

GRAFISCHE GUI MODELER

Scriptie



Fontys

Hogeschool ICT



THINKWISE
Software

10-06-2014

Stertefeld, M.M.

Microsoft
GOLD CERTIFIED
Partner

IBM
Business
Partner

ORACLE Silver
Partner

MySQL
Enterprise
READY PARTNER

Thinkwise Software Factory B.V.
Boogschutterstraat 7b, Apeldoorn

Colofon

Student

Naam	Stertefeld, M.M.
Studentnummer	2168041
Afstudeerrichting	Software Engineering - Voltijd
Afstudeerperiode	3-2-2014 – 1-7-2014

Bedrijf

Naam	Thinkwise Software Factory B.V.
Afdeling	Product Innovation
Adres	Boogschutterstraat 7b, 7324 AE
Plaats	Apeldoorn
Telefoon	+31 (0)55 312 82 80
Email	info@thinkwisesoftware.com
Web	http://www.thinkwisesoftware.com


Begeleiders

Fontys	Dhr. Dr. Lamers, M.J.M.
Bedrijf	Dhr. Winkels, R. – Product Innovation Specialist
Bedrijf	Dhr. Brink, D. – Product Innovation Specialist

Uitgifte

Datum	10-06-2014
Locatie	Apeldoorn

Ondertekening



Dhr. Winkels, R.
Product Innovation Specialist
10-06-2014

Voorwoord

Mijn naam is Mathijs Stertefeld. Ik ben een student aan de Fontys Hogeschool te Eindhoven. Hier studeer ik HBO ICT, waar ik momenteel in het vierde en laatste jaar van zit. De major van mijn studie is Software Engineering. Voor de opleiding heb ik een minor uitgevoerd aan een externe instelling, namelijk Advanced Programming aan de Hogeschool van Arnhem en Nijmegen te Arnhem.

Om de studie af te ronden voeren de studenten in het laatste jaar een afstudeerstage uit. Omdat mijn major Software Engineering is, was ik voornamelijk op zoek naar een bedrijf waar veel mogelijkheid is om software te ontwikkelen. Daarom had ik de voorkeur om bij een softwareontwikkelingsbedrijf te werken, in plaats van bij een ICT afdeling van een groter bedrijf. Na bij enkele bedrijven te hebben gesolliciteerd, ben ik uiteindelijk bij Thinkwise Software uitgekomen. Door te onderzoeken wat voor werk Thinkwise Software uitvoert, raakte ik erg geïnteresseerd in hun product. Het hoofdproduct van Thinkwise Software is namelijk de Software Factory, een geïntegreerde ontwikkelomgeving waarmee eenvoudig business software kan worden ontwikkeld. Tijdens mijn opleiding heb ik al vaker gewerkt aan ontwikkelomgevingen van software, dus ik had ook nog eens relevante ervaring.

Thinkwise Software had de afstudeeropdracht van de grafische modeler beschikbaar. Hiermee moet het mogelijk worden om het uiterlijk van de applicaties die ontwikkeld worden met de Software Factory, op grafische wijze aan te passen. De opdracht paste goed in de kennis die ik al bezat, maar bood ook nog genoeg uitdaging om erg leerzaam te kunnen zijn.

Deze scriptie is een documentatie van het werk dat ik heb uitgevoerd gedurende de afstudeerperiode. Het beschrijft waar ik precies aan heb gewerkt in mijn tijd bij Thinkwise Software. Het geeft ook de ervaringen aan die ik heb opgedaan tijdens het werken bij een daadwerkelijk bedrijf, in plaats van op school. Verder beschrijft het wat ik heb geleerd en hoe ik deze kennis kan toepassen om mijzelf en mijn werk te verbeteren.

Gedurende de afstudeerperiode ben ik begeleid door de docentbegeleider Martijn Lamers en mijn bedrijfsbegeleiders, Roel Winkels en Erik Brink. Ook betrokken was de afdelingsmanager en opdrachtgever Jasper Kloost. Ik wil hen graag bedanken voor de ondersteuning die zij hebben kunnen bieden en voor de mogelijkheid die zij mij hebben gegeven om mijzelf verder te kunnen ontwikkelen.

Ik wens u als lezer veel plezier met het lezen van deze scriptie. Ik hoop dat u ervan geniet om te leren over mijn periode bij Thinkwise Software, wat ik hier geleerd heb en welke ervaringen ik heb opgedaan.

Mathijs Stertefeld

Apeldoorn, 10 juni 2014

Inhoud

Samenvatting.....	5
Abstract.....	6
Verklarende Woordenlijst.....	7
1 Inleiding & Probleemstelling.....	10
2 Bedrijfsbeschrijving.....	11
2.1 Bedrijf.....	11
2.2 Medewerkers.....	11
3 Opdrachtformulering.....	13
3.1 Huidige Situatie Software Factory	13
3.2 Datamodel	14
3.3 Lijst en Formulier	15
3.4 Aanpassen Gebruikersinterface.....	15
3.5 Opdracht	17
3.6 Projectmethode	18
3.7 Projectuitvoering	19
4 Onderzoeksontwerp	20
4.1 Onderzoeksvragen	20
4.2 Onderzoeksmethode	21
5 Onderzoeksresultaten	22
5.1 Functionaliteit Huidige GUI Modeler.....	22
5.1.1 Lijst Modeler	22
5.1.2 Formulier Modeler.....	22
5.2 Gebruik Huidige GUI Modeler	22
5.3 Implementatie Grafische Modeler	23
5.3.1 Lijst Modeler	24
5.3.2 Formulier Modeler.....	25
5.4 Conclusie.....	26
6 Implementatieplan	27
6.1 Onderdelen.....	27
6.1.1 Basis Grafische Modeler	27
6.1.2 Lijst Modeler	27
6.1.3 Formulier Modeler.....	27
6.2 Planning	28
7 Implementatieproces.....	30
7.1 Basis Grafische Modeler	30
7.2 Lijst Modeler	31
7.3 Formulier Modeler.....	32
8 Conclusie En Aanbevelingen	35
8.1 Conclusie.....	35
8.2 Aanbevelingen	35

Evaluatie	36
Ervaringen	36
Leerpunten.....	37
Competenties.....	37
Referenties.....	38
Bijlages	39
Bijlage A: Project Initiation Document	40
Bijlage B: Onderzoeksverslag.....	67
Bijlage C: User Stories.....	88
Bijlage D: Use Cases	93
Bijlage E: Technische Beschrijving Implementatieproces	100
Bijlage F: Testresultaten	108
Bijlage G: Issue Lijst.....	114
Bijlage H: Competenties.....	115

Samenvatting

De afstudeerstage van Mathijs Stertefeld heeft plaatsgevonden bij Thinkwise Software, een ICT bedrijf in Apeldoorn. Het hoofdproduct van Thinkwise Software is de Software Factory, een geïntegreerde ontwikkelomgeving waarmee eenvoudig model gedreven business applicaties kunnen worden ontwikkeld. Voor deze applicaties kunnen veel onderdelen worden ingesteld, waaronder het aanpassen van de gebruikersinterface.

Het aanpassen van de gebruikersinterface gebeurt nu nog door het wijzigen van de waarden in een tabel, zonder direct te kunnen zien welk effect dit heeft op de eindapplicatie. Het doel van dit afstudeerproject is om de Software Factory uit te breiden met een grafische modeler voor het aanpassen van de gebruikersinterface van applicaties. Hiermee kan de gebruiker direct zien hoe de eindapplicatie er uit zal zien en welk effect wijzigingen hebben. Ook is het mogelijk om direct in deze weergave de onderdelen grafisch aan te passen, door gebruik te maken van verschillende grafische aanpassingstechnieken zoals Drag-And-Drop.

Gedurende het afstuderen is het project onderverdeeld in verschillende onderdelen. Eerst is vastgelegd wat het project inhoudt in het Project Initiation Document. Daarna is onderzocht wat er geïmplementeerd moet worden voor de grafische modeler en wat hiervoor de meest ideale manier is. Vervolgens is de daadwerkelijke grafische modeler ontwikkeld, getest en gedocumenteerd. Ten slotte is het volledige afstudeerproject gedocumenteerd in deze scriptie.

Aan het eind van de afstudeerperiode is het onderzoek uitgevoerd en is de grafische modeler ontwikkeld. Alle geplande hoofddoelen van het project zijn afgerond. Dit zorgt ervoor dat het project succesvol is afgerond.

Abstract

The graduation internship of Mathijs Stertefeld took place at Thinkwise Software, an ICT company located in Apeldoorn. Thinkwise Software's main product is the Software Factory, an integrated development environment which can be used to easily develop model driven business applications. The applications developed with it contain a great number of editable settings, among which is the ability to edit the user interface.

Editing the user interface is currently done by editing the values in a table, without being able to directly see what effect this has on the final application. The goal of this graduation project is to extend the Software Factory with a graphical modeler for editing the user interface of applications. This allows the user to directly see what the final application will look like and what effect the changes have on it. This view also allows for direct graphical manipulation of the various components, by use of various graphical editing techniques such as Drag-And-Drop.

The graduation project is divided into several parts. At first the contents of the project have been recorded in the Project Initiation Document. Next, research has been conducted to discover what has to be implemented and how this can be done in the most ideal way. Then the actual graphical modeler has been developed. Finally, the complete graduation project has been documented in this thesis.

At the end of the graduation period the research has been performed and the graphical modeler has been implemented. All the main goals of the project have been completed. This means that the project as a whole has been successfully completed.

Verklarende Woordenlijst

Term	Omschrijving
Agile	Bij agile software ontwikkeling wordt gebruik gemaakt van iteratief werken. In plaats van één groot ontwikkelingstraject wordt het ingedeeld in meerdere kleinere, losse ontwikkelingstrajecten.
Aggregatie	Een aggregatie is een functie die uitgevoerd wordt op een kolom van een tabel. Het wordt dan toegepast op alle rijen in die kolom. Een voorbeeld van aggregaties zijn bijvoorbeeld gemiddelde en totaal.
Auto hide	Auto hide is een manier om een scherm weer te geven in een interface. Normaal is dan alleen een verticaal tabblad zichtbaar aan de zijkant van het scherm. Door de muis over het tabblad te houden wordt het scherm uitgeklaapt naar de ware grote, waardoor de inhoud toegankelijk is. Ook is het mogelijk om het scherm vast te zetten, zodat het continu zichtbaar is.
Backlog	Binnen SCRUM wordt een backlog bijgehouden met user stories met taken. Elke sprint wordt er een aantal user stories van de backlog ingepland om uit te voeren.
Customer Solutions team	Het team dat applicaties ontwikkelt met behulp van de Software Factory voor en met klanten.
Software Factory database	In de database van de Software Factory worden de modellen van de verschillende applicaties opgeslagen. Voor elke applicatie wordt in aparte project versies het model voor elke versie apart opgeslagen.
Drag And Drop	Met drag en drop kunnen objecten verplaatst worden door er op te klikken en vervolgens met ingedrukte knop de cursor te verplaatsen. Het object wordt dan verplaatst naar de gewenste locatie.
Eindproduct	Een eindproduct van de Software Factory bestaat uit twee databases. De ene database bevat de gebruikersinformatie die wordt weergegeven in de applicatie. De andere database bevat het model van de applicatie, de informatie over hoe de applicatie is opgebouwd. Deze applicatie wordt ingeladen in de GUI om een functionele applicatie te vormen.
Formulier	Een formulier binnen de Software Factory is een overzicht van alle informatie van een entry in een database tabel. De informatie wordt weergegeven in velden en kan worden aangepast.
Formulier modeler	De formulier modeler is de ontwikkelde modeler die direct in de Software Factory toont hoe een formulier weergave van een tabel in de eindapplicatie er uit zal zien en waarin grafisch de instellingen hiervan gewijzigd kunnen worden.
Gebruikersinformatie	De informatie die de gebruiker invoert en kan wijzigen voor een applicatie. Deze wordt opgeslagen in de database van die applicatie.
Grafisch	Het direct visueel aanpassen van instellingen door gebruikersinterface interactie. Dit in plaats van indirect invoeren van de waardes.
Grafische modeler	De grafische modeler bestaat uit de lijst en de formulier modeler, waar mee op grafische wijze de instellingen voor een lijst en een formulier kunnen worden aangepast.
Grid	Een grid is een alternatieve benaming voor een lijst.
Huidige GUI modeler	De huidige GUI modeler bestaat uit een aparte modeler voor de lijst en het formulier. Hierin worden voor elke kolom in een lijst van een tabel de instellingen ingevoerd.

Klasse	Binnen object-georiënteerde programmeertalen is een klasse een blauwdruk voor objecten. Het voorziet de startwaardes en de implementatie van het gedrag.
Kolom	Binnen een lijst is een kolom een verticale verzameling van cellen. Een kolom beschrijft één eigenschap voor alle objecten in de lijst.
Label	Een label is een interface element waarin tekst wordt weergegeven. De gebruiker kan het label meestal niet wijzigen.
Lijst	Een lijst binnen de Software Factory is een tabel met rijen en kolommen waarin de entries van een database tabel worden weergegeven en kunnen worden gewijzigd.
Lijst modeler	De lijst modeler is de ontwikkelde modeler die direct in de Software Factory toont hoe een lijst weergave van een tabel in de eindapplicatie er uit zal zien en waarin grafisch de instellingen hiervan gewijzigd kunnen worden.
Model	Een model voor een Software Factory applicatie beschrijft de definitie van een Software Factory Applicatie. De Software Factory GUI interpreteert het model waardoor een volledig functionele applicatie ontstaat.
Model extender	Een uitbreiding van het model om extra functionaliteit toe te voegen aan een Software Factory applicatie. Deze worden in de Software Factory GUI ingeladen nadat het model van de Software Factory is ingeladen.
Model gedreven	De structuur en functionaliteit van een applicatie wordt opgeslagen in een model. Dit model wordt door een programma geïnterpreteerd waardoor een eindapplicatie ontstaat.
Modeler	Een modeler in de Software Factory biedt de mogelijkheid om aanpassingen te maken aan de gebruikersinterface van een applicatie.
MoSCoW	Een methodiek om prioriteit aan eisen te geven. De prioriteiten lopen van hoog naar laag: Must, Should, Could en Would.
Obj	In de GUI beschrijft een Obj de instellingen van een tabel.
Object	In object-georiënteerde programmeertalen is een object een instantie dat waardes en functionaliteit kan bevatten.
ObjProp	In de GUI beschrijft een ObjProp de instellingen van een kolom.
Property lijst	De lijst van eigenschappen van een lijst of een formulier. In de huidige GUI modeler wordt deze lijst weergegeven.
Product Innovation team	Het team van ontwikkelaars dat aanpassingen en nieuwe toevoegingen maakt aan de Software Factory.
Project	Een Software Factory project is een applicatie die gemaakt is en aangepast kan worden met de Software Factory. Een project kan bestaan uit meerdere project versies.
Projectversie	Projecten worden in versies opgeslagen. Hoe het model in elkaar zit kan verschillen per versie. Elke projectversie maakt echter gebruik van dezelfde database met data van de gebruikersinformatie.
Rij	Binnen een lijst is een rij een horizontale verzameling van cellen. De informatie in een rij hoort bij één object.
Schermbom	Een scherm bestaat uit een boom van schermelementen. Elk schermelement wordt weergegeven binnen de applicatie.
Schermelement	Een schermelement is een onderdeel van het totale scherm binnen de Software Factory. Het definieert welke componenten er in wordt weergegeven.

SCRUM	Een iteratieve agile ontwikkelingsmethode, waar korte sprints gebruikt worden om producten te ontwikkelen.
Software Factory	Een ontwikkelomgeving waarmee volledige model gedreven business applicaties kunnen worden ontwikkeld.
Software Factory GUI	De GUI is het programma dat wordt uitgevoerd om een applicatie te draaien. Het model dat wordt ingeladen bepaalt welke eindapplicatie er wordt uitgevoerd. Ook de Software Factory zelf wordt zo uitgevoerd. Er zijn verschillende GUI's, voor verschillende platforms.
Sprint	Binnen SCRUM is een sprint een korte ontwikkelingsperiode. Hierin wordt een product ontwikkeld en opgeleverd.
Tabblad	Wanneer een scherm niet genoeg ruimte heeft om alle informatie weer te geven, kan met een tabblad de overige informatie weergegeven worden.
Use case	In een use case is gewenste functionaliteit beschreven. Vervolgens kunnen tests worden uitgevoerd die controleren of het voldoet aan de eisen.
User story	Een user story beschrijft wat een interactie van een gebruiker met het systeem. Elke user story wordt onderverdeeld in taken die moeten uitgevoerd worden om de functionaliteit te ontwikkelen.
Veld	Binnen de Software Factory is een veld een interface element waarin data wordt weergegeven en kan worden gewijzigd. Voorbeelden van velden zijn textboxes, comboboxes en checkboxes.
What You See Is What You Get	Een What You See Is What You Get editor zorgt ervoor dat de interface die wordt gemaakt of aangepast gelijk wordt weergegeven zoals deze er in het eindproduct er uit zal zien.

1 Inleiding & Probleemstelling

Het doel van de afstudeerstage was om een uitbreiding te maken voor het hoofdproduct van Thinkwise Software, de Software Factory. Met de Software Factory kunnen volledige applicaties gemaakt worden. Van deze applicaties is het ook mogelijk om aan te passen hoe de user interface er uit zal zien. Momenteel wordt dit gedaan door een lijst van instellingen aan te passen, zonder direct en duidelijk te kunnen zien wat het effect van deze wijzigingen is op de applicatie.

De uitbreiding van de Software Factory moet ervoor zorgen dat het aanpassen van de user interface op grafische wijze gedaan kan worden, met behulp van de zogenoemde grafische modeler. Dit zorgt voor versimpeling en verduidelijking van het proces. Ook wordt hierin direct een beeld gegeven van hoe de eindapplicatie er uit zal komen te zien.

Het was de bedoeling dat er een prototype van de grafische modeler geïmplementeerd wordt binnen het afstudeerproject. Voordat het prototype echter kon worden gerealiseerd, is eerst onderzoek uitgevoerd moeten worden naar hoe deze het best ontworpen kon worden.

Voor het project is eerst onderzocht, door gebruik te maken van verschillende onderzoeksmethodes, wat de meest geschikte oplossingen zijn om de aanpasbare onderdelen op grafische wijze in te stellen. Uiteindelijk heeft dit een ontwerp opgeleverd voor de grafische modeler. Op dit ontwerp konden mogelijke implementaties gebaseerd worden. Door middel van literatuuronderzoek zijn ontwerpstandaarden geraadpleegd waarop het grafisch aanpassen van de onderdelen is gebaseerd. Diepte interviews met huidige gebruikers van de Software Factory hebben inzicht geven over gewenste manieren om de instellingen aan te passen. Verder is nog de al aanwezige kennis van de afstudeerder gebruikt voor eventuele oplossingen.

Met al deze verworven kennis is dan een ontwerp voor alle onderdelen van de grafische modeler opgesteld. Dit heeft dan antwoord geven op de hoofdvraag van het onderzoek: *“Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen om het ontwerpen van de user interface gebruiksvriendelijker te maken?”*

2 Bedrijfsbeschrijving

Hieronder is kort aangegeven wat Thinkwise Software als bedrijf doet en waar het hoofdproduct van de Software Factory voor wordt gebruikt. In het Product Initiation Document in [bijlage A](#) is een uitgebreidere beschrijving te vinden.

2.1 Bedrijf

“Thinkwise Software is een specialist in computergestuurde softwarebouw en uitvinder van de zeer innovatieve Software Factory. De Thinkwise Software Factory Suite is een geïntegreerde ontwikkelomgeving waarmee eenvoudig business software kan worden gebouwd, getest, gedeployed en beheerd. Het is zelfs mogelijk bestaande software automatisch te moderniseren.

De Thinkwise omgeving is vooral geschikt voor het bouwen van business software, zoals ERP applicaties, randsystemen, research applicaties en apps. Dit type software kan bestaan uit workflow, gebruikersinterface in Windows, Web of mobiele apparaten, applicatielogica/services en database/data services.

Software Factory

Hiermee bouwt en test u uw oplossing. U kunt de requirements vastleggen en met de business bespreken. Aan de hand van de requirements wordt een model gemaakt, bestaande uit workflow, procesmodel, datamodel, GUI model en applicatielogica. De modellen kunt u volledig grafisch valideren en analyseren.” (Thinkwise Software, 2013)

2.2 Medewerkers

Binnen Thinkwise Software zijn er teams met ieder verschillende werkzaamheden. Het Product Innovation team werkt aan de Software Factory zelf. Zij zorgen ervoor dat er nieuwe features worden geïmplementeerd, dat bestaande features worden aangepast en dat problemen met de Software Factory worden verholpen.

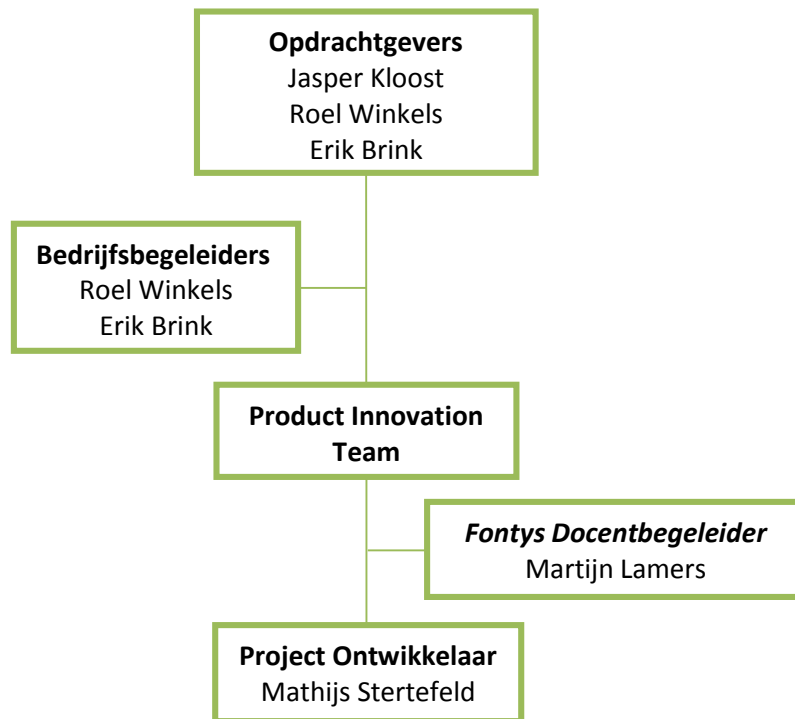
Het Customer Solutions team werkt aan het maken van applicaties met behulp van de Software. Deze applicaties kunnen voor zowel intern gebruik zijn, als voor klanten.

Het Sales team zorgt ten slotte er nog voor dat er nieuwe klanten worden aangetrokken, of dat er nieuwe projecten worden uitgevoerd voor bestaande klanten.

De afstudeerder is een onderdeel van het Product Innovation team. Gedurende de afstudeerperiode heeft hij gewerkt aan het uitbreiden van de functionaliteit van de Software Factory. De opdrachtgever van de afstudeeropdracht is tevens de manager van het Product Innovation team, Jasper Kloost. Hij zorgt voor definiëring van de opdracht en controleert of het project voldoet aan zijn eisen. Hiervoor woont hij ook de demonstraties van de tussenproducten bij.

Tijdens het afstudeerproject is de afstudeerder begeleid door twee bedrijfsbegeleiders. Dit zijn Product Innovation Specialists, Roel Winkels en Erik Brink. Zij hebben gezorgd voor begeleiding van de afstudeerder op verschillende manieren. Zij hebben ook directe voortgangsccontrole verzorgd, om zeker te zijn dat het project zich aan de planning houdt. Verder hebben zij technische ondersteuning geboden, door middel van toelichting op de werking van de Software Factory en haalbaarheid van geplande features van het afstudeerproject. Ten slotte hebben zij nog feedback geleverd op de gemaakte producten.

In het onderstaande organigram in figuur 1 is de structuur weergegeven van de betrokkenen bij het project.



Figuur 1. De organigram van de betrokkenen van het project.

Tijdens de implementatie fase van het project is gebruik gemaakt van de SCRUM ontwikkelmethode. (Schwaber & Sutherland, 2013) De opdrachtgever heeft hierin de rol van product-owner gespeeld. De bedrijfsbegeleiders hebben de rol van scrummasters en ook deels product-owners gespeeld.

3 Opdrachtformulering

3.1 Huidige Situatie Software Factory

De Software Factory is een geïntegreerde ontwikkelomgeving waarmee model gedreven applicaties kunnen worden ontwikkeld. Deze applicaties bestaan uit twee delen.

Het eerste deel is het model. Dit model definieert de structuur, logica en gebruikersinterface van de applicatie. Bij het maken van een applicatie wordt de volledige applicatie ontwikkeld naar de eisen. Het model wordt vervolgens opgeslagen in een relationele database.

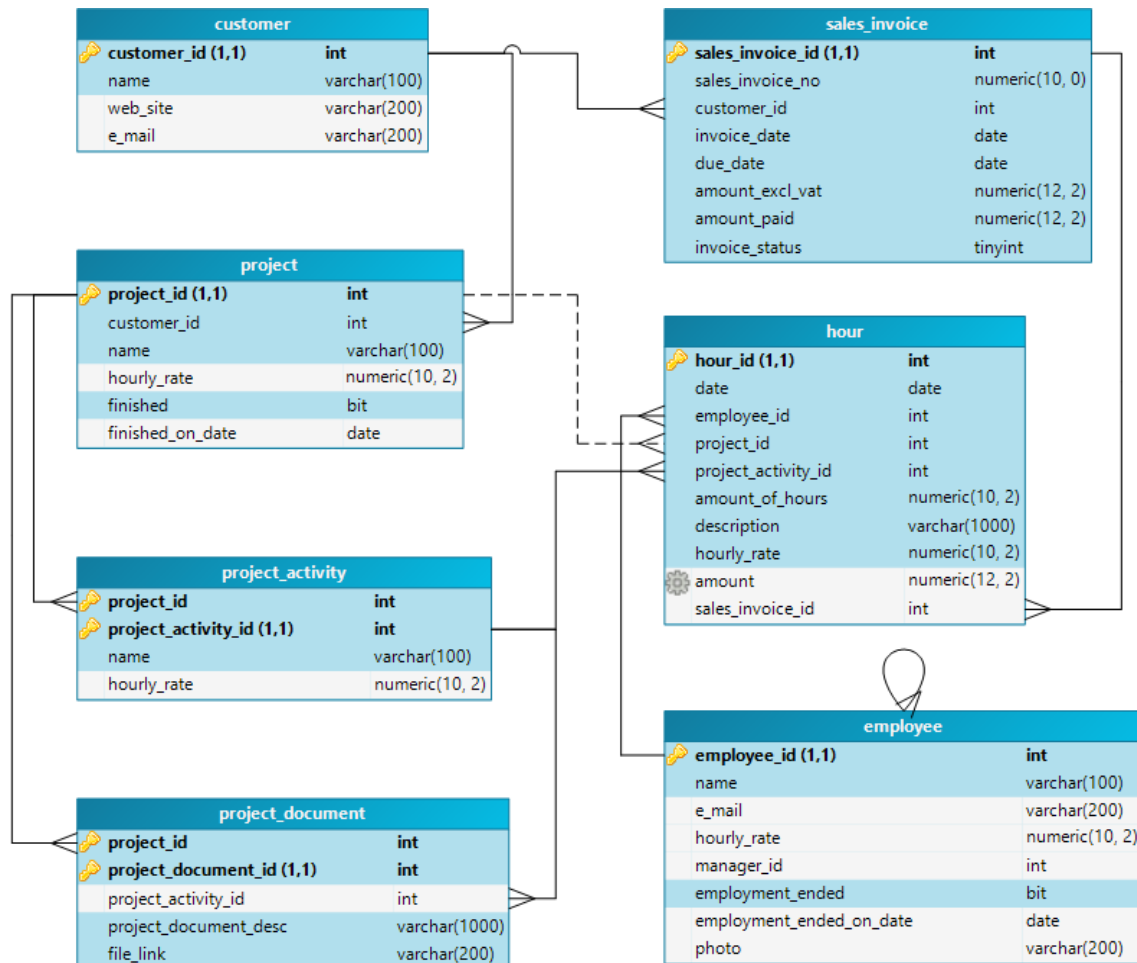
Dit model wordt dan geopend met het tweede deel, de Software Factory GUI. Deze interpreteert het model en levert een functionele applicatie op. Er zijn verschillende GUI's, voor verschillende platformen, wat het ontwikkelen van crossplatform applicaties zeer eenvoudig maakt.

De informatie die vervolgens in de applicatie wordt ingevoerd en gewijzigd, wordt opgeslagen in een aparte database.

3.2 Datamodel

Wanneer er in de huidige situatie een applicatie gemaakt wordt met de Software Factory, wordt eerst een project aangemaakt. Voor dit project wordt het model opgesteld om de applicatie te beschrijven. Het eerste onderdeel van het model dat wordt gemaakt is het datamodel.

Hierin stelt de gebruiker op welke data er in de applicatie komt en de relaties hiertussen. Deze data wordt opgeslagen door middel van tabellen en views. Per tabel wordt aangegeven welke velden er zijn. De relaties tussen de tabellen worden aangegeven door middel van sleutels. Een voorbeeld van een datamodel van een applicatie is te zien in figuur 2.



Figuur 2. Een datamodel in de Software Factory

Met het complete datamodel kan vervolgens het model in de Software Factory GUI worden geladen. Het model wordt dan geïnterpreteerd en de applicatie wordt opgestart. Als er verder niets is ingesteld, wordt de applicatie op de standaard manier weergegeven. Voor elke tabel in het datamodel is er een scherm. In dit scherm wordt de informatie van de tabel getoond en kan worden bewerkt.

3.3 Lijst en Formulier

Om gebruik te maken van de data in de tabellen in de eindapplicatie, zijn er twee belangrijke vensters. Het eerste venster wordt de “Lijst” weergave genoemd. Hierin worden in een lijst de velden van de tabel weergegeven. Vervolgens is er de “Formulier” weergave. Hierin wordt de informatie van één specifiek record in de tabel weergegeven in een aantal velden. Ook zijn er hier knoppen om verschillende acties uit te voeren, zoals het aanmaken, wijzigen of verwijderen van records. Als de gebruiker geen aanpassingen heeft gemaakt aan de gebruikersinterface, wordt voor een tabel de data standaard weergegeven in de lijst weergave bovenin en de formulier weergave onderin, zoals te zien in figuur 3.

Date	Employee ID	Project id	Project activity id	Amou	Description	Hourly rate	Amount	Invoice ID
02/06/2014	David Davidse	IKEA Kaste	Kast in elkaar zette	5	Kast bouwen	60.00	300.00	1002
02/03/2014	Jan Jansen	Project On	Do project	10	Make project	250.00	2500.00	1001
01/05/2014	Petra Petruse	IKEA Kaste	Kast in elkaar zette	2	Kast bouwen	30.00	60.00	1002
01/26/2014	Stephen Haw	Project Tw	Design project	1	Design project	250.00	250.00	1000
01/27/2014	Stephen Haw	Project Tw	Design project	2	Design project	250.00	500.00	1000
02/03/2014	Ella Alle	Project On	Do project	1	Make project	20.00	20.00	1001

Hour - 1

Hour - 2

Date: 1/26/2014

Employee ID: Stephen Hawking

Project id: Project Two

Project activity id: Design project

Amount of hours: 1

Figuur 3. Een lijst en een formulier van een bepaalde tabel.

