University of Nevada
Reno

Implementation of Interactive Course Web Site

A professional paper submitted in partial fulfillment
of the requirements for the degree of
Master of Science
with a major in Computer Science

by

Mohammad R. Islam

Dr. Frederick C. Harris, Jr., advisor
August 2000

Abstract

At the time we began development, we took it on faith that the Servlet API would soon be ported to other web servers (Apache, Netscape, IIS), and of course our faith has been amply rewarded. Not only are Apache, NSAPI, and ISAPI servlet plugins available from several vendors, but high-powered, Java technology-enabled application servers like WebLogic's Tengah and ATG's Dynamo are fully servlet-compliant. The HTML/JavaScript front end gave us the most lightweight platform-independent solution and was adequate for our needs because our interface, besides a few forms, consisted mostly of information presentation. The Servlet API provided us with a simple, robust, and powerful object framework for building this HTML-based application, including objects for retrieving arguments from a web-server request, a simple stream interface for sending the HTML response to the client, and even more advanced functionality, such as cookies and server-side includes of servlets.

This study involves extensive use of java servlets to develop an interactive course website. Some techniques of Online Educations are implemented in this paper. The packages used are Java Development Kit, Java Servlet Development Kit, a Relational Database Management System (miniSQL), JDBC driver Software to connect the database, Apache Web Server.

Contents

List of Figures

List of Tables

1. Introduction

There is much excitement over the Internet and World Wide Web. The Internet ties the “information world” together. The World Wide Web makes the Internet easy to use and gives it the flair and sizzle of multimedia. Organizations see the Internet and the Web as crucial to their information systems strategies. Java provides a number of built-in networking capabilities that make it easy to develop Internet-based and Web-based applications [2]**.**

Online Education refers to any form of learning and teaching that takes place via a computer network. The network could be a local bulletin board system (BBS) or it could be the global Internet and World Wide Web. The network could also be a local area network (LAN) or an intranet within a particular organization. Historically, online interaction has been called "computer mediated communication" (CMC), although this term covers applications beyond instruction (e.g., decision-making in work teams). The most common function used in online education is electronic mail, chat, bulletin board etc. Online education also involves access to databases in the form of text files or multimedia web pages, as well as the exchange of information (e.g., assignments, course materials) via file transfers [7].

Online education is becoming increasingly common in schools, colleges, and the training realm. Initially, it was used to supplement existing classroom instruction, but over time, online classes have become the primary form of interaction and information. The purpose of this ‘Implementation of Interactive Course Website’ is to construct some of the basic characteristics of online education.

Some of the available technologies to develop Web Applications are CGI, proprietary server APIs, server-side JavaScript, or Microsoft’s Active Server Pages [3]. All these are viable solutions, but they each have their own set of problem. Servlets enhance the functionality of World Wide Web servers. Servlet technology today is primarily designed for use with the HTTP protocol of the World Wide Web, but servlets are being developed for other technologies. Servlets are effective for developing Web based solutions that help provide secure access to a Web site, that interact with databases on behalf of a client, that dynamically generate custom HTML documents to be displayed by browsers and that maintain unique session information for each client [1]. Many developers feel that servlets are the right solutions for database-intensive applications that communicate with so-called thin clients-applications that require minimal client-side support. The server is responsible for the database access. Clients connect to the server using standard protocols available on all client platforms. Thus, the logic code can be written once and reside on the server for access by the clients.

2. Background Literature

In late 1996, Java on the server side was a very hot development topic. Several major software vendors were marketing technologies specifically aimed at helping server-side Java developers do their jobs more efficiently. Most of these products provided a prebuilt infrastructure that could lift the developer's attention from the raw socket level into the more productive application level. For example, Netscape introduced something it named "server-side applets"; the World Wide Web Consortium included extensible modules called "resources" with its Java-based Jigsaw web server; and with its Website server, O'Reilly Software promoted the use of a technology it (only coincidentally) dubbed "servlets." The drawback: each of these technologies was tied to a particular server and designed for very specific tasks.

In early 1970, JavaSoft (a company that has since been reintegrated into Sun Microsystems as the Java Software Division) finalized its definition of Java servlets [4]. This action consolidated the scattered technologies into a single, standard, generic mechanism for developing modular server-side Java code. Servlets were designed to work with both Java-based and non-Java-based servers. Supports for servlets has since been implemented in nearly every web server, from Apache to Zeus.

While servlets can be used to extend the functionality of any Java-enabled server, today they are most often used to extend web servers, providing a powerful, efficient replacement for CGI scripts. When you use a servlet to create dynamic content for a web page or otherwise extend the functionality of a web server, you are in effect creating a web application. While a web page merely displays static content and lets the user navigate through that content, a web application provides a more interactive experience.

A web application can be as simple as a keyword search on a document archive or as complex as an electronic storefront. Web application are being deployed on the Internet and on corporate intranets or extranets, where they have the potential to increase the productivity and change the way that companies, large and small, do business.

Servlets have been quick to gain acceptance because, unlike many new technologies that must first explain the problem or task they were created to solve, servlets are a clear solution to a well-recognized and widespread need: generating dynamic web content. From corporations down to individual web programmers, people who struggled with the maintenance and performance problems of CGI-based web programming are turning to servlets for their power, portability, and efficiency. Others, who were perhaps intimidated by CGI programmer’s apparent reliance on manual HTTP communication and the Perl and C languages, are looking to servlets as a manageable first step into the world of web programming.

3. An Invitation to Servlets

This chapter answers the question "What is a Servlet?", shows typical uses for Servlets, compares Servlets to CGI programs and explains the basics of the Servlet architecture and the Servlet lifecycle. It also gives a quick introduction to HTTP and its implementation in the HttpServlet class.

3.1 What is a Servlet?

Servlets are modules of Java code that run in a server application (hence the name "Servlets", similar to "Applets" on the client side) to answer client requests. Servlets are not tied to a specific client-server protocol but they are most commonly used with HTTP and the word "Servlet" is often used in the meaning of "HTTP Servlet" [1].

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

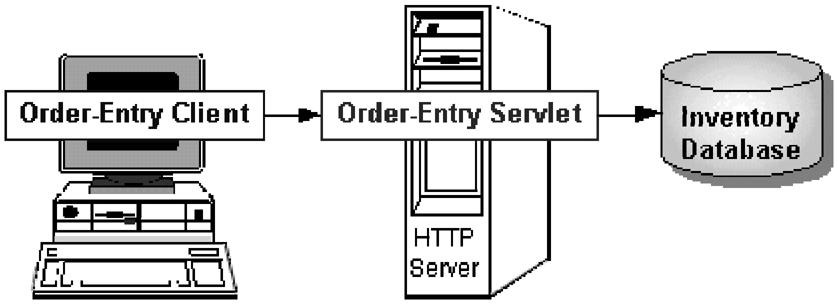


Figure1: A servlet communicating with database

Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface.

3.2. Practical Application for Java Servlets

Servlets can be used for any number of Web-related applications. Here are some important applications of servlets:

- Development of e-commerce “store fronts” is becoming one of the most common uses for java servlets. A servlet can build an online catalog based on the contents of a database. It can then present this catalog to the customer using dynamic HTML. The customer will choose the items to be ordered, enter the shipping and billing information, and then submit the data to a servlet. When the servlet received the posted data, it will process the order and place them in the database for fulfillment. Every one of these processes can easily be implemented using Java servlets.
- Servlets can be used to deploy Web sites that open up large legacy systems on the Internet. Many companies have massive amounts of data stored on large mainframe systems. These businesses do not want to re-architect their systems. So they choose to provide inexpensive Web interfaces into them. Because we have the entire JDK at our disposal and security provided by the Web server, we can use servlets to interface into these systems using anything from TCP/IP to CORBA.

- When developing a distributed object application that will be deployed to the Web, we run into access issues. If we choose to use applets in out client browser, we are only able to open a connection to the originating server, which might be behind a firewall. Getting through a firewall using RMI is a very common problem. If servlets are employed, we can tunnel through the firewall using a servlet technology called HTTP Tunneling. This enables the applet to access objects that can be running almost anywhere on the network.

3.3 Servlets vs. CGI

The traditional way of adding functionality to a Web Server is the Common Gateway Interface (CGI), a language-independent interface that allows a server to start an external process which gets information about a request through environment variables, the command line and its standard input stream and writes response data to its standard output stream [6]. Each request is answered in a separate process by a separate instance of the CGI program, or CGI script (as it is often called because CGI programs are usually written in interpreted languages like Perl).

Servlets has the following advantages over CGI:

- A Servlet does not run in a separate process. This removes the overhead of creating a new process for each request.
- A Servlet stays in memory between requests. A CGI program (and probably also an extensive runtime system or interpreter) needs to be loaded and started for each CGI request.
- There is only a single instance, which answers all requests concurrently. This saves memory and allows a Servlet to easily manage persistent data.
- A Servlet Engine can run a Servlet in a restrictive *Sandbox* (just like an Applet runs in a Web Browser's Sandbox), which allows secure use of untrusted and potentially harmful Servlets.

3.4 The Basic Servlet Architecture

A Servlet, in its most general form, is an instance of a class, which implements the *javax.servlet*. Servlet interface. Most Servlets, however, extend one of the standard implementations of that interface, namely *javax.servlet.GenericServlet* and *javax.servlet.http.HttpServlet*. The following paragraph will be discussing only HTTP Servlets, which extend the *javax.servlet.http.HttpServlet* class.

In order to initialize a Servlet, a server application loads the Servlet class (and probably other classes which are referenced by the Servlet) and creates an instance by calling the default constructor. Then it calls the Servlet's *init(ServletConfig config)* method. The Servlet should perform one-time setup procedures in this method and store the *ServletConfig* object so that it can be retrieved later by calling the Servlet's *getServletConfig()* method. GenericServlet handles this. Servlets which extend *GenericServlet* (or its subclass *HttpServlet*) should call *super.init(config)* at the beginning of the init method to make use of this feature. The *ServletConfig* object contains Servlet parameters and a reference to the Servlet's *ServletContext.* The *init* method is guaranteed to be called only once during the Servlet's lifecycle. It does not need to be thread-safe because the service method will not be called until the call to *init* returns.

When the Servlet is initialized, its *service(ServletRequest req, ServletResponse res)* method is called for every request to the Servlet. The method is called concurrently (i.e. multiple threads may call this method at the same time) so it should be implemented in a thread-safe manner.

When the Servlet needs to be unloaded (e.g. because a new version should be loaded or the server is shutting down) the *destroy()* method is called. There may still be threads that are executing the service method when destroy is called, so destroy has to be thread-safe. All resources which were allocated in init should be released in destroy. This method is guaranteed to be called only once during the Servlet's lifecycle.

A typical Servlet lifecycle

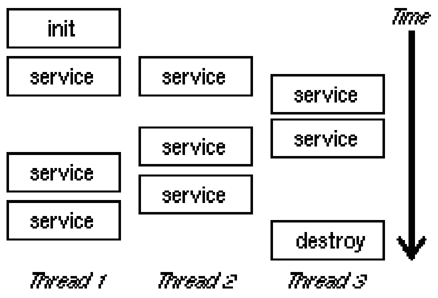


Figure 2: A servlet’s lifecycle

3.5 Reasons To Use Java Servlets

Java servlets are one of the most exciting new technologies. Servlets are efficient, persistent, portable, robust, extensible, secure, and they are receiving widespread acceptance [3]. If it is used only to replace CGI, we will have saved a lot of time and headache. Servlets solve many of the common problems we run into when using CGI, and they prove to have a clear advantage over many of the other alternatives.

3.5.1 Efficient

A servlets initialization code is executed only the first time the Web server loads it. After the servlet is loaded, handling new request is only a matter of calling a service method. This is a much more efficient technique than loading a completely new executable with every request.

