

2004

Visualizing student performance on CIMEL exercises and quizzes

Sharmeen Mecklai
Lehigh University

Follow this and additional works at: <http://preserve.lehigh.edu/etd>

Recommended Citation

Mecklai, Sharmeen, "Visualizing student performance on CIMEL exercises and quizzes" (2004). *Theses and Dissertations*. Paper 838.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

Mecklai, Sharmeen

**Visualizing
Student
Performance on
CIMEL Exercises
and Quizzes**

May 2004

“Visualizing student performance on CIMEL exercises and quizzes”

by

Sharmeen Mecklai

A Thesis
Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science

in
Computer Science and Engineering

Lehigh University

May 2004

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

April 30, 2004
Date

6

Thesis Advisor

Co-Advisor

Chairperson of Department

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, **Dr. Glenn Blank**, for all his guidance and support. This thesis work was enabled and sustained by his vision and ideas.

Special thanks to....

... Hussein, for being my husband and for putting up with me and this thesis. I am truly blessed to have you in my life. Thanks for your understanding, patience, and support.

... Murat, for introducing me to .NET technology and his continued support throughout.

... My mother who encouraged me that I could finish my thesis in time

Table of Contents

LIST OF FIGURES.....	v
ABSTRACT.....	1
1 INTRODUCTION.....	2
1.1 Database Tracking of exercises and quizzes:.....	5
1.1.1 Data Organization in the MySQL database:	7
1.2 Quizzes in the multimedia:	8
1.3 Thesis Organization	8
2 CIMEL VISUALIZATION TOOL DESIGN	10
2.1 Login Design.....	10
2.2 Tool Menus and Submenus.....	11
2.2.1 View record by student	11
2.2.2 View records for all the students in the class	14
2.3 Approach to CIMEL User Interface Design	17
3 TOOL ARCHITECTURE.....	19
3.1 Overview.....	19
3.1.1 Creating a new web application using Visual Studio .NET.....	20
3.1.2 Building and running an application	21
3.1.3 Tool Box and Properties window.....	22
3.2 User Authentication	22
3.2.1 Redirecting to a different web page	24
3.2.2 Session Object.....	24
3.2.3 Postback events	25
3.2.4 Page_Load event	25
3.3 Database Connection and ADO.NET	26
3.3.1 DBConnection Object	27
3.3.2 DBCommand Object	28
3.3.3 DataReader Object	28
3.3.4 DataAdapter Object.....	28
3.3.5 DataSet	29
3.3.6 Data Binding	29
3.3.7 Connecting to the CIMEL database	29
3.4 ADO.NET.....	30
3.5 SQL in Table Creation.....	31
3.6 Graphs.....	33
4 TOOL MAINTENANCE.....	34
5 RELATED WORK.....	35
6 RECOMMENDATIONS AND CONCLUSIONS.....	38
7 REFERENCES	40
8 APPENDIX	41
8.1 How to generate new users for CIMEL.....	41
9 VITA.....	42

LIST OF FIGURES

Figure 1: TA persona responds to a wrong choice in a multiple-choice question	4
Figure 2: An example of a simple drag and drop exercise.....	4
Figure 3: An example of a more involved constructive exercise.....	5
Figure 4: A snapshot of a quiz question with feedback	6
Figure 5: A snapshot of the final score a student received on a quiz.....	6
Figure 6: A snapshot of CIMEL Visualization Tool Login Page	11
Figure 7: Sample output for a student's performance on the exercises	12
Figure 8: Sample output for a student's performance on the quizzes.....	13
Figure 9: Graph showing the number of times a student got an exercise correct	13
Figure 10: Graph showing the number of questions a student got correct in a quiz.....	14
Figure 11: Sample output for a class's performance on the exercises.....	15
Figure 12: Sample output for a class' performance on the quizzes	15
Figure 13: The percent of questions a class got correct in each exercise.....	16
Figure 14: The percent of questions a class got correct in each quiz.....	17
Figure 15: Creating a new ASP.NET web application.....	21
Figure 16: ADO.NET object model	26
Figure 17: The dialog box to add a connection string.....	30

ABSTRACT

When students use CIMEL multimedia online, the database receives information about each page they visit, each step they make with interactive exercises, and each answer they give to quizzes which may be given throughout the content. Instructors who want to understand or evaluate how their students are doing may be interested in this data, but may find the sheer volume of it overwhelming. They need tools to help visualize all this data. A tool for this purpose has been created and is web-based and platform independent so that instructors everywhere can access this tool easily. *This thesis describes the tool and its architecture. It is especially intended for developers who want to extend the functionality of the tools.*

1 INTRODUCTION

CIMEL is a multimedia framework which stands for Constructive and collaborative, Inquiry-based E-Learning supplementing computer science courses. Pioneers of CIMEL have defined *Constructive*, *Collaborative* and *Inquiry-based* learning as follows:

“*Constructive* learning goes beyond learning by receiving knowledge, to learning by building systems, with immediate, visual feedback.” “*Collaborative* learning encourages students to interact with instructors and librarians, via live links and remote-controlled “show me” sessions and by reviewing multimedia FAQs of recorded “show me” sessions”. “*Inquiry-based* learning guides the student into pursuing exploratory research in a community of students and scholars”¹.

An important activity in the CIMEL multimedia development is keeping track² of how students perform on exercises and quizzes, what screens they have visited as well as keeping note of all the interactions with the interface (such as changes of PREFERENCES and switches to JUST THE FACTS mode) . This data is subsequently analyzed for various purposes, such as trends on the class’s understanding of a chapter material as well as a student’s comprehension of the material. Analyses are often supported by visualization of the gathered data (which for the CIMEL project is stored in a central MySQL database server), using graphs, and tables so that it is easier to read/understand rather than using raw data.

¹These definitions have been taken from
http://www.cse.lehigh.edu/~cimel/papers/SIGCSE2002_abstract.htm

² Here by tracking I am referring to recording data into the cimel database and not the track editor defined originally in the CIMEL documentation

Students learn material with the help of interactive quizzes and constructive exercises. For each of these exercises, a TA persona provides feedback guiding a student through each exercise. Figures 1-3 are some of the examples of the kinds of exercises in CIMEL. Figure 1, shows an example of a multiple choice style exercise where a TA persona changes expression and provides feedback to the student hinting at what is wrong with a user choice. The constructive exercises are much more complex for example; this kind of exercise may challenge a learner to build solutions to problems by dragging and dropping pieces of structures into place, incrementally. Figure 2 is a snapshot from a sequence of screens, in which a TA persona asks the user to drag the page frames to show the pages that swap. Figure 3 shows an exercise that is a little more complicated than exercises in figures 1 and 2. In this exercise, the professor persona walks the learner step by step through the process of building an abstract data type for an Apple. Thus, in this exercise, the learner goes through a series of screens, incrementally building up an ADT specification for class Apple. Note, that at each step, feedback is provided to help the learner learn from mistakes as well as correct actions.³

³ This material has been taken from "Adapting Multimedia For Diverse Student Learning Styles" (Blank et.al) 2003

