vrije Universiteit amsterdam

Master-Project Thesis

Developing content management and Presentation facilities

with

Adobe Flex and the Zend Framework

Document code	: Thesis_ska310_v1.0.docx
Version	: 1.0
Status	: Final
Created by	: Stefan van de Kaa
Date:	: July 21, 2009
Supervisor	: Anton Eliëns
Second reader	: Ronald Siebes

History of Master-Project Thesis

Overview of corrections

Date of this correction: July 21, 2009	**Date next correction:**

Correction date	**Previous correction date**	**Summary of corrections**	**Show corrections**
Juny 21, 2009	-	First edition	-

Approvals

This document needs the approval of the following persons.

Name	**Signature**	**Release date**	**Version**
Anton Eliëns		July 21, 2009	
Ronald Siebes		July 21, 2009	

Spread

This document has been send out to:

Name	**Spread date**	**Version**
Anton Eliëns		1.0
Ronald Siebes		1.0
Onderwijsbureau		1.0

Management summary

In the current website world there are a lot of available Content Management Systems. Most of them work in a very complex manner and are not user friendly. With this project I designed and implemented a working content management system which works user friendly.

Secondly there are multiple tools to create presentations but almost none of them run in a browser and have the possibilities to save and load the presentations again. This is another things which I have created in this project.

The tools I have used to come to these products are Adobe Flex 3 and the Zend Framework. To work with PHP in the backend I used PHPEclipse as a plug-in for eclipse. Also the Adobe Flex builder plug-in inside eclipse to get a complete working environment.

After the complete design the CMS was implemented. For data transfer from Flex to PHP I used the AMF protocol via PHP. The default Flex RichTextEditor has not been use due to lack of possibilities. Instead of this I worked with FCKEditor which does suit the needs.

For securing the CMS I have used an open source library which had MD5 hashing from within Flex so that the data wouldn't be transferred without being encrypted. Also there is a login system and due to the AMF protocol it's almost impossible to use the functionality without having an API to see what the possibilities are.

There have been a lot of issues during this project but all of them have been solved inside the time of the project. And the result is an presentation application and an user friendly content management system.

Table of Contents

Management summary 3

1 Preface 7

2 Introduction 8

3 Project description 9

4 Approach 10

4.1 Planning 10

4.2 Development model 10

5 Initiation 11

5.1 Process 11

5.1.1 Adobe Flex 11

5.1.2 Zend Framework 13

5.2 Issues 15

5.2.1 Zend documentation 15

5.2.2 Master project 15

6 Implementing Presentation Facilities 16

6.1 Process 16

6.1.1 Flex RichTextEditor 16

6.1.2 Flash player full screen 17

6.1.3 Layout 17

6.1.4 Effects 17

6.2 Documentation 18

6.3 Issues 18

6.3.1 Flex RichTextEditor 18

7 Analysis Content Management System 19

7.1 Process 19

7.1.1 Button selection 20

7.1.2 System Design 26

7.2 Issues 27

7.2.1 Large project 27

7.2.2 WYSIWYG editor 27

8 Implementation Content Management System 28

8.1 Process ... 28

8.1.1 Backend model ... 28

8.1.2 AMF communication ... 29

8.1.3 IFrame within Flex ... 31

8.1.4 MD5 hashing ... 32

8.1.5 Password generation ... 33

8.2 Issues ... 33

8.2.1 Debugging AMF ... 33

8.2.2 Eclipse warnings ... 35

8.2.3 AMF variables ... 36

8.2.4 Configuring FCKEditor ... 37

8.2.5 Language ... 38

9 Extra information ... 39

10 Conclusion ... 40

Appendix I: Planning ... 41

Appendix II: Dictionary ... 42

Appendix III: Literature ... 43

Books ... 43

Meetings ... 43

Websites ... 43

Appendix IV: Reflection ... 44

Flex and the Zend Framework ... 44

Assignment ... 44

Knowledge ... 45

Adobe Flex ... 45

Zend Framework ... 45

AMF Protocol ... 45

Rewrite modules ... 46

Appendixes in attached document:

- Appendix V: Zend with Amf
- Appendix VI: PowerPoint application
- Appendix VII: Survey results
- Appendix VIII: System Design

1 Preface

This document is the final part of my Master Computer Science with the specialization of Multimedia. After 2 years of studying on the Vrije Universiteit I am graduating. I have chosen to do this with a project for the VU. This project couldn't be done without the people who supported or helped me. Hereby I would like to thank these persons.

Anton Eliëns for supporting me with the project and giving his clear opinions on the project. Also for giving extra information on how to make this project to a success.

Ronald Siebes for being the second reader and having the confidence that this will be an interesting and challenging project.

My parents and sister with her partner for giving me the strength to start and successfully finish this master education.

Gert-Jaap Hertgers for giving me some technical information about how to efficiently use the Zend Framework in an object oriented way.

Of course I would like to thank all other teachers at the Vrije Universiteit for giving me the information and knowledge to get where I am today.

Stefan van de Kaa,
Renswoude, July 2009

2 Introduction

There are many solutions to create a website or a presentation. One possibility to change a website is to type plain text HTML codes. Another solutions is to use a tool like Microsoft Frontpage or Macromedia Dreamweaver. Then we have the What You See Is What You Get (WYSIWYG) editors which might be used from Adobe Flex, Javascript of many other solutions.

The most important question for this project is how we can create an application with which the user can edit his website in the most user-friendly way. And for this special case we will use a combination of Adobe Flex with the Zend Framework. This document will go into the process of creating an application like this and give an answer to the next question:

> *"How can we develop content management and presentation facilities with Adobe Flex and the Zend Framework in a way that it is user friendly?"*

By giving a clear answer to this question we can see if this project has reached the goals which were set at the beginning. This document also contains the problems which occurred during the project and the decisions made during the process. Finally it will give a brief overview of the application which is created.

3 Project description

Within this project I will build an Content Management System (CMS) in such a way that the users can edit their website in a very user-friendly manner. They don't need to know anything of what happens in the background. For example if a user want an image within a page, then in the background there will change a lot. First there has to be an image which has to be uploaded. Then the code has to be inserted into the page for that image and then the properties like alt text need to be altered.

The second part of the CMS is an user system for the website. For example if we use a login system on the website then we can add users and remove users in the CMS. Also we can generate new passwords for users.

Another part of this project where I will start with, creating presentation facilities with Adobe Flex. This means that a user can create a PowerPoint like presentation. The only difference with this is that it will be build using just the browser. Also the user can save and load his presentations and display the presentation within the browser. This application will use only Adobe Flex without the Zend Framework.

4 Approach

To build a fully working CMS with the features we want I have divided this project into multiple phases. By doing this there is a clear distinction between the different tasks within the project.

This project consists of the following phases:

- Initiation
- Implementing Presentation Facilities
- Analyses Content Management System
- Implementation Content Management System
- Documentation

The initiation phase was to get a clear idea of what I will do within this project and further arrange everything with the VU. In the implementing presentation facilities phase I have implemented a working application to create and alter a Flex presentation. Then the analyses phase began to make a design of the CMS. In the implementation content management system phase I implemented the design which I made before. And the final phase was the documentation phase. This phase contains the documenting everything. Which includes this document.

4.1 Planning

In the beginning I made a planning for the project. After working on the project for a while a created a more detailed planning which can be found in appendix I.

4.2 Development model

Since the content and goals became clear in the project it was easy to choose the correct phases. Because I have worked with some development models before I chose the “shashimi” model for this project. This model is a little variation for the waterfall method. The most important difference between these models is that the sashimi model offers room for overlap in some phases. If we look at the planning we see that the analysis, implementation and documentation phases overlap each other. The reason this is the case is because during the process of developing software you always find some problems which you need to solve. And sometimes this means altering the design.

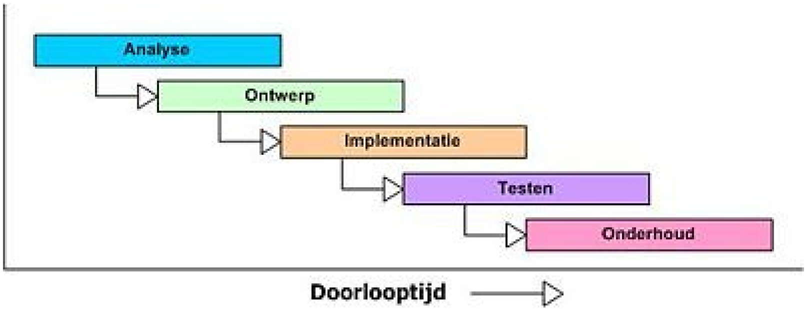


Figure 1: Sashimi model

5 Initiation

During the initiation I made the preparations to start the project in the right way. In this phase I made a detailed planning for myself.

5.1 Process

Before I began with this project I started with my literature study. Within this study I did research about all the different components of the Zend Framework. This gave me a good idea about the possibilities of the Zend Framework and how I could get this to work in combination with Adobe Flex. Also I noticed that there are many ways to use the Zend Framework and that the documentation isn't easy to use if you haven't worked with the Zend Framework before.

After this I made the preparations needed to start this project. I had a conversation with Anton Eliëns about the proposal of the project. We altered the initial proposal and came up with the final proposal. After this Anton gave me advice about a second reader. After this I contacted Ronald Siebes and he decided he wanted to be the second reader. After the approval of both readers I submitted the proposal into the wiki of the VU and submitted a copy to the student office.

When it was clear that I could start with my project I made a fresh installation of Adobe Flex with the Flex 3 SDK. Also installed WAMP (Windows Apache MySQL PHP) on it. After this I downloaded the Zend Framework and configured this for a new empty project. This was all I needed to do to get started with my project.

5.1.1 Adobe Flex

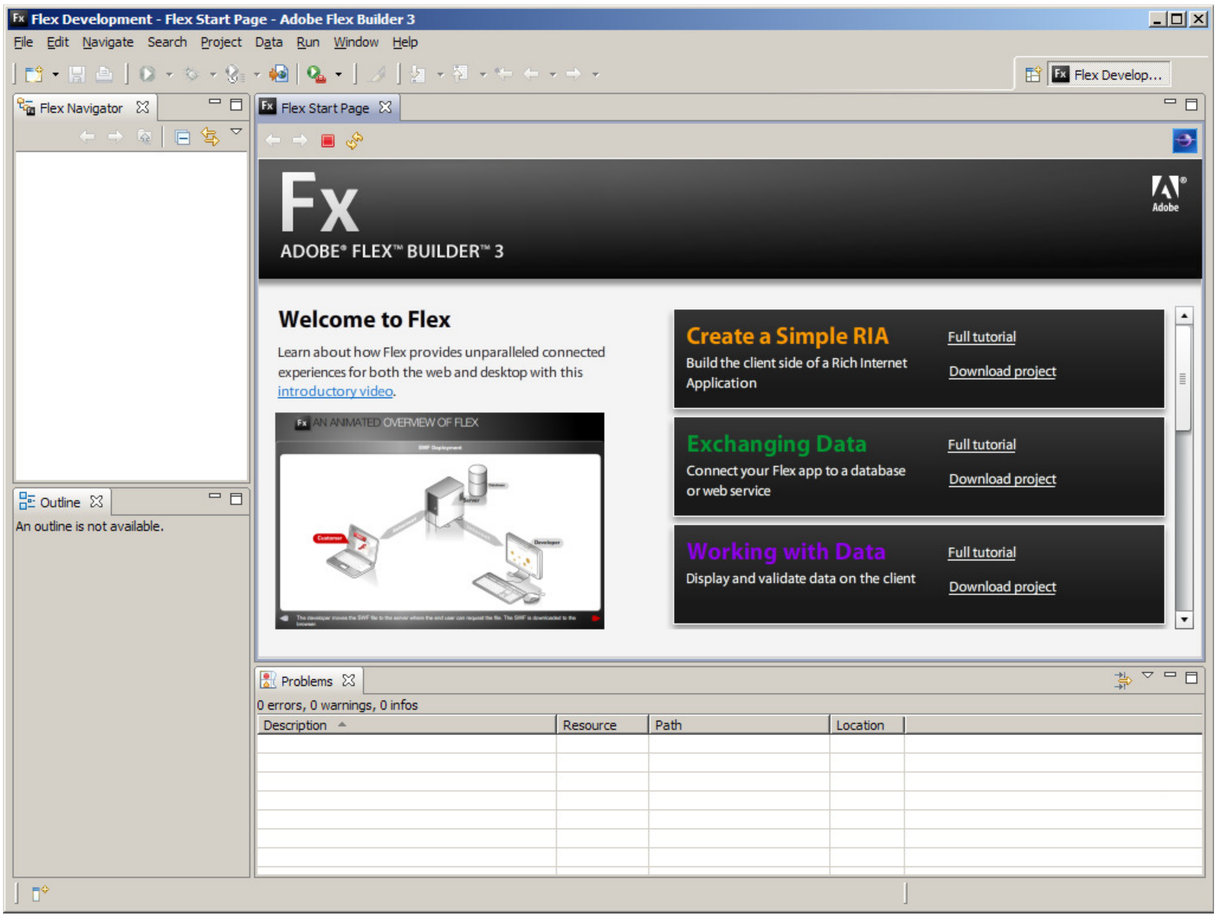
Here I will go into some more details of Adobe Flex. At the beginning Adobe Flex wasn't owned by Adobe. In the next table we will see the history of Adobe Flex. I excluded the alpha and beta versions in this overview.

