

Ontwikkelen van datawarehousing & dashboard applicatie bij 42

Bijlagen



DE HAAGSE
HOGESCHOOL

Inhoud

Requirements V1.0.....	1
Plan van Aanpak V1.0.....	4
Sprint 1 Retrospective V1.0.....	8
Spring Boot presentatie.....	11
Requirements V4.0.....	28
Klassendiagram sprint 3.....	46
Klassendiagram sprint 4.....	50
Sprint 4 retrospective V1.0.....	54
Requirements – final.....	58
Klassendiagram database – final.....	81
Klassendiagram systeem – final.....	83
Evaluatie bedrijfsbegeleider.....	87



Requirements V1.0

Bijlage A

Project Requirements

Functional requirements

Havik

#	Requirement	Oorsprong
1.1.	Havik moet de data uit Raaf kunnen laten zien.	Case
1.2.	Havik moet berekeningen kunnen toepassen op de data uit Raaf.	Case
1.3.1.	Havik moet kunnen laten zien hoe het ziekteverzuim in het verleden is geweest.	Case
1.3.2.	Havik moet het ziekteverzuim van verschillende jaren kunnen laten zien.	Case
1.3.3.	Havik moet het ziekteverzuim van verschillende maanden kunnen laten zien.	Case
1.4.	Havik moet gegevens uit het verleden kunnen opvragen.	Interview #1 Opdrachtgever
1.5.	Sommige gegevens moeten handmatig ingevoerd kunnen worden.	Interview #1 Opdrachtgever
1.6.1.	Havik moet kunnen laten zien hoeveel werknemers er in een jaar waren.	Interview #1 Opdrachtgever
1.6.2.	Havik moet kunnen laten zien hoeveel werknemers er in een maand waren.	Interview #1 Opdrachtgever
1.6.3.	Havik moet in de gevallen van 1.6.1. & 1.6.2. kunnen laten zien welke werknemers dit waren.	Interview #1 Opdrachtgever
1.6.4.	Havik moet de hoeveelheid werknemers kunnen afzetten tegen de omzet in één grafiek.	Interview #1 Opdrachtgever
1.7.1.	Havik moet de FTE per jaar kunnen laten zien.	Interview #1 Opdrachtgever
1.7.2.	Havik moet de FTE per maand kunnen laten zien.	Interview #1 Opdrachtgever
1.8.1.	Havik moet de omzet per jaar kunnen laten zien.	Interview #1 Opdrachtgever
1.8.2.	Havik moet de omzet per maand kunnen laten zien.	Interview #1 Opdrachtgever
1.9.	Havik moet kunnen laten zien wat voor soort projecten er lopen.	Interview #1 Opdrachtgever
1.10.1.	Havik moet de top 10 klanten laten zien.	Interview #1 Opdrachtgever
1.10.2.	De top 10 klanten moet gesorteerd kunnen worden op basis van facturatie.	Interview #1 Opdrachtgever
1.10.3.	De top 10 klanten, gesorteerd op facturatie, moet vervolgens gesorteerd kunnen worden op basis van de hoeveelheid projecten.	Interview #1 Opdrachtgever
1.10.4.	De top 10 klanten, gesorteerd op facturatie, moet vervolgens gesorteerd kunnen worden op basis van de hoeveelheid uren.	Interview #1 Opdrachtgever
1.10.5.	De top 10 klanten, gesorteerd op facturatie, moet vervolgens gesorteerd kunnen worden op basis van de service.	Interview #1 Opdrachtgever
1.11.	Havik moet de behaalde omzet tegen de originele begroting kunnen	Interview #1

uitzetten in één grafiek.

Opdrachtgever

De gegevens die bij 1.5. bedoeld worden zijn nog niet bekend; wanneer deze bekend zijn wordt deze requirement verder uitgebreid/verduidelijkt.

Raaf

#	Requirement	Oorsprong
2.1.	Raaf moet data kunnen ophalen.	Case
2.2.	Raaf moet data kunnen opslaan.	Case
2.3.	Kentallen moeten handmatig opgeslagen kunnen worden.	Interview #1 Opdrachtgever
2.4.	Bepaalde data moet achteraf toegevoegd kunnen worden.	Interview #1 Opdrachtgever

De gegevens die bij 2.4. bedoeld worden zijn nog niet bekend; wanneer deze bekend zijn wordt deze requirement verder uitgebreid/verduidelijkt.

Non-functional requirements

Numer	Requirement	Oorsprong
3.1.	De systemen moeten opgezet worden m.b.v. Springboot.	Afstudeerplan
3.2.	De systemen moeten gemaakt worden in Java.	Afstudeerplan
3.3.	De hoeveelheid data die toegevoegd wordt in de database is irrelevant.	Interview #1 Opdrachtgever

Business rules

Numer	Business rule
4.1.	Één FTE is 40 uur.
4.2.	FTE is proportioneel per medewerker; een medewerker die 32 per week werkt staat gelijk aan 0.8 FTE.
4.3.	Door veranderingen in berekeningen moeten sommige getallen niet meegeteld worden in bepaalde berekeningen.
4.4.	Een jaar wordt in juni van het volgende jaar afgesloten; vanaf dan is het niet meer mogelijk om cijfers te veranderen.

De gegevens die bij 4.3. bedoeld worden zijn nog niet bekend; wanneer deze bekend zijn wordt deze business rule verder uitgebreid/verduidelijkt en worden bijbehorende requirements toegevoegd.



Plan van Aanpak V1.0

Bijlage B

Aanleiding & Probleemstelling

De probleemstelling bestaat uit twee onderdelen. Ten eerste is er geen centrale plaats voor het bekijken van de bedrijfsgegevens. Hiermee wordt bedoeld dat er op verschillende systemen binnen het bedrijf ingelogd moet worden om de data van dat systeem te kunnen bekijken. Dit gaat ten koste van het overzicht en zorgt ervoor dat bepaalde data moeilijk aan elkaar gekoppeld kan worden.

Het tweede probleem is dat de huidige afspraken vastgesteld zijn op mail of alleen onthouden worden door de betrokkenen. Er is geen centrale plaats en toetsing is soms te subjectief. Dit gaat ten koste van de efficiëntie en maakt het sturen van een groot bedrijf moeilijker.

Doelstelling

Het doel van de opdracht is dat er twee nieuwe systemen komen. Systeem "Havik" moet data van de andere systemen ophalen en zelf opslaan. Vervolgens moet een nieuw systeem, "Raaf", gemaakt worden die deze data laat zien samen met nieuwe formules die hierop toegepast kunnen worden. Een voorbeeld van zo'n nieuwe formule is dat het zichtbaar moet worden of het ziekteverzuim is toegenomen of afgenomen het afgelopen jaar (en met hoeveel).

Projectinrichting

Tijdens het project wordt er gewerkt met SCRUM. Net zoals bij de rest van het bedrijf worden sprints van 2 weken aangehouden; in totaal zijn er negen sprints (inclusief de 0 sprint). De sprints zullen om en om toegeweid worden aan het andere systeem: na een sprint toegeweid aan Havik komt een sprint toegeweid aan Raaf en vice versa.

Voor het programmeren zal IntelliJ IDEA 15 gebruikt worden.

Het project zal 17 kalenderweken duren. De startdatum is 8 februari 2016; de einddatum 3 juni.

Scope

Binnen de scope van het project vallen alle onderdelen die in het volgende hoofdstuk benoemd worden. Onderdeel is om de systemen Havik & Raaf te maken. Raaf moet de database kunnen bevragen voor data en nieuwe data in de database kunnen plaatsen. Havik moet kunnen communiceren met Raaf en Postduif om deze data te verkrijgen en vervolgens te kunnen laten zien. Het ontwerpen van nieuwe database tabellen die benodigd zijn voor Raaf valt ook binnen de scope. Verdere specificaties van de scope zullen bepaald worden per sprint; vooraf zal bepaald worden welke requirements er in de sprint behandeld zullen worden. Vervolgens kan de scope hierop ingesteld worden.

Eindresultaat

Aan het eind van het project horen de volgende producten gemaakt en opgeleverd te zijn:

- Plan van Aanpak
- Requirements document
- Database & Systeem ontwerpen
- Raaf & Havik
- Tests & testontwerpen

Na dit project is het mogelijk voor het management om een beter overzicht te verkrijgen van verschillende statistieken over het bedrijf. Afspraken kunnen beter getoetst worden en andere statistieken met betrekking tot personeel en omzet zullen beter en sneller beschikbaar zijn; dit zorgt voor een tijdswinst bij het management.

Risico's

Tijdens het project moet er gewerkt worden met Java. Dit is een taal waar ik lang niet mee gewerkt heb. Het kunnen sommige elementen van programmeren langer duren dan verwacht. Om dit te beperken wordt er in de eerste week veel aandacht besteed aan het uitbreiden van mijn Java kennis.

Ditselfde geldt ook voor Spring en SpringBoot. Bij SpringBoot geldt echter dat dit niet in de sessies van de eerste week naar voren komt. Omdat dit ook een redelijk nieuwe techniek bij 42 zou zijn is de kennis hierover ook vrijwel niet binnen 42 te vinden; er is één medewerker die er ervaring mee heeft die één keer per week binnen bij 42 zit (buiten deze dag zit deze medewerker op locatie). Om de problemen die dit kan veroorzaken zo veel mogelijk te verminderen besteed ik zelf veel tijd aan het leren van SpringBoot in de eerste weken. Onderdelen hiervan zijn de manuals van Spring(Boot) en tutorials door te nemen.

Planning

Er wordt gewerkt met sprints van twee weken. De eerste week van het project is de 0-sprint; gebruikt voor het opzetten van het project. De 0-sprint is de enige sprint die één week lang is.

Globale Planning

Sprint 0 – 8 februari – 12 februari

- Opstellen Plan van Aanpak
- Opstellen eerste versie requirements
- Kennis verkrijgen Java (dit wordt gedaan d.m.v. sessies met Robert Bor)
- Kennis verkrijgen over de code van 42
- Werken aan stageverslag

Sprint 1 – 15 februari – 26 februari

- Requirements opstellen
- Opstellen use cases
- Opstellen datadictionary
- Werken aan stageverslag

De (globale) planning van de latere sprints zullen op basis van de opgestelde en geprioriteerde requirements zijn.

Detailplanning Sprint 0

Kennis verkrijgen over de code van 42	8 & 9 februari
Kennis verkrijgen Java	9, 10 & 12 februari
Opstellen eerste versie requirements	10 & 11 februari
Opstellen Plan van Aanpak	11 februari
Werken aan stageverslag	12 februari

Detailplanning Sprint 1

Opstellen requirements	15-18 & 22-25 februari
Opstellen datadictionary	18 & 24 februari
Opstellen use cases	23 & 25 februari
Tijd die niet besteed kan worden aan requirements (bv. omdat een meeting nodig is voor verdere input) wordt besteed aan Spring, SpringBoot & Havik	-
Werken aan stageverslag	19 & 26 februari

Sprint 1 Retrospective V1.0

Bijlage C

Eindresultaat Sprint #1

Requirements

Een groot gedeelte van de requirements zijn opgesteld; verdere requirements zijn specifiek aan de widgets en zullen in de sprints toegewijd aan deze widgets opgesteld worden. De requirements zijn driemaal naar de opdrachtgever opgestuurd; de review van de laatste versie is nog in afwachting. Na goedkeuring hiervan kunnen deze geprioritiseerd worden. Een semi-prioritisering is al besproken; de opdrachtgever heeft al aangegeven om met de widgets met betrekking tot de hoeveelheid werknemers of ziektepercentage te beginnen.

Eventuele “vage” definities zijn opgenomen voor duidelijkheid bij zowel de opdrachtgever als de opdrachtnemer.

Use cases stonden ingepland om gemaakt te worden; deze zijn niet gemaakt. Omdat de applicatie vooral een dashboard applicatie is met weinig functionaliteit behalve widgets te laten zien zouden use cases niet nuttig zijn. Hier is over gesproken met de begeleider.

Kennis

In sessions met Robert is er veel kennis bijgebracht over Java, Spring & de indeling van klassen binnen de systemen van 42. Sessions met Maarten hebben veel kennis bijgebracht over JavaScript; AngularJS sessions komen binnenkort.

Havik heb ik met behulp van Poujah onderzoek naar gedaan. Hieruit is gebleken dat er al een groot gedeelte van Havik bestaat. De code van Havik zit ook duidelijk in elkaar; widgets toevoegen is flexibel opgebouwd en zou zonder al te veel problemen moeten verlopen.

Spring-Boot heb ik zelf onderzoek naar gedaan; hier is een redelijke hoeveelheid kennis over vergaard. I.v.m. het grote gedeelte van Havik dat al bestaat is er besloten dat Spring-Boot niet terugkomt binnen het project zelf. (I.p.v. hiervan doe ik onderzoek hiernaar en wordt er een presentatie gegeven op de demo in de 2e week en op de KTM.)

Evaluatie

De requirements zijn nog niet geprioritiseerd; voor deze sprint is al duidelijk wat er gemaakt gaat worden, maar voor de komende sprints moet dit zeker gebeuren. Het proces zelf was duidelijk en volgens planning verlopen. De afwezigheid van de use cases is jammer, maar te verklaren.

De hoeveelheid kennis die is opgedaan is ontzettend hoog. Het begrijpen van Java & Spring is een essentieel onderdeel van dit project; dat de sessions perfect zijn gelopen is dus een geweldige uitkomst.

De sessions met Maarten over JavaScript zijn ook goed verlopen, alhoewel dit wel een lastiger concept is. Tijdens het uitbreiden van de Havik-UI zal moeten blijken of dit concept volledig over is gekomen; mijn verwachting is van wel. Blijk ik hier toch nog problemen mee te hebben dan is Maarten “altijd” bereikbaar.



Spring Boot presentatie

Bijlage D

Spring-Boot

Jordi Romeijn
Afstudeerder

Spring

- Nieuw “Hello World” Project

- Imports

- Spring
 - Tomcat

- Configuration Files

- @EnableWebMvc
 - @ComponentScan

- Controller

- @RestController
 - @RequestMapping



```
package demo.spring.controller;

import ...

@RestController
@RequestMapping("/hello")
public class HelloWorldController {
    @RequestMapping("/hello")
    public String home() { return "Hello World!"; }
}

@Configuration
@ComponentScan(basePackageClasses = ApplicationConfig.class,
    excludeFilters = {
        @ComponentScan.Filter({ ControllerAdvice.class,
            Controller.class,
            RestController.class })
    })
class ApplicationConfig {

}

@Configuration
@EnableWebMvc
@ComponentScan(basePackageClasses = demo.spring.ApplicationConfig.class,
    includeFilters = @ComponentScan.Filter({ RestController.class })
)
class WebMvcConfig extends WebMvcConfigurerAdapter {

}
```

Wat is Spring-Boot (niet)

- Toevoeging
- Geen vervanging!!
- Nieuw project
- Minder configuratie
- Sneller programmeren :)

Spring

- Nieuw “Hello World” Project

- Imports

- Spring
 - Tomcat

- Configuration Files

- @EnableWebMvc
 - @ComponentScan

- Controller

- @RestController
 - @RequestMapping



Spring-Boot (1)

