

Afstudeerverslag: Project Annotatiecomponent



Naam afstudeerder:	M.C.W. Zuidhoek
Studentnummer:	07009593
Afstudeerbedrijf:	MI Consultancy B.V.
Afstudeerbegeleider:	P.R.C. Breukel
2 ^e examiner:	A.M.J.J Lousberg-Orbons
Begeleider afstudeerbedrijf:	A. v.d. Lans, A. d. Haas
Periode:	2011-1.1 (07-02-2011 t/m 03-06-2011)

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Voorwoord

Dit verslag gaat over het verloop van mijn afstudeerstage. Deze stage liep van 7 februari 2011 tot en met 3 juni 2011. Tijdens de stage heb ik veel hulp gehad van de mensen die bij het afstudeerbedrijf (MI Consultancy) werken. Hiervoor ben ik hen allen zeer dankbaar en ik ben blij dat ik na mijn afstuderen mag blijven. Jullie zijn geweldig.

Verder wil ik ook Karin van Schie bedanken, een goede vriendin die mij getipt heeft toen ze zag dat MI afstudeerders zocht. Dankzij haar heb ik deze geweldige afstudeeropdracht kunnen vinden.

Allen hartelijk bedankt.

Mike

Referaat

Annotatiecomponent, Visual Basic.Net, Visual Basic 6, Interop Forms Toolkit, Elektronisch Medisch Dossier, Elektronisch Patiënten Dossier, MI Consultancy.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

1. Inleiding.....	1
2. Achtergrond	2
2.1. Het bedrijf.....	2
2.2. Opdrachtschrijving	3
3. Projectopzet	5
3.1. Situatie bij aanvang	5
3.2. Technieken en methodes	5
3.3. Initiële projectplanning.....	5
4. Opstellen Project Initiation Document (PID)	6
5. Opstellen Requirements Analyse.....	9
6. Hoofdfase 1: Overzetten huidige functionaliteit	12
6.1. Subfase 1: Opstellen Functioneel Ontwerp.....	12
6.2. Subfase 2: Opstellen Technisch Ontwerp.....	15
6.3. Subfase 3: Realisatie van het ontwerp	17
6.4. Subfase 4: Testen.....	19
6.5. Subfase 5: Implementatie.....	22
6.6. Subfase 6: Schrijven van documentatie	23
7. Hoofdfase 2: Toevoegen nieuwe functionaliteit	25
7.1. Subfase 1: Opstellen Functioneel Ontwerp.....	25
7.2. Subfase 2: Opstellen Technisch Ontwerp.....	27
7.3. Subfase 3: Realisatie van het ontwerp	29
7.4. Subfase 4: Testen.....	32
7.5. Subfase 5: Implementatie.....	33
7.6. Documentatie	33
8. Afwijkingen ten opzichte van het Afstudeerplan	34
8.1. Ontwerpen en bouwen van een database	34
8.2. Planning	34
9. Evaluatie.....	35
9.1. Product	35
9.2. Proces	36
10. Beroepstaken	38
10.1. Uitvoeren analyse door definitie van requirements	38

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

10.2.	Ontwerpen systeemdeel.....	38
10.3.	Bouwen applicatie.....	39
10.4.	Initiëren en plannen van het testproces.....	40
10.5.	Uitvoeren van en rapporteren over het testproces	40
11.	Literatuurlijst	42
12.	Afkortingen.....	43
Bijlagen		i
(Tussen)producten.....		i
Overig.....		lxxxvi

1. Inleiding

In dit document staat de opzet en uitvoering beschreven van het converteren van de annotatiecomponent in het Norma EMD/EPD systeem. In chronologische volgorde wordt uitgelegd hoe het functioneel ontwerp van de nieuwe versie tot stand kwam, hoe dit ontwerp vervolgens uitgebouwd is in een technisch ontwerp en hoe dit vervolgens is gerealiseerd tot een werkelijk component. Ook wordt beschreven hoe de component getest en geïmplementeerd is en hoe de gebruikersdocumentatie tot stand is gekomen.

Belangrijker dan het hoe, is het waarom. Daarom staat in dit document ook beschreven waarom op een aantal punten is afgeweken van de originele versie, waarom een onderdeel op een bepaalde manier is uitgevoerd en waarom bepaalde beslissingen genomen zijn.

Dit document is opgesplitst in een aantal delen. In het eerste deel zal gericht zijn op de inhoud en uitvoering van het project zelf. Dit wordt ingeleid met een uitleg over de achtergrond van het bedrijf en de context waarin het project is opgezet en uitgevoerd.

Het tweede deel zal voornamelijk reflectie zijn op het uitgevoerde project. Er zal uitgelegd worden welke vaardigheden in dit project zijn gebruikt, op welk niveau deze vaardigheden zijn gebruikt en waarom. Ook wordt het project en het resultaat ervan geëvalueerd. Wat ging goed of minder goed en waarom? Wat zou het de volgende keer anders moeten en waarom? Dat zijn de vragen die in dit deel beantwoord worden. Tenslotte zal uitgelegd worden van welke competenties gebruik is gemaakt. Ook zal er uitgelegd worden waaraan te zien is dat deze competenties gebruikt zijn en op welk niveau.

Regelmatig zal in het verslag verwezen worden naar externe bronnen. Een externe bron zal worden aangegeven met een getal tussen haakjes (bijvoorbeeld (1)). Dit nummer komt overeen met een onderdeel uit de literatuurlijst die aan het einde van dit document te vinden is. Een aantal van deze externe bronnen zal ook opgenomen zijn in de bijlagen van dit verslag.

In het verslag zal ook verwezen worden naar andere hoofdstukken in het verslag. Een interne verwijzing zal worden aangeduid met teksten in de vorm van “zie hoofdstuk 2.1” of “zie hoofdstuk 2”. Er kan ook verwezen zijn naar een afbeelding in het verslag. Deze verwijzingen worden aangegeven met de tekst “zie Figuur 1” en de afbeelding zelf zal zijn voorzien van een bijschrift.

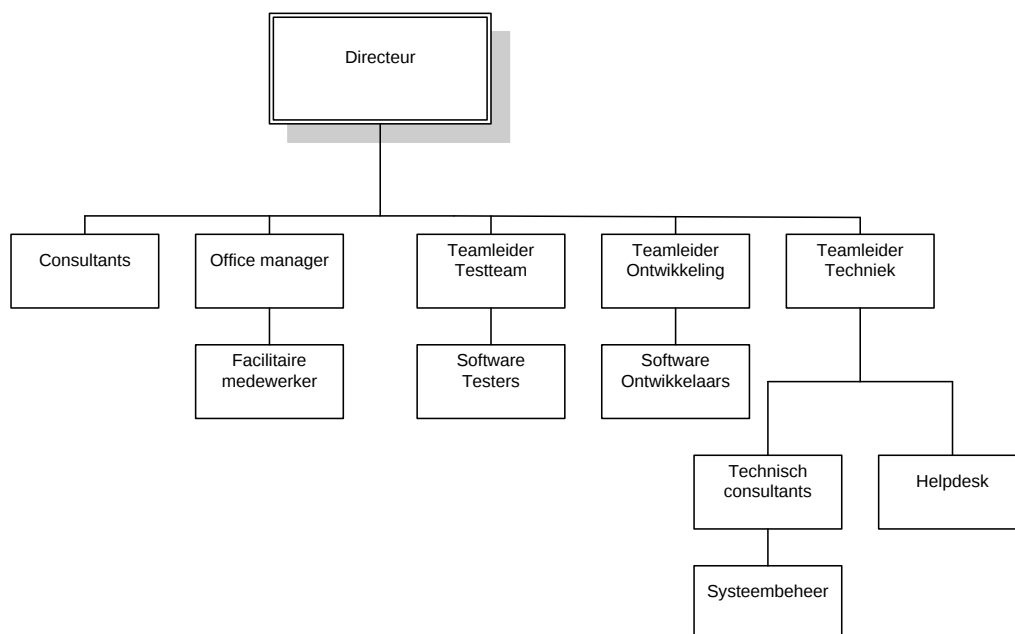
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

2. Achtergrond

2.1. Het bedrijf

MI Consultancy, staat voor Medned Information Consultancy BV, is opgericht in 1986 en tot op heden zijn er 23 mensen werkzaam bij dit bedrijf. Het bedrijf werkt aan het ontwikkelen en beheren van Norma. Norma is een softwarepakket voor ziekenhuizen en is een EPD (Elektronisch Patiënten Dossier).

Ziekenhuizen en zelfstandige behandelcentra (ZBC's) vormen de doelgroepen van MI Consultancy. In verschillende ziekenhuizen & ZBC's in Nederland wordt Norma gebruikt door een zeer diverse groep eindgebruikers, van de arts-assistent, specialist, verpleegkundige, SEH-afdeling tot de patholoog-anatoom. Norma biedt alle nodige registratiemogelijkheden en inzage in de andere gegevens, zoals laboratoriumuitslagen, opnameoverzichten en röntgenverslagen.



Figuur 1: Organogram MI Consultancy

Figuur 1 toont een organogram van het bedrijf. Tijdens de opdracht is gewerkt vanuit de afdeling ontwikkelaars, hoewel er ook contact is gehouden met de afdelingen consultants en testers.

Consultants zijn verantwoordelijk voor de communicatie met de klanten. Zij nemen de wensen en eisen van de gebruikers op en vertalen dit naar een concept voor een uitbreiding op het bestaande systeem. Dit concept bestaat meestal uit een concept functioneel ontwerp waar de basisfunctionaliteit van de te ontwikkelen uitbreiding in beschreven staat (2).

Ontwikkelaars nemen de concepten van de consultants over, werken deze verder uit en zetten deze om in werkende componenten. Ze zijn ook verantwoordelijk voor het maken en aanpassen van de database. Ook repareren zij alle bugs die gevonden worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Testers zijn verantwoordelijk voor het bepalen van de kwaliteit van de ontwikkelde componenten. Het is hun taak bugs te vinden voordat de klant dat doet.

2.2. Opdrachtoomschrijving

Het bedrijf werkt vooral aan het software pakket Norma. Norma is een Server-client gebaseerde applicatie. Het systeem wordt ondersteund door een database dat draait op het DBMS Pervasive.

In dit project zijn alleen wijzigingen aangebracht aan de Norma Client. In de client is het EMD (Elektronisch Medisch Dossier) opgenomen. In dit EMD worden alle ziektebeelden van een patiënt opgenomen. Dit gebeurt in verschillende formulieren die door de gebruiker zelf gedefinieerd worden. Deze formulieren bestaan weer uit verschillende componenten. Eén van deze componenten is de annotatiecomponent. In dit component kan de gebruiker op schematische weergaven aanduiden waar een patiënt klachten ondervindt. Met andere woorden, de gebruiker zet niet meer een kruisje op een tekening op papier (zie Figuur 2), maar kan dat direct in een formulier in het systeem doen.

Rug		
asdrukpijn		
Paravert. Idoppijn		
fixatie		
scoliose		
lasègue	R	L
Omgek lasègue	R	L
Gekr. lasègue	R	L
Kemp	R	L
Natfziger	R	L
Bragard	R	L
Heupen	R	L
S.I. gewricht	R	L

NEK		
fixatie		
lasègue arm	R	L
For. comp. test	R	L
Adson	R	L
Wright	R	L
Bew. schouder	R	L

Figuur 2: Een papieren annotatie formulier

Momenteel is het bedrijf bezig met het omzetten van deze componenten van VB6 naar VB.Net. Een gedeelte hiervan is al voltooid. Echter de annotatiecomponent dient nog wel herschreven te worden.

De VB6 versie van de annotatiecomponent is één van de oudere componenten die het systeem kent. Hierdoor voelt de component, in vergelijking met een aantal nieuwere componenten, soms wat ouderwets en stijf aan. Dit is door de jaren heen ook aangegeven door verschillende klanten, die ook een aantal suggesties en wensen hebben doorgegeven over hoe de component te verbeteren is. Denk hierbij aan wensen als meer bewerkingsmogelijkheden, of meer opties voor kleur of tekst.

Een ander groot nadeel van de VB6 versie, is de code zelf. VB6 is namelijk niet object georiënteerd. Dit zorgt ervoor dat de code snel rommelig aanvoelt. Ook is de code niet of nauwelijks herbruikbaar, zijn de UI mogelijkheden beperkt en VB6 heeft een beperkte performance in vergelijking met andere programmeertalen.

Het gebrek aan object oriëntatie zorgt ervoor dat de code al slecht te lezen is. Dit, samen met het feit dat door de jaren heen veel aanpassingen zijn gedaan om bijvoorbeeld fouten te herstellen, hebben ervoor gezorgd dat de code tegenwoordig nauwelijks nog leesbaar is. Dit

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

heeft tegelijkertijd ook een negatieve invloed op de onderhoudbaarheid, omdat het niet meer te overzien is welke gevolgen een bepaalde aanpassing heeft.

Door de jaren heen zijn de richtlijnen van het bedrijf met betrekking tot UI en code flink veranderd. De VB6 versie van de component voldeed ook niet meer aan deze richtlijnen. De slechte onderhoudbaarheid zorgde er ook voor dat het moeilijk was om de component alsnog aan deze richtlijnen te laten voldoen.

3. Projectopzet

3.1. Situatie bij aanvang

Bij aanvang van de afstudeerperiode, bestond de annotatiecomponent nog als een VB6 UserControl. Zoals in hoofdstuk 2.2 beschreven, voldeed deze niet meer aan de eisen van het bedrijf. Ook de klanten waren er niet meer tevreden mee. Een andere stagiair had in het verleden al geprobeerd deze component te converteren naar VB.Net, maar was daar niet in geslaagd. Hoewel onbruikbaar voor het systeem, kon dit project wel gebruikt worden om een basisidee te krijgen over de werking van de externe DLL die gebruikt zou worden voor dit project. Deze externe functionaliteit (Vintasoft (3)) zou de basisfunctionaliteit leveren voor de component. Er was ook een kort Functioneel ontwerp aanwezig waarin werd aangegeven waar de VB6 versie van de component toe in staat was en wat de wensen en eisen waren van de klanten van het bedrijf.

3.2. Technieken en methodes

Het ontwerpen van het systeem zou gedaan worden met behulp van UML klassen- en objectdiagrammen en Use-Cases, volgens de richtlijnen van het bedrijf. De diagrammen zijn opgesteld met behulp van de informatie uit (4). Een aantal diagrammen, zoals sequence diagrammen zijn niet gemaakt. Dit omdat ze dieper op bepaalde aspecten van het ontwerp ingingen dan nodig was. Het testen is uitgevoerd met behulp van de eigen methodes van het bedrijf. Het programmeren van de component is gedaan in Visual studio 2010 in de taal Visual Basic.Net (VB.Net). Het .Net Framework dat gebruikt is, is versie 2.0. Er is verder gebruik gemaakt van de frameworks van DevExpress (1) voor de uitgebreidere grafische opties. Microsoft Interop Forms Toolkit (5) tenslotte is gebruikt om het .Net component in een VB6 hoofdproject te kunnen plaatsen (zie hoofdstuk 6.5). Al deze onderdelen zijn volgens de standaarden van het bedrijf. Het project is uitgevoerd als een PRINCE2 project.

3.3. Initiële projectplanning

Voor aanvang van het project was de planning als volgt:

Activiteit	Tijd nodig (naar schatting)
Schrijven PID	1 week
Opstellen en analyseren requirements	2 weken
Analyse aangekochte onderdelen	1 week
Schrijven functioneel ontwerp	2 weken
Schrijven technisch ontwerp	2 weken
Realisatie van het ontwerp	4 weken
Testen	2 weken
Ondersteuning bij implementatie	1 week
Schrijven gebruiker en beheerdocumentatie	1 week
Overdracht	1 week

Tijdens het project is deze planning nog een aantal keren aangepast (zie hoofdstuk 4).

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

4. Opstellen Project Initiation Document (PID)

Het project begon met het maken van een Project Initiation Document (PID). In het PID is “de projectopzet getoond van het project Annotatiecomponent”(6). De eerste versie van het PID bevatte onder andere een doel- en probleemstelling, resultaten, randvoorwaarden & beperkingen en een projectplanning.

Nadat de opdrachtgever goedkeuring verleende op deze punten, is het PID uitgebreid met de punten aanpak en fasering, omvang, organisatiestructuur, beheersingsmechanismen en projectrisico's.

Eén van de belangrijkste risico's die duidelijk werd bij het opstellen, was het risico dat het project mogelijk niet op tijd voltooid zou worden, waardoor het bedrijf achter zou blijven met een niet-werkend component. Om dit risico af te kunnen vangen, is ervoor gekozen om het project op te hakken in twee grote fases.

Fase	Begin datum	Eind datum	Duur in weken	Activiteiten	Op te leveren product(en)
Initiërende fase	7-2	9-2	0.5	Opstellen Project Initiatie Document (PID)	PID
Analyse	9-2	16-2	1	- Analyse oorspronkelijke module - Analyse aanvullende wensen / eisen	Requirements analyse
Functioneel Ontwerp	16-2	23-2	1	- Use-Cases - UI ontwerp - Testplan - Opstellen functioneel ontwerp	Functioneel ontwerp Testplan
Technisch Ontwerp	23-2	2-3	1	- Klassediagrammen - Objectdiagrammen - Opzetten testomgeving - Maken testscripts	Technisch ontwerp
Realisatie	2-3	23-3	2 ⁺	Uitvoeren technisch ontwerp	Module
Testen	23-3	1-4	1.5	- Uitvoeren testscripts - Mogelijke scenario's - Gevonden fouten herstellen	Testverslag
Implementatie	4-4	8-4	1	Ondersteuning implementatie	<geen>
Beheer en Documentatie	11-4	15-4	1	- Gebruikershandleiding - Technische handleiding	Documentatie
Uitloop	18-4	22-4	1	N.V.T.	<geen>
Totale duur	7-2	22-4	10 ⁺	N.V.T.	<geen>

Figuur 3: PID planning Fase Huidige functionaliteit

Fase één zou enkel bestaan uit het overzetten van de bestaande functionaliteit van VB6 naar VB.Net. Pas als dit volledig was afgerond, zou er verder gegaan worden met fase twee: het toevoegen van nieuwe functionaliteit aan het VB.Net component. Hierdoor zou het bedrijf, zelfs al zou fase twee niet voltooid worden, in elk geval beschikken over een basisproject dat gelijkwaardig is aan de oorspronkelijke component in VB6.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Fase	Begin datum	Eind datum	Duur in weken	Activiteiten	Op te leveren product(en)
Functioneel Ontwerp	25-4	27-4	0.5	• Use-Cases • UI ontwerp • Testplan • Opstellen functioneel ontwerp	Functioneel ontwerp Testplan
Technisch Ontwerp	27-4	29-4	0.5	• Klassediagrammen • Objectdiagrammen • DB ontwerp • Opzetten testomgeving • Maken testscripts	Technisch ontwerp
Realisatie	2-5	13-5	2	• Uitvoeren technisch ontwerp	Module
Testen	16-5	20-5	1	• Uitvoeren testscripts • Mogelijke scenario's	Testverslag
Implementatie	23-5	25-5	1	• Ondersteuning implementatie	<geen>
Beheer en Documentatie	25-5	27-5	1	• Gebruikershandleiding • Technische handleiding • Gevonden fouten herstellen	Documentatie
Uitloop	30-5	3-6	1	N.V.T.	<geen>
Totale duur	25-4	3-6	7	N.V.T.	<geen>

Figuur 4: PID planning Fase Nieuwe functionaliteit

Het bepalen van de organisatiestructuur nam wat extra tijd in beslag omdat bepaald moest worden wie er zou dienen als het Project Assurance team. Dit team zou verantwoordelijk zijn voor het monitoren van (kwaliteits)aspecten van het project en de documentatie eromheen. De belangrijkste taak hierin was controleren of de component aan bleef sluiten op de requirements. Uiteindelijk heeft consultant Anne ten Hove deze rol op zich genomen. Verder zijn de procedures bij uitzonderingen uitgedacht.

In het hoofdstuk beheersingsmechanismen is aangegeven welke mechanismen worden toegepast om de beheersbaarheid van het project te waarborgen. Er is hier aangegeven dat de voortgangsrapportage voornamelijk via Exception reports en End Stage Reports gedaan zou worden.

Er is gekozen om niet alle documenten die PRINCE2 kent ook daadwerkelijk te gaan gebruiken. Het project zou groter en onhandelbaarder worden dan noodzakelijk. Een voorbeeld hiervan is een Timesheet rapportage. Ten eerste was in het PID al duidelijk beschreven welke werkzaamheden in elke fase uitgevoerd zouden worden evenals de looptijd van elke fase. Verder was er ook regelmatig mondeling overleg met de begeleiders / opdrachtgever. Hierdoor zou dit document dus niets toevoegen aan het project.

Ook in het PID zelf zijn een aantal onderdelen niet opgenomen. Ook hier geldt dat een aantal onderdelen niets toe zou voegen aan het project. Voorbeelden hiervan zijn de management samenvatting, de investeringsanalyse of de relaties met andere projecten. Deze onderdelen zouden niets toevoegen voor het bedrijf of voor het project zelf.

Het maken van het PID liep van 7 februari tot en met 15 februari. Dit was een stuk langer dan oorspronkelijk gepland. De vertraging werd onder andere veroorzaakt door het

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

opsplitsen van het project in de twee hoofdfases en de toevoeging van een aantal punten zoals beheersingsmechanismen en organisatiestructuur. Ook ontstond er vertraging doordat de omgeving van het bedrijf nog onbekend was.

5. Opstellen Requirements Analyse

Requirement oude functionaliteit

Algemeen

- De nieuwe component moet hetzelfde formaat hebben als de oude versie.
- Indien de afbeelding groter is dan de component, dienen schuifbalken getoond te worden waarmee de gebruiker door de afbeelding kan bewegen.
- De look van de nieuwe component dient zoveel mogelijk overeen te komen met de look van de oude component om het voor de eindgebruikers zo eenvoudig mogelijk te houden.
- De oude component controleert nu of de muis op een bepaald icoontje staat en laat aan de hand daarvan verschillende opties zien (als een soort uitschuif menu). Dit werkt echter niet helemaal zoals gewenst. Gekeken moet worden of dit in de nieuwe component anders kan.
- Afbeeldingen moeten versleept kunnen worden met het "handje".
- Indien er een nieuwe instantie of episode gemaakt wordt, dient er een nieuwe, lege (standaard) afbeelding in de component geplaatst te worden.

Annotaties

- De component moet de volgende typen annotaties kunnen plaatsen:
 - Pen
 - Tekst
 - Symbolen
- Annotaties moeten gedraaid, verplaatst en verwijderd kunnen worden en het moet mogelijk zijn om de annotatie te vergroten of te verkleinen.
- Annotaties moeten herhaaldelijk geplaatst kunnen worden.
- Indien een annotatie gekozen wordt dient de cursor te veranderen in de te plaatsen annotatie.
- Annotaties moeten ongedaan gemaakt kunnen worden met de knop ongedaan maken.
- De component mag alleen gebruikt worden indien de gebruiker het recht heeft om het EMD te wijzigen.
- Indien de component wordt aangeroepen vanuit gastgebruik, mag deze alleen gebruik worden als aangegeven staat dat gastgebruik toestemming heeft om het EMD te wijzigen.

Instellingen

- Alle instellingsschermen dienen in aparte vensters getoond te worden.
- Er dient een instellingenscherm te zijn voor de volgende onderdelen:
 - Tekst
 - Afbeeldingen
 - Kleuren
 - Lijn
 - Tekstinstellingen:
- Het lettertype en de lettergrootte dient aangepast te kunnen worden (standaard Arial 16).
- Afbeeldingsinstellingen:
 - Afbeeldingen kiezen die gedefinieerd zijn met standaard labels. Deze labels moeten ingesteld staan op actief.
 - De labels hebben standaard een prefix

Requirements nieuwe functionaliteit

Algemeen

- Huidige functionaliteit moet beschikbaar blijven.
- Annotaties moeten na opslaan nog bewerkt kunnen worden.
- Het terugdraaien van acties moet ongedaan gemaakt kunnen worden (redo functie).

Annotaties

- Er moet een eigen symboolannotatie ingeladen kunnen worden voor specialismen die aan de standaard symbolen niet voldoende hebben.
- De eigenschappen van geplaatste annotaties moeten aangepast kunnen worden per annotatie (kleur, lijnstijl enzovoort).

Instellingen

- Bij Lijninstellingen moeten de teksten die de verschillende lijnstijlen aangeven vervangen worden door afbeeldingen.
- De kleur van tekstannotaties moet gewijzigd kunnen worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

(XX—Label). Deze prefix mag niet getoond worden. ○ Afbeeldingen importeren vanaf de lokale PC. Alleen het JPG formaat is toegestaan. ○ Melding indien wijzigingen zijn opgeslagen voor het wisselen van de afbeelding • Lijninstellingen: ○ Lijndikte moet ingesteld kunnen worden (standaard 2 pixels). ○ Lijntype moet ingesteld kunnen worden (vast, streep, punt, streep-punt, streep-punt-punt). • Kleurinstellingen: ○ Er moet gekozen kunnen worden uit 2 standaardkleuren (blauw, rood waarvan blauw automatisch gekozen wordt indien de gebruiker geen kleur kiest) en 3 specifieke kleuren. ○ De standaardkleuren mogen niet gewijzigd worden.	
--	--

Figuur 5: Requirements

Deze fase had als doel om de wensen en eisen vast te leggen in een requirements document. Bij aanvang van het project was er een eenvoudig functioneel ontwerp (2) dat beschreef waar de VB6 versie van de component toe in staat was. Hiermee was de basis gelegd voor de requirements. Ook beschreef dit document de wensen van verschillende klanten. Deze wensen zijn gebruikt voor het opstellen van requirements voor de uitbreidingen op de component.

Vervolgens is er gekeken naar de oorspronkelijke VB6 code. Dit gaf een indruk van de samenhang tussen component en het geheel waarin deze geplaatst was. Hiermee kon bepaald worden wat er nodig was om de nieuwe component goed te laten passen in het geheel als deze afgerond was.

Er was al eerder geprobeerd de component te vertalen naar VB.Net. Deze poging was echter mislukt. Toch bood het project nog wel enig inzicht over hoe de externe functionaliteit toe te passen waren. Deze tweede versie van de component gaf, samen met de VB6 versie, nog een aantal extra requirements zoals bijvoorbeeld de standaard instellingen (lijn, tekst, kleur).

Tenslotte is er gekeken naar hoe de oorspronkelijke component zich gedroeg in de productieomgeving. Hierin was te zien hoe de component reageerde op input van de gebruiker en kon beter nagegaan worden hoe de user interface eventueel verbeterd kon worden. Uiteindelijk leverden deze punten een conceptversie op voor de requirements.

Deze conceptversie werd vervolgens voorgelegd aan de begeleiders. De gesprekken met hen hierover gaf nog een aantal extra requirements, bijvoorbeeld over het weglaten van de prefix (een prefix is een aantal tekens dat aangeeft bij welk specialisme een afbeelding hoort, bijvoorbeeld “kno—”). Om de lijst met requirements overzichtelijk te houden zijn alle requirements onderverdeeld in drie grote categorieën: Algemeen, Annotaties en Instellingen. Onder Algemeen zijn alle requirements te vinden die betrekking hebben tot de

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

GUI. Annotaties bevat alle requirements die te maken hebben met de annotaties zelf, zoals de soorten annotaties en de bewerkingsmogelijkheden ervan. Instellingen tenslotte bevatten alle requirements met betrekking tot de instellingen. Voor extra overzicht zijn ook deze weer onderverdeeld in een aantal categorieën. Uiteindelijk ontstond het requirements document dat te vinden is in Figuur 5.

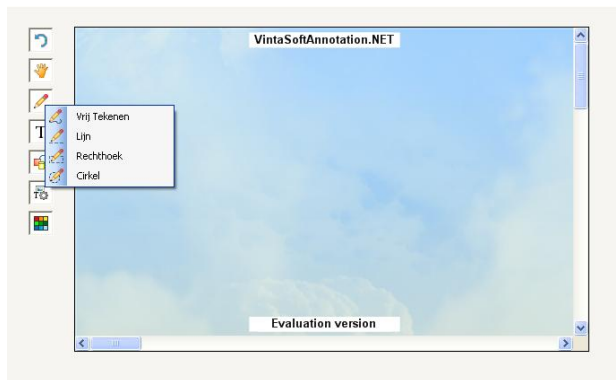
Deze fase liep van 9 februari tot en met 16 februari en begon al terwijl het PID nog verbeterd en uitgebreid werd. Dit zorgde ervoor dat de impact het tijdsverlies van het PID op de rest van de planning beperkt bleef.

6. Hoofdfase 1: Overzetten huidige functionaliteit

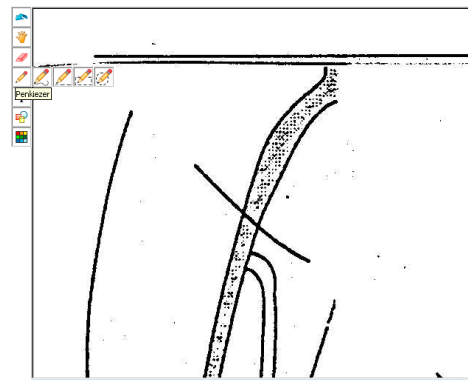
In deze fase is alle bestaande functionaliteit overgezet van VB6 naar VB.Net. In deze fase is een analyse gemaakt van de wensen en eisen, het opstellen van een functioneel en technisch ontwerp, het daadwerkelijke bouwen van de component, het testen van de component en het implementeren en maken van bijbehorende documentatie. De planning van deze fase liep van 7 februari tot en met 15 april, met een uitloopweek tot 22 april. De fase eindigde daadwerkelijk op 4 april. De reden voor het eerder afronden van deze fase is terug te vinden in hoofdstuk 6.4 en 6.5.

6.1. Subfase 1: Opstellen Functioneel Ontwerp

Nadat de requirements opgesteld waren, begon het werken aan het functioneel ontwerp. Er is begonnen met het maken van een ontwerp voor de grafische user interface (GUI) van de component, omdat de requirements voor dit onderdeel het meest strikt waren, wat het ook het eenvoudigste onderdeel maakte.



Figuur 6: Eerste UI ontwerp van de component



Figuur 7: UI van de oorspronkelijke component

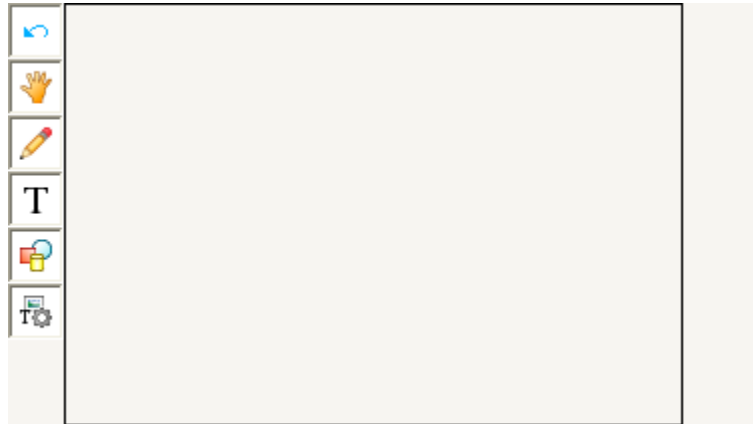
De voornaamste eis aan de GUI van de component was dat deze zoveel mogelijk gelijk bleef aan het origineel, om de overgang voor de gebruiker zo eenvoudig mogelijk te maken. Het belangrijkste verschil zit hem in de uitklapbare menu's. In de originele versie verscheen er een rij met icoontjes als er meerdere opties waren (zie Figuur 7). In de nieuwe versie zijn deze vervangen door Contextmenu's (zie Figuur 6). Dit heeft vooral te maken met het feit dat de rij afbeeldingen uit het origineel niet altijd goed reageren. De contextmenu's zijn een eenvoudige oplossing hiervoor, die ook gemakkelijk uit zijn te breiden met behulp van de designer van Visual Studio. Een andere optie was om aparte venstertjes te maken waarin de gebruiker een keuze kon maken, maar dit hield in dat de gebruiker extra handelingen moest verrichten. Hierdoor was deze oplossing minder geschikt dan de contextmenu's.

In de oorspronkelijke versie zaten de instellingen verstopt in een menu dat opgeroepen werd met de rechtermuisknop. Door een wijziging in de richtlijnen voor de user interface van het bedrijf is dit inmiddels niet meer wenselijk. Daarom is ervoor gekozen om de instellingen nu een eigen icoontje te geven aan de linkerkant van de component. De verschillende opties zijn nu onderverdeeld in kleine schermpjes om alles zo overzichtelijk mogelijk te houden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Oorspronkelijk was er niet aan gedacht dat het formaat van de component een belangrijk punt was. Echter, dit bleek wel degelijk belangrijk. Doordat de componenten naar eigen inzicht geplaatst kunnen worden in zogeheten formulieren en het formaat van de componenten door de gebruiker zelf in te stellen is, mocht de component in standaardafmetingen dus *absoluut* niet afwijken van de afmetingen van de VB6 versie. Dit betekende dat alle afbeeldingen en venstertjes tot op de pixel nauwkeurig gelijk moesten zijn aan de VB6 versie (zie Figuur 5).

Hierdoor is het UI ontwerp nog enigszins veranderd, hoewel dit verder geen ingrijpende veranderingen meer had op het definitieve ontwerp, aangezien deze al grotendeels overeenkwam met de oorspronkelijke UI.



Figuur 8: Definitieve User Interface

Na het toevoegen van de requirement met betrekking tot de afmetingen van de component, kon er verder gewerkt worden aan de Use Cases. Deze Use Cases zijn terug te vinden in (7).

Naam	Afbeelding wijzigen → Afbeelding importeren
In Beschrijving	Het wijzigen van de achtergrondafbeelding.
Aannamen	Geen
Resultaat	De afbeelding is veranderd en alle annotaties zijn verwijderd.
Stappenplan	1. Druk op het icoon Afbeelding instellen. 2. Kies de optie Bladeren 3. Zoek de juiste afbeelding op de lokale harde schijf 4. Indien er eerder opgeslagen wijzigingen bestaan, wordt er een melding getoond en om bevestiging gevraagd. 5. De oude afbeelding en alle annotaties die erop geplaatst zijn worden verwijderd. 6. De nieuwe afbeelding wordt ingeladen.
Uitzonderingen	Indien de afbeelding opgeslagen is, wordt een melding weergegeven dat eerder opgeslagen data verloren zal gaan. Stap 5 en 6 worden niet uitgevoerd als de gebruiker geen toestemming geeft bij 4. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden. Alleen .JPG of .JPEG afbeeldingen mogen ingeladen worden.

In Figuur 9 is een voorbeeld van een Use Case te zien. Deze Use Case is opgesteld aan de hand van de volgende requirements:

- Afbeeldingen importeren vanaf de lokale PC. Alleen het JPG formaat is toegestaan.
- Melding indien wijzigingen zijn opgeslagen voor het wisselen van de afbeelding.

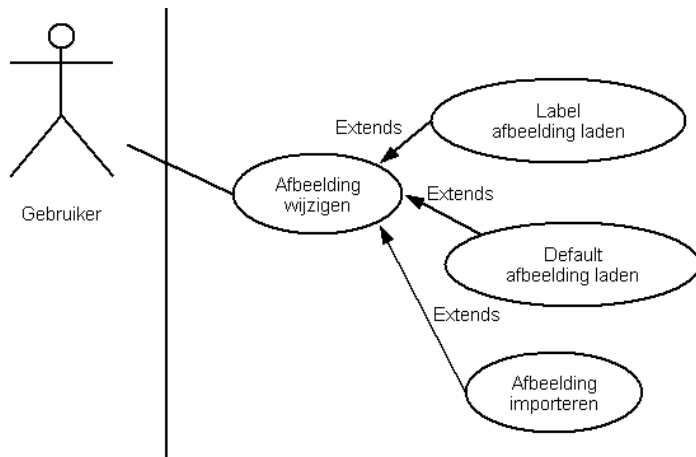
Zoals te zien is in het stappenplan wordt inderdaad gecontroleerd of de afbeelding al een keer is opgeslagen. De uitzonderingen geven aan dat alleen .jpg en .jpeg bestanden geopend mogen worden.

Een aantal van de Use cases maken ook gebruik van elkaar om hun

Figuur 9: Use Case

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

eigen functie uit te kunnen voeren, zoals de Use Cases in Figuur 10. De functie afbeelding wijzigen kan op meerdere manieren uitgevoerd worden, afhankelijk van de acties van de gebruiker. Deze kan ervoor kiezen om een afbeelding te kiezen uit een lijstje, zelf een



Figuur 10: Een deel uit het Use Case diagram

afbeelding opzoeken op de harde schijf, of de oorspronkelijke afbeelding opnieuw te laden. Het resultaat van elk van deze Use Cases is gelijk, maar de regels ervan verschillen. Daarom is ervoor gekozen om drie aparte Use Cases te maken. Om aan te duiden dat de Use Cases wel hetzelfde resultaat geven, hebben zij een extend relatie met een algemene Use Case Afbeelding wijzigen. De reden dat er voor een extends relatie is

gekozen en niet een include is omdat er slechts één van de drie Use Cases per keer uitgevoerd kan worden.

