Accreditation Units Calculator:
An Internet Application

Wu Mei Zhan

A Major Report
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

April 2003

0-612-77727-8

ABSTRACT

Accreditation Units Calculator: An Internet Application

Wu Mei Zhan

The project aims to develop an Internet application that provides the Software Engineering students with the information about Software Engineering courses and enables them to calculate the Accreditation Units for a single course or for a group of courses. The purposes and requirements of the project are described briefly in the first two chapters of this report.

Due to the nature of the application. different computer technologies are involved. The emphasis has been placed on research and learning the technologies of Web-based application with database accessibility. Chapter 3 has a review of the technologies used as well as the contribution of those technologies to the project.

Chapter 4 is a description about the design and implementation of the project, including the front-end interface as well as back-end database and Java Servlet program design and implementation.

Chapter 5 concluded the project and point out potential further work of the project. The referenced Web sites and books have been listed in the last chapter.

ACKNOWLEDGMENTS

I wish to express my deep gratitude to my supervisor. Dr. Peter Grogono, for his invaluable guidance and suggestions throughout this work.

I would also like to thank my family for their understanding and support throughout my education career in the Department of Computer Science of Concordia University.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

In today's fast-moving, competitive, global business environment, the Internet is growing at an astounding rate, and is becoming more and more important as an information medium. According to the survey result of Network Wizards (http://www.nw.com), there are over 162 million Internet hosts until July 2002, almost treble the amount of it at three years ago. Organizations see the Internet and the Web as crucial to their information systems strategies. Perhaps the most important trend is that the Internet is becoming a much more dynamic place; much more of the information available on the Internet is generated live. According to user's request, database are searched, data are counted, dynamically.

The purpose of my project is to:

- Provide a useful tool, the Accreditation Units Calculator (AU Calculator), for the undergraduate students of Software Engineering Department of Concordia University to dynamically check and calculate the Accreditation Units for a single course or a group of courses through the Web.
- Gain experience with Internet application programming through the design and implementation of the AU Calculator project.

2. Requirements of the AU Calculator

The intended users of the AU Calculator are the Software Engineering students of Concordia University.

The AU Calculator will provide the intended user with a faster, easier and convenience way of checking AU (Accreditation Units) information for a single course or for a group of courses, and calculating the total AUs they have accomplished and/or the total AUs they have to gain.

The system should respect the following basic requirements:

- Provide choices for all available option groups and academic years;
- Provide AU information for all of the courses available for an option group in an academic year;
- Allow users to choose a single or a group of courses from the course list available into their selected course list;
- Allow users to remove a course from the selected courses list;
- Show the total AUs for all courses the student selected;
- Show the total remaining AUs the students has to do;
- Allow the user to view and print the detailed AU information for all courses chosen;

- Allow the system administrator to add new database files without recompile the application;
- The most interaction with user should be done within a single interface;

Since the project is an Internet application. the interfaces of it are all Web pages.

- The first page allows the students to choose the academic year. and the option group (group 1 or 2) they enrolled in.
- The second page is the main interaction page of the project. which not only provides information of AUs for the courses of the selected option group and academic year. but also provides the functionality of calculator for total AUs of selection.
- When the user finished with the second page. with a simple click on a button at the bottom. he/she can go to the third page. which was produced live from the second page. and shows the detailed AU information for all of the courses selected in the previous interface. User can get a hard copy of the results by printing it with the embedded print function of the Web browser.

3. Tools to be Used by AU Calculator Project

To build the project, developer has to choose a suitable set of tools, develop and integrate a set of programs and applications. During this stage, the following requirements were considered:

- As an Internet application, the project should meet the needs of its intended user, allow them to interact with data via the Web;
- Aiming for Web features that are efficient to operate, elegant to use and easy to maintain;
- The interface should be helpful, that is, clear and intuitive;

Following sections are descriptions for the tools chosen to implement AU Calculator:

3.1 HTML

3.1.1 Purpose of HTML

HTML stands for the **H**yper**T**ext **M**arkup **L**anguage. *Hypertext* refers to text containing connections to other documents that contain information about related topics. By clicking a link, readers can get to another hypertext document, which in turn offers links to additional documents. and so on.

HTML is the major language of the Internet's World Wide Web. Web sites and web pages are written in HTML. Web pages written in HTML can bring text, pictures, sounds, and links... all together in one place.

HTML files are usually names with the extension .html or .htm. They are in fact standard ASCII files with formatting codes; commonly called *tags*. that contain information about layout, text styles. document titles, paragraphs, lists. and hypertext links etc.

HTML provides primitive user interface elements (also known as *controls* or *widgets*), such as push buttons. radio buttons, checkboxes. lists. text input fields. formatted text. images. hypertext links. and client-side image maps. and so on. HTML table gives HTML the ability to arrange the layout of those primitive UI elements on a Web page.

3.1.2 Advantages of Using HTML

- Browser independent

HTML is a browser independent language. Documents written in HTML can be interpret and displayed by any Web browser. regardless what kind of Web editor was used.

- Composition independent of platforms

HTML files are plain text files, so they can be composed and edited on any type of computer...Windows, Mac, UNIX, whatever.

- Easy to be mastered

HTML language itself is easy to master. The time spend on learning HTML is a small fraction of learning other languages. In addition, there exist some easy to use HTML editors that even the most non-experienced user can create HTML documents with no difficulties at all.

3.1.3 Limitations of HTML

Although HTML provides the building blocks of user interfaces and allows them to be laid out in pleasing ways within a Web page, In fact, it is not enough to produce a fully functional user interface with only HTML itself.

Two very important features of user interfaces are not provided by HTML.

- First, the ability to perform actions in response to user events is the most crucial feature of any kind of interactive user interface. Unfortunately, HTML did not provide enough event handlers to respond to the users interactions. It provides only two possible actions:
  - Reset form elements to their default state.

  - Submit the values of form elements to a CGI script.

  These actions are further limited by the fact that they must be triggered by special-purpose elements in the user interface: forms can be reset only through ***Reset*** buttons and can be submitted only through ***Submit*** buttons.

- In addition to the lack of event handling, HTML user interfaces also suffer from poor response time. It would be possible, for example, to create an HTML form that contains a number of different ***Submit*** buttons, each of which had some non-default label other than "Submit." Whenever the user clicked one of these buttons, the Web browser would contact the Web server, which could send a new HTML form that represented the updated state of the user interface. However, even with a very fast network connection between browser and server, the delay required for this kind of update would make the interface awkward to use. Over a slower connection the interface would probably not be worth using at all. So, clearly, due to the paucity of actions that HTML can perform, and due to the delay imposed by the client/server model of the Web, it is not possible to implement a fully interactive user interface in pure HTML.

In conclusion, HTML is only a markup language but not a programming language, it lacks statements to allow programmer to change the sequence of actions of execution. Web pages written in pure HTML, without adding to the HTML markup tags with some form of programming languages, such as Java or JavaScript, are actually static and flat.

3.1.4 Contribution of HTML in the AU Calculator Project

Since the AU Calculator is a Web based application, HTML is indispensable for the interface design and implementation of AU Calculator project. The primitives of user interfaces of AU Calculator are created with HTML control elements, and the layout of the primitives is arranged using HTML tables. By embedding JavaScript programs in the HTML pages, the user interfaces of the AU Calculator become fully interactive.

3.2 Java and Java Servlet

3.2.1 Purpose of Java and Java Servlet

Java, promoted as a means of adding "Dynamic comment" to World Wide Web pages, is one of the world's key general-purpose programming languages. It was created by Sun Microsystems based on two of the most widely used implementation languages, C and C++. Since the messier, complex, and error prone C/C++ features has been removed, Java is much more concise than its ancestors, and precisely satisfied the needs of information processing requirements of today's businesses and organizations.

Due to a number of built-in networking capabilities, particularly its portability, Java has been widely used for Internet-based and World Wide Web-based applications. Instead of static text and graphics, Web pages become "alive" with interactivities, audios, videos, three-dimensional imaging, animations etc.

The highest-level view of networking in Java is Java Servlet, which using the request-response model of communication. The clients (the World Wide Web browser) request that some action to be performed and the servers (the World Wide Web server) perform the action and response to the clients.

Servlets are server-side extensions programmed against the *Servlet API*, which are used to extend and enhance Web servers. Servlet interface is philosophically equivalent to the old CGI interface, but is both more powerful and extremely more efficient.

Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

A servlet can almost be thought of as an applet that runs on the server side. Servlet technology provides Web developers with a simple, consistent mechanism for building component-based, platform-independent Web-based applications, extending the functionality of a Web server and for accessing existing business systems, without the performance limitation of traditional CGI programs.

Today, servlets are a popular choice for building interactive Web applications, which:

- Helping provide secure access to a Web site,
- Interacting with database on behalf of a client,
- Dynamically generating custom HTML documents to be displayed by browsers
- Maintaining unique session information for each client.

3.2.2 Advantages of Using Java

Java has gained enormous popularity since it first appeared. Its rapid ascension and wide acceptance can be traced to its design and programming features, particularly in its promise that you can write a program once, and run it anywhere. As stated in Java language white paper by Sun Microsystems: "Java is *simple*, *object-oriented*, *distributed*, *interpreted*, *robust*, *secure*, *architecture neutral*, *portable*, *multithreaded*, and *dynamic*."

- *Java is simple.*

  - Java has replaced the complexity of *multiple inheritance* in C++ with a simple structure called *interface*, and has eliminated the use of *pointers* at the programming level, where used to be a source of memory leaking in C++ programs.

- Java uses *automatic memory allocation* and *garbage collection* technology, so that the programmer don't have to take care of memory allocation and garbage collection themselves anymore.
- Although been considered as a very powerful programming language, Java has only a small number of language constructs.
- The clean syntax makes Java programs easy to write and read.

- *Java is Object-oriented.*

Object-oriented programming provides several benefits, including greater flexibility, modularity, reusability, and extensibility etc. Java is purely an Object-oriented programming language. It consists extensive sets of pre-defined classes, which are arranged in treelike hierarchy and grouped in packages.

