

Web-based "Questions-Bank" System to Improve E-Learning Education in Qatari School

Hosam F. El-Sofany, Noor Al-Jaidah, Shaima Ibrahim and Salha Al-kubaisi
Department of Computer Science and Engineering,
College of Engineering, Qatar University, Doha, Qatar

Abstract: Problem statement: This study described the analysis, design and implementation of the Questions-Bank system that allows the students of primary, preparatory and secondary schools to take web-based quizzes and exams, to download course reviews and previous exams. "Questions-Bank", means that each question has a weight and area of knowledge deals with it. The system facilitated generation of automatic, balanced and different exam sheets, that containing different types of questions, covering the entire curriculum and displaying gradually from easiness to difficulty. The exam sheet produced by the system, takes into account the different levels of the students from excellent, good, to fair and avoids any mistakes of language and non-clear terminologies. **Approach:** We had used the V-model to develop the system, this model started by specifying the system requirements, analyzed and classified the requirements into functional and nonfunctional requirements, constructed the use cases and domain model, the system design was started by determining the database tables and appropriate interfaces and finally, the implementation phase was launched by implementing the functionalities of the system. **Results:** Our system was used successfully in distance learning as well as in self-training. We have tested the system with different type of courses taught in the schools, ranged from primary to secondary levels. The feedbacks of both teachers and students were highly promising. **Conclusion:** Implementing this system will allow the schools to create a Questions-Bank database. The system allowed the school 'students to take quizzes, free exercises and exams, to download course reviews and previous exams.

Key words: E-Learning, educational systems, web-based evaluation systems, interactive applications

INTRODUCTION

Introducing Internet at schools has encouraged the development of new tools and systems within the scope of education and training. As a result, a new era of learning and training approaches has emerged, where new educational models have been developed, students could learn independently at any time or location, simply by connecting to the Net along with the appropriate systems and tools. Teachers, at the same time, could teach through "on-line" setting and could schedule lessons and exams without the classical physical constraints. This electronic learning approach, also known as e-learning, has opened new horizons in teaching for both teachers and students^[1-4].

The assessment process in an educational system is an important and primordial part of its success to assure the correct way of knowledge transmission and to ensure that students are working correctly and succeed to acquire the needed knowledge. Many assessment

methods can be used: conducting some experiments, realizing different mini-projects, making quizzes and exams. The evaluation mechanism help students to discover whether they have gained the required knowledge as expected and provide appropriate and timely feedback to teachers to adjust their lectures, exercises and exams^[5-7].

In this research project we have introduced the web-based "questions-bank" system, which can be used with any school courses and may save hours in exams and quizzes preparation and correction, as well as may save resources like, photocopying and distributing the exams papers, locations of these exams, teachers and assistants. The proposed system has the following functionalities and features:

Knowledge evaluation: Students can evaluate their understanding levels and teachers can track the activities of their students and can guide them to reach the pre-determined objectives of the courses.

Corresponding Author: Hosam Farouk El-Sofany, Department of Computer Science and Engineering, College of Engineering, Qatar University, P.O. Box 2713, Doha, Qatar Tel: +974 5626625

Exams generation: Teachers can store many types of questions like multiple choices, true/false and fill in the blanks. The system uses an intelligent algorithm to generate balanced exam sheet, that containing different types of questions, covering the entire curriculum and displaying gradually from easiness to difficulty.

Exam grading: quizzes and exams can be marked automatically by the system. Thus, students can get instantly their marks in a given quiz immediately after submitting their responses and see also the corrected answers.

Communication: The system has a communication tool that allows students and teachers to interact together and discuss their activities.

Course management: The system is highly flexible and doesn't necessitate any programming skills from its potential users. Thus, teachers can add, delete and update their course materials at any moment through the web.

Questions-Bank database: The system allows the schools to create a Questions-Bank database that stores the previous exams, the model answers, the reviews and useful exercises for each course.

MATERIALS AND METHODS

System overview: The system interface is entirely web-based and doesn't necessitate any technical skills from the potential users. The system is divided into three applications, called administrator, teacher and student application. As shown in Fig. 1, the system is appropriate for on-campus teaching, with access to the Internet, or off-campus teaching using the local network.