Macromedia Flex 1.0 - March 2004
Macromedia Flex 1.5 - October 2004
Adobe Flex 2.0 - June 28, 2006
Adobe Flex 2.0.1 - January 5, 2007
Adobe Flex 3.0 - February 25, 2008
Adobe Flex 3.1 - August 15, 2008
Adobe Flex 3.2 - November 17, 2008
Adobe Flex 3.3 - March 4, 2009

Figure 2: Flex logo

So now we know that Adobe took over Macromedia, but what does Flex actually mean or do? Adobe Flex was supposed to be a version of Flash but then meant for the programmer instead of the designer. If I narrow this down a bit it means that where Flash works with the well known timeline in combination with ActionScript, Adobe Flex creates the Flash files different. With Adobe Flex you can build a similar application with MXML and ActionScript.

MXML is in fact a XML language which defines how the user interface should look like. This way the programmer can “program” how he wants the user to feel the GUI. The nice thing of Flex is that you can use it without having to pay for expensive licenses. You can use the SDK and use your own editor like for example notepad or editplus. The editor which Adobe has created is called Flex Builder and is build on top of Eclipse. This makes it easy for Java editors which are mostly familiar with Eclipse.



Welcome to Flex

Learn about how Flex provides unparalleled connected experiences for both the web and desktop with this introductory video.

Create a Simple RIA — Build the client side of a Rich Internet Application

Exchanging Data — Connect your Flex app to a database or web service

Working with Data — Display and validate data on the client

An outline is not available.

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Figure 3: Flex Builder

The version I have used for this project is Flex Builder 3 with the Flex 3.3 SDK. The reason to use this version was made because there are some changes between version 3.2 and 3.3 which enables it to load a file into the Flash player directly without use a server language like PHP, ASP or JAVA. At the moment of writing this document they are working on the new version of the Flex SDK and Flex Builder. If you are interested in this you can look on the Adobe website and search for Flash Builder 4. This is not a wrong typed name! With adobe they changed the naming of the products to keep them in a more general line.

5.1.2 Zend Framework

There are multiple frameworks which you can choose from when you want to work with one in the backend. In a previous project I have tested two of them and found out what would be the best framework for a combination with Adobe Flex. The framework I selected was the Zend Framework, the reason for this is that it can be adjusted to the wishes of the user. Also there is no generation module in it. This has both advantages and disadvantages. For instance you have to do the setup of a project yourself while with other frameworks this can be generated. On the other hand, because all these things aren't generated you know exactly what happens behind the screens and there is no code which is useless. There is no overhead.

This framework makes sure you program your code in a nice and clear manner. It almost forces the programmer to write his code Object Oriented. And if you know how the framework works you will not write any code twice. You can build all kind of helpers which you can use everywhere in the application.

In this framework there are a lot of things generalized. For instance the connections to multiple data providers. You can connect to a web service or just an AMF (Action Message Format) service of the flash player in exactly the same way.

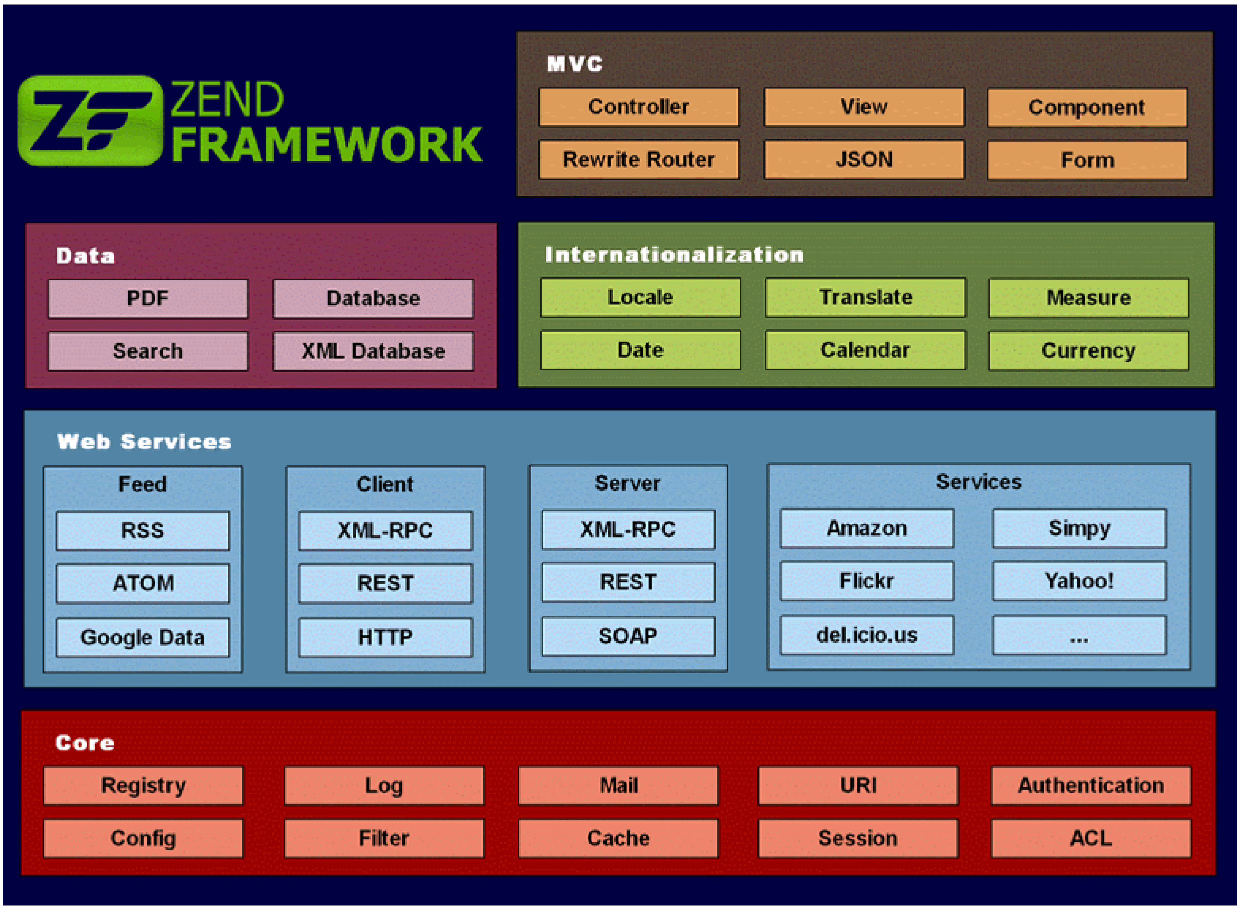


Figure 4: General overview of the Zend Framework

As we can see in the image above the Zend Framework is divided in multiple parts. This is also the best way to look at it. It starts with the idea to keep it Object Oriented. Then the default idea is to have a controller which will be called depending on the URL. And that controller calls the action which the URL has. Than that action handles the request and for example gets all books from a database and displays this in a view. And that view will be rendered and displayed to the user which did the request.

To clear it up, if we would browse to http://www.mywebsite.com the index file of the server will be called. In this case if we have set up Zend then Zend parses the URL and sees that we haven't defined a controller or action and will go to the default (IndexController) with the default action (indexAction).

And if we would browse to http://www.mywebsite.com/books/show then the server still loads the index file (this because Zend uses rewrite rules) but then Zend sees that we should go to the BooksController and there load the showAction.

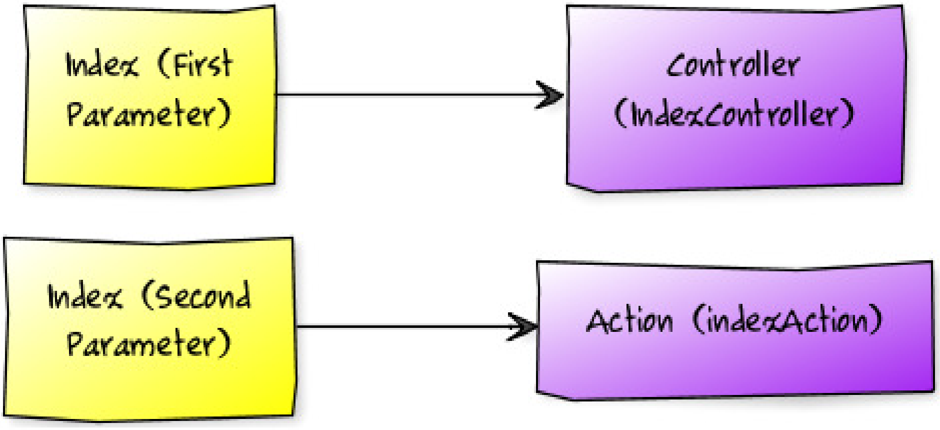


Figure 5: Controllers and Actions (ex: http://www.mywebsite.com/index/index).

Then we do the actions we want to do and set the variables in the view which is coupled to this action. For example the showAction is coupled to a view called show.phtml. And to get the information for the variables we can use one of the supported databases or if we want a web service.

For more information about the setup of a Zend Framework project visit VI: Zend with Amf. In this document I discuss how to set up the Zend Framework and one of the communication option of the Zend Framework namely Zend_Amf.

5.2 Issues

5.2.1 Zend documentation

One of the issues I noticed during the initiation (and previous multimedia projects concerning the Zend Framework) is that there is a lot of documentations about the framework but within this documentation it's not clear for which version of the framework it is. This makes it difficult to find out how to set up a project correct with the right version of the framework. Also the documentation is for advanced users and lacks proper examples.

Secondly I noticed that there is almost no information about the history of the Zend Framework. And in my personal opinion it's good to know something about the history of a product before you start using it. This made me read experiences of other people to get an idea about the state of the framework and about how mature the framework is. The framework is at the moment in version 1.9 but when I started at version 1.7. And has modules for the most common functionalities you might want to implement in your application.

5.2.2 Master project

Before I started with my master project I was told that my academic profile should be approved. When I looked for more information about this I noticed that this should be done in the first two months of the Master education. Because I had to do this first before starting my Master I had a small delay and could start a bit later. This however gave me the time to look take a better look at the exact project and how I wanted to run the project.

6 Implementing Presentation Facilities

In the second phase I tried to find out if it was possible to create presentation facilities with Adobe Flex. I did gave myself some requirements. First of all I wanted it to be run in the browser. Second I wanted to be able to save and load the presentations which we created. And finally to make it user friendly I wanted to add Drag'n'Drop functionality so that the user can reorganize the sheets by dragging them to the new location. Also inserting, deleting and updating was a must.

