Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Computer system to replace the directory of professors in the lobbies at EETAC buildings

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Francesc Ribera Hidalgo

DIRECTOR: Dolors Royo Vallés

DATE: April 15th 2011

Títol: Sistema informàtic per substituir directoris de professors als vestíbuls dels edificis de l'EETAC

Autor: Francesc Ribera Hidalgo

Director: Dolors Royo Vallés

Data: 15 d'abril de 2011

Resum

Els edificis de la UPC allotgen una gran quantitat de personal: personal d'administració i serveis (PAS), personal docent i investigador (PDI), becaris, professors visitants, estudiants de doctorat, estudiants post-doc, etc. Sovint és necessari poder trobar a les persones en multitud de possibles escenaris. El cas més habitual es dóna quan un nou visitant a l'edifici necessita trobar el despatx on es troba una persona.

La tecnologia que es fa servir actualment a la UPC està obsoleta i dóna un servei molt inferior a les capacitats de les tecnologies informàtiques actuals: actualment es fan servir directoris sobre suport físic, en general murals. No cal dir que el suport físic té infinitat de limitacions que van des del caràcter estàtic de la informació continguda, fins al seu elevat cost de manteniment.

La possibilitat de fer servir tecnologies informàtiques expandeix exponencialment els usos que pot tenir un directori. En primer lloc ocupa poc espai. L'espai ocupat es limita al necessari per un terminal informàtic. Típicament un directori físic pot ocupar entre 1m2 i 4 m2 a les parets del vestíbul. No sempre és possible trobar la seva posició òptima en la distribució espacial del vestíbul. A més, el directori informàtic pot ser interactiu, de manera que només presenta la informació d'interès per l'usuari. Un altre avantatge del directori informàtic és la seva facilitat de manteniment i cost marginal d'actualització de la informació.

Per tant, l'objectiu d'aquest PFC serà crear una eina informàtica amb interfície gràfica d'usuari que permeti accedir de manera interactiva al directori del personal dels edificis de la UPC. Per fer-ho s'ha fet un estudi sobre les tecnologies existents i de com fer-les servir per al nostre propòsit.

En aquest projecte hem fet servir el gestor de continguts Drupal com base de l'aplicació, ja que les seves característiques són les que més s'ajusten a les nostres necessitat. També ha calgut desenvolupar un mòdul propi per tal de poder establir la comunicació amb la base de dades de la UPC.

Title: Computer system to replace the directory of professors in the lobbies at EETAC buildings

Author: Francesc Ribera Hidalgo

Director: Dolors Royo Vallés

Date: April 15th 2011

Overview

UPC buildings host a large number of staff: administrative personnel and services (APS), teaching and research staff (TRS), interns, visiting professors, phd students and post-doc students among others. It is often necessary to find people in a multitude of possible scenarios. The most common situation is when a new visitor to the building needs to find the office where somebody is located.

The technology used in the UPC is now obsolete and gives a very inferior service to the capabilities of current computer technology: currently used physical location panels are generally decorations. Needless to say, the hardware has numerous limitations that range from the static nature of the information, to the high cost of maintenance.

The possibility of using computer technology expands exponentially the possible uses of a directory. First it occupies little space. The occupied space is limited to the need for a computer terminal. Typically a location panel can take between 4 m2 and 1m2 on walls in the lobby. It is not always possible to find its optimal position in the spatial distribution of the lobby. In addition, the computerized location panel can be interactive, so that it only presents information relevant to the user. Another advantage of computerized panel is its ease of maintenance and the marginal cost of information updating.

Therefore, the objective of PFC is to create a tool with a graphical user interface that allows interactive access to the directory staff of the buildings of the UPC, also to do a study on existing technologies and how to use them for our purpose.

ÍNDEX

LIST OF FIGURES

INTRODUCTION

Now a days, with the continued evolution of information technologies, the tendency is to computerize systems and technologies, that until recently, were used in a manual way. The evolution of programming languages, such as OOP, the evolution of the Web or data storage systems, provides some resources with many advantages over traditional manual systems.

Data storage systems, such as medical records, police records, traffic management, banking and many others have been computerized and have become much more efficient systems, allowing users to streamline their transactions and perform them in a more secure way.

Both, through the evolution of communication systems, the Internet has become an indispensable tool as currently not conceived of a world without it. Web 2.0 appeared to provide dynamism and interactivity the old static Internet world. Its appearing led to the possibility of developing web applications, so users could access traditional desktop applications via a web server using only a web browser. The main advantage of Web applications is the independence of the operating system and the ease to update and maintain web applications without distributing and installing software to thousands of potential users.

Gradually, companies need to manage large quantities of news, articles or documents on the web, so that dynamic systems were developed to facilitated this task. The first content management systems (CMS) were developed by organizations that published a lot of content online, and needed continuous updates, such as online journals, newspapers and corporate publications.

Before the advent of content management systems, the content publishing on the web was slow, cumbersome and unwieldy. The proud owners of a website had no choice but to learn to use a web editing program like Dreamweaver or FrontPage, creating the html files one by one and then had to be uploaded to the FTP server. Many who were encouraged to set up a website for their personal or small business, fell victim to discouragement, due to the difficulty in updating internet content, filling outdated and unattractive to visitors. Currently, CMS allow users to launch their websites or web applications quickly and easily without prior knowledge of web programming. For this reason CMS has become an effective tool, which provides a high scalability and flexibility to any website.

The objective of this project is to develop a computer system with graphic interface that allows access to the detailed information of professors teaching at the university. To connect our system to the UPC professor-database, the UPC has enabled web service to perform the queries. In conclusion, the result will be a SOA application which replaces the way you view the teacher's location information (panels in the lobby), saving time and space, making it easier to upgrade and maintain.

This work is divided in five chapters. The first one presents all necessary concepts, both conceptual and technological, for the proper understanding of the project. The second chapter provides a comparative of the different options found, selecting the most significants, and detailing the option selected. Third chapter is focused in the design and the final implementation of the application. The results are discussed in chapter four, and, finally, chapter five summarizes the most relevant conclusions from the project.

CHAPTER 1. PREVIOUS KNOWLEDGE

This chapter explains some introductory concepts for the reader to have a prior conceptual vision of the project, following a technical explanation of the technologies commonly involved.

1.1   Concepts

1.1.1   Web application

A web application is a computer software hosted on a web server, which has access, restricted or not, through a network like the Internet or an intranet. The main characteristic of web applications is that the application should only be installed on the web server where all users can access through their browser. This is a great advantage as much ease maintenance and updating. The web applications can be accessed from anywhere where network connectivity allowing each user to acces their personal data and their personalized application from anywhere.

The traditional architecture of Web applications is a client-server architecture and layered architecture. In this architecture, each level is developed as independent modules, often with different platforms.

1.1.2   Layered programming

Layered programming is a programming style where the primary objective is to separate business logic from the presentation logic. This architecture is usually separated into three layers: presentation, logic and data layer.

All these layers can reside on a single computer, even to the most common as they are spread over the three layers (depending on the needs of each application). The business layers and data can reside on the same computer, and if the growing needs advise, can be separated into two or more computers. If the size or complexity of the database increases, it can be separated into several computers which will receive requests from the computer where the business layer resides.

However, if the complexity in the business layer is which forces the separation, the business layer could reside in one or more computers which would perform requests to a single database. In very complex systems come to have a number of computers on which the business layer runs, and a number of computers on which the database runs.
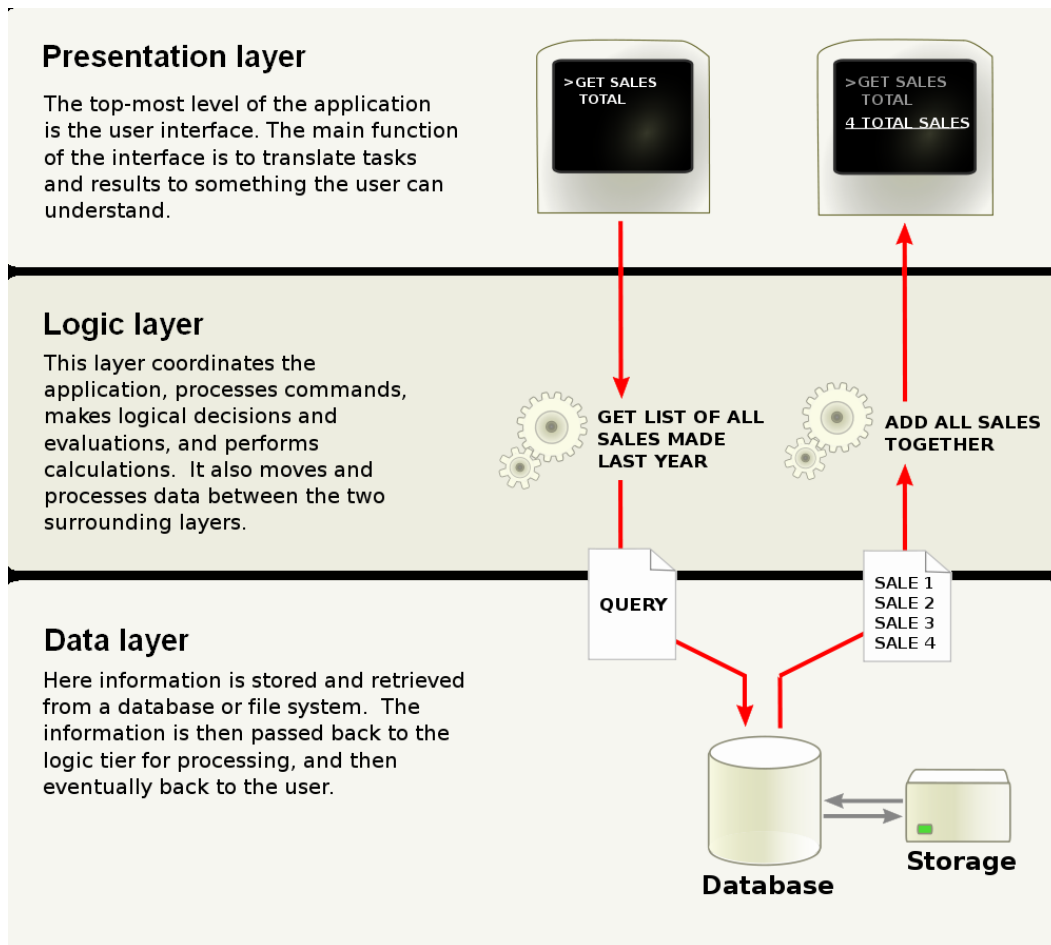
Fig. 1.1 Web application architecture. Three layer/tier architecture.

