



Technisch Ontwerp

Website WOSI

Document historie

Versie(s)

Versie	Datum	Status	Omschrijving / wijzigingen
0.1	20 nov 2008	Concept	Initiële versie
0.2	03 jan 2008	Revised	Nagekeken / verbeterde versie

Auteur(s)

Versie	Auteur	Email
0.1	Ruud Jungbacker	ruudjungbacker@gmail.com
0.2	Jan Willem de Birk	Jan.willem.de.birk@hva.nl

Controle en Goedkeuring

Versie	Datum	Naam	Functie	Goedgekeurd

Versie	Datum	Naam	Functie	Goedgekeurd

Inhoudsopgave

Technisch Ontwerp	1
Document historie	2
Versie(s)	
Auteur(s)	
Controle en Goedkeuring	
Inhoudsopgave	1
Inleiding	1
Database	2
Ontwerp	
Entiteiten	
<i>User.....</i>	<i>2</i>
<i>Agenda.....</i>	<i>2</i>
<i>News.....</i>	<i>2</i>
<i>Page.....</i>	<i>3</i>
<i>File.....</i>	<i>3</i>
Applicatiestructuur	4
Data Acces Object laag	
Servicelaag	
Domeinlaag	
Frontend	
Diagrammen	5
De algemene structuur	

Overige diagrammen

Gebruikte Technieken

6

Spring

Spring LDAP

JSTL

Hibernate

Inleiding

Dit document zal verder gaan waar het functioneel ontwerp is gebleven. In dit document zal de technische realisatie van het systeem worden besproken.

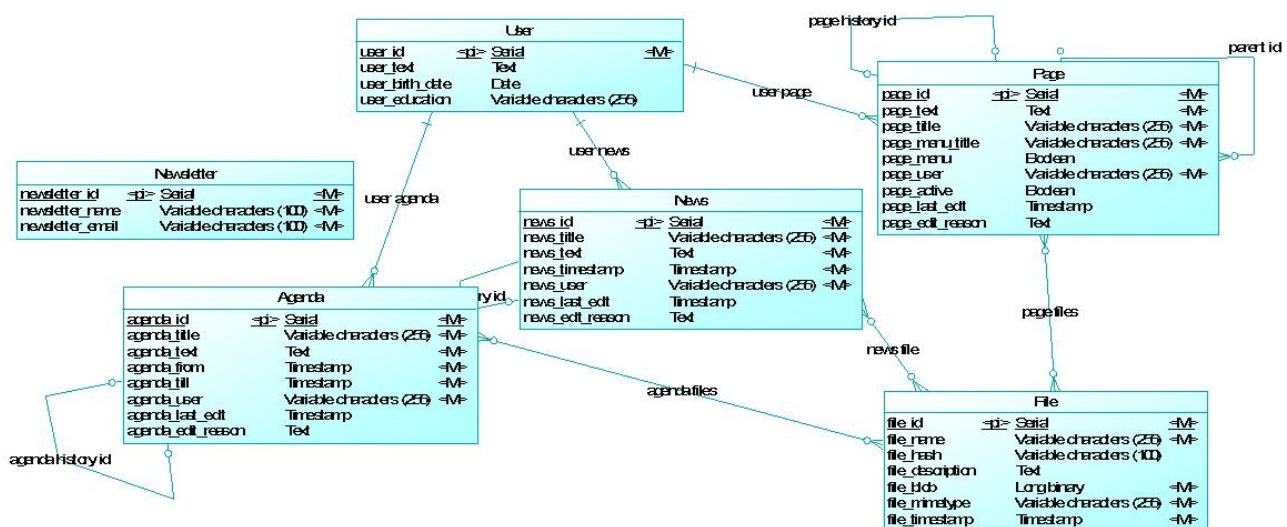
We zullen de technieken welke gebruikt worden beschrijven en een overzicht te geven over hoe de applicatie opgebouwd is.

Database

Voor dit project gebruiken we een MySQL 5.0.x database. In deze database zullen we alle gegevens opslaan welke nodig zijn voor de website. De teksten, maar ook bijlagen zoals tekstdocumenten, afbeeldingen en andere bestanden.

Ontwerp

In het ontwerp van de database is rekening gehouden met het feit dat deze zeer flexibel moet zijn. Er moeten verschillende crosslinks worden uitgevoerd en hiervoor moet de database geschikt zijn. Ook zal aan veel events zoals agenda items en project beschrijven een file kunnen worden gehangen.