6.1 Process

When I started with the creation of this project I wanted to build the presentation facilities in such a way that it will be easy to use and user friendly. So I started with inserting sheets and being able to update them. When this was done I wanted to be able to delete them from the list and rearrange them with Drag'n'Drop.

When this was also finished I wanted to be able to load and save the presentations to XML. The saving to XML seemed easy to work with. Most of the Flash players already support file saving to the browser which makes it possible to give a popup wherein the user can select where he or she wants to save the file.

The problem came with the loading of a presentation into the application. If I was only going to use Adobe Flex without an extra language as server side I had to use the Flash player 10. This because only flash player 10 supported the loading of files into the flash players memory directly. But to use a flash player 10 application I had to use the newer version of the Flex SDK namely Flex 3.3.

6.1.1 Flex RichTextEditor

The default Flex RichTextEditor doesn't support the inserting of images. Therefore I needed to extend this component to be able to put an extra button on the component which makes it possible to give in a location of an image and then insert that into the HTMLText of the component. This means a lot of extra work.

Another option for this would be to let the user insert for example one or two images on each sheet and give these images a default position. Since I personally don't think this is user friendly I didn't choose for this option.

There was another small problem of the default editor of Flex, namely the HTML which it creates contains all kinds of extra font information which will not be accepted by for example Internet Explorer. So if I wanted to use the HTML generated from the editor I needed to filter this out or display the sheets in the application itself. Displaying the HTML inside the application was for me the best option because the application then also looked like one complete application wherein you can do everything.

6.1.2 Flash player full screen

Since I wanted to display the presentations in the Flash player it would look the best when it would be shown full screen. When trying to implement this I found out that the flex project needed to be configured to be able to go into full screen state. The way to solve this is edit the default template of the flex project and add a property. The property allowFullscreen and set this property to true.

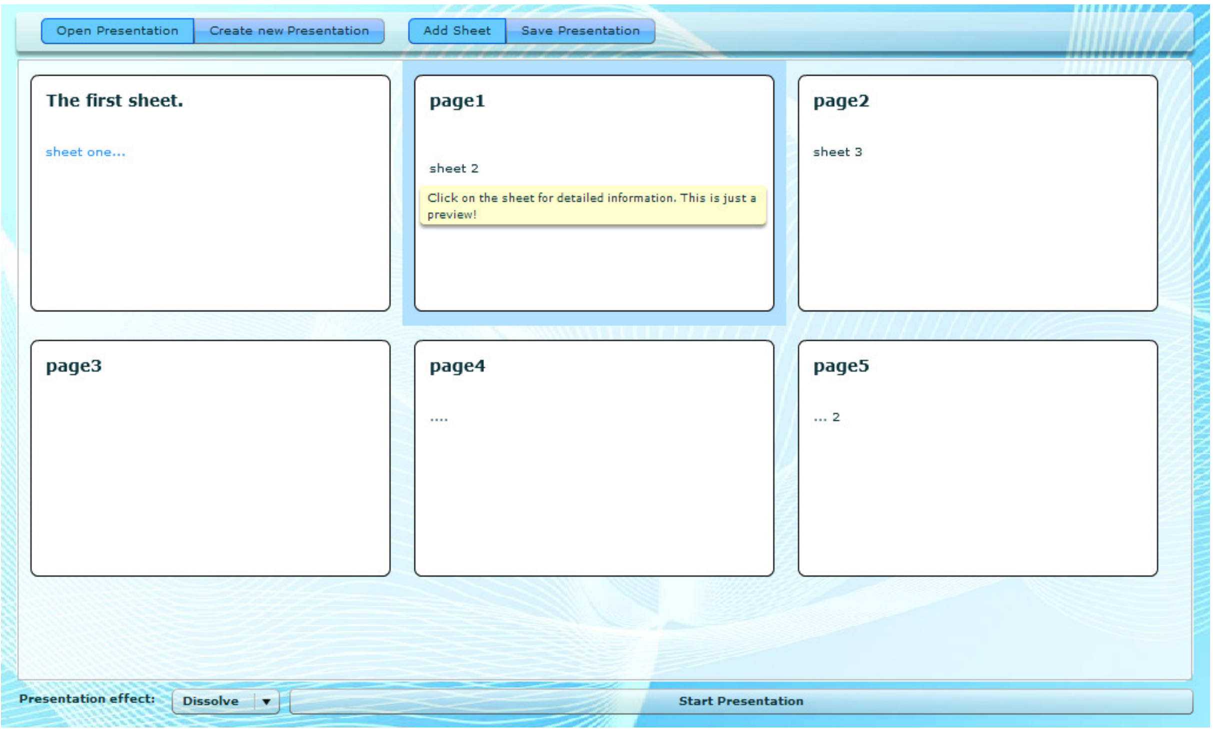
When this is done we can add a handler to a button which sets the state of the application to full screen. And when the user presses the escape button get back to the normal state. During the implementing of this I also noticed that when the application is in full screen all the input fields are disabled. So starting the application in full screen and edit the presentation there wasn't an option.

6.1.3 Layout

When you create a presentation you can normally also change the background and all kinds of styling issues. Because I am not a designer and wanted to keep it as simple as possible I chose to not include anything of this into the application. I did add a default background image which has a nice look and feel for the eye. It would be extremely simple to add functionality to this application to change the background.

6.1.4 Effects

One of the best features of Flex are the effects. By adding effects to an application the application will be experienced in a completely different way. If we would see a presentation without any effects the presentation will automatically look a bit boring. Since I think this is an important feature of presentation facilities I also added this into the application. I inserted a few default effects but this can also be extended to make the look and feel for the user very different.



The first sheet.

sheet one...

page1

sheet 2

Click on the sheet for detailed information. This is just a preview!

page2

sheet 3

page3

page4

....

page5

... 2

Presentation effect: Dissolve

Figure 6: Flex presentation application.

6.2 Documentation

To give the users an idea about how to build this application I have created a manual to do this on their own. Within this I have walked through all the features which I added into the application. This document is added as VI: PowerPoint application.

6.3 Issues

6.3.1 Flex RichTextEditor

When I was extending the default Flex RichTextEditor component to let it fit my needs I noticed that there was a bug with this component. When you add a image into the editor the editor adds a lot of extra font information. Within this information it says that the image has a font of size two. This wouldn't be a problem on its own but if you now get into a special position of the text near the image the font size changes to two pixels.

I have tried to build the entire RichTextEditor from scratch but since this is a but inside the HTML engine of the Flex SDK there is no way for me to fix it. Personally I think this is a big problem if this application would be used in a live environment.

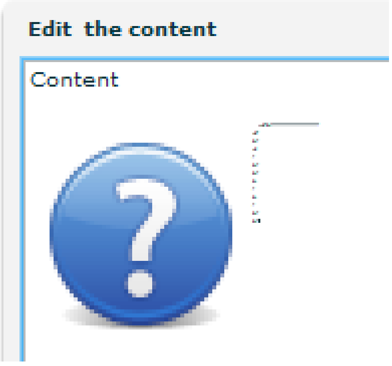


Figure 7: Bug in Flex RichTextEditor

In the image above we can see that next to the image the font size has changed from the default (twelve pixels) to the new font size (two pixels).

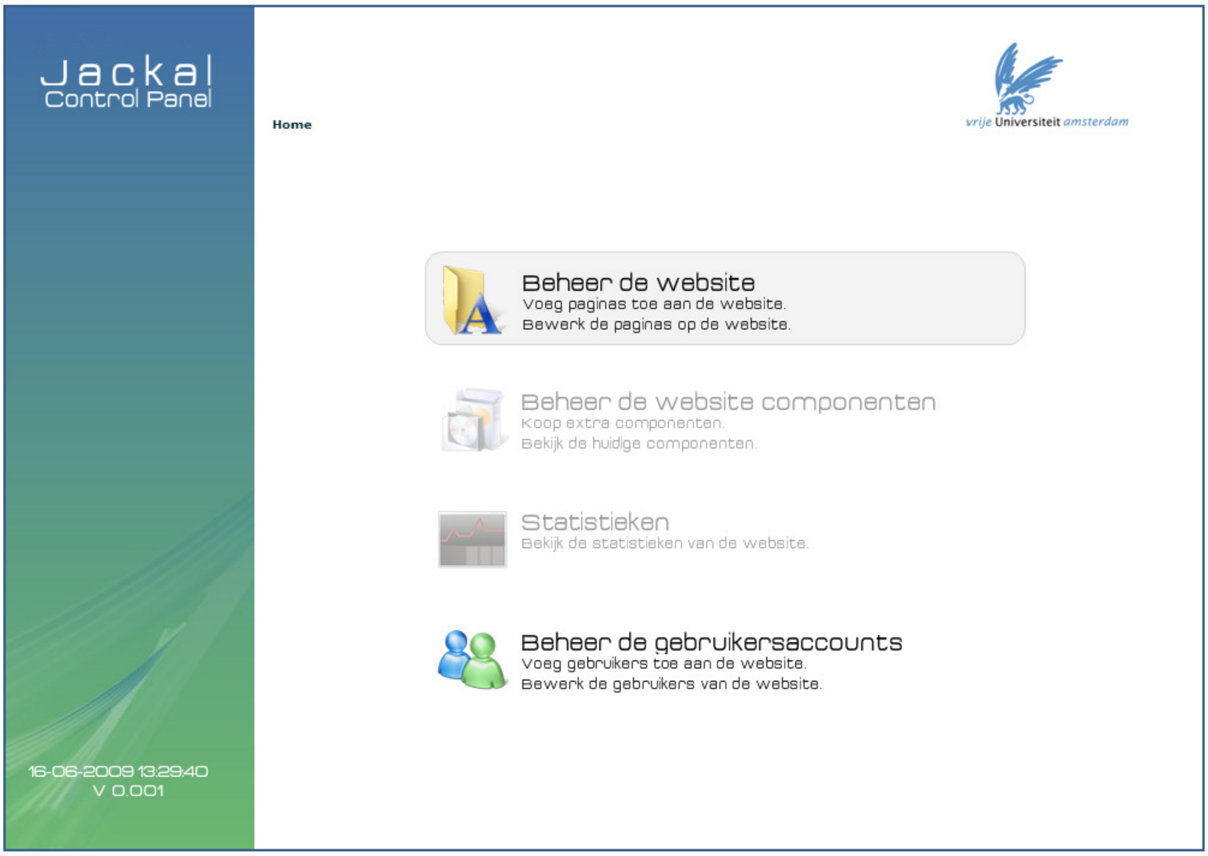
7 Analysis Content Management System

During the analysis phase I gathered information about how people experienced some features which I wanted to include in the Content Management System. When this was all clear to me I created the Graphical User Interface and started with the functional requirements.

7.1 Process

The first thing I did was create a survey for random people. In this survey I asked question like what they think of the windows control panel and what their experience was with for example Drag'n'Drop. I attached all the results in VII: Survey results.

The most important thing I noticed from this survey is that the most people know the windows control panel and that they also know how Drag'n'Drop works. But that their opinion about if Drag'n'Drop will work for Content Management is scattered. Because of this I choose not to implement Drag'n'Drop into the Content Management of the pages. And since the most users know how the control panel work I wanted to make a graphical user interface which works like that.



Beheer de website
Voeg paginas toe aan de website.
Bewerk de paginas op de website.

Beheer de website componenten
Koop extra componenten.
Bekijk de huidige componenten.

Statistieken
Bekijk de statistieken van de website.

Beheer de gebruikersaccounts
Voeg gebruikers toe aan de website.
Bewerk de gebruikers van de website.