Deze vensters worden met een standaard layout gegenereerd. Het is ook mogelijk om deze aan te passen, zo kan bijvoorbeeld alleen de lijst of alleen het formulier getoond worden. Ook kan een volledig andere weergave van de data worden gebruikt. Weergaves kunnen samen gecombineerd worden om complexe schermindelingen te maken, om zo schermen te kunnen vormen die aan specifieke wensen van de klant voldoen.

De standaard lijst en formulier worden echter het meeste gebruikt in applicaties. Voor de lijst en het formulier is er een groot aantal instellingen dat kan worden aangepast om de data te tonen.

3.4 Aanpassen Gebruikersinterface

Voor een lijst kan onder andere de standaard breedte van een kolom worden aangepast. Ook is het mogelijk om het type van de kolom binnen de lijst in te stellen, dit geeft de zichtbaarheid en bewerkbaarheid van een kolom aan. Verder is er nog de mogelijkheid om verschillende soorten van aggregatie voor een kolom in de lijst te tonen.

Deze instellingen zijn te zien in figuur 4. Verder zijn er een aantal instellingen die van toepassing zijn op de volledige lijst, in plaats van slechts op een kolom. Zo kan bijvoorbeeld ingesteld worden of de eindgebruiker de lijst kan bewerken of niet.

Tabel

Kolommen

Lijst

Formulier

Lookup tabellen

Detailtabbladen

Zoeken / Filteren

Sorteren

Kolom id	Handm. kolombreedte lijst	Lijst kolom breedte (pixels)	Afwijkend lijst type	Aggregatie in lijst tonen	Aggregatie type
▶ hour_id	<input checked="" type="checkbox"/>	25	Verborgen	<input type="checkbox"/>	
date	<input checked="" type="checkbox"/>	50	Regulier	<input type="checkbox"/>	
employee_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>	
project_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>	
project_activity_id	<input type="checkbox"/>		Regulier	<input type="checkbox"/>	
amount_of_hours	<input type="checkbox"/>		Regulier	<input checked="" type="checkbox"/>	Sommeren
description	<input type="checkbox"/>		Regulier	<input type="checkbox"/>	
hourly_rate	<input checked="" type="checkbox"/>	50	Regulier	<input type="checkbox"/>	
amount	<input checked="" type="checkbox"/>	50	Alleen lezen	<input checked="" type="checkbox"/>	Sommeren
invoice_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>	

Figuur 4. Het wijzigen van de GUI van de lijst van een tabel.

Voor een formulier kunnen nog meer waardes in worden gesteld. Zo kan bijvoorbeeld het formaat van velden en de bijbehorende labels worden ingesteld. Groepen van velden worden gemaakt met hun eigen labels en iconen. Ook kunnen de velden worden ingedeeld in tabbladen. Hier kan ook weer het type van een veld worden ingesteld, om de zichtbaarheid en aanpasbaarheid in te stellen.

Deze instellingen zijn te zien in figuur 5. Verder zijn er ook voor het formulier instellingen die voor het hele formulier van toepassing zijn. Zo kan bijvoorbeeld worden ingesteld of de eindgebruiker nieuwe records kan toevoegen of bestaande records kan aanpassen of verwijderen.

Tabel	Kolommen	Lijst	Formulier	Lookup tabellen	Detailtabbladen	Zoeken / Filteren	Sorteren	Prefilters				
Kolom id	Label	Vel	Vel	Veld	Veld in volg	Label	Volgende groep	Veld in volgen	Veld op volg	Label	Volgende	Afwijkend formul
▶ hour_id			1	<input checked="" type="checkbox"/>	general	D:\Software_fabr	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
date			1	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>			
employee_id			1	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>			
project_id			1	<input checked="" type="checkbox"/>	activity	D:\Software_fabr	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
project_activity_id			1	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>			
amount_of_hours			1	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>			
description			1	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>			
hourly_rate			1	<input checked="" type="checkbox"/>	financial	D:\Software_fabr	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		financial		

Figuur 5. Het wijzigen van het formulier van een tabel.

Na het aanpassen van deze instellingen voor de lijst en het formulier moet steeds in de eindapplicatie het model worden ververs om te kunnen zien welk effect de wijzigingen hebben op hoe de eindapplicatie er uit ziet. Dit zorgt voor erg veel testwerk bij het maken van een GUI, waarin de waardes steeds worden veranderd en dan in het programma moet worden gekeken hoe de interface er daadwerkelijk uit ziet.

3.5 Opdracht

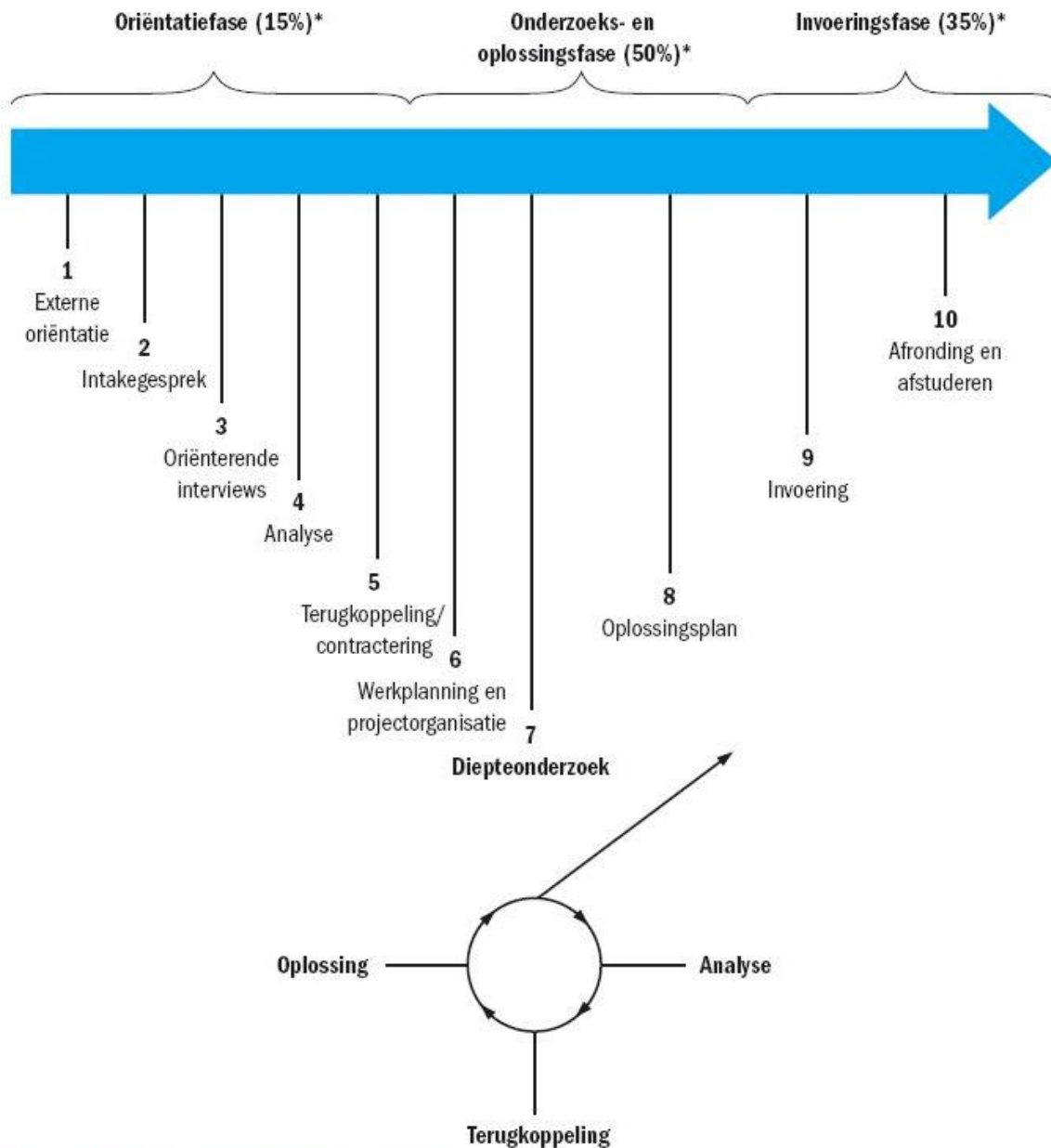
Momenteel is het ontwikkelen van de gebruikersinterface met de Software Factory dus nog niet ideaal. Om dit te verbeteren is deze afstudeeropdracht in leven geroepen. Thinkwise Software wil ervoor zorgen dat het aanpassen van de gebruikersinterface op een gebruiksvriendelijke manier kan worden gedaan. De eis van een grafische modeler zal hiervoor moeten zorgen.

Hierbij moet gebruik worden gemaakt van “What-You-See-Is-What-You-Get” en “Drag and Drop” functionaliteit om schermen in te richten. Hiermee wordt direct in de Software Factory weergegeven hoe de lijst en het formulier er in het eindproduct er uit zullen zien. De wijzigingen die dan worden gemaakt zijn dus direct zichtbaar. Dit zorgt ervoor dat makkelijker en sneller de gebruikersinterface van eindapplicaties kan worden ontwikkeld.

3.6 Projectmethode

Voor het afstudeerproject is gebruik gemaakt van het Tien Stappen Plan, te zien in figuur 6. Stap één tot en met zes, waarin de oriëntatie en planning van het project zijn uitgevoerd, zijn reeds al vastgelegd in het Project Initiation Document, te vinden in [bijlage A](#). De rest van het project bestaat de overige vier stappen: stap zeven Diepteonderzoek, stap acht Oplossingsplan, stap negen Invoering en stap tien Afronding en Afstuderen.

Voor de implementatie is gebruik van de SCRUM ontwikkelmethode (Schwaber & Sutherland, 2013), om de implementatie te plannen en te beheren.



*Aanbevolen tijdsverdeling van het totale project in fasen van het TSP

Figuur 6. Het Tien Stappen Plan. Bron: (Kempen & Keizer, 2011)

3.7 Projectuitvoering

In stap zeven is in het diepteonderzoek gekeken naar hoe de huidige GUI modeler in elkaar zit en welke mogelijkheden er zijn om de gebruikersinterface aan te passen. Ook is het gebruik van de huidige GUI modeler onderzocht om de belangrijkste onderdelen te kunnen bepalen. Hiermee kon dan een ontwerp worden opgesteld voor een grafische modeler, wat een deel beslaat van stap acht, het Oplossingsplan.

In de implementatie is het ontwerp dat uit het onderzoek komt, gerealiseerd. Eerst is de geplande implementatie ontworpen, wat de rest van stap acht beslaat. Er is vervolgens een prototype ontwikkeld waarmee de verschillende instellingen voor de lijst en het formulier op grafische wijze kunnen worden aangepast. Hierbij was het ook van belang dat de technische documentatie in orde was, zodat er na het project aan kan worden doorontwikkeld. Dit is stap negen van het Tien Stappen Plan, de Invoering.

Aan het eind van de implementatie is het volledige product gedocumenteerd in een scriptie, die uiteindelijk verdedigd wordt voor de begeleiders en assessors. Hierin is vastgelegd wat er is uitgevoerd en hoe dit proces is verlopen. Dit beschrijft dan de tiende stap van het Tien Stappen Plan, Afronding en Afstuderen.

Uiteindelijk zijn dus de volgende producten uit het project gekomen:

- Project Initiation Document
- Onderzoeksverslag
- Prototype Grafische Modeler
- Scriptie

4 Onderzoeksontwerp

4.1 Onderzoeksvragen

Het project heeft als doel om de huidige GUI modeler te vervangen met een grafische modeler. Dit moet ervoor zorgen dat het aanpassen van de user interface van applicaties eenvoudiger en efficiënter wordt en er direct overzicht is van hoe het eindproduct er uit zal zien. De hoofdvraag van het onderzoek luidt dan ook: *“Welke functionaliteiten bevat de huidige GUI modeler en hoe kunnen deze worden geïmplementeerd in een grafische modeler?”* Om deze hoofdvraag is deze onderverdeeld in een aantal deelvragen. Wanneer deze allemaal zijn beantwoord kan antwoord worden gegeven op de hoofdvraag.

Om ervoor te zorgen dat de grafische modeler aan deze eisen voldoet, moet het dezelfde functionaliteiten hebben als de huidige GUI modeler. Het is dus de bedoeling dat er geen verlies in functionaliteit is. In de nieuwe grafische modeler mogen dingen niet lastiger worden of niet gedaan kunnen worden. Om dit te garanderen, wordt gekeken hoe gebruikers momenteel werken met de huidige GUI modeler. Dit zal de eerste deelvraag beantwoorden, welke luidt: *“Hoe wordt momenteel gebruik gemaakt van de huidige GUI modeler?”*

Ook wordt onderzocht welke functionaliteiten de huidige GUI modeler heeft. Er wordt gekeken welke onderdelen van de GUI aanpasbaar zijn en op welke wijze deze aangepast kunnen worden. Dit zal antwoord geven op de tweede deelvraag: *“Welke onderdelen kunnen met de huidige GUI modeler aangepast worden?”*

Vervolgens wordt voor elk van deze onderdelen gekeken in hoeverre het mogelijk is om ze te bewerken met een grafische modeler. Verder wordt hier gekeken of er geen grafische oplossingen zijn om bepaalde instellingen aan te passen. Hier wordt dan gekeken wat een geschikte alternatieve oplossing kan zijn. Dit zorgt voor een antwoord op de derde deelvraag: *“Welke onderdelen zijn wel of niet geschikt om met een grafische modeler aan te passen?”*

Ten slotte wordt er gekeken naar alle onderdelen die aanpasbaar zijn. Er wordt onderzocht wat de meest geschikte manier is om elk onderdeel aan te passen op grafische wijze. Er wordt gekeken hoe dit gedaan kan worden op een dusdanige manier dat het eenvoudig en intuïtief is voor de gebruiker. Dit beantwoordt dan de vierde deelvraag: *“Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen?”*

Door al deze deelvragen te beantwoorden wordt een duidelijk overzicht gegeven van de functionaliteit van de huidige GUI modeler. Ook geeft het aan wat de functionaliteiteisen zijn van de grafische modeler. Dit zorgt dan voor een ontwerp van de grafische modeler, dat gebruikt kan worden voor de implementatie hiervan. Ook levert dit dan antwoord op de bovenstaande hoofdvraag.

4.2 Onderzoeksmethode

Voor het onderzoek is eerst gekeken naar de huidige GUI modeler. Alle onderdelen zijn onderzocht die hiermee kunnen worden aangepast en ingesteld. Deze zijn vervolgens allemaal in kaart gebracht.

Gebaseerd op alle aanpasbare instellingen is vervolgens een ontwerp gemaakt voor hoe deze verschillende instellingen grafisch aangepast kunnen worden. Hiervoor is gebruik gemaakt van tijdens de opleiding opgedane kennis, gecombineerd met literatuuronderzoek naar bestaande standaard interacties om gelijkwaardige instellingen aan te passen. Dit heeft uiteindelijk een ontwerp opgeleverd waarin wordt uitgelegd hoe het aanpassen van al deze instellingen op grafische wijze gedaan zou kunnen worden. Dit ontwerp is te vinden in het onderzoeksverslag in [bijlage B](#).

Hierna zijn er interviews gehouden met de gebruikers van de huidige GUI modeler in de Software Factory. Het ontwerp is aan hen voorgelegd, om aan te geven wat het plan is voor de grafische modeler. Daarna zijn er vragen gesteld over hoe zij de huidige GUI modeler gebruiken en welke eisen zij hebben aan de grafische modeler. Dit heeft voor extra inzicht gezorgd over hoe er precies gebruik van wordt gemaakt en welke onderdelen van de GUI modeler het belangrijkst zijn. Aan de hand van de informatie gewonnen uit deze interviews is het eerder in het onderzoek gemaakte ontwerp aangepast.

Dit ontwerp is vervolgens voorgelegd aan de ontwikkelaars van de Software Factory. Omdat zij veel kennis hebben over het ontwerp en het gebruik ervan, waren zij in staat om feedback te geven op het ontwerp. Aan de hand van deze feedback zijn wijzigingen en toevoegingen gemaakt aan het ontwerp.

Verder is er ook toelichting gegeven op de code van de huidige GUI modeler. Ook is de code zelf onderzocht om te kijken in hoe verre het ontwerp technisch mogelijk was om te implementeren. Omdat er vrij weinig documentatie beschikbaar was over de werking en structuur, is dit gedaan door veel testwerk en vragen aan de ontwikkelaars.

5 Onderzoeksresultaten

De resultaten uit het onderzoek hebben de nodige informatie opgeleverd om de deelvragen te kunnen beantwoorden. Hieronder zijn de resultaten van het onderzoek samengevat weergegeven en worden de deelvragen beantwoord. De volledige onderzoeksresultaten zijn te vinden in het onderzoeksverslag in [bijlage B](#).

5.1 Functionaliteit Huidige GUI Modeler

Met de Software Factory kan een groot aantal verschillende onderdelen van een applicatie worden aangepast. Niet alle onderdelen die aangepast kunnen worden, zitten ook in de huidige GUI modeler. Deze onderdelen moeten op andere plaatsen in de Software Factory aangepast worden.

De belangrijkste onderdelen van de GUI modeler zijn de lijst en de formulier modeler. Hiermee kan de indeling van lijsten en formulieren worden aangepast. Hieronder staan voor beide de onderdelen die aangepast kunnen worden die relevant zijn voor de GUI.

In hoofdstuk 3 Opdrachtformulering is in figuur 4 en 5 de huidige GUI modeler weergegeven.

5.1.1 Lijst Modeler

In de lijst modeler wordt de indeling van de kolommen in een lijst ingedeeld. In de huidige formulier modeler kan een aantal instellingen worden gewijzigd. Het volledige overzicht is te vinden in het onderzoeksverslag in [bijlage B](#).

5.1.2 Formulier Modeler

In de formulier modeler wordt de indeling van velden op een formulier ingedeeld. In de huidige formulier modeler kan per veld een aantal instellingen worden gewijzigd. Het volledige overzicht is te vinden in het onderzoeksverslag in [bijlage B](#).

5.2 Gebruik Huidige GUI Modeler

Om te onderzoeken hoe de huidige GUI modeler wordt gebruikt, zijn er interviews gehouden met medewerkers van Thinkwise Software van het Customer Solutions team. Deze medewerkers maken met grote regelmaat gebruik van de huidige GUI modeler.

De interviews zijn gehouden met:

- Arjan Sollie, Customer Solutions Creator
- Frank Wijnhout, Project Manager Customer Solutions
- Roland van Aggele, Project Manager Customer Solutions

Voor de interviews is eerst het bestaande ontwerp getoond en doorgesproken. Hierna zijn een aantal vragen gesteld over verschillende onderwerpen van zowel de huidige GUI modeler, als over de grafische modeler. Hieronder is voor alle onderdelen aangegeven wat de belangrijkste punten bleken uit de interviews. De interviews zelf zijn te vinden in het onderzoeksverslag in [bijlage B](#).

Wijzigen Schermen

Het scherm waar het meeste aan wordt aangepast is het formulier. Dit scherm wordt door de eindgebruikers erg veel gebruikt en heeft een groot aantal instellingen die aangepast kunnen worden. Hierdoor is dit de belangrijkste modeler.

Volgorde Van Wijzigingen

Bij het aanpassen van de gebruikersinterface van een applicatie begint de gebruiker gewoonlijk met het aanpassen van het formulier. Hier wordt dan de structuur van het formulier opgezet. De velden worden in de juiste volgorde gezet en er worden tabbladen en groepen aangemaakt om de velden die relevantie hebben bij elkaar te zetten. Daarna worden pas de kleinere details van de velden zelf aangepast, zoals bijvoorbeeld het formaat. Nadat het formulier is ingesteld, wordt de lijst aangepast.

Meest Gewijzigde Instellingen

Bij veel projecten wordt gebruik gemaakt van het wijzigen van de volgorde van velden in het formulier, om relevante velden bij elkaar te zetten. Vervolgens wordt er dan ook veel gebruik gemaakt van tabbladen en groepen, voor verdere indeling van relevante velden. Als laatst wordt het formaat van velden en labels ook nog vaak aangepast.

Minst Gewijzigde Instellingen

Bij het ontwerpen van formulieren wordt het open laten van lege regels tussen velden minder vaak gebruikt. Hiervoor wordt dan vaker gebruik gemaakt van een nieuwe groep.

Wijzigen Instellingen naar Zelfde Waarde

Soms passen gebruikers de instellingen van verschillende velden aan naar dezelfde waarde. In de GUI modeler wordt dit vooral gedaan bij het aanpassen van de breedte van velden. Deze worden dan allemaal op hetzelfde formaat gezet, afwijkend van het standaard formaat. Verder wordt ook vaak de zichtbaarheid van meerdere velden tegelijkertijd aangepast.

Missende Instellingen in Huidige GUI Modeler

Een feature die nu erg gemist wordt in de GUI modeler is het direct kunnen zien wat het effect is van de wijzigingen aan de interface. Er worden enkele workarounds gebruikt om op andere plaatsen in de Software Factory te kunnen zien welke invloed gemaakte wijzigingen hebben op het eindproduct.

5.3 Implementatie Grafische Modeler

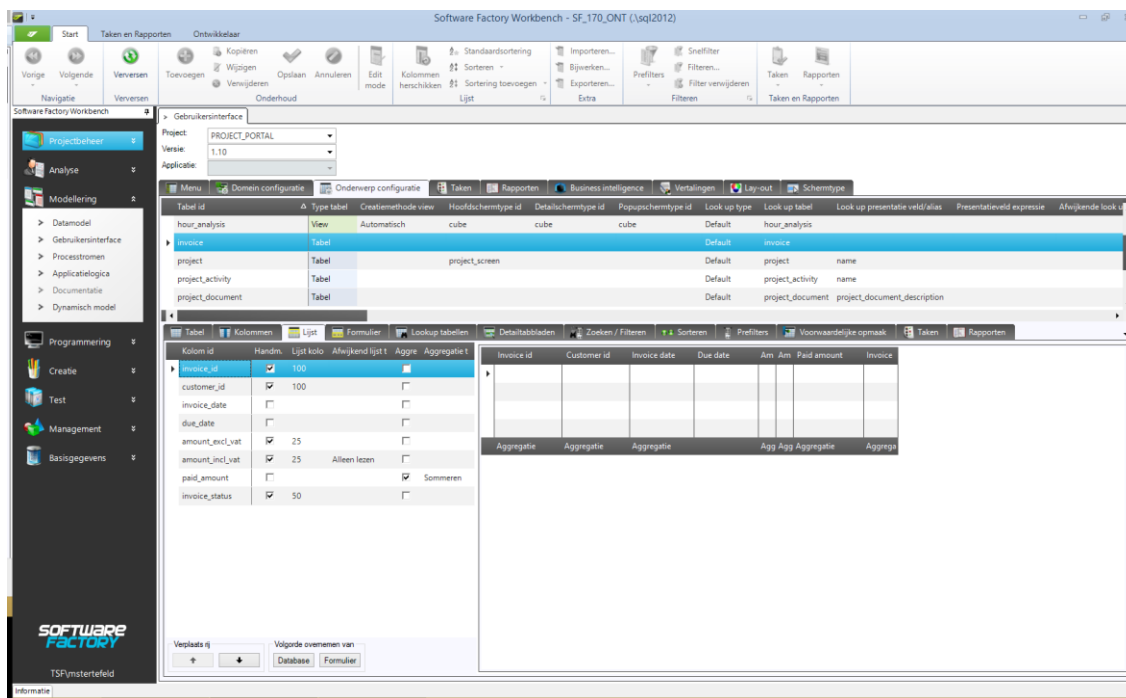
Door overleg met gebruikers en ontwikkelaars van de Software Factory en het maken van verschillende ontwerpen is er voor gekozen om de grafisch modeler te laten bestaan uit een “What-You-See-Is-What-You-Get” editor. In deze editor wordt de layout van de interface van de eindapplicatie op grafische wijze ingesteld. Hoe de applicatie er dan in deze editor uit ziet, komt overeen met hoe de eindapplicatie er dan uiteindelijk uit zal zien. Hierdoor hoeft niet de eindapplicatie zelf te worden opgestart om te bekijken hoe deze er uit ziet.

Om er voor te zorgen dat de grafische modeler geen functionaliteitsverlies heeft, vergeleken met de huidige GUI modeler, moeten alle onderdelen die in het vorige hoofdstuk zijn genoemd aanpasbaar zijn in de grafische modeler. Hieronder staat hoe de grafische modeler precies geïmplementeerd kan worden.

5.3.1 Lijst Modeler

In de lijst modeler worden de instellingen aangepast van de weergave van de informatie in de lijst. Voor het grafisch wijzigen van elk van de aanpasbare instellingen is in het onderzoeksverslag in [bijlage B](#) de gewenste implementatie gegeven.

In figuur 6 hieronder is het ontwerp weergegeven van hoe de implementatie van lijst modeler in de Software Factory er mogelijk uit zal komen te zien.



Figuur 6. Het ontwerp van de mogelijke implementatie van de lijst modeler.

5.4 Conclusie

Het onderzoeken van de Software Factory heeft een duidelijk overzicht gegeven van alle functionaliteiten die in de huidige GUI modeler zitten. Dit heeft helder in kaart gebracht welke onderdelen met de huidige GUI modeler aangepast kunnen worden.

De interviews met de gebruikers van de Software Factory hebben goed aangegeven wat de meest belangrijke onderdelen zijn. Hieruit blijkt dat het meest gewerkt wordt met de formulier modeler. Hier wordt dan vooral gebruik gemaakt van het aanpassen van de volgorde van velden en het indelen van de velden in verschillende soorten groepen. Deze interviews hebben een goed beeld kunnen geven van hoe momenteel gebruik wordt gemaakt van de huidige GUI modeler.

De informatie gekregen uit de interviews, gecombineerd met de feedback op het ontwerp, heeft gezorgd voor aanpassingen aan het oorspronkelijke ontwerp van de grafische modeler. Door vervolgens te overleggen met de andere ontwikkelaars van de Software Factory is gebleken welke onderdelen wel en niet geschikt zijn om aan te passen met de grafische modeler. Zo kunnen veel onderdelen worden geïmplementeerd in een “What-You-See-Is-What-You-Get” editor en kunnen de overige instellingen worden gewijzigd in een menu binnen de grafische modeler.

De informatie uit de interviews is gecombineerd met de onderzochte standaarden, te vinden in het onderzoeksverslag in [bijlage B](#), voor interface ontwerp. Dit heeft voor alle aanpasbare instellingen een manier opgeleverd om deze grafisch te bewerken. Hiermee is bepaald wat de optimale methodes zijn om de verschillende onderdelen met de grafische modeler aan te passen.

Met alle verzamelde informatie uit het onderzoek is dus vastgelegd wat de functionaliteit is van de huidige GUI modeler en hoe deze geïmplementeerd kan worden in een grafische modeler. Dit is vastgelegd in een ontwerp dat is gebruikt om de daadwerkelijke grafische modeler te ontwikkelen.

6 Implementatieplan

Uit het onderzoek is gebleken welke instellingen allemaal aanpasbaar zijn. Ook zijn voor alle aanpasbare instellingen mogelijkheden voorgesteld om deze te wijzigen. Tijdens de implementatieperiode is gewerkt om deze voorgestelde functionaliteiten te realiseren. De te realiseren functionaliteiten zijn onderverdeeld in verschillende onderdelen die los zijn geïmplementeerd. Deze verschillende onderdelen zijn vervolgens ingepland in de gegeven tijdsperiode.

6.1 Onderdelen

De implementatie van de grafische modeler is opgedeeld in drie onderdelen. Deze onderdelen zijn achter elkaar uitgevoerd. Voor het project gebruik is gemaakt van de SCRUM ontwikkelingsmethode. (Schwaber & Sutherland, 2013) Daarom zijn er voor elk onderdeel user stories met taken opgesteld. Voor de verschillende aanpasbare onderdelen zijn ook use cases opgesteld. Hiermee kan worden gecontroleerd of een onderdeel is voltooid en aan de eisen voldoet. Wanneer alle use cases van een onderdeel zijn voltooid, is het onderdeel succesvol geïmplementeerd.

6.1.1 Basis Grafische Modeler

Er is begonnen met de basis van de grafische modeler. Dit bestaat uit een scherm in de Software Factory. In dit scherm wordt aan de linkerkant de property lijst weergegeven. Hierin staan dezelfde instellingen als in de huidige GUI modeler.

Aan de rechterkant is de lijst of formulier modeler zichtbaar. Hier wordt de lijst of het formulier getoond zoals het in de eindapplicatie er uit zal zien. Ook is hier later functionaliteit aan worden toegevoegd om daadwerkelijk grafisch de instellingen aan te kunnen passen.

In hoofdstuk 3 is in figuur 4 en 5 te zien hoe het aanpassen van de instellingen voor de implementatie van de grafische modeler werd uitgevoerd. In hoofdstuk 5 is in figuur 6 en 7 te zien hoe dit in de grafische modeler weergegeven zal worden.

6.1.2 Lijst Modeler

Nadat de basis grafische modeler was geïmplementeerd, is begonnen met de lijst modeler. Deze was al eerste gekozen omdat het minder functionaliteit en complexiteit bevat dan de formulier modeler. Hierdoor kon er meer kennis over de Software Factory worden verzameld. Door middel van de lijst modeler kunnen op grafische wijze aanpassingen worden gemaakt aan de indeling van de lijst.

6.1.3 Formulier Modeler

Wanneer de lijst modeler was afgerond, is begonnen met de formulier modeler. Deze bevat verreweg de meeste functionaliteit. Doordat op dat punt in het proces de basis grafische modeler en de lijst modeler al waren ontwikkeld, was er een voldoende kennis over de code en interne werking van de Software Factory om de formulier modeler te kunnen implementeren.

6.2 Planning

Het implementeren van de grafische modeler was ingepland over een periode van negen weken. Voor het implementatie is gebruik gemaakt van de SCRUM ontwikkelingsmethode. (Schwaber & Sutherland, 2013) Hiervoor was dan voor elk van de drie onderdelen een sprint van drie weken ingepland. In elke sprint is begonnen met onderzoek naar hoe het onderdeel geïmplementeerd moet worden. Dit onderzoek bestaat uit het inspecteren van de huidige code en het technisch ontwerpen van de mogelijke implementatie.

Vervolgens is het onderdeel gerealiseerd. Hierin zijn de geplande functionaliteiten ontwikkeld. De gemaakte aanpassingen zijn overzien door de bedrijfsbegeleiders en eventuele feedback is hier op gegeven.

Aan het einde van de sprint is de functionaliteit nog een keer getest. In de use cases in [bijlage D](#) is voor alle verschillende onderdelen vastgelegd hoe het moet functioneren. Voor elke use case is de functionaliteit getest. Dit is gebeurd door het uitvoeren van black-box testing op een aantal verschillende scenario's. Dit houdt in dat de functionaliteit wordt getest zonder te kijken naar de interne werking van de applicatie. Er wordt alleen gekeken naar de invoer en de verwachte uitvoer.

De resultaten hiervan zijn vastgelegd in de testresultaten in [bijlage F](#). Als de test faalde, is het probleem vastgelegd in de issue lijst, te vinden in [bijlage G](#). Eventueel zijn er nog aanpassingen gemaakt om het wel correct te laten werken. Als de ontwikkelde functionaliteit voldeed aan de eisen, was het onderdeel succesvol geïmplementeerd.