Aan de hand van de Use Cases kon het eerste deel van het testplan gemaakt worden. Dit deel van het testplan was voornamelijk toegespitst op de vraag of het daadwerkelijke resultaat van de use cases gelijk was aan het verwachte resultaat. De Use cases gaven ook gelijk een stappenplan van hoe de test uitgevoerd kon worden. Hiermee kon de eerste versie van de checklist opgesteld worden.

Voor elke Use case werd gekeken wat het verwachte resultaat was. Deze punten werden allemaal in de checklist opgenomen. Vervolgens werden de stappenplannen van alle Use Cases bekeken om punten te ontdekken waar iets fout zou kunnen gaan (bijvoorbeeld een gebruiker die een verkeerd bestandstype in probeert te laden in de Use-Case in Figuur 9). Deze punten werden ook toegevoegd aan de checklist.

Nadat alle punten van de checklist waren verzameld, werden deze ingedeeld in categorieën om overzicht te houden. De categorieën opstarten, plaatsen en bewerken / verwijderen zijn gekozen omdat bijna alle punten in de checklist hiermee te maken hebben. De drie punten die overbleven zijn geplaatst in de categorie overig (7).

Vervolgens moest bepaald worden wat er wel en niet getest zou worden. Omdat alle functionaliteit van de grond af aan opnieuw opgebouwd zou worden in een andere ontwikkelomgeving, is bepaald om alle Use Cases uit het ontwerp te testen. Natuurlijk waren er ook triviale dingen waarvan testen niet echt prioriteit had. Een voorbeeld hiervan is het aanpassen van het formaat van de component. Dit testen zou totaal overbodig zijn, omdat in één oogopslag te zien is of een component zich aangepast heeft aan de opgegeven afmetingen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Tenslotte is er ook een globale planning opgesteld voor de testers. Hiermee konden zij inschatten hoeveel werk zij ongeveer zouden hebben aan het testen. Ook werd hierin aangegeven hoeveel tijd er vrijgemaakt was voor het repareren van gevonden fouten.

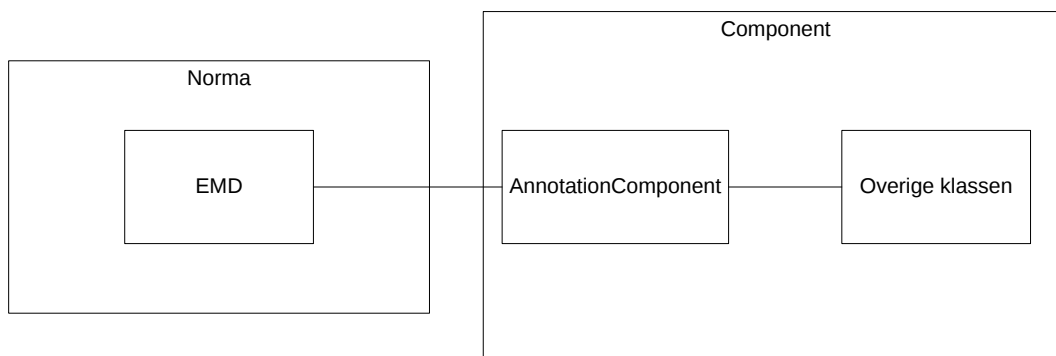
Het opstellen van het functioneel ontwerp liep van 16 februari tot en met 23 februari, precies volgens de planning van het PID. Op dit punt was de achterstand opgelopen tijdens het opstellen van PID volledig weggewerkt.

6.2. Subfase 2: Opstellen Technisch Ontwerp

Nadat het functioneel ontwerp voltooid was, kon het technisch ontwerp opgesteld worden. Voor zover bekend was er geen ontwerp meer van de VB6 versie van de component. De VB6 code was ook niet objectgeoriënteerd omdat dit niet mogelijk is in VB6 (zie ook hoofdstuk 2.2), maar wat wel een eis voor de nieuwe versie was. Hierop is besloten om het technisch ontwerp van de grond af aan opnieuw op te bouwen, zonder gebruik te maken van eerdere documentatie van de VB6 versie.

Het opstellen van het technisch ontwerp is begonnen met het klassendiagram omdat de rest van het ontwerp hierop zou steunen. Bij het opstellen van het klassendiagram is goed gekeken naar de beschreven functionaliteit in de use cases.

De basis van het klassendiagram was de klasse AnnotationComponent. Deze klasse kreeg de verantwoordelijkheid over de interface en het aanroepen en beheren van de andere objecten (zie Figuur 11). Het idee was dat deze klasse dienst zou doen als “aanspreekpunt” voor het geheel waar de component in geplaatst zou worden. Dit alles is volgens de regels van de Mediator design pattern. Alle communicatie tussen objecten worden vanuit deze klasse geregeld, wat ervoor zorgt dat de andere objecten beter herbruikbaar zijn.



Figuur 11: Component ten opzichte van het systeem

Om te kunnen communiceren met het Norma systeem, zijn er verschillende properties en publieke methoden ingebouwd in de klasse AnnotationComponent. Een UML klassendiagram ondersteunt echter geen properties. Hierdoor was het niet mogelijk deze op te nemen in het ontwerp.

In plaats van zelf de functionaliteit te maken die ervoor zorgt dat er annotaties op een afbeelding geplaatst kunnen worden, heeft het bedrijf voor aanvang van het project gekozen

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

om gebruik te maken van externe functionaliteit. Deze functionaliteit (Vintasoft (3)) biedt mogelijkheden voor het bewerken van afbeeldingen en deze bewerkingen als XML op te slaan, zodat deze hergebruikt kunnen worden. Ook biedt de externe functionaliteit mogelijkheden om verschillende soorten standaardannotaties te maken (zoals tekst of een cirkel) of anders eigen annotaties te ontwerpen (door het overerven van annotatieklassen).

De component moet verschillende typen annotaties ondersteunen. Omdat alle annotaties dezelfde basisvariabelen hebben (formaat, kleur, positie enzovoort), is ervoor gekozen om één abstracte klasse Annotation te maken, met verschillende subklassen die de specifieke gegevens voor een bepaald type annotatie ondersteunen (8). Verder is er de klasse ColorManager, die verantwoordelijk is voor het veranderen van de kleur van symboolannotaties en de icoontjes van de GUI. Deze klasse was nodig omdat symboolannotaties uit een bitmap afbeelding bestaan. De klasse ColorManager past de afbeelding pixel voor pixel aan als er een andere kleur gekozen is dan de standaardkleur van de afbeelding. Dit is niet de meest efficiënte methode, maar omdat alle symboolannotaties 32 bij 32 pixels zijn, zal dit niet veel impact hebben op de performance van de component.

Later in het ontwerpproces kwam naar voren dat de externe functionaliteit geen ondersteuning bood aan het terugdraaien van acties (ongedaan maken). Dit betekende dus dat deze functionaliteit zelf ontwikkeld moest worden. Er waren verschillende mogelijkheden om dit te doen.

Als eerste is geprobeerd om een soort backup te maken van de complete annotatie. Deze backup zou dan opgeslagen worden in een Array, en elke keer als op ongedaan maken werd geklikt werd de backup van de annotatie teruggeplaatst op de afbeelding. De nieuwe versie van de annotatie zou dan natuurlijk weggehaald worden. Het bleek echter dat op het moment dat de annotatie veranderde, ook de backup versie van de annotatie veranderde. Waarschijnlijk kwam dit doordat er maar één versie van een annotatieobject met een bepaalde GUID (Uniek ID) mag bestaan. Hierdoor bleek dit idee dus niet uitvoerbaar.

Uiteindelijk is de klasse UndoManager ontworpen. Deze klasse slaat de gegevens van de bewerkte annotatie op voor en na de bewerking, zodat deze later teruggedraaid konden worden. De opzet van de klasse bood ook de mogelijkheid om de acties later opnieuw uit te voeren.

Na de afronding van het klassendiagram, is het objectdiagram opgesteld. Dit diagram was bedoeld om een duidelijker beeld te geven van relaties tussen objecten onderling. In het objectdiagram zijn ook, voor de volledigheid, de standaard .NET componenten opgenomen, die samen de GUI vormen.

Als laatste onderdeel van het technisch ontwerp, is het testplan, waarmee begonnen is in het functioneel ontwerp, verder uitgebreid. Er zijn aanwijzingen opgenomen over hoe de testomgeving ingericht moest worden en de checklisten zijn verder uitgebreid met extra punten waarop getest moest worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Later in de realisatiefase zou blijken dat het technisch ontwerp nog niet voldoende was en dat niet alle functionaliteit gerealiseerd kon worden met de huidige opzet. Hierop is besloten het technisch ontwerp aan te passen (9). Meer hierover is te vinden in paragraaf 6.3. Ook zou blijken dat de gumfunctie niet uitvoerbaar bleek.

Het opstellen van het technisch ontwerp liep van 24 februari tot en met 2 maart, volgens de planning van het PID. Tijdens de realisatie fase is ook nog tijd uitgetrokken om het technisch ontwerp bij te werken.

6.3. Subfase 3: Realisatie van het ontwerp

Na het afronden van de ontwerpfase, was het tijd om het ontwerp om te zetten in een werkend component. Vanwege de strenge richtlijnen, welke dit onderdeel relatief eenvoudig maakten, is als eerste de GUI uitgewerkt: het aanpassen en plaatsen van de icoontjes, de ruimte waarin de afbeelding moest komen, het toevoegen van de contextmenus en de dialoogvensters. Hierna werd de functionaliteit toegevoegd.

Eén van de eisen gesteld in de requirements was dat de cursor moest veranderen in de geselecteerde annotatie (bijvoorbeeld een pen voor een lijnannotatie, of het plaatje van de gekozen symboolannotatie). Hier werd echter een probleem duidelijk. Zodra de cursor zich op de afbeelding bevond, veranderde deze terug naar de standaardcursor. Alle standaardmanieren om de cursor te veranderen werkten niet meer.

In een poging dit op te lossen is er contact gezocht met de leverancier van de externe functionaliteit via hun support forum (10). Hier werd aangegeven dat dit een bekend probleem was, dat opgelost was in een nieuwere versie. Echter, de licentie van het bedrijf gaf recht op een jaar aan updates en dit jaar was enkele weken voor deze versie uitkwam verstreken. Dit betekende dus dat er een nieuwe versie aangeschaft zou moeten worden voor ongeveer €800. Aangezien dit het enige probleem was dat we hadden, werd dit bedrag te hoog bevonden om alleen een cursor te laten veranderen.

Het vreemde aan dit probleem was dat de cursor wel werkte in het project van de vorige stagiair. Verdere analyse van dit project, toonde aan dat er een stukje code in het project zat dat handmatig een nieuwe cursor aanmaakte. Dit stuk code is vervolgens hergebruikt en veranderd in de klasse CursorManager, waardoor het probleem alsnog opgelost werd .

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

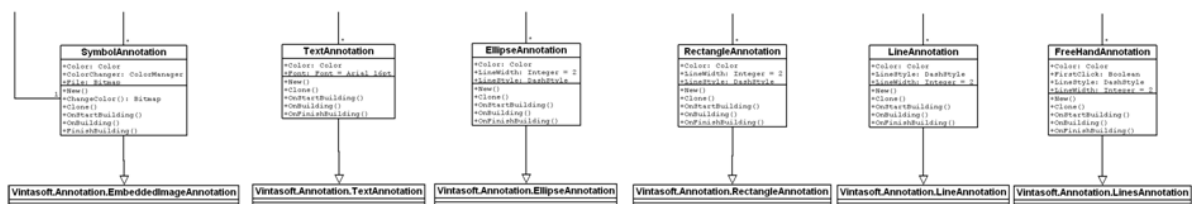
Zoals in hoofdstuk 6.2 al aangegeven was, ondersteunde de externe functionaliteit niet de mogelijkheid om gemaakte acties terug te draaien.

Actie 1		Actie 2	
gegevens oud	positie 0.0, grootte 0.0	gegevens oud	positie 3.8, grootte 5.5
gegevens nieuw	positie 3.8, grootte 5.5	gegevens nieuw	positie 5.6, grootte 5.5
type	aanmaken	type	verplaatsen

Figuur 12: Twee acties in de UndoManager

Hiervoor is de klasse UndoManager aangemaakt. Er is een hoop tijd gestopt in het ontwikkelen van deze klasse. Het eerste probleem was het moment waarop een actie geregistreerd moest worden. Er diende tenslotte gegevens van zowel voor als na de actie opgeslagen te worden om deze terug te kunnen draaien. Dit is opgelost door het afbeeldingsgebied evenementen af te laten vuren als er een bewerking wordt uitgevoerd. Als de bewerking van een type is dat geregistreerd moet worden (bijvoorbeeld aanmaken of verplaatsen), wordt er een actie aangemaakt.

Nu was het alleen nog zaak om de gegevens van de annotatie op te slaan voor de bewerking. Dit is opgelost door op het moment dat een actie toegevoegd wordt, terug te kijken in de lijst naar de meest recente actie die is uitgevoerd op een annotatie. De gegevens na bewerking werden daarna gebruikt als de gegevens voor de nieuwe bewerking (zie ook Figuur 12).



Figuur 13: De herziene annotatieklassen en hun superklassen

Het volgende punt was dat annotaties een aantal standaardinstellingen moesten hebben, bijvoorbeeld de geselecteerde kleur en lijnstijl. Symboolannotaties moesten een vast formaat hebben als de gebruiker enkel klikte (in plaats van klikken en slepen om zelf het formaat te bepalen). Al deze requirements bleken niet voldaan te kunnen worden met de opzet van de annotatieklassen zoals deze bedacht waren in het technisch ontwerp. Hierop is besloten om het klassendiagram aan te passen. De abstracte klasse Annotation verdween en de annotatieklassen erfden vanaf nu direct over van de klassen in de externe functionaliteit (zie Figuur 13).

Het laatste grote probleem dat opgelost moest worden was de gum functie. Omdat een annotatie een object was, zou het erg veel werk zijn om een aantal punten van de annotatie doorzichtig te maken. Er werd geprobeerd om een doorzichtige lijnannotatie over het origineel te leggen in de hoop dat dit de afbeelding weer naar boven zou halen. Dit werkte helaas niet. Gezien de hoeveelheid werk die het zou zijn om per pixel te bepalen of de kleur moet veranderen, is er in overleg met het bedrijf besloten deze functionaliteit te schrappen.

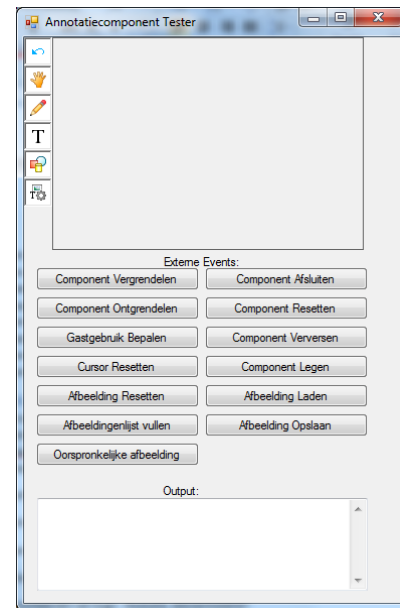
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

De externe functionaliteit bood uitgebreide bewerkingsmogelijkheden ten opzichte van de VB6 versie van de component. Zo uitgebreid, dat de gumfunctie toch niet heel veel zou toevoegen aan de component.

De fase liep van 2 maart tot en met 23 maart. In deze tijd zijn ook de aanpassingen aan het technisch ontwerp gedaan. De keren dat vanuit de testfase terug is gegaan naar de realisatie fase is niet meegeteld.

6.4. Subfase 4: Testen

De testfase begon met het opzetten van een testomgeving. Hoewel de component voltooid was, was deze nog niet geïmplementeerd in het systeem. Dit had als nadeel dat de publieke methodes en properties nog niet getest kon worden. Om dit op te lossen is er een test formulier gemaakt met knoppen die de publieke methodes aanroepen (zie Figuur 14).



Het testformulier was in feite bedoeld om de Norma applicatie te kunnen simuleren. Alle standaardacties (zoals het inladen van een afbeelding, of het vergrendelen van de component) zit in de publieke methodes van de component. Op het moment dat het systeem de component gebruikt, wordt deze dus ingesteld met behulp van deze properties en methodes. Er zijn ook een aantal publieke methodes die resultaten teruggeven aan het systeem bijvoorbeeld de bestandslocatie + naam van een opgeslagen afbeelding als de methode afbeelding opslaan wordt aangeroepen. Deze resultaten werden getoond in het vakje Output, zodat ook gecontroleerd kon worden of de gewenste gegevens teruggestuurd zouden worden.

Figuur 14: Het testformulier

Na het voltooien van het testformulier, is begonnen met het invullen van de checklisten uit het testplan. In Figuur 15 is een deel van deze lijst te zien.

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?	X	Kan niet getest worden			
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?	X	Kan niet getest worden			
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?	OK				

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?	X	Kan niet getest worden			
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?	X	Kan niet getest worden			

Figuur 15: Een gedeelte uit de checklisten van het testplan

Ondanks het testformulier was het niet mogelijk om alle punten in de checklisten als te testen. De items zichtbaar in Figuur 15 konden voornamelijk pas getest worden als de component geïmplementeerd was. Deze stappen zouden door het testteam na de implementatie herhaald moeten worden. De rest van de checklist is allemaal met OK afgerond.

De checklist werd nogmaals nagelopen door iemand van het testteam om er zeker van te zijn dat er niks over het hoofd was gezien. Het is namelijk mogelijk dat punten die een ontwikkelaar als voldoende ziet, misschien niet voldoende is in de ogen van een klant of tester. De tester gaf inderdaad nog commentaar op een aantal punten (zie Figuur 16).

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?		Na instellen van groter lettertype wordt de tekst afgekapt. Ipv van "dingen" wordt "ding" getoond. Indien je het annotatievlak vergroot, wordt de tekst zichtbaar.			

Figuur 16: Commentaar van het testteam

Indien het testteam een dergelijk punt had gevonden, werd er eerst besproken hoe belangrijk dit punt was. Indien dit punt belangrijk genoeg werd bevonden, ging het project terug naar de realisatie fase om het aan te passen (dit is gebeurd met het punt in Figuur 16). Andere punten waren niet belangrijk genoeg om gelijk op te lossen en werden opgeschoven naar de tweede hoofdfase of genegeerd. Een voorbeeld hiervan is het veranderen van de lijnstijl teksten in het menu lijninstellingen in plaatjes met de verschillende stijlen.

Na de checklists is het testteam verder gegaan met het testen van de component met behulp van het testformulier. Hier werd nauwkeuriger gekeken naar het gedrag van de component. Ook hier kwamen nog een aantal dingen uit waarin het gedrag van de component niet gelijk was aan het gewenste gedrag. Dit betrof veelal kleine punten zoals het standaardformaat van symboolannotaties dat veranderde naarmate de resolutie van de afbeelding veranderde. Ook in deze gevallen werd overlegd over hoe dringend de gevonden fout was en of directe aanpassing noodzakelijk was.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Er waren ook een aantal meer ernstige punten. Een goed voorbeeld hiervan was de ongedaan maken functie. Het kwam regelmatig voor dat er twee keer op de knop ongedaan maken geklikt moest worden om een actie terug te draaien. Soms werd dit veroorzaakt doordat de gebruiker bijvoorbeeld een annotatie 360 graden roteerde, wat natuurlijk geen merkbaar verschil oplevert als de actie teruggedraaid werd. Om dit tegen te gaan werd er een extra check in de UndoManager geplaatst die controleerde of de bewerking geen 360 graden rotatie was. Was dit wel het geval, dan werd de actie niet geregistreerd. In andere gevallen werden er twee acties geregistreerd doordat de annotatie bijvoorbeeld van grootte veranderde bij het plaatsen. Dit was voornamelijk het geval als er een symboolannotatie geplaatst werd die kleiner was dan het minimumformaat. In dat geval wordt een annotatie namelijk uitvergroot tot het minimum, wat een extra actie opleverde om ongedaan te maken. Hiervoor is een extra controle in de AnnotationComponent klasse gebouwd die zorgt dat bewerkingen op een annotatie niet geregistreerd worden als de annotatie nog gebouwd wordt.

Een minder belangrijk, maar wel buitengewoon onhandig punt, was dat tekstannotaties niet altijd goed geplaatst werden. Met name langere teksten werden afgekapt omdat de ruimte van de annotatie beperkt werd. Van het woord “dingen” was dan bijvoorbeeld alleen “ding” zichtbaar (hetzelfde werd ook aangegeven door de testers in Figuur 16). Geprobeerd werd om met behulp van een berekening (De lengte van de tekst vermenigvuldigd met de lettergrootte) dit te omzeilen. Een aantal varianten hierop waren of te groot of te klein. De huidige variant is eigenlijk ook niet helemaal goed (het resultaat is vaak te groot), maar de hele tekst is wel zichtbaar en dat was het belangrijkste. Verder kan de gebruiker na het bouwen toch zelf het formaat nog aanpassen voor als deze er niet tevreden over is.

Het was ook de bedoeling dat de component verder getest zou worden nadat deze geïmplementeerd was in het systeem. Hoewel de implementatie uiteindelijk voltooid was, is het testen van de component in het systeem niet voltooid. Dit had te maken met een tekort aan testcapaciteit. In de momenten dat er wel enigszins getest kon worden, waren er nog wel een aantal problemen gevonden. Deze fouten hadden vooral te maken met de communicatie tussen het .Net component en de VB6 omgeving waarin deze was geplaatst. Meer hierover in hoofdstuk 6.5.

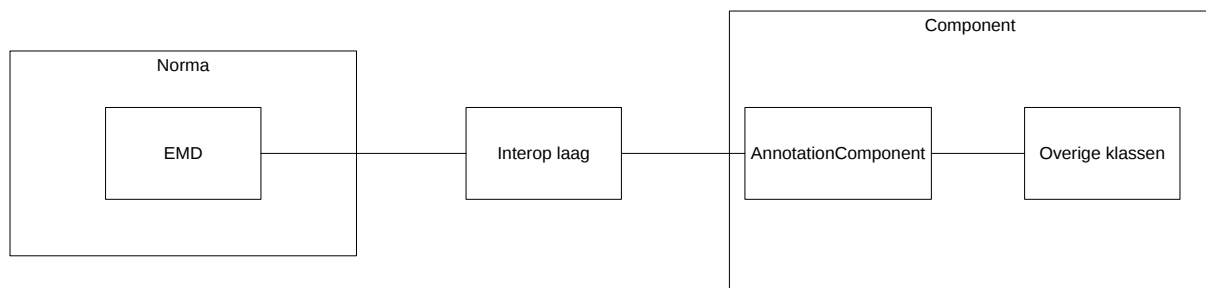
Deze fase was oorspronkelijk gepland tussen 23 maart en 1 april. Echter, door het gebrek aan capaciteit bij het testteam en een inschattingsfout in de planning, is er al begonnen aan de volgende fases vanaf 28 maart.

6.5. Subfase 5: Implementatie

Na het voltooien van de testen, is begonnen met de implementatie van de component in het systeem. In deze fase is vooral ondersteuning verleend aan de ontwikkelaars die de component in het systeem zouden plaatsen. Met ondersteuning wordt bedoeld het verhelpen van problemen met de communicatie tussen VB6 en .NET.

Het implementeren werd bemoeilijkt door het feit dat er twee verschillende talen gebruikt waren voor de twee onderdelen. Hierdoor was het moeilijk te bepalen of een fout zich aan de VB6 kant of de .Net kant bevond, omdat maar één van de twee kanten tegelijkertijd in Debug modus gedraaid kon worden. Dit betekende dat er goede communicatie onderhouden moest worden met degenen die verantwoordelijk waren voor de implementatie.

Een voorbeeld van een communicatieprobleem was de veranderde manier van opslaan. In de originele versie was de component zelf verantwoordelijk voor het plaatsen van de afbeelding in de database. Echter, voor de nieuwe versie was afgesproken om dit door het overkoepelende systeem te laten regelen. Hiervoor stuurt de methode een regel tekst met de bestandsnaam en locatie terug. Dit betekende dus dat het systeem aangepast moest worden om de tekst af te vangen en vervolgens de afbeelding op te halen en in de database te plaatsen. Ditzelfde probleem gold ook in omgekeerde volgorde voor de afbeeldingen laden. De afbeelding moest dus eerst uit de database gehaald worden en vervolgens doorgegeven aan de component.



Figuur 17: Implementeren van de component

Om communicatie tussen de .Net component en het VB6 systeem mogelijk te maken, was het nodig om een extra laag te maken tussen de twee delen. Deze laag is een zogeheten Interop laag, speciaal ontworpen om .Net code geschikt te maken voor VB6 (5). In deze laag zijn alle properties en publieke methodes uit de component opnieuw opgenomen. De VB6 code roept de methoden en properties van de interop laag aan en de interop laag roept vervolgens dezelfde methode van de component aan. Resultaten teruggeven gaat op dezelfde wijze. Deze wijze houdt in dat de interop laag aangepast moet worden als er extra publieke methoden of properties aangemaakt worden in de component. Maar als de code achter een property of methode wordt aangepast hoeft de interop laag dus niet aangepast te worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

De implementatie heeft behoorlijke vertraging opgelopen tijdens het project. Dit had te maken met de planning van het bedrijf en het onderschatten van de hoeveelheid werk die deze fase met zich mee zou brengen.

De ergste onderschatting was de communicatie tussen VB6 en VB.Net via Interop. Zoals eerder beschreven kan met deze methode maar één kant tegelijk getest worden (ofwel de VB6 kant danwel de VB.Net kant), wat er voor zorgde dat er continu overlegd en gediscussieerd moest worden met de persoon die verantwoordelijk was voor de implementatie. Ook ging er veel tijd zitten in het bruikbaar maken van de Interop laag voor VB6. Deze diende namelijk geregistreerd te worden en dit proces kostte veel tijd en moeite om goed uit te kunnen voeren.

In de tussentijd is er al verder gegaan met het ontwikkelen van de nieuwe functionaliteiten uit hoofdfase 2. De fase was gepland van 4 april tot en met 8 april. Dit gedeelte van de planning was dan ook niet haalbaar. Inmiddels is de component wel geïmplementeerd in het systeem.

6.6. Subfase 6: Schrijven van documentatie

Tijdens het testen van de component is ook alvast begonnen met het schrijven van de gebruikershandleiding. Dit was niet volgens de planning, maar het was op dat moment niet mogelijk om de testfase te versnellen vanwege de redenen genoemd in 6.4. In de gebruikershandleiding staat alle functionaliteit van de component beschreven in termen die voor iedereen te begrijpen zijn.

Er bestond al een handleiding voor de annotatiecomponent. Deze zou als basis gediend kunnen hebben voor de huidige handleiding, maar dit is niet gebeurd. Dit kwam voornamelijk doordat de functionaliteit van de component, veranderd is in de vorm van gebruik. De functionaliteit is in essentie gelijk gebleven, maar doordat de externe functionaliteit dingen met betrekking tot bewerkingen van annotaties op een andere manier aanpakt dan de oorspronkelijke versie van de component, voelt het werken met de component toch iets anders aan dan de oude versie. Ook was het bedrijf niet tevreden met de huidige versie van de handleiding door veranderde richtlijnen.

Na de gebruikersdocumentatie is er verder gegaan met de beheerdocumentatie. Deze documentatie is vooral intern bedoeld en gemaakt om onderhoud aan de component te vereenvoudigen. In de beheerhandleiding wordt een beschrijving gegeven van elk onderdeel van de component. Dit kan een klasse zijn, of een structure of enumerator die gebruikt wordt (zie Figuur 18).

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Public Structure AnnotationData Public Guid As Guid Public Location As PointF Public Size As SizeF Public Rotation As Single Public AnnotationType As String Public Visible As Boolean End Structure	In deze structure worden alle gegevens van een annotatie genoteerd voor de UndoManager. De volledige annotatie kan niet opgeslagen worden omdat alle versies van de annotatie worden bijgewerkt als deze veranderd. In plaats daarvan worden alleen relevante eigenschappen van de annotatie opgeslagen. Met andere woorden, de eigenschappen die ongedaan gemaakt kunnen worden. Als de UndoManager uitgebreid moet worden met extra acties, dienen de gegevens die nodig zijn voor de actie opgenomen te worden in AnnotationData. Alle eigenschappen die geregistreerd worden, hebben hetzelfde type en dezelfde naam als hun tegenhangers in de annotatie. Location in de annotatie is een PointF, dus de registratie in de AnnotationData wordt ook een PointF genaamd Location. De enige uitzondering hierop is AnnotationType, die alleen de naam van de klasse bevat waaruit de annotatie is opgemaakt (bijvoorbeeld ctlAnnotation.EllipseAnnotation)
---	---

Figuur 18: Structure AnnotationData uit de beheerhandleiding

Het idee achter deze handleiding was dat ontwikkelaars snel konden zien welke verantwoordelijkheden bij welke klasse lagen, zonder eerst de ontwerpdocumentatie door te hoeven lezen (voor wanneer ze bijvoorbeeld een fout zouden moeten repareren). Het kan dus eigenlijk gezien worden als een soort samenvatting.

Oorspronkelijk was deze fase gepland van 11 april tot en met 15 april. Doordat de planning op dat moment al in de war was, was deze fase veel eerder afgerond dan oorspronkelijk gedacht. De lengte van de fase is echter wel gelijk gebleven.

7. Hoofdfase 2: Toevoegen nieuwe functionaliteit

In deze fase lag de nadruk voornamelijk op het toevoegen van extra functionaliteit aan de component. In de loop van de jaren zijn er verschillende wensen geuit door klanten. Deze wensen waren opgenomen in de requirements van het project (zie hoofdstuk 5). Zoals in hoofdstuk 4 beschreven, was het toevoegen van nieuwe functionaliteit gescheiden van het overzetten van de bestaande functionaliteit. Deze hoofdfase liep van 6 april tot en met 10 mei. Dit is veel eerder en sneller dan oorspronkelijk gepland (zie Figuur 4). Dit heeft weer te maken met het gebrek aan testcapaciteit en de planning van het bedrijf.

7.1. Subfase 1: Opstellen Functioneel Ontwerp

Om ervoor te zorgen dat de nieuwe functionaliteit zo goed mogelijk aansluit op de bestaande functionaliteit, is er opnieuw een functioneel ontwerp gemaakt. Dit functioneel ontwerp (11) is alleen toegespitst op de uitbreidingen. De huidige functionaliteit is niet nogmaals uitgemeten en daarom moet dit functioneel ontwerp naast het origineel gehouden worden (7).

Een aantal van de requirements voor de uitbreiding (zie Figuur 5) zijn niet ontstaan door wensen van klanten. Een voorbeeld hiervan is het herhalen van ongedaan gemaakte acties. De suggestie voor deze functionaliteit is ontstaan na het maken van de UndoManager klasse. Dankzij de opzet van deze klasse zou het erg eenvoudig moeten zijn om deze functionaliteit toe te voegen. Na overleg is dan ook besloten om deze requirement toe te voegen.

Een andere, zeer belangrijke requirement was de mogelijkheid om geplaatste annotaties na het opslaan alsnog te kunnen bewerken. Tot nu toe werden de annotaties bij het opslaan samengevoegd met de afbeelding tot één bestand en vervolgens opgeslagen. Deze methode had als nadeel dat het niet omgekeerd werkte. De annotaties konden dus niet meer losgehaald worden van de afbeelding.

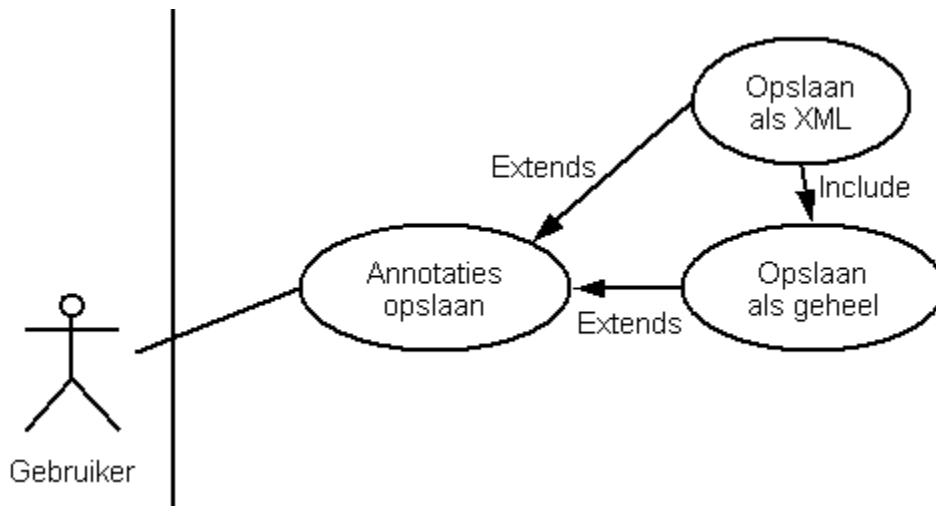
Naam	Annotaties opslaan → Opslaan als XML
Beschrijving	Het in een los bestand opslaan van de annotaties zodat deze later alsnog bewerkt kunnen worden.
Aannamen	Er is tenminste één annotatie geplaatst. De instelling 'gescheiden opslaan van annotaties' moet ingeschakeld zijn.
Resultaat	De annotaties zijn opgeslagen
Stappenplan	1. De gebruiker klikt op [Opslaan]. 2. De annotaties worden in een XML bestand gezet. 3. De Use Case Opslaan als geheel wordt uitgevoerd. 4. De bestandslocaties en bestandsnamen van het XML bestand en de afbeelding worden teruggegeven aan het EMD. 5. Het EMD plaatst de bestanden in de database.
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Het was gelukkig niet nodig om zelf de functionaliteit te ontwikkelen die ervoor zorgde dat de annotaties los van de afbeelding opgeslagen konden worden. De externe functionaliteit bood namelijk de mogelijkheid om annotaties op te slaan als een XML bestand. Door deze functionaliteit te gebruiken werd het dus mogelijk om aan deze requirement te voldoen. Helaas leidde dit wel tot een conflict met een andere

Figuur 19: Use Case Opslaan als XML

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

requirement, namelijk de eis dat alle bestaande functionaliteit behouden moest blijven. De bestaande opslaan functionaliteit kon dus niet zomaar veranderd worden. Hierdoor was het nodig om de Use Case Afbeelding Opslaan uit het oorspronkelijke functioneel ontwerp dus aan te passen (zie Figuur 20).



Figuur 20: Use Case Annotaties opslaan

Voor de uitbreiding is ook een nieuw testplan gemaakt. Als basis hiervoor is hetzelfde testplan gebruikt als voor het testplan voor de bestaande functionaliteit. In het testplan is wel aangegeven dat de bestaande functionaliteit niet zo uitgebreid getest hoeft te worden. Alle bestaande functionaliteit zou al getest zijn voordat deze versie van de component voltooid is. Het was wel belangrijk om de bestaande functionaliteit nog even snel na te kijken om te controleren of er geen nieuwe fouten ontstonden door het toevoegen van de nieuwe functies.

De fase liep van 4 april tot en met 8 april. Dit was één dag langer dan gepland, maar de fase is veel eerder uitgevoerd dan de oorspronkelijke planning (deze was van 25 april tot en met 27 april), waardoor er nog voorgelopen werd op de totale planning.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

brengen, zou te groot kunnen zijn om de splitsing te verantwoorden. Hierom is eerst het Request for Change opgesteld, om de opdrachtgever op de hoogte te brengen van de risico's en voordelen van de splitsing.

Nadat het Request for Change was goedgekeurd, werd een deel van de klasse AnnotationComponent overgeplaatst in een nieuwe klasse genaamd AnnotationManager. AnnotationManager (zie Figuur 23) kreeg alle verantwoordelijkheden voor de annotaties, terwijl AnnotationComponent (zie Figuur 22) de verantwoordelijkheden voor de communicatie met systeem en gebruiker behield. Dit veranderde klassendiagram leidde tot een nieuwe versie van het technisch ontwerp (14), dat vervolgens uitgevoerd is in de realisatiefase in hoofdstuk 7.3.