- Nieuw “Hello World” Project
 - Imports
 - Spring-Boot-Starter
 - Spring-Boot-Web
 - ~~Configuration Files~~
 - Controller
 - @RestController
 - @RequestMapping
 - @EnableAutoConfiguration
 - Opinionated



spring
boot

```
package demo.spring.controller;

import ...

@RestController
@RequestMapping("/hello")
public class HelloWorldController {
    @RequestMapping("/hello")
    public String home() { return "Hello World!"; }
}

@Configuration
@ComponentScan(basePackageClasses = ApplicationConfig.class,
    excludeFilters = {
        @ComponentScan.Filter({ ControllerAdvice.class,
            Controller.class,
            RestController.class })
    })
class ApplicationConfig {

}

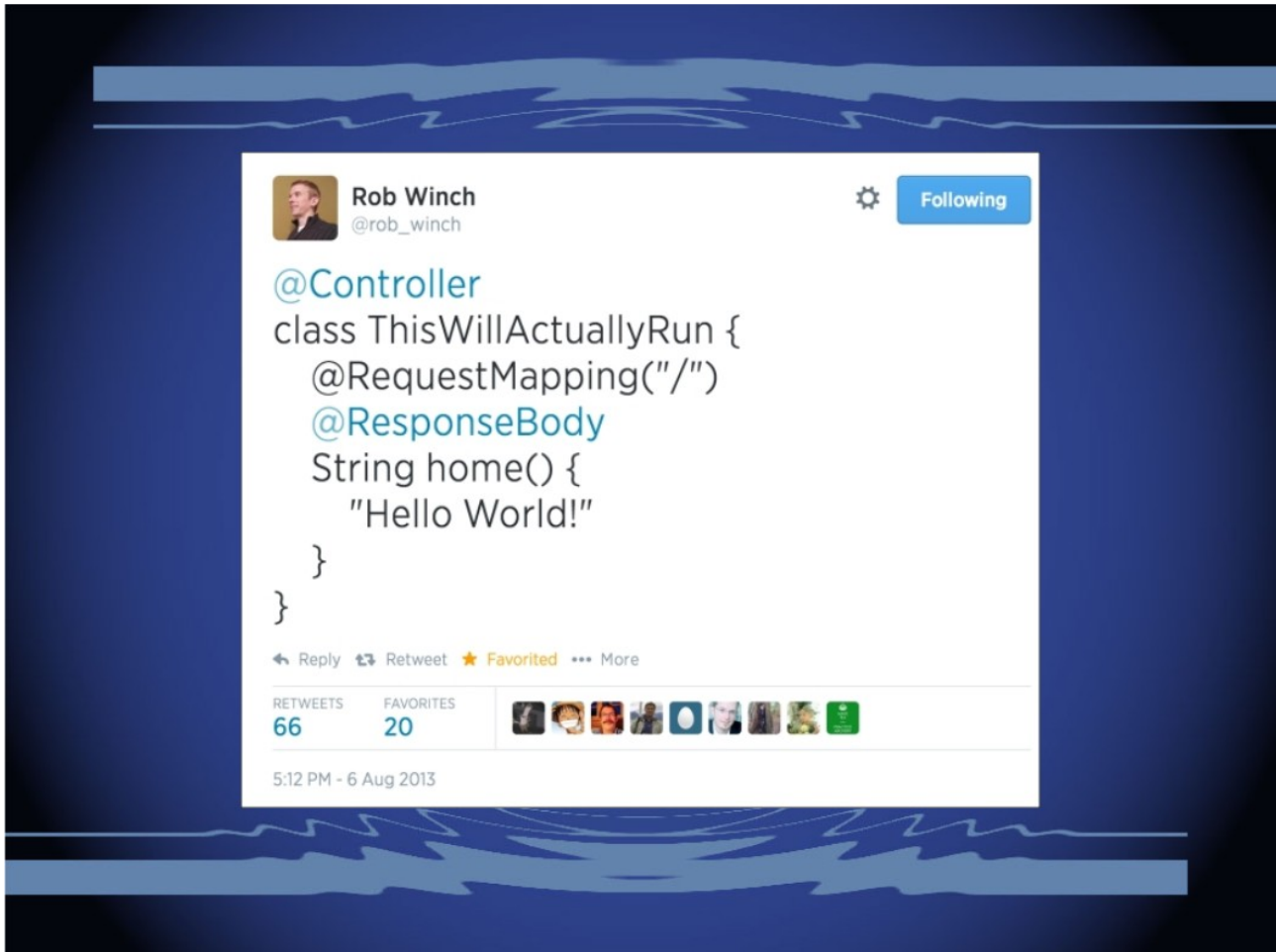
@Configuration
@EnableWebMvc
@ComponentScan(basePackageClasses = demo.spring.ApplicationConfig.class,
    includeFilters = @ComponentScan.Filter({ RestController.class })
)
class WebMvcConfig extends WebMvcConfigurerAdapter {

}
```

```
package demo.springboot.controller;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@EnableAutoConfiguration
@RequestMapping("/demo")
public class DemonstrationController {
    @RequestMapping("/hello")
    String home() { return "Hello World!"; }
}
```



Spring-Boot (2)

- Configuratie aanpassen??

- Geen probleem

- `@EnableAutoConfiguration(exclude={foo.class})`
 - Eigen beans toevoegen
 - Bv. DataSource
 - Eigen config file toevoegen



spring
boot

Spring-Boot Starters (1)

- Spring-boot-starter
 - Autoconfiguration
 - Logging
 - YAML
- Spring-boot-starter-actuator
 - Server
 - Security
 - Logging
 - Externalized configuration
 - Management endpoints
 - And more!!



Spring-Boot Starters (2)

- Batch
 - Batch
 - HSQLDB
- Data-jpa
 - Spring-data-jpa
 - Hibernate
- Test
 - Spring test
 - Junit
 - +2
- Web
 - Tomcat
 - Spring-webmvc
 - And more!!



spring
boot

Name	Description
<code>spring-boot-starter</code>	The core Spring Boot starter, including auto-configuration support, logging and YAML.
<code>spring-boot-starter-actuator</code>	Production ready features to help you monitor and manage your application.
<code>spring-boot-starter-amqp</code>	Support for the "Advanced Message Queuing Protocol" via <code>spring-rabbit</code> .
<code>spring-boot-starter-aop</code>	Support for aspect-oriented programming including <code>spring-aop</code> and AspectJ.
<code>spring-boot-starter-artemis</code>	Support for "Java Message Service API" via Apache Artemis.
<code>spring-boot-starter-batch</code>	Support for "Spring Batch" including HSQLDB database.
<code>spring-boot-starter-cache</code>	Support for Spring's Cache abstraction.
<code>spring-boot-starter-cloud-connectors</code>	Support for "Spring Cloud Connectors" which simplifies connecting to services in cloud platforms like Cloud Foundry and Heroku.
<code>spring-boot-starter-data-elasticsearch</code>	Support for the Elasticsearch search and analytics engine including <code>spring-data-elasticsearch</code> .
<code>spring-boot-starter-data-gemfire</code>	Support for the GemFire distributed data store including <code>spring-data-gemfire</code> .
<code>spring-boot-starter-data-jpa</code>	Support for the "Java Persistence API" including <code>spring-data-jpa</code> , <code>spring-orm</code> and Hibernate.
<code>spring-boot-starter-data-mongodb</code>	Support for the MongoDB NoSQL Database, including <code>spring-data-mongodb</code> .
<code>spring-boot-starter-data-redis</code>	Support for the REDIS key-value data store, including <code>spring-data-redis</code> .
<code>spring-boot-starter-data-rest</code>	Support for exposing Spring Data repositories over REST via <code>spring-data-rest-webmvc</code> .
<code>spring-boot-starter-data-solr</code>	Support for the Apache Solr search platform, including <code>spring-data-solr</code> .
<code>spring-boot-starter-freemarker</code>	Support for the FreeMarker templating engine.
<code>spring-boot-starter-groovy-templates</code>	Support for the Groovy templating engine.
<code>spring-boot-starter-hateoas</code>	Support for HATEOAS-based RESTful services via <code>spring-hateoas</code> .
<code>spring-boot-starter-hornetq</code>	Support for "Java Message Service API" via HornetQ.
<code>spring-boot-starter-integration</code>	Support for common <code>spring-integration</code> modules.
<code>spring-boot-starter-jdbc</code>	Support for JDBC databases.
<code>spring-boot-starter-jersey</code>	Support for the Jersey RESTful Web Services framework.
<code>spring-boot-starter-jta-atomikos</code>	Support for JTA distributed transactions via Atomikos.
<code>spring-boot-starter-jta-bitronix</code>	Support for JTA distributed transactions via Bitronix.
<code>spring-boot-starter-mail</code>	Support for <code>javax.mail</code> .
<code>spring-boot-starter-mobile</code>	Support for <code>spring-mobile</code> .
<code>spring-boot-starter-mustache</code>	Support for the Mustache templating engine.
<code>spring-boot-starter-security</code>	Support for <code>spring-security</code> .
<code>spring-boot-starter-social-facebook</code>	Support for <code>spring-social-facebook</code> .
<code>spring-boot-starter-social-linkedin</code>	Support for <code>spring-social-linkedin</code> .
<code>spring-boot-starter-social-twitter</code>	Support for <code>spring-social-twitter</code> .
<code>spring-boot-starter-test</code>	Support for common test dependencies, including JUnit, Hamcrest and Mockito along with the <code>spring-test</code> module.
<code>spring-boot-starter-thymeleaf</code>	Support for the Thymeleaf templating engine, including integration with Spring.
<code>spring-boot-starter-velocity</code>	Support for the Velocity templating engine. Deprecated in 1.4.
<code>spring-boot-starter-web</code>	Support for full-stack web development, including Tomcat and <code>spring-webmvc</code> .
<code>spring-boot-starter-websocket</code>	Support for WebSocket development.
<code>spring-boot-starter-ws</code>	Support for Spring Web Services.

Spring-Boot Starters (3)

- Zelf toevoegen
 - Eenmaal doen binnen 42
 - Gebruiken bij alle toekomstige projecten
 - Meest gebruikte dependencies
 - Meest gebruikte configuraties



spring
boot

Recap

- Nieuw project
- Autoconfiguratie → minder configuratie
- Opinionated
- Sneller (beginnen met) programmeren

Recap (2)/Meer informatie

- Klein gedeelte behandeld
 - Starter POMs
- Reference guide op spring.io



Requirements V4.0

Bijlage E

Definities

Begrip	Afkorting	Definitie
Full-Time Equivalent	FTE	Dienstverband van een medewerker ten opzichte van een volledige werkweek van 40 uur. Een medewerker die 40 uur per week werkt staat gelijk aan 1 FTE; een medewerker die 32 uur per week werkt staat gelijk aan 0.8 FTE.
Functie		Dat wat uiteindelijk culmineert in een widget.
Grootboekrekening		Een herkenningnummer voor boekhouding waar meerdere posten onder gegroepeerd kunnen worden. Iedere factuur is gekoppeld aan een grootboekrekening. Voor de huidige specs kan worden aangenomen dat er een 1-1 relatie is. Strikt genomen is een factuur verbonden met 3 of meer grootboekrekeningen, maar vanuit Snip is er maar 1 per factuur belangrijk (en mogelijk).
Prijs Index Cijfer		“Een prijsindex is een genormaliseerd gemiddelde (meestal een gewogen gemiddelde) van prijzen voor een gegeven klasse van goederen of diensten in een gegeven regio, gedurende een gegeven tijdsinterval. Het is een statistiek die ontworpen is om te vergelijken hoe, als geheel genomen, deze prijzen tussen tijdsperiodes of geografische locaties kunnen verschillen.” - https://nl.wikipedia.org/wiki/Prijsindex
Statische/historische gegevens		Gegevens die niet meer veranderen, meestal omdat er genoeg tijd over heen is gegaan dat deze niet meer hoeven, of zelfs mogen, aangepast te worden.
Verplichte verlofdagen		Werkdagen die een medewerker verplicht vrij krijgt, bijvoorbeeld de dag na hemelvaart.
Werkdagen		De werkbare dagen in een week; kan per medewerker anders zijn.
Widget		Een klein onderdeel van een webpagina die iets nuttigs met informatie opgehaald uit andere locaties doet. Geeft deze informatie vervolgens ook weer.
Ziekteverzuim/ Ziektepercentage		Ziekteverzuim wordt per maand berekend. Het aantal uren ziek is bekend, het geldige weekrooster van de medewerker staat vast. Het ziektepercentage is het percentage ziekuren ten opzichte van het aantal uren dat maximaal gewerkt had kunnen worden in deze periode. Gewerkte uren: is de optelling van alle uren waarop niet ziek-uren geschreven zijn. Ziekte uren: is de optelling van alle uren waarop ziek-uren geschreven zijn.

Maximaal werkbare uren: is de optelling van alle uren in alle dagen waarop contractueel gewerkt zou moeten worden. Dagen waarop contractueel gewerkt zou moeten worden bestaan uit de contractdagen min de feestdagen.

Ziekteverzuim: ziekte uren / maximaal werkbare uren * 100% (individueel per medewerker).

Voorbeeld:

Medewerker A: 160 uren werkbaar; 16 uur ziek; 10% ziek

Medewerker B: 140 uren werkbaar; 14 uur ziek; 10% ziek

Medewerker C: 20 uren werkbaar; 10 uur ziek; 50% ziek

<Scenario A>

Maandverzuim: het gemiddelde ziekteverzuim van alle medewerkers in een maand.

In het voorbeeld: 23.3% (gemiddelde van de percentages)

</Scenario A>

<Scenario B>

Maandverzuim: $\frac{\sum \text{maximaal werkbare uren per maand (van alle medewerkers)}}{\sum \text{ziekte-uren per maand (van alle medewerkers)}} * 100\%$

In het voorbeeld: 12.5% (uren bij elkaar opgeteld)

</Scenario B>

Project Requirements

Functional requirements

#	Benodigd Requirement	Oorsprong
1.1.	Havik moet grafieken kunnen laten zien.	Case
1.2.	Havik moet tabellen kunnen laten zien.	Case
1.3.	1.1. Havik moet een lijndiagram kunnen weergeven die het ziektepercentage van de afgelopen 10 jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet. Elke lijn stelt het ziektepercentage van één jaar voor bij één werkmaatschappij. Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.	Review #1 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever
1.4.	1.1. Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien. Havik moet een staafdiagram kunnen weergeven die het ziektepercentage van één jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet. Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij. De werkmaatschappijen moeten apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December). Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever Review #1 Opdrachtgever Interview #3 Opdrachtgever

#	Benodigd Requirement	Oorsprong
1.5.	1.1.	Havik moet in een staafdiagram alle werknemers van een jaar kunnen laten zien. Hierbij wordt de hoeveelheid werknemers uitgezet tegenover de maanden. Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij. De werkmaatschappijen moeten apart aan en uit te zetten zijn. Ook moeten de werknemersgroepen apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December). De selectie moet niet verder terug in de tijd kunnen gaan dan 2003. De selectie mag ook niet verder in te tijd kunnen gaan dan het huidige jaar. Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien. Werknemers die later in de maand zijn aangenomen (dus niet de gehele maand hebben gewerkt) tellen mee voor de hele maand; deze hoeven niet speciaal aangegeven worden.
1.5.1.	1.5.	Bij de bovenstaande grafiek moet de gebruiker de optie hebben om in plaats van de hoeveelheid werknemers de hoeveelheid FTE te zien. Alle andere opties moeten ook bij deze grafiek mogelijk zijn.

