

Metropolia University of Applied Sciences
Institute of Technology
Degree Programme in Media Engineering

Minna Suhonen

Re-designing a Commercial Website

Bachelor's Thesis. 3 June 2009

Instructor: Joonas Turkama, Managing Director
Supervisor: Merja Bauters, Lecturer
Language advisor: Anne Perälampi, Lecturer

Author	Minna Suhonen
Title	Re-designing a commercial website
Number of Pages	29
Date	3 June 2009
Degree Programme	Media Engineering
Instructor Supervisor	Joonas Turkama, Managing Director Merja Bauters, Lecturer
<p>The goal of this final year project was to redesign a commercial web site Rips.fi for company CoreFinland in such a way that it can be updated virtually by anyone. The existing site was developed with Flash and required knowledge of the technology to be able to update the information. Also, heavy-weight and poor scalability were listed as problems and needed to be solved.</p> <p>As a solution, a redesigned, PHP and MySQL based site and an administration area were created, with some Ajax features. The previous layout of the site was preferred and thus, a very similar design was produced. Although problems were met with scripts, the objectives were met relatively well. The new site is significantly easier to maintain and does not require coding skills. The design lacks the previous animation and moving parts, but is lighter and loads faster. Ajax was used to validate feedback and order forms and to create an image gallery with thumbnails.</p> <p>In hindsight, a ready-made content management system (CMS) might have been easier and a more credible solution for renewing a site than self-coding. However, the site is relatively simple and therefore self-coding was found to be appropriate. As a site grows to contain more interactive sections, such as blogging, feeds, commenting and tagging, simple solutions often lack the needed security and usability features. The resulting site of this project can be developed further to have more sophisticated scripting, or it could be re-designed again to use a free open source CMS.</p>	
Keywords	web design, content management, PHP, MySQL, Ajax

Tekijä Otsikko	Minna Suhonen Kaupallisen verkkosivuston uudelleensuunnittelu
Sivumäärä Aika	29 3.6.2009
Koulutusohjelma	Media Engineering
Tutkinto	Bachelor of Engineering
Ohjaaja Ohjaava opettaja	toimitusjohtaja Joonas Turkama lehtori Merja Bauters
<p>Insinööriyön tavoitteena oli suunnitella uusi versio kaupallisesta internet-sivustosta ruis-snackseja maahantuovalle yritykselle. Tarkoituksena oli ennen kaikkea ratkaista edellisen sivuston ongelmakohdat ja luoda hallintaosio, jotta sivua voidaan jatkossa päivittää suoraan selaimella ilman HTML- tai PHP-kielten tuntemusta. Vanhojen sivujen päivittäminen vaati Flash-osaamista, jota monella yrityksen työntekijällä ei ole. Lisäksi sivusto oli raskas ja vaikea laajentaa.</p> <p>Tuloksena syntyi PHP-pohjainen sivusto, joka käyttää MySQL-tietokantaa ja joitakin Ajax-kirjastoja vähentämään sivulatausten määrää. Ulkoasun tyyli haluttiin pitää mahdollisimman samankaltaisena kuin Flash-sivuissa, joten rakennekin säilyi yhtäläisenä. Vaikka ongelmiakin ilmeni skriptien kanssa, tavoitteissa onnistuttiin. Sivuston ulkoasusta puuttuvat Flashin liikkuvat osat, mutta kevyempänä ja yksinkertaisempana se latautuu nopeammin ja toimii sujuvammin. Ajax-tekniikoiden avulla tarkistetaan lomakkeiden tiedot, vaihdetaan ulkoasun tyyliä ja luotiin entistä toimivampi kuvagalleria. Hallintaosion ansiosta sisällön muokkaaminen on yksinkertaista käytännössä kenelle tahansa.</p> <p>Valmis sisällönhallintajärjestelmä olisi saattanut olla huomattavastikin helpompi ja nopeampi ottaa käyttöön kuin itse luotu, mutta koska sivusto oli kohtalaisen yksinkertainen, päätettiin pysyä alkuperäisessä suunnitelmassa. Jos sivusto sisältäisi esimerkiksi blogin, tunnisteita, kommentointimahdollisuuden ja syötteitä, olisi yksinkertainen itse tehty järjestelmä varmasti riittämätön käytettävyyden ja turvallisuuden kannalta. Tuloksena saatua sivustoa ja sen skriptausta voidaan tarpeen mukaan helposti kehittää edelleen niin, että sen ominaisuudet ovat kehittyneempiä, tai sivut voidaan suunnitella jälleen uudelleen niin, että ne käyttävät valmista avoimen lähdekoodin sisällönhallintaa.</p>	
Hakusanat	verkkosivujen suunnittelu, sisällönhallinta, PHP, MySQL, Ajax

Contents

1	Introduction	5
2	Background and Fundamental Concepts	5
2.1	Flash	7
2.2	Web 2.0	8
2.3	Usability	9
2.4	Security	10
3	Methods and Technologies	12
3.1	CSS, XHTML	12
3.2	PHP	13
3.3	MySQL	13
3.4	Ajax	14
4	Results	15
4.1	Defining the Requirements	16
4.2	Creating Mock-ups	17
4.3	Defining the Database and Architecture	18
4.4	Creating the Layout	20
4.5	Creating the Administration Area	22
4.6	Confronting Problems	24
4.7	Testing and Validating	25
5	Discussion	25
6	Conclusions	27
	References	29

1 Introduction

The aim of this project was to redesign a website in such a way that both the company and the customers benefit from it. This means fixing the issues that currently exist, as well as creating some new features. The company, CoreFinland Ltd, imports rye snacks to Finland and the website Rips.fi is about the products.

