

Automated webstore and order processing

Plaetinck, Dieter

May 23, 2007

“KaHo Sint Lieven” Department Industrieel Ingenieur
G.Desmetstraat.1, B-9000 GENT
dieter@plaetinck.be

Abstract

This paper documents the development of a webstore application. The application is tailored towards the specific requirements of the meat selling company *Plaetinck D. Vleeshandel*. The application consists of multiple parts: the webstore itself and the interface for the administrator. The administration of the store is straightforward and automated in several ways. Included in the admin interface is a planner for itineraries and deliveries of orders. This paper deals with the platform research, the software development and a study about how an application like this could integrate with the Semantic Web. Throughout the entire process, the aspect of security has been especially important.

Keywords: *Webstore, semantic web.*

1 Goals

The goal of this project is to develop a webstore application. The application has to be suited for the specific needs of the company *Plaetinck D. Vleeshandel*. The request is a webstore where customers can buy goods, but the goods have to be priced according to the group to which the customer belongs. (e.g. butchers and restaurants have to pay different prices for the same goods).

Also the tracking of both orders and new customers is not like regular webstores: after registering, the account of the customer is not valid until approved by the administrator. The reason for this is that the background of each company has to be checked before they are allowed making (expensive) orders. Likewise, orders made by customers with valid accounts have to be reviewed by the administrator, because stocks change rapidly and last-minute corrections are often necessary. Another unique feature is

that the system has to be aware of the opening hours of the customer, and the latest delivery date a customer agrees on. This is needed because the administrator uses a planner to schedule deliveries of orders and - apart from the availability of drivers - three factors play a crucial role in the scheduling of deliveries: the location of the customer, the opening hours, and the time a customer allows to wait for his goods. The reason for this is because usually each driver visits lots of customers consequently, so the closer customers are located to each other, and the more their opening hours match, the more efficient deliveries can be scheduled.

2 Technologies used

The whole application is web based. First existing webstores were researched, but it was clear that for an application with such specific needs, something custom had to be built. After careful research, the LAMP platform was chosen both because of the attractive pricing but also because this technology is well suited for such an application. LAMP consists of the Linux operating system, the Apache webserver, the MySQL database system and the PHP scripting language. For faster (and better) development the CakePHP 1.2 framework was used, along with the prototype & scriptaculous libraries who aid in Ajax development. The structure of the application is showed in figure 1. Additionally, the CakePHP Image Resize Helper is used

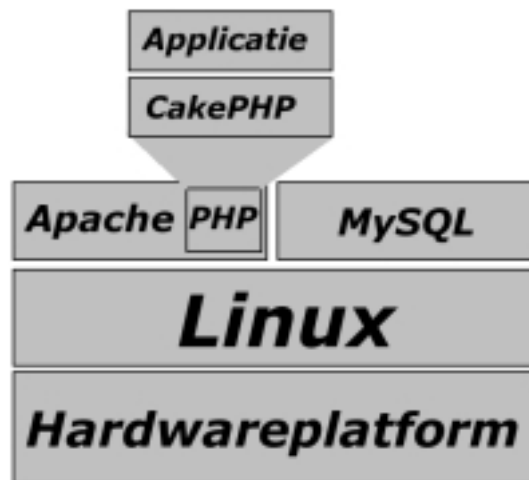


Figure 1: Application structure

and so is the TinyMCE wysiwyg-editor. For automating image uploads, the Xfce desktop environment and the Imagemagick toolset was used.

3 Implementation

3.1 Modelling

First models were constructed: use-case diagrams were used to make it clear which functionalities were required, and an ER-diagram was built to make it clear what the problem domain entities are, and how their relations were to be implemented in the application. The Model-View-Controller pattern was used to construct the application.

3.2 Security

Because security plays a big role in this application, much effort has been put into it. The application uses a challenge-response authentication component with built-in brute force detection, along with a rules-based access control system. Also popular attacks such as XSS¹, SQL-injections and session hijackings are prevented, although these features are built in the CakePHP framework. The combination of all these mechanism results in the flexible, powerfull and safe protection system that such an application needs.