The main advantage of this architecture is that development can be done at several levels and if it was necessary to do some changes, it would only be necessary to modify the required level without affecting the other levels.

It is important to note that the terms of layers and levels not refer to the same concept. The layer term refers to how a solution is segmented from the logical point of view. In contrast, the level term corresponds to the way in which the logical layers are distributed in physical form. For example, a solution of three layers (presentation, business logic and data) that reside on a single computer is said to be the architecture of the solution would be three layers and one level. In contrast, a three-layer solution that resides on two computers (presentation and logic on one hand, logic and data on the other side) would be the architecture of the solution is developed in three layers and two levels.

1.1.3   WCMS (Web Content Management System)

A Content Management System (CMS) is software that allows you to create a support structure (framework) for the creation and administration of content. In particular, a Web Content Management System is a software system which provides website authoring, collaboration and administration tools designed to

allow users with little knowledge of web programming languages or markup languages to create and manage the site's content with relative ease.

Normally it is used to control a dynamic collection of Web material (HTML documents, images and other forms of media) and facilitates document control, auditing, editing, and timeline management.

1.2   Technologies

1.2.1   PHP

PHP (PHP Hypertext Pre-processor [1]) is an open source code interpreted high-level scripting language. PHP is inserted into the HTML document but it is a server language. This means that every time someone visits a page, PHP is executed on the server. This interprets the code and generates the resulting HTML content, which is what the user will see in his browser. This process is shown in the Fig. 1.2.
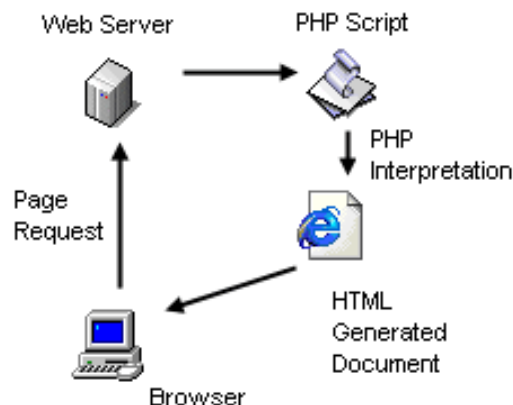
Fig. 1.2 PHP Request process

The goal of PHP is to create dynamic Web pages, providing the dynamism missing in the HTML. A dynamic website is a website that changes with time, the user, the user interaction, the context, the content, or any combination of the previous.

1.2.2   Web services

According to the W3C a Web service is "a software system designed to support interoperable machine-to-machine interaction over a network". Communication is usually done via XML based messages over the HTTP protocol, while the interface of the web service is described in a machine-readable format. Fig. 1.3 shows how a Web service works and which protocols are used.

Fig. 1.3 Web Service operation and used protocols

The party providing the Web service is called the service provider. It publishes all Web services provided via the Universal Description, Discovery and Integration (UDDI) protocol to a service registrar, which can be seen as the yellow pages of Web services. The service consumer, which uses the services provided by the service provider, discovers these Web services via the service registrar, again using the UDDI protocol. The service provider also provides a description of the Web services to the consumer, using the Web Service Description Language (WSDL). Basically, without a WSDL document, service consumers would not be able to use Web services properly simply because they would not know which methods, properties and communication method the Web services support.

Communication between the service provider and consumer can be performed using different protocols. A very well known protocol is the Simple Object Access Protocol (SOAP), but XML-RPC and REST (Representational State Transfer) web services are also used in practice. They only differ in the way messages are packaged.

1.2.3   SOA

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains.

SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. An endpoint is the entry point for such a SOA implementation.

CHAPTER 2. CMS SELECTION

This chapter will show what CMS have been selected. To do this, firstly the selection criteria is exposed, and the comparison between the most popular CMS that fulfill these criteria.

To have an objective opinion about each of the CMS selected, a simple test is done in order to compare their main characteristics. Some of the information of this chapter has been extracted from [2][3][4] and the CMS test had been done to verify some of the affirmations found in these documents.

2.1   CMS selection criteria

In order to decide which CMS platform will be used to develop our application, we will choose among all those that fulfill the following criteria:

- **Developed in PHP**. Although there are content management systems developed with other programming languages, PHP has the advantage that it can be deployed on most web servers and on almost all operating systems and platforms.

- **Free software**. The reason is not only to avoid the cost of a license, but also keeping it with an accessible code which can be modified to adapt the CMS if it were necessary.

- **Relevance**. Not always the most well known is the best, but using a CMS that is well known and has many users is beneficial to our purpose. We will have more components to suit our needs, we will have better support because there is more information on the Internet, we make sure the software will remain available and updating versions, etc.

Based on these criteria, three of them have been selected, the ones with greater acceptance and more widespread use. These CMS are: WordPress, Joomla! and Drupal.

2.2   CMS comparison

To perform the comparison between the three selected CMS, the following will be evaluated: the complements and modules, their usability and the existing documentation.

2.2.1   Complements and modules

One advantage of using a CMS is that you can use themes or templates for our website or web application to has a professional appearance without knowing anything about web design, and using the modules to provide a lot of features without programming skills. The most popular CMS provide more number of subjects and modules, making it easier to do what we need.

It is important to note that some of these allowances may be payable. In the case of Joomla!,it is commonly found that the higher-quality modules are not free, with costs typically between 20 and 200 euros. Drupal and WordPress publish free complements in its pages and with free licenses.

The Fig. 2.1 shows the data about the number of complements and themes of each CMS [5][6][7].
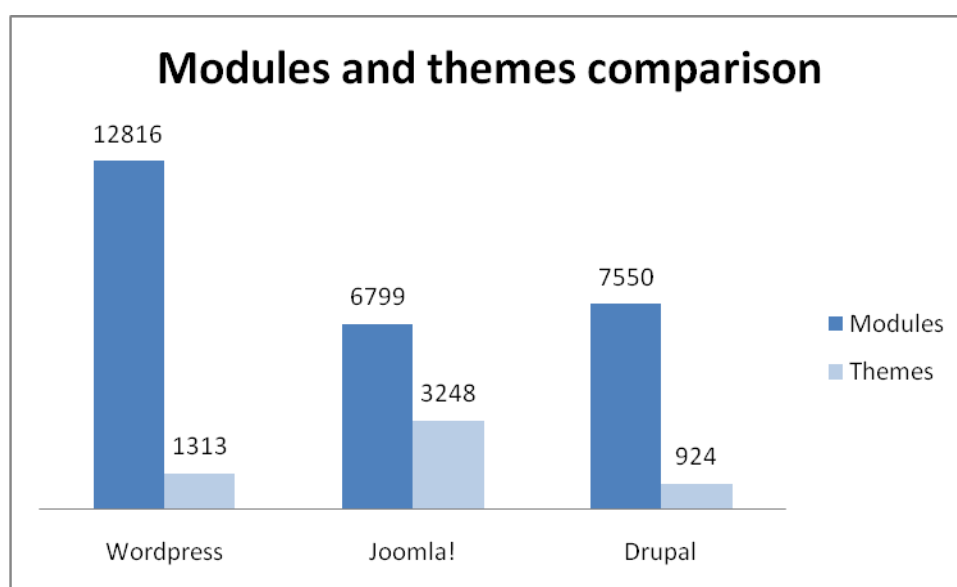

Fig. 2.1 Modules and themes comparison in 2010

Some conclusions can be extracted from the Fig. 2.1. WordPress is the CMS that has the major quantity of modules available. WordPress is an excellent system to use when creating a website that lets you quickly get your thoughts out on the web, but while it is often used as a blog manager, it can be configured to work in many other interesting ways as well.

Joomla! is the CMS that has de major number of themes and templates, but the appearance of the web application isn't the most important feature for the aim of this project. Despite having half number of modules than WordPress, surely Joomla! modules can make a web application much more functional than one made with WordPress.

Finally, Drupal has a little bit more modules than Joomla! but has much less than WordPress. Despite this fact, Drupal modules add new functionalities to the web applications with a very professional quality. On the other hand, it has the smallest number of templates of the three CMS. This is the weakest point of Drupal, though as said before, this isn't the most important feature for the aim of this project.

2.2.2   Usability

Although a user without programming skills and design may be able to create a website with any of the three systems, this CMS they are not equally easy to use.

2.2.2.1   WordPress

WordPress is more immediate to implement for a simple website, but if we want do something more complex; in comparison to the other CMS the implementation is more complex due to the blog manager simplicity. There are plugins that can be helpful to use WordPress as a CMS, but these have their limitations. However, the evolution leads to contemplate WordPress increasingly used as a generic CMS and not as manager of blogs, so we will see in the near future developments in this regard. It is possible to use WordPress as a common CMS with the latest versions using some plugins, but it is not really simple as in others CMS like Joomla! or Drupal.