3.5.2 Persistent

Servlets can maintain states between requests. When a servlet is loaded, it stays resident in memory while serving incoming requests. A simple example of this would be a Vector that holds a list of categories used in an online catalog. When the servlet is initialized, it queries the database for a list of categories and stores these categories in a Vector. As it services requests, the servlets accesses the Vector that holds the categories instead of querying the database again. Taking advantage of the persistent characteristics of servlets can improve the performance of the application drastically.

3.5.3 Portable

Servlets are developed using Java; therefore, they are portable. This enables servlets to be moved to a new operating system without changing the source. Codes compiled on a Windows NT platform can easily moved to Solaris box without changing anything.

3.5.4 Robust

Because servlets are developed with access to the entire JDK, they are very powerful and robust solutions. Java provides a very well defined exception hierarchy for error handling. It has a garbage collector to prevent problems with memory leaks. In addition, it includes a very large class library that includes network support, file support, database access, distributed object components, security, and many other classes.

3.5.5 Extensible

Another advantage servlets gain by being developed in an object-oriented language like Java is they can be extended and polymorphed into new objects that better suit our needs. A good example of this is an online catalog. We might want to display the same catalog search tool at the top of every dynamic page throughout our Web site. We definitely don’t want to add this code to every one of our servlets. So, we implement a base servlet that builds and initializes the search tool and then extend it to display transaction-specific responses.

3.5.6 Secure

Servlets run on the server side, inheriting the security provided by the Web server. Servlets also can take advantages of the Java Security Manager.

3.5.7 Widespread Acceptance

Because of all there is to be gained from using Java servlets, they are being widely accepted. Vendors are providing servlet support in two main forms. The first is servers that have built-in support for servlets, and the second is by using third-party add-ons.

4. Database Access with JDBC

JDBC technology is a Java API that lets us access virtually any tabular data source from the Java programming language. It provides cross-DBMS connectivity to a wide range of SQL databases, and now, with the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files [9].

The JDBC API allows developers to take advantage of the Java platform's "Write Once, Run Anywhere" capabilities for industrial strength, cross-platform applications that require access to enterprise data. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment.

The "Write Once, Run Anywhere" Java 2 Platform is a safe, flexible, and complete cross-platform solution for developing robust Java applications for the Internet and corporate intranets. The open and extensible Java Platform APIs are a set of essential interfaces that enable developers to build their Java applications and applets. The Java 2 Platform provides uniform, industry-standard, seamless connectivity and interoperability with enterprise information assets [6].

4.1 JDBC API Overview

The JDBC API makes it possible to do three things:

- Establish a connection with a database or access any tabular data source
- Send SQL statements
- Process the results

4.2 JDBC Architecture

The JDBC API contains two major sets of interfaces: the first is the JDBC API for application writers, and the second is the lower-level JDBC driver API for driver writers.

JDBC technology drivers fit into one of four categories. Applications and applets can access databases via the JDBC API using pure Java JDBC technology-based drivers, as shown in the following figure:

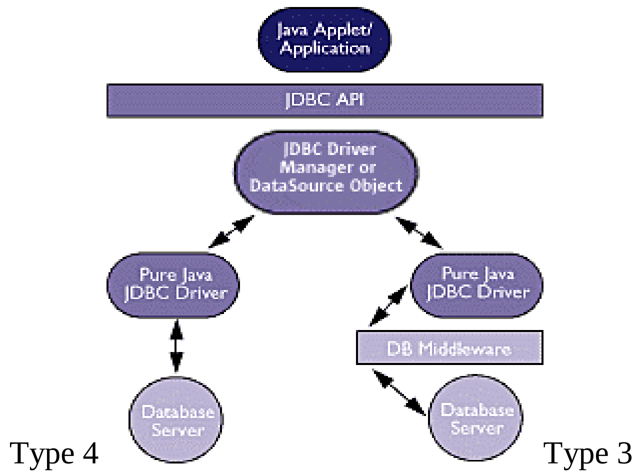


Figure 3: Pure Java JDBC Technology based drivers

Type 4: Direct-to-Database Pure Java Driver: This style of driver converts JDBC calls into the network protocol used directly by DBMSs, allowing a direct call from the client machine to the DBMS server and providing a practical solution for intranet access.

Type 3: Pure Java Driver for Database Middleware: This style of driver translates JDBC calls into the middleware vendor's protocol, which is then translated to a DBMS protocol by a middleware server. The middleware provides connectivity to many different databases

The graphic below illustrates JDBC connectivity using ODBC drivers and existing database client libraries.

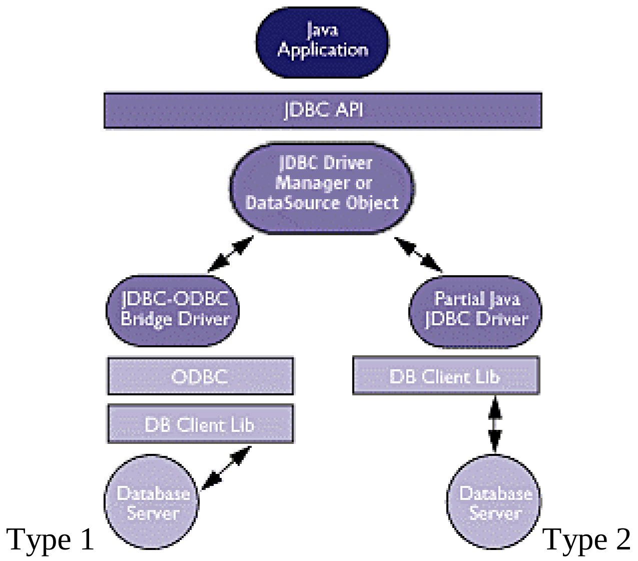


Figure 4: JDBC-ODBC drivers

Type 1: JDBC-ODBC Bridge plus ODBC Driver: This combination provides JDBC access via ODBC drivers. ODBC binary code--and in many cases, database client code--must be loaded on each client machine that uses a JDBC-ODBC Bridge. Sun provides a JDBC-ODBC Bridge driver, which is appropriate for experimental use and for situations in which no other driver is available.

Type 2: A native-API partly Java technology-enabled driver: This type of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

4.3 Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access. In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a *client/server* configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an intranet, which, for example, connects employees within a corporation, or it can be the Internet.

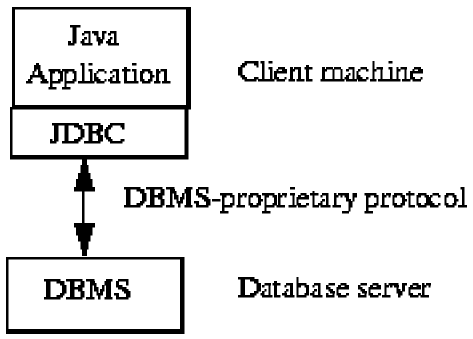


Figure 5. JDBC two-tier model

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

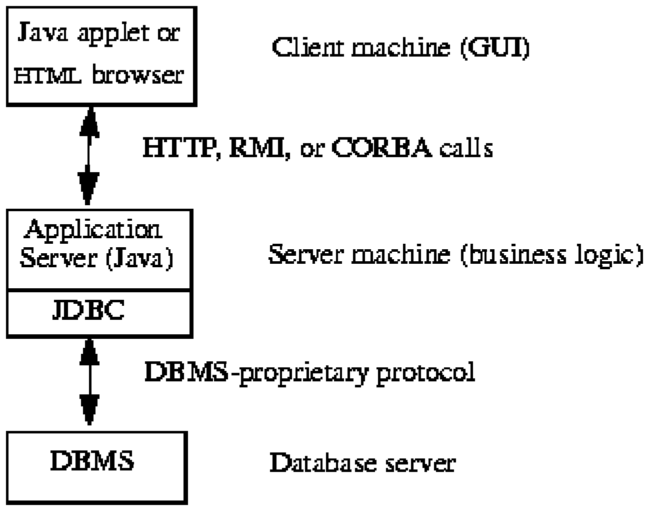

Java applet or HTML browser
Client machine (GUI)
HTTP, RMI, or CORBA calls
Application Server (Java)
JDBC
Server machine (business logic)
DBMS-proprietary protocol
DBMS
Database server

Figure 6. JDBC three-tier model

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java [8]. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features. JDBC is important to allow database access from a Java middle tier.

5. Project Overview

The number of online users increases day by day. One of the powerful aspects of the Web is information sharing. We can hardly find someone going online and not browsing the net. One of the fast growing areas of the Web is distance education (or distance learning). The reason distance education on the Web is getting popular is because it has advantages over other types of distance education programs. It gives much more flexibility to the users. The users can take the courses they registered for at any computer connected to the Internet. They usually have a more flexible time frame to take their classes and their tests.

In terms of programming of these sites developers had until recently to use limited range of technology to choose from. These technologies usually involved Common Gateway Interface (CGI) programming, Javascript, and Microsoft's Active Server Pages. This paper demonstrates that Java servlets and JDBC can be used programming for these sites.

This paper discusses how Java Servlets technology can be used to develop a dynamic web page. The goal is not to develop an 'Online Course'. We rather prefer to implement some features of online education. In the Computer Science Department at UNR, we have about 90 students in every semester for CS202. And it is becoming very time consuming for the TA to grade the assignment and quiz in every week. If we can take advantage of some features of online education, we can save a lot of time.

5. 1 Description of the Project

The web site is available only for the student, TA and the instructor of the course. Any of the above users can login to the secure site. If the user has the administrative privileges, he can do some administrative tasks such as add a user, view the list of users, build a quiz, grade a quiz, grade the assignment etc. Students can only submit quiz question, take a quiz, view quiz grade, submit assignment, change password etc. To implement all these features we have some servlets programs and a database program. The servlet generates the web page, gets information from the user, if necessary communicate with the database server via the database class. The database class is the middle tier between the servlet and database server. The following table tells the name of the servlet I am running for this project.