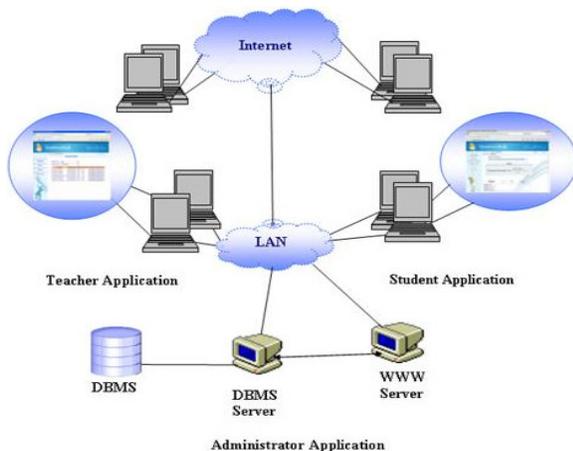


Fig. 1: System overview

The administrator application: This application provides the school administrator with simple tools to manage the information of the Questions-Bank system. The following are some features provided by the administrator application:

- Activate/Inactivate the system
- Import and Export the questions-bank database
- Send automatic email immediately containing the new password whenever the user changes his/her password
- Administer the basic information of the system such as: Levels data, subjects or courses data, classes' data, teacher's data and students' data

The teacher application: This application provides the teachers with various tools to construct the system services. The following are some features provided by the teacher application:

- Change account password
- Manage (add, update, delete and display) all the system services (reviews, exam questions, quizzes, previous exams and model answers)

The student application: This application provides the students with interactive tools to use the services of the system. The following are some features provided by the student application:

- Change account password
- Download assignments, revision documents, previous exams and other files that are uploaded by the teachers
- Take web-based quizzes, exams, assignments, interactive tutorials and reviews

Adopted software development process: Software development process represents the structure imposed on the development of a software product, which includes software life cycle and software process. There are several models for such processes, each describing an approach to a variety of tasks or activities that takes place during the process. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code and sometimes installing and checking out the software for operational activities. Note that, these activities may overlap or be performed iteratively^[8].

Life cycle model: Numerous models have been proposed to handle the problems of defining activities

and associating them together. Our adopted software development process is the V-model, as shown in Fig. 2, which can be presumed to be the extension of the Waterfall model^[9]. Instead of moving down in a linear way, the process steps are bent upwards after the implementation phase, to form the typical V shape. In the development of the Questions-Bank system, the V-model was followed by:

- Specifying the system requirements using different gathering approaches
- The requirements were analyzed and classified into functional and nonfunctional requirements and use cases and domain model were constructed
- The system design was started by determining the database tables, appropriate interfaces, class diagram: To show a static view of used classes and interaction diagrams: To show the collaborations among the objects of the system to complete an action
- The implementation phase was launched by implementing the functionalities of the system

Each of the previously mentioned phases is associated to a certain phase of testing to verify and to validate that the work is accurately executed, if problems are found during the verification or validation phases, then a re-execution of left side of the V is performed to make fixes and improvements.

Methods and techniques to analyze the problem:

Scenarios: A scenario is an informal description of actions done by a specific user in order to make a use of the system. It can be considered as an instance of a use case. It is expressed as a story of how a user can access the system.

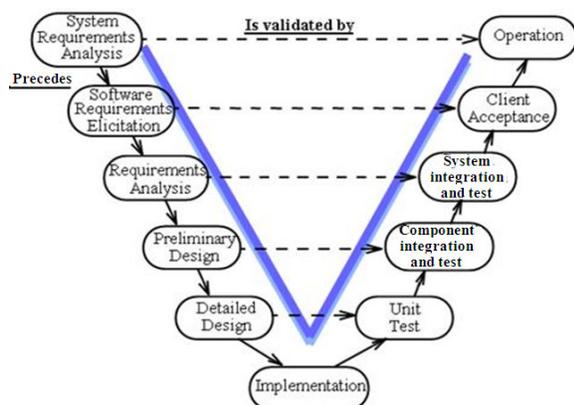


Fig. 2: The V-model - software development process

Use-cases: A use-case describes the interaction between the users of the system and the system itself through a sequence of simple steps that show how the system responds to a request from a certain user. It is mainly used to capture the functional requirements of the system. Use cases must be written in a language that is understandable by clients. They must describe the events from the point of view of the users.

Initial UML interaction diagrams: An interaction diagram demonstrates a certain aspect of a system by modeling its behavior. It is used to visualize the collaboration among several objects to complete a certain task.

Initial static class diagram: An initial class diagram (also called the domain model) consists of class and object diagrams. It is a static model that furnishes a visual representation of the real-world concepts that are visible to the user not of software components. Elements of a domain model are classes and the relationships between them. It is widely used as a source of inspiration for designing software objects.

System analysis: The analysis phase is the chief phase in which the system requirements are identified in more details. The goal of this phase in the system development is to refine the system goals into defined functions and operation of the intended application. System requirements are documented in a complementary set of artifacts: flow charts use case diagrams, system sequence diagrams and so on. Each artifact provides a different perspective of the system under design and contains distinct requirements. The combination of these perspectives establishes the tasks that are to be accomplished by the system.