In de database is te zien dat verschillende entiteiten zijn gekoppeld aan een user (gebruiker). Op deze manier kan worden bijgehouden wie, wat en hoe er wijzigingen aan de site zijn aangebracht. Hiervan wordt van verschillende entiteiten (agenda, news en page) dan ook een geschiedenis bijgehouden.

Entiteiten

De verschillende entiteiten zijn:

User

- Als iemand inlogt worden de naam en het ID opgeslagen vanuit de LDAP database van het project. Ook kunnen hier profielen worden aangemaakt doormiddel van een tekst welke een gebruiker zelf kan invullen. Hierdoor kan een persoonlijke pagina over deze persoon worden aangemaakt.

Agenda

- Hierin zullen de verschillende agenda punten worden opgeslagen. Hierbij zal ook een geschiedenis worden bijgehouden zodat eventuele verkeerde wijzigingen ook kunnen worden teruggedraaid.

News

- Hier zullen de verschillende nieuwsitems worden opgeslagen. Hierdoor kunnen de recentste nieuwsitems worden weergegeven op de frontpagina van de website. Verder wordt ook hiervan de geschiedenis bijgehouden zodat foutieve wijzigingen kunnen worden teruggedraaid.

Page

- *Hierin kunnen verschillende pagina's worden opgeslagen. De hele inhoud van de pagina wordt hierin opgeslagen zodat er een volledige vrijheid is hoe een pagina wordt ingedeeld. Verder wordt hier ook weer een geschiedenis van bijgehouden.*

File

- *Hierin kunnen bestanden worden opgeslagen welke dan gekoppeld kunnen worden aan een nieuwsitem, agenda item of een pagina. Een file kan aan een of meerdere van deze entiteiten worden gekoppeld.*

Applicatiestructuur

Om de website zo overzichtelijk en onderhoudbaar mogelijk te maken is het project opgedeeld in vier lagen, elke laag heeft zijn eigen verantwoordelijkheid. Deze verschillende lagen worden hier beschreven.

Data Acces Object laag

De Data Access Object (DAO) laag bevat methodes om domein-objecten uit een databron te laden, zoeken, veranderen en verwijderen. Voor elk DAO object bestaat een interface welke alle functies beschrijft, deze interface maakt het mogelijk om de gehele DAO laag te vervangen voor bijvoorbeeld een andere ORM methode. Binnen deze laag wordt enkel data doorgespeeld naar een databron, of de servicelaag; er mag dus geen data aangepast worden. De DAO laag mag nooit direct gebruikt worden vanuit de frontend; al het dataverkeer moet via de servicelaag gaan, zodat eventuele business rules niet genegeerd worden.

Servicelaag

De Servicelaag ligt boven de DAO laag, deze bevat functies die business rules afdwingen en andere functionaliteiten mogelijk maken. Deze laag bevat in ieder geval de functies die ook in de DAO laag voorkomen, maar er kunnen ook extra functies in komen. Omdat de servicelaag boven de DAO laag ligt kunnen er bewerkingen op de data gedaan worden voordat het via de DAO laag opgeslagen wordt in de gekoppelde databron (denk hierbij aan bijvoorbeeld aanmaakdatum van objecten) Bij deze laag is hetzelfde principe toegepast als bij de DAO laag, voor elke service is er een interface, dit maakt het mogelijk om de hele servicelaag te vervangen; en zo dus een andere set business rules toe te passen.

Domeinlaag

De domeinlaag bevat de objectstructuur die nodig is om de data nodig voor het gehele website op te slaan. Dit zijn de objecten die via de servicelaag, daarna via de DAO laag opgeslagen en opgevraagd worden in de database. Denk hierbij bijvoorbeeld aan een klant met alle bijbehorende gegevens; de objecten in de domeinlaag bevatten dus dezelfde velden als de bijbehorende tabellen in het datamodel.