Gedurende de implementatieperiode is wekelijks een vergadering gehouden met de bedrijfsbegeleiders om te rapporteren over de voortgang en eventuele problemen of wijzigingen. De bedrijfsbegeleiders vervulden de rol van scrummasters en product-owners. Er is dagelijks een standup meeting met hen gehouden, waarin kort werd beschreven welke activiteiten de vorige dag zijn uitgevoerd en welke activiteiten gepland waren voor die dag. De activiteiten waren gebaseerd op de per user story gedefinieerde taken in de backlog, te vinden in [bijlage C](#). Aan alle taken is met behulp van de MoSCoW methode (Brennan, 2009) een prioriteit toegekend. Ook is aan de taken een tijd toegevoegd van hoe lang wordt ingeschat er aan te werken.

Verder is elke week met de docentbegeleider gecommuniceerd over de voortgang en uitgevoerde activiteiten. Aan het eind van de sprint is ook de functionaliteit van het onderdeel gedemonstreerd aan de bedrijfsbegeleider en de opdrachtgever.

De planning voor de implementatieperiode is weergegeven in figuur 8. In het PID in [bijlage A](#) was eerst vastgelegd dat ook een modeler gemaakt zou worden voor de taken en rapporten. Gedurende het onderzoek is dit echter naar voren gekomen dat dit een minder belangrijk onderdeel was. Verder is de benodigde tijd voor de implementatie hoger gebleken. Daarom is dit onderdeel vervallen en is er gefocust op de lijst en formulier modeler.

[illegible]

Figuur 8. De planning voor de implementatieperiode. Voor elk van de onderdelen is drie weken ingedeeld.

7 Implementatieproces

De implementatie is ingedeeld in drie verschillende fases: de basis grafische modeler, de lijst modeler en de formulier modeler. Voor elke fase is hier het ontwikkelingsproces globaal vastgelegd. Een uitgebreidere en technischere beschrijving per onderdeel is te vinden in de technische beschrijving van het implementatieproces in [bijlage E](#).

7.1 Basis Grafische Modeler

De Software Factory GUI bestaat uit Microsoft C# code en wordt beheerd met behulp van het versiebeheersysteem SVN. Aan het begin van het implementatieproces, is in SVN een aparte branch aangemaakt voor het project van de grafische modeler. Dit zorgt ervoor dat de code geïsoleerd blijft van de rest van het project. (Collins-Sussman, Fitzpatrick, & Pilato, 2013) Het kan dan vervolgens later, wanneer het project succesvol is gerealiseerd, in de hoofdversie van de Software Factory GUI worden opgenomen.

Vervolgens is gekeken naar waar in de code aanpassingen moesten worden gemaakt en het technisch ontwerpen van hoe deze aanpassingen gemaakt moesten worden om de basis van de grafische modeler te implementeren. Door middel van toelichting van de bedrijfsbegeleiders en door analyse van de code, is onderzocht wat er uitgevoerd moet worden om het te implementeren.

De Software Factory is zelf ook een Software Factory project. Om de nieuwe modelers zichtbaar te maken binnen de Software Factory, zijn er aanpassingen gemaakt aan het project van de Software Factory. Hierdoor worden de benodigde schermen van de grafische modeler weergegeven.

Hierna is begonnen met de realisatie van de basis grafische modeler. Omdat de Software Factory GUI ontwikkeld is in Microsoft C#, wordt binnen Thinkwise Software voor de realisering gebruik gemaakt van Microsoft Visual Studio. Er is eerst een apart Microsoft C# project aangemaakt. Alle functionaliteit van de grafische modeler is hierin geïmplementeerd. Dit project wordt bij het uitvoeren van de Software Factory ingeladen om de nieuwe functionaliteit toe te voegen.

In dit project is begonnen met het zichtbaar maken van de benodigde schermen in de Software Factory. Voor de property lijst, de lijst modeler en de formulier modeler zijn schermelementen aangemaakt. Eerst is de functionaliteit van de property lijst ontwikkeld, zodat deze hetzelfde werkt als de huidige GUI modeler. Hierna is begonnen met het weergeven van de lijst en het formulier in de modelers zoals ze er uit zien in de eindapplicatie.

Eerst is geprobeerd om de lijst en het formulier te tonen door de informatie te halen uit het geopende project in de Software Factory. Het project bevatte hier echter niet alle benodigde informatie voor, zoals het weergeven van de lijst met de juiste stijl en het weergeven van de data in de lijst.

Daarom is na overleg besloten om direct verbinding te maken met de database, waar het model van de applicatie is opgeslagen. Dit wordt gedaan op dezelfde wijze als dat het model wordt ingeladen wanneer de applicatie normaal wordt opgestart. Hierdoor komen er geen extra problemen met de veiligheid.

De juiste informatie kan vervolgens uit de database worden gehaald van hoe de lijst en het formulier er in de eindapplicatie uit zien. Deze informatie wordt vervolgens gebruikt om in de lijst en de formulier modeler correct weer te geven hoe de lijst en het formulier er uit zien in de eindapplicatie. De gebruiker die een applicatie maakt met de Software Factory kan dan direct zien hoe het eindproduct er uit zal komen te zien.

Om een aantal wijzigbare instellingen voor de lijst en het formulier weer te geven, is een auto hide scherm toegevoegd waarin. Dit scherm verbergt zich automatisch aan de zijkant van het scherm, zodat het geen extra ruimte van de lijst of formulier modeler inneemt. Het is echter wel mogelijk om dit scherm constant zichtbaar te hebben.

Na de voltooiing van de realisatie van de basis grafische modeler, is de functionaliteit hiervan getest. De resultaten van het testen zijn vastgelegd in [bijlage F](#). De problemen die nog voorkomen in de basis grafische modeler zijn vastgelegd in de lijst met issues in [bijlage G](#). Ook is gedocumenteerd welke handelingen zijn uitgevoerd en hoe de geïmplementeerde functionaliteit werkt.

Aan het einde van deze fase is een demonstratie gegeven van het prototype. Hierin is de functionaliteit van de basis grafische modeler getoond. Deze demonstratie is uitgevoerd voor de opdrachtgever en de bedrijfsbegeleiders.

7.2 Lijst Modeler

Na de voltooiing van de basis grafische modeler, is begonnen met de implementatie van de lijst modeler. Ook hier is weer eerst begonnen met onderzoeken en technisch ontwerpen, om er achter te komen wat er gedaan moet worden en hoe de lijst modeler geïmplementeerd moet worden. Hiervoor is weer aanvullende informatie door de bedrijfsbegeleiders gegeven en verdere analyse van de bestaande code uitgevoerd.

De functionaliteit van de lijst modeler is weer geïmplementeerd in hetzelfde Microsoft C# project en in dezelfde model extender.

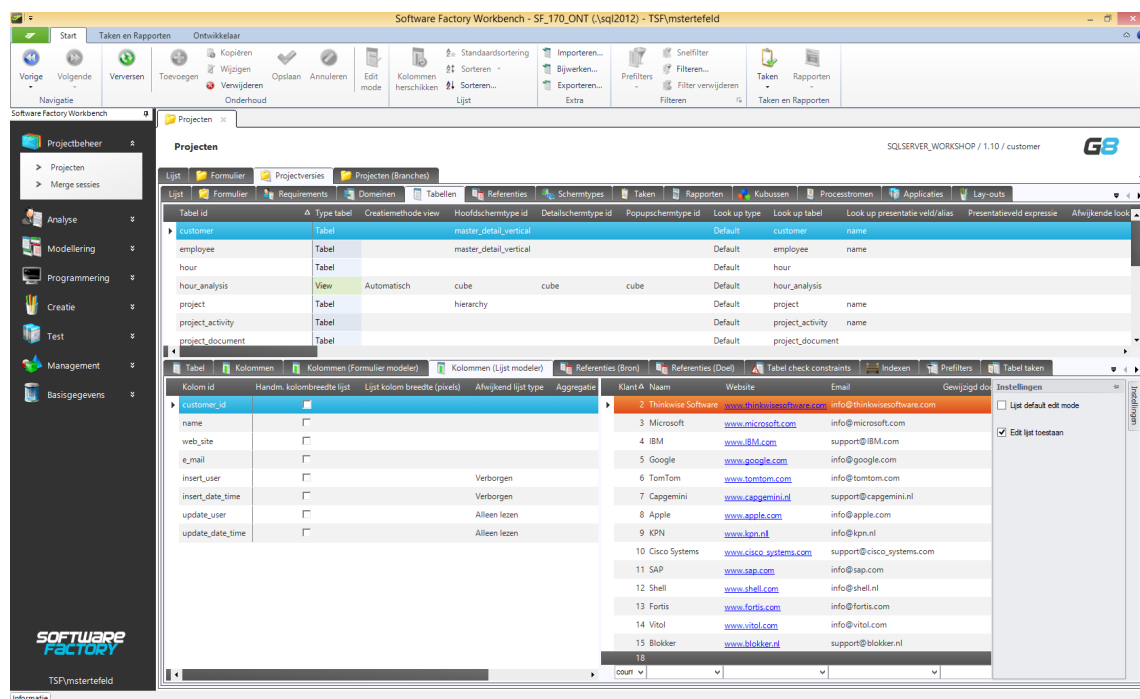
Eerst is er voor gezorgd dat wijzigingen die in de lijst modeler worden gemaakt aan de eindapplicatie, worden doorgevoerd naar het model. Hierdoor kunnen de aangepaste instellingen voor de applicatie worden opgeslagen in de database.

Vervolgens zijn de verschillende functionaliteiten toegevoegd aan de lijst modeler om de instellingen van de lijst grafisch aan te kunnen passen. Ook zijn aan het auto hide scherm de benodigde wijzigbare instellingen toegevoegd.

Nadat de implementatie van de lijst modeler is afgerond, is de functionaliteit weer getest, waarvan de resultaten te vinden zijn in [bijlage F](#). Ook zijn eventuele problemen met de lijst modeler vastgelegd in de lijst met issues in [bijlage G](#). Verder zijn de uitgevoerde handelingen weer gedocumenteerd en de toegevoegde functionaliteit vastgelegd.

Ook hier is aan het einde van deze ontwikkelingsfase een demonstratie gegeven aan de opdrachtgever en de bedrijfsbegeleiders, waar de nieuw ontwikkelde functionaliteit van de lijst modeler is getoond.

In figuur 9 is weergegeven hoe de lijst modeler er in de Software Factory uit ziet.

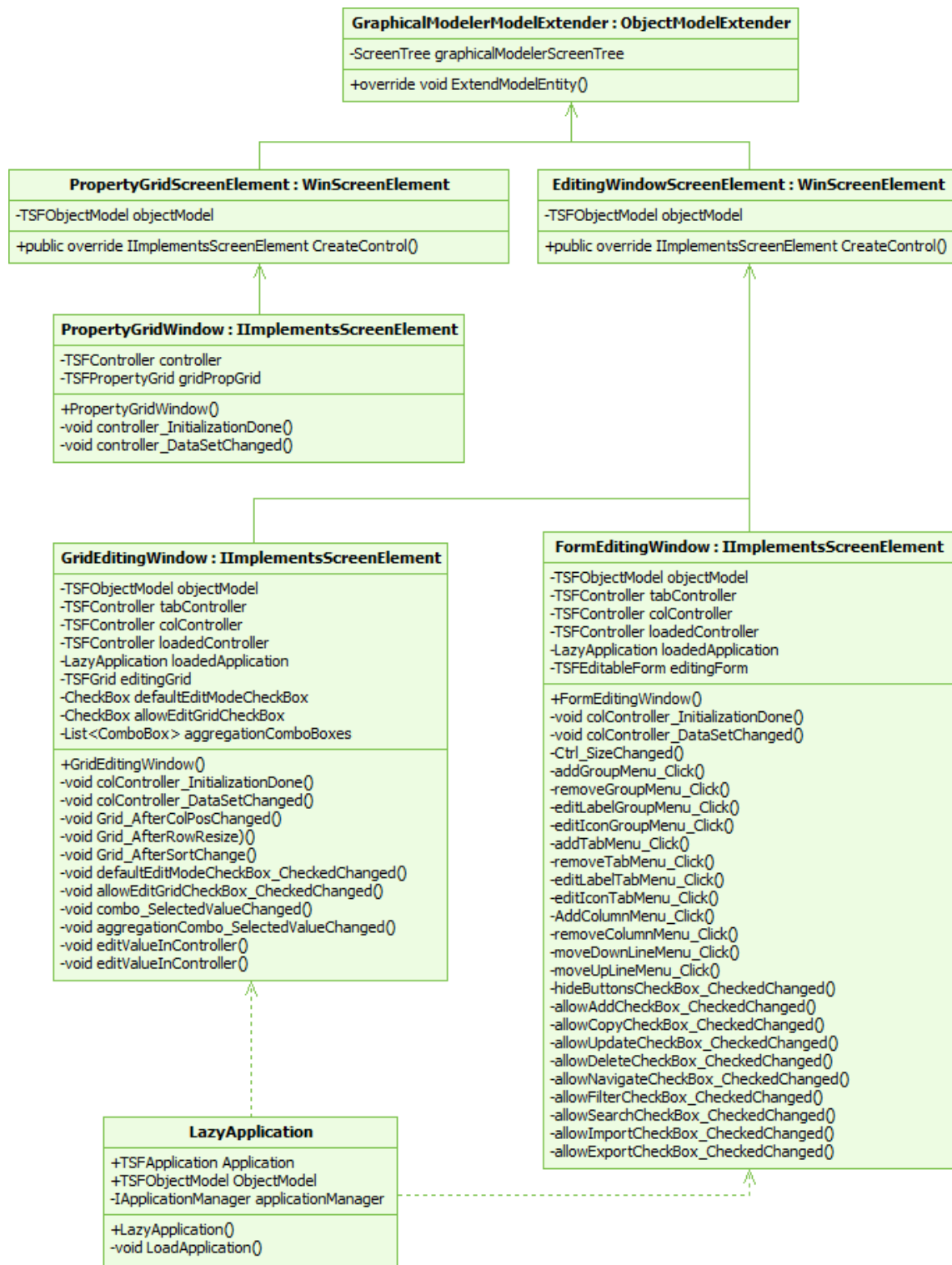


Figuur 9. De lijst modeler geïmplementeerd in de Software Factory. Aan de rechterkant is het auto hide scherm met instellingen te zien.

7.3 Formulier Modeler

Na het ontwikkelen van de lijst modeler, is het ontwikkelen van het laatste onderdeel van de grafische modeler gerealiseerd, de formulier modeler. Eerst is weer gestart met het onderzoek, om er achter te komen wat er uitgevoerd moet worden om de formulier modeler te implementeren. Dit is wederom gebeurd door code analyse en toelichting van de bedrijfsbegeleiders.

Hier is eerst is de code georganiseerd. Omdat alle onderdelen tot dusver in dezelfde klasse zat, met verschillende functionaliteiten, zorgde het voor een gebrek aan overzicht. De code is opgedeeld in verschillende onderdelen voor de model extender, de property lijst, de lijst modeler en de formulier modeler. Hierdoor blijft de code voor elk onderdeel geïsoleerd. Alle code bevindt zich nog wel binnen hetzelfde Microsoft C# project. In figuur 10 is het klassendiagram weergegeven waar de structuur van het project in zichtbaar is.



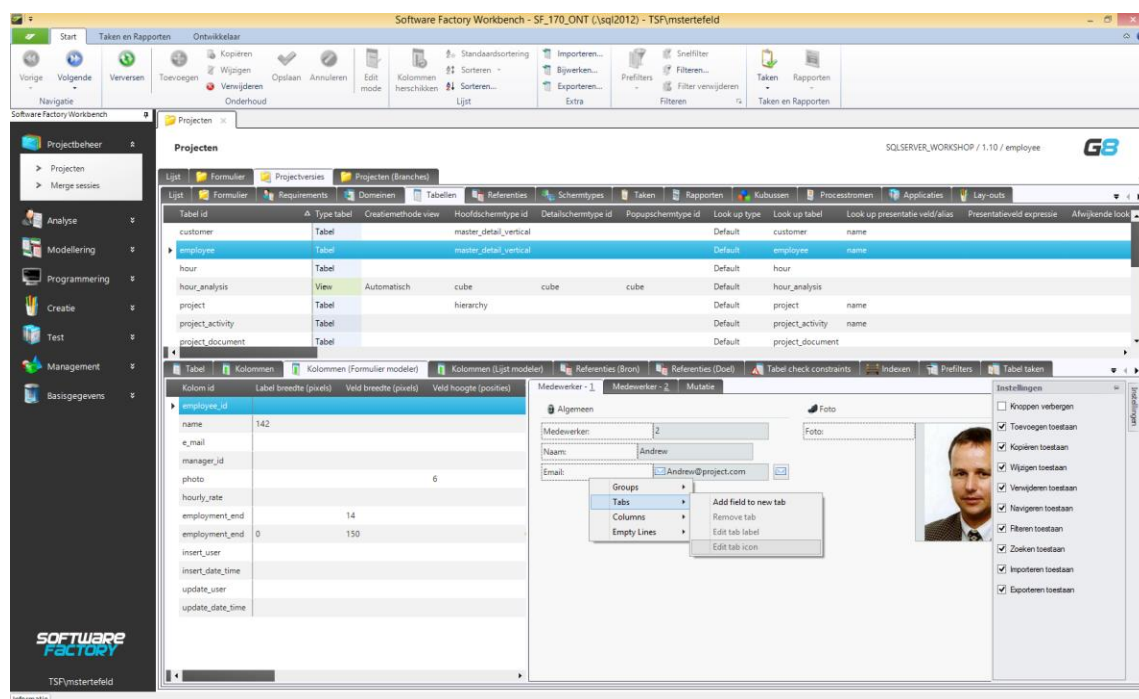
Figuur 10. Het klassendiagram van de Microsoft C# code van het project.

Hierna is de functionaliteit toegevoegd aan de formulier modeler om de verschillende instellingen van het formulier grafisch te kunnen wijzigen. Aan het auto hide scherm zijn ook weer de instellingen toegevoegd die voor het formulier wijzigbaar zijn.

Na de afronding van het implementeren van de formulier modeler, is de functionaliteit van de formulier modeler getest. De resultaten zijn vastgelegd in [bijlage F](#). De problemen er van zijn vastgelegd in de issue lijst in [bijlage G](#). Er is ook weer vastgelegd welke handelingen zijn uitgevoerd en hoe de toegevoegde functionaliteit werkt.

Aan het einde van deze fase is het resultaat weer gedemonstreerd aan de bedrijfsbegeleiders en de opdrachtgever om de functionaliteit van de formulier modeler te tonen.

In figuur 11 is te zien hoe de formulier modeler in de Software Factory er uit ziet.



Figuur 11. De formulier modeler geïmplementeerd in de Software Factory. Aan de rechterkant is het auto hide scherm met instellingen te zien. Ook is te zien hoe een tabblad aangepast kan worden.

8 Conclusie En Aanbevelingen

8.1 Conclusie

Het project van de grafische modeler is naar de eisen afgerond. Voor het project was gepland om een prototype te maken van de grafische modeler, waar het mee mogelijk is om de verschillende instellingen van de lijst en het formulier te kunnen aanpassen.

Door het onderzoek is gebleken wat de ideale manieren zijn om alle instellingen te wijzigen. Vervolgens is het grootste deel van de functionaliteit hiervan gerealiseerd. Er is een aantal wijzigingen gemaakt ten opzichte van het originele ontwerp, maar de gemaakte wijzigingen wijken niet drastisch af van het originele plan.

Uiteindelijk is er een functioneel prototype opgeleverd met de kernfunctionaliteit, waarmee de instellingen van de lijst en het formulier in de Software Factory grafisch kunnen worden aangepast. Thinkwise Software kan hierop verder bouwen om de huidige problemen en de nog missende functionaliteit te realiseren. Hiermee kan de grafische modeler dan worden afgerond en worden geïmplementeerd in de standaardversie van de Software Factory.

8.2 Aanbevelingen

Om de functionaliteit van de grafische modeler volledig te implementeren, zullen eerst alle issues moeten worden verholpen en de missende functionaliteiten worden geïmplementeerd, die zijn vastgelegd in [bijlage G](#). Zeker het kunnen aanpassen van de volgorde van velden op het formulier is een functionaliteit die bij de gebruikers die de Software Factory gebruiken om applicaties te maken, erg belangrijk bleek. Ook het weergeven van de grafische modeler op de juiste plaats, in het gebruikersinterface scherm, in plaats van in het project scherm, is belangrijk om er voor te zorgen dat gebruikers de functionaliteit gemakkelijk kunnen vinden. Deze onderdelen zijn tijdens het project niet gerealiseerd omdat de functionaliteit te complex bleek om te implementeren in de beschikbare tijdsperiode.

Tijdens het ontwikkelen van de grafische modeler is duidelijk geworden dat op enkele plaatsen de documentatie van de Software Factory GUI gebrekkig is. Voor de code mist commentaar om de toepassing van verschillende onderdelen toe te lichten. Ook de informatie over de structuur van de code van de Software Factory GUI schiet soms tekort. Hierdoor wordt het ontwikkelen van toevoegingen aan de Software Factory lastig. Er moet extra tijd worden besteed aan onderzoek naar de Software Factory zelf. Daarom kan de documentatie van de Software Factory GUI nog sterk worden verbeterd door Thinkwise Software.

Voor dit project is door middel van commentaar in de code en de documentatie in deze scriptie de functionaliteit vastgelegd. Voor toekomstige projecten, zeker voor andere stagiaires en afstudeerders, zal het erg nuttig zijn om ook voor de rest van de Software Factory GUI goede documentatie vast te leggen. Dit zorgt er voor dat ze minder tijd hoeven te besteden aan het onderzoeken van de structuur en werking. Hierdoor hebben ze meer tijd om te besteden aan de daadwerkelijke implementatie van projecten.

Evaluatie

Ervaringen

Het werken bij Thinkwise Software en specifiek binnen het Product Innovation team heeft voor mij voor goede ervaringen gezorgd. Het is een bedrijf met een zeer innovatief en uniek product. Het concept van model gedreven software wordt door slechts weinig bedrijven gebruikt.

Thinkwise Software is een professioneel en zakelijk bedrijf. In vergelijking met het los en informeel werken op school, is dit heel anders. Het heeft mij duidelijk gemaakt wat ik kan verwachten van het werken na de opleiding. Verder was het interessant en leerzaam om te werken met de collega's binnen het Product Innovation team. Alle collega's beschikten over uitgebreide kennis van hun product en hun werk.

Aan het begin van het afstuderen heb ik bij Thinkwise Software een workshop gevolgd over de Software Factory. Hierin is het gebruik van de Software Factory uitgelegd. Hierdoor heb ik inzicht gekregen in hoe de Software Factory werkt en hoe het gebruikt moet worden. Dit heeft geholpen met het toevoegen van de nieuwe functionaliteit aan de Software Factory.

Het uitvoeren van professioneel onderzoek is iets waar tijdens de opleiding nog niet veel aandacht aan is besteed. Doordat dit tijdens het afstudeerproject is toegepast, heb ik er veel over geleerd. Het onderzoeken was erg interessant en zorgde voor veel relevante informatie, die vervolgens weer nuttig was voor het ontwerpen van het geïmplementeerde systeem.

Aan het begin van het afstuderen heb ik in het persoonlijke ontwikkelplan aangegeven aan welke ontwikkelpunten ik zou werken. Het eerste punt hierin is het plannen van een agile project. Het werken aan dit project heeft hier aan bijgedragen. Tijdens het project is het werk ingedeeld in een backlog van taken. Dit heeft ervoor gezorgd dat het werk goed verdeeld kon worden over de ontwikkelingsperiode. Tijdens de implementatieperiode is overgegaan van wekelijkse meetings naar daily standup meetings, om de planning, communicatie en het overzicht voor de begeleiders en de afstudeerder te verbeteren. Het plannen van de implementatie heeft ervoor gezorgd dat ik nauwkeuriger de hoeveelheid werk van bepaalde werkzaamheden kan inschatten, om toekomstige projecten ook vlot te kunnen laten verlopen.

In het ontwikkelplan is ook het ontwikkelen van een uitbreiding van een bestaande applicatie met professionele codebase aangegeven. Dit is uitstekend van toepassing geweest op het afstudeerproject. De implementatie van de grafische modeler was een uitbreiding op het bestaande systeem. Alhoewel in een eigen project is gewerkt, heb ik genoeg interactie gehad met het bestaande systeem. Hierdoor heb ik extra inzichten verkregen in hoe professionele applicaties zijn gestructureerd en hoe er wijzigingen aan worden gemaakt.

Ten slotte is in het persoonlijke ontwikkel plan het volledig documenteren van een project aangegeven. Omdat dit een afstudeerproject is, moet er erg veel worden gedocumenteerd over wat er precies tijdens het project is uitgevoerd. Omdat het een uitbreiding is van een bestaande applicatie, die later zal worden gebruikt door het bedrijf, is ook de technische documentatie belangrijk geweest. Documenteren is een kernvaardigheid van een software ontwikkelaar. Door voor dit project uitgebreid te hebben gedocumenteerd heb ik een belangrijke vaardigheid verder ontwikkeld.

Leerpunten

Een punt waar ik nog aandacht aan kan besteden is het samenwerken met collega's. Omdat de afstudeerstage een individueel project is, wordt er snel geneigd om veel dingen zelf te doen. Echter, bij problemen of tegenslagen is het raadplegen van collega's of begeleiders geen probleem. In het afstudeerproject zou ik wel vaker en sneller op bepaalde punten van deze ondersteuning gebruik kunnen maken.

Competenties

Tijdens het afstudeerproject zijn de verschillende competenties toegepast. In [bijlage H](#) is voor elke competentie aangegeven hoe deze aan bod zijn gekomen tijdens het afstudeerproject.

Referenties

- Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis. Retrieved 6 6, 2014
- Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. (2013). *Version Control With Subversion*. O'Reilly Media. Retrieved 6 6, 2014, from <http://svnbook.red-bean.com/en/1.7/svn.branchmerge.whatis.html>
- Kempen, P., & Keizer, J. (2011). *Competent afstuderen en stagelopen*. Noordhoff Uitgevers B.V. Retrieved 6 6, 2014
- Schwaber, K., & Sutherland, J. (2013, juli). *The Scrum Guide™*. Retrieved 6 6, 2014, from The home of Scrum: <https://www.scrum.org/Portals/0/Documents/Scrum Guides/2013/Scrum-Guide.pdf>
- Thinkwise Software. (2013, April 2). *Grafische User Interface Modeler*. Retrieved from Thinkwise Software: <http://www.thinkwisesoftware.com/wat-is-het/>

Bijlages

Bijlage A: Product Initiation Document

Hierin wordt het project beschreven en toegelicht waarom en hoe het uitgevoerd zal worden. Ook is de planning van het volledige afstudeerproject hierin vastgelegd.

Bijlage B: Onderzoeksverslag

In het onderzoeksverslag is vastgelegd hoe het onderzoek is uitgevoerd en welke resultaten het heeft opgeleverd.

Bijlage C: User Stories

In de user stories zijn de geplande werkzaamheden ingedeeld in user stories met bijbehorende taken.

Bijlage D: Use Cases

De use cases beschrijven de functionaliteit van de verschillende onderdelen van het project. Deze worden gebruikt om te controleren of het product voldoet aan de eisen.

Bijlage E: Technische Beschrijving Implementatieproces

Hierin is een uitgebreide en technische beschrijving gegeven van de uitgevoerde werkzaamheden van het implementatieproces.

Bijlage F: Testresultaten

De resultaten van de uitgevoerde tests zijn hier vastgelegd.

Bijlage G: Issue Lijst

De issue lijst bevat een overzicht van welke problemen en missende functionaliteiten nog in de implementatie aanwezig zijn.

Bijlage H: Competenties

In de competenties is vastgelegd hoe de verschillende competenties aan bod zijn gekomen tijdens het afstudeerproject.

Bijlage A: Project Initiation Document
Grafische GUI Modeler

Project Initiation Document



Project	Grafische GUI Modeler
Datum	12-3-2014
Auteur	Mathijs Stertefeld
Versie	1.30
Status	Voltooid

Inhoud

1	Inleiding	4
2	Achtergrond	5
2.1	Bedrijf.....	5
2.2	Opdracht	6
3	Projectdefinitie	9
3.1	Projectdoelstellingen	9
3.2	Onderzoek.....	9
3.3	Gekozen Oplossing.....	10
3.4	Scope van het Project	10
3.5	Project Decompositie Structuur (PDS) op Projectniveau	11
3.6	Productbeschrijving op Projectniveau.....	11
3.7	Projectbudget	11
3.8	Projectmanagement	12
3.9	Wijzigingsbudget.....	12
3.10	Benodigde Resources.....	12
3.11	Uitsluitingen.....	12
3.12	Afhankelijkheden	12
3.13	Randvoorwaarden	12
3.14	Aannames	12
4	Onderzoeksontwerp	13
4.1	Probleemstelling.....	13
4.2	Vooronderzoek	13
4.3	Doelstelling	13
4.4	Onderzoeksvragen	13
4.5	Gegevensverzameling.....	14
4.6	Onderzoeksplanning	14
4.7	Bronnen	14
5	Project Organisatie Structuur	15
5.1	Projectorganisatie.....	15
5.2	Opdrachtgevers	15
5.3	Bedrijfsbegeleiders	15
5.4	Product Innovation Team	15
5.5	Fontys Docentbegeleider.....	16
5.6	Projectontwikkelaar.....	16
5.7	Communicatie.....	16
5.8	Project Regels	16
6	Project Beheersing.....	17
6.1	Rapportage en Voortgangsbewaking	17
6.2	Toleranties	17
6.3	Risicomanagement	17
6.4	Kwaliteitsbewaking.....	17
6.5	Wijzigingsprocedure	17

6.6 Escalatieprocedure	18
Referenties.....	19
Bijlage A: Communicatieplan.....	20
Bijlage B: Acceptatie Criteria	22
Bijlage C: Business Case	23
Bijlage D: Product Beschrijving	25
Bijlage E: Planning.....	27

1 Inleiding

Doel van dit Document

Dit is het Project Initiation Document (PID) van de afstudeerstage bij Thinkwise Software. Hierin worden de doelen van het project beschreven en wordt beschreven wat er nodig is om deze doelen te bereiken. Ook wordt aangegeven waarom het nodig is om deze doelen te bereiken.

Dit PID bevat een beschrijving van wat er gerealiseerd zal worden binnen het project. Ook beschrijft het de management van het project, wie er betrokken zijn bij het project en ieders rol hierbinnen.

Dit document wordt gebruikt om te garanderen dat het project een verantwoorde en correcte basis heeft, voordat de ontwikkeling van het daadwerkelijke project wordt gestart.

Het document dient ook ter referentie, waarmee de betrokken leden van het project de voortgang en wijzigingen die gemaakt worden tijdens de ontwikkeling van het project, kunnen controleren.

Redenen

De reden dat de afstudeerder bij Thinkwise Software werkzaam is, is voor het afstuderen voor de studie bij de Fontys. Thinkwise Software is de ontwikkelaar van de Software Factory, waarmee eenvoudig business software kan worden gemaakt. Het ontwerpen van de GUI's hiervoor werkt met het huidige systeem niet optimaal. De aanleiding van de opdracht is het verbeteren van dit proces. De verbetering van het systeem zorgt ervoor dat het intuïtiever is voor de gebruiker en de workflow verbeterd wordt.

