White Board

Getahun Alemu

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd-project

Part of the Databases and Information Systems Commons

Recommended Citation
https://scholarworks.lib.csusb.edu/etd-project/2195

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
WHITE BOARD

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Getahun Alemu

June 2003
WHITE BOARD

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Getahun Alemu
June 2003

Approved by:

[Signature]
Josephine Mendoza, Chair, Computer Science

[Signature]
Richard Botting

[Signature]
Ernesto Gomez

6-4-03
Date
ABSTRACT

This project designs and implements a tool to enhance the current means of availing coursework information in educational systems. The project addresses current issues in availing coursework information like, decentralized locations of coursework information, handling different types of questions, randomization of coursework contents, classification of questions by topic, and sharing of questions among instructors. The project encompasses review of related application tools to address and identify the limitations that exist compared to the common day-to-day activities that are inherent in the teaching environment. The new tool is developed using a component-based methodology and has a capability of being installed as client/server application and as a web-based application with minimal configuration setups as opposed to complete change of software program. The tool utilizes the state-of-the-art Oracle database system to store coursework data, Oracle Internet Application Server (OiAS) to process the logic and setup the communication media between the client and server application. The user interface is implemented using Oracle Forms 6i tool. The product has undergone two
test methods: a Unit test plan where modules are tested separately and an integration test plan where modules are tested as one complete unit after integration. The new tool incorporates additional features which includes capabilities such as: Sharing of questions amongst instructors, classification of questions by topic and complexity, randomization of course work contents, and user friendly iconic coursework status indicators.
ACKNOWLEDGMENTS

I would like to thank Dr. Josephine Mendoza, my advisor, for her strong assistance and valuable advise to make this project a reality. In addition, I would like to extend my gratitude to my committee members Dr. Botting and Dr. Gomez for their helpful insight and Mr. Ken Han for his advice and support in hardware and installation issues.

I would also like to thank my sister Aida and my brother-in-law Tamrat for providing me strong support and dedication without which I would not have been able to complete my graduate studies.

The support of the National Science Foundation under award 9810708 is gratefully acknowledged.
TABLE OF CONTENT

ABSTRACT .......................................................... iii
ACKNOWLEDGMENTS .................................................. v
LIST OF TABLES ...................................................... ix
LIST OF FIGURES ..................................................... x

CHAPTER ONE: SOFTWARE REQUIREMENTS SPECIFICATION

1.1 Introduction ...................................................... 1
1.2 Purpose of the Project ........................................... 5
1.3 Context of the Problem .......................................... 5
1.4 Significance of the Project ...................................... 5
1.5 Preliminary Design .............................................. 6
   1.5.1 Description of Use Case Diagram .................... 6
1.6 Assumptions ..................................................... 8
1.7 Limitations ..................................................... 9
1.8 Definition of Terms ........................................... 9

CHAPTER TWO: SOFTWARE DESIGN

2.1 Architecture .................................................... 11
   2.1.1 Client Tier ................................................. 11
   2.1.2 Middle Tier ............................................... 12
   2.1.3 Database Tier .............................................. 13
   2.1.4 Communication Process .................................. 13
2.2 Problem Description ........................................... 15
   2.2.1 Problem Description ...................................... 16
   2.2.2 Database Design .......................................... 18
2.3 Detailed Design .................................................. 19
  2.3.1 Stored Program Packages ............................. 20
  2.3.2 Triggers ..................................................... 20
  2.3.3 Database Procedures and Packages ........... 20

CHAPTER THREE: MAINTENANCE

  3.1 Installation and Configuration of Application Servers .................. 22
    3.1.1 Forms Server Installation and Configuration ..................... 22

  3.2 Secure Shell Applets and Secured File Transfer Protocol .............. 25

CHAPTER FOUR: USERS MANUAL

  4.1 Login Interface ............................................ 26
  4.2 Main Interface .............................................. 26
    4.2.1 Main Menu ............................................ 27
    4.2.2 Pop-Up Menu .......................................... 29

  4.3 WORK_ON Interface .......................................... 33
  4.4 SOLUTION_BLK Interface .................................... 36
  4.5 Comment Interface .......................................... 39
  4.6 CREATE_QUESTION Interface .................................. 40
  4.7 Load_Question Interface .................................... 44
  4.8 Register_Exercise Interface .................................. 44
  4.9 Add_Exerno Interface ........................................ 45
  4.10 Register>User Interface ..................................... 47
  4.11 Control Interface ........................................... 49
CHAPTER FIVE: CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusion ........................................ 50
5.2 Future Directions ............................... 51