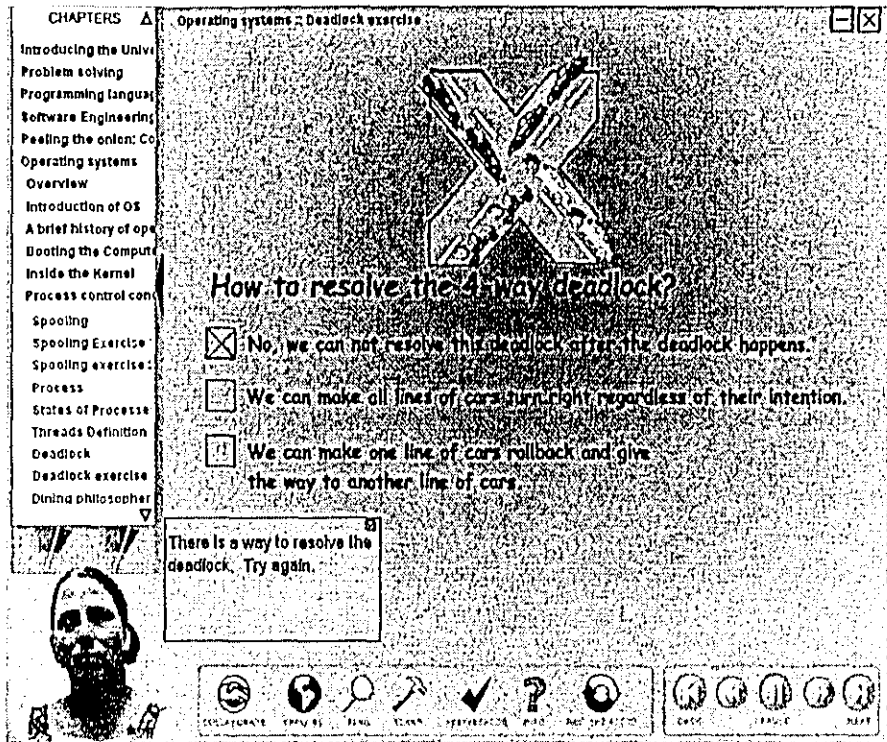


Figure 1: TA persona responds to a wrong choice in a multiple-choice question

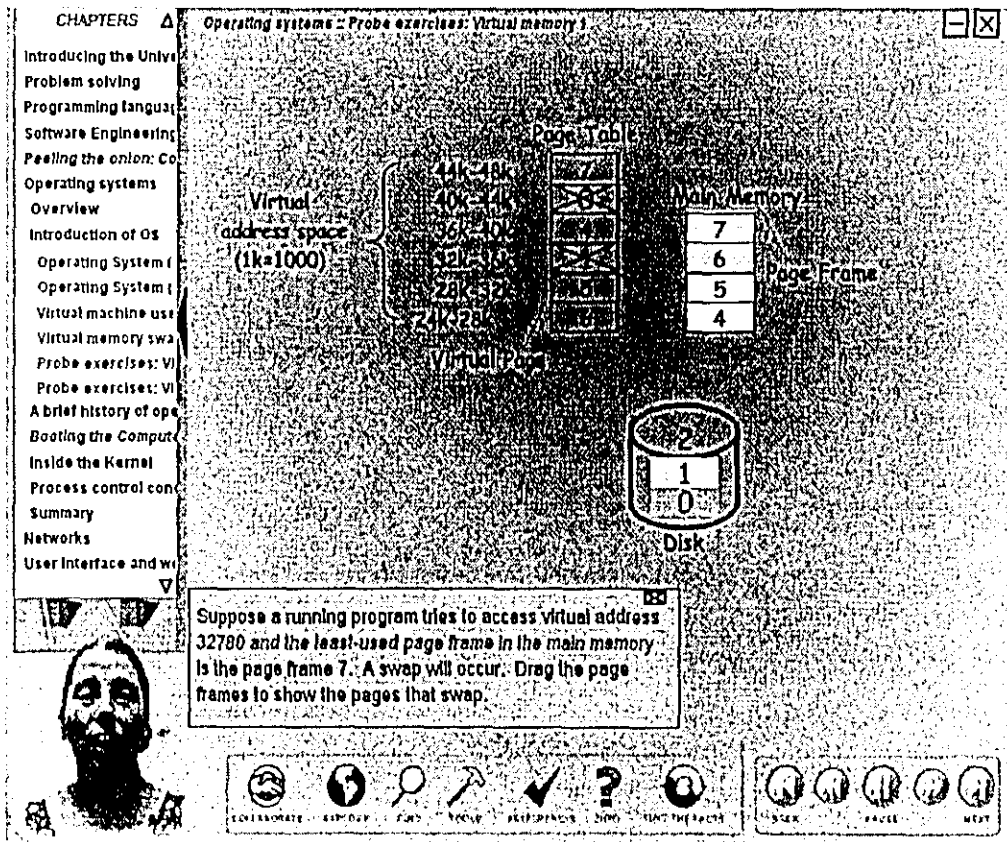


Figure 2: An example of a simple drag and drop exercise

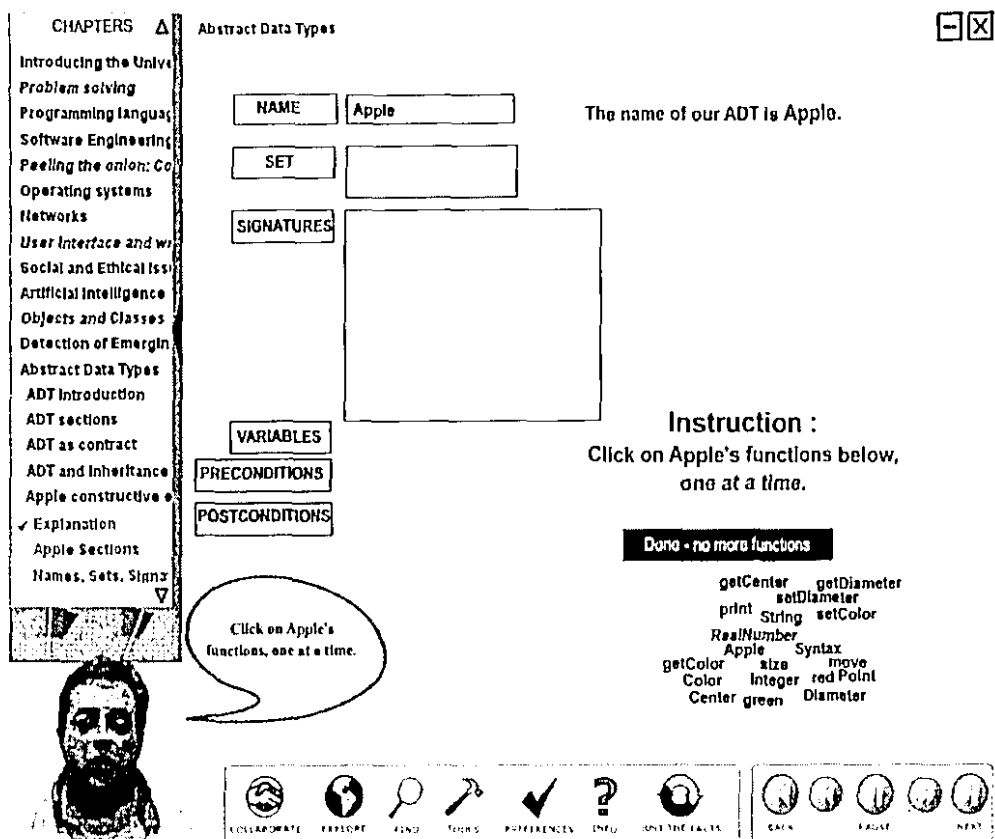


Figure 3: An example of a more involved constructive exercise

1.1 Database Tracking of exercises and quizzes:

As mentioned earlier, a central database keeps track of all the interactive exercises.

Each chapter consists of multiple exercises and quizzes. Figure 4 shows a snapshot of a quiz question with a feedback after a student selects an answer choice. Figure 5 shows the final score the student received on completion of the quiz. The format of these exercises can be multiple choice, true/false, or drag and drop. However, the quizzes have a format of multiple choice only. The exercises are primarily meant to facilitate active learning, whereas the quizzes are meant to assess the student's understanding on a topic. Note that the exercises give the user the flexibility to change their answers based on the feedback they get. The quizzes on the other hand, even though do give the user some form of

feedback, do not allow the users to change their answers or go back to a previous question.

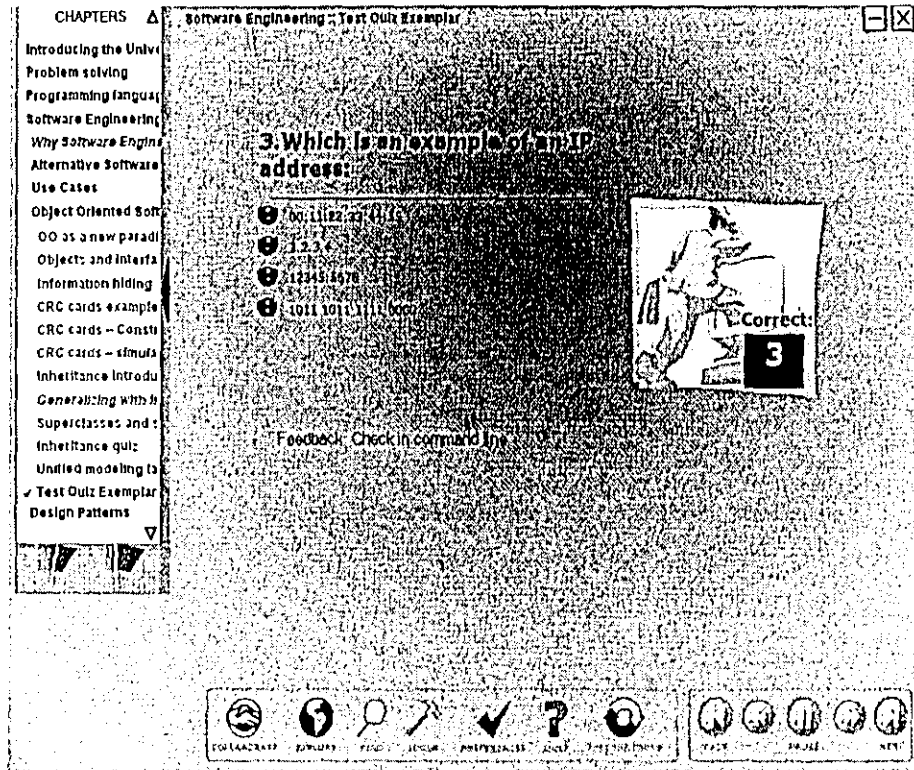


Figure 4: A snapshot of a quiz question with feedback

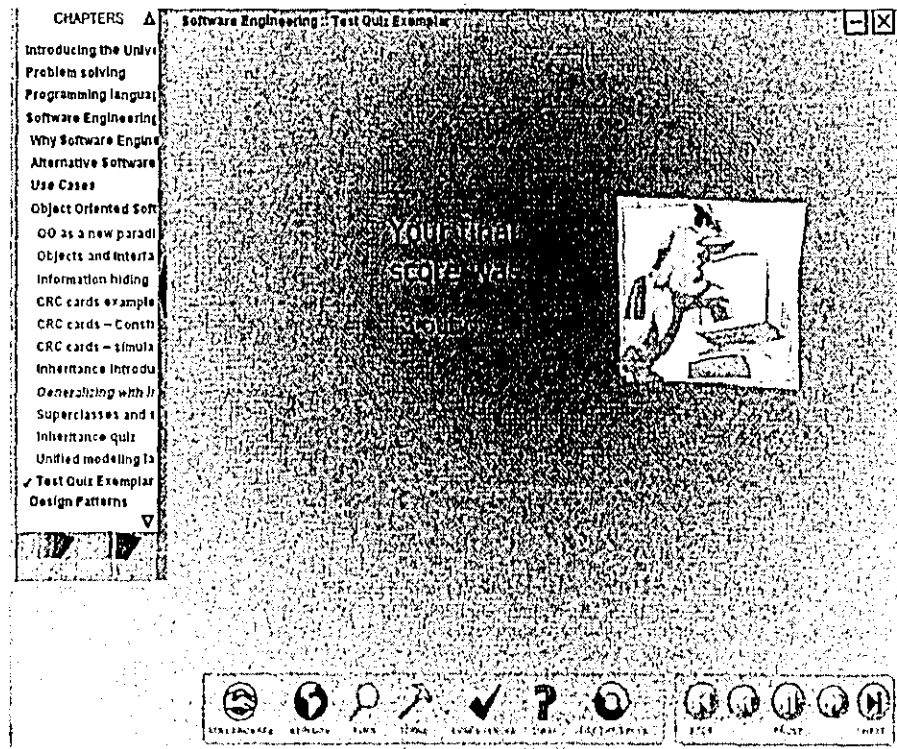


Figure 5: A snapshot of the final score a student received on a quiz

feedback, do not allow the users to change their answers or go back to a previous question.