Available information gathering technique: It's difficult to build a solution if we don't know the requirements. The "elicitation" step is where the requirements are first gathered from the client. Many techniques are available for gathering requirements. Each has a value in certain circumstances and in many cases, we need multiple techniques to gain a complete picture from a diverse set of clients and stakeholders. Here is a look at some of the approaches that could be taken.

Interviews: The most common technique for gathering requirements is to sit down with the clients and ask them what they need. Our discussion would help us plan out ahead of time, based on the type of requirements that we are looking for:

- Gather facts, opinions and speculations
- Ask about specific details
- Ask for other sources of information
- Ask about the stakeholders' vision for the future
- Ask if they have alternative ideas
- Ask them to draw diagrams

Prototyping: Prototyping is a relatively modern technique for gathering requirements. In this approach, we gather preliminary requirements that we use to build an initial version of the solution. We show this to the client, who then gives us additional requirements. We change the application and cycle around with the client again. This repetitive process continues until the product meets the critical mass of business needs or for an agreed number of iterations.

Use case: Use cases are basically stories that describe how discrete processes work. The stories include people (users) and describe how the solution works from a user perspective. Use cases may be easier for the users to articulate, although the use cases may need to be distilled later into the more specific detailed requirements. At this stage, client requirements are gathered. This is done on the basis of information provided by the client in the form of documents. Requirements could be classified in two types: Functional and Non-Functional requirements.

System specification:

Functional requirements: Describe the interactions between the system and its environment independent of its implementation. The environment includes the end user and any other external system with which the system interacts.

The Questions-Bank system should allow different types of users to interact with the system according to given privileges. There are three main users, as shown in Fig. 3, school administrator, teacher and student. It provides various services for each one:

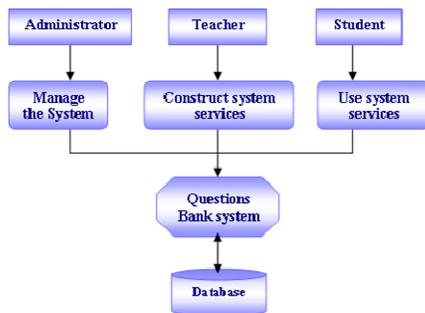


Fig. 3: The three main users of the system

- Questions-Bank system allows the school administrator to:
 - Login (as administrator)
 - Change account password
 - Get new password instead of the lost one
 - Activate/Inactivate the system
 - Renew data (Import the data from the Excel file to the SQL server database, delete data from specific table)
 - Send an email immediately containing the new password whenever the user changes his/her password
 - Manage (add, update, delete and display) the basic information of the system
- Questions-Bank system allows the teacher to:
 - Login (as teacher)
 - Change account password
 - Manage (add, update, delete and display) all the system services (reviews, exam questions, quizzes and previous exams with their model answers)
- Questions-Bank system allows the student to:
 - Login (as student)
 - Change account password
 - Get new password instead of the lost one
 - Download assignments, revision documents, previous exams and other files that are uploaded by the teachers
 - Take web-based quizzes, exams, assignments and interactive tutorials

Non-functional requirements: Describe aspects of the system that are not directly related to the functional behavior of the system. Non-functional requirements include a broad variety of the requirements such as implementation requirements, interface requirements, operation requirements and others.

- Usability: The system must provide a coherent user interface that is consistent with the organization interface. The system must be obvious, allowing the user to navigate the system easily
- Reliability: The system must perform and maintain its functions in routine circumstances
- Supportability: The system must be easily modified or extended by more functions
- Performance: Performance requirements are concerned with quantifiable attributes of the system, such as response time, throughput availability and accuracy

- Security: The system must prevent unauthorized users to access the system. Authorized users must be provided with user ids and passwords. In order to achieve high level of security, the password must be encrypted

Hardware and software resources: We have used some software during our system development such as: Windows XP professional, Windows Server O/S, Microsoft office Excel, Microsoft Word, SQL Server database system, MS Visual Studio.Net and Photo Shop.

System models:

Scenario: A scenario is a specific sequence of actions and interactions. It is one particular story of using a system, or one path through the use case. In our system, we have developed a simple scenario that illustrated the basic actions in the system.

UML use case: UML use-cases, were used in the analysis, in which the functional requirements are extracted and described along with a scenario of the flow of events. Universal Modeling Language (UML) is an industry standard language for specifying, constructing, visualizing, documenting artifacts of a software system.

System design: The design goal of our system is to resolve some of the issues, specifically; it is to be made easily accessible and customizable. Students will not be afraid to use our system. The system will be designed as a web application requiring only a web browser administer and use. Hence, no additional software needs to be downloaded.

