ONLINE WARE MANAGING IN A LOGISTICAL WAREHOUSE VIA INTERNET

Mihály Varga – Etele Csanády – Zalán Koppány Kovács – Zoltán Kocsis

Abstract

The aim of this project was to create a multilingual web-interface between our partner – who is a national spedition company– and its partners, to cut lower the order processing time, and to increase the ability and the satisfaction of its partners. To realize that, we used a classic LAMP system, which is not just relaible, flexible, secure, but also cost-effective. This was proved to a good selection, because we managed to not just fullfill the project, but we could gave a really usable, simple interfaced, parametrizable solution to our partner by minimal development cost. © University of West Hungary.

Key words: *logistics, Internet, web, Linux, MySQL, PHP*

INTRODUCTION

In this project our task was to create and fit a web-interface to an exiting ERP (Enterprise Resource Planning) system of the company. It had to be cheap, parametrizale, easy to develop in the future, and of course safe. Implementation of security was the most difficult job, because we had to accomodate to the existing source system and to the content provider too. (Each of these systems have their features and limitations.) We solved this "security problem" with a special implementation: the program is totally indipendent from the source system, and also completes the (security) demands of the content provider. This means, if somebody hacks the web-interface, he will not gets into the real system. He can change values, but it is totally useless, because the web-system gets every value from the source system with automatic refresh, and between the two system there are no direct connection. The web-interface itself has been made according to the *xhtml* standards extended with *CSS* (Cascading Style Sheets).

In a few point our tasks were the following:

1. Increase the comfort of the partners
2. Decrease the time spending with managing orders
3. Cheap system
4. Wordwide (Europe-wide) accessibility and availability
5. To adapt to the customised source system
6. Customizable functionality and surface

1. IT BACKEND

Final environment	
Operating system	*FreeBSD 6.1*
Database	*MySQL 5.0.51a*
Web server	*Apache 2.2.8*
PHP	*PHP 5.2.5*

Development and test environments	
Operating system	*Fedora 7, Ubuntu 7.10*
Database	*MySQL 5.0.45*
Web server	*Apache 2.2.4*
PHP	*PHP 5.2.3*

Table 1: Software environments

1.1 Infrastructure

1.1.1 Linux

Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The kernel, at the heart of all Linux systems, is developed and released under the GNU General Public License and its source code is freely available to everyone. It is this kernel that forms the base around which a Linux operating system is developed. There are now literally hundreds of companies and organizations and an equal number of individuals that have released their own versions of operating systems based on the Linux kernel.

Apart from the fact that it's freely distributed, Linux's functionality, adaptability and robustness, has made it the main alternative for proprietary Unix and Microsoft operating systems. IBM, Hewlett-Packard and other giants of the computing world have embraced Linux and support its ongoing development. Well into its second decade of existence, Linux has been adopted worldwide primarily as a server platform. Its use as a home and office desktop operating system is also on the rise. The operating system can also be incorporated directly into microchips in a process called "embedding" and is increasingly being used this way in appliances and devices.

Throughout most of the 1990's, tech pundits, largely unaware of Linux's potential, dismissed it as a computer hobbyist project, unsuitable for the general public's computing needs. Through the efforts of developers of desktop management systems such as KDE and GNOME, office suite project OpenOffice.org and the Mozilla web browser project, to name only a few, there are now a wide range of applications that run on Linux and it can be used by anyone regardless of his/her knowledge of computers.

1.1.2 Apache Web Server

The Apache HTTP Server, commonly referred to simply as Apache, is a web server notable for playing a key role in the initial growth of the World Wide Web. Apache is developed and maintained by an open community of developers under the auspices of the Apache

Software Foundation. The application is available for a wide variety of operating systems, including Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, and Microsoft Windows. Released under the Apache License, which is characterized as free software and open source software.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support *mod_perl*, *mod_python*, Tcl, and PHP. Popular authentication modules include *mod_access*, *mod_auth*, and *mod_digest*. A sample of other features include SSL and TLS support (*mod_ssl*), a proxy module, a useful URL rewriter (also known as a rewrite engine, implemented under *mod_rewrite*), custom log files (*mod_log_config*), and filtering support (*mod_include* and *mod_ext_filter*).