As a background to this project, I did my industrial placement for CoreFinland and my tasks concentrated mainly on maintaining the websites and fixing their problems. During this period I evaluated the sites and elicited the development ideas and requirements for the sites. The website that I am concentrating on is aimed at the customers, which makes it very important for the site to be functional, as well as to look professional for creating a beneficial image for the company.

The main renewal will be done with the technical part, even though the visual appearance will also change. All the information will be entered to a database for an easy, browser-based maintaining of the site. For this purpose, an administration area will be created. The objectives are to achieve a dynamic site that can be updated easily without a profound technical knowledge, that is scalable, and that looks professional.

2 Background and Fundamental Concepts

The site to be renewed has been created using Flash and ActionScript 2.0 by a previous trainee. There are simple instructions for future trainees on how to update different parts of the site, such as events, product information and image gallery. In addition to the Finnish version, there is the English site which is updated alongside with the Finnish one on the relative parts. Also a Swedish version is expected to be published. The website, as illustrated in Figure 1, contains product and company information, a competition, media section with images, a video and sound clips, and a possibility to order the products or a t-shirt as well as to send feedback. Recently added new sections

include recipes and information about the health effects of rye. On the right hand side there is space for Flash banners.



Figure 1. The original Flash site.

Currently, the company CoreFinland Ltd is financially unable to hire long term staff for maintaining its websites and other affairs on the Internet. The work is done by trainees who stay for about couple of months each. Naturally all of the trainees have different kind of backgrounds and there is no certainty that there will be a person available who has some experience with the used technologies. Until the company has sufficient financing for real employees, it is a notable benefit if the web sites can be updated by anyone who is familiar with Internet browsing. This way the company will be able to use the budget into other urgent issues, not to mention the conventionality of web-based administration. The administration area displays information from a database and has in minimum the functions to add, delete and modify data.

After recognizing the requirements and restrictions, the following step is to evaluate the current site and find the existing issues that need to be solved. Several issues have been pointed out to be corrected to the next version. The scalability of the site is difficult as

the navigation is basically an animation. This requires re-doing the animation to be able to add new sections. Especially a page for news or current affairs is on the top of the wish list. Also the ability to add some interaction to the site in a form of forum for recipes has been discussed. Overall the site would benefit from having a factor that would keep the visitors coming over again. A simple Flash-based game featuring the products could serve as one alongside with the forum. Whether or not that is developed during this project, at least there should be a suitable environment for it in terms of the space available.

2.1 Flash

Originally, Flash was created to be a vector animation software. Since then it has developed to be more controllable by ActionScript scripting, and thus it became a method to create interactive web sites. However, a Hypertext Markup Language (HTML) based website, in addition to a programming language, can be interactive as well. The strengths of Flash include that complex animations can be created with it, and video and audio can easily be played. Basic text, however, is not handled very impressively, as the rendering is slower and not as clear as with HTML. Also, despite the fact that vector graphics are light-weight, Flash sites are often heavy and take a long time to load especially on slow connections. The ideal solution would be to mix HTML and Flash on the relevant sections in such a way that the best features of both can be used. [1]

A Flash site in itself is basically a series of animation with static content in spite of the used scripting, unless it uses external files or a database. All of the text content is included in the animations and thus also modifying the text requires some knowledge of Flash, even though the animations or scripting are not altered. However, the text can be retrieved for example from an external text file. This would eliminate the need of Flash knowledge in updating but would not make the file size significantly smaller and thus reduce the loading time. Furthermore, an administration area can be created to control the contents from a database. This would be a noteworthy solution if the site otherwise was perfectly usable, except for the lack of expertise. In the case of Rips.fi, the site

definitely needs thorough redesigning and, even though a technologically aware person could maintain the site, it would be unnecessarily complex to continue using it as a Flash-based site when more simple solutions can be created.

2.2 Web 2.0

Desktop applications have had usability techniques applied to them that have not been available for web applications. However, the Internet has developed into a more mature state enabling increasing use of potent technologies for enhancing user experience. Features that are familiar from desktop applications, such as drag and drop, have not been available on web before. [2] Benefits of web applications include high accessibility, no required installation, easy upgrading and access to large amounts of data. The costs of development and support are lower than that of desktop applications, in addition to shorter time to market. [3]

A Rich Internet Application (RIA) is an application that bridges the usability gap between desktop and web applications by offering higher levels of interactivity compared to normal Internet applications. This requires more code on the browser and most importantly JavaScript, which is provided directly by the browser without add-ons. [3] From technological perspective, the term Web 2.0 is to refer to web development that tries to find new ways to make existing technologies to work together. The technical foundation includes HTML, CSS, HTTP, JavaScript, XML and server side programming among others. Often developers are specialized on one technology and have limited knowledge on others, which makes it important to know enough of them all to understand the division between them. For instance, it makes no sense to do something in JavaScript with a complicated function that can easily be done in CSS. [4]

2.3 Usability

For the purpose of designing a website, the broad definition of usability can be summarised to refer how well the site can be understood and used effectively to achieve certain goals. According to usability expert Jakob Nielsen, web site usability breaks down to five components: learnability, efficiency, memorability, errors, and satisfaction. Learnability of a site can be evaluated by how easy it is to perform tasks on a first visit. In web 2.0 sites this may become a problem if a site acts differently than a user is accustomed to, for example due to use of Ajax. Thus, simplicity and consistency help in learning an interface. [5] Remembering the user expectations with the development process eliminates unexpected actions effectively. All links and buttons should have relevant terms or concepts so that the user is aware what will happen next. Knowing the users, their context of use as well as actual tasks and practices, helps with designing better user interaction of the product. [6]