- *Java is distributed*

Distributed computing involves several computers on a network working together. Programs running on different systems communicate with each other to perform a joint task. Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file.

- *Java is platform independent.*

One of the most important advantages of Java is its portability. There are no platform-specific features on the Java language specification. Java programs can run on any platform, including Microsoft Windows, Macintosh, and most versions of UNIX, such as Solaris, without having to be recompiled.

- *Java is secure*

Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind. The compiler, interpreter, and Java-compatible browsers all contain several levels of security measures that are designed to reduce the risk of security compromise, loss of data and program integrity, and damage to system users.

- *Java facilitate multimedia*

Most programming languages do not have built-in multimedia capabilities. However, through the packages of classes that are an integral part of the Java programming world. JAVA provides extensive multimedia facilities that will enable a programmer to start developing powerful multimedia applications immediately.

- *Java is robust*

- Java puts a lot of emphasis on early checking for possible errors; Java compilers are able to detect many problems that would first show up during execution time in other languages.
- Java eliminates certain types of programming constructs in other languages that are prone to errors. For instance, Java does not support pointers, which eliminates the possibility of overwriting memory and corrupting data.
- Java has a runtime exception-handling feature to provide programming support for robustness, and can catch and respond to an exceptional situation so that the program can continue its normal execution and terminate gracefully when a runtime error occurs.

- *Java is Multithreaded*

Multithreading is the capability for a program to perform several tasks simultaneously within a program. Not as other programming languages that have to call operating system-specific procedures to enable multithreading, Java has smoothly integrated multithreading capabilities into it.

- *Java is Dynamic*

The Java programming language was designed to adapt to an evolving environment. New methods and properties can be added freely in a class without affecting their clients. Also, Java is able to load classes as needed at runtime.

3.2.3 Limitations of Java

The main disadvantage of Java is its speed. Java programs execute more slowly than platform-dependent compiled languages such as C/C++.

Unlike natively compiled code, which is a series of instructions that correlate directly to a microprocessors instruction set and executed by the system directly, programs written in Java has to be compiled into byte code first, and then an interpreter, called a Java Virtual Machine, specifically designed for a computer architecture, must first translate the Java binary code into the equivalent microprocessor instruction. Obviously, this translation takes time and so that; it is inherently slower than platform-dependent compiled language, such as C/C++. However, some modern implementations of Java, such as "just in time" (JIT) compilation, are almost as fast as C++ by using some tricky techniques.

3.2.4 Contribution of Java Servlet to the AU Calculator Project

As stated in the previous sections, the Internet is becoming a dynamic place for information publishing. The AU Calculator project is trying to allow the user interacts with *live* data of Accreditation Units information for courses of Software Engineering Department of Concordia University. So that the Web page has to be dynamic and have the ability to produce Web pages containing information specially targeted at each user.

To accomplish dynamic publishing and interacting on the Internet, usually the following components are involved in the execution of an Internet application:

- Web server providing Internet service;
- Web browser allowing the user interaction with the Web server via Internet;
- CGI programs responding to the user input, bridging the Web browser and Web server;
- Database server storing the course information about the Accreditation units.

Where CGI (Common Gateway Interface) program is important component, which enable Web pages to provide real-time and dynamic information.

The AU Calculator project uses Java Servlet to serve the purpose of a CGI program, to enable communications between Web server and Web browser. Using request-response communication model, Java servlet provides a straightforward means of accessing data in relational database via the Web.

A connection between the client browser and the Web server starts when the client makes a request to the port where the server resides. The server then start the Java servlet program with the user's inputs to gather the information needed. The Java servlet program connects to the appropriate database, retrieves the right data, dynamically generates a Web page that contains the requested information, and the server send the generated Web page to the client as a response.

The following figure briefly described how the AU Calculator application works and how the Java servlets contributes:

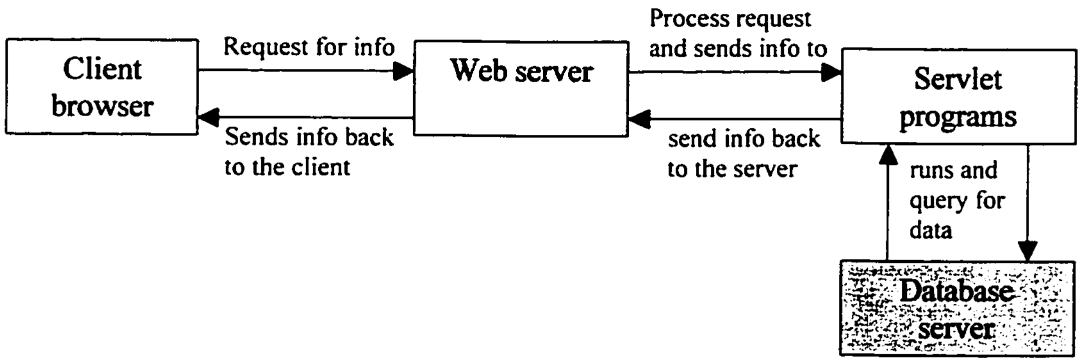


Figure 3.1 The interactions between components of AU Calculator

3.3 JavaScript

3.3.1 Purpose of JavaScript

As its name implies. JavaScript is a scripting language, developed by Sun Microsystems. The major purpose of JavaScript is to create client-side executable to enhance the functionality of existing Web pages and to nudge Web pages away from static displays of data towards interactive applications. JavaScript contains a set of pre-built objects that reflects characteristics of the Web page, and allows JavaScript program to manipulate, modify, and react to the Web page. Most JavaScript programs are launched by actions, which occur on the Web page, often by the user.

When a Web page uses Perl or Java, people usually cannot see the Perl or Java. Programs written in such language are actually staying on the server, running in the back-end of the Internet. Where JavaScript is a front-end language, the program of JavaScript runs on the user's machine directly. When we open a Web page, we are actually seeing both the HTML and JavaScript.

HTML provides the form elements, hypertext links, and other primitive building blocks that form the basis of user interfaces. But, due to the paucity of actions that HTML can perform and the delay imposed by the round-trips of client/server model of the Web. Web pages using pure HTML is not capable to handle events--to respond when the user interacts with it, which is the most crucial feature of any kind of interactive user interface. JavaScript extends HTML by adding event handlers that are invoked when the user interacts with the primitive elements of a user interface. By embedding JavaScript executable content into HTML, interactivity between the user and the computer can take place on the client side.

3.3.2 Advantages of Using JavaScript

- JavaScript has the ability to read the user's input from HTML form elements and to set the value displayed within these elements, to perform interesting actions in response to user interaction.
- JavaScript is notified of various events that occur when a user interacts with the elements of an HTML form, and can execute arbitrary code to "handle" those events
- JavaScript is a general purpose programming language. Arbitrary complex "executable content" can be embedded into Web pages.
- With JavaScript embedded Web pages, the round trip to the server is unnecessary, interaction between user and the Web page is much faster.
- In addition, JavaScript is simpler and easier to learn.

3.3.3 Limitations of JavaScript

JavaScript may not be able to create a CGI program like Java or Perl to communicate with the server. You won't hear anyone boasting about the power of JavaScript as the universal solution to all your programming needs.

3.3.4 Contribution of JavaScript to the AU Calculator

As an Internet application. AU Calculator involves interactions between the intended user and the Web pages that serve as the interface of AU Calculator. Without JavaScript, every user event has to be transmitted to the server. and the server's response to be transmitted back to the user. This will lead to a non-useful user interface due to the round trip delay of the Internet.

In the AU Calculator project, JavaScript was used to embed executables into the HTML page. which is produced dynamically by the Java Servlet program. This largely reduced the number of times of client/server communications via the Internet.

3.4 Open Database Connectivity (ODBC)

3.4.1 Purpose of ODBC

Open Database Connectivity (ODBC) is a widely accepted call-level application-programming interface for manipulating relational data using SQL query syntax across disparate data sources. It provides a standard API that allows a client (i.e. Windows program) to access a wide range of servers (i.e. data sources). When writing code to interact with a database, programmers usually have to add code that talks to a particular database using a proprietary language. This can be quite the daunting task causing much grief.

With ODBC, application developers can simply write to the ODBC API and let the ODBC Manager and Driver takes care of the database language specifics. Regardless of the database type used, all of the calls will be to the ODBC API. Depending on the requirements, the data source to be accessed can be flat file or a relational database. It is also possible to access multiple data sources simultaneously regardless of the database provided. All that developers need to do is to have an ODBC driver, which is specific to the type of database that will be used, installed.

3.4.2 Advantages of Using ODBC

- ODBC is portable. It is the most universally supported API that allows a programmer to abstract a program from a database on different platforms. It is actually available in shipping implementations. with support from more third parties than any other alternative (including non-shipping ones).
- Being an international standard. ODBC allows developers to manipulate a vast array of relational data sources. such as Microsoft Access, Oracle, Fox. etc., through numerous ODBC Drivers from both Microsoft and third party vendors.
- ODBC simplifies and speeds up the development of generic. scalable client applications for distributed by providing a single interface for communicating with multiple, concurrent back-end services.
- ODBC provides crucial capabilities to client/server **O**n-**L**ine **T**ransaction **P**rocessing and **D**ecision **S**upport **S**ystems.

3.4.3 Limitations of ODBC

- The key disadvantage of ODBC is that it is limited to relational, SQL-syntax based database.
- The functionality supported by ODBC is limited. For example, it does not support nested outer joins, etc.

- Since ODBC is also an additional layer between the application's code and the data, it may restrict developers from taking maximum advantage of the specific DBMS features available on the system, such as Groups/Roles/Database events.
- ODBC may slow down applications runtime performance, since it is an extra layer between the application's code and the data; applications need to connect to the ODBC before the ODBC connect to the Database.
- With ODBC, any front-end tool could feasibly modify the databases. To ensure secure of the data, developers must enforce security at a server level, such as assigning groups and roles to the databases accessible by ODBC.