In bijlage C is de Businesscase te vinden waarin de aanleiding van het project verder wordt toegelicht.

Globale Aanpak

Het project wordt uitgevoerd door één enkele afstudeerder. De afstudeerder krijgt begeleiding van verschillende collega's binnen het bedrijf. Extra ondersteuning is beschikbaar indien nodig. De afstudeerder communiceert ook regelmatig met de school, om de voortgang en status aan te tonen.

Globale Kosten en Tijd

Het afstuderen is ingepland voor minimaal 85 dagen. Elke dag bevat 8 werkuren. Dit komt neer op een totaal van 680 manuren. Het project is verdeeld in verschillende fases, voor de verschillende onderdelen die aan bod komen. In Bijlage E is een overzicht van de verschillende fases en de indeling hiervan.

2 Achtergrond

2.1 Bedrijf

“Thinkwise Software is een specialist in computergestuurde softwarebouw en uitvinder van de zeer innovatieve Software Factory. De Thinkwise Software Factory Suite is een geïntegreerde ontwikkelomgeving waarmee eenvoudig business software kan worden gebouwd, getest, gedeployed en beheerd. Het is zelfs mogelijk bestaande software automatisch te moderniseren.

De Thinkwise omgeving is vooral geschikt voor het bouwen van business software, zoals ERP applicaties, randsystemen, research applicaties en apps. Dit type software kan bestaan uit workflow, gebruikersinterface in Windows, Web of mobiele apparaten, applicatielogica/services en database/data services.

De suite bestaat uit de volgende onderdelen:

Software Factory

Hiermee bouwt en test u uw oplossing. U kunt de requirements vastleggen en met de business bespreken. Aan de hand van de requirements wordt een model gemaakt, bestaande uit workflow, procesmodel, datamodel, GUI model en applicatielogica. De modellen kunt u volledig grafisch valideren en analyseren.

UpCycler

Om bestaande applicaties te moderniseren kunt u de UpCycler gebruiken. Met behulp van de UpCycler wordt een model van uw bestaande applicatie afgeleid en in de Software Factory opgeslagen. Dit is de eerste versie van uw splinternieuwe applicatie die u vervolgens weer kunt doorontwikkelen. U houdt wat goed is en verbetert waar nodig.

Abstracte GUI's

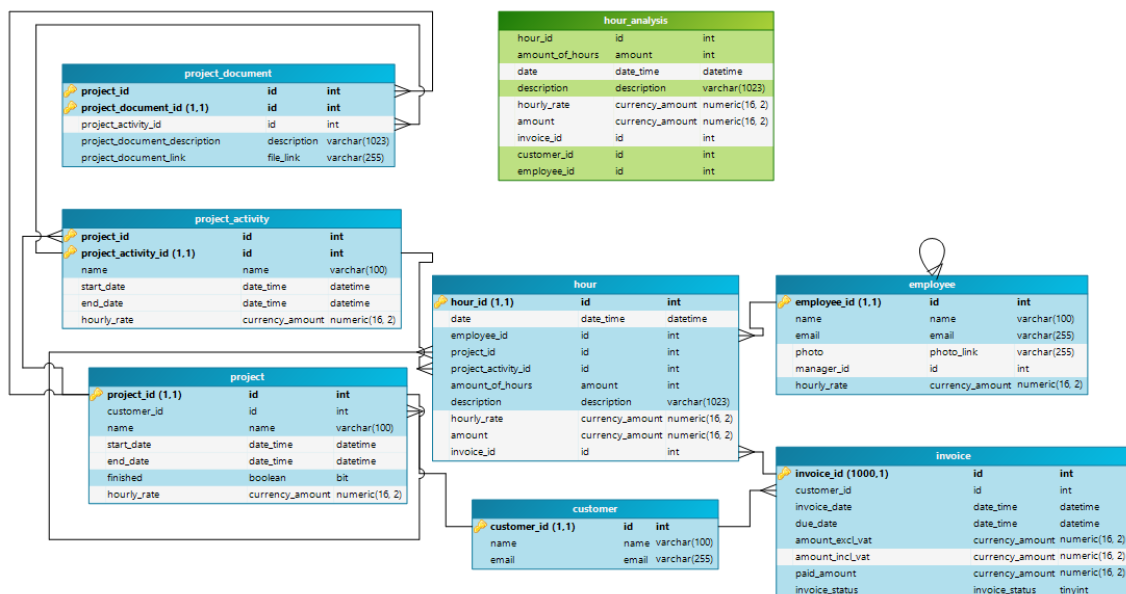
De gebruikersinterfaces (GUI's) worden door Thinkwise Software zowel in Windows en Web beschikbaar gesteld. Door middel van runtime interpretatie anticiperen de GUI's voor 100% op uw unieke model. Elke wijziging en uitbreiding van het model is direct in beide GUI's beschikbaar. Zodra er nieuwe technieken beschikbaar komen, verwerkt Thinkwise Software deze in haar abstracte GUI's. U hoeft hier dus geen tijd aan te besteden en bent altijd up-to-date.

Intelligent Application Manager (IAM)

Hiermee kunt u de applicaties in productie beheren. Uiteraard kunt u de autorisatie inrichten en desgewenst koppelen aan Active Directory. Daarnaast kan elke gebruiker via de GUI zijn voorkeuren vastleggen in de IAM. Ten slotte kan het gebruik worden geanalyseerd, waardoor de GUI's zelflerend worden.” (Thinkwise Software, 2013)

2.2 Opdracht

Bij het maken van een applicatie met de Software Factory wordt er eerst een datamodel opgesteld van de gewenste applicatie. Hierin wordt aangegeven welke tabellen en views er in de eindapplicatie zijn en welke velden er zijn per tabel. Ook worden de relaties tussen de verschillende tabellen aangegeven. Dit is te zien in Figuur 1.



Figuur 1. Een datamodel in de Software Factory

Voor de tabellen in het datamodel zijn er schermen in de eindapplicatie, om de data in te zien en te bewerken. De Software Factory genereert een model van de eindapplicatie. De abstracte GUI, die voor elke applicatie hetzelfde is, interpreteert dit model en bouwt zichzelf aan de hand van dit model op. De ontwikkelaar heeft echter nog wel de mogelijkheid om in te stellen hoe de schermen er uit zullen zien.

Om gebruik te maken van de data in de tabellen in de eindapplicatie, zijn er twee belangrijke vensters. Het eerste venster wordt de “Lijst” weergave genoemd. Hierin wordt in een lijst de velden van de tabel weergegeven. Vervolgens is er de “Formulier” weergave. Hierin wordt de informatie van één specifieke record in de tabel weergegeven in een aantal velden. Ook zijn er hier knoppen om verschillende acties uit te voeren, zoals het aanmaken, wijzigen of verwijderen van records. Beide vensters zijn te zien in Figuur 2.

Date	Employee ID	Project id	Project activity id	Amou	Description	Hourly rate	Amount	Invoice ID
02/06/2014	David Davidse	IKEA Kaste	Kast in elkaar zette	5	Kast bouwen	60.00	300.00	1002
02/03/2014	Jan Jansen	Project On	Do project	10	Make project	250.00	2500.00	1001
01/05/2014	Petra Petruse	IKEA Kaste	Kast in elkaar zette	2	Kast bouwen	30.00	60.00	1002
01/26/2014	Stephen Haw	Project Tw	Design project	1	Design project	250.00	250.00	1000
01/27/2014	Stephen Haw	Project Tw	Design project	2	Design project	250.00	500.00	1000
02/03/2014	Ella Alle	Project On	Do project	1	Make project	20.00	20.00	1001

Hour - 1

Hour - 2

Date: 1/26/2014
Employee ID: Stephen Hawking
Project id: Project Two
Project activity id: Design project
Amount of hours: 1

Figuur 2. Een Lijst en een Formulier van een bepaalde tabel.

Deze vensters worden met een standaard layout gegenereerd. Het is echter ook mogelijk om deze aan te passen. Aan de Lijst en aan het Formulier kunnen verschillende dingen worden aangepast.

Voor een Lijst kunnen de standaard kolombreedte worden aangepast, het type van de kolom binnen de Lijst en de mogelijkheid om verschillende soorten van aggregatie voor de lijst te tonen. Deze instellingen zijn te zien in Figuur 3.

Tabel	Kolommen	Lijst	Formulier	Lookup tabellen	Detailtabbladen	Zoeken / Filteren	Sorteren
Kolom id	Handm. kolombreedte lijst	Lijst kolom breedte (pixels)	Afwijkend lijst type	Aggregatie in lijst tonen	Aggregatie type		
hour_id	<input checked="" type="checkbox"/>	25	Verborgen	<input type="checkbox"/>			
date	<input checked="" type="checkbox"/>	50	Regulier	<input type="checkbox"/>			
employee_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>			
project_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>			
project_activity_id	<input type="checkbox"/>		Regulier	<input type="checkbox"/>			
amount_of_hours	<input type="checkbox"/>		Regulier	<input checked="" type="checkbox"/>	Sommeren		
description	<input type="checkbox"/>		Regulier	<input type="checkbox"/>			
hourly_rate	<input checked="" type="checkbox"/>	50	Regulier	<input type="checkbox"/>			
amount	<input checked="" type="checkbox"/>	50	Alleen lezen	<input checked="" type="checkbox"/>	Sommeren		
invoice_id	<input checked="" type="checkbox"/>	25	Regulier	<input type="checkbox"/>			

Figuur 3. Het wijzigen van de GUI van de Lijst weergave van een tabel.

Voor een Formulier kunnen nog meer waardes in worden gesteld. Zo kunnen de breedte en hoogte van velden en diens labels worden ingesteld. Groepen van verschillende velden worden gemaakt met hun eigen labels en iconen. Ook kan worden ingesteld op welk tabblad een waarde staat. Ten slotte kan ook nog het type van een veld worden aangepast. Deze instellingen zijn te zien in Figuur 4.

Kolom id	Label breedte (pix)	Veld breedte (pix)	Veld hoogte (pos)	Veld aantal pos	Veld in volgende groep	Label	Volgende groep icoon	Veld in volgende kol	Veld op volgende tab	Label	Volgende tab ico	Afwijkend formulier type
hour_id	25	10		1	<input checked="" type="checkbox"/>	Uren		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Verborgen
date				1	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			Regulier
employee_id				1	<input checked="" type="checkbox"/>	info		<input type="checkbox"/>	<input type="checkbox"/>			
project_id				1	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
project_activity_id	40	10		0	<input type="checkbox"/>			<input checked="" type="checkbox"/>	<input type="checkbox"/>			
amount_of_hours	40			1	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
description		100	3	2	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			
hourly_rate				1	<input checked="" type="checkbox"/>	Kosten		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Factuu	D:\Software_fabri	
amount				1	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>			Alleen lezen
invoice_id		10		2	<input type="checkbox"/>			<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Figuur 4. Het wijzigen van de Formulier weergave van een tabel.

Na het aanpassen van deze waardes moet steeds in de eindapplicatie het model worden ververst om te kijken hoe de wijzigingen er uit zullen zien. Dit zorgt voor erg veel testwerk bij het maken van een GUI, waarin de waardes steeds worden veranderd en dan in het programma moet worden gekeken hoe de interface er daadwerkelijk uit ziet.

Om de gebruikersinterface dus op een gebruiksvriendelijke manier te kunnen modeleren zal dit grafisch moeten gebeuren. Hierbij moet gebruik worden gemaakt van “What-You-See-Is-What-You-Get” en “Drag and Drop” functionaliteit om schermen in te richten. Dit zorgt voor makkelijker en sneller ontwikkelen van eindapplicaties.

De opdracht bestaat uit het onderzoeken naar wat de mogelijkheden zijn voor GUI modeling en het ontwerpen en ontwikkelen van een prototype van een GUI modeler, waarmee volgorde, zichtbaarheid, grootte van velden, indeling in groepen en tabbladen, stijlen en afbeeldingen aangepast kunnen worden op grafische wijze. Hierbij is het ook van belang dat de technische documentatie in orde is, zodat er na het project er aan kan worden doorontwikkeld.

3 Projectdefinitie

3.1 Projectdoelstellingen

In de huidige versie van de Software Factory kan bij het maken van een applicatie worden ingesteld hoe de schermen er uit komen te zien door middel van het invoeren van een aantal waardes in een formulier. Om er vervolgens achter te komen hoe het eindresultaat er daadwerkelijk uit ziet, moet dit bekeken worden in de applicatie. Dit zorgt voor extra tijdsbesteding aan het controleren of de interface voldoet aan de eisen van de ontwikkelaar.

Het nieuwe systeem zorgt ervoor dat ontwikkelaars efficiënter en sneller applicaties kunnen maken. Ze kunnen gemakkelijker instellen hoe de vensters zijn ingedeeld en zien direct hoe het eindresultaat er uit zal komen te zien, zonder de applicatie te hoeven draaien. Dit zorgt voor een verduidelijking en versimpeling van het gehele proces.

3.2 Onderzoek

Voor het project zal een onderzoek worden uitgevoerd om er achter te komen welke functionaliteiten in de huidige GUI modeler zitten en wat de mogelijkheden zijn om deze op een grafische te implementeren. De hoofdvraag van het onderzoek is dus als volgt: *“Welke functionaliteiten bevat de huidige GUI modeler en hoe kunnen deze worden geïmplementeerd in een grafische modeler?”*

Deze hoofdvraag kan worden beantwoord door de deelvragen te beantwoorden, welke als volgt zijn:

- Hoe wordt momenteel gebruik gemaakt van de huidige GUI modeler?
- Welke onderdelen kunnen met de huidige GUI modeler aangepast worden?
- Welke onderdelen zijn wel of niet geschikt om met de grafische modeler aan te passen?
- Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen?

Om deze vragen te beantwoorden, wordt er op verschillende manieren onderzocht. Eerst wordt gekeken naar de functionaliteiten van huidige GUI modeler. Aan de hand hiervan worden ontwerpen opgesteld voor hoe deze functionaliteiten uitgevoerd kunnen worden in een grafische modeler. Vervolgens worden interviews uitgevoerd met gebruikers van de huidige GUI modeler, om extra informatie te winnen en feedback te krijgen op de ontwerpen. De ontwerpen worden vervolgens aangepast.

Voor de ontwerpen wordt dan gekeken in hoeverre het technische mogelijk is om de ontwerpen te realiseren. Indien nodig kunnen dan nog aanpassingen worden gemaakt aan de ontwerpen. Om de technische haalbaarheid te onderzoeken worden er enkele prototypes ontwikkeld.

Een uitgebreid ontwerp van het onderzoek is beschreven in hoofdstuk 4.

Met de kennis opgedaan uit het onderzoek en de gemaakte prototypes wordt de basis gelegd voor het eindproduct zelf.

3.3 Gekozen Oplossing

Voor dit project is er gekozen om het instellen van de layout van de schermen grafisch te doen. Hierbij wordt gebruik gemaakt van “What-You-See-Is-What-You-Get” en “Drag & Drop”. Hiermee kunnen volgorde, zichtbaarheid, grootte van velden, indeling in groepen en tabbladen, stijlen en afbeeldingen worden aangepast op grafische wijze.

Door het aanpassen van de layout grafisch te maken, wordt de gebruiksvriendelijkheid sterk verbeterd. Zo wordt het toegankelijker voor meer mensen om applicaties met aangepaste layouts te ontwikkelen. Omdat het ontwikkelen intuïtief en eenvoudiger gemaakt wordt, kunnen ook mensen met minder technische kennis gebruik make van de grafische modeler.

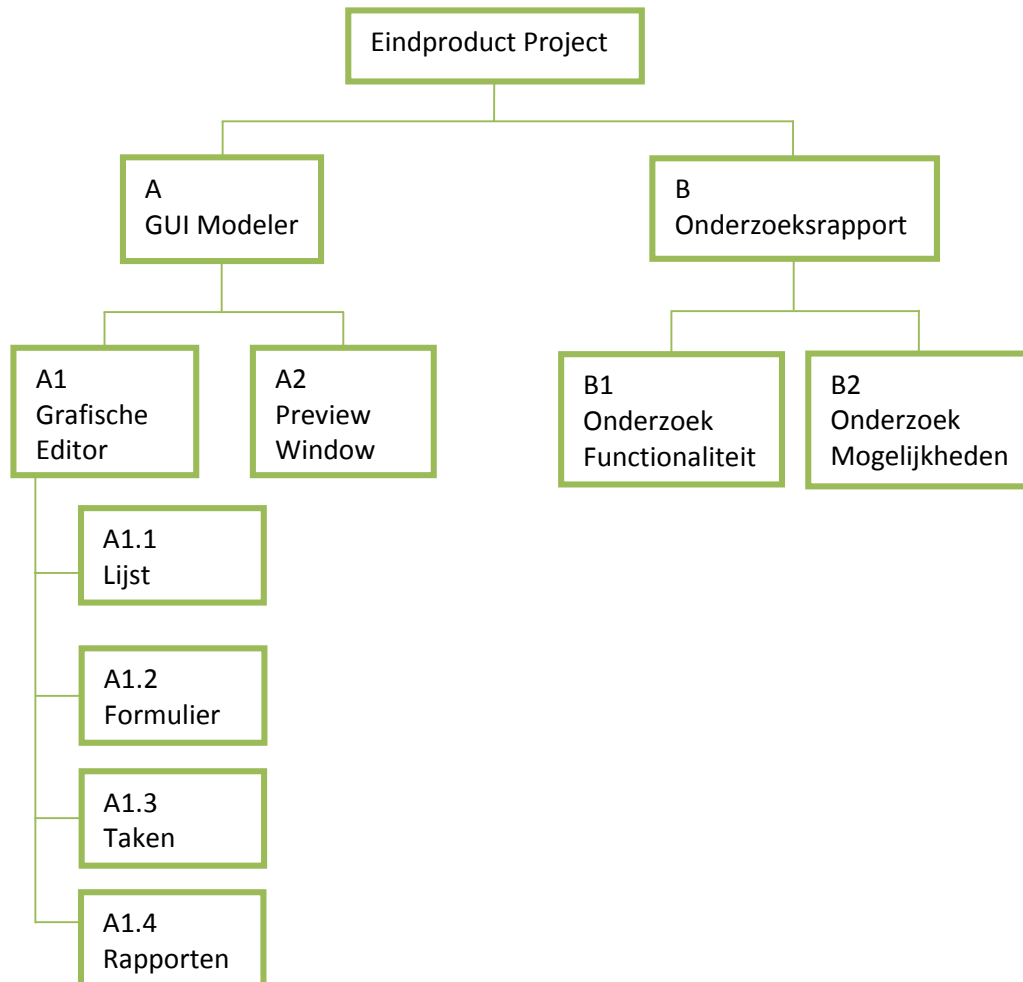
3.4 Scope van het Project

Bij dit project zijn er verschillende mensen betrokken. De originele ontwikkelaars van de Software Factory, de leden van het Product Innovation team, zorgen voor zowel informatie als begeleiding. Verder zal de opdrachtgever beslissen of het plan voor het project in overeenkomst is met de visie voor het project.

Voor het project wordt onderzoek gedaan naar wat een mogelijke geschikte grafische modeler is. De inhoud van dit onderzoek is verder beschreven in bijlage D. Vervolgens wordt hiervoor een prototype ontwikkeld. Met dit prototype moet het mogelijk zijn om de layout van applicaties grafisch te kunnen aanpassen, zonder verlies van de functionaliteit die in de huidige GUI modeler zit. Hierbij wordt vooral gefocust op de Lijst en de Formulier modeler. De Taken en Rapporten modelers zijn ook gepland, maar zijn minder essentieel. Ten slotte kan bij een overschot van tijd nog worden gekeken naar het aanpassen van de Stijl van de verschillende schermen met behulp van de grafische modeler. Er zijn geen eisen aan het toevoegen van extra functies die in de huidige GUI modeler nog niet mogelijk zijn.

3.5 Project Decompositie Structuur (PDS) op Projectniveau

In figuur 5 is de Project Decompositie Structuur weergegeven. Hierin worden de verschillende onderdelen van het project weergegeven.



Figuur 5. De Project Decompositie Structuur van het project.

3.6 Productbeschrijving op Projectniveau

Zie Bijlage D voor de volledige Product Beschrijving.

3.7 Projectbudget

De afstudeerstage duurt 85 dagen. Elke dag wordt 8 uur besteed aan het werken aan het project. Dit zorgt voor een totaal van 680 uur om te besteden aan het project. De beschikbare tijd is opgedeeld in een aantal verschillende fases, waarin aan specifieke onderdelen van het project wordt gewerkt.

Zie bijlage E voor een volledig overzicht van de planning.

3.8 Projectmanagement

Voor het project wordt gebruik gemaakt van de SCRUM ontwikkelmethode (Schwaber & Sutherland, 2013) om agile te kunnen ontwikkelen. Hier wordt continu gerapporteerd over de activiteiten en gemaakte keuzes, waardoor de begeleiders ook overzicht houden over het project. Voor elke fase worden een aantal sprints ingedeeld. In deze sprints wordt steeds aan een aantal taken van bepaalde onderdelen gewerkt.

3.9 Wijzigingsbudget

Wanneer er wijzigingen worden gemaakt tijdens het project die de lengte van de verschillende projectonderdelen beïnvloeden, zullen er dusdanige aanpassingen gemaakt worden aan het project om te garanderen dat het project binnen de gegeven tijd kan worden afgerond.

Er is een doorlooptijd van twee weken die gebruikt kan worden om het project af te ronden in het geval dat het project nog niet is afgerond binnen de gegeven tijd.

3.10 Benodigde Resources

Er zijn niet veel middelen nodig voor het project. De afstudeerder werkt alleen aan het project, maar rapporteert wel aan de begeleiders over de voortgang van het project. De volledige werktijd is ingepland aan het werken aan het project en aan de zaken gerelateerd aan het project. Er wordt gebruik gemaakt van één enkele werkplek met computer. De software waar gebruik van wordt gemaakt is:

- Microsoft Visual Studio 2012 voor het ontwikkelen van de code.
- Microsoft Office 2013 voor de documentatie.
- TortoiseSVN 1.8 voor versiebeheer.
- Thinkwise Software Factory zelf om te testen.

3.11 Uitsluitingen

Het project wordt uitgevoerd in een aparte branch van de code. Hierin wordt alleen gewerkt aan de GUI modeler van de Software Factory. Aan de rest van de Software Factory wordt niet gewerkt. Verder wordt er slechts een prototype ontwikkeld. Het is mogelijk dat nog niet alle functionaliteit hier in ontwikkeld is.

3.12 Afhankelijkheden

Het project is afhankelijk van de huidige versie van de Software Factory, vooral het onderdeel van de GUI modeler. Daarom is het afhankelijk van de originele ontwikkelaars van de Software Factory, het Product Innovation team.

3.13 Randvoorwaarden

De Software Factory heeft documentatie, zodat ontwikkeling van een grafische GUI modeler mogelijk is zonder de volledige code door te hoeven lezen.

3.14 Aannames

De huidige GUI modeler is al functioneel en gedocumenteerd tot een niveau dat niet de volledige code geïnspecteerd hoeft te worden om de grafische modeler te implementeren.

4 Onderzoeksontwerp

4.1 Probleemstelling

Momenteel, wanneer een applicatie wordt gemaakt met de Software Factory, kan worden ingesteld hoe de schermen er uit komen te zien, door middel van het invoeren van aantal waarden in een formulier. Om vervolgens er achter te komen hoe het eindresultaat er ook daadwerkelijk uit ziet, moet dit bekeken worden in de applicatie. Dit zorgt dat er extra tijd moet worden besteed aan het controleren of de interface voldoet aan de eisen van de ontwikkelaar.

Het onderzoek vindt plaats om er achter te komen welke functionaliteiten in de huidige applicatie zitten, en wat de mogelijkheden zijn om deze op een grafische wijze uit te voeren.

4.2 Vooronderzoek

Voor het begin van het onderzoek is al enige kennis opgedaan over de huidige GUI modeler. Door het ontwikkelen van het Project Initiation Document is al duidelijk geworden welke eisen er zijn aan het project. Ook heeft de initiële workshop over de Software Factory grotendeels uitgelegd wat er gedaan kan worden met de huidige GUI modeler.

4.3 Doelstelling

Het onderzoek moet duidelijk maken hoe de functionaliteit van de huidige GUI modeler is geprogrammeerd. Ook wordt er gekeken hoe deze momenteel wordt gebruikt. Aan de hand van deze kennis kan worden onderzocht welke functionaliteiten hiervan op grafische wijze gedaan kunnen worden, en wat de meest optimale uitwerkingen zijn van de grafische modeler.

De informatie die hieruit wordt opgedaan kan vervolgens worden gebruikt om een goed ontwerp te maken voor de grafische modeler, en wat hier wel en niet in moet, en om ervoor te zorgen dat de grafische modeler volledig naar wens is van de gebruikers. Ook kunnen alvast enkele prototypes worden ontwikkeld, waarop het eindproduct kan worden gebaseerd. In deze prototypes wordt geëxperimenteerd met verschillende technieken, zoals drag & drop en met verschillende layouts van de grafische modeler zelf.

De uitwerking tot de daadwerkelijke grafische modeler zorgt ervoor dat ontwikkelaars efficiënter en sneller applicaties kunnen maken. Ze kunnen gemakkelijker instellen hoe de vensters er uit zien, en zien direct hoe het eindresultaat er uit zal komen te zien, zonder de applicatie te hoeven draaien. Dit zorgt voor een verduidelijking en versimpeling voor het gehele proces.

4.4 Onderzoeksvragen

Het doel van het onderzoek is om antwoord te geven op de hoofdvraag. Om tot deze conclusie te komen wordt een onderverdeling gemaakt in deelvragen. Het beantwoorden van de deelvragen levert een antwoord op voor de hoofdvraag.

Hoofdvraag

Welke functionaliteiten bevat de huidige GUI modeler en hoe kunnen deze worden geïmplementeerd in een grafische modeler?

Deelvragen

- Hoe wordt momenteel gebruik gemaakt van de huidige GUI modeler?
- Welke onderdelen kunnen met de huidige GUI modeler aangepast worden?
- Welke onderdelen zijn wel of niet geschikt om met de grafische modeler aan te passen?
- Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen?

4.5 Gegevensverzameling

Voor het onderzoek vindt vooral deskresearch plaats. Door de huidige GUI modeler te inspecteren, kan worden uitgezocht welke functionaliteit er precies in zit. Door te kijken naar de codebase van de Software Factory kan bekeken worden hoe deze functionaliteiten precies zijn geïmplementeerd.

Door het interviewen van de leden van het Customer Solutions team kan worden uitgezocht hoe momenteel de GUI modeler wordt gebruikt, en welke eisen zij hebben aan de grafische modeler.

De informatie die hieruit gewonnen is kan dan weer gebruikt worden om te onderzoeken wat de beste oplossingen zijn voor de grafische modeler. Hiervoor kan ook gebruik worden gemaakt van het experimenteren met prototypes. Door enkele simpele prototypes te ontwikkelen kan de beste oplossing worden uitgezocht.

Verder kan van het Product Innovation team nog meer informatie gewonnen worden over de eventuele mogelijkheden en limieten van de Software Factory.

Ten slotte kan door tests met users gekeken worden hoe intuïtief de grafische modeler is voor standaard gebruik.

4.6 Onderzoeksplanning

Voor het onderzoek wordt eerst de huidige codebase bestudeerd. Wanneer duidelijk is hoe deze precies functioneert, en welke mogelijkheden er zijn voor de uitbreiding hiervan in de vorm van een grafische modeler, kan begonnen worden met het experimenteren met prototypes. Na het maken van een aantal prototypes, en het eventuele testen hiervan op gebruikers van de Software Factory, kan de beste oplossing gekozen worden om deze volledig uit te werken tot het daadwerkelijke product.

4.7 Bronnen

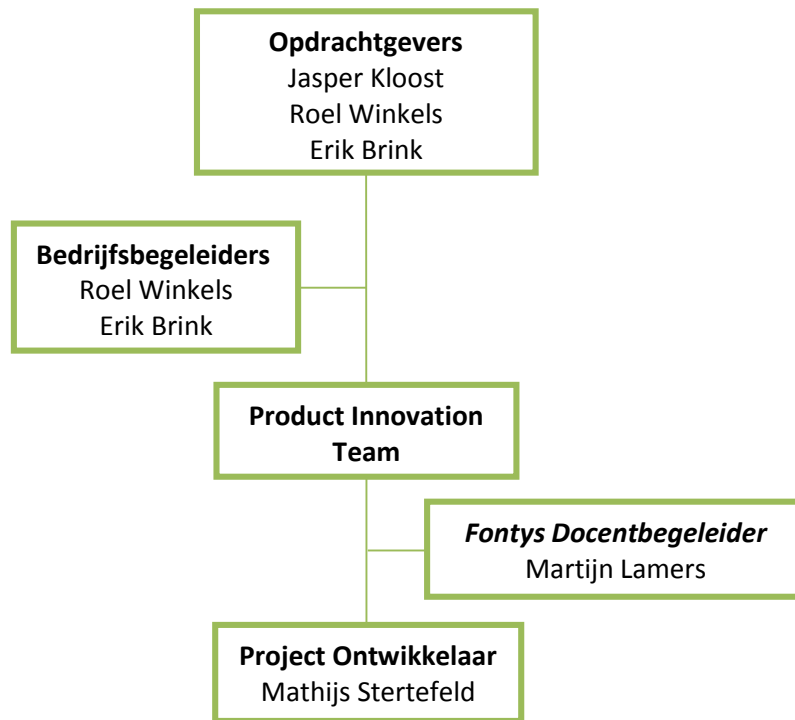
Er zijn dus een aantal verschillende bronnen voor dit onderzoek waar informatie vandaan kan worden gehaald:

- De code en documentatie van de Software Factory
- Het Product Innovation team
- De gebruikers van de Software Factory, het Customer Solution team
- Experimenten met prototypes
- Gebruikerstests van de nieuwe features.
- De documenten van het OND61 vak van de Fontys.

5 Project Organisatie Structuur

5.1 Projectorganisatie

Bij het uitvoeren van het project zijn een aantal verschillende mensen betrokken. Hieronder is in figuur 6 de hiërarchie te zien van de betrokkenen van het project.



Figuur 6. Het organigram van de betrokkenen.

5.2 Opdrachtgevers

De opdrachtgevers hebben het project gecreëerd. Zij overzien het project en zorgen ervoor dat het eindresultaat voldoet aan de eisen en overeenkomt met wat origineel was gepland. Wijzigingen aan het project zullen worden overlegd met de opdrachtgevers, om er voor te zorgen dat zij in overeenstemming zijn met het plan.

De opdrachtgevers zullen aan het einde van het project een advies geven voor de beoordeling van het project.

5.3 Bedrijfsbegeleiders

De bedrijfsbegeleiders bieden begeleiding voor de afstudeerder binnen het bedrijf. Zij zorgen ervoor dat de afstudeerder toegang heeft tot alle benodigdheden en zullen, wanneer er problemen ontstaan, hem ondersteuning bieden of doorverwijzen naar anderen. Zij communiceren ook met de docentbegeleider. Tevens adviseren zij de docentbegeleider over de beoordelingen van de afstudeerproducten.

5.4 Product Innovation Team

De huidige versie van de Software Factory wordt ontwikkeld door het Product Innovation Team. Leden van dit team zullen assistentie bieden en benodigde informatie verstrekken gedurende het project.