AnnotationComponent
-c_highlightColor: Color - _colorChanger: ColorManager - _textDialog: TextInputDialog - _lineDialog: LineDialog - _imageDialog - b_textSet: Boolean - _customSymbols: CustomSymbolManager - s_bRetrievePosition: String - b_eigendomOverTeNemen: Boolean - b_controlLocked: Boolean - s_itemOptions: String - i_subTypeItem: Integer - i_selectedSubType: Integer - s_writeOnlyParameters: String - b_gastGebruik: Boolean - d_otherImages: Dictionary(Of Integer, String) - d_otherSymbols: Dictionary(Of Integer, String) - b_saveSeperate: Boolean
-RegisterVintasoftDLL() -Initialize() -InitializeBooleanValues() -InitializeDictionaries() -InitializeDialogs() -ChangeCMSymbolImages() -MarkPictureBox() -InternalClear() +LockControl() +UnlockControl() +SetGastGebruik() +ResetMousePointer() +ResetCompletely() +Shutdown() +Refresh() +Clear() +ResetPicture() +LoadList(Options: String) +LoadCustomAnnotation(Options: String) +ReloadCustomSymbolMenu() +LoadInitialPictureFile() +LoadPictureFile(FileName: String) +SavePictureFile(): String +TriggerCustomAnnotations()

Figuur 22: De klasse AnnotationComponent na de splitsing

AnnotationManager
-c_selectedColor: Color - _colorChanger: ColorManager - f_selectedFont: Font - i_selectedLineWidth: Integer - _scrollHand: Vintasoft.Imaging.VisualTools.Pan - _cursorChanger: CursorManager - _undoController: UndoManager - b_undoing: Boolean - _currentlySelected: SelectionOptions - b_placeText: Boolean - b_annotationSelected: Boolean - b_wasPictureEdited: Boolean - s_imageFile: String - s_initialImageFile: String - b_imageSaved: Boolean - s_annotationText: String - _selectedLineStyle: DashStyle - _customSymbol: Bitmap - _annotationViewer: Vintasoft.Annotation.AnnotationViewer
-Initialize() -InitializeLineSettings() -InitializeTextSettings() -InitializeManagers() -InitializeBooleanValues() -InitializeHand() +SaveImage(): String +SaveAnnotationsSeperate(): String +LoadPictureFile(FileName: String) +LoadInitialPictureFile() -Resample() +LoadSavedAnnotations() +CreateSymbolAnnotation(Image: Bitmap) +CreateCustomSymbolAnnotation(Image: Bitmap) +CreateEllipseAnnotation() +CreateRectangleAnnotation() +CreateFreehandAnnotation() +CreateLineAnnotation() +CreateTextAnnotation(Text: String) +ActivateHand() +StopBuilding() +Clear() +ResetMousePointer() +ResetPicture() -IsAnnotationSelected(sender: Object, e: EventArgs): Boolean +ChangeSelectedAnnotationColorSettings(NewColor: Color) +ChangeSelectedAnnotationLineSettings(NewLineStyle: DashStyle, NewLineWidth: Integer) +ChangeSelectedAnnotationFont(NewFont: Font) -AnnotationControllerChanged(sender: Object, e: Vintasoft.Annotation.AnnotationControllerChangedEventArgs) -RepeatAnnotation() -CancelBuilding(sender: Object, e: MouseEventArgs) +UndoAction() +RedoAction() -DeleteKeyPressed(sender: Object, e: EventArgs)

Figuur 23: De nieuwe klasse AnnotationManager

De fase was oorspronkelijk gepland van 27 april tot en met 29 april. De fase is echter al uitgevoerd tussen 8 april en 10 april. Het aanpassen van het technisch ontwerp samen met het Request for Change heeft plaatsgevonden van 21 april tot en met 24 april. In de tussentijd was al verdergegaan met de realisatiefase in 7.3.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

7.3. Subfase 3: Realisatie van het ontwerp

Tijdens het afronden van het technisch ontwerp werd al begonnen met het opschonen en optimaliseren van de code. Op een aantal plaatsen stond nog code die in een comment geplaatst was omdat deze niet meer gebruikt werd. Ook zijn op een aantal plaatsen methoden opgesplitst in kleinere methoden om het overzicht te bewaren. Een voorbeeld hiervan is het venster voor de afbeeldinginstellingen. Door deze op te schonen werd het venster beter onderhoudbaar en makkelijker te bewerken. Het opschonen van de code is gebeurd met behulp van de regels in (15)

Het bouwen van het ontwerp is begonnen met het inbrengen van het singleton pattern in de klasse ColorManager. Dit omdat het invoeren hiervan een aanpassing is die weinig tijd kost en weinig invloed heeft op de rest van de component. Vervolgens is een aantal van de klassen opgeschoond. Grote methodes zijn opgesplitst in meerdere kleine methodes. Het opsplitsen van de methodes had als doel de onderhoudbaarheid van de component te vergroten.

De volgende stap was het uitbreiden van de UndoManager, zodat het mogelijk werd om ongedaan gemaakte acties opnieuw uit te voeren. Deze methode was precies het omgekeerde van de methode UndoAction.

Hierna werd gewerkt aan het bewerken van afzonderlijke annotaties. Hiermee moest het voor de gebruiker mogelijk worden om van de geselecteerde annotatie de kleur en, afhankelijk van het type annotatie, de lijnstijl dan wel het lettertype aan te passen. Hier ontstond het probleem dat de kleur van symboolannotaties niet veranderde als er een andere kleur werd gekozen. Het probleem bleek te liggen bij de kleurmethode van de ColorManager. Deze veranderde de kleur van een pixel alleen als deze een bepaalde kleur had. Doordat de geplaatste symboolannotaties deze kleur niet meer hadden, veranderden ze ook niet van kleur. Dit is opgelost door een extra methode toe te voegen aan de ColorManager, die controleert of een pixel niet transparant is. Is dit niet het geval, dan wordt de kleur van de pixel veranderd. Dit kon niet gebruikt worden voor het eerste keer plaatsen van symboolannotaties, omdat deze nog geen transparante pixels had. De transparante pixels worden gemaakt tijdens de eerste kleurverandering (wat gebeurt tijdens het plaatsen).

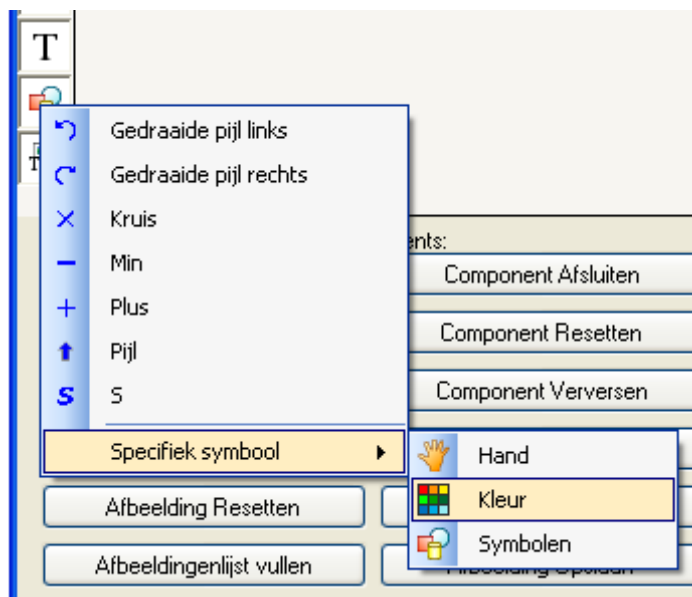
Het veranderen van lijnstijlen of lettertypen werd zonder problemen ingevoerd.

Hierna werd de klasse CustomSymbolAnnotation gemaakt. Aangezien deze (op de ColorManager na) identiek was aan de klasse SymbolAnnotation, waren er geen problemen met het maken van deze klasse. De moeilijkheden hier lagen in het integreren van de specifieke symbolen in de UI. Het probleem was namelijk dat er een variabel aantal symbolen ingeladen kon worden. Het toevoegen van een specifiek symbool gebeurde op een andere plek dan in de component. Het probleem was dan ook de manier waarop de gebruiker de annotatie kon kiezen. Eerst was er een apart venstertje ontworpen waarin de

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

annotaties werden getoond en waar de gebruiker kon kiezen. Dit idee werd echter afgekeurd omdat het te onhandig zou kunnen zijn voor de gebruiker. Bij voorkeur zou de gebruiker de annotatie kunnen kiezen met behulp van een contextmenu.

Alle bestaande contextmenu's waren echter statische menu's gedefinieerd in de Designer van Visual Studio. Het was dus niet mogelijk om dezelfde methode te gebruiken voor dit menu, omdat onbekend was hoeveel en welke symbolen er beschikbaar waren. Het was dus nodig om een dynamisch menu te maken. Hiervoor is een aparte klasse ontworpen, die regelt dat er een contextmenu toegevoegd wordt en voor elke annotatie een extra item in die lijst maakt. Deze CustomSymbolManager wordt gekoppeld aan een item in het symbool contextmenu.



Figuur 24: Het CustomSymbolMenu in de component

Vervolgens is er gewerkt aan het opslaan van annotaties als een XML bestand. Om dit te bewerkstelligen is de bestaande methode voor opslaan opgesplitst in twee methodes. De methode die eerst wordt aangeroepen (SavePictureFile) voert een controle uit of het nodig is om de annotaties los op te slaan. Afhankelijk van het resultaat worden de juiste opslagmethodes aangeroepen. De andere methode (SaveImage) zorgt ervoor dat de afbeelding en de annotaties samengevoegd worden en slaat het resultaat op in een tijdelijke map. Vervolgens wordt de locatie en bestandsnaam teruggegeven. Als de annotaties wel los opgeslagen moeten worden, wordt een andere methode (SaveAnnotationsSeperate) aangeroepen die daarvoor zorgt.

Tijdens het opslaan ontstond het probleem dat symbool annotaties en specifieke symbool annotaties na het opslaan niet meer werden ingeladen. Toch waren ze wel terug te vinden in het XML bestand. Ook ontstonden er geen foutmeldingen tijdens het inladen van de XML bestanden waardoor niet na te gaan was wat er precies fout ging tijdens het inladen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Om dit probleem op te lossen is weer een beroep gedaan op het support forum van de externe functionaliteit (16) met de vraag of zij begrepen wat er fout ging. Na enig overleg met de medewerkers van VintaSoft, kwamen zij met een oplossing.

Om symboolannotaties vanuit een XML bestand in te kunnen laden, was het nodig om een extra override van de constructor te maken. De constructor van de SymbolAnnotation en CustomSymbolAnnotation klassen hebben namelijk een afbeelding als parameter. Doordat de afbeeldingen worden ingeladen vanuit een XML bestand, kennen deze versies de parameter niet. Door een override van de constructor te doen zonder de parameter, is de externe functionaliteit zelfstandig in staat de afbeelding in te laden in het object. Hierdoor was het weer mogelijk om symboolannotaties in te laden na opslaan.

Ook al worden de annotaties los opgeslagen, dan nog moest er ook een samengevoegde versie van de afbeelding opgeslagen worden. Dit komt omdat het systeem in staat moet zijn om automatisch brieven te kunnen genereren en daarin kan een annotatieafbeelding opgenomen worden. Daarom voert SaveAnnotationsSeperate zelf ook de methode SaveImage uit. Hier waren wat problemen met de timing. Als namelijk de afbeelding eerst werd samengevoegd en daarna opgeslagen verdwenen namelijk alle geplaatste annotaties. Maar als eerst de annotaties werden opgeslagen en daarna de afbeelding werd samengevoegd konden de annotaties niet meer bewerkt worden omdat de samengevoegde afbeelding was ingeladen. Dit betekende dus dat na het opslaan, de standaardafbeelding opnieuw ingeladen moest worden en vervolgens worden de annotaties opnieuw ingeladen. De SaveAnnotationsSeperate methode stuurde vervolgens zowel de bestandsnaam en locatie van het XML bestand alsmede de bestandsnaam en locatie van de samengevoegde afbeelding terug. Het EMD plaatst de bestanden vervolgens in de database.

Het splitsen van de klasse AnnotationComponent was tot het laatst bewaard, voornamelijk omdat bij enige tegenslag het splitsen heel veel tijd in beslag kon nemen, die wellicht beter besteed had kunnen worden. Ook moest er gewacht worden op goedkeuring van het Request for Change in 7.2 Om de splitsing te kunnen maken is eerst de nieuwe klasse AnnotationManager gemaakt. Deze klasse is een UserControl. Hierdoor kon de nieuwe klasse met behulp van de Visual Studio designer ingebracht worden in de UI. Voordeel hiervan is dat deze Control herbruikbaar is als de UI vervangen moet worden.

Vervolgens is begonnen met het overzetten van methodes en variabelen van de klasse AnnotationComponent naar de nieuwe AnnotationManager. De overgeplaatste methoden in de AnnotationManager werkten bijna allemaal direct. Er waren slechts enkele kleine dingen zoals namen van variabelen die hier en daar aangepast moesten worden. De methoden die achterbleven in de AnnotationComponent werkten een stuk minder goed. Een hoop methoden maakten gebruik van variabelen die nu niet meer bestonden. Ook bleef er een hoop overbodige code achter. In plaats van proberen de bestaande code net zolang aan te passen totdat deze wel werkte, is gekozen om een aantal methoden van de grond af aan

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

opnieuw op te bouwen. Dit was minder tijdrovend dan proberen de oude methoden te repareren en verminderde de kans dat er fouten achter bleven.

De fase was oorspronkelijk gepland van 2 mei tot en met 13 mei. In werkelijkheid liep de fase van 8 april tot en met 9 mei. De fase duurde aanzienlijk langer dan gepland door het Request for Change en de uitvoering ervan.

7.4. Subfase 4: Testen

Na het voltooien van de component is er weer een testfase opgestart. Zoals in hoofdstuk 7.1 al was aangegeven, was deze testfase vooral gericht op de toegevoegde functionaliteit. Vooral dankzij de splitsing van de AnnotationComponent klasse bleef het echter wel belangrijk om de bestaande functionaliteit opnieuw te controleren.

Net als in de vorige testfase, begon de testfase met het uitvoeren van de checklist die was opgesteld in het functioneel ontwerp. Om de checklist uit te kunnen voeren werd weer gebruik gemaakt van hetzelfde testproject als de vorige keer, met als extra toevoeging een knop om een aantal specifieke symboolannotaties in te laden (dit was een publieke methode die toegevoegd was. Zie hoofdstuk 7.3).

In Figuur 25 is een item uit de checklist te zien. Het opvallende aan dit punt is dat deze is goedgekeurd terwijl niet aan alle criteria in het item voldaan wordt. Vooraf was afgesproken dat de kleur van specifieke symboolannotaties niet veranderd kon worden (zie hoofdstuk 7.2). Daarom was dit type annotatie buiten beschouwing gelaten tijdens het uitvoeren van de checklist. Omdat dit mogelijk onduidelijk zou kunnen zijn voor anderen die de checklist nalopen, is dit in de bevindingen erbij gezet. Achteraf had het item zelf aangepast moeten worden omdat niet zeker is of een ander de resultaten ook zou lezen.

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Is de kleur van de geselecteerde annotatie aan te passen?	OK	Met uitzondering van gespecialiseerde symbolen (klopt)			

Figuur 25: een deel uit de checklist voor de nieuwe functionaliteit

Hierna was het de bedoeling dat de testers weer verder zouden gaan met het testen van de component. Echter vanwege dezelfde redenen als in hoofdstuk 6.4 is het testen uitgesteld. De prioriteit lag nog bij het testen van de eerste versie van de component. Dit komt omdat deze volgens de regels van het bedrijf eerst goed geïmplementeerd moest zijn voordat de uitbreidingen geïmplementeerd mochten worden.

Deze fase was oorspronkelijk gepland van 16 mei tot en met 20 mei. De checklists zijn al uitgevoerd op 4 mei. Hierna zijn nog enkele fouten opgelost. Momenteel ligt de component bij het testteam welke nog niet klaar is met het testen ervan. Het is dus nog mogelijk dat er fouten gevonden worden, waardoor de fase nog niet volledig is afgerond.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

7.5. Subfase 5: Implementatie

Zoals in hoofdstuk 7.4 al aangegeven, kon de nieuwe versie van de component pas geïmplementeerd worden nadat de implementatie van de eerste versie voltooid en de nieuwe versie volledig getest was. Dit is voor het einde van de opdracht niet meer gelukt.

Om er toch voor te zorgen dat de implementatie van de nieuwe versie zo soepel mogelijk verloopt, is ervoor gezorgd dat er nauwelijks toevoegingen of aanpassingen zijn aan de bestaande publieke methoden en properties. De enige aanpassingen zijn de methode `Savelmage`, die afhankelijk van een instelling één of meerdere bestandsnamen en locaties geeft, een extra property om aan te kunnen geven of annotaties los opgeslagen moeten worden en een methode om specifieke symboolannotaties toe te voegen. Hierdoor zijn er minimale wijzigingen aan het systeem nodig om de nieuwe versie te kunnen implementeren.

De fase was oorspronkelijk gepland op 23 mei tot en met 25 mei. Omdat de eerste versie van de nieuwe component echter nog niet volledig geïmplementeerd was, is deze fase nog niet gestart. Inmiddels is de eerste versie succesvol geïmplementeerd, en wordt er nu gewerkt aan het testen van de tweede versie. Indien het testen succesvol is zal de component alsnog opnieuw geïmplementeerd worden. Dit zal echter niet meer in het project zelf gebeuren.

7.6. Documentatie

In het laatste deel van deze hoofdfase is de gebruikers en beheer documentatie uit hoofdstuk 6.6 bijgewerkt. In de aangepaste versie van de gebruikershandleiding is de extra functionaliteit die is toegevoegd (zoals de herhaal functie) opgenomen (17).

In de aangepaste beheer handleiding (18) is te zien wat de functies zijn van de twee klassen die zijn ontstaan na de splitsing van de klasse `AnnotationComponent` (zie hoofdstuk 7.2)

8. Afwijkingen ten opzichte van het Afstudeerplan

8.1. Ontwerpen en bouwen van een database

Oorspronkelijk werd gedacht dat er een aanpassing / toevoeging aan de bestaande database nodig was om de nieuwe manier van annotaties opslaan te kunnen ondersteunen. Echter, in de loop van het project is gebleken dat dit door het bedrijf zelf geregeld zou gaan worden. Dit had vooral te maken met het feit dat het bedrijf momenteel bezig is met een overstap naar een ander DBMS. Het bedrijf maakte in eerste instantie gebruik van het DBMS Pervasive, maar schakelt nu over op Microsoft SQL Server 2008.

Ook bleek dat de component zelf niet communiceert met de database, maar afbeeldingen en annotaties opslaat in een lokale tijdelijke map. Vervolgens wordt de locatie van deze bestanden teruggegeven aan het systeem waar de component in geplaatst is (het EMD, Elektronisch Medisch Dossier), die vervolgens de bestanden in de database plaatst. De ontwikkeling van de component stond helemaal los van het oorspronkelijke systeem, met uitzondering van publieke methoden en properties, die door het EMD aangeroepen konden worden en viel daarom niet in de scope van het project. Hierdoor is het niet gelukt deze beroepstaken (beroepstaak 2.1 en 2.2), die oorspronkelijk gepland stonden op niveau 4 en 3, te behalen.

8.2. Planning

Voor aanvang van het project werd er al een ruwe planning opgesteld. In deze planning werd er nog vanuit gegaan dat het project in één grote fase uitgevoerd zou gaan worden.

Tijdens het opstellen van de risicoanalyse in het PID werd duidelijk dat er een risico was dat er niet genoeg tijd zou zijn om alle functionaliteit te voltooien. Dit zou betekenen dat het bedrijf mogelijk achterbleef met een niet-werkend component aan het einde van het project. Omdat dit niet wenselijk was, is er gekeken naar methodes om de impact van dit risico te verlagen.

De meest eenvoudige methode leek het aanpassen van de planning te zijn. Door de planning op te splitsen in een fase waarin de bestaande functionaliteit wordt overgebracht en een fase waarin nieuwe functionaliteit wordt toegevoegd, zou het bedrijf in het minst gunstige geval achterblijven met een component dat gelijkwaardige functionaliteit heeft ten opzichte van de oorspronkelijke VB6 versie.

Behalve tijd, had het opsplitsen nog andere voordelen. Zo zorgde de splitsing voor een extra punt waarop gekeken kon worden of de component nog aansloot op de wensen en eisen van het bedrijf en de klanten. Door de nieuwe functionaliteit pas later toe te voegen, was er extra tijd om de bestaande functionaliteit te testen terwijl er verder gewerkt werd aan de nieuwe functionaliteit.

9. Evaluatie

In dit hoofdstuk wordt teruggeblikt op de resultaten van het project. Er wordt uitgelegd wat goed ging in het project, wat minder goed ging en hoe het anders gedaan had kunnen worden. Omdat dit een nogal breed onderwerp is, is dit hoofdstuk opgesplitst in twee delen. Eerst wordt het resultaat bekeken, de voltooide component en bijbehorende documentatie. Vervolgens wordt gekeken naar het proces waarmee het product tot stand kwam.

9.1. Product

De voltooide component is uiteindelijk erg goed gelukt. Het bedrijf en de begeleiders zijn er zeer tevreden over en er zijn tijdens de implementatie weinig ernstige fouten gevonden. De documentatie is ook in orde. Iemand van buitenaf zou de functionele en technische ontwerpen van de component moeten kunnen gebruiken om een replica van de component te kunnen maken.

Natuurlijk zijn er ook dingen minder goed gegaan. Door een ontwerpfout kreeg de klasse AnnotationComponent veel te veel verantwoordelijkheden. Dit leidde ertoe dat de klasse nagenoeg onbeheersbaar was toen de realisatie van de tweede hoofdfase begon (zie hoofdstuk 7.2). Uiteindelijk was er een Request for Change en een tweede iteratie van het technisch ontwerp nodig om dit te kunnen verbeteren. De splitsing heeft echter wel het beoogde resultaat gehaald.

Een ander punt is de ongedaan maken functionaliteit. Zodra opgemerkt werd dat deze zelf gemaakt moest worden doordat de externe functionaliteit hier geen oplossing voor had, werd al gewaarschuwd dat dit een zeer intensieve onderneming zou worden. Dit bleek ook te kloppen. Er is erg veel ontwikkeltijd in de klasse gaan zitten, waar verschillende ontwerpen ontstonden, maar die bij uitvoer niet leken te werken (zie hoofdstuk 6.2). Toen uiteindelijk wel een goede methode was gevonden om de annotaties op te slaan, was het volgende probleem het moment van registreren. Zoals verwacht was er dus veel tijd en aandacht besteed om deze functie in te kunnen bouwen. Dit liep allemaal wel goed met de planning, maar er was vooraf geen rekening gehouden met het feit dat dit zoveel werk zou kunnen kosten.

Verder was er de gum functie. Zoals in hoofdstuk 6.3 beschreven, bleek de gumfunctie erg veel werk te gaan kosten om te realiseren. Dit zorgde voor enige tegenstrijdigheid in het bedrijf. Aan de ene kant was de gumfunctie wel een requirement omdat alle bestaande functionaliteit uit de VB6 versie bewaard moest blijven, en de consultant begeleider vond dan ook dat deze wel gemaakt moest worden. Aan de andere kant werd de gumfunctie door de directeur en de ontwikkelaar begeleider niet belangrijk genoeg gevonden om er zoveel tijd en aandacht aan te besteden. Na overleg tussen de twee begeleiders werd besloten om de functionaliteit toch te schrappen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

9.2. Proces

Ook het proces dat leidde tot de voltooide component verliep zo goed als gehoopt had kunnen worden. Er waren niet veel problemen en de paar die er wel waren hadden voornamelijk met planning te maken.

Een groot deel van de planning die opgesteld was in het PID, werd ook daadwerkelijk nageleefd. Hoofdfase 1 verliep exact volgens schema tot aan de fase testen. Vanaf dat moment werd het lastig om de planning na te leven doordat de testers ook hun eigen planning hadden die nageleefd moest worden. Hetzelfde gold voor de implementatie fase omdat ook dit door anderen gedaan zou worden.

Het was verstandig om verder te gaan met de planning terwijl er gewacht werd op de andere partijen in deze fases. Hierdoor raakte er geen tijd verloren. Integendeel, er werd zelfs tijd gewonnen. De ondersteuning die geboden moest worden tijdens de testen en implementatie fases bleek achteraf zodanig beperkt dat er erg veel tijd verspild zou zijn als er niets gedaan werd in de tussentijd.

Ook was het goed om de planning op te splitsen in twee hoofdfasen. Zoals in hoofdstuk 8.2 beschreven was dit voornamelijk om te voorkomen dat het bedrijf achterbleef met een niet-werkend of onvolledig component. Uiteindelijk bleek dat beide versies van de component op tijd voltooid waren.

Wat misschien wel beter had gekund, was dat de tweede hoofdfase opnieuw gepland werd nu er zoveel extra tijd vrijgekomen was. Hoewel de planning qua duur in grote lijnen overeenkwam met de werkelijkheid, ontstond er toch enige verwarring door de afwijkende data die aangegeven was op de planning. Dit werd echter wat verhelderd met behulp van de fase einde rapporten.

Een onderwerp dat erg goed ging tijdens het project was het werken met de taal VB.Net. Voor het begin van het project was er nog geen ervaring met het ontwikkelen in VB.Net. Tijdens het project is voor het eerst gewerkt met de programmeertaal. Hierdoor ontstond er wel een klein risico voor het project. Echter, de taal werd erg snel opgepikt. Tijdens het hele project waren er geen problemen met de programmeertaal.

Het gebruik van externe functionaliteit was ook een nieuw onderwerp. Om te voorkomen dat er veel moeilijkheden ontstonden tijdens het werken met externe de functionaliteit, is er veel analyse gedaan naar de functionaliteit. Hierdoor was er een goed idee van de sterke en minder sterke punten van de functionaliteit. Tijdens deze analyse werd onder andere ontdekt dat de functionaliteit geen ongedaan maken ondersteunde (zie hoofdstuk 6.2), waardoor hier snel op ingespeeld kon worden door zelf een ontwerp te maken voor het ongedaan maken.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Het gebruik van PRINCE2 tijdens dit project verliep goed. Toch was het misschien niet verstandig om dit project als een PRINCE2 project op te zetten. Het bedrijf zelf maakt namelijk gebruik van SCRUM, een zogeheten Agile ontwikkelmethode. Bij SCRUM wordt gewerkt in korte sprints waarin stukjes software worden opgeleverd. Deze methode had ook prima gebruikt kunnen worden voor dit project. Zo zou de eerste sprint voor het functioneel ontwerp zijn, een tweede sprint voor het technisch ontwerp enzovoort.

Het gebruik van de UML methodiek sloot goed aan op het project. Doordat het bedrijf zelf ook gebruik maakt van UML, sloot dit mooi op elkaar aan. Het enige nadeel dat UML met zich meebracht ten opzichte van dit project, was dat een UML klassendiagram geen ondersteuning bood voor de properties die VB.Net kent. Hierdoor kon minder duidelijk aangegeven worden welke variabelen wel/van buitenaf bereikbaar waren.

Tijdens het project zijn een aantal dingen teruggekomen die behandeld waren op de opleiding Informatica. Een voor de hand liggend voorbeeld hiervan is natuurlijk het gebruik van UML of PRINCE2. Een compleet overzicht is terug te vinden in hoofdstuk 10.

Er waren echter ook een aantal die niet op de opleiding behandeld zijn. Voorbeelden hiervan zijn het gebruik van externe functionaliteit of communicatie tussen verschillende talen zoals VB6 en VB.Net.

10. Beroepstaken

10.1. Uitvoeren analyse door definitie van requirements

Beroepstaak code:	1.4
Beschrijving:	Uitvoeren analyse door de definitie van requirements
Niveau:	3
Toegepast in:	Opstellen requirements document

Aan deze taak is voldaan doordat de wensen en eisen uit verschillende bronnen bij elkaar gebracht zijn en op elkaar afgestemd. De verschillende bronnen bestaan hier uit een oorspronkelijk functioneel ontwerp dat de bestaande functionaliteit van de VB6 versie van de component beschreef, gesprekken met de opdrachtgever en de verschillende verzoeken van klanten die zowel in het functioneel ontwerp als in het rapportagesysteem van het bedrijf vermeld staan. Het resultaat hiervan was een lijst met punten waaraan het systeem moest voldoen.

Deze punten zijn vervolgens onderverdeeld in de categorieën huidige functionaliteit en nieuwe functionaliteit, als gevolg van het besluit om het project in deze twee fases op te delen. Hierdoor ontstond er uit de requirements lijst twee sets use-cases.

10.2. Ontwerpen systeemdeel

Beroepstaak code:	3.2
Beschrijving:	Ontwerpen systeemdeel
Niveau:	3
Toegepast in:	Functioneel ontwerp / technisch ontwerp

Om aan deze taak te kunnen voldoen is veel tijd uitgetrokken om een goed ontwerp op te kunnen stellen. Om het ontwerp goed uit te kunnen beelden, is gebruik gemaakt van UML Use-cases, klassendiagrammen en objectdiagrammen. Ook is er in een later stadium gebruik gemaakt van het Singleton Designpattern. Dit pattern wordt gebruikt in de ColorManager om te voorkomen dat er meerdere instanties met verschillende kleuren ontstaan, die ertoe kunnen leiden dat er annotaties met onverwachte kleuren verschijnen. Het risico op meerdere instanties van deze klasse is ontstaan doordat meerdere klassen (AnnotationComponent, AnnotationManager en SymbolAnnotation) allemaal gebruik maken van deze klasse.

Bij de GUI was de manier waarop subitems getoond worden de grootste afweging die gemaakt moest worden (zie hoofdstuk 6.1). Er kon gekozen worden om de PictureBox methode over te nemen uit de oorspronkelijke versie, maar vanwege de problemen die dat met zich mee kon brengen (de functionaliteit werkte in de oorspronkelijke versie al niet volledig naar wens) is gekozen om dit te doen via het .Net component contextmenu.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

10.3. Bouwen applicatie

Beroepstaak code:	3.3
Beschrijving:	Bouwen applicatie
Niveau:	4
Toegepast in:	Realisatie

Aan deze taak is voldaan doordat de component aan moest sluiten op een groter geheel. Het feit dat dit geheel uit twee verschillende programmeertalen bestaat (VB.Net en VB6).

Vanzelfsprekend is de component objectgeoriënteerd opgebouwd, waardoor verschillende onderdelen opnieuw te gebruiken zijn.

Zodra de eerste versie van de component voltooid was (aan het einde van Hoofdfase 1), is deze in het versiebeheersysteem Sourcevault gehangen. Oorspronkelijk was het de bedoeling dat verdere wijzigingen (alles in Hoofdfase 2) dus via dit systeem doorgevoerd zou worden. Echter, door het gebrek aan testcapaciteit uit hoofdstuk 6.4 is dit niet mogelijk gebleken. Hoewel het uiteraard mogelijk is om eerdere versies op te halen met een versiebeheersysteem, is het moeilijk om reparaties aan te brengen in de huidige versie, terwijl er op hetzelfde moment aan een nieuwe versie gewerkt wordt.

Ook het feit dat er met externe functionaliteit gewerkt is (Vintasoft.Imaging en Vintasoft.Annotation (3)) gaf extra uitdaging aan deze taak. Doordat de component afgesteld moest worden op de wensen en eisen terwijl er tegelijkertijd rekening gehouden moest worden met de beperkingen van de externe functionaliteit. Dit uitte zich onder andere in het terugdraaien van acties, waar geen ondersteuning voor was. Hierdoor is de klasse UndoManager in het leven geroepen.

Tenslotte moest de component ook voldoen aan de kwaliteitseisen van het bedrijf. Dit betekende dus dat de component zich staande moest kunnen houden in een testomgeving. Dit legde nog meer nadruk op het creëren van een stabiel component.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

10.4. Initiëren en plannen van het testproces

Beroepstaak code:	3.4
Beschrijving:	Initiëren en plannen van het testproces
Niveau:	3
Toegepast in:	Functioneel ontwerp / Technisch ontwerp

Aan deze taak is voldaan door het opstellen van een testplan. Dit testplan is verdeeld over het functioneel ontwerp en het technisch ontwerp. In het functioneel ontwerp staat beschreven wat er wel en niet getest moet worden, een globale planning en een voorlopige checklist met punten die getest worden (zie hoofdstuk 6.1). Hierbij moest een goede afweging gemaakt worden over welke onderdelen wel of niet getest moesten worden.

In het technisch ontwerp zijn vervolgens uitspraken gedaan over de testomgeving, en is de definitieve checklist toegevoegd met alle punten die getest moeten worden (zie hoofdstuk 6.2).

Er is hierbij gebruik gemaakt van de testmethode van het bedrijf zelf. Dit komt doordat ook mensen buiten het project de component moesten testen, die standaard toch al gebruik maken van deze methode. Het zou onnodig werk hebben opgeleverd om een andere methode te kiezen.

10.5. Uitvoeren van en rapporteren over het testproces

Beroepstaak code:	3.5
Beschrijving:	Uitvoeren van en rapporteren over het testproces
Niveau:	3
Toegepast in:	Testen

Het uitvoeren van de tests is op een aantal manieren gebeurd. Als eerste is de checklist zowel intern als door de testers uitgevoerd. Deze resultaten zijn vervolgens met elkaar vergeleken, en afwijkingen zijn besproken.

De tweede manier van testen gebeurde door regelmatige kliktesten in verschillende stadia van de realisatie. Deze gebeurden op willekeurige momenten, en hadden als doel mogelijke feiten vroegtijdig op te sporen. Deze testen werden zowel intern gedaan als door het lid van de Project Assurance groep (zie (6)).

Punten die met behulp van deze kliktests werden gevonden zijn alleen mondeling besproken, er is geen documentatie meer van. Wel is de feedback van deze testen gelijk verwerkt. De resultaten van de checklisten werden geregistreerd in een resultatenbestand (19)

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Fouten die gevonden werden met behulp van de testen werden ook in dit bestand aangegeven, samen met reproductiestappen. Dit kon vervolgens gebruikt worden om de fouten te repareren.

De rest van het testproces viel niet onder het project. Dit bleef de verantwoordelijkheid van de testers van het bedrijf

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

11. Literatuurlijst

1. **Developer Express.** *DevExpress*. [Online] Developer Express. <http://www.devexpress.com>.
2. **Hove, Anne ten.** Functioneel ontwerp - Annotatiecomponent. Katwijk : sn, 2009.
3. **VintaSoft. Ltd.** VintaSoftImaging.NetSDK. *Vintasoft*. [Online] VintaSoft Ltd. <http://vintasoft.com/vsimaging-dotnet-index.html>.
4. **Kleppe, Jos Warmer & Anneke.** *Praktisch UML 4de editie*. sl : Pearson Education Benelux bv., 2007. 978-90-430-1265-2.
5. **Microsoft.** Microsoft Interop Forms Toolkit. [Online] <http://msdn.microsoft.com/en-us/vbasic/bb419144>.
6. **Zuidhoek, Mike.** Project Initiation Document - Project: Annotatie component. Katwijk : sn, 2011.
7. —. Functioneel Ontwerp - Huidige functionaliteit. Katwijk : sn, 2011.
8. —. Technisch ontwerp - Huidige functionaliteit V1. Katwijk : sn, 2011.
9. —. Technisch ontwerp- Huidige functionaliteit V2. Katwijk : sn, 2011.
10. **VintaSoft Ltd.** *Vintasoft Forums*. [Online] <http://vintasoft.com/forums/viewtopic.php?f=16&t=1317&sid=49239b21369dd19f1374cd8a2639e74e>.
11. **Zuidhoek, Mike.** Functioneel ontwerp - Nieuwe functionaliteit. Katwijk : sn, 2011.
12. —. Technisch ontwerp - Nieuwe functionaliteit. Katwijk : sn, 2011.
13. —. Request for Change document KD. Katwijk : sn, 2011.
14. —. Technisch ontwerp - Nieuwe functionaliteit iteratie. Katwijk : sn, 2011.
15. **Martin, Robert C.** *Clean Code - A Handbook of Agile Software Craftmanship*. sl : Prentice Hall, 2009. 780132350884.
16. **VintaSoft Ltd.** Vintasoft Community - Forums. *VintaSoft*. [Online] VintaSoft Ltd., 14 4 2011. [Citaat van: 27 4 2011.] <http://vintasoft.com/forums/viewtopic.php?f=16&t=1353&sid=6a25e3816e6ef710af5286bca6fde345>.
17. **Zuidhoek, Mike.** Annotatiecomponent - Handleiding V2. Katwijk : sn, 2011.
18. —. Beheer handleiding V2. Katwijk : sn, 2011.
19. —. *Checklist.xls*. [Excel Werkblad] Katwijk : MI Consultancy, 2011.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

12.Afkortingen

[illegible]

Bijlagen

(Tussen)producten

Project Initiation Document

Inleiding en achtergrond project

In dit document wordt de projectopzet getoond van het project Annotatie component. Momenteel is de organisatie bezig met het herschrijven van hun systeem naar Visual Basic .NET (VB.NET). Een groot gedeelte hiervan is al voltooid. Echter, een aantal componenten, waaronder de annotatie component, dienen nog herschreven te worden.

In de volgende hoofdstukken komt achtereenvolgens aan de orde:

- de projectdefinitie (o.a. doelstellingen, aanpak en resultaten);
- de organisatiestructuur;
- de initiële projectplanning;
- de beheersingsmechanismen (tolerantie, rapportage en uitzonderingsprocedure);
- de projectrisico's, inclusief tegenmaatregelen.

2 Projectdefinitie

2.1 Doelstellingen

Het doel van de opdracht is het herschrijven van de huidige annotatiecomponent. Ook dient er extra functionaliteit toegevoegd te worden aan de nieuwe component. De module dient te voldoen aan de richtlijnen van het bedrijf voor zowel de GUI, als de onderliggende code en dataopslag.

Uiteraard dienen ook gebruikers en technische documentatie voor de module gemaakt te worden.