3.4.4 Contribution of ODBC to the AU Calculator

With ODBC, we can do all database access though the Web server, or can access the database directly from the user's machine. In the AU Calculator project, the ODBC client is built in Java Servlet, which uses SQL to request data from or to send data to the server DBMS, which is built in Microsoft Access. Since the DBMS doesn't understand the ODBC client's request until the command passes through the ODBC Driver for that specific DBMS. This ODBC driver is software that can translate the command into a format that the ODBC Server can understand. The ODBC Server sends the answer back to the ODBC Driver, which translates the answer into a format that the ODBC client can understand.

The ODBC has four components, as show in the following figure:

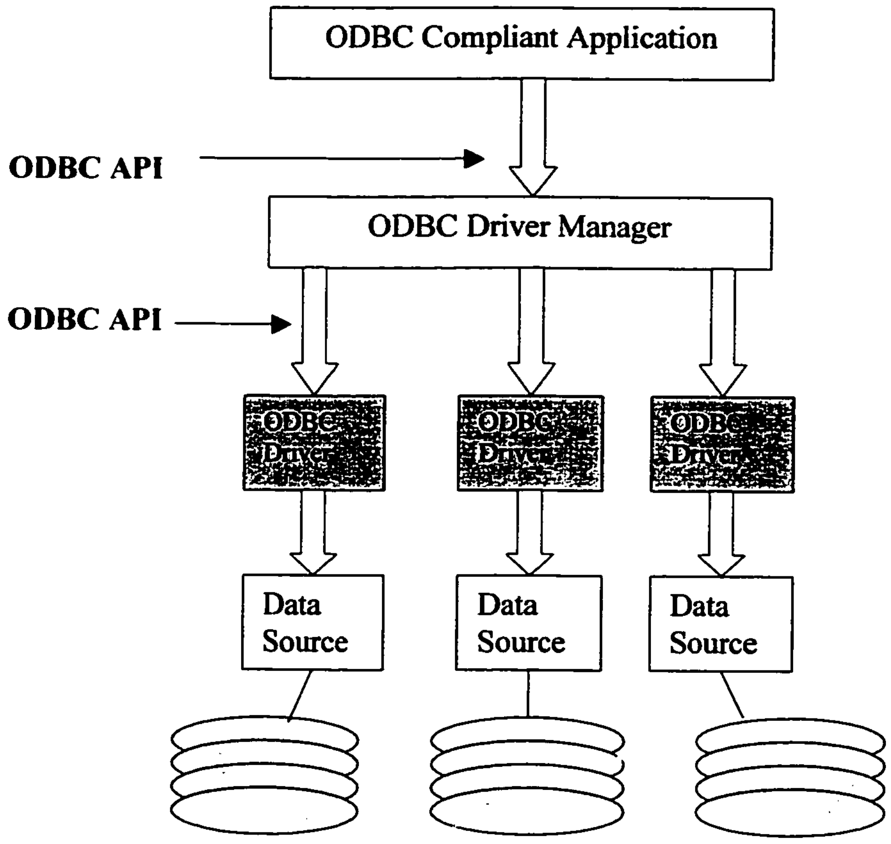


Figure 3.2 The ODBC Architecture

- The primary task of the *application* is to perform processing by passing SQL statements to and receiving results (if any) from the ODBC Driver Manager.
- *Driver Manager* is a *Dynamic Link Library* that manages communication between applications and drivers, it determines which driver to load based on a data source name, loads and unloads drivers, processes ODBC function calls or passes them to a

driver.

- A *driver* is also a *Dynamic Link Library* that implements the functions in the ODBC API. Each driver is specific to a particular DBMS. The drivers processes ODBC function calls received from the Driver Manager, submitting the resultant SQL requests to a specific data source, and returns results to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by the associated DBMS processes the ODBC function calls.
- A *data source* is simply the source of the data. It can be a file or a particular database on a DBMS.

3.5 Microsoft Access

3.5.1 Purpose of MS Access

Microsoft Access is a powerful and robust 32-bit **R**elational **D**ata**b**ase **M**anagement System (RDBMS) for creating desktop and client/server database applications that run under Windows 9x and Windows NT 4x. Basically, a DBMS is a program that facilitates the storage and retrieval of structured information on a computer's hard drive.

A Microsoft Access database is a file with the extension .mdb, which includes different parts of the database, such as tables, forms, reports, queries, as well as macros and visual basic programs for extending the functionality of database applications. It can be used for personal information management, in a small business to organize and manage all data, or in an enterprise to communicate with servers.

Microsoft Access provides in one package the database information and the tools needed to use the database. Microsoft Access Package includes the following elements:

- A relational database system that supports two industrial standard query languages: Structured Query Language (SQL) and Query By Example (QBE);
- A full featured programming language, essentially a subset of Visual Basic;
- A rapid application development environment complete with visual form and report development tools;
- Various wizards and builders to make development easier.

3.5.2 Advantages of Using MS Access

- *Ease of use*

Microsoft Access is designed to take advantage of the Windows operating systems. It provides a broad range of tools offering great simplicity for the first-time users, that even the most inexperienced database users can easily master, making databases building more intuitive. An extensive collection of wizards and add-ins handle most of the mundane chores involved in creating and modifying tables, queries, forms, graphs, and reports that even the most experienced user can take advantage of. Filtering, sorting, querying, and form and report creation are fairly simple tasks, especially when using the toolbars, wizards, and graphical interfaces provided with Microsoft Access. This flexibility allows the experienced developer to build framework database applications, which can be later modified and added to by the client.

Developers can quickly design database systems that provide maximum ease of use for end-users. Instead of putting 100 percent of the database application's development and upgrades into the hands of the developer, the end-user has the flexibility to add to the application with his/her own hands.

Quick development and ease of use are the two primary advantages of choosing Microsoft Access for the database design and use.

- *Data migration*

When comparing various database design systems, Microsoft Access is by far the easiest way to share and move data. Its single file system makes it a breeze to upload, download, or copy and paste entire databases to other computers. Instead of tying up the network, you can upload a copy of your entire database to migrate the database to another department.

This ease of migration makes Microsoft Access a favored tool when developing small databases. The developer and client can easily exchange prototypes of a database design, without having to recreate and run new installations for each updated version. This saves time for the developer and cuts costs for the client.

- *Simple backups/archiving*

The Microsoft Access one-file system provides a simple way to make backups. Instead of backing up several to hundreds of data files, only one single file needs to be backed up. Depending on the size of your database and location of your backup medium, this can be as easy as copying and pasting the file to another drive.

There is no easier product for making an archive of the database. Simply copy the file that contains the data, paste it to a safe place, and rename the file as an archive. The original file can be continually used as is or an empty copy of the database can be opened to start a new day, week, month, quarter, etc. In addition, the archives can be combined through queries to analyze and report summaries for a particular group of data or period of time.

- *Powerful macros*

Access macro language is a surprisingly powerful substitute for traditional database programming techniques. Macros offer Access users the ability to automate database applications by responding to a wide variety of externally and internally generated events with several dozen predefined macro actions. The 40+ macro instructions were so powerful that you could create quite sophisticated database applications by using only Access macros.

- *Network Client/server capability*

Microsoft Access duplicates on the PC desktop many of the features of client/server relational database systems. It is specifically designed for creating multi-user applications where database files are shared on networks. And also, it incorporates a sophisticated security system to prevent unauthorized persons from viewing or modifying the databases. The User-Level Security Wizard makes secure applications easier to implement and administer.

- *Internet features*

Microsoft Access offers a variety of new Internet-related features for creating HTML documents for use on intranet and the Internet. End users now can quickly create quite sophisticated enterprise-wide databases solutions that integrate and leverage XML, XSL, and dynamic Web pages to better allow for the sharing and presentation of data across the intranet and Internet.

3.5.3 Limitations of MS Access

- *Limited to relational databases*

Microsoft Access is a relational DBMS that limits with only relational databases. In recent years, OOP concepts have been widely accepted and object oriented database management systems may be a trend in the future.

- *Limited to small databases*

Microsoft Access is a low capacity database system. It is designed to efficiently manage relatively small numbers of database records. Microsoft Access 2000 has a 2-gigabyte file size limit for an entire database (MDB file). Although it is possible to append hundreds of thousands of records into MS Access database tables, performance will noticeably began to decrease when it reaches a volume of approximately 25,000 records.

Performance will decrease even further as more records and/or additional concurrent users are added in.

The ultimate effect that a large volume of records has on performance is partially influenced by the database design. A Microsoft Access table record can contain anywhere from 1 field to 255 fields. Thus. a 2-field record will require a lot less resources than a 50-field record. The type and form of data being stored in a record also effects performance. Pasting a large JPG or GIF image into a record could make the size of one image record be equal to 50 or more "text only" records.

Although it lacks in high volume capacity. the good news is that efficient database design, data management. and archiving techniques can often work around the capacity limitations of Microsoft Access.

- *Limited to low transaction rates*

Using Microsoft Access for database design. allows the developer to utilize a wealth of input validation, formatting, data display, querying, and reporting tools. As far as user-friendliness. these tools are unparalleled by any other database system currently found on the market. However. pre-built user-friendly tools come with a downside.

Microsoft Access was designed with the primary goal of providing ease of use for inexperienced and experienced database users/developers. To accomplish this goal. Microsoft had to use a one-file system and embed many of its ease of use tools into the actual MDB file, data tables, and form objects. In fact, all of the objects and tables reside within one file format. This type of structure results in Microsoft Access performing sluggishly in high production environments, which require large volumes of data input per hour. Microsoft Access is definitely not a good choice for high production data entry environments.

- *Limited to low concurrent database usage*

Access database runs best as a singer-user system or a small multi-user system. Although Microsoft Access claims that 50 or more users can use Access database concurrently with no problems, that is not always recommended. As mentioned previously, the structural design of Microsoft Access will limit performance in high production environments. This also applies to high levels of concurrent usage, especially concurrent data entry operations. If concurrent usage does not entirely involve data entry operations, then higher concurrent usage may be possible without performance issues. For example, 10 users running reports and read-only queries may not conflict with an additional 5-10 users inputting data at the same time.