#	Benodigd Requirement	Oorsprong
1.6.	1.1. Havik moet in een staafdiagram het totale facturatie bedrag per jaar kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar. Elke staaf is hierin opgedeeld in verschillende partities. De gebruiker moet de optie hebben om de staven op te delen per werkmaatschappij of per grootboeknummer. De staaf-partities moeten apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Q1 t/m Q4). Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij of grootboeknummer laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien.	Interview #1 Opdrachtgever Interview #3 Opdrachtgever Review #1 Opdrachtgever Review #1 Opdrachtgever Interview #3 Opdrachtgever Review #1 Interview #3 Opdrachtgever
1.7.	1.1. Havik moet in een lijndiagram het totale facturatie bedrag van meerdere jaren kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar. Elke lijn stelt hierbij het facturatiebedrag van één werkmaatschappij van één jaar voor. Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (het facturatiebedrag van alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn. Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever

#	Benodigd Requirement	Oorsprong
1.8.	1.2. Havik moet in een tabel alle klanten kunnen weergeven met het totale facturatiebedrag per klant. Ook de hoeveelheid projecten die bij deze klanten lopen en de hoeveelheid uren die besteed wordt aan een klant moet zichtbaar zijn. De tabel moet gesorteerd kunnen worden op de drie zichtbare gegevens. (Voorbeeld: de klant met het hoogste facturatie bedrag staat bovenaan, etc.) Het moet mogelijk zijn om werkmaatschappijen aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezette werkmaatschappij niet meegenomen in de tabel. Het moet mogelijk zijn om grootboeknummers aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezet grootboeknummer niet meegenomen in de tabel.	Interview #1 Opdrachtgever Interview #3 Opdrachtgever Interview #1 Opdrachtgever Interview #1 Opdrachtgever Review #1 Opdrachtgever Interview #3 Opdrachtgever Interview #1 Opdrachtgever
1.9.	Havik moet in een tabel de prijsindexcijfers van verschillende statistieken kunnen laten zien. Mogelijkheden zijn bijvoorbeeld de prijsindexcijfer van de kosten, tarieven, bruto salaris, totale kosten, Deze requirement wordt later verder uitgewerkt; voor nu is deze niet relevant of interessant.	Interview #4 Opdrachtgever
1.10.	Als er in de widgets geklikt wordt op historische data moet het scherm binnen Havik zelf zichtbaar worden. Wordt er geklikt op data van de huidige maand moet er een nieuw scherm geopend worden welke uitkomt in het relevante systeem (deeplink); in het relevante systeem moet meteen de geselecteerde data zichtbaar zijn (de gebruiker hoeft niet eerst zelf te zoeken). Zie de bijborende tabel voor de systemen waar naar gelinkt moet worden voor de verschillende types gegevens.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever
1.11.	Er moet een scherm zijn waar de admin instellingen kan wijzigen.	Interview #3 Opdrachtgever Interview #4 Opdrachtgever
1.12.	1.11. In het instellingen scherm moet er ingesteld kunnen worden wat de eerste acht widgets op het dashboard zijn.	Interview #3 Opdrachtgever
1.13.	Widgets na de eerste acht moeten op alfabetische volgorde staan. Widgets die handmatig ingesteld staan op de eerste 8 plekken moeten hier niet in terugkomen; er staan geen dubbele widgets in.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever

#	Benodigd Requirement	Oorsprong
1.14.	1.11. Widgets kunnen een drempelwaarde hebben waarbij deze zichtbaar een waarschuwing moeten afgeven als de dat hiervan boven of onder de drempelwaarde komt. Deze drempelwaardes moeten per widget handmatig ingesteld kunnen worden in het instellingenschermb. Welke widgets er een drempelwaarde kunnen hebben staat nog niet vast. De drempelwaarde geldt alleen voor de laatste mogelijke periode (oftewel “nu”). De periode kan per widget anders zijn; zie de bijbehorende tabel voor de periodes.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever
1.15.	Er moet een functie zijn die de weergegeven data voor alle widgets herberekend. De admin moet deze zelf kunnen activeren	Interview #3 Opdrachtgever Interview #5 Opdrachtgever
1.16.	1.11. In het instellingenschermb moet het mogelijk zijn om handmatig data in te voeren voor berekeningen. Wanneer er een herberekening plaats vindt moet de gebruiker de optie krijgen om de handmatig ingevoerde data te gebruiken voor de berekening of de berekening uit te voeren met de data die in de database staat. Zie de bijbehorende tabel voor de gegevens die handmatig ingevoerd moeten kunnen worden.	Interview #3 Opdrachtgever

“Havik moet een lijndiagram kunnen weergeven die het ziektepercentage van de afgelopen 10 jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet.

Elke lijn stelt het ziektepercentage van één jaar voor bij één werkmaatschappij.

Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.

Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.”

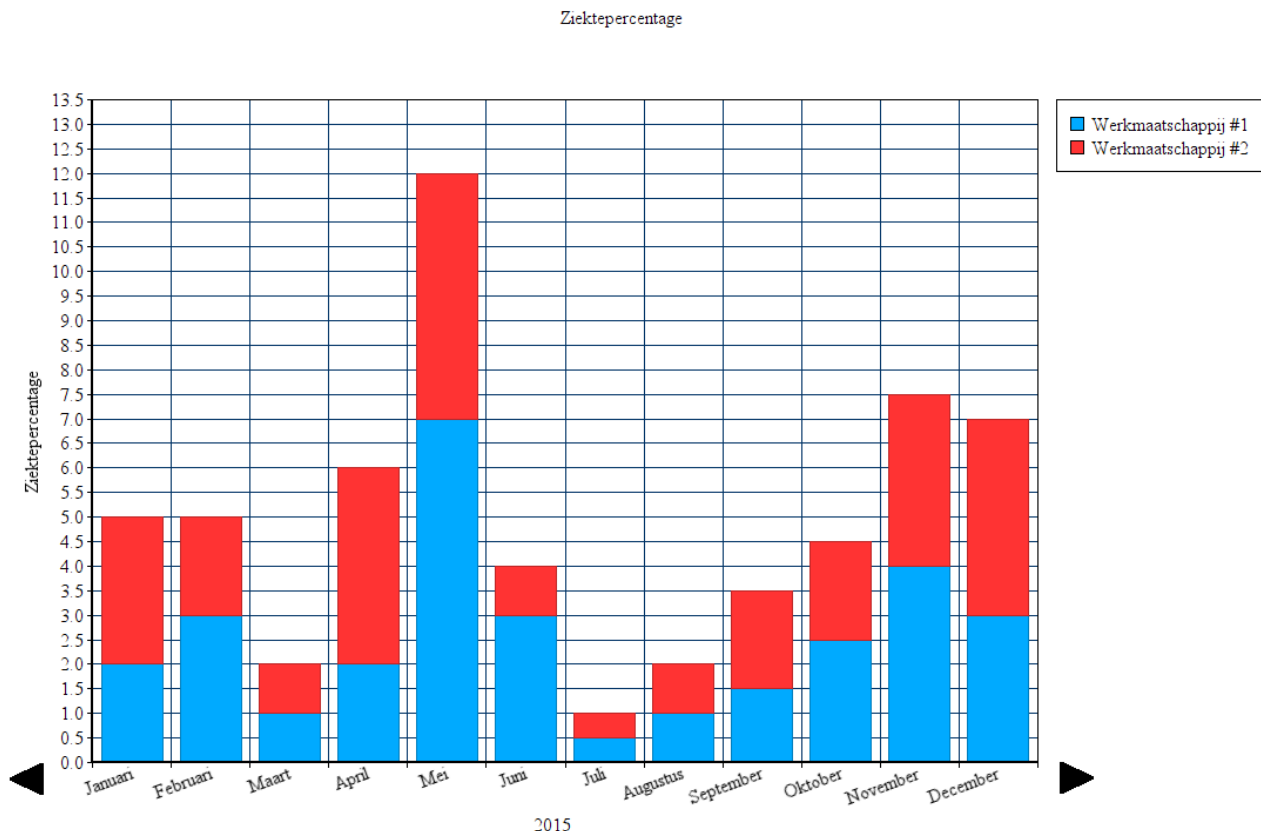


Illustration 2: Requirement 1.4.

“Havik moet een staafdiagram kunnen weergeven die het ziektepercentage van één jaar laat zien.

Hierbij wordt het percentage tegenover de maanden uitgezet.

Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij.

De werkmaatschappijen moeten apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December).

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.”

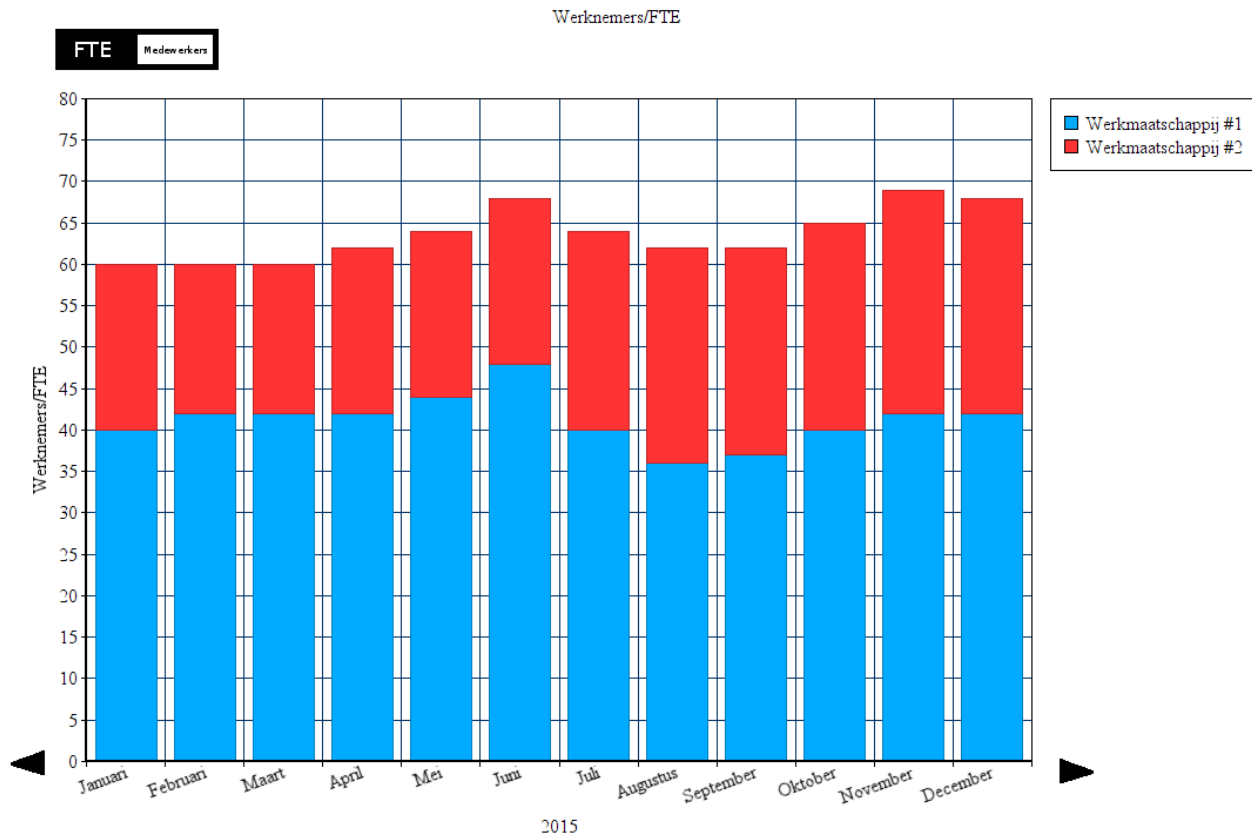


Illustration 3: Requirement 1.5.

“Havik moet in een staafdiagram alle werknemers van een jaar kunnen laten zien. Hierbij wordt de hoeveelheid werknemers uitgezet tegenover de maanden.

Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij.

De werkmaatschappijen moeten apart aan en uit te zetten zijn. Ook moeten de werknemersgroepen apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December).

De selectie moet niet verder terug in de tijd kunnen gaan dan 2003.

De selectie mag ook niet verder in de tijd kunnen gaan dan het huidige jaar.

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.

Werknemers die later in de maand zijn aangenomen (dus niet de gehele maand hebben gewerkt) tellen mee voor de hele maand; deze hoeven niet speciaal aangegeven worden.”

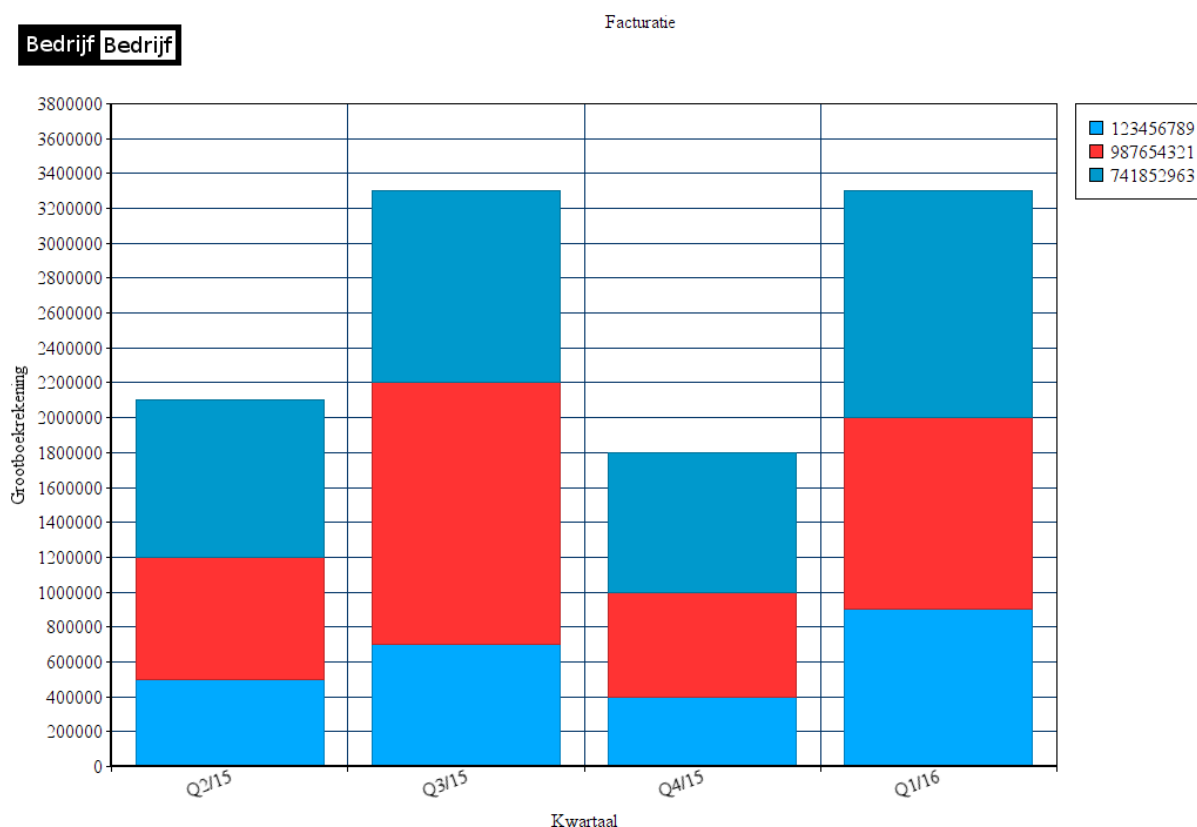


Illustration 4: Requirement 1.6.

“Havik moet in een staafdiagram het totale facturatie bedrag per jaar kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar.

Elke staaf is hierin opgedeeld in verschillende partities. De gebruiker moet de optie hebben om de staven op te delen per werkmaatschappij of per grootboeknummer.

De staaf-partities moeten apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Q1 t/m Q4).

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij of grootboeknummer laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien.”

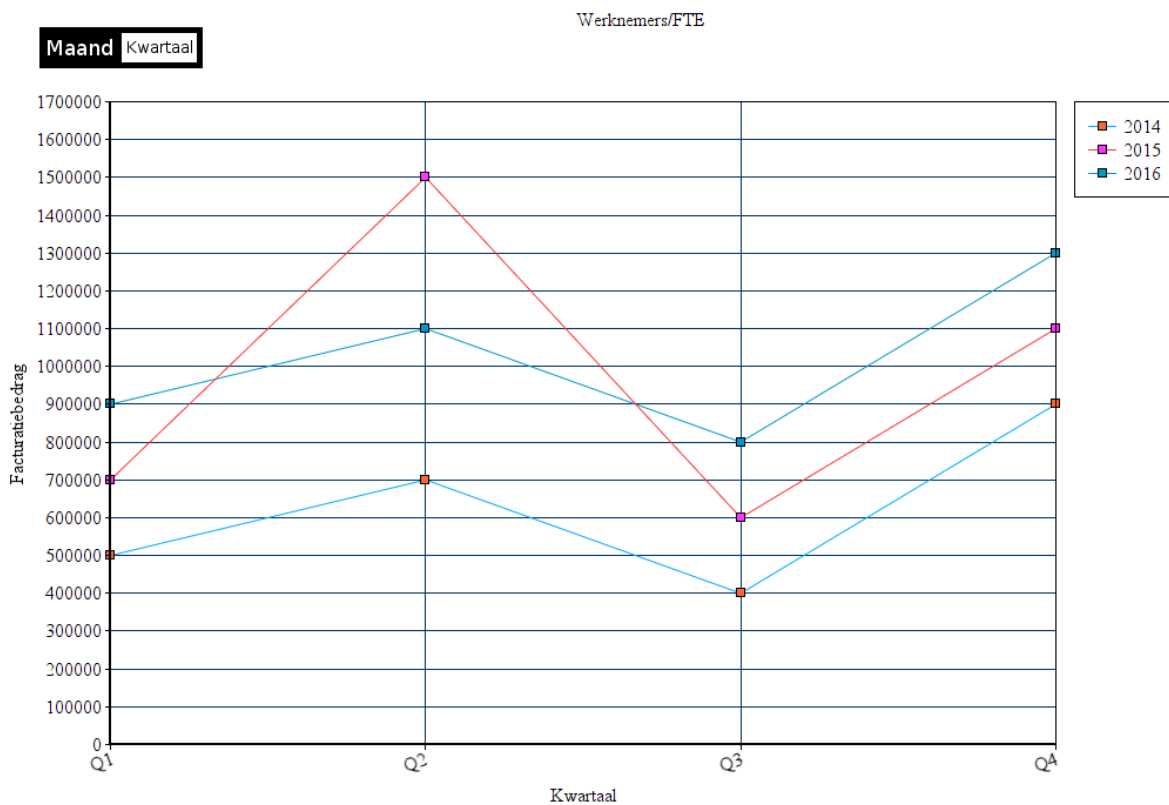


Illustration 5: Requirement 1.7.