2.2 Aanpak en fasering

Het project wordt aangepakt volgens de Prince2 methodiek en bestaat uit de volgende fasen:

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

- Analyse
- Functioneel Ontwerp Oude functionaliteit
- Technisch Ontwerp Oude functionaliteit
- Realisatie Oude functionaliteit
- Testen Oude functionaliteit
- Implementatie Oude functionaliteit
- Beheer en Documentatie Oude functionaliteit
- Functioneel Ontwerp Nieuwe functionaliteit
- Technisch Ontwerp Nieuwe functionaliteit
- Realisatie Nieuwe functionaliteit
- Testen Nieuwe functionaliteit
- Implementatie Nieuwe functionaliteit
- Beheer en Documentatie Nieuwe functionaliteit

2.3 Resultaten

Het project levert de volgende hoofdproducten op:

- Functioneel en Technisch ontwerp voor de oude functionaliteit en de toegevoegde functionaliteit.
- Twee versies van de component. De oude versie, welke in essentie een vertaling is van de oorspronkelijke component naar VB.NET, en een versie met toegevoegde functionaliteiten.
- Testplannen voor beide versies van de component, inclusief testresultaten.
- Beheer en gebruikersdocumentatie voor beide versies.

2.4 Omvang

Het project zal naar schatting 17 weken gaan duren.

2.5 Randvoorwaarden & beperkingen

- Er wordt gebruik gemaakt van een externe DLL. Het is dus niet nodig om alle functionaliteit van begin af aan op te bouwen.
- Er dient wel uitgezocht te worden of de DLL alle functionaliteit ondersteunt.
- Er mag pas begonnen worden met het toevoegen van nieuwe functionaliteit als de bestaande functionaliteit uit de eerste versie overgenomen is.
- De GUI moet voldoen aan de standaard van MI Consultancy.

3 Organisatiestructuur

Voor de duur van het project wordt een tijdelijke organisatie opgezet. Hieronder worden de verantwoordelijkheden van de diverse betrokken groepen algemeen toegelicht.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Project Board	
Namen	Rol
Alex van der Lans	Business Executive (zaakbehartiger)
Anne-Marie de Haas	Senior User (vertegenwoordiger gebruiker)

De project board is eindverantwoordelijk voor het overall eindresultaat van het project en voor het ter beschikking stellen van mensen en middelen. De project board komt formeel alleen bij elkaar voor het bespreken van faseplanningen en uitzonderings-rapportages. De Project Board rapporteert aan corporate/programme management.

Project Assurance	
Namen	Rol
Anne ten Hove	Consultant

Project Assurance is verantwoordelijk voor het monitoren van (kwaliteits)aspecten van het project en de opgeleverde producten, onafhankelijk van de Projectmanager. Project Assurance vergewist zich ervan dat het project conform de afspraken wordt beheerst en uitgevoerd. Bovendien houdt het de toegevoegde waarde van het project voor de bedrijfsdoelstellingen in de gaten, indien dit door de Project Board (Executive) is gedelegeerd. Project Assurance rapporteert aan de Project Board.

Project Manager & Project Support	
Naam	Rol
Alex van der Lans	Project Manager

De Projectmanager krijgt van de Project Board de verantwoordelijkheid en bevoegdheid voor de dagelijkse gang van zaken in het project. De Project Manager kan beslissingen nemen binnen de grenzen zoals aangegeven door de Project Board

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

(tolerantie). De belangrijkste verantwoordelijkheid van de Project Manager is het zekerstellen dat het project de gewenste producten oplevert, binnen tijd, binnen budget en volgens de afgesproken kwaliteit. De Project Manager rapporteert aan de Project Board.

Team	
Naam	Rol
Mike Zuidhoek	Ontwikkelaar

Het team is verantwoordelijk voor het opleveren van producten volgens de specificaties zoals aangegeven door de Projectmanager.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

4 Initiële Projectplanning

De projectplanning geeft aan welke acties wanneer door wie worden. Deze planning kan gedurende de loop van het project aangepast worden op basis van ervaringen in het project en / of nieuwe gegevens.

4.1 Projectplanning

De planning is als volgt:

Oude functionaliteit:

Fase	Begin datum	Eind datum	Duur in weken	Activiteiten	Op te leveren product(en)
Initiërende fase	7-2	9-2	0.5	Opstellen Project Initiatie Document (PID)	PID
Analyse	9-2	16-2	1	- Analyse oorspronkelijke module - Analyse aanvullende wensen / eisen	Requirements analyse
Functioneel Ontwerp	16-2	23-2	1	- Use-Cases - UI ontwerp - Testplan - Opstellen functioneel ontwerp	Functioneel ontwerp Testplan
Technisch Ontwerp	23-2	2-3	1	- Klassediagrammen - Objectdiagrammen - Opzetten testomgeving - Maken testscripts	Technisch ontwerp
Realisatie	2-3	23-3	2 ¹	Uitvoeren technisch ontwerp	Module
Testen	23-3	1-4	1.5	- Uitvoeren testscripts - Mogelijke scenario's - Gevonden fouten herstellen	Testverslag
Implementatie	4-4	8-4	1	Ondersteuning implementatie	<geen>
Beheer en Documentatie	11-4	15-4	1	- Gebruikershandleiding - Technische handleiding	Documentatie

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Uitloop	18-4	22-4	1	N.V.T.	<geen>
Totale duur	7-2	22-4	10 ¹	N.V.T.	<geen>

Nieuwe functionaliteit:

Fase	Begin datum	Eind datum	Duur in weken	Activiteiten	Op te leveren product(en)
Functioneel Ontwerp	25-4	27-4	0.5	• Use-Cases • UI ontwerp • Testplan • Opstellen functioneel ontwerp	Functioneel ontwerp Testplan
Technisch Ontwerp	27-4	29-4	0.5	• Klassediagrammen • Objectdiagrammen • DB ontwerp • Opzetten testomgeving • Maken testscripts	Technisch ontwerp
Realisatie	2-5	13-5	2	• Uitvoeren technisch ontwerp	Module
Testen	16-5	20-5	1	• Uitvoeren testscripts • Mogelijke scenario's	Testverslag
Implementatie	23-5	25-5	1	• Ondersteuning implementatie	<geen>
Beheer en Documentatie	25-5	27-5	1	• Gebruikershandleiding • Technische handleiding • Gevonden fouten herstellen	Documentatie
Uitloop	30-5	3-6	1	N.V.T.	<geen>
Totale duur	25-4	3-6	7	N.V.T.	<geen>

5 Beheersingsmechanismen

In dit hoofdstuk wordt aangegeven welke mechanismen worden toegepast om te waarborgen dat het project beheersbaar blijft.

¹ Afwezig in de week van 11-3 tot 20-3. Deze week wordt daarom niet meegeteld

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

5.1 Toleranties

Tijdens de uitvoering van het project controleert de Projectmanager regelmatig de voortgang. Indien de afwijking van de plannen naar verwachting groter is dan de afgesproken tolerantie, wordt daarover apart gerapporteerd aan de Project Board. De tolerantie op projectniveau is als volgt gedefinieerd:

-	doorlooptijd plus of min 10%
---	------------------------------

Per fase kunnen afwijkende toleranties worden opgenomen in de faseplanningen, afhankelijk van de omvang en impact van risico's.

5.2 Voortgangsrapportage

Er zijn twee soorten voortgangsrapportages van de Projectmanager aan de Project Board:

- Exception Report (rapportage indien de tolerantiegrenzen van het project of de fase in tijd dreigt te worden overschreden);
- End Stage Report (terugblik op afgelopen fase (evaluatie) en vooruitblik op eerstkomende fase (faseplanning)).

5.3 Uitzonderingsprocedure ('Management by Exception')

De uitzonderingsprocedure treedt in werking als van een fase of van het project verwacht wordt dat het niet binnen de afgesproken tolerantiegrenzen ten aanzien van tijd blijft. Zodra dit dreigt te gebeuren, wordt het Project board middels een Exception Report op de hoogte gebracht. In onderling overleg wordt de aanleiding en oorzaak besproken. Daarna wordt besloten tot één van de volgende situaties:

- de Project Board treft maatregelen ter voorkoming/opheffing van de aanleiding;
- de Project Board besluit geen actie te ondernemen, omdat het denkt dat de overschrijding van de tolerantie niet plaats zal vinden;
- de toleranties voor de fase worden verruimd;
- er worden concessies gedaan ten aanzien van tijd, kwaliteit of omvang van het op te leveren resultaat (bereik).

De eerste drie situaties worden vermeld in het eerstvolgende End Stage Report. De laatste situatie kan op verzoek van de Project Board leiden tot het opstellen van een Exception Plan met alternatieven voor de aanpak van het vervolg van het project en aangepaste planningen.

6 Projectrisico's

De volgende tabel geeft een overzicht van de tot nu toe onderkende bedreigingen ten aanzien van het project met voorgestelde tegenmaatregelen, de kans van optreden en de mate van negatief effect op het project (aangegeven op een schaal van 1 tot 5)². De laatste kolom geeft het risico aan (kans*effect). Op deze wijze kan gefundeerd worden afgewogen welke bedreigingen de meeste aandacht of hulpbronnen verdienen.

Tijdens het project wordt deze lijst voortdurend bijgehouden in een Logboek risico's, met een uitgebreidere beschrijving van de risico's.

Bedreiging	Tegenmaatregel	Kans	Effect	Risico
Vertraging door ziekte of privé omstandigheden	Een goede planning maken die ruimte biedt voor dit soort omstandigheden (uitloophase)	2	3	6
Onbekend met de programmeertaal	Verdiepen in de programmeertaal	1	3	3
Niet eenduidige requirements	Goede requirements analyse en overleg hierover. Bij aanhoudende onenigheid de beslissing laten maken door de Project Manager	3	3	6
Externe software heeft onverwachte beperkingen	Goede analyse externe software	2	4	8
Door voortschrijdend inzicht wijzigen het Functioneel of Technisch ontwerp tijdens een	Geen	3	4	12

² 1 = kleine kans of gering effect, 5 = grote kans of groot effect

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Bedreiging	Tegenmaatregel	Kans	Effect	Risico
vervolgfase, waardoor deze fase uitloopt.				

Functioneel Ontwerp – Huidige functionaliteit

Doel van de component

Het doel van de component is om een gebruiker in staat te stellen visueel aan te geven waar een patiënt klachten heeft. De gebruiker kan op schematische weergaven van lichaamsdelen met behulp van lijntjes, vormen en symbolen zeer nauwkeurig aangeven waar en wat voor soort klachten een patiënt heeft. Dit zorgt ervoor dat gebruikers in één oogopslag kunnen zien wat de bevindingen van de eerste gebruiker zijn.

Requirements

Algemeen

- De nieuwe component moet hetzelfde formaat hebben als de oude versie.
- Indien de afbeelding groter is dan de component, dienen schuifbalken getoond te worden waarmee de gebruiker door de afbeelding kan bewegen.
- De look van de nieuwe component dient zoveel mogelijk overeen te komen met de look van de oude component om het voor de eindgebruikers zo eenvoudig mogelijk te houden.
- De oude component controleert nu of de muis op een bepaald icoontje staat en laat aan de hand daarvan verschillende opties zien (als een soort uitschuif menu). Dit werkt echter niet helemaal zoals gewenst. Gekeken moet worden of dit in de nieuwe component anders kan.
- Afbeeldingen moeten verslept kunnen worden met het “handje”.
- Indien er een nieuwe instance of episode gemaakt wordt, dient er een nieuwe, lege (default) afbeelding in het component geplaatst te worden. Deze default afbeelding wordt gedefinieerd tijdens het maken van het formulier

Annotaties

- De component moet de volgende typen annotaties kunnen plaatsen:
 - Pen
 - Tekst
 - Symbolen
- Pen annotaties bestaan uit:
 - Lijn
 - Vierkant
 - Cirkel
 - Vrij tekenen
- Symbool annotaties bestaan uit:
 - Pijlen (in de vier windrichtingen)
 - Kruis
 - Plus en min
 - S
 - Gedraaide pijlen
- Annotaties moeten gedraaid, verplaatst en verwijderd kunnen worden, en het moet mogelijk zijn om de annotatie te vergroten of te verkleinen.
- Annotaties moeten herhaaldelijk geplaatst kunnen worden.
- Indien een annotatie gekozen wordt dient de cursor te veranderen in de te plaatsen annotatie.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

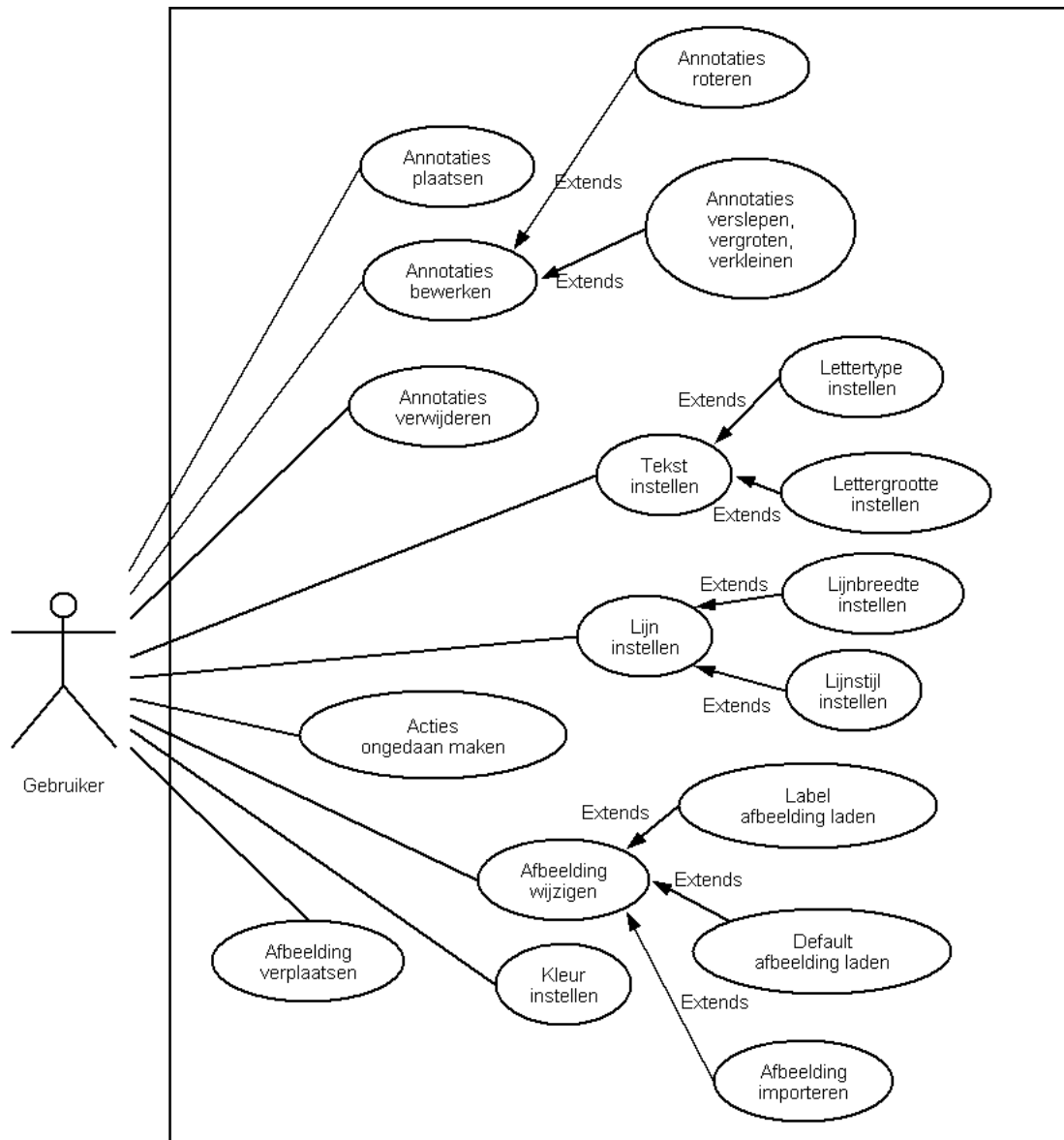
- Annotaties moeten ongedaan gemaakt kunnen worden met de knop ongedaan maken.
- (Delen van) annotaties moeten verwijderd kunnen worden met het gummetje.
- De component mag alleen gebruikt worden indien de gebruiker het recht heeft om het EMD te wijzigen.
- Indien de component wordt aangeroepen vanuit gastgebruik, mag deze alleen gebruik worden als aangegeven staat dat gastgebruik toestemming heeft om het EMD te wijzigen.

Instellingen

- Alle instellingsschermen dienen in aparte vensters getoond te worden.
- Er dient een instellingenscherm te zijn voor de volgende onderdelen:
 - Tekst
 - Afbeeldingen
 - Kleuren
 - Lijn
- Tekstinstellingen:
 - Het lettertype en de lettergrootte dient aangepast te kunnen worden (standaard Arial 16).
- Afbeeldingsinstellingen:
 - Afbeeldingen kiezen die gedefinieerd zijn met standaard labels. Deze labels moeten ingesteld staan op actief.
 - De labels hebben soms een prefix (XX—Label). Deze prefix mag niet getoond worden.
 - Afbeeldingen importeren vanaf de lokale PC. Alleen het JPG formaat is toegestaan.
 - Melding indien wijzigingen zijn opgeslagen voor het wisselen van de default afbeelding
- Lijninstellingen:
 - Lijndikte moet ingesteld kunnen worden (standaard 2 pixels).
 - Lijntype moet ingesteld kunnen worden (vast, streep, punt, streep-punt, streep-punt-punt).
- Kleurinstellingen:
 - Er moet gekozen kunnen worden uit 2 standaardkleuren (blauw, rood waarvan blauw automatisch gekozen wordt indien de gebruiker geen kleur kiest) en 3 specifieke kleuren.
 - De standaardkleuren mogen niet gewijzigd worden.

Use Cases

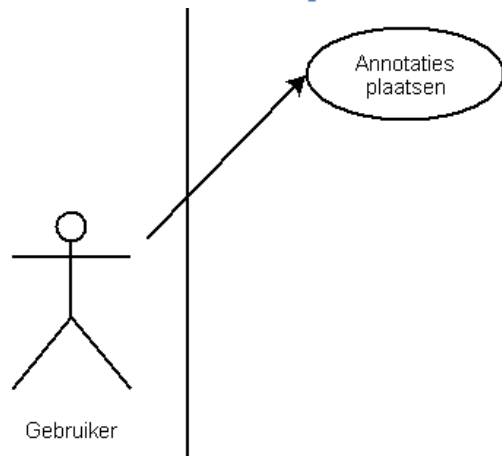
Use Case Diagram



Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Use Case Beschrijvingen

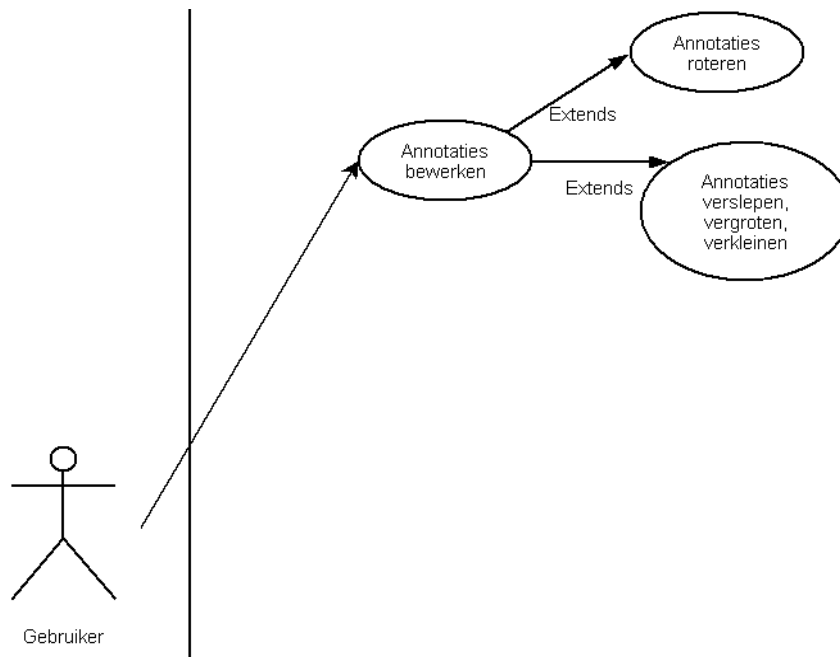
Use Case: Annotaties plaatsen



Naam	Annotaties plaatsen
Beschrijving	Het plaatsen van een annotatie op een bestaande afbeelding.
Aannamen	Er is al een afbeelding in de component ingeladen.
Resultaat	Er is een annotatie toegevoegd aan de afbeelding.
Stappenplan	1. De gebruiker kiest de te plaatsen annotatie. 2. De cursor verandert in de te plaatsen annotatie. 3. De gebruiker klikt (in de afbeelding) op de plek waar de annotatie moet komen. 4. De annotatie wordt op de afbeelding geplaatst.
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Use Case: Annotaties bewerken (Extends roteren en verslepen, vergroten, verkleinen)



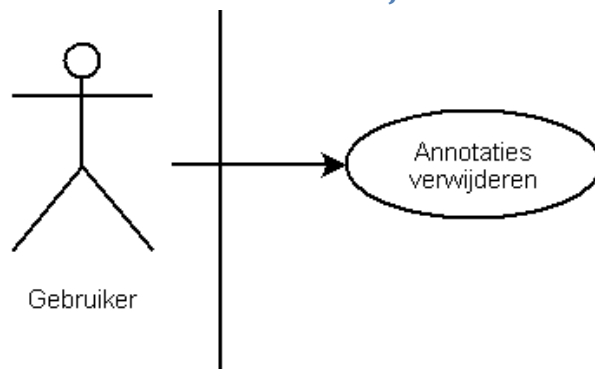
Naam	Annotaties bewerken (roteren)
Beschrijving	Het uitvoeren van een bewerking op een bestaande annotatie.
Aannamen	Er is tenminste één annotatie op de afbeelding geplaatst. De afbeelding is niet opgeslagen.
Resultaat	De annotatie is bewerkt
Stappenplan	1. Klik op de te bewerken annotatie. 2. Klik en sleep op het groene bolletje dat boven de annotatie verschijnt.
Uitzonderingen	Nadat de afbeelding is opgeslagen kunnen de annotaties niet meer bewerkt worden. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Naam	Annotaties bewerken (Verslepen,
------	---------------------------------

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

	vergroten en verkleinen)
Beschrijving	Het uitvoeren van een bewerking op een bestaande annotatie.
Aannamen	Er is tenminste één annotatie op de afbeelding geplaatst. De afbeelding is niet opgeslagen.
Resultaat	De annotatie is bewerkt
Stappenplan	1. Klik op de te bewerken annotatie. 2. Klik, afhankelijk van de gewenste actie, op het juiste witte bolletje en sleep.
Uitzonderingen	Nadat de afbeelding is opgeslagen kunnen de annotaties niet meer bewerkt worden. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Use Case: Annotaties verwijderen

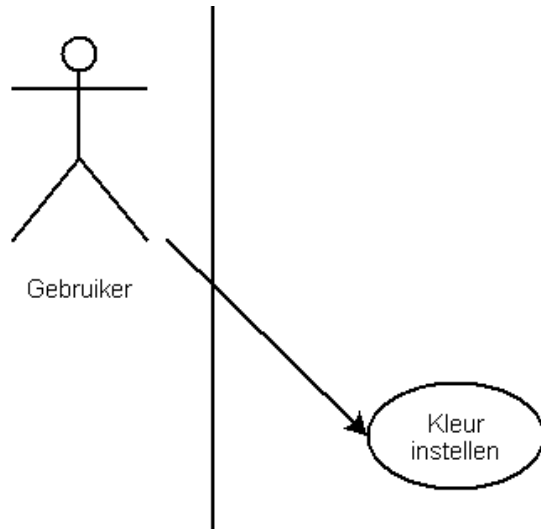


Naam	Annotaties verwijderen
Beschrijving	Het verwijderen van een annotatie.
Aannamen	Er is tenminste één annotatie geplaatst. De afbeelding is niet opgeslagen.
Resultaat	De geselecteerde annotatie is verwijderd.
Stappenplan	1. Klik op de te verwijderen annotatie. 2. Druk op de DELETE toets.
Uitzonderingen	Nadat de afbeelding is opgeslagen

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

	kunnen de annotaties niet meer verwijderd worden. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.
--	---

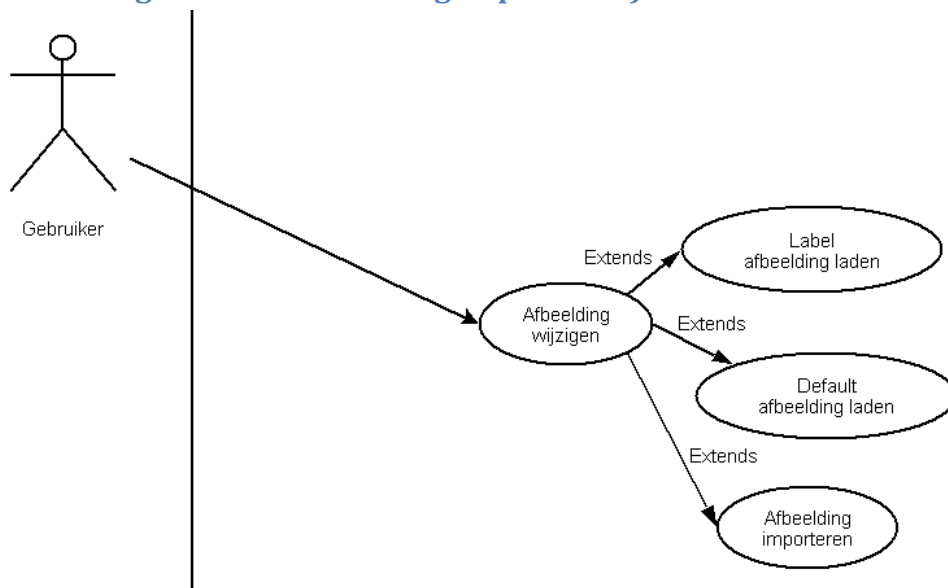
Use Case: Kleur instellen



Naam	Kleur instellen
Beschrijving	Het kiezen van de te gebruiken kleur voor nieuwe annotaties.
Aannamen	Geen
Resultaat	Alle annotaties die geplaatst worden hebben de gekozen kleur.
Stappenplan	1. Klik op het icoon kleuren. 2. Kies één van de twee standaard kleuren, of één van de drie specifieke kleuren. 3. Nieuwe annotaties zullen de geselecteerde kleur krijgen.
Uitzonderingen	Annotaties die geplaatst zijn voordat de kleur veranderd is, veranderen niet mee. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Use Case: Afbeelding wijzigen (Extends Label afbeelding laden, Default afbeelding laden en Afbeelding Importeren)



Naam	Afbeelding wijzigen → Label afbeelding laden
Beschrijving	Het wijzigen van de achtergrondafbeelding.
Aannamen	Geen
Resultaat	De afbeelding is veranderd en alle annotaties zijn verwijderd.
Stappenplan	1. Druk op het icoon Afbeelding instellen. 2. Kies één van de vooraf ingestelde labels. 3. Indien er eerder opgeslagen wijzigingen bestaan, wordt er een melding getoond en om bevestiging gevraagd. 4. De oude afbeelding en alle annotaties die erop geplaatst zijn worden verwijderd. 5. De nieuwe afbeelding wordt ingeladen.
Uitzonderingen	Indien de afbeelding opgeslagen is, wordt een melding weergegeven dat eerder opgeslagen data verloren zal gaan. Stap 4 en 5 worden niet uitgevoerd als de

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

	gebruiker geen toestemming geeft bij 3. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.
--	---

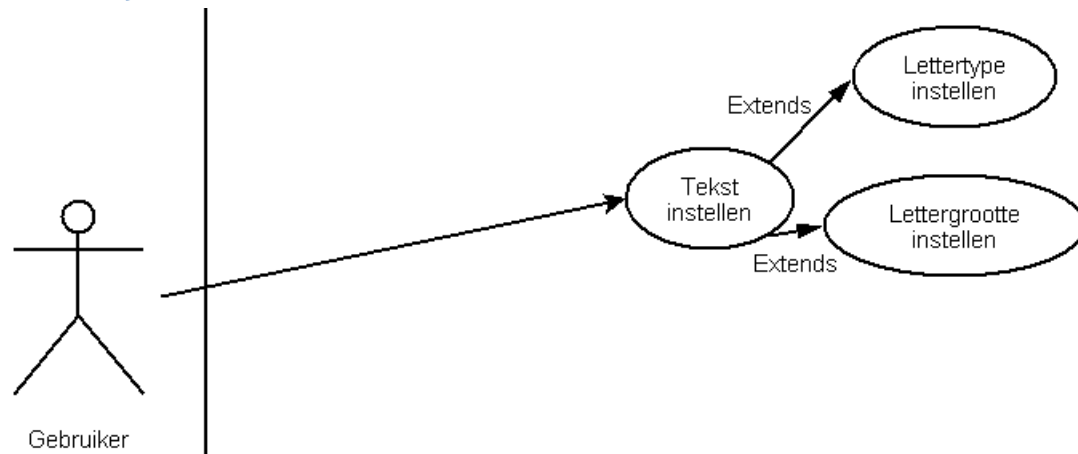
Naam	Afbeelding wijzigen → Default afbeelding laden
Beschrijving	Het wijzigen van de achtergrondafbeelding.
Aannamen	Geen
Resultaat	De afbeelding is veranderd en alle annotaties zijn verwijderd.
Stappenplan	1. Druk op het icoon Afbeelding instellen. 2. Kies de optie Standaardafbeelding herstellen. 3. Indien er eerder opgeslagen wijzigingen bestaan, wordt er een melding getoond en om bevestiging gevraagd. 4. De oude afbeelding en alle annotaties die erop geplaatst zijn worden verwijderd. 5. De nieuwe afbeelding wordt ingeladen.
Uitzonderingen	Indien de afbeelding opgeslagen is, wordt een melding weergegeven dat eerder opgeslagen data verloren zal gaan. Stap 4 en 5 worden niet uitgevoerd als de gebruiker geen toestemming geeft bij 3. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Naam	Afbeelding wijzigen → Afbeelding importeren
Beschrijving	Het wijzigen van de achtergrondafbeelding.
Aannamen	Geen
Resultaat	De afbeelding is veranderd en alle annotaties zijn verwijderd.
Stappenplan	1. Druk op het icoon Afbeelding instellen. 2. Kies de optie Bladeren 3. Zoek de juiste afbeelding op de lokale harde schijf 4. Indien er eerder opgeslagen wijzigingen bestaan, wordt er een melding getoond en om bevestiging gevraagd. 5. De oude afbeelding en alle annotaties die erop geplaatst zijn worden verwijderd. 6. De nieuwe afbeelding wordt ingeladen.
Uitzonderingen	Indien de afbeelding opgeslagen is, wordt een melding weergegeven dat eerder opgeslagen data verloren zal gaan. Stap 5 en 6 worden niet uitgevoerd als de gebruiker geen toestemming geeft bij 4. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Use Case: Tekst instellen (Extends lettertype instellen en lettergrootte instellen)



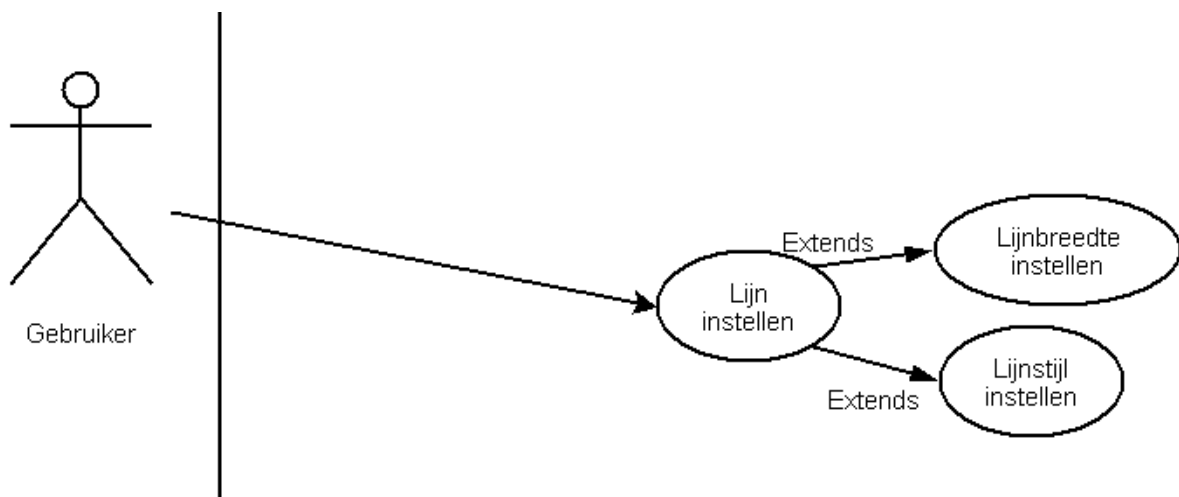
Naam	Tekst instellen → Lettertype instellen
Beschrijving	Het lettertype van tekstannotaties instellen.
Aannamen	Geen
Resultaat	Het lettertype is gewijzigd voor nieuwe tekstannotaties.
Stappenplan	1. Klik op instellingen → Tekstinstellingen. 2. Kies het juiste lettertype.
Uitzonderingen	Eerder geplaatste tekstannotaties zullen niet veranderen. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.

Naam	Tekst instellen → Lettergrootte instellen
Beschrijving	De lettergrootte van tekstannotaties instellen.
Aannamen	Geen
Resultaat	De lettergrootte is gewijzigd voor nieuwe

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

	tekstannotaties.
Stappenplan	1. Klik op instellingen → Tekstinstellingen. 2. Kies de juiste grootte.
Uitzonderingen	Eerder geplaatste tekstannotaties zullen niet veranderen. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.

Use Case: Lijn instellen (Extends Lijnbreedte instellen en Lijnstijl instellen)



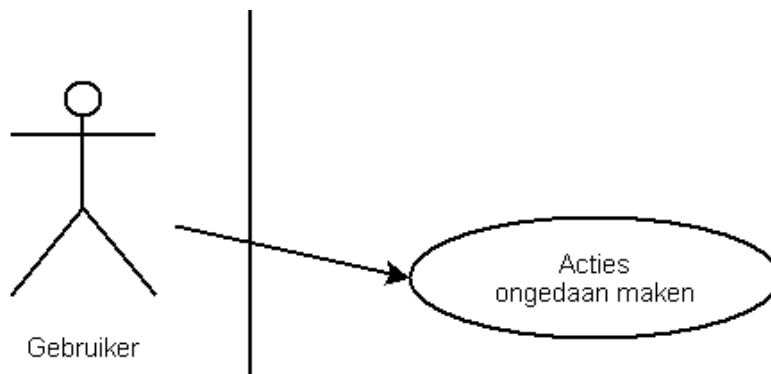
Naam	Lijn instellen → Lijnbreedte instellen
Beschrijving	Het instellen van de dikte van nieuwe lijnannotaties.
Aannamen	Geen
Resultaat	Nieuwe lijnannotaties worden met de nieuwe breedte getekend.
Stappenplan	1. Klik op instellingen → Lijninstellingen. 2. Kies de te gebruiken breedte.
Uitzonderingen	Bestaande lijnannotaties worden niet aangepast. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

	geopend worden.
--	-----------------

Naam	Lijn instellen → Lijnstijl instellen
Beschrijving	Het instellen van de stijl van nieuwe lijnannotaties.
Aannamen	Geen
Resultaat	Nieuwe lijnannotaties worden met de nieuwe stijl getekend.
Stappenplan	3. Klik op instellingen → Lijninstellingen. 4. Kies de te gebruiken stijl.
Uitzonderingen	Bestaande lijnannotaties worden niet aangepast. Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Dit scherm kan maar één keer tegelijk geopend worden.