16-06-2009 13:29:40
V 0.001

Figure 8: Content Management System GUI

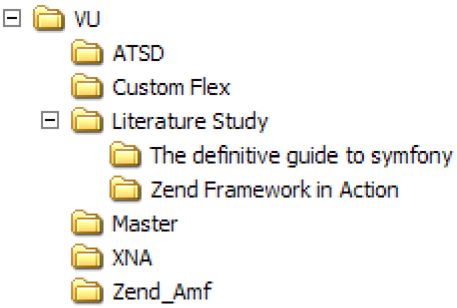
7.1.1 Button selection

When I was thinking about what the Content Management System should support I came up with the following selection.

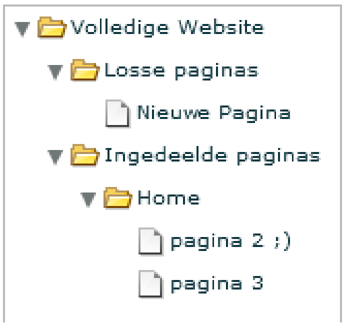
7.1.1.1 Page management

Since this is the most important part of a Content Management System this is the first button. Here the user can add pages to his website, edit his pages and if he wants also delete them from his website. Also can he edit the keywords and metadata for the pages. The reason for this is that the CMS should handle working with Google.

One of the results of the survey was that most people are used to the windows structure and control panel. Therefore I decided to use the folder structure of windows to display the pages. This makes it the easiest for users to work with and it also speaks for itself. A nice advantage of this is that Adobe Flex has a build in tree component which I can use to represent the windows folder structure.



```
⊟ VU
    ATSD
    Custom Flex
  ⊟ Literature Study
      The definitive guide to symfony
      Zend Framework in Action
    Master
    XNA
    Zend_Amf
```



```
▼ Volledige Website
  ▼ Losse paginas
      Nieuwe Pagina
  ▼ Ingedeelde paginas
    ▼ Home
        pagina 2 ;)
        pagina 3
```

Figure 9: on the left the Windows tree structure
Figure 10: on the right the Adobe Flex tree structure

Another result of the survey was that most users say that the easiest solution to edit a website is by using an What You See Is What You Get editor. Because I also think this is the nicest solutions I decided that I will use an WYSIWYG editor for the managing of the content. As I previously noted that the RichTextEditor of the Flex SDK has a bug with images I still had to decide if I will still use this editor or will use another editor and integrate this into my CMS.

Because there are many open source WYSIWYG editors I took a look at the two most familiar. Namely FCKEditor and TinyMCE. Both the editors have nice features and work very well. Also is it for both editors possible to change them the way you want them to use. There are however some differences which made the decision clear if I would choose not to use the default Flex RichTextEditor. Next is a table with the most important differences.

	FCKEditor	TinyMCE
Output code	After update 2.6 RC Valid XHTML	Valid XHTML
Installation	Possible with: -ASP.Net -ASP -ColdFusion -PHP -Java -Active-FoxPro -Lasso -Perl -Python	Possible with: -PHP -ASP.Net -JSP -ColdFusion
Uploading files	Without any plug-ins	Need to buy a plug-in to upload files.
Image Management	You can upload and insert images directly.	You can't manage the images with TinyMCE
Load time	Slower than TinyMCE	Faster than FCKEditor
Browser support	Support for: -Internet Explorer 5.5+ -Firefox 1.5+ -Safari 30+ -Opera 9.0+ -Netscape 7.1+ -Camino 1.0+	Support for: -Internet Explorer 5.5+ -Firefox 1.5+ -Safari 3 -Opera 9.0+ -Camino 1.0 -SeaMonkey 1.0.5
Installation	Copy files into folder	Copy files into folder
Language support	Language packs	Language packs

When I took a look at the table above I decided that I would choose FCKEditor above TinyMCE even though TinyMCE is faster. The biggest advantage is that FCKEditor has build in support for files and images and has themes which can be used.

Since I wanted the CMS to have a look and feel like the windows control panel this is also a nice feature since FCKEditor has an Office 2003 theme which would look familiar for almost all windows users.

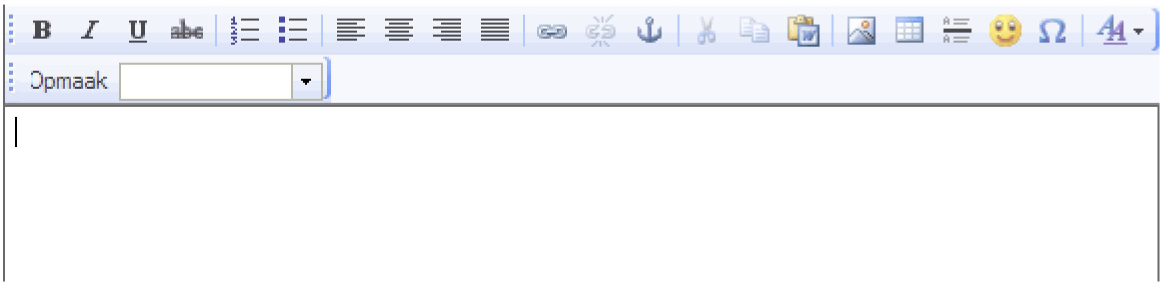


Figure 11: FCKEditor User Interface.

After this I had to make the final decision, will I use the Flex RichTextEditor or go for FCKEditor. For me the bug in Flex which made working with images to a nightmare was the downside and therefore I decided to use FCKEditor. Even though I didn't knew how to get this to work inside a Flex application.

7.1.1.2 Website components

The reason this button should be on the starting page of the Content Management System is very logical. I wanted the CMS to be build with modules. The idea behind this is for management and also to be able to create extra profits if the CMS will become commercial.

With a normal Content Management System you can create new pages, edit them and if you want delete them. This is all very straightforward. But what if we want to choose to insert a project page instead of a new page. Where the difference is that we can add a special number of photos to the project to be viewed on the side.

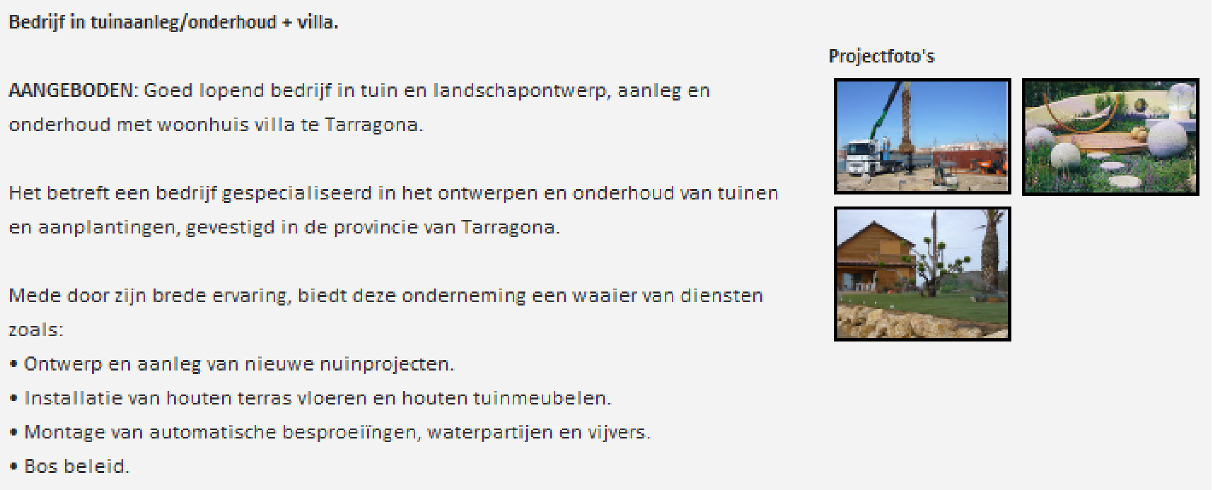
The idea is that these special types of pages (components) can be bought and activated by clicking on this button. But then is the question what kind of components do we want to offer here. Well actually all kinds of components you can think of.

Some examples would be:

- Guestbook
- Forum
- Picture gallery
- Web shop
- Forms
- Etc.

Since making all these components take a lot of time this feature isn't enabled in the working application. But the button stays in the design to show that it will be implemented. Also it shows very nice how the Content Management System should work in the near future.

Another reason to put the button in there and not hide it for eventual customers is that they have the idea that there are other components which they might want to use. And if they really want some components then they will ask for it. Then I can always mention that if the component doesn't exist that its always possible to create a custom component which will suit their needs.



Bedrijf in tuinaanleg/onderhoud + villa.

AANGEBODEN: Goed lopend bedrijf in tuin en landschapontwerp, aanleg en onderhoud met woonhuis villa te Tarragona.

Het betreft een bedrijf gespecialiseerd in het ontwerpen en onderhoud van tuinen en aanplantingen, gevestigd in de provincie van Tarragona.

Mede door zijn brede ervaring, biedt deze onderneming een waaier van diensten zoals:

- Ontwerp en aanleg van nieuwe nuinprojecten.
- Installatie van houten terras vloeren en houten tuinmeubelen.
- Montage van automatische besproeiïngen, waterpartijen en vijvers.
- Bos beleid.

Figure 12: Example of how a Project component might look like on a website

7.1.1.3 Statistics

Nowadays almost everything in real life depends on statistics. For this reason I think statistics are mandatory in a advanced Content Management System. There are a lot of ways to set up website statistics. One of them is to read the log files of the web server and then parse them into valid statistics. Another one is to use a already available tool and use this to view the statistics of the website. And a third solutions is to use the Google API and display the information you get from there into your own application. Let’s go into all options a bit deeper.

Parsing log files:

Parsing log files is a very complex task. To do this correctly you need very advanced knowledge of how the log files are build up. Another very important thing to know about this is that it works different on every single server application on which you might want to put your website. For example if you would use Apache then the log files are entirely different than when you would use Internet Information Services. And if you would use shared hosting then it might not even be possible to access the log files of the server because they might be protected or located in a location where you don’t have access to.

Because of all these downsides of parsing log files I decided this also isn’t an option to implement in the CMS. If I would choose for this solution I would have to know for sure that the CMS would on one server only where of the configuration wouldn’t change in the near future. And that I of course have access to the files needed for this.

Using available tools:

Using the tools which are available usually also use the log files of the server to generate the statistics. There are however some tools which only requires to insert a script tag which would then count the user statistics of your website. There is again a downside to this. This is a very basic statistics solutions which only counts how many users there are on a specific day. Sometimes you can see where the users come from, but this is also very basic and usually loaded from a table by checking IP addresses.

My conclusion to this is that this solution is even worse than when you would parse the log files of the server yourself. If you would parse the log files you could at least filter out all the information you want to see and display it exactly the way you want to see it. So this is not an option.

Google API:

With the Google API you can use Google Analytics to display the information you want on your website. To do this you also need to insert a script tag into the source of the website to retrieve the statistics. Then you can use the API with the login of the Google account to fetch it via a feed.

Once you have all the data information you can use this any way you want. There are some examples on what you can do with it. They can be found on the following website:

http://code.google.com/intl/nl/apis/analytics/docs/gdata/gdataGallery.html

The nice thing of this is that it can be integrated with Flex very nice since Flex has support for charts and very nice other features like data grids. The next image shows an example which can be found on the website mentioned above. These are screenshots of an mobile phone application build on top of the API.

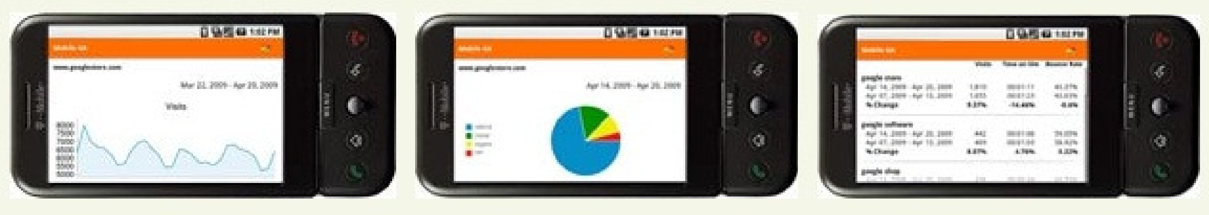


Figure 13: Example of Google Analytics API possibilities.

