4 Design

In this chapter it will be explained all the design decisions taken while programming the application News Hunter. It will be described the final architecture of the program and explain in detail each one of the parts that form it.

4.1 System architecture

The program News Hunter is an online application that everybody that has an internet connection will be able to use so the architecture of the system has been divided in three big blocks. First of all, the web interface that will allow the user to use the News Hunter services. The other big block of the program will be the search engine that will be hosted in the server. The users won’t have any kind of interaction with this block. It will work automatically looking for the news and the preferences that the users decide. To connect these two big blocks there will be a third smaller part that will be the database. The database will contain all the information about the users and their newspapers.

The web interface will provide all the information about the users to the database while the search engine will just check that information in order to look for the news and decide which ones suits better the users’ preferences. When the news has been chosen, the search engine will store them in the database and send the newspapers to the users.

4.1.1 The web interface

In the next sections will be explained the design of the web interface that allow the users to use the News Hunter application.

4.1.1.1 The design

In the image below you can see the design of the website for the News Hunter application. According to these days, the design is simple but at the same time attractive for the users. The main objective was to create an interface easy to use and not ambiguous, where the user could easily find everything he wants and use without problems or learning previously how to use the website, all the services.
All the logical of the website has been programmed using PHP language. That gave the possibility to create a website easy to modify and expand if it's necessary. To do that it's been used the include function provided by the PHP libraries that allow to include another file in a PHP script. That way you can generate a page by including other files with HTML tags.

That system allow us to have a PHP file with the header of the website, another file with the menu, another file with the bottom of the page and then include all the files in the page that will only contain the body of the page. That system is very useful because a website is composed of different pages and if the menu, for example, is generated in each one of the pages, to change one option in the menu you would have to change all the pages that contains the menu. Using includes allow you to change only the menu file and all the pages that include that file will change automatically.

The different parts that are generated in separate files are the menu, the header of the website and the bottom. Those parts are generated in the files: head.php, menu.php, feu.php. In the image below you
can see an example of code using the include function with the three files that contains the different parts of the page.

```php
<?php

//DEFINES/
include_once('parts/defines.php');

$title="Welcome to the Ultimate Newsmastering eXperience";

/\HEAD
include(ABSOLUTA.'/parts/head.php');

/\MENU
include(ABSOLUTA.'/parts/menu.php');
?>

</div class="contenttext">
</div class="bodytext" style="padding:15px;" align="justify">
</strong>What is News Hunter</strong><br />
</div>
</div class="panel" align="justify">
</span class="orangetitle">Utility of News Hunter</span>
For example one company that want to send periodically news related with the activity of the company

to some workers, and probably add own news internals to the company, or maybe from some blog of

one of the chief...<br />
People that want to generate his own thematic news paper with news from a lot of different sources

but speaking about one specific theme and share it with some friends.<br />
People that wants to write his own news and make it public.<br />
People that is only interested in read news, but don't like to read only one newspaper, and don't like to

change the website...<br />
</span>
</div>
</div>

<?php

/*EU*/
include(ABSOLUTA.'/parts/peu.php');
?>
```

At the same time in the image is possible to see that there's another include of a file called defines.php. That file contains some defined variables in order to be used during the programming of the site. One example of a defined variable could be the word “ABSOLUTA” that we can see in the includes of the different parts of the page. That word has been defined as the absolute path where the website is hosted. That's very useful because if the website is hosted in one domain and one day the owner of the site decides to change the name of the domain, changing the path of the domain in the file defines.php you will change all the paths that can be found in the different pages that as includes or as link, for example.

The line to define a variable in PHP is something like the next one:

```php
define ("ABSOLUTA", "C:/wamp/www", true);
```
As it can be seen in the line, to create a defined variable is used the define function provided by the PHP libraries. In the first field of the function you decide the name of the variable. In the second field you write the value of that variable. The third field is to decide if that variable will be case-sensitive or not. Set at TRUE will mean that it’s not important if it’s written in capital letters or not.

4.1.1.2 Packages

The web interface can be divided in different packages depending on the functionalities that the users can use in the application. Those different packages are:

- Registration system
- Newspaper management system
- Mail sending system
- Captcha system
- Database module
- Other functionalities

4.1.1.2.1 Registration system

This package is the one that takes care about all the users’ information and allow the users to register in the system of News Hunter. At the same time, it will give to the user the possibility of modifying his information or resigning from the system.

Files from the registration system:

- Register.php

This file is the first step for the user to get registered in the News Hunter system. If a user wants to use the services of News Hunter it’s a requirement to get registered in the website in order to get his information. In the image below you can see the information asked to the user to get registered.
As you can see, the information asked for is very short because it’s not necessary to have a lot of the information of the users. The name and surname are just to identify the users but the real important fields will be the e-mail and the password. The e-mail must be unique in the News Hunter database. It means that one e-mail only can be one time registered in the database. That’s because the e-mail will be used for the user to login into the website as well as the password.

The four fields are obligatory to fill in so if the user forget to fill one, the website will give an error and will obligate the user to fill the missing field in order to finish the registration. Another error will be given if the e-mail introduced by the user is not a correct e-mail. The last security step will be the security code that the user has to copy from the security image. That’s important in order to be sure that the person who wants to get registered is a human person and not a robot. If the security code written is not the same in the image the registration won’t finish correctly and will give another error.

When all the information is correct the user just has to push the register button and the website will take him to the next step in the file register2.php.

- Register2.php

This file is a merely informative page for the user. When the user has introduced all the information correctly will be taken to this page where will be congratulate for the registration in the News Hunter system. At the same time this file sends an e-mail to the address that the user decided as a login. This e-mail contains a link to the News Hunter website with a random code
generated while the registration. When the user pushes the link in the e-mail the code in the link and the code generated are compared and if it’s the same the user will be confirmed in the system.

That’s useful in order to be sure that the e-mail introduced by the user exists and is in use at the moment. At the same time, the random generated code is necessary to be sure that the user that confirms the e-mail is the same that did the registration to the system. The e-mail sent to the user is shown in the image below.

Welcome to News Hunter

From: News Hunter (info.newshunter@gmail.com)
To: Dolor
Date: martes, 15 de enero de 2008 11:00:10

Welcome to News Hunter, to confirm your registration you have to click in this link or copy it and open it in your navigator.
127.0.0.1/registered.php?code=152294921&m=blindzick@hotmail.com

Another important thing that this file do is treat the information sent by the user with the PHP function trim(). That function trim whitespace (or other characters) from the beginning and end of a string. That’s useful to store always in the database the information in the same format and avoid the users to write strange thinkgs in the fields of the registration formulary. An example of code using this function is shown below.

```php
$family = trim($_POST['family']);
$name = trim($_POST['name']);
$mailU = trim($_POST['mail']);
$password = trim($_POST['pass']);
```

- Registered.php

This file is where the confirmation link in the e-mail sent will take the user when pushed. Is just a confirmation page that will give the opportunity to the user to create the newspaper or just go to the users profile menu.