Servlet Name	**File Name**
Welcome	Welcome.java
AddUser	AddUser.java

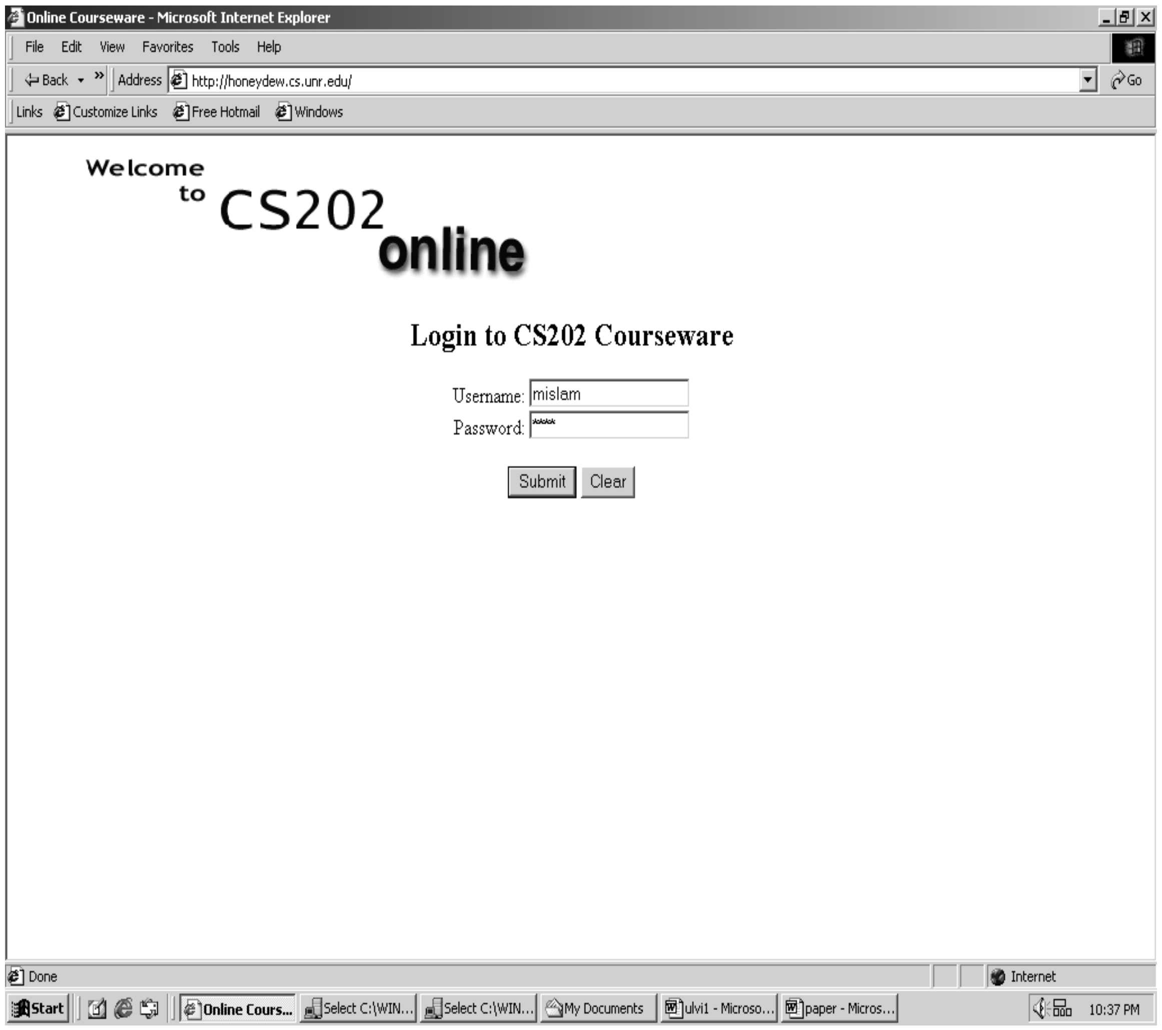
AddUserProcess	AddUserProcess.java
ChangePassword	ChangePassword.java
ChangePasswordProcess	ChangePasswordProcess.java
DeleteUser	DeleteUser.java
DeleteUserProcess	DeleteUserProcess.java
UserList	UserList.java
BuildQuiz	BuildQuiz.java
BuildQuizProcess	BuildQuizProcess.java
BuildMCPart	BuildMCPart.java
BuildTFPart	BuildTFPart.java
BuildTXPart	BuildTXPart.java
GradeQuiz	GradeQuiz.java
GradeQuizProcess	GradeQuizProcess.java
InsertGrade	InsertGrade
InsertQue	InsertQue.java
InsertQueProcess	InsertQueProcess.java
InsertMCQue	InsertMCQue.java
InsertTFQue	InsertTFQue.java
InsertTXQue	InsertTXQue.java
Quiz	Quiz.java

QuizProcess	QuizProcess.java
Answer	Answer.java
FormUpload	FormUpload.java
UploadServlet	UploadServlet.java

Table 1: List of servlets

From the above list, we will talk about some of the important servlets. We will skip some servlets are used to generate only for HTML Forms.

Welcome.java: This servlet that gets the username and password from a HTML Form. It calls *loginCheck()* method of database class, and matches them with the username and password in the database. If the user is valid, then it calls *adminCheck()* method of the database to see if the user has the admin privileges. Then it directs the user to a different page according to the user type. Also, it generates a unique session object, which contains the username and passes it to the next servlet.

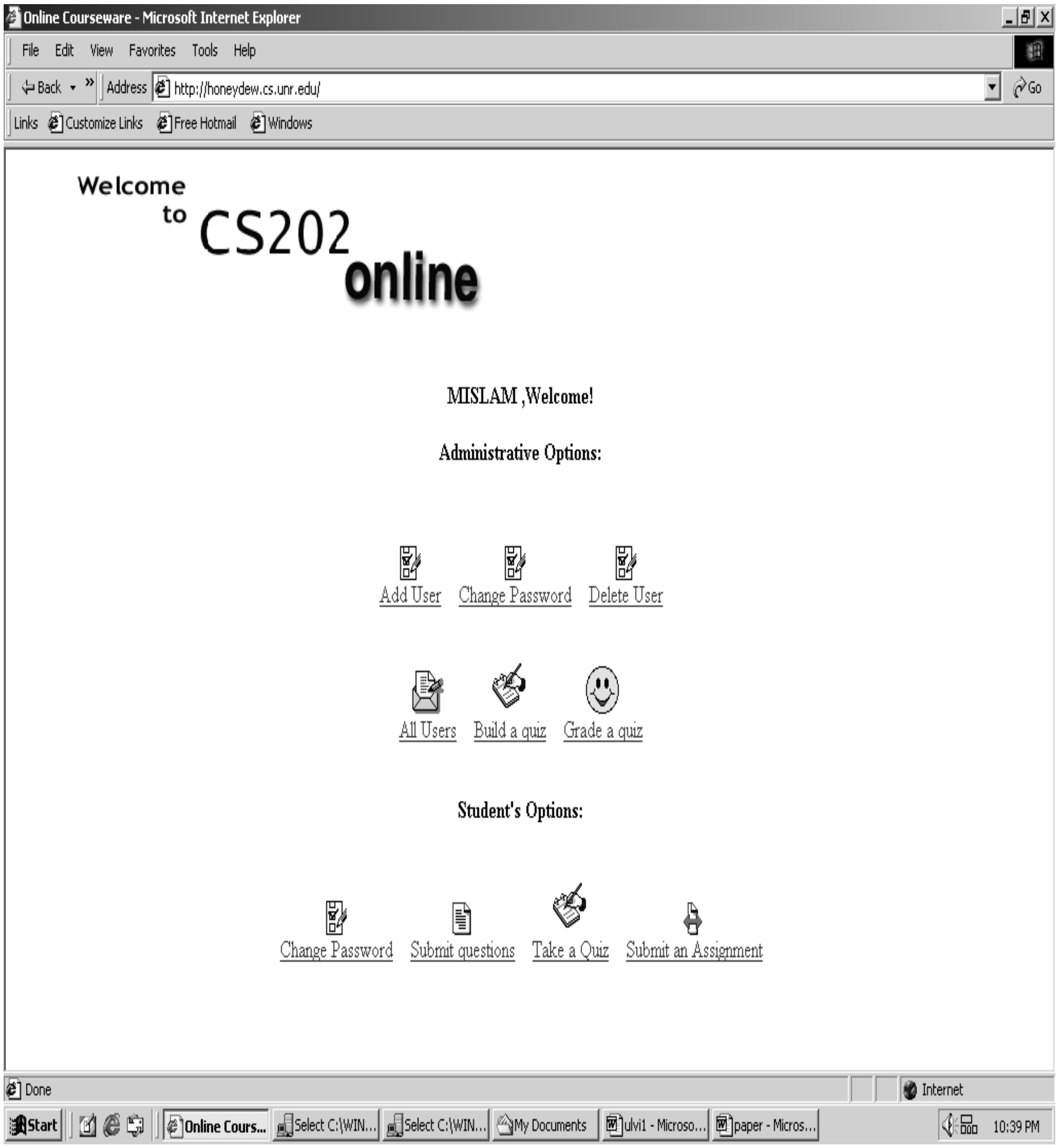

Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online

Login to CS202 Courseware

Username: mislam
Password: ****

Submit
Clear
Done
Internet
Start
Online Cours...
Select C:\WIN...
Select C:\WIN...
My Documents
ulvi1 - Microso...
paper - Micros...
10:37 PM

Figure 7: The login screen that connects to Welcome.java


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done
Internet
Start
Online Cours...
Select C:\WIN...
Select C:\WIN...
My Documents
ulvi1 - Microso...
paper - Micros...
10:39 PM

MISLAM ,Welcome!

Administrative Options:

Add User Change Password Delete User

All Users Build a quiz Grade a quiz

Student's Options:

