A Proposed Model for A Web-Based Course On "Introduction to Computers"

Yousry S. El-Gamal, Abd-Elhamid Elmaghrabi, Sherif M. Tawfik
Arab Academy for Science and Technology and Maritime Transport

Abstract

During the past 5 years, the manner in which information is shared over the Internet has been dramatically improved by the introduction of World Wide Web (WWW). The World Wide Web provides standards for representation of multimedia information and protocols, which allow access and navigation to multimedia documents in the Internet. Recently, the World Wide Web has become a major contender in distance education. Many faculties have found the World Wide Web to be useful to enhance presentations, and to create a learning environment for students outside of class. This paper presents a model Web-Based course On "Introduction To Computers" offered to freshmen of the college of Engineering at the Arab Academy for Science & Technology and Maritime Transport (AAST). The model covers both the course material and student evaluation through exercises and quizzes.

Different tools such as Perl, JavaScript, Cookies, and HTML are utilized to implement the model in order to make use of the best features of each one for the optimal implementation of the model. Perl is utilized to implement automatic feedback, and record keeping of student's progress. JavaScript and Cookies help in controlling the quizzes access and timing, and automatic quiz submission. HTML offers the displaying of all the Web pages and the hyperlinks between them. Special emphasis has been given to exercises and quiz functions to achieve flexibility and automatic feedback without loss of control. The proposed course offers numerous advantages including convenience in use, better accessibility, enhanced communication with instructor and peers, automatic testing, quick feedback, and enriched presentation through hyperlinks with related material.

1. Introduction

The subject of distance education through the World Wide Web (WWW) is gaining increased consideration. Researchers and education organizations have observed that the WWW is a potentially valuable tool for education. By its nature, the WWW facilitates the distribution of knowledge and instructional resources. A recent survey reported that for 50% of Internet users in the US and Europe, the primary use of the Web was for education[1]. It's a popular and useful instructional medium for number of reasons[1, 2, 3]: The ability of WWW to integrate graphics, text, and sound inside one multimedia document called Web page, the ability of WWW to handle interactive Web pages, the ease of use of WWW through navigation software, which integrates access to all types of Internet resources and its ability to access the latest version of documents. As a result of the ability of WWW to handle interactive Web pages (forms), it can be used to produce certain types of learning materials as quizzes and exercises. This encouraged a number of universities world-wide to establish a number of their offered courses on the Web. Some of them even allowed free access to their

Web-Based courses. However, implementations were quite different both in functions and utilized tools. The issue of using the Web to present exercises and conduct examinations for example has been handled differently in various models[4, 5, 6, 7, 8, 9]. While some models relied on e-mail for that purpose, others recommended the traditional written examinations for assessment. The issues of examinations access, control, timing, and security still need further investigation for effective Web-Based instruction models

The main objectives of this paper is to introduce a model Web based course, which can achieve the features of: Convenience in use, Better accessibility, Enhanced communication with instructor and peers, Automatic testing, Quick feedback, and Enriched presentation through hyperlinks with related material

The target course for implementing the model has been chosen to be (CC111) "Introduction to Computers".

The motivation for choosing the "Introduction to Computers" course is because the course has no prerequisites and aims at providing basic knowledge about computer building blocks, problem solving methodology, and simple programming concepts. Also, because this course is offered to all freshmen of the college of Engineering and Technology at AAST, and it is a prerequisite to all other computer courses.

2. Tools Used

The model has been implemented using Perl, JavaScript, Cookies and HTML. Processing the information entered into an interactive Web page such as quiz form is by far the most common use of CGI scripts. These scripts act as a link between the server and the external application for creating dynamic Web pages that are viewed by the Web browser[10]. ***Perl*** is one of the best utility languages that can be used to provide CGI scripts. Generating dynamic HTML pages is relatively a simple process when using Perl language. Popularity of Perl for creating dynamic Web pages can be attributed to several factors: cost, functionality, availability, and platform. ***JavaScript*** is one of a new breed of Web languages called scripting languages[11]. JavaScript is object based script language that supports the development of client and server applications[10,12]. It can be used to provide dynamic information, validating forms, and making pages interactive[11]. JavaScript offers several advantages to the programmer including a short development cycle, ease-of-learning, and small size scripts. These advantages mean that JavaScript can be easily and quickly used to extend HTML pages that are already on the Web[13,11]. ***Cookies*** were developed by Netscape. A cookie is a small piece of information, which is stored on a special text file called cookie.txt by the browser. Cookies store information between browser sessions; it survives when the user turns off his machine. Cookies allow Web site developers to store information on the client harddisk[12]. Using CGI scripts or JavaScript can create cookies. Behind every Web page we can view the HyperText Markup Language ***(HTML)***. It allows us to create Web pages that incorporate text, images, sounds, movies, and hyperlinks. HTML is a WYSIWYG (what you see is what get) typesetting language. HTML is not a programming language; it is simply a set of instructions, called tags, which are used to tell the browser how to present the Web page[14], and it is platform independent.

3. The Proposed Model

Figure (1) depicts the proposed model diagram that has been introduced to develop the "Introduction to Computers" course according to objectives of section (1).