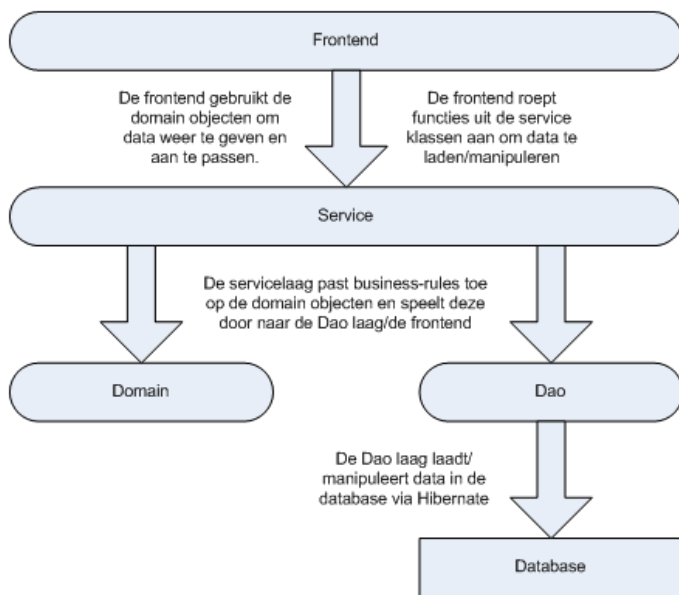
Frontend

De frontend laag bestaat uit de JSP pagina's, Servlets, javascript bestanden etc. Dit is de laag welke de eindgebruiker krijgt te zien. Hier wordt de informatie uit het systeem op een logische manier weergegeven en kan een gebruiker deze informatie wijzigen.

Diagrammen

Er volgen nu een aantal diagrammen die de structuur van de applicatie schematisch weergeven. Deze diagrammen zijn puur om een beter inzicht in de structuur te geven, en bevatten daarom niet alle methodes en velden die in de echte situatie wel bestaan. Deze keuze hebben we ook gemaakt om de leesbaarheid en het overzicht te behouden.

De algemene structuur



Overige diagrammen

Hier zullen de diagrammen komen te staan over het domein model, de service laag en de DAO laag. Deze zijn op dit moment nog niet beschikbaar aangezien er hier nog vele wijzigingen in zullen komen. Deze zullen dus gedocumenteerd worden zodra de applicatie gereed is.

<< Diagrammen nodig >>

Gebruikte Technieken

Spring

Het Spring Framework, of kortweg Spring is een open source framework gericht op ontwikkeling van software in het Javaplatform. Het raamwerk combineert een aantal API's en ideeën op een manier die een alternatief biedt voor de standaard manier van het ontwikkelen. Dankzij diverse uitbreiding wordt het vooral gebruikt als alternatief of uitbreiding van technologieën uit J2EE platform.

Dit framework wordt ook binnen de WOSI applicaties gebruikt, en zorgt er voor dat elke WOSI ontwikkelaar kan verder ontwikkelen aan de applicatie.

Spring LDAP

Is een library die het makkelijker maakt om LDAP applicaties te ontwikkelen in Java, in samenwerking met het Spring framework

JSTL

JSTL staat voor JavaServer Pages Standard Tag Library (JSTL) en is een component van het Java EE Web applicatie ontwikkel platform. Het is een uitbreiding op JSP, en kent eenvoudige 'tags' toe aan taken zoals XML data processing, condities, loops en internationalization. Het levert een eenvoudige manier om logica met de view te combineren.

Hibernate

Hibernate is een Object/Relational Mapping (ORM) oplossing voor Java. Het is een gratis en open source framework. Hibernate werd door een internationaal team van Java ontwikkelaars gemaakt. Het is een eenvoudig te gebruiken framework voor het koppelen van een object georiënteerd domeinmodel aan een traditionele relationele database.

Hibernate zorgt voor de koppeling van Java klassen aan databasetabellen (en voor het koppelen van Java datatypes aan SQL datatypes), maar zorgt tevens voor de ophaalfuncties waardoor het ontwikkeltraject aanmerkelijk kan verkorten. De ontwikkelaar hoeft zich door gebruik te maken van dit framework niet meer bezig te houden met SQL en JDBC verbindingen.

Het doel van Hibernate is om de ontwikkelaar zoveel als mogelijk te vrijwaren van programmeertaken die gerelateerd zijn aan het vastleggen van de gegevens. Hibernate is eenvoudig in een project te incorporeren, of dat nu een nieuwe ontwikkeling, of het ontsluiten van bestaande systemen betreft. Hibernate genereert de benodigde SQL instructies, en de ontwikkelaar hoeft geen code op te nemen voor het verwerken van de resultset en conversie van gegevenstypes. Daardoor kan een applicatie worden ontwikkeld die eenvoudig portable is te maken naar alle SQL databases.