

Technisch Ontwerp

Regisseur

Datum:  
Woensdag 4 mei 2005

Auteur:  
B. Passon

Versie:  
0.5

## Inhoudsopgave

Inleiding .....	3
Functionaliteiten .....	3
Registratie van contact moment .....	3
Opslag van het formulier.....	3
Distributie van de aanvraag .....	4
Architectuur Regisseur.....	4
Ontvangst van DFS formulieren .....	4
Verwerking van flows voor formulieren .....	4
Verwerkingsflow .....	4
Flow Actie: Opslag van formulier .....	5
Flow Actie: Email van formulier .....	5
Opslag van gegevens .....	6
Java Classes.....	6
Regisseur.....	6
FlowFactory.....	6
Flow .....	6
FlowStep.....	7
FlowAction .....	7
Backoffice .....	7
Inloggen .....	7
Hoofdpagina Regisseur .....	8
Configuratie Regisseur.....	8
Zoeken bestaande flows .....	8
Invoercontrole .....	8
Resultaten.....	9
Aanmaken nieuwe flow .....	9
Stap 1: Opgeven Flow Gegevens .....	9
Stap 2: Definieren van een FlowStep .....	9
Stap 3: Definieren van flow acties .....	10
Stap 4: Vervolg van wizard.....	11
Bekijken en bewerken van bestaande flows.....	11
Zoeken ontvangen formulieren .....	11
Invoercontrole .....	11
Resultaten.....	11
Bekijken formulier.....	12
Database .....	12
RegisseurProperties.....	12
DFSDocumenten .....	13
Flows .....	13
FlowSteps .....	14
FlowActions .....	14
FlowActionProperties .....	15
FlowStatusen.....	15
FlowStepStatussen.....	16
FlowActionStatussen .....	17
Opmerkingen .....	17

## **Inleiding**

Na afronding van het project waarbij een ondernemersportaal gerealiseerd is binnen het glazen stadhuis voor de Haagse ICT diensverlening, bleek het noodzakelijk om te zorgen voor een betere 'tracking' en 'tracing' van de aangeboden producten via het portaal. De oplossing voor dit probleem is beschreven in het functioneel ontwerp voor de Regisseur.

In dit document wordt de technische implementatie van deze beoogde regisseur besproken. Als achtergrond voor dit document hebben gediend:

- Functioneel Ontwerp voor de Regisseur.
- Een gesprek met Arjan Otter over de te realiseren Regisseur.

## **Functionaliteiten**

Uit het functioneel ontwerp komen een drietal functionaliteiten naar voren die aangeboden moeten worden door de Regisseur.

1. Registratie van contact moment; dit om het moment tussen aanvraag en registratie van een aanvraag in het processysteem vast te leggen en te bewaken.
2. Opslag document; de opslag van de oorspronkelijke aanvraag is vereist.
3. Distributie; een verzoek moet naar meerdere instanties verstuurd kunnen worden.

Om de bovenstaande punten te kunnen realiseren, zal de Regisseur het eerste aanspreekpunt zijn voor alle aanvragen van het ondernemersportaal. De regisseur zal dan de verdere distributie op zich nemen.

### **Registratie van contact moment**

Op dit moment is het zo dat als de aanvraag door het Digitaal Formulieren Systeem verstuurd is, is er geen enkele mogelijkheid om te achterhalen wanneer het verstuurd is en waar het formulier gebleven is. De Regisseur zal in de Klant Gegevens Bank (KGB) een contact moment opnemen welke aangeeft wanneer het formulier ontvangen en verwerkt is door de Regisseur.

### **Opslag van het formulier**

Als een formulier door het Digitaal Formulieren Systeem verstuurd is, is het onmogelijk om het originele formulier nog ter inzage te krijgen. De regisseur gaat deze functionaliteit aanbieden door de originele aanvraag op te slaan.

In eerste instantie zal de opslag extern plaats vinden in de KGB. Er zijn ontwikkelingen in gang gezet voor een betere opslag locatie, het DMS.

Echter deze is nog niet gerealiseerd, waardoor de Regisseur alle aanvragen voorlopig zal opslaan in de KGB. De externe opslag kan gebruikt worden om de originele aanvraag zichtbaar te maken voor de aanvrager.