Use Case: Acties ongedaan maken



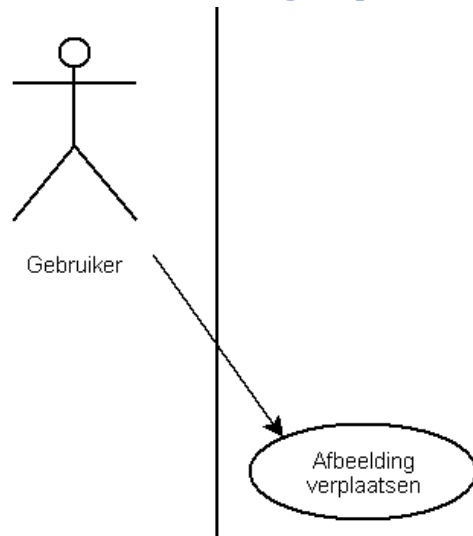
Naam	Acties ongedaan maken
Beschrijving	Het ongedaan maken van de laatst uitgevoerde actie(s)
Aannamen	Er is tenminste 1 actie ondernomen
Resultaat	De laatst uitgevoerde actie(s) zijn ongedaan gemaakt

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Stappenplan	1. Klik op de knop ongedaan maken. 2. De laatst uitgevoerde actie wordt ongedaan gemaakt.
Uitzonderingen	Het verplaatsen van de afbeelding wordt niet als een actie gerekend. Na opslaan kan er niet meer ongedaan gemaakt worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

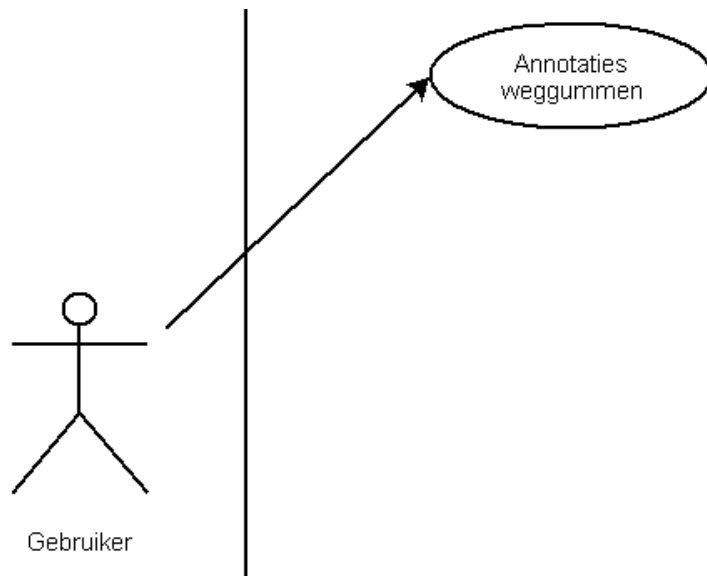
Use Case: Afbeelding verplaatsen



Naam	Afbeelding verplaatsen
Beschrijving	Het verslepen van de afbeelding om delen te zien die buiten de component vallen.
Aannamen	De afbeelding is te groot voor de component
Resultaat	De afbeelding is verplaatst
Stappenplan	1. Klik op het handje. 2. De cursor is veranderd. 3. Klik en sleep ergens in de afbeelding.
Uitzonderingen	Indien de afbeelding klein genoeg is om volledig in de component te passen, kan er niet verslept worden

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Use Case: Annotaties weggummen

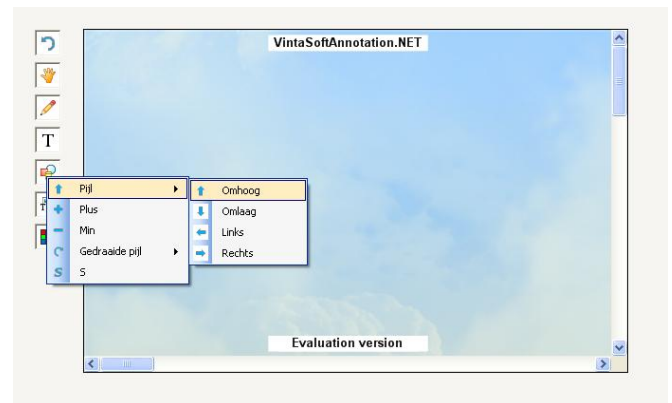
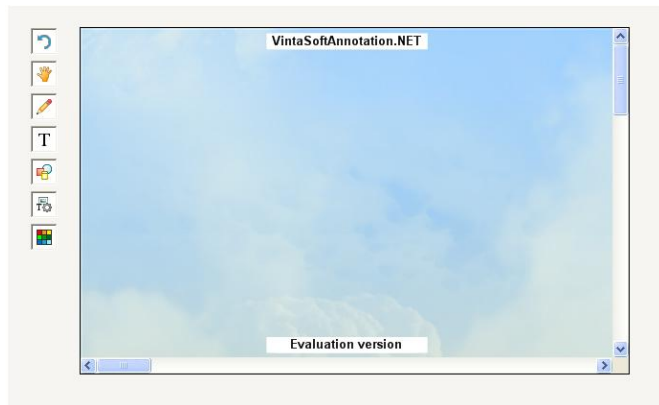


Naam	Annotaties weggummen
Beschrijving	Verwijdert (delen van) annotaties
Aannamen	Afbeelding is niet opgeslagen. Er is tenminste één annotatie geplaatst.
Resultaat	Een gedeelte van een annotatie is verwijderd
Stappenplan	1. Klik op het gummetje. 2. De cursor verandert in een gummetje. 3. Klik en sleep over het uit te gummen gedeelte.
Uitzonderingen	Opgeslagen annotaties kunnen niet worden uitgegumd.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

UI Ontwerp

Schermafbeeldingen



Verschillen met eerdere versie

Het grootste verschil tussen de twee ontwerpen is de manier waarop de



verschillende annotaties gepresenteerd worden aan de gebruiker. In de huidige versie (zie afbeelding links) wordt een rij aan pictureboxes aangemaakt zodra de gebruiker zijn muis op het icoontje Drag & Drop plaatst. In het nieuwe ontwerp verschijnt er een drop down menu zodra de gebruiker op het icoontje klikt. Dit biedt als voordeel dat het menu overzichtelijk

blijft als er extra annotaties toegevoegd worden (zeker als er gebruik gemaakt wordt van categorieën zoals te zien is in de afbeelding rechtsboven).

Tijdens het opslaan worden de afbeelding en de annotaties samengesmolten, en kan daarna niet meer bewerkt worden. In een later stadium zal gekeken worden of dit anders kan.

Testplan

Wat wordt er getest?

Alle genoemde Use-Cases worden getest.

Wat wordt er niet getest?

- Aanpassen formaat component.
- Opslaan.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

- Afbeelding aanpassen na opslaan (alleen annotaties die na opslaan geplaatst worden kunnen bewerkt worden).

Test taken

Elk van de genoemde testitems moet volgens de use case in het functioneel ontwerp uitgevoerd kunnen worden. Ook moet het resultaat gelijk zijn aan het verwachte resultaat van de use case. Tenslotte moet er een testomgeving ingericht worden in de vorm van een nieuw formulier met de nieuwe component erin.

Planning

Tijd/datum	Omschrijving	Wie?	(ok/x)
23-3	Aandragen onderdelen, begin eerste testronde		
25-3	Feedback eerste testronde		
28-3	Bugfixing a.d.h. van feedback eerste ronde		
30-3	Begin tweede testronde		
1-4	Voltooien testen		

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Risico

- Tijdens het testen zijn er de volgende risico's die het testen kunnen belemmeren:
- **Tijdstekort:** Het kan zijn dat de geplande tijd tekort blijkt te zijn om alle genoemde onderdelen voldoende te kunnen testen. In de algemene planning is een uitloopweek opgenomen die (gedeeltelijk) gebruikt kan worden om dit op te vangen. Wel moet er gekeken worden of er onderwerpen geschrapt kunnen worden uit de tests indien deze uitloopweek nodig is.
 - **Stoppers:** Er is een kans dat tijdens het testen een fout gevonden wordt die het onmogelijk maakt om verder te kunnen met testen. Dit risico wordt teruggebracht door de ontwikkelaar alvast een aantal onderdelen te laten testen voor de testfase ingaat. Tijdens deze snelle testen wordt vooral gekeken of er geen crashes optreden bij grote functies, er wordt minder nauwkeurig gekeken dan wanneer het echte testen begint.

Checklisten

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij opstarten component een afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Annotaties plaatsen					
Lukt het maken van een lijn met Vrij tekenen?					
Lukt het maken van een lijn met Lijn tekenen?					
Lukt het maken van een vierkant met Vierkant tekenen?					
Lukt het maken van een cirkel met Cirkel tekenen?					
Lukt het om een tekst te plaatsen met Tekst annotatie?					
Is het mogelijk om een symbool annotatie te plaatsen?					
Als er een andere kleur gekozen wordt in de kleurenkiezer, is dit dan terug te zien in nieuwe annotaties?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Als er een andere lijndikte of lijnstijl gekozen wordt, is dit dan terug te zien in nieuwe pen annotaties? (lijn, vrij, cirkel, vierkant)					
Als het lettertype of lettergrootte wordt aangepast, is dit dan terug te zien in nieuwe tekstannotaties?					
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?					
Is de geselecteerde annotatie te vergroten / verkleinen?					
Is de geselecteerde annotatie te roteren of te verplaatsen?					
Is de geselecteerde annotatie te verwijderen?					
Overig					
Is het mogelijk om (delen van) annotaties te verwijderen met de gumfunctie?					
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?					
Indien er geen acties ongedaan te maken zijn, wordt er dan een melding weergegeven als iemand toch op ongedaan maken klikt?					

In de kolom Bevinding wordt aangegeven wat er daadwerkelijk te zien is na het uitvoeren van de check. Denk hierbij aan bijvoorbeeld foutmeldingen. In de kolom Reproductie wordt aangegeven hoe een eventuele afwijking te herproduceren is. Belangrijk is om hier de volgende punten in aan te geven:

- Omgeving (Bijvoorbeeld: welke gebruiker is er ingelogd?)
- Patiëntnummer van de testpatiënt.
- Welk formulier?
- Welke instance / episode van het formulier?
- Stappen die geleid hebben tot het afwijkend gedrag.

In de kolom Gewenst gedrag komt te staan wat het resultaat zou moeten zijn. Indien hier niet aan voldaan is, is er dus sprake van een fout. De kolom Prioriteit geeft

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

tenslotte aan hoe groot de impact is van de gevonden fout en hoe snel deze opgelost moeten worden. Prioriteit kent drie categorieën:

- Laag: Dit zijn kleine foutjes in de UI (zoals een pictogram dat op de verkeerde plek staat) of foutjes die makkelijk te omzeilen zijn.
- Normaal: Dit zijn de soort fouten die ervoor zorgen dat bepaalde onderdelen niet naar behoren werken (zoals wanneer een plus wordt geplaatst terwijl de min annotatie is geselecteerd of bepaalde instellingen die niet gewijzigd kunnen worden).
- Hoog: Dit zijn de fouten die ervoor zorgen dat de component crasht (stoppers), of waardoor werken met de component nagenoeg onmogelijk wordt (bijvoorbeeld: annotaties plaatsen werkt niet)

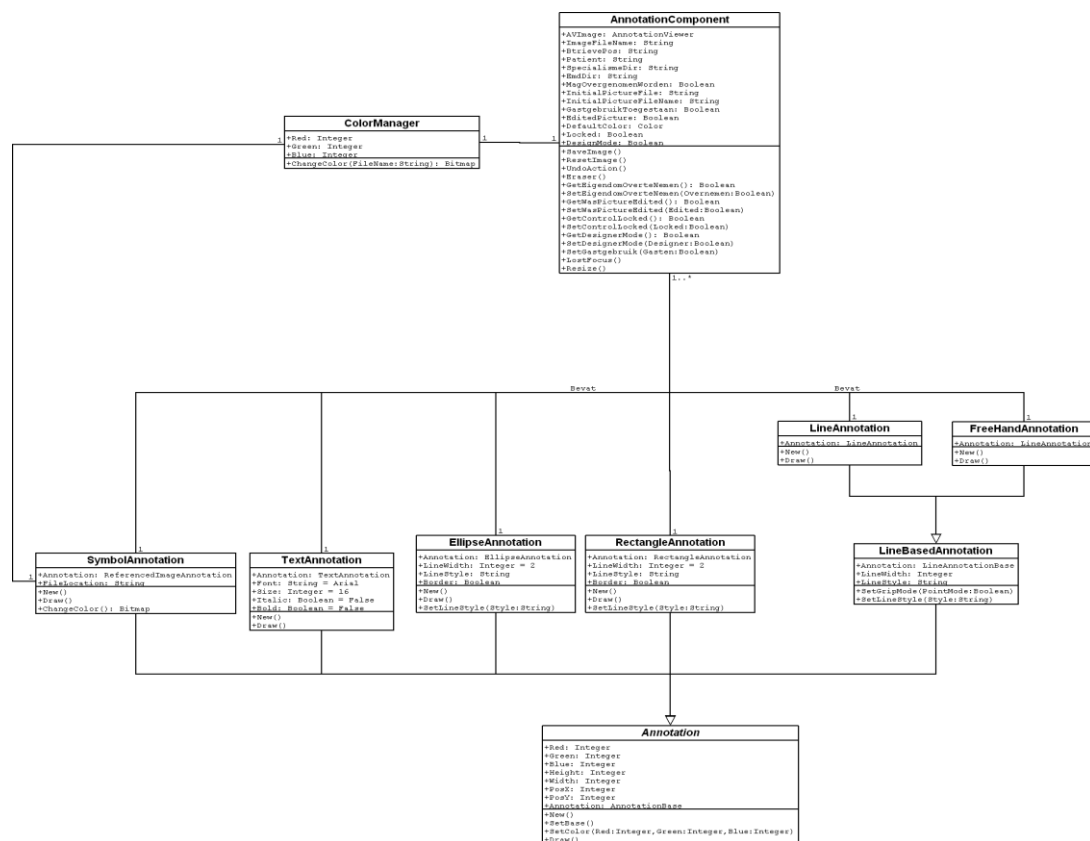
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Technisch ontwerp – Huidige functionaliteit V1

Inleiding

In dit document wordt uitgelegd hoe het eerder opgestelde functioneel ontwerp uitgewerkt zal worden. Om dit toe te lichten wordt als eerste het klassediagram getoond. Vervolgens wordt in het object diagram aangegeven welke overige objecten er aangemaakt worden om de component te kunnen creëren (hier vallen ook standaard objecten van VB.NET onder). Tenslotte wordt een uitbreiding op de checklist voor het testen getoond, samen met de eisen waar de testomgeving aan moet voldoen. Opmerking: omdat de afbeeldingen wellicht wat moeilijk te lezen zijn, zijn deze ook terug te vinden onder Y:\Medewerkers\Mike\Technisch ontwerp

Klassediagram



In dit klassediagram is de schematische opzet van de component aangegeven. In het diagram is te zien dat alle klassen een verband hebben met de klasse AnnotationComponent. Andere objecten zullen vanuit deze klasse aangeroepen worden. Ook de GUI is gedefinieerd in deze klasse. Op het moment dat de gebruiker een annotatie wil plaatsen of bewerken, wordt één van de annotatie klassen aangeroepen.

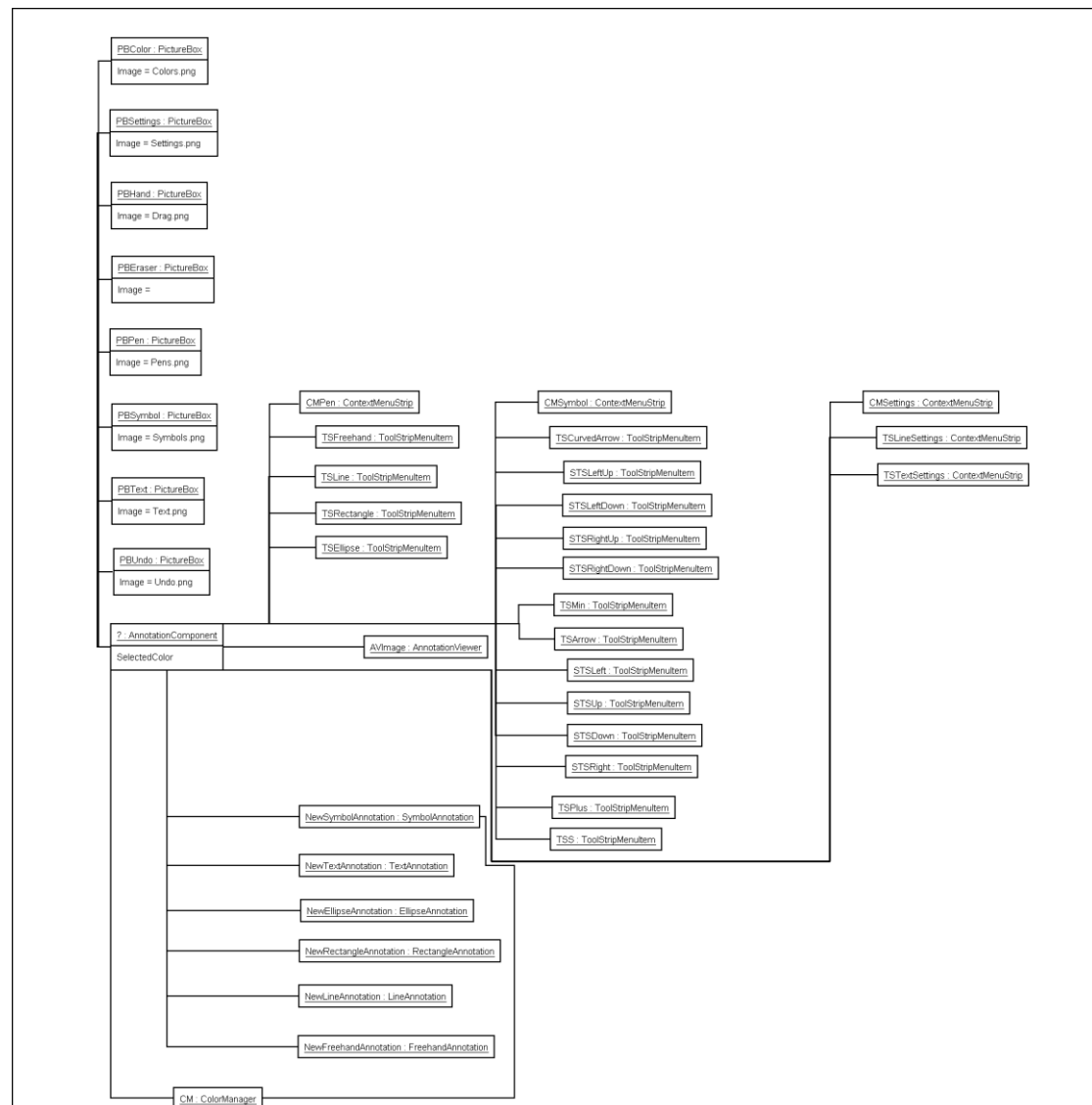
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Deze annotatieklassen hebben allemaal dezelfde basis die gedefinieerd is in de abstracte klasse Annotation. In deze basis staat de kleur die de annotatie moet krijgen, plus methodes die alle annotaties met elkaar delen (denk hierbij aan tekenmethodes bijvoorbeeld). Deze opzet zorgt ervoor dat de component makkelijk uitgebreid / aangepast kan worden.

De klasse ColorManager tenslotte, is verantwoordelijk voor het veranderen van de kleuren van afbeeldingen. Op het moment dat de gebruiker een andere kleur kiest, zal elke nieuwe annotatie deze kleur aannemen. Symboolannotaties werkt echter met vooraf gedefinieerde afbeeldingen en kunnen dus niet zo makkelijk van kleur veranderen als bijvoorbeeld een lijn of tekstannotatie. Deze klasse zal de bronafbeelding van de annotatie nemen, en pixel voor pixel controleren of de kleur aangepast moet worden en zo ja, in welke kleur. Dit gebeurt ook voor de afbeeldingen die te zien zijn in de contextmenu's.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Object Diagram



In dit diagram is een overzicht te vinden van alle objecten en de onderlinge relatie ertussen. Ook hier valt weer op dat AnnotationComponent één van de belangrijkste onderdelen is van het project.

Een hoop van de objecten in dit diagram zijn standaard controls. Verder zijn er een aantal objecten die helpen met het aanmaken van de verschillende soorten annotaties, plus een ColorManager object dat helpt met het aanpassen van de kleur in symboolannotaties en de symbolen in de contextmenu's.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Testomgeving

Om de component na realisatie goed te kunnen testen moet er een testomgeving opgezet worden voor de component. Om de component te kunnen testen volstaat het om een nieuw formulier aan te maken waar de component in geplaatst kan worden. Dit zorgt ervoor dat er geen risico is dat andere componenten de testresultaten vervuilen.

Behalve een eigen formulier zijn er geen eisen voor de testomgeving.

Checklists

Naar aanleiding van de ontwikkelingen tijdens de ontwerpfase, zijn de checklists gemaakt in het functioneel ontwerp uitgebreid met extra checkpunten.

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Lukt het maken van een lijn met Vrij tekenen?					
Lukt het maken van een lijn met Lijn tekenen?					
Lukt het maken van een vierkant met Vierkant tekenen?					
Lukt het maken van een cirkel met Cirkel tekenen?					
Lukt het om een tekst te plaatsen met Tekst annotatie?					
Is het mogelijk om een symbool annotatie te plaatsen?					
Is het mogelijk om een annotatie meerdere malen te plaatsen?					
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?					
Is de geselecteerde annotatie te vergroten / verkleinen?					
Is de geselecteerde annotatie te roteren?					
Is de geselecteerde annotatie te verplaatsen?					
Is de geselecteerde annotatie te verwijderen?					
Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?					
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?					
Kan er een specifieke kleur gekozen worden?					
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?					
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?					
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?					
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?					
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?					
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?					
Wordt een eventuele prefix van een label niet weergegeven?					
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?					
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?					
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?					
Overig					
Is het mogelijk om (delen van) annotaties te verwijderen met de gumfunctie?					
Is het mogelijk om de afbeelding te verplaatsen met het handje?					
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?					
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan					

Naam: M.C.W. Zuidhoek

Studentnummer: 07009593

maken dan grijs?

--	--	--	--	--	--

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Technisch ontwerp – Huidige functionaliteit V2

Inleiding

In dit document wordt uitgelegd hoe het eerder opgestelde functioneel ontwerp uitgewerkt zal worden. Om dit toe te lichten wordt als eerste het klassediagram getoond. Vervolgens wordt in het object diagram aangegeven welke overige objecten er aangemaakt worden om de component te kunnen creëren (hier vallen ook standaard objecten van VB.NET onder). Tenslotte wordt een uitbreiding op de checklist voor het testen getoond, samen met de eisen waar de testomgeving aan moet voldoen. Opmerking: omdat de afbeeldingen wellicht wat moeilijk te lezen zijn, zijn deze ook terug te vinden onder Y:\Medewerkers\Mike\Technisch ontwerp

Klassediagram



In dit klassediagram is de schematische opzet van de component aangegeven. In het diagram is te zien dat alle klassen een verband hebben met de klasse AnnotationComponent. Andere objecten zullen vanuit deze klasse aangeroepen worden. Ook de GUI is gedefinieerd in deze klasse. Op het moment dat de gebruiker een annotatie wil plaatsen of bewerken, wordt één van de annotatie klassen aangeroepen.

Deze annotatieklassen zijn allemaal overervingen van klassen uit de VintaSoft onderdelen. In deze klasse worden alle instellingen voor de annotatie goed gezet (kleur, lijnstijl, lettertype etc.). Ook wordt na het bouwen de grootte op 24 bij 24

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

pixels gezet als de annotatie kleiner is. Dit is om te voorkomen dat de annotaties zo klein worden dat ze niet meer te zien zijn. Dit brengt het extra voordeel mee dat de gebruiker aan één klik voldoende heeft om een annotatie te plaatsen, in plaats van klikken en slepen.

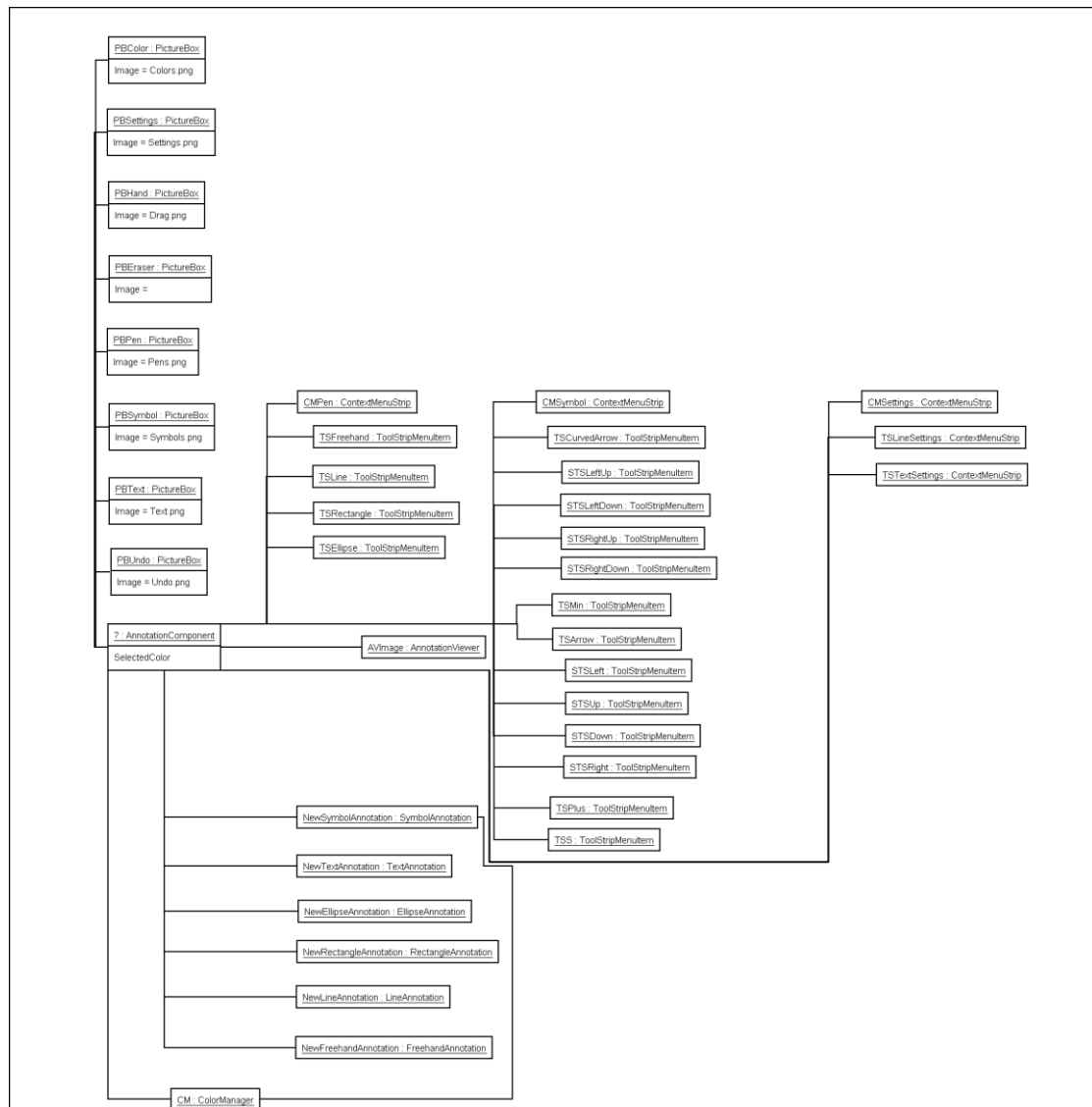
De klasse ColorManager, is verantwoordelijk voor het veranderen van de kleuren van afbeeldingen. Op het moment dat de gebruiker een andere kleur kiest, zal elke nieuwe annotatie deze kleur aannemen. Symboolannotaties werkt echter met vooraf gedefinieerde afbeeldingen en kunnen dus niet zo makkelijk van kleur veranderen als bijvoorbeeld een lijn of tekstannotatie. Deze klasse zal de bronafbeelding van de annotatie nemen, en pixel voor pixel controleren of de kleur aangepast moet worden en zo ja, in welke kleur. Dit gebeurt ook voor de afbeeldingen die te zien zijn in de contextmenu's.

De klasse UndoManager is verantwoordelijk voor het registreren van de acties die de gebruiker uitvoert. Hierdoor heeft de gebruiker de mogelijkheid zijn of haar eigen acties terug te draaien.

De klasse CursorManager tenslotte zorgt ervoor dat de cursor verandert kan worden. In de versie van VintaSoft die gebruikt wordt, bevindt zich een fout waardoor de cursor niet altijd verandert. Met deze klasse kunnen we de fout omzeilen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Object Diagram



In dit diagram is een overzicht te vinden van alle objecten en de onderlinge relatie ertussen. Ook hier valt weer op dat AnnotationComponent één van de belangrijkste onderdelen is van het project.

Een hoop van de objecten in dit diagram zijn standaard controls. Verder zijn er een aantal objecten die helpen met het aanmaken van de verschillende soorten annotaties, plus een ColorManager object dat helpt met het aanpassen van de kleur in symboolannotaties en de symbolen in de contextmenu's.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Testomgeving

Om de component na realisatie goed te kunnen testen moet er een testomgeving opgezet worden voor de component. Om de component te kunnen testen volstaat het om een nieuw formulier aan te maken waar de component in geplaatst kan worden. Dit zorgt ervoor dat er geen risico is dat andere componenten de testresultaten vervuilen.

Behalve een eigen formulier zijn er geen eisen voor de testomgeving.

Checklists

Naar aanleiding van de ontwikkelingen tijdens de ontwerpfase, zijn de checklists gemaakt in het functioneel ontwerp uitgebreid met extra checkpunten.

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Lukt het maken van een lijn met Vrij tekenen?					
Lukt het maken van een lijn met Lijn tekenen?					
Lukt het maken van een vierkant met Vierkant tekenen?					
Lukt het maken van een cirkel met Cirkel tekenen?					
Lukt het om een tekst te plaatsen met Tekst annotatie?					
Is het mogelijk om een symbool annotatie te plaatsen?					
Is het mogelijk om een annotatie meerdere malen te plaatsen?					
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?					
Is de geselecteerde annotatie te vergroten / verkleinen?					
Is de geselecteerde annotatie te roteren?					
Is de geselecteerde annotatie te verplaatsen?					
Is de geselecteerde annotatie te verwijderen?					
Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?					
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?					
Kan er een specifieke kleur gekozen worden?					
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?					
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?					
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?					
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?					
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?					
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?					
Wordt een eventuele prefix van een label niet weergegeven?					
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?					
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?					
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?					
Overig					
Is het mogelijk om (delen van) annotaties te verwijderen met de gumfunctie?					
Is het mogelijk om de afbeelding te verplaatsen met het handje?					
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?					
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan					

Naam: M.C.W. Zuidhoek

Studentnummer: 07009593

maken dan grijs?

--	--	--	--	--	--

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Testresultaten – Huidige functionaliteit

Resultaten Mike

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?	OK				
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					
Lukt het maken van een lijn met Vrij tekenen?	OK				
Lukt het maken van een lijn met Lijn tekenen?	OK				
Lukt het maken van een vierkant met Vierkant tekenen?	OK				
Lukt het maken van een cirkel met Cirkel tekenen?	OK				
Lukt het om een tekst te plaatsen met Tekst annotatie?	OK				
Is het mogelijk om een symbool annotatie te plaatsen?	OK				
Is het mogelijk om een annotatie meerdere malen te plaatsen?	OK				
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?	OK				
Is de geselecteerde annotatie te vergroten / verkleinen?	OK				
Is de geselecteerde annotatie te roteren?	OK				
Is de geselecteerde annotatie te verplaatsen?	OK				
Is de geselecteerde annotatie te verwijderen?	OK				

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?	OK				
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?	OK				
Kan er een specifieke kleur gekozen worden?	OK				
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?	OK				
Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?	OK				
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?	OK				
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?	OK				
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?	OK				
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?	OK				
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?	OK				
Wordt een eventuele prefix van een label niet weergegeven?	OK				
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?	OK				
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?	OK				
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?	OK				
Overig					
Is het mogelijk om de afbeelding te verplaatsen met het handje?	OK				
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?	OK				
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan maken dan grijs?	OK				

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Resultaten testteam

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					
Lukt het maken van een lijn met Vrij tekenen?	OK				
Lukt het maken van een lijn met Lijn tekenen?	OK				
Lukt het maken van een vierkant met Vierkant tekenen?	OK				
Lukt het maken van een cirkel met Cirkel tekenen?	OK				
Lukt het om een tekst te plaatsen met Tekst annotatie?	OK				
Is het mogelijk om een symbool annotatie te plaatsen?	OK				
Is het mogelijk om een annotatie meerdere malen te plaatsen?	OK				
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?	OK				
Is de geselecteerde annotatie te vergroten / verkleinen?	OK				
Is de geselecteerde annotatie te roteren?	OK				
Is de geselecteerde annotatie te verplaatsen?	OK				
Instellingen					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?	OK				
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?	OK				
Kan er een specifieke kleur gekozen worden?	OK				
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?	OK				
Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?	OK				
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?		Na instellen van groter lettertype wordt de tekst afgekapt. Ipv van "dingen" wordt "ding" getoond. Indien je het annotatievlak vergroot, wordt de tekst zichtbaar.			
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?	OK				
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?	OK				
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?	OK				
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?					
Wordt een eventuele prefix van een label niet weergegeven?					
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?					
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?					
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?					
Overig					
Is het mogelijk om de afbeelding te verplaatsen met het handje?					

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?	x	Deze werkt niet consistent. Soms is 1 keer klikken genoeg. Soms moet je 2 keer klikken om een actie ongedaan te maken.			
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan maken dan grijs?					

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Fase einde rapporten – Huidige functionaliteit

Requirements analyse

End Stage Report			
Project:	Annotatie component	Fase:	Requirement analyse
Project manager:	Alex van der Lans	Datum:	16-02-2011
Paraaf PM:			

Bereikte resultaten

- Requirements analyse component. (Bijgevoegd)
- Extra wensen en eisen component. (Bijgevoegd)

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:	7	7	0	0

Uitleg:

.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

- Scheiding van de extra wensen en eisen. Deze staan nu apart onder de noemer nieuwe functionaliteit.
-
-

Toetsing doelstelling en risico's (zijn ze nog steeds valide en acceptabel ?)

- Geen veranderingen in risico's en doelstelling nodig.
-

|

Requirement analyse oude functionaliteit

Algemeen

- De nieuwe component moet hetzelfde formaat hebben als de oude versie.
- Indien de afbeelding groter is dan de component, dienen schuifbalken getoond te worden waarmee de gebruiker door de afbeelding kan bewegen.
- De look van de nieuwe component dient zoveel mogelijk overeen te komen met de look van de oude component om het voor de eindgebruikers zo eenvoudig mogelijk te houden.
- De oude component controleert nu of de muis op een bepaald icoontje staat en laat aan de hand daarvan verschillende opties zien (als een soort uitschuif menu). Dit werkt echter niet helemaal zoals gewenst. Gekeken moet worden of dit in de nieuwe component anders kan.
- Afbeeldingen moeten versleept kunnen worden met het “handje”.
- Indien er een nieuwe instantie of episode gemaakt wordt, dient er een nieuwe, lege (standaard) afbeelding in het component geplaatst te worden.

Annotaties

- De component moet de volgende typen annotaties kunnen plaatsen:
 - Pen
 - Tekst
 - Symbolen
- Pen annotaties bestaan uit:
 - Lijn
 - Vierkant
 - Cirkel
 - Vrij tekenen
- Symbool annotaties bestaan uit:
 - Pijlen (in de vier windrichtingen)
 - Kruis
 - Plus en min
 - S
 - Gedraaide  pijlen
- Annotaties moeten gedraaid, verplaatst en verwijderd kunnen worden, en het moet mogelijk zijn om de annotatie te vergroten of te verkleinen.
- Annotaties moeten herhaaldelijk geplaatst kunnen worden.
- Indien een annotatie gekozen wordt dient de cursor te veranderen in de te plaatsen annotatie.
- Annotaties moeten ongedaan gemaakt kunnen worden met de knop ongedaan maken.
- De component mag alleen gebruikt worden indien de gebruiker het recht heeft om het EMD te wijzigen.
- Indien de component wordt aangeroepen vanuit gastgebruik, mag deze alleen gebruik worden als aangegeven staat dat gastgebruik toestemming heeft om het EMD te wijzigen.

Instellingen

- Alle instellingsschermen dienen in aparte vensters getoond te worden.
- Er dient een instellingscherm te zijn voor de volgende onderdelen:
 - Tekst
 - Afbeeldingen
 - Kleuren
 - Lijn
- Tekstinstellingen:
 - Het lettertype en de lettergrootte dient aangepast te kunnen worden (standaard Arial 16).
 -
- Afbeeldingsinstellingen:
 - Afbeeldingen kiezen die gedefinieerd zijn met standaard labels. Deze labels moeten ingesteld staan op actief.
 - De labels hebben standaard een prefix (XX—Label). Deze prefix mag niet getoond worden.
 - Afbeeldingen importeren vanaf de lokale PC. Alleen het JPG formaat is toegestaan.
 - Melding indien wijzigingen zijn opgeslagen voor het wisselen van de afbeelding
- Lijninstellingen:
 - Lijndikte moet ingesteld kunnen worden (standaard 2 pixels).
 - Lijntype moet ingesteld kunnen worden (vast, streep, punt, streep-punt, streep-punt-punt).
- Kleurinstellingen:
 - Er moet gekozen kunnen worden uit 2 standaardkleuren (blauw, rood waarvan blauw automatisch gekozen wordt indien de gebruiker geen kleur kiest) en 3 specifieke kleuren.
 - De standaardkleuren mogen niet gewijzigd worden.

Requirement analyse nieuwe functionaliteit

Algemeen

- Afbeelding moet na opslaan verder bewerkt kunnen worden. Annotaties worden na opslaan namelijk samengevoegd met de afbeelding en als 1 geheel opgeslagen.

Annotaties

- Zelf gemaakte annotaties toevoegen.
- Annotaties na opslaan alsnog kunnen bewerken.
- Indien er een annotatie geselecteerd is moeten de instellingen van die specifieke annotatie gewijzigd kunnen worden.