5.5 Fontys Docentbegeleider

De Fontys docentbegeleider houdt gedurende de afstudeerperiode contact met de afstudeerder. De afstudeerder rapporteert wekelijks over de voortgang van het project. De docentbegeleider bemiddelt de communicatie tussen de school, het afstudeerbedrijf en de afstudeerder.

Daarnaast bepaalt de docentbegeleider of het PID goedgekeurd wordt, wat een voorwaarde is om door te mogen gaan met het afstudeerproject. Ook beoordeelt de docentbegeleider samen met de tweede assessor de afstudeerproducten.

5.6 Projectontwikkelaar

De projectontwikkelaar is de afstudeerder. Hij werkt bij het bedrijf voor een vooraf vastgestelde tijdsperiode, waarin hij probeert het project te voltooien. Hij is de enige ontwikkelaar van het project, maar kan assistentie krijgen van collega's binnen het bedrijf, indien nodig.

5.7 Communicatie

Een overzicht van de communicatie tussen de verschillende betrokkenen van het project is te vinden in Bijlage A.

5.8 Project Regels

Thinkwise Software maakt gebruik van een interne portal genaamd de Thinkwise Community Portal. Hierin leggen de werknemers vast hoeveel uur ze iedere week hebben gewerkt en waar zij deze uren aan hebben besteed. Tijdens het gehele project wordt hierin de projectwerkzaamheden bijgehouden. Ook afwezigheden of schoolbijeenkomsten worden hier ingevoerd.

6 Project Beheersing

6.1 Rapportage en Voortgangsbewaking

Tijdens de projectperiode zullen er wekelijkse meetings zijn die normaal plaatsvinden op dinsdag om 11:00. Deze meetings worden gedaan met de projectbegeleiders. Hierin wordt gekeken welke voortgang is gemaakt en eventuele problemen besproken. Aan de hand van de feedback verkregen tijdens deze meetings, kunnen eventuele veranderingen worden gemaakt aan het project.

Er zal ook een wekelijkse rapportage zijn aan de docentbegeleider. Deze wordt uitgevoerd aan het einde van de week. Hierin wordt beschreven welke activiteiten zijn uitgevoerd in die week, zodat de docentbegeleider een beeld heeft van de voortgang die de afstudeerder maakt.

Verder wordt een urenverantwoording bijgehouden. Hierin wordt aangegeven hoeveel uur per dag is gewerkt aan welke onderdelen. Dit wordt bijgehouden in het TCP system van het bedrijf zelf. Het systeem vereist dat voor een week voor elke werkdag 8 uur is ingevuld. Hierin is het ook mogelijk om eventuele afwezigheid aan te geven.

6.2 Toleranties

Gedurende het project zal er een tolerantie zijn van 10% per fase van het tijdsbudget. Voor elke fase die het geplande budget overschrijdt, wordt het budget van de volgende fase aangepast aan de hand van de verandering.

6.3 Risicomanagement

Risico's worden tijdens de initialisatiefase vastgelegd in het PID. Deze zijn te vinden in bijlage C. Gedurende de loop van het project wordt hier op gecontroleerd. Voor alle mogelijke risico's die tijdens het project kunnen voorkomen, zullen er specifieke oplossingen zijn. Dit zorgt voor een gemakkelijke oplossing wanneer ze risico's ontstaan.

Verder is het project geen kernproduct voor het bedrijf. Als het project daarom niet succesvol is, zijn er geen grote consequenties voor het bedrijf.

6.4 Kwaliteitsbewaking

Voor het project is een lijst van requirements gemaakt. Deze requirements zijn geprioriteerd aan de hand van de MoSCoW methode (Brennan, 2009). Het project is succesvol uitgevoerd als voldaan is aan alle Must en Should requirements. Zie bijlage B voor de lijst van requirements.

Om te garanderen dat de kwaliteit van het project veilig is gesteld, is er regelmatige communicatie met de begeleiders, over zowel de voortgang als de producten. Dit zorgt er voor dat problemen die de kwaliteit in gevaar kunnen brengen snel opgemerkt en verholpen kunnen worden.

Gedurende de loop van het project wordt tijdens en na het realiseren getest. Dit zorgt ervoor dat het project voldoet aan de gestelde eisen.

6.5 Wijzigingsprocedure

Wijzigingen aan het project moeten eerst worden bekeken door de opdrachtgevers. Wanneer de opdrachtgevers de wijzigingen goedkeuren, zal worden overwogen of de wijzigingen voltooid kunnen worden binnen het huidige budget of dat er enkele aanpassingen moeten worden aan de wijziging.

6.6 Escalatieprocedure

Wanneer er escalaties van het budget plaats vinden, zal er een review plaatsvinden om uit te zoeken waarom dit is gebeurd. Naderhand zal de planning aangepast worden om te compenseren voor de escalatie en om mogelijke toekomstige escalaties te voorkomen.

Referenties

- Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis. Opgeroepen op 6 6, 2014
- Schwaber, K., & Sutherland, J. (2013, juli). *The Scrum Guide™*. Opgeroepen op 6 6, 2014, van The home of Scrum: [https://www.scrum.org/Portals/0/Documents/Scrum Guides/2013/Scrum-Guide.pdf](https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf)
- Thinkwise Software. (2013, April 2). *Grafische User Interface Modeler*. Opgehaald van Thinkwise Software: <http://www.thinkwisesoftware.com/wat-is-het/>

Bijlage A: Communicatieplan

Introductie

Dit communicatieplan toont de verschillende partijen die een interesse hebben bij het project en hoe zij betrokken zijn bij het project. Het geeft ook een overzicht van welke communicatievorm wordt gebruikt voor elke stakeholder. Dit bevat de partijen beschreven in de Project Organisatie Structuur.

Belanghebbenden van het Project

Wie	Namens	Belang	Communicatievorm
Externe gebruikers	Klanten	Moeten het uiteindelijke product gebruiken.	Eventuele feedback van klanten kan worden gebruikt om het product aan te passen en te verbeteren.
Opdrachtgevers	Thinkwise Software	Overziet het project en geeft het globale doel aan.	Controleren het plan van het project en enige veranderingen worden gerapporteerd aan hen. Mondelinge communicatie en email contact.
Product Innovation Team	Thinkwise Software	Bieden begeleiding gedurende het project.	Mondelinge communicatie en advies, email support.
Projectbegeleider	Thinkwise Software	Heeft controle over het project en zorgt voor behoud van voortgang.	Mondelinge communicatie en advies, email support.
Fontys Docentbegeleider	Fontys	Is verantwoordelijk dat de afstudeerder de juiste stappen zet en het juiste werk oplevert.	Email contact en bedrijfsbezoeken.
Project Ontwikkelaar	Fontys & Thinkwise Software	Ontwikkelt het daadwerkelijke project. Zorgt ervoor dat het project de planning volgt.	Communiqueert met alle betrokken partijen door middel van verschillende vormen van communicatie.

Communicatie Kanalen

Van	Naar	Informatie	Medium	Frequentie van data
Afstudeerder	Opdrachtgevers	Controle	Mondeling en email	Wijzigingen aan het plan moeten worden goedgekeurd door de Opdrachtgevers.
Projectbegeleider	Afstudeerder	Controle en begeleiding	Mondeling en email	De projectbegeleider stuurt en geeft feedback aan de afstudeerder.
Afstudeerder	Product Innovation Team	Advies	Mondeling en email	Bij eventuele problemen of verzoeken voor advies kan het Product Innovation team worden geraadpleegd.
Afstudeerder	Fontys Docentbegeleider	Controle	Mondeling en email	De afstudeerder rapporteert wekelijks over de voortgang. Ook zijn er geplande bedrijfsbezoeken.

Bijlage B: Acceptatie Criteria

Preview Window	Prioriteit: Must
Tijdens het modelen van de GUI is een preview window binnen het formulier zichtbaar waarin te zien is hoe het eindresultaat er uit zal zien.	
Positie Velden	Prioriteit: Must
In het preview window is het mogelijk om de positie van de velden en labels aan te passen.	
Formaat Velden	Prioriteit: Must
In het preview window is het mogelijk om het formaat van de velden en labels aan te passen.	
Zichtbaarheid Velden	Prioriteit: Must
In het preview window is het mogelijk om de zichtbaarheid van de velden en labels aan te passen.	
Elementen Slepen	Prioriteit: Must
In het preview window is het mogelijk om drag en drop functionaliteit te gebruiken om de layout aan te passen.	
Lijst Modeleren	Prioriteit: Must
Het is mogelijk om de Lijst aan te passen met de grafische GUI modeler.	
Formulier Modeleren	Prioriteit: Should
Het is mogelijk om het Formulier aan te passen met de grafische GUI modeler.	
Taken Modeleren	Prioriteit: Could
Het is mogelijk om de Taken aan te passen met de grafische GUI modeler.	
Rapporten Modeleren	Prioriteit: Could
Het is mogelijk om de Rapporten aan te passen met de grafische GUI modeler.	
Stijl Aanpassen	Prioriteit: Would
Het is mogelijk om de uiterlijke stijl aan te passen met de grafische GUI modeler.	

Bijlage C: Business Case

Inleiding

De business case bevat de overwegingen om het huidige project te starten. Het vormt de rechtvaardiging van het project. De onderbouwing van het project zal regelmatig worden geëvalueerd op basis van de Business Case. Het beschrijft de verwachte te investeren kosten ten opzichten van de verwachte voordelen en besparingen en de projectrisico's. De business case begint met een beschrijving van de gemaakte afwegingen voor het project.

Redenen

Het instellen van de layout van de schermen van applicaties gemaakt met de Software Factory gebeurt nu nog vooral door het invullen van lijsten en formulieren. Deze methode werkt prima, maar de gebruiksvriendelijkheid en eenvoudigheid kan nog zeer verbeterd worden.

Overwogen Alternatieven

Er is gekozen om het instellen van de layout grafisch te doen. Hierbij wordt gebruik gemaakt van "What-You-See-Is-What-You-Get" en "Drag & Drop" functionaliteit. Hiermee kunnen volgorde, zichtbaarheid, grootte van velden, tabbladen, stijlen en afbeeldingen ingesteld en aangepast worden op een grafische wijze.

Voordelen

Door het instellen van de layout van de schermen grafisch te maken, wordt de gebruiksvriendelijkheid sterk verbeterd en wordt het toegankelijker voor meer mensen om applicaties met aangepaste layout te ontwikkelen. Omdat het intuïtief en eenvoudiger gemaakt wordt, kunnen ook minder technische mensen het eenvoudig begrijpen en gebruiken.

Risico's

Hieronder zijn de risico's binnen het project beschreven. Voor elk risico is aangegeven op een schaal van 1 tot 5 hoe groot de kans is dat dit risico voorkomt en hoe ernstig de impact van dit risico is op het project. Verder is voor elk risico een geplande actie die kan worden ondernomen om dat risico te voorkomen, te verminderen of op te lossen.

Risico	Kans (1-5)	Impact (1-5)	Acties
Huidige functionaliteit van de GUI modeler valt weg met de grafische modeler.	2	2	De huidige GUI modeler ook nog beschikbaar laten in de Software Factory.
Het gebruik van de grafische modeler is minder intuïtief dan de huidige GUI modeler.	2	4	De grafische modeler herontwerpen om ervoor te zorgen dat deze intuïtiever is.
Het ontwerp voor de grafische modeler is technisch niet haalbaar.	4	4	Het ontwerp moet aangepast worden om handelingen op een andere manier uit te voeren.
De grafische modeler kan niet worden afgerond binnen de gegeven periode.	2	2	Er is in de planning tijd over voor een uitlooperperiode, waarin alsnog het project afgemaakt kan worden. Anders kan het na de afstudeerperiode door het bedrijf worden afgerond. Verder is het een niet essentieel
De grafische modeler is onvoldoende gedocumenteerd.	1	1	De begeleiders houden overzicht over de werkzaamheden en sturen aan indien er onvoldoende documentatie is.

Tijd en Kosten

De afstudeerperiode is gepland voor 85 dagen. Elke werkdag duurt 8 uur. Dit levert 680 werkuren op om het volledige project te voltooien. Deze tijd wordt verdeeld in verschillende fases. De eerste fase bevat de ontwikkeling van het PID. De tweede fase bevat het onderzoeken van de functionaliteit en mogelijkheden. De derde, vierde en vijfde fase bevat het ontwikkelen van de verschillende functionaliteiten van de applicatie. De zesde en laatste fase bevat het testen en distribueren van de gemaakte applicatie en het maken van de afstudeerscriptie en afstudeerpresentatie.

Bijlage D: Product Beschrijving

Doel

Het project wordt uitgevoerd omdat de gebruiksvriendelijkheid van de huidige Software Factory kan worden verbeterd. Door het modeleren van de GUI grafisch te maken, wordt het een stuk intuïtiever en eenvoudiger, ook voor minder technische mensen.

Samenstelling

Het eindresultaat bestaat uit het onderzoeksrapport en het eindproduct. Voor het onderzoek wordt onderzoek gedaan naar de functionaliteiten van de huidige GUI modeler en de mogelijkheden om deze grafisch te implementeren. De hoofdvraag van het onderzoek is dus als volgt: *“Welke functionaliteiten bevat de huidige GUI modeler en hoe kunnen deze worden geïmplementeerd in een grafische modeler?”*

Deze hoofdvraag kan worden beantwoord door de deelvragen te beantwoorden, welke als volgt zijn:

- Hoe wordt momenteel gebruik gemaakt van de huidige GUI modeler?
- Welke onderdelen kunnen met de huidige GUI modeler aangepast worden?
- Welke onderdelen zijn geschikt om met de grafische modeler aan te passen?
- Welke onderdelen zijn niet geschikt om met de grafische modeler aan te passen?
- Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen?

Om deze vragen te beantwoorden, wordt er op verschillende manieren onderzocht. Eerst wordt gekeken naar de functionaliteiten van huidige GUI modeler. Aan de hand hiervan worden ontwerpen opgesteld voor hoe deze functionaliteiten uitgevoerd kunnen worden in een grafische modeler. Vervolgens worden interviews uitgevoerd met gebruikers van de huidige GUI modeler, om extra informatie te winnen en feedback te krijgen op de ontwerpen. De ontwerpen worden vervolgens aangepast.

Voor de ontwerpen wordt dan gekeken in hoeverre het technische mogelijk is om de ontwerpen te realiseren. Indien nodig kunnen dan nog aanpassingen worden gemaakt aan de ontwerpen. Om de technische haalbaarheid te onderzoeken worden er enkele prototypes ontwikkeld.

Met de kennis opgedaan uit het onderzoek en de gemaakte prototypes wordt de basis gelegd voor het eindproduct zelf.

Het product bestaat uit een functionerende GUI modeler, waarmee de GUI grafisch opgebouwd kan worden. De grafische modeler heeft een Preview Window, zichtbaar naast de huidige modelers voor verschillende onderdelen. In dit venster kunnen verschillende eigenschappen van de velden van de bijbehorende tabel op grafische wijze aangepast worden. Zo kunnen de velden in breedte of hoogte veranderd worden door met de randen te slepen en kan de indeling van de velden worden veranderd door ze te verslepen. Uiteindelijk moet er de mogelijkheid zijn voor de grafische modeler om volgorde, zichtbaarheid, grootte van velden, indeling in groepen en tabbladen, stijlen en afbeeldingen aan te passen op grafische wijze.

Bronnen

De belangrijkste bron van informatie is het Product Innovation Team. Als ontwikkelaars van de Software Factory kunnen zij ondersteuning en begeleiding aanbieden. Verder kan via het internet erg veel globale informatie gevonden worden die mogelijk nuttig kan zijn voor dit project.

Formaat

De gehele Software Factory en de programma's gemaakt met de Software Factory hebben dezelfde stijl en uiterlijk. Het eindproduct moet dezelfde stijl hebben.

De Software Factory is grotendeels geprogrammeerd in Microsoft C#. Dit project is een uitbreiding op de huidige codebase en wordt dus ook geprogrammeerd in Microsoft C#.

Producent

De afstudeerder is de enige projectontwikkelaar die werkt aan de grafische modeler. Het Product Innovation team is beschikbaar voor informatie en om ondersteuning te bieden. De product eigenaren hebben controle over het project en kunnen zien of het volgens het plan verloopt.

Kwaliteitscriteria

Het belangrijkste criterium van het project is dat de functionaliteit en bruikbaarheid niet verminderd mag worden. Het mag niet moeilijker of onduidelijker worden om een layout samen te stellen.

Vereiste Vaardigheden

Bij het project komt een groot aantal van de competenties aan bod. Bij het opstellen van het PID en het uitvoeren van het onderzoek komen vooral de competenties Analyseren, Ontwerpen en Adviseren aan bod. Bij het ontwikkelen van het daadwerkelijke project komen Realiseren en Beheren goed aan bod. Professioneel en Methodisch Handelen komen sterk aan bod, omdat in een zakelijke bedrijfsomgeving moet worden gewerkt. Hierbij moet gebruik worden gemaakt van de ontwikkelingsregels en standaarden van het bedrijf. Samenwerken komt aan bod bij het communiceren met collega's over het project.

Om het project succesvol af te ronden, moet kennis van de verschillende competenties aanwezig zijn.

Verder moet er ook kennis zijn van Microsoft C#, waarin het product geprogrammeerd wordt.

Bijlage E: Planning

Voor het afstuderen is het project ingedeeld in verschillende fases. Er wordt begonnen met een Initiatiefase. Hierin wordt kennis gemaakt met het bedrijf en diens software. Ook wordt hierin het PID volledig geschreven.

Vervolgens is er de onderzoeksfase. Hierin wordt onderzoek uitgevoerd naar de huidige structuur en functionaliteit van de Software Factory en wat de mogelijkheden zijn om deze uit te breiden.

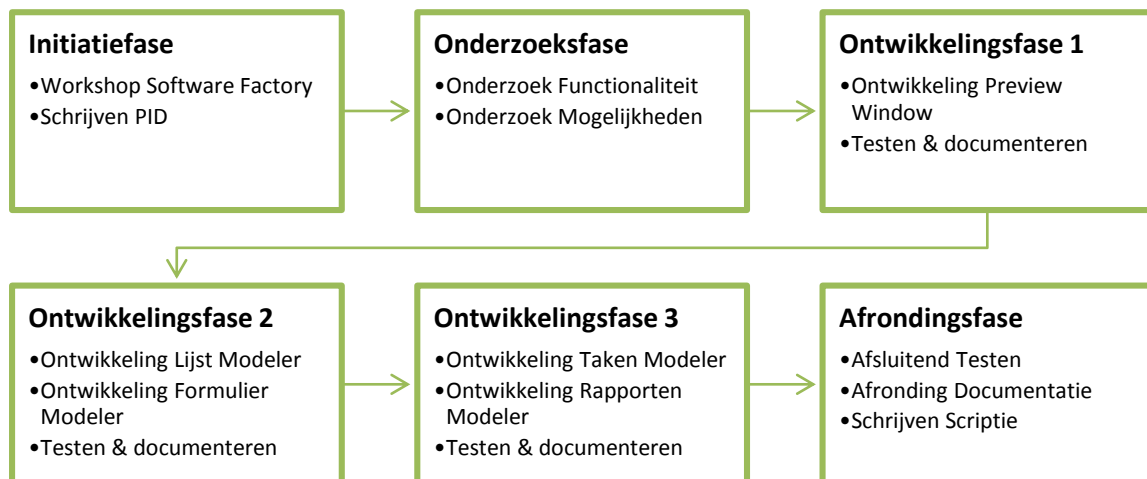
Hierna komen de drie ontwikkelingsfases. Het opdelen van de geplande tijd in fases zorgt ervoor dat iteratief gewerkt kan worden. Binnen elke fase wordt er voor een deel van het project aan een aantal taken gewerkt. Hierin wordt ook continu ontworpen, gebouwd, getest, opgeleverd en de documentatie bijgehouden.

In de eerste ontwikkelingsfase wordt het Preview Window ontwikkeld. Hiervan wordt dan aan het einde van de fase een prototype opgeleverd waarin het Preview Window functioneel is.

In de tweede ontwikkelingsfase wordt eerst de Lijst Modeler ontwikkeld en vervolgens de Formulier Modeler. Hiervan wordt dan ook weer een functioneel prototype opgeleverd.

In de derde ontwikkelingsfase wordt nog de Taken Modeler en de Rapporten Modeler ontwikkeld. Ook wordt hier nog gekeken naar eventuele mogelijkheid om de stijl functionaliteiten aan te passen via de Modeler. Afhankelijk van de hoeveelheid tijd die over is na ontwikkeling, wordt een prototype van deze functionaliteiten opgeleverd.

Ten slotte is er nog de afrondingsfase. Hierin wordt het geheel nog een keer getest om te kijken of het product voldoet aan de acceptatiecriteria, te vinden in bijlage B. Verder wordt de documentatie afgerond en de eindscripctie geschreven.



Bijgevoegd in het bestand Planning Afstudeerstage is een planning weergegeven van wanneer de verschillende onderdelen van het project worden uitgevoerd.

Bijlage B: Onderzoeksverslag

Grafische GUI Modeler

Onderzoeksverslag



THINKWISE
Software



Fontys

Hogeschool ICT



Project	Grafische GUI Modeler
Datum	23-5-2014
Auteur	Mathijs Stertefeld
Versie	1.0
Status	Voltooid

Inhoud

1	Inleiding	3
2	Deelvragen	4
3	Onderzoeksmethode	5
4	Onderzoekresultaten	6
4.1	Functionaliteit Huidige GUI Modeler	6
4.1.1	Lijst Modeler	6
4.1.2	Formulier Modeler	7
4.2	Gebruik Huidige GUI Modeler	9
4.3	Implementatie Grafische Modeler	10
4.3.1	Lijst Modeler	10
4.3.2	Formulier Modeler	12
4.4	Conclusie	17
5	Referenties	18
6	Bijlages	19
6.1	Interview Arjan Sollie	19
6.2	Interview Frank Wijnhout	20
6.3	Interview Roland van Aggele	21

1 Inleiding

Het doel van de afstudeerstage was om een uitbreiding te maken voor het hoofdproduct van Thinkwise Software, de Software Factory. Met de Software Factory kunnen volledige applicaties gemaakt worden. Van deze applicaties is het ook mogelijk om aan te passen hoe de user interface er uit zal zien. Momenteel wordt dit gedaan door een lijst van instellingen aan te passen, zonder direct en duidelijk te kunnen zien wat het effect van deze wijzigingen is op de applicatie.

De uitbreiding van de Software Factory moet ervoor zorgen dat het aanpassen van de user interface op grafische wijze gedaan kan worden, met behulp van de zogenoemde grafische modeler. Dit zorgt voor versimpeling en verduidelijking van het proces. Ook wordt hierin direct een beeld gegeven van hoe de eindapplicatie er uit zal komen te zien.

Het was de bedoeling dat er een prototype van de grafische modeler geïmplementeerd wordt binnen het afstudeerproject. Voordat het prototype echter kon worden gerealiseerd, is eerst onderzoek uitgevoerd moeten worden naar hoe deze het best ontworpen kon worden.

Voor het project is eerst onderzocht wat de meest geschikte oplossingen zijn om de aanpasbare onderdelen op grafische wijze aan te passen, door gebruik te maken van verschillende onderzoeksmethodes. Uiteindelijk heeft dit een ontwerp opgeleverd voor de grafische modeler. Op dit ontwerp konden mogelijke implementaties gebaseerd worden. Door middel van literatuuronderzoek zijn ontwerpstandaarden geraadpleegd waarop het grafisch aanpassen van de onderdelen is gebaseerd. Diepte interviews met huidige gebruikers van de Software Factory hebben inzicht geven over gewenste manieren om de instellingen aan te passen. Verder is nog de al aanwezige kennis van de afstudeerder gebruikt voor eventuele oplossingen.

Met al deze verworven kennis is dan een ontwerp voor alle onderdelen van de grafische modeler opgesteld. Dit heeft dan antwoord geven op de hoofdvraag van het onderzoek: *“Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen om het ontwerpen van de user interface gebruiksvriendelijker te maken?”*

2 Deelvragen

Het project heeft als doel om de huidige GUI modeler te vervangen met een grafische modeler. Dit moet ervoor zorgen dat het aanpassen van de user interface van applicaties eenvoudiger en efficiënter wordt en er direct overzicht is van hoe het eindproduct er uit zal zien.

Om ervoor te zorgen dat de grafische modeler aan deze eisen voldoet, moet het dezelfde functionaliteiten hebben als de huidige GUI modeler. Het is dus de bedoeling dat er geen verlies in functionaliteit is. In de nieuwe grafische modeler mogen dingen niet lastiger worden of niet gedaan kunnen worden. Om dit te garanderen, wordt gekeken hoe gebruikers momenteel werken met de huidige GUI modeler. Dit zal de eerste deelvraag beantwoorden, welke luidt: *“Hoe wordt momenteel gebruik gemaakt van de huidige GUI modeler?”*

Ook wordt onderzocht welke functionaliteiten de huidige GUI modeler heeft. Er wordt gekeken welke onderdelen van de GUI aanpasbaar zijn en op welke wijze deze aangepast kunnen worden. Dit zal antwoord geven op de tweede deelvraag: *“Welke onderdelen kunnen met de huidige GUI modeler aangepast worden?”*

Vervolgens wordt voor elk van deze onderdelen gekeken in hoeverre het mogelijk is om ze te bewerken met een grafische modeler. Verder wordt hier gekeken of er geen grafische oplossingen zijn om bepaalde instellingen aan te passen. Hier wordt dan gekeken wat een geschikte alternatieve oplossing kan zijn. Dit zorgt voor een antwoord op de derde deelvraag: *“Welke onderdelen zijn wel of niet geschikt om met een grafische modeler aan te passen?”*

Ten slotte wordt er gekeken naar alle onderdelen die aanpasbaar zijn. Er wordt onderzocht wat de meest geschikte manier is om elk onderdeel aan te passen op grafische wijze. Er wordt gekeken hoe dit gedaan kan worden op een dusdanige manier dat het eenvoudig en intuïtief is voor de gebruiker. Dit beantwoordt dan de vierde deelvraag: *“Wat zijn de optimale methodes om de verschillende onderdelen met de grafische modeler aan te passen?”*

Door al deze deelvragen te beantwoorden wordt een duidelijk overzicht gegeven van de functionaliteit van de huidige GUI modeler. Ook geeft het aan wat de functionaliteiteisen zijn van de grafische modeler. Dit zorgt dan voor een ontwerp van de grafische modeler, dat gebruikt kan worden voor de implementatie hiervan. Ook levert dit dan antwoord op de hoofdvraag: *“Welke functionaliteiten bevat de huidige GUI modeler en hoe kunnen deze worden geïmplementeerd in een grafische modeler?”*

3 Onderzoeksmethode

Voor het onderzoek is eerst gekeken naar de huidige GUI modeler. Alle onderdelen zijn onderzocht die hiermee kunnen worden aangepast en ingesteld. Deze zijn vervolgens allemaal in kaart gebracht.

Gebaseerd op alle aanpasbare instellingen is vervolgens een ontwerp gemaakt voor hoe deze verschillende instellingen grafisch aangepast kunnen worden. Hiervoor is gebruik gemaakt van tijdens de opleiding opgedane kennis, gecombineerd met literatuuronderzoek naar bestaande standaard interacties om gelijkwaardige instellingen aan te passen. Dit heeft uiteindelijk een ontwerp opgeleverd waarin wordt uitgelegd hoe het aanpassen van al deze instellingen op grafische wijze gedaan zou kunnen worden.

Hierna zijn er interviews gehouden met de gebruikers van de huidige GUI modeler in de Software Factory. Het ontwerp is aan hen voorgelegd, om aan te geven wat het plan is voor de grafische modeler. Daarna zijn er vragen gesteld over hoe zij de huidige GUI modeler gebruiken en welke eisen zij hebben aan de grafische modeler. Dit heeft voor extra inzicht gezorgd over hoe er precies gebruik van wordt gemaakt en welke onderdelen van de GUI modeler het belangrijkst zijn. Aan de hand van de informatie gewonnen uit deze interviews is het eerder in het onderzoek gemaakte ontwerp aangepast.

Dit ontwerp is vervolgens voorgelegd aan de ontwikkelaars van de Software Factory. Omdat zij veel kennis hebben over het ontwerp en het gebruik ervan, waren zij in staat om feedback te geven op het ontwerp. Aan de hand van deze feedback zijn wijzigingen en toevoegingen gemaakt aan het ontwerp.

Verder is er ook toelichting gegeven op de code van de huidige GUI modeler. Ook is de code zelf onderzocht om te kijken in hoe verre het ontwerp technisch mogelijk was om te implementeren. Omdat er vrij weinig documentatie beschikbaar was over de werking en structuur, is dit gedaan door veel testwerk en vragen aan de ontwikkelaars.

4 Onderzoekresultaten

4.1 Functionaliteit Huidige GUI Modeler

Met de Software Factory kan een groot aantal verschillende onderdelen van een applicatie worden aangepast. Niet alle onderdelen die aangepast kunnen worden, zitten ook in de huidige GUI modeler. Deze onderdelen moeten op andere plaatsen in de Software Factory aangepast worden.

De belangrijkste onderdelen van de GUI modeler zijn de lijst en de formulier modeler. Hiermee kan de indeling van lijsten en formulieren worden aangepast. Hieronder staan voor beide de onderdelen die aangepast kunnen worden die relevant zijn voor de GUI.

4.1.1 Lijst Modeler

In de lijst modeler wordt de indeling van de kolommen in een lijst ingedeeld. In de huidige lijst modeler kunnen een aantal instellingen worden gewijzigd. Hieronder is een overzicht van de verschillende onderdelen die aan kunnen worden gepast.

Kolom Volgorde

De volgorde van de kolommen binnen de lijst. Deze worden in de volgorde weergegeven zoals ze zijn opgeslagen.

Kolom Breedte

De breedte van een kolom binnen de lijst. Deze wordt opgeslagen in het aantal pixels.

Rij Hoogte

De hoogte van een rij binnen de lijst. Deze wordt opgeslagen in het aantal pixels. Deze wordt ingesteld op tabel niveau en is dus voor alle rijen hetzelfde. Dit is echter de minimum hoogte en de hoogte van een rij kan uit worden gerekend wanneer de inhoud niet binnen een rij past.

Aggregatie

Per kolom kan een aggregatie ingesteld worden. De aggregatie zorgt voor een functie op de alle rijen van een kolom. Zo kan bijvoorbeeld alle getallen in een kolom worden opgeteld om het totaal weer te geven.

Sorteren

Kolommen binnen een lijst kunnen gesorteerd worden. Dit sorteren kan vervolgens oplopend of aflopend gebeuren.

Wijzigingsinstellingen

Voor een lijst kunnen een aantal wijzigingsinstellen worden aangepast. Hieronder staan de aanpasbare instellingen.

Waardes binnen een lijst kunnen direct worden aangepast met Edit Lijst functionaliteit. Hierdoor hoeft geen formulier gebruikt te worden, maar kunnen de waardes in de lijst zelf worden gewijzigd. Dit kan worden uitgeschakeld en kan ook standaard aangezet worden.

Ook kan met Auto Resize de kolommen binnen een lijst worden ingesteld om automatisch de breedte aan te passen aan de inhoud van de kolom.