Change Password Submit questions Take a Quiz Submit an Assignment

Figure 8: Options for TA/Instructor generated by Welcome.java

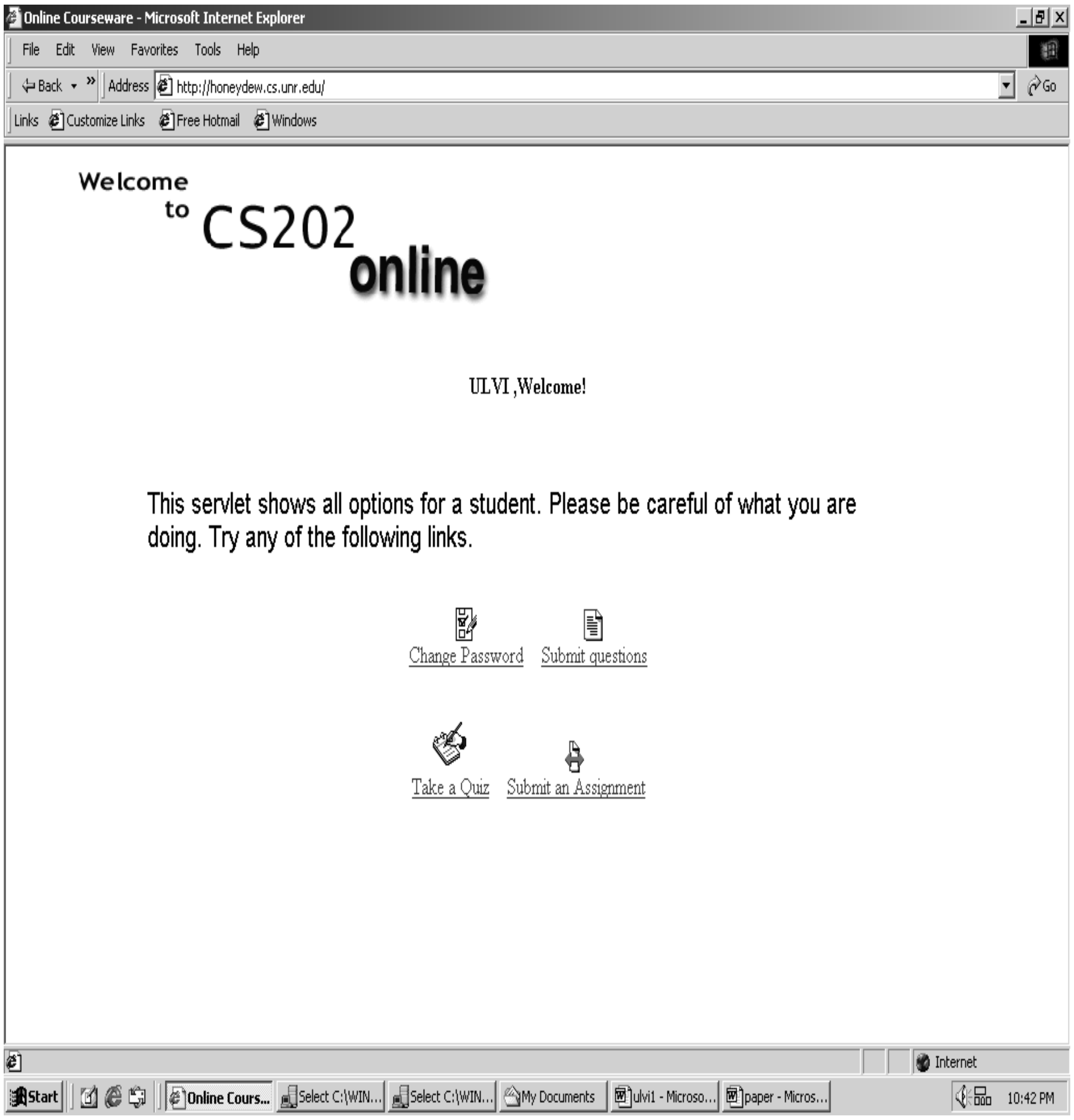
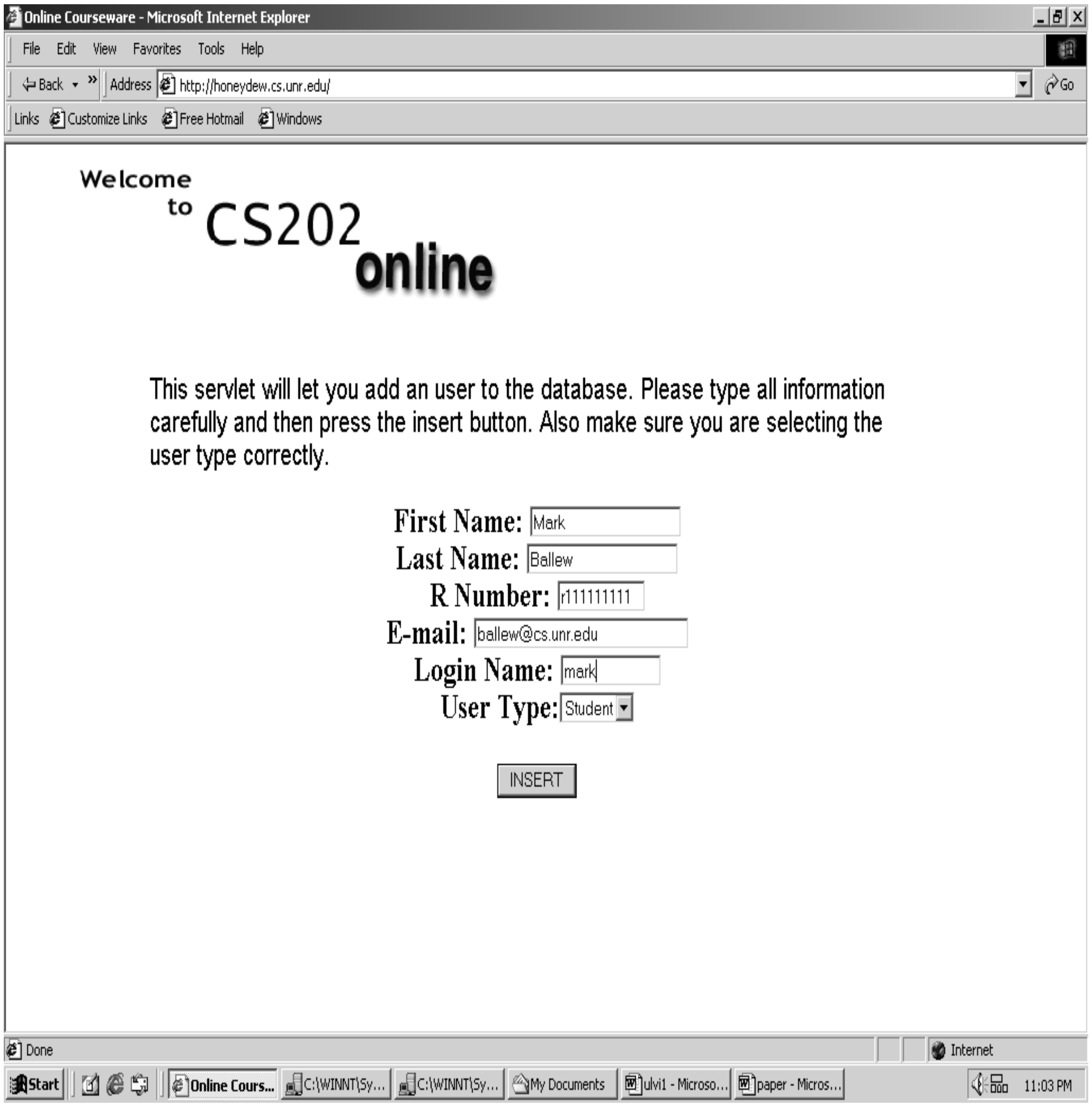

This servlet shows all options for a student. Please be careful of what you are doing. Try any of the following links.

Figure 9: Options for Students generated by Welcome.java

AddUserProcess.java: This servlet gets the user information from the form generated by *AddUser.java*. Then it calls *newEntry()* to add the user in the database.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online

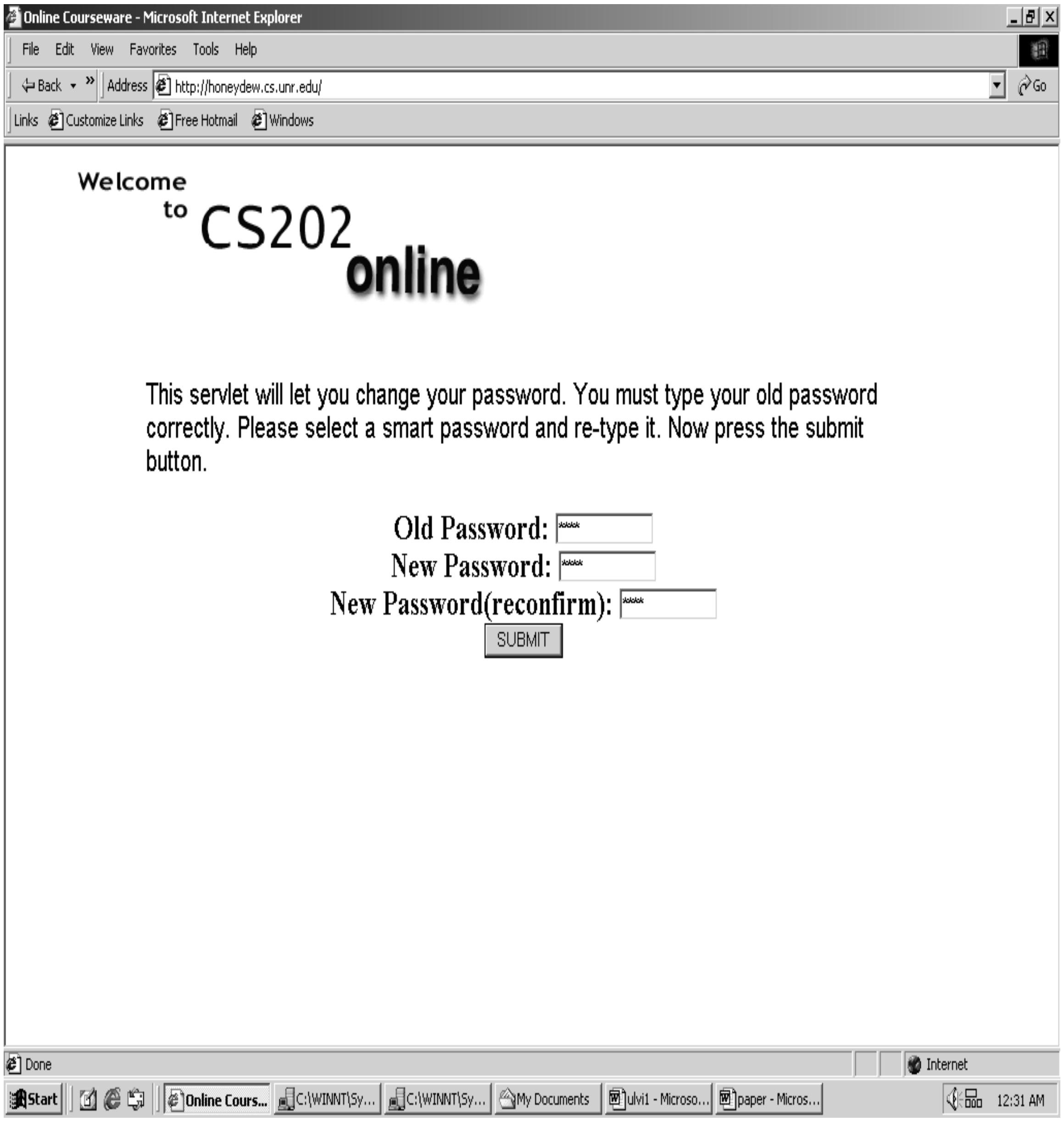
This servlet will let you add an user to the database. Please type all information carefully and then press the insert button. Also make sure you are selecting the user type correctly.

First Name: Mark
Last Name: Ballew
R Number: r111111111
E-mail: ballew@cs.unr.edu
Login Name: mark
User Type: Student

INSERT
Done
Internet
Start Online Cours... C:\WINNT\Sy... C:\WINNT\Sy... My Documents ulvi1 - Microso... paper - Micros... 11:03 PM

Figure 10: Adding an user

ChangePasswordProcess.java: This servlet gets the password information from the *ChangePasswd.java*. The user has to type the new password twice. Ihe new password has to match before it calls the *changePassword()* method of the database class.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online

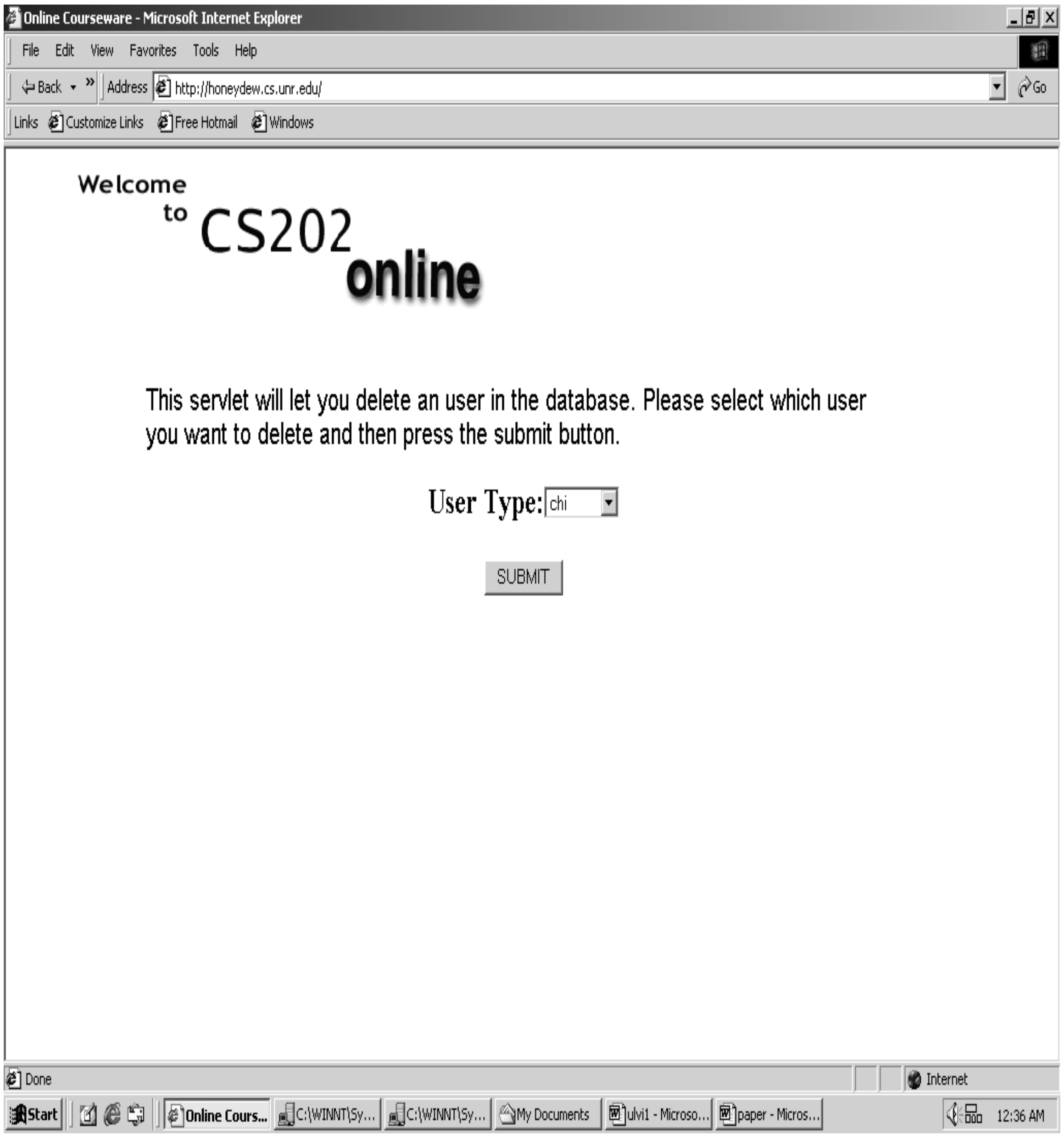
This servlet will let you change your password. You must type your old password correctly. Please select a smart password and re-type it. Now press the submit button.