3.3 The application itself

All needed requirements have been implemented using the technologies mentioned above. Both the planner of deliveries and the shopping cart work with an Ajax drag 'n drop system. When shopping, a customer can drag and drop articles into the cart, and when satisfied with the chosen articles, checkout the order. At this point he/she can provide a date on which he/she must have received the goods. On the planner a listing of all approved orders is displayed, with each order showing the name of the customer, the opening hours, the adress and the above mentioned date of latest delivery. The administrator can then create new routes, assign drivers do it and configure the time of departure. Then the orders for customers with similar opening hours and who are located relatively closely to each other can be dropped on the routes to shedule deliveries. This is displayed on figure 2

Images of articles or categories can be uploaded either via the web interface itself (through html forms), or fully automatically: the Xfce desktop environment is configured to call a shellsript upon attachment of a digital camera on the system. This script downsizes the images on the camera and uploads them to the webapplication Resizing of the images to sizes as needed

¹Cross side scripting

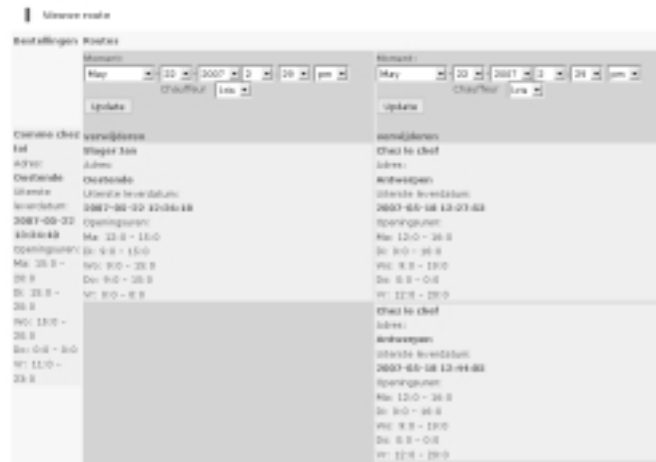


Figure 2: Planner

by the webstore, is also done automatically: images are resized to the needed size and cached for further usage. The TinyMCE editor mentioned above allows the administrator to post news on the frontpage using rich markup.

4 The Semantic Web

An interesting question is: “What is the Semantic Web, and what does it have to do with this application?”. The answer is even more interesting: the Semantic Web is about two things: “it is about common formats for integration and combination of data drawn from diverse sources, and it is also about a languages for recording how the data relates to real world objects.” [1]

In short: it is the next step in making the web do even more for humans. By using the Semantic Web one could - for example - have an agent application at the client-side automatically create orders based on information it retrieves from another source, such as current stocks, or even stocks of the customers of that customer.

Since the Semantic Web is still a work in progress, there are no definitive answers to how an application could be built for it. However, languages are being developed such as RDF (a language to express statements that can be interlinked about entities). There are also applications being developed on these new technologies such as OpenID (a distributed authentication system) and FOAF (an RDF-based social network). Due to this interlinking of data from various sources, trust becomes a critical aspect in the Semantic Web, especially for critical applications such as online stores. Unless the early stage of the Semantic Web, an approach has been thought out

that could be used for this (or any) webstore in the Semantic Web. The approach consists of a distributed authentication system (such as OpenID) along with a “Web of Trust”-based approach. [2] Encrypted xml documents on the location specified by the nodes in the Web of Trust could then be used to display information about orders of each customer.

5 Conclusion

The goal of this project was to develop a webstore that fullfills some very specific needs, such as a pricing system that accounts different prices depending on the group a customer belongs to. Worth noting is also the planner of deliveries where parameters such as opening hours, location of the customer and the latest delivery date are taken into account. The application that has been developed, fullfills these requirements. However there is still room for improvement:

By using the encryption provided by https transfers, security could be taken to a higher level. Also the usage of transactions when performing important database logic (such as insertion of an order, which needs multiple queries) would be more robust.²

The application is not yet ready for production usage, but it is a good starting point to build upon.

References

- [1] World Wide Web Consortium (2007) “Semantic Web” <http://www.w3.org/2001/sw/>
- [2] WikPedia (2007) ”Web of trust” http://en.wikipedia.org/wiki/Web_of_trust

²Unfortunately this feature will only be available in the upcoming release of the used framework.