Afwijkend Lijst Type

Een kolom kan drie afwijkende lijst types hebben. De drie opties zijn Regulier, Alleen Lezen en Verborgen. Deze worden dan toegepast op een kolom binnen de lijst. Standaard heeft een kolom een waarde van Regulier. Hierin is de kolom gewoon zichtbaar, inclusief de waarde die er in staat. Ook is het nog mogelijk om de waarde in de kolom te veranderen. Wanneer de kolom op Alleen Lezen wordt gezet, is nog steeds de kolom zichtbaar, alleen is de kolom grijs en kan de waarde in de kolom niet worden aangepast. Bij de instelling Verborgen is de kolom volledig verwijderd van het scherm.

4.1.2 Formulier Modeler

In de formulier modeler wordt de indeling van velden op een formulier ingedeeld. Een veld bestaat uit een label en een inputveld, dat van verschillende types kan zijn. Afhankelijk van het type veld kan er ook een knop achter het veld staan die een relevante functie van het veld uitvoert.

In de huidige formulier modeler kunnen per veld een aantal instellingen worden gewijzigd. Hieronder is een overzicht van de verschillende onderdelen die aan kunnen worden gepast.

Veld Volgorde

Alle velden van een formulier worden in volgorde opgeslagen. De velden worden weergegeven op de opgeslagen volgorde.

Label Breedte

De breedte van een label dat voor een veld staat wordt opgeslagen in het aantal pixels.

Veld Breedte

De breedte van een veld wordt ook opgeslagen in het aantal pixels.

Veld Hoogte

De hoogte van een veld wordt opgeslagen in posities, waar één positie gelijk staat aan een regel in het formulier.

Veld Aantal Posities Verder

Een veld kan een aantal posities over slaan na het voorgaande veld. Hierdoor komen er lege regels tussen de velden te staan.

Groepen

Binnen het formulier kunnen groepen van velden worden gemaakt. Er kan zelf worden ingesteld bij welke velden nieuwe groepen beginnen. Bij een groep van velden wordt boven het eerste veld van de groep de groepstitel weergegeven. Deze groepstitel is ingesprongen en onderstreept.

Groepslabel

De tekst in het label van de groepstitel kan zelf worden ingesteld. De titel van een groep staat ingesprongen en is onderstreept.

Groepsicoon

Er kan een icoon voor de groep worden ingesteld. Dit icoon wordt weergegeven voor de groepstitel. Het is niet verplicht voor een groep om een icoon te hebben.

Kolommen

De velden kunnen binnen verschillende kolommen op het formulier geplaatst worden. Schermen vormen automatisch nieuwe kolommen wanneer de huidige kolom te vol raakt. Het is echter ook mogelijk om zelf in te delen vanaf welk veld een nieuwe kolom begint.

Tabbladen

De velden kunnen binnen verschillende tabbladen op het formulier geplaatst worden. Van een tabblad wordt de titel weergegeven. Het is mogelijk om zelf in te stellen vanaf welk veld een nieuw tabblad begint. Schermen vormen echter automatisch nieuwe tabbladen wanneer het huidige tabblad te vol raakt, om de plaats te hebben voor de velden. Deze nieuwe tabbladen hebben nog wel dezelfde titel. Al deze tabbladen krijgen vervolgens onderstreepte nummers achter de titel.

Tabblad Titel

De tekst in het label van de tabblad titel kan zelf worden ingesteld.

Tabblad Icoon

Er kan een icoon voor het tabblad worden ingesteld. Dit icoon wordt weergegeven voor de tabblad titel. Een tabblad hoeft niet verplicht een icoon te hebben.

Wijzigingsopties

Er zijn een aantal functies die per tabel kunnen worden ingesteld met betrekking tot het wijzigen van de inhoud. Deze functies worden uitgevoerd met behulp van de knoppen aan de onderkant van het formulier of via de ribbon. Deze instellingen zijn:

- Toevoegen
- Kopiëren
- Wijzigen
- Verwijderen

Verder kunnen in de ribbon een aantal andere instellingen worden aangepast. Deze instellingen zijn:

- Filteren
- Zoeken
- Importeren
- Exporteren

Afwijkend Formulier Type

Een veld kan drie verschillende afwijkende formulier types hebben. Deze verschillende types zijn Regulier, Alleen Lezen en Verborgen. Standaard heeft een veld een waarde van Regulier. Hierin is het veld en label gewoon zichtbaar, inclusief de waarde die er in staat. Ook is het nog mogelijk om de waarde in het veld te veranderen. Wanneer het veld op Alleen Lezen wordt gezet, is nog steeds het veld inclusief label zichtbaar, alleen is het veld grijs en kan de waarde in het veld niet worden aangepast. Bij de instelling Verborgen is het veld inclusief label volledig verwijderd van het scherm.

4.2 Gebruik Huidige GUI Modeler

Om te onderzoeken hoe de huidige GUI modeler wordt gebruikt, zijn er interviews gehouden met medewerkers van Thinkwise Software van het Customer Solutions team. Deze medewerkers maken met grote regelmaat gebruik van de huidige GUI modeler.

De interviews zijn gehouden met:

- Arjan Sollie, Customer Solutions Creator
- Frank Wijnhout, Project Manager Customer Solutions
- Roland van Agele, Project Manager Customer Solutions

Voor de interviews is eerst het bestaande ontwerp getoond en doorgesproken. Hierna zijn een aantal vragen gesteld over verschillende onderwerpen van zowel de huidige GUI modeler, als over de grafische modeler. Hieronder is voor alle onderdelen aangegeven wat de belangrijkste punten bleken uit de interviews.

De interviews zijn bijgevoegd in de bijlages in hoofdstuk 6.

Wijzigen Schermen

Het scherm waar het meeste aan wordt aangepast is het formulier. Dit scherm wordt door de eindgebruikers erg veel gebruikt en heeft een groot aantal instellingen die aangepast kunnen worden. Hierdoor is dit de belangrijkste modeler.

Volgorde van Wijzigingen

Bij het aanpassen van de gebruikersinterface van een applicatie begint de gebruiker gewoonlijk met het aanpassen van het formulier. Hier wordt dan de structuur van het formulier opgezet. De velden worden in de juiste volgorde gezet en er worden tabbladen en groepen aangemaakt om de velden die relevantie hebben bij elkaar te zetten. Daarna worden pas de kleinere details van de velden zelf aangepast, zoals bijvoorbeeld het formaat. Nadat het formulier is ingesteld, wordt de lijst aangepast.

Meeste Gewijzigde Instellingen

Bij veel projecten wordt gebruik gemaakt van het wijzigen van de volgorde van velden in het formulier, om relevante velden bij elkaar te zetten. Vervolgens wordt er dan ook veel gebruik gemaakt van tabbladen en groepen, voor verdere indeling van relevante velden. Als laatst wordt het formaat van velden en labels ook nog vaak aangepast.

Minst Gewijzigde Instellingen

Bij het ontwerpen van formulieren wordt het open laten van lege regels tussen velden minder vaak gebruikt. Hiervoor wordt dan vaker gebruik gemaakt van een nieuwe groep.

Wijzigen Instellingen naar Zelfde Waarde

Soms passen gebruikers de instellingen van verschillende velden aan naar dezelfde waarde. In de GUI modeler wordt dit vooral gedaan bij het aanpassen van de breedte van velden. Deze worden dan allemaal op hetzelfde formaat gezet, afwijkend van het standaard formaat. Verder wordt ook vaak de zichtbaarheid van meerdere velden tegelijkertijd aangepast.

Missende Instellingen in Huidige GUI Modeler

Een feature die nu erg gemist wordt in de GUI modeler is het direct kunnen zien wat het effect is van de wijzigingen aan de interface. Er worden enkele workarounds gebruikt om op andere plaatsen in de Software Factory te kunnen zien welke invloed gemaakte wijzigingen hebben op het eindproduct.

4.3 Implementatie Grafische Modeler

Door overleg met gebruikers en ontwikkelaars van de Software Factory en het maken van verschillende ontwerpen is er voor gekozen om de grafisch modeler te laten bestaan uit een “What-You-See-Is-What-You-Get” editor. In deze editor wordt de layout van de interface van de eindapplicatie op grafische wijze ingesteld. Hoe de applicatie er dan in deze editor uit ziet, komt overeen met hoe de eindapplicatie er dan uiteindelijk uit zal zien. Hierdoor hoeft niet de eindapplicatie zelf te worden opgestart om te bekijken hoe deze er uit ziet.

Om er voor te zorgen dat de grafische modeler geen functionaliteitsverlies heeft, vergeleken met de huidige GUI modeler, moeten alle onderdelen die in het vorige hoofdstuk zijn genoemd aanpasbaar zijn in de grafische modeler. Hieronder staat hoe de grafische modeler precies geïmplementeerd kan worden.

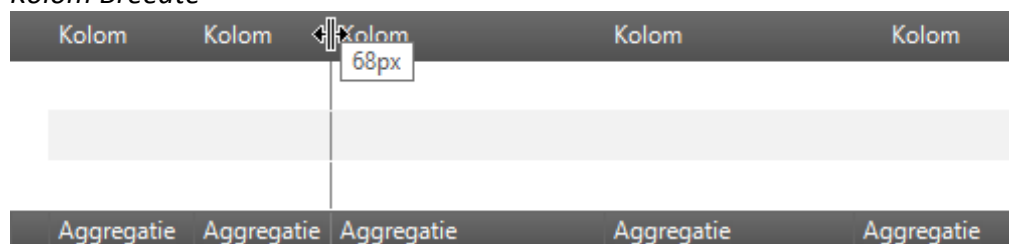
4.3.1 Lijst Modeler

In de lijst modeler worden de instellingen aangepast van de weergave van de informatie in de lijst. Voor het grafisch wijzigen van elk van de aanpasbare instellingen is hieronder de gewenste implementatie er van gegeven.

Kolom Volgorde

De kolommen in de lijst worden in de volgorde weergegeven zoals ze zijn opgeslagen. Door de kolommen te verslepen kan de volgorde van de kolommen in de lijst worden aangepast.

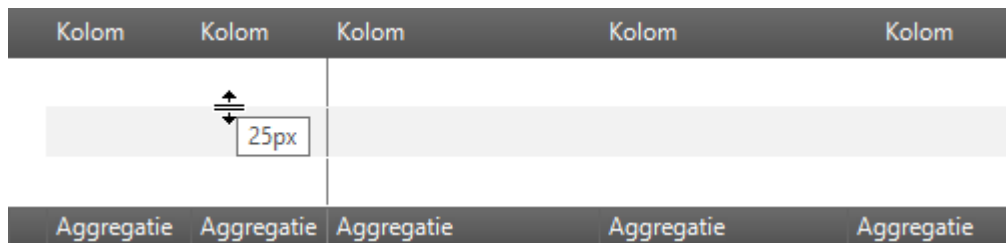
Kolom Breedte



Figuur 1. Het aanpassen van de breedte van een kolom.

Het aanpassen van de breedte van de kolommen kan worden gedaan door de randen van de kolommen te verslepen, zoals zichtbaar in figuur 1. Tijdens het verslepen kan dan zichtbaar zijn hoeveel pixels breed het label is met behulp van een tooltip. Dit is gebaseerd op het aanpassen van kolombreedte in onder andere Microsoft Office producten zoals Excel en Word. (Microsoft, 2014)

Rij Hoogte



Figuur 2. Het aanpassen van de hoogte van de rijen.

Het aanpassen van de hoogte van de rijen kan gedaan worden door de randen van een rij te verslepen, zoals zichtbaar in figuur 2. Omdat een de hoogte van de rijen voor de gehele tabel hetzelfde is, wordt het hier direct voor alle rijen ingesteld. (Microsoft, 2014)

Aggregatie



Figuur 3. Het aanpassen van de aggregatie van een kolom.

Het toevoegen van een aggregatie aan een kolom kan gedaan worden door op de aggregatie knop aan de onderkant van de kolom te klikken. Er verschijnt vervolgens een dropdown menu waarin gekozen kan worden welke aggregatie gebruikt moet worden, zoals zichtbaar in figuur 3.

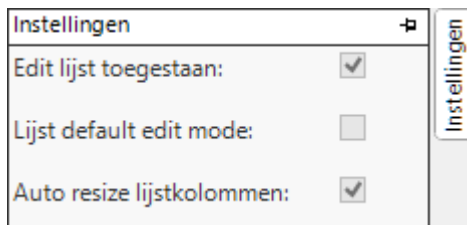
Sorteren

Kolom	▼ Kolom	Kolom	Kolom	Kolom
Aggregatie	Aggregatie	Aggregatie	Aggregatie	Aggregatie

Figuur 4. Het aanpassen van de sortering van een lijst.

Door op de titel van een kolom te klikken, kan ingesteld worden dat de lijst op die kolom moet worden geselecteerd. Er kan worden ingesteld of de lijst oplopend of aflopend moet worden gesorteerd. Er wordt dan in de kolom een pijl naast de titel weergegeven, zoals zichtbaar in figuur 4. Dit is gebaseerd op hoe in de huidige lijst de sortering aangepast kan worden. (Infragistics NetAdvantage, 2012)

Wijzigingsinstellingen



Figuur 5. Het auto hide scherm met de aanpasbare instellingen.

Het instellen van de verschillende wijzigingsopties kan gedaan worden door weer een optiemenu aan de rechterkant van de modeler weer te geven, zoals zichtbaar in figuur 5. Om ervoor te zorgen dat het scherm niet constant in het zicht is, kan het met auto hide verborgen worden zodat het alleen verschijnt als de muis er overheen wordt gehouden, of wanneer het vastgezet wordt. Hier is het auto hide scherm ook weer gebaseerd op de auto hide mogelijkheden van schermen in Microsoft Visual Studio. (Microsoft, 2014)

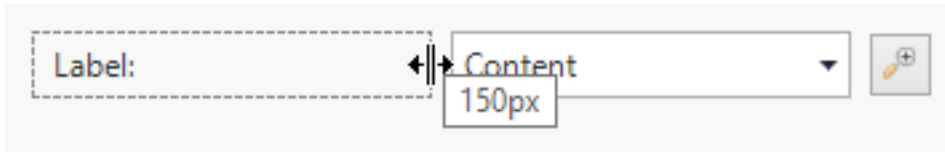
4.3.2 Formulier Modeler

In de formulier modeler worden de instellingen aangepast van hoe de informatie in het formulier wordt weergegeven. Voor het grafisch wijzigen van elk van de aanpasbare instellingen is hieronder de gewenste implementatie er van gegeven.

Veld Volgorde

De velden worden in het formulier in de volgorde weergegeven zoals ze zijn opgeslagen. Door de velden te verslepen naar een andere positie, kan de volgorde van de velden in het formulier worden aangepast.

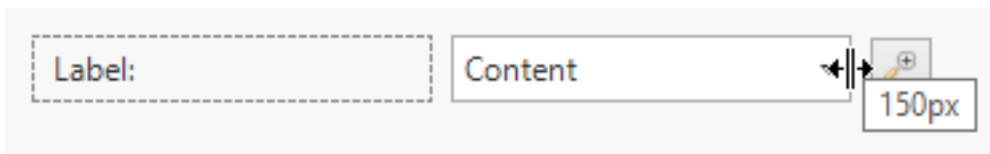
Label Breedte



Figuur 6. Het aanpassen van de breedte van een label.

Het aanpassen van de breedte van de labels kan worden gedaan door de randen van de labels te verslepen, zoals zichtbaar in figuur 6. Tijdens het verslepen kan dan zichtbaar zijn hoeveel pixels breed het label is met behulp van een tooltip. Omdat labels echter geen duidelijk zichtbare randen hebben, kan met stippellijn worden aangegeven waar de randen hiervan lopen. In de eindapplicatie zijn deze randen dan weer onzichtbaar. Het verslepen van de randen van objecten is gebaseerd op het aanpassen van het formaat in onder andere verscheidene Microsoft Office producten. (Microsoft, 2014)

Veld Breedte



Figuur 7. Het aanpassen van de breedte van een veld.

Het aanpassen van de breedte van de velden kan, net zoals bij de labels, gedaan worden door de randen van de velden te verslepen, zoals zichtbaar in figuur 7. (Microsoft, 2014) Ook kan hier weer worden weergegeven hoeveel pixels breed het veld is met behulp van een tooltip.

Veld Hoogte



Figuur 8. Het aanpassen van de hoogte van een veld.

Het aanpassen van de hoogte van de velden kan ook weer gedaan worden door het verslepen van de randen, zoals zichtbaar in figuur 8. (Microsoft, 2014) Omdat de hoogte wordt bepaald in posities, wat het aantal regels op het formulier is, gebeurt het verslepen per positie en niet per pixel. Hier kan dan het aantal posities worden weergegeven in een tooltip.

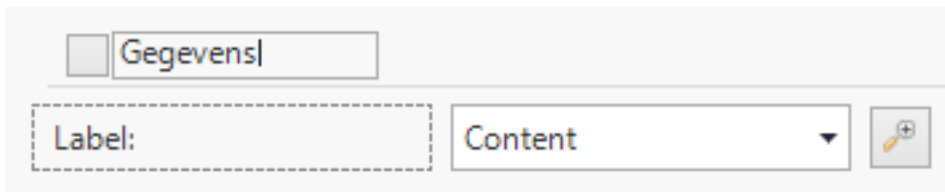
Veld Aantal Posities Verder



Figuur 9. Het invoegen van een lege regel of groep tussen velden.

Wanneer de cursor tussen twee velden wordt gehouden, kan er een lijn met een plusteken worden weergegeven. Wanneer hierop wordt geklikt komt er een menu tevoorschijn. Hierin kan dan worden gekozen om een enkele of meerdere nieuwe lege regels er tussen toe te voegen, zoals zichtbaar in figuur 9. Het tonen van een toevoegmogelijkheid tussen twee velden is gebaseerd op het invoegen van rijen in een tabel in Microsoft Word. (Microsoft, 2014)

Groepen



Figuur 10. Het aanpassen van het groep label en het groep icoon.

Na het klikken op de lijn die tussen regels verschijnt, kan in het menu worden gekozen om een nieuwe groep toe te voegen. Wanneer een groep is toegevoegd, kan deze weer verwijderd worden door er op te rechteklikken.

Groep Titel

Wanneer een groep is toegevoegd, kan de titel worden gewijzigd door er op te dubbelklikken, waardoor de titel geselecteerd en gewijzigd kan worden, zoals zichtbaar in figuur 10.

Groep Icoon

Voor de titel van de groep wordt een leeg vierkant weergegeven, zoals zichtbaar in figuur 10. Wanneer hierop geklikt wordt, kan de locatie van de afbeelding worden ingevoerd. Vervolgens wordt dit icoon in het vierkant weergegeven en kan weer gewijzigd worden door er op te klikken.

Kolommen



Figuur 11. Het aangeven van een handmatig toegevoegde kolom.

Het verslepen van een veld naar een nieuwe kolom zorgt ervoor dat vanaf dat veld een nieuwe kolom begint. Omdat de Software Factory ook automatisch nieuwe kolommen kan toevoegen wanneer er niet genoeg ruimte is, wordt door middel van een streep onder aan de voorgaande kolom duidelijk gemaakt dat deze kolom handmatig is beëindigd en een volgende kolom is toegevoegd, zoals zichtbaar in figuur 11.

Tabbladen



Figuur 12. Het aanpassen van de tabblad titel en tabblad icoon, en het toevoegen van nieuwe tabbladen.

Het verslepen van een veld naar een nieuw tabblad zorgt ervoor dat vanaf dat veld een nieuw tabblad begint, zoals zichtbaar in figuur 12. Omdat de Software Factory ook automatisch nieuwe tabbladen kan toevoegen wanneer er niet genoeg ruimte is, wordt door middel van het toevoegen van onderstreepte nummers achter de titel van tabbladen duidelijk gemaakt dat dit tabblad automatisch is toegevoegd.

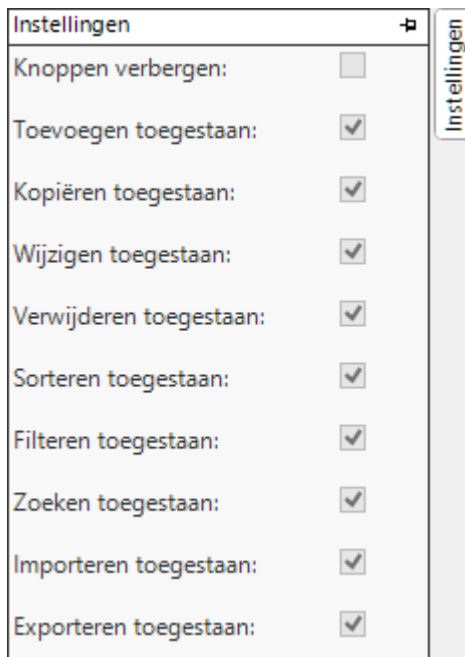
Tabblad Titel

Wanneer een tabblad is toegevoegd, kan de titel worden gewijzigd door er op te dubbelklikken, waardoor de titel geselecteerd en gewijzigd kan worden, zoals zichtbaar in figuur 12. Een automatisch toegevoegd tabblad heeft dezelfde titel als het voorgaande tabblad. Het wijzigen van de titels hiervan zorgt ervoor dat alle automatisch toegevoegde tabbladen naar dezelfde titel worden gewijzigd.

Tabblad Icoon

Voor de titel van het tabblad wordt een leeg vierkant weergegeven, zoals zichtbaar in figuur 12. Wanneer hierop geklikt wordt, kan de locatie van de afbeelding worden ingevoegd. Vervolgens wordt dit icoon in het vierkant weergegeven en kan weer gewijzigd worden door er op te klikken.

Wijzigingsopties



Figuur 13. Het auto hide scherm met de aanpasbare instellingen.

Het instellen van de verschillende wijzigingsopties kan gedaan worden door een optiemenu aan de rechterkant van de modeler weer te geven, zoals zichtbaar in figuur 13. In dit menu kunnen de verschillende soorten van wijzigingsmogelijkheden worden in- of uitgeschakeld.

Om ervoor te zorgen dat het scherm niet constant in het zicht is, kan het met auto hide verborgen worden zodat het alleen verschijnt als de muis er overheen wordt gehouden, of wanneer het vastgezet wordt. Het auto hide scherm is gebaseerd op de auto hide mogelijkheden van schermen in Microsoft Visual Studio. (Microsoft, 2014)

Afwijkend Formulier Type



Figuur 14. Het weergeven van verborgen en alleen lezen velden, en het aanpassen van het type van een veld.

Het is mogelijk om tussen twee velden een streep weer te geven, om aan te geven dat er een Verborgen veld tussen zit, zonder de velden verder uit elkaar staan dan hoe ver ze in het eindproduct uit elkaar zullen staan. Wanneer er meerdere Verborgen velden tussen staan is het mogelijk om het getal bij de streep weer te geven van hoeveel verborgen velden er tussen bevinden.

Wanneer een veld op Alleen Lezen staat dan kan het veld grijs worden weergegeven.

Het aanpassen van het type van een veld kan worden gedaan door er rechts op te klikken, en vervolgens te selecteren welke type het moet krijgen, zoals zichtbaar in figuur 14.

4.4 Conclusie

Het onderzoeken van de Software Factory heeft een duidelijk overzicht gegeven van alle functionaliteiten die in de huidige GUI modeler zitten. Dit heeft helder in kaart gebracht welke onderdelen met de huidige GUI modeler aangepast kunnen worden.

De interviews met de gebruikers van de Software Factory hebben goed aangegeven wat de meest belangrijke onderdelen zijn. Hieruit blijkt dat het meest gewerkt wordt met de formulier modeler. Hier wordt dan vooral gebruik gemaakt van het aanpassen van de volgorde van velden en het indelen van de velden in verschillende soorten groepen. Deze interviews hebben een goed beeld kunnen geven van hoe momenteel gebruik wordt gemaakt van de huidige GUI modeler.

De informatie gekregen uit de interviews, gecombineerd met de feedback op het ontwerp, heeft gezorgd voor aanpassingen aan het oorspronkelijke ontwerp van de grafische modeler. Door vervolgens te overleggen met de andere ontwikkelaars van de Software Factory is gebleken welke onderdelen wel en niet geschikt zijn om aan te passen met de grafische modeler. Zo kunnen veel onderdelen worden geïmplementeerd in een "What-You-See-Is-What-You-Get" editor en kunnen de overige instellingen worden gewijzigd in een menu binnen de grafische modeler.

De informatie uit de interviews is gecombineerd met de onderzochte standaarden voor interface ontwerp. Dit heeft voor alle aanpasbare instellingen een manier opgeleverd om deze grafisch te bewerken. Hiermee is bepaald wat de optimale methodes zijn om de verschillende onderdelen met de grafische modeler aan te passen.

Met alle verzamelde informatie uit het onderzoek is dus vastgelegd wat de functionaliteit is van de huidige GUI modeler en hoe deze geïmplementeerd kan worden in een grafische modeler. Dit is vastgelegd in een ontwerp dat is gebruikt om de daadwerkelijke grafische modeler te ontwikkelen.

5 Referenties

- Infragistics NetAdvantage. (2012). *Sorting Columns*. Opgeroepen op 03 13, 2014, van Infragistics NetAdvantage Windows Forms: http://help.infragistics.com/Help/Doc/WinForms/2012.2/CLR4.0/html/WinGrid_Sorting_Columns.html
- Microsoft. (2014). *Add columns and rows in a table*. Opgeroepen op 03 10, 2014, van Microsoft Office Support: <https://office.microsoft.com/en-001/word-help/add-columns-and-rows-in-a-table-HA102840121.aspx>
- Microsoft. (2014). *Change the column width and row height*. Opgeroepen op 03 10, 2014, van Microsoft Office Support: <https://office.microsoft.com/en-001/excel-help/change-the-column-width-and-row-height-HA102749049.aspx>
- Microsoft. (2014). *How to: Arrange and Dock Windows*. Opgeroepen op 03 12, 2014, van Microsoft Developer Network: <http://msdn.microsoft.com/en-us/library/z4y0hsax.aspx>
- Microsoft. (2014). *Resize a picture or shape*. Opgeroepen op 03 10, 2014, van Microsoft Office Support: <https://office.microsoft.com/en-001/powerpoint-help/resize-a-picture-or-shape-HP005195097.aspx?CTT=1>

6 Bijlages

6.1 Interview Arjan Sollie

Geïnterviewde: Arjan Sollie, Customer Solutions Creator
Interviewer: Mathijs Stertefeld, Afstudeerder
Locatie: Thinkwise Software, Boogschutterstraat 7b, Apeldoorn
Datum: 14-03-2014

Bij het interview zijn eerst door de afstudeerder de ontwerpen getoond van de grafische modeler. Hierdoor heeft Arjan een beeld kunnen krijgen van de plannen. Er is toelichting gegeven op de verschillende onderdelen die in de grafische modeler zitten en hoe deze functioneren.

Vervolgens zijn er een aantal vragen gesteld over hoe Arjan gebruik maakt van de huidige GUI modeler. De eerste gestelde vraag ging over welke schermen hij het meest aanpast.

Arjan gaf hier aan dat het scherm waarin hij de meeste wijzigingen maakt voor applicaties voor klanten het formulier is. Verder gaf hij aan dat hij ook vaak wijzigingen maakt aan de schermen voor de Taken en die van de Prefilters.

De volgende vraag die gesteld is ging over in welke volgorde hij de verschillende instellingen in de grafische modeler aanpast.

Hier gaf Arjan aan dat hij vaak begint bij het aanpassingen maken aan het formulier. Hij begint met het toekennen van de velden aan verschillende tabbladen, om deze zo te groeperen. Vervolgens past hij de mogelijkheid aan om via het formulier data te kunnen toevoegen, wijzigen of aanpassen. Hierna wordt de volgorde van de velden binnen de tabbladen ingedeeld. Ook deelt hij de relevante velden in binnen groepen van bij elkaar passende informatie. Daarna deelt hij de velden en groepen in verschillende kolommen in. Hierna past hij de grootte van de verschillende velden aan. Ten slotte past hij voor de velden de formulier types aan. Hiermee kunnen velden als niet aanpasbaar of onzichtbaar worden weergegeven.

Hierna is gevraagd over welke instellingen hij het meest aanpast.

Arjan antwoordde hier dat bij het aanpassen van een applicatie, het aanpassen van de grootte van velden en de bijbehorende labels het meest wordt gedaan.

Vervolgens is gevraagd welke instellingen hij zelden aanpast.

Hier zei Arjan dat hij niet vaak open regels invoert tussen velden. Ook gaf hij aan dat hij de sorteerinstellingen van lijsten zelden aanpast.

De volgende vraag ging over welke instellingen het belangrijkst zijn om aan te passen.

Arjan was hier van mening dat het indelen van de velden in verschillende tabbladen wel een cruciale functie is. Ook gaf hij aan dat het aanmaken van Prefilters belangrijk is.

Hierna luidde de vraag of er instellingen zijn die hij veel naar dezelfde waardes aanpast.

Volgens Arjan stelde hij vaak de grootte van velden voor bedragen in op een kleiner formaat dan de standaard grootte.

De laatste vraag ging over of hij nog instellingen mist in de huidige GUI modeler.

Arjan gaf aan dat hij geen autofilters kon instellen op tabelniveau.

Hiermee waren alle vragen beantwoord en is het interview afgerond.

6.2 Interview Frank Wijnhout

Geïnterviewde: Frank Wijnhout, Project Manager Customer Solutions
Interviewer: Mathijs Stertefeld, Afstudeerder
Locatie: Thinkwise Software, Boogschutterstraat 7b, Apeldoorn
Datum: 17-03-2014

Bij het interview zijn eerst door de afstudeerder de ontwerpen getoond van de grafische modeler. Hierdoor heeft Frank een beeld kunnen krijgen van de plannen. Er is toelichting gegeven op de verschillende onderdelen die in de grafische modeler zitten en hoe deze functioneren.

Vervolgens zijn er een aantal vragen gesteld over hoe Frank gebruik maakt van de huidige GUI modeler. De eerste gestelde vraag ging over welke schermen hij het meest aanpast.

Frank gaf hier aan dat hij het meest aanpassingen maakt aan het formulieren. Ook de lijst past hij vaak aan, maar hier maakt hij minder aanpassingen.

De volgende vraag die gesteld is ging over in welke volgorde hij de verschillende instellingen in de grafische modeler aanpast.

Hier gaf Frank aan dat hij begint met het aanpassen van het formulier. Hij begint met het bepalen van een juiste structuur voor de velden, aan de hand van welke velden bij elkaar passen. Hij start dan met het indelen van de tabbladen. Vervolgens deelt hij de velden in groepen in. Hierna past hij de velden zelf aan. Na het aanpassen van het formulier, gaat hij verder met het aanpassen van de lijst.

Hierna is gevraagd over welke instellingen hij het meest aanpast.

Frank antwoordde hier dat hij het meest gebruik maakt van het indelen van de velden in tabbladen en groepen.

Vervolgens is gevraagd welke instellingen hij zelden aanpast.

Hier zei Frank dat hij de breedte van velden niet vaak aanpast. Hij gaf aan dat hij dit niet vaak deed om formulieren uniform en overzichtelijk te houden.

De volgende vraag ging over welke instellingen het belangrijkst zijn om aan te passen.

Frank vond hier dat de tabbladen en groepen, die hij ook het vaakst aanpast, het belangrijkst zijn om aan te passen.

Hierna luidde de vraag of er instellingen zijn die hij veel naar dezelfde waardes aanpast.

Volgens Frank stelde hij de dynamische procedures vaak op dezelfde waardes in.

De laatste vraag ging over of hij nog instellingen mist in de huidige GUI modeler.