Old Password: ****
New Password: ****
New Password(reconfirm): ****
SUBMIT

Done
Internet
Start Online Cours... C:\WINNT\Sy... C:\WINNT\Sy... My Documents ulvi1 - Microso... paper - Micros... 12:31 AM

Figure 11: Changing password

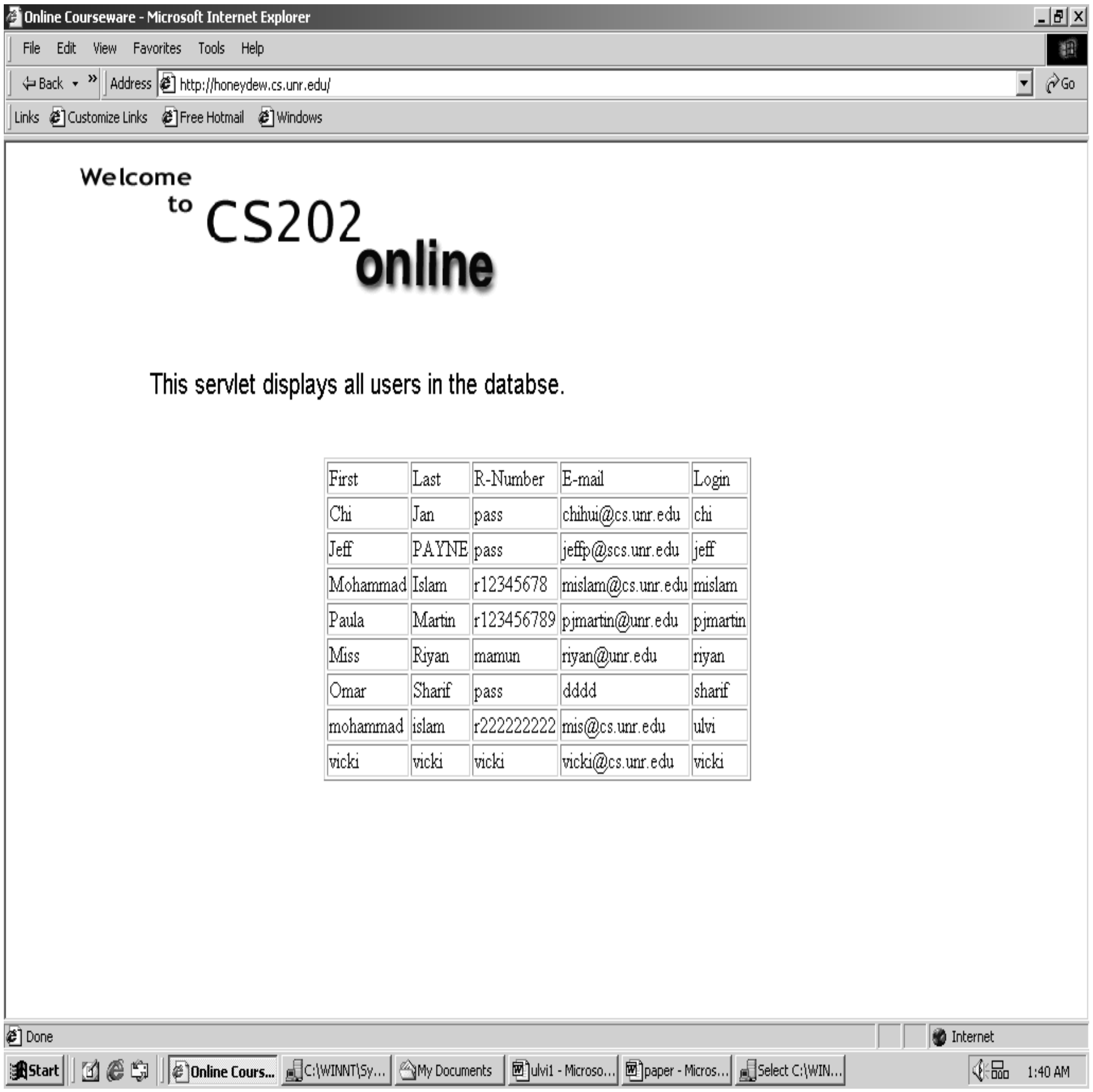
DeleteUserProcess.java: This servlet gets the user information from the form generated by *DeleteUser.java*. We have drop down list from where we can select which user will be deleted. It gets the user name and calls *deleteUser()* method of the database class to delete the user.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
User Type: chi
SUBMIT
Done
Internet
Start
12:36 AM

This servlet will let you delete an user in the database. Please select which user you want to delete and then press the submit button.

Figure 12: Deleting an user

UserList.java: This servlet calls a method *displayUser()* of the database class and show the users on the screen.

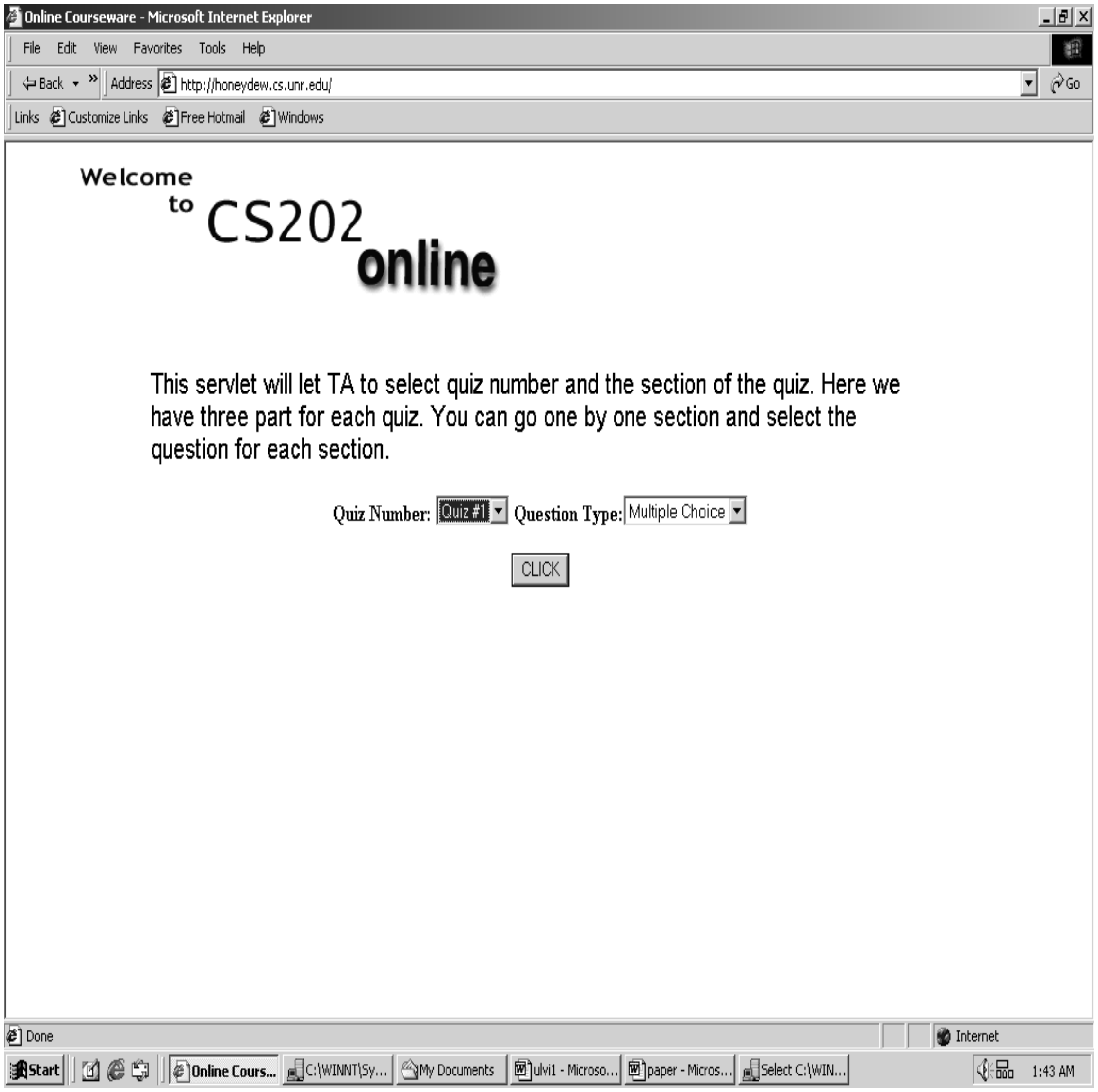

Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done
Internet
Start Online Cours... C:\WINNT\Sy... My Documents ulvi1 - Microso... paper - Micros... Select C:\WIN... 1:40 AM

This servlet displays all users in the databse.