Since there are very much options and that there is a lot of information stored by Google in my opinion this is the best solutions for integrating statistics into the Content Management System. The only downside of it is that the API is very large and that it is very new so there are not much coding examples online.

Because this project runs six months integrating this into the Content Management System would probably take at least one month. In the original planning it was only the idea to integrate page management with the content and the user system. Therefore I will also disable this button in the menu. But I will not remove it since I still want to implement it later on.

7.1.1.4 User management

The final button for the main menu will be the user management button. To log in to the Content Management System you need to be a verified user. Therefore the user must be in the database. With the user management part you can add and delete users from the system.

In contradiction to the rest of the CMS I will not use the layout of the windows Control Panel for this. The main reason is that I want to keep it as simple as possible and use the most functionalities provided by the Flex SDK. I have seen some examples on the internet of how I could use Flex to display for example items. One of them it the Flex Store which can be found here:

http://www.adobe.com/devnet/flex/samples/flex_store/

In this store there are some phones which are displayed in thumbs. If there are more or less items in there the thumbs can change size depending on how many items there are left. I also want to use this for the user management. If there are only four users show two rows. If there are five to nine then show three rows. And if more than show four rows. Also there will be some effects in here to give a nice look and feel for the user.

Here is an image of the Flex Store to give an idea about how it should look like.

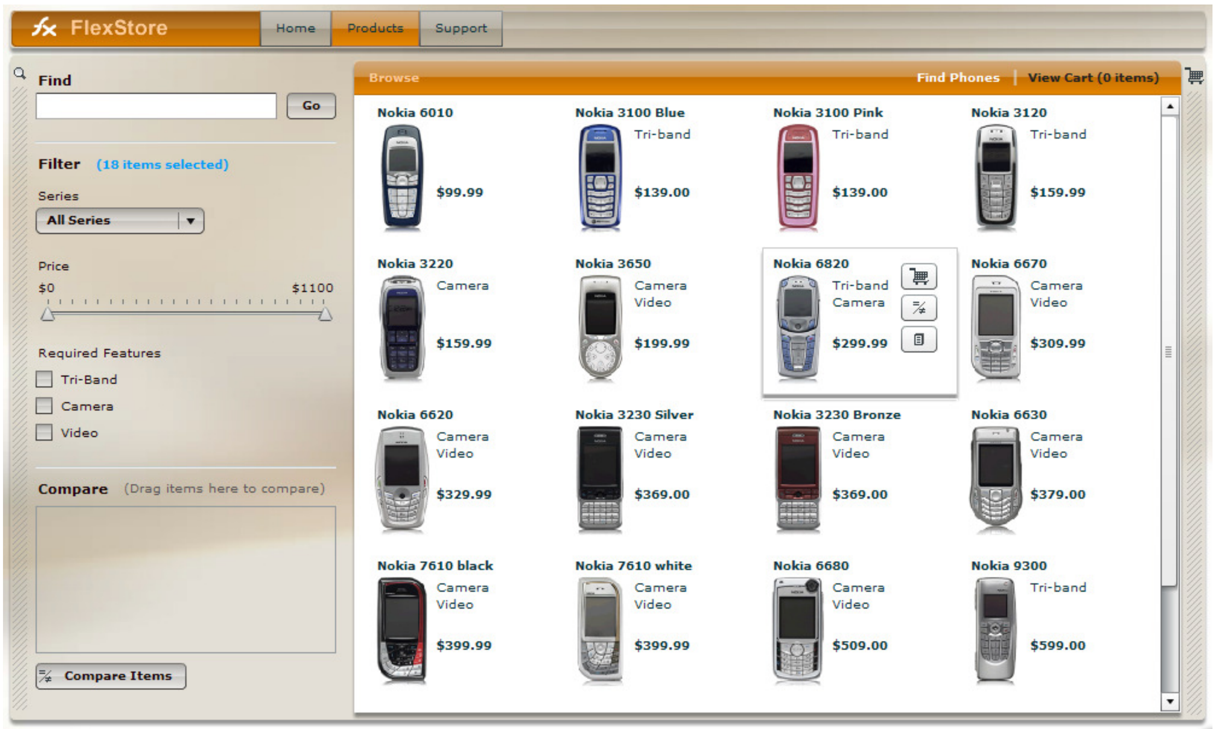


Figure 14: Flex Store

7.1.2 System Design

During this phase I created a system design. This design is attached as VIII: System Design. To create a good functional and fast CMS there were some things which needed extra attention. One of this is the design of the database. Since everything needs to work tight with each other and this is very important.

Things to think about when creating the database diagram which had extra value compared to a diagram for a normal CMS were the user rights part and the part where the components (For example the pages) had to be stored into.

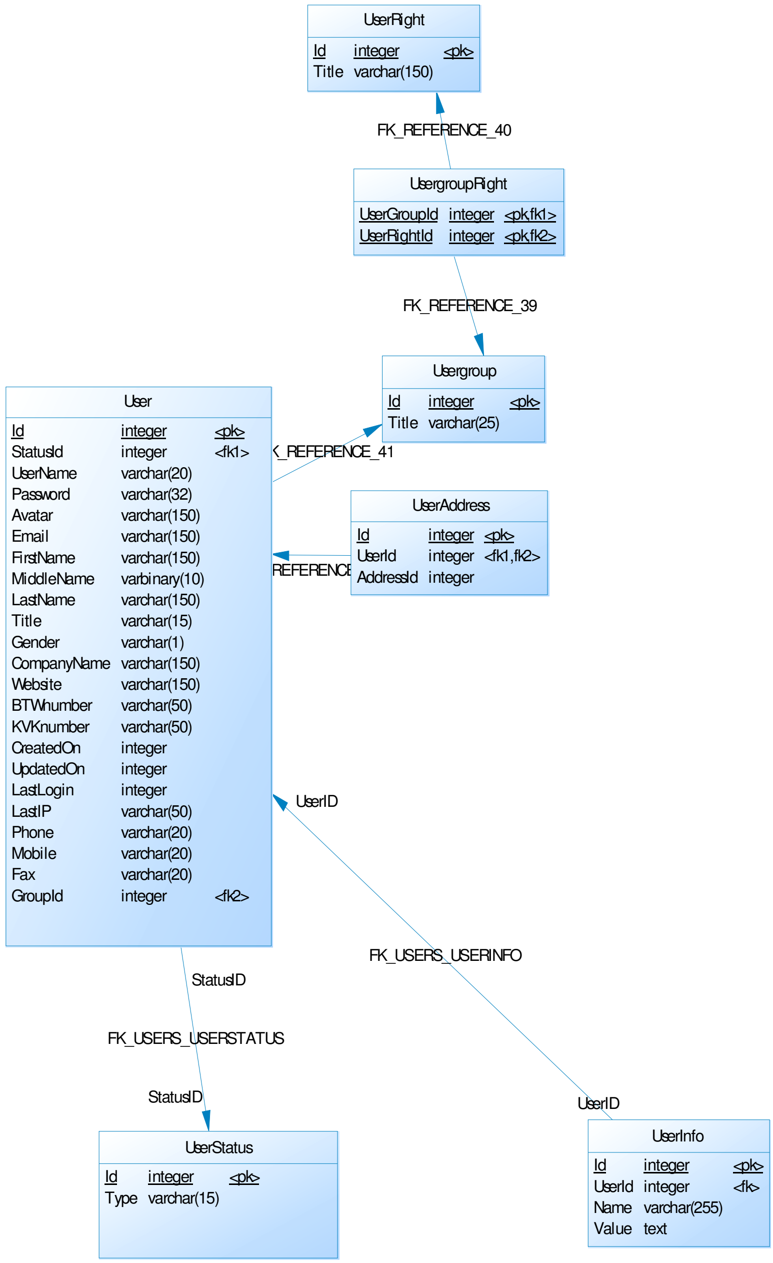


Figure 15: Database model, User part

In the figure above the partial user tables are visible. Here we can see that a user belongs in a group. The group can have several rights which he then owns. Since at the moment users van only login into the CMS the default group is administrator. Also there are other things stored like the address, extra info and his status.

This entire diagram is also included in the system design attachment where it is described into more details. Further all the requirements of the system are written down in the document.

7.2 Issues

7.2.1 Large project

This project can be expanded to a very large scale. To keep in manageable and acceptable within the time zone I had, I have chosen to create the user system and the page managing system. This because they are the most important parts of a Content Management System. If I would include the statistics and components than this project could be expanded to take so much time that I will be working on it for another year.

I actually am really enjoying working on this project which will probably mean that I will continue working on the CMS after the project is finished. Then I will build in the other features to make it more elaborate and insert components for eventual users which might use the system in a live situation.

7.2.2 WYSIWYG editor

After I concluded that the Flex RichTextEditor doesn't suit my needs I decided that I want to use the FCKEditor within Flex. Since I haven't worked with including HTML into an Flex application I needed to figure out if this was possible at all. After looking around on the internet I did find a solution to use the FCKEditor within my application.

Searching for a good solution however did took some extra time. There is also a component available online which can be bought and then used within the Flex application but I decided to work around this and write my own code to get it to work. The advantage of this is that I can manage my own code and if there are fix the bugs which isn't possible if I would use a component build by someone else.

8 Implementation Content Management System

Within this phase I implemented the Content Management System according to the requirements which are described in the appendix VIII. For this I used Eclipse with the Flex Builder plug in. For the PHP programming of the Zend Framework I used the PHPEclipse plug in for Eclipse. This made working with two different languages easier.

8.1 Process

The first thing I did in implementing the Content Management System was finding out how to set up the model with PHP. The default solutions would be very general and I wouldn't have to do much to get this to work.

8.1.1 Backend model

I have worked with the Symfony framework before and in this framework there are a lot of classes which are generated. The classes use very nice collections and are very advanced. In these classes you can define specific functionalities which can be very useful for example when checking certain entries in the database.

Since this is a very clean solutions I wanted to implement this into the Content Management System. The general idea how this work is the following. We have an object class which contains the information which I want every object to have. For example an ID. Then we get for each table another class. This class contains the information which is different for the tables. For example an user table contains an address and a lot of other information.

Then we get a manager class. Using this manager we can get entries from the database. An example function which could be in this manager might be UserManager::Get($userId). This in his place returns the user object.

If we would have a return value which contains more than one user, then this will be returned into a collection. A collection is sort of an Array but with more options. For example filtering, updating and iterating through it.

There is one final thing which makes the database interaction complete and that is the database classes. These files are necessary to define which table corresponds to which PHP class. Without these files the entire solutions will not work.

Working with these classes takes a lot of extra time since there are a lot of files which need to be created for every table in the database. There are however enough advantages which make it worthwhile. The most important one is that the entire model looks very general and work in a very easy and clean manner. If you want to add functionality to any table this is done very easily. Also add checks to inserting, updating or deleting is very easy. This solutions also speeds things up. You can check if the records are already fetched from the database. If this is the case return the active data and otherwise fetch the new data. Also with saving records this is easy. You can check if there are any changes to an object before saving it. So if there are no changes you also don't need to save the record into the database.

8.1.2 AMF communication

For the communication between the Flash player and the PHP backend I choose to use the AMF protocol. When implementing this into the content management system I ran into some problems. With the AMF protocol we can send objects from PHP which can be mapped to Flex objects. This looked like a nice feature. But when implementing this I noticed that this is almost impossible. There are a lot of errors which aren't documented anywhere.

So therefore I tried to find a workaround. Looking on the internet didn't give me any ideas on how to get this to work. I however did figure out that I could transmit data from PHP to Flex via AMF if the data was wrapped into an ArrayCollection. So when I inserted the variables from the PHP object into an ArrayCollection and send that over to Flex I could work with it.

