGINT
Sort	VARCHAR
VOSclim	BOOL
CompanyID	BIGINT
DateInput	DATE
DateOutput	DATE

2.3.20 Observer Ship

Deze tabel bevindt de gegevens aan een schip aan de waarnemers.

Naam	Type
IMO number	BIGINT
ObserverID	BIGINT

2.3.21 Foreign_ ship

In deze tabel worden de buitenlandse schepen opgeslagen.

Naam	Type
IMO number	BIGINT
Name	VARVARCHAR
CallLetters	VARVARCHAR
Sort	VARVARCHAR
CompanyID	BIGINT

2.3.22 VisitForeign ship

In deze tabel worden de buitenlandse schepen aan een bezoek gekoppeld.

Naam	Type
IMO number	BIGINT
CompanyID	BIGINT

2.3.23 Visit Foreign

In deze tabel worden de buitenlandse bezoeken aan schepen opgeslagen.



Ontwerp Document

Naam	Type
IMO number	BIGINT
Date	DATE
VisitBy	VARCHAR
Output	VARCHAR
Correction	INT
Adjusted	BOOL
EyeHeight	FLOAT
SeawaterTemperature	FLOAT
Extra	MEDIUMTEXT

2.3.24 Montab

In deze tabel worden de montab gegevens opgeslagen.

Naam	Type
IMO number	BIGINT
ReferenceDate	DATE
Pressure Deviation	FLOAT
Pressure stdev	FLOAT
Pressure number observation	INT
Wind deviation	FLOAT
Wind stdev	FLOAT
Wind number observation	INT
Number observations	INT
Extra	MEDIUMTEXT

2.4 Normaliseren

De database wordt omgezet in een normaalvorm. Er wordt express niet voor een al te hoge normaalvorm gekozen omdat hierdoor de performance van de database en dus de snelheid van de applicatie te ver achteruit gaat.

2.4.1 Eerste normaalvorm

Een relatie is in de eerste normaalvorm als hierbij alle attributen van deze relatie enkelvoudig zijn. Er komen dus geen repeating groups voor in de relatie. Tevens is elk attribuut atomair en bevat dus een enkele waarde. De database die ontwikkeld is staat in de eerste normaalvorm in de database komen:

- Geen repeating groups voor.
- Elk attribuut is atomair.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp Document

2.4.2 Tweede normaalvorm

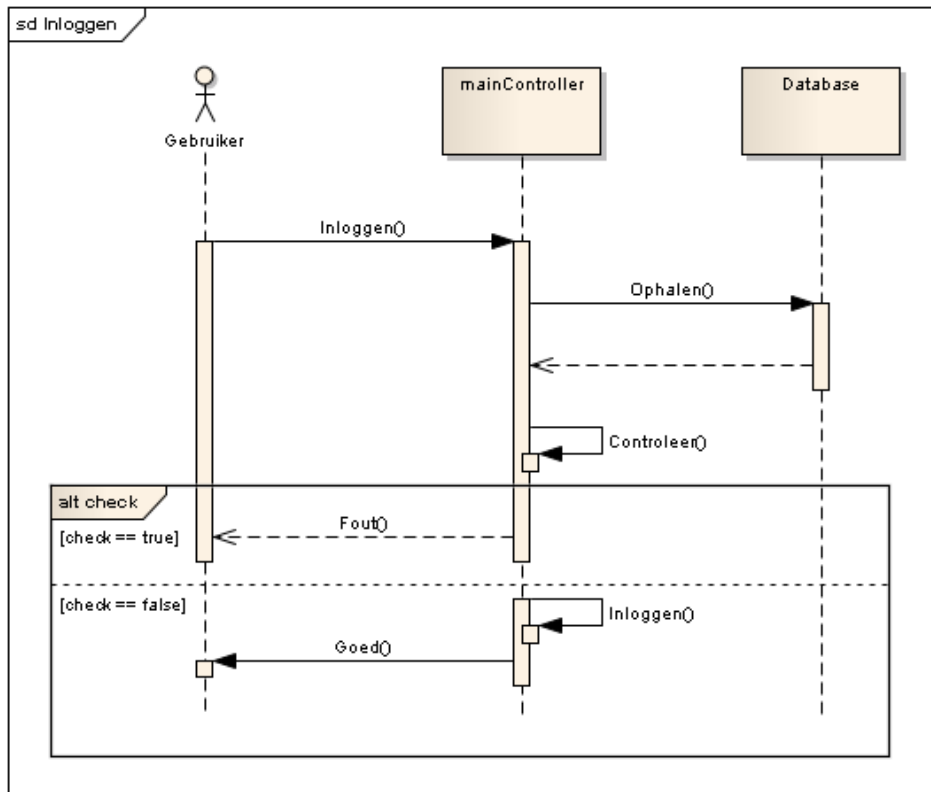
Een relatie is in de tweede normaalvorm indien de relatie in de eerste normaalvorm is en ieder niet sleutel attribuut volledig functioneel afhankelijk is van iedere kandidaat sleutel. De database die ontwikkeld is staat ook in de tweede normaalvorm aangezien voldaan wordt aan bovenstaande voorwaarden.

Ontwerp Document

Hoofdstuk 3: sequence diagrams

In dit hoofdstuk worden de beschreven usecases omgezet in sequence diagrams. Deze diagrammen geven een beter beeld over de werking van het systeem. De acties die het systeem zal uitvoeren worden in dit hoofdstuk behandeld.

3.1 inloggen

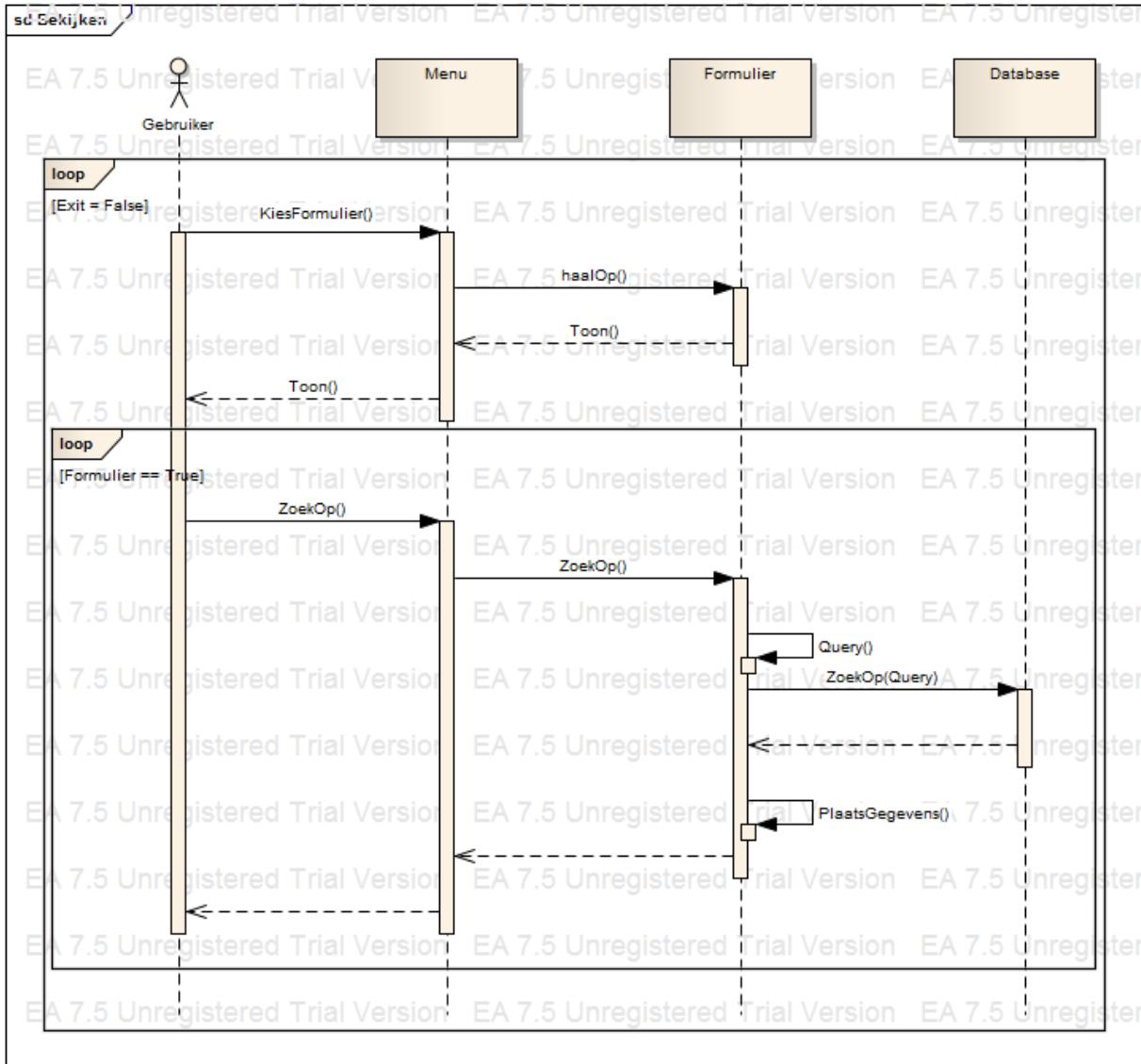


Figuur 4 Sequence Diagram Inloggen

De gebruiker geeft zijn gebruikersnaam en wachtwoord op. Deze worden vervolgens gecheckt in de database. Zijn de gegevens goed dan kan de gebruiker inloggen. Indien de gegevens fout zijn volgt een foutmelding.

Ontwerp Document

3.2 Gegevens bekijken



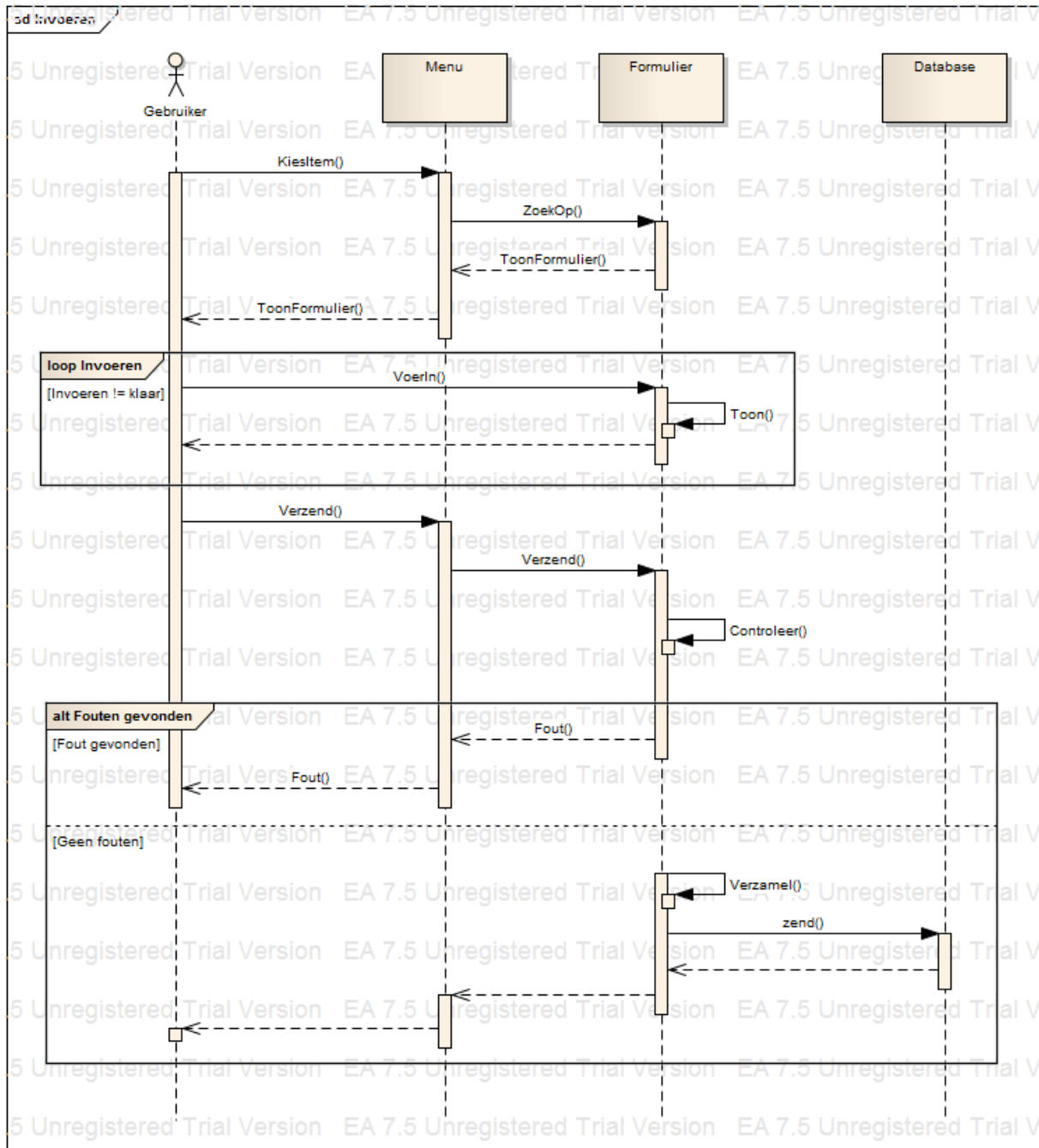
Figuur 5 Sequence diagram gegevens bekijken

De gebruiker geeft aan de mainController door welke gegevens bekeken moeten worden. De mainController stuurt naar de klassen die het formulier bevat van de gewenste gegevens een verzoek om het formulier te tonen. Deze klassen 'maakt' vervolgens het formulier en zendt dit naar de mainController waarna de gebruiker de gegevens kan opzoeken. Bij elke zoekactie zal een query door de klassen behorende bij het formulier uitgevoerd worden op de database om de juiste gegevens te tonen in het formulier.