On the other hand, modifying a template is relatively simple, and the organization can be very flexible, so we can see examples of pages made with WordPress and organization that look completely different content.

2.2.2.2   Joomla!

Joomla! has the problem of being too rigid in its content organization, with a two level hierarchical structure. This forces to plan exactly what content the website will have and how it will be structured, because it is not easy to change once we have a lot of content published.

The presentation of content is also more rigid than in other CMS, which means that some websites developed with Joomla! often look very similar and have very similar structure organization.

2.2.2.3   Drupal

Drupal is more difficult to install and use than the previous ones, the modules are more difficult to manage, but there are optional modules for creating authentic web applications without programming. If is needed to create a complex web, the effort spent learning to use Drupal certainly will be rewarded, because is a very complete and powerful CMS.

On the other hand, Drupal has a more complex presentation system, of the three CMS compared and it is not a coincidence that Drupal offers far fewer themes and templates than WordPress or Joomla!. However, it also offers great flexibility, so it may be worthwhile to spend some time to work on the design of our website.

2.2.3   Documentation

All three systems provide full documentation on its website. Anyway, there is enough information on the web, so if there is any problem the solution can be achieved in a few minutes doing a few searches.

On the other hand, in the website of each system, forums exist where issues are raised that are usually answered by other users or the developers themselves.

2.2.4   Summary

After the previous comments are made of each proposed CMS, a simple test is done to summarize the main characteristics of each CMS.

2.2.4.1 WordPress

WordPress is a great choice for a direct way to set up a web site. The system is easier to understand and install and is simple to maintain and update, allowing you to have configured the system without being a technology expert. Loads of modules and themes exist to ad optional functionalities. However, WordPress cannot scale to configure complex web pages as the other two systems. There is a basic distinction of roles, with no quick way to support publication or issue permits based on a section or type of content.

WordPress pros:

- Simple to use. No need for modifications.
- Excellent for blogging or sharing thoughts in a sequential manner.
- Even the most basic users can get the hang of it quickly.

WordPress cons:

- Not developer friendly.
- The forums tend to be less helpful .
- Upgrades tend to bring more problems than solutions.

This is not a good choice for the aim of this project due to its simplicity and we need to implement some complex features that WordPress cannot provide.

2.2.4.2 Joomla!

Joomla! is a robust system, good for different stages of development. It is relatively easy to install and configure. The system allows you to create a solid infrastructure for different web structures that allow complex and extensive web sites. The optional modules are not as numerous as in WordPress but have the necessary to enable the implementation of a wide variety of features.
Joomla! is not the best choice for sites that are produced by different types of authors. Although Joomla! supports more complex structures that WordPress is not as flexible as Drupal. Each of the content is usually associated with a page. This makes the system more comfortable to understand but limited advanced structures (such as arranging a website across multiple taxonomies). It is not easy, for example, to create your own type of content and then display it in different views on different pages on your website.

Joomla! pros:

- Friendly for all types of users - designers, developers and administrators.
- Huge community is awesome for assisting with creation of websites.
- Has been rapidly growing and improving itself.

Joomla! cons:

- Still not user-friendly enough for everyone to understand.
- Not quite as powerful as Drupal, and can be a bit confusing.
- Recently rebuilt the entire system from ground-up, and so there are still many out there sticking to the old versions

Joomla! seems to be a good CMS but I think it is not the best choice because is not enough flexible to implement the features and functionalities that the project needs.

2.2.4.3 Drupal

Drupal is flexible and powerful, and is a great option for complex sites. Supports a wide variety of web structures - you can define very detailed rules about what content you want to show where and build your own content type. It is easy for administrators to find and update the content and has a strong types of configurations through the management windows (has a huge variety of settings). But the power of Drupal brings more complexity. Understand what the system offers and how to configure is more difficult than with WordPress and Joomla!.

About the appearance, Drupal has a very hard presentation plane and is a bit difficult to implement a custom theme or template if you don't have a little knowledge about HTML and CSS. Despite this, it has a number of ready-made templates and themes, so changing the appearance is not a difficult task if you do not mind the fact that is not customized to suit the author.

Drupal pros:

- Extremely developer friendly.
- Strong community to help you.
- Can be used to create some really spectacular websites due to the fact that Drupal seems to have a pretty professional core development.

Drupal cons:

- Not very designer and user-friendly. It's hard for someone with little code knowledge to make the leaps required to do the very cool things that Drupal is becoming known for.
- Drupal theming often has some problems (until recently). Probably because developers, not designers, are who make the themes.
- Getting a Drupal website published could cost you more time than WordPress or Joomla!.

After analyzing and testing the three proposed systems, I think the best option is Drupal, allowing me to develop my own capabilities in case of not finding any that fulfill the project needed. The fact that has the worst presentation plane of the three CMS is not a determinant factor to discard it, because the appearance is not the most important factor for this project.

Once tested some of its most popular modules, I'm sure it's the best choice. In addition, the fact that in the CBL campus has already some applications implemented with Drupal is decisive, and it would be easier to integrate with these applications if it were necessary someday.

CHAPTER 3. SYSTEM DESIGN

Having chosen the CMS, this chapter shows the steps followed in the application design. Firstly the system design criteria will be explained and then the final design of the system is shown with an explanation of each part. During the design has taken into account the recommendations and examples suggested of some Drupal Books [8][9][10]

3.1   System design requirements

Before starting the application development, the following design requirements were raised in order to fulfill the project necessities in the most efficient way:

- **Robust application.** May be the most important feature of the system. It can seem obvious, but if the new technology fails more than the previous technology, the change has no sense. Also, having an application that doesn't work in the university lobby does not transmit a good impression.

- **Intuitive user interface.** To perform a search in the system, the user interface must be simple and understandable for any user. A person who uses the application for the first time must be able to use all the features without any difficulty.

- **Database operations.** How to access to the database is a fundamental problem in the web developing. In this case, the data is stored in a remote machine and this increases the complexity of the problem.

- **Navigation and style design.** All the web applications have to be clean, structured and must have a structured information design. There needs to be a consistent look and feel throughout the site. Users need to be able to immediately recognize their location on the site.

These design requirements, are known as a non-functional requirements. During the application development, attempts were made to adjust the application features to this design criteria within the possibilities offered by each of the technologies used.

3.2   System schema

As is explained in the introduction, the function of this system is to read the professors data from the UPC database. The queries to the UPC database are based on the users request and will be done via web service through the UPC network. The schema that implements the system to be developed is shown in the Fig. 3.1.
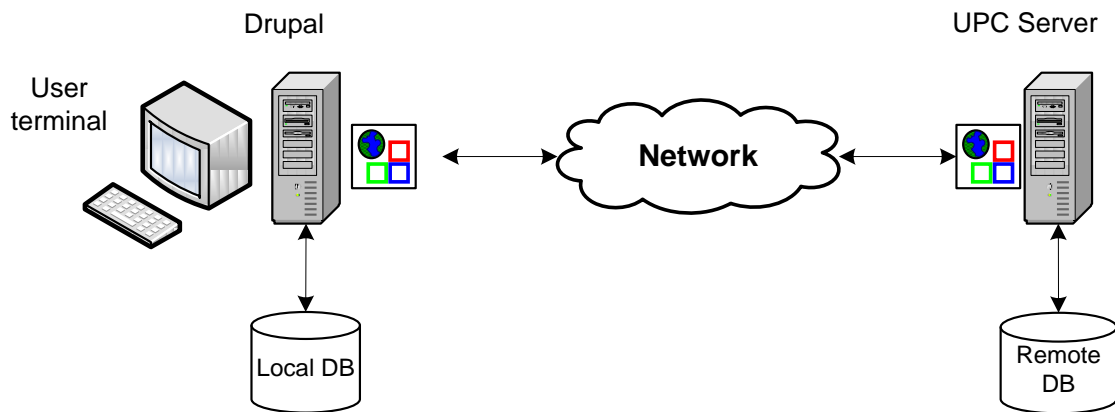
Fig. 3.1 Total system schema

In the system schema the elements involved can be appreciated. In the first part we can see the user terminal where the search into the database will be done. The Drupal will make the queries to its own database (Local DB) and display the information found in the terminal.

The communication between the Drupal computer and the UPC server will be done periodically in order to keep the local database updated with the data of the UPC database.

3.3   System layers

As explained in the section 1.1.1, the traditional architecture of web application is a three-layer architecture (Fig. 1.1). This system has been developed following this type of architecture and is tried to develop the system to be more modular as possible.

In this section the developments steps followed in each of the three layers will be explained.

3.3.1   Logic layer

As discussed in CHAPTER 2, the selected CMS is Drupal. Drupal system will manage the entire application logic and will interact with the data layer. The Drupal installation requires a local database installation and this fact is perfect to implement the data layer in this database. It implies that the interaction between Drupal and the data layer is included in the Drupal installation and is not needed to implement it.

3.3.1.1   Required modules

A new Drupal installation includes the necessary modules to create a web application but this will be quite simple and its possibilities are very limited. The new features are implemented by modules. Drupal is a modular application and

if you want to add a new feature, you only need to install a new module that implements it. For the aim of this project, new modules are needed which will provide Drupal the possibility to manage web services, databases and web forms.

In the Drupal community there are some existing modules which are exactly what is needed for the project necessities. These modules are:

- SOAP Client: Implements a SOAP client that allows Drupal to operate with an external web services [11].
- WebForms: Allows Drupal to create customized web forms to allow the data input to the system by the users.
- Views: This tool is essentially a smart query builder that, given enough information, can build the proper query, execute it, and display the results.