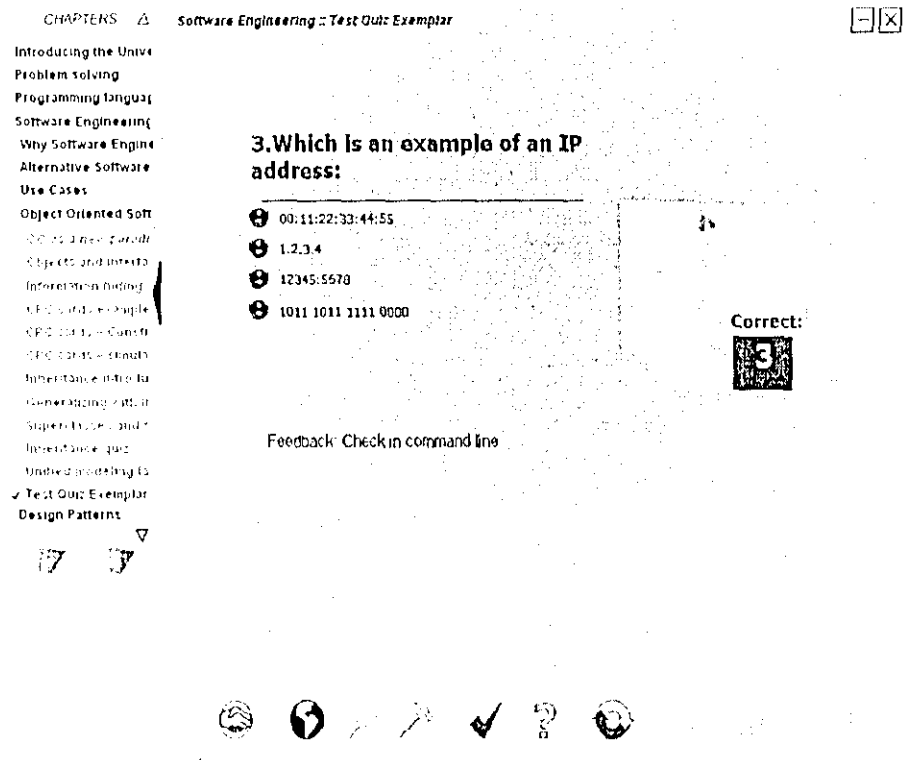


Figure 4: A snapshot of a quiz question with feedback

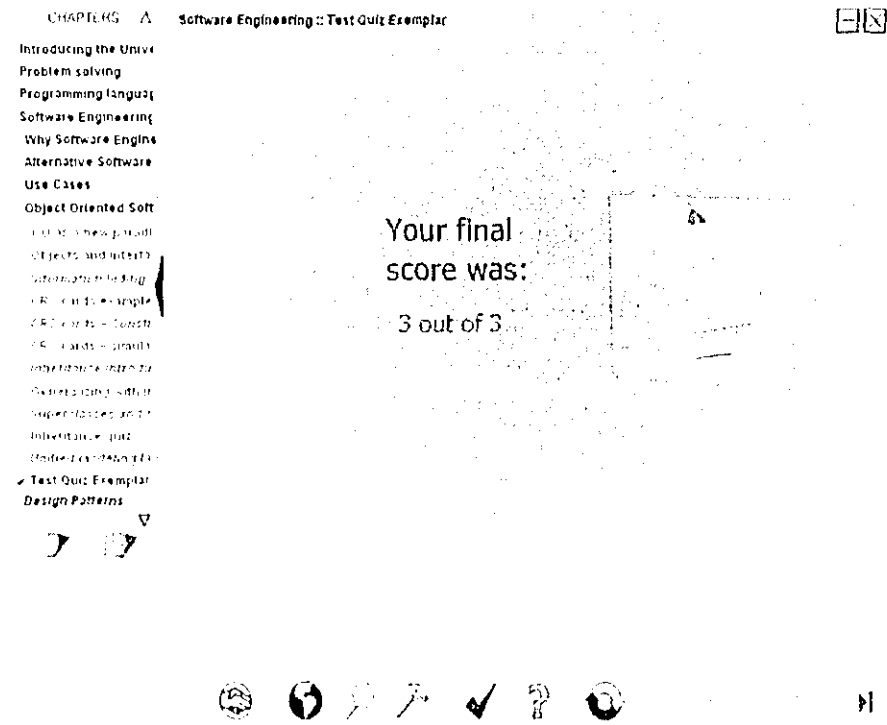


Figure 5: A snapshot of the final score a student received on a quiz

1.1.1 Data Organization in the MySQL database:

When a student works on an exercise or a quiz, the data is sent to the CIMEL database on a central server.⁴ The **exercise** table stores the exercise data and the **quizattempted** and **quizcompleted** tables store the quiz data.

The **exercise table** consists of the following fields: **userID**, **currScreen**, **questionNumber**, **questionLevel**, **correct**, **currScreenlabel**, **currChaplabel**, **timestamp**. Every student using the multimedia will be assigned a unique **userID** by a system administrator.⁵ These **userIDs** will be then stored in the database under another table called **users**. Every screen in the multimedia has a unique ID of type integer, hence when a user goes to an exercise screen the associated current screen ID is sent to the exercise table and stored in **currScreen**. Each exercise has a question number which is stored in the field called **questionNumber**. The **questionLevel** for each exercise is the number associated with each answer choice. So for example, for a multiple choice exercise having four answer choices (a-d), **questionLevel** will be a = 1, b = 2 and so on. Depending on whether a student chooses a correct or an incorrect answer, this information will be stored in the table as 0 or 1, where 0 infers incorrect and 1 means correct. The **currChaplabel** holds the name of the chapter, which has the exercise and the **currScreenlabel** holds the name of the particular exercise screen. This information is very important to keep track of which exercises in what chapters have the student attempted and whether they got it correct or not. **Timestamp** stores the time the student went to the particular exercise screen.

⁴ Note that there are 2 databases mentioned in this report, **cimel_logon** stores **userID** and password and the **cimel** database stores all the other information about exercises and the quizzes.

⁵ Look in the Appendix for more details on how a system administrator adds new users to the CIMEL system

The **quiz** information is stored in two tables called `quizattempted` and `quizcompleted`. `Quizattempted` keeps track of all the quiz questions that a student attempts and `quizcompleted` keeps track of whether the student finished a quiz. So if a student quits before completing a quiz, the `quizcompleted` table will not be updated. The `quizattempted` table has similar fields as the `exercise` table. The fields in the `quizattempted` table are: `userID`, `currScreen`, `questionNumber`, `correct`, `currChaplabel`, `currScreenlabel` and `timestamp`. Besides `userID`, `currScreen`, `currScreenlabel`, `currChaplabel` and `timestamp`, `quizcompleted` has two extra fields: `ScoreCorrect` and `totalNoQues`. `ScoreCorrect` consists of the number of questions the student got correct and the `totalNoQuestion` field holds the total number of questions in the quiz.

1.2 Quizzes in the multimedia:

A quiz template was recently created so that the instructors can use this as a template to add quizzes to any of the sections. There are two parts to the quiz exemplar. The quiz template was created in flash, but it uses xml data from "quiz.xml". The quiz template allows the instructor to add as many as twenty questions to the quiz. Also, the instructor can specify how many questions should be displayed to the student. Thus, the instructor can have twenty questions stored in the xml file, but can specify in the "quiz.xml" file that the system display only five questions at random to the student. The randomization of the questions allows different students to receive different sets of questions.⁶

1.3 Thesis Organization

The remainder of this report is organized as follows:

⁶ Further details on how to use the quiz exemplar can be found on:
<http://www.cse.lehigh.edu/~cimel/documentation.html>

- Section 2 is intended for the end user and is an overview of the different characteristics of the visualization tools
- Section 3 is intended for developers who want to extend or modify the tool; it describes the tool architecture and shows how new types of graphs (characteristics) can be added.
- Section 4 will be for the future developers on how to maintain this tool.
- Section 5 will be on related work on this topic.
- Section 6 concludes the work, by giving recommendations on how the tool could have been better and some suggestions on how to improve this tool.

2 CIMEL VISUALIZATION TOOL DESIGN

2.1 *Login Design*

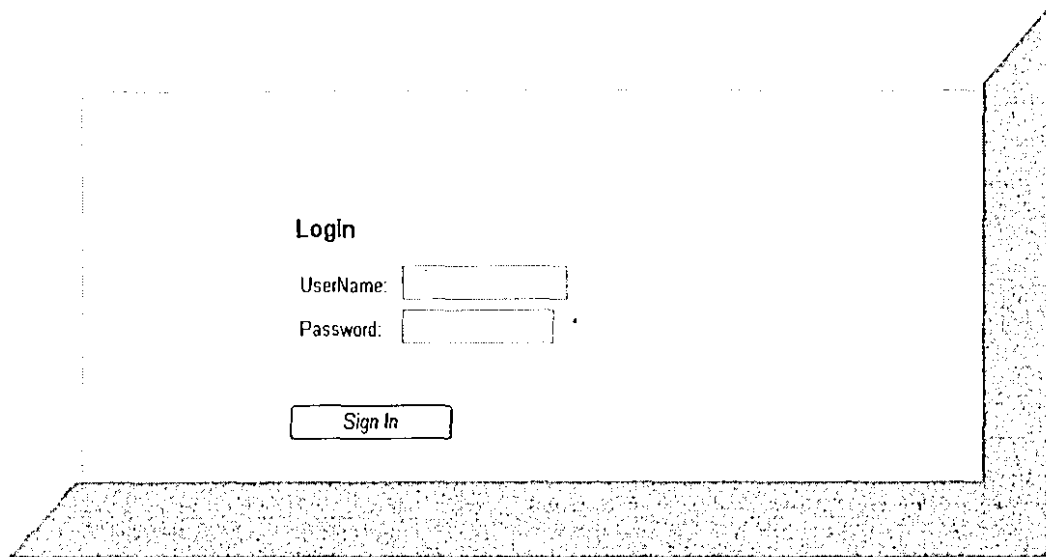
At present, the visualization tool has been set up so that only administrators and instructors can view it. A future extension might be to let students visualize their own progress using a subset of these tools.