Ontwerp Document

3.3 Gegevens invoeren



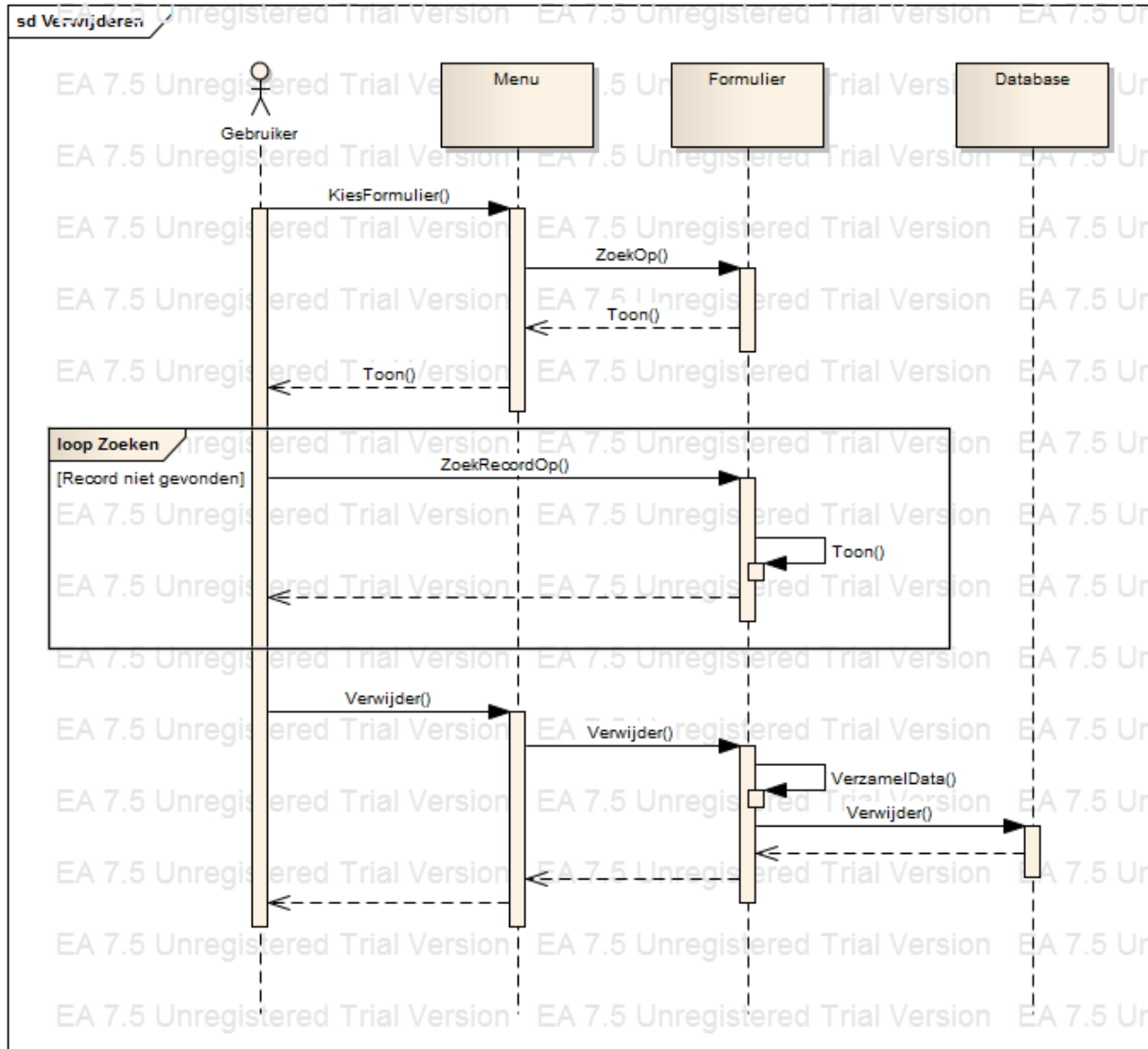
Figuur 6 Sequence Diagram Gegevens invoeren

De gebruiker geeft aan de MainController door welke gegevens men wil invoeren. De mainController stuurt naar de klasse welke een formulier bevat voor de gewenste gegevens invoer een verzoek om het formulier te tonen. Deze klassen zend het formulier naar de mainController waarna de gebruiker de gegevens kan invoeren. Indien de gebruiker klaar is en op verzenden drukt. Zal de klassen behorende bij het formulier kijken of de gegevens juist zijn ingevoerd. Indien dit niet het geval is krijgt de gebruiker een foutmelding en de mogelijkheid de gegevens aan te passen. Indien de gegevens

Ontwerp Document

goed zijn worden deze verzameld en in een query geplaatst vervolgens stuurt de klasse de gegevens naar de database.

3.4 Gegevens verwijderen



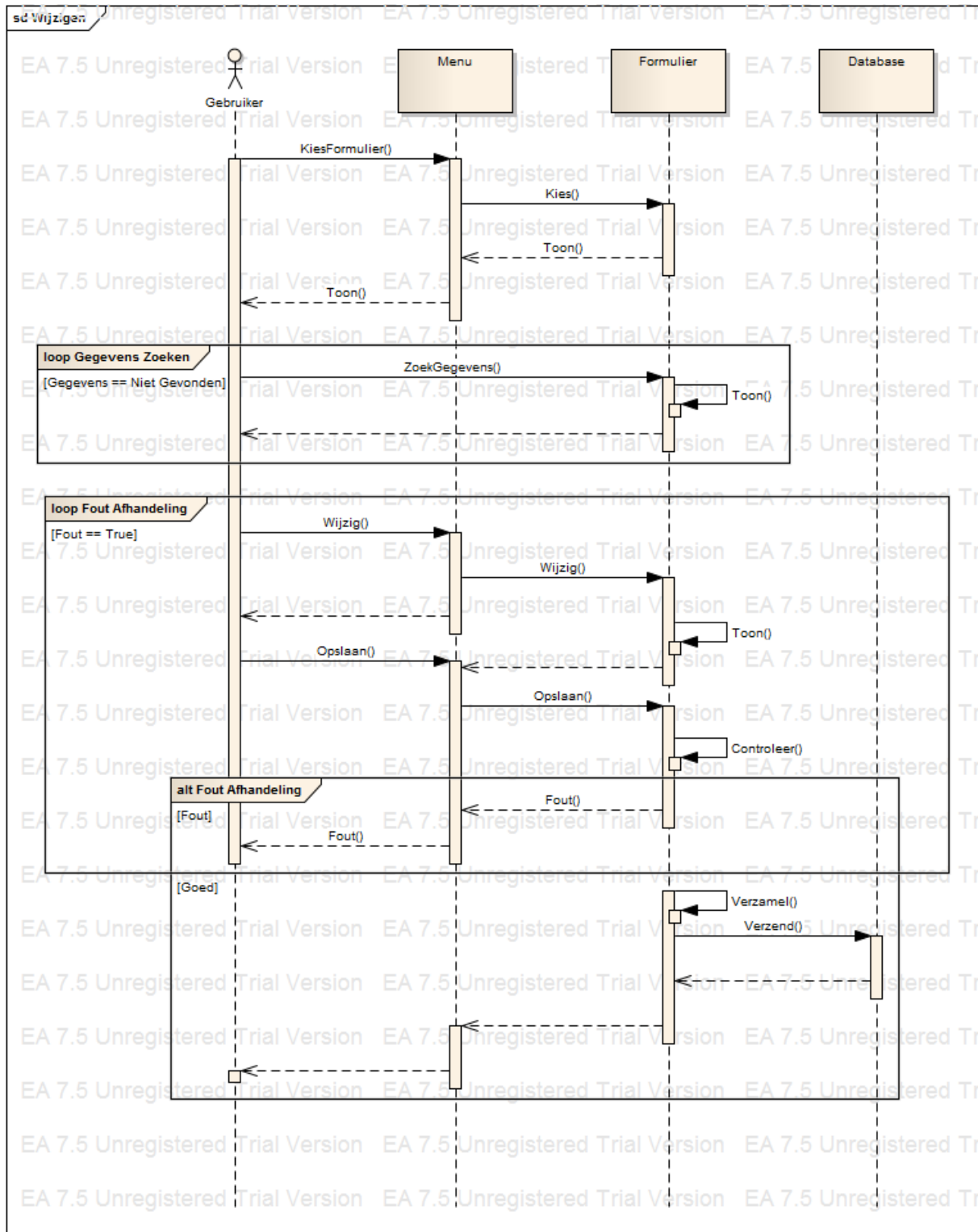
Figuur 7 Sequence diagram Gegevens verwijderen

De gebruiker geeft aan de mainController door gegevens te willen verwijderen. De mainController geeft een bericht aan de klassen welke deze gegevens afhandelt dat het formulier getoond moet worden. Deze klassen zend zijn formulier en de gebruiker kan de gegevens die men wil verwijderen opzoeken. Indien de gegevens zijn gevonden en de gebruiker op verwijderen drukt zal de klasse behorende bij het formulier de gegevens bekijken welke getoond worden. Indien er geen noodzaak is om de gegevens te bewaren (de gegevens zijn dus niet meer nodig bij andere elementen in het systeem) wordt in een query aan de database bekend gemaakt dat deze gegevens verwijderd kunnen worden.



Ontwerp Document

3.5 Gegevens Wijzigen



Figuur 8 Sequence Diagram Gegevens wijzigen

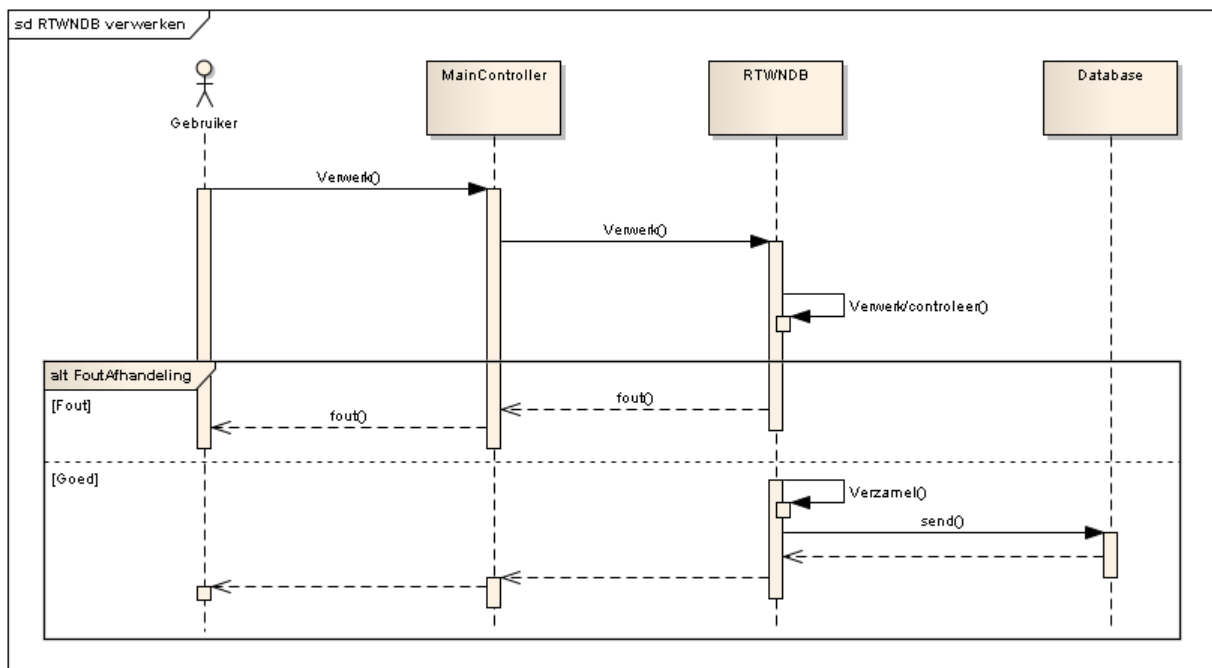
De gebruiker geeft aan de mainController door gegevens te willen wijzigen. De mainController geeft aan de klasse door die deze gegevens afhandelt dat het formulier



Ontwerp Document

getoond moet worden. De gebruiker zoekt vervolgens de juiste gegevens op welke getoond worden in het formulier. De gebruiker kan deze gegevens aanpassen en vervolgens opslaan. Indien men kiest om de gegevens op te slaan worden deze door de klasse behorende bij het formulier gecheckt. Indien er fouten zijn krijgt de gebruiker een foutmelding en de mogelijkheid de gegevens aan te passen. Indien de gegevens juist zijn worden deze verzameld en in een query naar de database gestuurd.

3.6 RTWNDB bestanden verwerken

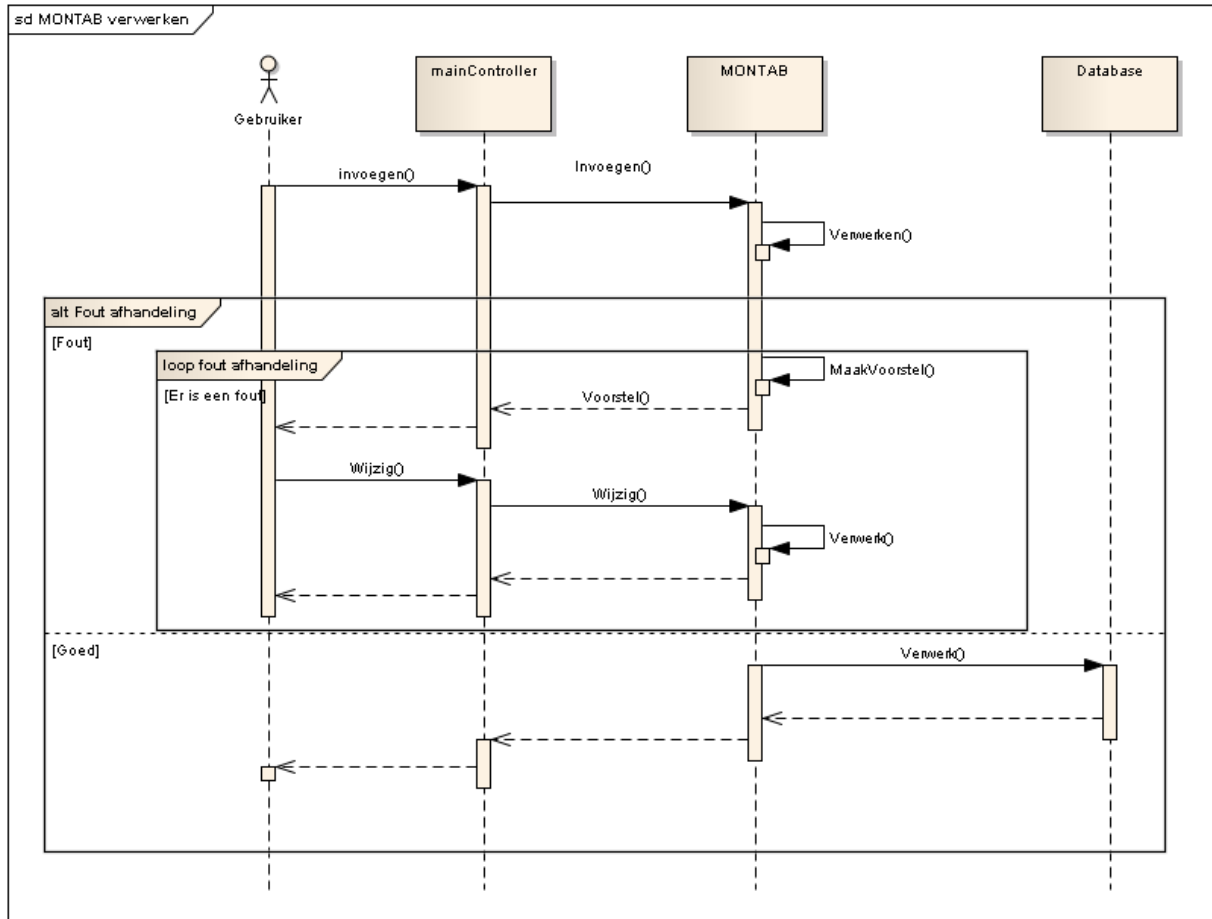


Figuur 9 Sequence Diagram RTWNDB verwerken

De gebruiker maakt aan de mainController duidelijk dat er RTWNDB bestanden verwerkt moeten worden. De mainController geeft aan de klassen RTWNDB door waar het bestand zich bevindt. Vervolgens verwerkt en controleert de klasse RTWNDB het bestand op fouten. Indien er een fout is gevonden zal een foutmelding volgen. Als alles goed is ingelezen zullen de gegevens klaar gemaakt worden voor verzending en in de database geplaatst worden.

Ontwerp Document

3.7 Montab bestanden verwerken

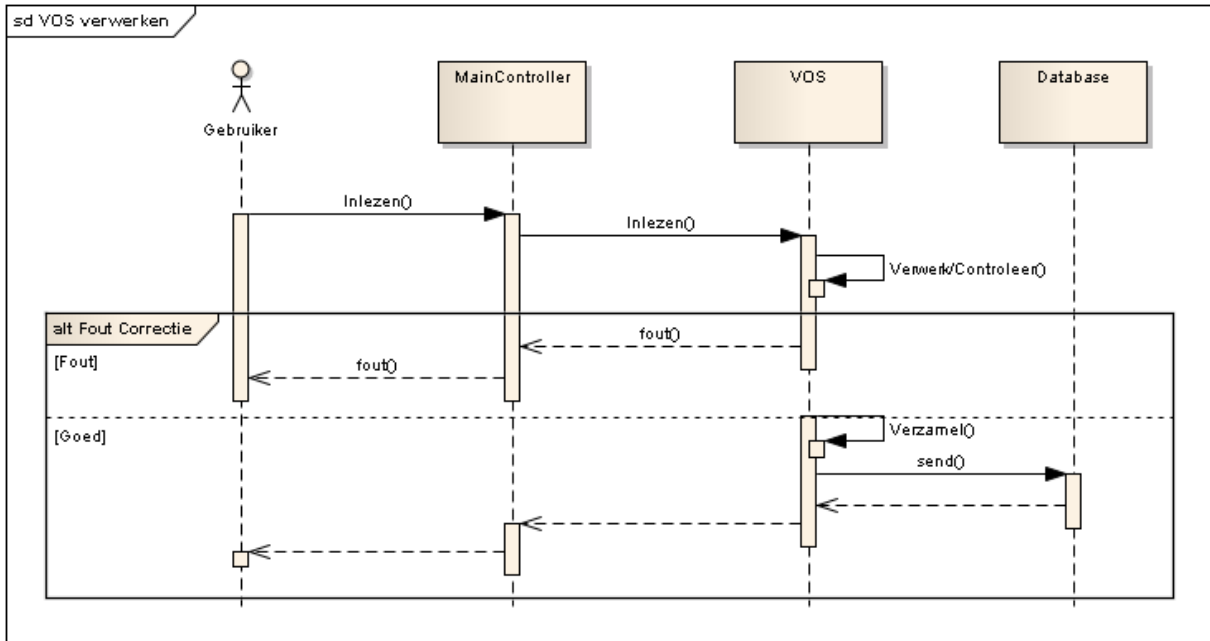


Figuur 10 Sequence diagram MONTAB verwerken

De gebruiker geeft aan de mainController door een MONTAB diskette te verwerken. De locatie van het bestand wordt doorgegeven aan de klassen MONTAB. Deze klassen haalt vervolgens het bestand op en verwerkt dit. Indien er fouten worden gevonden zullen er voorstellen gemaakt worden en dit aan de gebruiker getoond worden. Vervolgens krijgt de gebruiker de mogelijkheid de fouten te verbeteren. Hierna wordt het bestand opnieuw verwerkt en gecontroleerd op fouten. Indien het bestand volledig goed is zullen de gegevens in de database geplaatst worden.

Ontwerp Document

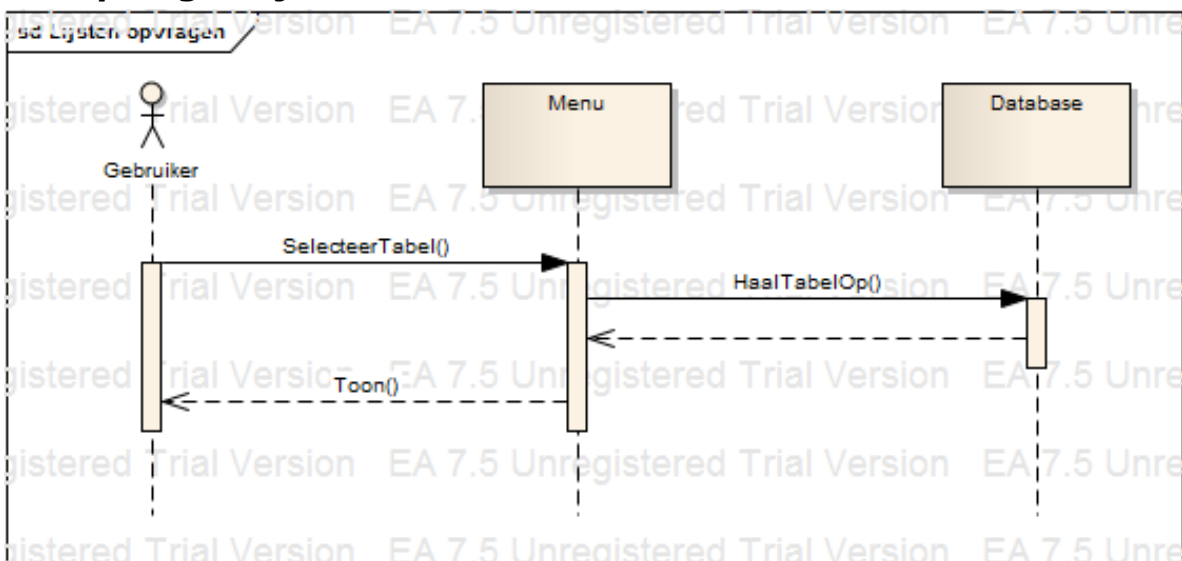
3.8 VOS verwerken



Figuur 11 Sequence diagram VOS verwerken

De gebruiker geeft aan de mainController door dat er VOS gegevens verwerkt moeten worden. Vervolgens geeft de mainController aan de klassen VOS door waar dit bestand zich bevind. De klassen VOS leest het bestand in en verwerkt/controleert dit op fouten. Indien er fouten aanwezig zijn volgt een foutmelding. Indien de gegevens juist zijn zullen de verzameld worden en in de database geplaatst worden.

3.9 Opvragen lijsten



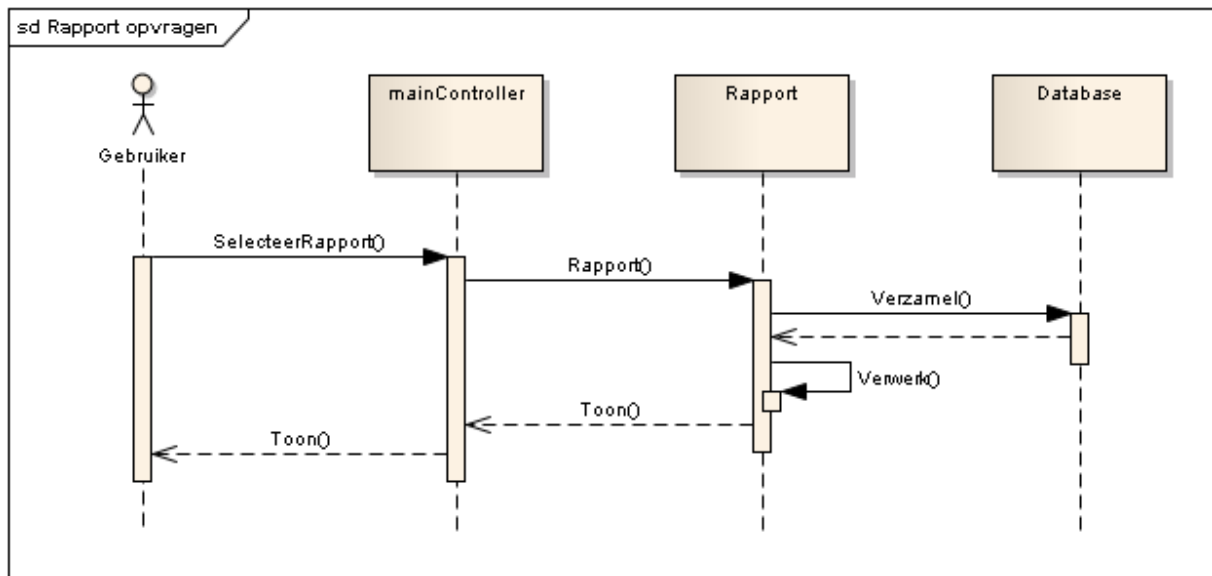
Figuur 12 Sequence diagram Lijsten opvragen

De gebruiker geeft aan de mainController door dat men een lijst wil ophalen. Aangezien lijsten een weergave zijn van de inhoud van een tabel geeft de gebruiker dus eigenlijk

Ontwerp Document

door welke tabel men wil zien. De MainController geeft aan de klassen lijsten door welke tabel getoond moet worden. De klassen lijsten haalt vervolgens alle gegevens behorende bij deze tabel uit de database en verzend deze naar de mainController.

3.10 Opvragen Rapporten

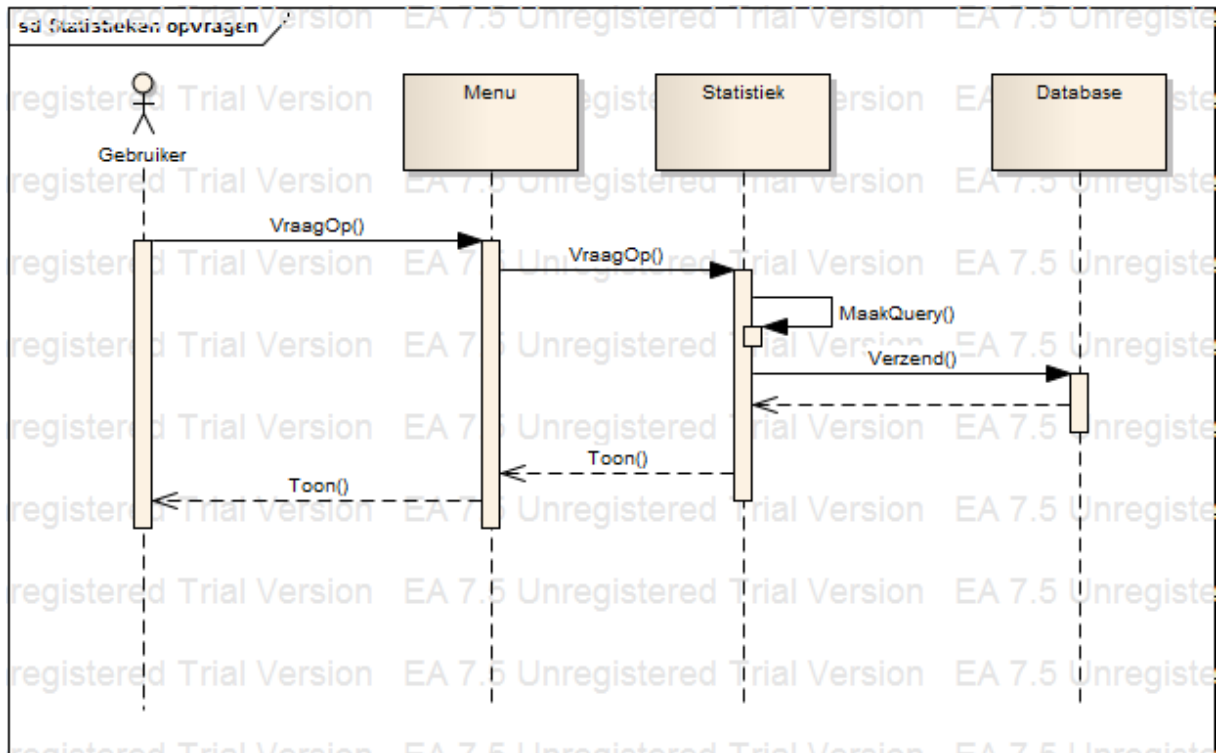


Figuur 13 Sequence Diagram Rapport opvragen

De gebruiker geeft aan de mainController door een rapport te willen maken. Na het invoeren van de gegevens die benodigd zijn zal de mainController de juiste functie aanroepen bij rapport. Indien men een rapport over de schepen wil maken zal dus een functie `RpSchepen(char,char,char)` verzonden worden. En wanneer men een verzendingsrapport wil maken zal de functie `Rpverzending(char,char,char)` verzonden worden. Rapport haalt vervolgens de juiste gegevens op en verwerkt deze in een rapport wat behoort bij deze gegevens. Dit rapport wordt vervolgens aan de gebruiker getoond.

Ontwerp Document

3.11 Opvragen Statistieken

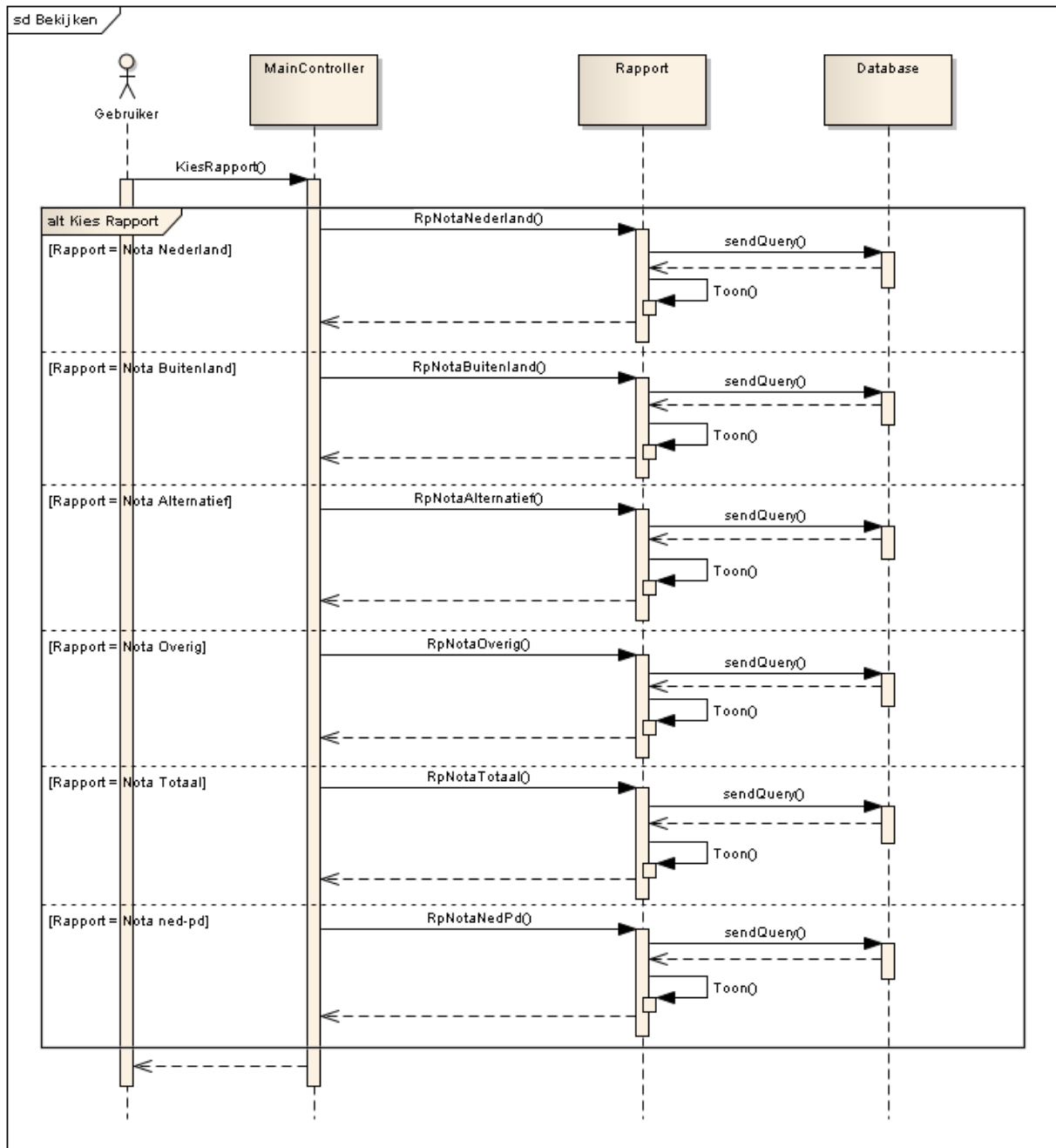


Figuur 14 Sequence Diagram Opvragen statistieken

De Gebruiker geeft aan de mainController door dat er statistieken opgevraagd moeten worden. De mainController geeft vervolgens door aan de klassen statistieken welk statistiek opgevraagd moet worden. De klassen statistiek maakt hier vervolgens een query van en verzend deze naar de database. Vervolgens worden de gegevens die terugkomen gepresenteerd aan de gebruiker.

Ontwerp Document

3.12 Nota bekijken



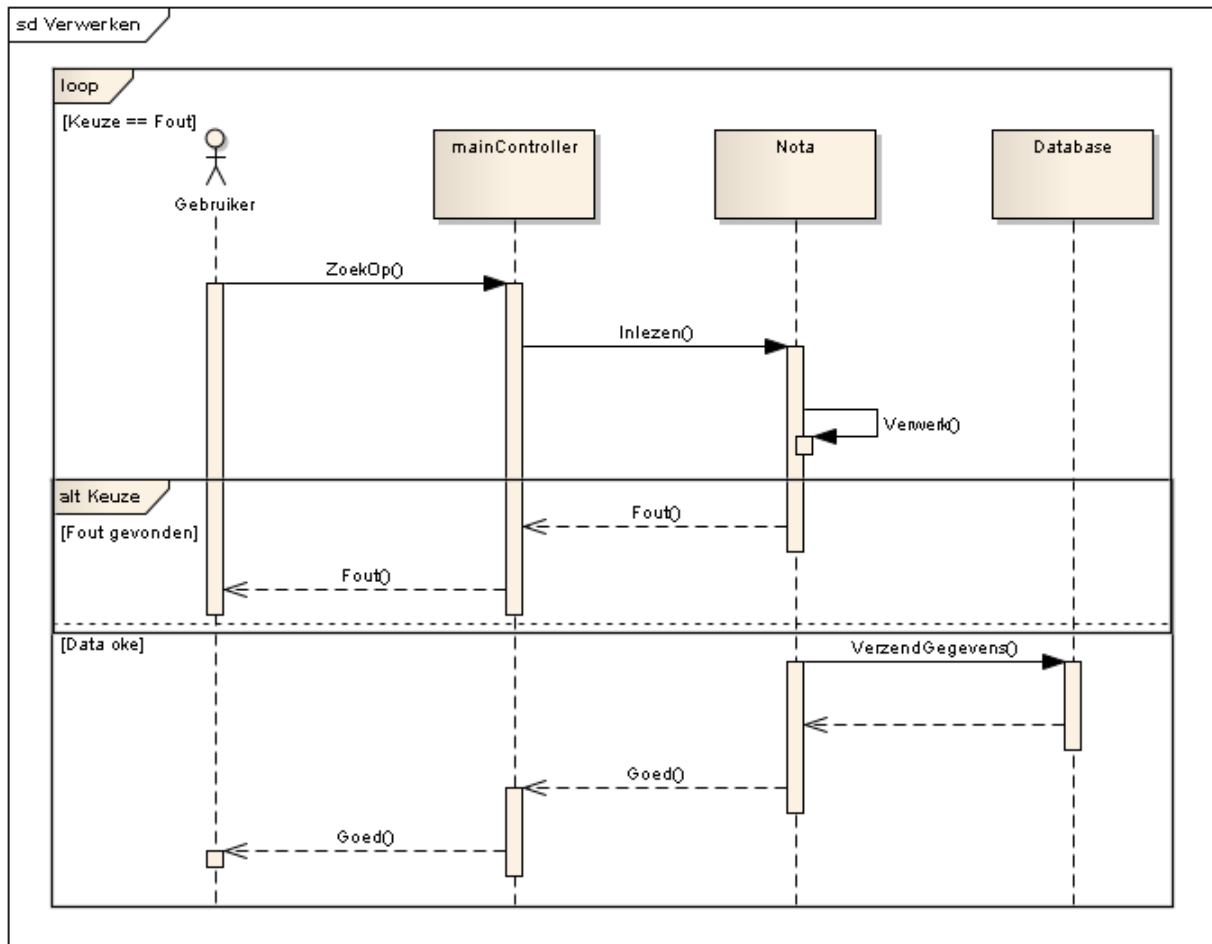
Figuur 15 Sequence Nota bekijken

Bij het ophalen van de nota's geeft de gebruiker aan de mainController door welke nota bekeken moet worden. De MainController geeft vervolgens door aan de klasse rapport welk rapport opgehaald moet worden. Vervolgens zal de functie die aangeroepen is bij MainController de juiste gegevens ophalen uit de database en het juiste rapport produceren.



Ontwerp Document

3.13 Nota verwerken



Figuur 16 Sequence Diagram Nota verwerken

De gebruiker geeft aan de mainController door dat er nota's verwerkt moeten worden. Vervolgens wordt aan de klasse nota doorgegeven welke file ingelezen moet worden. De klasse nota verwerkt de file. Indien er fouten zijn wordt dit aan de gebruiker kenbaar gemaakt. Als alles goed is worden de gegevens uit de file in de database geplaatst en krijgt de gebruiker een bericht dat de gegevens verwerkt zijn.

Hoofdstuk 4: klassendiagram

[illegible]

Figuur 17 Klassendiagram

- 29 -



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp Document

Hoofdstuk 5: HTML lay-out

De HTML die de lay-out van de website zal bepalen wordt gemaakt in een aparte file. Hierdoor wordt het mogelijk de lay-out file aan te passen zonder hiervoor bij de php code te moeten komen. Het is hierdoor mogelijk een totaal andere lay-out te ontwikkelen zonder veel verstand te moeten hebben van de echte functionaliteit van de website.

De Lay-out zal worden geïmplementeerd in de php pagina's doormiddel van het commando: `$layout = file_get_contents("layout.html")`. Door onder aan de pagina het volgende commando uit te voeren: `$layout = str_replace("%body%", $input, $layout); print($layout);`. Wordt het mogelijk de php code te plaatsen op het `%body%` punt in de html. Deze `%body%` zal geplaatst worden op het toekomstige infoveld (zie **hoofdstuk 1**).



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwerp Document

Hoofdstuk 6: CSS

Om de lay-out van de applicatie consequent te houden zal er gebruik gemaakt worden van een CSS¹ file. De CSS file zal onder andere de volgende elementen beschrijven:

Koptekst

Lettertype: Verdana

Grootte: 14 px

Kleur: zwart

Stijl: vet

Normale tekst

Lettertype: Verdana

Grootte: 10 px

Kleur: Zwart

Tevens zullen de menu's van de lay-out in deze CSS file beschreven worden.

¹ Cascading Style Sheets



Ontwerp Document

Hoofdstuk 7: XML

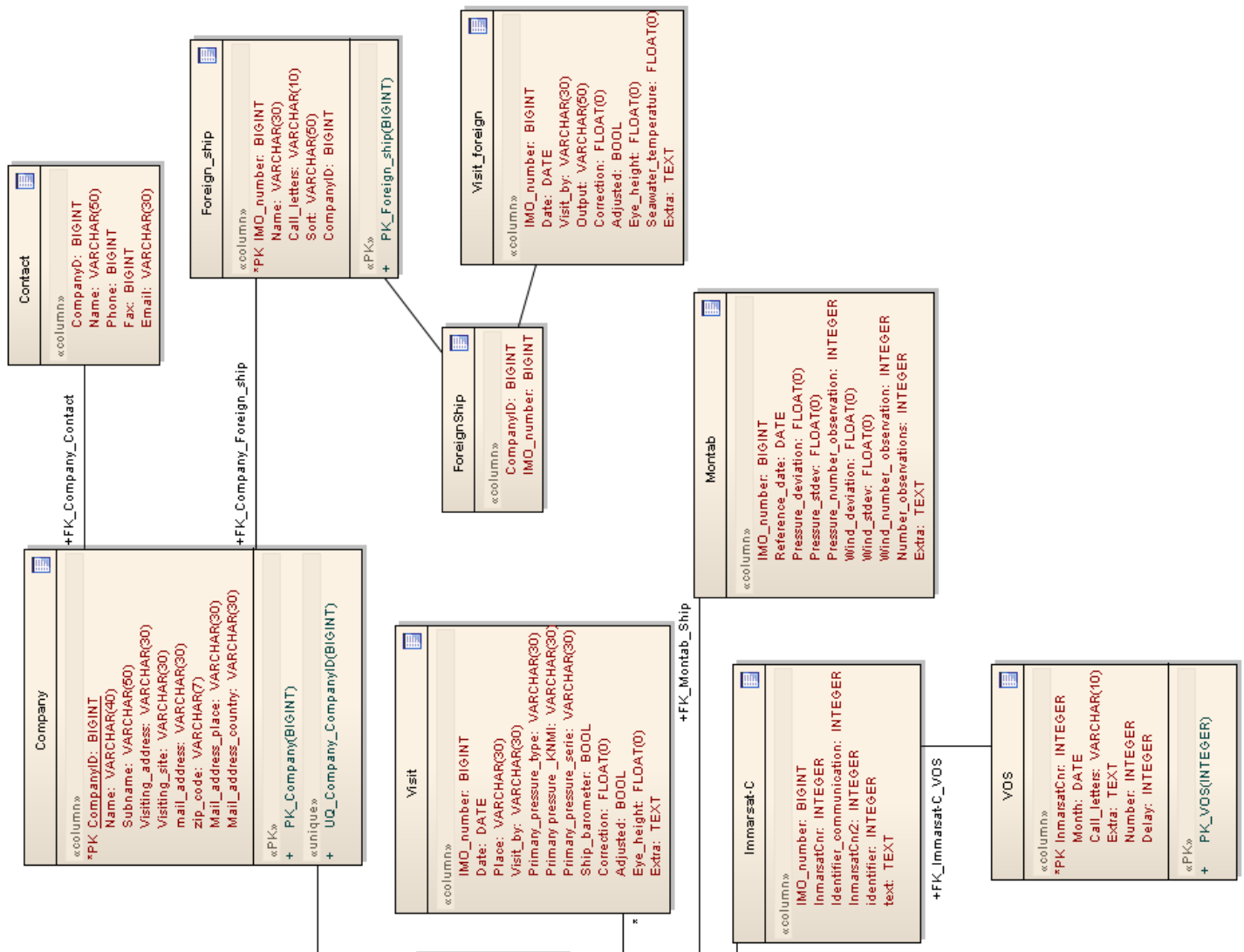
Voor het bijhouden van instellingen zal gebruik gemaakt worden van een XML¹ file. Het voordeel hiervan is dat makkelijk centraal instellingen aangepast kunnen worden. Voor het uitlezen van de XML zal de XML dom parser gebruikt worden. Zaken die in de XML vermeld kunnen worden zijn, database locatie, naam, gebruikersnaam en wachtwoord. Tevens kunnen ook andere zaken waarvan het handig is dat dit vanaf een centraal punt in te stellen is in de XML file opgenomen

¹ Extensible Markup Language



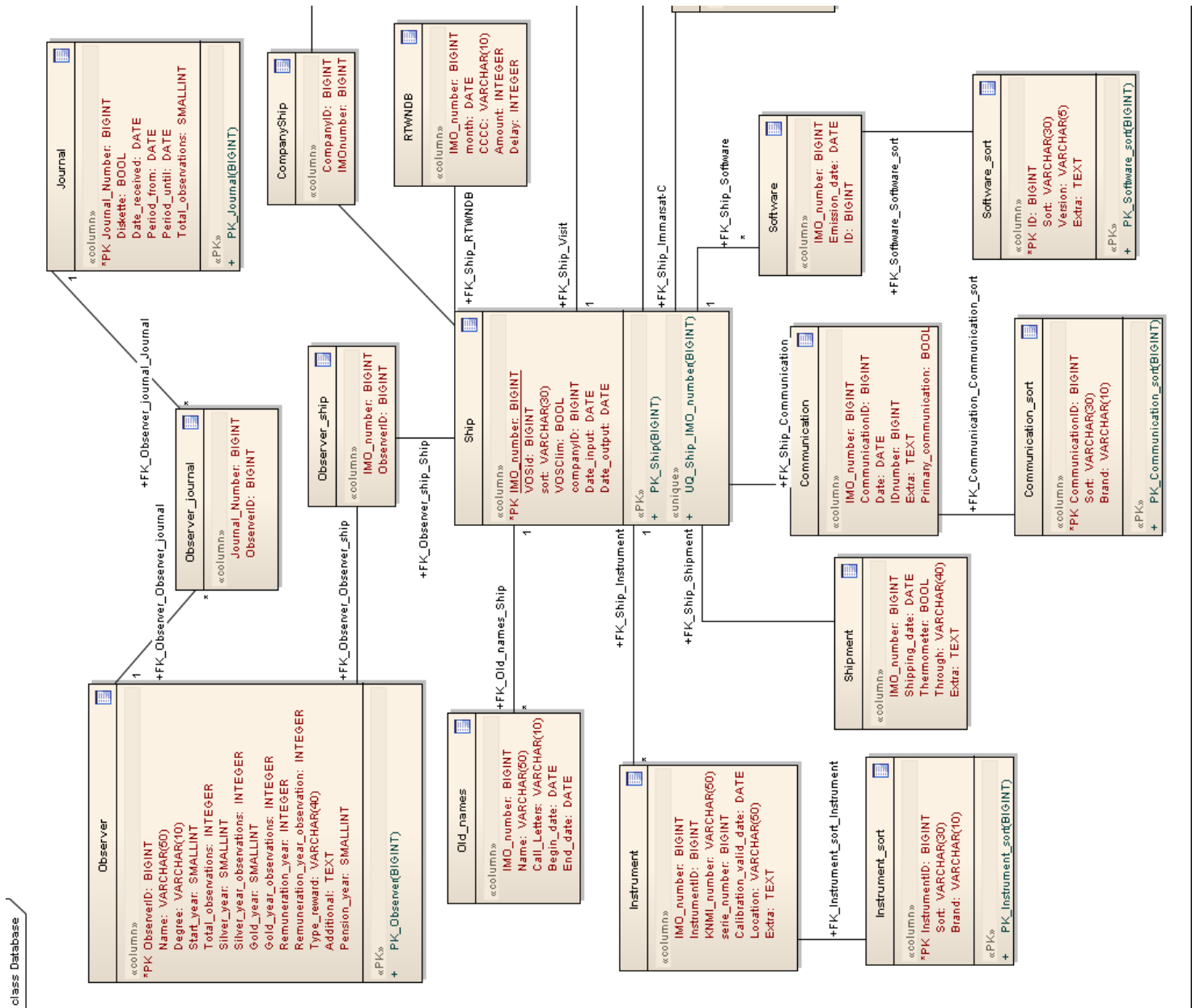
Ontwerp Document

Bijlage 1: Conceptueel schema





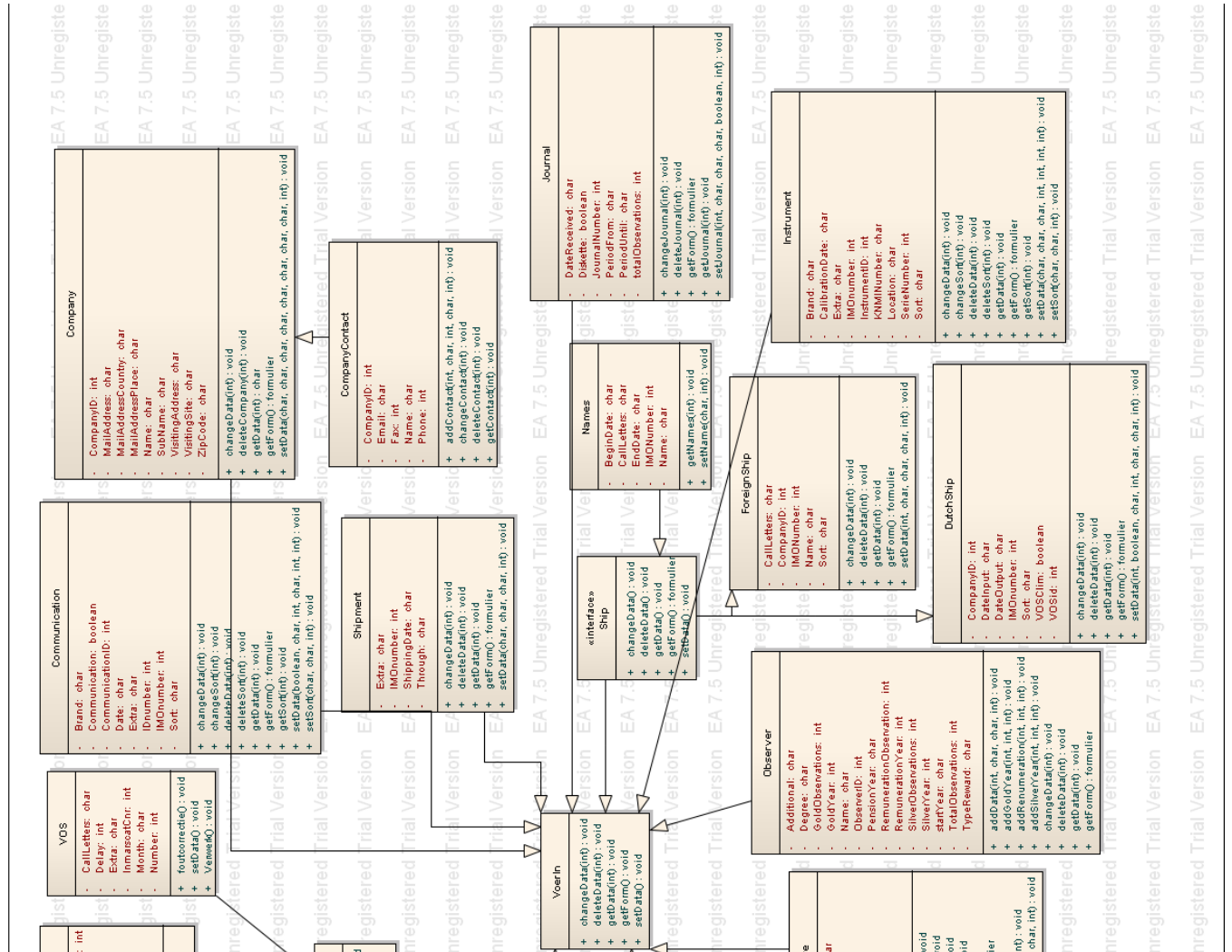
Ontwerp Document





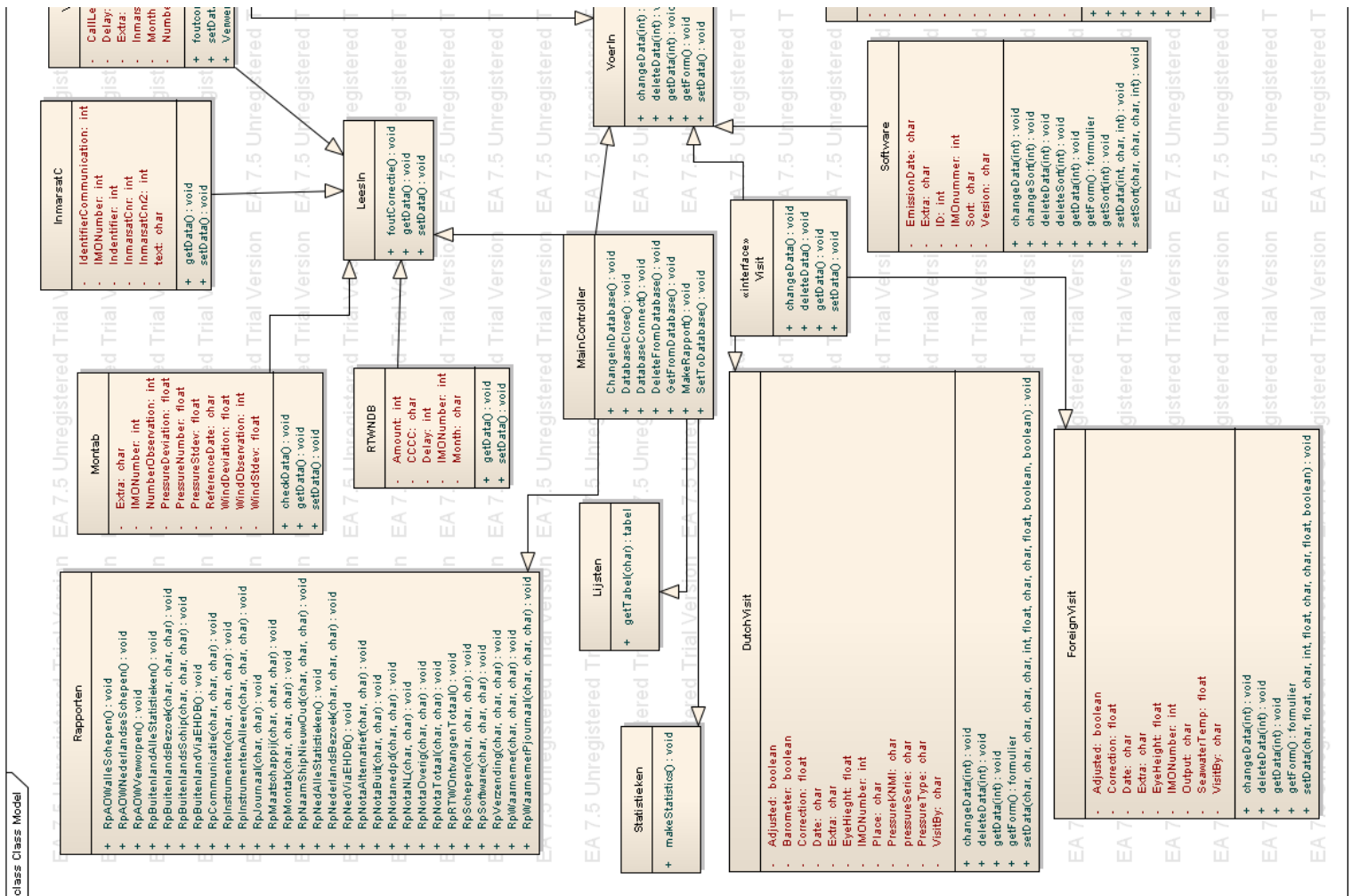
Ontwerp Document

Bijlage 2: klassendiagram





Ontwerp Document





Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Handleiding SEA



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Handleiding

Inhoud


1. Item toevoegen	3
2. Items bekijken	4
3. Items wijzigen	5
4. Item verwijderen	6
5. Lijsten	7
6. Read files	8
7. Reports	9
8. SQL builder	11
9. Statistieken	12



Handleiding

1. Item toevoegen

In het menu ADD kunnen verschillende elementen aan de database toegevoegd worden. De naamgeving verduidelijkt waar wat ingevuld kan worden.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Add

[Company](#)
[Company Contact](#)
[Company Ship](#)
[Company Foreign Ship](#)
[Instrument](#)
[Instrument Sort](#)
[Shipment](#)
[Communication](#)
[Communication Sort](#)
[Software](#)
[Software Sort](#)
[Visit](#)
[Ship](#)
[Observer](#)
[Foreign Visit](#)
[Foreign ship](#)
[Journal](#)
[Ship Name](#)
[Observer Company](#)
[Observer Journal](#)

View
Change
Delete
List
Read files
Reports
SQL Builder
Statistieken

Welcome!

Company:

Name:

Subname:

Visiting:

Address:

Place:

Zipcode

Mail

mail address:

zip code:

mail address place:

mail address country:

Figuur 1 Menu ADD



Handleiding

2. Items bekijken

In het menu View kunnen gegevens uit de database bekeken worden. Via een drop down box kan in de meeste gevallen voor een schip gekozen worden. Hierna worden de elementen behorend bij dat schip getoond. Indien een schip bijvoorbeeld meerdere instrumenten heeft zullen deze dus allemaal verschijnen.



Add	View Instrument
View	
Instrument	Geef de zoekactie op:
Instrument_Sort	<input type="text" value="jeo - 111"/> <input type="button" value="Search"/>
Shipment	
Communication	
Communication_Sort	
Software	
Software_Sort	
Visit	
Company	
Ship	
Observer	
Foreign_Visit	
Foreign_ship	
Journal	
Change	
Delete	
List	
Read files	
Reports	
SQL Builder	
Statistieken	

Figuur 2 Menu View



Handleiding

3. Items wijzigen

In het menu Change kan de data die ingevoerd is gewijzigd worden. Doormiddel van een drop down box kan het item wat gewijzigd moet worden, worden geselecteerd. Hierna kan doormiddel van een formulier de data aangepast worden en opnieuw in de database worden opgeslagen.

The screenshot shows a web application interface. On the left is a vertical menu with blue buttons: 'Add', 'View', 'Change', 'Delete', 'List', 'Read files', 'Reports', 'SQL Builder', and 'Statistieken'. Below the 'Change' button is a list of links: [Instrument](#), [Instrument_Sort](#), [Shipment](#), [Communication](#), [Communication_Sort](#), [Software](#), [Software_Sort](#), [Visit](#), [Company](#), [Ship](#), [Observer](#), [Foreign_Visit](#), [Foreign_ship](#), and [Journal](#). To the right of the menu is a section titled 'View InstrumentSort'. Below the title is the text 'Geef de zoekactie op:' followed by a search box containing 'thermo - meter' and a 'Search' button. In the top right corner, there is a logo and the text 'Koninklijk Nederlands Meteorologisch Instituut Ministerie van Verkeer en Waterstaat'.

Figuur 3 Change Menu



Handleiding

4. Item verwijderen

Doormiddel van een drop down box kan het element wat verwijderd moet worden, worden geselecteerd. Door op select te drukken zal het element direct uit de database verwijderd worden.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Add
View
Change
Delete

[Instrument](#)
[Instrument_Sort](#)
[Shipment](#)
[Communication](#)
[Communication_Sort](#)
[Software](#)
[Software_Sort](#)
[Visit](#)
[Company](#)
[Ship](#)
[Observer](#)
[Foreign_Visit](#)
[Foreign_ship](#)
[Journal](#)

List
Read files
Reports
SQL Builder
Statistieken

View Instrument

Geef de zoekactie op:


Figuur 4 Verwijder items



Handleiding

5. Lijsten

In het menu lijsten kan de inhoud van de verschillende tabellen opgevraagd worden. De informatie die getoond wordt is een 1 op 1 overzicht van de gegevens die zich in die tabel bevinden.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Add

View

Change

Delete

List

[Instrument](#)

[instrument_sort](#)

[Shipment](#)

[Communication](#)

[communication_sort](#)

[Software](#)

[Software_sort](#)

[visit](#)

[company](#)

[ship](#)

[observer](#)

[visit_foreign](#)

[foreign_ship](#)

[journal](#)

Read files

Reports

SQL Builder

Statistieken

Welcome!

IMO_number	InstrumentID	KNMI_number	serie_number	Calibration_valid_date	Location	Extra	Placed	Primary
656	3	111	111	12-03-2010	1333	333	333	0
123	3	111	111	15-03-2010	111	111	22	0
656	3	33	33	13-09-1989	33	333	333	1
656	3	11	11	14-05-2006	11	11	11	0
13	1	123	11	13-09-1989	233	344	4443	1
656	3	123	6666	13-09-1988	66	666	666	0
13	3	test	0	13-09-1989	test2	lal	test4	0
656	3	11	11	13-09-1988	11	111	111	0

Figuur 5 Lijsten



Handleiding

6. Read files

In het menu read file kunnen de verschillende bestanden ingelezen worden. Bestanden die aangeleverd worden met de extensie .txt kunnen direct ingelezen worden. Excel bestanden moeten eerst omgezet worden naar een ; gescheiden .csv bestand voordat deze ingelezen kunnen worden.

Wanneer een journaaldiskette ingelezen wordt kunnen de fouten in die gevonden worden aangepast worden. Door op de link "again" te drukken zal het bestand opnieuw gecontroleerd worden op fouten nadat de gegevens zijn aangepast.

Doormiddel van de link get journalfiles kunnen alle bestanden die ingelezen zijn als zip bestand gedownload worden. Een kopie van de zip file blijft als backup op de server opgeslagen. Indien de gedownloaden zip bestanden kwijt mochten raken kunnen deze dus nog opgevraagd worden.



Add	Welcome!	Filename: <input type="text"/>	Bladeren...
View		<input type="button" value="Submit"/>	
Change			
Delete			
List			
Read files			

[RTWNDB](#)
[VOS](#)
[Nota](#)
[Montab](#)
[Journal_Disk](#)
[Alternative_Lijst](#)
[Get_journalFiles](#)

Reports
SQL Builder
Statistieken

Figuur 6 Read Files



Handleiding

7. Reports

Onder het menu reports kunnen verschillende rapporten gegenereerd worden. In het submenu reports kan een rapport gegenereerd worden van de volgende items:

Ship
Visit
Shipment
Software
Communication
Instrument
Foreign ship
Visit foreign ship

Vervolgens kan gekozen worden welke beginletters meegenomen moeten worden in het rapport. Tevens is het nog mogelijk de gegevens te sorteren op:

Name
Callsign
Company

In het submenu reports nota kunnen rapporten van nota's uitgeprint worden. Er kan gekozen worden voor de volgende rapporten:

NotaNL
NotaBuitenland
NotaTotaal
NotaOverig
NotaAlternatief

Vervolgens kan gekozen worden welke beginletters van schepen meegenomen moeten worden in het overzicht. Hierna kan de koers ingevuld worden waarna een rapport getoond, wordt waarin de kosten zowel in euro's als in dollars getoond worden.

Bij het submenu raport DMG kunnen de volgende rapporten gegenereerd worden:

RTW ontvangen totalen
Nederlandse schepen via alle stat
Nederlandse schepen via EHDB
Rpt AOW Nederlandse schepen
Rpt AOW Verworpen berichten
Buitenlandse schepen via alle stat
Buitenlandse schepen via EHDB
AOW alle schepen
Nederlandse schepen (VOS id) via alle stat
Nederlandse schepen (VOS id) via EHDB
Rpt AOW Nederlandse schepen (VOS id)

Hierna kan het jaar en de maand geselecteerd worden waarvan het rapport getoond moet worden.

Bij het submenu overig kunnen de volgende rapporten gegenereerd worden:

Company



Handleiding

Instrument Sort
Journal
MONTAB
Observer
Observer journal

Vervolgens kan nog gekozen worden welke beginletters meegenomen mogen worden.

Voor elk rapport kan gekozen worden of deze als .doc of .odt gegenereerd moet worden.



Add	<h2>Welcome!</h2> <p>Rapport: <input type="text" value="Ships"/></p> <p>From: <input type="text" value="A"/></p> <p>until with: <input type="text" value="A"/></p> <p>Order By: <input type="text" value="Name"/></p> <p>Doc Type: <input type="text" value=".doc"/></p> <p><input type="button" value="Get Rapport"/></p>
View	
Change	
Delete	
List	
Read files	
Reports	
Reports	
Raport_Nota	
Raport_DMG	
Raport_Overig	
SQL Builder	
Statistieken	

Figuur 7 Reports




Handleiding

8. SQL builder

In de SQL builder is het mogelijk SQL query's direct op de database uit te voeren. Indien de query bewaard moet worden is dit ook mogelijk onder het submenu Build And Save. Een query moet een naam krijgen zonder spaties.

Opgeslagen query's kunnen opnieuw uitgevoerd worden en indien men dit wenst eventueel worden verwijderd.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Add

View

Change

Delete

List

Read files

Reports

SQL Builder

Build Query

Build AND Save

Load Query

Delete Query

Statistieken

Welcome!

Query Builder:

```
select * from ship
```

Submit

IMO_number	VOSid	sort	VOSclim	companyID	Date_input	Date_output
2905	22	22	22	1	21-05-1988	22-05-1999
2333	lol	lol	0	1	15-04-2001	15-05-2001
554	eette	etetw	0	1	14-09-1989	15-09-1989
656	wtrwe	ytyrere	0	1	14-09-1989	15-09-1989
777	rgr	gfgh	0	1	13-09-1989	14-09-1989
909	909	909	127	335	13-09-1989	14-09-1989
123	1323	gg	0	1	13-02-2009	13-04-2009
1988	PCCL	HIS	33	6	13-09-1989	14-09-1989
12345	2	HIS	1	15	13-04-2010	14-04-2010

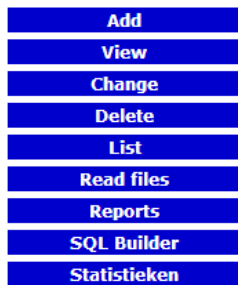
Figuur 8 SQL builder



Handleiding

9. Statistieken

Bij statistieken kan gekeken worden wanneer een schip voor het laatst bezocht is. De maanden waartussen het laatste bezoek plaats heeft gevonden moeten opgegeven worden waarna een tabel met de betreffende schepen getoond zal worden.



[statistiek](#)

Welcome!

From: Month:
Until: Month:



Figuur 9 Statistieken



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwikkelaars handleiding

Nieuwe SEA



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwikkelaars handleiding

Inhoud

1. Inleiding	3
2. handmatige invoer	4
3. Inlezen bestanden	6
3.1 RTWNDB	6
3.2 VOS	6
3.3 Nota bestanden	6
3.4 MONTAB	6
3.5 Journaalsdiskette	6
4. reports	8
5. SQL builder	9
6. statistiek	10
7. Lijsten	11
8. Layout	12



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwikkelaars handleiding

1. Inleiding

Dit document beschrijft kort de werking van het systeem. Hierbij wordt vooral ingegaan op zaken die niet voor zich zelf spreken of niet in de software zelf voldoende gedocumenteerd zijn.



Ontwikkelaars handleiding

2. handmatige invoer

Elke formulier beschikt over een eigen klassen. Deze klassen beschikken over de volgende functies:

getForm(\$i)

Deze functie toont het gewenste formulier. *\$i* bepaald de manier waarop het formulier getoond wordt (alleen lezen of wijzigen). De values die getoond worden in de invoervelden zijn de private variabelen. Indien men een formulier invuld zijn deze dus leeg. Echter met *getData()* worden deze variabelen gevuld waardoor ze in het formulier verschijnen.

Een formulier is opgebouwd in een tabel op de volgende manier

Naam	Invulveld	Error veld
------	-----------	------------

In het naam veld wordt weergegeven wat ingevuld moet worden.

In het invulveld kan men de data invoeren. Indien men data veranderd of bekijkt staat hier de oude data. In het error veld worden de fouten weergegeven indien de ingevulde data voor dat veld niet klopt.

De error messages worden door de functies die op fouten controleren neergezet. Deze functies plaatsen in de array *error* de fout. Welke vervolgens op de juiste plaats in het error veld verschijnt. Indien er geen fout is wordt " " weggeschreven in het array. Er verschijnt dan geen error boodschap naast het veld.

Process()

Dit is de functie die aangeroepen wordt door *getForm(\$i)*. In deze functie worden de verschillende functies aangeroepen die de data controleren. Als eerste wordt hier de data van de verschillende velden opgehaald en naar de controleerfuncties gestuurd.

De controleerfuncties, controleren de data en zetten indien er een fout is, een error message in de error array. Indien er een fout ergens is wordt check 1 gemaakt. Hierdoor zal de data door de process functie niet weggeschreven worden naar de database maar dient de gebruiker de data eerst te herstellen.

Indien er geen fouten gevonden zijn wordt door de functie process de data weggeschreven. De waarden van de variabelen change bepaald de manier waarop. Indien change ongelijk aan 1 is betreft het nieuwe data en wordt *setData()* aangeroepen. Indien change gelijk is aan 1 betreft het al bestaande data en wordt *changeDate()* aangeroepen.

setData()

Deze functie zet de ingevulde data in de database. De functie wordt alleen aangeroepen als in process blijkt dat alle data goed is ingevuld.

getData()

De functie haalt de data op uit de database en zet deze in de private variabelen. Zodra hierna *getForm()* wordt aangeroepen zal de data hierin in de juiste velden verschijnen.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwikkelaars handleiding

DeleteData(\$id)

Functie verwijderd de data met behorende bij \$id uit de database.

changeData()

Deze functie verandert het record wat al in de database bestaat.



Ontwikkelaars handleiding

3. Inlezen bestanden

Het inlezen van bestanden begint met de basisklasse ReadFile. Deze functie beschikt over de basisfunctionaliteit zoals het selecteren van de file. Ook wordt hier het bestand gecontroleerd. Doormiddel van een XML bestand wordt gekeken of de extensie van het bestand klopt. Dit wordt gecontroleerd aan de hand van de opbouw van de bestandsnaam waaruit is af te leiden welk bestand het is.

De overige klassen die per bestand zijn aangemaakt overerven de functionaliteit van ReadFile. Hierdoor hoeft in veel gevallen alleen de functie process() voor elk bestand opnieuw geschreven te worden.

3.1 RTWNDB

Het bestand wat verwacht wordt is een ; gescheiden bestand. Er wordt steeds opzoek gegaan naar de ; aangezien men weet welke waarden verwacht wordt, wordt deze tijdelijk opgeslagen in een variabelen. Zodra de regel is ingelezen wordt deze juist weggeschreven naar de database waarna het proces zich herhaald.

3.2 VOS

VOS bestanden worden op dezelfde manier verwerkt als RTWNDB. Het bestand wordt eerst weer omgezet naar ; gescheiden csv bestand. Vervolgens wordt gezocht naar deze ;.

3.3 Nota bestanden

Nota bestanden worden ingelezen doormiddel van een .txt bestand. In een XML bestand is opgeslagen welke data zich waar op de regel bevind. Dit zorgt ervoor dat de data in de juiste kolommen wordt geplaatst. De volgorde van de elementen in de XML is gelijk aan de volgorde van de kolommen in de database. Een nieuwe kolom toevoegen aan het einde van de database plus aan het einde van de XML de naam en locatie van de data opgeven. Is genoeg om nieuwe data op te slaan in de database.

3.4 MONTAB

Bij MONTAB bestanden wordt eerst bepaald wat het onderwerp van de data is. Zodra een onderwerpsregel is gevonden is het type data bekend. Hierdoor is duidelijk waar de data die eronder staat over gaat. Het type bepaald in de switch statement waar de data naartoe weggeschreven wordt. Voor de switch statement wordt de data in een paar variabelen gezet die in het switch statement afhankelijk van het type naar de juiste kolommen worden weggeschreven.

3.5 Journaalsdiskette

Het journaalsdiskette bestand wordt, regel voor regel in de database gezet met een oplopend ID. Elke regel wordt gecontroleerd. Indien een fout gevonden wordt, wordt het ID van die regel opgeslagen in een array. Tevens wordt een regel getoond waarop aangegeven wordt dat er een fout gevonden is. Via de functie error kan de fout hersteld



Ontwikkelaars handleiding

worden. Hier wordt het item uit de database met het juiste ID getoond plus de items met de twee ID's er onder en boven. Indien de gebruiker op delete drukt zal DELETE in het record worden ingelezen. Wanneer het bestand opnieuw verwerkt wordt, worden deze records niet meegenomen. Hierbij wordt de functie ToTXT() aangeroepen die alle data uit de database in een nieuw .txt bestand zet. Records waarin DELETE staat worden overgeslagen.

De functie getZIP() maakt van alle bestanden die op de server staan een zipfile welke aan de gebruiker ter download aangeboden wordt. De bestanden worden vervolgens uit de directory op server verwijderd. Echter blijft wel een kopie van de zipfile op de server opgeslagen zodat oude data altijd terug te halen is.



Ontwikkelaars handleiding

4. reports

via verschillende formulieren kunnen rapporten opgevraagd worden. Elk rapport bestaat uit één of meerdere query's de query's staan opgeslagen in de map Reports/Query's op de server. De functie `getQuery($query)` haalt hiervan een query op die als string teruggegeven wordt. Omdat in de forms door de gebruiker gekozen kan worden tussen welke letters de naam moet zitten moet dit op het laatste moment aan de query toegevoegd worden. De query's bevatten daarom de regel `Between #from AND #until`. De functie `setBetween($Query,$from,$until)` kan na opgave van de query, start letter en tot letter de query aanpassen tot wat de gebruiker wenst. De juiste query wordt hierna teruggegeven door de functie.

De functie `Output($sql)` van de klassen `sql builder` geeft een tabel van een query terug. Deze functie wordt aangeroepen om de query daadwerkelijk uit te voeren. Het resultaat wordt weggeschreven naar een bestand.

Aan het begin van de functie `process` behorende bij het formulier wordt het bestand waarin de data wordt weggeschreven geopend. Dit is afhankelijk van de keuze van de gebruiker een `.doc` of `.odt` bestand. Indien het bestand niet bestaat wordt dit aangemaakt. Door de `fwrite()` functie wordt de data in het bestand weggeschreven. Indien dit gedaan is wordt het bestand gesloten met `fclose()` waarna het ter download wordt aangeboden.



Ontwikkelaars handleiding

5. SQL builder

De klassen SQL builder stelt de gebruiker in staat sql query's te bouwen en uit te voeren op de database. Deze query's kunnen opgeslagen worden waarna ze later nogmaals makkelijk kunnen worden uitgevoerd

De functie Output(\$sql) maakt van een query een tabel. Als eerste zal de titel van de tabel gemaakt worden. Dit wordt gedaan doormiddel van mysql_field_name() de namen van de velden op te vragen en deze in de verschillende kolommen te plaatsen.

Hierna wordt doormiddel van mysql_fetch_row de waardes in de juiste kolommen neergezet. Het resultaat van de hele tabel die ontstaat wordt via een return teruggegeven.

De functie builder() stelt de gebruiker in staat een query te bouwen en naar de functie Output te sturen. Doormiddel van een try en catch wordt gekeken of er een bestand TEMP.sql aanwezig is dit bestand bevat de laatste gedraaide query welke in het formulier getoond kan worden. Hierdoor hoeft de gebruiker indien zijn laatste query aangepast moet worden en deze niet opgeslagen is niet de hele query opnieuw in te voeren.

De functie process() stuurt de ingevoerde query naar de functie Output().

De functie save() stelt de gebruiker in staat de query op te slaan. Hierbij wordt de query weggeschreven in een .sql bestand met een naam naar keuze. Tevens wordt de naam van de query samen met de naam van het sql bestand opgeslagen in de XML file.

De xml file wordt door de functie load() uitgelezen. Hierdoor kan de gebruiker een opgeslagen query opnieuw uitvoeren. Doordat in de XML file staat welk bestand geladen moet worden kan hierbij makkelijk het bestand ingelezen worden en aan de functie Output() doorgegeven worden.

De functie delete() stelt de gebruiker in staat een query aan te wijzen die verwijderd moet worden. De functie deletequery() verwijderd vervolgens de query uit de XML file en verwijderd het bestand van de server.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Ontwikkelaars handleiding

6. statistiek

de klassen statistiek zorgt ervoor dat de gebruiker statistieken kan raadplegen. De functie `getFom()` toont het formulier waarin de gebruiker het tijdsbestek van de statistieken kan aangeven. Hierna wordt de functie `Process()` aangeroepen. Deze functie haalt doormiddel van de functie `getQuery()` de statistiek query op van de server. Vervolgens wordt doormiddel van de functie `setBetween()` het tijdsbestek in de query ingevoerd. Hierna wordt doormiddel van de functie `Output()` van de klassen SQL builder de query uitgevoerd.



Ontwikkelaars handleiding

7. Lijsten

De klassen list laat afhankelijk van de naam die binnen komt een directe weergave zien van een tabel in de database dit wordt op de volgende manier gedaan:

```
public function getList($list)
{
    $this->query = "SELECT * FROM `".$list."`";
    $sql = new SQLbuilder;
    return $sql->Output($this->query);
}
```



Ontwikkelaars handleiding

8. Layout

De layout is opgebouwd uit een paar items. Het bestand begin.html bevat de HTML van de layout. De layout is in combinatie met het bijgevoegde bestand opmaak.css verantwoordelijk voor de layout. In het bestand begin.html is de tekst %body% te vinden. Door de verschillende php pagina's die de layout laden zal hierna gezocht worden. %body% wordt hierbij vervangen door wat daadwerkelijk getoond moet worden.

De pagina home.php handelt een groot gedeelte van aanvragen af. Alle zaken die door een formulier ingevuld moeten worden, evenals het maken van rapporten, lijsten, query's en statistieken worden door home.php afgehandeld. De url naar deze pagina is als volgt opgebouwd: home.php?i=adCompany*a&variant=normal. I bepaald hier welke klassen aangeroepen moet worden. Aangezien veel klassen over dezelfde functies beschikt is alleen de klassen nodig om de zaken aan te roepen.

Op de volgende manier wordt dit geregeld:

Via verschillende array's wordt bepaald tot welke groep iets behoort. Zo horen alle klassen die een bestand inlezen tot dezelfde groep, ze gebruiken namelijk dezelfde namen voor dezelfde functies:

```
$getFile = array(  
"RTWNDB" => $rtwndb = new RTWNDB,  
"VOS" => $vos = new VOS,  
"Nota" => $nota = new NOTA,  
"montab" => $montab = new MONTAB,  
"JournalDisk" => $journaldisk = new JournalDisk,  
"German" => $Alternative = new AlternativeLijst  
);
```

De eerste twee letters bij de link achter de i geven aan dat we iets willen toevoegen. Wat hierna komt betreft de klassen waarop dit betrekking heeft tot het *. Achter het * staat hoe het formulier getoond moet worden (bewerking of alleen lezen) Door nu met een if else statement te kijken of dit item in een array staat kan de juiste functie aangeroepen worden. In het geval van company gaat dit als volgt:

Company behoort tot de volgende array:

```
$arr = array(  
"Company" => $Company = new Company,  
"Ship" => $Ship = new Ship,  
"Observer" => $Observer = new Observer,  
"visitforeign" => $visitforeign = new VisitForeign,  
"shipforeign" => $shipforeign = new ShipForeign,  
"journal" => $journal = new Journal,  
"shipname" => $shipname = new ShipName,  
"companycontact" => $companycontact = new CompanyContact
```




Ontwikkelaars handleiding

```
);
```

Hierdoor komen we uiteindelijk bij het volgende if statement terecht:

```
else if(array_key_exists($a, $arr))
{
    if($value != "ad")
    {
        $b = trim($_GET['s']);
        $arr[$a]->getData($b);
    }

    $input .= $arr[$a]->getForm($act);

    if($value == "dl")
    {
        $b = trim($_GET['s']);
        $arr[$a]->deleteData($b);
    }
}
```

Zodra de eerste twee letter ad waren betreft dit dat we iets toe gaan voegen er wordt dan gelijk het juiste formulier getoond. Door `$arr[$a]->getForm($act)`. De `$a` bepaald welke klassen voor `->getForm($act)` komt te staan en dus welk formulier wordt getoond. De `$act` bepaald

De waarden die `$act` meekrijgt is de waarden die achter `*` stond in de url deze waarden bepaald of we zaken in het formulier mogen invullen of alleen mogen kijken.

Indien de eerste 2 letters geen ad betreft, betekend dit dat we zoiezo de data in het formulier willen tonen hierom wordt dan de functie `getData` aangeroepen.

Indien de eerste twee letters dl zijn willen we de data deleten waardoor `deleteData` wordt aangeroepen.

`&sort=Normal` is belangrijk voor `change.php`, `delete.php` en `view.php`. Indien de sort IMO is wordt een drop down box getoond met de schepen waaruit gekozen kan worden. In dit geval kunnen meerdere items aan boord van een schip zijn. Indien sort normal is wordt een drop down box met de items getoond het betreft hier dan de items zelf los van het schip.



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Verkeer en Waterstaat

Test rapport

Naam: Sybren Wille
Datum: 28-05-2010
Applicatie: Nieuwe versie SEA.



Test rapport

Inhoudsopgave

1. Inleiding	3
2 Soorten tests	4
2.1 Eenheidstest	4
2.2 Integratietest	4
2.3 Systeemtest	4
3. Add element	5
3.1 company	5
3.2 Company contact	5
3.3 Company ship	5
3.4 Company foreign ship	5
3.5 Instrument	5
3.6 Instrument sort	5
3.7 Shipment	5
3.8 Communication	6
3.9 Communication sort	6
3.10 Software	6
3.11 Software sort	6
3.12 Visit	6
3.13 Ship	6
3.14 Observer	6
3.15 Foreign visit	7
3.16 Foreign ship	7
3.17 Journal	7
3.18 Ship name	7
3.19 Observer company	7
3.20 Observer journal	7
4. View element	7
5. Change element	7
6. Delete element	8
7. List	8
8. Readfiles	8
8.1 RTWNDB	8
8.2 VOS	8
8.3 Nota	8
8.4 MONTAB	8
8.5 journaal diskette	9
8.6 get journal files	9
9. Reports	9
10. SQL builder	9
11. statistieken	9
12. Ondersteuning browsers	9



Test rapport

1. Inleiding

Om te zorgen dat de software goed werkt is testen noodzakelijk. Willekeurig verschillende functies testen werkt niet, omdat de kans dan groot is dat er iets wordt overgeslagen. Gestructureerd testen is daarom de methode om fouten direct te vinden, zodat deze zo snel mogelijk kunnen worden opgelost.

In hoofdstuk 2 staat beschreven welke tests zijn uitgevoerd tijdens het doorlopen van het project. Er is tevens gekeken naar de ondersteuning van de applicatie bij verschillende browsers. Aangezien elke browser HTML anders weergeeft moet gekeken worden of verschillende browsers de applicatie goed kunnen weergeven.



Test rapport

2 Soorten tests

Na en tijdens het programmeren is het programma steeds getest op het juist functioneren. De software heeft de volgende tests doorlopen:

- Eenheidstest.
- Integratietest.
- Systeemtest.

2.1 Eenheidstest

De eenheidstest is er op gericht om een eenheid van een programma met een specifieke functie die kan worden getest. Bij deze test wordt er gecontroleerd of een fase van het systeem zijn functie uitvoert zoals verwacht. De eenheidstests hebben plaatsgevonden tijdens het schrijven van het programma. Er is in deze fase getest of de juiste functies aangeroepen worden.

2.2 Integratietest

De Integratietests beoordelen of de verschillende fases die moeten samenwerken dit zonder problemen doen. Ze controleren of de overgang van de fases goed verlopen, en ook of de goede informatie wordt doorgestuurd. Deze test is uitgevoerd nadat alle functies hun individuele eenheidstests hebben doorstaan. Bij de integratietest ligt dus de nadruk op de verbinding tussen de fases en de gegevens die tussen hen worden uitgewisseld. Zo is er bijvoorbeeld hier getest of rapporten van ingevoerde data uit de vorige fase gemaakt kunnen worden.

2.3 Systeemtest

Na het voltooien van de eenheidstest en de integratietest wordt de systeemtest uitgevoerd, deze dient ter controle of alles goed samenwerkt. Deze systeemtest lijkt op de integratietest, maar heeft een veel breder bereik aangezien alles hierbij betrokken is. Tijdens de systeemtest zijn nog een aantal "fouten" naar voren gekomen. In de volgende hoofdstukken volgt een uitgebreid verslag van de systeemtesten. Voor de eenheidstest is gebruik gemaakt van black box testen. Hierbij wordt goede en foute data in de applicatie ingevoerd. Vervolgens wordt de uitvoer bekeken en bepaald of dit is wat men verwachtte.



Test rapport

3. Add element

Bij deze test is gekeken of het mogelijk is data in de database in te voeren.

3.1 company

Waarde bij Company ingevuld en database gecontroleerd. Alle waardes staan op de verwachte plaats

3.2 Company contact

Waardes bij Company contact ingevuld en database gecontroleerd. Alle waardes zijn zoals verwacht aanwezig.

3.3 Company ship

Waardes bij company ship ingevuld en database gecontroleerd. Begin date company werd niet opgeslagen. Fout hersteld waarna data goed in database verschijnt.

3.4 Company foreign ship

Waarde ingevuld bij company foreign ship. De waardes die ingevuld zijn komen juist in de database terecht.

3.5 Instrument

Waardes ingevuld in formulier. Waarde van checkbox werd niet in database opgeslagen. Fout gevonden en hersteld waarna de data goed in database komt.

3.6 Instrument sort

Waardes ingevuld en database gecontroleerd. De waardes staan juist in de database.

3.7 Shipment

Waardes ingevuld en database gecontroleerd. Waarde van de checkbox werd weer niet in database opgeslagen. Zelfde fout als bij instrument. Code aangepast waarna alles correct wordt opgeslagen.



Test rapport

3.8 Communication

Waardes ingevuld en database gecontroleerd. Waarde van checkbox werd weer niet in database opgeslagen. Probleem kwam al eerder voor en is in de code snel hersteld. Hierna werden alle gegevens correct in de database opgeslagen.

3.9 Communication sort

Waardes ingevuld in database en gegevens gecontroleerd. De gegevens werden juist in de database getoond.

3.10 Software

Waardes ingevuld en database gecontroleerd. Alle data werd juist in de database getoond.

3.11 Software sort

Waardes ingevuld en database gecontroleerd. Alle data werd juist in de database getoond.

3.12 Visit

Waardes ingevuld en database gecontroleerd. De twee checkboxes werden niet goed weergegeven in de database. Fout kwam al eerder voor en is eenvoudig hersteld. Hierna werd alle data juist in de database getoond.

3.13 Ship

Waardes ingevuld en database gecontroleerd. Sort werd niet goed in de database getoond. Kleine typfout in query aangepast waarna de data goed weggeschreven werd.

3.14 Observer

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.



Test rapport

3.15 Foreign visit

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.

3.16 Foreign ship

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.

3.17 Journal

Informatie ingevuld en de database gecontroleerd. Checkbox bij diskette functioneert niet zoals gewenst. Code aangepast waarna alle data juist in de database verschijnt.

3.18 Ship name

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.

3.19 Observer company

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.

3.20 Observer journal

Informatie ingevuld en database gecontroleerd. Alle ingevulde data werd correct weergegeven in de database.

4. View element

Om te kijken of de juiste informatie opgevraagd wordt uit de database. Is gekeken welke data getoond zou moeten worden volgens de gegevens uit de database. Dit bleek in alle gevallen te kloppen.

5. Change element

Om te kijken of elementen veranderd kunnen worden is voor elk element data opgevraagd. Deze data is gewijzigd waarna in de database gekeken is of deze data



Test rapport

inderdaad gewijzigd werd. Dit bleek voor alle gevallen inderdaad waar te zijn. Hieruit is te concluderen dat het mogelijk is de data goed te wijzigen.

6. Delete element

Om te kijken of elementen uit de database verwijderd kunnen worden, zijn elementen de applicatie verwijderd. Vervolgens is gekeken of deze daadwerkelijk uit de database verdwenen waren. Dit bleek het geval, hieruit kunnen we concluderen dat de data inderdaad juist verwijderd kan worden.

7. List

Aangezien een lijst een directe één op één weergave is van een tabel uit de database, is gecontroleerd of de tabel inderdaad volledig getoond wordt. Dit bleek voor alle lijst elementen het geval te zijn.

8. Readfiles

Het inlezen van bestanden is ook getest.

8.1 RTWNDB

Er is getest op het inlezen van RTWNDB bestanden. Hierbij is gekeken of de bestanden juist in de database komen en eventuele eerdere data van dezelfde maand verwijderd wordt. Ook is gekeken of het niet mogelijk is om bestanden met een andere indeling in te lezen. Deze test is succesvol uitgevoerd.

8.2 VOS

Er is getest op het inlezen van VOS bestanden. Hierbij is gekeken of de bestanden juist in de database komen en eventuele eerdere data van dezelfde maand verwijderd wordt. Ook is gekeken of het niet mogelijk is om bestanden met een andere indeling in te lezen. Deze test is succesvol uitgevoerd.

8.3 Nota

Er is getest op het inlezen van Nota bestanden. Hierbij is gekeken of de bestanden juist in de database komen en eventuele eerdere data van dezelfde maand verwijderd wordt. Ook is gekeken of het niet mogelijk is om bestanden met een andere indeling in te lezen. Deze test is succesvol uitgevoerd.

8.4 MONTAB

Er is getest op het inlezen van MONTAB bestanden. Hierbij is gekeken of de bestanden juist in de database komen en eventuele eerdere data van dezelfde maand verwijderd wordt. Ook is gekeken of het niet mogelijk is om bestanden met een andere indeling in te lezen. Deze test is succesvol uitgevoerd.



Test rapport

8.5 *journaal diskette*

Er is getest op het inlezen van Journaal diskette bestanden. Hierbij is gekeken of de bestanden juist in de database komt en of de fouten in het bestand herkend worden. Tevens is gekeken of de fouten in het programma hersteld kunnen worden, en een foutloos bestand gegeneerd wordt. Ook is gekeken of het niet mogelijk is om bestanden met een andere indeling in te lezen. Deze test is succesvol uitgevoerd.

8.6 *get journal files*

Bij deze test is gekeken of de eerder ingelezen journaalbestanden als een zip bestand gedownload kunnen worden. Dit bleek het geval te zijn.

9. Reports

Rapporten met data zijn bekeken. Aan de hand van de data in de database is gekeken of de juiste data in het rapport stond. Aan de hand hiervan kon de conclusie getrokken worden dat dit inderdaad het geval is.

10. SQL builder

Getest is of Query's op de database uitgevoerd konden worden. Tevens is getest of deze query's konden worden opgeslagen, opnieuw konden worden uitgevoerd en worden verwijderd. Query's bleken alleen niet verwijderd te kunnen worden als er een spatie in de naam aanwezig was. Verder is deze test succesvol uitgevoerd.

11. statistieken

Het formulier voor statistieken is ingevuld. Er werd hierbij zoals verwacht een statistiek van de verwachte data uitgevoerd.

12. Ondersteuning browsers

Gekeken is naar de ondersteuning van de applicatie met verschillende browsers. Naar voren is gekomen dat de applicatie functioneert zoals verwacht met de laatste versie van Microsoft internet Explorer, Mozilla Firefox en Google Chrome.