Naast de externe opslag maakt de Regisseur ook gebruik van een interne opslag zodat als er eventueel fouten optreden bij de externe opslag er nog steeds een kopie van het originele formulier aanwezig is.

### **Distributie van de aanvraag**

Het Digitaal Formulieren Systeem is op dit moment niet in staat een aanvraag bij meerdere instanties te bezorgen. Met de verandering in de flow van aanvragen naar de situatie dat de Regisseur alle aanvragen in eerste instantie zal opvangen, is het mogelijk de distributie zo op te zetten dat er meerdere instanties voorzien kunnen worden in hun informatie behoefte met betrekking tot aanvragen die gedaan zijn.

### **Architectuur Regisseur**

De architectuur van de Regisseur zal bestaan uit een stappeling van drie lagen, waarbij de lagen van boven naar beneden de volgende functies vervullen:

- Ontvangst van DFS formulieren.
- Verwerking van flows voor formulieren.
- Opslag van gegevens.

### **Ontvangst van DFS formulieren**

De formulieren zullen vanuit de DFS middels een SOAP aanroep bij de Regisseur aankomen. De SOAP interface is vastgelegd middels een WSDL die door de DFS beschikbaar gesteld is.

### **Verwerking van flows voor formulieren**

Elk formulier dat via de SOAP interface bij de Regisseur binnen komt moet verwerkt worden. Een formulier dat bij de Regisseur aan komt is een in XML opgemaakt document dat beschrijft welke informatie er opgegeven is door de aanvrager.

Aan de hand van dit XML document bepaalt de Regisseur welke flow er gekozen moet worden voor de verwerking van het document. Elk type formulier dat bekend is binnen de Regisseur heeft een gedefinieerde flow.

#### *Verwerkingsflow*

Zoals als hierboven al is opgemerkt is er binnen de Regisseur voor elke type formulier een flow gedefinieerd waarin is opgeslagen welke verwerking er plaats moet vinden. Een flow is een reeks gekoppelde stappen die elke weer één of meer acties kan bevatten. De stappen van

een flow worden achtereenvolgens verwerkt met als voorwaarde dat elke voorgaande stap correct afgerond moet zijn.

Vooralsnog is er geen terugkoppeling vanuit andere systemen naar de Regisseur, waardoor op dit moment flows alleen zullen bestaan uit één stap met aantal acties die uitgevoerd moeten worden. In de toekomst kan het zo zijn dat er wel terugkoppeling vanuit andere instanties gerealiseerd wordt. De Regisseur kan daar op inspringen door meer stappen toe te kennen aan een flow en de Regisseur uit te breiden met meer acties als dat nodig mocht zijn.

Elke stap in een flow bestaat uit een aantal acties die achter elkaar afgewerkt moeten worden. Er zijn op dit moment een tweetal acties gedefinieerd die nodig zijn om de functionaliteit beschreven in het functioneel ontwerp te realiseren:

- Opslag van formulier
- Email van formulier

#### *Flow Actie: Opslag van formulier*

Deze actie zorgt er voor dat een formulier wordt opgeslagen zodat deze later nog bekeken kan worden. Vooralsnog wordt deze actie altijd standaard gedaan omdat het wenselijk is dat formulieren ten alle tijden bekeken moeten kunnen worden. Deze actie kan middels een HTTP POST of een SOAP aanroep naar de KGB uitgevoerd worden. Deze actie maakt gebruik van de volgende properties voor de configuratie:

- store.url – de url waar de aanroep op moet gaan plaatsvinden
- store.type – het type van de aanroep welke gebruikt moet worden; post/soap

#### *Flow Actie: Email van formulier*

Deze actie zorgt er voor dat een formulier wordt verstuurd naar de ontvangende partij. Deze actie maakt gebruik van de standaard email server instellingen voor de Regisseur. Middels SMTP wordt het formulier verstuurd naar de ontvanger. Deze actie maakt gebruik van de volgende properties voor de configuratie:

- email.from – het e-mail adres dat als versturende partij gebruikt moet worden.
- email.rcpt – het e-mail adres van de ontvangende partij.
- email.subject – het onderwerp van het te versturen e-mail bericht.
- email.body – de tekst voor het te versturen e-mail bericht.
- email.attachment.plaintext – true/false instelling om het bericht te vertalen van XML naar plaintext leesbare opmaak.