Using that as the idea for transmitting the data I added a wrapper class into the backend which would handle all the data transactions. This class contains five functions which could handle all the transactions namely:

- callUserFunc()
  This function can handle function calls onto a given object. This means that I will have to give the object to this function and the name of the function which should be called upon the object with the parameters. To give an example:

  ```
  callUserFunc($object, “setName”, “NewName”);
  ```

  If we would call this function from Flex and the $object variable would be the referenced object we can set the name of the user and update it. This would be nice since there are a lot of things we want to set from Flex.

- getObjectFromManager()
  Sometimes we want to receive an object from a manager class. For example if we want information about the user which has just logged in. Then we can use this function to get the object and all his properties. Once we have them in Flex we can do everything with it. To give an example:

  ```
  getObjectFromManager(      "UserManager",
                             "ValidateUser",
                             new Array(
                                   usrename.text,
                                   MD5.hash(password.text), true

                             )
                       );
  ```

  This functionality called from within Flex can check if an user is valid. If the user is valid it returns the user object itself. To get this to work and return the user object as an Array I needed to add the final parameter true. This way the function will give an ArrayCollection back instead of the real user object.

- getObjectsFromManager()
  This is of course also possible with more than one object. So if we want to get more than one object back we can call this function. The difference in the PHP code is that the return of one record is just a plain object. When it's more than one the model turns it into a collection. This collection cannot be parsed by the AMF and Flex. Therefore this function changes it into an ArrayCollection with the objects and their information.

  This way the Flex application can work with it and display all the information it needs. This is for example done with the part where the pages are loaded. And also where the users are displayed.

  The advantage of loading all the objects at once is the time of the loading. If we would load all the pages one by one the loading will take more than a minute because of all the requests. And if we load them all at once it's done with one request and is done in less than five seconds.

- save()
  Since there are caching mechanisms build into the model it is better to keep a separated save function. The special reason for this is that we need to do at least two actions. The first are setting the settings of the object which we want to save. Then when that's done actually save it into the database. If we wouldn't do the last it will be changed into the model where it's cached but not saved.

  This function handles saving for two different types. The first one is when it's a new object. When it's a new object it saves the new object into the database en than gives that object back. Because of the fact that we didn't have the id of the record this id is added into the ArrayCollection which is returned.

  If the record already existed in the database we update the record and return the updated record. This way the Flex application always works with the latest version of the database information.

- delete()
  The final function which is in this wrapper class is the delete function. With this function we can delete records from the database. This does require the object itself to be given as parameter. This is a check so we know that we will not delete entries which shouldn't be deleted.

8.1.3 IFrame within Flex

In the design I decided that I didn't want to use the default RichTextEditor from Flex. For this reason I needed to insert the FCKEditor into Flex. After searching around for a while I noticed that the only solution to do this is by using an IFrame within the HTML page which renders the editor.

There are some details which were required for this to work. One of them was changing the preferences of the flash player. In here the "wmode" property needed to be set to opaque. What this actually does is that it makes it possible to show HTML properties in front of the Flash movie which is the actual application.

Then by accessing JavaScript functions within Flex we can set the source of the IFrame in the HTML page and then load the HTML from the database into the editor. This is a strange workaround but the only working workaround which I was able to find and get working. I added some extra JavaScript functions to be able to load the content from the editor into the Flex application. This was necessary for saving it back into the database. Also I added functionality to hide and show the IFrame depending on the actions that the user takes.

The complete version of the editor within the Flex application is visible in the next image.

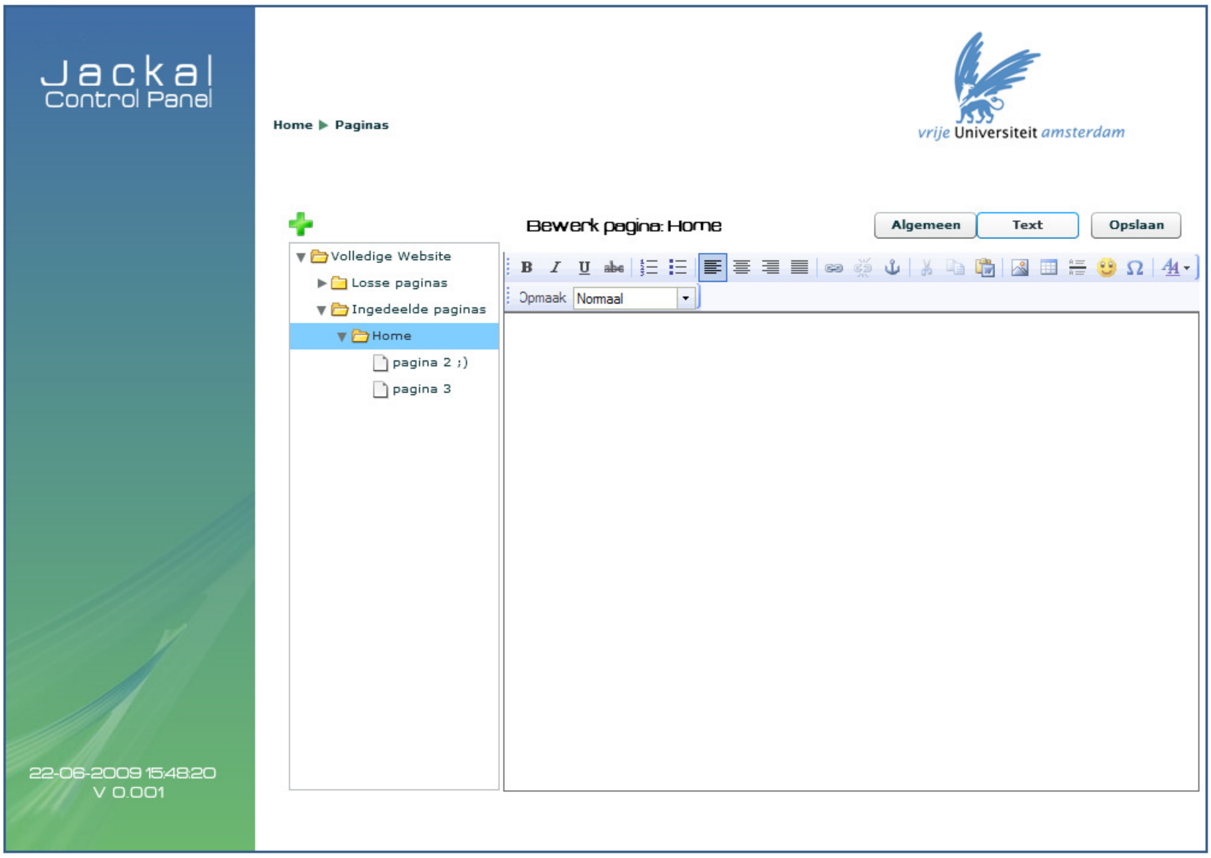


Figure 16: WYSIWYG Editor (FCKEditor)

For this editor I choose to disable the most features. The reason to do this is to keep it as tight and easy as possible for the user. The most features will not be used when editing a normal website. These are only for the advanced users. And if necessary this can be enabled after all.

8.1.4 MD5 hashing

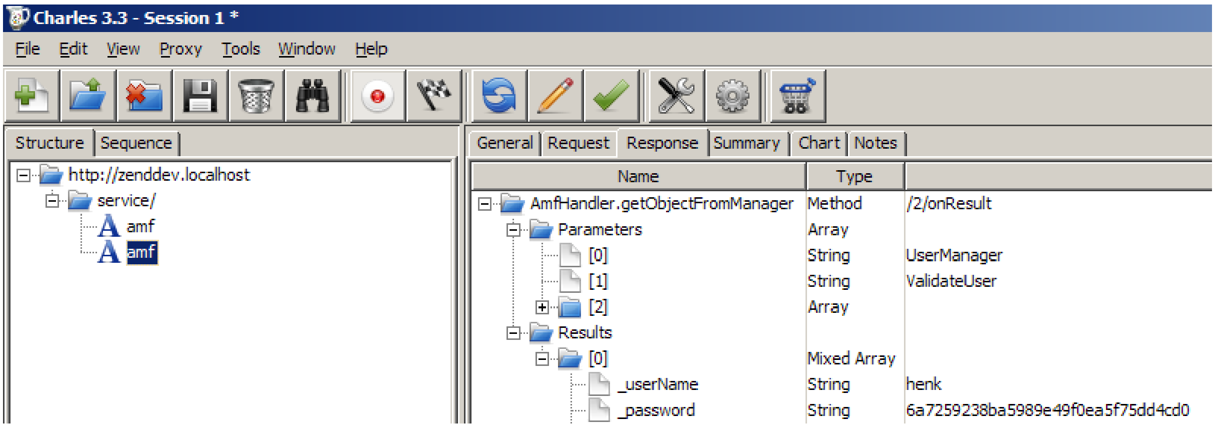
Working with the Content Management System has to be secure. In the database all the passwords are stored with MD5 hashes so that it cannot be leaked in any way. But because the information need to be transferred from the Flex application to the PHP backend it can be found using a packet sniffer. The passwords will namely be send as plain text to the PHP files.

Since this isn't a good solutions I wanted to do something about this problem. I was searching for a decent solution for this and I know it is not a good solutions to implement a hashing algorithm by myself. Then I ran across the Google code database and found an implementation build by adobe which made it possible to hash it with MD5 from the application. Then it will be transferred encrypted to the model and the problem is gone.

The information about the files I used can be found here:

http://code.google.com/p/as3corelib/

The only thing I needed to do is include the files into the project and then call the right function which would look like this: MD5("Encryption String"). Personally I think it is better to use code which is already there freely available knowing the code is tested and fully working, then when you want to implement the same code yourself. This because people are known to make mistakes!


Charles 3.3 - Session 1 *
File Edit View Proxy Tools Window Help
Structure Sequence
http://zenddev.localhost
service/
amf
amf
General Request Response Summary Chart Notes

Name	Type	
AmfHandler.getObjectFromManager	Method	/2/onResult
Parameters	Array	
[0]	String	UserManager
[1]	String	ValidateUser
[2]	Array	
Results		
[0]	Mixed Array	
_userName	String	henk
_password	String	6a7259238ba5989e49f0ea5f75dd4cd0

Figure 17: login results from Flex to PHP (MD5 working)

8.1.5 Password generation

To make the system very user friendly and secure I choose for the option to generate a new password for an user. This in case if he or she forgot her password. When the new password is generated and the mail server is configured in the configuration file of the Zend project the user will automatically receive the newly generated password in his email.

The generation is done in the backend. The usermanager is the correct place to handle this. According to the most recent discussions a password should be minimal six characters and containing at least one number or special character. Within the function I use the password will be eight characters or numbers. The characters could be capital or lowercase. The password will be mailed directly before inserted into the database for optimal security.

Implementing the mail functionality is very easy within the Zend Framework since they already have modules for sending email. With this module you can choose for plain text mail or HTML mail. I chose for a plain text mail since HTML emails are sometimes blocked by spam filters.

8.2 Issues

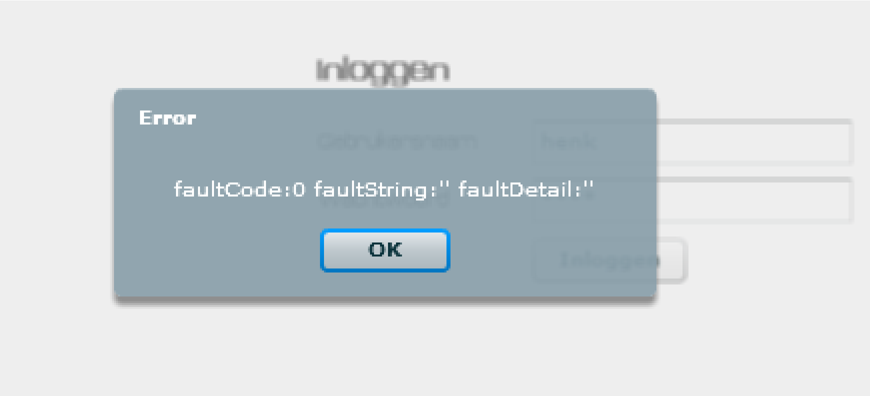
During the implementation of the CMS I ran across some issues which are noted below.

