Administration Interface in an e-Testing Software Platform

Diana BUTUCEA
Academy of Economic Studies, Bucharest, Romania
dianabutucea@gmail.com

Administration interfaces represent a subject that requires an in-depth analysis. Due to the complex tasks it needs to execute, the developer of such interfaces must find a perfect balance between aesthetics, functionality, ergonomics and modularity of it. While web users become more and more selective into choosing the websites they browse, the interface of any application must be impeccable. Since e-testing software platforms require a complex content management system, the paper will provide arguments over functionality, aesthetics, ergonomics and modularity of such platforms. An in-depth comparison over four CMSs will be made and a model of implementation will also be presented focusing on the administration interface and the functionalities it offers.
Keywords: *administration interface, user interface, database, software system, e-testing*

1 Introduction

Over the last centuries innovative efforts have produced impressive technological achievements: sophisticated medical cures, agricultural methods, new modes of transport, communication media, information technologies etc. These achievements keep fostering the optimism for prosperity, higher standards of living or, in a broader sense, better conditions of life. The cradle of the optimism goes back to the Enlightenment, an intellectual movement in the seventeenth and eighteenth century that strongly influenced the portrayal of mankind. It is the era of great scientists, philosophers and writers, like Descartes, Newton, Leibnitz, Locke, Kant, Voltaire and Diderot. They claim that man is rational and good by nature. Also Darwin should be mentioned, whose theory of evolution reflected the conflict between science and religion, while it rejected the idea of creation of life according to the Bible book of Genesis. Rather than the creationist belief that every species was created individually by God and is not subject to change or progress, Darwin claimed that life has developed in a progressive way from primitive forms to complex organisms.

The Enlightenment marked the liberation from the medieval doctrines of magic, superstition, prejudices and the fear of God by replacing it with human rationality. The fear of God made way for a scientific description and explanation of the world. Beliefs weren't anymore accepted on the authority of priests, sacred texts, or tradition, but only on the basis of reason. Reinforced by the idea of natural regularity and material cause the Scientific Revolution successfully proclaimed the ideology of upward development, progress and improvement of the world, encouraged by an ever-increasing knowledge, understanding and control of nature's processes. It asserts that the individual as well as humanity as a whole can progress to perfection. Being tightly linked with the starting points of modern society, innovation is a necessary condition for all economic functioning. Innovations further the creation of new products, services and production processes, which will give an economic actor an advantage over its competitors. The predominant motto is "innovate or pine away" and the concepts of growth, progress, innovation and change seem to have become

self-evident within the context of our societal system [9].

In recent years the pace at which new technologies become available has increased rapidly. Internet, tablets, e-readers and smart phones are fast-growing markets. So far direct technology push on the educational system remained largely without effect. Educational institutions, positioned as public utilities rather than competitive business, lacked the stimulus for displaying innovative power and could easily ignore new technological trends and resist radical changes. Today, the conditions have changed: the pressure on education is high, new technologies are flooding the markets, and learners, be it schoolchildren or adult professionals, grow up immersed in new digital communication technologies and will demand high quality, flexible, modern and tailored learning services. [10]

Within the technology framework illustrated in Section 2, the paper wishes to provide a model of implementation for an administration module used in an e-testing software system. The proposed architecture supports fully interactive operation of the administrator over the server through the user interface and is open towards the integration of more advanced adminstration functionalities. Section 3 will provide the functional description of the application's components and design features whereas Section 4 concludes the paper.

2. User interfaces features

2.1. Design and aesthetics

Wiktionary [11] defines **aesthetics** (*uncountable*) as the study or philosophy of beauty. Beauty in World Wide Web stands for a creative, original, functional and usable layout. An interface that would generate admiring comments, lots of traffic, link exchange and link passing (a term proposed for the transmission of the link between Internet users through different communication channels, e.g. Yahoo! Messenger or Facebook). Aesthetics inside a web application is mainly due to the web designer's experience and creativity. He is the person who has the vision to *draw* a layout following a list of specifications but also putting his fingerprint on the project (his vision over the website theme and purpose). A good layout would guarantee at least one thing to a web application: that people would not instantly click the browser's *Close* button when they start browsing it; instead they would be tempted to continue using it. [5]

Regarding the user interface matter, the distributed information approach to display design identified three forms of interface communication: information retrieval, comparison, and interaction. This theory is seeded in traditional cognitive theory stemming from connectionist networks and mapping, but the resources model acknowledged that task success is partially display dependent. According to distributed information resources research, display design impacted information-processing tasks. Various studies have examined each type of interface communication and the role of display design concluded that attention to the technology components is essential for effective positive results. [1]

After checking and optimizing the design part, it becomes the developer's job to create the functionality as appealing as possible. Therefore, different technologies (both client-side and server-side) are used in order to achieve spectacular visual effects like multimedia files embedded into web pages, special features modal windows, hover, floating or fading effects, small details that make an interface user-friendly, functional and browse-appealing.