The model contains the course registration, course information, course materials, course evaluation, student's follow-up, and their controller processes. These design issues of the model are summarized as follows:

Registration
Passwords of the first chapter and login Ids are provided for the registered students taking into consideration that the registration form is filled correctly, and every student have a unique e-mail address.

Course information
Allowing free entry for the (visitors/students) to view the course information needed to run the system.

Course material
1- The entry to the course chapters is controlled by chapter's passwords. Students have to solve the exercise of every chapter to get the password of the next chapter.
2- Presents the course in a segmented manner to help the students to focus on relatively small and manageable tasks.

Course evaluation
1- The student is asked to enter the chapter password and a valid e-mail address in order to start the exercise and submit it.
2- The student is allowed to access the exercise function as many times as needed to attain the success level required
3- The student is allowed to access the quiz only once.
4- The student is asked to provide a valid e-mail address in order to start the quiz.
5- The quiz form is automatically submitted if the quiz time is over.
6- The student can submit the quiz form through a submit key if he/she finishes before the quiz specified time.
7- An automatic feedback is provided immediately to the student.

Student's follow-up

Only the instructor is allowed to keep track of all the information that had been stored about students during their activities on the course home page.

4. Functions Of The Proposed Model

Figures (2,3) depict the structure function diagrams that have been introduced to develop "Introduction to Computers" course home page according to the proposed model. These functions can be classified as follows:

Static functions: Include all the functions of the system that can be viewed by the (visitors/students) of the site. Materials in these functions are normally static during the

course period. These functions are: - main course page, introduction, webmaster, teaching staff, course advisor, software needed, objectives, course catalog, course evaluation, and references.

Dynamic functions: Include all the functions of the system that can be viewed by the students when they access the site but with some conditions. The materials in some of these functions are dynamically changed in response to the information stored from the previous visits of the students. The materials of the other functions don't get changed (static), but the entries to these functions are dynamically changed as a response to the information entered by the students. These functions include registration form, course materials (chapters, and exercises), schedule (quizzes), and follow.