The next step is to evaluate the efficiency of performing a task once it is learned. In this, new technologies can make a positive difference in reducing the time needed by combining multistep processes. That has also an effect on how easily a user can memorize the use of an interface after a period of time. However, by using Ajax, unique widgets can be created that have no points of reference to the users. When changing the way of operation, the user should always be notified. For example, users do not expect that a hyperlink will submit forms when mostly buttons are used. Furthermore, the back and forward buttons in many cases do not work with Ajax interfaces. The user must be provided with information of what is going on and how to interact with the product if it does not follow the established conventions of use. [7]

Having only few errors in a system is self-evidently an important usability goal. The focus is also on the ease of recovering from the received errors. Network problems may become more of a problem when using Ajax than in conventional web development because the browser's reload button will not send the Ajax action again. Instead, reload will return to the state before the action was called for. Finally, the satisfaction of a user is strongly affected by the visual design, but also Ajax features can make the site

significantly more satisfying than a site that requires several page reloads. A good example of this is the suggestions in a search field that appear as the user writes. [3]

2.4 Security

Security of web applications is an important aspect. However, it cannot be evaluated simply as whether or not a script is secure but more like a scale that can be increased and decreased. Security often results in more code and checks, and compromises on convenience, both for the user and the programmer. One should keep in mind that in spite of what the initial purpose of a site is, it can and will be used for something else either accidentally or on purpose. Showing detailed error messages on site and using simple database references with users are a risk. On the other hand, some scripts are vulnerable to spam being sent through them if the data is not validated. [8] A simple method of spamming is to use a contact form on a site. Adding to an email field a line terminator and Bcc: with other addresses sends the message to all of them. This can be prevented by validating the email address field. Automated submissions can be prevented with something that only humans can understand, such as a question requiring an answer or a string of letters and numbers from an image. [9]

A crucial part of web site security is the checking of authorization which should be done wherever needed. Showing the users more than is required and they need is, besides a security risk, needless. Validating and filtering input data is done without disturbing the user. Because of Ajax, no more security holes exist in web development than without it. However, Ajax has created an environment for more sophisticated attacks than before by exposing more of the application to client-side scripting. Creating dynamic web applications increases the risk of making mistakes. Thus developers must ensure that no doors are unintentionally left open in an application. Only the XMLHttpRequest object brings a new form of technology. The object can make requests only to the same domain name, this way eliminating the possibility of cross-domain requests. [9]

If private information and sessions are sent, traffic sniffing can be effectively prevented by using HTTPS. This method encrypts the data being sent and significantly more effort is required for breaking into it. However, not all information is reasonable to pass over HTTPS due to the negative impact on performance. Selective encryption, commonly the authentication process, suits most web applications best. HTTPS does not, however, protect against attacks such as cross-site scripting (XSS), cross-site request forgeries (CSRF) or SQL injection. [9]

Unescaped values are a security risk that could be used to run statements and commands not intended by the developer. Thus, the strings containing characters that have a special meaning in code, such as single or double quotation marks, should be escaped in such a way that they are not confused as being code. A text string such as "You aren't logged in" would leave a hole for an attack if the apostrophe was not escaped with a backslash. Therefore, all code around data sent for outputting should be escaped properly, ensuring that the data does not break rendering and will remain untouched. In SQL injection attacks, SQL commands are run by abusing unescaped strings via GET or POST requests. This way the attacker can run any SQL commands that the database has permission to run and, for example, retrieve account information. XSS attacks have the same principle as SQL injections. However, in XSS attacks the vulnerability is used to allow attacker's scripting or markup to get evaluated. Data that either fails to get escaped when displayed or is sent by a browser can be used. Filtering the requests can prevent the attacks, accepting only expected data types, and escaping all parameters in SQL statements and output data. [9]

CSRF attacks send GET or POST requests from another location via a browser. Actions are attempted to be performed as a user viewing the exploit. Therefore, prevention is based on reinforcing user authentication. The strongest method for this is creating a random, unique token for every request. While the tokens cannot change for every request because of the asynchronous behaviour, they can be reused in parts of an application where only one request can be made. This way it gets difficult for the attacker to steal the token in time to use it. [9]

3 Methods and Technologies

In this chapter, the methods and technologies that were chosen to be used in the redesign project are presented. In addition to the basic web technologies discussed in section 3.1, a scripting language and a database were needed to enable dynamic output and web-based updating. These are studied in sections 3.2 and 3.3. Finally, some dynamic features were added and the technologies utilised for them are discussed in section 3.4.