The first thing that the users of this tool will see is the login page where they are asked to input their user name and password. A system administrator will be able to see all the courses in the database, however if the user is an instructor for a course then they will only be able to view the information relating to the courses they teach.

The username and passwords that are entered by the user are checked against the information stored in the users table in the cimel_logon database. The instructors that would like to use the CIMEL multimedia and this visualization tool will have to get a logon and password first. This password and logon information is then stored in the cimel_logon database. When a user inputs their username and password, this information is matched against the information stored in this database. Note that the cimel_logon table also stores the information whether a user is a student, instructor or an administrator. Hence, if the username and password match and it is determined that the user is an administrator or a professor, they are given authorization to view this tool.

Figure 6 is a snapshot of the login page.

Welcome to Cimel Visualization tool



Login

UserName:

Password:

Figure 6: A snapshot of CIMEL Visualization Tool Login Page

2.2 Tool Menus and Submenus

Once the user successfully logs in, an administrator can view all the courses in the database, but an instructor can only view the courses they are teaching. Once an instructor chooses the course s/he would like to view, he/she gets another drop down menu from which they can select from the following options: “view records for all the students in the class” and “view record by student.”

2.2.1 View record by student

When an instructor selects this option, a list box appears which consists of all the students in the class. The instructor can then select the student for whom they would like to see the exercise and quiz information. Once a student is selected two tables will appear, one consists of details about the student’s performance on the exercises and the other table consists of details about the student’s performance on the quizzes.

The exercise table holds the following details:

- The chapter names of each of the exercise the student has attempted

- The name of the exercise that the student has attempted
- The number of tries/attempts on each exercise before getting it either correct or giving up.
- The last column informs the instructor whether the student eventually got the question correct or not. Hence, if the number is zero in this column it means that the student did not get this question correct. If the number is greater than zero, it means that the student got this question that many times correct. Note, that usually for drag and drop exercises this number tends to be “greater than one”, as there is more than one correct answer.

Figure 7 shows a sample output for a student’s performance on the exercises.

EXERCISE

CurrentChapterLabel	CurrentScreenLabel	No. Of Tries	No Times Correct
Operating systems	Probe Question 1	2	1
Peeling the onion: Computer architecture	AND Gate output?	2	1
Peeling the onion: Computer architecture	Match the true pigs	2	1
Peeling the onion: Computer architecture	OR Gate output?	2	1
Problem solving	State representations	2	1
Problem solving	State representations-2	1	1
Problem solving	Well-defined versus ill-defined	11	4
Software Engineering	Quiz: Purpose of Design Patterns	5	1

Figure 7: Sample output for a student's performance on the exercises

The quiz table holds the following details:

- Similar to the exercise table, this table has information about which quizzes the student has attempted in what chapters.
- The total number of questions in the quiz.
- The final score/number of questions student got correct.

Figure 8 shows a sample output for a student’s performance on the quizzes.

QUIZ

CurrentChapterLabel	CurrentScreenLabel	totalNoQues	ScoreCorrect
Software Engineering	Test Quiz Exemplar	3	1
Introducing the Universal Computer	Quiz: History of Computers	2	0

Figure 8: Sample output for a student's performance on the quizzes

There is another button here called “more details.” When this button is clicked the instructor can view which questions on the quizzes a student got wrong/incorrect. The “more details” button is a toggle button. So when it is clicked it will change its text to “less details” and when clicked again, the table consisting of information on what questions the student got incorrect on a quiz will disappear.

Besides the above tables, there will be two graphs that will be a supplement to the tables. One graph will show the information about exercises and the other graph will display data for the quizzes. Figure 9 and 10 show these graphs.

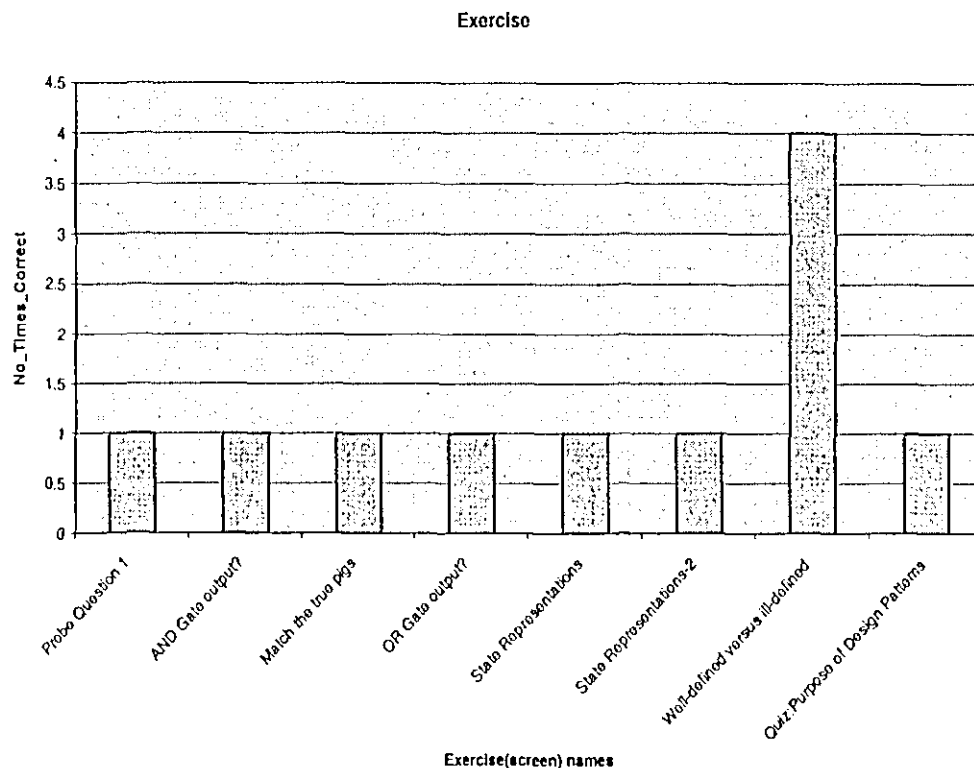


Figure 9: Graph showing the number of times a student got an exercise correct

Note that the reason for the current design of the graphs was for an instructor to see a trend on how a student is doing on each exercise/quiz compared to the entire class. So that if need be the professor might discuss the topic again in class. By plotting the number of times correct versus the exercise screen names (as in Figure 9), the instructor can easily visualize how a student did on an exercise, that is whether they got an exercise correct or not. Also, this graph tells the instructor what exercises the student has attempted.

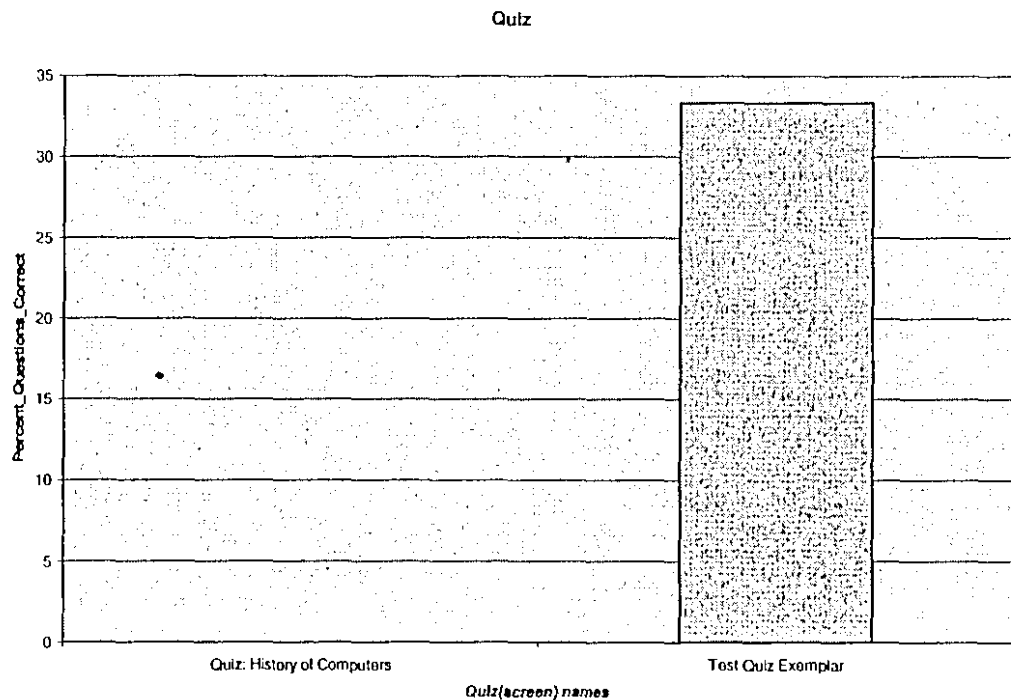


Figure 10: Graph showing the number of questions a student got correct in a quiz

2.2.2 View records for all the students in the class

There are two tables displayed here, one for the exercise and one for the quizzes. The purpose of the data in these tables is for the instructors to see a trend of how the class did as a whole on each of the exercises and the quizzes. Therefore, the following information is displayed in the exercise table:

- Similar to the previous tables, this table also has information about which exercises the student has attempted in what chapters.
- The total number of times this question was correct.
- The number of times the students in this course who got this question correct expressed as a percentage.