Design methodology and patterns: A methodology is a collection of methods for solving a class of problems. It specifies which method should be used and the manner in which it is utilized. Software development methodologies decompose the process into activities. They provide methods for several activities as shown in Fig. 4.

Design pattern is a general re-usable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or a template for solving a problem that is common among different situations. It typically shows relationships and interactions between classes or objects.

Database design: Database is an important component in our system. Microsoft SQL Server database is the database management system used for the system.

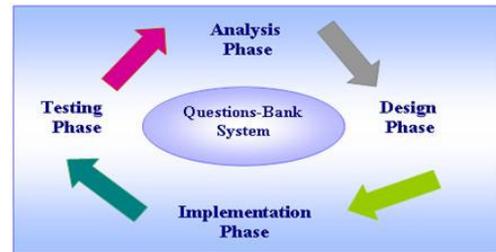


Fig. 4: Software development methodologies

The implementation of the database has been developed and modified many times to be applicable with the system requirement.

The DBMS stores the subjects in each school level, the reviews, the previous examinations, the exams questions, the quizzes, the student grades and the basic information of the administrator, teacher and student. We have chosen the entity-relationship model to represent the database design^[10]. Figure 5, shows the basic entity-relationship diagram, that contains the main tables used for creating the Questions-Bank database.

Application logic design: Application logic design will deal with three main parts of the application design:

- First part: The architectural design of the system which is concerned with the tiers of the system
- Second part: The upper level design is further introduced as an identification of the distinct subsystems and their functionalities
- Third part: The detailed design is discussed with the class diagram and sequence diagrams

System architecture: The web-based "questions-bank" system consists of three components as shown in Fig. 1. The client, the server and the database management system (DBMS). The client side application is developed by web-based programming languages such as XHTML, ASP.NET, Macromedia Flash and Java script. The server application is developed by Microsoft Visual Studio (ASP.NET, C#.NET and VB.NET) and acted as a gateway between the database and the clients.

The school information, user's information, course materials, questions, quizzes, exams, model answers, reviews and tutorial are stored and managed using Microsoft SQL Server database. The database and the server applications will run on Microsoft Windows Server.

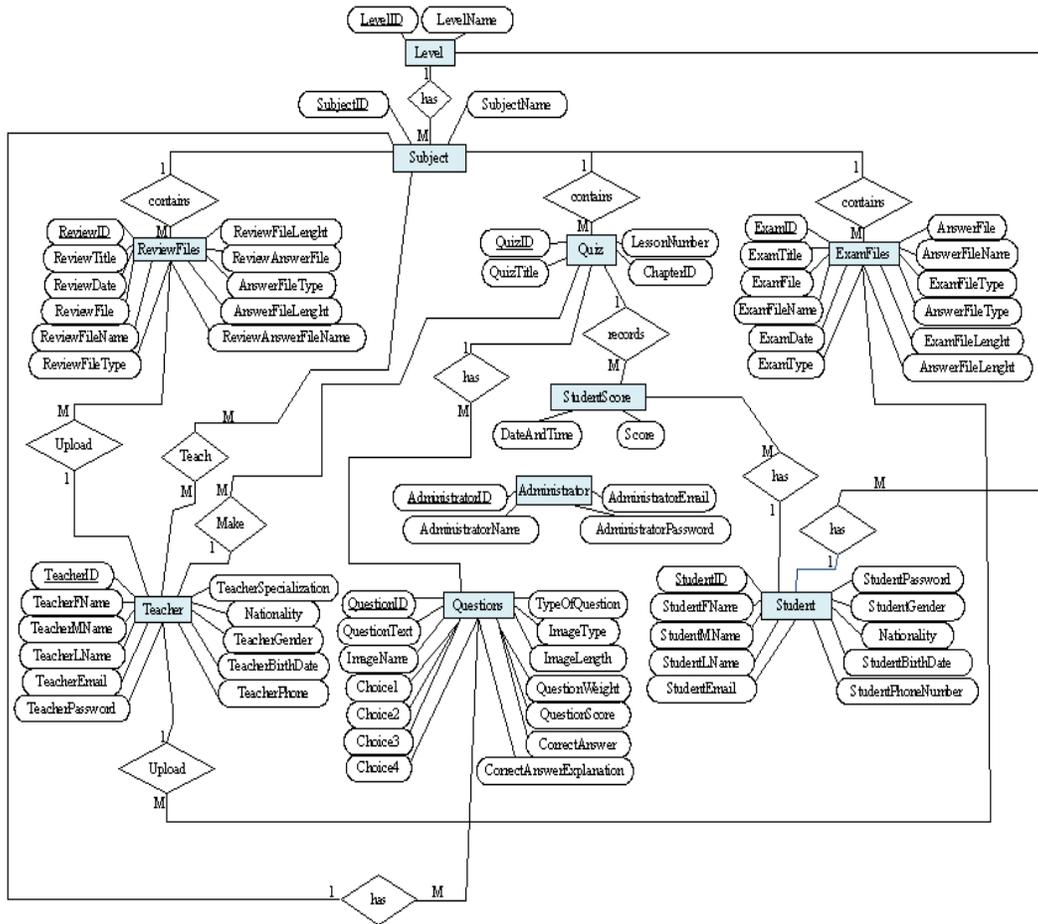


Fig. 5: The basic Entity Relationship Diagram (ERD)