If the random generated code and the code from the link are different this page will give an error, remembering the user to check it’s e-mail to push the link sent, so the registration won’t be finished correctly. If the registration is finished correctly, the user will be automatically logged in
the website so the menu in the left side of the website will be different, as you can see in the images below.

That’s possible because of the session variables. In order to maintain some information during the session, PHP provides the session variables. In order to store that session variables it’s necessary to use the PHP function session_start() at the beginning of all the files where you want to maintain the session information. In News Hunter website are used 3 different session variables:

$_SESSION['dintre'] -> Is the session variable that allows the system to know if the user is logged in or not. If the value is 1, it means that the user is logged in, otherwise the value will be 0 or NULL.

$_SESSION['user'] -> This variable contains the e-mail of the user logged in. It is useful when creating newspapers or just consulting information because the website knows all the time which user is asking for it.

$_SESSION['error'] -> This variable is to know which kind of error has been produced depending on the numeric value that it has assigned.
When the user is correctly registered in the News Hunter system he can enjoy all the functionalities that the website gives to him.

- **Modify.php**
The user will have the chance in each moment to modify the information introduced while the registration process. This page will show to the user that information and will allow him to change it. The file load the data from the user checking the variable $SESSION['user'] and looking for it in the database to show it in the page as is shown in the image below.

![Modify your profile](image)

After saving the changes, the website takes the user to the confirmation of modification page.

- **Modify2.php**
In this page the website confirms to the user that the information he changed has been successfully updated. The file checks that the data is correct and modify the information in the database. If there’s something wrong in the modified information the page shows an error to the user. Another e-mail of confirmation is sent if the user decided to change the e-mail information.

- **Resign.php**
If the user arrives at this page it means that he wants to resign from the News Hunter system and give up using the services that the application can offer. In resign.php the website will ask to the user for a confirmation of resign. If the user pushes the no button he will be taken to the users menu again. If the user decide to resign and pushes the yes button it will take the user to the next step of the resign process.
- Resign2.php
  This page is just a confirmation to the user that won’t be able to use the News Hunter services anymore. The page will logout automatically the user and the PHP logic of the file will remove all the information about the user stored in the database including all the related information about the newspapers.

- Profile.php
  This page is the users menu. From here the user will be able to use the services of News Hunter like all the related things about the newspapers and all the related things about his profile.

4.1.1.2.2 Newspaper management system

This is the package in charge of the generation of the newspapers for the users and of showing their historical of newspapers. As it was explained before, the user of News Hunter will be able to create a custom newspaper giving the rules to create it and the sources where to take the news. He will be also able to choose the format of the newspaper and the periodicity of it and at the same time, create a list of e-mails to send the publication of the newspaper to.

This functionality will allow the user to see all the historical of newspaper since he started using News Hunter classified by date of publication and shown in HTML format in the website. Another
functionality that the user will be able to use is to add an delete news to the actual publication of the newspaper before it is published.

The files that of the newspaper management system are:

- **Management.php**
  This page shows the main menu of the newspaper management system. From this page you can access to the newspaper functionalities that News Hunter gives to the users. The main particularity of the page is that if the user has created a newspaper the menu will give the opportunity to modify or remove it, as well as add or delete news from the actual newspaper before its publication, and if the user has no newspaper created will give only the opportunity to create a new one. The two options will give the opportunity to the user to consult the historical of newspapers as is shown in the images below.

That’s possible using a created function in the database module that checks if the user logged in News Hunter has or not newspaper created. Depending on the link pushed by the user, the website will take him to the next step of the process.

- **Step1.php**
  When the user decides to create a new newspaper the website will take him to the first step of the generation. That step is shown in this file that asks to the user the first information about the newspaper he’s going to create.
That information consists in giving a name to the newspaper and a description for it. At the same time, in this first step the user can decide in which frequency he and his colleagues will receive the newspaper. The possible frequencies to choose are: daily, weekly or monthly.

When the information is introduced, the user will go to the next step of the newspaper generation. It's not allowed to leave those fields in blank.

- Step2.php

After giving a name, a description and a frequency, the user is taken to this page. This file will check if the information introduced in the first step is correct and if there's a problem with that will take the user again to the first step to allow him to correct the mistake.

If the information is correct, the page will give to the user two more options to customize the newspaper. The first one is decide in which format the user wants News Hunter to generate the newspaper. The two possibilities are PDF file or HTML format. Depending on the format chosen bye the user, the newspaper will be sent to the e-mails in one way or in the other.

The second option given to the user is to create a list of e-mails, as long as possible, to who News Hunter will send the newspaper generated in the frequency chosen before.
After introducing that information, the user will be taken to the third step of the generation if the information is correct.

- Step3.php

This third step is the most important in the generation of the custom newspaper. It will give to the user the possibility to generate the rules that will decide in the future, which news suits the newspaper and which not.

First of all, the file checks if there were some errors in the list of e-mails given by the user in the step before. It’s not possible to find an error in the choice of the format of the newspaper because from the beginning the website gives the HTML option as default and then it’s not possible not to choose the format. If there are errors in the list of e-mails, the user is taken to the second step again in order to mend the possible mistakes.
If everything is correct in the second step the user will be able to write a list of words with a value that goes from “Automatic accepts” to “Not important”. Depending on the importance of the word, the user will give a value to it and then the search engine of News Hunter will use those rules to decide which news are important or not for the user. As an important thing to say, the words that have the value “Automatic accepts” will mean that if a new contains that word, it will be automatically accepted to be in the newspaper. An example of list is shown in the image below.

![News Hunter Create newspaper: step 3](image)

When the user has decided a list of word that will be used a rules to choose the news for News Hunter, is sent to the next step of the generation. With that system of rules is difficult to generate a perfect list the first time you create the newspaper but the user will have the chance of modify and enlarge the list of rules in order to improve the efficiency of the search engine of News Hunter.

- Step4.php

This fourth step will give to the user the opportunity of choose from which sources he wants to obtain the news for his custom newspaper. First of all, the file will check if all the rules introduced by the user in the step before has a value and a correct word. If there’s any problem with the list of words the user will be taken again to the third step to mend the possible mistakes.
If the list of the third step is correct the website will give in this fourth step a list of default sources that the News Hunter database contains stored. The user will be able to choose as many sources as he wants from this list and at the same time, to add own sources by his own.

When the user finishes generating the list of sources, all the information necessary to create the newspaper has been introduced so the user will be sent to the last step of the newspaper generation.

- **Step5.php**
  That's the last step in the newspaper creation for the user. If the list of sources is not correct, this file will send the user again to the fourth step to mend the errors. If everything is correct it will mean that the newspaper creation is now possible and an information page will be shown to the user confirming the successfully creation of the newspaper.
At the same time, the script of the file will do different process to store all the information in the database and allow the creation of newspaper for that user start. First of all, the script will send an e-mail to each one of the e-mails from the list given by the user. The user will also be informed in the information page that those e-mails were sent, as you can see in the last image. That e-mail is very similar to the registration e-mail sent to the user to confirm that the e-mail is in use and its functionality is exactly the same, be sure that the users from the list of e-mails are using that e-mail and avoid possible spamming generated from our website. Below you can see the sent e-mail.