3.1 CSS, XHTML

HyperText Markup Language (HTML) is the basic building tool of a web site. However, the latest version of it, 4.01, was released in 1997 and thus it hardly keeps up with the changes and trends in a world wide web. As a successor, XHTML 1.0 was developed in 2000 and XHTML 1.1 in 2001. Basically, XHTML 1.0 transitional is the same as HTML 4.01 except that it uses XML syntax, which essentially means that all tags have either a starting tag and an ending tag, or they close themselves. Furthermore, XHTML 1.1 differs only little from its previous version. However, it is only based on XHTML 1.0 strict which results in some features not being available, such as frames and the style associated markup. [4]

As XHTML strongly encourages separating style from content, cascading style sheets (CSS) are used for defining the style of a web page. This makes it also easy to change the layout without having to touch the content, which remains unchanged. [4] Using separate XHTML and CSS reduces the amount of code which results in smaller file sizes and increased maintainability. Multiple different stylesheets can be made available on a web site so that a user can select the preferred one, as was done in this project. Also a site can be optimized for different devices or for printing. In the beginning of 2009, the latest version of CSS is level 2 revision 1. [10]

3.2 PHP

There are at least a couple of methods by which the site renewal could be carried out. Popular technology for this is PHP which is a widely supported, server-side, cross-platform, open source scripting language. [10] PHP was designed in 1994 to write web scripts and not stand-alone applications as opposed to programming languages such as Java. PHP runs on most operating systems and a script written on one server usually works on another. Furthermore, PHP can be integrated with virtually any database available and it has high extensibility. [8] In March 2009, PHP is at version 5.2.9.

As a scripting language, PHP can be embedded into HTML page to have a dynamic result instead of static page. When a request is made on a static page, HTML data is returned without any server-side action. The PHP scripts run after an event, often when a URL is entered, and everything PHP does occurs on a server. The server processes the code and returns appropriate HTML code to the browser. However, to a user the process looks the same whether or not server-side action is happening. [8] There are many reasons why PHP was preferred in this project over, for instance, ASP.NET. The most important one is that, from programming languages, only Perl, Common Gateway Interface (CGI) and Server Side Includes (SSI) were supported by the web hosting service, in addition to PHP. Also, PHP is a common language choice for web sites similar to this project, and open source was preferred by the company.

3.3 MySQL

With HTML and PHP, the data is on a hard-coded basis. For a flexible basis and dynamic output, a back-end data storing is necessary. A popular choice for this is an open-source database MySQL. It is a portable and reliable relational database management system (RDBMS) that uses multiple tables to store data. In a nutshell, working with a database includes few steps. Tables can be manipulated by SQL queries. A connection to the database is established using a programming language, such as PHP, and SQL queries are sent. Finally, the results of a query are retrieved and often displayed. [8] Crucially, MySQL was chosen to be the database in this project because it

is already available in the web hosting service that Rips.fi uses. Also, as it is open source and works well with PHP, the choice was clear.

3.4 Ajax

Ajax stands for Asynchronous JavaScript and XML (Extensible Markup Language), although it does not necessarily use either XML or asynchronous exchanges any more. The term first occurred in 2005 to describe a combination of technologies. It is considered to be one of the pillars of Web 2.0 and thus an essential tool. [4] None of the components of Ajax are new, however, but now they are being standardized and have a clear direction of evolution. Ajax changes the typical page flow where user performs a request by clicking a link or typing a URL and a server responds. In an Ajax application, a new type of request is added. To a server, it is a normal request but the browser does not need to reload. These kind of requests can be small, for example validating fields in a form or preloading data, as well as large, such as submitting the form or returning HTML code to replace the content on a page. As the request frequency is high, it sets expectations to capabilities of servers. [3]

JavaScript is the essential part of Ajax. The functions make heavy use of Document Object Model (DOM) to manipulate the HTML page. The XMLHttpRequest object enables asynchronous access to the server in background while the user can continue working. In addition, server-side technologies are required to handle the requests. Multiple programming languages can be used for this, for instance PHP. [2] Data encoding in Ajax can be done with a format that is most convenient for the situation. XML is a popular one because of its large scale support in programming languages. With it, describing arbitrary data types is easy. [3]

Ajax becomes beneficial for the user if a functionality that can make a task easier can be achieved. Overusing it wastes server resources and results in longer loading time of a web page. Instead, a light-weight script that loads content whenever needed can be used. Furthermore, the design of functionalities should concentrate on helping the user to

proceed without distracting with the dynamic elements. There has to be a balance with the asynchronous actions so that they are not intrusive to the user but the change is still apparent enough. [9] Increasing interactivity with the user provides more information dynamically, for instance giving suggestions as a user types on the search field. When the extra data is loaded only when it is needed, searching and browsing become faster and natural. Once a task that takes a long time is found, an Ajax solution for making it easier might be found by using in-place editing or drag-and-drop. [3]

In this project, in-place editing could have been considered as a method to modify the information in a database. Also, the ordering form could have utilised drag-and-drop to add products to the shopping cart. However, more conventional PHP-based solutions were created, keeping in mind that the future users may need time to get accustomed to the new site and the administration area even without these still relatively rarely seen Ajax methods. Furthermore, no real problem or difficulty were found in using the PHP functionalities, and thus the Ajax solutions would not have provided anything that were not already present. Search field was not considered essential for the rather small-scale site. In a suggestion list for further development, these possibilities of Ajax are mentioned as options for creating new features or developing the existing ones.

4 Results

After it was agreed with the company that the current site needed renewing, the project started by collecting the requirements for a new site. The target group for the site had been defined by the client. The users of the site are consumers that are interested in Rips rye snacks. For professional customers, namely from hotels, restaurants and cafés, there is a separate website. The company defines its target group as being health-oriented people that are from 20 to 35 years old. On the consumer site, the products are promoted as being an option to the traditional potato chips that is healthier and has less fat. Ingredients and guideline daily amount (GDA) information are presented. Also, there is a separate section for information about the health effects of rye.