Figure 11 shows a sample output for a class's performance on the exercises.

EXERCISE

CurrentChapterLabel	CurrentScreenLabel	Total	Correct	Percent	Correct
Peeling the onion: Computer architecture	AND Gate output?	1		50	
Peeling the onion: Computer architecture	Match the true pigs	1		50	
Peeling the onion: Computer architecture	OR Gate output?	1		50	
Operating systems	Probe Question 1	1		50	
Software Engineering	Quiz: Purpose of Design Patterns	1		20	
Problem solving	State representations	1		50	
Problem solving	State representations-2	1		100	

Figure 11: Sample output for a class's performance on the exercises

The quiz table displays the following information:

- The information about which quizzes the student has attempted in what chapters.
- The number of questions correct in each quiz out of the total number of questions in a quiz expressed as a percentage.

Figure 12 shows a sample output for a class's performance on the quizzes.

QUIZ

CurrentChapterLabel	CurrentScreenLabel	Percent	Correct
Introducing the Universal Computer	Quiz: History of Computers	0	
Software Engineering	Test Quiz Exemplar	33.33	

Figure 12: Sample output for a class' performance on the quizzes

There will be two graphs represented here that will be supplements to figures 11 and 12.

One graph will show the information about exercises and the other graph will display data for the quizzes. Figures 13 and 14 show these graphs.

As mentioned above, the reason for this tool is for instructors to see the trends on how the class's/student's performance is on an exercise/quiz. By plotting percent questions right versus the screen names (as shown in figures 13 and 14), this will give the instructor an idea on what topics the students are struggling with.

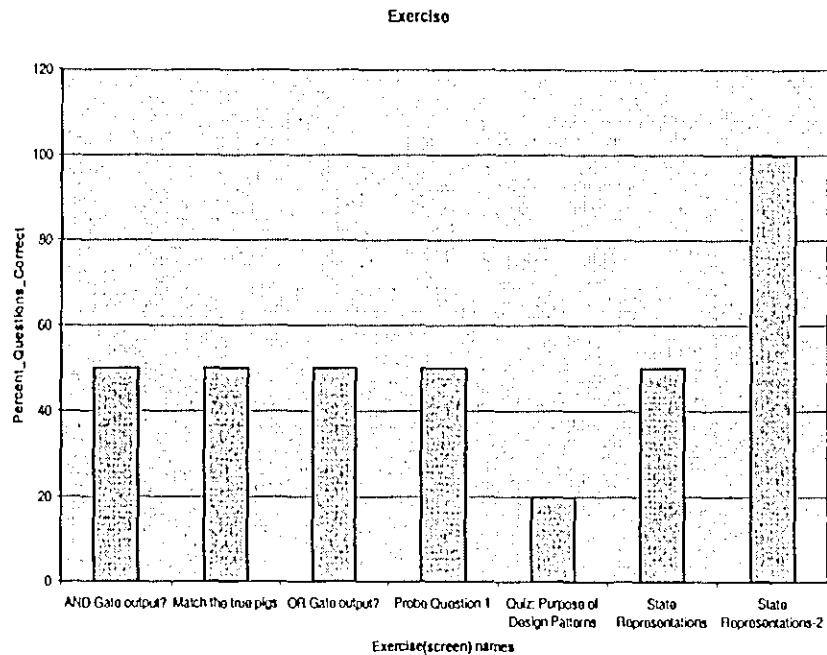


Figure 13: The percent of questions a class got correct in each exercise

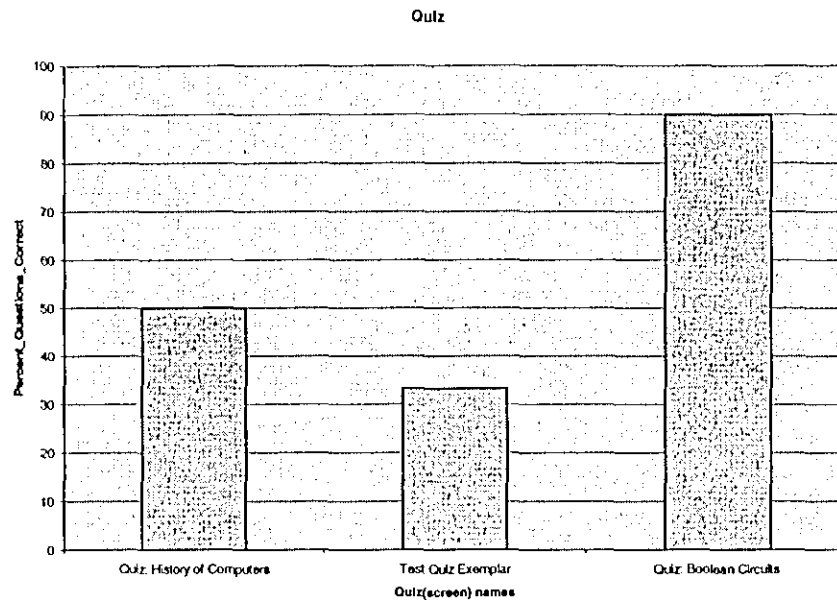


Figure 14: The percent of questions a class got correct in each quiz

While an instructor is viewing the data, it may be that some students could be working on the multimedia at the same time, and every time a student completes an exercise or a quiz, this new data is stored in the database. However, since the data in the tool is being generated dynamically from the database, the instructor will be able to view the most updated data every time they choose a menu selection. Also, these tables and graphs are dynamic, so when new data comes into the database, this new data is also displayed in the tables and graphs.

2.3 Approach to CIMEL User Interface Design

Above sections explain how the user interface has been designed but do not explain why the developer has chosen this design. One of the key features of the design is to have everything on one page. This is so that the instructor can find everything on one page and this makes it easy for the instructor to visualize and compare data. The other design feature was to have the graphics and the tables for any selection to be on one page. This

design decision was influenced by Clark and Mayer. Clark and Mayer in their book *e-Learning and the Science of Instruction* (2003) discuss the “contiguity principle.”

Contiguity principle states that printed words should be placed near corresponding graphics. Hence following this principle, the drop down menus are all in close proximity to the tables (an example of a simple graphic).

Extending this idea further, the tables are presented with fixed length and so as the new data comes in, the scroll bar will appear in the table. This was done so that when the tables become big, the instructor does not have to scroll down a long page to see the graphs. The thought was that instructors should be able to see the table data and the graphs side by side; hence this would make the visualization of the data easier for the instructor.

3 TOOL ARCHITECTURE

3.1 Overview

CIMEL visualization tool was created using Microsoft Visual Studio.NET 2003. What is .NET and why was .NET chosen to build this tool? The reason for choosing .NET to create this tool was that .NET is very suitable for what needed to be accomplished. The big advantage of .NET was that it did not have a very big learning curve and is fairly easy to use. Also, since this tool has been around for a while, a lot of help and documentation is available online. Finally, .NET unlike other similar tools, due to its multiple built-in features allowed for a relatively rapid development of this tool which was a very important requirement for the developer and the CIMEL team.

“a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of Web services—small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet”.⁷

Liberty in his book “Programming C#” explains that .NET platform as

“a new development framework that provides a fresh application programming interface(API) to the services and APIs of classic Windows operating system, while bringing together a number of disparate technologies that emerged from Microsoft in the late 1990s”

This tool has been created specifically with a Visual Studio Web application using C#, which is built around ASP.NET. ASP.NET is a platform which is used for developing and running applications on a Web server. ASP.NET is also part of the .NET Framework, which then allows it to access all of the features of this framework. For example, one can create ASP.NET Web applications using any .NET programming language (Visual Basic, C#, C++, and many others) and .NET debugging facilities. In web applications, one can

access data using ADO.NET. ASP.NET Web applications run on a Web server configured with Microsoft Internet Information Services (IIS). Note that Visual Studio creates the IIS applications when needed and provides ways for deploying the Web applications to IIS; hence one never has to work directly with IIS.⁸

Visual Studio.NET provides tools such as drag-and-drop controls that make application development much faster, easier, and more reliable. An advantage of web applications is that they can be accessed from any browser that can connect to the server. The tools for creating web applications in .NET are called Web forms. Hence with web forms, a web application is deployed to a web server and users interact with the application through a standard browser. With web forms, an HTML page with static content is created and then C# code is used to generate dynamic content. The C# code runs on the server, and the data produced is integrated with the static HTML to create a web page. Thus what is then sent to the browser is standard HTML.

Web forms divide the user interface into two parts, the visual part or user interface (UI) and the logic that lies behind it. The UI page is stored in file with the extension .aspx. The logic (code) for this page can be stored in a separate code-behind C# source file.

3.1.1 Creating a new web application using Visual Studio .NET

To begin work on a new web application, first open Visual Studio and choose New Project under the file Menu. In the New Project window, create a new C# ASP.NET Web Application and name it as letsstart, as shown in figure 15.