These modules are excellent modules but when I was testing them, I could not do the interaction between them. For this reason I had to develop my own module that includes the needed parts for the three modules exposed above.

3.3.1.2   *Use case diagram and user stories*

To start designing the new module (UPC_Services module) the user's necessities are needed to be known. User stories describe the functionality of a software system that brings value to the user and / or customer. In this case, the user and the customer is the same person or actor in the system. I could differentiate between different types of users that could use the application, for example: student, faculty, staff... But these users are essentially the same, because the operations that they can perform on the application do not differ from the rest of the users.
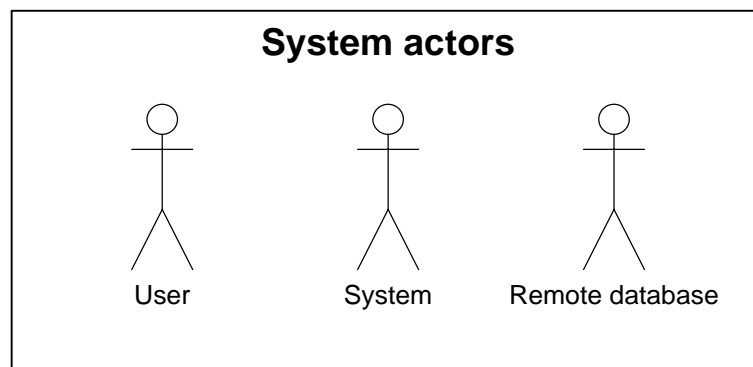

Fig. 3.2 System actors

In the Fig. 3.2 I've added two extra non-personal users because they use a functionality of the application which cannot be used by any other user (update local database). The process of updating the local database is a non-application

process; this process is done executing a script with the tasks scheduler of Microsoft Windows.
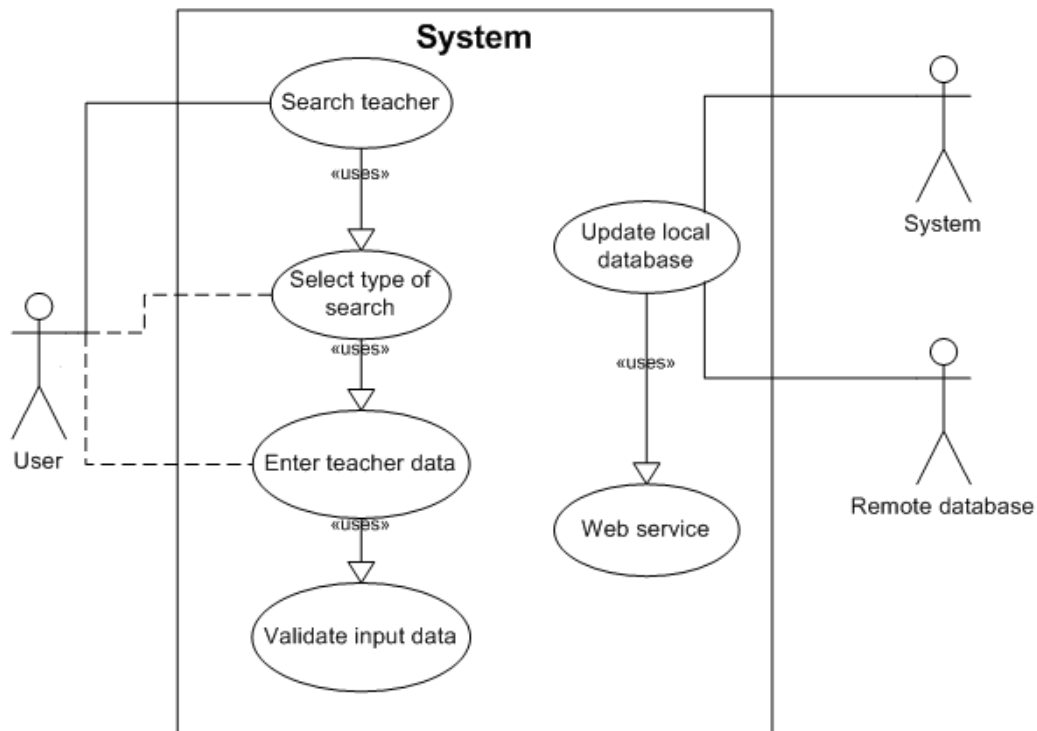

Fig. 3.3 Use cases diagram

In the use case diagram of the Fig. 3.3, the actors and the use cases of the application is shown. The search teacher functionality shows how the search by the user is performed. This sequential process needs the action from the user to finish and finally is done the data validation and the systems constructs the appropriate query to the local database.

User stories include motivations objectives much more specific than a use case. In fact, each of the paths of a use case diagram could be a user story. For this reason I will use the user story methodology to define which types of searches could be done by the user. The User Stories of the search teacher functionality are the following:

- As a user, I want to search for the teachers by their first and last names.
- As a user, I want to search for the teachers by their department.
- As a user, I want to search for the teachers by their office building.
- As a user, I want to search for the teachers on the complete directory.

I've proposed these user stories to the search teacher functionality because as a student, sometimes I seek this type of classification in the actual panels.

3.3.1.3 Data flow diagram

The UPC_Services module will contain a main form in order to allow the users to select the type of search to perform. After this selection is done, the module will redirect the user to the new form to allow the user to enter the data on the type of search selected. These actions are shown in the flowchart of the Fig. 3.4:
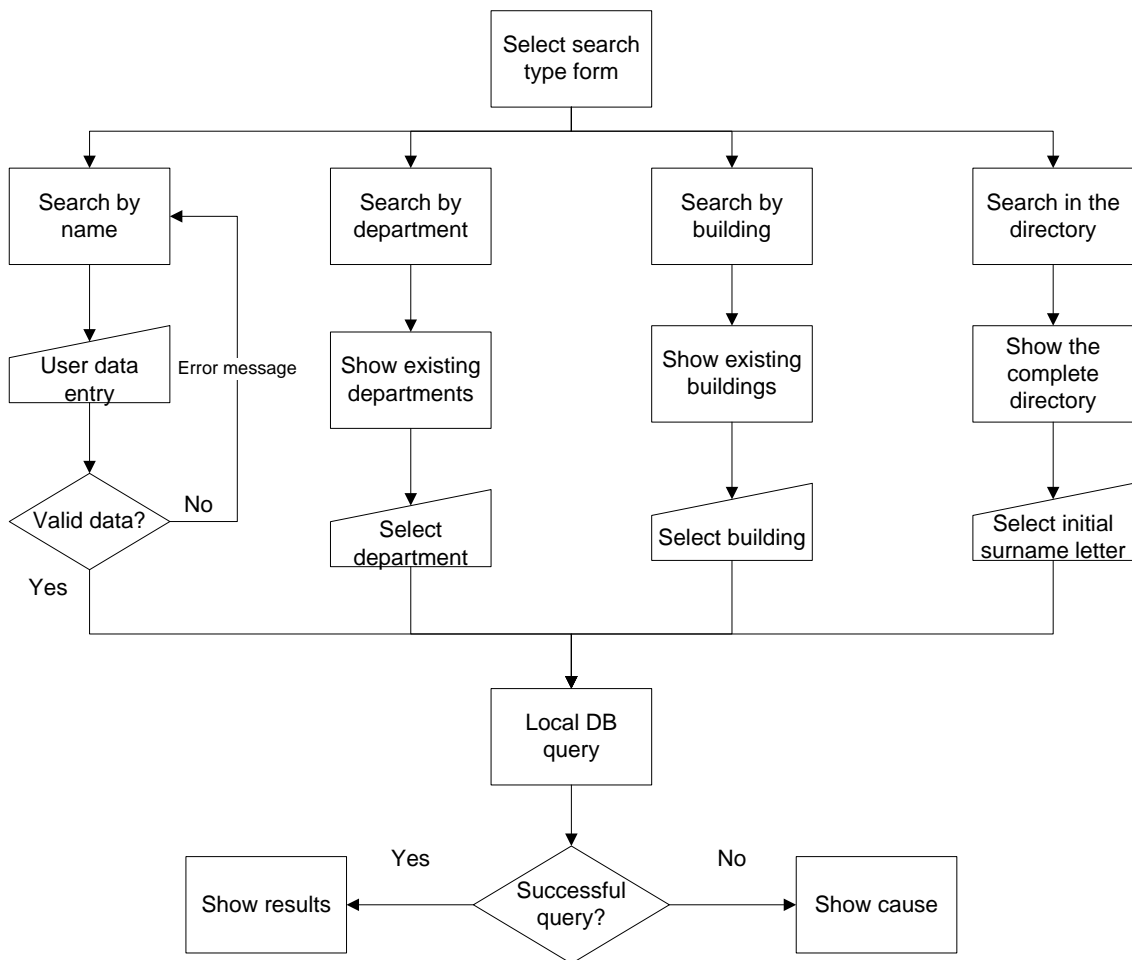

Fig. 3.4 Search process flowchart

Once the data is entered, validated, then the query to the local database is built. If the query obtains any result, they are displayed in a new form; otherwise the user is informed with the reason. Obviously, the database query function belongs to the data layer and the result representation function belongs to the presentation layer. The rest of the functions belong to the logic layer.