De aanvraag zal als een attachment meegestuurd worden. Deze attachment kan in twee formaten verstuurd worden. Plain text en XML. Het formaat dat gebruikt moet worden is in te stellen via de properties.

Het XML formaat is de aanvraag zoals hij door de DFS bij de Regisseur is aangeboden. Het plain text formaat is een opsomming van de name/value pairs voor de verschillende delen van een aanvraag. De vertaling van het XML bericht naar plain text zal plaatsvinden middels een XSL translatie.

## **Opslag van gegevens**

De gegevens binnen de Regisseur worden opgeslagen in de database van de Regisseur. Deze database is te bewerken middels Java Entity Beans met Container Managed Persistence. Er wordt gebruik gemaakt van een Oracle database. Voor de specificatie van de database verwijs ik naar hoofdstuk 0.

## **Java Classes**

Voor de implementatie in Java zijn een aantal classes nodig, we bespreken hier de belangrijkste classes en geven aan wat hun relatie is tot het functioneren van de Regisseur. Niet alle classes worden hier dus vermeld.

### **Regisseur**

Deze class is centraal binnen de Regisseur. Hij regelt de ontvangst van formulieren en start de verwerking van formulieren.

### **FlowFactory**

De class wordt gebruikt voor het aanmaken en ophalen van Flow objecten. Eveneens is dit het entypoint voor FlowStatus objecten. We maken gebruik van een Factory pattern om er voor te zorgen dat het in de toekomst mogelijk blijft om snel een andere manier van opslag onder de regisseur te hangen.

### **Flow**

Deze class is een interface die alle functionaliteiten van een Flow definieert. De volgende methoden zijn gedefinieerd in de interface:

- public String getName()
- public Date getDateCreated()
- public void execute()  
throws FlowException
- public void executeFlowStep( Integer nr )  
throws FlowException
- public FlowStep getFlowStep( Integer nr )  
throws FlowException
- public Integer getFlowStepCount()
- public List getFlowSteps()

## **FlowStep**

Deze class is een interface die all functionaliteiten van een FlowStep definieerd. De volgende methoden zijn gedefinieerd in de interface:

- public String getName()
- public Date getDateCreated()
- public Date getDateLastExecuted()
- public void execute()  
throws FlowException
- public void executeFlowAction( Integer nr )  
throws FlowException
- public Integer getFlowActionCount()
- public List getFlowActions()

## **FlowAction**

Deze class is een interface die alle functionaliteiten van een FlowAction definieerd. De volgende methoden zijn gedefinieerd in de interface:

- public String getName()
- public String getType()
- public Date getDateCreated()
- public Date getDateLastExecuted()
- public void execute()  
throws FlowException
- public Properties getProperties()  
throws FlowException
- public List getPropertyNames()

## **Backoffice**

De backoffice van de Regisseur biedt alle functionaliteit die nodig is om de Regisseur te configureren en om de verschillende flows voor inkomende formulieren te definiëren. Daarnaast kan er naar ontvangen formulieren gezocht worden en de bijbehorende flows bekeken en beïnvloed worden.

De layout van de backoffice zal zo veel mogelijk gelijk gehouden worden aan de layout van de huidige portal. De backoffice wordt gerealiseerd met behulp van JSP pagina's.

## **Inloggen**

Het beheer over wie toegang krijgt tot de backoffice van de Regisseur wordt volledig gedaan door het CDS systeem. Er wordt een koppeling gerealiseerd die de gebruiker die toegang vraagt autoriseert tegen de CDS. Als de autorisatie positief is, zal de hoofdpagina van de backoffice

van de Regisseur te zien zijn. Als de autorisatie negatief is, zal er een pagina getoond worden met een melding dat er geen toegang tot het systeem verleend wordt.

## Hoofdpagina Regisseur

De hoofdpagina van de Regisseur bestaat uit de een bericht waar de gebruiker uitleg vindt over de backoffice van de Regisseur en wordt hem de mogelijkheid gegeven te kiezen uit de onderstaande opties voor het beheer van de Regisseur.

- Configuratie van de Regisseur.
- Zoeken naar reeds bestaande flows voor formulieren.
- Aanmaken van nieuwe flows voor onbekende formulieren.
- Zoeken van ontvangen formulieren.