Instellingen

- Tekstinstellingen
 - Er moet gekozen kunnen worden tussen de opties normaal, **vet**, *cursief* en **vet-cursief**.
- Afbeeldingsinstellingen:
 - Sorteren op volgnummer.
- Kleurinstellingen
 - Specifieke kleuren moeten opgeslagen kunnen worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Functioneel ontwerp

End Stage Report			
Project:	Annotatiecomponent	Fase:	Functioneel ontwerp
Project manager:	Alex van der Lans	Datum:	23-02-2011
Paraaf PM:			

Bereikte resultaten

- Functioneel ontwerp:
 - Use Cases
 - UI Ontwerp
 - Testplan

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: Uren	7	7	Geen	Geen

Uitleg:

.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

-
-
-

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

- Geen verandering in risico's en doelstelling nodig
-
-

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Technisch ontwerp

End Stage Report			
Project:	Annotatiecomponent	Fase:	Technisch ontwerp
Project manager:	Alex van der Lans	Datum:	02-03-2011
Paraaf PM:			

Bereikte resultaten

- Technisch ontwerp:
 - Klassendiagram
 - Object diagram
 - Aangevulde checklisten testplan

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: • Uren	7	7	Geen	Geen

Uitleg:

.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

-
-
-

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

- Geen verandering in risico's en doelstelling nodig
-

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

•

Realisatie

End Stage Report			
Project:	Annotatiecomponent	Fase:	Realisatie
Project manager:	A. v.d. Lans	Datum:	23-03-2010
Paraaf PM:			

Bereikte resultaten

- De huidige functionaliteit van de annotatiecomponent is overgezet van VB6 naar VB.Net
- De wens om toegevoegde annotaties te verplaatsen of te bewerken voor opslaan is toegevoegd aan de component
- Standaardkleuren kunnen niet meer gewijzigd worden. (ticket 3005)

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: • Uren	10	10	0	0

Uitleg:

.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

- Door voortschrijdend inzicht is het technisch rapport meerdere malen gewijzigd (met name het klassediagram).
- Tijdens het realiseren diende een complete klasse gemaakt te worden om de functie ongedaan maken te ondersteunen. Hier was oorspronkelijk geen rekening mee gehouden. Het technisch ontwerp is aangepast om deze wijziging weer te geven. Er is hierbij geen tijdsverlies ontstaan.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

- Tijdens het realiseren bleek er een fout in de VintaSoft module te zitten die ervoor zorgde dat de cursor in de annotatieafbeelding niet veranderde. Dit is opgelost door een extra klasse te bouwen die zich focust op het veranderen van de cursor. Dit is aangepast in het technisch ontwerp. Er is hierbij geen tijdsverlies ontstaan.
- In het klassediagram was oorspronkelijk uitgegaan van één hoofdklasse annotatie met subklassen voor elk type annotatie. Dit bood echter niet voldoende flexibiliteit. Om dit op te lossen is dit oorspronkelijke idee aangepast. Nu heeft elke annotatie zijn eigen klasse die overerft van de desbetreffende klasse in de VintaSoft module. Deze wijziging is opgenomen in het technisch ontwerp. Er is hierbij geen tijdsverlies ontstaan.

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

- Er wordt nog steeds voldaan aan alle business case's.
- Er zijn geen nieuwe risico's ontdekt in deze fase.

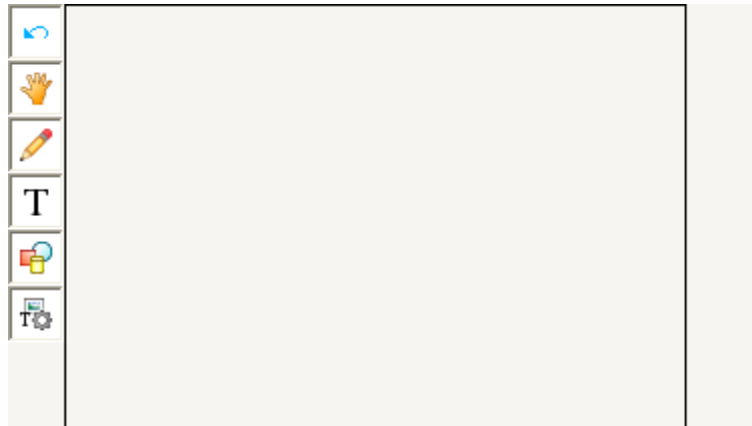
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Gebruikershandleiding – Huidige functionaliteit

Algemeen

Inleiding

In de Annotatiecomponent kan op verschillende afbeeldingen annotaties gemaakt worden bij de patiënt. Dit aangeven kan met verschillende lijnen, symbolen, teksten en figuren.



Aan de rechterkant van de component is de afbeelding te zien die op dat moment geselecteerd is. Aan de linkerkant zijn de verschillende acties te vinden.

Gebruik

Het plaatsen van annotaties

De annotatie die geplaatst moet worden kan gekozen worden door op lijnannotaties (🖋️), tekstannotaties (T) of symboolannotaties (🗨️) te klikken. Bij lijn- en symboolannotaties verschijnt er een keuzemenu waarin de juiste annotatie geselecteerd kan worden. Bij tekstannotaties verschijnt er een invoerveld waarin u de tekst opgeeft die in de annotatie geplaatst moet worden. Na het selecteren van de annotatie, kan deze geplaatst worden door op de gewenste locatie in de afbeelding te klikken. U kunt desgewenst dezelfde annotatie meerdere malen plaatsen. Met de rechter muisknop stopt u het plaatsen van annotaties.

Het bewerken van annotaties

Nadat u annotaties geplaatst heeft, kunt u deze bewerken. Selecteer de annotatie die u wilt bewerken. Vervolgens ziet u negen witte bolletjes en één groen bolletje om de annotatie verschijnen. Hierna kan een aantal bewerkingen op de geselecteerde annotatie gedaan worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Verplaatsen

Het middelste witte bolletje kan gebruikt worden om de annotatie te verplaatsen. De annotatie is te verplaatsen door te klikken en te slepen met het middelste bolletje. Laat de muisknop los als u tevreden bent over de nieuwe positie van de annotatie.

Vergroten / verkleinen

Na het plaatsen, kan het formaat van de annotatie nog aangepast te worden. Door te klikken en te verslepen van één van de witte bolletjes om de annotatie, is deze te vergroten of te verkleinen. De bolletjes in de hoeken kunnen zowel horizontaal als verticaal versleept worden, terwijl de bolletjes aan de boven en onderkant alleen verticaal, en de bolletjes links en rechts alleen horizontaal aangepast kunnen worden. Om de annotatie te spiegelen versleept u het gekozen bolletje tot voorbij zijn tegenligger (sleep bijvoorbeeld het bovenste bolletje onder het onderste bolletje om verticaal te spiegelen).

Roteren

Het roteren van de annotatie gebeurt met behulp van het groene bolletje. Klik en sleep het bolletje rond de annotatie om deze te roteren. Tijdens het verslepen verschijnt er een groene cirkel in de annotatie met een aantal lijnen erin. Beweeg de muis in deze cirkel om de annotatie in sprongen van 45 graden te roteren. Beweeg de muis buiten de cirkel om vrij te roteren.

Het verwijderen van annotaties

Geplaatste annotaties zijn te verwijderen, zolang de afbeelding nog niet is opgeslagen. Om een annotatie te verwijderen selecteert u de annotatie die verwijderd moet worden. Vervolgens drukt u op de DELETE toets om de annotatie te verwijderen.

TIP: ANNOTATIES KUNNEN NA OPSLAAN NIET MEER BEWERKT OF VERWIJDERD WORDEN.

Instellingen

In de annotatiecomponent kunnen een aantal instellingen veranderd worden. De instellingen zijn te vinden onder . Voor alle instellingen (met uitzondering van afbeeldinginstellingen) geldt dat ze alleen van toepassing zijn op annotaties die erna geplaatst worden. Eerder geplaatste annotaties zullen niet veranderen als de instellingen aangepast worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Tekstinstellingen

Onder tekstinstellingen kan het lettertype en lettergrootte van tekstannotaties aangepast worden. Standaard staat deze ingesteld op het lettertype Arial en lettergrootte 16.

Lijninstellingen

Lijnstijl	Voorbeeld
Vast	—————
Streep	- - - - -
Punt	*****
Streep-punt	- * - * - * -
Streep-punt-punt	- * - * - * - *

Hier kunnen de instellingen voor lijnannotaties (✎) aangepast worden. Lijnstijl verandert het patroon van de lijn, terwijl lijndikte aangeeft hoe breed de lijn moet worden. Standaard staan de lijninstellingen op een vaste lijn van 2 pixels.

Afbeeldinginstellingen

Onder afbeeldinginstellingen kan ingesteld worden welke afbeelding er in de component getoond moet worden. Er kan gekozen worden uit één van de vooraf gedefinieerde afbeeldingen of u kunt met de optie 'Nieuwe afbeelding' een afbeelding van uw eigen computer inladen. Tenslotte kunt u ook de standaardafbeelding opnieuw inladen. De standaardafbeelding wordt aangegeven met de tekst "(standaard)" achter de naam van de afbeelding. Hou er rekening mee dat alle geplaatste annotaties verloren zullen gaan als u de afbeelding verandert.

Tip: Als u de afbeelding wijzigt terwijl u al eens heeft opgeslagen, zal er een melding verschijnen om u te waarschuwen dat alle eerder opgeslagen gegevens ook verloren zullen gaan.

Kleurinstellingen

Onder kleurinstellingen kunt u de kleur kiezen voor de annotaties die u wilt plaatsen. U kunt kiezen uit één van de twee standaardkleuren (blauw en rood) of u kunt zelf een kleur kiezen met de optie 'Kleur kiezen'. In dit venster kunt u uit nog meer vooraf gedefinieerde kleuren kiezen of u kiest [Aangepaste kleur definiëren] en gaat zelf aan de slag om uw eigen kleur te maken.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Overige functies

Ongedaan maken

Het plaatsen, verwijderen en bewerken van annotaties kan ongedaan gemaakt worden met . Hiermee kunt u snel kleine vergissingen corrigeren.

Handje

Met het handje () kunt u de afbeelding verslepen. Dit werkt alleen als de afbeelding te groot is voor de component en geeft u een extra hulpmiddel om snel door de afbeelding te navigeren.

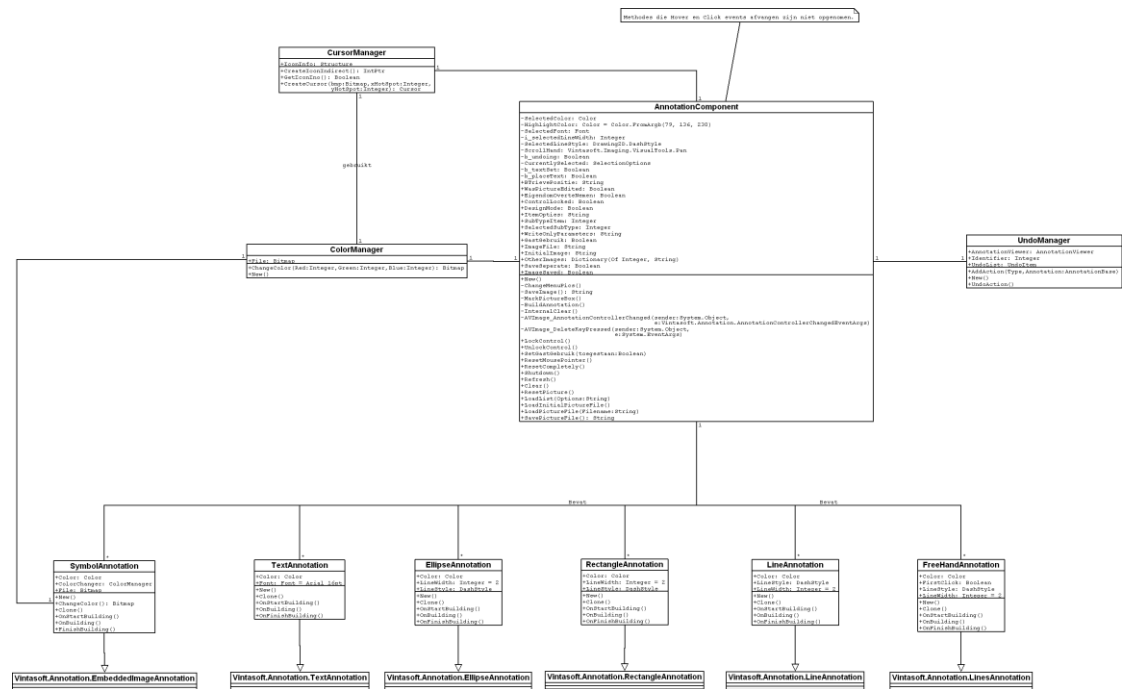
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Beheer handleiding – Huidige functionaliteit
Inleiding

Naam:
Studentnummer:

M.C.W. Zuidhoek
07009593

Klassediagram



Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

AnnotationComponent

AnnotationComponent
-SelectedColor: Color -HighlightColor: Color = Color.FromArgb(79, 136, 230) -SelectedFont: Font -i_selectedLineWidth: Integer -SelectedLineStyle: Drawing2D.DashStyle -ScrollHand: Vintasoft.Imaging.VisualTools.Pan -b_undoing: Boolean -CurrentlySelected: SelectionOptions -b_textSet: Boolean -b_placeText: Boolean +BTrievePositie: String +WasPictureEdited: Boolean +EigendomOverteNemen: Boolean +ControlLocked: Boolean +DesignMode: Boolean +ItemOpties: String +SubTypeItem: Integer +SelectedSubType: Integer +WriteOnlyParameters: String +GastGebruik: Boolean +ImageFile: String +InitialImage: String +OtherImages: Dictionary(Of Integer, String) +SaveSeperate: Boolean +ImagesSaved: Boolean
+New() -ChangeMenuPics() -SaveImage(): String -MarkPictureBox() -BuildAnnotation() -InternalClear() -AVImage_AnnotationControllerChanged(sender: System.Object, e: Vintasoft.Annotation.AnnotationControllerChangedEventArgs) -AVImage_DeleteKeyPressed(sender: System.Object, e: System.EventArgs) +LockControl() +UnlockControl() +SetGastGebruik(toegestaan: Boolean) +ResetMousePointer() +ResetCompletely() +Shutdown() +Refresh() +Clear() +ResetPicture() +LoadList(Options: String) +LoadInitialPictureFile() +LoadPictureFile(Filename: String) +SavePictureFile(): String

De klasse AnnotationComponent is de belangrijkste klasse in de component. AnnotationComponent is verantwoordelijk voor het afvangen van alle gebruikersinvoer, het aansturen van alle andere objecten (zoals bijvoorbeeld de UndoManager) en bevat alle methodes en variabelen die van buitenaf aangeroepen kunnen worden (zoals bijvoorbeeld de LockControl methode).

Lijnannotaties

EllipseAnnotation
+Color: Color +LineWidth: Integer = 2 +LineStyle: DashStyle
+New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

RectangleAnnotation
+Color: Color +LineWidth: Integer = 2 +LineStyle: DashStyle
+New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

LineAnnotation
+Color: Color +LineStyle: DashStyle +LineWidth: Integer = 2
+New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

FreeHandAnnotation
+Color: Color +FirstClick: Boolean +LineStyle: DashStyle +LineWidth: Integer = 2
+New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

EllipseAnnotation

Erft over van Vintasoft.Annotation.EllipseAnnotation. Bij het aanmaken van een nieuwe instantie wordt automatisch een standaardkleur meegegeven aan de annotatie. Deze wordt overschreven tijdens het plaatsen door de kleur die de gebruiker heeft gekozen (variabele SelectedColor). Tijdens het plaatsen worden ook de lijninstellingen doorgegeven (variabelen SelectedLineStyle en I_selectedLineWidth). Na het bouwen wordt gecontroleerd of de annotatie kleiner is dan 24 X 24 pixels, en indien nodig, vergroot naar 24 X 24.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

RectangleAnnotation

Erft over van Vintasoft.Annotation.RectangleAnnotation. Verder gelden dezelfde opmerkingen als bij EllipseAnnotation

LineAnnotation

Erft over van Vintasoft.Annotation.LineAnnotation. Bij het aanmaken van een nieuwe instantie wordt automatisch een standaardkleur meegegeven aan de annotatie. Deze wordt overschreven tijdens het plaatsen door de kleur die de gebruiker heeft gekozen (variabele SelectedColor). Tijdens het plaatsen worden ook de lijninstellingen doorgegeven (variabelen SelectedLineStyle en l_selectedLineWidth). Lijnannotaties lopen altijd van de linkerbovenhoek naar de rechteronderhoek.

FreeHandAnnotation

Erft over van Vintasoft.Annotation.FreeHandAnnotation. Ook hier worden weer een kleur en lijninstellingen overschreven bij het bouwen van de annotatie. Bij dit type annotatie is echter de property BuildingMode op Freehand gezet en daardoor kan er met deze annotatie vrij getekend worden. Tijdens het bouwen wordt ook gecontroleerd of de gebruiker de linkermuisknop ingedrukt blijft houden. Op het moment dat de gebruiker de muisknop loslaat, wordt het bouwen gestopt.

TextAnnotation

TextAnnotation
+Color: Color
+Font: Font = Arial 16pt
+New()
+Clone()
+OnStartBuilding()
+OnBuilding()
+OnFinishBuilding()

Erft over van Vintasoft.Annotation.TextAnnotation. Bij het aanmaken van een instantie wordt een kleur en een lettertype meegegeven aan de annotatie. Tijdens het bouwen wordt er een rand om de annotatie geplaatst om de grootte ervan beter te kunnen tonen. Tijdens het voltooiën van de annotatie wordt deze rand automatisch weer uitgeschakeld. Ook wordt tijdens het voltooiën, indien de gebruiker geen formaat heeft aangegeven, Het formaat van het tekstvak bepaald. Hierna wordt de positie van het tekstvak gecorrigeerd.

SymbolAnnotation

SymbolAnnotation
+Color: Color
+ColorChanger: ColorManager
+File: Bitmap
+New()
+ChangeColor(): Bitmap
+Clone()
+OnStartBuilding()
+OnBuilding()
+FinishBuilding()

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Erft over van Vintasoft.Annotation.EmbeddedImageAnnotation. Bij het aanmaken van een instantie wordt een kleur en een ColorManager meegegeven aan de annotatie (meer over de ColorManager is te vinden in de paragraaf ColorManager). Voor het bouwen verandert de ColorManager de kleur van de afbeelding in de geselecteerde kleur. Bij het bouwen van de annotatie wordt de startpositie iets opgeschoven om de annotatie onder de cursor te laten vallen. Hierdoor komt de annotatie op exact dezelfde plek als de cursor als de gebruiker enkel klikt in plaats van slepen. Na het bouwen wordt het formaat op 24X24 gezet als de annotatie kleiner is dan 24X24 (wat voornamelijk gebeurt als de gebruiker klikt in plaats van sleept).

Managers

UndoManager	ColorManager	CursorManager
+AnnotationViewer: AnnotationViewer +Identifier: Integer +UndoList: UndoItem +AddAction(Type: Annotation:AnnotationBase) +New() +UndoAction()	+File: Bitmap +ChangeColor(Red:Integer,Green:Integer,Blue:Integer): Bitmap +New()	+IconInfo: Structure +CreateIconIndirect(): IntPtr +GetIconInfo(): Boolean +CreateCursor(hbm:Bitmap,xHotSpot:Integer,yHotSpot:Integer): Cursor

UndoManager

Deze klasse is verantwoordelijk voor het registreren en terugdraaien van alle acties die de gebruiker uitvoert. Bij het aanmaken van een instantie wordt de AnnotationViewer meegestuurd aan het object. Hierdoor kan de UndoManager zelf de acties terugdraaien. Het aanmaken van een actie wordt geregeld vanuit de AnnotationComponent klasse. Als er een actie wordt uitgevoerd in de AnnotationViewer in de component, wordt er een methode aangeroepen die kijkt wat voor soort actie er wordt uitgevoerd. Als het een actie is die ondersteund wordt door de UndoManager, wordt er een registratie gemaakt van de actie. Momenteel ondersteunt de UndoManager de acties Create, Delete, Move, Rotate en Resize, met andere woorden de standaardacties die op elk soort annotatie uitgevoerd kan worden.

Bij het registreren van acties worden een aantal basisgegevens van de annotatie opgeslagen in een UndoItem (zie ook de paragraaf UndoItem). Gegevens na de transformatie worden geplaatst in de AnnotationNew variabele van het UndoItem. Gegevens voor de transformatie komen in de AnnotationOld variabele terecht.

Het terugdraaien van acties verschilt een beetje per soort. Voor elk type actie geldt echter grofweg dat eerst de AnnotationOld variabele wordt opgehaald. Vervolgens wordt de GUID van deze annotatie vergeleken met de GUIDs van alle annotaties die te vinden zijn in de AnnotationViewer. Als de GUIDs gelijk zijn aan elkaar, wordt de

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

annotatie in de AnnotationViewer aangepast met de gegevens van de AnnotationOld variabele.

ColorManager

De ColorManager is verantwoordelijk voor het veranderen van de kleuren in nieuw te plaatsen symboolannotaties of de geselecteerde symboolannotatie. Bij het aanmaken van een nieuwe instantie wordt er gelijk een afbeelding ingeladen. Met de ChangeColor methode wordt de kleur die is meegegeven aan de methode ingevuld op alle pixels die de RGB code 0,174,255 hebben. Dit is de standaardkleur van de oorspronkelijke symboolannotatie bestanden. Doordat elke pixel in een afbeelding gecontroleerd wordt, is het zaak om niet te grote afbeeldingen te gebruiken voor de ColorManager, omdat dit impact kan hebben op de performance van de component.

CursorManager

De CursorManager is verantwoordelijk voor het uiterlijk van de cursor als deze zich in de AnnotationViewer bevindt. Als de gebruiker een annotatie heeft geselecteerd, wordt dit aangegeven door de cursor te veranderen in een symbool zodra deze de AnnotationViewer betreedt. Voor lijnannotaties is dit een potlood, voor tekstannotaties een T en voor symboolannotaties het symbool zelf. Voor symbolen geldt ook dat de cursor de gekozen kleur aangeeft. Daarom bevat de klasse ook weer een instantie van de ColorManager. Let erop dat deze momenteel niet automatisch de kleur overneemt van de ColorManager instantie die in de AnnotationComponent zit.

Structures

AnnotationData

Public Structure AnnotationData
Public Guid As Guid
Public Location As PointF
Public Size As SizeF
Public Rotation As Single
Public AnnotationType As String
Public Visible As Boolean
End Structure

In deze structure worden alle gegevens van een annotatie genoteerd voor de UndoManager. De volledige annotatie kan niet opgeslagen worden omdat alle versies van de annotatie worden bijgewerkt als deze veranderd. In plaats daarvan worden alleen relevante eigenschappen van de annotatie opgeslagen. Met andere woorden, de

eigenschappen die ongedaan gemaakt kunnen worden. Als de UndoManager uitgebreid moet worden met extra acties, dienen de gegevens die nodig zijn voor de actie opgenomen te worden in AnnotationData.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Alle eigenschappen die geregistreerd worden, hebben hetzelfde type en dezelfde naam als hun tegenhangers in de annotatie. Location in de annotatie is een PointF, dus de registratie in de AnnotationData wordt ook een PointF genaamd Location. De enige uitzondering hierop is AnnotationType, die alleen de naam van de klasse bevat waaruit de annotatie is opgemaakt (bijvoorbeeld `ctlAnnotation.EllipseAnnotation`)

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

UndoItem

```
Public Structure UndoItem
    Public AnnotationOld As AnnotationData
    Public AnnotationNew As AnnotationData
    Public Type As ActionType
End Structure
```

Om een actie terug te kunnen draaien is het nodig om de gegevens van een annotatie op twee momenten op te slaan. Namelijk vlak voor en vlak na de bewerking. Deze structure zorgt ervoor

dat we de gegevens van beide momenten op kunnen slaan. De structure maakt gebruik van twee AnnotationData structures, en een ActionType Enumerator (meer over ActionType is te vinden in de paragraaf ActionType).

Enumerators

ActionType

```
Public Enum ActionType
    Create = 0
    Move = 1
    Rotate = 2
    Resize = 3
    Delete = 4
End Enum
```

De enumerator ActionType wordt gebruikt om bij het registreren van acties het type vast te stellen. De UndoManager gebruikt dit type vervolgens om de bewerkingen uit te voeren die nodig zijn om een actie terug te draaien. Als er extra soorten acties om terug te draaien worden toegevoegd aan de UndoManager

(bijvoorbeeld kleur veranderen), dan dient er ook een extra type aan deze enumerator toegevoegd te worden.

SelectionOptions

```
Public Enum SelectionOptions
    None = 0
    Cancel = 1
    'DragTool
    Hand = 2
    'PenAnnotations
    Ellipse = 3
    Line = 4
    Rectangle = 5
    Freehand = 6
    'SymbolAnnotations
    CurvedArrowLeft = 7
    CurvedArrowRight = 8
    Cross = 9
    Minus = 10
    Plus = 11
    Arrow = 12
    S = 13
    Text = 14
End Enum
```

Deze enumerator geeft alle mogelijke acties waaruit een gebruiker kan kiezen. Het gaat hier dan voornamelijk om acties die te maken hebben met het plaatsen van annotaties. De enige uitzonderingen hierop zijn Hand, Cancel en None.

- Hand: Deze optie zorgt ervoor dat de AnnotationViewer zo wordt ingesteld, dat de gebruiker er doorheen kan bewegen door te klikken en te slepen met de muis. Deze functie werkt alleen als de afbeelding groter is dan de viewer.
- Cancel: Deze optie wordt gekozen als de gebruiker op de rechter muisknop klikt. Op dat moment wordt de huidige selectie gestopt, en worden de PictureBoxes weer terug gezet op hun standaardwaarden. Ook de cursor wordt teruggezet op het standaardicoon. Als al deze bewerkingen uitgevoerd zijn, wordt er automatisch doorgeschakeld naar de optie None.
- None: Geeft aan dat er op dit moment niks geselecteerd is en dat de component dus niets hoeft te bouwen of te verplaatsen als de gebruiker klikt.

Functioneel ontwerp – Nieuwe functionaliteit

Inleiding

In dit document wordt het ontwerp voor de vernieuwde annotatiecomponent verder uitgebreid met extra functionaliteit. Het gaat dan om uitbreidingen waarvoor de wens eerder is geuit door de klant. Ook zijn er uitbreidingen opgenomen waarvan het idee ontstaan is tijdens de ontwikkeling van de component. Tenslotte wordt nader ingegaan op hoe de component getest moet worden nadat deze gerealiseerd is.

Doel van de component

Het doel van de component is om een gebruiker in staat te stellen visueel aan te geven waar een patiënt klachten heeft. De gebruiker kan op schematische weergaven van lichaamsdelen met behulp van lijnen, vormen en symbolen zeer nauwkeurig aangeven waar en wat voor soort klachten een patiënt heeft. Dit zorgt ervoor dat gebruikers in één oogopslag kunnen zien wat de bevindingen van de eerste gebruiker zijn.

Requirements

Algemeen

- Huidige functionaliteit moet beschikbaar blijven.
- Annotaties moeten na opslaan nog bewerkt kunnen worden.
- Het terugdraaien van acties moet ongedaan gemaakt kunnen worden (redo functie).

Annotaties

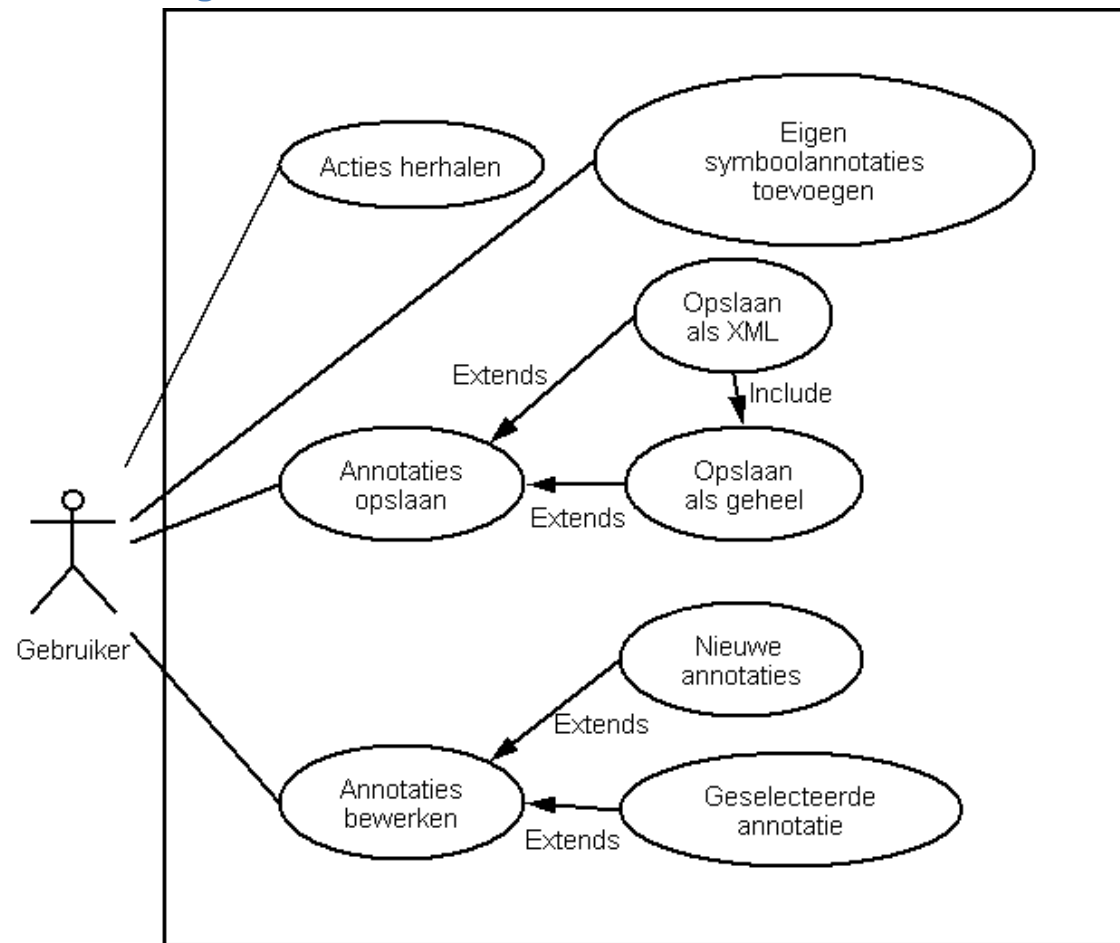
- Er moet een eigen symboolannotatie ingeladen kunnen worden voor specialismen die aan de standaard symbolen niet voldoende hebben.
- De eigenschappen van geplaatste annotaties moeten aangepast kunnen worden per annotatie (kleur, lijnstijl enzovoort).

Instellingen

- Bij Lijninstellingen moeten de teksten die de verschillende lijnstijlen aangeven vervangen worden door afbeeldingen.
- De kleur van tekstannotaties moet gewijzigd kunnen worden.

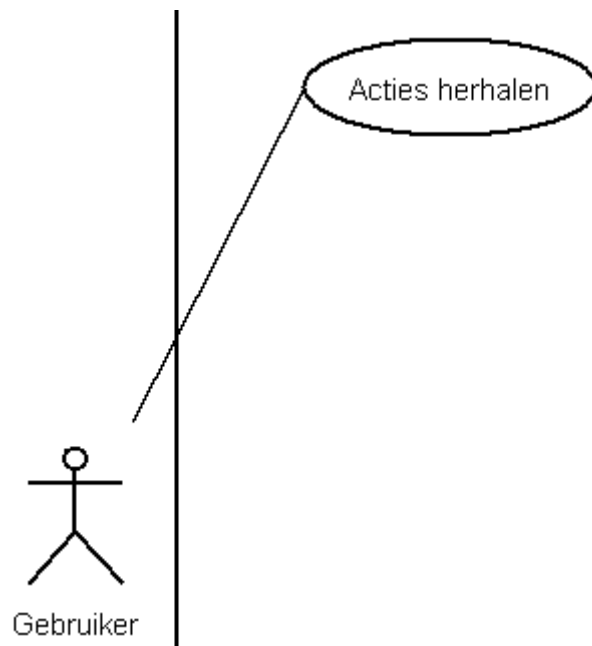
Use Cases

Use Case Diagram



Use Case Beschrijvingen

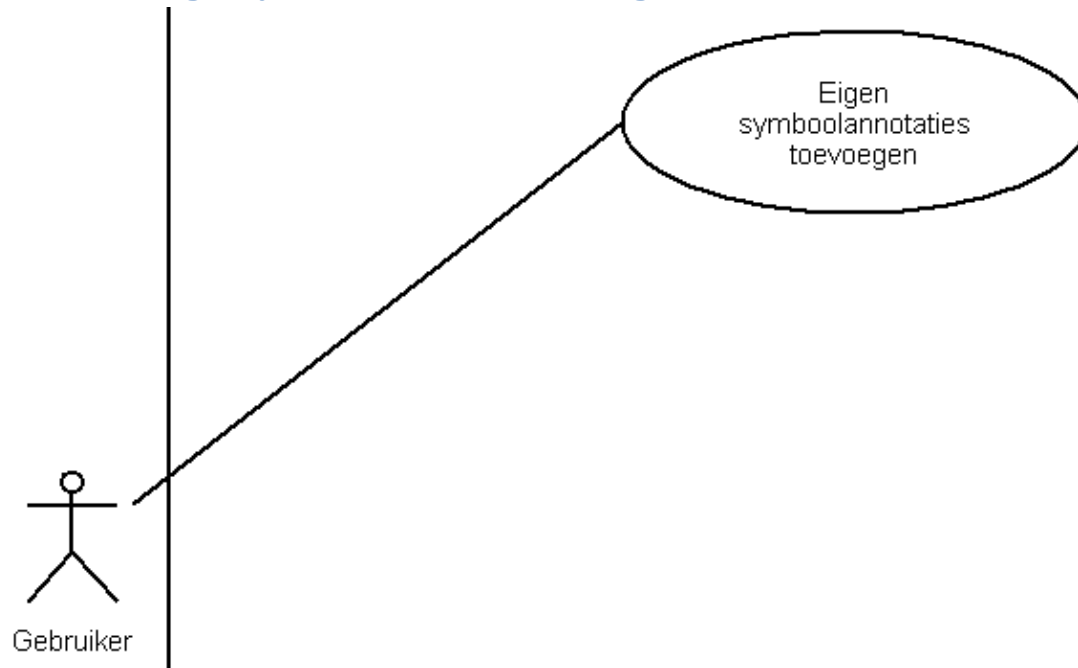
Use Case: Acties herhalen



Naam	Acties herhalen
Beschrijving	Het herhalen van eerder ongedaan gemaakte acties.
Aannamen	Er is tenminste één actie ongedaan gemaakt.
Resultaat	Een ongedaan gemaakte actie is herhaald.
Stappenplan	5. De gebruiker klikt op de knop herhalen. 6. De actie die als laatste ongedaan gemaakt is wordt opnieuw uitgevoerd.
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Als na het ongedaan maken een nieuwe actie is gedaan, werkt herhalen niet.

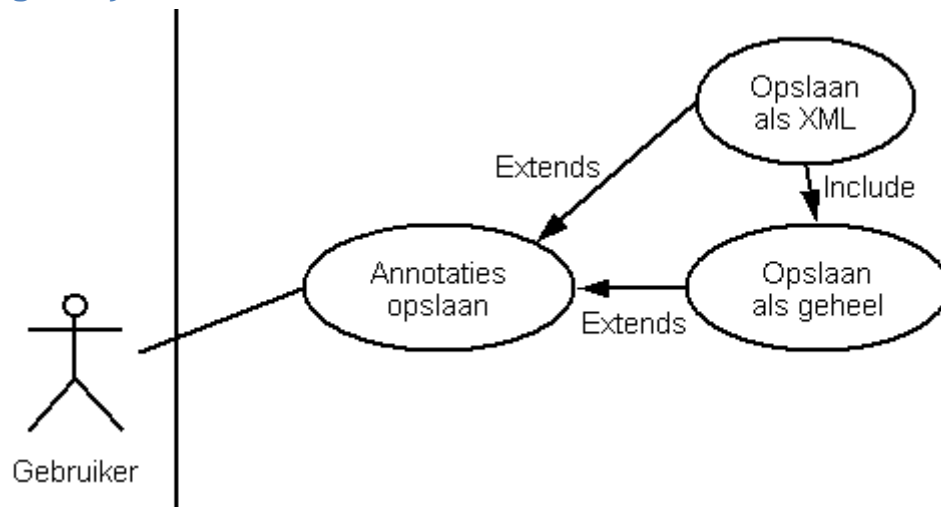
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Use Case: Eigen symboolannotaties toevoegen



Naam	Eigen symboolannotaties toevoegen
Beschrijving	Het inladen van gespecialiseerde symboolannotaties.
Aannamen	Er is tenminste één symboolannotatie gedefinieerd in het ontwikkeldossier.
Resultaat	Een gespecialiseerd symbool is ingeladen en kan als annotatie gebruikt worden.
Stappenplan	1. De gebruiker opent een formulier dat een annotatiecomponent bevat. 2. Het EMD stuurt de component een lijst met alle symboolannotaties die bij het specialisme van dat formulier horen. 3. De component voegt de symboolannotaties toe aan de lijst met symboolannotaties.
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. De standaardannotaties zullen altijd zichtbaar zijn. De specialisme symbolen kunnen niet van kleur veranderen.