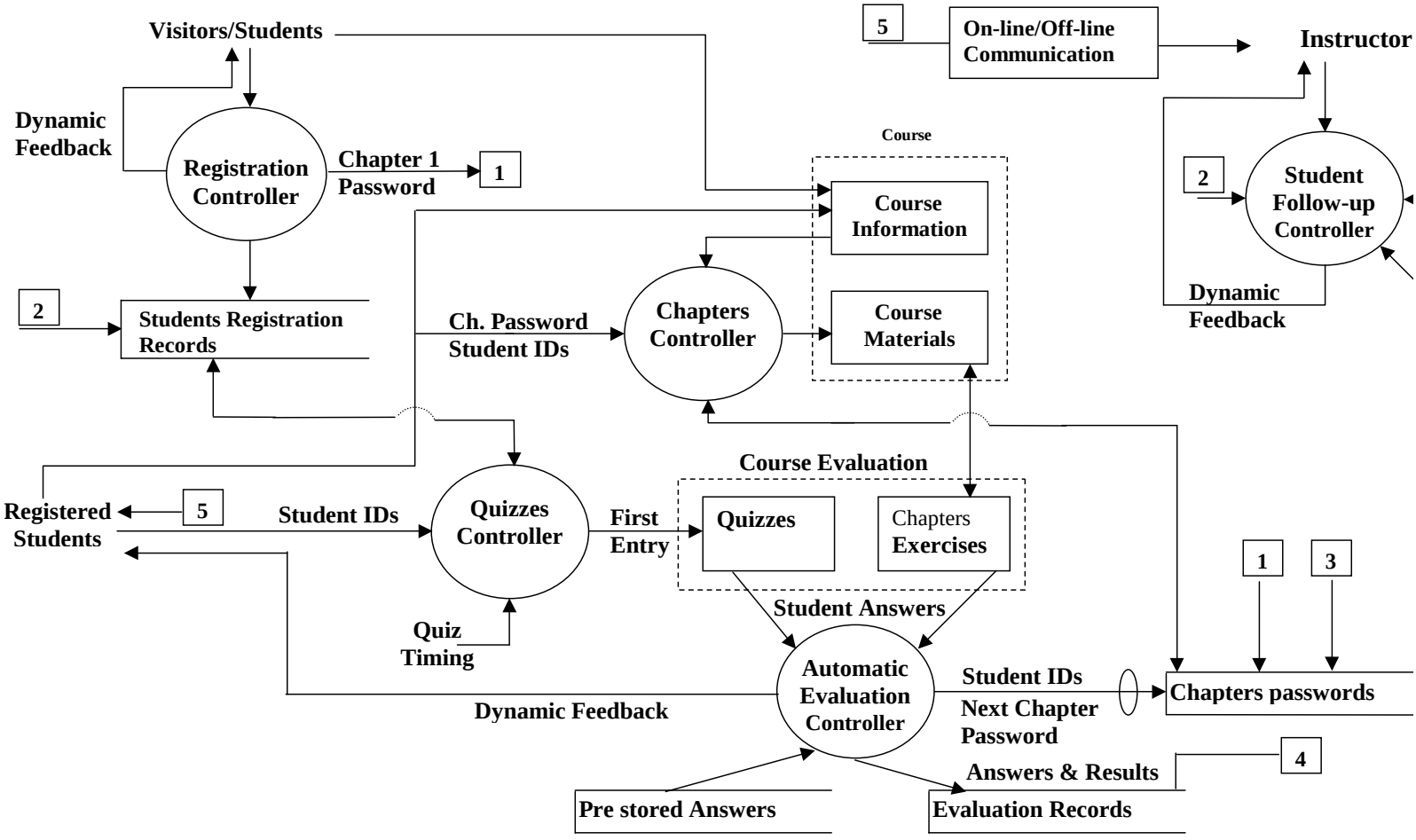

Figure (1) The Proposed Model Diagram

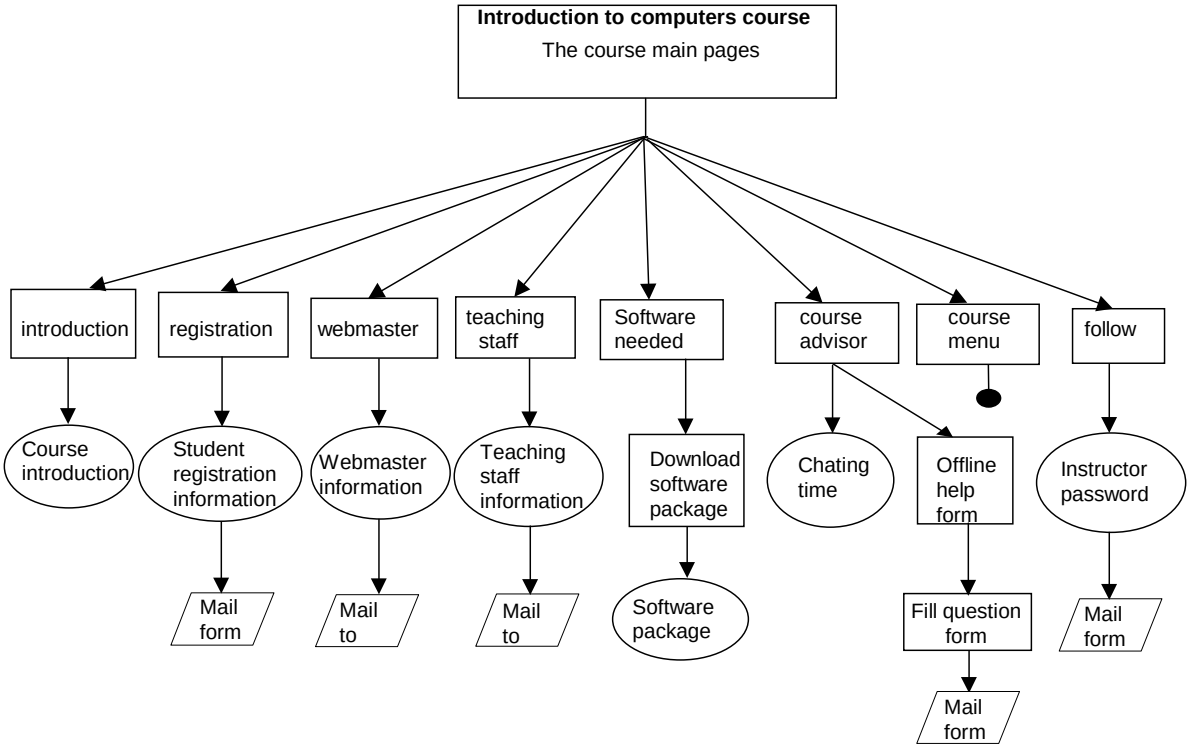

Figure (2) Course Main Web Page

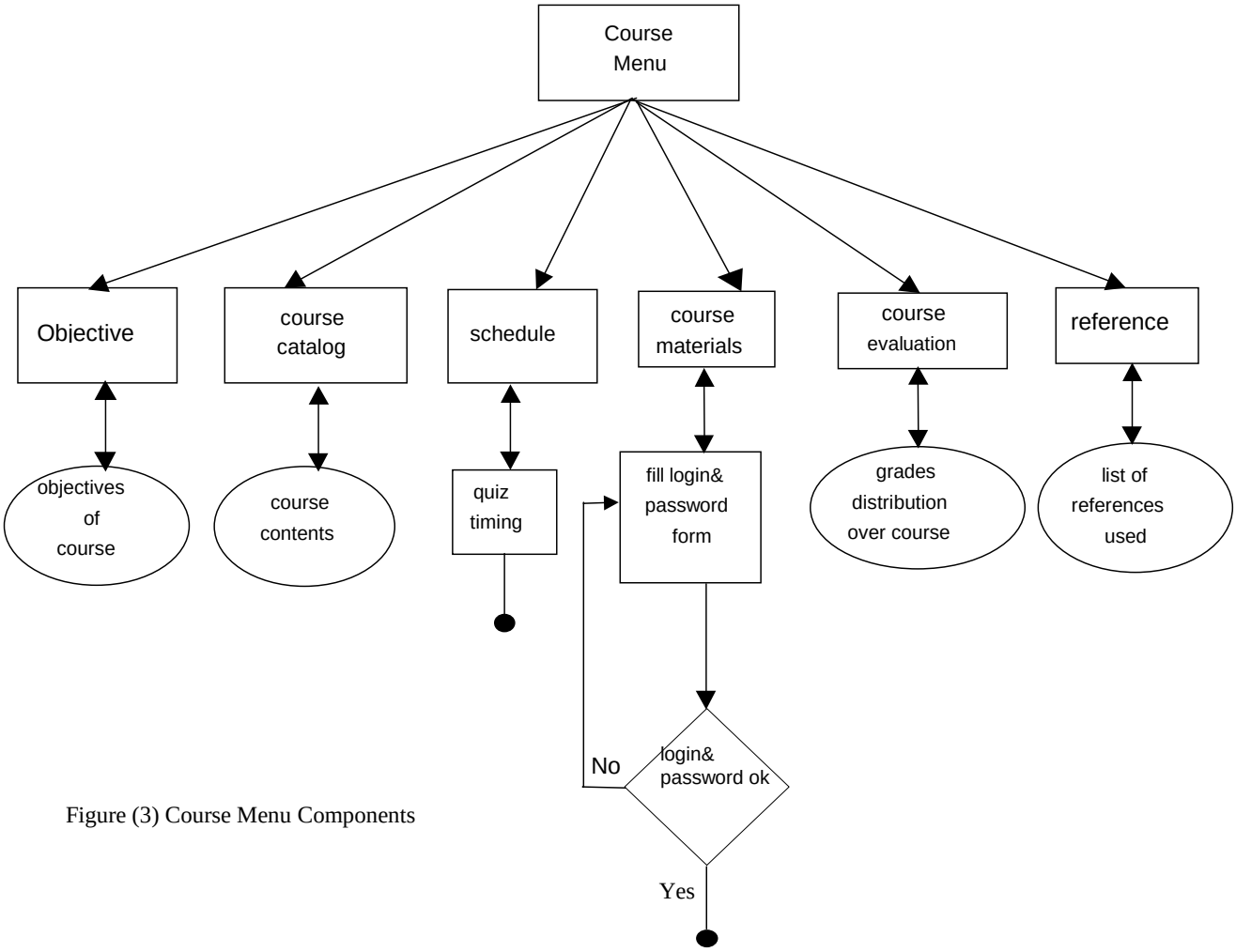

Figure (3) Course Menu Components

4.1 Registration Form Function

Figure (4) depicts the operation flow of the functionality of the registration form.

The registration form contains some basic fields that can be used to gather information about the students. Students must fill this form correctly to register in the course.

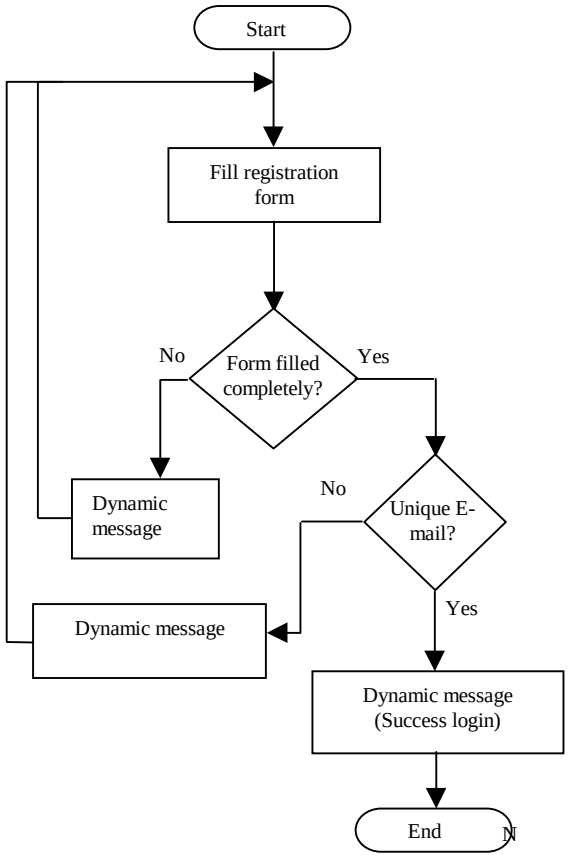

Figure (4) Registration process flowchart

The following checks are performed on the fields of the registration form:

1- A check on the (fullname, and city) fields to ensure that both fields accept characters only.
2- A check for the correctness of the (e-mail) field by checking for the "@" character, and the uniqueness of e-mail address for every student.
3- A check on the (phone, and fax) fields to ensure that both fields accept numbers only
4- A check to ensure that there are no any empty fields in the form

There are no other checks for the rest of the fields in the form, because they are menu selection item fields. This means that the student can select by clicking the mouse over an item from items list.

Once the form field checks are passed correctly then all the information of the student is saved into files and a dynamic message is provided to the student to give him the password, registration number, and login id.

4.2 Exercise Function

Figure (5) depicts the operation flow of the exercise function. Any student at the end of each chapter can recall the function, where questions are provided to cover each unit of the chapter contents.