“Havik moet in een lijndiagram het totale facturatie bedrag van meerdere jaren kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar.

Elke lijn stelt hierbij het facturatiebedrag van één werkmaatschappij van één jaar voor.

Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (het facturatiebedrag van alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.

Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien. “

Klant	Totale Facturatiebedrag	Projecten	Uren
Klant A	2000000	1	1.000
Klant B	1500000	2	200
Klant C	800000	2	120
Klant D	500000	1	40

Table 1: Requirement 1.8.

Sorteren op... ▲ ▼

- ☐ Werkmaatschappij
- ☐ Werkmaatschappij
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer

Illustration 6: Requirement 1.8.

“Havik moet in een tabel alle klanten kunnen weergeven met het totale facturatiebedrag per klant. Ook de hoeveelheid projecten die bij deze klanten lopen en de hoeveelheid uren die besteed wordt aan een klant moet zichtbaar zijn.

De tabel moet gesorteerd kunnen worden op de drie zichtbare gegevens. (Voorbeeld: de klant met het hoogste facturatie bedrag staat bovenaan, etc.)

Het moet mogelijk zijn om werkmaatschappijen aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezette werkmaatschappij niet meegenomen in de tabel.

Het moet mogelijk zijn om grootboeknummers aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezet grootboeknummer niet meegenomen in de tabel.”

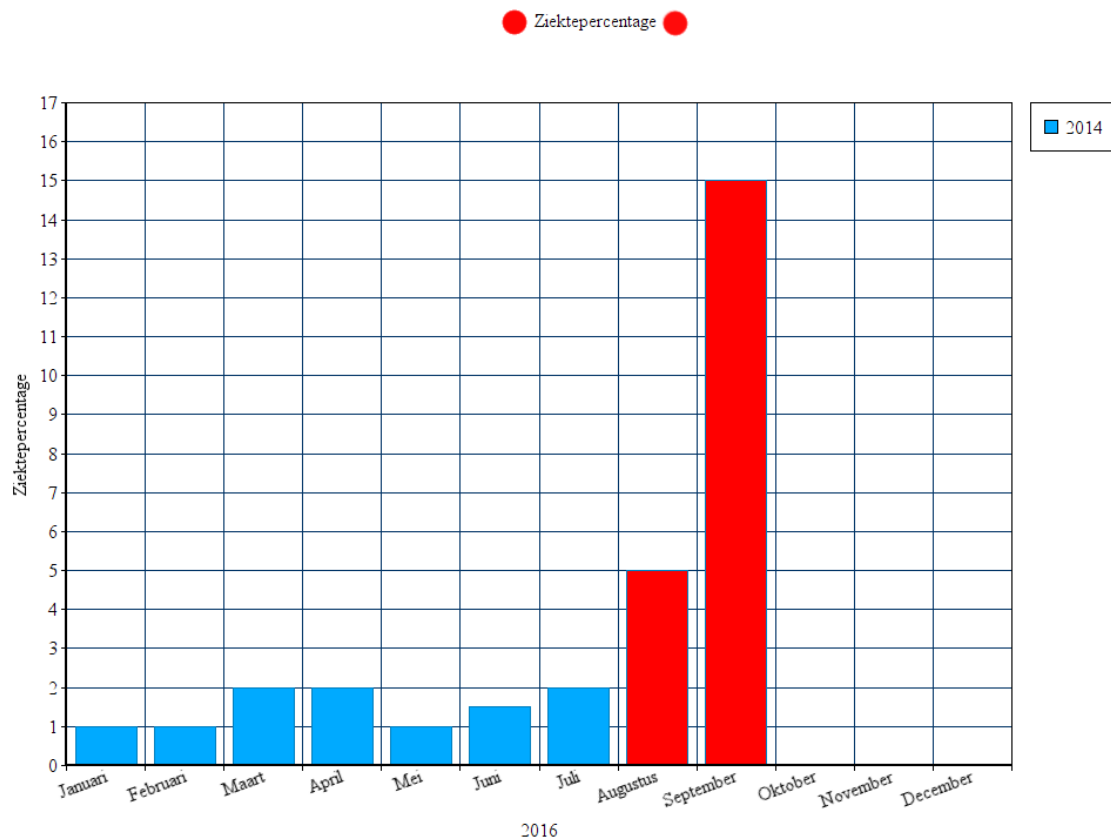


Illustration 7: Voorbeeld van de implementatie van requirement 1.14.

“Widgets kunnen een drempelwaarde hebben waarbij deze zichtbaar een waarschuwing moeten afgeven als de dat hiervan boven of onder de drempelwaarde komt.

Deze drempelwaardes moeten per widget handmatig ingesteld kunnen worden in het instellingenscherf.

Welke widgets er een drempelwaarde kunnen hebben staat nog niet vast.

De drempelwaarde geldt alleen voor de laatste mogelijke periode (oftewel “nu”). De periode kan per widget anders zijn; zie de bijbehorende tabel voor de periodes.”

Requirements 1.3. - 1.8.: Gegevens die geselecteerd worden Detailinformatie

Gegevens met betrekking tot facturaties	De gefactureerde klanten, De facturatie per klant
Medewerkersaantallen	Namen van de medewerkers
Gegevens met betrekking tot ziekte	Namen van de medewerkers die ziek waren

Requirement 1.10.: Gegevens die geselecteerd worden Systeem waar naar gelinked wordt

Gegevens met betrekking tot facturaties	Snip
Medewerkersaantallen	Mus
Gegevens met betrekking tot werktijden	Mus
Gegevens met betrekking tot werknemers	Kauw

Requirement 1.14.: Gegevens Relevant tijdsbestek voor waarschuwingen

Facturatie	Afgelopen twee maanden + huidige maand
------------	--

Requirement 1.16.: Gegevens die handmatig ingevoerd moeten kunnen worden

Omzet medewerkers
Omzet licenties
Omzet service
Omzet derden
Omzet overige
Budget medewerkers
Budget licenties
Budget service
Budget derden
Budget overige

Deze tabellen worden uitgebreid wanneer de bijbehorende requirements worden behandeld in een sprint.

Non-functional requirements

<i>Nummer Requirement</i>	<i>Oorsprong</i>
2.1. Het systeem moet data in de database kunnen opslaan	Case
2.2. De backend moet gemaakt worden in Java.	Case
2.3. De backend moet gemaakt worden met Spring.	Review #1 Architect
2.4. De backend moet gemaakt worden met JPA.	Review #1 Architect
2.5. De frontend moet gemaakt worden in JavaScript.	Case
2.6. De frontend moet gemaakt worden met AngularJS.	Review #1 Architect
2.7. De hoeveelheid data die toegevoegd wordt in de database is irrelevant.	Interview #1 Opdrachtgever
2.8. De REST API moet onder Tomcat draaien.	Interview #1 Architect
2.9. De webclient monder onder een Apache HTTP server draaien.	
2.10. De blokken waar de widgets inkomen moeten allemaal even groot zijn.	Interview #3 Opdrachtgever
2.11. Grafiek-widgets moeten in hun blokken passen.	Review #2 Opdrachtgever
2.12. Tabellen kunnen groten zijn dat de blokken waar deze ingeplaatst worden.	Interview #3 Opdrachtgever
2.12.1. In dit geval moet in het blok zelf gescrolled kunnen worden.	Interview #3 Opdrachtgever
2.13. Wanneer er data herberekend wordt waar in verloop van tijd meerdere berekeningen van toepassing zijn moet de berekening toegepast worden die destijds relevant was. Het kan dus voorkomen dat bij een herberekening van één type data meerdere berekeningen toegepast worden. ^[1]	Interview #3 Opdrachtgever

Hoewel technisch gezien geen requirement heeft de opdrachtgever aangegeven dat het niet uitmaakt hoeveel data er in de database wordt toegevoegd.

^[1]: Doordat er bijvoorbeeld vanaf een bepaald moment geen BTW meer moet worden berekend in een berekening veranderd deze berekening. Alle data van voor dat moment moet dan berekend worden met de oude formule (met BTW) en alle data van na die datum moet met de nieuwe formule berekend worden (zonder BTW). Het systeem hoeft alleen met meerdere berekeningen te kunnen werken; de gebruiker hoeft deze niet zelf toe te kunnen voegen via het dashboard.

Business rules

Nummer Business rule

- | | |
|------|---|
| 3.1. | In verloop van tijd kunnen berekeningen veranderen. Historische data moet dan met de oude berekening berekend worden. |
| 3.2. | De datum van een factuur komt niet altijd overeen met de datum waarin de omzet gemaakt wordt. |
| 3.3. | Medewerkers worden opgedeeld in de volgende groepen: Overhead; Stagiare/Afstuderen. Medewerkers die niet in een groep zijn ingedeeld zijn inzetbare mensen. |
| 3.4. | Klanten kunnen facturen ontvangen van meerdere werkmaatschappijen. |
| 3.5. | Na een bepaalde datum/verlopen tijd wordt bepaalde data niet meer veranderd. (zie bijbehorende tabel) |
| 3.6. | Personeelsnummers worden niet hergebruikt. |

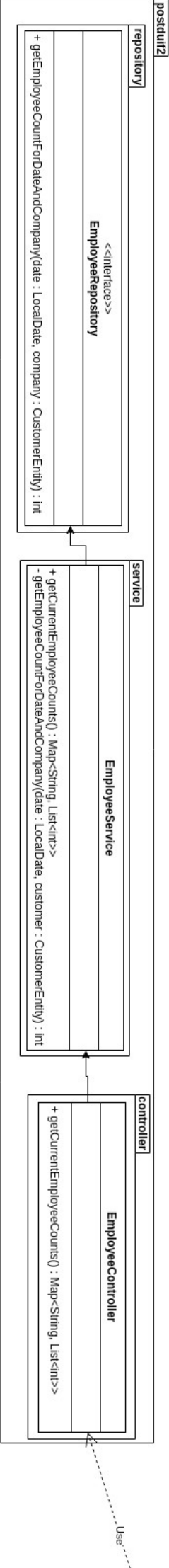
Business Rule 3.5.: Data die niet meer veranderd Tijd/Datum die verstreken moet zijn

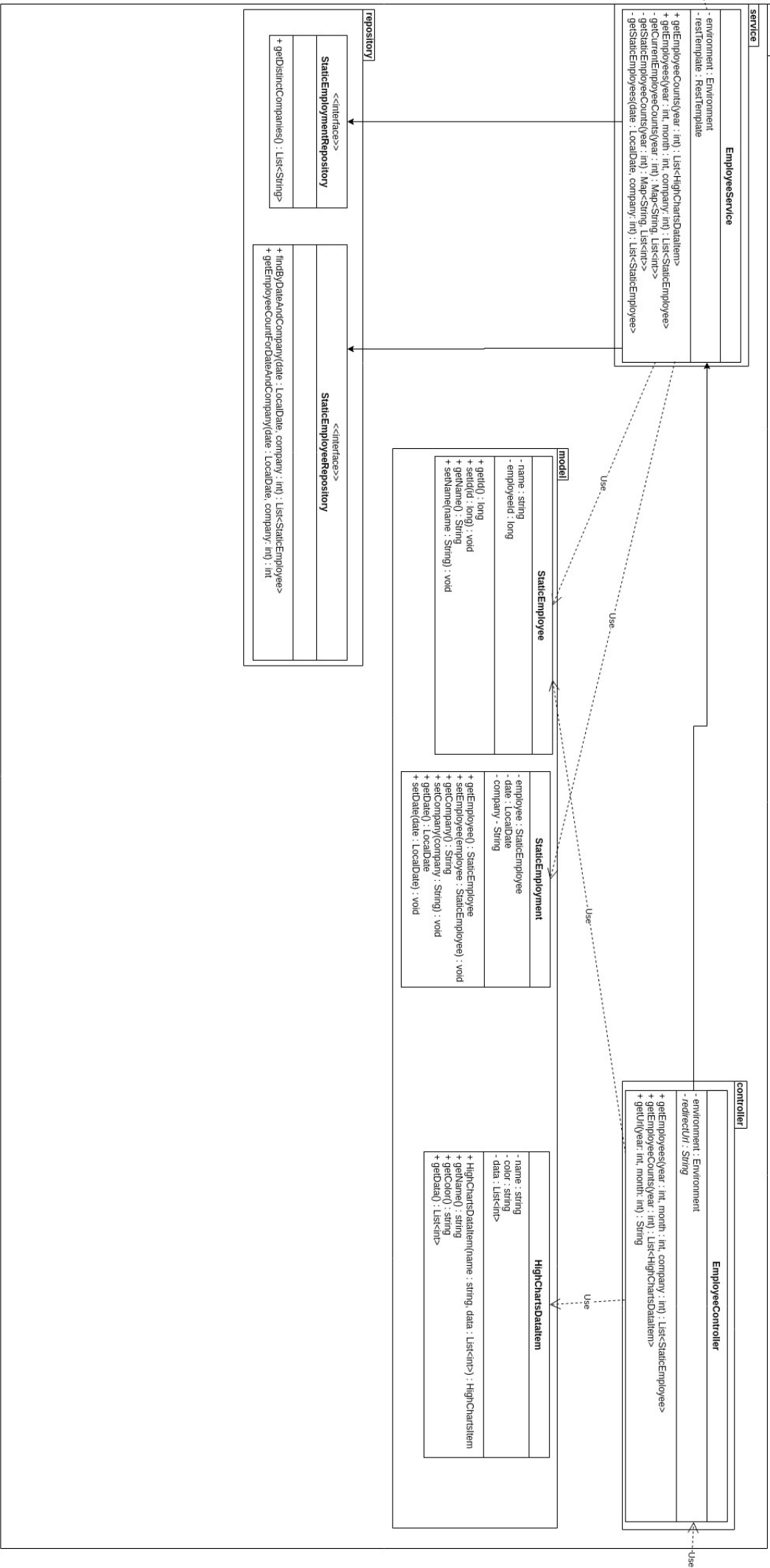
Financiële data	Juni van het volgende jaar
Data met betrekking tot ziekte	Één maand
Data met betrekking tot aantallen medewerkers	Één maand