3.3.2   Data layer

The Fig. 3.1 shows that the data layer is divided in two parts: local and remote databases. Initially, was only contemplated to use the remote database in the data layer like is show in the Fig. 3.5.
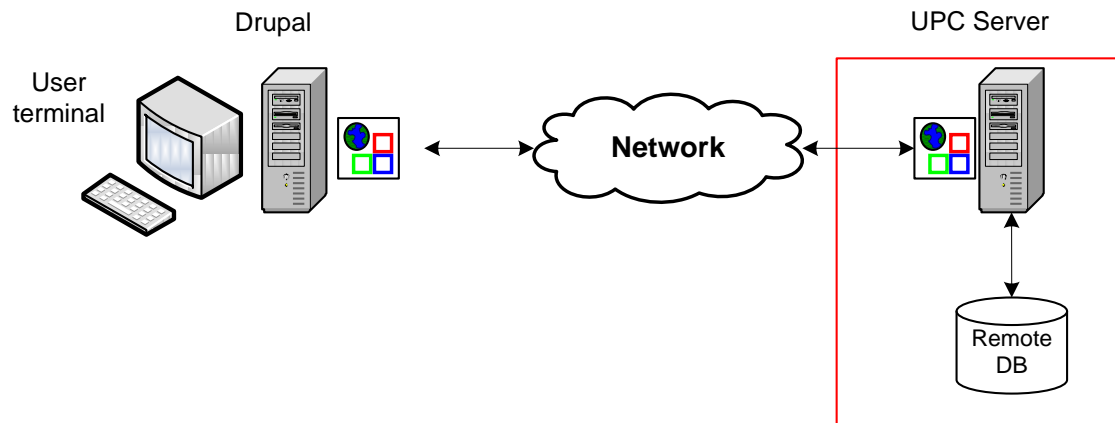

Fig. 3.5 First data layer design

Firstly accessing only to the remote database was thought to be sufficient for the requirements of the application. From the functional point of view is entirely correct, but it was discarded for being inefficient.

This is an inefficient design because querying a remote database for each of the requests that are performed in the terminal, has no sense. The amount of data is not very large and in our case, data from the database are not often updated. Once obtained the information from the remote database, it makes sense to store them locally, so we do not charge the remote database with repetitive queries.

On the other hand, querying a remote database will be subject to the state of the network and the database workload. For this reason, this option was rejected.

Then was decided to create a local database in the same place where the user terminal will be placed. Creating a local database is avoided dependence of the network status and the workload state of the remote database. Obviously, to maintain the local database consistent with the remote data, regular updates are needed to be done to the remote database in order to update only the fields which need to be updated.
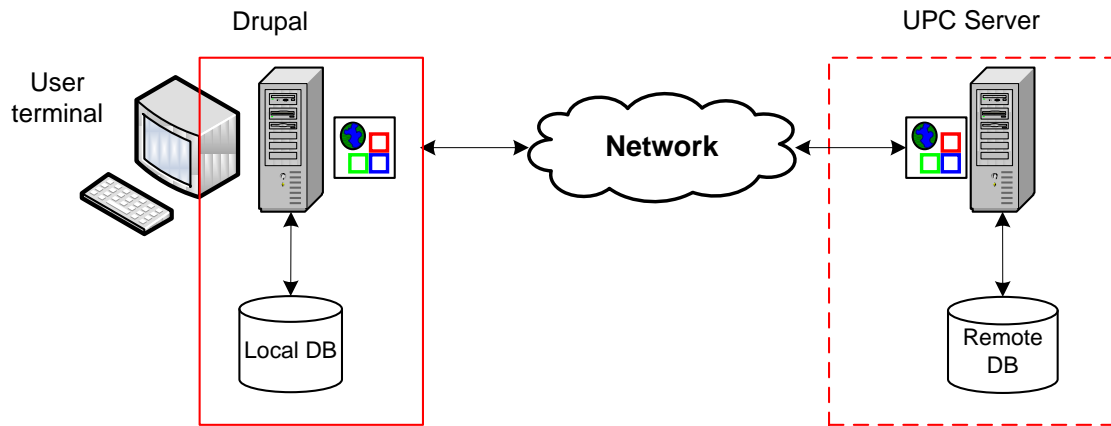
Fig. 3.6 Final data layer design

With this data layer architecture, only a database updating web service function is needed. The UPC only has to enable a web service to permit the Drupal web service client to do the data update.

In the Fig. 3.6 the final data layer design is shown. In this figure, the remote database appears with a dotted line. The reason is because this database is not considered a part of the data layer because the Drupal application will never do any type of operation over this database. For the Drupal application, this database not exists, although this database is the real source of data.

The function that performs the queries to the local database is the same because each type of query is constructed out of the function and the query function only does the connection to the database, send the query and return the result. The result returned by this function is the new form generated and Drupal only has to paint it in the user interface. Here is the function:

```
function doDBQuery($sqlQuery, $formName)
{
        //Connection and connection validation statements
        //Data processing and data validation statements

        //Form construction statements

        return $formName;
}
```

The local database has only one table to store the teachers' data. This table has the following columns:

- name: The teacher name. Is a index field.
- surname1: The first teacher surname. Is a index field.
- surname2: The second teacher surname.
- department: The department of the teacher.
- email: The e-mail of the teacher.
- office: The location of the teacher's office.
- campus: The university campus where is the teacher office
- telephone: The teacher's office telephone.
- PDI_ID: An internal teacher identifier. Is an auto incremental field (primary key).

The name and surname1 columns are an index field in order to have an efficient access of ordered records. For the local database, a MySQL 5.1 database management system (DBMS) was chosen with an InnoDB storage engine. It is the most common database system and is the standard database used by Drupal.

3.3.3   Presentation layer

The presentation layer creates a different form depending on the user selections. For each user action, the presentation layer creates a new form, but when this layer has to represent the results of each type of query, the query sender form, is recycled by the logic layer clearing its content in order to create the new content to display.

The reason to do that is by the way that Drupal has to represent the web forms. Each web form has some properties, and one of these properties is the submitted values. If I recycle the submitted forms, I don't need to redirect the user to a new url and send the submitted values by POST or GET.

In the following points, the screen sequence of the application and the used template will be described.

3.3.3.1   UPC template

In order to give an institutional appearance to the application, I have used a page style similar to the UPC home page style. The reason is because this application will be placed in the lobbies of the buildings of the UPC and if I use a custom page style this may seem to be a little bit serious.

On the other hand, following the fourth design requirements exposed in the point 3.1, I've tried to use a template design as simplest as possible to show the information in a clearly way.
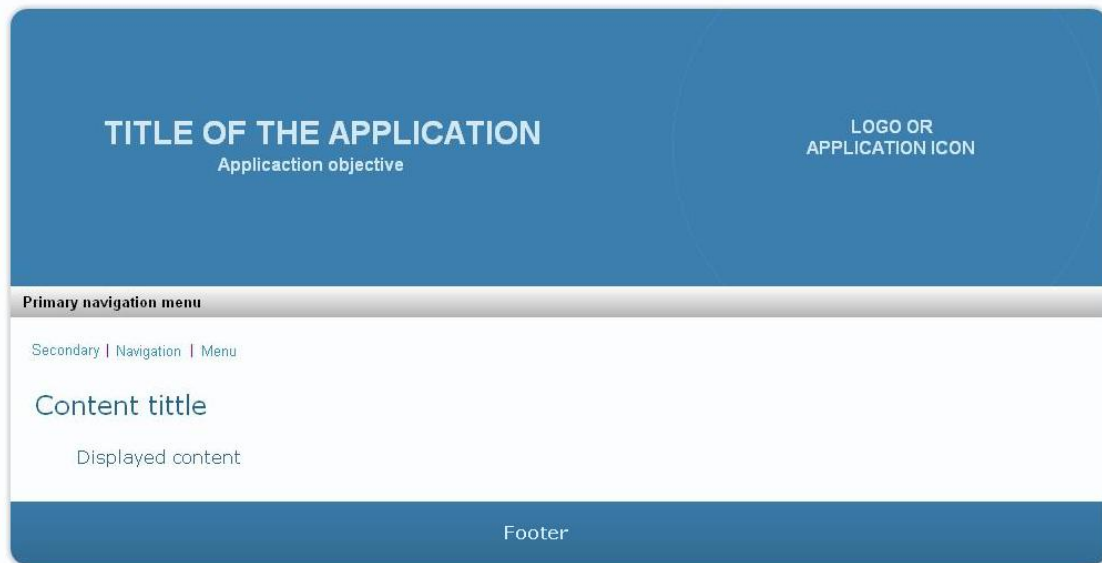
Fig. 3.7 Template design

The template showed in the Fig. 3.7 has the following elements:

- **Header:** Contains the title and subtitle of the application and the university logo.
- **Primary navigation menu:** Used to navigate through the application functionalities.
- **Secondary navigation menu:** Used to navigate through each of the elements of the functionality. Showing where the user is at present moment and allows them to return back on the selection.
- **Content title:** Title of the displayed content.
- **Displayed content:** Displayed content information.
- **Footer:** Information about the application and author.

CHAPTER 4. IMPLEMENTATION AND TESTING

In this chapter, firstly, the implementation of the application is explained, in particular how is the user interface and the options that are showed to the user.

In the second point, it is explained the testing method that had been used to test the application with real users, taking care their opinion about the usability of the application. Its results are commented.

4.1   User interface

4.1.1   Home page

As explained in the section 3.1, the user interface has to be intuitive. In this point is explained, the user interfaces that implements the search process described in the search process flowchart in the Fig. 3.4.