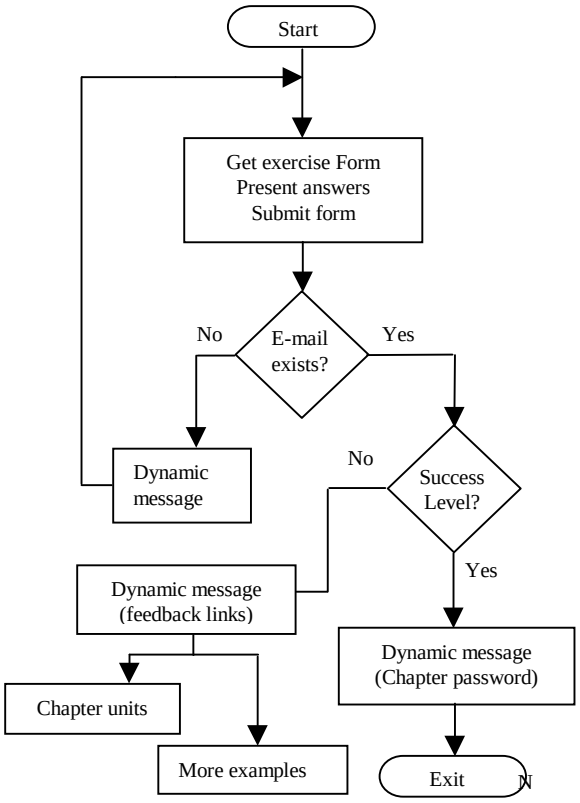

Figure (5) Exercise flowchart

Each chapter, except for the first one, has a special password, which is provided to the student when finishing the exercise of the previous chapter successfully. The student is allowed to access this function as many times as needed to attain the success level required. The student is asked to provide the chapter password and a valid e-mail address in order to start the exercise. Failure to do so results in a dynamic message generated by the system.

The student e-mail address should be provided to submit an exercise form. This would inhibit the student from accessing the exercise directly by providing the full URL address of the exercise form. Once the entry requirements are justified, the student takes the exercise and submits his answers at the end. His answers, scores, and the trial number are recorded and an automatic feedback is provided to the student. If success level is attained at that time, the score of the present chapter and the password of the next chapter are provided by the system. Otherwise, a dynamic message is generated carrying information regarding wrong answers, tips for the student, directions and hyperlinks to specific units of the chapter to help the student for a better understanding, and hence obtaining the required success level.

4.3 Quiz Function

Figure (6) depicts the operation flow of the quiz function. The function can be activated be any student registered in the course, through the hyperlinks in the quizzes schedule Web page as shown in figure (3).

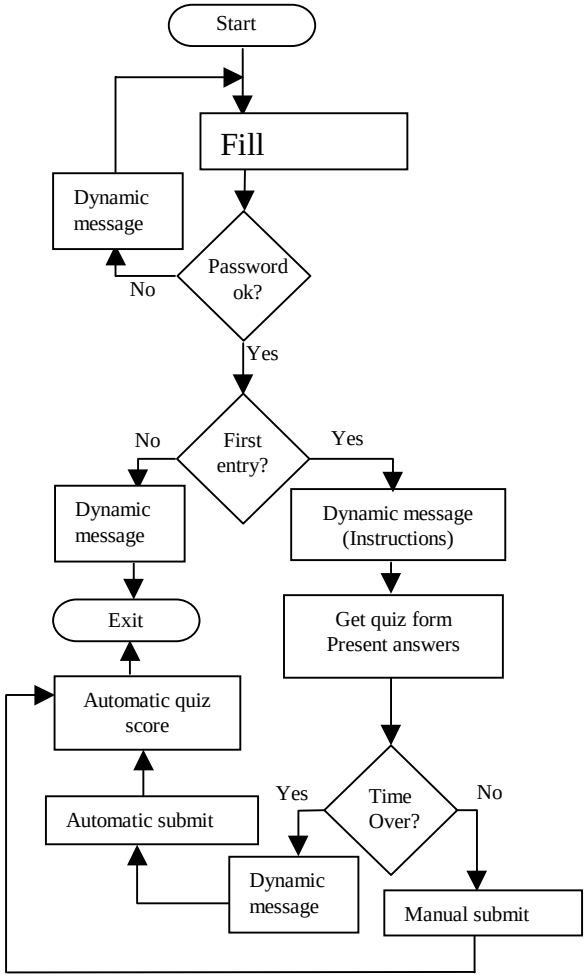

Figure (6) Quiz function

The hyperlinks to quizzes will be active only on the specified day for every quiz. Otherwise, all links will not be active.

A student is allowed to access this function only once. The student is asked to provide a valid e-mail address in order to start the quiz. Failure to do so results in a dynamic message generated by the system. If the student had a previous access to the function then the system will reject the access attempt, a dynamic message is generated, and the student will be forced to log out of this Web page.

Once the entry requirements are justified, a series of dynamic messages are generated to provide the student with all the instructions of the quiz. Then the student takes the quiz.

If the quiz time is finished before the student completes the quiz or submits the quiz by the submit key, a dynamic message is generated to tell him that the time is over and the quiz

form is automatically submitted. Otherwise, if the student finishes the quiz before the quiz time then he can submit the form through the submit key. In both cases, the answers and the final score of the student are recorded and an automatic feed back is provided to the student with his score.

4.4 Follow Function

Figure (7) depicts the operation flow of the follow function. Follow function is dedicated only to the instructor. The aim of the follow function is to allow the instructor to keep track of all the information about students that had been stored during student's activity on the course home page.

The follow function enables the instructor to obtain the following students information: (a) registration data of a particular students, (b) results of quizes/exercises of one or all students, and (c) lest of students that are logged on a specific chapter.