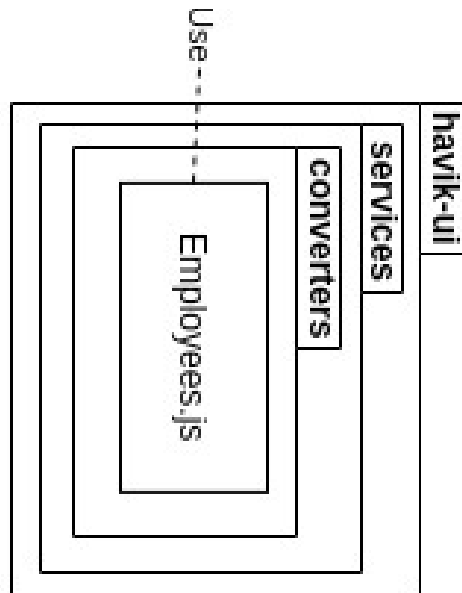


Klassendiagram sprint 3

Bijlage F

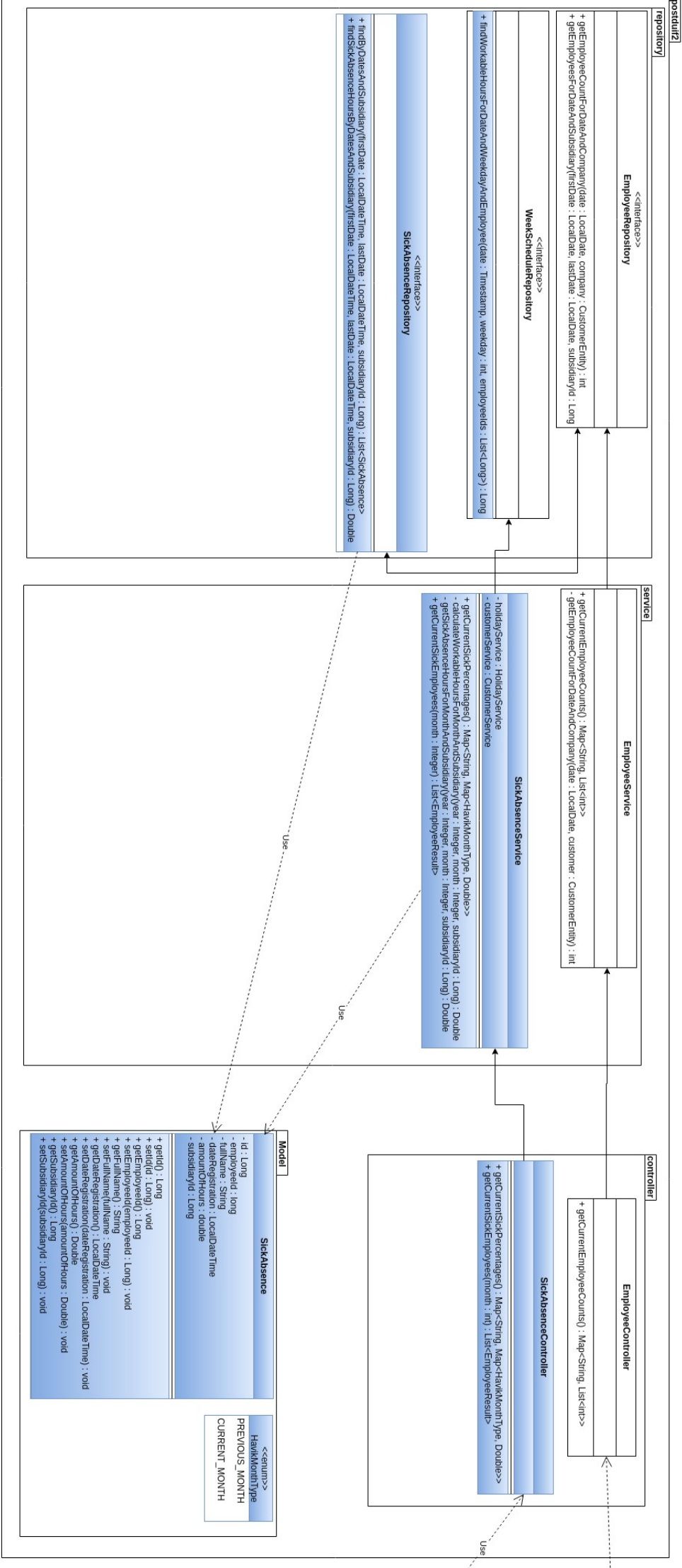




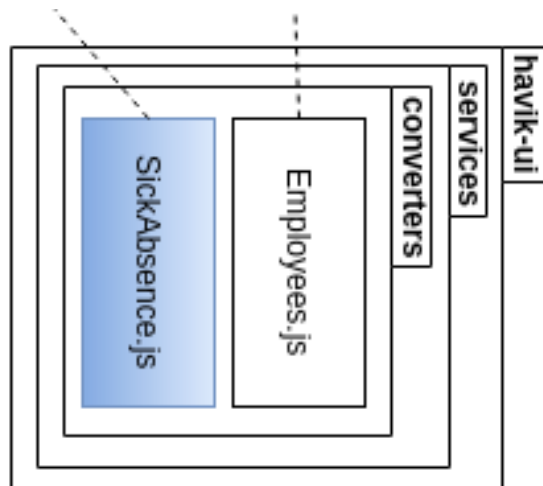


Klassendiagram sprint 4

Bijlage G







Sprint 4 retrospective V1.0

Bijlage H

Eindresultaat Sprint #4

Medewerkers-widget

Deze is nu af. De laatste functionaliteiten zijn toegevoegd en deze is volledig getest.

Requirements

Er zijn een aantal nieuwe requirements bijgekomen. De grootste uitbreiding was voor requirement 1.4: de requirement voor het ziekteverzuim-widget. Feedback van de begeleider geeft aan dat de requirements klaar zijn voor de prioritering; alleen feedback van de opdrachtgever is hier nog voor benodigd.

UML Diagrammen

Beide diagrammen zijn geupdatet. De database van Havik wordt vergroot met een extra tabel die informatie per maand opslaat; deze kan ook de medewerkersaantallen per maand opslaan. Ook het design van de programmatuur is aangepast om de nieuwe widget te kunnen accomoderen. N.a.v. de code review op maandag 11-04-16 is deze niet meer volledig up-to-date.

Ziekteverzuim-widget

Elementen die functioneel zijn in de widget:

- De widget zelf is geïmplementeerd;
- Hardcoded data wordt vanuit Havik-core doorgegeven naar de UI;
- Database data wordt opgehaald uit Postduif en laten zien in de UI;
- Het jaartal is te veranderen.

Functionaliteit die nog toegevoegd moet worden:

- De in-memory database van Havik-core moet uitgebreid worden om ziekteverzuim op te slaan;
- De in-memory database van Havik-core moet gebruikt worden;
 - Beide bovenstaande punten gelden ook voor de PostgreSQL database;
- De medewerkers die ziek waren moeten worden laten zien wanneer er op een staaf geklikt wordt;
 - Als er vervolgens op een medewerker geklikt wordt moet er via een deeplink een nieuwe pagina bij Mus openen;
- De widget moet goed getest worden; een aantal tests zijn al gemaakt, maar het grootste gedeelte moet nog getest worden.

Evaluatie

Requirements

Behalve een klein aantal nieuwe requirements en veranderingen aan de requirement voor het ziekteverzuim-widget is er weinig aan het document veranderd. Dit is ook al een indicatie dat de requirements klaar zijn voor de prioritering. De prioritering is nu wel een activiteit wat op korte termijn moet gebeuren; vanaf ongeveer de eerste sprint is al gesproken over de prioritering terwijl we nu de vijfde sprint ingaan.

Ziekteverzuim-widget

De vorige sprint had in totaal 7 dagen waarin ik kon werken aan Havik (1 dag pasen, 2 dagen afstudeerverslag).

- 1.5 dag hiervan is besteed aan het afmaken van de medewerkers-widget;
- 1 dag hiervan is besteed aan de requirements en het design van het ziekteverzuim-widget;
- 1 dag hiervan is besteed aan het maken van de UI van het ziekteverzuim-widget;
- 2 dagen hiervan zijn besteed aan het maken van de twee grote queries;
- 1.5 dag hiervan is besteed aan het maken van de back-end van het ziekteverzuim widget.

Waarom is de widget niet afgekomen in de sprint/wat is er fout gegaan? Kort gezegd, er was teveel werk geplanned in te weinig tijd.

- Het afmaken van de medewerkers-widget heeft meer tijd gekost dan verwacht;
- De requirements zijn wel volgens planning verlopen
- Het design is later nog veranderd door een beslissing over een extra tabel in de Havik database; dit heeft later in de sprint ook nog tijd gekost (ongepland);
- De UI zelf had 2 dagen ingepland gekregen; deze moet nu nog getest worden. Had er inderdaad 2 dagen besteed kunnen worden aan de UI had dit waarschijnlijk af geweest.
- De twee grote queries zijn twee onverwachte activiteiten die ongepland veel tijd hebben gekost;
- De back-end heeft niet genoeg tijd kunnen krijgen i.v.m. andere activiteiten die teveel tijd in beslag namen. Hier zal nu nog wel een dag of 2-3 aan besteed moeten worden; Havik-core + de Database van Havik moeten aangepast worden voor de nieuwe widget en de tests voor Havik & Postduif moeten gemaakt worden.
- De sprint was een dag korter i.v.m. pasen.

All in all: activiteiten hadden meer nodig dan gepland.

Planning Sprint #5

Het doel van de vijfde sprint is om de widget van de vorige sprint (ziekteverzuim) af te maken. De bijbehorende requirement hierbij is 1.4. Daarnaast wordt er deze sprint tijd besteed aan het deployen naar de testomgeving.

<i>Activiteit</i>	<i>Datum</i>
Code Review Sprint #4	11-4, 12-4
Sprint #4 Retrospective	11-4, 12-4
Updaten Design (o.a. n.a.v. de code review)	12-4
Programmeren & testen ziekteverzuim-widget	12-4, 13-4, 14-4
Deployen naar testomgeving*	18-4, 19-4, 20-4, 21-4
Werken aan afstudeerverslag	15-4, 22-4

*: Dit is inclusief alle benodigdheden die gedaan moeten worden voordat er gedeployed kan worden. Omdat ik nog niet weet wat hiervoor benodigd is kan ik deze planning niet opsplitsen.

Requirements – final

Bijlage I

Definities

Begrip	Afkorting	Definitie
Full-Time Equivalent	FTE	Dienstverband van een medewerker ten opzichte van een volledige werkweek van 40 uur. Een medewerker die 40 uur per week werkt staat gelijk aan 1 FTE; een medewerker die 32 uur per week werkt staat gelijk aan 0.8 FTE.
Functie		Dat wat uiteindelijk culmineert in een widget.
Grootboekrekening		Een herkenningnummer voor boekhouding waar meerdere posten onder gegroepeerd kunnen worden. Iedere factuur is gekoppeld aan een grootboekrekening. Voor de huidige specs kan worden aangenomen dat er een 1-1 relatie is. Strikt genomen is een factuur verbonden met 3 of meer grootboekrekeningen, maar vanuit Snip is er maar 1 per factuur belangrijk (en mogelijk).
Prijs Index Cijfer		“Een prijsindex is een genormaliseerd gemiddelde (meestal een gewogen gemiddelde) van prijzen voor een gegeven klasse van goederen of diensten in een gegeven regio, gedurende een gegeven tijdsinterval. Het is een statistiek die ontworpen is om te vergelijken hoe, als geheel genomen, deze prijzen tussen tijdsperiodes of geografische locaties kunnen verschillen.” - https://nl.wikipedia.org/wiki/Prijsindex
Historische gegevens		Gegevens die niet meer veranderen, meestal omdat er genoeg tijd over heen is gegaan dat deze niet meer hoeven, of zelfs mogen, aangepast te worden.
Verplichte verlofdagen		Werkdagen die een medewerker verplicht vrij krijgt, bijvoorbeeld de dag na hemelvaart.
Werkdagen		De werkbare dagen in een week; kan per medewerker anders zijn.
Widget		Een klein onderdeel van een webpagina die iets nuttigs met informatie opgehaald uit andere locaties doet. Geeft deze informatie vervolgens ook weer.
Ziekteverzuim/ Ziektepercentage		Ziekteverzuim wordt per maand berekend. Het aantal uren ziek is bekend, het geldige weekrooster van de medewerker staat vast. Het ziektepercentage is het percentage ziekuren ten opzichte van het aantal uren dat maximaal gewerkt had kunnen worden in deze periode.
Gewerkte uren: is de optelling van alle uren waarop niet ziek-uren geschreven zijn.		

Ziekte uren: is de optelling van alle uren waarop ziek-uren geschreven zijn.

Maximaal werkbare uren: is de optelling van alle uren in alle dagen waarop contractueel gewerkt zou moeten worden. Dagen waarop contractueel gewerkt zou moeten worden bestaan uit de contractdagen min de feestdagen.

Ziekteverzuim: ziekte uren / maximaal werkbare uren * 100% (individueel per medewerker).

Voorbeeld:

Medewerker A: 160 uren werkbaar; 16 uur ziek; 10% ziek

Medewerker B: 140 uren werkbaar; 14 uur ziek; 10% ziek

Medewerker C: 20 uren werkbaar; 10 uur ziek; 50% ziek

Maandverzuim: $\sum$ maximaal werkbare uren per maand (van alle medewerkers) / $\sum$ ziekte-uren per maand (van alle medewerkers) * 100%

In het voorbeeld: 12.5%

Project Requirements

Functional requirements

#	Benodigd Requirement	Oorsprong	Storypoints	Prioriteit
1.1.	Havik moet grafieken kunnen laten zien.	Afstudeerplan	-	-
1.2.	Havik moet tabellen kunnen laten zien.	Afstudeerplan	-	-
1.3. 1.1.	Havik moet een lijndiagram kunnen weergeven die het ziektepercentage van de afgelopen 10 jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet. Elke lijn stelt het ziektepercentage van één jaar voor bij één werkmaatschappij. Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.	Review #1 Opdrachtgever Interview #3 Opdrachtgever Interview #5 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever	13	-

#	Benodigd Requirement	Oorsprong	Storypoints Prioriteit
1.4. 1.1.	Havik moet een staafdiagram kunnen weergeven die het ziektepercentage van één jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet. Hierbij zijn er 3 staven per maand: • één voor 42; • één voor Prepend; • één voor alle werkmaatschappijen bij elkaar opgeteld. De staven moeten apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December). De selectie moet niet verder terug in de tijd kunnen gaan dan 2003. De selectie mag ook niet verder in de tijd kunnen gaan dan het huidige jaar. Wanneer er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijbehorende tabel welke gegevens er van de medewerkers moet worden laten zien. Wordt er vervolgens op één van de werknemers in dit scherm geklikt moet er een deeplink zijn naar systeem Mus. Bij Mus moet vervolgens automatisch ingelogd worden met dezelfde gegevens die gebruikt zijn om bij Havik in te loggen. Mus moet vervolgens uitkomen op de urenregistratiepagina van de specifieke werknemer waar op geklikt is voor die maand.	Interview #3 Opdrachtgever - (21) Interview #5 Opdrachtgever Interview #3 Opdrachtgever Interview #5 Opdrachtgever *: 04-04-2016 Interview #3 Opdrachtgever Review #1 Opdrachtgever Interview #5 Opdrachtgever Interview #5 Opdrachtgever Interview #3 Opdrachtgever Interview #5 Opdrachtgever Interview #5 Opdrachtgever *: 04-04-2016	-

#	Benodigd Requirement	Oorsprong	Storypoints Prioriteit
1.5. 1.1.	Havik moet in een staafdiagram alle werknemers van een jaar kunnen laten zien. Hierbij wordt de hoeveelheid werknemers uitgezet tegenover de maanden. Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij. De werkmaatschappijen moeten apart aan en uit te zetten zijn. Ook moeten de werknemersgroepen apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December). De selectie moet niet verder terug in de tijd kunnen gaan dan 2003. De selectie mag ook niet verder in te tijd kunnen gaan dan het huidige jaar. Als er geklikt wordt op een staaf, welke historische data weergeeft, moet er een scherm zichtbaar worden die de werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien. Wordt er geklikt op een staaf die betrekking heeft op de huidige periode moet er een deeplink zijn naar systeem Mus. Bij Mus moet vervolgens automatisch ingelogd worden met dezelfde gegevens die gebruikt zijn om bij Havik in te loggen. Mus moet vervolgens uitkomen op de pagina die alle medewerkers van dit moment laat zien. Werknemers die later in de maand zijn aangenomen (dus niet de gehele maand hebben	Interview #1 Opdrachtgever - (13) Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever Review #1 Opdrachtgever Review #1 Opdrachtgever Interview #4 Opdrachtgever Interview #4 Opdrachtgever Interview #3 Opdrachtgever Interview #4 Opdrachtgever Interview #3 Opdrachtgever Interview #4 Opdrachtgever	-