Frank gaf hier aan dat hij instellingen graag zowel relatief wilde aanpassen, in toevoeging van het huidige systeem waar aanpassingen voornamelijk absoluut worden gedaan.

Hiermee waren alle vragen beantwoord en is het interview afgerond.

6.3 Interview Roland van Aggele

Geïnterviewde: Roland van Aggele, Project Manager Customer Solutions
Interviewer: Mathijs Stertefeld, Afstudeerder
Locatie: Thinkwise Software, Boogschutterstraat 7b, Apeldoorn
Datum: 03-04-2014

Bij het interview zijn eerst door de afstudeerder de ontwerpen getoond van de grafische modeler. Hierdoor heeft Arjan een beeld kunnen krijgen van de plannen. Er is toelichting gegeven op de verschillende onderdelen die in de grafische modeler zitten en hoe deze functioneren.

Vervolgens zijn er een aantal vragen gesteld over hoe Arjan gebruik maakt van de huidige GUI modeler. De eerste gestelde vraag ging over welke schermen hij het meest aanpast.

Roland gaf hier aan dat hij vooral aanpassingen maakt aan het formulier en de lijst. Verder past hij ook de sorteringen en de prefilters aan.

De volgende vraag die gesteld is ging over in welke volgorde hij de verschillende instellingen in de grafische modeler aanpast.

Hier gaf Roland aan dat vaak begint met het formulier. Hier past hij eerst de opmaak aan. Dan zet Roland de velden in de juiste volgorde. Vervolgens deelt hij de velden in tabbladen en groepen in. Hierna stelt hij de specifieke instellingen per veld in.

Hierna is gevraagd over welke instellingen hij het meest aanpast.

Roland antwoordde hier dat hij het meest de volgorde van velden aanpast. Ook past hij vaak de zichtbaarheid van velden aan. Ten slotte gaf Roland aan dat hij de breedte van velden nog vaak wijzigt.

Vervolgens is gevraagd welke instellingen hij zelden aanpast.

Hier zei Roland dat hij niet vaak de afwijkende lookup instelde. Verder past hij zelden het zoeken en het filteren aan. Ook maakt hij weinig gebruik van de voorwaardelijke opmaak.

De volgende vraag ging over welke instellingen het belangrijkst zijn om aan te passen.

Roland vond hier het instellen van de volgorde van velden het belangrijkst. Verder vond hij het aanpassen van de sortering erg belangrijk en het instellen van de prefilters erg belangrijk.

Hierna luidde de vraag of er instellingen zijn die hij veel naar dezelfde waarden aanpast.

Volgens Roland stelde hij vaak de breedte van velden en labels aan naar dezelfde waarden. Ook zet hij vaak vele velden tegelijk op dezelfde zichtbaarheid.

De laatste vraag ging over of hij nog instellingen mist in de huidige GUI modeler.

Roland gaf aan dat hij een preview window mist, waarin hij direct kan zien welk effect de wijzigingen hebben op het eindproduct.

Hiermee waren alle vragen beantwoord en is het interview afgerond.

Bijlage C: User Stories

Hieronder zijn voor de verschillende onderdelen van de implementatie de user stories weergegeven. Elke user story heeft een prioriteit toegewezen gekregen aan de hand van de MoSCoW methode (Brennan, 2009). Per user story zijn er een aantal taken die uit moeten worden gevoerd om het onderdeel te voltooien. Per taak is vervolgens aangegeven hoe lang het onderdeel wordt geschat te duren.

Basis Grafische Modeler

User Story	MoSCoW	Taak	Tijd
De property lijst voor de lijst en formulier modeler wordt weergegeven in de Software Factory.	Must	Eigen schermtype voor property lijst	2
		Tonen lijst met aanpasbare eigenschappen	16
		Invullen van de lijst met informatie uit het model	16
		Doorvoeren wijzigingen naar model	2

User Story	MoSCoW	Taak	Tijd
De grafisch aanpasbare lijst dat er hetzelfde uit ziet als in het eindproduct wordt weergegeven in de lijst modeler in de Software Factory.	Must	Eigen schermtype voor lijst modeler	2
		Tonen grafisch aanpasbare lijst zoals in eindproduct	16
		Invullen van de lijst met informatie uit de eindapplicatie	16
		Doorvoeren wijzigingen naar model	2

User Story	MoSCoW	Taak	Tijd
Het grafisch aanpasbare formulier dat er hetzelfde uit ziet als in het eindproduct wordt weergegeven in de formulier modeler in de Software Factory.	Must	Eigen schermtype voor formulier modeler	2
		Tonen grafisch aanpasbaar formulier zoals in eindproduct	16
		Invullen van het formulier met informatie uit de eindapplicatie	16
		Doorvoeren wijzigingen naar model	2

User Story	MoSCoW	Taak	Tijd
Het auto hide scherm met aanpasbare instellingen wordt aan de zijkant van de modeler weergegeven.	Must	Apart scherm voor instellingen	8
		Auto hide functionaliteit	4
		Doorvoeren wijzigingen naar model	2

Lijst Modeler

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de breedte van kolommen in pixels aanpassen door het verslepen van de randen.	Must	Slepen rechterkant kolom	16
		Beïnvloeding andere kolommen	8
		Doorvoeren wijzigingen naar model	4

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de hoogte van rijen in pixels aanpassen door het verslepen van de randen.	Should	Slepen onderkant rij	16
		Beïnvloeding andere rijen	8
		Doorvoeren wijzigingen naar model	4

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de aggregatie van een kolom aanpassen met een menu onderaan de lijst.	Must	Weergeven aggregatie knop	16
		Tonen menu met aggregatieopties	8
		Doorvoeren wijzigingen naar model	4

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de wijzigingsinstellingen aanpassen door middel van een auto hide menu aan de zijkant van de modeler.	Must	Tonen auto hide window met opties	8
		Aanpassen verschillende instellingen	8
		Doorvoeren wijzigingen naar model	4

Formulier Modeler

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de breedte van labels in pixels aanpassen door het verslepen van de randen.	Must	Weergeven rand om label	1
		Slepen rechterkant label	1
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de breedte van velden in pixels aanpassen door het verslepen van de randen.	Must	Slepen rechterkant veld	1
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de hoogte van velden in posities aanpassen door het verslepen van de randen.	Should	Slepen onderkant veld	1
		Aanpassing in posities	1
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de volgorde van velden aanpassen door ze te verslepen.	Would	Verslepen veld naar andere positie	4
		Beïnvloeding andere velden	3
		Doorvoeren wijzingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan lege regels invoegen tussen velden door het klikken op het verschijnende menu.	Should	Popup tonen bij muis tussen velden	3
		Selecteren van invoeren regel	1
		Invoegen lege regel	2
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan velden in groepen indelen door het klikken op het verschijnende menu.	Must	Popup tonen bij muis tussen velden	3
		Selecteren van invoeren regel	1
		Invoegen groep	2
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan het label van een groep aanpassen door er op te klikken.	Should	Label als wijzigbaar veld	2
		Klikken op label om te wijzigen	2
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan het icoon van een groep aanpassen door er op te klikken.	Would	Icoon als knop	3
		Ingeven directory voor icoon	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan een tabblad invoegen door een veld te slepen naar de nieuwe tabblad knop.	Must	Nieuwe tabblad knop	2
		Slepen van velden op knop	4
		Aanmaken van nieuw tabblad	2
		Beïnvloeding andere velden	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan het label van een tabblad aanpassen door er op te klikken.	Should	Label als wijzigbaar veld	3
		Klikken op label om te wijzigen	2
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan het icoon van een tabblad aanpassen door er op te klikken.	Would	Icoon als knop	3
		Ingeven directory voor icoon	1
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan de wijzigingsinstellingen aanpassen door middel van een auto hide menu aan de zijkant van de modeler.	Must	Tonen auto hide window met opties	1
		Aanpassen verschillende instellingen	2
		Doorvoeren wijzigingen naar model	1

User Story	MoSCoW	Taak	Tijd
De gebruiker kan het afwijkend formulier type van een veld aanpassen door er op te klikken.	Could	Klikken om menu te tonen	2
		Kiezen formulier type	1
		Beïnvloeding andere velden	2
		Doorvoeren wijzigingen naar model	1

Referenties

Brennan, K. (2009). *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis. Opgeroepen op 6 6, 2014

Bijlage D: Use Cases

Hieronder is per onderdeel de verschillende toegevoegde functionaliteiten weergegeven. Voor elke functionaliteit is beschreven hoe het zou moeten werken. Door de geïmplementeerde functionaliteit te vergelijken met de use cases kan worden bepaald of het project voldoet aan de eisen.

Basis Grafische Modeler

Property Lijst: Aanpassen Waarde

Verloop

1. De gebruiker past een waarde aan in de property lijst
2. Het programma past de waarde in de controller aan.
3. De modeler wordt ververst met de nieuwe instelling.

Precondities

De gebruiker is werkzaam in de lijst of formulier modeler

Postcondities

De gewijzigde instelling is opgeslagen en de relevante modeler is ververst.

Grafische Modeler: Veranderen Tabel

Verloop

1. De gebruiker kiest een andere tabel.
2. De modeler wordt ververst met de nieuwe inhoud.

Extensies

1a. De tabel heeft geen inhoud.

Precondities

De gebruiker is werkzaam in de lijst of formulier modeler

Postcondities

De relevante modeler is ververst om de lijst of het formulier van de nieuwe geselecteerde tabel te tonen.

Lijst Modeler

Lijst: Instellen Kolom Breedte

Verloop

1. De gebruiker sleept de rand van een kolom in de lijst modeler.
2. Het programma past de waarde van de kolom breedte in de controller aan.
3. De lijst modeler wordt ververst met de nieuwe kolom breedte.

Extensies

1b. De rand wordt naar een negatieve waarde gesleept.

Precondities

De gebruiker is werkzaam in de lijst modeler.

Postcondities

De breedte van de kolom is aangepast naar hoe breed de gebruiker het heeft gesleept.

Lijst: Instellen Rij Hoogte**Verloop**

1. De gebruiker sleept de rand van een rij in de lijst modeler.
2. Het programma past de waarde van de rij hoogte in de controller aan.
3. De lijst modeler wordt ververst met de nieuwe rij hoogte.

Extensies

1a. De rand wordt naar een negatieve waarde gesleept.

Precondities

De gebruiker is werkzaam in de lijst modeler.

Postcondities

De hoogte van de rij is aangepast naar hoe hoog de gebruiker het heeft gesleept.

Lijst: Instellen Aggregatie**Verloop**

1. De gebruiker kiest voor een kolom een aggregatie type in de combobox.
2. Het programma past het aggregatie type in de controller aan.
3. De lijst modeler wordt ververst met de nieuwe aggregatie.

Extensies

3a. De aggregatie is niet van toepassing op het type kolom.

Precondities

De gebruiker is werkzaam in de lijst modeler.

Postcondities

Het type aggregatie is aangepast naar het type dat de gebruiker heeft geselecteerd.

Lijst: Instellen Wijzigingsinstellingen**Verloop**

1. De gebruiker opent het instellingen scherm aan de zijkant van de lijst modeler.
2. De gebruiker wijzigt de checkbox van de gewenste instelling.
3. Het programma past de waarde van instelling in de controller aan.
4. De lijst modeler wordt ververst met de nieuwe instelling.

Precondities

De gebruiker is werkzaam in de lijst modeler.

Postcondities

De wijzigingsinstelling is aangepast naar wat de gebruiker heeft geselecteerd.

Formulier Modeler**Formulier: Instellen Label Breedte****Verloop**

1. De gebruiker sleept de rand van een label in de formulier modeler.
2. Het programma past de waarde van de label breedte van het veld in de controller aan.
3. De formulier modeler wordt ververst met de nieuwe label breedte.

Extensies

1a. De rand wordt naar een negatieve waarde gesleept.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Postcondities

De breedte van het label is aangepast naar hoe breed de gebruiker het heeft gesleept.

Formulier: Instellen Veld Formaat**Verloop**

1. De gebruiker sleept de rand van een veld in de formulier modeler.
2. Het programma past de waarde van de veld hoogte of breedte van het veld in de controller aan.
3. De formulier modeler wordt ververst met het nieuwe veld formaat.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Postcondities

Het formaat van het veld is aangepast naar hoe groot de gebruiker het heeft gesleept.

Formulier: Toevoegen Groep**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het toevoegen van een groep.
3. De gebruiker voert een naam voor de groep in.
4. Het programma past de waarde voor de groep van het veld in de controller aan.
5. De formulier modeler wordt ververst met de nieuwe groep.

Extensies

2a. Het veld bevindt zich al in een eigen groep.

3a. De gebruiker voert geen naam in.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Het geselecteerde veld zit nog niet in een eigen groep.

Postcondities

Het veld is toegevoegd aan een nieuwe groep.

Formulier: Verwijderen Groep**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het verwijderen van een groep.
3. Het programma past de waarde voor de groep van het veld in de controller aan.
4. De formulier modeler wordt ververst met de groep verwijderd.

Extensies

2a. Het veld bevindt zich nog niet in een eigen groep.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Het geselecteerde veld zit al in een eigen groep.

Postcondities

De groep van het veld is verwijderd.

Formulier: Instellen Label Groep**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het wijzigen van het label van een groep.
3. De gebruiker voert een naam voor de groep in.
4. Het programma past de waarde voor het label van de groep van het veld in de controller aan.
5. De formulier modeler wordt ververs met de nieuwe naam voor de groep.

Extensies

3a. De gebruiker voert geen naam in.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Het geselecteerde veld zit al in een eigen groep.

Postcondities

Het label van de groep is aangepast naar wat de gebruiker heeft ingevoerd.

Formulier: Instellen Icoon Groep**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het wijzigen van het icoon van een groep.
3. De gebruiker voert een directory voor het icoon voor de groep in.
4. Het programma past de waarde voor het icoon van de groep van het veld in de controller aan.
5. De formulier modeler wordt ververs met het nieuwe icoon voor de groep.

Extensies

3a. De gebruiker voert geen directory in.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Het geselecteerde veld zit al in een eigen groep.

Postcondities

Het icoon van de groep is aangepast naar wat de gebruiker heeft ingevoerd.

Formulier: Toevoegen Tabblad**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het toevoegen van een tabblad.
3. Het programma past de waarde voor het tabblad van het veld in de controller aan.
4. De formulier modeler wordt ververs met het nieuwe tabblad.

Extensies

2a. Het veld bevindt zich al in een eigen tabblad.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Het geselecteerde veld zit nog niet in een eigen tabblad.

Postcondities

Het veld is toegevoegd aan een nieuw tabblad.

Formulier: Verwijderen Tabblad**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het verwijderen van een tabblad.
3. Het programma past de waarde voor het tabblad van het veld in de controller aan.
4. De formulier modeler wordt ververst met het tabblad verwijderd.

Extensies

2a. Het veld bevindt zich nog niet in een eigen tabblad.

Precondities

De gebruiker is werkzaam in de formulier modeler.
Het geselecteerde veld zit al in een eigen tabblad.

Postcondities

Het tabblad van het veld is verwijderd.

Formulier: Instellen Label Tabblad**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het wijzigen van het label van een tabblad.
3. De gebruiker voert een naam voor het tabblad in.
4. Het programma past de waarde voor het label van het tabblad van het veld in de controller aan.
5. De formulier modeler wordt ververst met de nieuwe naam voor het tabblad.

Extensies

3a. De gebruiker voert geen naam in.

Precondities

De gebruiker is werkzaam in de formulier modeler.
Het geselecteerde veld zit al in een eigen tabblad.

Postcondities

Het label van het tabblad is aangepast naar wat de gebruiker heeft ingevoerd.

Formulier: Instellen Icoon Tabblad**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het wijzigen van het icoon van een tabblad.
3. De gebruiker voert een directory voor het icoon voor het tabblad in.
4. Het programma past de waarde voor het icoon van het tabblad van het veld in de controller aan.
5. De formulier modeler wordt ververst met het nieuwe icoon voor het tabblad.

Extensies

3a. De gebruiker voert geen directory in.

Precondities

De gebruiker is werkzaam in de formulier modeler.
Het geselecteerde veld zit al in een eigen tabblad.

Postcondities

Het icoon van het tabblad is aangepast naar wat de gebruiker heeft ingevoerd.

Formulier: Toevoegen Kolom**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het toevoegen van een kolom.
3. Het programma past de waarde voor de kolom van het veld in de controller aan.
4. De formulier modeler wordt ververst met het veld in de nieuwe kolom.

Extensies

2a. Het veld bevindt zich al in een nieuwe kolom.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Postcondities

Het veld is toegevoegd aan een nieuwe kolom.

Formulier: Verwijderen Kolom**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het verwijderen van een kolom.
3. Het programma past de waarde voor de kolom van het veld in de controller aan.
4. De formulier modeler wordt ververst met de kolom verwijderd.

Extensies

2a. Het veld bevindt zich niet in een nieuwe kolom.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Postcondities

De kolom van het veld is verwijderd.

Formulier: Invoegen Lege Regel**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het toevoegen van een lege regel.
3. Het programma past de waarde voor de lege regels van het veld in de controller aan.
4. De formulier modeler wordt ververst met het veld op de volgende regel.

Precondities

De gebruiker is werkzaam in de formulier modeler.

Postcondities

Het veld is een regel naar beneden verplaatst op het formulier.

Formulier: Verwijderen Lege Regel**Verloop**

1. De gebruiker klikt op een veld in de formuliermodeler.
2. De gebruiker kiest in het contextmenu het verwijderen van een lege regel.
3. Het programma past de waarde voor de lege regels van het veld in de controller aan.
4. De formulier modeler wordt ververst met de kolom verwijderd.

Extensies

1a. Het veld heeft nog geen lege regels.

Precondities

De gebruiker is werkzaam in de formulier modeler..
Het geselecteerde veld heeft al minstens één lege regel.

Postcondities

Het veld is een regel omhoog verplaatst op het formulier.

Formulier: Instellen Wijzigingsinstellingen**Verloop**

1. De gebruiker opent het instellingen scherm aan de zijkant van de formulier modeler.
2. De gebruiker wijzigt de checkbox van de gewenste instelling.
3. Het programma past de waarde van de instelling in de controller aan.
4. De formulier modeler wordt ververst met de nieuwe instelling.

Precondities

De gebruiker is werkzaam in de formulier modeler.

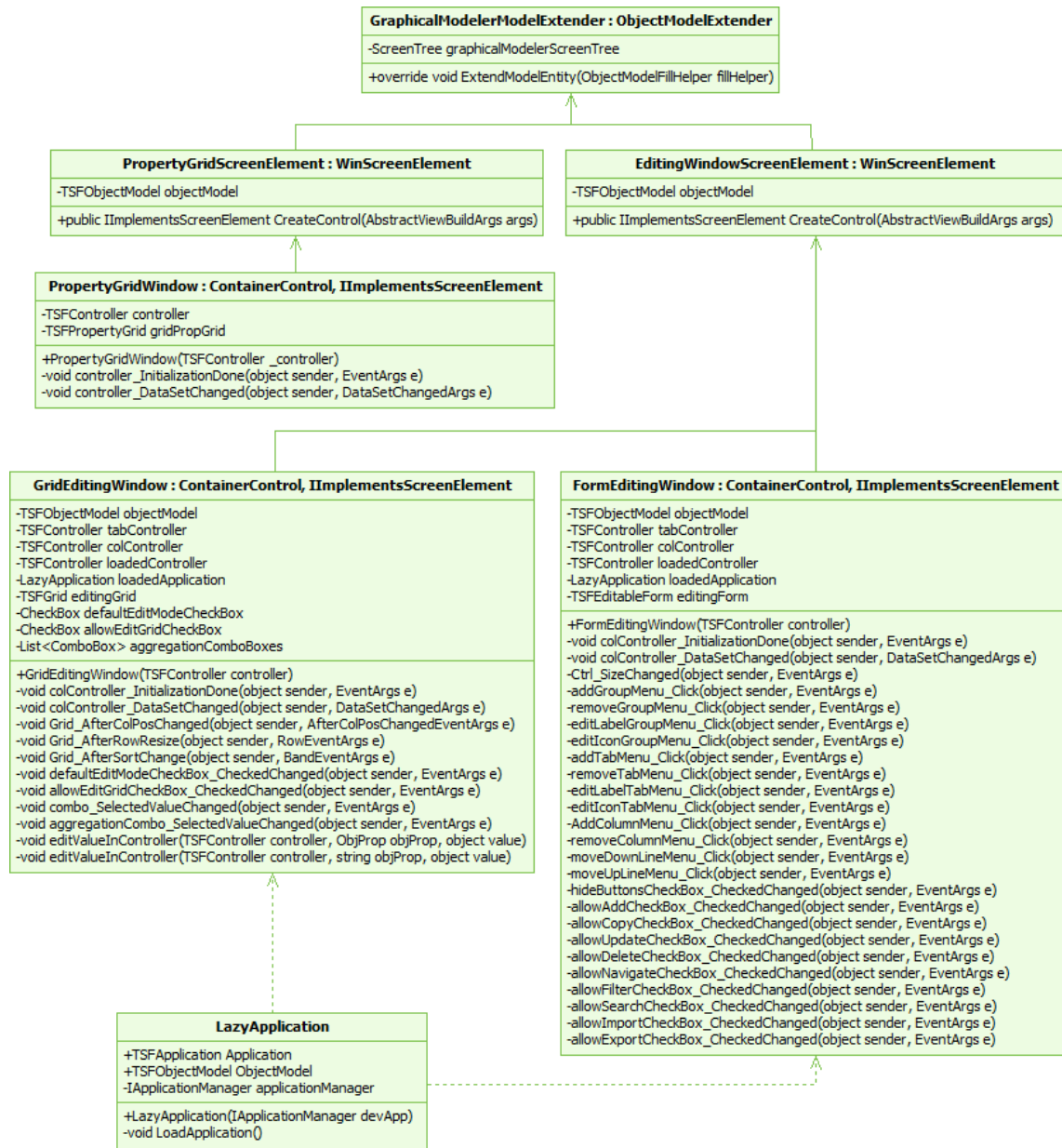
Postcondities

De wijzigingsinstelling is aangepast naar wat de gebruiker heeft geselecteerd.

Bijlage E: Technische Beschrijving Implementatieproces

Ontwerp

Hieronder in figuur 1 is het klassendiagram van de grafische modeler te zien. De basis van het project bestaat uit de GraphicalModelerModelExtender. Deze bevat twee schermelementen, het PropertyGridScreenElement en het EditingWindowScreenElement.



Figuur 1. Het klassendiagram dat een overzicht geeft van de structuur van het gerealiseerde product.

In het PropertyGridScreenElement wordt het PropertyGridWindow weergegeven. Dit is het scherm component dat de property lijst toont met alle aanpasbare instellingen van de lijst of het formulier.

Het `EditingWindowScreenElement` kan twee verschillende schermen tonen. Afhankelijk van of de lijst of het formulier wordt aangepast wordt hierin het `GridEditingWindow` of het `FormEditingWindow` getoond. In beide schermen wordt gebruik gemaakt van een `LazyApplication`. Hiermee wordt de eindapplicatie ingeladen vanuit de database, om zo de lijst of het formulier te tonen zoals het in de eindapplicatie er uit zal komen te zien.

In beide van deze schermen is de functionaliteit geïmplementeerd om grafische aanpassingen te kunnen maken aan de lijst of het formulier. De gewijzigde aanpassingen worden vervolgens opgeslagen in de database en getoond in de grafische modeler.

Basis Grafische Modeler

De Software Factory bestaat uit een model en een GUI. Het model is opgeslagen in de database en bevat de data van de Software Factory. De Software Factory GUI bestaat uit Microsoft C# code en wordt beheerd met behulp van het versiebeheersysteem SVN.

Voordat is begonnen met de implementatie van de grafische modeler, is in SVN een branch aangemaakt voor het project van de grafische modeler. Dit zorgt ervoor dat de code geïsoleerd blijft van de rest van het project. Het kan dan vervolgens later, wanneer het project succesvol is gerealiseerd, in de hoofdversie van de Software Factory GUI worden opgenomen.

Vervolgens is onderzocht welke handelingen uitgevoerd moeten worden om de basis van de grafische modeler te implementeren. Hierin wordt aan de linkerkant de lijst van eigenschappen weergegeven van de lijst of het formulier dat wordt bewerkt. Aan de rechterkant wordt, afhankelijk van of de lijst of het formulier wordt aangepast, weergegeven hoe de lijst of het formulier er uit zal zien in het eindproduct. Door middel van toelichting van de bedrijfsbegeleiders en door analyse van de code, is onderzocht wat er uitgevoerd moet worden om dit te implementeren.

Omdat de Software Factory zelf een project is dat ontwikkeld is met de Software Factory, is dit project ook aangepast, om de benodigde schermen weer te geven in de Software Factory. Dit zorgt ervoor dat de grafische modeler kan worden gebruikt binnen de Software Factory.

Voor de implementatie van de basis grafische modeler is eerst een model extender aangemaakt. Dit is een losstaand onderdeel dat kan worden ingeladen in de Software Factory GUI om extra functionaliteit toe te voegen. Door de gehele grafische modeler te implementeren in een model extender, blijft de functionaliteit in een apart project, zonder dat dit invloed heeft op de rest van de code van de Software Factory GUI. De model extender is dan simpelweg een losse DLL die kan worden toegevoegd aan een project zoals de Software Factory. Deze model extender bevindt zich in een losstaand Microsoft C# project.

Vervolgens zijn in de model extender schermelementen toegevoegd die de property lijst en de lijst en formulier modeler weergeven. Een schermelement is een deel van het scherm binnen de Software Factory dat data uit een tabel op bepaalde wijze weergeeft. Door eigen schermelementen aan te maken kunnen de lijst en formulier modeler worden ontwikkeld met unieke functionaliteit.

Hierna is begonnen met het weergeven van de property lijst aan de linkerzijde van de grafische modeler. Door deze te implementeren, zijn nog steeds alle aangepaste waardes duidelijk zichtbaar en kunnen op de oude wijze de aanpassingen ook nog worden uitgevoerd.

Om de wijzigingsinstellingen voor de lijst en het formulier weer te geven, is een auto hide scherm toegevoegd waarin verschillende instellingen kunnen worden gewijzigd. Dit scherm verbergt zich automatisch aan de zijkant van het scherm, zodat het geen extra ruimte van de lijst of formulier modeler inneemt. Het is echter wel mogelijk om dit scherm constant zichtbaar te hebben.

Binnen de model extender is daarna begonnen met het kunnen weergeven van hoe de lijst of het formulier er in de eindapplicatie uit zal komen te zien in de lijst en formulier modeler. In eerste instantie is geprobeerd om dit uit te voeren door de benodigde informatie uit het huidig geopende project in de Software Factory te halen. Dit bleek echter voor een aantal problemen te zorgen. Niet alle benodigde informatie bleek op eenvoudige wijze uit het huidige project te kunnen worden gehaald. Om dit op deze wijze verder uit te kunnen voeren, zou een groot deel van de huidige Software Factory code moeten worden aangepast.

Daarom is besloten om in plaats hiervan verbinding te maken met de database, waarin de daadwerkelijke eindapplicatie is opgeslagen. Deze applicatie wordt dan deels ingeladen, waarna de benodigde data voor het wijzigen van de lijst of het formulier kan worden opgehaald.

Met behulp van de data in het eindproduct, kunnen de lijst en het formulier precies zo worden weergegeven zoals ze er in het eindproduct uit zullen zien. Het maken van wijzigingen met behulp van de property lijst zorgt ervoor dat deze wijzigingen worden opgeslagen in de database. Ook zorgt het ervoor dat de lijst of formulier modeler wordt aangepast, om de wijzigingen zichtbaar te maken.

Na de voltooiing van de realisatie van de basis grafische modeler, is de functionaliteit hiervan getest. De problemen die nog voorkomen in de basis grafische modeler zijn vastgelegd in de lijst met issues in [bijlage G](#). Ook is gedocumenteerd welke handelingen zijn uitgevoerd en hoe de geïmplementeerde functionaliteit werkt.

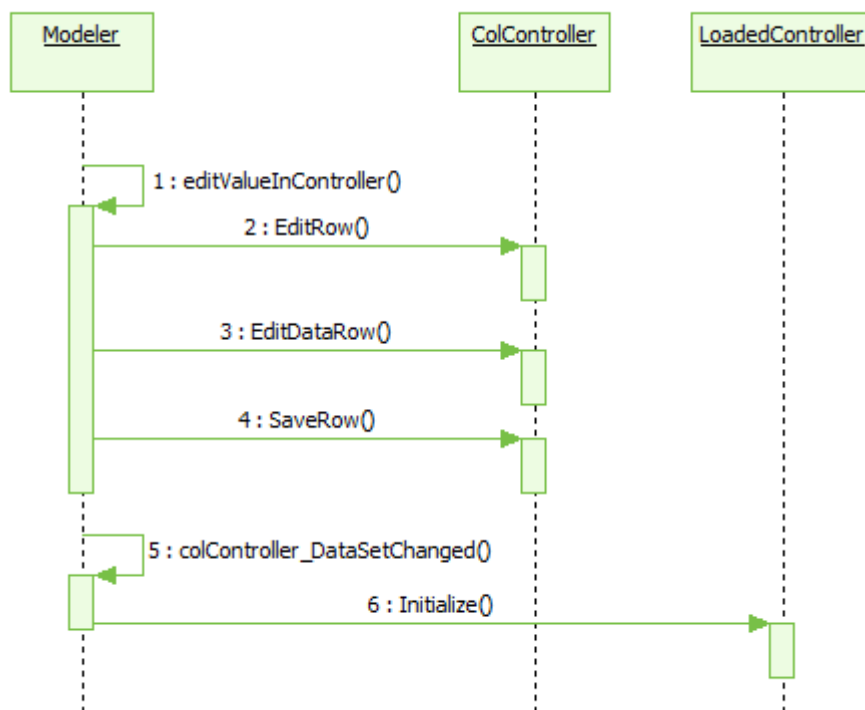
Aan het einde van deze fase is een demonstratie gegeven van de functionaliteit van de basis grafische modeler. Deze demonstratie is uitgevoerd voor de opdrachtgever en de bedrijfsbegeleiders.

Lijst Modeler

Na de voltooiing van de basis grafische modeler, is begonnen met de implementatie van de lijst modeler. Ook hier is weer eerst begonnen met onderzoek, om er achter te komen wat er uitgevoerd moet worden om de lijst modeler te kunnen implementeren. Hiervoor is weer toelichting door de bedrijfsbegeleiders gegeven en verdere analyse van de code uitgevoerd.

De functionaliteit van de lijst modeler is weer geïmplementeerd in hetzelfde Microsoft C# project en in dezelfde model extender.

Er is begonnen met het ervoor zorgen dat wijzigingen die in het lijst modeler worden gemaakt aan de eindapplicatie, worden doorgevoerd naar de Software Factory. Hierdoor kunnen de aangepaste instellingen voor de applicatie worden opgeslagen in de database. Alle verschillende instellingen die worden gewijzigd kunnen zo simpelweg via dezelfde methode worden aangepast in de database. Ook wordt de grafische modeler dan verversd om de wijziging te tonen. Deze manier van het aanpassen van instellingen is ook gebruikt voor de formulier modeler. Het proces is beschreven in het sequence diagram in figuur 2.



Figuur 2. Het sequence diagram voor het opslaan van wijzigingen naar de database en verversen van de grafische modeler.