Quiz generation algorithm: The DBMS of the "Questions-Bank" system includes an innovative and intelligent algorithm for managing the difficulty of the quiz questions^[2]. The algorithm uses the difficulty attribute set by the user as input parameter. The teachers set the difficulty levels of their questions at creation phase using a numbering system. The value assigned to a specific question identifies how difficult (or easy) it would be for a student to reply correctly to the question. Our algorithm can generate automatically different quizzes with different levels of difficulties. Obviously, teachers are not always as accurate as would be desirable in estimating the difficulty of a question (as they don't know the exact level of the students using the system) our algorithm can update the difficulty and the easiness of the questions based on the responses given by the students. The algorithm has the ability to detect the difficulty/easiness of a question based on students reply and adjust the difficulty value

accordingly. As a result, some of the feedbacks of the students in a given course are used to adjust the level of the questions. The following are the methods of our algorithm:

- **Get_Difficulty():** This method determines the difficulty of a specific question
- **Calc_Margn():** This method calculates the lower and the upper bounds of error percentage in the students' responses
- **Calc_Error():** This method calculates the percentage of wrong responses to a question, as a ratio between errors and a constant value named Max_Num_Access. The involved variables are reset after each difficulty evaluation
- **Mark_Ques():** This method activates a Boolean field in the register of a question if the difficulty level reaches the thresholds

Table 1: Questions levels

Difficulty	Lower bound (%)	Upper bound (%)
1	10	30
2	30	50
3	50	70
4	70	90

Four difficulty degrees have been considered rangy from level one to level four. If a question has a value higher than four, it could be too difficult or it could be ambiguous. Similarly, if a question has a value of 0, the question could be too easy, or too obvious and nearly all the students can answer it correctly. The questions that are out of bounds are marked, so they cannot be used for any quiz and the teachers must update them. Table 1 shows the upper and lower bounds for determining the difficulty of the questions.

Algorithm: ManageQuestion (Difficulty):

1. Access random question from the "questions" database table.
2. Check the Max_Num_Access of the question
Get_Difficulty(); Calc_Margn();
Calc_Error();
3. Recalculate_Difficulty();
4. if (Error < LowerBound) then
Difficulty--;
else if (Error > UpperBound) then
Difficulty++;
5. if Difficulty is out-of-bounds then
Mark_Ques();

RESULTS

System implementation: The web-based "Questions-Bank" system takes the advantage of the latest development technologies in its implementation. It is an ASP.NET web application to be used on the Internet and was written using Microsoft's Visual Studio .NET development environment in the C# programming language. Code that executes on the server to process event handlers for events such as button clicks is written in C#. Microsoft's SQL Server 2008 is used for database operations, including the storage and retrieval of login information. Successfully logging into the system, users are taken to an action page that dynamically presents options to them based on their privileges.

Database implementation: The database is designed to have the following main tables: Level, Subject, ReviewFile, Quiz, ExamFiles, Questions, StudentScore, Student, Teacher and Administrator.



Fig. 6: The Questions-Bank sub-systems

During the development of the system, data for the testing of different functionalities was needed to be imported into the database tables. The tables in the database are filled using two different ways .The first way is to fill the tables by importing the data from an MS Excel sheet that contains the basic data from the school's database. The second way is done during the run time of the system.

System requirements implementation: To prove the functionality of the "Questions-Bank" system, we have implemented several web-based applications as components of the system. The Questions-Bank system is divided into three sub-systems, as shown in Fig. 6, the administrator, the teacher and the student sub-system.

The administrator sub-system: This sub-system provides the school administrator with web-based services to manage the information of the Questions-Bank system. The following are some services provided by the administrator sub-system:

Login: Before users are granted access to the "Questions-Bank" system, they must successfully login to the system. The user's accounts are associated with different levels of privileges. Username and password is the most common authentication method, used to access the system, as shown in Fig. 7. Login information is verified by comparing the entered username and password with the stored ones in the database. The authentication program on the Web server opens a database connection and retrieves the user record.

If the user input data matches a database record, the user is granted to access the system services. Otherwise, an "invalid login" message is displayed and the login page is displayed again.