#	Benodigd Requirement	Oorsprong	Storypoints Prioriteit
1.5.1.1.5.	Bij de bovenstaande grafiek moet de gebruiker de optie hebben om in plaats van de hoeveelheid werknemers de hoeveelheid FTE te zien. Alle andere opties moeten ook bij deze grafiek mogelijk zijn.	Interview #1 Opdrachtgever 8 Interview #3 Opdrachtgever	-
1.6. 1.1.	Havik moet in een staafdiagram het totale facturatie bedrag van een jaar kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar. Elke staaf is hierin opgedeeld in verschillende partities. De gebruiker moet de optie hebben om de staven op te delen per werkmaatschappij of per grootboeknummer. De staaf-partities moeten apart aan en uit te zetten zijn. De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Q1 t/m Q4). Als er geklikt wordt op een staaf, welke historische data weergeeft, moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij of grootboeknummer laat zien. Zie de bijbehorende tabel welke gegevens er van de klanten moet worden laten zien. Onduidelijk wat er moet gebeuren bij het klikken op een staaf van de huidige periode.	Interview #1 Opdrachtgever 13 Interview #3 Opdrachtgever Review #1 Opdrachtgever Review #1 Opdrachtgever Interview #3 Opdrachtgever Review #1 Interview #3 Opdrachtgever	-

#	Benodigd Requirement	Oorsprong	Storypoints Prioriteit
1.7. 1.1.	Havik moet in een lijndiagram het totale facturatie bedrag van meerdere jaren kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen van een kalenderjaar. Elke lijn stelt hierbij het facturatiebedrag van één werkmaatschappij van één jaar voor. Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (het facturatiebedrag van alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn. Als er geklikt wordt op een punt, welke historische data weergeeft, in de grafiek moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijbehorende tabel welke gegevens er van de klanten moet worden laten zien. Onduidelijke grafiek; teveel data weergave in 1 grafiek (3 lijnen per jaar). Mogelijk opsplitsen?	Interview #3 Opdrachtgever 13 Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #3 Opdrachtgever	-

#	Benodigd Requirement	Oorsprong	Storypoints	Prioriteit
1.8.	1.2. Havik moet op de locatie van één widget twee tabellen laten zien. In beide tabellen moet de top 10 aan klanten worden laten zien met hun: • klantnaam; • hoeveelheid uren. De eerste tabel laat deze gegevens zien over de afgelopen 12 maanden. De tweede tabel laat deze gegevens zien over de afgelopen 2 maanden. De tabellen moeten gesorteerd worden op de hoeveelheid uren. Het moet mogelijk zijn om werkmaatschappijen aan en uit te zetten. Hierbij worden uren bij een uitgezette werkmaatschappij niet meegenomen in de tabellen. Wanneer er geklikt wordt op de naam van een klant moet er een nieuw scherm openen waarin via een deeplink naar Kauw de gegevens van die klant wordt laten zien. Het moet mogelijk zijn om de huidige bekeken maand (vanaf waar er 2/12 maanden worden teruggeteld) te veranderen. Hiervoor moeten het jaar en de maand apart van elkaar veranderd kunnen worden.	Prioritering Interview #1 Opdrachtgever Interview #3 Opdrachtgever Prioritering Prioritering Interview #1 Opdrachtgever Prioritering Prioritering Prioritering	8	1
1.9.	Havik moet in een tabel de prijsindexcijfers van verschillende statistieken kunnen laten zien. Mogelijkheden zijn bijvoorbeeld de prijsindexcijfer van de kosten, tarieven, bruto salaris, totale kosten, Deze requirement wordt later verder uitgewerkt; voor nu is deze niet relevant.	Interview #4 Opdrachtgever	13	-

#	Benodigd Requirement	Oorsprong	Storypoints	Prioriteit
1.10.	Er moet een instellingen scherm zijn waar alleen de admin bij mag. Vanaf het instellingenscherf kan er doorgeklikt worden naar de verschillende instellingsmogelijkheden.	Interview #3 Opdrachtgever Interview #4 Opdrachtgever	3	
1.11. 1.10.	In het instellingen scherm moet er ingesteld kunnen worden wat de eerste acht widgets op het dashboard zijn. Deze instelling geldt vervolgens voor alle gebruikers; de gebruikers kunnen deze niet apart aanpassen. Alle gebruikers hebben dus dezelfde widgets op de eerste acht plekken staan.	Interview #3 Opdrachtgever Interview #4 Opdrachtgever	8	-
1.11. 1	Widgets na de eerste acht moeten op alfabetische volgorde staan. Widgets die handmatig ingesteld staan op de eerste 8 plekken moeten hier niet in terugkomen; er staan geen dubbele widgets in.	Interview #3 Opdrachtgever Interview #3 Opdrachtgever	3	-

#	Benodigd Requirement	Oorsprong	Storypoints Prioriteit
1.12. 1.10.	Widgets kunnen een drempelwaarde hebben waarbij deze zichtbaar een waarschuwing moeten afgeven als de dat hiervan boven of ander de drempelwaarde komt. Deze drempelwaardes moeten per widget handmatig ingesteld kunnen worden in het instellingenscherf. Welke widgets er een drempelwaarde kunnen hebben staat nog niet vast. De drempelwaarde geldt alleen voor de laatst mogelijke periode (oftewel “nu”). De periode kan per widget anders zijn; zie de bijbehorende tabel voor de periodes. De ingestelde drempelwaardes zijn voor alle gebruikers gelijk; de admin bepaald deze, de gebruikers zelf kunnen deze niet veranderen. Alle gebruikers moeten dus dezelfde waarschuwingen zien. Hoe deze waarschuwing er vervolgens uit ziet is nog niet bepaald. Voorbeelden zijn wel benoemd: Het relevante element (bijvoorbeeld de staaf in een staafgrafiek) is rood gekleurd; De x- & y-assen zijn rood gekleurd; De tekst in de widget is rood gekleurt. Het besluit hoe deze eruit komt te zien moet nog genomen worden; ondertussen kunnen er voorbeelden bijkomen/afgaan.	Interview #3 Opdrachtgever 21 Interview #3 Opdrachtgever Interview #3 Opdrachtgever Interview #4 Opdrachtgever Interview #5 Opdrachtgever	-

#	Benodigd Requirement	Oorsprong	Storypoints	Prioriteit
1.13.	Er moet een functie zijn die de data in de database herberekent. De admin moet deze zelf kunnen activeren. Hierbij dient rekening gehouden te worden met data die toegevoegd, veranderd of verwijderd is na de vorige (her)berekening.	Interview #3 Opdrachtgever Interview #5 Opdrachtgever	34	2
1.14. 1.10.	In het instellingenschermb moet het mogelijk zijn om handmatig data in te voeren voor berekeningen. Wanneer er een herberekening plaats vindt moet de gebruiker de optie krijgen om de handmatig ingevoerde data te gebruiken voor de berekening of de berekening uit te voeren met de data die in de database staat. Zie de bijbehorende tabel voor de gegevens die handmatig ingevoerd moeten kunnen worden.	Interview #3 Opdrachtgever	21	4
1.15.	Werkmaatschappijen moeten in verschillende grafieken dezelfde kleur krijgen. "42 BV" moet in alle grafieken gerepresenteerd worden door de kleur blauw. "Prepend" moet in alle grafieken gerepresenteerd worden door de kleur groen.	Interview #5 Opdrachtgever Interview #5 Opdrachtgever Interview #5 Opdrachtgever	-	-
1.16.	Alle widgets moeten gelijk getrokken worden qua style, zodat het hele dashboard volgens dezelfde style is opgebouwd. Voorbeeld: Alle widgets moeten conform requirement 1.15. werken. Dit betreft ook de widgets die al bestonden voor dit project begon; heel Havik moet hier dus op gecontroleerd worden.	Interview #5 Opdrachtgever	13	-

: Deze requirements zijn aangepast naar aanleiding van een kort gesprek (maximaal 5 minuten) met de opdrachtgever. De datum achter de "" geeft aan wanneer dit gesprek heeft plaatsgevonden. Deze korte gesprekken zijn niet opgenomen en er zijn geen notulen van. (open door policy)

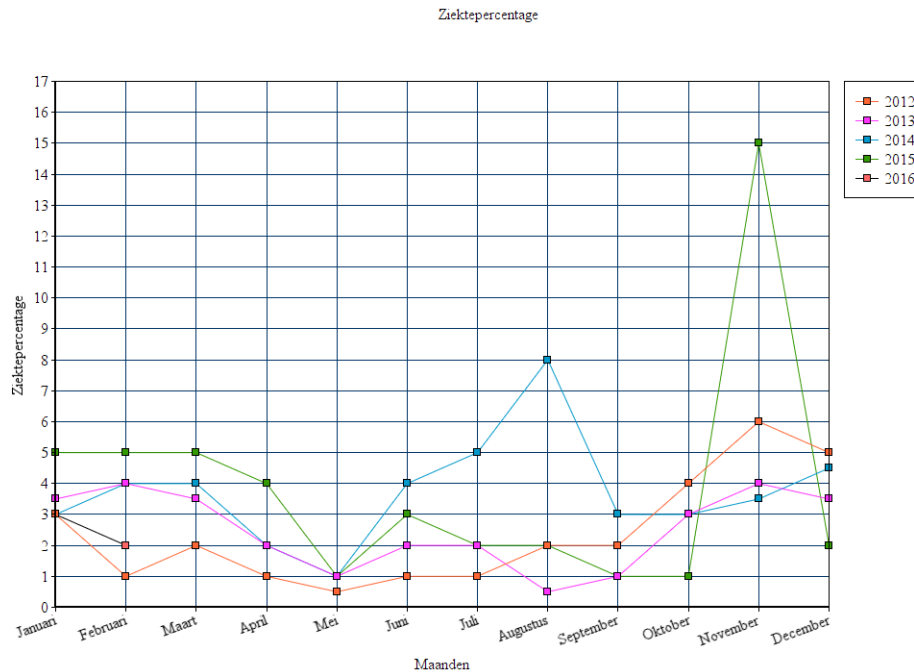


Illustration 8: Requirement 1.3.

“Havik moet een lijndiagram kunnen weergeven die het ziektepercentage van de afgelopen 10 jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet.

Elke lijn stelt het ziektepercentage van één jaar voor bij één werkmaatschappij.

Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.

Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.”

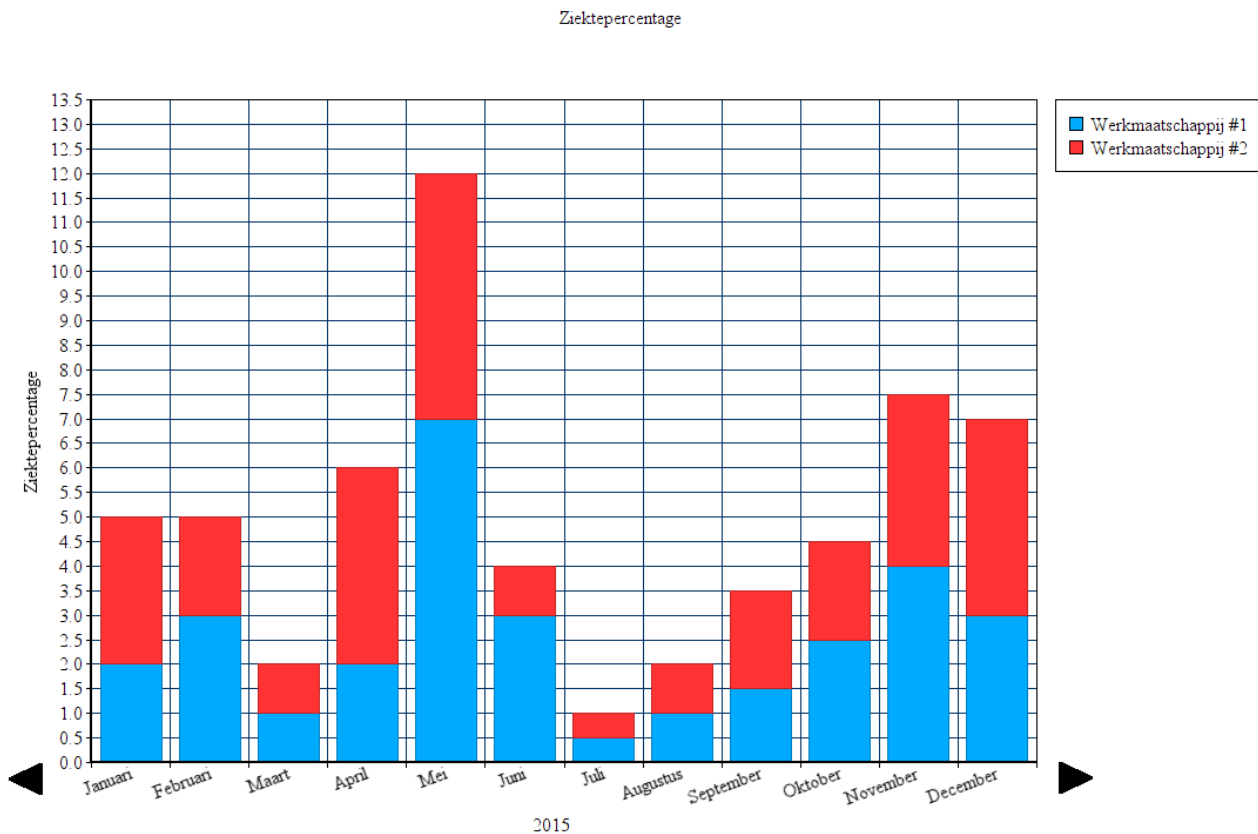


Illustration 9: Requirement 1.4.

“Havik moet een staafdiagram kunnen weergeven die het ziektepercentage van één jaar laat zien. Hierbij wordt het percentage tegenover de maanden uitgezet.

Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij.

De werkmaatschappijen moeten apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December).

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de zieke werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.”

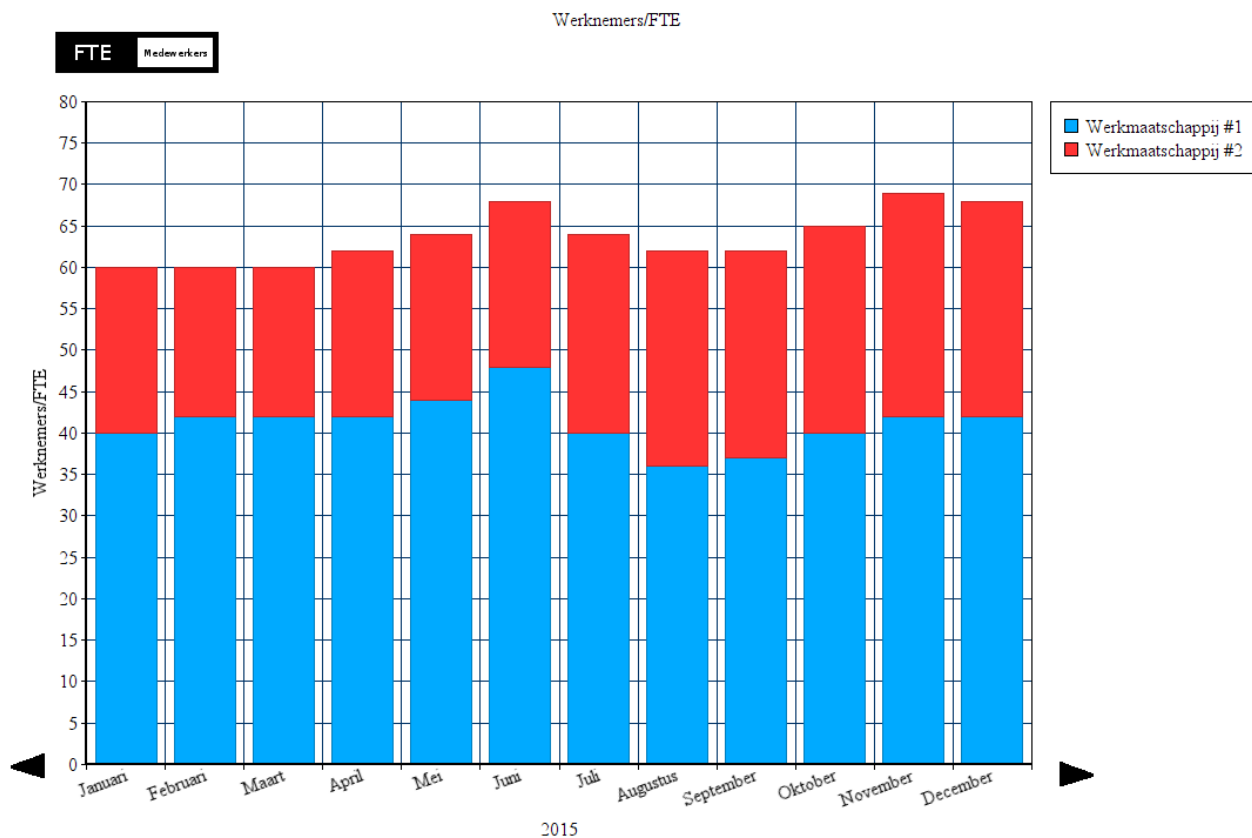


Illustration 10: Requirement 1.5.