Use Case: Annotaties opslaan (Extends Opslaan als XML & Opslaan als geheel)



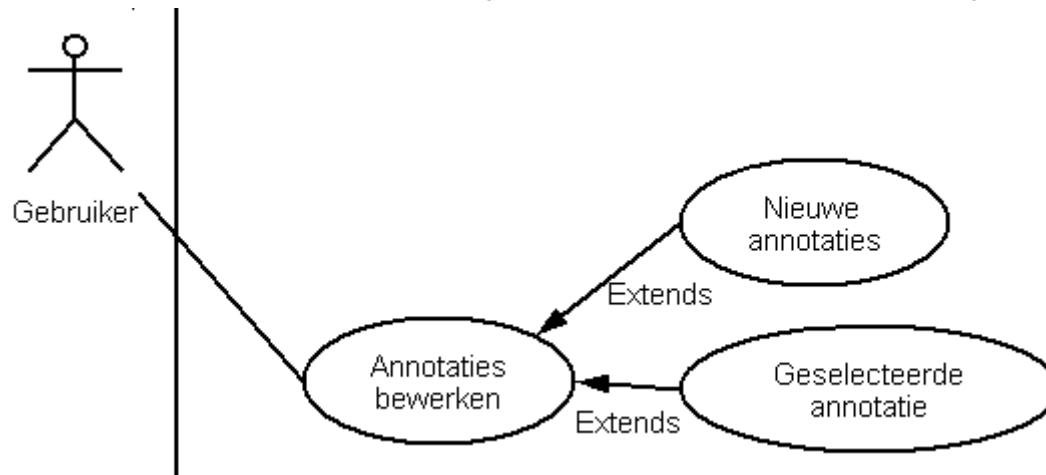
Naam	Annotaties opslaan → Opslaan als XML
Beschrijving	Het in een los bestand opslaan van de annotaties zodat deze later alsnog bewerkt kunnen worden.
Aannamen	Er is tenminste één annotatie geplaatst. De instelling 'gescheiden opslaan van annotaties' moet ingeschakeld zijn.
Resultaat	De annotaties zijn opgeslagen
Stappenplan	1. De gebruiker klikt op [Opslaan]. 2. De annotaties worden in een XML bestand gezet. 3. De Use Case Opslaan als geheel wordt uitgevoerd. 4. De bestandslocaties en bestandsnamen van het XML bestand en de afbeelding worden teruggegeven aan het EMD. 5. Het EMD plaatst de bestanden in de database.
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Naam	Annotaties opslaan als geheel
Beschrijving	Het samenvoegen van de annotaties en de afbeelding tot één Jpg afbeelding.
Aannamen	Er is tenminste één annotatie geplaatst. De instelling 'gescheiden opslaan van annotaties' moet uitgeschakeld zijn.
Resultaat	De annotaties zijn opgeslagen
Stappenplan	1. De gebruiker klikt op [Opslaan]. 2. De annotaties worden samengevoegd met de afbeelding tot één geheel. 3. Deze afbeelding wordt opgeslagen 4. De bestandslocatie en bestandsnaam worden teruggegeven
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft. Indien deze use case aangeroepen wordt vanuit Opslaan als XML, wordt stap 1 niet uitgevoerd.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Use Case: Annotaties bewerken (Extends Geselecteerde annotatie)



Naam	Annotaties bewerken → Geselecteerde annotatie
Beschrijving	Het bewerken van de geselecteerde annotatie.
Aannamen	Er is een annotatie geselecteerd.
Resultaat	De geselecteerde annotatie is bewerkt
Stappenplan	1. Ga naar het juiste instellingenschermb, afhankelijk van de instelling die veranderd moet worden. 2. Het instellingenschermb toont de <i>huidige</i> instellingen van de annotatie. 3. Verander de instelling. 4. De component verandert de instelling van de geselecteerde annotatie
Uitzonderingen	Gebruiker mag dit niet uitvoeren als deze geen rechten heeft.

Testplan

Wat wordt er getest?

Alle genoemde Use-Cases, plus de ongedaan maken functie worden getest. Alle eerder gemaakte functionaliteit, met uitzondering van de functie ongedaan maken, hoeft niet grondig getest te worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Wat wordt er niet getest?

Test taken

Elk van de genoemde testitems moet volgens de use case in het functioneel ontwerp uitgevoerd kunnen worden. Ook moet het resultaat gelijk zijn aan het verwachte resultaat van de use case. Tenslotte moet er een testomgeving ingericht worden in de vorm van een nieuw formulier met de nieuwe component erin.

Planning

Tijd/datum	Omschrijving	Wie?	(ok/x)
16-5	Aandragen onderdelen, begin eerste testronde		
17-5	Feedback eerste testronde		
18-5	Bugfixing a.d.h. van feedback eerste ronde		
19-5	Begin tweede testronde		
20-5	Voltooien testen		

Risico

- Tijdens het testen zijn er de volgende risico's die het testen kunnen belemmeren:
- **Tijdstekort:** Het kan zijn dat de geplande tijd tekort blijkt te zijn om alle genoemde onderdelen voldoende te kunnen testen. In de algemene planning is een uitloopweek opgenomen die (gedeeltelijk) gebruikt kan worden om dit op te vangen. Wel moet er gekeken worden of er onderwerpen geschrapt kunnen worden uit de tests indien deze uitloopweek nodig is.
 - **Stoppers:** Er is een kans dat tijdens het testen een fout gevonden wordt die het onmogelijk maakt om verder te kunnen met testen. Dit risico wordt teruggebracht door de ontwikkelaar alvast een aantal onderdelen te laten testen voor de testfase ingaat. Tijdens deze snelle testen wordt vooral gekeken of er geen crashes optreden bij grote functies, er wordt minder nauwkeurig gekeken dan wanneer het echte testen begint.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Checklisten

In de kolom Bevinding wordt aangegeven wat er daadwerkelijk te zien is na het uitvoeren van de check. Denk hierbij aan bijvoorbeeld foutmeldingen. In de kolom Reproductie wordt aangegeven hoe een eventuele afwijking te herproduceren is. Belangrijk is om hier de volgende punten in aan te geven:

- Omgeving (Bijvoorbeeld: welke gebruiker is er ingelogd?)
- Patiëntnummer van de testpatiënt.
- Welk formulier?
- Welke instance / episode van het formulier?
- Stappen die geleid hebben tot het afwijkend gedrag.

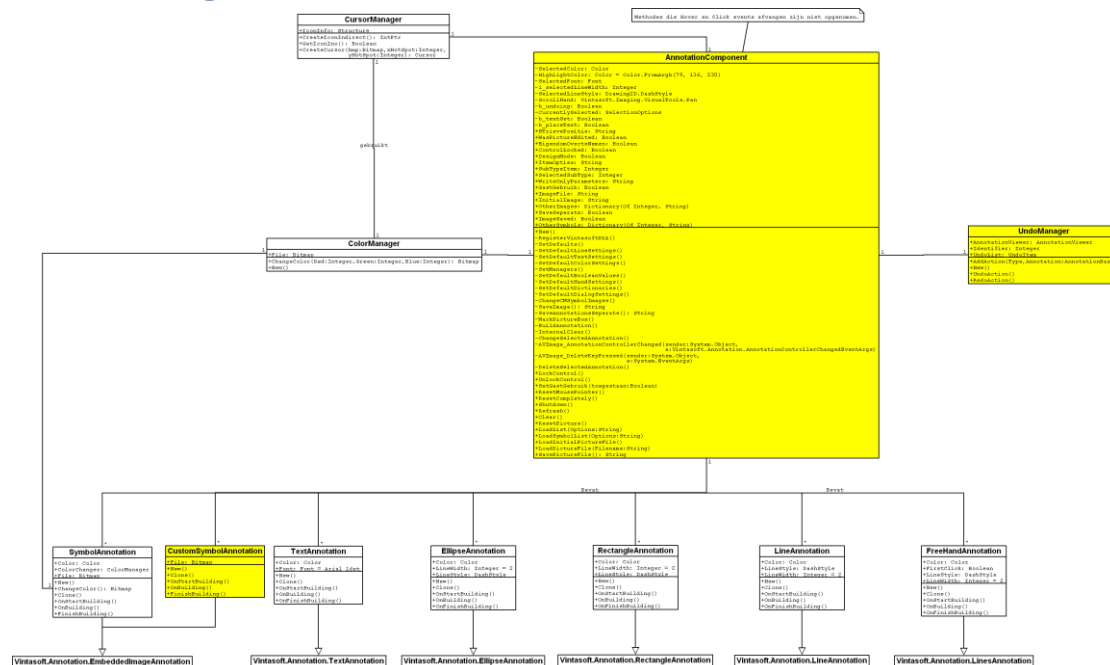
In de kolom Gewenst gedrag komt te staan wat het resultaat zou moeten zijn. Indien hier niet aan voldaan is, is er dus sprake van een fout. De kolom Prioriteit geeft tenslotte aan hoe groot de impact is van de gevonden fout en hoe snel deze opgelost moeten worden. Prioriteit kent drie categorieën:

- Laag: Dit zijn kleine foutjes in de UI (zoals een pictogram dat op de verkeerde plek staat) of foutjes die makkelijk te omzeilen zijn.
- Normaal: Dit zijn de soort fouten die ervoor zorgen dat bepaalde onderdelen niet naar behoren werken (zoals wanneer een plus wordt geplaatst terwijl de min annotatie is geselecteerd of bepaalde instellingen die niet gewijzigd kunnen worden).
- Hoog: Dit zijn de fouten die ervoor zorgen dat de component crasht (stoppers), of waardoor werken met de component nagenoeg onmogelijk wordt (bijvoorbeeld: annotaties plaatsen werkt niet)

Technisch ontwerp – Nieuwe functionaliteit

Inleiding

Klassendiagramm



In dit klassendiagram is de schematische opzet van de component te vinden. Dit klassendiagram is op veel punten gelijk aan het klassendiagram van de huidige versie van de component. De onderdelen waarop deze nieuwe versie afwijkt, zijn geel aangegeven.

In de klasse AnnotationComponent zijn een aantal nieuwe methoden toegevoegd. Zo is er de methode SaveAnnotationsSeperate, die verantwoordelijk is voor het maken van het XML bestand waarin de annotaties opgeslagen worden. Met dit XML bestand kunnen annotaties opnieuw ingeladen worden, zodat deze na opslaan bewerkt kunnen worden. Ook is de methode LoadSymbolList toegevoegd, waarmee

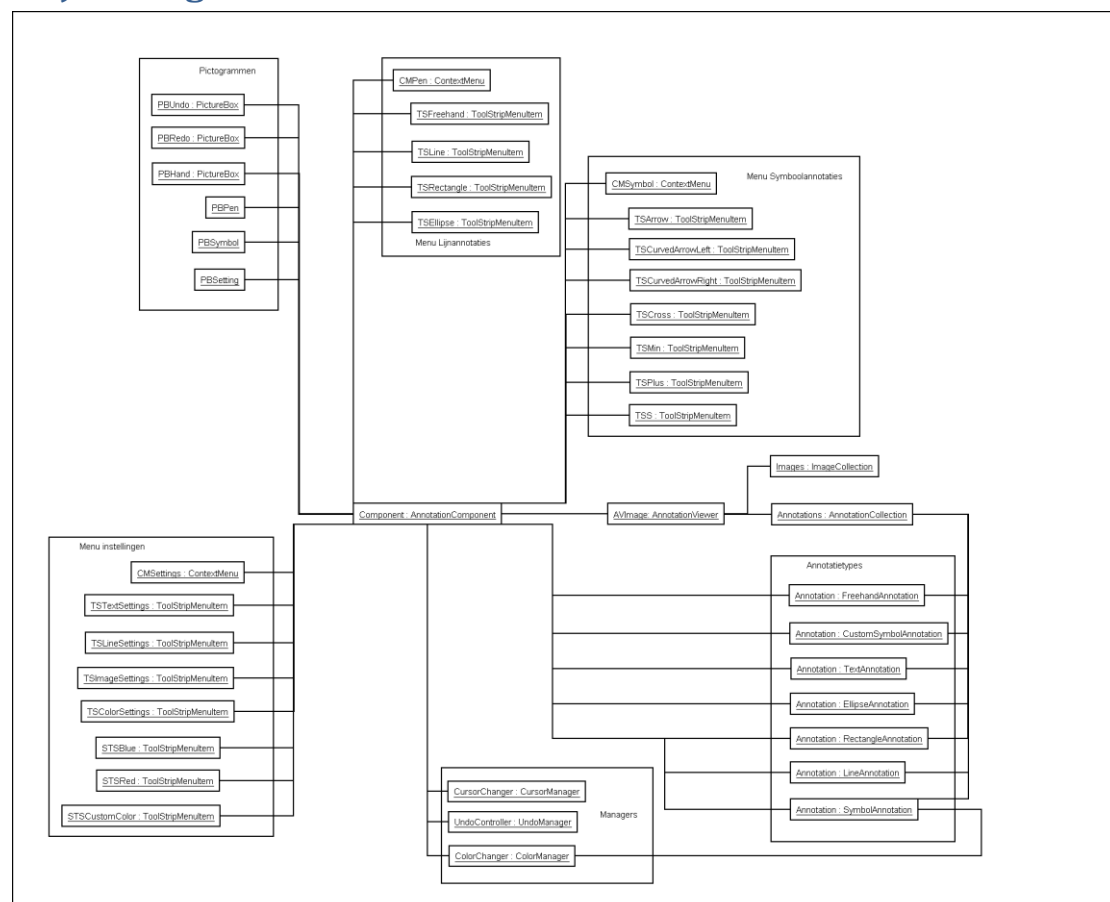
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

gespecialiseerde symboolannotaties ingeladen kunnen worden. Verder zijn er een aantal methoden toegevoegd om de klasse wat beter leesbaar te kunnen maken.

De UndoManager is uitgebreid met de methode RedoAction. Met deze methode kan een teruggedraaide actie opnieuw uitgevoerd worden.

Tenslotte is de nieuwe klasse CustomSymbolAnnotation toegevoegd. Deze klasse toont veel overeenkomsten met de klasse SymbolAnnotation. Het grootste verschil is dat CustomSymbolAnnotation niet over een ColorManager beschikt. Omdat er geen eisen gesteld kunnen worden aan de afbeeldingen die de symboolannotaties vormen, kan de kleur van de annotatie niet gewijzigd worden. De enige eis die wel aan deze annotaties gesteld wordt is dat ze een formaat van 24 bij 24 pixels hebben.

Object diagram



In dit diagram is een overzicht te vinden van alle objecten in de component. Net

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

zoals in de voorgaande versie, is ook hier het object van de klasse AnnotationComponent het belangrijkste. Dit object is verantwoordelijk voor het aanmaken en beheren van bijna alle andere objecten.

De objecten in het diagram zijn opgedeeld in een aantal groepen om het overzicht te bewaren. De groepen dienen dan ook geen enkele andere functie dan dit (ze zullen in de code dan ook niet meer terug te vinden zijn).

Menu groepen

De menu objecten zijn opgedeeld in drie categorieën: Lijnannotaties, Symboolannotaties en Instellingen. Elke groep bestaat uit een ContextMenu object, met één of meerdere ToolStripMenuItems. ToolStripMenuItems kunnen ook een aantal ToolStripMenuItems hebben. De voorvoegsels TS en STS geven aan op welk niveau het ToolStripMenuItem getoond wordt (ToolStrip en SubToolStrip). Alle objecten die een menu vormen, worden aangemaakt door het AnnotationComponent object.

Annotaties

Als een gebruiker een nieuwe annotatie wil maken, wordt deze aangemaakt door het AnnotationComponent object. Als het aanmaken voltooid is, wordt de controle over het object verplaatst naar de AnnotationCollection van de AnnotationViewer. Alle bewerkingen die vervolgens op de annotatie worden uitgevoerd (hier valt ook het daadwerkelijke tekenen van de annotatie) worden afgehandeld door de AnnotationViewer.

Testomgeving

Om de component na realisatie goed te kunnen testen moet er een testomgeving opgezet worden voor de component. Om de component te kunnen testen volstaat het om een nieuw formulier aan te maken waar de component in geplaatst kan worden. Dit zorgt ervoor dat er geen risico is dat andere componenten de testresultaten vervuilen. Het is voldoende om hetzelfde formulier te gebruiken als waarin de vorige versie van de component is getest, mits er knoppen worden toegevoegd voor de nieuwe publieke methoden (LoadSymbolList en SaveImage wanneer annotaties los opgeslagen moeten worden).

Behalve een eigen formulier zijn er geen eisen voor de testomgeving.

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Checklists

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					
Lukt het maken van een willekeurige annotatie?					
Is het mogelijk om een annotatie meerdere malen te plaatsen?					
Lukt het maken van een gespecialiseerde symboolannotatie?					
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?					
Is de geselecteerde annotatie te vergroten / verkleinen?					
Is de geselecteerde annotatie te roteren?					
Is de geselecteerde annotatie te					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

verplaatsen?					
Is de kleur van de geselecteerde annotatie aan te passen?					
Is het lettertype van de geselecteerde Tekstannotatie aan te passen?					
Is de lettergrootte van de geselecteerde Tekstannotatie aan te passen?					
Is de lijnstijl van de geselecteerde lijnannotaties aan te passen?					
Is de geselecteerde annotatie te verwijderen?					
Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?					
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?					
Kan er een specifieke kleur gekozen worden?					
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?					
Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?					
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?					
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?					
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?					
Als er een annotatie geselecteerd is terwijl één van de instellingenschermen is geopend, toont dit scherm dan de instellingen van de annotatie?					
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?					
Wordt een eventuele prefix van een label niet weergegeven?					
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?					
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?					
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?					
Overig					
Is het mogelijk om de afbeelding te verplaatsen met het handje?					
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?					
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan maken dan grijs?					

Request for Change – Klassendiagram

Request for Change			
Productnaam:	Annotatiecomponent	Versienummer:	1.0.1.0
Indiener:	M. Zuidhoek	Impact³:	T / G / K / O

³ T=Tijd, G=Geld, K=Kwaliteit, O=Omvang, BC=Business Case, R=Risico's

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

			BC / R
Datum laatste update:	22-04-2011	Naam Besliser: Paraaf Besliser:	A. van der Lans
Datum geïdentificeerd:	21-04-2011	Prioriteit⁴:	H / M / L
Beslissing⁵:	T / A / U	Status:⁶	
Beslissingsdatum:			

Beschrijving voorgestelde wijziging

Opsplitsen klasse AnnotationComponent in twee kleinere klassen: AnnotationComponent en AnnotationViewer. AnnotationComponent wordt verantwoordelijk voor de GUI en communicatie met het EMD. AnnotationViewer wordt verantwoordelijk voor het aanmaken en beheren van annotaties, en het opslaan van de afbeelding (ie opmerkingen voor het klassediagram van deze twee nieuwe klassen).

Impact voorgestelde wijziging (vb. Tijd, Geld, Kwaliteit, Omvang, Business Case, Risico's)

Het opsplitsen naar deze twee klassen zal er voor zorgen dat er meer overzicht is in de component zelf. Momenteel is alles op één grote hoop gegooid in de klasse AnnotationComponent, wat ervoor zorgt dat het moeilijker wordt om de component later aan te passen.. Het opsplitsen zal ongeveer twee dagen in beslag lopen, maar zal geen impact hebben op de planning omdat het project voor op schema ligt. Door de splitsing zou de component beter beheersbaar moeten blijven.

Indien beslissing positief: toewijzingsdetails (wie, wat, wanneer)

Indien beslissing negatief: korte motivering (waarom is de RFC afgewezen of uitgesteld)

.

⁴ H=Hoog, M=Middel, L=Laag

⁵ T=Toegewezen, A=Afgewezen, U=Uitgesteld

⁶ OK= akkoord

Ondernomen Acties: (b.v. doorverwezen naar Follow-on Actions)

Opmerkingen:

AnnotationComponent	AnnotationViewer
-HighlightColor: Color -ColorChanger: ColorManager -TIPText: TextInputDialog -b_textSet: Boolean -CMMCustomSymbol: CustomSymbolManager -s_bTrievePosition: String -b_eigendomOverTeNemen: Boolean -b_controlLocked: Boolean -s_itemOpties: String -i_subTypeItem: Integer -i_selectedSubType: Integer -s_writeOnlyParameters: String -b_gastGebruik: Boolean -d_otherImages: Dictionary(Of Integer, String) -d_customSymbols: Dictionary(Of Integer, String) -b_saveGenerate: Boolean -Initialize() -InitializeBooleanValues() -InitializeDictionaries() -InitializeDialogs() -ChangeCMSymbolImages() -MarkPictureBox() -CreateCustomSymbolAnnotation(Image:Bitmap) -InternalClear() +LockControl() +UnlockControl() +SetGastGebruik() +ResetMousePointer() +ResetCompletely() +Shutdown() +Refresh() +Clear() +ResetPicture() +LoadList(Options:String) +LoadCustomAnnotation(Options:String) +ReloadCustomSymbolMenu() +TriggerCustomAnnotations() +LoadInitialPictureFile() +LoadPictureFile(FileName:String) +SavePictureFile(): String	-SelectedColor: Color -ColorChanger: ColorManager -SelectedFont: Font -i_selectedLineWidth: Integer -ScrollHand: Vintasoft.Imaging.VisualTools.Pan -CursorChanger: CursorManager -UndoController: UndoManager -b_undoing: Boolean -CurrentlySelected: SelectionOptions -b_placeText: Boolean -AnnotationSelected: Vintasoft.Annotation.AnnotationBase -b_wasPictureEdited: Boolean -s_imageFile: String -s_initialImageFile: String -b_imageSaved: Boolean -RegisterVintasoftDLL() -SetDefaults() -SetDefaultLineSettings() -SetDefaultTextSettings() -SetManagers() -SetDefaultBooleanValues() -SetDefaultHandSettings() +SaveImage(): String +SaveAnnotationsSeparate(): String +ReloadSavedAnnotations() -BuildAnnotation() -Clear() -IsAnnotationSelected(): Boolean +ChangeSelectedAnnotationColorSettings() +ChangeSelectedAnnotationLineSettings() -AnnotationControllerChanged(sender:Object, e:Vintasoft.Annotation.AnnotationControllerChangedEventArgs) -DeleteKeyPressed() +ResetPicture() +LoadInitialPictureFile() +LoadPictureFile(FileName:String)

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

verantwoordelijkheden had, en moeilijk onderhoudbaar zou worden. De twee klassen zijn als volgt:

- De klasse AnnotationComponent is nu verantwoordelijk voor de User Interface en het afhandelen van de public methods / properties.
- De klasse AnnotationManager is verantwoordelijk voor het plaatsen, bewerken en verwijderen van annotaties. Ook methodes die te maken hebben met de annotatieafbeelding (zoals inladen en opslaan) zijn hier te vinden.

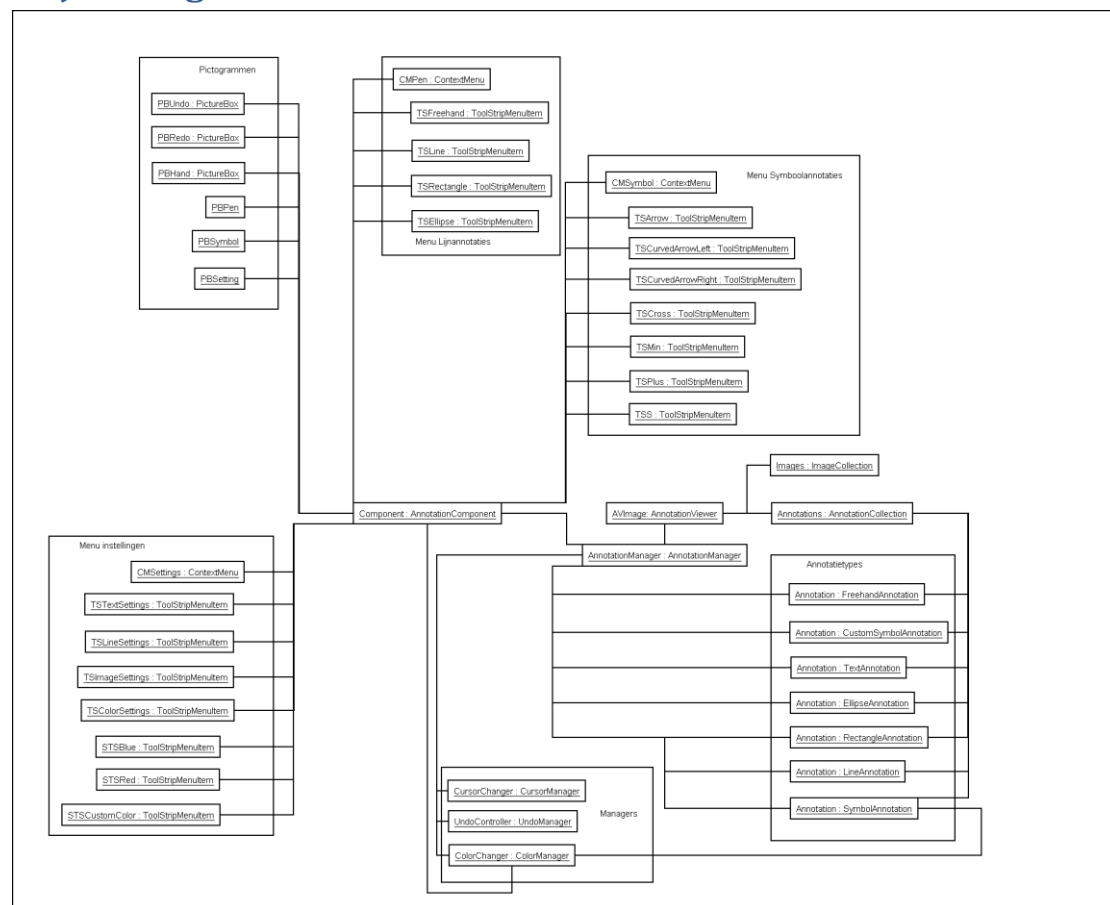
Doordat de GUI nu een aparte klasse is in de component, is het eenvoudiger om deze aan te passen of zelfs helemaal te vervangen. Ook wordt het eenvoudiger om extra functionaliteit aan de component toe te voegen, doordat deze op één locatie te vinden is, en niet vermengd met de GUI.

Tenslotte is de nieuwe klasse CustomSymbolAnnotation toegevoegd. Deze klasse toont veel overeenkomsten met de klasse SymbolAnnotation. Het grootste verschil is dat CustomSymbolAnnotation niet over een ColorManager beschikt. Omdat er geen eisen gesteld kunnen worden aan de afbeeldingen die de symboolannotaties vormen, kan de kleur van de annotatie niet gewijzigd worden. De enige eis die wel aan deze annotaties gesteld wordt is dat ze een formaat van 24 bij 24 pixels hebben.

Een aantal klassen hebben extra methoden gekregen. Zo is de UndoManager bijvoorbeeld uitgebreid met de methode RedoAction. Met deze methode kan een teruggedraaide actie opnieuw uitgevoerd worden. Ook zijn er een aantal extra methoden aan deze klasse toegevoegd om de leesbaarheid te vergroten zoals de methode FullRotationCheck(), die kijkt of een rotatie niet exact 360 graden of een veelvoud hiervan is. Ook zijn er de methodes NextItemExists() en PreviousItemExists(), die controleren of er acties voor of na de geselecteerde actie ingevuld zijn. Het resultaat van deze methodes kunnen o.a. gebruikt worden om de redo en undo knoppen in of uit te schakelen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Object diagram



In dit diagram is een overzicht te vinden van alle objecten in de component. Ook hier is het object `AnnotationComponent` opgesplitst in twee delen.

Als eerste is er nu een nieuw object `AnnotationComponent`. Deze nieuwe versie is verantwoordelijk voor het beheer van alle UI elementen (zoals menu's en `PictureBoxes`). Ook wordt vanuit dit object de opdracht tot het maken van annotaties doorgegeven.

Het object dat nu verantwoordelijk is voor alles dat met annotaties te maken heeft, is `AnnotationManager`. Dit object is verantwoordelijk voor het aanmaken van nieuwe annotaties, het bijwerken van de `UndoManager` en de `CursorManager`, en natuurlijk de `AnnotationViewer` zelf, die zorgt voor het tonen van de afbeelding en de daarop geplaatste annotaties.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

De objecten in het diagram zijn opgedeeld in een aantal groepen om het overzicht te bewaren. De groepen dienen dan ook geen enkele andere functie dan dit (ze zullen in de code dan ook niet meer terug te vinden zijn).

Menu groepen

De menu objecten zijn opgedeeld in drie groepen: Lijnannotaties, Symboolannotaties en Instellingen. Elke groep bestaat uit een ContextMenu object, met één of meerdere ToolStripMenuItems. ToolStripMenuItems kunnen ook een aantal ToolStripMenuItems hebben. De voorvoegsels TS en STS geven aan op welk niveau het ToolStripMenuItem getoond wordt (ToolStrip en SubToolStrip). Alle objecten die een menu vormen, worden aangemaakt door het AnnotationComponent object.

Annotaties

Als een gebruiker een nieuwe annotatie wil maken, wordt deze aangemaakt door het AnnotationManager object. Als het aanmaken voltooid is, wordt de controle over het object verplaatst naar de AnnotationCollection van de AnnotationViewer. Alle bewerkingen die vervolgens op de annotatie worden uitgevoerd (hier valt ook het daadwerkelijke tekenen van de annotatie) worden afgehandeld door de AnnotationViewer.

Testomgeving

Om de component na realisatie goed te kunnen testen moet er een testomgeving opgezet worden voor de component. Om de component te kunnen testen volstaat het om een nieuw formulier aan te maken waar de component in geplaatst kan worden. Dit zorgt ervoor dat er geen risico is dat andere componenten de testresultaten vervuilen. Het is voldoende om hetzelfde formulier te gebruiken als waarin de vorige versie van de component is getest, mits er knoppen worden toegevoegd voor de nieuwe publieke methoden (LoadSymbolList en SaveImage wanneer annotaties los opgeslagen moeten worden).

Behalve een eigen formulier zijn er geen eisen voor de testomgeving.

Checklists

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
------	--------------	-----------	-------------	----------------	------------

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?					
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?					
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?					
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?					
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?					
Annotaties plaatsen					
Lukt het maken van een willekeurige annotatie?					
Is het mogelijk om een annotatie meerdere malen te plaatsen?					
Lukt het maken van een gespecialiseerde symboolannotatie?					
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?					
Is de geselecteerde annotatie te vergroten / verkleinen?					
Is de geselecteerde annotatie te roteren?					
Is de geselecteerde annotatie te verplaatsen?					
Is de kleur van de geselecteerde annotatie aan te passen?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Is het lettertype van de geselecteerde Tekstannotatie aan te passen?					
Is de lettergrootte van de geselecteerde Tekstannotatie aan te passen?					
Is de lijnstijl van de geselecteerde lijnannotaties aan te passen?					
Is de geselecteerde annotatie te verwijderen?					
Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?					
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?					
Kan er een specifieke kleur gekozen worden?					
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?					
Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?					
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?					
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?					
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?					
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Als er een annotatie geselecteerd is terwijl één van de instellingenschermen is geopend, toont dit scherm dan de instellingen van de annotatie?					
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?					
Wordt een eventuele prefix van een label niet weergegeven?					
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?					
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?					
Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?					
Overig					
Is het mogelijk om de afbeelding te verplaatsen met het handje?					
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?					
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan maken dan uitgeschakeld?					

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Testresultaten – Nieuwe functionaliteit

Resultaten Mike

Item	Check (OK/X)	Bevinding	Reproductie	Gewenst gedrag	Prioriteit
Opstarten van de component					
Wordt bij het starten van een nieuwe instance/episode de standaardafbeelding ingeladen?	X	Niet getest in Norma			
Wordt bij het starten van een bestaande instance / episode de eerder gemaakte afbeelding ingeladen?	X	Niet getest in Norma			
Verschuiven er schuifbalken indien de afbeelding groter is dan de component?	OK				
Als de gebruiker niet het recht heeft om de component over te nemen, Kan de gebruiker de component dan inderdaad niet gebruiken?	X	Niet getest in Norma			
Als de gebruiker aangemeld is als gastgebruiker, mag deze dan de component niet gebruiken als gastgebruik is ingesteld op niet toegestaan?	X	Niet getest in Norma			
Annotaties plaatsen					
Lukt het maken van een willekeurige annotatie?	OK				
Is het mogelijk om een annotatie meerdere malen te plaatsen?	OK				
Lukt het maken van een gespecialiseerde symboolannotatie?	OK				
Annotaties bewerken / verwijderen					
Is een annotatie te selecteren?	OK				
Is de geselecteerde annotatie te vergroten / verkleinen?	OK				
Is de geselecteerde annotatie te roteren?	OK				
Is de geselecteerde annotatie te verplaatsen?	OK				
Is de kleur van de geselecteerde annotatie aan te passen?	OK	Met uitzondering van gespecialiseerde symbolen (klopt)			
Is het lettertype van de geselecteerde Tekstannotatie aan te passen?	OK				

Naam: M.C.W. Zuidhoek
 Studentnummer: 07009593

Is de lettergrootte van de geselecteerde Tekstannotatie aan te passen?	OK				
Is de lijnstijl van de geselecteerde lijnannotaties aan te passen?	OK				
Is de geselecteerde annotatie te verwijderen?	OK				
Instellingen					
Kan er één van de twee standaardkleuren (blauw en rood) ingesteld worden?	OK				
Als er geen kleur aangegeven is, wordt er dan gebruik gemaakt van de kleur blauw?	OK				
Kan er een specifieke kleur gekozen worden?	OK				
Veranderen de icoontjes in de contextmenu's mee als de kleur verandert wordt?	OK				
Hebben annotaties die geplaatst worden na het veranderen van de kleur, de nieuwe kleur meegekregen?	OK				
Kan de lettergrootte voor nieuwe tekstannotaties ingesteld worden?	OK				
Kunnen de tekstinstellingen voor nieuwe tekstannotaties ingesteld worden?	OK				
Kan de lijndikte voor nieuwe lijnannotaties ingesteld worden?	OK				
Kan de lijnstijl voor nieuwe lijnannotaties ingesteld worden?	OK				
Als er een annotatie geselecteerd is terwijl één van de instellingschermen is geopend, toont dit scherm dan de instellingen van de annotatie?	OK				
Afbeeldingen					
Kan de afbeelding veranderd worden aan de hand van een label?	OK				
Wordt een eventuele prefix van een label niet weergegeven?	OK				
Kan er een .JPG afbeelding ingeladen worden van de lokale harde schijf?	OK				
Kan de standaardafbeelding opnieuw ingeladen worden als deze veranderd is?	OK				

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Indien de afbeelding gewijzigd wordt terwijl deze al eens is opgeslagen, wordt er dan een melding getoond?	OK				
Overig					
Is het mogelijk om de afbeelding te verplaatsen met het handje?	OK				
Is het mogelijk om acties ongedaan te maken met de knop ongedaan maken?	OK				
Indien er geen acties ongedaan te maken zijn, is de knop ongedaan maken dan grijs?	OK				

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Fase einde rapporten – Nieuwe functionaliteit

Functioneel ontwerp

End Stage Report			
Project:	Annotatiecomponent	Fase:	Functioneel ontwerp nieuwe functionaliteit
Project manager:	A. v.d. Lans	Datum:	08-04-2011
Paraaf PM:			

Bereikte resultaten

- Use Case
- Testplan
-

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: • Uren	3	4	1	-19

Uitleg:

.Door afwijkend verloop van de test en implementatie fase is alvast vooruit gewerkt om tijdsverlies te voorkomen. De test en implementatie fases zijn op dit moment in handen van personen buiten de projectgroep.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

- Voorlopen op schema door onverwachte wendingen in fases testen en implementatie
-
-

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

- Er zijn geen wijzigingen nodig in de business case en / of risico's

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

•

Technisch ontwerp

End Stage Report			
Project:	Annotatiecomponent	Fase:	Technisch ontwerp nieuwe functionaliteit
Project manager:	A. v.d. Lans	Datum:	13-4-2011
Paraaf PM:			

Bereikte resultaten

- Klassendiagram
- Objectdiagram
- Uitgewerkte checklisten

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: • Uren	3	3	0	-19

Uitleg:

.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

- Suggestie om Singleton pattern te gebruiken voor de ColorManager klasse. Dit zal geen tijdsverlies opleveren.
-
-

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

- Geen wijzigingen in Business Case en / of Risico's nodig
-

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

•

Realisatie

End Stage Report			
Project:	Annotatiecomponent	Fase:	Realisatie Nieuwe functionaliteit
Project manager:	A. v.d. Lans	Datum:	09-05-2011
Paraaf PM:			

Bereikte resultaten

- De bestaande functionaliteit van de annotatiecomponent is uitgebreid met extra functionaliteit.
- Eigen symbolen toevoegen is nu mogelijk (Ticket 63)
- Wijzigingen aanbrengen in eerder opgeslagen is nu mogelijk door de annotaties los als XML op te slaan (Ticket 4079)
- Instellingen kunnen nu voor alleen de geselecteerde annotatie aangepast worden.
- Er is een redo functie toegevoegd om ongedaan gemaakte acties ongedaan te maken.