First	Last	R-Number	E-mail	Login
Chi	Jan	pass	chihui@cs.unr.edu	chi
Jeff	PAYNE	pass	jeffp@scs.unr.edu	jeff
Mohammad	Islam	r12345678	mislam@cs.unr.edu	mislam
Paula	Martin	r123456789	pjmartin@unr.edu	pjmartin
Miss	Riyan	mamun	riyan@unr.edu	riyan
Omar	Sharif	pass	dddd	sharif
mohammad	islam	r222222222	mis@cs.unr.edu	ulvi
vicki	vicki	vicki	vicki@cs.unr.edu	vicki

Figure 13: List of the users

BuildQuizProcess.java: This servlet gets the quiz information from the previous servlet. The quiz question could be three types such as multiple choice, true/false, and text type. This servlet will display the entire quiz question we have in the database for this particular quiz. The instructor can build any part of the quiz at a time. It calls *displayMC(), displayTF(), displayTX(), chapterCheckMC(), chapterCheckTX(), chapterCheckTF()* methods of the database class.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done
Internet
Start
Online Cours...
C:\WINNT\Sy...
My Documents
ulvi1 - Microso...
paper - Micros...
Select C:\WIN...
1:43 AM

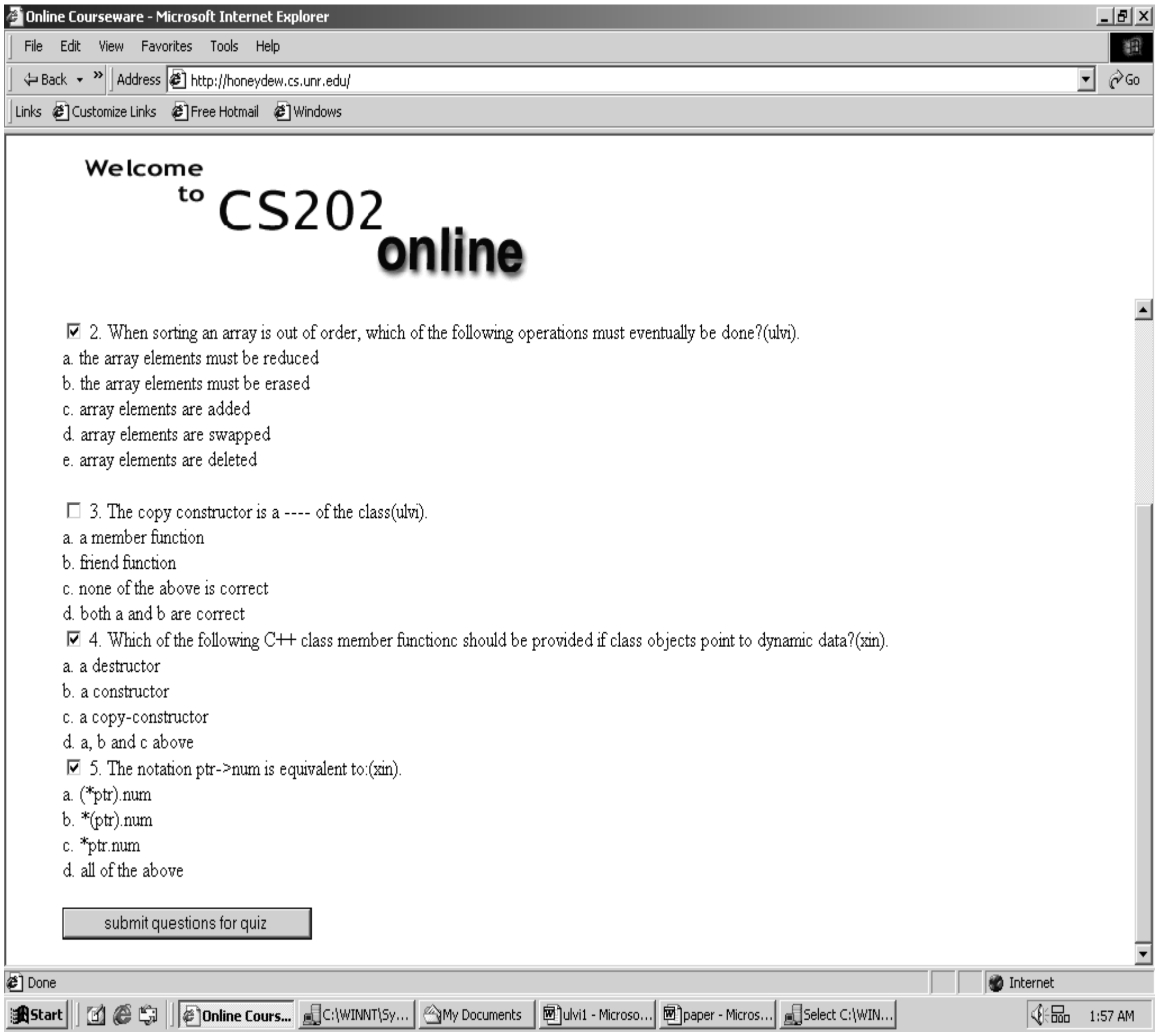
This servlet will let TA to select quiz number and the section of the quiz. Here we have three part for each quiz. You can go one by one section and select the question for each section.

Quiz Number: Quiz #1 Question Type: Multiple Choice

CLICK

Figure 14: Building a new quiz

BuildMCPart.java/ BuildTFPart.java/ BuildTXPart.java: This servlet gets the selected question, point for each question from the previous servlet and checks if the question exists in the database for this particular quiz by calling *q_check()* method of the database. It either displays an error message when the question is in the database or inserts the question by calling *newQuiz()* method of the database class.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done
Internet
Start
Online Cours...
C:\WINNT\Sy...
My Documents
ulvi1 - Microso...
paper - Micros...
Select C:\WIN...
1:57 AM

☑ 2. When sorting an array is out of order, which of the following operations must eventually be done?(ulvi).
a. the array elements must be reduced
b. the array elements must be erased
c. array elements are added
d. array elements are swapped
e. array elements are deleted

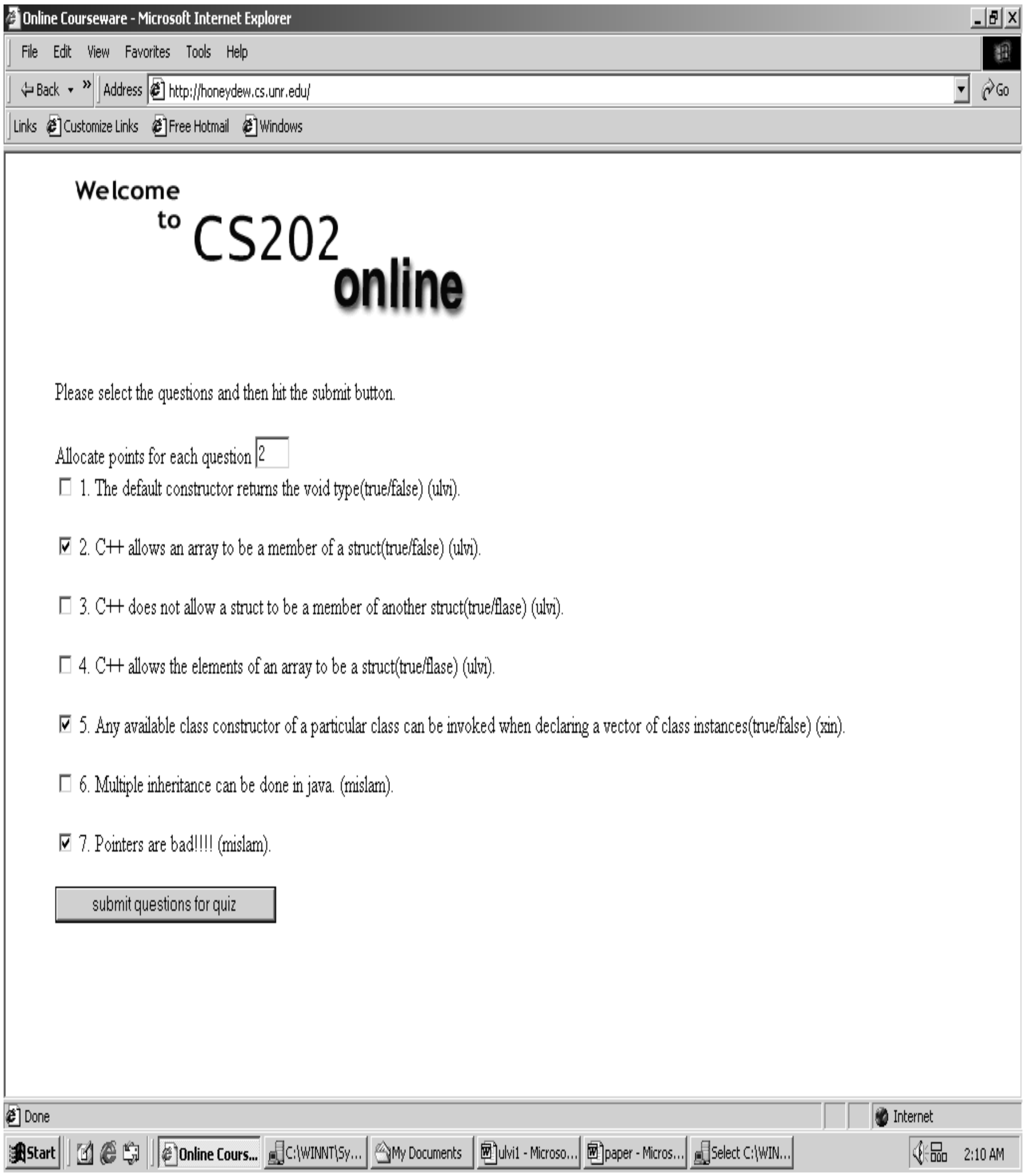
☐ 3. The copy constructor is a ---- of the class(ulvi).
a. a member function
b. friend function
c. none of the above is correct
d. both a and b are correct

☑ 4. Which of the following C++ class member functionc should be provided if class objects point to dynamic data?(xin).
a. a destructor
b. a constructor
c. a copy-constructor
d. a, b and c above

☑ 5. The notation ptr->num is equivalent to:(xin).
a. (*ptr).num
b. *(ptr).num
c. *ptr.num
d. all of the above

submit questions for quiz

Figure 15: Selecting multiple choice questions for a new quiz

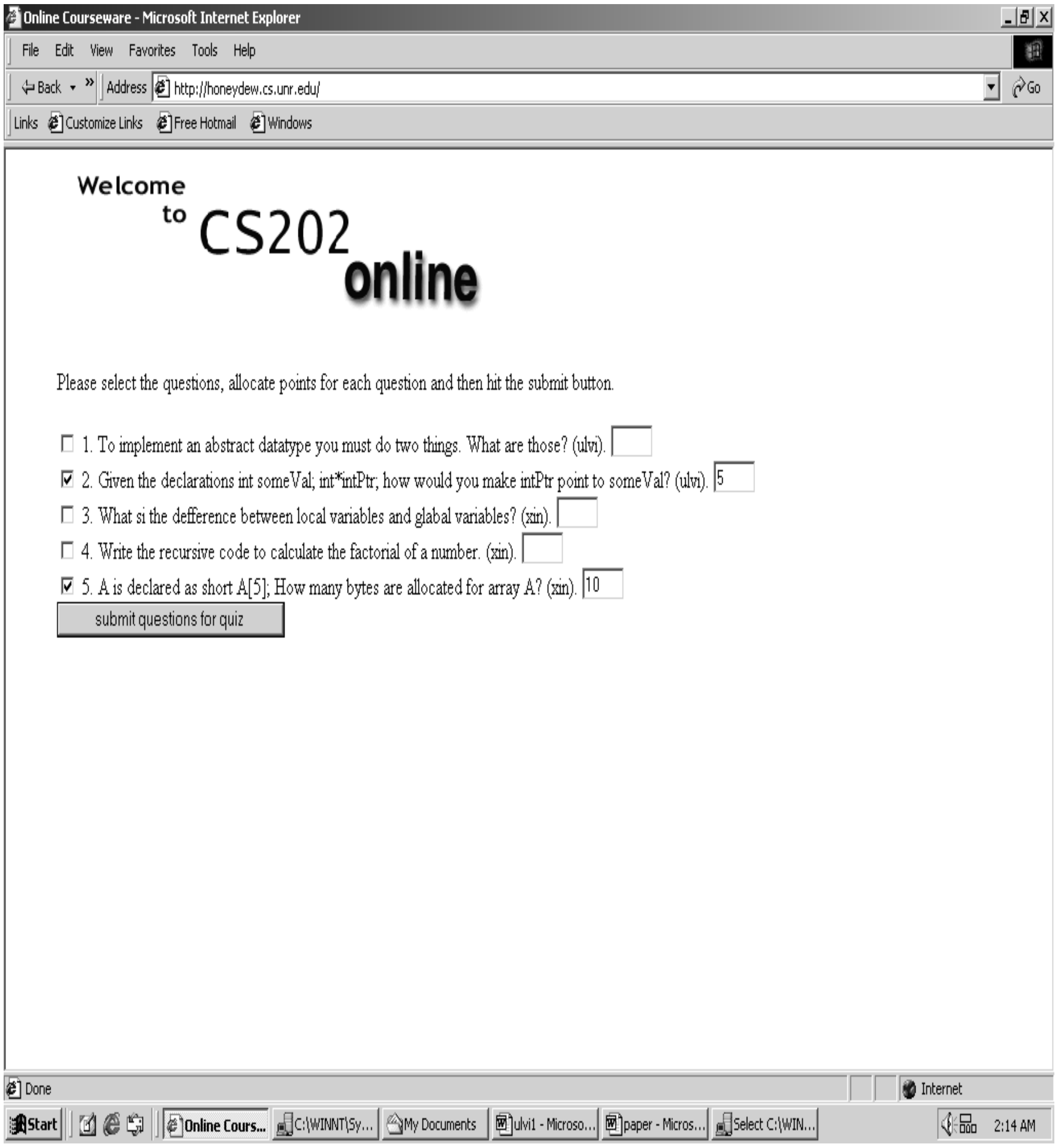

Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online

Please select the questions and then hit the submit button.