2.2. Ergonomics

Wiktionary [11] defines **ergonomics** (*plural* **ergonomics**) as the science of the

design of equipment, especially so as to reduce operator fatigue, discomfort and injury. Starting form this definition and extrapolating, on the web, ergonomics stands for ease of use and a page's property to give the user access to any information, link, button or functional need, in order to create a non-obstructive browsing experience. Therefore, this aspect is very important and must seriously be taken into consideration when starting to develop a complex administration interface.

2.3. Modularity

Modularity (uncountable) – the property of being modular [11] states, in terms of user interfaces, a CMS's ability to easily create custom or new modules. Modern software platforms need to meet this standard and even unexperienced users have to be able to create such modules.

A comparison over all of these features and not only has been made further in the paper.

3. Administration interfaces

3.1. Overview

Talking about an e-Testing software platform requires a good, in-depth analysis. Since developing such a system can become quite challenging in both overall and step-by-step development process, a solid ground in matters of documentation and system requirements must exist even before the project begins. In this phase, a developer must find the answer to several questions which include: *What is the main functionality of the platform?*, *Who is going to operate the platform?, What are the expected results?* or *Which are the system's limitations?* These questions have to find full answers in the analysis phase of development in order to go on with the project. It is the software engineer's job to define its platform guidelines, to choose its models of implementation, to coordinate the development workflow and to assure smooth and in-time project hand-over.

In the development process of an e-testing software platform, the answers to the first set of questions were extremely clear – the developer must build up a software system that would generate real-time tests and deliver instant results to the examined students, therefore, a software platform that would complete these tasks is compulsory. The decision was to split the work into three separate modules – the user interface, the administration interface and the database system (as will be shown below).

3.2. Open Source CMSs vs. Custom made CMSs

An administration interface for a software platform has to meet a few criteria: good aesthetics (as shown in Section 2), ease of use (ergonomics), fully customisable modules, clean and fully customisable core code, non-restrictive administration interface and page templates. Considering these rather few criteria, a decision must be taken by the software developer over which administration interface to use for its platform. Therefore, a study over four major CMSs (Content Management Systems) has been made in order to decide which administration interface suits the platform best. Since the research wanted to point out the advantages and disadvantages of using these open-source tools, the paper will present the author's conclusions after using each of these interfaces and, in parallel, developing a custom one.

a. The Joomla! CMS

The Joomla! project is one of the most rapidly growing open-source Content Management Systems in the market. It has a large support community and lots of developers base their projects on it. Even though Joomla! has a good set of modules

that can be used by unskilled developers, it becomes extremely difficult for anyone to create a custom module and place it in its project. The five criteria defined before are not satisfied, since the administration interface is non-intuitive, the modules are un-customisable and the code is almost impossible to change. Considering these aspects, as well as Table 1, the decision was not to use Joomla! as an administration interface.

b. The Mambo CMS

Since Mambo is the predecessor of Joomla!, it has almost the same advantages and disadvantages for the end-user. The administration interface is good enough to develop template-based projects, it includes lots of modules, but does not allow too much customisation, neither on the administration interface and on the user interface. The CMS has proven itself not suitable for the e-testing software platform.

c. The Drupal CMS

The Drupal CMS is mostly preferred by enterprise corporations due to its scalability and power to process and store large amounts of data and files. Drupal has an easy to use administration interface, easy user management, lots of plugins available free and built-in modules, but, again, customising a module has proven to be a difficult task, therefore, even though this CMS gained the best score (as shown in Table 1), it still didn't meet the requirements for the development of the e-testing software platform.

d. The WordPress CMS

Considering the aspects that have been taken into considerations with all of the previous CMSs, WordPress, as the last open source platform to be tested proved itself extremely user-friendly in terms of interface customisation but really restrictive in terms of code editing and also in terms of security (the upgrade cycle is too fast, therefore updating the versions is compulsory).

After testing these four administration interfaces, the results were as follows:

Obs. The metrics defined for the research has been chosen as follows:

- if the aesthetics of the administration interface is good, the CMS received one point; else, it received zero points;
- if the CMS has been considered ergonomic and intuitive, it received one point; else – zero;
- if the CMS had customisable modules – one point; else zero;
- if the CMS had customisable core code – one point; else zero;
- if the CMS had a non - restrictive administration interface – one point; else zero.

Table 1. CMSs' scores

Criteria	Joomla!	Mambo	Drupal	WordPress
Aesthetics	1	0	1	0
Ergonomics	1	1	1	1
Customisable modules	0	0	0	0
Customisable core code	0	0	0	0
Non-restrictive administration	0	0	1	1

interface				
Total	**2**	**1**	**3**	**2**

After analyzing the scores and the main features these types of administration interfaces can offer, the decision was taken over implementing a custom-made administration interface rather than use an existing one, since none of them has proven itself reliable for the software platform that needed to be developed. Even though the amount of work is considerably larger, a skilled developer can easily create its own administration interface, build his own modules and be able to obtain good results on both client-side and administrator interface.

4. Model of implementation

4.1. Web technologies integrated in the administration module

Considering the wide variety of web technologies and because each technology has its strong and weak parts, it remains to the developer's choice and work experience which one to use in its software platform. Depending on the tasks the application needs to execute and on the developer's work experience, a software system can be build using different web-based technologies.

Server-side technologies refer to scripting where a client generates a request to a server, the server analyzes this request and generates a response back to the client. The process is executed on the server and the response returns as HTML code to the client (the web browser). These technologies are mostly preferred by developers due to ease of integration within mark-up language and because of their capability to generate dynamic HTMLs. [2]

Since its release, in 1994, PHP (the acronym for Personal Home Page) became very popular due to its ease of nesting into HTML code, but started to fully expand since its fourth release, when Zeev Suraski and Andi Gutmans rewritten the PHP engine under the acronym Zend.[8]

PHP can be regarded as a mature technology in the sense that it has a large user base, is widely supported and has many advanced features.

Opposed to server-side technologies, **client-side technologies** are the ones that generate the events directly on the client application (usually, the web browser), in order to cut out the time needed for the server's interrogation and response delay and, most important, to create visual and functional effects similar to desktop applications, for an excellent browsing experience to the user. The resemblance to desktop applications is quality in ease and use while the difference comes from the mobility and accessibility [3]. In this context, RIAs (Rich Internet Applications) have emerged in the last few years, providing developers a large set of tools to use.

AJAX (Asynchronous JavaScript and XML) allows every element within a web interface to be individually and quickly updated without affecting the rest of the interface. This, of course, is not what most web users are accustomed to. Initiating an action within most web sites triggers the inevitable blank screen and page loading process. Though not very responsive, the full-page update makes it very clear to users that their action has resulted in a reaction and that a response will be available as soon as the page is refreshed. Because AJAX - based updates are very fast and incremental (often affecting only a small portion of the user interface), users may not notice them, especially when they are used to seeing full-page rewrites, but they make the web pages work more efficiently, increasing the browsing speed.

In matters of markup languages, the simplicity of HTML has captivated many

skillful programmers and creative web developers, enabling them to create all sorts of websites. Over the years, HTML has withstood many challenges and evolved with new features as the web technology advances at an astonishing pace. Today, HTML is still the predominant technology for website designs. These websites provide mostly browseable-only contents with form interaction using Common Gateway Interface (CGI).

Dynamic web content generation and interaction are achieved by using a database system as the backend data storage and retrieval system in conjunction with a client (web browser) and a web server. The majority of these types of Internet applications are most used by the e-commerce and entertainment sectors. However, most of these more sophisticated interactive applications still provide only simple interactions such as product navigation, shopping cart storage and form processing. More complex interaction is achieved by using Javascripts embedded in HTML or plug-ins. [6]

Two-tiers approaches to client / server architectures followed the file server approach. In this system, the client workstation is responsible for managing the user interface, including presentation logic, data processing logic and business rules logic and the database server is responsible for database storage, access and processing. Therefore, user authorisation, integrity checking, data dictionary maintenance, query and update processing are all performed at one location, on the database server. [7]

4.2. Custom-made administration interface

Since none of the four open source administration interfaces did not prove themselves efficient or customisable enough to meet the demanded requirements (as shown in Section 3), an administration module for an e-testing software system has been developed. The solution has been implemented using PHP as server-side technology and AJAX as client-side scripting (technologies which are similar to the ones the four CMSs that have been tested use) and is composed of three major modules: the student module, the administration module and the database structure. What I needed was a tool that would dynamically generate tests, randomize questions and / or answers, generate instant results and have the capability to save and manipulate the tests' results. Therefore, the platform I have developed is fully customizable (the modules can be easily upgraded, if necessary), has an easy module implementation system and is able to manage large amounts of data and files. The implemented algorithms have proven themselves efficient and the database structure is stable and hasn't encountered errors of any kind.

The users that would interact with the application were divided in three separate categories: the master administrator, the administrator and the students.

- the master administrator is a special type of user and has special administrative roles, mostly into assigning roles and privileges to the application's users.
- after logging into the application, the administrator has many different management options, including disciplines, chapters, questions and answers, sections that define the content of the online tests; besides these sections, tests and exams management sections are different and manage the creation, distribution and manipulation of the online tests.
- students are only allowed to take the exams

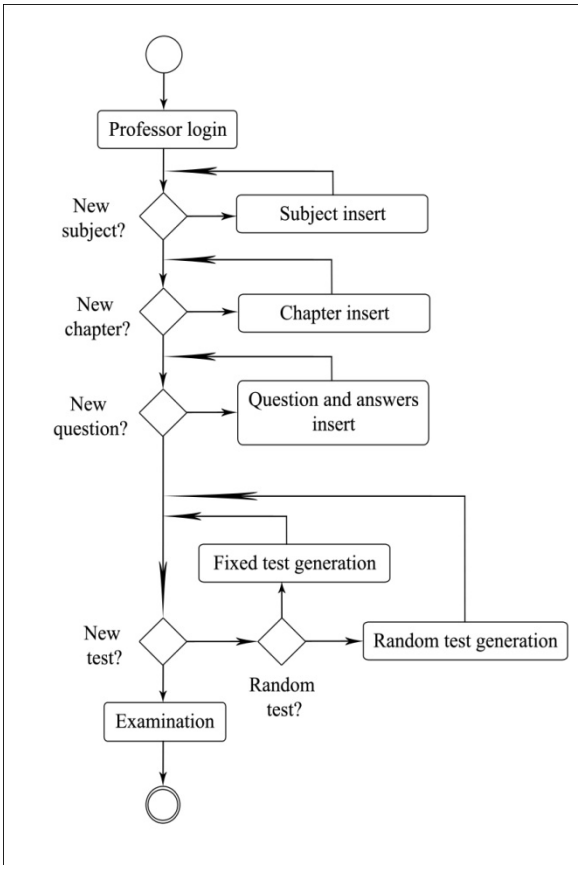

Figure 1. Activities diagram

Figure 1 shows the activities diagram and presents the workflow and the conditional steps an administrator does in order to deliver the tests to his students. As previously mentioned, after logging into the application, the administrator has to insert a new subject, link chapters to this new subject and then, in a cascade mode, insert his questions. After completing these steps, the administrator must create his test, add questions to it (if it is a fixed test) or just name it, if he wants to create a random-generated new test. All these options are available through the administration module presented in Figure 2.

After creating the tests, inside the Exams Management section, the administrator authorizes the students to take the exam, after they have previously registered through an online form. The proposed authorization algorithm counts the students that have registered, validates the list of students and, with the professor's permission, triggers the exam.

Figure 2. Administration module interface

If a student is not authorized by the professor, he would immediately be removed from the waiting list. A log file is automatically generated, allowing the teacher to have full control over his admission lists and his taken exams. Figure 2 presents the main interface of the administration module. Note that, if the test is a random-generated one, each student will have different sets of questions, receiving their mark separately, according to their test.

Since this interface is a custom-built one and since every implemented module is known by me, I can clearly state that building this administration platform from scratch has been the best solution for my type of software system. Also, because classic CMSs are mostly targeted to build

commercial websites and to be used by unskilled users, the choice of building a custom administration interface has become compulsory in this situation.

4.3. Future work

Being aware that the presented subject has not yet reached its full potential and that there still are a lot of aspects to cover, the research will to continue by studying the fields of semantic web and ontologies. Since a standardization method has not yet been defined in this research area, I hope that my future studies and research would become useful materials for other researchers. Meanwhile, I wish to improve and develop the functionalities of the software model, and, in parallel with semantic web, to fulfil some objectives like implementation of an e-testing platform that would be used by blind people, implementation of a single sign on mechanism and gradual transition to web 3.0. The main interest would also imply improving the user interface, adding new question types, new test types and proposing new implementation algorithms to the present software system.

5. Conclusions

An administration interface represents the communication channel of a user with any application. In the development process of an interface, the developer must consider at least two aspects: the aesthetic of the interface and the functionality of it.

Recent studies have revealed a high correlation between how an application looks and the user's desire to continue browsing it [5], therefore the way an administration interface is designed could easily define a user's will of using it.

In addition, the term administration interface strongly relates to the database it uses. Since all data is gathered from the database system, the information must be unaltered and ready to be delivered to the user when he needs it. A powerful content management system is the software platform that does this delivery whenever necessary. Even though, overall, all open source CMSs are working on the same technologies and use the same DBMS, major differences appear in terms of presentation and modularisation. Data flows are also different and modifying any of them proves to be a difficult task even for skilled developers.

Linking a database to an administration interface is usually done through configuration files, two examples of such files being listed below:

```
<?php
      //php database configuration
file:
      $conn =
mysql_connect('host_name','user_name
','password') or die('db connection
failed');
      $db =
mysql_select_db('db_name');
?>

<!-- asp.net connection string -->
<connectionStrings>
      <add
      name="name_of_db_connection"
      connectionString="Data
      Source=data_source_name;Initia
      l Catalog=db_name;Persist
      Security Info=True;User
      ID=user_id;Password=user_passw
      ord"
      providerName="System.Data.SqlC
      lient"/>
</connectionStrings>
```

The paper presents the implementation model of an e-testing administration interface, analyzing some of its features, functional mechanisms and implementation algorithms. E-testing software systems represent a special kind of web applications. Because of their functional purpose, they need to be robust

and reliable. These two major characteristics are implemented through the web technologies used in the development process. In chapter 4.1, a set of arguments to support the decisions regarding the technologies used in this process have been presented. Nevertheless, although the market offers a wide variety of web technologies, in the end it comes to the developer's decision and work experience which one to use for its software platform [4].

While server-side technologies represent the basis of the software platform, client-side technologies manage to bring in front the sparks that make the difference between a static, dynamic, aesthetic or functional web application.

The paper also presents a comparative study made by the author over some of the most commonly used Content Management Systems. The study revealed a rather intuitive result (which has now been confirmed, through its research, by the author as well) – every CMS, no matter its name, producer or level of popularity, is a restrictive tool when it comes to building a custom-made web application. Even though four of these types of software platforms have been tested, in the end the author's custom-made administration interface has proven itself the best solution into building the desired e-testing software system.

In conclusion, a good balance between aesthetics, functionality, ergonomics and modularity of the administration interface, a solid structured database system, good security and reliable software development will create the premises for any administration module or interface to become widely used and appreciated.

5. Acknowledgments

This work was co financed from the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards".

References

[1] Austin, K., „Multimedia learning: Cognitive individual differences and display design techniques predict transfer learning with multimedia learning modules", Computers & Education 53, 2009, pp. 1339–1354

[2] Butucea, D., Cervinschi, C., "Development of e-learning software systems using web technologies", The Tenth International Conference on Informatics in Economy IE, 2011

[3] Castellanos-Nieves, D., Fernandez-Breis, J. T, Valencia-Garcia, R., Martinez-Bejar, R., Iniesta-Moreno, M., "Semantic Web Technologies for supporting learning assessment", Information Sciences, No. 181, 2011, pp. 1517 – 1537

[4] Cervinschi, C., Butucea, D., "Integration of web technologies in software applications. Is Web 2.0 a solution?", Database Systems Journal, 1, 2010, pp. 39-45

[5] Cervinschi, C., Butucea, D., "Impact of web technologies in functionality and aesthetics of web applications", 18th International Economic Conference – IECS, 2011

[6] Chang, G., Hsieh, J., Calixto, P., „Components for Building Desktop-Application-Like Interface in Web Applications", APWeb, LNCS 3399, 2005, pp. 960–971

[7] Hoffer, A., Prescott, B., Topi, H., "Modern Database Management", Pearson Prentice Hall, 2009, pp. 406-407

[8] Săulescu, O., "Creați uşor un Website dinamic cu PHP şi MySQL", Chip, Nr.2, 2003
[9] Westera, W., „Beyond functionality and technocracy: creating human involvement with educational technology.", Educational Technology and Society, 8, 2005, pp. 28–37
[10] Westera, W., „The eventful genesis of educational media", Educational Technology and Society, Published online: 21 May 2011
[11] http://en.wiktionary.org

Diana BUTUCEA graduated in 2008 from the Economic International Relations Department of the Academy of Economic Studies in Bucharest, in 2009 from the Faculty of Automatics and Applied Informatics of the Transilvania University of Brasov and in 2010 from the Economic Informatics masters at the Academy of Economic Studies in Bucharest. Her parallel interests, in economy and software engineering, are now merging into her studies and researches since she is PhD student at the Academy of Economic Studies, Bucharest, studying integrated software systems, web technologies and e-learning platforms.