Belangrijkste cijfers van de afgelopen periode

	Gepland (in dagen)	Werkelijk (in dagen)	Afwijkingen in deze fase	Total project afwijking
Doorlooptijd:				
Kosten: • Uren	14	22	8	-4

Uitleg:

.De uitloop in deze fase is voor een groot deel veroorzaakt door het uitsplitsen van de klasse AnnotationComponent (zie ook Y:\Medewerkers\Mike\Nieuwe functionaliteit\Request for Change document KD.doc). Omdat we voorliepen op de planning is dit verder niet van invloed op de planning.

Gebeurtenissen van invloed op de voortgang van het project (negatief en positief)

- Klasse AnnotationComponent is gesplitst om de onderhoudbaarheid te vergroten.
-
-

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Toetsing business case en risico's (zijn ze nog steeds valide en acceptabel ?)

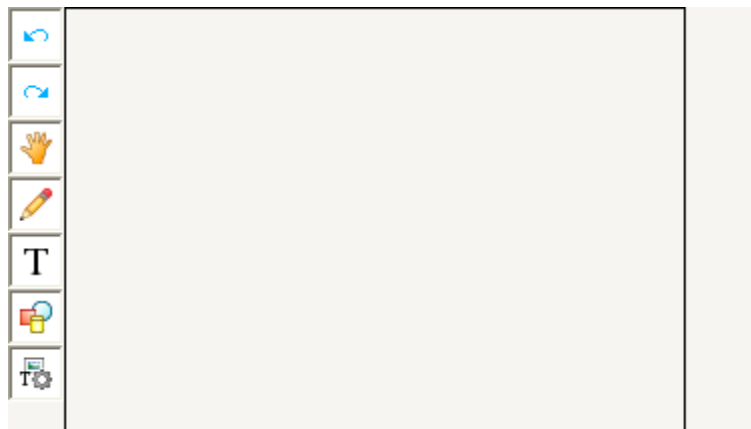
- Business case en Risico's zijn nog steeds valide en acceptabel
 -
 -
-

Gebruikershandleiding – Nieuwe functionaliteit

Algemeen

Inleiding

In de Annotatiecomponent kan op verschillende afbeeldingen annotaties gemaakt worden bij de patiënt. Dit aangeven kan met verschillende lijnen, symbolen, teksten en figuren.



Aan de rechterkant van de component is de afbeelding te zien die op dat moment geselecteerd is. Aan de linkerkant zijn de verschillende acties te vinden.

Gebruik

Het plaatsen van annotaties

De annotatie die geplaatst moet worden kan gekozen worden door op lijnannotaties (🖋️), tekstannotaties (T) of symboolannotaties (📌) te klikken. Bij lijn- en symboolannotaties verschijnt er een keuzemenu waarin de juiste annotatie geselecteerd kan worden. In het keuzemenu van symboolannotaties is ook de optie Specifiek symbool te vinden. Hieronder staan annotaties die toegespitst zijn op de specialisme(n) van het formulier. Indien er geen specifieke annotaties gedefinieerd zijn, is de optie uitgegrijsd. Bij tekstannotaties verschijnt er een invoerveld waarin u de tekst opgeeft die in de annotatie geplaatst moet worden. Na het selecteren van de annotatie, kan deze geplaatst worden door op de gewenste locatie in de

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

afbeelding te klikken. U kunt desgewenst dezelfde annotatie meerdere malen plaatsen. Met de rechter muisknop stopt u het plaatsen van annotaties.

Het bewerken van annotaties

Nadat u annotaties geplaatst heeft, kunt u deze bewerken. Selecteer de annotatie die u wilt bewerken. Vervolgens ziet u negen witte bolletjes en één groen bolletje om de annotatie verschijnen. Hierna kan een aantal bewerkingen op de geselecteerde annotatie gedaan worden.

Verplaatsen

Het middelste witte bolletje kan gebruikt worden om de annotatie te verplaatsen. De annotatie is te verplaatsen door te klikken en te slepen met het middelste bolletje. Laat de muisknop los als u tevreden bent over de nieuwe positie van de annotatie.

Vergroten / verkleinen

Na het plaatsen, kan het formaat van de annotatie nog aangepast te worden. Door te klikken en te verslepen van één van de witte bolletjes om de annotatie, is deze te vergroten of te verkleinen. De bolletjes in de hoeken kunnen zowel horizontaal als verticaal versleept worden, terwijl de bolletjes aan de boven en onderkant alleen verticaal, en de bolletjes links en rechts alleen horizontaal aangepast kunnen worden. Om de annotatie te spiegelen versleept u het gekozen bolletje tot voorbij zijn tegenligger (sleep bijvoorbeeld het bovenste bolletje onder het onderste bolletje om verticaal te spiegelen).

Roteren

Het roteren van de annotatie gebeurt met behulp van het groene bolletje. Klik en sleep het bolletje rond de annotatie om deze te roteren. Tijdens het verslepen verschijnt er een groene cirkel in de annotatie met een aantal lijnen erin. Beweeg de muis in deze cirkel om de annotatie in sprongen van 45 graden te roteren. Beweeg de muis buiten de cirkel om vrij te roteren.

Het verwijderen van annotaties

Geplaatste annotaties zijn te verwijderen, zolang de afbeelding nog niet is opgeslagen. Om een annotatie te verwijderen selecteert u de annotatie die verwijderd moet worden. Vervolgens drukt u op de DELETE toets om de annotatie te verwijderen.

TIP: ANNOTATIES KUNNEN NA OPSLAAN NIET MEER BEWERKT OF VERWIJDERD WORDEN.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Instellingen

In de annotatiecomponent kunnen een aantal instellingen veranderd worden. De instellingen zijn te vinden onder . Voor alle instellingen (met uitzondering van afbeeldinginstellingen) geldt dat ze alleen van toepassing zijn op annotaties die erna geplaatst worden. Eerder geplaatste annotaties zullen alleen veranderen als de instellingen aangepast worden, terwijl de annotatie geselecteerd is.

Tekstinstellingen

Onder tekstinstellingen kan het lettertype en lettergrootte van tekstannotaties aangepast worden. Standaard staat deze ingesteld op het lettertype Arial en lettergrootte 16.

Lijninstellingen

Lijnstijl	Voorbeeld
Vast	—————
Streep	- - - - -
Punt	*****
Streep-punt	- * - * - * -
Streep-punt-punt	- * - * - * -

Hier kunnen de instellingen voor lijnannotaties () aangepast worden. Lijnstijl verandert het patroon van de lijn, terwijl lijndikte aangeeft hoe breed de lijn moet worden. Standaard staan de lijninstellingen op een vaste lijn van 2 pixels.

Afbeeldinginstellingen

Onder afbeeldinginstellingen kan ingesteld worden welke afbeelding er in de component getoond moet worden. Er kan gekozen worden uit één van de vooraf gedefinieerde afbeeldingen of u kunt met de optie 'Nieuwe afbeelding' een afbeelding van uw eigen computer inladen. Tenslotte kunt u ook de standaardafbeelding opnieuw inladen. De standaardafbeelding wordt aangegeven met de tekst "(standaard)" achter de naam van de afbeelding. Hou er rekening mee dat alle geplaatste annotaties verloren zullen gaan als u de afbeelding verandert.

Tip: Als u de afbeelding wijzigt terwijl u al eens heeft opgeslagen, zal er een melding verschijnen om u te waarschuwen dat alle eerder opgeslagen gegevens ook verloren zullen gaan.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Kleurinstellingen

Onder kleurinstellingen kunt u de kleur kiezen voor de annotaties die u wilt plaatsen. U kunt kiezen uit één van de twee standaardkleuren (blauw en rood) of u kunt zelf een kleur kiezen met de optie 'Kleur kiezen'. In dit venster kunt u uit nog meer vooraf gedefinieerde kleuren kiezen of u kiest [Aangepaste kleur definiëren] en gaat zelf aan de slag om uw eigen kleur te maken. De kleur van elk type annotatie kan bewerkt worden, met uitzondering van specifieke symboolannotaties.

Overige functies

Ongedaan maken

Het plaatsen, verwijderen en bewerken van annotaties kan ongedaan gemaakt worden met . Hiermee kunt u snel kleine vergissingen corrigeren.

Herhalen

Het tegenovergestelde van ongedaan maken. Hiermee kunnen teruggedraaide acties opnieuw uitgevoerd worden. Deze actie kan uitgevoerd worden met .

Handje

Met het handje () kunt u de afbeelding verslepen. Dit werkt alleen als de afbeelding te groot is voor de component en geeft u een extra hulpmiddel om snel door de afbeelding te navigeren.

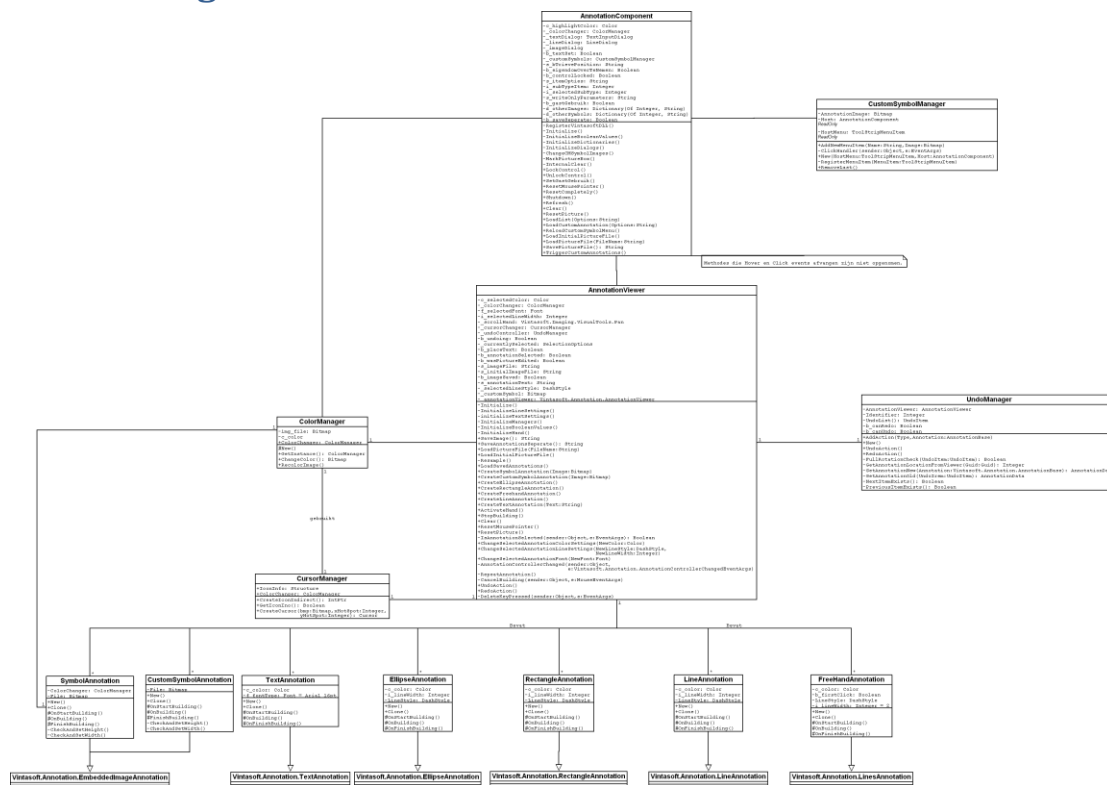
Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Beheer handleiding – Nieuwe functionaliteit

Inleiding

M.C.W. Zuidhoek
07009593

civ



Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

AnnotationComponent

De klasse AnnotationComponent is verantwoordelijk voor de communicatie met de gebruiker en het EMD. In deze klasse bevinden zich de public methods en properties waar het EMD gebruik van maakt. In een enkel geval verwijst de public method in AnnotationComponent gelijk door naar een Method in een andere klasse (meestal AnnotationManager).

De klasse genereert de interface, en geeft de acties van de gebruikers door aan de juiste klassen.

AnnotationComponent
-c highlightColor: Color - colorChanger: ColorManager - _textDialog: TextInputDialog - _lineDialog: LineDialog - _imageDialog -b textSet: Boolean - _customSymbols: CustomSymbolManager -s bTrievePosition: String -b eigendomOverTeNemen: Boolean -b controlLocked: Boolean -s itemOpties: String -i subTypeItem: Integer -i selectedSubType: Integer -s writeOnlyParameters: String -b gastGebruik: Boolean -d otherImages: Dictionary(Of Integer, String) -d otherSymbols: Dictionary(Of Integer, String) -b saveSeparate: Boolean -RegisterVintasoftDLL() -Initialize() -InitializeBooleanValues() -InitializeDictionaries() -InitializeDialogs() -ChangeCMSymbolImages() -MarkPictureBox() -InternalClear() +LockControl() +UnlockControl() +SetGastGebruik() +ResetMousePointer() +ResetCompletely() +Shutdown() +Refresh() +Clear() +ResetPicture() +LoadList(Options:String) +LoadCustomAnnotation(Options:String) +ReloadCustomSymbolMenu() +LoadInitialPictureFile() +LoadPictureFile(FileName:String) +SavePictureFile(): String +TriggerCustomAnnotations()

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

AnnotationManager

AnnotationViewer
<pre>-c_selectedColor: Color -c_colorChanger: ColorManager -f_selectedFont: Font -i_selectedLineWidth: Integer -ScrollHand: Vintasoft.Imaging.VisualTools.Pan -c_cursorChanger: CursorManager -undoController: UndoManager -b_undoing: Boolean -CurrentlySelected: SelectionOptions -b_placeText: Boolean -b_annotationSelected: Boolean -b_wasPictureEdited: Boolean -s_imageFile: String -s_initialImageFile: String -b_imageSaved: Boolean -s_annotationText: String -SelectedLineStyle: DashStyle -customSymbol: Bitmap -annotationViewer: Vintasoft.Annotation.AnnotationViewer -Initialize() -InitializeLineSettings() -initializeTextSettings() -InitializeManagers() -InitializeBooleanValues() -InitializeHand() +SaveImage(): String +SaveAnnotationsSeperate(): String +LoadPictureFile(Filename:String) +LoadInitialPictureFile() -Resample() +LoadSavedAnnotations() +CreateSymbolAnnotation(Image:Bitmap) +CreateCustomSymbolAnnotation(Image:Bitmap) +CreateEllipseAnnotation() +CreateRectangleAnnotation() +CreateFreehandAnnotation() +CreateLineAnnotation() +CreateTextAnnotation(Text:String) +ActivateHand() +StopBuilding() +Clear() +ResetMousePointer() +ResetPicture() -IsAnnotationSelected(sender:Object,e:EventArgs): Boolean +ChangeSelectedAnnotationColorSettings(NewColor:Color) +ChangeSelectedAnnotationLineSettings(NewLineStyle:DashStyle, NewLineWidth:Integer) +ChangeSelectedAnnotationFont(NewFont:Font) -AnnotationControllerChanged(sender:Object, e:Vintasoft.Annotation.AnnotationControllerChangedEventArgs) -RepeatAnnotation() -CancelBuilding(sender:Object,e:MouseEventArgs) +UndoAction() +RedoAction() -DeleteKeyPressed(sender:Object,e:EventArgs)</pre>

De klasse AnnotationManager is verantwoordelijk voor het aanmaken en beheren van annotaties en de afbeelding waarop deze geplaatst worden. De standaardinstellingen voor annotaties worden ook in deze klasse bijgehouden (c_selectedColor, f_selectedFont, i_selectedLineWidth en _selectedLineStyle). Er zijn properties aanwezig die de status van het _annotationviewer object aangeven (zoals AnnotationSelected of CurrentlySelected), en

methodes die het aanmaken van de verschillende annotaties afhandelen. Hierdoor is deze klasse makkelijk herbruikbaar voor wanneer er besloten wordt de GUI (AnnotationComponent) te vernieuwen of te vervangen.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Lijnannotaties

EllipseAnnotation
+Color: Color +LineWidth: Integer = 2 +LineStyle: DashStyle +New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

RectangleAnnotation
+Color: Color +LineWidth: Integer = 2 +LineStyle: DashStyle +New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

LineAnnotation
+Color: Color +LineStyle: DashStyle +LineWidth: Integer = 2 +New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

FreeHandAnnotation
+Color: Color +FirstClick: Boolean +LineStyle: DashStyle +LineWidth: Integer = 2 +New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

EllipseAnnotation

Erft over van Vintasoft.Annotation.EllipseAnnotation. Bij het aanmaken van een nieuwe instantie wordt automatisch een standaardkleur meegegeven aan de annotatie. Deze wordt overschreven tijdens het plaatsen door de kleur die de gebruiker heeft gekozen (variabele SelectedColor). Tijdens het plaatsen worden ook de lijninstellingen doorgegeven (variabelen SelectedLineStyle en I_selectedLineWidth). Na het bouwen wordt gecontroleerd of de annotatie kleiner is dan 24 X 24 pixels, en indien nodig, vergroot naar 24 X 24.

RectangleAnnotation

Erft over van Vintasoft.Annotation.RectangleAnnotation. Verder gelden dezelfde opmerkingen als bij EllipseAnnotation

LineAnnotation

Erft over van Vintasoft.Annotation.LineAnnotation. Bij het aanmaken van een nieuwe instantie wordt automatisch een standaardkleur meegegeven aan de annotatie. Deze wordt overschreven tijdens het plaatsen door de kleur die de gebruiker heeft gekozen (variabele SelectedColor). Tijdens het plaatsen worden ook de lijninstellingen doorgegeven (variabelen SelectedLineStyle en I_selectedLineWidth). Lijnannotaties lopen altijd van de linkerbovenhoek naar de rechteronderhoek.

FreeHandAnnotation

Erft over van Vintasoft.Annotation.LinesAnnotation. Ook hier worden weer een kleur en lijninstellingen overschreven bij het bouwen van de annotatie. Bij dit type annotatie is echter de property BuildingMode op Freehand gezet en daardoor kan er met deze annotatie vrij getekend worden. Tijdens het bouwen wordt ook gecontroleerd of de gebruiker de linkermuisknop ingedrukt blijft houden. Op het moment dat de gebruiker de muisknop loslaat, wordt het bouwen gestopt.

TextAnnotation

TextAnnotation
+Color: Color +Font: Font = Arial 16pt +New() +Clone() +OnStartBuilding() +OnBuilding() +OnFinishBuilding()

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Erft over van Vintasoft.Annotation.TextAnnotation. Bij het aanmaken van een instantie wordt een kleur en een lettertype meegegeven aan de annotatie. Tijdens het bouwen wordt er een rand om de annotatie geplaatst om de grootte ervan beter te kunnen tonen. Tijdens het voltooiën van de annotatie wordt deze rand automatisch weer uitgeschakeld. Ook wordt tijdens het voltooiën, indien de gebruiker geen formaat heeft aangegeven, Het formaat van het tekstvak bepaald. Hierna wordt de positie van het tekstvak gecorrigeerd.

SymbolAnnotation

SymbolAnnotation
+Color: Color +ColorChanger: ColorManager +File: Bitmap +New() +ChangeColor(): Bitmap +Clone() +OnStartBuilding() +OnBuilding() +FinishBuilding()

Erft over van Vintasoft.Annotation.EmbeddedImageAnnotation. Bij het aanmaken van een instantie wordt een kleur en een ColorManager meegegeven aan de annotatie (meer over de ColorManager is te vinden in de paragraaf ColorManager). Voor het bouwen verandert de ColorManager de kleur van de afbeelding in de geselecteerde kleur. Bij het bouwen van de annotatie wordt de startpositie iets opgeschoven om de annotatie onder de cursor te laten vallen. Hierdoor komt de annotatie op exact dezelfde plek als de cursor als de gebruiker enkel klikt in plaats van slepen. Na het bouwen wordt het formaat op 24X24 gezet als de annotatie kleiner is dan 24X24 (wat voornamelijk gebeurt als de gebruiker klikt in plaats van sleept).

Managers

UndoManager
+AnnotationViewer: AnnotationViewer +Identifier: Integer +UndoList: UndoItem +AddAction(Type, Annotation: AnnotationBase) +New() +UndoAction()

ColorManager
+File: Bitmap +ChangeColor(Red: Integer, Green: Integer, Blue: Integer): Bitmap +New()

CursorManager
+IconInfo: Structure +CreateIconIndirect(): IntPtr +GetIconInfo(): Boolean +CreateCursor(hbm: Bitmap, xHotSpot: Integer, yHotSpot: Integer): Cursor

UndoManager

Deze klasse is verantwoordelijk voor het registreren en terugdraaien van alle acties die de gebruiker uitvoert. Bij het aanmaken van een instantie wordt de AnnotationViewer meegestuurd aan het object. Hierdoor kan de UndoManager zelf de acties terugdraaien. Het aanmaken van een actie wordt geregeld vanuit de AnnotationComponent klasse. Als er een actie wordt uitgevoerd in de AnnotationViewer in de component, wordt er een methode aangeroepen die kijkt wat voor soort actie er wordt uitgevoerd. Als het een actie is die ondersteund wordt door de UndoManager, wordt er een registratie gemaakt van de actie. Momenteel ondersteund de UndoManager de acties Create, Delete, Move, Rotate en Resize, met andere woorden de standaardacties die op elk soort annotatie uitgevoerd kan worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Bij het registreren van acties worden een aantal basisgegevens van de annotatie opgeslagen in een UndoItem (zie ook de paragraaf UndoItem). Gegevens na de transformatie worden geplaatst in de AnnotationNew variabele van het UndoItem. Gegevens voor de transformatie komen in de AnnotationOld variabele terecht.

Het terugdraaien van acties verschilt een beetje per soort. Voor elk type actie geldt echter grofweg dat eerst de AnnotationOld variabele wordt opgehaald. Vervolgens wordt de GUID van deze annotatie vergeleken met de GUIDs van alle annotaties die te vinden zijn in de AnnotationViewer. Als de GUIDs gelijk zijn aan elkaar, wordt de annotatie in de AnnotationViewer aangepast met de gegevens van de AnnotationOld variabele.

ColorManager

De ColorManager is verantwoordelijk voor het veranderen van de kleuren in nieuw te plaatsen symboolannotaties of de geselecteerde symboolannotatie. Bij het aanmaken van een nieuwe instantie wordt er gelijk een afbeelding ingeladen. Met de ChangeColor methode wordt de kleur die is meegegeven aan de methode ingevuld op alle pixels die de RGB code 0,174,255 hebben. Dit is de standaardkleur van de oorspronkelijke symboolannotatie bestanden. Doordat elke pixel in een afbeelding gecontroleerd wordt, is het zaak om niet te grote afbeeldingen te gebruiken voor de ColorManager, omdat dit impact kan hebben op de performance van de component.

CursorManager

De CursorManager is verantwoordelijk voor het uiterlijk van de cursor als deze zich in de AnnotationViewer bevindt. Als de gebruiker een annotatie heeft geselecteerd, wordt dit aangegeven door de cursor te veranderen in een symbool zodra deze de AnnotationViewer betreedt. Voor lijnannotaties is dit een potlood, voor tekstannotaties een T en voor symboolannotaties het symbool zelf. Voor symbolen geldt ook dat de cursor de geklozen kleur aangeeft. Daarom bevat de klasse ook weer een instantie van de ColorManager. Let erop dat deze momenteel niet automatisch de kleur overneemt van de ColorManager instantie die in de AnnotationComponent zit.

Structures

AnnotationData

```
Public Structure AnnotationData
    Public Guid As Guid
    Public Location As PointF
    Public Size As SizeF
    Public Rotation As Single
    Public AnnotationType As String
    Public Visible As Boolean
End Structure
```

In deze structure worden alle gegevens van een annotatie genoteerd voor de UndoManager. De volledige annotatie kan niet opgeslagen worden omdat alle versies van de annotatie worden bijgewerkt als deze veranderd. In plaats daarvan worden alleen relevante eigenschappen van de annotatie opgeslagen. Met

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

andere worden, de eigenschappen die ongedaan gemaakt kunnen worden. Als de UndoManager uitgebreid moet worden met extra acties, dienen de gegevens die nodig zijn voor de actie opgenomen te worden in AnnotationData.

Alle eigenschappen die geregistreerd worden, hebben hetzelfde type en dezelfde naam als hun tegenhangers in de annotatie. Location in de annotatie is een PointF, dus de registratie in de AnnotationData wordt ook een PointF genaamd Location. De enige uitzondering hierop is AnnotationType, die alleen de naam van de klasse bevat waaruit de annotatie is opgemaakt (bijvoorbeeld ctlAnnotation.EllipseAnnotation)

UndoItem

```
Public Structure UndoItem
    Public AnnotationOld As AnnotationData
    Public AnnotationNew As AnnotationData
    Public Type As ActionType
End Structure
```

Om een actie terug te kunnen draaien is het nodig om de gegevens van een annotatie op twee momenten op te slaan. Namelijk vlak voor en vlak na de bewerking. Deze structure zorgt ervoor dat we de gegevens van beide

momenten op kunnen slaan. De structure maakt gebruik van twee AnnotationData structures, en een ActionType Enumerator (meer over ActionType is te vinden in de paragraaf ActionType).

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Enumerators

ActionType

```
Public Enum ActionType  
    Create = 0  
    Move = 1  
    Rotate = 2  
    Resize = 3  
    Delete = 4  
End Enum
```

De enumerator ActionType wordt gebruikt om bij het registreren van acties het type vast te stellen. De UndoManager gebruikt dit type vervolgens om de bewerkingen uit te voeren die nodig zijn om een actie terug te draaien. Als er extra soorten acties om terug te draaien worden toegevoegd aan de UndoManager (bijvoorbeeld kleur veranderen), dan dient er ook een extra type aan deze enumerator toegevoegd te worden.

SelectionOptions

Deze enumerator geeft alle mogelijke acties waaruit een gebruiker kan kiezen. Het gaat hier dan voornamelijk om acties die te maken hebben met het plaatsen van annotaties. De enige uitzonderingen hierop zijn Hand, Cancel en None.

```
Public Enum SelectionOptions  
    None = 0  
    Cancel = 1  
    'DragTool  
    Hand = 2  
    'PenAnnotations  
    Ellipse = 3  
    Line = 4  
    Rectangle = 5  
    Freehand = 6  
    'SymbolAnnotations  
    CurvedArrowLeft = 7  
    CurvedArrowRight = 8  
    Cross = 9  
    Minus = 10  
    Plus = 11  
    Arrow = 12  
    S = 13  
    Text = 14  
End Enum
```

- Hand: Deze optie zorgt ervoor dat de AnnotationViewer zo wordt ingesteld, dat de gebruiker er doorheen kan bewegen door te klikken en te slepen met de muis. Deze functie werkt alleen als de afbeelding groter is dan de viewer.
 - Cancel: Deze optie wordt gekozen als de gebruiker op de rechter muisknop klikt. Op dat moment wordt de huidige selectie gestopt, en worden de PictureBoxes weer terug gezet op hun standaardwaarden. Ook de cursor wordt teruggezet op het standaardicoon. Als al deze bewerkingen uitgevoerd zijn, wordt er automatisch doorgeschakeld naar de optie None.
- None: Geeft aan dat er op dit moment niks geselecteerd is en dat de component dus niets hoeft te bouwen of te verplaatsen als de gebruiker klikt.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Overig

Oorspronkelijk functioneel ontwerp (Anne ten Hove)

Inleiding

De bestaande annotatiecomponent dient te worden herschreven in VB.NET. Om deze reden én omdat er meerdere issues zijn met betrekking tot de bestaande versie worden in dit document de functionele eisen van de nieuwe component beschreven.

Doel

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Beschrijving huidige functionaliteit

Definiëren

Bij het definiëren van het Elektronisch medisch dossier kunnen verschillende labels voor annotatieafbeeldingen aangemaakt worden, waarbij d.m.v. de knop “Afbeeldingen” ook de bijbehorende bestandsnamen gedefinieerd worden

Elektronisch medisch dossier definiëren

Codetabel DBC's Dossier instellingen Inlezen lijsten Overgevoeligheden Data dictionary Meta dossier

Specialisme specifiek Codetabel diagnose Codetabel verrichtingen Codetabel complicaties Codetabel problematiek Codetabel medicatie

☐ Vrije tekst ☐ Ziektebeelden
☐ Tekstlabels ☐ Diagnose
☐ Waarde ☐ Verrichting
☐ Aanvink ☐ Complicatie
☐ Ja / Nee ☐ Problematiek
☐ Datum / Tijd ☐ Geluidsfragmenten
☒ Annotatieafbeeldingen ☐ Video
☐ Documenten ☐ Journaal
☐ Afbeeldingen ☐ Verpleegafspraken
☐ Bestanden te archiveren ☐ Spreekuurlijst
☐ Relaties tussen patiënten

Item toevoegen / wijzigen

Labelnaam:
been

Vertrouwelijkheid:
Zichtbaar voor andere specialismes

☒ Label actief Volgordenummer 0

Data dictionary

Voeg toe Wijzig

test
been

Afbeeldingen

Sorteervolgorde
☐ Alfabetisch
☒ Volgorde van invoer

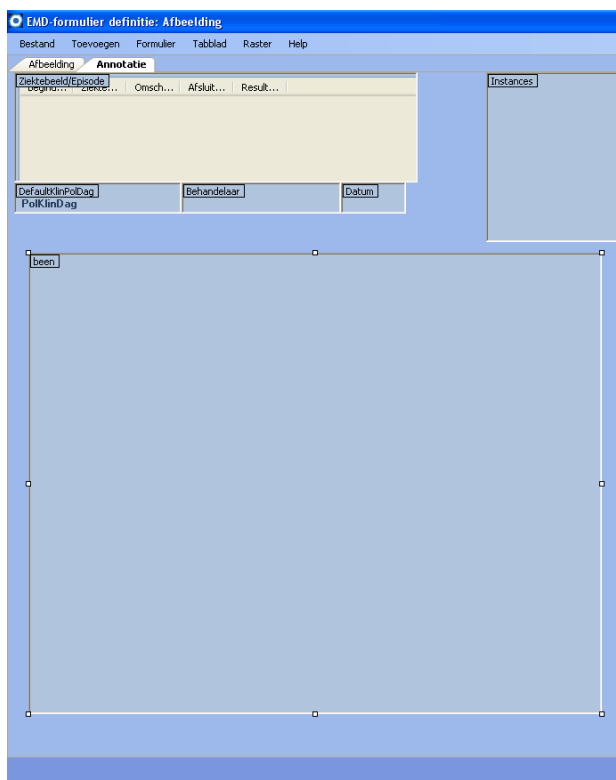
Labels zoeken
< >

OK Info

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

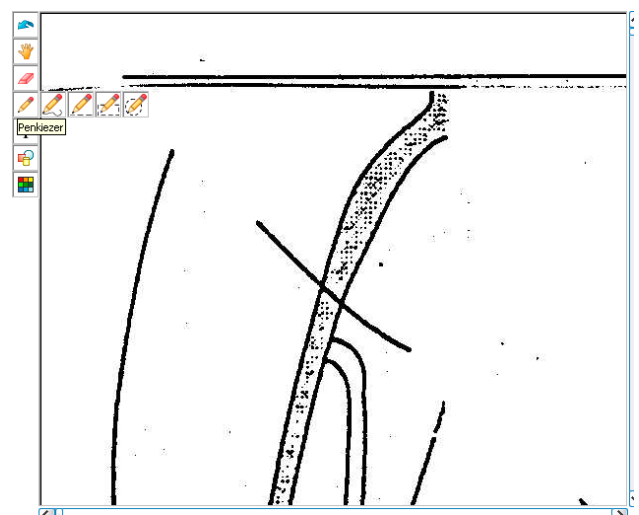


Vervolgens kan bij het definiëren van de EMD-formulieren de annotatiecomponent geplaatst worden, qua grootte en positie aangepast én gekoppeld worden aan een label.



Functionaliteit

Met de huidige annotatiecomponent kunnen aan afbeeldingen annotaties worden



Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

toegevoegd. Het toevoegen van annotaties kan door zelf te tekenen op de geopende afbeelding, door standaard iconen toe te voegen of door tekst toe te voegen aan de afbeelding. Voor al deze opties kunnen verschillende kleuren gebruikt worden.

Menu

Vanuit het menu aan de linkerzijde van de component kunnen de verschillende functies gebruikt worden.



Alle acties kunnen met behulp van de functie ongedaan maken geannuleerd worden.



Met het handje kan de afbeelding verslept worden. Deze functie wordt gebruikt wanneer de afbeelding groter is dan de component zelf, aangezien de component de afbeelding op werkelijke grootte toont.



Toegevoegde annotaties kunnen ook met de gumfunctie verwijderd worden. Wanneer de annotatieafbeelding echter wordt opgeslagen, worden van de annotaties als een geheel met de afbeelding opgeslagen, waardoor op opgeslagen afbeeldingen niet meer gegumd kan worden.



Met de penkiezer kan getekend worden op de afbeelding. Hiervoor kan gekozen worden om vrij te tekenen of een lijn, vierkant of cirkel te tekenen.



Voor het toevoegen van tekst kan op een willekeurige plaats in de afbeelding geklikt worden. Dit punt wordt als startpunt voor de tekst gebruikt.



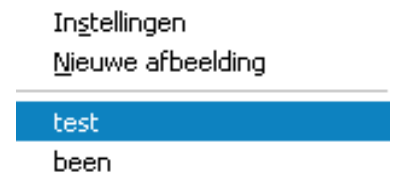
Met de “drag & drop” functie kunnen standaard icoontjes op de afbeelding geplaatst worden.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593



De kleurkiezer kent drie standaardkleuren, blauw, rood en zwart. Door dubbel te klikken op de kleurkiezer kunnen ook andere kleuren gekozen worden. Deze worden vervolgens op positie 4 en 5 in de kleurkiezer getoond. Wanneer echter de applicatie afgesloten wordt, zijn zelf toegevoegde kleuren niet meer standaard beschikbaar.

Onder de rechtermuisknop bevindt zich nog een menu met extra opties. Onder “Instellingen” kunnen de lijnstijl, lijndikte, lettertype en lettergrootte ingesteld worden. Met “Nieuwe afbeelding” kan een andere afbeelding op de eigen computer geopend worden. Onder de horizontale lijn worden de verschillende labels van annotatieafbeeldingen getoond. Hiermee kan direct een ander label in de component geopend worden.



Wensen / Verbeterpunten

Voor de huidige annotatiecomponent zijn een aantal verbeterpunten en gebruikerswensen bekend:

- Toegevoegde annotaties kunnen niet meer verplaatst worden. Wanneer bijvoorbeeld een cirkel getekend wordt, moet deze dus direct precies juist geplaatst zijn.
- Eigen symbolen toevoegen aan annotatiecomponent (ticket 63). Momenteel is het alleen mogelijk om standaardiconen te gebruiken voor de “drag & drop” functie, een aantal specialismen hebben de wens geuit om eigen iconen hieraan toe te voegen.
- - De standaard kleuren mogen niet worden gewijzigd.
- De zelf gekozen kleur moet eventueel kunnen worden opgeslagen (ticket 3005).
- AMC zou graag de mogelijkheid willen hebben om wijzigingen in een annotatiecomponent te kunnen aanbrengen die al bewaard is (ticket 4079). Zoals beschreven onder functionaliteit kunnen eenmaal opgeslagen afbeeldingen niet meer gewijzigd worden.

Ook voor het plaatsen van de component in het Ontwikkeldossier (EMD-formulieren definiëren) is een wens geuit:

- Aanpassen component (in ontwikkeldossier) aan grootte van geselecteerd plaatje/label direct na selectie van annotatielabel. Daarna mag de gebruiker natuurlijk de component kleiner of groter maken. Er wordt dus alleen initieel en bij

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

een nieuwe selectie van een annotatielabel een grootte bepaald. Goed opletten:
grootte component is niet hetzelfde als grootte van het plaatje. Ook bestaat er een
minimum grootte van de component. (ticket 2901)

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

Beschrijving nieuwe functionaliteit

Functionaliteit

De functionaliteit van de .NET annotatiecomponent dient minimaal gelijk te zijn aan de functionaliteit van de huidige component. Zo mogelijk dienen de genoemde wensen en verbeterpunten ook mogelijk gemaakt te worden.

Eisen aan design

Om de overgang naar de nieuwe component voor eindgebruikers te vereenvoudigen, dient het design bij voorkeur niet teveel af te wijken van de oude component. Ook de bestaande iconen uit het menu moeten bij voorkeur hergebruikt worden. Onnodige of ongewenste functies worden niet getoond om de component zo eenvoudig mogelijk te houden voor eindgebruikers.

Naam: M.C.W. Zuidhoek
Studentnummer: 07009593

bijlage: Black ice annotatiecomponent

<http://www.blackice.com/annotation.htm>