“Havik moet in een staafdiagram alle werknemers van een jaar kunnen laten zien. Hierbij wordt de hoeveelheid werknemers uitgezet tegenover de maanden.

Elke staaf is hierin opgedeeld in verschillende partities: één per werkmaatschappij.

De werkmaatschappijen moeten apart aan en uit te zetten zijn. Ook moeten de werknemersgroepen apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Januari t/m December).

De selectie moet niet verder terug in de tijd kunnen gaan dan 2003.

De selectie mag ook niet verder in de tijd kunnen gaan dan het huidige jaar.

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de werknemers van die maand van de bijbehorende werkmaatschappij laat zien. Zie de bijborende tabel welke gegevens er van de medewerkers moet worden laten zien.

Werknemers die later in de maand zijn aangenomen (dus niet de gehele maand hebben gewerkt) tellen mee voor de hele maand; deze hoeven niet speciaal aangegeven worden.”

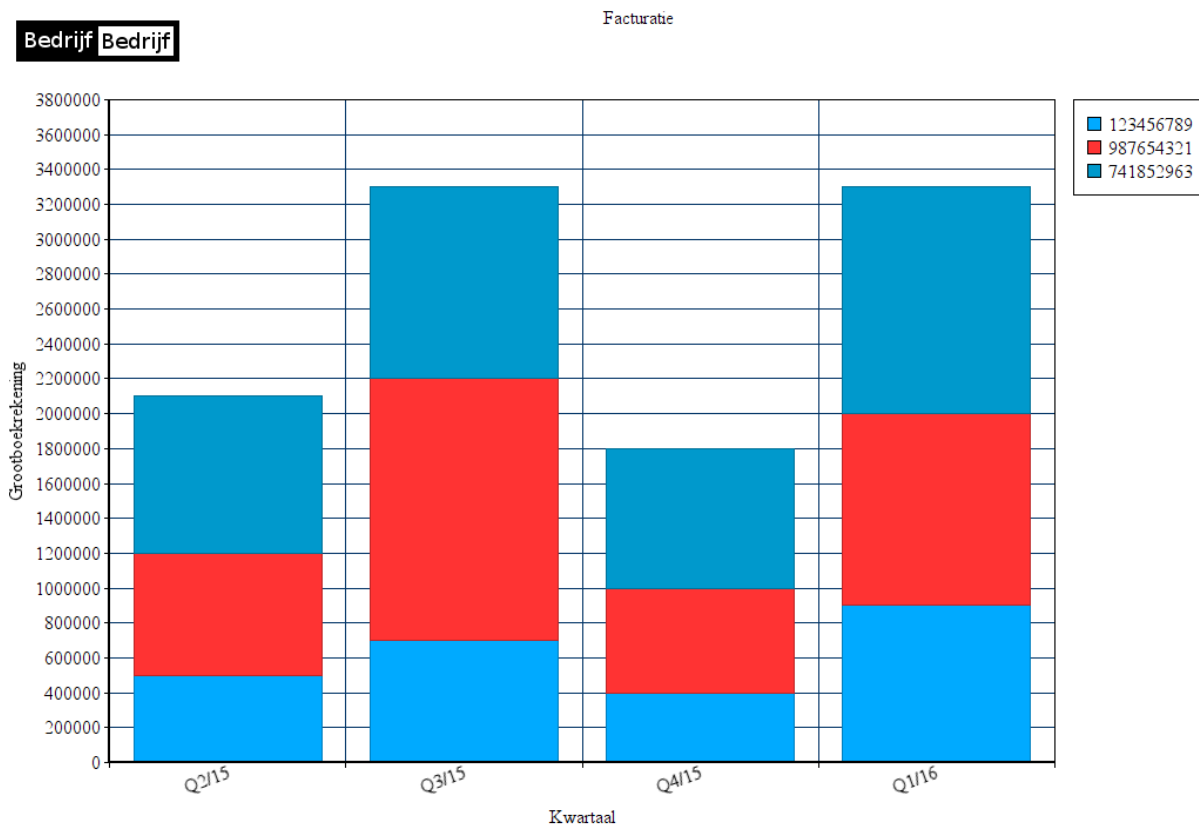


Illustration 11: Requirement 1.6.

“Havik moet in een staafdiagram het totale facturatie bedrag per jaar kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar.

Elke staaf is hierin opgedeeld in verschillende partities. De gebruiker moet de optie hebben om de staven op te delen per werkmaatschappij of per grootboeknummer.

De staaf-partities moeten apart aan en uit te zetten zijn.

De selectie moet voor-/achteruit in de tijd kunnen scrollen met sprongen van één jaar (waardoor er op het scherm altijd een volledig jaar zichtbaar is, van Q1 t/m Q4).

Als er geklikt wordt op een staaf moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij of grootboeknummer laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien.”

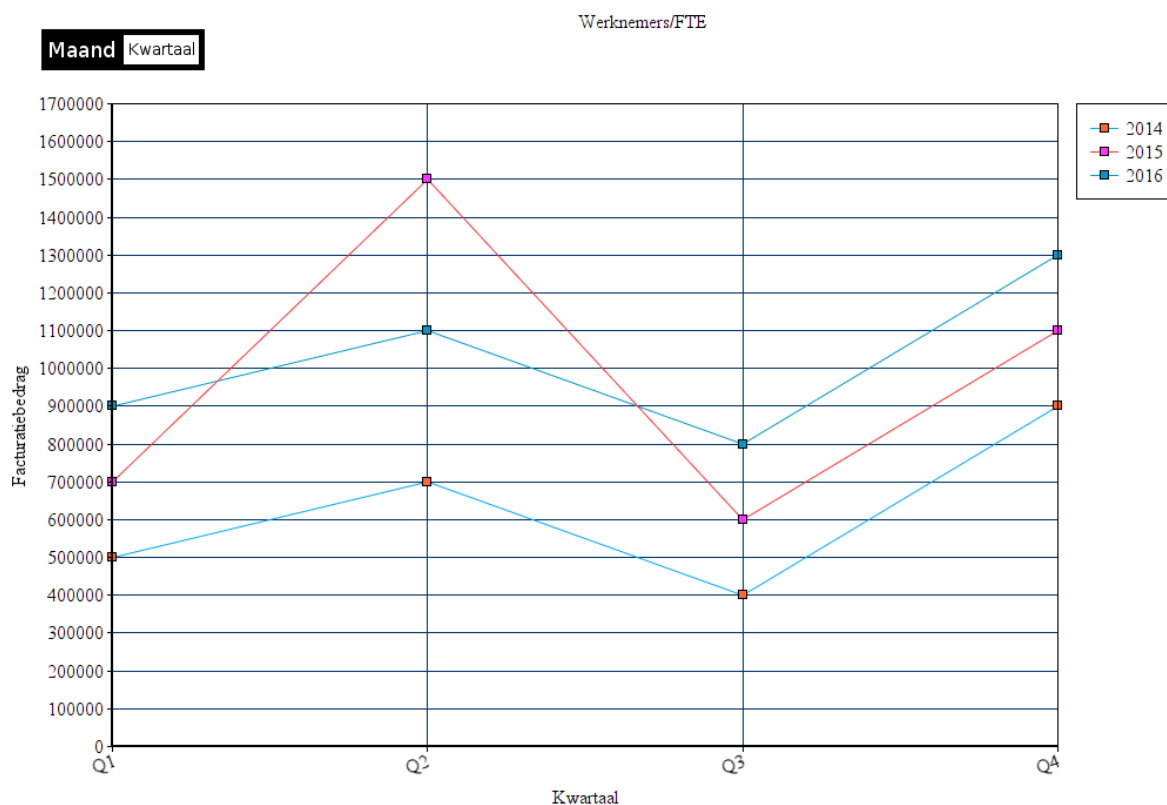


Illustration 12: Requirement 1.7.

“Havik moet in een lijndiagram het totale facturatie bedrag van meerdere jaren kunnen laten zien. Hierbij wordt het bedrag uitgezet tegenover de vier kwartalen in een jaar.

Elke lijn stelt hierbij het facturatiebedrag van één werkmaatschappij van één jaar voor.

Lijnen moeten apart aan en uit te zetten zijn. Ook het totaal (het facturatiebedrag van alle werkmaatschappijen bij elkaar opgeteld) moet zichtbaar kunnen zijn en aan en uit te zetten zijn.

Als er geklikt wordt op een punt in de grafiek moet er een scherm zichtbaar worden die de klanten van die maand van de bijbehorende werkmaatschappij(en) laat zien. Zie de bijborende tabel welke gegevens er van de klanten moet worden laten zien. “

Klant	Totale Facturatiebedrag	Projecten	Uren
Klant A	2000000	1	1.000
Klant B	1500000	2	200
Klant C	800000	2	120
Klant D	500000	1	40

Table 2: Requirement 1.8.

Sorteren op... ▲ ▼

- ☐ Werkmaatschappij
- ☐ Werkmaatschappij
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer
- ☐ Grootboeknummer

Illustration 13: Requirement 1.8.

“Havik moet in een tabel alle klanten kunnen weergeven met het totale facturatiebedrag per klant. Ook de hoeveelheid projecten die bij deze klanten lopen en de hoeveelheid uren die besteed wordt aan een klant moet zichtbaar zijn.

De tabel moet gesorteerd kunnen worden op de drie zichtbare gegevens. (Voorbeeld: de klant met het hoogste facturatie bedrag staat bovenaan, etc.)

Het moet mogelijk zijn om werkmaatschappijen aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezette werkmaatschappij niet meegenomen in de tabel.

Het moet mogelijk zijn om grootboeknummers aan en uit te zetten. Hierbij worden facturaties, projecten & uren bij een uitgezet grootboeknummer niet meegenomen in de tabel.”

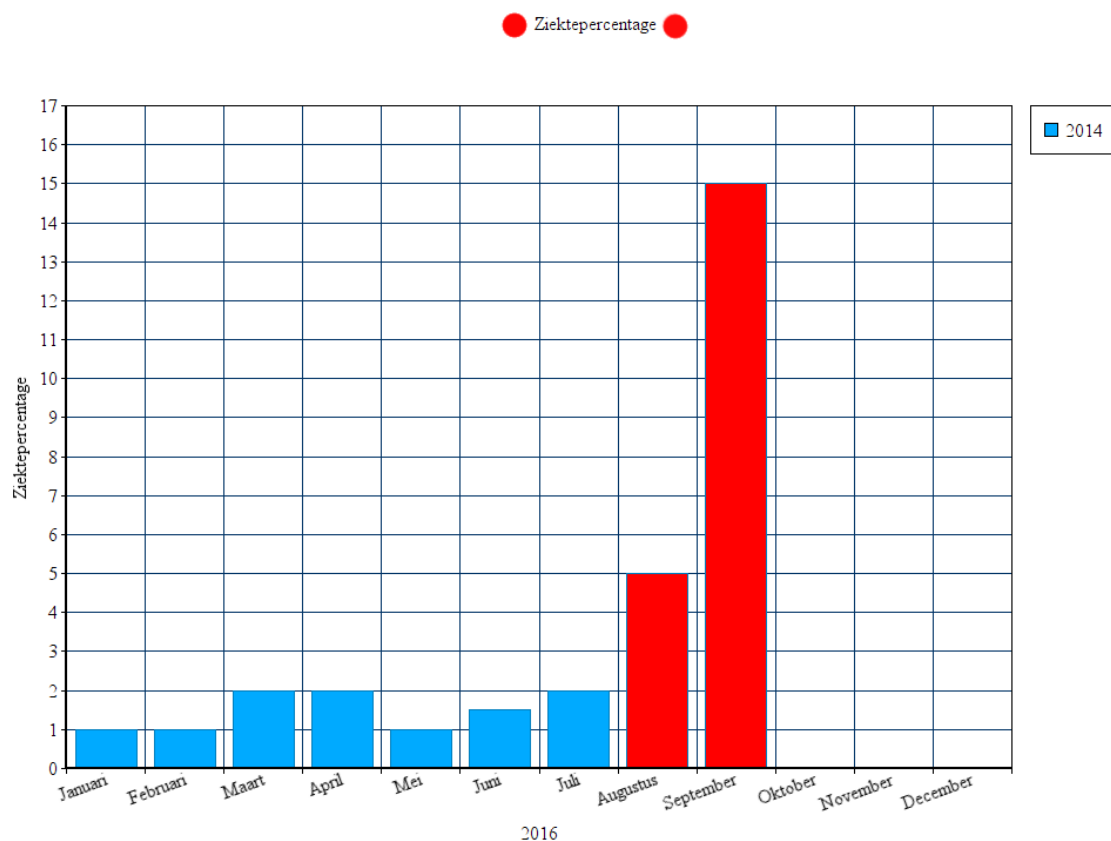


Illustration 14: Voorbeeld van de implementatie van requirement 1.12.

“Widgets kunnen een drempelwaarde hebben waarbij deze zichtbaar een waarschuwing moeten afgeven als de dat hiervan boven of onder de drempelwaarde komt.

Deze drempelwaardes moeten per widget handmatig ingesteld kunnen worden in het instellingenscherf.

Welke widgets er een drempelwaarde kunnen hebben staat nog niet vast.

De drempelwaarde geldt alleen voor de laatste mogelijke periode (oftewel “nu”). De periode kan per widget anders zijn; zie de bijbehorende tabel voor de periodes.”

Requirements 1.3. - 1.8.: Gegevens die geselecteerd worden Detailinformatie

Gegevens met betrekking tot facturaties	De gefactureerde klanten; De facturatie per klant
Medewerkersaantallen	Namen van de medewerkers
Gegevens met betrekking tot ziekte	Namen van de medewerkers die ziek waren; Per medewerker het aantal uren dat deze ziek was

Requirement 1.10.: Gegevens die geselecteerd worden Systeem waar naar gelinked wordt

Gegevens met betrekking tot facturaties	Snip
Medewerkersaantallen	Mus
Ziekteverzuim	Mus
Gegevens met betrekking tot werktijden	Mus
Gegevens met betrekking tot werknemers	Kauw

Requirement 1.12.: Gegevens Relevant tijdsbestek voor waarschuwingen

Facturatie	Afgelopen twee maanden + huidige maand
Ziekteverzuim	Huidige maand

Requirement 1.14.: Gegevens die handmatig ingevoerd moeten kunnen worden

Omzet medewerkers
Omzet licenties
Omzet service
Omzet derden
Omzet overige
Budget medewerkers
Budget licenties
Budget service
Budget derden
Budget overige

Deze tabellen worden uitgebreid wanneer de bijbehorende requirements worden behandeld in een sprint.

Non-functional requirements

<i>Nummer Requirement</i>	<i>Oorsprong</i>
2.1. Het systeem moet data in de database kunnen opslaan	Case
2.2. De backend moet gemaakt worden in Java.	Case
2.3. De backend moet gemaakt worden met Spring.	Review #1 Architect
2.4. De backend moet gemaakt worden met JPA.	Review #1 Architect
2.5. De frontend moet gemaakt worden in JavaScript.	Case
2.6. De frontend moet gemaakt worden met AngularJS.	Review #1 Architect
2.7. De hoeveelheid data die toegevoegd wordt in de database is irrelevant.	Interview #1 Opdrachtgever
2.8. De REST API moet onder Tomcat draaien.	Interview #1 Architect
2.9. De webclient monder onder een Apache HTTP server draaien.	
2.10. De blokken waar de widgets inkomen moeten allemaal even groot zijn.	Interview #3 Opdrachtgever
2.11. Grafiek-widgets moeten in hun blokken passen.	Review #2 Opdrachtgever
2.12. Tabellen kunnen groten zijn dat de blokken waar deze ingeplaatst worden.	Interview #3 Opdrachtgever
2.12.1. In dit geval moet in het blok zelf gescrolled kunnen worden.	Interview #3 Opdrachtgever
2.13. Wanneer er data herberekend wordt waar in verloop van tijd meerdere berekeningen van toepassing zijn moet de berekening toegepast worden die destijds relevant was. Het kan dus voorkomen dat bij een herberekening van één type data meerdere berekeningen toegepast worden. ^[1]	Interview #3 Opdrachtgever

Hoewel technisch gezien geen requirement heeft de opdrachtgever aangegeven dat het niet uitmaakt hoeveel data er in de database wordt toegevoegd.