Fig. 7: Administrator login



Fig. 8: Administrator sub-system

System management: The administrator sub-system provides the school administrator with web-based services to add (update, or delete) the basic data needed for the other two sub-systems, such as: levels data, subjects data in each level, school data, classes data, teachers data and students data, as shown in Fig. 8.

Change password: The "questions-bank" system provides the user to change his/her account password:

- The user login into the system from the user login page. The user clicks on the link "Change Password" and then he/she will be directed to the suitable page
- The user enters his/her old password, new password and re-enters the new password again, then he presses the "Change" button
- A validation is applied on the input data to ensure that they are filled with required information
- The user's old password is deleted and the new one is inserted using an update query



Fig. 9: Review administration (delete)



Fig. 10: Previous exams administration (upload)

The teacher sub-system: This sub-system provides the teacher with web-based tools to manage the information of the Questions-Bank system. In addition to the login and change password, the following services are available in the teacher sub-system:

Reviews administration: Teacher can use this service to upload (add) the MS Word, PP presentation and video files that contain the reviews for each subject. Also teacher can delete specific review(s) from the database and can display the stored reviews, as shown in Fig. 9.

The teacher can store the following data for each review: Level name, subject name, review title, review date, review file and the review answer file (if exist).



Fig. 11: Quiz administration (add new quiz)

Previous exams administration: Teacher can use this services to upload (add) the previous exams and their model answers for each subject, as shown in Fig. 10. Also teacher can delete specific exam(s) from the database and can display the stored exams.

The teacher can store the following data for each previous exam: Level name, subject name, exam type (midterm, final), exam title, exam date, exam file and the exam answer file (if exist).

Exam administration: Teachers can use this on-line tool to add, update and delete questions and answers for their exams and quizzes. The questions are inserted into the database dynamically, so new questions can be added during the configuration of a exam or quiz, as shown in Fig. 11.

The teacher can store the following data for each exam/quiz: level name, subject name, chapter number, lesson number, quiz title, quiz type (multiple choices, true or false, fill in the blank), question text, question image file, choice1, choice2, choice3, choice4, correct answer, question weight, question score and question explanation.

Print students scores: Teacher can use this tool to print the grade reports for the students who took the web-based quizzes (exams) proposed by the system, as shown in Fig. 16.



Fig. 12: Download review

The student sub-system: This sub-system allows the student to download reviews, to download the previous exams and their model answers and to take selected quizzes. In addition to the login and change password, the following services are available in the student sub-system:

Download reviews: Student can use this service to download the reviews files for his/her subjects, as shown in Fig. 12. The student select the subject name and the teacher name (as input) from the available list of values, the system search for the required query and displays all the stored reviews from the database, Fig. 13, shows the implementation of the "Review-algorithm", that used for this service.

Download previous exams: Student can use this service to download previous exams and their model answers for his/her subjects, as shown in Fig. 14. The student select the subject name and the teacher name (as input) from the available list of values, the system search for the required query and displays all the stored exams from the database.

On-line quiz: After a student login into the system from the student login page, he/she can select the quiz, where the questions will be selected randomly by our algorithm from the database. Our system covers many types of questions as mentioned before, such as: multiple choices, True/False and fill in the blank.

```

public partial class Review : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label12.Text = (string)Session["StudentName"];
    }
    protected void Download_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        SqlDataAdapter adapter;
        byte[] filedata, Answerfile;
        string fileName, fileType, answerType, AnswerName;
        SqlConnection connection1 = new SqlConnection(@"Data Source=" + "MCOELHELSPAFM"
        + @"\SQLEXPRESS;Initial Catalog=QBANK;Integrated Security=True");
        SqlCommand comm = new SqlCommand();
        DataSet Rev = new DataSet();
        comm.Connection = connection1;
        if (e.CommandName == "Download")
        {
            int rowIndex = Convert.ToInt32(e.CommandArgument);
            DataRow rowKeys = GridView1.DataKeys[rowIndex];
            int id = (int)rowKeys.Values["ReviewID"];
            try
            {
                connection1.Open();
                comm.CommandText = "Select ReviewFile From ReviewFiles Where ReviewID = @id ";
                comm.Parameters.AddWithValue("@id", id);
                filedata = (byte[])comm.ExecuteScalar();
                if (filedata.Length == 0)
                {
                    Response.StatusCode = 404;
                }
                else
                {
                    fileName = (String)comm.ExecuteScalar();
                    fileType = (String)comm2.ExecuteScalar();
                    Response.ContentType = fileType;
                    Response.AddHeader("Content-Disposition", "attachment; filename=" + fileName);
                    Response.Charset = "";
                    Response.Cache.SetCacheability(HttpCacheability.NoCache);
                    Response.BinaryWrite(filedata);
                    Response.End();
                }
            }
            catch
            {
            }
        }
    }
}

```

Fig. 13: The implementation of the "Review-algorithm"



Fig. 14: Download previous exams

The quiz will be over either whenever the student finishes it and select the "Finish" button, or whenever a pre-allocated time will be over. The system will calculate the number of correct answers and display the result. Finally, the student score will be saved in the corresponding database for possible tracking, as shown in Fig. 15 and 16.