Popular compression methods on Apache include the external extension module, *mod_gzip*, implemented to help with reduction of the size (weight) of web pages served over HTTP. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different actual websites. For example, one machine, with one Apache installation could simultaneously serve www.example.com, www.test.com, test47.test-server.test.com, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs) which permit easier, more intuitive configuration of the server.

1.1.3 MySQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software â including industry leaders such as Yahoo! , Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

1.1.4 PHP

PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

2. THE SYSTEM

2.1 Superuser

Structure of the menu of the superuser:

1. **Administration**
   - (a) *Create new user*

     If the superuser wants to create a new user
   - (b) *Modify user*

     If the superuser wants to modify some data of the already created user, or wants to chage its privilegs
   - (c) *Delete user*

     If the superuser wants to delete a user
   - (d) *Refresh partnerlist*

     The refresing of partners is not automatic. The source system is uploading the new parnerlist. When the superuser logs in, and clicks to this menupoint the refresing will be done.

Stock

- (a) *Stock at the moment*

  The superuser can see what is the acual state. He can see all of the stored goods, or just the goods fo a specific partner.
- (b) *Uncompleted orders*

  The superuser can see all of the uncompleted orders, and of course he has the ability to delete them.

On road

- (a) *Orders on road*

  In this menupont the system lists all of the completed orders that are already on the road and still haven't arrived to the customer and how much time needs to have arrived. (This time is estimated.) After the estimated arriving, the completed order will be listed for 2 more hours. After the 2 more hours, the order is not shown any more (but the system itself is sores it for amonth for security reasons).

Exit

2.2 Normal user

Structure of the menu of the normal user:

1. **Userdata**
   (a) *Change data*
   The user can some of change his data, like his real name, e-mail address, etc..

Orders

(a) *Place an order*
Here the user can place an order. He can pick the wished amount of the good he wants from a list, and the system automaticly sends this order to the spedition company who process the order. While placing the order, the system of course checks whether there are the ordered amount goods in the stock or not, or the user can place an order at all. The user can specify the buyer, the transporter, he can specify the time the time of start of the transportation, and he can even upload his own transportlist.

(b) *Follow orders*
Here the user can see his placed, but haven't processed orders.

(c) *Orders on road*
The user can follow his own orders that are processed by the spedition company already on the road.