Obviously, the first screen that any user will find is the home page of the application that not contains any type of content. The home page contains the welcome text of the application and the primary navigation like is shown in the Fig. 4.1.
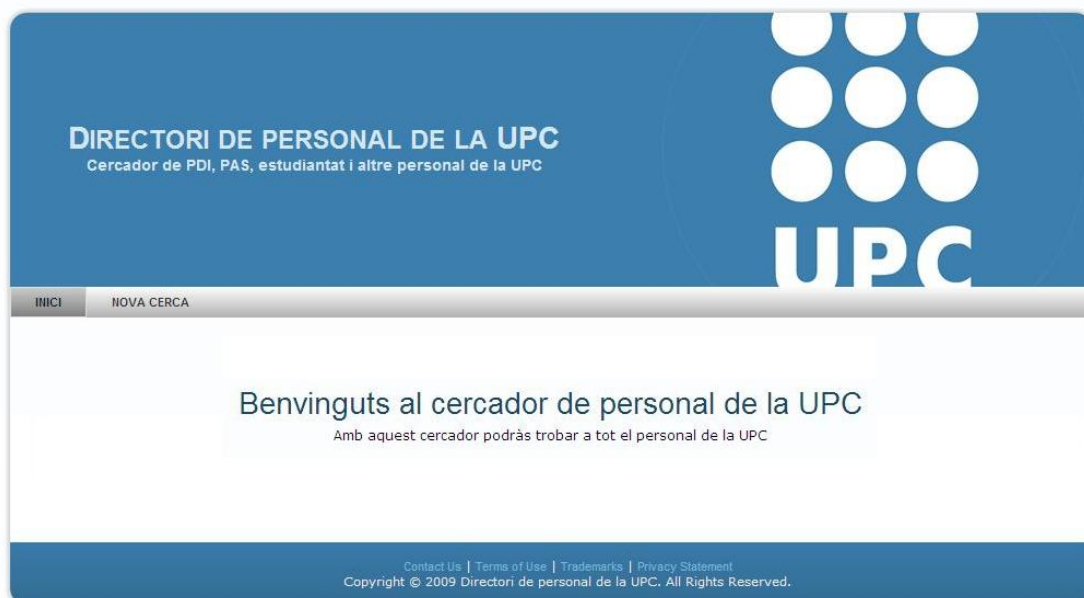

Fig. 4.1 Application home page screen

In the home page, the user can only select the new search option located in the primary navigation menu. After that, the user will be redirect to the select search type form.

4.1.2   Search type forms

For the following screenshots, the header has been removed in order to reduce the image size. The first search type form commented is the select search type form (Fig. 4.2).

Fig. 4.2 Search type form screen

In this page, the user can choose the search criteria exposed in the section 3.3.1.2 as a user stories. These options are shown in a submit buttons control and each of these buttons will creates the new form of the search type selected.

Fig. 4.3 Search by name form

In the Fig. 4.3 is shown the search by name form. This form allows the user to enter the name and the surname of the professor. Only letters are allowed in this entry. The form validation function will throw an error if another character is introduced by the user like is shown in the Fig. 4.4, otherwise the search result is displayed in the search result form.

Fig. 4.4 Search by name form validation error

If the user selects the search by department option, the new form will list the actual UPC department like is shown in the Fig. 4.5. When the user clicks any of the departments button the result is displayed in the result form.

Fig. 4.5 Search by department form

The search by building form shown in the Fig. 4.6, is similar to the search by department but can display more results than the search by department because a university building could contains more than one professor in the department.

Fig. 4.6 Search by building form

The last search type form is a little bit different from the rest. The complete directory form, will display all the alphabet letters and the first time that is displayed shows the search result for the letter A.

Fig. 4.7 Search on the complete directory form

In the Fig. 4.7 is shown the search on the complete directory form but in this case, there are no results for the letter A and it is notified to the user by a message under the alphabet buttons.

4.1.3   Results forms

In this section are commented all the different types of result forms with their particularities. All the result forms are the same result form except the directory search form, which doesn't have a result form because its result form is itself.

Fig. 4.8 Search on the complete directory result form

In the Fig. 4.7 is shown a search with no result and in this case, in the Fig. 4.8 is displayed one result for the letter R. All the results are displayed into a collapsible fieldset in order to represent only the information that the user want.

For example, one search can provide more than 10 results to the user, and in this case, display all information of the professors in the same page has no sense. If the user wants to view the teacher information, only has to click over the teacher name and the data will appear under the teacher's name.

The result search form of the other search types is shown in the Fig. 4.9. In this case is shown how is the appearance of this form with a lot of search result and one of the fieldsets collapsed.

Fig. 4.9 Search result form with a non collapsed fieldset

This is a clear way of displaying data results, but the main reason for designing like this is related with the design requirement of the database operations.

With this style of result representation, the form has all the data queried to the local database, but even though the user only can see the related data to the professor's name selected. Doing that, the application save a number of queries to the data base and only need to do one query and store the information in the fieldsets.

4.2   Testing

In order to evaluate the usability of the application, Computer System Usability Questionnaire (CSUQ) has been used [12]. The CSUQ has a greatest

acceptance and also it's appropriated to this application type. The questionnaire has been done by 42 users and after analyzing the answers, the scores were calculated based on the rules shown in the Table 4.1.

4.2.1   Computer System Usability Questionnaire (CSUQ)

CSUQ consists of 19 items aimed to address the following system usability characteristics: quick completion of work, ease of learning, high-quality documentation and online information, functional adequacy and rapid acquisition of usability experts. Each of the 19 items can be scored with 7 score levels. I've inverted the score levels because I think it is clearer. In the original CSUQ, the strongly agree rating has the less punctuation (1 point) and I think it is more clear if it has the maximum punctuation (7 points).

The answers of the CSUQ items can be calculated into four score levels:

- OVERALL: The overall satisfaction score.
- SYSUSE: System usefulness.
- INFOQUAL: Information quality.
- INTERQUAL: Interface quality.

The rules to calculate the CSUQ scores are showed in the Table 4.1:

Score name	Average the Responses to:
OVERALL	Items 1 through 19
SYSUSE	Items 1 through 8
INFOQUAL	Items 9 through 15
INTERQUAL	Items 16 through 18

Table 4.1 Rules to calculate CSUQ scores

4.2.2   CSUQ results

The results had been represented in graphs for an easy interpretation. The four score types are represented in a circular graph and finally is represented the average grades of each of the score types.

For the OVERALL score results showed in the Fig. 4.10, it can be appreciated that only the 2% of the total scores of the questionnaire had been scored with a score lower than 4. Otherwise, the 59% of the total scores were the maximum grade. It only means that over all the users' responses, the 59% were a 7 points grade and only a 2% were lower than 4. It is a very good result, but is no determinant in order to evaluate the quality of the application.
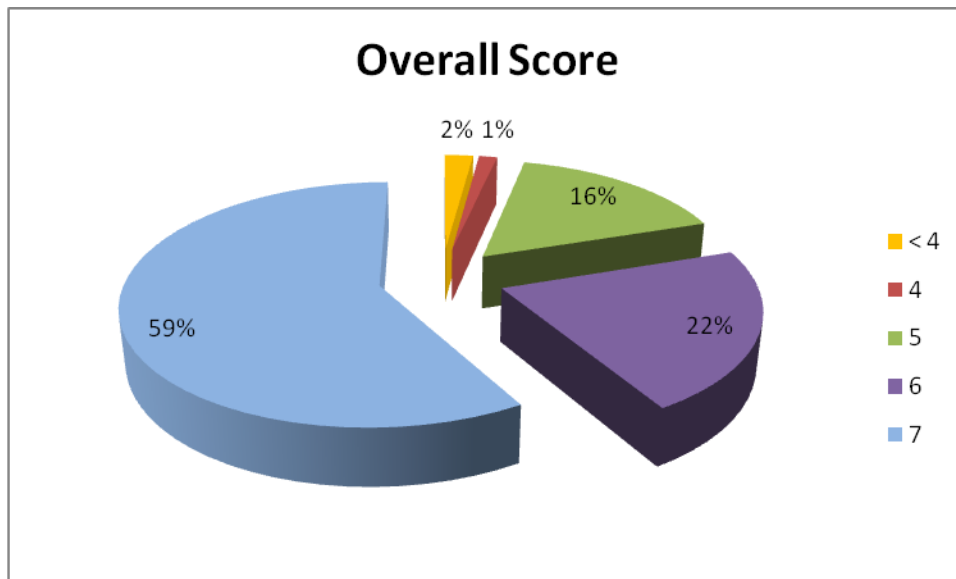
Fig. 4.10 OVERALL score results

From my point of view, the other three scores (SYSUSE, INGOQUAL and INTERQUAL) are much more important in order to evaluate the application quality. In the Fig. 4.11, are shown the SYSUSE score results. These results are very important because they indicate the users' satisfaction with the application usability.
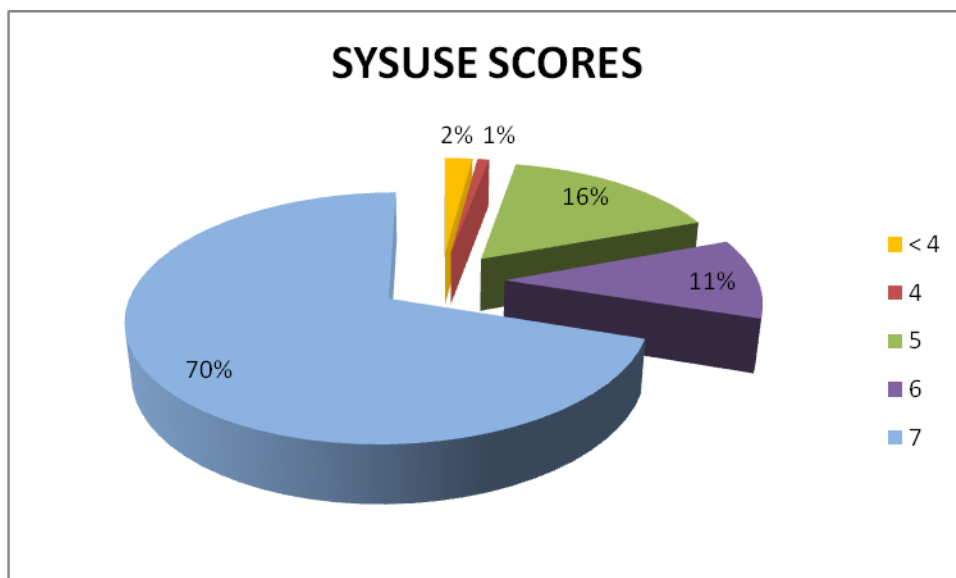
Fig. 4.11 SYSUSE score results

In the section 3.1 are exposed the system design requirements, where one of them is the intuitive user interface. It is very important to fulfill the design requirement because it allows users to use the application without any type of prior training. The 70% of the users have graded this feature with 7 points, and only the 2% of the users have graded this feature under the 4 points. This result

means that the system is very easy to use without any type of previous knowledge and is very easy to finalize all the tasks without any problem.