**Newspaper confirmation**

![Newspaper confirmation](image)

Welcome to News Hunter, Francesc added you to receive a newspaper from our website. To accept the newspaper you have to click in this link or copy it and open it in your navigator.

127.0.0.1/newspaper/confirmation.php?code=15&m=blindzick@gmail.com

Another process that the script will do is to store all the information introduced by the user about the newspaper into the News Hunter database. To do that complex insertion it will be needed
several inserts and relations created in the tables of the database, but the structure of it will be explained in next sections.

The last important thing that the script will do is create in the server the different folders where the newspapers files will be stored. Exactly, it will generate a folder with a name that will be a combination of the id of the user plus three random generated numbers. That will be useful to not allow easily other users to check the newspapers of that user. Inside of this folder it will be created three more folders. One called “html”, another one called “pdf” and a last one called “rss”. The “html” and “pdf” folders will contain the actual version of the newspaper depending on the format chosen by the user. The “rss” folder will contain all the newspapers generated until that moment by the News Hunter application for that user in order to show the historical when required.

After all that processes, the newspaper will be created correctly and the user will start enjoying the publication if it with the frequency chosen before.

- Confirmation.php

When the people that receive the e-mail to be included into the list of e-mails to send the newspaper clicks in the link sent, the website will take them to this page. Another comparison between the code and the e-mail will be done and if everything is correct the e-mail of that person will be confirmed.

If the code and the e-mail don’t correspond to the appropriate values a message of error will be shown to the user recommending checking the email link again.

- Historical.php

This is the file that makes a list of all the newspapers generated until the moment by News Hunter. The script opens the folder where all the rss of the newspaper are stored and make a list of all of them ordered by date of creation. The user only has to push the button on the desired date and a page with all the news in HTML will be open.
• Transformar.php

When the user chooses one newspaper from the list of the historical of newspapers the website takes the user to this script giving the direction of the rss file that has to be opened. The script takes that file and shows it to the user giving the format designed in the file "plantilla.xslt". That way is really easy to change the format that you want to show the news. The only thing to do is change the "plantilla.xslt" file. The image below is an example of how can be shown one newspaper.
### News Hunter

#### Newspaper

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Putin chooses nationalist as Russian ambassador to NATO</td>
<td>The appointment of an outspoken nationalist is the latest reflection of Putin's assertive stance toward the West.</td>
</tr>
<tr>
<td>British government approves new nuclear power plants</td>
<td>The announcement signals a new enthusiasm for atomic energy after years in which the industry stagnated in much of Europe.</td>
</tr>
<tr>
<td>Georgia charges opposition tycoon with coup plot</td>
<td>Georgian prosecutors on Thursday charged opposition politician and tycoon Badri Patarkatsishvili, in self-imposed exile in Israel and Britain, with plotting a coup and attempting to organize a terrorist attack.</td>
</tr>
<tr>
<td>New U.S. president won't fulfill all Europeans' desires</td>
<td>The Europeans may be clutching at straws if they expect new leadership in the White House to herald a fundamental shift in the trans-Atlantic relationship.</td>
</tr>
</tbody>
</table>
| Kosovo leader promises independence in weeks | |}

- **Delete.php**
  
  When the user decides to delete the newspaper he created will arrive at this script. This page will ask for the confirmation that the newspaper has to be deleted. If the user decides that at the end he wants to maintain the newspaper and pushes the no button will return back to the newspaper management menu.

  If on the contrary, he decides to delete the newspaper completely the website will send him to the next step of the deleting process.

- **Deleted.php**
  
  This page will confirm to the user that the newspaper has been successfully deleted from the database but before, the script will remove from the database all the information related to the newspaper stored in the tables. The historical of newspapers is not deleted to allow the user to check the old versions every time he wants.
• Modify1.php
The modification process of the newspaper is really similar to the creation process. The only
difference is that where in the creation process the information of the newspaper appeared empty,
here will appear with the information added during the creation process. That’s why it won’t be so
extensive the explanation of the whole files during the process.

In this first step of modification the user will be able to change the name, the description and the
frequency of the project. The user will have the chance to change that information or not.

• Modify2.php
In that second step will be checked if the information sent from the first step is correct. If not, the
user will be sent to the first step again. If everything is correct, the user will be able to decide again
in which format want the newspaper to be published, HTML format or PDF file format. At the
same time it will be possible to modify the list of e-mails to send the newspapers to.

• Modify3.php
In the third step it will be checked if there are errors in the new list of e-mails given. When
everything is correct the website will show all the actual rules of the newspaper and will give to the
user the possibility to add more rules or delete the existing ones.

• Modify4.php
In the fourth step of the modification process the script will check if the information about the rules
is correct. If there are no problems the website will show to the user the list of sources where to
take the news from. The user again will be able to add more sources or remove the existing ones.

• Modify5.php
In the last step, and if all the information coming from the last step is correct, the script will send
again the confirmation e-mail to the news e-mails of the list (if there are). Then will update all the
information stored in the database and will show to the user a message of confirmation that
everything has been ok.

• Adddel.php
When the user wanted to check the news added until that moment to the next newspaper publication will arrive at this page. This script will show the list of news added until that moment to the next publication giving the opportunity to the user to delete the ones that are not interesting for him or adding new news written by the user using a formulary.

- Adddel2.php
  This script will remove the new chosen by the user from the next publication of the newspaper or add a new written by the user. After doing one of the two things, the page will automatically take the user again to the list of news giving again the opportunity to add or remove news.

4.1.1.2.3 Mail sending system

In order to send the automatic e-mails from News Hunter website we looked for an existing library. It was chosen PHPMailer, a PHP written class that makes easy the sending of emails in HTML format, with personalised headers and working with multiple mail servers.

To use this class we only had to include the file class.phpmailer.php and generate a PHPMailer object in order to provide to it all the information to send the e-mail and use the send() method existing in the class. Below you can see an example of function created send emails with an HTML message using the PHPMailer class.
4.1.1.2.4 Captcha system

The word CAPTCHA means Completely Automated Public Turing test to tell Computers and Humans Apart. A common type of CAPTCHA requires that the user type the letters of a distorted image, sometimes with the addition of an obscured sequence of letters or digits that appears on the screen.

As it was explained before, that system has been used in the registration system to be sure that the one was getting registered was a human person. In order to use a CAPTCHA system in the website it was looked for a created PHP library. The file used is CaptchaSecurityImages.php. That file generates an image with a code and at the same time creates a session variable with that code. After that, you only have to use HTML tags to show the image and allow the user to write the code. When the user write
the code, it is compared with the code of the session variable and if it’s not the same, the system will give an error.

The HTML tags to show the image and allow the user to write the code are the next ones:

```
     valign="MIDDLE" align="MIDDLE" alt="captcha" />
<input id="security_code" name="security_code" type="text" style="width:127px" class="ingran"/>
```

Where `img` shows the image generated by the script CaptchaSecurityImages.php and the `input` allows the user to write the code. After that, as it was explained before, you only have to compare that code with the assigned by the script into the session variable.