3.5.4 Contribution of MS Access to the AU Calculator

The characteristics of MS Access make it an ideal candidate for developing database of the AU Calculator project. The AU Calculator project needs to build several database tables to store the AU information for all courses in the two option groups of the undergraduate programs of Software Engineering Department of Concordia University. Due to the simplicity and the availableness, especially the characteristics of easy to maintain and migrate, MS Access has been chosen as the DBMS of AU Calculator project.

3.6 Jigsaw (Java Servlet Server)

3.6.1 The Purpose of Jigsaw

Jigsaw is a Web server that developed by W3C (World Wide Web Consortium). Its main purpose is to demonstrate the new protocol features of W3C Jigsaw server as they are defined, and to provide the basis for experimentations in the field of server.

3.6.2 The Advantages of Jigsaw

Since Jigsaw is entirely written in Java, it offers the following advantages:

- Portability

Jigsaw will run on any platform that supports Java, with no changes!

- Extensibility

Jigsaw is made of a core and a set of extension modules. Developers can add their own modules, dynamically, to the server. In addition, since the Java runtime comes with both threads and garbage collection, as an extension writer, the developers' job is largely simplified.

- Object Oriented design

The full code of the Jigsaw server is Object Oriented. Instead of consider a resource as being either a CGI script or a file, as most of the existing Web servers did, all of the resources of Jigsaw are objects. Jigsaw allows any object to become accessible via HTTP or whatever protocol implemented.

3.6.3 The limitations of Jigsaw

- Since Jigsaw is written totally in Java, it is considered as slower than some other Web servers. But as W3C claimed, the evaluation of performance of Jigsaw shows that, it performs at least as well as the CERN Web server.

- Since the purpose of Jigsaw is for demonstration of protocol features and for experimental of Web server applications, it is probably not appropriate to use Jigsaw as Web server for Web applications that are for commercial purpose.

3.6.4 The Contributions of Jigsaw to the AU Calculator

The choice of using Jigsaw as the Web server of AU Calculator is not only based on the advantages of Jigsaw as declared by W3C and mentioned above. but also based on the following factor:

- Available on the Web free of charge

Jigsaw can be downloaded from the W3C's Web site with no charges. This is the basic reason for its having been chosen as the Web server of AU Calculator project.

- Ease of use

The Jigsaw Web server is simple to be installed and to be used. To configure Jigsaw, all that of the user has to do is to open an auto-run script file *Install.ba.* The same simplicity applies to running the Jigsaw server, simply clicking on file *Jigsaw.bat*.

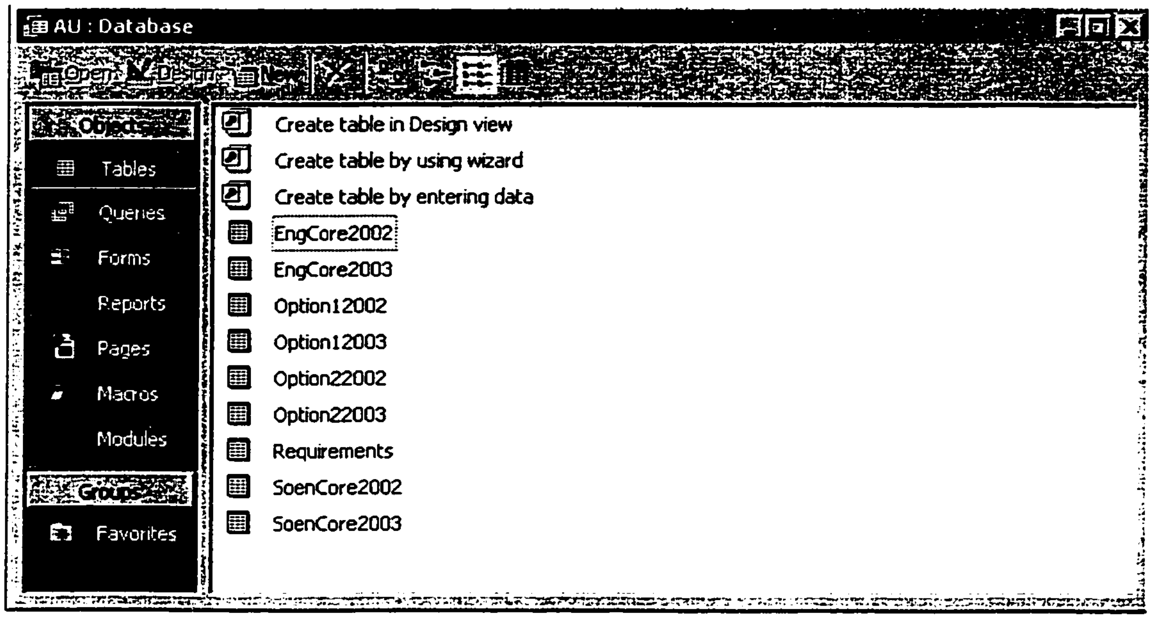
As the Web server, Jigsaw connects the Web pages of AU Calculator with the back-end Java servlet program, which retrieves the user requests, communicates with ODBC, and dynamically generates interfaces of AU Calculator according to user's requests.

4. Implementation of the AU Calculator

The Au Calculator can be divided into two parts: front end and back end. The front-end is made by HTML with embedded JavaScript executables, which serve as the user interfaces of the AU Calculator. The back-end was made by Java Servlet programs and Microsoft Access database.

4.1 Back-end: The Database

The database of AU Calculator AU.mdb is built in MS Access, which is used to store the AU information of the courses of Software Engineering courses. The Figure 4.1 below is the database window of AU.mdb:



Create table in Design view
Create table by using wizard
Create table by entering data
EngCore2002
EngCore2003
Option12002
Option12003
Option22002
Option22003
Requirements
SoenCore2002
SoenCore2003

Figure 4.1 Database objects of AU Calculator

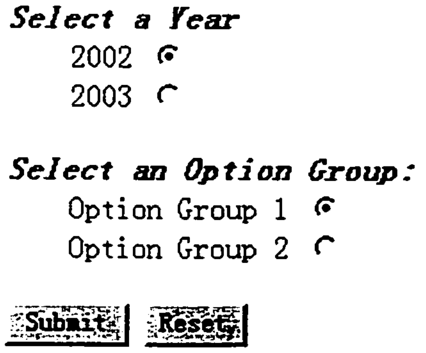
As shown above, to indicate the option groups and the academic years, the database objects have the following name style:

- EngCore2002, EngCore2003, etc.
- SoenCore2002, SoenCore2003, etc.
- Option12002, Option12003, etc.
- Option22002, Option22003, etc.

With this kind of naming system, the new objects of database can be easily added to the system without recompile the AU Calculator program.

4.2 Front-end: Prepare Pre-required Information

Consider the Internet traffic will greatly limit the speed of data transfer, to faster the interactivity of the end user with the AU Calculator, we has to minimize the communications between the client and the server. For this purpose, all of the data needed by a specific user should be retrieved within a single call to the server. This is accomplished by enforcing the end user to select the year and the option group before s/he starts the AU Calculator as showed in the following figure:



Select a Year

[✓ 2002]
2003

Select an Option Group:

[✓ Option Group 1]
Option Group 2

Figure 4.2 Submitting pre-needed information

The *method* of the HTML *form* is POST. As the sample HTML source code shown:

```
<form method="POST" action="http://localhost:8001/servlet/AUServletStart">
 <p><b><i>Select a Year</b></i>
```

```
        <br>2002
    <input type="radio" value="2002" checked name="Year">
        <br>2003
    <input type="radio" value="2003" name="Year" ></p>
  <p><b><i>Select an Option Group:</b></i>
        <br>Option Group 1
    <input type="radio" value="1" checked name="numOptionGroup">
        <br>Option Group 2
    <input type="radio" value="2"  name="numOptionGroup"></p>
  <p><input type="submit" value="Submit" name="B1">
    <input type="reset" value="Reset" name="B2"></p>
  <p><input type="hidden" value="localhost" name="Address">
</form>
```

The Web Server *Jigsaw* will invoke the back end Java Servlet program *AUServletStart.class* when the *Submit* button was clicked.

4.3 Back-end: The Java Servlet Program 1

The information submitted by the user will be passed to an instance of the Java Servlet program *AUServletStart.class*. The invoked instance of *AUServletStart.class* will first initialize the system by calling function *init()* as shown below:

```
  public void init( ServletConfig config ) throws ServletException
  {
    super.init( config );
    try     //try to connect to the database driver on server
    {
      Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
      connection =
        DriverManager.getConnection( URL, "Concordia", "Computer" );
    }
    catch ( Exception e )
    {
      e.printStackTrace();
      connection = null;
    }
  }
```

The above *init()* function doing the following:

- Configuring the system by calling super class function *init()*,
- Connecting to the JDBC-ODBC driver manager for the database *AU.mdb*, with user name 'Concordia', and password 'Computer',
- Handling exceptions for connection failure.

Since the user requested action is POST, the next function called by the Web Server is the callback function *doPost()* as shown below:

```
public void doPost( HttpServletRequest request, HttpServletResponse response )
        throws ServletException, IOException
```

```
  {
    //get user input from the parent page
    strNumOptionGroup = request.getParameter( "numOptionGroup" );
    strYear           = request.getParameter( "Year" );
    strAddress        = request.getParameter( "Address" );
    //initialize the class viariables
    initialize();
    //set the new page for the AU Calculator
    doGet( request, response );
    //do the garbage collection
    cleanup();
  }
```

The function *doPost()* does the following:

- Retrieving the user's inputs.
- Initializing the system,
- Calling the callback function *doGet(),*
- Cleaning up.