## Configuratie Regisseur

Het configuratie scherm van de Regisseur stelt de gebruiker in staat de instellingen die de Regisseur gebruikt voor het uitvoeren van de gestelde taken. Het scherm zal bestaan uit een lijst van properties die ingesteld kunnen worden.

## Zoeken bestaande flows

Het zoekscherm voor bestaande flows biedt alle functionaliteit die nodig is voor het zoeken naar al eerder gedefinieerde flows voor formulieren. Er kan gezocht worden op de volgende parameters:

- Formulier Code
- Naam van de Flow.
- Datumbereik waartussen de Flow aangemaakt is.

De zoekparameters wordt automatisch gecombineerd middels een AND tot een totale zoekparameter. Er wordt case-insensitive gezocht op de naam van een formulier en tevens wordt de zoekparameter naam gezien als een substring voor resultaten. Wordt er gezocht op de waarde '**lier**', dan zal als resultaat alle formulieren getoond worden met de waarde '**lier**' in hun naam hebben staan.

### *Invoercontrole*

De opgegeven zoekparameters worden gecontroleerd op correctheid. De volgende regels worden toegepast.

- Voor zoekparameters van het type **String** geldt een minimum lengte van drie tekens.
- Datum parameters moeten voldoen aan het volgende formaat: dd-MM-jjjj

## *Resultaten*

De resultaten van een zoekopdracht worden getoont in een tabel-overzicht dat paginageerbaar is, met een maximum van 25 resultaten per scherm.

Elke regel van de tabel is aan te klikken om gevonden resultaat te bewerken of te bekijken.

## **Aanmaken nieuwe flow**

Het formulier voor het aanmaken van een nieuwe flow biedt alle functionaliteit die nodig is om een nieuwe flow aan te maken. Het aanmaken van een nieuwe flow zal plaatsvinden in de vorm van een wizard die uit een viertal stappen zal bestaan, waarvan de laatste drie stappen te herhalen zijn.

### *Stap 1: Opgeven Flow Gegevens*

De eerste stap van de wizard zal bestaan uit het opgeven van de informatie die nodig is voor de identificatie van de flow. De gebruiker moet de volgende gegevens opgeven. Alle gevraagde informatie is verplicht.

- De naam van de flow.
- Het document type waar de flow bij hoort.

Als de gebruiker deze informatie opgegeven heeft, kan gebruik maken van de knop '**verder**' om door te gaan naar de volgende stap in de wizard.

Als de gebruiker naar de volgende stap wil gaan worden de gegevens die hij heeft opgegeven gecontroleerd. De volgende controles vinden plaats:

- De naam van de flow bestaat uit alfanumerieke tekens.
- De naam van de flow bestaat uit minimaal 3 en maximaal 100 tekens.
- De naam van de flow is nog niet eerder gebruikt.
- Er bestaat nog geen flow voor het opgegeven document type.
- Het document type bestaat uit minimaal 3 tekens en maximaal 30 tekens.

Als één van de bovenstaande controles faalt, zal de gebruiker hier op attent gemaakt worden en moeten de gevonden fouten opgelost worden alvorens het systeem het verder gaan met stap twee van de wizard toestaat.

### *Stap 2: Definieren van een FlowStep*

De tweede stap van de wizard stelt de gebruiker in staat om een stap binnen een flow aan te maken. De gebruiker moet de volgende gegevens opgeven. Alle gevraagde informatie is verplicht:

- De naam van de flow step.

Als de gebruiker deze informatie opgegeven heeft, kan gebruik maken van de knop '**verder**' om door te gaan naar de volgende stap in de wizard.

Als de gebruiker naar de volgende stap wil gaan worden de gegevens die hij heeft opgegeven gecontroleerd. De volgende controles vinden plaats:

- De naam van de flow step bestaat uit alfanumerieke tekens.
- De naam van de flow step bestaat uit minimaal 3 en maximaal 100 tekens.
- De naam van de flow step is nog niet eerder gebruikt binnen de huidige flow.

Als één van de bovenstaande controles faalt, zal de gebruiker hier op attent gemaakt worden en moeten de gevonden fouten opgelost worden alvorens het systeem het verder gaan met stap drie van de wizard toestaat.