Exit

3. FUNCTIONALITY IN SCREENSHOTS

3.1 Superuser

Figure 1: Modify user data

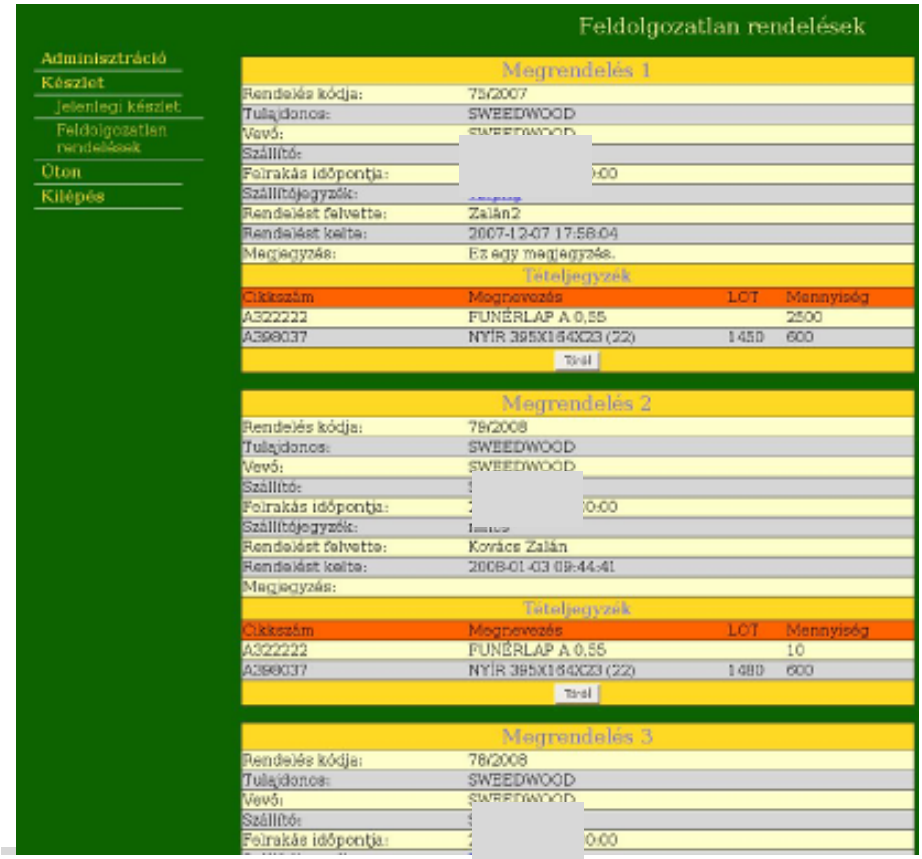

Figure 2: List of uncompleted orders

Figure 3: List of orders on road

3.2 Normal user

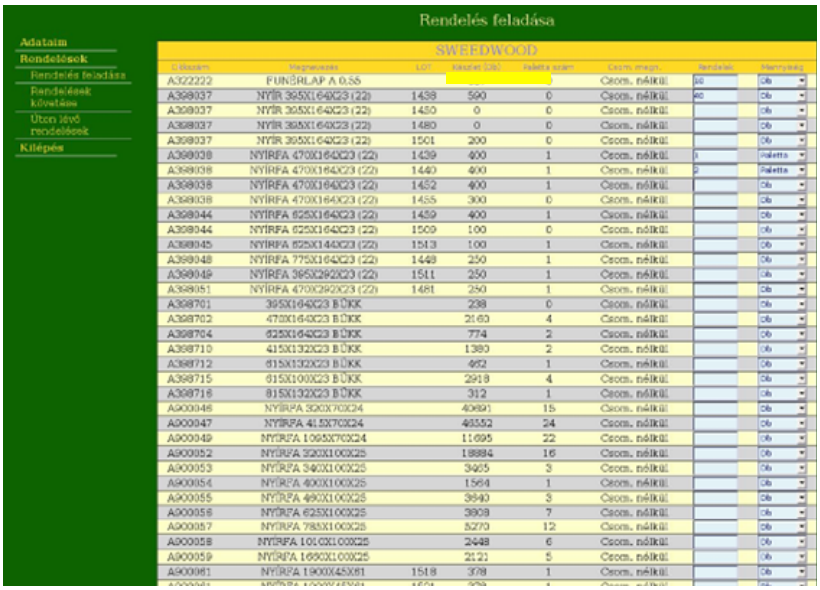

Figure 4: Selecting goods

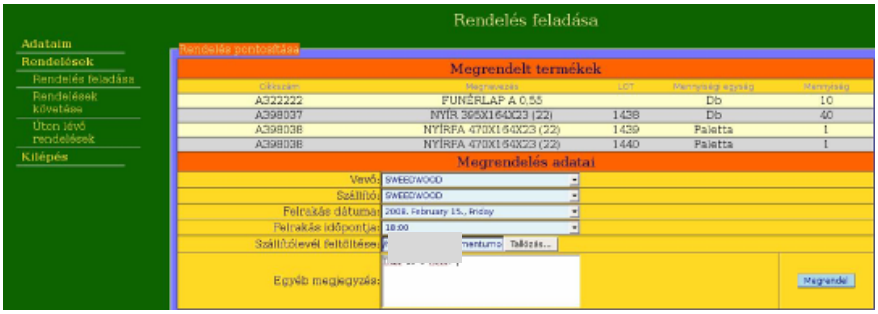

Figure 5: Placeing the order

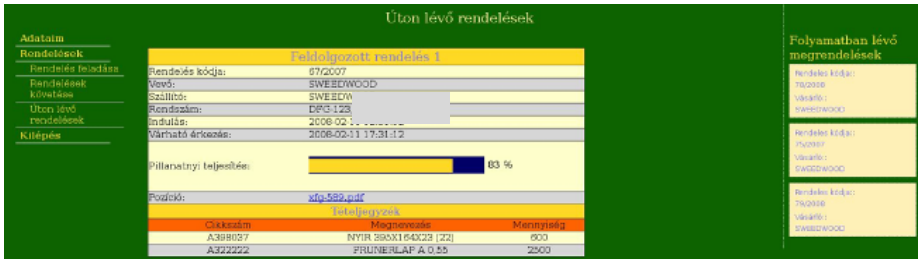

Figure 6: Completed orders

REFERENCES

[1.] www.linux.com
[2.] www.mysql.hu
[3.] www.phpstudio.hu