5.2.1 Grade Availability through Wireless Technology ................. 51
5.2.2 Artificial Intelligent Capability ..... 51
5.2.3 Rich Text Questions ....................... 51
5.2.4 Answer Statistics ......................... 52

APPENDIX A: DATABASE TABLES ....................... 53
APPENDIX B: DATA DICTIONARY ....................... 56
APPENDIX C: CONFIGURATION FILES ................... 61
APPENDIX D: TEMPLATE FOR QUESTION FILES .............. 66
REFERENCES ............................................. 68
LIST OF TABLES

Table 1. Comparison Table .................................. 4
LIST OF FIGURES

Figure 1. Use Case Diagram ............................................. 7
Figure 2. Application Architecture ........................................ 11
Figure 3. Communication Process ............................................. 13
Figure 4. Conceptual Model Diagram ......................................... 15
Figure 5. Extended Entity Relationship Diagram ......................... 19
Figure 6. Login Interface ..................................................... 26
Figure 7. Main Interface ....................................................... 27
Figure 8. Pop-Up Menu I ....................................................... 30
Figure 9. Pop-Up Menu II ....................................................... 31
Figure 10. WORK_ON Interface ................................................ 34
Figure 11. SOLUTION_Blk Interface .......................................... 36
Figure 12. Comment Interface .................................................. 39
Figure 13. Create_Question Interface ........................................ 41
Figure 14. Register_Exercise Interface ..................................... 44
Figure 15. Add_Exercise_No Interface ...................................... 46
Figure 16. Register_User Interface .......................................... 48
Figure 17. Control Interface .................................................... 49
CHAPTER ONE
SOFTWARE REQUIREMENTS
SPECIFICATION

1.1 Introduction

The development of the Internet has enabled many applications to be accessed and be shared amongst users located at different geographical locations. It has created an environment where remoteness is no longer a barrier to accessing information. Despite the current trend of moving businesses on to the Internet, information retrieval and sharing on the Internet is particularly evident in the educational system.

This project designs and implements a tool to enhance the current means of accessing coursework information via the Internet.

Instructors are currently utilizing the Internet to share information, post research papers, and distribute instructional materials. As can very well be seen, the course work of each section a student takes is available on a specific site where the instructor posts it (usually in the instructor’s web page). This has a number of limitations as outlined below:
1. Students have to visit a number of sites to get the information pertaining to the course work of each course they are taking.

2. Sharing of course work information amongst instructors can only be made by the cut and paste method from/to respective pages.

3. Grade information can be posted but has to be done manually.

Black Board (Bb), WebCT, and Computer-Based Training (CBT) are being used widely by educational institutions, and are designed to overcome the limitations mentioned above, including additional features. CSUSB is one institution that is using Bb. The following information is gathered from Mr. Michael Casadonte who was the Coordinator of The Instructional Materials in Office of Distributed Learning (ODL) in CSUSB.

1. Bb, a very big software system, requires instructors to attend a training on how to use it.

2. Bb incorporates a number of features, like chat facilities, which are not often utilized by students.
3. Intensive support is being provided for instructors.

4. Even though file transfer feature is incorporated in Bb, there is no feature included which enables computer science students to remotely login and access to the file server located in the computer science department enabling them to work on course assignments/projects.

5. Bb, WebCT, and CBT are fully web-based application. They cannot work on client/server architecture as they are coded in HTML.

The detailed comparison of the existing application software (Bb, WebCT) and the proposed system (WB) is tabulated in Table 1.
Table 1. Comparison Table

<table>
<thead>
<tr>
<th>Features Supported</th>
<th>Bb</th>
<th>WebCT</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Question Types</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple Choice</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Essay Type</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>True/ False</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td><strong>Classification of questions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>By topic</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>By complexity</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td><strong>Question uploading</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>from file</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>with/without answer correct answers</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td><strong>question ownership (sharing)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>private</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>public</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td><strong>Coursework preparation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>random selection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual selection</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>from file (mixed)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Image attachment</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>With Questions</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>With Answers</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td><strong>remote login to servers</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>included</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td><strong>Grading</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>automatic (multiple choice type)</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>By student</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td><strong>Web-Based only</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
1.2 Purpose of the Project

The purpose of the project is two-fold. The first purpose is to develop a tool that enables instructors to post coursework information such as homework, projects, grades online, and enable easy access and sharing of coursework details like questions among instructors.

The second purpose of the project is to develop a structured mechanism for users to access coursework information, work on the respective coursework, and view solutions and grades.