For the INFOQUAL score is obtained a similar punctuation to the SYSUSE score. In this case, the user has grade the information quality showed by the application with 7 points in the 72% of cases and only 1% of the grades ara under the 4 points.
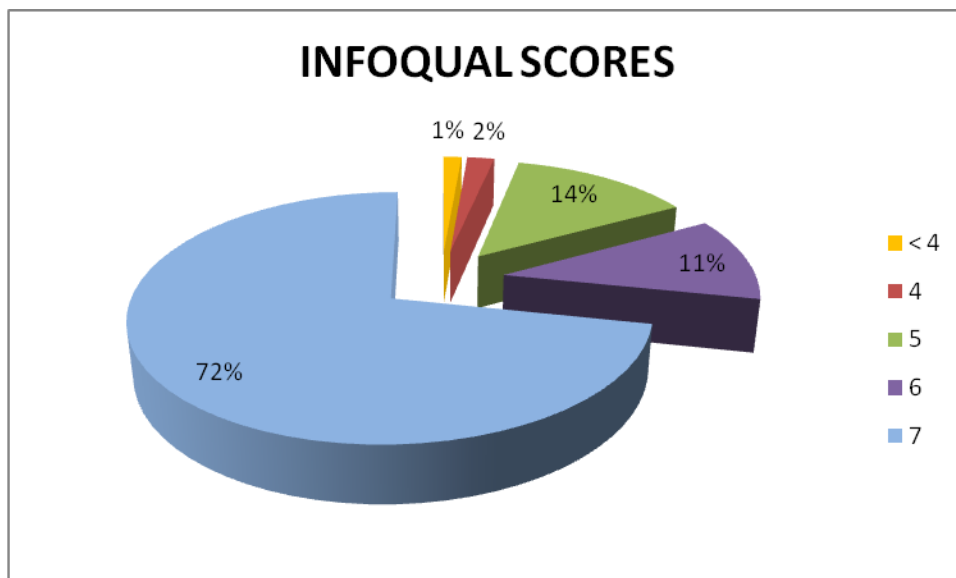

Fig. 4.12 INFOQUAL score results

These results represents that the users are strongly satisfied with the information shown by the application when their actions are successfully complete and when the application show any type of error.

I think that this result is very important because if the user is satisfied with the information displayed, he doesn't need to do any type of question in order to finalize their searches. For example, if any user input an invalid character in the teacher's name or teacher's surname, the displayed error has to be very clear in order to allow the user to understand the error and correct it.

Otherwise, the displayed results has to be atractive and esary to understand by the users to ensure their satisfaction using the application and to avoid the query repetition or the stop of using the application for an unclear displayed result.

Finally, the INTERQUAL score represents the user interface quality evaluation. This score seems to be extremly related with the SYSUSE and INFOQUAL scores but it only evaluates the items that the user use to interact with the system (like controls, images, screen design,…).

Fig. 4.13 INTERQUAL score results

The Fig. 4.13 shows the INTERQUAL score results. In this case, the major scores wher 6 points with the 65% of the grades. But this score has the worst punctuation under the 4 points with the 5% of the punctuations.
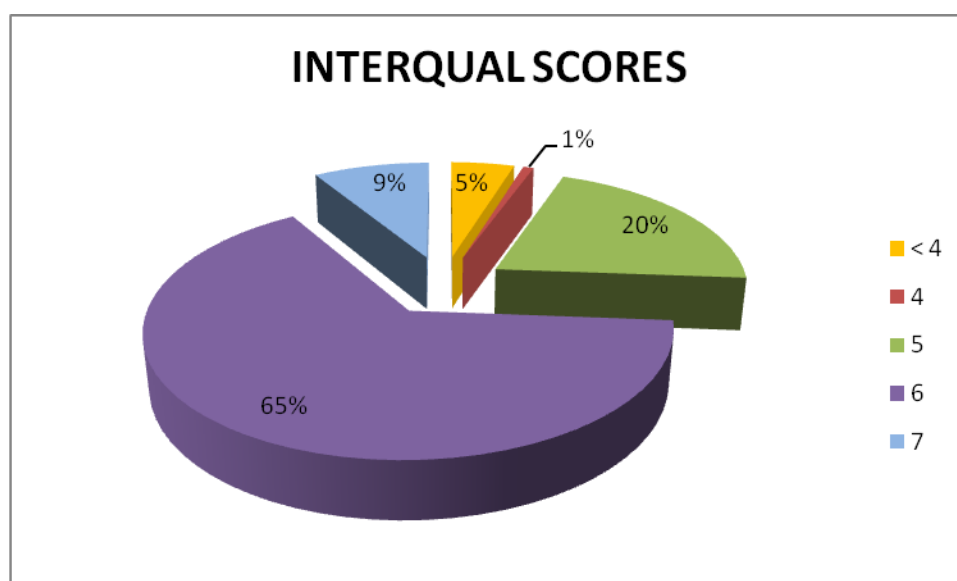
These results not seems to be a worring data, but in the 16th, 17th and 18th questions have been those that have had the most number of comments and some of them quite negative.

For example, one comment that caught my attention was a reference to the controls used in the complete directory search form (Fig. 4.7). This comment says that the buttons used to show alphabet were too small and generally he did not like the effects that had the application. This user was strongly disagree in the three question about the user interface of the application.

For example, on the result forms showed in Fig. 4.8 and Fig. 4.9, the user can show and hide information from the professors by clicking on its name. In the application development I have not used any kind of effect (I do not use JavaScript or other scripting technologies) and what is mentioned in that comment as effects, are simply properties of the controls. The "efect" of show or hide teachers' information by clicking on its name, is simply a fieldset control property.

On the other hand, about the button size of the alphabetthe, I decide to keep them like this becauseI don't know if the application will be used with a touch screen monitor or will be used with a mouse and keyboard.

To conclude this section, in the Fig. 4.14 are shown the average grades of the application. These average grades are the average punctuation of each of the CSUQ scores.
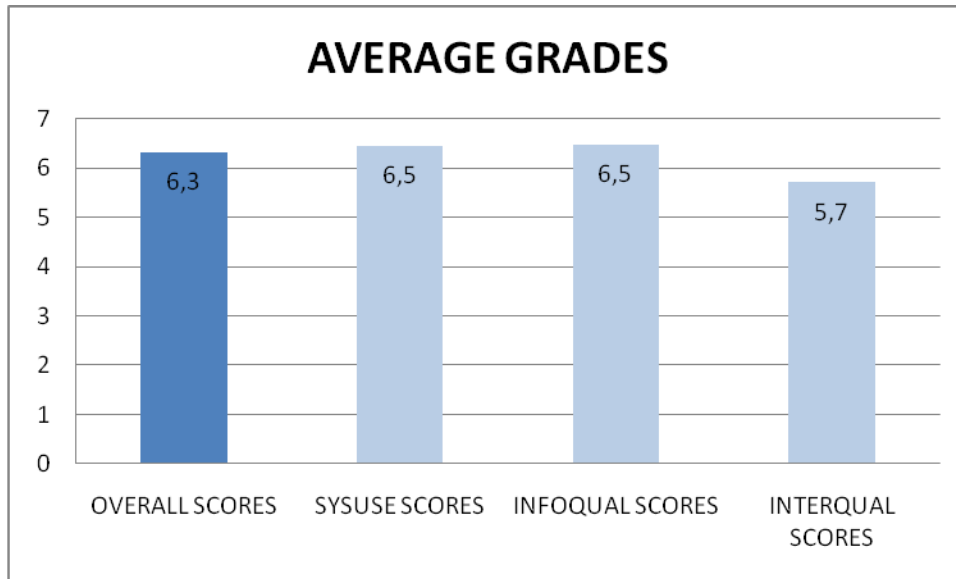
Fig. 4.14 Average grades of the application

The conclusion that can be extracted from this result is that all the users who have completed the questionnaire, like the application. In average, they have scored the OVERALL score with 6.3 points over 7, and it is an excellent result. But in the INFOQUAL score, the average grade was 5.7 over 7, which is a very good numeric result, but this is the score that have had the worst user comments. It is necessary to consider these comments as they can help me to improve application functionality and quality of the user interface, once the application is deployed in the building lobby's.

CHAPTER 5. CONCLUSIONS

In this last chapter, I have summed up, from a critical point of view, all the things done in this master thesis. And finally, some future work is proposed.