### *Stap 3: Definieren van flow acties*

De derde stap stelt de gebruiker in staat om acties te definiëren voor de zojuist aangemaakte flow step. Er moet minimaal één actie gedefinieerd worden voor een flow step. Er is geen bovengrens voor het aantal acties dat een gebruiker aan mag maken. Voor een te definiëren actie moet de gebruiker de volgende informatie opgeven. Alle informatie is verplicht.

- De naam van de actie.
- Het type actie selecteren uit de lijst met mogelijke types.
- Opgeven van properties die nodig zijn voor het geselecteerde actie type.

Als de gebruiker deze informatie opgegeven heeft, kan gebruik maken van de knop '**verder**' om door te gaan naar de volgende stap in de wizard.

Als de gebruiker naar de volgende stap wil gaan worden de gegevens die hij heeft opgegeven gecontroleerd. De volgende controles vinden plaats:

- De naam van de flow action bestaat uit alfanumerieke tekens.
- De naam van de flow action bestaat uit minimaal 3 en maximaal 100 tekens.
- De naam van de flow action is nog niet eerder gebruikt binnen de huidige flow step.
- Er is een type actie geselecteerd.
- Alle properties voor het geselecteerde type zijn opgegeven.

Als één van de bovenstaande controles faalt, zal de gebruiker hier op attent gemaakt worden en moeten de gevonden fouten opgelost worden alvorens het systeem het verder gaan met stap drie van de wizard toestaat.

#### *Stap 4: Vervolg van wizard*

De vierde stap stelt de gebruiker in staat aan te geven wat hoe het vervolg van de wizard moet zijn. De gebruiker kan hier kiezen uit de volgende mogelijkheden:

- Wizard afronden en alles opslaan.
- Een nieuwe actie toevoegen aan de huidige stap
- Huidige stap afsluiten en een nieuwe stap aanmaken

Afhankelijk van de gekozen optie zal de gebruiker of naar een van de pagina's binnen de wizard gaan of terug gaan naar het hoofdmenu.

### **Bekijken en bewerken van bestaande flows**

Het bekijken van een gekozen flow vindt plaats op één pagina. Deze pagina toont alle informatie die bekend is over een flow. In de vorm van tabellen worden alle stappen binnen een flow getoond. Binnen deze tabellen wordt alle informatie die beschikbaar is over de gedefinieerde acties zichtbaar gemaakt.

Achter de verschillende regels van de tabellen wordt de mogelijkheid geboden om de gegevens van de regel te bewerken. Ook kunnen hier flows actief en in-actief gemaakt worden.

Voor alle informatie die bewerkt wordt gelden exact dezelfde controles als voor de invoer van nieuwe flows.

### **Zoeken ontvangen formulieren**

Het zoeken naar ontvangen formulieren biedt de mogelijkheid om te zoeken in alle formulieren die ontvangen zijn. Er kan gezocht worden op de volgende zoek parameters.

- Het type van het ontvangen formulier.
- Datumbereik waarbinnen het formulier ontvangen is.

De zoekparameters wordt automatisch gecombineerd middels een AND tot een totale zoekparameter.

#### *Invoercontrole*

De opgegeven zoekparameters worden gecontroleerd op correctheid. De volgende regels worden toegepast.

- Voor zoekparameters van het type **String** geldt een minimum lengte van drie tekens.
- Datum parameters moeten voldoen aan het volgende formaat: dd-MM-jjjj

#### *Resultaten*

De resultaten van een zoekopdracht worden getoont in een tabel-

overzicht dat paginageerbaar is, met een maximum van 25 resultaten per scherm.

Elke regel van de tabel is aan te klikken om gevonden resultaat te bekijken en eventueel mogelijke acties te ondernemen.

### **Bekijken formulier**

Het bekijken van een gekozen formulier vindt plaats op één pagina. Deze pagina toont alle informatie die bekend is over het formulier en de gekoppelde flow. In de vorm van tabellen wordt alle bekende informatie getoont.

De pagina biedt de mogelijkheid om het originele formulier te downloaden. Naast deze functionaliteit kan er ook ingegrepen worden in de verwerkingsflow van het formulier. Zowel flow steps als flow actions kunnen opnieuw uitgevoerd worden middels deze pagina.