The main tasks, such as retrieving data and creating user interface of AU Calculator etc. will be done by the other callback function *doGet()*, as shown below:

```
public void doGet( HttpServletRequest request, HttpServletResponse response )
    throws ServletException, IOException
  {
    response.setContentType( "text/html" );
    PrintWriter output = response.getWriter();
    boolean success1 = getOption();
    boolean success2 = getCore( "SoenCore" );
    boolean success3 = getCore( "EngCore" );
    boolean success4 = getRequirements();
    boolean success5 = getPriorStudy();
    if ( !success1 || !success2 || !success3 || !success4 || !success5 ) {
      output.print( "An error occurred. Try it later." );
      try {
        connection.close();
      }
      catch( Exception e ) {
        System.err.println("Problem closing the database" );
      }
      return;
    }
    StringBuffer buf = new StringBuffer();
    SetBuffer ( buf );
    output.println( buf.toString() );
    output.close();
  }
```

The *doGet()* function does the following:

- Set the response content in HTML/Text format,
- Getting the PrintWriter object that can send the response content to the user, from the server,

- Calling member functions *getOption()*, *getCore()*, *getRequirement()*, *getPriorStudy* to retrieves the requested data.
- Testing the results of the *get* functions, and handling the failure,
- Creating an output buffer, and then calling member function *setBuffer()* to set the buffer contents,
- Telling the Web server to call the *PrintWriter* object to print the output content to the window of the user's Web browser, and close the *PrintWriter* when finished.

The communications to the database server is done by function *getOption()*, *getCore()*, *getPriorStudy()*, *getRequirements()*. These functions have the same structure as function *getOption()* shown below:

```
private boolean getOption()
{
  String query = "SELECT * FROM Option";
  query += strNumOptionGroup;
  query += strYear;
  try
  {
    statement = connection.createStatement();
    resultSet = statement.executeQuery( query );
    boolean moreRecords = resultSet.next();
    if ( !moreRecords )
    {
      return false;
    }
    ResultSetMetaData rsmd = resultSet.getMetaData();
```

```
        do
        {
          strOptionList[numOption] = "";
          int column;
          for ( column = 1; column < rsmd.getColumnCount(); ++column )
          {
            strOptionList[numOption] += resultSet.getString( column );
            strOptionList[numOption] += "~";
          }
          strOptionList[numOption] += resultSet.getString( column );
          numOption += 1;
        } while ( resultSet.next() );
        statement.close();
        return true;
      }
      catch( SQLException sqle )
      {
        sqle.printStackTrace();
        return false;
      }
    }
```

They are doing mainly the following:

- Creating SQL statement that will be used to query the database.
- Calling the database driver to execute the query,
- Retrieving and formatting the resulting set of requested data.

- Storing the retrieved information into member variables,
- Handling exceptions of failure,
- Closing the query statement

At this stage, all of the data needed for the AU Calculator has been stored in the member variables of the instance of *AUServletStart.class*. The next thing *doGet()* does is to create output content in HTML format by calling member function *setBuffer()*:

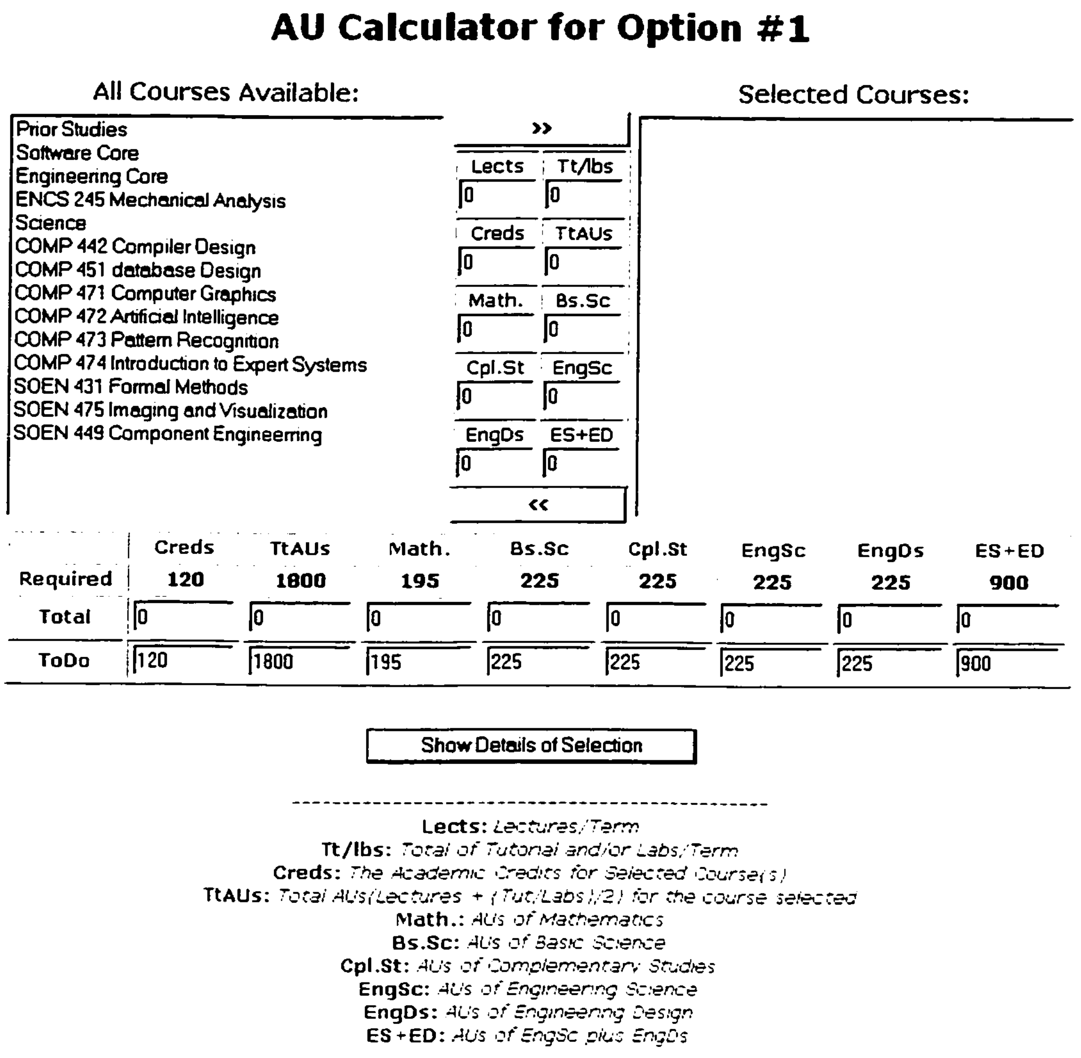
```
private void setBuffer( StringBuffer buf )
{
  buf.append( "<HTML><HEAD><TITLE>AU Calculator</TITLE>\n\n" );
  buf.append( "<SCRIPT LANGUAGE=Javascript>\n" );
  buf.append( "<!--\n" );
  ......
  buf.append( "//-->\n" );
  buf.append( "</SCRIPT>\n\n" );
  buf.append( "</HEAD>\n" );
  buf.append( "<BODY bgColor=#ffeec0>\n" );
  ......
  buf.append( "</BODY></HTML>" );
}
```

The function *setBuffer()* appends to output buffer text in the HTML format, that can serve as the User Interface of the AU Calculator. The embedded JavaScript functions in the generated HTML content can handle interactions between the intended user and the Interface of AU Calculator. Since all of the information stored in the member variables has been put into the hidden fields of the generated HTML content quietly as well, no further client-server communication is needed; therefore, the performance of the AU Calculator can be independent of the Internet traffic.

As a response, the Web server then sends the dynamically generated Web page, the interface of AU Calculator, to the user's browser via the Internet.

4.4 Front-end: The User Interface

As stated in the previous section, after the user clicking on the *submit* button of the HTML form mentioned in section 4.2, the user interface of AU Calculator is generated dynamically by the back-end Java Servlet program *AUServletStart.class* as shown in the following figure:


AU Calculator for Option #1

All Courses Available:

- Prior Studies
- Software Core
- Engineering Core
- ENCS 245 Mechanical Analysis
- Science
- COMP 442 Compiler Design
- COMP 451 database Design
- COMP 471 Computer Graphics
- COMP 472 Artificial Intelligence
- COMP 473 Pattern Recognition
- COMP 474 Introduction to Expert Systems
- SOEN 431 Formal Methods
- SOEN 475 Imaging and Visualization
- SOEN 449 Component Engineering

Selected Courses:

Lects	Tt/lbs
0	0
Creds	**TtAUs**
0	0
Math.	**Bs.Sc**
0	0
Cpl.St	**EngSc**
0	0
EngDs	**ES+ED**
0	0