1.3 Context of the Problem

The context of the problem is to address the limitations of currently available systems by designing and implementing software that eliminates current limitations.

1.4 Significance of the Project

The project is dedicated only to coursework. The limited function makes the proposed tool is easy to use. The project makes use of graphical icons that make the tool user-friendly. The project is more significant because it can be installed on a client/server architecture giving users the same interface irrespective
of the architecture employed. This feature has the advantage that users are not required to get trained to utilize the tool if the on-line version is installed on a client/server architecture. The additional features such as, sharing of questions among instructors, randomizing the contents of coursework, automatic grading system for multiple choice type questions give the tool features that facilitates the overall coursework preparation processes.

1.5 Preliminary Design

The functional analysis of the WB project is depicted in Figure 1 below using Use Case Diagram. Here, three users types are identified: Student User type, Professor user type, and Teacher Assistant (TA) User type. The corresponding functionalities pertaining to each user type are noted.

1.5.1 Description of Use Case Diagram

1.5.1.1 Access Information. WB shall provide information about the current class section a student is taking including the breakdown of its components.
1.5.1.2 **DO Exercises.** WB will provide an environment for users to perform course work such as homework, projects, and exams for the section a student is registered in. The system takes into account the
possibility of having a number of sections for a specific course per quarter.

1.5.1.3 Change Account Information. WB shall provide a means of altering/changing password by respective user.

1.5.1.4 Registering Users/TA. A professor assigned to teach a course shall be given full privilege to add/drop students from the user list. In addition, Teaching Assistant shall be given access to the system by the corresponding professor he/she is assigned to.

1.5.1.5 Update Information. WB will provide a means for updating course work information posted on WB. This includes status of a course work (posted, submitted, graded), due date, and add/remove questions.

1.5.1.6 Post Grade/Solution. WB will enable the respective professors and TAs to post grades and solutions for each exercise type (homework, laboratory exercise, and projects)

1.5.1.7 Print. WB will provide users a means for printing the necessary reports and exercises.

1.6 Assumptions

The following assumptions were made regarding the project:
1. Users know how to use the Internet.

2. User computers are equipped with java enabled browser(s): Netscape version 4.1 and above, Internet Explorer 5.0 and above.

1.7 Limitations

The following limitations apply to the project:

1. Formatted text cannot be stored in the database.

2. Questions having graphics are limited to one graphic.

3. Data having special characters such as mathematical expressions cannot be stored into the database.

1.8 Definition of Terms

The following terms are defined as they apply to the project.

1. WB - White Board: - name of this project

2. Bb - Black Board ®: - a web-based software application designed to provide online courses

3. OS - Operating System: - the collection of software that directs a computer's operations, controls and schedules the execution of other
programs, and manages storage, input/output, and communication resources

4. GUI - Graphical User Interface: - techniques of using graphics, keyboard, and mouse to provide an easy-to-use, user interface to an application.

5. UML - Unified Modeling Language: - a standard notation for the modeling of real-world objects as a first step in developing an Object Oriented Design methodology.

6. JVM - Java Virtual Machine: - software that interprets and executes the byte codes in Java class files.

7. CGI - Common Gateway Interface: - a standard for programs to interface with information servers such as HTTP (web) servers

8. SSH - Secured Shell: - a protocol for securely getting access to a remote computer

9. SFTP - Secured File Transfer Protocol: - a protocol for securely transferring file(s) to a remote computer
CHAPTER TWO  
SOFTWARE DESIGN  

2.1 Architecture  

WB utilizes the state-of-the-art Oracle software development tools for the design of the GUI, and Oracle application servers and database system as the back-end. The three-tier application architecture is shown in Figure 2 below.

![Application Architecture Diagram]

Figure 2. Application Architecture

2.1.1 Client Tier  

Java enabled web browsers are used as a client tier. These browsers deploy the java client of the GUI designed using Oracle Forms. When a user initiates WB, this java client provides the user interface for the associated
Forms Server Run time engine on the server and handles user interaction and visual feedback.

2.1.2 Middle Tier

The middle tier also known as the application server consists of three components: Forms Listener, Runtime Engine and Forms Cartridges.

The Forms Listener acts as an agent taking connection request from the java client processes and initiating Forms server runtime processes on their behalf. It also maintains a pool of running engines for handling connections from java clients as quickly as possible.

The Forms Run Time Engine is a process that maintains a connection to the database on behalf of the java client process. It is composed of application logics, data management and PL/SQL engine.