Allocate points for each question 2

- [ ] 1. The default constructor returns the void type(true/false) (ulvi).
- [x] 2. C++ allows an array to be a member of a struct(true/false) (ulvi).
- [ ] 3. C++ does not allow a struct to be a member of another struct(true/flase) (ulvi).
- [ ] 4. C++ allows the elements of an array to be a struct(true/flase) (ulvi).
- [x] 5. Any available class constructor of a particular class can be invoked when declaring a vector of class instances(true/false) (xin).
- [ ] 6. Multiple inheritance can be done in java. (mislam).
- [x] 7. Pointers are bad!!!! (mislam).

Figure 16: Selecting true/false questions for a new quiz


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://honeydew.cs.unr.edu/
Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done
Internet
Start
2:14 AM

Please select the questions, allocate points for each question and then hit the submit button.

☐ 1. To implement an abstract datatype you must do two things. What are those? (ulvi).

☑ 2. Given the declarations int someVal; int*intPtr; how would you make intPtr point to someVal? (ulvi). 5

☐ 3. What si the defference between local variables and glabal variables? (xin).

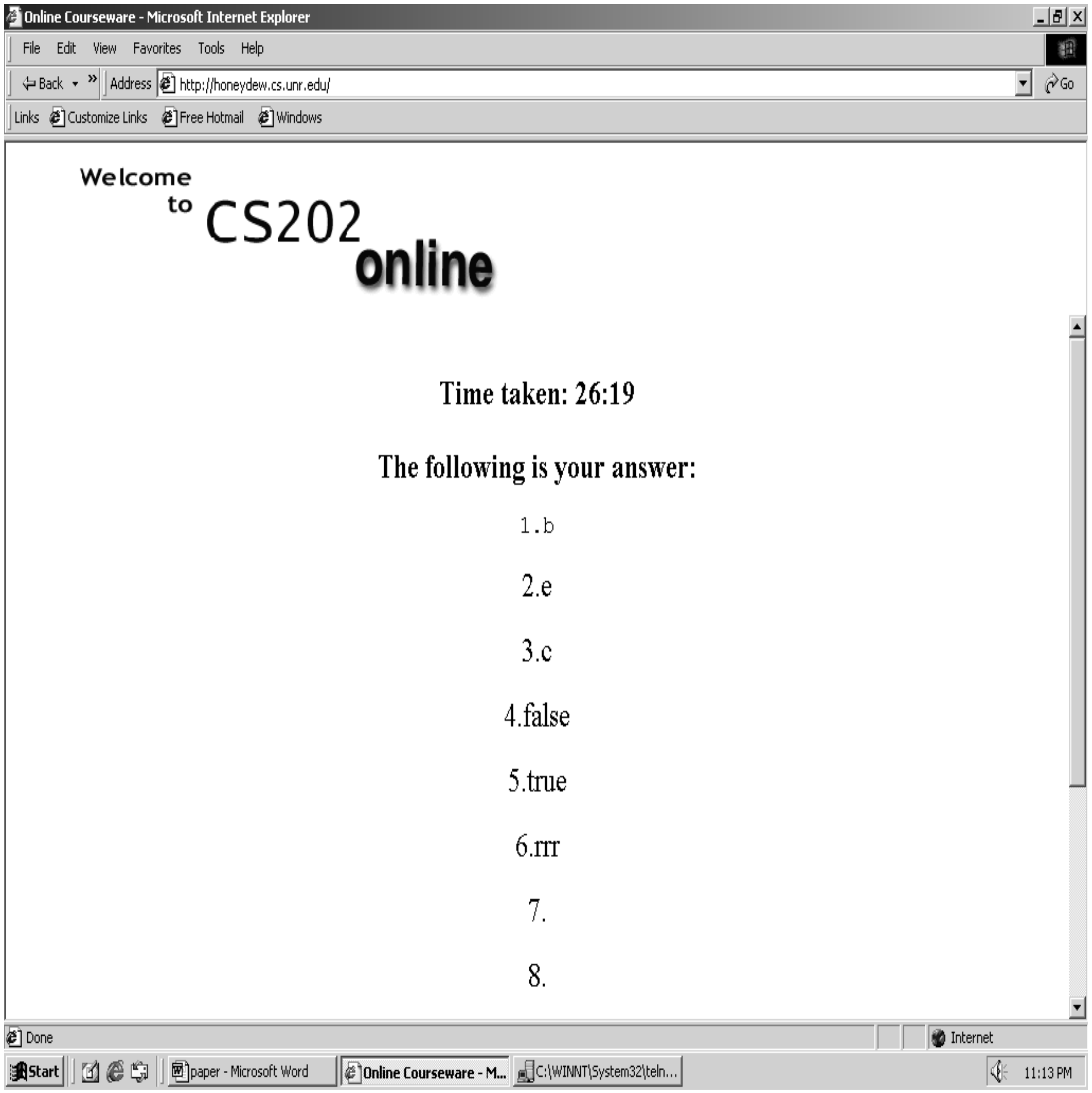
☐ 4. Write the recursive code to calculate the factorial of a number. (xin).

☑ 5. A is declared as short A[5]; How many bytes are allocated for array A? (xin). 10

submit questions for quiz

Figure 17: Selecting text questions for a new quiz

GradeProcess.java: This servlet gets the information from the *GradeQuiz.java* and calls *displayAnswer()* method of the database class to show the answer of the quiz submitted by the student. It has a text box for TA/Instructor to insert the grade and then it connects to *InsertGrade.java* that will insert the grade in the grade table by calling *insertGrade()*. Before insertion it checks if the student was graded before.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done Internet
Start paper - Microsoft Word Online Courseware - M... C:\WINNT\System32\teln... 11:13 PM

Time taken: 26:19

The following is your answer:

1.b

2.e

3.c

4.false

5.true

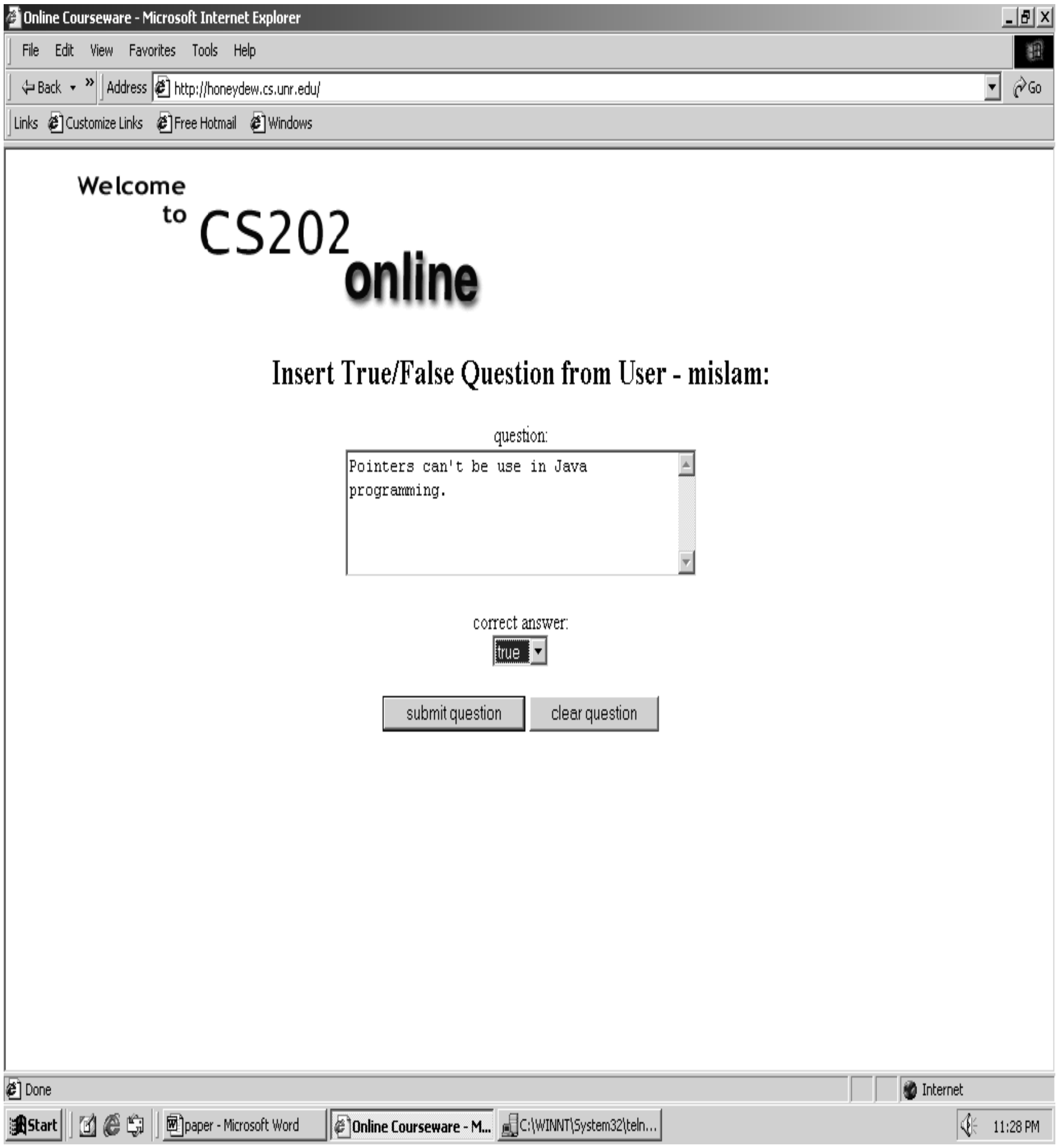
6.rrr

7.

8.