5.1   Overview

As a result of this master thesis has explained the concepts and features of the Drupal CMS, and its possibilities. It has been made a search and comparison of the three CMS with greater acceptance and more widespread use: WordPress, Joomla! and Drupal. The great debate is between Joomla! and Drupal. Both CMS are, from my point of view, the two open source CMS more complete that are currently available. But they have considerable differences.

During the project development was necessary to develop my own module. The existing Drupal modules were very good choices, but I was not able to perform the interaction between them. In particular, for the SOAP client module there was very little documentation and after checking the source code I was unable to find how to instance it from the WebForms module.

During the development of the module, the UPC did not enable any web service to carry out the appropriate queries to their database in order to update the local database of Drupal. So in order to make my module tests I had to develop my own web service and let the update feature made without any real test passed. This function is made without knowing what would be the role of the UPC, it will therefore be necessary to adapt to the Web service function when it will be enabled.

When the application development was completed, user test were performed. I provided them a questionnaire and their ratings and reviews were taken into account in assessing whether or not, to perform design changes. Since the results have been very positive, it was not considered necessary to make any changes. However, I have paid attention to every point of criticism.

After the test, I can conclude that the objectives outlined in the section 3.1 have been met and therefore the project as well.

5.2   Review

After all the CMS comparison made, all three CMS are a good option to do your own web application, even though each is more oriented to a certain application. Below is an overview of each one of them.

5.2.1   WordPress

WordPress is extremely easy to configure and manage. It allows you to get your own website very quickly. It's oriented to be a blog manager but its recent posts plugin can provide WordPress webs many more functionalities. Although in this project I have rejected it very quickly, it doesn't mean that WordPress is a bad CMS, but simply means that it was not the best option for my application.

5.2.2   Joomla!

Joomla! is easier to configure and manage than Drupal, besides possessing a lot of themes and templates that make it visually attractive. Joomla! is a CMS thought for all type of users, whether novice users, advanced users, designers or developers.

5.2.3   Drupal

Drupal is probably much harder to configure, yet its setup menu is fairly intuitive. But its learning curve is higher than the other two. Unlike WordPress and Joomla! this CMS is geared for users with a basic knowledge of programming in PHP. Once you understand its operation and its potential is discovered, becomes a powerful tool in the development of web applications. Its nucleolus is developed in a very professional way and its community of users and developers bring warm modules and accessories that let you add a lot of new features to your web application.

5.2.4   Conclusion

For this project I chose Drupal because I think it is more appropriate for the aim of this project and fulfills the project requirements in a better way. In the EETAC an application made with Drupal already exists and it is another reason for my choice because would be easier to adapt them if it was necessary.

Developing my own Drupal module I think that the Drupal's API is very simple because it is a modular application. What I mean is that if we extrapolate to the OOP, Drupal core acts as an interface and the modules would be as the implementation of this interface. When implementing the modules, Drupal has hooks that are responsible for rendering the content on the site where you want as specified. Drupal's API has a great documentation and a lot of tutorials and high quality examples.

5.3   Future work

In this last section, some future work is proposed in case of continuing with the all achievements done in this master thesis.

I think it may be interesting to change the way I create the forms in the module. For example, I propose to use AJAX because the application would gain in appearance and quality of presentation. Drupal has an API for the AJAX functions implementation.

Another improvement for the application would be allowing professors to add comments to the available information. For example, if a student is looking for a professor in the application and in that moment he is out of the office, the professor would be able to show when he will be back again.

For the web service, I think the best option is that the UPC has a timestamp in its database to check the latest update of a record. If this timestamp exist, the UPC would only have to implement a function in which you pass the timestamp as a parameter and the function returns an array or array of objects which contain the records to update in the local database.

Continuing with the web service, I think the best option would be a REST web service with JSON data interchange format. JSON is lighter than XML. JSON is based on JavaScript and thus would be easier to use and exchange of data between PHP and JavaScript.

REFERENCES

[1]   PHP Group (2001). *PHP: Hypertext Preprocessor*. Retrieved December 2010, from http://www.php.net

[2]   Southwest Interactive (2009). *The competition: The Ultimate Showdown of Content Management System Destiny*. Retrieved December 2010, from http://www.cmsshowdown.com

[3]   New Local Media (2008). *A Comparison of Capabilities and Features on Drupal, Joomla! And Wordpress*. Retrieved December 2010, from http://www.newlocalmedia.org/downloads/file/9-cmsfeaturecomparison

[4]   Good web practices (2009). *Which CMS should you choose?.* Retrieved December 2010, from http://www.goodwebpractices.com/other/wordpress-vs-joomla-vs-drupal.html

[5]   Mullenweg, M., Boren, R., & Caoimh, D. O. (2003). *WordPress*. Retrieved December de 2010, from Blog tool and Publishing Platform: http://WordPress.org

[6]   OpenSourceMatters. (2005). *Joomla!* Retrieved December de 2010, from http://www.Joomla!.org

[7]   Drupal Association (2001). *Drupal - Open Source CMS*. Retrieved December de 2010, from http://drupal.org

[8]   Byron, A., Berry, A., Nathan, H., Eatton, J., Walker, J., & Robbins, J. (2008). *Using Drupal.* (J. Steele, Ed.) Sebastopol: O'Reilly Media.

[9]   VanDyk, J. K., & Westgate, M. (2007). *Pro Drupal Development.* (M. Wade, Ed.) Berkeley: Apress.

[10]   Mercer, D. (2008). *Building Powerful and Robust Websites with Drupal 6.* (A. Deobhakta, Ed.) Birmingham: Packt Publishing.

[11]   James, T. (2010). *Drupal Web Services.* (A. Kumar, Ed.) Birmingham: Packt Publishing.

[12]   Lewis, J. R. (1993). *IBM Computer Usability Satisfaction Questionnaires.* Technical Report 54.786, IBM Corporation, Human Factors Group, Boca Raton, FL. http://drjim.0catch.com/usabqtr.pdf

ANNEX

The Computer System Usability Questionnaire (CSUQ)

This questionnaire gives you an opportunity to express your satisfaction with the usability of your primary computer system. Your responses will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you.

To as great a degree as possible, think about all the tasks that you have done with the systemwhile you answer these questions.

Please read each statement and indicate how strongly you agree or disagree with the statement bycircling a number on the scale. If a statement does not apply to you, circle N/A.

Whenever it is appropriate, please write comments to explain your answers.

Thank you!

---

1. Overall, I am satisfied with how easy it is to use this system.

STRONGLY							**STRONGLY**
DISAGREE 1	2	3	4	5	6	7	**AGREE**

COMMENTS:

2. It is simple to use this system.

STRONGLY							**STRONGLY**
DISAGREE 1	2	3	4	5	6	7	**AGREE**

COMMENTS:

3. I can effectively complete my work using this system.

STRONGLY							**STRONGLY**
DISAGREE 1	2	3	4	5	6	7	**AGREE**

COMMENTS:

4. I am able to complete my work quickly using this system.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

5. I am able to efficiently complete my work using this system.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

6. I feel comfortable using this system.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

7. It was easy to learn to use this system.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

8. I believe I became productive quickly using this system.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

9. The system gives error messages that clearly tell me how to fix problems.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

10. Whenever I make a mistake using the system, I recover easily and quickly.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

11. The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

12. It is easy to find the information I need.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

13. The information provided with the system is easy to understand.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

14. The information is effective in helping me complete my work.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

15. The organization of information on the system screens is clear.

STRONGLY   **STRONGLY**
DISAGREE   1   2   3   4   5   6   7   AGREE

COMMENTS:

Note: *The interface includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the screens (including their use of graphics and language).*

16. The interface of this system is pleasant.

     STRONGLY   **STRONGLY**
     DISAGREE   **1**   **2**   **3**   **4**   **5**   **6**   **7**   **AGREE**

COMMENTS:

17. I like using the interface of this system.

     STRONGLY   **STRONGLY**
     DISAGREE   **1**   **2**   **3**   **4**   **5**   **6**   **7**   **AGREE**

COMMENTS:

18. This system has all the functions and capabilities I expect it to have.

     STRONGLY   **STRONGLY**
     DISAGREE   **1**   **2**   **3**   **4**   **5**   **6**   **7**   **AGREE**

COMMENTS:

19. Overall, I am satisfied with this system.

     STRONGLY   **STRONGLY**
     DISAGREE   **1**   **2**   **3**   **4**   **5**   **6**   **7**   **AGREE**

COMMENTS:

CSUQ results:

Items	Score						
	1	2	3	4	5	6	7
1	0	0	0	0	0	13	29
2	0	0	0	0	0	0	42
3	0	0	0	0	0	0	42
4	0	0	0	0	0	0	42
5	0	0	0	0	3	8	31
6	0	0	7	3	14	15	3
7	0	0	0	0	0	0	42
8	0	0	0	0	38	0	4
9	0	0	0	0	0	0	42
10	0	0	0	0	0	4	38
11	0	0	0	0	0	9	33
12	0	0	0	0	13	3	26
13	0	0	0	0	0	0	42
14	0	0	0	0	0	12	30
15	2	1	1	6	27	5	0
16	0	2	4	1	1	33	1
17	0	0	0	0	25	9	8
18	0	0	0	0	0	40	2
19	0	0	0	1	10	21	10
Total	2	3	12	11	131	172	467

Most relevant comments:

Item 4

Comment: The user interface is not the best, but the system is very fast and I think it is very important!

Item 15

Comment: The user interface has a nice design and the information organization is clear, but the continuous movement of the footer is very annoying for me. For that reason I'm strongly disagree in this point.

Item 16

Comment: The information organization is good, but if I access to the application with a computer screen little than 15'' the controls go out of position and give a bad impression. I don't like it.