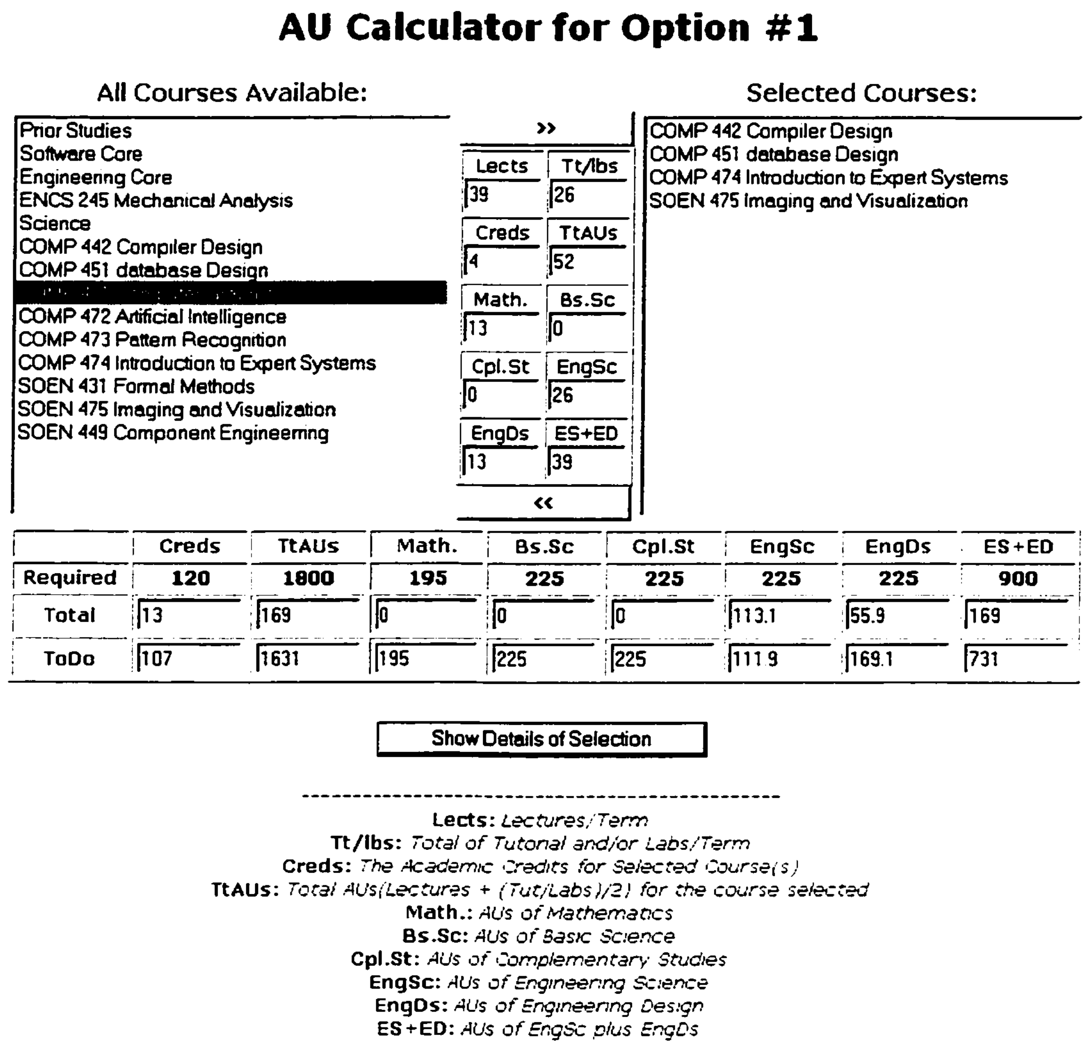
	Creds	TtAUs	Math.	Bs.Sc	Cpl.St	EngSc	EngDs	ES+ED
Required	120	1800	195	225	225	225	225	900
Total	0	0	0	0	0	0	0	0
ToDo	120	1800	195	225	225	225	225	900

Lects: *Lectures/Term*
Tt/lbs: *Total of Tutorial and/or Labs/Term*
Creds: *The Academic Credits for Selected Course(s)*
TtAUs: *Total AUs(Lectures + (Tut/Labs)/2) for the course selected*
Math.: *AUs of Mathematics*
Bs.Sc: *AUs of Basic Science*
Cpl.St: *AUs of Complementary Studies*
EngSc: *AUs of Engineering Science*
EngDs: *AUs of Engineering Design*
ES+ED: *AUs of EngSc plus EngDs*

Figure 4.3 Main Interface of the AU Calculator

With the embedded JavaScript functions to handle all of the user actions, this AU Calculator Interface is fully interactive.

When the user clicks any course in the *All Courses Available* list, the selected course will be highlighted, and the AUs of it will appear in the middle boxes, as shown below:


AU Calculator for Option #1

All Courses Available:

Prior Studies
Software Core
Engineering Core
ENCS 245 Mechanical Analysis
Science
COMP 442 Compiler Design
COMP 451 database Design
[illegible]
COMP 472 Artificial Intelligence
COMP 473 Pattern Recognition
COMP 474 Introduction to Expert Systems
SOEN 431 Formal Methods
SOEN 475 Imaging and Visualization
SOEN 449 Component Engineering

»

Lects	Tt/lbs
39	26
Creds	**TtAUs**
4	52
Math.	**Bs.Sc**
13	0
Cpl.St	**EngSc**
0	26
EngDs	**ES+ED**
13	39

«

Selected Courses:

COMP 442 Compiler Design
COMP 451 database Design
COMP 474 Introduction to Expert Systems
SOEN 475 Imaging and Visualization

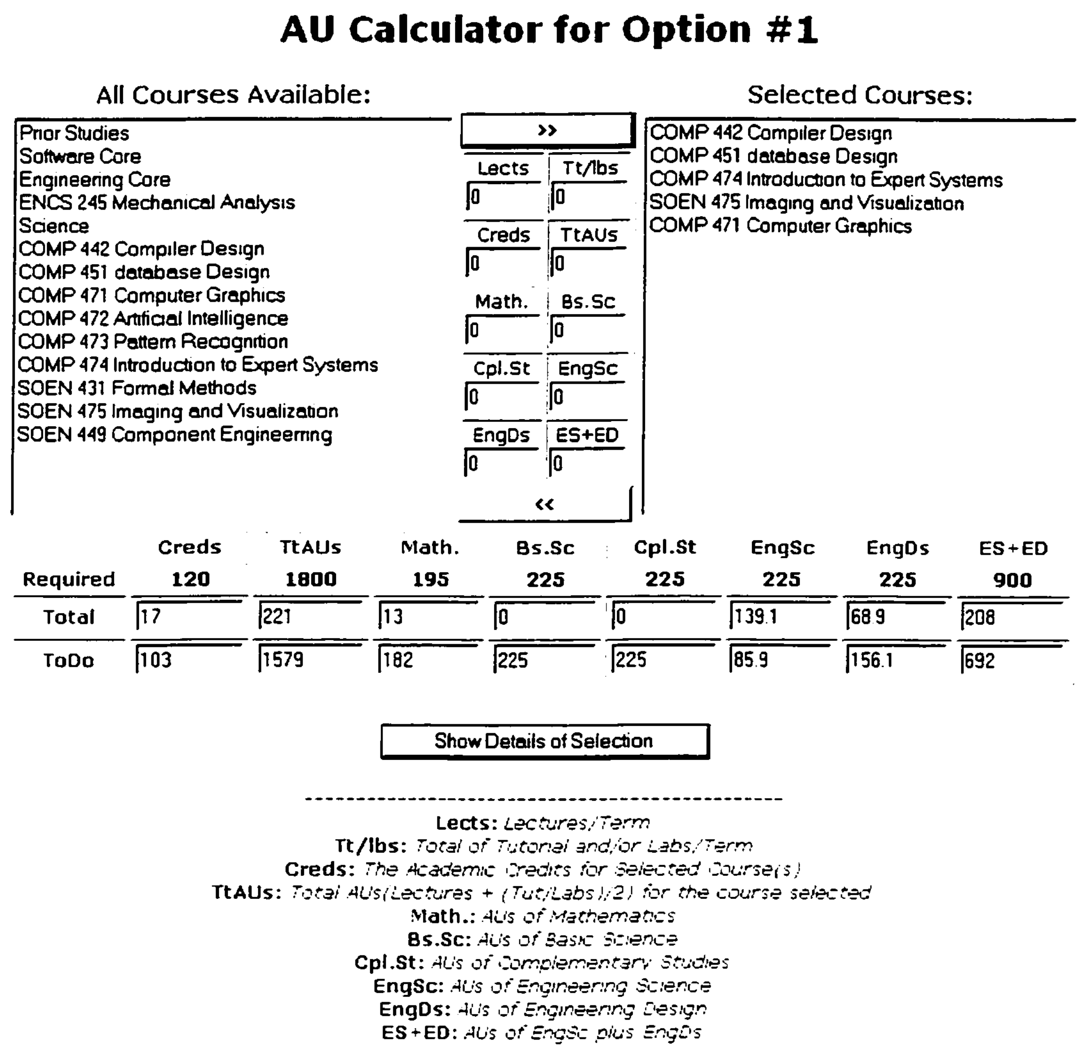
	Creds	TtAUs	Math.	Bs.Sc	Cpl.St	EngSc	EngDs	ES+ED
Required	120	1800	195	225	225	225	225	900
Total	13	169	0	0	0	113.1	55.9	169
ToDo	107	1631	195	225	225	111.9	169.1	731

Show Details of Selection

Lects: *Lectures/Term*
Tt/lbs: *Total of Tutorial and/or Labs/Term*
Creds: *The Academic Credits for Selected Course(s)*
TtAUs: *Total AUs(Lectures + (Tut/Labs)/2) for the course selected*
Math.: *AUs of Mathematics*
Bs.Sc: *AUs of Basic Science*
Cpl.St: *AUs of Complementary Studies*
EngSc: *AUs of Engineering Science*
EngDs: *AUs of Engineering Design*
ES+ED: *AUs of EngSc plus EngDs*

Figure 4.4 Select a course from All Courses Available list

With a course highlighted, the user can click the button >>, to add that course into the *Selected Courses* list, and the AUs of that course will be added/reduced automatically into/from the bottom *Total/ToDo* boxes, as shown below:



AU Calculator for Option #1

All Courses Available:

- Prior Studies
- Software Core
- Engineering Core
- ENCS 245 Mechanical Analysis
- Science
- COMP 442 Compiler Design
- COMP 451 database Design
- COMP 471 Computer Graphics
- COMP 472 Artificial Intelligence
- COMP 473 Pattern Recognition
- COMP 474 Introduction to Expert Systems
- SOEN 431 Formal Methods
- SOEN 475 Imaging and Visualization
- SOEN 449 Component Engineering

Lects	Tt/lbs
0	0
Creds	TtAUs
0	0
Math.	Bs.Sc
0	0
Cpl.St	EngSc
0	0
EngDs	ES+ED
0	0

Selected Courses:

- COMP 442 Compiler Design
- COMP 451 database Design
- COMP 474 Introduction to Expert Systems
- SOEN 475 Imaging and Visualization
- COMP 471 Computer Graphics

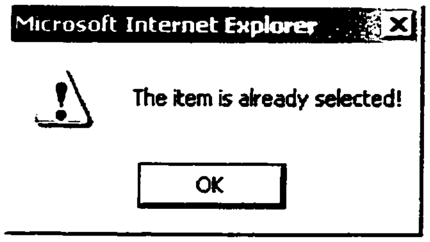
	Creds	TtAUs	Math.	Bs.Sc	Cpl.St	EngSc	EngDs	ES+ED
Required	**120**	**1800**	**195**	**225**	**225**	**225**	**225**	**900**
Total	17	221	13	0	0	139.1	68.9	208
ToDo	103	1579	182	225	225	85.9	156.1	692

Lects: *Lectures/Term*
Tt/lbs: *Total of Tutorial and/or Labs/Term*
Creds: *The Academic Credits for Selected Course(s)*
TtAUs: *Total AUs(Lectures + (Tut/Labs)/2) for the course selected*
Math.: *AUs of Mathematics*
Bs.Sc: *AUs of Basic Science*
Cpl.St: *AUs of Complementary Studies*
EngSc: *AUs of Engineering Science*
EngDs: *AUs of Engineering Design*
ES+ED: *AUs of EngSc plus EngDs*

Figure 4.5 Added a course into Selected Courses list

If the highlighted course is already in the selected course list, then clicks the button

will pop up message:

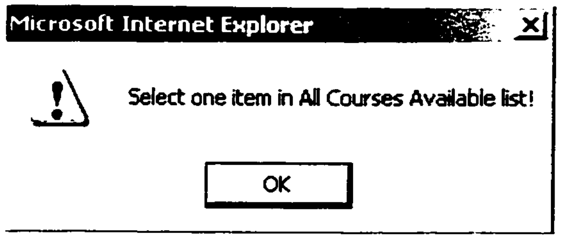

Microsoft Internet Explorer

The item is already selected!