The function starts by asking the instructor to enter his password and the choice of (1) to select one student, or (2) for selection of all students. If password is valid then the choice field selection is examined. If the selection is one then a dynamic table is provided with all students registered for the course. Then he can select one of these students and choose the option (type of information) needed. If no data exists, a dynamic message is provided. Else, a dynamic page provides the required data. If the selection was two (all students), then the instructor also has to choose one option for the type of information requested.

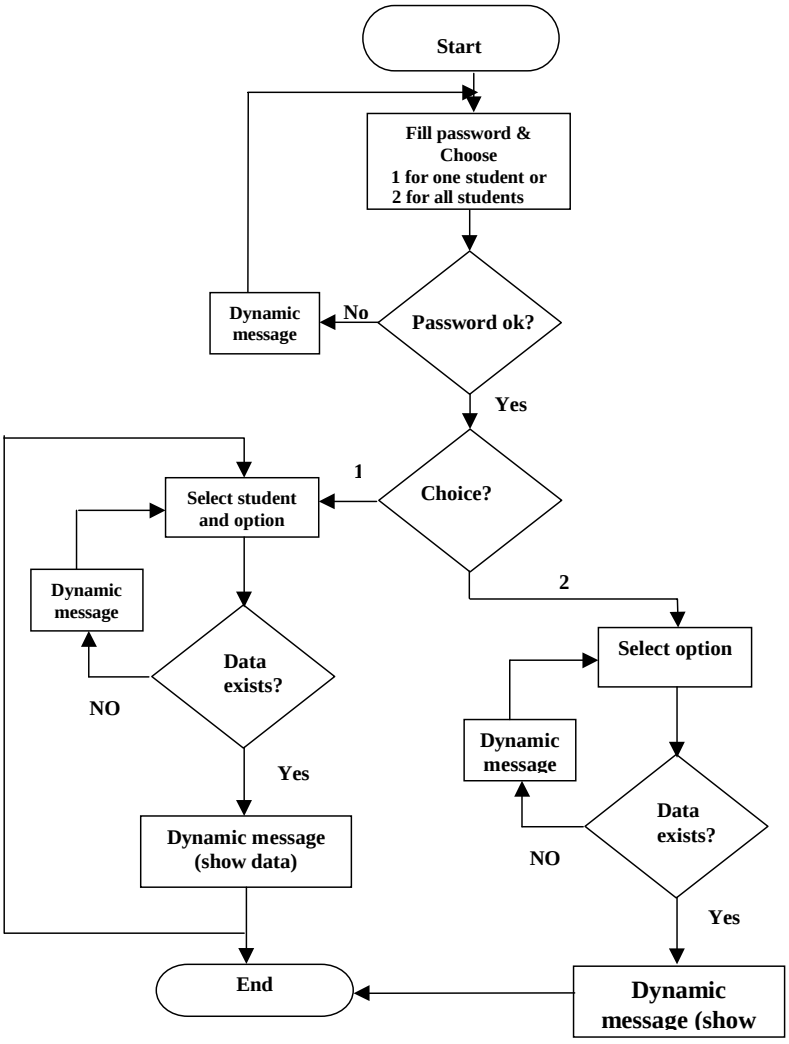

Figure (7) Follow function flowchart

5. Model Implementation

5.1 Registration Function

The registration Web page has been implemented using HTML and JavaScript. HTML is used to construct the form, which contains the entry fields. This is accomplished by using the "FORM and INPUT" tags. The HTML uses the FORM tag to begin constructing the form. The form has a name "registration", which is assigned to the form by using the NAME attribute. This form is submitted to the Web server by using the Action attribute, which illustrates the full path of the CGI-scripts written by ***Perl*** language. The INPUT tag is used to allow a user to enter data in an entry field.

JavaScript has been used to check the validity of the entry field of the form. Special code is written and embedded in the HTML file. If there is any unacceptable data entered in any field then the JavaScript code will stop processing the rest of the code and give an alert message through an alert window. All functions that have been written are called through using the Javascript event handler. These event handlers are linked into the INPUT tag, except the onSubmit event handler that must be linked into the FORM tag.

Using JavaScript in this way not only insures the correctness of entered data, but also saves the students time and frees the server for other processes.

When submitting the registration form to the server, one of two dynamic Web pages is expected to return to the student, indicating accepted or unaccepted registration form. Both dynamic pages are handled by the CGI-Script that resides on the server, written in Perl language.

Perl starts by reading the registration passed data through using special ready function from “CGI-lib” code library. This library is available free on the internet. Perl code extracts the e-mail field from the stream data, then its file handling features are used to open chapter password file. Perl code starts to compare the student e-mail that is entered from the form and all the e-mails that are pre stored in the file. If e-mail exists then a dynamic Web page is provided to the student, which contains the message "Your e-mail is the same as another student e-mail. Correct it and try again". Else, the Perl code opens the registration data file and stores all the form field, opens password data file and store the e-mail of the student and chapter password and opens the counter data file and increments a counter by one. Finally a dynamic Web page is created, which tells the student that his registration is accepted and gives him the password of chapter1 and his counter number. Perl can imbed HTML tags by using the print command.