Taking into account the needs of the target group, as well as the needs for the maintenance of the site, I formed a list that I sent to the manager and few other workers knowledgeable of the topic that could have had suggestions and who could provide relative feedback. At the same time, I created two mock-ups of the layout structure. The list was approved as it was, in addition to the second mock-up. Based on these points, the process of creating a new site started. The outcome of the accepted decisions directed the choices for the programming languages, which were the following: PHP, a MySQL based database, in addition to Ajax technologies for the relevant sections of the site. The next phase for me was testing. I conducted the testing during the project with different up-to-date browsers that were known to be potentially used by the target group, and also feedback was collected from the relevant company representatives, including those responsible for the company's visual image. Based on it, I modified mainly the visual layout. Furthermore, I validated the XHTML code at the end with Markup Validation Service offered by the World Wide Web Consortium.

4.1 Defining the Requirements

The first phase of the project was to collect a list of the issues and development areas with comments. The previously mentioned, approved list included the following five points that were seen as problems on the existing site. Firstly, the site was heavy for old computers and slow connections, such as those that often use USB-modems. Secondly, updating the site was not possible for those who were not experienced with Flash and thus guidance was needed. Thirdly, due to the complex layer structure and frequent use of animations, modification and extension of the sections required inspection and was time consuming. Fourthly, the graphics were the main focus of the site. There was only limited space available for text and to be able to add more, the only option would have been to reduce the text size. Finally, the image gallery was troublesome to update. All of the images were placed in a line and the simple solution would have been to add new images to the end. To add the newest ones to the beginning meant that every image had to be separately moved backwards. Furthermore, images were not categorized and there was no space for description.

In addition, new sections for the site emerged when the list was discussed with company representatives. One would be news or current events. Recipes would bring new ideas of using the rye snacks, for instance with soups and salads. For keeping the visitors at the site, a simple Flash game featuring the snacks was suggested. Also, some interaction was discussed in form of a possibility to comment and add own recipes. Finally, an administration area for updating the site was needed. In addition to the image gallery, ordering the products would benefit from improvements, as it was only a simple form with fields for name and address, and a box for writing the ordered items. Generally, the resulting site should be lighter and take less time to load than the Flash one. My proposal was that everything that needs to be updated should be in a database so that the updating can be done in the administration.



Figure 2. A rough mock-up of the structure, version 1.

4.2 Creating Mock-ups

After defining the requirements, I designed two mock-ups of the structure using Photoshop, as illustrated in Figure 2 and 3. Out of these version two, Figure 3, was preferred by the company due to the similar structure to the existing site. In this version, the navigation links are on left and on the right side of the content box is a Flash banner, as well as on top of the page. The logo is on the top left corner, thus no significant differences were created. The first suggested version in Figure 2 is otherwise similar but it has the navigation on top of the content box. The manager of the company noted that in this version, there would be less space for the links, and thus if the site expands more, another line would have to be used. Also, he wished that the site would not look very different with the Flash version, so that no impression of going backwards in technology could be made.



Figure 3. The version of structure mock-up that was chosen.

4.3 Defining the Database and Architecture

According to the decisions I made for the database, all the text in the main sections of the site is inputted into a MySQL database in tables, as illustrated in Figure 4. The sections Competition, Company and Healthy Rye contain some paragraphs of text and thus only one text field was created with no identification fields. The other tables have several records in them and therefore id field is relevant to be able to modify the respective record. In addition, all of them have title and other fields that include text content. In Figure 4, the Users table is displayed with grey background because it is the only field used exclusively by the administration area. Figure 4 does not display all the security features, such as field for privileges. The privileges were added to the list of development ideas to be created in future, due to the schedule restrictions during this project.

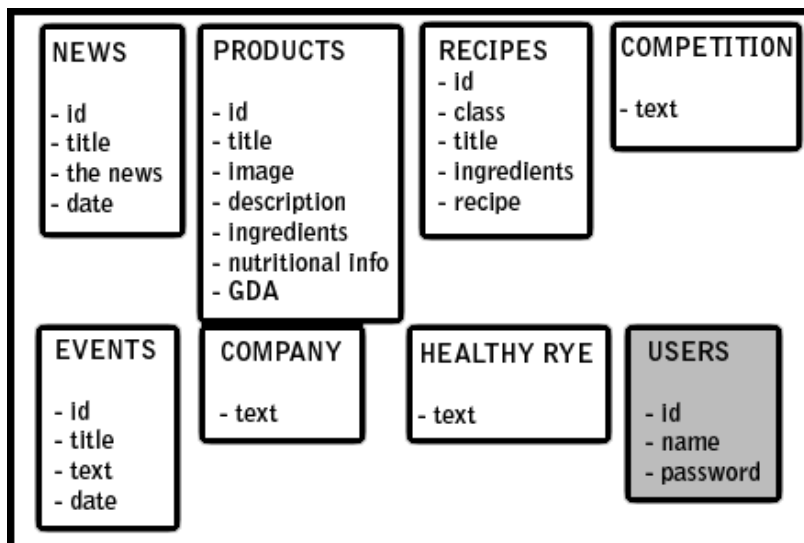


Figure 4. The structure of the database.

The architecture of the main site that I created is illustrated in Figure 5. All of the sections, excluding ordering and feedback forms, retrieve their content from the database using the file databaseconnection.php. The forms for ordering and feedback use Ajax functionalities for validation and the media section has an Ajax based image gallery.