4.1.1.2.5 Database module

The database module is a file where you can find all the functions related with storing and getting information from the database. All that functions will be needed during the different possible processes to do with the News Hunter website. There is too a function that allow the website to connect with the database in order to do all the transactions.

The name of the file is bd.php and the functions will contain different types of SQL language queries because the database system is mySQL.

4.1.1.2.6 Other functionalities

To make News Hunter a complete website it has to have other typical functionalities that help the users to know more about the creators of the project or gives facilities to use the website. At the same time, always there’s a possibility of improving the job done so there’s the possibility of receiving the users’ advices or complaining about the use and working o the application.
The different files that complete the website are the following:

- Contact.php

Every website requires the contact functionality in order to allow the users to help the administrators of the site to improve the services or to complain about them. In the image below you can see the formulary to contact with the administration of News Hunter.

As you can see, the only requirement is to write the name, decide a topic, give an e-mail of contact and write the message that the user want to send.

The possible topic messages are:

- About how News Hunter works
- Problems with the registration
- Suggestions
- Love declarations to the staff

It’s not necessary to be logged in to News Hunter to contact the administrators. That’s because maybe there can be users that have problems with the registration process, for example. At the same time, the number of topics is limited because we don’t think that is necessary another kind of
contact to improve the website. When the user pushes the send button the website takes him to the next step of the contact.

- **Contact2.php**
  This file treats the information sent by the user in order to check that everything is correct. If there’s some errors on the data or the email is not correct it will be shown a page explaining the error and giving the chance to correct it.

  If all the information is correct, the page will show a message of confirmation to the user and the logic of the file will send an e-mail with all the information to the gmail account created to administrate the website.

- **About.php**
  This page is merely informative and explains what’s exactly News Hunter and it’s functionalities. At the same time explains to the user which has been the motivation of the creators to undertake this project.

- **Sitemap.php**
  The map of the site of the site page gives all the possible functionalities and links together in one page in order to allow the user to find easily what he is looking for. At the same time gives an structure of the website so is an easy way of having a first contact with News Hunter and know about it’s services.

- **Us.php**
  This page is also informative and pretends to bring nearer the users to the creators of the website explaining a little bit who they are, what are they doing in life and which are their motivations and next projects.

- **Index.php**
  As the file name shows, that’s the main page of the website. It’s if the first contact with News Hunter so here it’s explained briefly what News Hunter is and why it can be useful.
4.1.2 The database

The database used to store all the information about the users and the newspapers is a MySQL 5.0.45 database. This SQL database management system is a free software system and one of the most used nowadays as databases, especially while using PHP websites so it was a good solution according to this project.

4.1.2.1 Administration

In order to administrate the News Hunter database it has been used phpMyAdmin. PhpMyAdmin is a tool written in PHP intended to handle the administration of MySQL databases. With this tool it was possible to create all the tables with the fields that at the end will store all the information related with the users of News Hunter and their newspapers.

4.1.2.2 Design

The database used to administrate the information of the News Hunter application is formed by six tables with different fields. In this section it will be explained each table and the fields that compose them.

The six tables that compose the News Hunter database are:

- Users
- News
- Emails
- Rules
- Sources
- Relusersources
That table is the one that will contain all the information related with the user and some information about the newspaper of each user. In the image below you can see the structure of this table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surname</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pass</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirmed</td>
<td>binary(1)</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegistrationDate</td>
<td>datetime</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastConnection</td>
<td>datetime</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>int(10)</td>
<td></td>
<td>No</td>
<td></td>
<td>auto_incre</td>
<td></td>
</tr>
<tr>
<td>codigo</td>
<td>varchar(10)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>varchar(5)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NextDelivery</td>
<td>datetime</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastDelivery</td>
<td>datetime</td>
<td></td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NameNewspaper</td>
<td>varchar(200)</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DescriptionNewspaper</td>
<td>text</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Htul</td>
<td>tinyint(1)</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pdf</td>
<td>tinyint(1)</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The fields Name, Surname, Mail and Pass are the fields that contain the information introduced by the user when he is registered. Confirmed is the field that tells if the e-mail of the user has been confirmed. Id is the field that will identify the user and is assigned automatically by the database. The field codigo contains the random code generated to check if the person that confirmed the e-mail of the user is the one that received the e-mail of confirmation. At the same time, the three first numbers of this field are used, with the id, to generate the folders for the user where the newspaper files will be stored.

The fields RegistrationDate and LastConnection are not used at the moment but can be useful to generate statistics about the user and his using of the application.
The rest of the fields are storing information about the newspaper of that user. Meanwhile the user don’t create a newspaper those fields will be set at NULL. When the user generates the newspaper, the information introduced by the user will be stored in the appropriated field.

4.1.2.2.2  News

This table is the one that will contain the news selected by the News Hunter application. The image below shows the fields that compose this table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>User</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>text</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>text</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link</td>
<td>text</td>
<td>latin1_swedish_ci</td>
<td>Yes</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateFound</td>
<td>datetime</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selected</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tractada</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td>No</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The field Id identifies the new and is assigned automatically by the database. The field User contains the user to who is addressed that new. The field Source contains the Id of the source where the new was found from. The Title contains the title of the new as it was written in the rss file. The same happens with the description and the Link. The DateFound field contains the date when the new was found. The score contains the sum of the scores of the rules founded in that new. The field Rules contain which rules are found in that new. The field Tractada will be set if the new is sent to the newspaper. That way if a new is deleted it can’t be selected again if the field Tractada is set. The field selected will be set if the new is selected as interesting.
4.1.2.3 Emails

This table contains the e-mails to whom the newspapers had to be sent. In the image below you can see the structure of the table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>varchar(255)</td>
<td>latin1_spanish_ci</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm</td>
<td>varchar(2)</td>
<td>latin1_spanish_ci</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The field User contains the Id of the user that added that e-mail to the list of e-mails to send the newspaper. The field e-mail contains the e-mails to whom the system has to send the newspapers. The field Confirm says if that e-mail has been confirmed or not.

When the newspaper of a user has to be sent, the application will check the list of e-mails of that user with the Id and then will send the newspaper only to the e-mails that has been confirmed.

4.1.2.4 Rules

The table rules will contain the rules created by the users while the generation of the newspaper. In the image below you can see the structure of that table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td>auto_increment</td>
</tr>
<tr>
<td>UserId</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td>int(11)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word</td>
<td>varchar(250)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accept</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All the rules will be identified with the Id, assigned automatically by the database. The rules will contain too the user id assigned in the field UserId and the score given by the user. The field word will contain the word that will be searched in the news and if the word has the field AutomaticAccept at 1 it will mean that the news that contains that word will be accepted automatically.

4.1.2.2.5 Sources

This table is the one that contains the sources where to look for the news. In the image below you can see the structure of that table.

The field Id is assigned automatically by the database and identifies the source. The field Nom contains the name of the source while the field URL contains the internet address of it. The field Public will be used to indicate which sources are provided by the system and which ones had been introduced by the user.