The quiz is displayed based on question-by-question methodology. A navigator is included so the student can go backward and forward over the questions in single steps. When the last question is reached, the student can go back to the previous one and if the first question is reached, the student can go forward to the next questions.

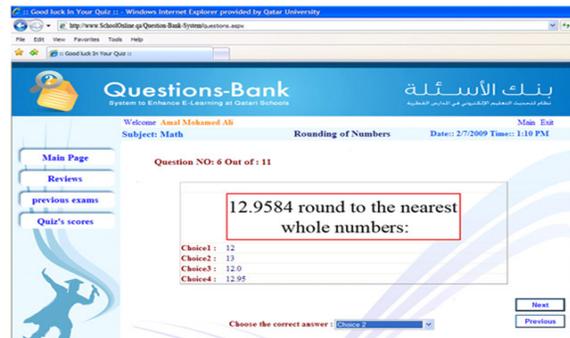


Fig. 15: Student quiz

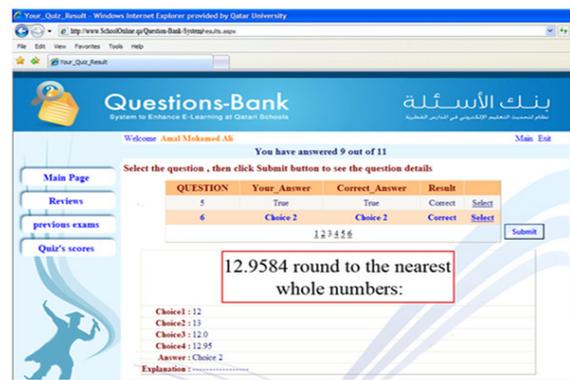


Fig. 16: Student quiz' scores

When the student finishes the quiz, a record will be created in the StudentScore table and stored in the database. Teachers can consult this table to track the activities of their students.

DISCUSSION

To obtain some feedback about the "Questions-Bank" system, we have prepared a questionnaire and distribute it to the potential students registered in the system. Sixty students have been randomly selected to use as a testing pool. A group of 30 students have used the hard paper quiz (and review) and another similar group has used our web-based system. Table 2 shows that both groups obtained approximately similar results. Students had to obtain a grade of 60% to pass the quiz. After taking the quiz, all the students had to fill in a questionnaire with general and specific questions related to the method of testing. The most significant questions reflecting student opinion are set out in Table 3, with the relevant group responses. The results of question 2 indicate that, independently of the method used, a vast majority of students want to know their grade as soon as possible as well as the correct answers.

Table 2: Quiz results of the two groups of students

Student Groups	Passed	Percentage
A-paper quiz and review	22	73.33
B-computer quiz and review	25	83.33

Table 3: Results of the Questionnaires of both Groups

	Group A (%)	Group B (%)
Do you trust the on-line evaluation system?		
Yes	65	90
No	35	10
When do you prefer to be told your quiz score?		
At the end	80	95
Later	20	5
How would you describe the process of entering answers in the computer?		
Easy	-	75
Acceptable	-	22
Difficult	-	3
Very difficult	-	-
How do you feel your result would have been, if the quiz had been on paper as the traditional way?		
Worse	-	25
Same	-	40
Better	-	5
Do not know	-	30
How do you feel your result would have been, if the quiz had been by the on-line system (i.e., computer-based)?		
Worse	40	-
Same	30	-
Better	20	-
Do not know	10	-

The results of question 5 indicate that, a large percentage of students who did not use the computer-based system felt that their marks would have been worse if a computer had been used. Similarly, question 4 shows that a percentage of student using computers felt that they might have done better using traditional methods.

Finally, the majority of the students trust more a computer-based evaluation system than classical methods. Some students did comment on the absence of printed copies of their answers and the fact that they could not compare their answers with the correct results after the quiz.

CONCLUSION

This study described the analysis, design and implementation of the Questions-Bank system that allows the students to take web-based quizzes and exams, to download course reviews and previous exams. The system facilitated generation of automatic, balanced and different exam sheets, that containing different types of questions, covering the entire curriculum and displaying gradually from easiness to difficulty. The system allows the schools to create a Questions-Bank database that stores the exams questions, the previous exams, the model answers, the reviews and useful exercises for each course.