### **Database**

De database van de Regisseur bevat alle informatie die nodig is om de Regisseur te laten functioneren zoals beschreven in het Functioneel Ontwerp. De volgende tabellen zijn gedefinieerd:

- RegisseurProperties; bevat alle instellingen voor de regisseur.
- DFSDocumenten; bevat alle originele ontvangen formulieren.
- Flows; bevat alle beschikbare flows voor afhandeling van formulieren.
- FlowSteps; bevat alle beschikbare flow steps voor de gedefinieerde flows.
- FlowActions; bevat alle beschikbare flow actions voor de gedefinieerde flow steps.
- FlowActionProperties; bevat alle instellingen voor een flow action.
- FlowStatussen; bevat de status informatie voor een lopende flow.
- FlowStepStatussen; bevat de status informatie voor een mogelijke flow step van een lopende flow.
- FlowActionStatussen; bevat de status informatie voor de mogelijke flow acties van een mogelijke flow step voor een lopende flow.

De verschillende tabellen zijn hieronder verder uitgewerkt en toegelicht.

### **RegisseurProperties**

De regisseur properties worden opgeslagen in de RegisseurProperties tabel. Deze tabel bevat alle instellingen die nodig zijn om de regisseur te laten functioneren. De structuur van de tabel is hieronder weergegeven.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
property	varchar(100)	ja	ja
value	varchar(255)	ja	nee

De naam van de property moet uniek zijn. Dit is de naam van een instelling voor de regisseur. De value is de waarde van de instelling. Dit is een tekstuele invulling

### **DFSDocumenten**

De binnenkomende documenten worden altijd opgeslagen in deze tabel. Ze worden misschien doormiddel van acties ook nog extern opgeslagen, maar we willen altijd controle hebben over eventueel uit te voeren acties. De structuur van de tabel is hieronder weergegeven.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
document_guid	varchar(50)	ja	nee
document_type_id	varchar(30)	ja	nee
date_received	date and time	ja	nee
content	text	ja	nee

De document\_guid moet uniek zijn, omdat elk formulier door middel van deze guid geïdentificeerd wordt binnen externe systemen. De waarde voor deze guid is te herleiden aan de hand van het ontvangen document. De document type id is de identificatie van het type formulier. Deze wordt gebruikt voor de identificatie van de flow die voor verwerking van het formulier gebruikt moet worden. De date received is de datum en tijd waarop het document ontvangen is. De content is hierin het document zoals het origineel ontvangen is door de regisseur.

Naast de guid die uniek moet zijn, wordt er intern binnen de regisseur nog gebruik gemaakt van een uniek id-veld welke automatisch gevuld wordt met behulp van een sequence.

### **Flows**

Voor elke flow die aangemaakt wordt voor een unieke document type id is er in deze tabel een record aanwezig. Dit record is de ingang naar de configuratie van de flow steps en de flow actions.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
name	varchar(100)	ja	nee
document_type_id	varchar(30)	ja	nee
date_created	date time	ja	nee
active	varchar(1) [y/n]	ja	nee

De name is de naam van de flow zoals de beheerder hem heeft gedefinieerd. De document type id is de document type id waarvoor deze flow gebruikt moet worden. De document type id is uniek binnen deze tabel. Er kan maximaal één flow per document type gedefinieerd zijn. De date created is de datum waarop deze flow gedefinieerd is. Active wil zeggen of de flow actief is en dus beschikbaar voor de Regisseur om te gebruiken voor verwerking van formulieren.

Het id-veld is de unieke identificatie van een flow binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowSteps**

Elke stap binnen een flow is geregistreerd in deze tabel. Flow steps zijn gekoppeld aan de flow waar ze bij horen.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_id	integer	ja	nee
name	varchar(100)	ja	nee
step_number	integer	ja	nee

De flow id is de identificatie van de flow uit de Flows tabel waar deze stap bij hoort. De name is de naam van de stap binnen de flow. Het step number is het nummer van de stap binnen de flow. De nummering van stappen begint bij 1 en loopt incrementeel op naar n.