4.1.2.2.6 Reluserssources

This table is the one that contains the relation between the users and the sources used to look for the news for the newspaper. In the image below you can see the structure of that table.
The user Id contains the Id of the user that uses the source source Id to look for the news for his newspaper.

4.1.3 News Hunter application

The program hunter is the part of the software that should be working all the time to check all the sources that our users put in their newspapers, use that news contained in the different rss feeds and pass the personal rules filter of each user assigning for each new a final score.

4.1.3.1 How the Hunter Program works

The main objective of the program is check every determined time interval (default 30 min) the different sources that process is called “Hunting process”, one we have all the rss feeds stored in our local disk, the program starts checking the different sources with the rules list of each user, checking every new with all the user list of rules and assigning a final score, we call that process as a “Match Filter” (explained with more detail later).

After the “Match Filter” we’ll know if the new shout be accepted or not, and we can store that information, but is important to say that the new is never deleted, it means that every user have inside the Database all the news published in all the sources that are from his interest, that’s because two important reasons, first to do not being all the time putting the same news to analyze because we don’t know if it’s a new new or if on the other side is a new already checked several times but that never passed the Match Filter, putting the new always in the Database we are only checking the news that are new in the database, it means that each new will be checked once with the efficiency that this method implies, for the second and more important new it’s important to have all the news stored because our program have an “Experience improve method”, that basically will try to make the selection better with the experience, (that method will be explained with more detail later).

After we have a list of all the news of all the users marked according if our filter considered them interesting or not the program starts the “Documents generation procedure”, which will check if it’s time to deliver the newspaper, it depends because we can set our newspaper to be generated every week, two weeks, month... and if the program discover that is time to regenerate a new edition of the
newspaper starts the procedure. That procedure will create a PDF document, rss document and HTML document. All that sort of documents will be stored in a personal folder of the user for fast access.

After the generation of all the required documents we are ready to send to all the email address that the user gave us (and are validated, answering to the mail that we have sent at the registration moment, to avoid spam attempts.) in that mail we will attach the document that the user ask us to sent in the registration process the PDF, HTML or the link to the html page in our server.

After that process the “Experience improve method” will start and the process will sleep for the time that we have set, after that sleep time the process will start, and that way over and over until we decide for some reason to stop the application.
Installing the Program

Opening & Checking DB

Opening & executing

Hunting Process

Match Filter

Document Generation Method

RSS

Using the rss we can generate(XSLT) HTML

HTML

Using the html we can generate(.NET Library) PDF

PDF

Only if it is time for deliver

Experince improve Method

Send

Only if it is time for the generation
4.1.3.1.1 Hunting process

In that process the Hunter Program is hunting the rss feeds from the external sources, the critical point here is to avoid be consider as an a hacker attack trying to prompt a Denial of service (DOS) or only flooding the servers making a huge quantity of documents demands. That could be possible because in our site we can have a lot of registered users, and each user can have as many sources as he want, in that context is easy to think that some determined popular sources can be used for a quite important quantity of users, and if we don’t make a fetch of their rss we can ask for the feed once for each user, it could mean easily miles of times, every half an hour (or the time that is set between executions), is easy to see that that process can easily be detected for some firewall as an attack or by the webmaster as a not desired traffic and be blocked.

For that reason the first procedure of the program is discover which are the feeds needed in all the process for all the users, and make a list to fetch them downloading first to our local hard drive save it there and use it all the times that we need. The program will create a folder in our main hard drive (usually c:) called rss, and in that place it will download all the feeds. With that procedure we can be safe because we are going to check all the sources every half an our at maximum, by the other side the after effect of that decision is that we’ll need to have some disk space to store it but if you think that we are speaking about files that have a size of 7Kb in average is easy to understand that won’t be a problem with the actual size and price of the disk space.

Assuming that fetching decision had implications in the Database design where we had a different table to store the sources, and every source is only stored once, using the numeric code to make different relations between users and sources. With that decision we are simplifying the process to know which sources are needed to fetch, because we will have to fetch all the items in that table, and when we download the rss we save it using the numeric code as a file name, making direct the posterior use for each user.

On the other hand that decision to put in a different table all the sources and store the relations between sources and users separately permits to have high performance on the hunting procedure which is executed every half an hour(default) and in the haunting program who will have a lot of work to do,
but we are putting more work in the registration process because we will have to check for each source if it's already in the database or not and put it, is easy to see that the time that will take is not important almost not perceptible, and we will do it only once at the registration process.

With the interface of the program we will be able to check in which step we are and which sources we have already download, using the log tab

And we will see the log's table.

Here we can see the date and time of each action, and at the beginning the four sources that we had downloaded in that execution, we can see that only after that download we start with the other parts of the process.
And finally after all that decision we can check that the algorithm to download the sources is very small and simple, in code and in time of execution.

```java
private void DownloadSources()
{
    string cadena = "";
    string url = "";
    string nom = "";
    int idSource;
    MySqlConnection BD = new MySqlConnection(conexion);
    MySqlCommand SQL = new MySqlCommand();
    progressBar1.Minimum = 0;
    progressBar1.Value = 0;
    progressBar1.Maximum = numSources;

    //ELIMINEM DIRECTORI
    if (Directory.Exists(rutaDescarrega))
    { Directory.Delete(rutaDescarrega, true); }
    //RECREEM DIRECTORI
    Directory.CreateDirectory(rutaDescarrega);

    //BUCLE DE DESCARREGA
    cadena = "SELECT * FROM Sources";
    SQL.Connection = BD;
    SQL.CommandText = cadena;
    BD.Open();
    MySqlDataReader lector = SQL.ExecuteReader();
    while (lector.Read())
    {
        url = Convert.ToString(lector["URL"]);
        idSource = Convert.ToInt32(lector["Id"]);
        nom = Convert.ToString(lector["Nom"]);
        XmlDocument arxiuXML = new XmlDocument();
        try
        {
            arxiuXML.Load(url);
            arxiuXML.Save(rutaDescarrega + "/" + idSource.ToString() + ".xml");
            WriteRich("Downloaded source: " + nom);
        }
        catch (Exception e)
        {
            WriteRich("Error Downloading source: " + url + " Problem: " + e.Message);
        }
        progressBar1.Increment(1);
        sources.Text = progressBar1.Value.ToString();
        progressBar1.Refresh();
    }
    lector.Close();
    BD.Close();
}
```
4.1.3.1.2 Match Filter

In that step the objective is filter the news that are relevant to the ones that aren’t relevant, for that reason we use a list of rules that every user have to filter. That rules could be create completely by the user or could be of our special templates with some user changes. That process we have to do it for each user and for every source and every new, it means that we will have to pass all the rules to every new for every user.

To understand that process is necessary to understand how the rules are made and how they work. The rules are made by a word or a group of words that if are found in a new we consider are relevant, it can be relevant to find the new interesting or the opposite relevant to find the new inappropriate.