Figure 18: Grading a quiz

InsertQueProcess.java: This servlet will pop up different type form depending on the question type. It can connect to any of the servlet i.e. *InsertMCQue.java, InsertTFQue.java, InsertTXQue.java.* One of the above servlets will insert the question in the database by calling *newMCEntry()*, or *newTFEntry()*, or *newTXEntry()* method of the database class.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online

Insert True/False Question from User - mislam:

question:

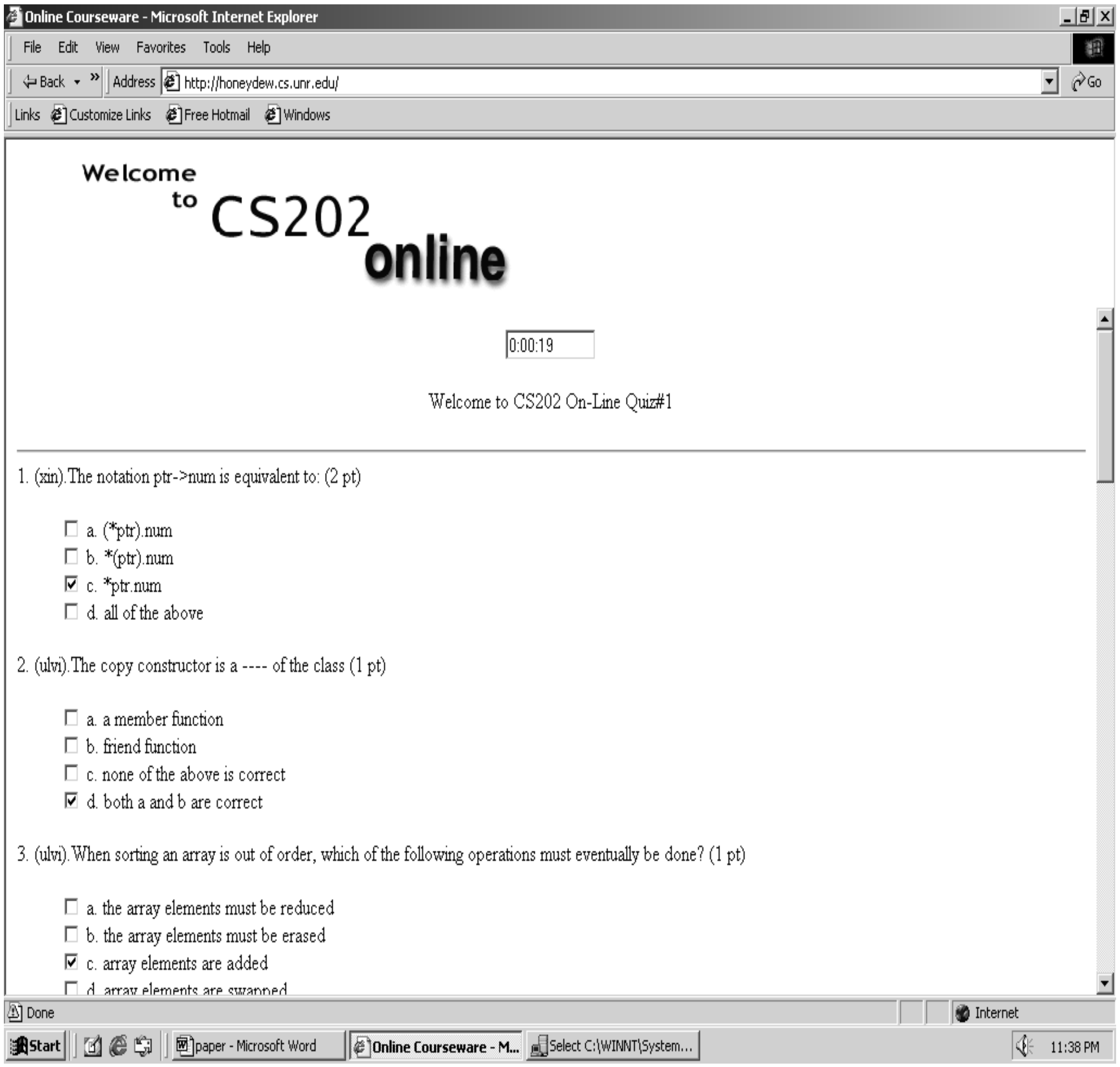
```
Pointers can't be use in Java
programming.
```

correct answer: true

submit question
clear question
Done
Internet
Start
paper - Microsoft Word
Online Courseware - M...
C:\WINNT\System32\teln...
11:28 PM

Figure 19: Quiz question Submission

QuizProcess.java: This servlet will let the student to take the quiz. It gets the quiz information from the *Quiz.java* servlet. It calls *displayMC(), displayTF(), displayTX()* methods of the database class to display the quiz on the screen. It also generates a clock to monitor the time, a student spending to take the quiz. After student takes the quiz, it connects to the Answer.java to insert the student's answers.


Online Courseware - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Address http://honeydew.cs.unr.edu/ Go
Links Customize Links Free Hotmail Windows
Welcome to CS202 online
Done Internet
Start paper - Microsoft Word Online Courseware - M... Select C:\WINNT\System... 11:38 PM

0:00:19

Welcome to CS202 On-Line Quiz#1

1. (xin).The notation ptr->num is equivalent to: (2 pt)

- ☐ a. (*ptr).num
- ☐ b. *(ptr).num
- ☑ c. *ptr.num
- ☐ d. all of the above

2. (ulvi).The copy constructor is a ---- of the class (1 pt)

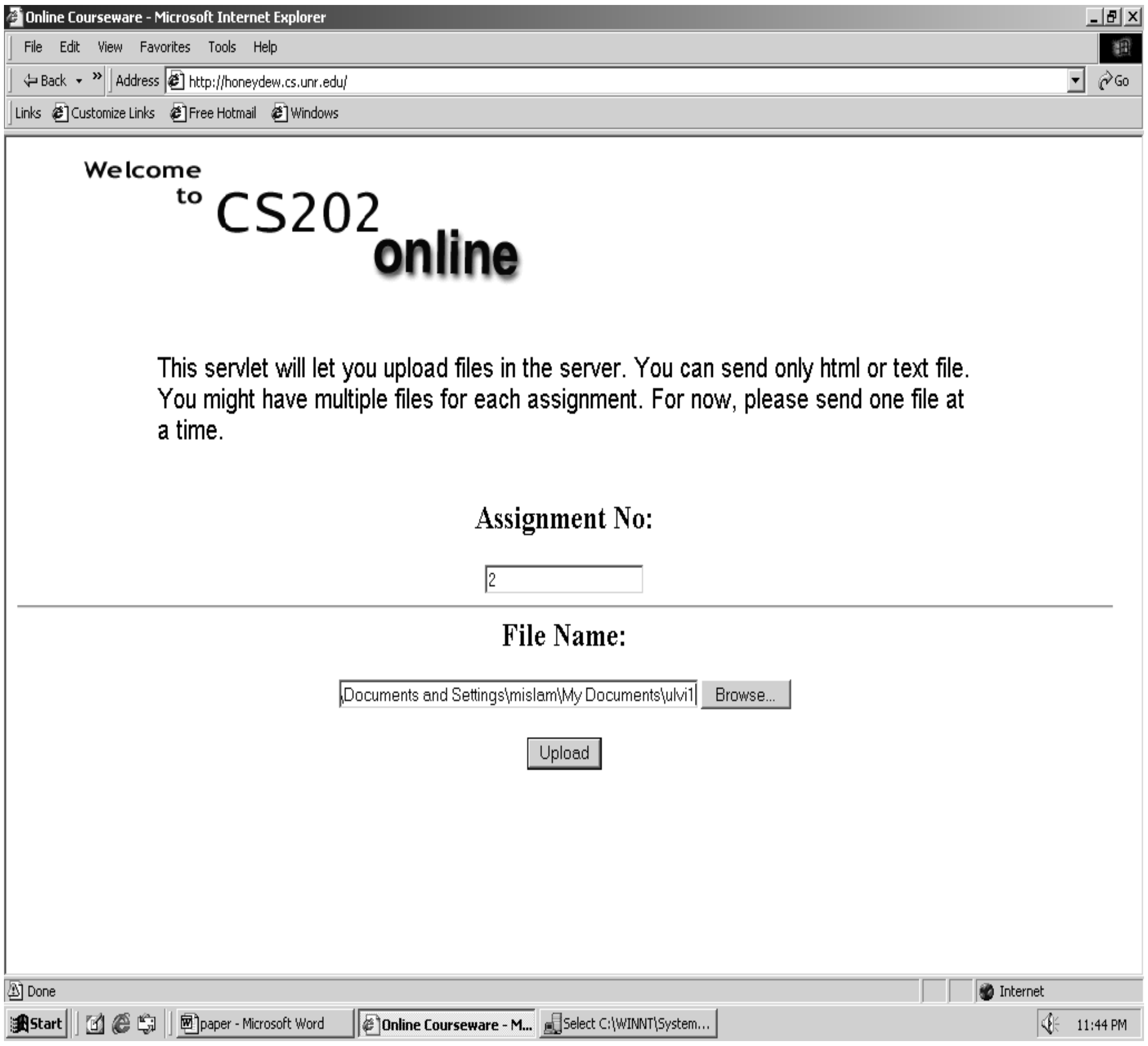
- ☐ a. a member function
- ☐ b. friend function
- ☐ c. none of the above is correct
- ☑ d. both a and b are correct

3. (ulvi).When sorting an array is out of order, which of the following operations must eventually be done? (1 pt)

- ☐ a. the array elements must be reduced
- ☐ b. the array elements must be erased
- ☑ c. array elements are added
- ☐ d. array elements are swapped

Figure 20: Student taking the quiz

UploadServlet.java: This servlet gets the file information from the *FormUpload.java* and uploads the file in the `/home/mislam/cs202/username/assignment# directory`.


Online Courseware - Microsoft Internet Explorer

http://honeydew.cs.unr.edu/

Welcome to CS202 online

This servlet will let you upload files in the server. You can send only html or text file. You might have multiple files for each assignment. For now, please send one file at a time.

Assignment No:

2

File Name:

Documents and Settings\mislam\My Documents\ulvi1

Fig21: File Upload

6. Conclusion

Servlets are a powerful embodiment of Java Philosophy, which are only now gaining attention. Servlets have some unique features: platform independence, ease of implementation, performance, threading, session control, and HTTP specific characteristics that will grow to displace CGI-Bin over time.

This study implements some features of online course design. It shows how Servlet technology could be used for automatic quiz generation, session tracking, file submission, grading assignment and quiz etc. A tree tier database model was implemented to store the information.

Servlets are very useful for providing dynamic content to users. The solidification of XML and the demand forever more sophisticated content will drive the advance and acceptance of Servlet technology.

7. Future Developments

This project provides the basic structure for an online courseware application. Future additions can easily be added to it to expand its functionality and operation. The additions we have planned for it include the following:

- Online assignment grading.
- Improved security and authentication scheme.

The more complex of these additions is the online assignment grading. This will include execution of a student's assignment through the use of a server side script, and automatic return of the output of the assignment in HTML. The grader will then be able to view both the source code and the output from the script at the same time. The grader will also be provided with a way to logically attach comments to the source code by referencing the line numbers. The comments will be kept in a text file available to the student for review.

The security and authentication scheme could be improved rather easily through the use of a more advanced encryption algorithm and a secure connection to the server. Since there is no ultra-sensitive information, i.e. social security numbers or credit card numbers, being transferred the secure connection is probably not as important as improving the encryption algorithm.

These short-term improvements could be made fairly easily, but there is also a question of long-term improvements. For instance, at the moment this application is not very portable to other fields of study besides Computer Science. A truly useful online courseware application would be portable or at least extendable to other areas of study. Improvements on this scale would require an almost complete rewrite of the application.

References:

[1] Dustin R. Callaway. *Inside Servlets: Server-Side Programming for the Java^{TM} Platform*. Addison Wesley Longman, Inc., Redaing, Massachusetts, 1999.

[2] David Flanagan. *Java In A Nutshell*, *second edition*. O'Reilly and Associates, Inc., Sebastopol, CA, 1997.

[3] James Goodwill. *Developing Java Servlets*. Sams Publishing, Indianapolis, IN, 1999.

[4] Jason Hunter and William Crawford. *Java Servlet Programming*. O'Reilly and Associates, Inc., Sebastopol, CA, 1998.

[5] William E. Weinman. *The CGI Book*. New Riders Publication, Indianapolis, IN, 1996.

[6] URL: http://java.sun.com/products/jdbc/datasheet.html current as of August 24, 2000.

[7] URL: http://www.online.edu current as of August 24, 2000.

[8] URL: http://java.sun.com/products/jdbc/datasheet.html current as of August 24, 2000.

[9] URL: http://www.webadvisor.com/jdbc.html current as of August 24, 2000.