Het id-veld is de unieke identificatie van een flow step binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowActions**

Elke actie binnen een flow step is geregistreerd binnen deze tabel. Flow

actions zijn gekoppeld aan de flow step waar ze bij horen.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_id	integer	ja	nee
flow_step_id	integer	ja	nee
name	varchar(100)	ja	nee
class_name	varchar(255)	ja	nee

De flow id is de flow waar deze actie bij hoort. De flow step id is de flow step waar deze actie deel van uit maakt. De name is de naam van de actie zoals deze gespecificeerd is door de beheerder. De class name is de FQCN van de Java class die deze actie implementeerd.

Het id-veld is de unieke identificatie van een flow action binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowActionProperties**

Alle instellingen voor acties die gebruikt worden binnen de regisseur staan opgeslagen in deze tabel.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_action_id	integer	ja	nee
property	varchar(100)	ja	nee
value	varchar(255)	ja	nee

De flow action id is de flow action waar deze instelling voor gebruikt moet worden. De property is de unieke property naam voor een flow action instelling. De value is de waarde van de instelling.

Het id-veld is de unieke identificatie van een flow action property binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowStatuseren**

De status van een flow wordt bijgehouden in de tabel flow statussen. Voor elk document dat verwerkt wordt binnen een flow is er een status record

in deze tabel te vinden.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_id	integer	ja	nee
document_id	integer	ja	nee
date_created	date time	ja	nee
date_changed	date time	ja	nee
current_step	integer	ja	nee

Het flow id veld is de identificatie van de flow die bij het betreffende status record hoort. De document id is de identificatie van het document uit de DFSDocuments tabel welke door deze flow verwerkt wordt. De date created is de datum waarop het flow status record is aangemaakt en dus de flow in gang gezet is. De date changed is de datum waarop voor het laatst dit record bijgewerkt is. De current step is de stap in de flow waar de regisseur gebleven is met de afhandeling.

Het id-veld is de unieke identificatie van een flow status binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowStepStatussen**

De status van een flow step wordt in deze tabel bijgehouden. Voor elke step binnen een flow is er een record te vinden in deze tabel.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_step_id	integer	ja	nee
date_created	date time	ja	nee
date_last_execute	date time	ja	nee
times_executed	integer	ja	nee

De flow step id is de identificatie van de flow step waar dit status record bij hoort. De date created is de datum waarop dit record is aangemaakt en tevens de datum waarop de flow step gestart is. De date last execute is de datum waarop de flow step voor het laatst uitgevoerd is. De times

executed geeft het aantal keren van executie van deze flow step aan sinds de datum dat hij is aangemaakt.

Het id-veld is de unieke identificatie van een flow step status binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **FlowActionStatussen**

De status van een flow action wordt in deze tabel bijgehouden. Voor elke actie binnen flows wordt er een record bijgehouden in deze tabel.

<b>Naam</b>	<b>Type</b>	<b>Verplicht</b>	<b>PK</b>
id	integer	ja	ja
flow_action_id	integer	ja	nee
date_created	date time	ja	nee
date_last_execute	date time	ja	nee
times_executed	integer	ja	nee

De flow action id is de identificatie van de flow action waar dit status record bij hoort. De date created is de datum waarop dit record is aangemaakt en tevens de datum waarop de flow action gestart is. De date last execute is de datum waarop de flow action voor het laatst uitgevoerd is. De times executed geeft het aantal keren van executie van deze flow action aan sinds de datum dat hij is aangemaakt.

Het id-veld is de unieke identificatie van een flow action status binnen de regisseur. Dit veld krijgt automatisch een waarde middels een sequence.

### **Opmerkingen**

In dit deel van het functioneel ontwerp hebben we een aantal opmerkingen en punten opgenomen die nog open blijven voor onderzoek of discussie.

- Authenticatie van hosts die data sturen naar de Regisseur. Het is vooralsnog onduidelijk hoe dit aan te pakken. SOAP zoals aanwezig bij de DFS biedt geen mogelijkheid hier iets mee te doen. Misschien dat de Novell eXtend applicatie server client/server authenticatie ondersteund.
- Berichten worden niet ontleed, maar in zijn geheel als XML of plain text doorgezeten naar de ontvangende partijen.
- Er is nog geen duidelijkheid over wat er moet gebeuren met formulieren waarvoor er geen verwerkende flow gedefinieerd is.

- Het kan zijn dat de classes en interfaces uiteindelijk meer methoden bevatten dan nu het geval is om alles te laten functioneren.