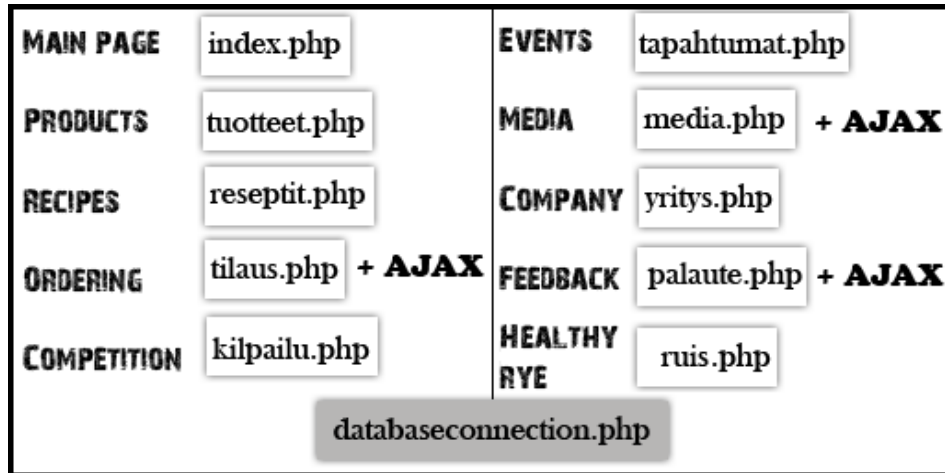


Figure 5. The architecture of the main site.

4.4 Creating the Layout

In the next phase I created the actual layout and site. I directed special effort to the visual design and Ajax functions to minimize the impression of a static and old-fashioned site. As the Flash site relies heavily on animation and sound effects that are not present to the same extent in Ajax, something substitutive was needed. I designed several versions of possible layouts during the project, out of which I selected the best two to be the default and the optional style. The default style is illustrated in Figure 6. In this design, the background image consists of pictures of a bunch of the snacks, and in text background is an image of grain field. Font style used in navigation is the same as in the Flash site, and thus the links are images. In the optional stylesheet, the background images and heading colours are different.



Figure 6. The default style miming the original, without banners.

Functionalities that were actualized using Ajax ended up being validating a feedback form due to its superior speed compared to a PHP-based version. Asynchronous requests validate the fields straight after a user has input information. Also, the image gallery uses Ajax libraries. The used script Lightbox 2 is created by Lokesh Dhakar and it overlays the images on top of the site and darkens the background, as illustrated in Figure 7. Changing the used stylesheet is also an Ajax-based function.

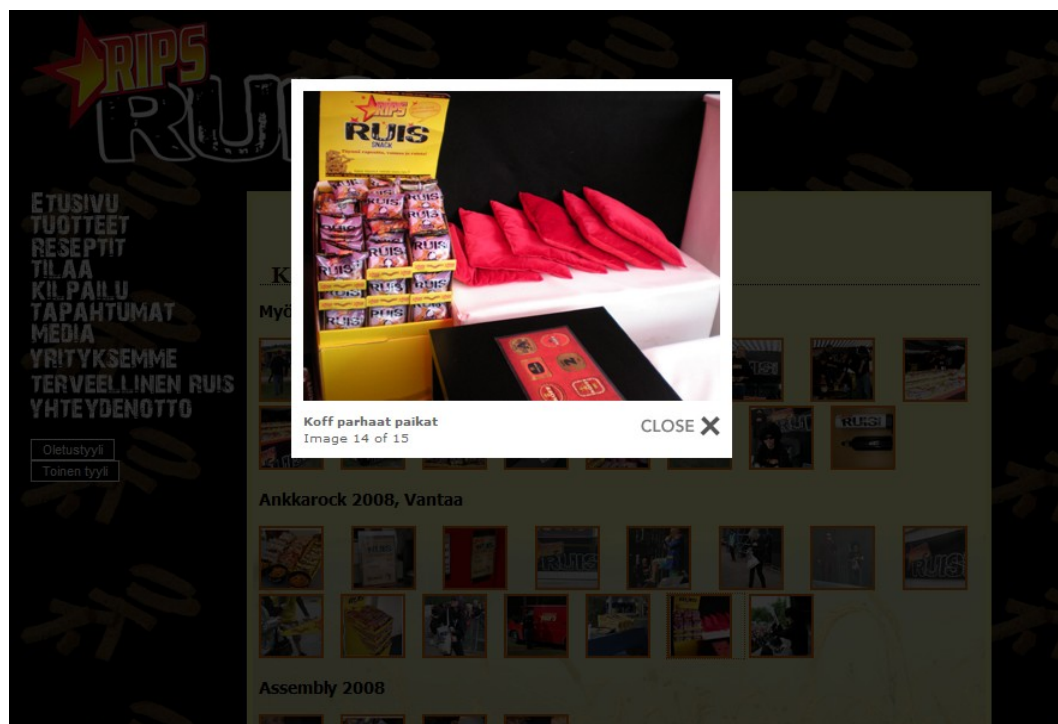


Figure 7. The Ajax-based image gallery.

4.5 Creating the Administration Area

Among the accepted list of requirements, the most relevant features of the administration area were the possibility to update and modify all the information on the site, as well as to add news, events and recipes. This was inspired by my personal experience of updating the site, as those were the most common actions the webmaster had to do along with fixing the bugs. I planned and executed the whole implementation of the administration area. There were no mock-ups of the structure or layout of the administration, and it basically follows the style of the main site to be consistent with the look and feel, except that there are two navigation bars. The final user interface is illustrated in Figure 8.



Figure 8. The user interface of administration area after logging in.