8.2.1 Debugging AMF

When I was working with AMF between Zend and Flex I noticed that there is no decent way to start debugging. Normally within Flex you can start debugging and see what goes in and out. With AMF this is not possible. The reason for this is that it is asynchronous.

Within the model it is also not possible to test if your classes are working. If you would call the AMF interface you will only get a text file with one line in it which says “<p>Zend Amf Endpoint</p>”. This way you can only see if the AMF endpoint is configured correctly.

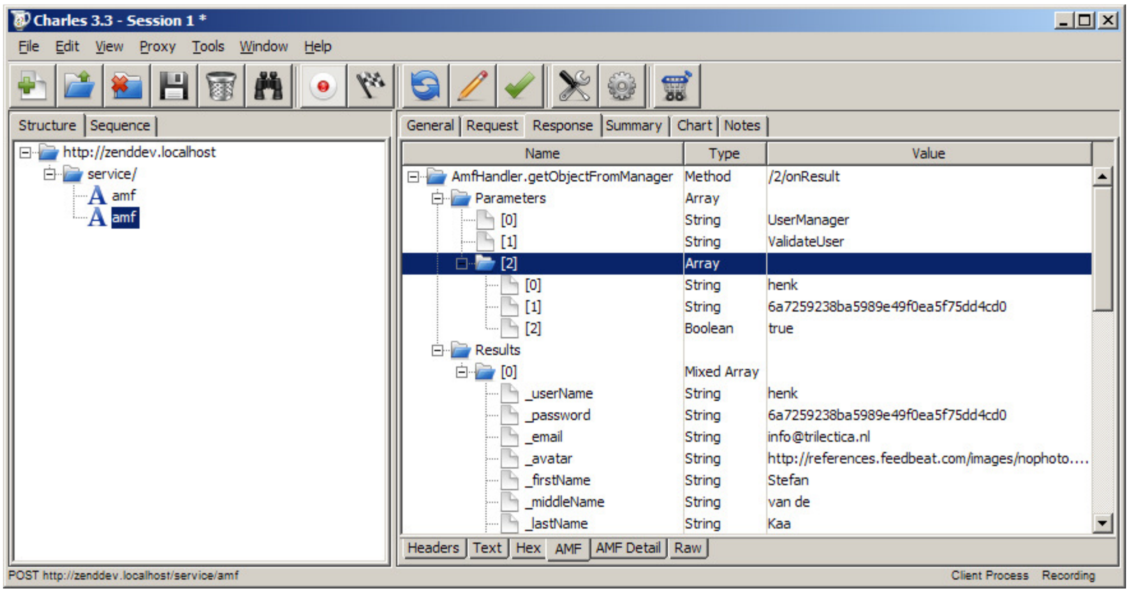
One of the errors which I couldn’t figure out easily was the next one.



faultCode:0 faultString:'' faultDetail:''

Figure 18: AMF error.

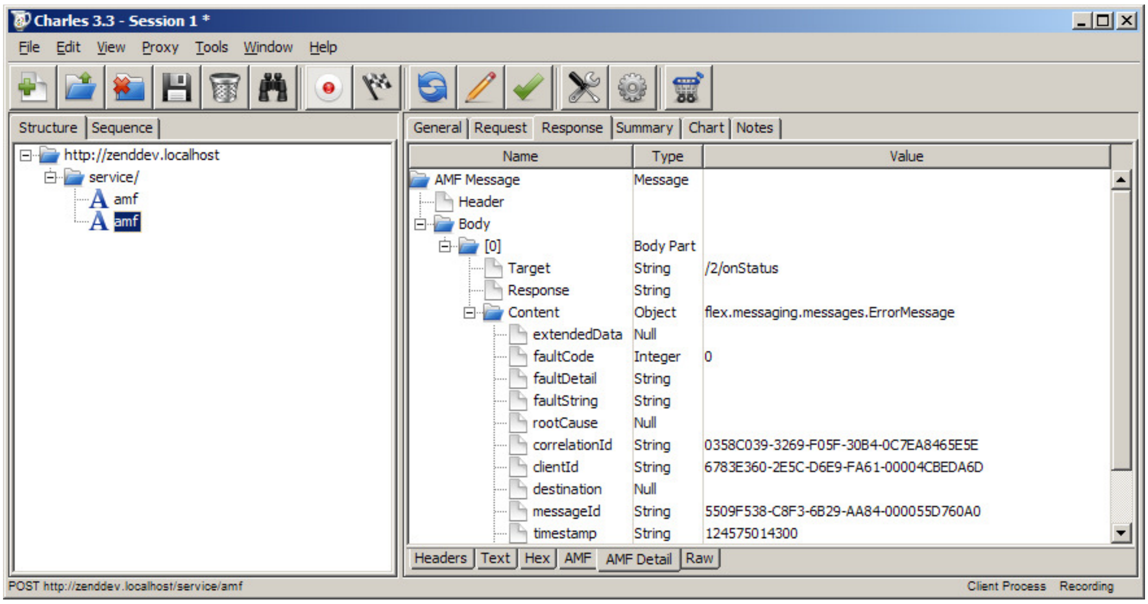
After trying to figure out how I was able to debug this I read a post on the internet about people who were struggling with exactly the same problem. In this thread one person mentioned a program called Charles. This program works like a package sniffer which reads all packages and formats in a way which is readable with the normal eye. This program also displays the AMF packages in a very readable way.



Name	Type	Value
AmfHandler.getObjectFromManager	Method	/2/onResult
Parameters	Array	
[0]	String	UserManager
[1]	String	ValidateUser
[2]	Array	
[0]	String	henk
[1]	String	6a7259238ba5989e49f0ea5f75dd4cd0
[2]	Boolean	true
Results		
[0]	Mixed Array	
_userName	String	henk
_password	String	6a7259238ba5989e49f0ea5f75dd4cd0
_email	String	info@trilectica.nl
_avatar	String	http://references.feedbeat.com/images/nophoto....
_firstName	String	Stefan
_middleName	String	van de
_lastName	String	Kaa

Figure 19: AMF response

But when I started using this program I noticed that if you get normal PHP errors, like forgetting an '$' sign or so, then you can debug very well with Charles. But when you have configuration errors or errors within the Flex application Charles doesn't help you at all. To give an example, I have changed the database configuration and now the password is invalid. The following image shows the result from within Charles.



Name	Type	Value
AMF Message	Message	
Header		
Body		
[0]	Body Part	
Target	String	/2/onStatus
Response	String	
Content	Object	flex.messaging.messages.ErrorMessage
extendedData	Null	
faultCode	Integer	0
faultDetail	String	
faultString	String	
rootCause	Null	
correlationId	String	0358C039-3269-F05F-30B4-0C7EA8465E5E
clientId	String	6783E360-2E5C-D6E9-FA61-00004CBEDA6D
destination	Null	
messageId	String	5509F538-C8F3-6B29-AA84-000055D760A0
timestamp	String	124575014300

Figure 20: AMF error response

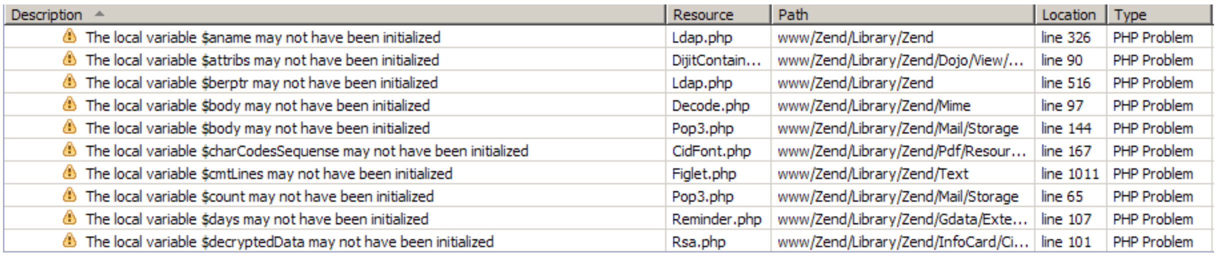
In this image we see the content which contains the error: “flex.messaging.messages.ErrorMessage”. This doesn’t tell me much about where I should search in the source code. If I look at the raw response then it doesn’t give me any more information about where to look. The best thing to do is walk through the code piece by piece. And for the model, check the functionalities of the managers from within another PHP class. Debugging this way really takes a lot of extra time but it the only way to debug the application.

One other bug which I ran into was that the application just wouldn’t respond and also didn’t gave an error. With Charles it also didn’t have any result but a timeout. I tried to find a solutions but the only one I could fine was coding some PHP classes from scratch.

In my opinion this is one part of the Flex framework which could be improved. The options at this moment take a lot of extra time and don’t work on a very easy way.

8.2.2 Eclipse warnings

When working with the Zend Framework there are a lot of warnings displayed in the problems bar. The warnings displayed are mostly from variables which haven’t been initialized. Some of the warnings are displayed in figure 21.



Description	Resource	Path	Location	Type
⚠ The local variable $aname may not have been initialized	Ldap.php	www/Zend/Library/Zend	line 326	PHP Problem
⚠ The local variable $attribs may not have been initialized	DijitContain…	www/Zend/Library/Zend/Dojo/View/…	line 90	PHP Problem
⚠ The local variable $berptr may not have been initialized	Ldap.php	www/Zend/Library/Zend	line 516	PHP Problem
⚠ The local variable $body may not have been initialized	Decode.php	www/Zend/Library/Zend/Mime	line 97	PHP Problem
⚠ The local variable $body may not have been initialized	Pop3.php	www/Zend/Library/Zend/Mail/Storage	line 144	PHP Problem
⚠ The local variable $charCodesSequense may not have been initialized	CidFont.php	www/Zend/Library/Zend/Pdf/Resour…	line 167	PHP Problem
⚠ The local variable $cmtLines may not have been initialized	Figlet.php	www/Zend/Library/Zend/Text	line 1011	PHP Problem
⚠ The local variable $count may not have been initialized	Pop3.php	www/Zend/Library/Zend/Mail/Storage	line 65	PHP Problem
⚠ The local variable $days may not have been initialized	Reminder.php	www/Zend/Library/Zend/Gdata/Exte…	line 107	PHP Problem
⚠ The local variable $decryptedData may not have been initialized	Rsa.php	www/Zend/Library/Zend/InfoCard/Ci…	line 101	PHP Problem

Figure 21: Zend warnings within Eclipse

I tried to instantiate some variables manually inside the functions to let the warnings disappear. Unfortunately this didn’t work the way I hoped. When doing this the code didn’t work the way it should. The variables which aren’t instantiated are global variables which are defined somewhere above in the architecture.

So the best thing to do is ignore the warnings and don’t try to get them disappeared from the error bar. Some of the error can however be removed. The solutions to do this is add the Zend library to the include paths of the project. When this is done the references of the files are found inside the project and the warnings concerning this problem will be removed.

8.2.3 AMF variables

As mentioned above is working with AMF between PHP and Flex not very programmer friendly. There are some other downsides which I noticed when working with it. When I first tried to transmit data from PHP to Flex and I wanted to send the variables back into an ArrayCollection I noticed that only the public variables were send back to the Flex application.

When I searched around on the internet I didn't find any solutions for this except for working with public variables instead of private variables. While this might be a solutions to get everything to work I decided this wasn't an option for this project. The main reason not to make all the variables public was because of the fact that this would ruin the integrity of the application within the PHP model.