^[1]: Doordat er bijvoorbeeld vanaf een bepaald moment geen BTW meer moet worden berekend in een berekening veranderd deze berekening. Alle data van voor dat moment moet dan berekend worden met de oude formule (met BTW) en alle data van na die datum moet met de nieuwe formule berekend worden (zonder BTW). Het systeem hoeft alleen met meerdere berekeningen te kunnen werken; de gebruiker hoeft deze niet zelf toe te kunnen voegen via het dashboard.

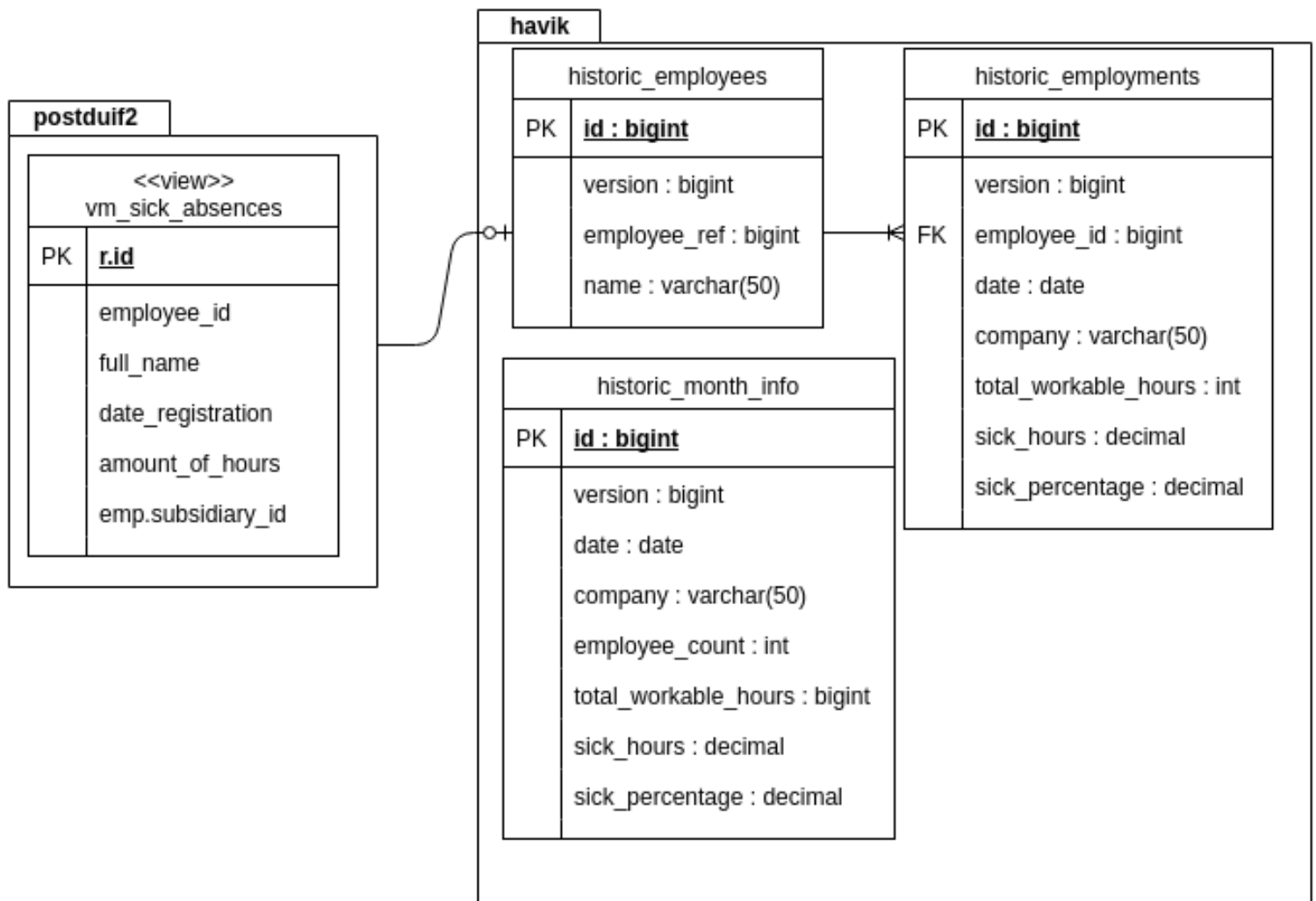
Business rules

<i>Nummer Business rule</i>	
3.1.	In verloop van tijd kunnen berekeningen veranderen. Historische data moet dan met de oude berekening berekend worden.
3.2.	De datum van een factuur komt niet altijd overeen met de datum waarin de omzet gemaakt wordt.
3.3.	Medewerkers worden opgedeeld in de volgende groepen: Overhead; Stagiare/Afstuderen. Medewerkers die niet in een groep zijn ingedeeld zijn inzetbare mensen.
3.4.	Klanten kunnen facturen ontvangen van meerdere werkmaatschappijen.
3.5.	Na een bepaalde datum/verlopen tijd wordt bepaalde data niet meer veranderd. (zie bijbehorende tabel)
3.6.	Personeelsnummers worden niet hergebruikt.

<i>Business Rule 3.5.: Data die niet meer veranderd Tijd/Datum die verstreken moet zijn</i>	
Financiële data	Juni van het volgende jaar
Data met betrekking tot ziekte	Één maand
Data met betrekking tot aantallen medewerkers	Één maand

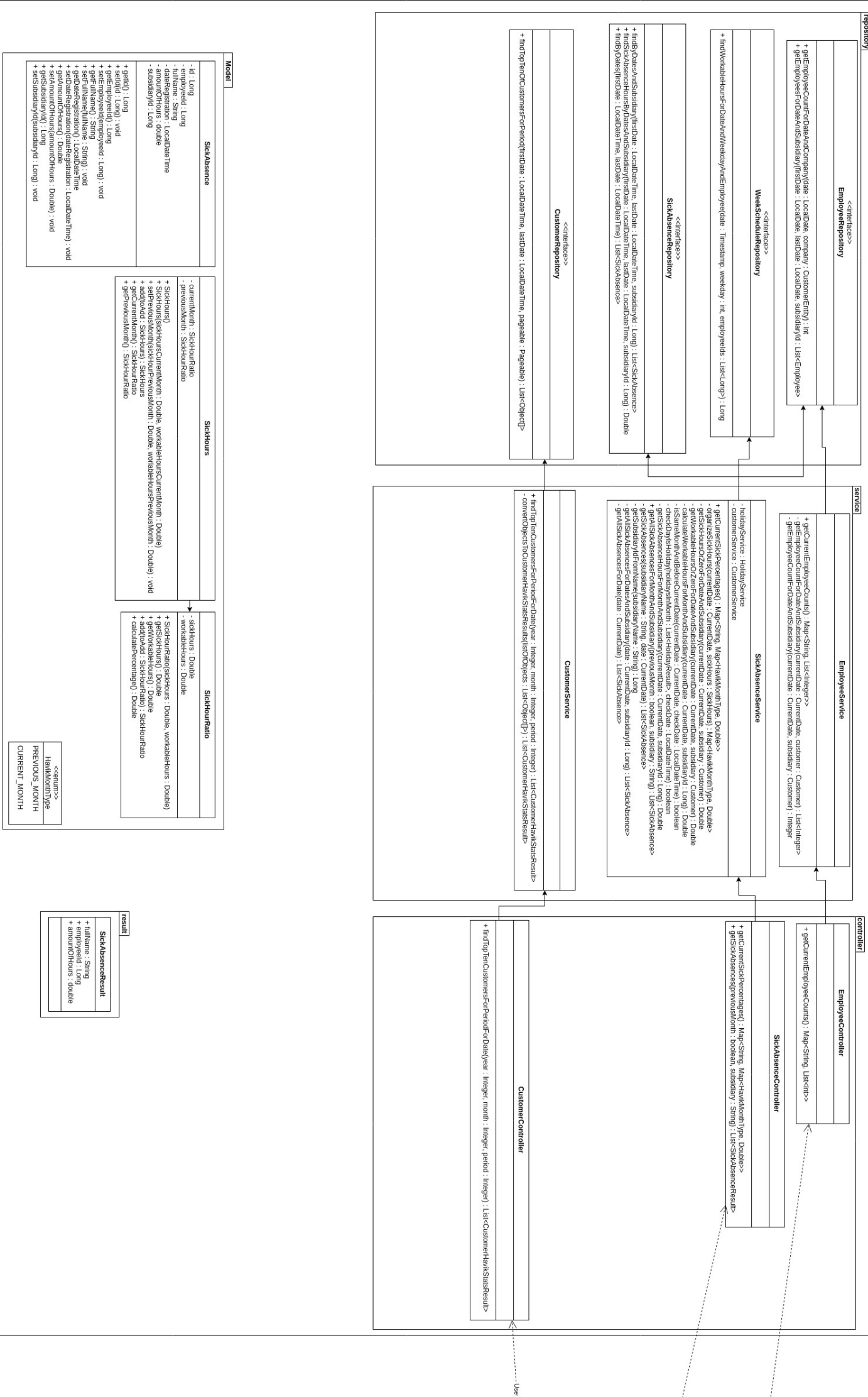
Klassendiagram database – final

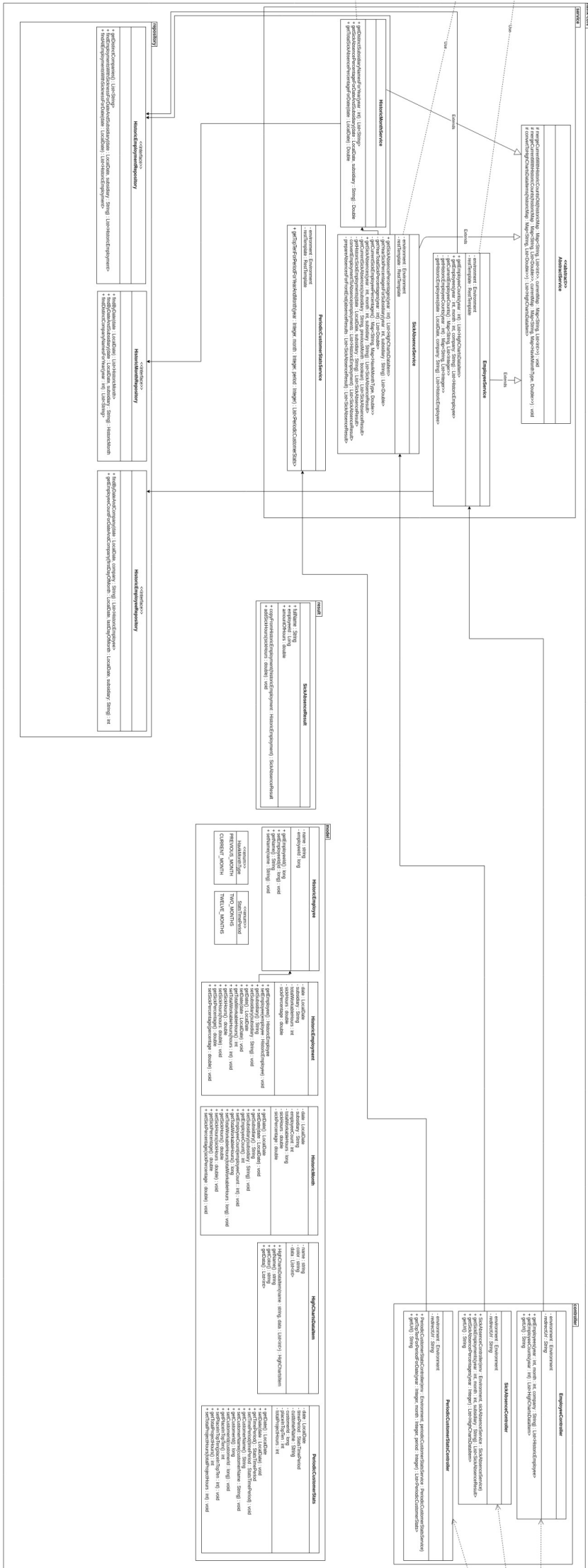
Bijlage J

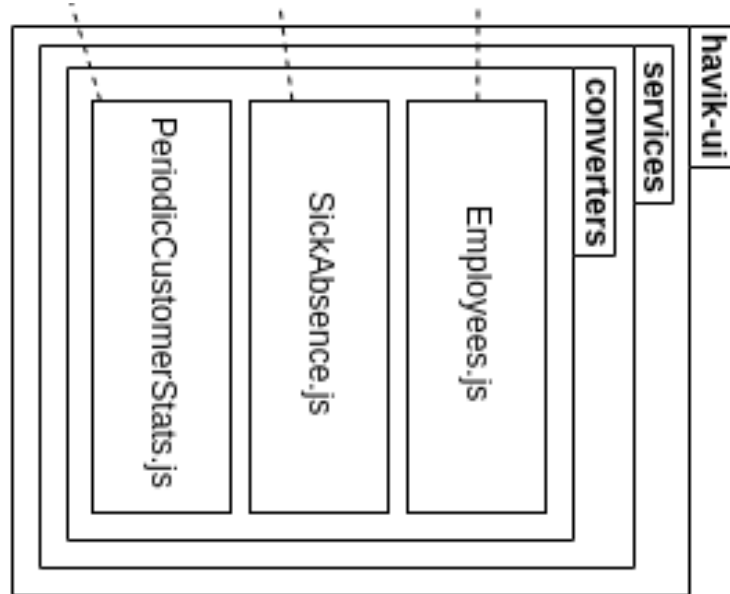


Klassendiagram systeem – final

Bijlage K







Evaluatie bedrijfsbegeleider

Bijlage L



Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: Jordi Romeijn

Periode: Februari 2016 - Juni 2016

Bedrijf c.q. instelling: 42 BV

Bedrijfsmentor: Robert Bor

Plaats: Zoetermeer

Datum: 27 mei 2016

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

Jordi komt uit een C# wereld. Al bij zijn intake bij 42 bleek dat hij geen enkele moeite had met de Java opdrachten (5 van de 5 goed). Die trend wist hij prima vast te houden.

Inwerken in de uitgebreide technologie stack van 42, alsmede het functionele domein ging hem gemakkelijk af.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

Zeer duidelijk in communicatie, vooral als het feitelijkheid betreft. Schriftelijk was dit in eerste instantie minder scherp dan verbaal. Gedurende de opdracht is dat bijgetrokken en werden de requirements verlost van ambiguïteit.

3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

- | | |
|---|------|
| • Qua verantwoordelijkheid | goed |
| • Qua zelfstandigheid | goed |
| • Qua planmatig werken | goed |
| • Qua creativiteit | goed |
| • Qua productiviteit | goed |
| • Qua samenwerken met collega's | goed |
| • Qua draagvlakontwikkeling | goed |
| • Qua inspelen op bedrijfscultuur | goed |
| • Qua rekening houden met de specifieke context van het bedrijf | goed |
| • Qua het op gang brengen van de nodige veranderingen | goed |

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

Zeer goed. Jordi voldoet ruim aan het niveau van wat wij van een professional op dit punt in de carrière verwachten.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

Prima. De opdrachtgever is tevreden met het geleverde resultaat.

6. Bent u tevreden over het opgeleverde (eind)product?

- **In hoeverre heeft u gekregen wat is afgesproken?**

Het product is conform afspraak opgeleverd.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Het product voldoet aan de eindverwachtingen.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

Het product is doorgezet naar de acceptatie-omgeving. De verwachting is dat het binnen afzienbare tijd doorgezet wordt naar productie.

- **Wat gebeurt er met het opgeleverde (eind)product?**

Dat gaat door het MT gebruikt worden om inzicht te krijgen in operationele kencijfers.

- **Kunt u direct met het opgeleverde product aan de slag?**

Ja.

7. **Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?**

8. **Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?**

Zeker.