For that reason we can think in a rule as a combination of a chain (word or more than a word) and a score, the score is a numeric value that will be used if the chain is found. The numeric value can be positive or negative, depending if it’s a word that makes the new more relevant or if the new give us the prove of that it won’t be interesting.

4.1.3.1.2.1 How use the score value

When we are using the rules for the new and we find a rule that we have to apply, it means we have found the chain, it’s time to use the score value, at the beginning of the process the new have score 0, and every time that we have a match with a rule we shout add a numeric number to that final score at the end the new will have a final score and comparing this score with border score (as default is set to 5) if the final score is higher that the border score we know that the new is interesting if not we will know that is not.

But the question is if we find that we have a match, we only add the rule score to the new score? We found after some experiments that the answer to this question shout be no, in order to improve the results.
And we found out that is not the same to find the chain in the description or in the title, obviously shout be more important in the title, and that is not the same to find it once that more times.

For that reason we used a distribution using the number of matches and places of the match.

![Found in Description](image)

That means that if we have a match in the Description of the new we will add 80%-145% of the rule score depending of the number of times that we found the match.

For example, if we have the word: Vilnius, and a score of 3 it means that if the new appears Vilnius twice we will add 3.6 to the new score.

Now let’s check the distribution that we use for the matches in the title of the new.
The score of the description and the title is independent, it means that following with the same example of before with the new with the chain: "Vilnius" and score: 3, if the word Vilnius appears once in the title we shout add 3 to the new score, because with one apparition we shout use 100% of the rule score value, for that reason after applying that rule the new will have $3.6 + 3 = 6.6$ of a partial score value. Means that thing that the new would be accepted? No, because we can have negative score. That change the final value of that score from 6.6 to 2 for example.

To improve the performance of that process and thinking that for being interesting that program could need a lot of different rules. We disposed the rules ordered by rule score ascending. Using that algorithm means that if in some point we reach the barrier score we can stop checking the next rules, because we will be sure no one of the following rules will decrease the partial new score. That means that only in the cases that we don’t find interesting a rule we’ll pass all the rules.

Now we can see how applying that simple method we can make the execution of all the program quite faster.

```csharp
private bool CheckRules(string title, string link, string description, List<rule> rules)
{
    float score = 0;
    float scorePartial = 0;
    int i = 0;
    bool accept = false;
    
    while (i < rules.Count && accept == false && score < scoreBarrier)
    {
        // Check each rule
        if (rule matches conditions)
        {
            scorePartial += rule.score;
            accept = true;
        }
    }
    return accept;
}
```
scorePartial = GiveScore(title, rules[i].keyword, norms[i].score, 2);

scorePartial += GiveScore(description, rules[i].keyword, rules[i].score, 1);
if (norms[i].automaticAccept && scorePartial > 0)
{
    accept = true;
}

score += scorePartial;
i++;

if(score >= scoreBarrier)
    accept = true;

return accept;

The method GiveScore return us a float with the score of applying a rule to a text that we will give, the last member of that method is the kind of the text, 1 for description, 2 for title.

4.1.3.1.3 Special Rules / Sources

Our system also permits to introduce some special rules or sources, that special rules or sources will permit the program an automatic accept of the new, or all the news of a source.

Will be easier to explain in it separately:

Rules:
In that case all the news that have a match, even only matching once, will be accepted, without checking the other rules and don’t caring about the final new score.
That can be very useful for some critical chain as the name of your company, a name of some people, or any other words that give us enough information to be sure that new will be interesting for us.

Sources:
Very often we will have a feed source that is enough specialized to be sure that all the news that appears in it is interesting. For example if we want to reed all the news about Google, we can have different rules to filter the most generals news feeders as times, nbc... but the Google rss, for sure we will find interesting all the news.
For that reason our program will allow to put a source as a automatic accept and all the news will pass without checking any rule.

4.1.3.2 Document Generation

When we have download and filtered the news is time to check if we shout make our next delivery of the newspaper, for example if the newspaper is set to have a new edition every week and we have done the last one a week before is time to generate the newspaper if don’t, easy we only jump that step waiting for a next execution of the program, and for sure in some execution of the program we will realize that is time to the generation.

The first document that we will generate is the rss, because using that document we will be able to generate all the other documents.

4.1.3.2.1 How to generate the rss

That point will be very simple, we have to chose all the news that we found interesting since the last newspaper and create and xml document writing the description, title, date...that we have in the Database, and after creating that xml document we shout mark all the news that we have used as used in the database to can differentiate in the future which news we have used and which ones not.

The news are never deleted to avoid using the same new twice or more in different editions. And to don’t filter more than one time the same new.

Example of a generated rss:

```xml
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title></title>
    <link>http://localhost/rss/10_317/2008_1_15.rss</link>
    <description></description>
    <language>en</language>
    <lastBuildDate>Tuesday, 15 of january 2008</lastBuildDate>
  </channel>
</rss>
```
<item>
  <title>Visual Trophies</title>
  <description>Among the homely staples of twentieth-century life that have been unceremoniously retired by the microchip revolution—the typewriter, the pressed-wax record, the card catalogue—the camera loaded with film has met a swift and stealthy end. Digital cameras look much like their analog predecessors, but the viewfinder is</description>
  <pubDate>18/12/2007 0:00:00</pubDate>
  <link>http://www.newsisfree.com/iclick/i,249263953,13276,f/</link>
</item>

<item>
  <title>3 million learner drivers# details lost, says Ruth Kelly</title>
  <description>The personal details of three million UK learner drivers have been lost in the American state of Iowa, the Government announced tonight.</description>
  <pubDate>18/12/2007 0:00:00</pubDate>
  <link>http://www.timesonline.co.uk/tol/news/politics/article3064090.ece#cid=OTC-RSS&attr=2015164</link>
</item>

<item>
  <title>Fraud claims made as S.Korea votes</title>
  <description>South Korean voters Wednesday are choosing between a presidential frontrunner who faces renewed fraud allegations and two opponents who have failed to capture the public’s imagination.</description>
  <pubDate>19/12/2007 0:00:00</pubDate>
</item>

<item>
  <title>Turkish troops end Iraq incursion</title>
  <description>Turkish troops that entered Iraq early Tuesday have returned to the Turkish side of the border, according to a regional government spokesman. He said the incursion involved about 300 troops in and lasted about 13 hours.</description>
  <pubDate>19/12/2007 0:00:00</pubDate>
</item>

</channel>
</rss>
After the generation of a new rss the program saves a copy in the historic folder with the date as a name of the file, to make easier consulting and listing the historical archive of rss, and a copy in the public directory called "last_news.rss", over writing the previous version if that existed.

The idea to make that is to allow our own newspaper to be used as a feed, for example in some news reader, Google homepage or wherever.

With that method we allow a rss generated by News Hunter being a source for another newspaper generated for News Hunter.

4.1.3.2.2 How to generate the HTML document

Using the generated rss we can automatically generate the html document using XSLT, that is a method that with a rss, a style sheet CSS and a document XSLT explaining how show information should be showed generates an HTML, that system offer us a big amount of improves:

1. - Separation between design and structure.
2. - Separation of the procedure of generated HTML from our program.
3. - Possibility to offer different newspaper design to chose very easily
4. - Generation of the PDF automatically
5. - Easy to implement

To understand how the XSLT works is necessary to read the following chapter.