⁷ This definition has been taken from <http://www.microsoft.com/net/basics/whatis.asp>

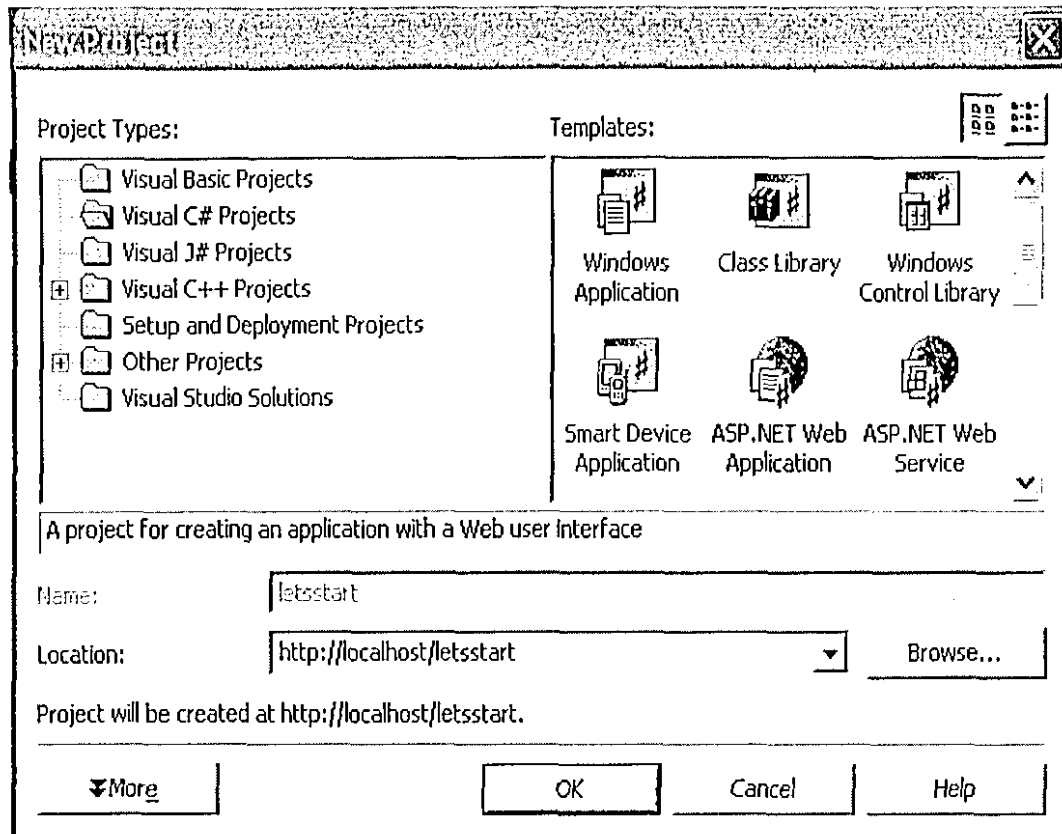


Figure 15: Creating a new ASP.NET web application

When a new web application is created, Visual Studio places a number of files in the project. The web form itself is stored in a file named `WebForm1.aspx`, which contains only HTML. A second, file called `WebForm1.aspx.cs`, stores C# associated with the form; this is the code-behind file.

3.1.2 Building and running an application

In order to build the code in the web form, go to the build menu and select “Build `letsstart`”. This will build the application. In order to run the application select “Start without debugging” under the Debug menu. Note that if the application has more than one web form and if someone wants to run any particular web form they will need to set the respective form as the “start page.” This can be done by first opening the

⁸ This material has been taken from MSDN website

solution explorer from the view menu and then right clicking on the appropriate web form and than choosing “Set as start page”.

3.1.3 Tool Box and Properties window

A toolbox window is available which has a selection of Windows widgets and controls. If the Toolbox is not displayed by default, it can be found under the view menu. The Toolbox is filled with controls that can be added to the web form. In the lower right corner, a properties window can be found, which displays all the properties of the currently selected item. The properties window can be used to set the static properties of the various controls. For example, to add text “Hello World” to a label, type the words into the box to the right of the Text property. Similarly, the font property can be used to change the font of the text “Hello World”.

Once, a form has been laid out, event handlers can be created for the different widgets by double-clicking the appropriate widget/control. This will lead to the code-behind page, which holds the source code for this form. Here the event-handling logic can be coded.

In every web form, the constructor calls a private method, *InitializeComponent()* which is used to define and set the properties of all the controls. The properties are set based on the default values or the values chosen by the programmer.

3.2 User Authentication

The first thing a user will see when they go to the CIMEL visualization tool website is a login page, where they will have to identify themselves by entering a valid user name and

a password. This user name and password are matched against the uid and password in the cimel_logon database. This concept is a simple one; however the implementation was not that easy. For security reasons the passwords in the users table of the cimel_logon database are encrypted using the MD5 algorithm. Once the MD5 algorithm is encrypted, it cannot be decrypted for security reasons. Therefore, the only way to compare the two passwords was to use hashing. Hashing an MD5 algorithm has been made very easy by the namespace System.Web.Security in the .NET framework class library. This namespace has a class called FormsAuthentication which contains a method `HashPasswordForStoringInConfigFile`, that does all the work of hashing a password the user enters and stores it as a string and hence the developers do not have to write lengthy code. This string is then compared to the password in the users table in cimel_logon database (note that the password in the users table is stored as string) and if they match the user is given access to the tool. Before a password is hashed, the program first checks for whether a username is valid or not. If the username is invalid then a message is displayed which says "not authorized". If the username is valid but password is invalid then a message "password invalid" is displayed.

Another thing that is checked for in this program is whether a user is an administrator or an instructor. If either is untrue then the user is denied access. If the user is an administrator then the user will be given access to all the courses stored in the database. If the user is an instructor (in the users table of cimel_logon database this information is stored under groups field and the instructor is "Prof" and an administrator is "admin"), then the instructor will only be able to see the courses they are teaching. In the program, this is done with "select" statements.

3.2.1 Redirecting to a different web page

One of the advantages of creating web applications using .NET is that it is very easy to redirect one's web browser to a new URL. Hence in this application, the programmer chose to create different web pages and then connected them using the "response.redirect" method in system.web namespace. In the visualization tool, the web page <http://128.180.123.160/letsstart/WebForm2.aspx> will be redirected to <http://128.180.123.160/letsstart/WebForm3.aspx> if the user is given authentication.

3.2.2 Session Object

In order to remember a value when switching between the different web pages, session object (an ASP built-in object) is very useful. Session objects are used to store information needed for a particular user session. Variables stored in the Session object are not discarded when the user jumps between pages in the application and persist for the entire user session. For purposes of this application, there were two values that needed to be remembered. One is the user name and second is what group does the user belong to, that is whether a user is an administrator or an instructor. The user name is used to display the message "Welcome username!". The group information is used for displaying the different courses for the different type of users.

A session object can be used as follows:

- First the information needs to be stored. This needs to be done on the web page where the information is. This can be done by:

```
Session["userName"] = UserName.Text;
```

- Next, this information can be called on another web page as follows:

```
userName = (string) Session["userName"];
```


3.2.3 Postback events

Postback events cause the web form to be posted back to the server immediately.

Example of these events are click-type events such as the Button Click event. There are many events that are non-postback events and therefore are not posted back to the server immediately; rather they are cached by the control until the next time a postback event occurs. However, non-postback controls can be forced to behave in a postback manner by setting their `AutoPostBack` property to `true`⁹. In the tool, most of the dropdown menus and the list box as well as the data grid control's `AutoPostBack` property were set to `true`.

3.2.4 Page_Load event

This is the most important initial event, which is fired every time a web form is loaded. Every web form page has the property `IsPostBack`, which will be `true` if the page is being loaded in response to a client postback, and `false` if it is being loaded for the first time. Hence, if the value of `IsPostBack` equals `false`, the page is being displayed for the first time and therefore the values need to be retrieved from the database.

In this tool, when `webform3` is first loaded, the `dropdownlist1` needed to be filled with values from the database and therefore the code to handle `dropdownlist1` was added in the `Page_Load` method. Note, that the values in `dropdownlist1` should be loaded only the first time the page is loaded. If the user clicks on a value or takes another action that sends the page back to the server, the values in the `dropdownlist1` should

⁹ This can be done in the property menu of the respective control

not be retrieved again when the page is reloaded. The code to check for when a page is loaded for the first time is as follows:

```
private void Page_Load(object sender, System.EventArgs e)
{
    if(!IsPostBack)
    {
        //..
    }
}
```

3.3 Database Connection and ADO.NET

The .NET framework provides a rich set of objects to manage database interaction and these classes are collectively referred to as ADO.NET (Liberty). Data can be retrieved from the database using ADO.NET. ADO.NET consists of a set of objects that interact to provide the required functionality. Figure 16 shows a simplified view of the primary objects in the ADO.NET object model.

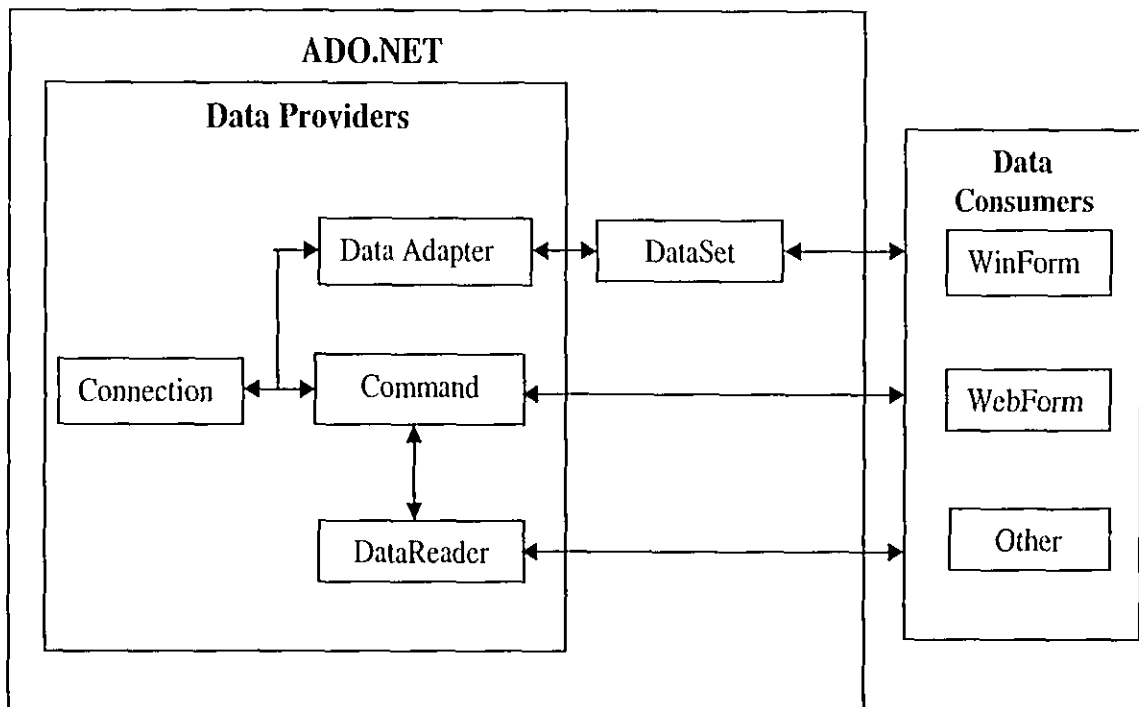


Figure 16: ADO.NET object model

There are two components in ADO.NET classes. These are Data Providers and the data set. The data set represents the actual data and the data providers handle the communication with a physical data store. Either of these components can communicate with data consumers such as Web forms.