Then I thought of another workaround for this problem. It might be possible to create an function which would return all the variables into an array and pass that information into an ArrayCollection to the Flex application. This is the function I came up with.

```
public function parseToArray(){

        return array(get_object_vars($this));

}
```

If I would use this within the object class and attach the original object to this array I can use all the functionalities I want. The reason why I attached the original object to the array was because I needed the reference within the Flex application to be able to edit the variables of the object. For example when I change the name of an user the original object with the new values will be send back to the backend and there the values will be changed and then saved to the database.

This entire situation will not work if the object will be send normally since Flex has no idea what to do with an PHP object. Therefore I chose to serialize the object. This way it's a plain text string which can be send over AMF and can be stored within the Flex application.

Another thing to think about when implementing this is that when the object is updated you do give back the updated object to the Flex application. If this isn't done then an eventual next update will not be executed since the object where the action should be executed on isn't the latest version.

Implementing this took some extra time which I didn't plan when I started. But once this was working I didn't need to think about anything else because all the transaction work in exactly the same way.

Returning the values to Flex would look like the following piece of source code.

```
// $obj is the actual object.

$objArr = $obj->parseToArray();

$resultArr = array_merge(array(serialize($obj)), $objArr);

return $resultArr;
```

8.2.4 Configuring FCKEditor

When installing the FCKEditor from the download there is no working server language configured. So to get the FCKEditor working with PHP I needed to say I want to use PHP as the server language. When this is done the editor automatically works and you can upload files and images the way it should work. The project then should off course be placed into the folder of the webserver. The images folder should also have write permissions.

Since I wanted to have all the uploaded files to be placed in one and the same folder, I needed to change the configuration of the FCKEditor. Normally all file types will be separated into different directories. For example, images are placed in a different directory as movies will be. To change this I changed the upload configuration folders inside the PHP configuration.

These actions weren't described on the website of the FCKEditor. It was however easy to figure out how to change these parameters and to get it to work. This actually took less time then I first thought it would take.

To change the buttons within the editor I use the following configuration file.

```
FCKConfig.ToolbarSets["JackalToolbar"] = [

        ['Bold','Italic','Underline','StrikeThrough',

        '-','OrderedList','UnorderedList',

        '-','JustifyLeft','JustifyCenter','JustifyRight','JustifyFull',

        '-','Link','Unlink','Anchor',

        '-','Cut','Copy','PasteWord',

        '-','Image','Table','Rule','Smiley','SpecialChar',

        '-', 'TextColor'],

        ['FontFormat']

] ;

FCKConfig.ToolbarCanCollapse = false;
```

Using this configuration shows only the default buttons which make it look like an simple editor. The last line makes sure the toolbar cannot be folded in. If without this line the toolbar can be hidden. The reason for doing this is that I wanted to let the editor look like Microsoft Office Word.

8.2.5 Language

The final issue I had was choosing the language for the GUI. I could implement the entire application with an English backend, but since we are still in Holland I chose to create the GUI in Dutch.

But since the application is build with Flex adapting it to English wouldn't take very much time. Even the FCKEditor can be changed into the English version with little time. Actually, FCKEditor checks which language the browser is using and sets itself to that language. But if we want we can force the editor to use a specified language.

9 Extra information

In the past I have visited some of Adobe's meetings. June the 3th there was another meeting which was about the newest version of the Flex SDK (V4.0) and the newly called Flash Builder. At that time I was already half way of my project and figured out how to solve the problems with AMF.

However I think it's worthwhile to notice that in the newer version there is a possibility to generate PHP classes which should make it possible to have an active reference to the real objects. If this would work well all the difficulties I ran into with AMF would be obsolete.

How it should work is that you can then map the returning object which will come from the generated PHP class to an active Flex object. This would then automatically map the functions like save and setUsername to the appropriate function in the PHP classes.

Since I had already implemented this it would cost me a lot of extra time to implement this into the version and convert everything to Flex 4. Therefore I didn't do this. I do have some questions about this mapping but the persons who were present at the meeting couldn't answer those questions for me.

One of the most important things which I am curious about is how much requests will be send if we would update an entire user object and save this into the database. Would it be one for every parameter of the user or would it be one, just the save action?

I will take a look at this in the future and if there will be enough benefits I will convert the project into and Flex 4.0 project.

More information about the Flex 4.0 SDK can be found here:

http://labs.adobe.com/technologies/flex4sdk/

http://www.adobeusergroup.nl/site/list_messages/287

10 Conclusion

Inside the introduction I asked the question how we can develop Content Management Systems with Adobe Flex and the Zend Framework which are user friendly. Well, Flex offers a lot of features which can be used to create an user friendly interface. The two problems are to figure out which features the customers find user friendly and how to use them.

The survey showed that a lot of people thought it would be a great solution to use Drag’n’Drop for a Content Management System. Personally I also think this would be very nice if it could be working in a very easy way. But implementing Drag’n’Drop to be able to change text and images would result in an application which most probably would be unusable for the normal computer user.

Since the survey also showed that the users are familiar with the Control Panel of Microsoft windows and the menu structures of windows this would be a much better solutions to implement an Content Management System. Therefore this is also the solutions which I used for implementing it.

Effects are also included into the CMS because they give a nice look and feel and also can hide delays when loading data from the database. This also improves the user friendliness of the application.

All together I think the application works very nice and user friendly. Some parts like statistics aren’t implemented which spared a lot of time within the project. Unfortunately implementing and debugging the AMF protocol between Flex and PHP took a lot of time extra.

So the answer to the research question is: We can implement an user friendly CMS or presentation application by using the correct features of the Flex SDK in combination with the Zend Framework at the correct place. For example Drag’n’Drop and effects within the user system and the Flex tree for simulating a directory structure in the file managing system.

For the presentation facilities we can use all the Flex features because a presentation should look as fancy as possible.

Appendix I: Planning

Activity	<5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
		2-2	9-2	16-2	23-2	2-3	9-3	16-3	23-3	30-3	6-4	13-4	20-4	27-4	4-5	11-5	18-5	25-5	1-6	8-6	15-6	22-6	29-6	6-7	13-7	20-7	27-7	3-8	10-8	17-8
Initiation	🟦																													
Orientation	🟦	🟦	🟦																											
Literature Study		🟦	🟦	🟦	🟦																									
Powerpoint App						🟦	🟦	🟦																						
Powerpoint tutorial							🟦	🟦	🟦																					
CMS																														
Analyse									🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦														
Implementation															🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦					
Documentation																									🟦	🟦	🟦	🟦	🟦	🟦

Appendix II: Dictionary

AMF

Action Message Format which is a binary format based on the SOAP protocol but binary.

PHP

Hypertext Preprocessor is an server side html embedded scripting language.

CMS

Content Management System.

Drag'n'Drop

Dragging objects, for example files, to handle an action. Then drop them to confirm the action. For example dragging an file into the bin to delete it from the file system.

SDK

Software Development Kit. This make developing software easier because it contains defined functionalities.

WYSIWYG

What You See Is What You Get. This can be an editor which creates code but in a way that you don't need to know these codes to build it.

Appendix III: Literature

Books

Rob Allen, Nick Lo, 2007, Zend Framework in action, Manning Publications

Jeff Tapper, Matthem Boles, James Tabot, Benjamin Elmore, Michael Labriola, 2007, Adobe Flex 2 Training from the Source, Peachpit

Meetings

Flash Catalyst & Flash Builder, June 12th, Amsterdam, Pakhuis de Zwijger

Websites

http://nl.wikipedia.org/wiki/Watervalmethode
http://www.google.nl
http://en.wikipedia.org/wiki/Adobe_Flex
http://framework.zend.com/
http://www.adobe.com
http://www.fckeditor.net/
http://en.wikipedia.org/wiki/Action_Message_Format
http://www.charlesproxy.com/
http://examples.adobe.com/flex2/inproduct/sdk/flexstore/flexstore.html
http://opensource.adobe.com/wiki/display/flexsdk/Targeting+Flash+Player+10
http://farm1.static.flickr.com/202/440815933_e278ef856e_o.gif
http://yuml.me/diagram/scruffy/class/%5BIndex%20(First%20Parameter)%7Bbg:yellow%7D%5D-%3E%5BController%20(IndexController)%7Bbg:purple%7D%5D,%5BIndex%20(Second%20Parameter)%7Bbg:yellow%7D%5D-%3E%5BAction%20(indexAction)%7Bbg:purple%7D%5D

Appendix IV: Reflection

Flex and the Zend Framework

As the final part of my two years of studying at the Vrije Universiteit I started with my final project. I first looked for a project which has my special interests. I choose to do a project internally. Now at the end of the project a lot of things are a lot more clear to me than before.

The reason for this is that when I did my HBO education at the Hogeschool van Arnhem en Nijmegen I had two internships and with one of them I did an project which was outside its properties. To project was to build a system which would work for a company with more than ten employees. But the company only had three employees. After the project the system wasn't used at all. This was something which I didn't want to see again. Then I thought of working with Flex and Zend which has my interest for a while now. Now what would be a better project then build a custom Content Management System with the language you like?

So therefore I chose to build an CMS with the programming languages which have my interests. Unfortunately my laptop was a bit slow when working with this all together.

Assignment

I really enjoyed the assignment. I think Flex is a really good language to learn and works very well once you get familiar with it. Another thing I liked about this project that it walked me through the entire process of making a fully working application. I started with thinking about the assignment. To get a good idea about what customers of an CMS want I did a survey across random age groups. Then when I had an idea about what people really wanted I created the design for the system and the database. After I had a general idea about how I wanted to set up the system I started programming.

During the programming I ran into a lot of nice things and also a lot of problems which I had to solve. The most difficult problem was transferring data between Zend and Flex via AMF. Also this gave me a good change to improve my PHP skills. It has been a while that I had been working with PHP as server language.

I do think it's too bad that when I was working on the project the beta versions of the Flex SDK 4.0 was being developed. If this would have been a bit earlier I could have build this project with the latest version of Flex. But this would probably also given me a lot of other problems since a was a bit familiar with the 3.0 SDK and not with the newest version.

Knowledge

Adobe Flex

Before this project I have been working with Flex. In my opinion Flex is a very nice language to build very rich applications. If you want to build a normal application this is very easy with Flex. But when you want more than the default components Flex can become very difficult to work with.

I haven't worked on a project of this size with Flex before. I learned a lot about the effects of flex and how to arrange projects as they get bigger. I also noticed that if you want to upgrade the active project to work with a newer version of the Flex SDK that you might get yourself into a lot of problems.

I think Flex is a very good language to use for building applications like an CMS and I will be using Flex in the future.

Zend Framework

I have been working with some PHP frameworks before. One of them was the Symfony Framework and the other was the Zend Framework. Both are very good to work with but when you use a different frontend like Flex you can do a lot more with the Zend Framework. It is very easy to adapt to your own wishes and requirements.

The downside of the Zend Framework is that there isn't that much documentation and the documentation that is there might only be working with another version that the one you are using. But once you have a general idea about the working of the Zend Framework it gets easier and works like a charm.

Setting up the project does take a lot of time and requires you to think of it before doing it. One good example of the set up is the configuration files. Within these files you can define the most important information. Because changing it later might cost you a lot of time.

In my opinion the Zend Framework is very good for larger projects like this and works the way it should be.

AMF Protocol

The AMF protocol was completely new to me. But since the Zend Framework handles an AMF server exactly the same as an WSDL server this also wasn't very difficult. Within Flex the request is also exactly the same.

One big advantage of the AMF protocol is the speed. The speed is much faster than a normal WSDL request. This because the data is binary instead of XML. I think that if I would have used WSDL/SOAP for the data transfer that the system would become very slow. Therefore I am glad I have learned some things of AMF.

Rewrite modules

Getting the Zend Framework working on a web server requires you to put an .htaccess file into the root of the project. This file contains some rewrite lines which pass all the requests to the index file. I have noticed that working with these files is very difficult and that the files are very complex.

I am really glad the information which should be in the file is available on the internet and I hope that I don’t have to build any of these files on my own.