4.1.3.2.3 How to generate the PDF document

We could use XSLT, to generate the PDF document, but after some exhausting tries with different solutions we always found some library problems for that reason we decided to change the strategy and use a library that transform the html document to PDF, with that solution we keep the benefits of the different templates and a design independent of the data, because we are using the html previously generated which at the same time was generated by XSLT, we can say that we inherit the benefits.
For that purpose to generate the PDF using the HTML page as a source we decided to use the .NET library wnvhtmlconverter which allows us exactly and quite accurately to do this.

4.1.3.2.4 HTML to PDF Converter for .NET 2.0

The HTML to PDF Converter library for .NET can be linked into any .NET application, either ASP.NET web sites or Windows Forms applications, to add html to pdf conversion capabilities to your application.

You can use the HTML to PDF Converter for .NET as general purpose tool for converting web pages and HTML code to PDF and images or you can use it as part of our .NET Reporting Toolkit to easily create PDF reports directly from ASP.NET pages and to benefit from the great power of standard ASP.NET databound server controls.

The integration is extremely easy and no additional installation is necessary in order to get started. It's just a .NET 2.0 assembly that you have to reference in your application. The pdf document resulted after the conversion preserves the exact aspect of the source HTML page.

4.1.3.3 What is XSLT

Extensible Style sheet Language Transformations (XSLT) is an XML-based language used for the transformation of XML documents into other XML or "human-readable" documents. The original document is not changed; rather, a new document is created based on the content of an existing one.[2] The new document may be serialized (output) by the processor in standard XML syntax or in another format, such as HTML or plain text.[3] XSLT is most often used to convert data between different XML schemas or to convert XML data into HTML or XHTML documents for web pages, creating a dynamic web page, or into an intermediate XML format that can be converted to PDF documents.

As a language, XSLT is influenced by functional languages,[4] and by text-based pattern matching languages like SNOBOL and awk. Its most direct predecessor was DSSSL, a language that performed the same function for SGML that XSLT performs for XML. XSLT can also be considered as a template processor.
XSLT is Turing complete.

4.1.3.3.1 Overview

The XSLT processing model involves:

- one or more XML source documents;
- one or more XSLT style sheet modules;
- the XSLT template processing engine (the processor); and
- one or more result documents.

The XSLT processor ordinarily takes two input documents – an XML source document, and an XSLT style sheet – and produces an output document. The XSLT style sheet contains the XSLT program text (or ‘source code’ in other languages) and is itself an XML document. It describes a collection of template rules: instructions and other directives that guide the processor in the production of the output document.

4.1.3.3.2 The role of XSLT

XSLT has its origins in the aspiration to separate information content from presentation on the Web. HTML, as originally defined, achieved a degree of device independence by defining presentation in
terms of abstractions such as paragraphs, emphasis, and numbered lists. As the Web became more commercial, publishers wanted the same control over quality of output that they had with the printed medium. This gradually led to an increasing use of concrete presentation controls such as explicit fonts and absolute positioning of material on the page. The unfortunate but entirely predictable side effect was that it became increasingly difficult to deliver the same content to alternative devices such as digital TV sets and WAP phones (repurposing in the jargon of the publishing trade).

Drawing on experience with SGML in the print publishing world, XML was defined early in 1998 as a markup language to represent structured content independent of its presentation. Unlike HTML, which uses a fixed set of concepts (such as paragraphs, lists, and tables), the tags used in XML markup are entirely user defined, and the intention is that they should relate to objects in the domain of interest (such as people, places, prices, and dates). Whereas the elements in HTML are essentially typographic (albeit at a level of abstraction), the aim with XML is that the elements should describe real-world objects. For example, Listing 1 shows an XML document representing the results of a soccer tournament.

Listing 1. An XML document representing the results of a soccer tournament

<br>
<results group="A">
  <match>
    <date>10-Jun-1998</date>
    <team score="2">Brazil</team>
    <team score="1">Scotland</team>
  </match>
  <match>
    <date>10-Jun-1998</date>
    <team score="2">Morocco</team>
    <team score="2">Norway</team>
  </match>
  <match>
    <date>16-Jun-1998</date>
    <team score="1">Scotland</team>
    <team score="1">Norway</team>
  </match>
  <match>
    <date>16-Jun-1998</date>
    <team score="3">Brazil</team>
    <team score="0">Morocco</team>
  </match>
  <match>
    <date>23-Jun-1998</date>
    <team score="1">Brazil</team>
    <team score="2">Norway</team>
  </match>
  <match>
    <date>23-Jun-1998</date>
    <team score="0">Scotland</team>
  </match>
</results>
If you want to display these soccer results through a Web browser, you can't expect the system to come up with a sensible layout. Some other mechanism is needed to tell the system how to display the data on a browser screen, a TV set, a WAP phone, or indeed on paper. This is where the style sheet comes in. The style sheet is a declarative set of rules that defines how information elements identified by tags in the source document should be rendered.

The W3C has defined two families of style sheet standards. The first, known as CSS (Cascading Style Sheets), is widely used with HTML, though it can also be used with XML. CSS can be used, for example, to say that when displaying an invoice, the total amount payable should be shown in 16 point Helvetica bold. However, CSS has no ability to perform computations, to rearrange or sort the data, to combine data from multiple sources, or to personalize what is displayed according to characteristics of the user or session. In the case of our soccer results, CSS (even the latest version, CSS2, which is not yet fully implemented in products) is not a powerful enough language to handle the task. For these reasons W3C embarked on the development of a richer style sheet language to be known as XSL (Extensible Stylesheet Language), taking many of the intellectual ideas from DSSSL (Document Style, Semantics, and Specification Language), as developed in the SGML community.

During the development of XSL (and this had already been foreshadowed in DSSSL), it emerged that the tasks to be performed in preparing an XML document for display could be split into two stages: transformation and formatting. Transformation is a process of converting one XML document (or its in-memory representation) into another. Formatting is the process of converting the transformed tree structure into a two-dimensional graphical representation, or perhaps a one-dimensional audio stream. XSLT was developed as a language to control the first stage, transformation. Development of the second stage, formatting, is work still in progress. But in practice, most people are currently using XSLT to transform XML documents into HTML, and using an HTML browser as the formatting engine. This is possible because, for all intents and purposes, HTML is just one example of an XML vocabulary, and XSLT is capable of using any XML vocabulary as its target.

Separating transformation into one language and formatting into another proved to be a really good decision, because it turned out that there are lots of applications for a transformation language that
have nothing to do with displaying documents to the user. As XML becomes more widely used as a data interchange syntax in electronic business, there is an increasing need for applications to convert data from one XML vocabulary to another. For example, an application might extract details of TV programs from an electronic program guide and insert them into a monthly bill for a pay-per-view customer. Equally, there are many useful data transformations in which the source and target vocabularies are the same. These include data filtering, as well as business operations such as applying a price increase. Increasingly therefore, as data starts to flow around the system in XML syntax, XSLT starts to become a ubiquitous high-level language for manipulating it.