Today, Mobile learning (M-learning) can use a collection of tools that tutors and developers can combine to deliver their teaching resources.

We intend to improve our research work, that introduced in^[11], by implementing the prototype "Mobile Questions-Bank" system (MQB), that provides the students to take wireless quizzes, wireless reviews and wireless exams.

In order to develop the MQB system, three mobile technologies will be evaluated to use:

Short Messaging Service (SMS): The questions could be sent to students via SMS messages, after which the students could reply with the answers via SMS messages. This approach is susceptible to delays because delivery times of SMS messages may vary greatly. When responding to a stream of questions, there is possibility of the SMS messages being duplicated or delivered out of sequence, causing errors in scoring.

Wireless Application Protocol (WAP): The questions could be displayed in the student's mobile phone as a very simple Internet web page based on the WAP format. The student could then select the response using the mobile phones navigation keys. The questions and answers would be transmitted using GPRS (General Packet Radio Service) or 3G (3rd. Generation) mobile data networks. However WAP generally offers an unfriendly user interface that may be confusing to students. When the mobile data transmission breaks due to network issues, it is not possible to recover from the error and show the student a helpful message.

Java 2 Micro Edition (J2ME) and eXtensible Markup Language (XML): The student could download and install a small J2ME application that would interact with the server over the GPRS and 3G mobile data networks, accessing XML questions file and transmitting responses as XML answer file to the server. Unlike WAP, J2ME applications have full control over the mobile phone screen to ensure that the application looks and operates the same on any mobile phone. J2ME applications also have full control over the data transmission and can retransmit the data in case of error.

ACKNOWLEDGMENT

The researchers acknowledge the financial support (Qatar National Research Fund- UREP 4-2-3) received from Qatar Foundation, via Qatar University, Qatar.

REFERENCES

1. El-Sofany, H.F., S.A. El-Seoud, F.F.M. Ghaleb, S.S. Daoud, J. M. AL Jaam and A.M. Hasna, 2007. XML and databases for e-learning applications. *Int. J. Emerg. Technol. Learn.*, 2: 6-12 <http://online-journals.org/i-jet/article/viewArticle/190>
2. El-Sofany, H.F., A. Hasna, J. Jaam, F. Ghaleb and S.A. El-Seoud, 2006. A web-based e-learning system experiment. *Int. J. Comput. Inform. Sci.*, 4: 22-29. http://www.ijcis.info/International%20Journal%20of%20Computing%20and%20Information%20Sciences_files/Vol4N1_EBEL/pp%2022%20-%2029.pdf
3. Abdel Wahab, M.S. *et al.*, 2006. Case study: Design and development of an online course. *Proceeding of the 9th International Conference on Interactive Computer Aided Learning*, Sept. 27-29, Villach, Austria.
4. Badariah, S. and S. Rosnafisah, 2006. Rapid e-learning content management system. *Int. J. Comput. Inform. Sci.*, 4: 1-9. http://www.ijcis.info/International%20Journal%20of%20Computing%20and%20Information%20Sciences_files/Vol4N1_EBEL/pp%2001%20-%2008.pdf
5. Lister, R. and P. Jerram, 2001. Design for web-based on-demand multiple choice exams using XML. *Proceeding of the IEEE International Conference on Advanced Learning Technologies*, Aug. 6-8, Madison, WI., USA., pp: 383-384. <http://portal.acm.org/citation.cfm?id=876869.878058&coll=&dl=>
6. Fox, G.C., 1999. Initial discussion of use of XML for universal access, shared places on the web: XML for web-based collaboration and distance education. *Proceeding of the Conference on XML Developers*, Aug. 19-20, Montreal, Canada, pp: 1-40. <http://www.ibiblio.org/bosak/conf/xmldev99/fox/fox.ppt>
7. Bota, F., L. Farinnetti and A. Rarau, 2000. An educational-oriented framework for building on-line courses using XML. *Proceeding of the IEEE International Conference on Multimedia and Expo*, July 30-Aug. 2, Paris, France, pp: 61-75. DOI : 10.1109/ICME.2000.869536
8. Bruegge, B. and A.H. Dutoit, 2004. *Object-Oriented Software Engineering using UML, Patterns and Java*. 2nd Edn., United States, ISBN: 10: 0130471100, pp: 800.
9. Elmasri, R. and S.B. Navathe, 2007. *Fundamentals of Database System*. 5th Edn., Addison-Wesley, USA., ISBN: 0-321-41506-X.
10. El-Sofany, H.F. and S.A. El-Seoud, 2008. Towards the development of an m-learning system: A new stage to enhance higher education. *Proceeding of the 11th International Conference on Interactive Computer Aided Learning*, Sept. 24-26, Villach, Austria, pp: 1-4.