Figure 4.6 Pop up message 1

On the other hand, if there is no any course highlighted, clicking on

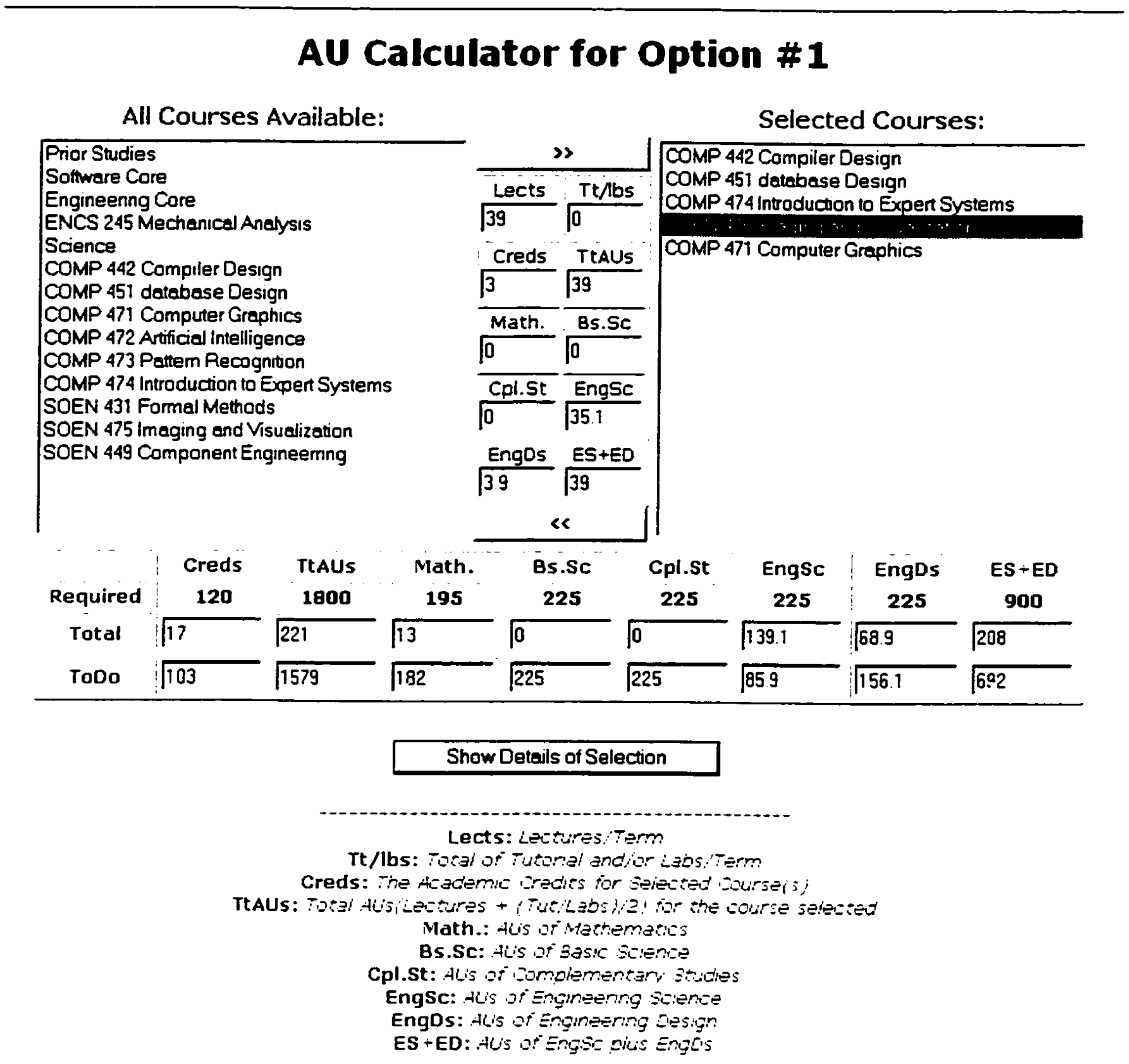
will pop up message:


Microsoft Internet Explorer

Select one item in All Courses Available list!

Figure 4.7 Pop up Message 2

When the user clicks any course in the *Selected Courses* list, the selected course will be highlighted, and the AUs of it will appear in the middle boxes, as shown below:



AU Calculator for Option #1

All Courses Available:

Prior Studies
Software Core
Engineenng Core
ENCS 245 Mechanical Analysis
Science
COMP 442 Compiler Design
COMP 451 database Design
COMP 471 Computer Graphics
COMP 472 Artificial Intelligence
COMP 473 Pattern Recognition
COMP 474 Introduction to Expert Systems
SOEN 431 Formal Methods
SOEN 475 Imaging and Visualization
SOEN 449 Component Engineerıng

Selected Courses:

COMP 442 Compiler Design
COMP 451 database Design
COMP 474 Introduction to Expert Systems
[illegible]
COMP 471 Computer Graphics

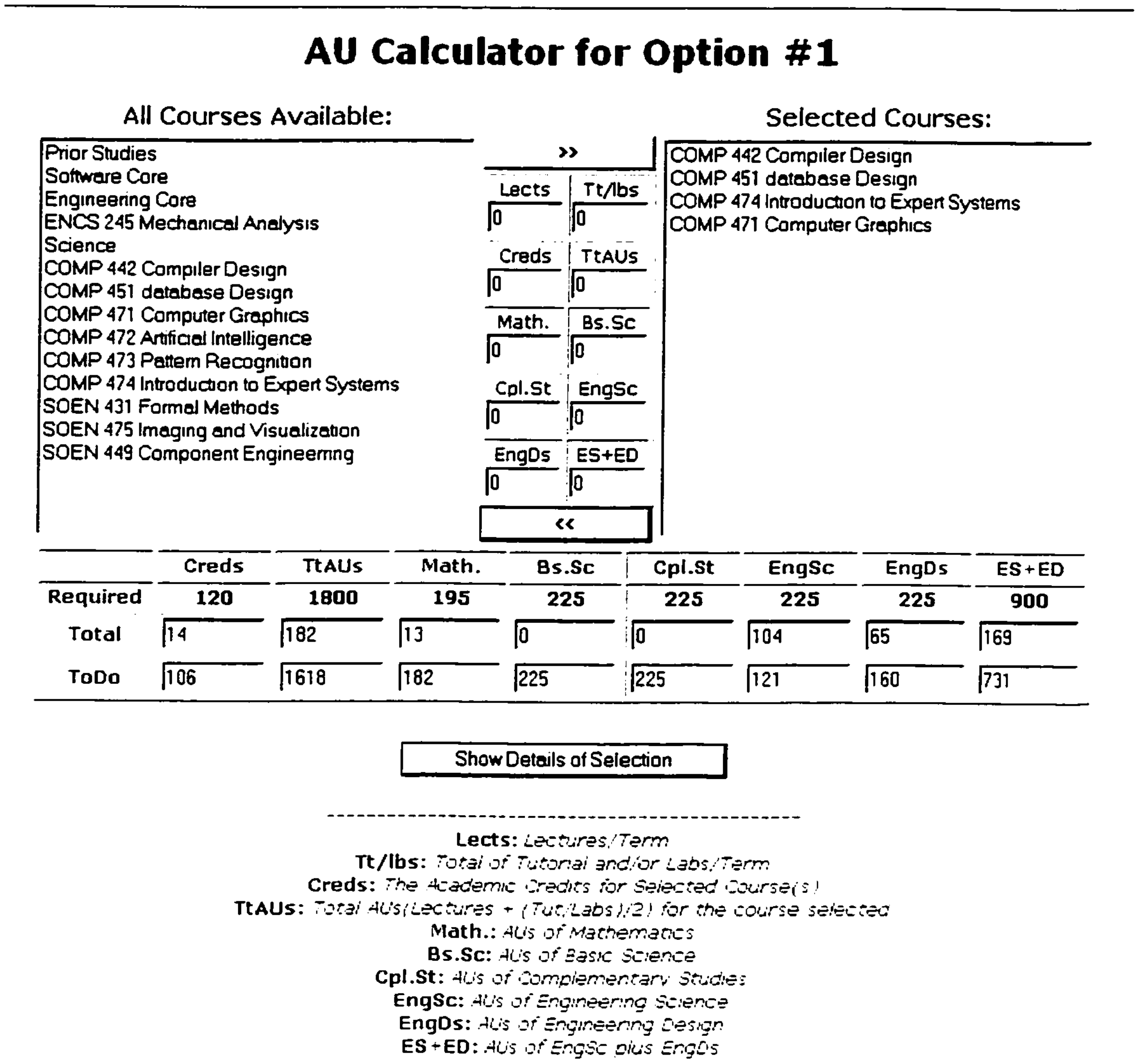
Lects	Tt/lbs
39	0
Creds	TtAUs
3	39
Math.	Bs.Sc
0	0
Cpl.St	EngSc
0	35.1
EngDs	ES+ED
3.9	39

	Creds	TtAUs	Math.	Bs.Sc	Cpl.St	EngSc	EngDs	ES+ED
Required	120	1800	195	225	225	225	225	900
Total	17	221	13	0	0	139.1	68.9	208
ToDo	103	1579	182	225	225	85.9	156.1	692

Lects: *Lectures/Term*
Tt/lbs: *Total of Tutorial and/or Labs/Term*
Creds: *The Academic Credits for Selected Course(s)*
TtAUs: *Total AUs(Lectures + (Tut/Labs)/2) for the course selected*
Math.: *AUs of Mathematics*
Bs.Sc: *AUs of Basic Science*
Cpl.St: *AUs of Complementary Studies*
EngSc: *AUs of Engineering Science*
EngDs: *AUs of Engineering Design*
ES+ED: *AUs of EngSc plus EngDs*

Figure 4.8 Select a course from the Selected Courses list

With the course highlighted, the user can click the button <<, to remove that course from the *Selected Courses* list, and the AUs of that course will be reduced/added automatically from/into the bottom *ToDo/Total* boxes, as shown below:



AU Calculator for Option #1

All Courses Available:

- Prior Studies
- Software Core
- Engineering Core
- ENCS 245 Mechanical Analysis
- Science
- COMP 442 Compiler Design
- COMP 451 database Design
- COMP 471 Computer Graphics
- COMP 472 Artificial Intelligence
- COMP 473 Pattern Recognition
- COMP 474 Introduction to Expert Systems
- SOEN 431 Formal Methods
- SOEN 475 Imaging and Visualization
- SOEN 449 Component Engineering

Selected Courses:

- COMP 442 Compiler Design
- COMP 451 database Design
- COMP 474 Introduction to Expert Systems
- COMP 471 Computer Graphics

Lects	Tt/lbs
0	0
Creds	**TtAUs**
0	0
Math.	**Bs.Sc**
0	0
Cpl.St	**EngSc**
0	0
EngDs	**ES+ED**
0	0

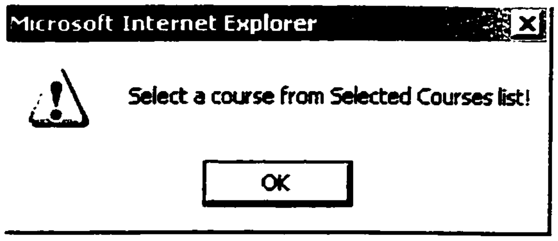
	Creds	TtAUs	Math.	Bs.Sc	Cpl.St	EngSc	EngDs	ES+ED
Required	**120**	**1800**	**195**	**225**	**225**	**225**	**225**	**900**
Total	14	182	13	0	0	104	65	169
ToDo	106	1618	182	225	225	121	160	731

Lects: *Lectures/Term*
Tt/lbs: *Total of Tutorial and/or Labs/Term*
Creds: *The Academic Credits for Selected Course(s)*
TtAUs: *Total AUs(Lectures + (Tut/Labs)/2) for the course selected*
Math.: *AUs of Mathematics*
Bs.Sc: *AUs of Basic Science*
Cpl.St: *AUs of Complementary Studies*
EngSc: *AUs of Engineering Science*
EngDs: *AUs of Engineering Design*
ES+ED: *AUs of EngSc plus EngDs*

Figure 4.9 Remove a course from the Selected Courses list

If there is no any course highlighted in the *Selected Course* list, clicking on

<< will pop up message:



Select a course from Selected Courses list!

Figure 4.10 Pop up Message 3

4.5 Back-end: The Java Servlet Program 2

As mentioned above, the main User Interface of the AU Calculator is a Web page, and its HTML form method is also POST, the ACTION of the Web server is to call the back-end Java Servlet program *AUSelectionDetail.class*:

```
......
<FORM NAME=cours METHOD=post ACTION=http://localhost:8001/servlet/AUSelectionDetail>
......
```

When the user want to view/print the final results of her/his selection, by clicking on the button Show Details of Selection, the information inputted by the user with the main interface page are send to the Web server, and the Web server will start the back-end Java Servlet program *AUSelectionDetail.class* to dynamically generating another Web page that containing the detailed information of AUs of all of the courses selected.

The following callback function *doPost()* of class *AUSelectionDetail* is called by the Web server:

```
public void doPost( HttpServletRequest request, HttpServletResponse response )
  throws ServletException, IOException
{
  initialize();
  for ( int i = 0; i < NUM_VALUES; i++ )
  {
    req[i] = request.getParameter( "req" + i );
  }
  strTotal = request.getParameter( "total" );
  strTodo  = request.getParameter( "todo" );
  strSelectedList = request.getParameter( "Sel" );
  doGet( request, response );
  cleanup();
}
```

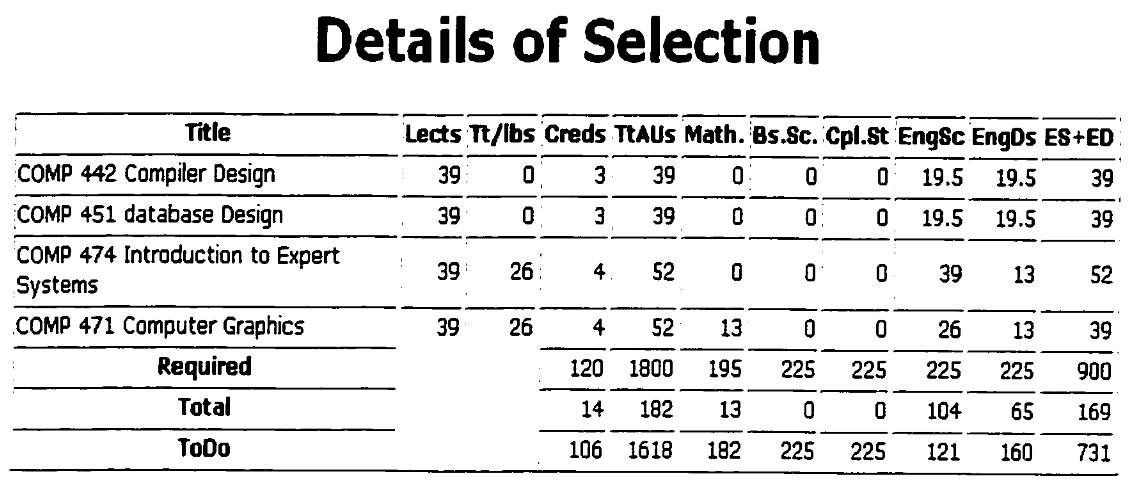
The function *doPost()* does the following:

- Retrieves the user inputs,
- Stores the retrieved information into member variables.
- Calls callback function doGet(),
- Cleanup the system.

What the *doGet()* function does is to create the output buffer in the HTML format, and formatting it in a table view, and inserting the information stored in the member variables into the fields of the HTML table created.

4.6 Front-end: The Resulting Page

The Web server then sends the dynamically generated resulting Web page, which contains the detailed AU information for every selected course, to the user's Web browser, as shown in Figure 4.11:



Details of Selection

Title	Lects	Tt/lbs	Creds	TtAUs	Math.	Bs.Sc.	Cpl.St	EngSc	EngDs	ES+ED
COMP 442 Compiler Design	39	0	3	39	0	0	0	19.5	19.5	39
COMP 451 database Design	39	0	3	39	0	0	0	19.5	19.5	39
COMP 474 Introduction to Expert Systems	39	26	4	52	0	0	0	39	13	52
COMP 471 Computer Graphics	39	26	4	52	13	0	0	26	13	39
Required			120	1800	195	225	225	225	225	900
Total			14	182	13	0	0	104	65	169
ToDo			106	1618	182	225	225	121	160	731

Figure 4.11 The Details of Selection page

5. Conclusions

The Accreditation Units Calculator (AU Calculator) project is tool designed for Software Engineering students to viewing and/or calculating the Accreditation Units for the Software Engineering courses available at the Software Engineering Department of Concordia University.

Several computer technologies have been used in the project. The following are some brief conclusions:

- Since the project is a Web based application, in order to minimize the overhead of Internet transfer time, the interface of the AU Calculator was purposely designed simple, and the embedded JavaScript code enhanced the functionality of the interface, and the communication between the browser and server has been reduced greatly, so that performance of the AU Calculator is independent of the Internet traffic.

- Instead of been a CGI based Internet application, the AU Calculator project is a Java Servlet based Internet application. The Java Servlet provides Web developers with a simple, consistent mechanism for building component-based, platform-independent Web-based interactive applications, extending the functionality of a Web server and for accessing existing business systems, without the performance limitation of traditional CGI programs.

- The database for the AU Calculator project, built in Microsoft Access, is pretty simple and in a rather small scale. It is extendable in the future for the Accreditation Units information of the Software Engineering courses in the new academic years.

- New technologies in the high-tech industry are always emerging in an astounding rate; therefore, some tools and methods used in this project might be replaced by new technologies in the future.

6. References

- *HTML Publishing on the Internet*
  2nd Edition ©1998
  Brent Heslop & David Holzgang
  ISBN 1566046254

- Frontiernet's web page for HTML Tag List
  http://www.frontiernet.net/~gene/html/index.html

- NetScape's Website for *JavaScript References* and *Core Guide* –
  http://developer.netscape.com/library/manuals

- W3C's Java Server - *Jigsaw Tutorials*
  http://www.w3.org/Jigsaw/Overview.html

- *A Tutorial for Servlets and JavaServer Pages (JSP)* 1:0
  http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

- *Java™ How to Program*
  Third Edition, H.M.Dietel, P.J.Dietel, Dietel & Associates, Inc.
  © 2000 by Prentice-Hall, Inc. Upper Saddle River, New Jersey 07458
  ISBN: 0130125075

- *Advanced Java™ 2 Platform How to Program*
  H.M.Dietel, P.J.Dietel, S. E. Santry, Dietel & Associates, Inc
  © 2002 by Prentice-Hall, Inc. Upper Saddle River, New Jersey 07458
  ISBN: 0130895601

- Microsoft's *MSDN library* Web site:
  http://msdn.microsoft.com/library/default.asp