As shown in Figure 8, the bar on left has links to display the data of the chosen category and also possibility to delete or modify the existing records. On right there are quicklinks to adding new records, as I noted it was a common task for the webmaster. On top of the content and navigation boxes, the username of the currently logged user is shown, next to a link to log out. Furthermore, on left under the content modification links, there is a set of instructions written by me for using the administration and technical information and suggestions for future developers. The instructions were done in case the user needed guidance in what can be done with the administration and they are directed to the ones that have no experience in web technologies and managing the

content. Technical information is targeted to the future webmasters who will fix the bugs they encounter and possibly extend the features. I also listed the suggestions that I had in mind during and after the process. Below the content box is the email address of webmaster to be contacted if facing technical difficulties.

NEWS	-> Modify & Delete & Add .php	+ AJAX
PRODUCTS	-> Modify & Delete & Add .php	+ AJAX
RECIPES	-> Modify & Delete & Add .php	+ AJAX
EVENTS	-> Modify & Delete & Add .php	+ AJAX
COMPETITION	-> Modify .php	
MEDIA	-> Modify & Delete & Add .php	
HEALTHY RYE	-> Modify .php	
COMPANY	-> Modify .php	
FEEDBACK	-> Modify .php	
databaseconnection.php		

Figure 9. The structure of the administration area.

As Figure 9 illustrates, modifying and deleting was established by using PHP but the adding function is an Ajax-based editing box that appears on top of the site. Deletion and adding files are in use in relevant sections that have multiple records. Similarly as on the main site, all the functions use the databaseconnection.php file. A form for logging in was created using PHP. Due to security concerns, all passwords are encrypted before entering them to the database to prevent them from leaking. Encryption is done with md5-function when administrators create new user accounts. The function transforms the password into a 32-character hexadecimal number. [11] Thus, even administrators who have access to the database records cannot see passwords in clear text. Furthermore, a method for changing a password was also created. Also, SQL injection was taken into consideration by using stripslashes and mysql_real_escape_string functions as follows:

```
$myusername = stripslashes($myusername);  
$mypassword = stripslashes($mypassword);  
$myusername = mysql_real_escape_string($myusername);  
$mypassword = mysql_real_escape_string($mypassword);
```

Stripslashes takes the quotes away from a quoted string [12] and `mysql_real_escape_string` escapes special characters that could be used in attack, enabling log in without a password. [13]

4.6 Confronting Problems

I met several technical difficulties during the process, both in main site and administration. For some reason the used javascript functions did not have any effect and thus the Ajax-based image gallery plainly displayed all the images after one another. Also, only Firefox browser displayed images from database in product information page. Google Chrome and Internet Explorer did not display the images at all – even when the file location was addressed. The validation of forms also posed slight technical problems. The form for ordering the snacks continued to order confirmation page even when nothing was entered to the fields. Thus, PHP based solution were created for validating the forms while the aforementioned problems were solved, and another Ajax libraries were searched for the image gallery. Debugging Javascript scripts is difficult because distinguishing whether the error is on the server-side or client-side code – or a cascade of problems between them – is troublesome. [3]

Problems that occurred while creating the administration area included that the initial solution for logging in was rather simple and lacking features in security and usability points of view. The time that the system keeps an account logged in seemed to be inconsistent and unaffiliated with whether or not the user was idle. Also all of the users had the same permissions as the administrator to perform any task. As with the main site, there were problems with Ajax libraries not functioning and other libraries had to be searched for, which delayed finishing of the project. Especially with the image gallery I had to test several libraries before finding Lightbox 2.

4.7 Testing and Validating

During the project, I used mainly Mozilla Firefox 3 browser on Windows Vista operating system to preview and test the site, in addition to occasional testing with Google Chrome and Internet Explorer. Besides the image problems discussed in section 4.6, some minor differences were found in the visual appearance that were not regarded as significant that actions should be made. Validation of the XHTML 1.0 markup was done with a validator by the World Wide Web Consortium, and corrections were made to meet the standards. When the layout was ready, the site was reviewed by the company's graphical designer who gave feedback. He noted several issues and development areas, for instance in colours, background images and text sizes. Obviously, I corrected all the aforementioned parts.

5 Discussion

As the Rips.fi site contain only few sections and relatively static information, self-coding a simple administration area was an interesting possibility to be able to control every feature. If the site was larger and included such elements as blogs and tags, a ready-made content management system (CMS) would have been significantly easier and faster to establish, not to mention a web shop with online payment. The same applies to discussion board that was mentioned before the project began. Even though the board would have been small scale, mainly adding new recipes and commenting them, it would have required security features such as validating the input data, preventing spam and identification of users. On the other hand, a Flash game was also in demand to attract the visitors to stay at the site. For maximizing the impact, a leader board would encourage competition – as well as prizes for the winners. However, animating and scripting the game would have required too extensive work and, thus, the project was left for the future so that the renewing schedule would not be outstretched exceedingly.

As noted in the results, coding an entire site is time consuming. Knowledge and research is required to be aware of all the aspects of web development, especially usability and security issues. Using and debugging Ajax scripts can be challenging for those who have little to no experience with JavaScript. Also, maintaining a site is a continuous work to be done as the technology evolves and new bugs in code are found that create security gaps. Thus, a ready-made CMS is an enticing option. Installing a CMS might require an experienced web developer but after that the site can be fully managed without coding skills. Furthermore, some systems are free of charge and open source, such as Joomla! and Drupal. There is also a wide variety of extensions for demanding use. [14] However, if the site was Flash-based and focuses on being visually imposing with mostly static content, it does not necessarily benefit from such systems.