Vervolgens is de functionaliteit om de kolombreedte aan te kunnen passen geïmplementeerd. Door het verslepen van de randen van de kolommen, wordt ingesteld dat die kolom een aangepaste breedte heeft en wordt de daadwerkelijke breedte opgeslagen in de database. Ook wordt de weergegeven lijst aangepast om de wijziging te tonen.

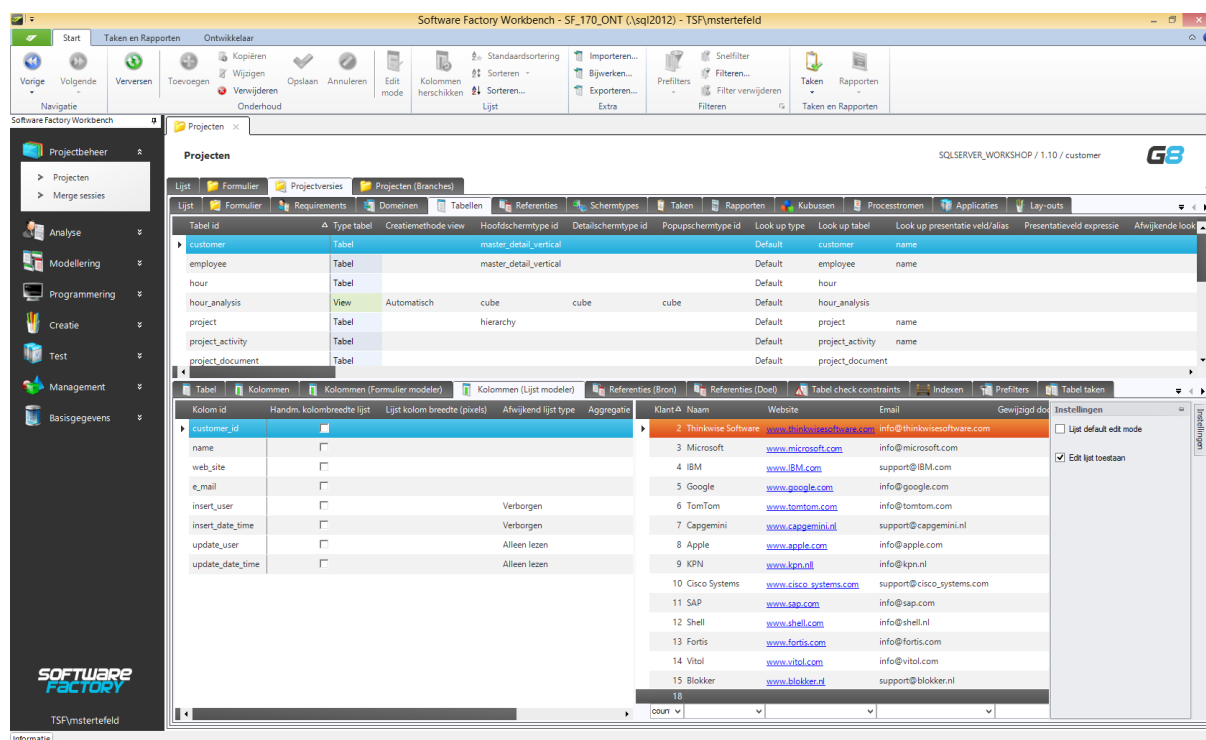
Hierna is het kunnen instellen van een aggregatie van een kolom toegevoegd. Door middel van een menu aan de onderkant van de kolom ingesteld kan worden welk type aggregatie er uitgevoerd moet worden op die kolom. Vervolgens wordt de aggregatie toegepast en opgeslagen in de database.

Aan het auto hide scherm is vervolgens een aantal functies toegevoegd. Eerst is het mogelijk gemaakt om voor de lijst in te stellen of deze in het eindproduct te wijzigen is of niet. Vervolgens is het instellen of de lijst standaard op edit modus staat toegevoegd. In edit modus kunnen direct de waarden in de lijst worden bewerkt. Deze instellingen zijn boolean waarden en kunnen dus met behulp van een checkbox simpelweg worden in en uitgeschakeld.

Nadat de implementatie van de lijst modeler is afgerond, zijn weer tests uitgevoerd om de functionaliteit te bevestigen. Ook zijn eventuele problemen met de lijst modeler vast gelegd in de lijst met issues in [bijlage G](#). Verder zijn de uitgevoerde handelingen weer gedocumenteerd en de toegevoegde functionaliteit vastgelegd.

Ook hier is aan het einde van deze ontwikkelingsfase een demonstratie gegeven aan de opdrachtgever en de bedrijfsbegeleiders, waar de functionaliteit van de lijst modeler is getoond.

Hieronder in figuur 3 is te zien hoe de lijst modeler er uit ziet in de Software Factory.



Figuur 3. De lijst modeler geïmplementeerd in de Software Factory. Aan de rechterkant is het auto hide scherm met instellingen te zien.

Formulier Modeler

Na het realiseren van de lijst modeler, is het laatste onderdeel van de grafische modeler begonnen, de formulier modeler. Hier is weer begonnen met te onderzoeken naar de uit te voeren handelingen om de formulier modeler te implementeren, door code analyse en toelichting van de bedrijfsbegeleiders.

Er is eerst begonnen met het organiseren van de code. Omdat alle code tot dusver in dezelfde klasse zat, zorgde het voor een gebrek aan overzicht. De code is opgedeeld in verschillende onderdelen voor de model extender, de property lijst, de lijst modeler en de formulier modeler. Hierdoor blijft de code voor elk onderdeel geïsoleerd. Alle code bevindt zich nog wel binnen het zelfde Microsoft C# project.

Voor de implementatie van de formulier modeler is eerst het aanpassen van velden en labels gerealiseerd. Er is functionaliteit toegevoegd dat het mogelijk maakt dat de zijkant van labels en van velden te verslepen is. Het veld of label neemt vervolgens het formaat aan waar naar toe is gesleept. Voor deze functionaliteit is gebruik gemaakt van het voorbeeld *“Move and Resize Controls on a Form at Runtime”* (Zomorrod, 2014)

Vervolgens is het auto hide scherm voor de formulier modeler geïmplementeerd. Hierin is het mogelijk om een aantal instellingen voor een formulier te wijzigen. Er kan worden ingesteld of de wijzigingsknoppen zichtbaar zijn. Ook kan voor elke functie van de wijzigingsknoppen worden ingesteld op de handeling mag worden uitgevoerd. Deze instellingen zijn: Toevoegen, kopiëren, wijzigen, verwijderen, navigeren, filteren, zoeken, importeren en exporteren. Al deze waardes zijn ook weer boolean waardes, waardoor ze met checkboxes kunnen worden in en uitgeschakeld.

Hierna is begonnen met tonen van een menu wanneer op een veld of label wordt geklikt. In dit menu zijn een aantal opties beschikbaar om de verschillende velden binnen het formulier in te kunnen delen. De eerste optie is om een veld toe te voegen aan een groep. De tweede optie is om een veld toe te voegen aan een nieuw tabblad. De derde optie is om velden in nieuwe kolommen op het formulier te zetten, of ze uit de kolom te verwijderen. De laatste optie is om lege regels tussen velden op het formulier in te voegen of te verwijderen.

Vervolgens is begonnen met het toevoegen van functionaliteit aan de opties binnen dit menu. Eerst is de mogelijkheid toegevoegd om velden regels over te laten slaan op het formulier, zodat er lege regels tussen dat veld en het vorige veld staat. Ook de mogelijkheid toegevoegd om deze lege regels weer te verwijderen. Verder is de mogelijkheid toegevoegd om velden in een volgende kolom op het formulier te tonen en om het veld weer aan de vorige kolom toe te voegen.

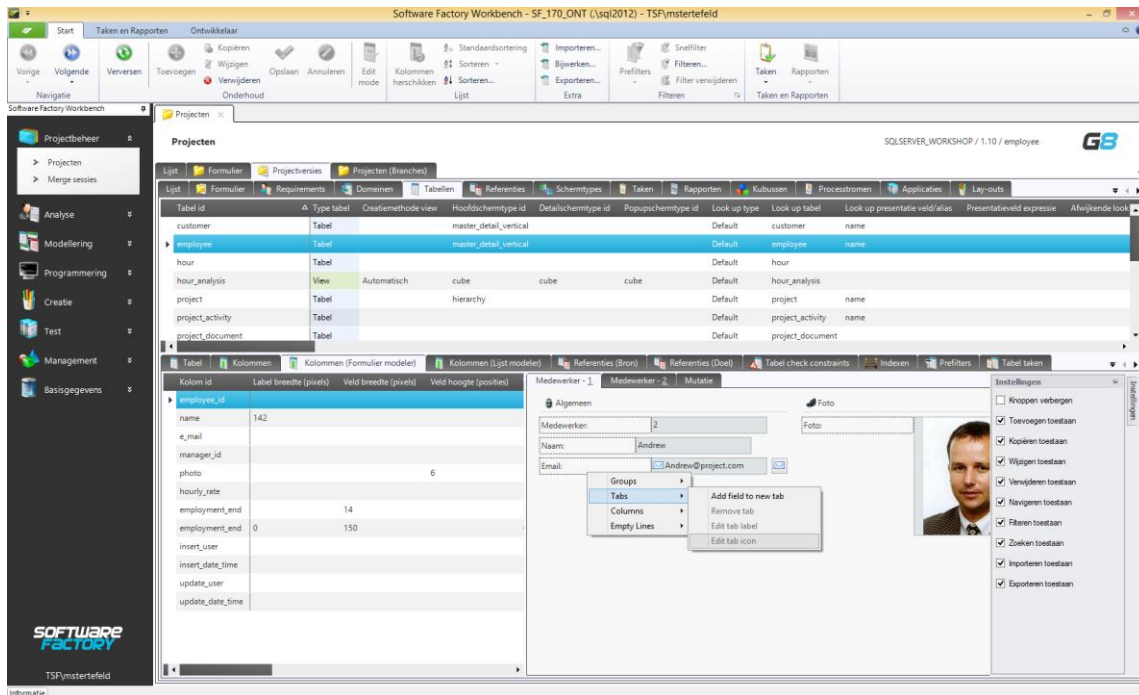
Daarna is begonnen met het indelen van velden in groepen en tabbladen. Eerst is de functionaliteit toegevoegd om velden te kunnen toevoegen aan een nieuwe tabblad en ze hier weer van te verwijderen. Vervolgens is dezelfde functionaliteit geïmplementeerd voor het toevoegen en verwijderen van velden in groepen. Er is een invoer dialoog toegevoegd om de verplichte titel van de groep in te kunnen vullen bij het toevoegen van een veld aan een groep.

Vervolgens is met behulp van het invoer dialoog de mogelijkheid toegevoegd om van groepen en tabbladen de titel te kunnen wijzigen. Ook is met dit dialoog het mogelijk gemaakt om de iconen van de groepen en tabbladen te wijzigen. De iconen worden opgeslagen als directories, waardoor simpelweg de directory van het icoon wordt ingevoerd.

Na de afronding van het implementeren van de formulier modeler, is de functionaliteit van de formulier modeler getest. De problemen zijn vastgelegd in de issue lijst in [bijlage G](#). Ook is weer vastgelegd welke handelingen zijn uitgevoerd en hoe de toegevoegde functionaliteit werkt.

Aan het einde van deze fase is weer gedemonstreerd aan de bedrijfsbegeleiders en opdrachtgever om de functionaliteit van de formulier modeler te tonen.

Hieronder in figuur 4 is te zien hoe de formulier modeler in de Software Factory er uit ziet.



Figuur 4. De formulier modeler geïmplementeerd in de Software Factory. Aan de rechterkant is het auto hide scherm met instellingen te zien. Ook is te zien hoe een tabblad aangepast kan worden.

Referenties

Zomorrod. (2014, 1 13). *Move and Resize Controls on a Form at Runtime (With Mouse)*.
Opgehaald van CodeProject: <http://www.codeproject.com/Tips/709121/Move-and-Resize-Controls-on-a-Form-at-Runtime-With>

Bijlage F: Testresultaten

Om te controleren of de functionaliteit van de grafische modeler voldoet aan de gestelde eisen, zijn aan de hand van de use cases beschreven in [bijlage C](#) tests uitgevoerd. Voor elke use case zijn er een met behulp van black-box testing een aantal scenario's uitgevoerd. De testresultaten voor elke use case is hier beschreven.

Basis Grafische Modeler

Property lijst: Aanpassen Waarde

De lijst modeler is geopend binnen het project. In de property lijst worden de verschillende soorten waardes aangepast voor de lijst. De wijzigingen worden opgeslagen en de lijst modeler wordt ververst met de nieuwe instellingen. Hierna is de formulier modeler getest. Ook hier zijn weer de verschillende soorten waardes voor het formulier aangepast in de property lijst. De aangepaste instellingen worden weer opgeslagen en de formulier modeler is ververst met de nieuwe instellingen

Hiermee is aangetoond dat de functionaliteit van het aanpassen van een waarde in de property lijst naar de eisen voldoet.

Grafische Modeler: Veranderen Tabel

De lijst modeler is geopend binnen het project. Standaard wordt hierin de eerste tabel geselecteerd. Het wisselen van de tabel zorgt ervoor dat de lijst modeler de nieuwe lijst toont. Ook het aanpassen van de waardes in de property lijst beïnvloedt de nieuw geselecteerde tabel. Hierna is de functionaliteit van de formulier modeler getest. Hier wordt weer de inhoud getoond van de geselecteerde tabel. Het wisselen van tabellen zorgt er weer voor dat de getoonde formulier verandert naar die van de nieuwe tabel. Het maken van aanpassingen beïnvloedt ook weer de nieuw geselecteerde tabel.

Dit toont aan dat de functionaliteit van het veranderen van geselecteerde tabel voldoet aan de eisen in de use case.

Lijst Modeler

Lijst: Instellen Kolom Breedte

De lijst modeler is geopend binnen het project. Hierin wordt van verschillende kolommen de breedte aangepast. Er wordt onder andere geprobeerd om de breedte negatief te maken. De breedte van de kolom wordt juist aangepast naar de versleepte breedte. Ook wordt de breedte van de kolom correct opgeslagen.

Hierdoor blijkt dat de functionaliteit van het aanpassen van de kolom breedte in de lijst modeler voldoet aan de eisen.

Lijst: Instellen Rij Hoogte

De functionaliteit van het aanpassen van de rij hoogte is nog niet geïmplementeerd. Daarom voldoet dit niet aan de eisen.

Lijst: Instellen Aggregatie

De lijst modeler is geopend binnen het project. Voor alle kolommen is een aggregatie combobox weergegeven. Deze box heeft dezelfde breedte als de kolom zelf. In de combobox wordt de waarde aangepast naar een vorm van aggregatie. Alle mogelijke aggregatietypes worden uitgetest. De aggregatie wordt succesvol opgeslagen. De lijst wordt echter nog niet succesvol verversd om de aggregatie te tonen. Het opnieuw openen van het project toont wel de juiste aggregatie.

De functionaliteit van het instellen van de aggregatie is hierdoor dus nog niet compleet geïmplementeerd en voldoet nog niet volledig aan de eisen.

Lijst: Wijzigingsinstellingen

De lijst modeler is geopend binnen het project. Door de muis over de instellingen knop te houden wordt het auto hide scherm getoond. Het verplaatsen van de muis buiten het instellingen scherm zorgt ervoor dat het weer verdwijnt. Door op de pin knop te drukken blijft het scherm continu zichtbaar. Binnen het auto hide scherm zijn de verschillende functionaliteiten getest. Eerst is getest om de default edit mode van de lijst in en uit te schakelen. Deze instelling wordt correct opgeslagen in de database. Hierna is het toestaan van het editen van de lijst in en uit geschakeld. Ook deze instelling wordt correct opgeslagen.

Dit toont dat de functionaliteit van de wijzigingsinstellingen in het auto hide scherm correct is geïmplementeerd.

Formulier Modeler

Formulier: Instellen Label Breedte

De formulier modeler is geopend binnen het project. De randen van een label worden weergegeven door middel van een stippellijn. Door de muis over de rechter zijkant van een label te slepen, verandert de cursor naar een breedte aanpassingscursor. Door vervolgens te klikken en te slepen wordt de breedte van het label aangepast. De waarde wordt correct opgeslagen in de database en de modeler wordt verversd met de nieuwe waarde. Het verslepen van een veld naar een negatieve waarde is niet mogelijk.

Hiermee is aangetoond dat de functionaliteit van de aanpasbare label breedte in het formulier aan de eisen voldoet.

Formulier: Instellen Veld Formaat

De formulier modeler is geopend binnen het project. Door de muis te bewegen over de randen aan de onder- en zijkant van het veld verandert de cursor naar een hoogte of een breedte aanpassingscursor. Door het klikken en verslepen van de breedte of de hoogte wordt deze waarde aangepast en opgeslagen in de database. De modeler wordt ook verversd om de nieuwe waarde te tonen. Het verslepen naar een negatieve waarde is niet mogelijk. Het verslepen van de hoogte gebeurt in regels. De hoogte wordt afgerond naar een volledige regel hoogte.

Dit geeft aan dat geïmplementeerde functionaliteit van het aanpassen van het formaat van velden naar de eisen is geïmplementeerd.

Formulier: Toevoegen Groep

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor groepen. Met een veld nog niet in een eigen groep is alleen de toevoegen optie beschikbaar. Door er op te klikken verschijnt er een dialoog. In het dialoog moet het label worden ingevoerd. Als er niets wordt ingevuld of als op cancel wordt gedrukt in het menu sluit het menu af en wordt er geen groep toegevoegd. Wanneer er wel iets wordt ingevuld wordt de groep succesvol toegevoegd. De groep wordt opgeslagen in de database en de modeler wordt ververst om de nieuwe groep te tonen.

De functionaliteit van het toevoegen van een groep is dus gerealiseerd naar de eisen.

Formulier: Verwijderen Groep

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor groepen. Met een veld in een eigen groep is de verwijderen optie beschikbaar. Door er op te klikken wordt de groep verwijderd. De groep en diens label worden verwijderd in de database. De modeler wordt ververst om de groep niet meer te tonen.

Hiermee voldoet de functionaliteit die is geïmplementeerd van het verwijderen van een groep aan de gestelde eisen.

Formulier: Instellen Label Groep

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor groepen. Met een veld in een eigen groep is de optie voor het wijzigen van het label beschikbaar. Door er op te klikken verschijnt er een dialoog. In dit dialoog kan het label worden ingevoerd. Als er niets wordt ingevuld of als op cancel wordt gedrukt in het menu sluit het menu af en wordt het label niet gewijzigd. Wanneer er wel iets wordt ingevuld wordt het label van de groep succesvol gewijzigd. Het label van de groep wordt opgeslagen in de database en de modeler wordt ververst om het nieuwe label te tonen.

De functionaliteit van het instellen van een groepslabel is hiermee geïmplementeerd naar de eisen.

Formulier: Instellen Icoon Groep

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor groepen. Met een veld in een eigen groep is de optie voor het wijzigen van het icoon beschikbaar. Door er op te klikken verschijnt er een dialoog. In dit dialoog kan de directory voor het icoon worden ingevoerd. Als er niets wordt ingevuld of als op cancel wordt gedrukt in het menu sluit het menu af en wordt het icoon niet gewijzigd. Wanneer er wel iets wordt ingevuld wordt het icoon van de groep succesvol gewijzigd. Het icoon van de groep wordt opgeslagen in de database en de modeler wordt ververst om het nieuwe icoon te tonen.

Dit toont dat de functionaliteit van het instellen van het icoon van een groep voldoet aan de eisen.

Formulier: Toevoegen Tabblad

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor tabbladen. Met een veld nog niet in een eigen tabblad is alleen de toevoegen optie beschikbaar. Door er op te klikken wordt het tabblad toegevoegd. Het tabblad wordt opgeslagen in de database en de modeler wordt ververst om het nieuwe tabblad te tonen.

De functionaliteit van het toevoegen van een tabblad is dus gerealiseerd naar de eisen.

Formulier: Verwijderen Tabblad

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor tabbladen. Met een veld in een eigen tabblad is de verwijderen optie beschikbaar. Door er op te klikken wordt het tabblad verwijderd. Het tabblad en diens eventuele label worden verwijderd in de database. De modeler wordt ververst om het tabblad niet meer te tonen.

Hiermee voldoet de functionaliteit die is geïmplementeerd van het verwijderen van een tabblad aan de gestelde eisen.

Formulier: Instellen Label Tabblad

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor tabbladen. Met een veld in een eigen tabblad is de optie voor het wijzigen van het label beschikbaar. Door er op te klikken verschijnt er een dialoog. In dit dialoog kan het label worden ingevoerd. Als er niets wordt ingevuld of als op cancel wordt gedrukt in het menu sluit het menu af en wordt het label niet gewijzigd. Wanneer er wel iets wordt ingevuld wordt het label van het tabblad succesvol gewijzigd. Het label van het tabblad wordt opgeslagen in de database en de modeler wordt ververst om het nieuwe label te tonen.

De functionaliteit van het instellen van het label van een tabblad is hiermee geïmplementeerd naar de eisen.

Formulier: Instellen Icoon Tabblad

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor tabbladen. Met een veld in een eigen tabblad is de optie voor het wijzigen van het icoon beschikbaar. Door er op te klikken verschijnt er een dialoog. In dit dialoog kan de directory voor het icoon worden ingevoerd. Als er niets wordt ingevuld of als op cancel wordt gedrukt in het menu sluit het menu af en wordt het icoon niet gewijzigd. Wanneer er wel iets wordt ingevuld wordt het icoon van het tabblad succesvol gewijzigd. Het icoon van het tabblad wordt opgeslagen in de database en de modeler wordt ververst om het nieuwe icoon te tonen.

Dit toont dat de functionaliteit van het instellen van het icoon van een tabblad voldoet aan de eisen.

Formulier: Toevoegen Kolom

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor kolommen. Met een veld nog niet in een eigen kolom is alleen de toevoegen optie beschikbaar. Door er op te klikken wordt het veld aan een nieuwe kolom toegevoegd. De kolom wordt opgeslagen in de database en de modeler wordt ververst om de nieuwe kolom te tonen.

De functionaliteit van het toevoegen van een kolom is dus gerealiseerd naar de eisen.

Formulier: Verwijderen Kolom

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor kolommen. Met een veld in een eigen kolom is de verwijderen optie beschikbaar. Door er op te klikken wordt de kolom verwijderd. De kolom wordt verwijderd in de database. De modeler wordt ververst om de kolom niet meer te tonen.

Hiermee voldoet de functionaliteit die is geïmplementeerd van het verwijderen van een kolom aan de gestelde eisen.

Formulier: Invoegen Lege Regel

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor lege regels. Hier binnen is de toevoegen optie beschikbaar. Door er op te klikken wordt een regel overgeslagen voor het veld. Wanneer er al lege regels worden weergegeven, wordt een extra lege regel toegevoegd. De lege regel wordt opgeslagen in de database en de modeler wordt ververst om de lege regel te tonen.

De functionaliteit van het toevoegen van een lege regel is dus gerealiseerd naar de eisen.

Formulier: Verwijderen Lege Regel

De formulier modeler is geopend binnen het project. Door op een veld of label te klikken verschijnt een menu. In dit menu is een submenu voor lege regels. Met een veld dat lege regels er voor heeft is de verwijderen optie beschikbaar. Door er op te klikken wordt er een lege regel verwijderd. Wanneer er meerdere lege regels worden weergegeven, wordt er slechts één verwijderd. Deze optie kan herhaaldelijk worden uitgevoerd tot ze allemaal zijn verwijderd. De lege regel wordt verwijderd in de database. De modeler wordt ververst om de lege regel niet meer te tonen.

Hiermee voldoet de functionaliteit die is geïmplementeerd van het verwijderen van een lege regel aan de gestelde eisen.

Formulier: Instellen Wijzigingsinstellingen

De formulier modeler is geopend binnen het project. Door de muis over de instellingen knop te houden wordt het auto hide scherm getoond. Het verplaatsen van de muis buiten het instellingen scherm zorgt ervoor dat het weer verdwijnt. Door op de pin knop te drukken blijft het scherm continu zichtbaar. Binnen het auto hide scherm zijn de volgende functionaliteiten die zijn getest:

Binnen het scherm zijn de volgende mogelijkheden:

- Knoppen verbergen
- Toevoegen toestaan
- Kopiëren toestaan
- Wijzigen toestaan
- Verwijderen toestaan
- Navigeren toestaan
- Verwijderen toestaan
- Filteren toestaan
- Zoeken toestaan
- Importeren toestaan
- Exporteren toestaan

Het wijzigen van alle waardes zorgt ervoor dat het correct wordt opgeslagen in de database.

Dit toont dat de functionaliteit van de wijzigingsinstellingen in het auto hide scherm correct is geïmplementeerd.

Bijlage G: Issue Lijst

Basis Grafische Modeler

De splitter van de schermelementen, die er voor zorgt dat de property lijst en de lijst of formulier modeler beide worden weergegeven, heeft een incorrecte startpositie. De splitter staat volledig naar links geschoven, waardoor de property lijst niet zichtbaar is totdat de splitter positie wordt versleept.

De verschillende weergegeven teksten in de grafische modeler zijn nu nog hard coded in het project vastgelegd. Dit moet worden aangepast om het ingebouwde taalsysteem van de Software Factory wordt gebruikt.

Het maken van de verbinding met het eindproduct wordt nu direct gedaan, zonder dat de gebruiker daar wijzigingen aan kan maken. Door het connectiedialoog weer te geven kan de gebruiker nog eventuele veranderingen kunnen maken om met de database te verbinden.

De lijst en formulier modeler bevinden zich nu in de Software Factory in het project scherm. Deze moeten verplaatst worden naar het gebruikersinterface scherm.

Om het aanpassen van bepaalde eigenschappen mogelijk te maken, zijn er in enkele classes buiten het project van de grafische modeler wijzigingen gemaakt. Deze wijzigingen moeten geïntegreerd worden in het project, om conflicten met andere onderdelen te voorkomen.

Lijst Modeler

Het aanpassen van de rijhoogte in de lijst modeler is nog niet mogelijk om op grafische wijze te doen. Dit komt omdat in de implementatie van de lijst in de code van de Software Factory, het verslepen van de onderkant of bovenkant van een rij niet mogelijk is.

Het verslepen van kolommen in de lijst, om zo de volgorde aan te passen, is nog niet geïmplementeerd in de lijst modeler.

In de lijst modeler wordt de lijst weergegeven zoals deze in het eindproduct zal zijn. Dit is inclusief de data die de lijst zal vullen. Door slechts een beperkte hoeveelheid rijen in te laden in de lijst kan het laden van de lijst modeler worden versneld.

In de lijst modeler wordt nog niet het afwijkende lijst type weergegeven. Ook is het nog niet mogelijk om dit in de lijst modeler aan te passen.

Formulier Modeler

Het verslepen van velden om de volgorde ervan aan te passen is nog niet geïmplementeerd in de formulier modeler.

In het formulier komt het nu nog voor dat soms een aantal velden wordt bedekt met een rechthoek met dezelfde kleur als de achtergrond. Hierdoor zijn deze velden niet meer zichtbaar en niet meer te wijzigen.

Het aanpassen van de iconen voor de groepen en tabbladen, wordt nu nog gedaan met het zelf aangemaakte invoerdialoog, dat ook wordt gebruikt voor het aanpassen van de labels hiervan. Dit dialoog moet worden vervangen met het file dialoog dat ook op andere plaatsen in de Software Factory wordt gebruikt.

Het weergeven van velden met een afwijkend formulier type wordt nog niet gedaan. Het is niet te zien in de formulier modeler of er verborgen velden zijn. Ook is het nog niet mogelijk om de afwijkende formulier types in de formulier modeler aan te passen.

Bijlage H: Competenties

Tijdens het afstudeerproject zijn de verschillende competenties toegepast. Hieronder is voor elke competentie aangegeven hoe deze gebruikt is tijdens het afstudeerproject.

Professioneel Handelen

Binnen Thinkwise Software heerst een zakelijke en professionele cultuur. Ook is het werk dat verricht wordt aan de Software Factory, van belang voor verschillende klanten en voor intern gebruik. Hierdoor is het belangrijk dat op serieuze wijze aan het project is gewerkt. Ook is communicatie binnen het bedrijf op professionele wijze verlopen, door middel van vergaderingen en overleggen.

Methodisch Handelen

Tijdens het afstudeerproject zijn op verschillende methodische manieren gewerkt. Voor het gehele project is gebruik gemaakt van het tien stappen plan. Hierdoor worden de verschillende fases van het project ingedeeld in fases waarin de verschillende onderdelen worden uitgevoerd.

Voor het onderzoek is gebruik gemaakt van de onderzoeksmethodes die geleerd zijn op de Fontys.

Voor het implementatieproces is gebruik gemaakt van de SCRUM ontwikkelmethode. (Schwaber & Sutherland, 2013) Hierdoor wordt het project goed ingepland en is er altijd sprake van gestructureerde voortgang.

Samenwerken

Alhoewel het afstudeerproject individueel is uitgevoerd, is er toch sprake geweest van samenwerking. Door te overleggen met de begeleiders over de verschillende opties binnen het project, zijn de beste oplossingen verkregen voor de verschillende keuzes in het project.

Analyseren

In het onderzoek gedeelte van het project is veel analyse gedaan. De functionaliteit van de huidige GUI modeler is nauwkeurig geanalyseerd, om vervolgens een goed ontwerp te maken. Ook tijdens het implementatieproces is een hoop analyse uitgevoerd. Vanwege gebrekkige documentatie, is veel onderzoek gedaan naar de structuur van de code en welke onderdelen gebruikt moeten worden om de functionaliteit te implementeren.

Adviseren

Voor het project is in het onderzoek gedeelte een ontwerp opgesteld van de mogelijke implementatie van de grafische modeler. Er is geadviseerd aan de bedrijfsbegeleiders en de opdrachtgever over het geplande ontwerp, gebaseerd op de informatie verzameld in het onderzoek.

Ontwerpen

In het onderzoek is aan de hand van de verzamelde informatie, een ontwerp opgesteld voor de grafische modeler. Dit ontwerp is gebaseerd op een combinatie van tijdens de opleiding opgedane kennis, gecombineerd met literatuuronderzoek naar bestaande standaard interacties. Ook de technische implementatie is gebaseerd op de ontwerp principes geleerd tijdens de opleiding, gecombineerd met de structuur van de Software Factory.

Realiseren

Voor de project is een uitgebreid prototype ontwikkeld met verscheidene functionaliteiten. Doordat er extra functionaliteit is ontwikkeld voor de Software Factory GUI, is er ook uitgebreid gewerkt met de bestaande code hiervan. De realisatieperiode is verder goed ingepland met behulp van de SCRUM ontwikkelmethode (Schwaber & Sutherland, 2013), waardoor het grootste deel van de realisatie binnen de geplande tijd kon worden uitgevoerd.

Beheren

In het afstudeerproject komt het beheren vrij weinig aan bod. Om er voor te zorgen dat het ontwikkelde project goed gebruikt kan worden, is voor de juiste documentatie gezorgd. Dit zorgt ervoor dat Thinkwise Software er gebruik van kan maken en ook wijzigingen een toevoegen kan maken.

Referenties

Schwaber, K., & Sutherland, J. (2013, juli). *The Scrum Guide™*. Opgeroepen op 6 6, 2014, van The home of Scrum: [https://www.scrum.org/Portals/0/Documents/Scrum Guides/2013/Scrum-Guide.pdf](https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf)