TESTING ASSISTANT – AN INTERACTIVE TRAINING TOOL FOR EVALUATING STUDENTS KNOWLEDGE

Adrian Florea[1], Traian Anghel[1], Delilah Florea[2]

[1] *"Lucian Blaga" University of Sibiu, Computer Science Department, Emil Cioran Street, No. 4, 550025 Sibiu, Romania,*
[2] *"Samuel von Brukenthal" National College of Sibiu, Informatics and Computer Science Department, Huet Square, No. 5, 550182 Sibiu, Romania,*

adrian.florea@ulbsibiu.ro, traian_gk@yahoo.com, delilah_florea@yahoo.com

ABSTRACT

Since many times the quotes about professors "who talks in someone else's sleep" are true and because the research studies have shown that the concentration degree decreases after every 10-15 minutes, applying changes into the teaching / learning / testing process became imperative. This should be realized through an attractive and interactive manner based on modern techniques and tools. Using simulators, different software tools, multimedia technology and formative interaction in teaching act became a real necessity. Testing Assistant represents an interactive training tool for evaluating student's knowledge and consists in a first step toward implementation of a modern multimedia learning and testing application devoted to teachers and students from Computer Science, Information Technology and Multimedia Systems specializations from Computer Science Department, University of Sibiu. Testing Assistant should combine the five fundamental components of an e-Learning project: Digital Curriculum, Professional Development, Connectivity, Technology and Methods Development for learning and testing.

Besides that is free and easy to use, the Testing Assistant tool provides the following advantages: flexibility, portability, extensibility, interactivity and performance. In the academic world the professor is time-pressed due to many activities besides teaching: research grants proposals, reviewing process, conferences, writing keynotes and articles, leading PhD students, etc. In this case, Testing Assistant tool could be a solution because it reduces the time required for correcting exams. Additionally, it permits both for students and for certain professors, generally self taught persons, to evaluate themselves and even to learn in advanced various improvement courses, providing the feedback regarding to their actual performance potential.

Keywords: testing tool, evaluating, teaching process, exam, database

INTRODUCTION

The fast development of computer science and information technology domains reflects itself primarily in higher education. The teaching / learning / testing process should be realized through an attractive and interactive manner based on modern techniques and tools (Excel / PowerPoint / Internet / Flash / Php /

Microsoft .NET Framework / Sql); otherwise we risk becoming like in Auden's quote "A professor is one who talks in someone else's sleep" according to (Dictionaries of Quotes). The educational content must be ported on digital assets realized with products such as Word and text documents, PowerPoint slides, Instructional Authoring docs (Authorware, ToolBook, etc.), Flash animation, HTML docs, illustrations and graphics, audio and video clips, quiz questions, etc. Using simulators, different software tools, multimedia technology and formative interaction in the act of teaching became a real necessity as (Grindei, Orza & Vlaicu, 2008; Moreland, Jones, Chambers, 2001) pointed out. For example, in microprocessors systems domain, it is difficult to explain concepts like caches, out-of-order and speculative execution, power consumption, and the interactions among the architecture components without visual aids. Graphical simulations of these architectures allow students to easily grasp the concepts of the architectures by observing the flow of instructions in time, also exploring the impact of different processors configuration on performance, dissipated energy and temperature.

Nowadays the teaching / learning process has become "online" or "virtual". The introduction of the aforementioned modern software instruments and techniques in the act of teaching involves changing the professor's roles and duties (Ilie, Jugureanu, Pacurari, Dragomirescu, Vladoiu, 2008). The cognitive role, which relates to mental processes of learning, information storage, and thinking, shifts to one of deeper cognitive complexity. The affective role, which relates to influencing the relationships between students, the instructor, and the classroom atmosphere, required faculty to find new tools to express emotion yet they found the relationship with students more intimate. The managerial role, which deals with class and course management, requires greater attention to detail, a better lesson / testing structure, and additional student monitoring according to (Guerra, Fabbri, Maeran, 2008). The professor should document for preparing new lessons / chapters making the bridge with prerequisite courses and announcing the future topics. The information must be exposed through an attractive and interactive manner by "bringing the student to play". Also, the professor must elaborate partial or final assessments and obviously, must keep contact through Internet with professors and researchers from all over the world that teach the same curricula, to be up-to-date with the most recent information from teaching / research domain. This involves from the teacher's side certain skills and competences for proper usage of information technology, for the right selection of multimedia materials and electronic lessons according to learning objectives including capabilities of selecting relevant information for educational purposes.

In this work we present the Testing Assistant (higher education tool assistant), an interactive graphical training tool for evaluating students knowledge. The Testing Assistant software tool is currently used in undergraduate and graduate courses / laboratories in Computer Science, Information Technology and Multimedia Systems specializations from "Lucian Blaga" University of Sibiu, Romania. The main topics approached through Testing Assistant application are Computer Organization and Design, Microprocessing Systems, Object Oriented Programming, Algorithms and Programming Languages due to specializations of the articles authors. The *Testing Assistant* application can be launched by typing in any internet browser (Internet Explorer, Mozilla Firefox, Opera, Google Chrome, etc) the address http://testingassistant.gotdns.com:800. We developed Testing Assistant application using Academic Resource Kit provided free of charge by Microsoft and that contain Express Edition of SQL Server, Visual Studio, .NET Framework, Shared Source Common Language Interface, Windows PowerShell and Virtual Server.

However, evaluating student's knowledge based on partial assessment and quizzes represented the starting point of Testing Assistant application, as of its current state, adaptations and improvements are constantly being made to make it better, more flexible and more easy to use. In its final stages of development, it could be considered as a real and feasible learning content management system.

International e-Learning platforms such as IBM Education Assistant (http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp), Oracle (https://academy.oracle.com/) and Cisco (http://www.cisco.com/web/learning/netacad/index.html) Academies, besides teaching modules include an evaluation component based on quizzes, partial and final exams, and practice exam in Flash. Also, for a fair student evaluation this should be realized in many stages, not just a final one. Besides homework during the semester it is required to undertake quizzes and partial assessments from studied chapters. Because assessment leads to training, students learn more and become more confident, self-directed learners. Additionally, with Testing Assistant both students and the academic staff, generally self taught persons, will have the possibility of a continuous improvement, learning in advanced various courses, taking advantage of the feedback regarding to the performance potential reached (what chapter should be study thoroughly). Finally, from an economically point of view, using Testing Assistant, it can save a lot of paper (less chopped down trees) with positive impact on life and the surrounding environment. On the other side, although the important names from computer science, Intel (http://www.intel.com/education/highered/modelcurriculum.htm), Hewlett-Packard (http://www.hp.com/education/) beside the formerly mentioned – IBM, Oracle, Cisco, provide different courses (operation system, microprocessing, parallel programming, business analysis, project management, database design and programming, etc), each of these courses cost starting from 500 Dollars to 2000 Dollars. Therefore, Testing Assistant tool is dedicated for students and professors from academia especially from "Lucian Blaga" University of Sibiu.

The organization of the rest of this paper is as follows. In section 2 we review the software design of *Testing Assistant*, the information database of learning / testing content management system, namely the tables implied and relationships among them. Section 3 describes the user interface and system requirements. Finally, section 4 suggests directions for future works and concludes the paper.

SOFTWARE DESIGN OF *TESTING ASSISTANT*

The learning content management system (LCMS) represents a new application segment in e-Learning domain. LCMS constitutes a multi-user environment dedicated for creating, storing, reusing, assembling and distributing the digital information with educational character from a central object repository.
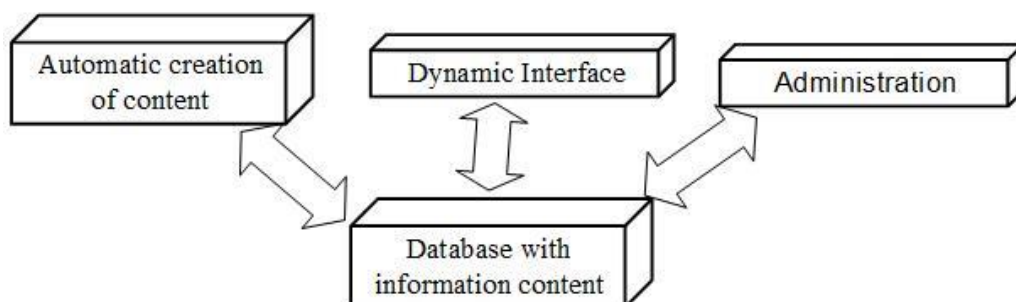

Figure 1. The architecture of a LCMS

As it can be seen in the Figure 1, there are 4 main components of a LCMS:
- **Database with information content** functions as a centralized repository. It has the role of storing and managing the information content. This content is available

on the World Wide Web, like in the case of Testing Assistant and is detailed below in Figure 2. Other LCMS offer the database on CD.

- **Administration** provides tools to manage the authorization of content, users and user groups, the organization of content – curriculum, courses, modules, learning objects, and units of digital learning content/assets. It facilitates the management of every educational activity, being used for launching available subjects / topics, introduction of professors, students or specializations, monitoring and reporting the progress made by students and any other administration functionalities.
- **Dynamic delivery interface** realizes dynamic distribution of information in web format making content visible for users. Also helps to monitor the student's activities.
- **Automatic creation of content** allows users, such as administrators, authors, professors and subject matter experts, to create new instructional content, or to upload existing third-party instructional content.

As it can be observed in the figure below (Figure 2) the database with information content of learning content management system includes 17 tables. We implemented a relational database management system that meets non-redundancy, integrity, consistency, atomicity restrictions. The arrows illustrate one-to-many relationships between linked tables. For example, each *Specialization* contains many *Subjects* (topics).

Each *Specialization* from Computer Science Department contains many *Subject*s (topics of study). *Teaching* table contains a Professor ID and a Subject ID so that it can uniquely identify a many-to-many relationship between *Professor* and *Subject* (each professor can have one or more subjects and vice versa).

A *Questionnaire* is one of the most important tables in this database. It specifies a set of questions that will be used later in the program to generate exams. It can be used only by the professor who generates it, encapsulating and protecting his work. Questionnaires can contain an unlimited number of *Chapters*, each chapter unlimited *Questions*, and each question unlimited *Answers* but at least one correct. The professor can add a description to a *Questionnaire* if he is adding several questionnaires for a subject, so that, later on, he can easily recognize which one to choose (this could be the case of a professor generating separate questionnaires for each preliminary examination and one for a final exam).
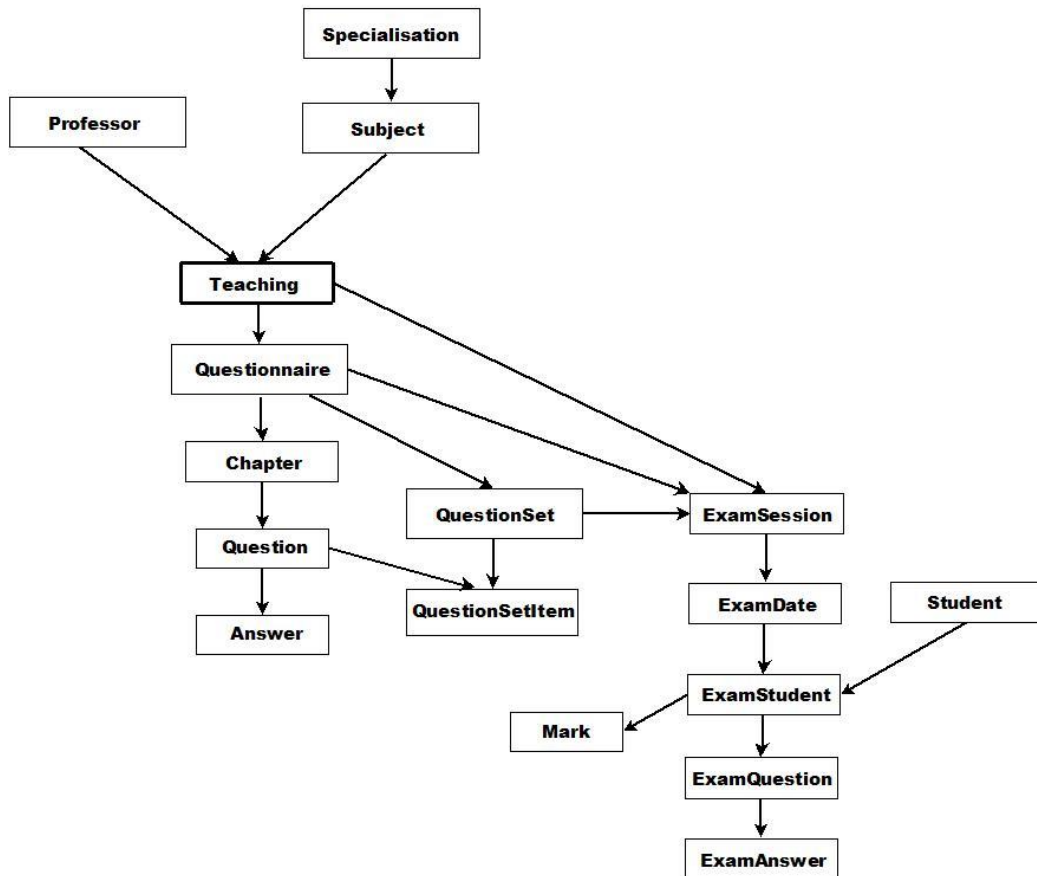
Figure 2. The database content of LCMS