There are three data providers that are included in the Visual studio .NET: Sql Server provider objects (these begin with SQL), OLE DB objects (these begin with OLEDB) and Oracle objects (these begin with Oracle). The database used for this tool is MySQL and the data provider for MySQL is not included in Visual studio .NET. There are two data providers that can be used here: either the MySQL data provider or the ODBC data provider. For the tool, the ODBC data provider was used¹⁰. This is because, it was easily found on the web and there was no difficulty in establishing the connection with the CIMEL database¹¹. After the data provider has been downloaded, to add them to Visual studio .NET, just right click “data” in the tool box and choose “Add/Remove Items” and choose the necessary .NET framework components. For creation of this tool, OdbcDataAdapter, OdbcCommand and OdbcConnection were added. The Connection, DataAdapter, Command, DataReader and DataSet objects are important and they are each discussed below.

3.3.1 DBConnection Object

The Connection object represents the physical connection to a data source. Its properties determine the data provider, the data source and database to which it will connect, and

¹⁰ This data provider can be downloaded from the msdn Microsoft website

the string to be used during connecting called the *connection string*. One can open and close the connection, change the database, and manage transactions with this object.

3.3.2 DBCommand Object

The Command object is typically a SQL statement or a stored procedure to be executed at the data source. Command objects can be created and executed independently and are used by the DataAdapter object to handle communications from a DataSet back to a data source. Command objects support SQL statements and stored procedures that return single values, one or more set of rows, or no values at all.

3.3.3 DataReader Object

A DataReader is a fast, low-overhead object for obtaining a forward-only, read-only stream of data from a data source. They cannot be created directly in the code, rather can only be created by calling the *ExecuteReader* method of the Command object.

3.3.4 DataAdapter Object

The DataAdapter object is used by ADO.NET to mediate between the DataSet object and the database. The DataAdapter contains four Command objects. They are: the *SelectCommand*, *UpdateCommand*, *InsertCommand* and *DeleteCommand*. The DataAdapter uses *SelectCommand* to fill a DataSet and uses the remaining three commands to send the changes back to the data source.

¹¹ The developer tried to use a MySQL adapter downloaded from the ByteFX site which provides the ByteFX MySQL driver, however, the developer had problems connecting to the database with this driver.

3.3.5 DataSet

The *DataSet* represents a subset of the entire database, cached on one's machine without a continuous connection to the database. Hence a *DataSet* consists of tables and their relations. The *DataSet* is always disconnected from the database and therefore it can contain data from multiple sources.¹²

3.3.6 Data Binding

Web forms allow binding of controls to the data so that as the data is modified, the controls respond automatically.

3.3.7 Connecting to the CIMEL database

To connect to the *CIMEL* database, in the toolbar in the web form, choose *OdbcConnection* under "Data" and drag it onto the web form (.aspx extension). In the properties window, click on the *ConnectionString*. This will open a dialog box, where the host/server name (or IP), database name, username and password need to be specified. To make sure that the connection has been established successfully, click the "test data source" button.

Figure 17 shows the dialog box¹³.

¹² The material on different types of objects has been taken from "Microsoft ADO.NET Step by Step" 2002, chapter 1, pages 3-19

¹³ Note that this dialog box may look different for different database language drivers like SQL Server and Oracle.

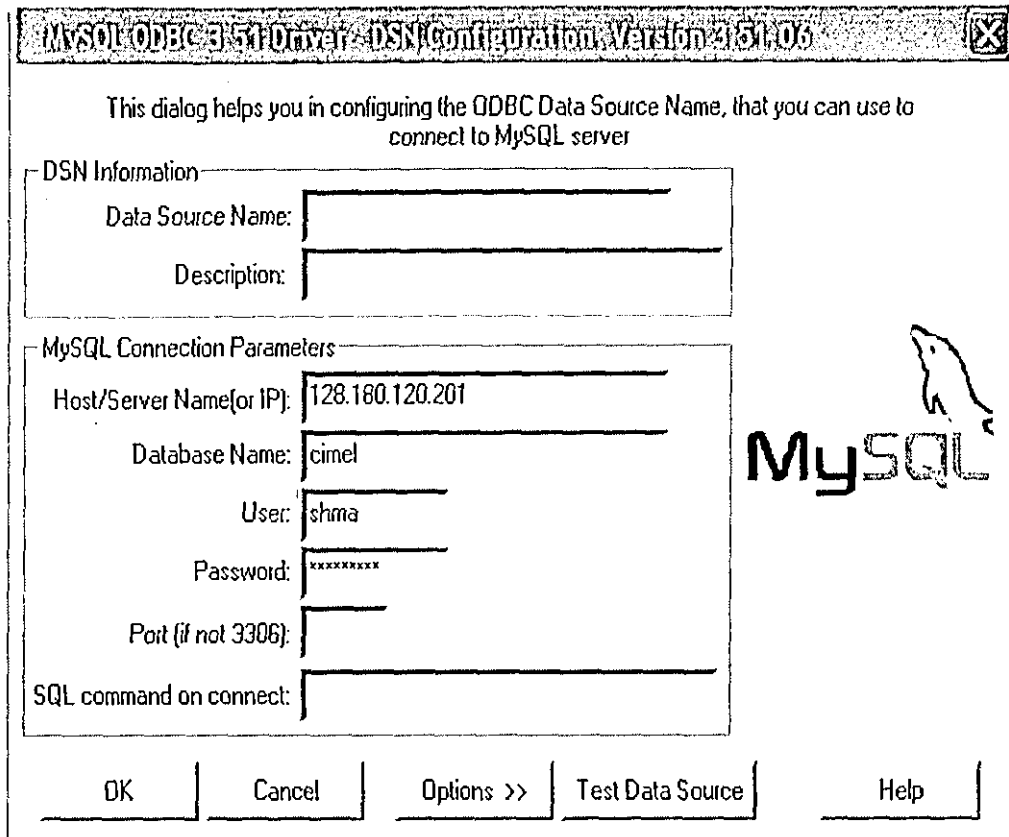


Figure 17: The dialog box to add a connection string

3.4 ADO.NET

The following is a list of some of the basic code used in the program:

- Create a DataAdapter and pass in the selection and connection string

```
OdbcDataAdapter da = new OdbcDataAdapter ("select currChaplabel from
quizattempted",odbcConnection1);
```

- Create a new DataSet object:

```
DataSet ds = new DataSet();
```

- Fill the dataset from the quizattempted table using the DataAdapter:

```
da.Fill(ds);
```

- Fill a data grid control using the dataset

```
DataGrid.DataSource = ds;
DataGrid.DataBind();
```

- For a list box, a datacommand object and datareader objects are needed and the code is as follows:

```
OdbcCommand cmd = new OdbcCommand("select userID from  
exercise",odbcConnection1);  
odbcConnection1.Open();  
OdbcDataReader reader = cmd.ExecuteReader();  
  
ListBox1.DataTextField = "userID";  
ListBox1.DataValueField = "userID";  
  
ListBox1.DataSource = reader;  
ListBox1.DataBind();  
reader.Close();  
odbcConnection1.Close();
```

Note that when a connection is explicitly opened, it needs to be explicitly closed.

3.5 SQL in Table Creation

Some of the SQL statements are very simple and therefore they are not mentioned here.

A few SQL statements are a little complex as they deal with joins. They are as follows:

- For putting the students that are in the course currently selected by the instructor into the listbox from the database, the following SQL statement was used:

```
Select distinct exercise.userID from exercise, course_students where  
exercise.userID = course_students.userID and course_students.courseID  
=""+course+""
```

Here, the two tables that are being joined are exercise, which holds the exercises a student has done and the table course_students which holds the data about which students belong to which course. In order to know which course has been chosen, the string course holds the value that the instructor has chosen.

- In order to know how all the students did in an exercise, that is to get the total number of questions correct and the percent correct for the entire class, the following SQL statement was used:

```
Select exercise.currChaplabel as CurrentChapterLabel,
exercise.currScreenlabel as CurrentScreenLabel, sum (exercise.correct) as
Total_Correct,
sum(exercise.correct)/count(exercise.questionNumber)*100 as
Percent_Correct from exercise, course_students where
course_students.courseID = "+course+"and exercise.userID =
course_students.userID group by exercise.currScreenlabel
```

Once again a join is being performed between the exercise and the course_students table. Here the aggregate function sum is being used to sum all the correct responses pertaining to the exercise and the aggregate function count is used to count the number of times this question was attempted by all the students in the class. The “group by” will group all the exercises with the same screen name together. Note, since the quiz table has very similar SQL statement, it is not mentioned here.

- The following SQL statement was used to find how a student in any course did on the different exercises.

```
Select exercise.currChaplabel as CurrentChapterLabel,
exercise.currScreenlabel as CurrentScreenLabel, count
(exercise.currScreenlabel)as No_Of_Tries, sum(exercise.correct)as
No_Times_Correct from exercise, course_students where exercise.userID
= "+checking+" and course_students.userID ="+checking+" group by
exercise.userID, exercise.currChaplabel, exercise.currScreenlabel
```

Once again here the tables exercise and course_students are joined to get the data for only the students that are in the course selected by the instructor. Here, “checking” is a string that holds the value of the student that an instructor has

selected to view the data for. The aggregate function count is being used to count the number of times a student attempted an exercise and the aggregate function sum is used to count the number of times an exercise was correct. So if this value is greater than zero, the instructor knows that a student eventually got an exercise question correct.

3.6 Graphs

This part of the tool has not been implemented yet and therefore currently there is no documentation available on this section. After the tool is completed this part of the documentation will be added and will be found at the following website:

<http://www.cse.lehigh.edu/~cimel/documentation.html> under the publications section.

4 TOOL MAINTENANCE

The developers will not need to maintain the tool itself as the tool is currently implemented; however there are a couple of tables in the CIMEL database that will need to be maintained. The two tables that the developer has to maintain are: *course_instructor* and *course_students*. The *course_instructor* table holds the fields *courseID* and *userID*. The *courseID* in the *course_instructor* table holds the course name and the *userID* holds the ID of the professor. The *course_students* table also has the fields called *courseID* and *userID*. In this table, the *courseID* once again refers to the course name. The *userID* in this table refers to the students. Hence the *course_students* table holds the data about which student belongs to what course and the *course_instructor* table holds the data about which professor teaches what course.

However, as mentioned above, there might be future versions of this tool where the students are allowed to see their own data or the future developers might also be interested in changing the way the data is currently being presented to the instructors. If this is the case, then the developers will be developing newer versions of the tool.¹⁴

¹⁴ The developers should take a look at the recommendations section to learn about how the tool can be improved

5 RELATED WORK

Blackboard has a facility to create quizzes.¹⁵ There are options to create multiple choice, true/false, matching, ordering, fill in the blank as well as short essay questions.

Blackboards, Online grade books (Schmidt 2002) and PageOut all allow basic evaluation of assignments and quizzes. For example, statistics for assignments such as the curve, mean and standard deviation can easily be calculated and made available to the instructor. One disadvantage of this type of evaluation technique and presentation of results is that it requires a lot of input from the instructor in terms of setting up quizzes and assignments. Secondly, unlike CIMEL visualization tool, the instructor can only receive basic statistics on the student's performance and do not get any graphical representation. One of the advantages of CIMEL is that it requires little user input; aside from selecting what to view, the user does not have to do anything else. The data is automatically generated for the use.

There are more sophisticated tools like WebCT (Schmidt) that allow instructors to track student progress. This tool has been mainly used for online taught courses. This tool can be used to check when and how often a student has accessed a course website. With WebCT questions can be developed and added to an online database, which then allows the instructor to create new quizzes by just extracting the questions from the database. There is another tool called Mallard¹⁶ that is similar to WebCT and provides excellent assessment and quiz tools. Mallard also enables instructors to observe a student progress

¹⁵ This material was taken from "Getting Started with Blackboard" An Introductory training on Blackboard for University of Northumbria staff, 2001

¹⁶ Details about Mallard can be found at:

http://wocket.csl.uiuc.edu/~ktechow/Mallard/Gradebook/Mallard_quiz_grdbook.html

even before the due date. Both *Mallard* and *WebCT* allow quizzes to be randomly generated. This is similar to the quizzes being generated in *CIMEL*. Hence, different students can be given different quizzes as well as the same student can see a different quiz each time s/he retakes it. Another aspect of *WebCT* quizzes that is very similar to *CIMEL* quizzes is that the feedback is provided immediately as well as what grade the student got on the quiz. There is one thing that is currently in *Mallard* that is not in *CIMEL* is that there is a student view to view quiz grades as well as an instructor view for viewing the quiz grade book. Also *Mallard* has an interesting feature that when a student does not hand in a quiz at the time it was due, the grade is displayed in red to let the instructor know that the student handed the quiz in late. Another interesting feature of *Mallard* is that the grade policy for exercises (not the quizzes) is that a student can answer parts of questions on separate days and the grade records will show the time at which the student received his/her highest score on each of these parts. One another thing that *CIMEL* does not have is tracking how long the student took on completing each quiz. *Mallard* also keeps track of how many times a student has retaken a quiz. The "grade boxes" displaying students' quiz grades are shaded progressively darker as the number of quiz submissions increases. Hence, a dark column in the gradebook indicates many students are retaking this quiz many times: perhaps a hint or more class discussion is needed. If a row in the gradebook is dark, then a particular student is likely having difficulty: perhaps a one-on-one meeting would be helpful. This feature is particularly useful and future developers may want to implement this feature in *CIMEL*.

There are many other tools like JExam, TestPilot, QMark¹⁷, that are assessment software tools that will allow instructors to create quizzes and get student performance statistics.

The one thing that is unique about the CIMEL visualization tool, is that the instructor can also see graphical representation of data. Moreover, the graphs and tables are dynamically generated, thus every time the instructor will use this tool, s/he will view the updated information from the database.

¹⁷ Details about these tools can be found at <http://www.chem1.com/chemed/program.html#TEST>

6 RECOMMENDATIONS AND CONCLUSIONS

CIMEL Visualization tool was designed for instructors to evaluate how the students are doing on exercises and quizzes. This report discusses the tool design, the tool architecture, and instructions for developers on how to maintain the tool as well as highlights other work done in this area.

The developer feels that the following improvements could be made to the current version of the tool:

- Due to the short time span that the tool was developed aesthetically, the tool is not as appealing and can be improved. For example, there could be an image icon of CIMEL on the page. Also, future users might want to add more color to the web page. Another improvement could be to make more interesting buttons than the ones that are currently implemented.
- Statistics such as curve, mean and standard deviation data about exercises and quizzes is not currently implemented. This can be added to the next version of the tool.
- The graphs are made in two dimensions using only two sets of information, for example, for the exercises instead of just plotting the No_times_correct versus exercise screen names. It would also have been nice if the graph can hold the number of times a student attempted an exercise. To build graphs in this manner using .NET is very difficult to do. One of the reasons for this is that the graphs are being created dynamically and at run time and so to build better looking graphs is

very difficult. That is why the developer felt that some other tool rather than .NET might have been better to use for the creation of the graphs.

- One of the problems the developer faced when trying to implement this tool, is the poor database schema and design of the tables in the CIMEL database. Since MySQL does not have a very powerful query engine, the developer was not able to write complex queries. This of course affected the user interface design of the tool. There was some information like how long a student took to do an exercise or a quiz could not be traced. Also, due to bad database design of the tables, some queries ended up being more complex than needed. Hence, a note to future developers, poor database design can be very detrimental and lead to a user interface that is not very good.
- A future implementation of the tool might want to track how many times a student ends up taking a quiz. This is particularly useful for the instructor. This is because if many students are retaking a quiz many times, this would be a hint for the instructor to have more class discussion on this topic.
- A future version of this tool might be to add a component where the students can view their grades on each of the quiz and exercise.
- Later, a more interactive CIMEL visualization tool can be made where an instructor can click on a student's login name to send e-mail.

In conclusion, since this is the first version of the tool, there are many possible improvements as well as extensions that can be made to this tool and hopefully this report will be a good help/guide for the future developers.

7 REFERENCES

Blank D. G, Roy S, Sahasrabudhe, S, Pottenger M. W, Kessler G. D Adapting multimedia for diverse student learning styles, J. Comput. Small Coll., 18, 3, 2003, 45-58, The Consortium for Computing in Small Colleges.

Blank D. G, Pottenger M. W, Kessler G. D Kay J. E, Teaching Java and Abstract Data Type Semantics with Multimedia. Computer Science and Engineering Department, Lehigh University. Available online at http://www.cse.lehigh.edu/~cimel/papers/SIGCSE2002_abstract.htm

Details about JExam, TestPilot, QMark:
<http://www.chem1.com/chemed/program.html#TEST>

Details about Mallard:
http://wocket.csl.uiuc.edu/~ktechtow/Mallard/Gradebook/Mallard_quiz_grdbook.html

Getting Started with Blackboard. Introductory training on Blackboard for University of Northumbria staff, version 1.3, September 2001. Available online at http://online.northumbria.ac.uk/LTA/media/docs/teaching_handbook_Sep01.doc

Liberty, J. (2003) Programming C#. (3rd ed.). Sebastopol, CA: O'Reilly & Associates, 689 pages.

MSDN website:
<http://msdn.microsoft.com/library/?url=/nhp/default.asp?contentid=28000440>

Riordan, M. R. (2002) Microsoft ADO.NET Step by Step. Washington:Microsoft Press

Ruth Colvin Clark O and Richard E. Mayer, e-Learning and the Science of Instruction, Jossey-Bass/Pfeiffer 2003

Schmidt, K. The Web-Enhanced classroom. Journal of Industrial Technology 18,2, 2002. Available online at <http://www.nait.org>

8 APPENDIX

8.1 *How to generate new users for CIMEL*

Following these steps, a system administrator can add users to CIMEL:

1. Log in to CIMEL using your username and password (faw2/2faw)
2. Hit the 'Minimise' button (the [-])
3. In the just fact window go to the following URL:

http://cimel.cse.lehigh.edu/scripts/login_secure_user_admin.pl

On this page the administrator can add, edit and remove users.

9 VITA

Sharmeen Mecklai was born in Karachi, Pakistan on May 28, 1977 to Abdul Aziz and Gulaby. She graduated from University of Toronto, Toronto, Canada, in 2000, with a Bachelor of Science in Human Biology. She then went to Cedar Crest College, Allentown, Pennsylvania, to pursue her Bachelors in Computer Science and graduated in 2002 with *Great Distinction*.

**END OF
TITLE**