The administration area in this project was designed for updating and managing only content and not design or structure. To achieve a fully browser-based administration, basically all text on the site should be retrieved from a database. Furthermore, the next step would be enabling also the complete change of layout straight on a browser, which certainly is worth considering. Instead, a simple css-based layout variation was created so that the visitor can change the appearance of the site. Adding a functionality to create new colour styles might have been possible even during this project. However, the real use for this would be questionable in an environment designed for people who use it for maintaining the contents and are perhaps only familiar with browsing. Another option would be to enable stylesheet creation for visitors as they are the ones for who the design is aimed at, but again this would mainly be a curiosity and would probably be mostly used by those who dislike the existing styles.

The objectives for the project emerged from real need and were the following: to achieve a browser-based site management and not to step back to old-fashioned appearance. The first objective was met as there now is an administration area that is usable by people with no technical knowledge. Maintaining the site is now easier and faster than before and can be done by virtually anyone, in comparison to the previous situation where only webmasters could perform the updating. However, as this is the first time this kind of content management is in use in the company, there is no previous experience of it and more detailed requirements will eventually be collected. Thus, when the site is developed or altered from this state, the objectives will be more

specified.

The second objective is more or less a matter of opinion. In terms of technology and usage, HTML, PHP, MySQL and JavaScript indeed are more aged than Flash and ActionScript. [15] This way a simple speculation could be drawn that Flash in itself is modern. However, the Flash site in the beginning of this project mainly used features that are not available in the older technologies to display images, video and audio, in addition to short animation clips. The renewed site still has embedded Flash objects in form of banners and players for video and audio, as Flash currently is the most convenient format for displaying animation and media files. As a result, nothing from the contents is missing and the site is much more light-weight. Furthermore, any text can now be highlighted and copied by using a mouse. This was basically possible in Flash only in the sections where the feature was enabled. Also, the different sections can be bookmarked and linked to without having to navigate to them. In terms of graphical design, the layout is rather simple and not especially ambitious. However, styles can be easily altered afterwards.

6 Conclusions

The aim of this project was to redesign a static Flash-based web site to be dynamic without having to compromise too much on the previous visual and functional features. Heavy weight and poor scalability were also listed as problems that needed to be solved. An important goal was to make the site manageable by virtually anyone. The objectives were met relatively well. Creating an administration area and entering most of the text into a database has made the site significantly easier to update, particularly for those without knowledge of HTML, PHP or Flash. Especially the image gallery is now more convenient to update than the complex set of Flash layers. However, the design has less dynamic, moving parts which inevitably makes the site look more static than before. This was not seen as a major weakness as the design otherwise is similar to the old layout, as well as functional. The site is now light compared to the previous one, and it can be easily extended. This, however, requires a developer experienced in PHP.

Although the site is functioning as it was meant to, many features of the administration would benefit from further development. These points were collected as a list with some suggestions. Besides utilizing the site as it is, the options are to continue with the existing code and to expand it with more sophisticated features, or to start again with a ready-made CMS that has all the desired facilities. It is also possible to return to Flash in such a way that all the information is retrieved from a database. In this case, however, the problems of the previous Flash site should be taken into consideration to be able to avoid them.

References

- 1 Mischook S: What is flash, when and why to use it [online].
URL: http://www.killersites.com/articles/articles_FlashUse.htm. Accessed 16 April 2009.
- 2 Darie C, Brinzarea B, Cherecheș-Toșa F, Bucica M. Ajax and php: building responsive web applications. Birmingham, UK: Packt Publishing; 2006.
- 3 Eichorn J. Understanding Ajax. Upper Saddle River, NJ: Prentice Hall; 2007.
- 4 Van der Vlist E, Ayers D, Bruchez E, Fawcett J, Vernet A. Professional web 2.0 programming. Indianapolis, IN: Wiley Publishing; 2007.
- 5 Krug S. Don't make me think: a common sense approach to web usability. 2nd ed. Berkeley, CA: New Riders; 2006.
- 6 Sinkkonen I, Kuoppala H, Parkkinen J. Professional psychology of usability. Helsinki: IT Press; 2003.
- 7 Kuutti W. Käytettävyys, suunnittelu ja arviointi. Helsinki: Talentum Media; 2003.
- 8 Lauriat S. Advanced ajax. Boston, MA: Prentice Hall; 2008.
- 9 Ullman L. Php 6 and mysql 5. Berkeley, CA: Peachpit Press; 2008.
- 10 York E. Beginning css. Cascading style sheets for web design. Indianapolis, IN: Wiley Publishing; 2005.
- 11 The PHP Group. Php: md5 [online].
URL: <http://fi2.php.net/md5>. Accessed 20 May 2009.
- 12 The PHP Group. Php: stripslashes [online].
URL: <http://fi2.php.net/stripslashes>. Accessed 28 March 2009.
- 13 The PHP Group. Php: mysql_real_escape_string [online].
URL: <http://fi2.php.net/manual/en/function.mysql-real-escape-string.php>. Accessed 28 March 2009.
- 14 Open Source Matters, Inc. What is Joomla? [online].
URL: <http://www.joomla.org/about-joomla.html>. Accessed 28 March 2009.
- 15 Gay J. The history of flash [online].
URL: http://www.adobe.com/macromedia/events/john_gay/page04.html. Accessed 16 April 2009.