A *QuestionSet* is basically a subset of questions from a Questionnaire. The professor selects specific questions that will be used later on in an exam. This is an alternative to generating a random subset of questions, as we will explain later.

ExamSession is the central core of an exam. It contains various data about the exam, like:

- Duration: length of the exam (minutes)
- Contribution: relevant for the final mark calculation (%)
- QuestionGenerator:
  - Random: a fixed number of questions will be shown to the student (e.g. out of 70 questions that a Questionnaire has, the system picks randomly 20).
  - Manual: the user gets to choose a QuestionSet subordinated to a previously chosen Questionnaire.
- QuestionAmount: visible only if QuestionGenerator "Random" is chosen. Represents the amount of questions the system should randomly pick.
- QuestionSetID: visible only if QuestionGenerator "Manual" is chosen.

Also, the professor has the freedom to set up some *ExamDates* for the upcoming exams. For each ExamDate chosen, a number of *ExamStudents* can be assigned. These are the students that will be evaluated later (they can start the exam).

An ExamStudent is what it takes for a *Student* to attend an exam. It contains the questions, the answer possibilities and what he has selected. There is possibility to select all students from a specialization to be evaluated. The *ExamAnswers* will be evaluated when the student presses the button *<Finish Exam>* and a *Mark* will be calculated, alongside with some raw statistics. A crucial feature here is that the mark can be changed by the professor who generated the exam, therefore allowing

increasing or decreasing marks for various reasons. Also, the student has the opportunity, in case of interrupting the exam due to unexpectedly events, to restore and continue the exam from the place it was interrupt.

THE SYSTEM REQUIREMENTS AND USER INTERFACE

For a proper use of Testing Assistant tool, some system requirements should be accomplished both from server and from client side. Thus, it is recommended to have a processor with at least 1 GHz frequency. The RAM memory recommended is 256Mbytes. To have a good view, it is required a 1024x768 minimum screen resolution. From *server* point of view there are following system requirements:
- .NET Framework 3.5. can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs. It provides a large range of features including user interface, data and data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The space required by .NET Framework is 375 Mbytes.
- IIS 5 (Internet Information Services) as web server. It is free available for users of Microsoft Windows 2003 Server or XP Professional. The space required by IIS 5 is 13.5 Mbytes.
- Microsoft SQL Server 2005 – program packages for database management system. It should be installed on web server and uses SQL languages for querying the database. Microsoft SQL Server requires most space of 1070 Mbytes.
- The space required for source code of Testing Assistant is 2.26 Mbytes.

From *client* point of view there are following system requirements:
- Any kind of browser that interprets HTML code (Internet Explorer, Mozilla Firefox, Opera, Google Chrome, etc.).
- For accessing the application the user has to open the browser and complete in address bar the server address and the port such as

  http://testingassistant.gotdns.com:800.