5.2 Exercise Function

The chapter exercises function Web page is linked to the main chapter units by a hyperlink. This page is implemented using the HTML and JavaScript. This page contains multiple choice questions and e-mail entry field, which are constructed using the FORM and INPUT tags of HTML. Each question has several choices. To form

these choices, the INPUT tag has options called “radio buttons”, which have been used to construct the questions. JavaScript is used to insure that the format of the e-mail entry field provided by the student is entered correctly.

A CGI script written in Perl processes the exercise form. Perl code uses its features to open the chapter password file and compare the student e-mail entered through form and all the e-mails that are pre stored in this file. If an e-mail is not found, it stops processing the form and creates a dynamic Web page that contains a message “e-mail does not exist, try with a correct password”. Else, Perl code compares the student answers with pre stored actual answers, correcting every question and storing the answers, results, and the number of iterations into a file. If success level is reached, Perl code creates a dynamic Web page, which contains the password of the next chapter and passes it to the student browser. Else, Perl code creates a dynamic Web page, which contain every wrong answer and a hyperlink explanation for the weak points. Using the imbedded HTML tags inside the Perl script create these links.

5.3 Quiz Function

The quiz function is linked to the on-line course model through an intermediate Web page. It contains all the hyperlinks to all quizzes. This page is implemented using HTML. The first Web page in the quiz function asks the student for an entry password. HTML and JavaScript have been used to develop this page. JavaScript has been used to check the validity of the entry field and to check for the first time entry. Reading a Cookie from the client harddisk that recorded if the student had taken the quiz before does this. If not, it creates a Cookie for every student entry to the quiz on his hard disk. This form is processed by CGI script, which is written in Perl. Perl code resides on the server and it checks for the student password by comparing it with the previously stored passwords that were recorded into a stored file. Perl will return a dynamic page with an error message (password failure), if the password is wrong. Using the print command, which is capable to imbed HTML tags directly, does this, or loads the second page of the model “instruction page”. This page uses HTML and JavaScript. An important feature of JavaScript, which is the alert window, is used to construct this page. Each instruction is loaded into one of these alert windows. Students view every alert and move to the next one by clicking on it. The last alert leads the students to the quiz page. Controlling the quiz time is accomplished in this page; JavaScript has a ready-made method it that can be used to set the quiz time. When the quiz time is finished, an alert is provided to the student. The student is forced to click this alert. Clicking this alert leads to automatic submit of the quiz form to the server, which is done by another JavaScript method. A CGI script written also in Perl automatically corrects this form. The answers and scores of students are recorded into a file by using a group of simple Perl commands that are capable of handling files. At the end, Perl script creates a dynamic page, which contains the final score of the quiz.

5.4 Follow Function

The follow function is linked to the on-line course model through the follow key in the left part of the course main Web page. The first Web page in the follow function asks the instructor for an entry password and information choice. HTML and

JavaScript have been used to develop this page. This page is constructed similar to the quiz password Web page, but with an extra choice field; the selection field. This is another type of tags that can be used inside the FORM tag. The SELECT tag allows the user to select an item from many items. This is accomplished by using it in concatenation with the OPTION tag. Here, the instructor has two options: first choice if he wants information for only one student. The second if he wants information for all students. After selection of his choice, the form is submitted to be processed on the server by a CGI-Script. Perl code checks for the instructor password and the item selected by comparing it to previous stored password and item selection number into the CGI-Scripts. CGI-Script is constructed as similar to the quiz password CGI-Script. The Perl code that loads the selection items for all student HTML files is similar to the Perl code that has been used in loading the main chapter HTML file. On the other hand, the next Perl code depicts the operation of creating the Web page for the selection item for one student. This page contains two-entry selection field. The first is a selection field that contains all the e-mail address for all students registered in the course. The second is the item selection for the information that the instructor can activate.

6. Testing the System

Several tests have been carried out to examine the functionality of the system components using special set-up in a computer lab as follows:

6.1 Configuration

The system has been implemented and run in a computer laboratory in the College of Engineering at AAST. The lab contains 10 personal computers (IBM Compatible) with the following specifications: 32 MB RAM, 2.4 MB harddisk, and 133MHz Pentium processor. The ten PCs are connected together to construct a network. One of the PCs is selected to be the server, while the others act as clients. Several software packages are established on the server and clients. The server has the following software:

- Red Hat LINUX Release 4.0 Kernel version 2.0.18
- HTTP Server Apache 1.1.1 (Web server software)
- Perl Interpreter version 5.003

The clients have the following software:

- Windows 95
- Netscape Communicator version 4.0

6.2 Test Procedures

The following tests and checks are established:

Registration tests

1- A test for the uniqueness of e-mail of every student. The system rejects to register more than one student with the same e-mail.
3- A test for checking that there are no empty fields and all the fields of the registration form are filled correctly before submission.

Exercises test

1- A test for checking that the feedback result from the CGI scripts that handles the exercises Web pages to the students are correct.
2- A test for checking that the system accepts multi entry to the chapter exercise, and that every entry is recorded.
4- A test for checking that the system gives the password of the next chapter if and only if the student obtains the success level of the previous chapter.

Quizzes tests

1- A test for checking that the feedback result from the CGI scripts that handles the quizzes Web pages to the students are correct.
2- A test for checking that the system accepts only the first student trial to access the quiz and rejects all the next trials.
3- A test for checking that the quiz is automatically submitted to the server once the allowed quiz time is finished.

Security tests

1- A test for checking that the system will reject entry to the quiz, if the cookies are canceled.
2- A test for checking that the system refuses the direct entry to the quiz by its direct full path address (full URL).

Global tests

1- A test for checking that all the CGI scripts handles the data files correctly.
2- A test for checking that the entered data by the student at any submission processes is saved correctly in the specified files.
3- A test for checking that the system gives corrects feedback data to the instructor for one student or for all students.

7. Conclusions

In this paper, a Web based course model has been proposed. The proposed model covers both the course material and student evaluation. The proposed model offers the advantages of proper control, correct timing, automatic feedback, and student's progress record keeping. Also, the model helps the instructor to follow-up the progress of his students and evaluates.

The "Introduction to Computers" course (CC111) has been selected as a target course for implementing the proposed model, since it is offered to all freshmen of Engineering students with no prerequisites. A number of Web tools are utilized to implement the model in order to make use of the merits of each tool. HTML is employed for implementing static functions as course materials and hyperlinks. Perl is utilized for the automatic feedback to students in quizzes and exercises as well as

implementing student progress record keeping. Access control, time keeping and submission of quizzes are implemented using JavaScript and Cookies.

All of the functions introduced in this work are field-tested using a special set-up in a lab consisting of a server and 10 PC’s using Netscape browsers.

Bibliography

1. Jed S. Lyons and Stephen R. McNill, The Design of Material World, an Internet-Based Educational Environment, *Proceeding of the ASEE Annual Conference & Exposition*, Seattle, Washington, June 28- July 1, 1998.
2. M.M. Morcos, et.al, On Continuing Education Through Distance Learning, *Proceeding of the ASEE Annual Conference & Exposition*, Seattle, Washington, June 28- July 1, 1998.
3. Donn C. Ritchie & Bob Hoffman, Using Instructional Design Principles To Amplify Learning On The World Wide Web, *Published in SITE96 (society for Info. Tech. and Teacher Education 7th World Conference)*, 1996.
4. Steven Downes, Web-Based Courses: The Assiniboine Model, *Proceedings of N.A.Web'98, Fourth International North America World Wide Web Conference*, University of New Brunswick. Canada, 1998.
5. David B. Oglesby, et.al, Statics On-Line: A Project Review, *Proceeding of the ASEE Annual Conference & Exposition*, Seattle, Washington, June 28- July 1, 1998.
6. James A. Rehg, Developing Web-Based Courses Using an Online Development Guide and Templates, *Proceeding of the ASEE Annual Conference & Exposition*, Seattle, Washington, June 28- July 1, 1998.
7. Rod Byrnes, et.al, The Development of a Multiple-Choice and True-False Testing Environment on the Web, *Proceedings of AusWeb95*, 1995.
8. Curtis Dyreson, An Expermient in Class Management Using the World Wide Web, *Proceedings of AusWeb96*, 1996.
9. H. Jack, A paperless (almost) Stastics Course, *Proceeding of the ASEE Annual Conference & Exposition*, Seattle, Washington, June 28- July 1, 1998.
10. Jeffry Dwight and Michael Erwin, "Special Edition Using CGI", Que Corporation, 1996.
11. Laura Lemag and Michael G. Moncur, *Laura Lemag's Web Workshop JavaScript*, Sams.net Publishing, 1996.
12. James Jaworski, *Mastering JavaScript*, Sybex. Inc., 1997.
13. Arman Danesh, Teach Yourself JavaScript 1.1 in a Weak Second Edition, Sams.net Publishing, 1996.
14. Tom Savola, et.al, *Special Edition Using HTML Second Edition*, Que Corporation, 1996.

YOSRY S. AL-GAMAL

Dr. Yousry S. El-Gamal is the vice president of education and research of the Arab Academy for Science and Technology and Maritime Transport, Alexandria, Egypt. He is also the chairman of the Scientific Computer Society of Alexandria, Egypt. His active areas of research are computing systems, parallel architectures, educational software, Database systems and software reuse.
Dr. El-Gamal obtained his D.Sc. in commputer science from the George Washington Univeristy, USA 1985, his M.Sc. degree from Ain-Shams Univeristy, Cairo Egypt 1977, and his B.Sc. degree in Electrical Engineernig from Alexandria Univeristy 1968.

ABD-ELHAMID ELMAGHRABI

Dr. El-Maghrabi is a professor of marine engineering and assistant dean for education and research at Arab Academy for Science and Technology and Maritime Transport (AASTMT) at Alexandria, Egypt. He serves as a director of the learning resources center, and director of curriculm and teaching methods at AASTMT. He received a B.Sc. degree in marine engineering and naval architecture from faculty of engineering Alexandria University, Egypt and a Ph.D. in Education from Al-Azhar Univeristy, Cairo, Egypt

SHERIF M. TAWFIK

Eng. Sherif M. Tawfik is an analyst programmer in Arab Academy for Science and Technology. He received his B.Sc. in computer engineering from Suez Canal University, Port-Said, Egypt 1991, his M.Sc. in computer engineering from Arab Academy for Science and Technology. In addition to his work, sherif teaches in the American University Programs held at Alexandria branch. He also, work as teaching assistant under supervision of Dr. Yousry in the field of Internet applications in education.