In my book I make the case that XSLT is to XML what SQL is to tabular data. The relational model gets its power not from the idea of storing data in tables, but from the high-level data manipulation possible in SQL, based on the relational calculus. Equally, the hierarchic data model of XML in itself does very little to help the application developer. It is XSLT as a high-level manipulation language for XML data that provides the power.

4.1.3.3.3 Processor implementations

XSLT processor implementations fall into two main categories: server-side, and client-side.

Although client-side XSLT processing has been available in Microsoft's Internet Explorer since 1999 (or even earlier, but in a form that was incompatible with the W3C specifications), adoption has been slower because of the widespread deployment of older and alternative browsers without XSLT support. For similar reasons, adoption of XSLT 2.0 in such environments is likely to be some years away.

XSLT processors may be delivered as standalone products, or as components of other software including web browsers, application servers, frameworks such as Java and .NET, or even operating systems. For example, Windows XP comes with the MSXML3 library, which includes an XSLT processor. Earlier versions may be upgraded and there are many alternatives, see the External Links section.

4.1.3.3.4 Performance

The performance of XSLT processors has steadily improved as the technology has become more mature, although the very first processor, James Clark's xt, was unbeaten for several years.
Most of the earlier XSLT processors were interpreters; in more recent products, code generation is increasingly common, using portable intermediate languages such as Java bytecode or .NET Common Intermediate Language as the target. However, even the interpretive products generally offer separate analysis and execution phases, allowing an optimized expression tree to be created in memory and reused to perform multiple transformations: this gives substantial performance benefits in online publishing applications where the same transformation is applied many times per second to different source documents. [10] This separation is reflected in the design of XSLT processing APIs such as JAXP (Java API for XML Processing).

Early XSLT processors had very few optimizations; style sheet documents were read into Document Object Models and the processor would act on them directly. XPath engines were also not optimized. Increasingly, however, XSLT processors use the kind of optimization techniques found in functional programming languages and database query languages, notably static rewriting of the expression tree for example to move calculations out of loops, and lazy pipelined evaluation to reduce the use of memory for intermediate results and allow "early exit" when the processor can evaluate an expression such as following-sibling::*[1] without a complete evaluation of all subexpressions. Many processors also use tree representations that are much more efficient (in both space and time) than general purpose DOM implementations.

4.1.3.3.5 Benefits of XSLT

Why should you consider using XSLT?

XSLT gives you all the traditional benefits of a high-level declarative programming language, specialized to the task of transforming XML documents.

The usual benefit cited for higher-level languages is development productivity. But in truth, the real value comes from potential for change. An XSLT application for transforming XML data structures can be made much more resilient to changes in the details of the XML documents than a procedural application coded using the low-level DOM and SAX interfaces. In the database world this feature is known as data independence, and it was the quest for data independence that led to the success of declarative languages like SQL and the demise of the older navigational data access languages. I firmly believe the same will happen in the XML world.
As with all declarative languages, there is a performance penalty. But for the vast majority of applications, the performance of today's XSLT processors is already good enough to meet the application requirements, and it is getting better. In my second article, I'll discuss the kind of optimization techniques that are being used in XSLT processors such as my own Saxon product.

4.1.3.4 Experience improve method

The last method is optional for the users but is highly recommended to improve the hunting. This method tries to use the experience that the program will have after some time using our service to improve the results. In other words is to use the knowledge that the program will have from the user to improve his results.

At the beginning the program doesn't know anything about the user and he will be forced to introduce what kind of news he wants, from which sources, and all the information. But after some time using the service we will have some information about the user: what kind of news he liked, from which sources, what kind of news he didn't like also from which sources, what rules are implied in each case, and finally how the news that he introduce by himself are.

For that reason all the sources have an trust rate, that is a float number that will modify the final score of all the news arriving form his feeds. It means that after the final score we will multiply this for this number to get the final corrected score:

$$\text{Final New Score} \times \text{trust source rate} = \text{final corrected score}$$

At the beginning all the sources have the same trust level score that is 1, in other words that correction will leave always the number exactly equal. But after some time and storing all the false positives, the news that the program found interesting and the user decided that wasn't interesting, we can start using that number.

The way to use that is:
If the source has more than 10% of false positives, decrement the trust source rate 0.5 for each 10% of his hits of false positives.

Using percentages we control that the correction won’t be too much fast or strong.

By the way we will have information about which rules are implied in the positives hits, and in the negatives hits.

And if the number of negative hits of a rule is twice the number of positive hits we will subtract 0.2 to the score value.

Another interesting idea is related with the news those are hand typed, we can check all the words contained in them and filter them taking out the most common used English words, using a filter with that words and automatically proposing words to be included to the rules to the user.

4.1.3.5 How to use News Hunter Program

First of all we need the file called setup.exe, and a computer with windows and the mysql server with all the necessary tables created on it, after that we can start with the installation of that software by double clicking on the setup icon.

4.1.3.5.1 Installing the software

We have to double click on the icon called setup.

Then we Hill see that screen and we have to click Next.
The next screen asking us for the path where we want to install the program, it will suggest as a default the program files directory, creating a folder called LithuanianHunters and into it a folder called newsHunter.
Then the program is ready to start the installation, is time to click next and wait some seconds until the process is done.

And the installation process will start showing us a progress bar.
And finally the process is over, all what we have to do now is click on Close and the program will be successfully installed.

4.1.3.5.2 Uninstalling the software

First we have to open the Installing and Uninstalling programs options in our control panel.
When we are there we have to scroll by all the programs searching for the news hunter, and then click over it and choose uninstall. And will appear that screen:

![Welcome to the NewsHunter Setup Wizard](image)

Select whether you want to repair or remove NewsHunter.

- [ ] Repair NewsHunter
- [x] Remove NewsHunter

[Cancel] [<< Back] [Finish]

We have to choose Remove NewsHunter, and click finish. After all the uninstall process we will see that screen warning us that the procedure is successful completed.

![Installation Complete](image)

NewsHunter has been successfully removed.
Click "Close" to exit.

[Cancel] [<< Back] [Close]
4.1.3.5.3 Executing Hunter Program

When we have installed the program we will have to click on the icon that we will have on the desktop or in the programs menu, and the program will open.

Before doing that we will have to be sure the mysql sever is working, because the first thing that the program will do is checking the number of users and sources in the database.

In that program screen we will have the number of users registered and the number of sources to check, when we click the start button the program will execute for the first time doing all the process previously explained, we will e able to check in which step we are looking to the progress bar and reading the text that is showed just up that (now stopped).

When the process is done will start the count down to the time of the next execution, we can see now the seconds left to the next execution 1800 seconds on the middle left part of the screen.
And finally we will have a complete log of all the process in the tab called log, to check what exactly the program is doing, what happened track possible errors... In that Tab we will find a button to reset the log erasing the old information.