As it can be seen, the system requirements don't exceed the hardware resources provided by nowadays ordinary desktop computer.

After launching the application, the first window that appears is the login page. Its purpose is to authenticate the application users. As it can be seen in the following figures there are three kinds of users:
- o Administrator:
  - Can add / edit / delete super-users with administrator's rights (admin), professors, students, subjects and specialization.
- o Professors:
  - Can add / edit / delete questionnaires with questions and answers, question sets for exams.
  - Can schedules exam sessions: select questionnaire, students, dates, exam type generator.
  - Can view the exam results of students and modify the mark if needed. Also, if some questions require manual evaluation (essay-type questions), the professor must correct the respective answers and award points according to those answers.
- o Students:
  - Can access their assigned exams. After finishing the exam they can view the result and also receive a feedback with their answers (correct or

wrong). The system also indicates the correct answer (for wrong answers).

From the Administrator panel (see Figure 3), this can add a new Professor by simply clicking on *<Professor>* and then on *<New>* button. The insertion of a new article in the database (in this case Professor table) will take place with observance of oneness condition (the Professor ID represents the unique key). Pushing the *<Remove>* button the selected Professor will be deleted but if and only if he hasn't assigned Questionnaires and Exam sessions (integrity condition of relational database management system). The *<Edit>* button allows modifying the Professor's attributes. Choosing the *<View>* button on screen will show the subjects taught by the selected Professor.

Figure 3. Professors panel within Administrator's account
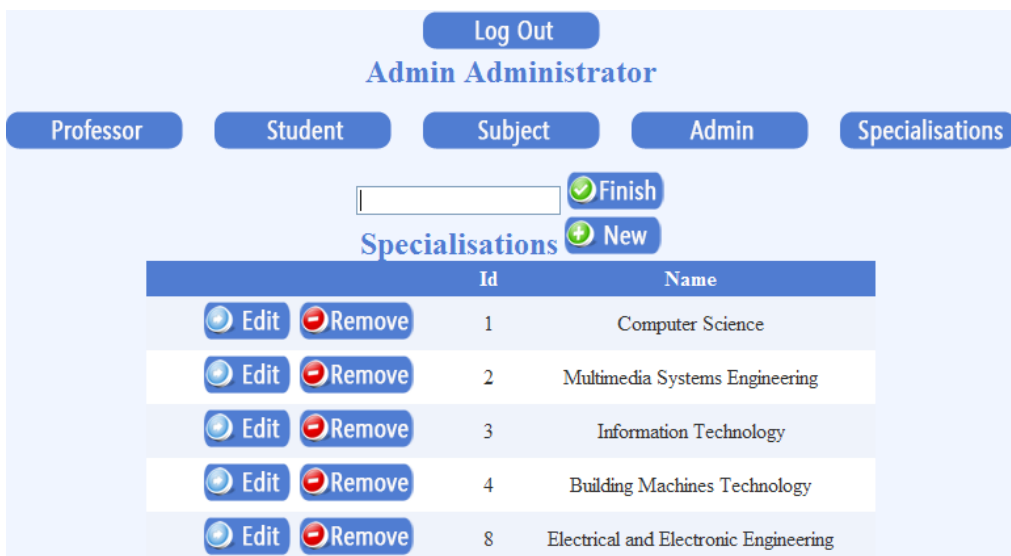
Figure 4. Specialization panel within Administrator's account

From the same panel, the Administrator can select the *<Specialisations>* button, then *<New>* button, specify the name of new specialization and then click on the *<Finish>* button (as in Figure 4).

If the Administrator clicks on *<Student>* button, on screen will appear a picture like that from Figure 5. At request, results can be filtered depending on Specialization/Year/Group/Semi-Group. By pushing the *<New>* button, a new student can be added with full-name, username, email, specialization and group name. Pushing the *<Remove>* button the selected Student will be deleted but if and only if he isn't involved in any Exam sessions (integrity condition of relational database management system). The *<Edit>* button allows modifying the Student's attributes.
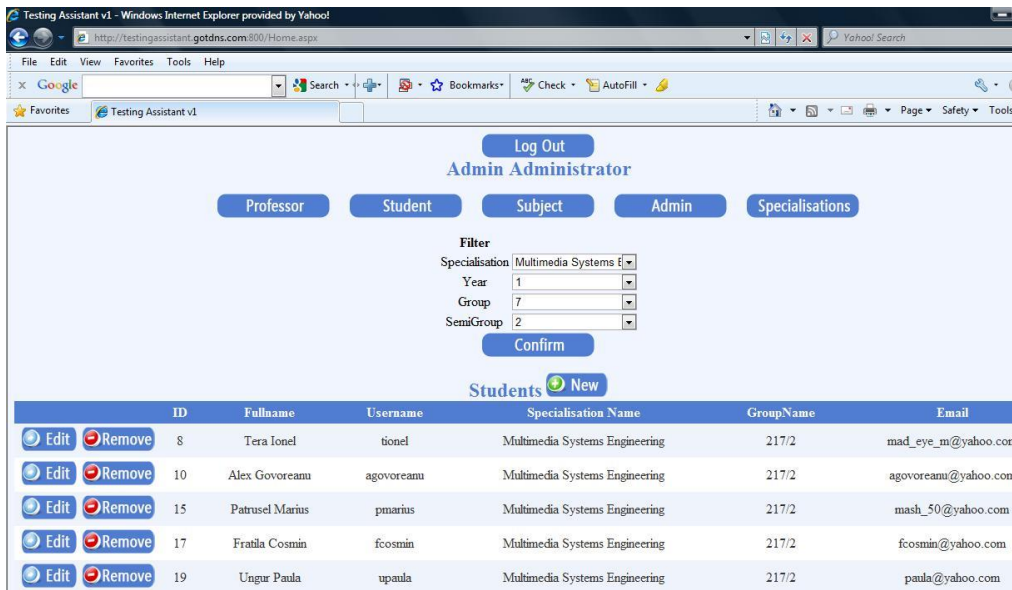
Figure 5. Filtering the Students from a certain Specialization/Year/Group/Semi-Group
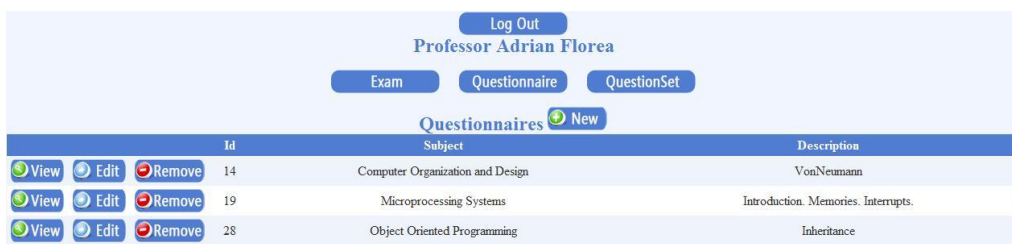
Figure 6. Questionnaire panel within Professor's account

If the Professor adds, modifies or removes a certain number of questions into a new Questionnaire and wants to see how it would appear into Exam session, it is necessary to be checked out in QuestionSet.

Figure 7. Establishing the Exam details by the Professor to a certain Subject from a previously introduced Questionnaire

An important component when the Professor schedules an Exam Session is the *Exam Status* that limits the student's access to the exam depending on each setting (see Figure 7). This *Status* can be modified by clicking on *<Change Status>* button from any row of *ExamDates*, and could be in one of the following states:

➢ *<Assigned>* – the students will have the access blocked to the exam but they can see the date and hour at which they were planned.

➢ *<In Progress>* – the students can access their exams and proceed to succeed.

➢ *<Finalized>* – the students that did not start in time, have the access blocked to the respective exam.

The Figure 8 illustrates changing the exam status within an *ExamSession* started by Professor.
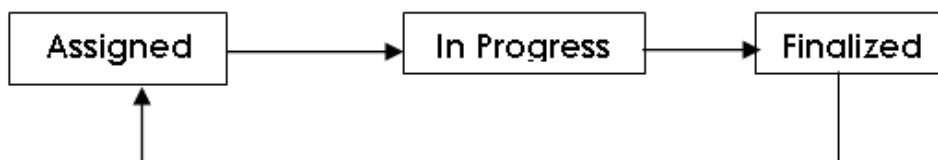
Figure 8. Changing the exam status within an *ExamSession*

In order to view the exam details of students assigned to a certain *<ExamDate>*, the Professor must push the *<View>* button. Thus, on the screen will be displayed a list with students involved in exam session and with their marks after finishing the exam (see Figure 11). On the left of the student ID from the list there is the *<ExamDetails>* button which displays private details for that student (see Figure 12).

The Professor can change the obtained mark within the same page and, by pushing the *<Go to Questions>* button, will see the student's answers.

Figure 9. Screenshot with an Exam: different type of questions

Testing Assistant includes the e-Testing module in which the professor defines a set of questions (questionnaire), eventually subsets for partial assessments, schedules exams and chooses students for that exam. As you partially can see in Figure 9, the types of questions are:

- Multiple choice single answer (MCSA);
- Multiple choice multiple answer (MCMA) with 2 or 3 correct answers;
- Textbox questions in which the student should indicate the answer within a single sentence (or word);
- Essay questions in which the student may introduce his detailed answer (see Figure 10);
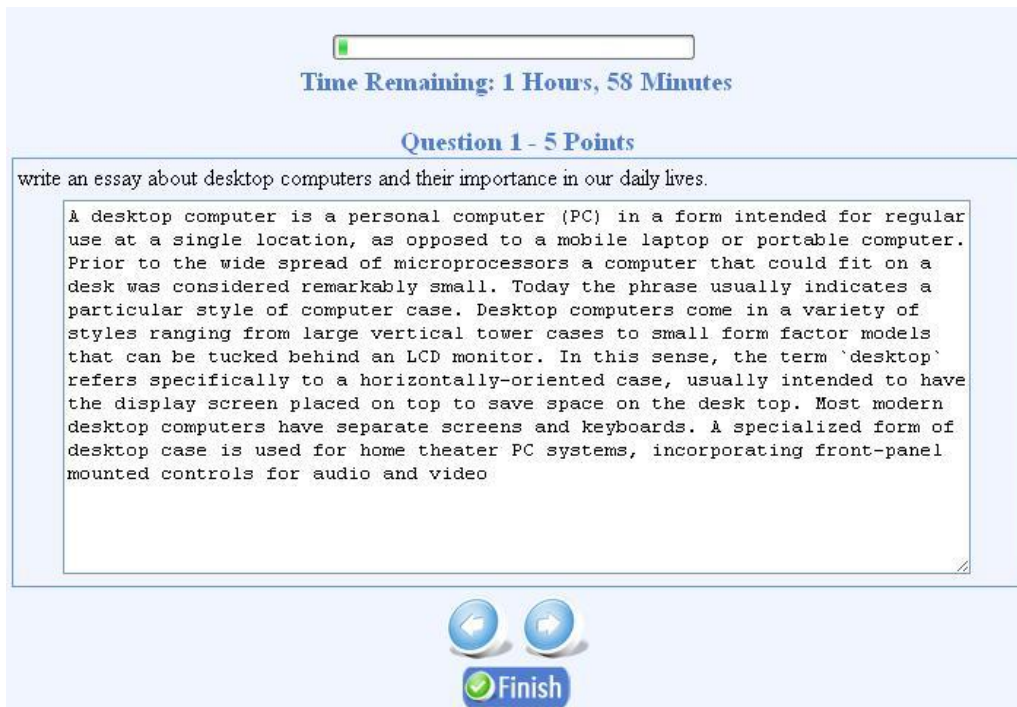- Questions that require settling the stages of answer in a proper order.

Figure 10. Essay Question Type

The multiple choice single answer and multiple choice multiple answer are treated separately (in MCSA if the student give a wrong answer he looses the points of the question whereas in MCMA the system will give partial points). Also, there are questions with only two answer options: Yes/No and True/False respectively.

The application is structured on chapters having homework to each of them. After pushing the <*Finish*> button, the Student can view his result that will be also inserted in his grade book, as well as feedback from the questions he just answered, to justify his mark. The grade book includes the percentage obtains by everybody on chapters. As it can be seen in Figure 11, the professor will see, sorted by their mark, all students that finalized the exam, as well as the students that have assigned the respective exam.

Exam Details	Id	Student Name	Student Group	Exam Status	Mark
	749	Tescut Ovidiu	217/1	Finalized	8.5
	748	Brad Radu	217/1	Assigned	

Figure 11. Global statistic obtained by Professor with the Exam results

Figure 12 illustrates the individual statistic obtained by Professor with the Exam result of a certain Student. In a few words we want to explain why there are differences between *Percentage* and *Mark*. The Percentage is simply obtained by dividing the achieved points to the total points. The Mark is determined taking into account that it will range from 1 (the worst result) to 10 (the best result). Thus, the Student should increase with other 9 points to receive the best mark. In the example

from Figure 12, if he gets 50 out of 60 exam points, his mark will be computed with the relation: $1 + \dfrac{50}{60} \times 9 = 1 + 7.5 = 8.5$. If the professor has a good reason he could change the obtained mark (e.g. increasing all marks for an exam with a certain value or taking into account each student's activity during the semester).

Figure 12. Individual statistic obtained by Professor with the Exam result

Every mark that is generated gets also a timestamp and can be seen at any time within the student's account by clicking on the button "Gradebook" (see Figure 13). An important notice here is that the marks remain even if the professor deletes the exam that has generated them. Exams will be periodically moved into tables specially designed to store historical content some time after they were finalized, because otherwise they would hold vast amounts of database data (example: 5 specializations x 500 students x 7 final exams x 30 questions x 4 answer possibilities = 2.100.000 entries in ExamAnswer table – it would make database access to this table almost impossible in a timely manner).

Professor	Subject	Mark	Date & Time
Traian Anghel	Algorithms and Programming Languages	8.75	7/14/2009 8:07:39 PM
Adrian Florea	Computer Organization and Design	9.5	6/17/2009 1:28:33 AM

Figure 13. Grade book for a student

For Essay question type, the professor will have to manually rate each answer – this can be done under the evaluation section of each exam as it can be seen in Figure 14.

Figure 14. Professors will manually rate essay type questions

CONCLUSIONS AND FURTHER WORK

Testing Assistant is an interactive training tool for evaluating student's knowledge and can represent a feasible solution for testing the knowledge from various educational domains. Testing Assistant is a very useful instrument in teaching process and could be a solution because reduces the time required for correcting exams. Also, the examination results are correct and it removes any suspicions about the obtained results. Testing Assistant provides online course management, continuing professional development for teachers, career development, and improving the educational process.

As further work we will improve the e-Learning module permitting to define lessons with text, images and video clips. Also, we will extend the e-Management module with proper administration of marks, averages, lessons and student's presence at courses / laboratories. These reports would be printed. Finally, we will implement facilities such as Blog, Forum and FAQ (frequently ask questions) as learning management units.

REFERENCES

Dictionary of Quotes, http://www.dictionary-quotes.com/a-professor-is-one-who-talks-in-someone-else-s-sleep-w-h-auden/

Grindei L., Orza B., Vlaicu A. (2008), Multimedia technologies with interactive applications in e-Learning (in romanian), *Blue Publishing House*, Cluj Napoca.

Guerra L., Fabbri M., Maeran G. (2008), A problematic teaching model for the implementation of e-learning formative pathways, *Proceedings of the Conference Knowledge Construction in E-learning Context: CSCL, ODL, ICT and SNA in education*, Cesena, Italy.

http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp

http://www.cisco.com/web/learning/netacad/index.html

http://www.hp.com/education/

http://www.intel.com/education/highered/modelcurriculum.htm

https://academy.oracle.com/

Ilie M., Jugureanu R., Pacurari O.S., Dragomirescu E., Vladoiu D. (2008), Teacher training manual for using e-Learning platform (in romanian), *International Letter Publishing Hall*, Siveco Romania.

Moreland J., Jones A., Chambers M. (2001), Enhancing Student Learning in Technology through Enhancing Teacher Formative Interactions, *Research Information for Teachers*, 3, Wellington, New Zealand.