The Forms Cartridge is an optional feature of the middle tier that gives provision for single point entry to many different Forms application and is useful for load balancing during distributed application server configuration at the middle tier.
2.1.3 Database Tier

The database tier is the back end application server where data are stored for future retrieval and updates. WB utilizes Oracle database version 9i as a back-end.

2.1.4 Communication Process

The detail communication steps are shown in Figure 3. The arrow lines in the figure indicates the direction of communication flow while the numbers attached with each arrow relates the figure with the steps in the communication process detailed in Section 2.1.4.1.

![Diagram of communication process]

Figure 3. Communication Process
2.1.4.1 Communication Process Steps.

1. User request from a browser or applet viewer.
2. Web server processes the request and fetches the requested page.
3. The client process requests a Java class implementation whose location is defined in the <APPLET> tag of the HTML code.
4. The web server furnishes the Java class file and starts the applet.
5. The Java client applet communicates with the Forms Listener via TCP/IP socket.
6. The Forms Listener, which is already started, starts a new Run Time Engine upon receipt of connection request from the client process. A new socket is allocated to this new run time engine to communicate with the Java client.
7. The Forms Listener hands off the connection between the java client and runtime engine.
8. Connection to the database server is established.
2.2 Problem Description

WB keeps track of information pertaining to coursework activities for a specific academic system, quarter or semester. This information is classified as Student information, Coursework information, and Department information. The objects associated with the design of WB system are also shown in Figure 4 using the Rational Rose UML Modeling system.

Figure 4. Conceptual Model Diagram
2.2.1 Problem Description

WB keeps track of information pertaining to coursework activities for a specific academic system, quarter or semester. This information is classified as Student information, Coursework information, and Department information.

2.2.1.1 Student Information. A student can register for one or more courses in a single academic term. However, a student can only register for one section per course. WB enables a student to perform course work for the courses he/she is registered in. WB stores the first name, last name and student identification number of a student.

2.2.1.2 Course Information. A course is identified by a unique course number and has a course description. A course is offered by a single department. Each course may have more than one section. Each section is taught by one instructor. Sections are distinguished from each other by unique numbers. For each section of a course, there exists course work composed of questions. Course work is classified by type: laboratory exercises, homework, and projects. Each course work type is further broken down by exercise numbers. Each exercise number is given a weight.
that totals the sum of points given to each question. For a given course work, the aggregate sum of points of each exercise number will be normalized to the points assigned by the instructor. In addition, each section of a course will have a grading scale to map the numerical percentile value to a letter grade. Other information to be stored with each exercise number includes due-dates, status of a course work indicating if an exercise has been submitted, posted, or graded.

Questions that are stored in the system belong to a specific course offered by a department. Each question will include the following information: question number, question description, corresponding weight, answer, owner, question type (essay, multiple choice), and complexity level (simple, medium, complex). Questions may be set public or private. Questions having a public property are sharable among instructors where as private types are not sharable. Attempted answers by each student, earned point and instructor’s comment for each question will also be handled by WB.

2.2.1.3 Department Information. A department will be identified by a unique department code and name. Each department will have a number of professors and offer a
number of courses. A professor can teach more than one section in a single academic term but each section is taught by one professor. First name, last name, and professor identification number will be stored in WB to identify a professor. A professor can assign a graduate assistant (TA) for the section(s) he/she is teaching. WB allows professors to organize course work for the sections they are teaching.

2.2.2 Database Design

The ER diagram in Figure 5 shows the database design for the WB project. The entities with corresponding attributes and the relationships among the entities are defined. This design is further normalized to third normal form to define and create the database tables required by the WB project. These details are documented in Appendix A.

Appendix B defines the data dictionary of the database design for WB project. The data types of each attributes, the constraints associated with the attribute, and attributes associated with each table are tabulated here.
2.3 Detailed Design

The detail design of the WB system refers to the problem description of WB system as outlined in section 2.2.1 above. Oracle Forms 6i is used to design the user interfaces. A tree-structured menu which groups course details by course drives the user interface. Three database roles are created which depicts the user type. These database roles are student role, professor role,
and Teacher Assistant (TA) role. These roles group and give access to common functionalities to user groups. The available functions related to a course and user type (role) is activated with a mouse click on the menu items. The interfaces defined and the corresponding usages are discussed in chapter four. The implementation details are grouped in the following software program components:

2.3.1 Stored Program Packages

Stored program packages are program modules that are bundled together in a package. The program logics are related functionally and are composed of functions and procedures. These packages are stored in the client side of the application.

2.3.2 Triggers

Triggers are program segments that are executed by user actions such as, pressing a button, mouth click, etc. or database system events like, database startup and instance shutdown.

2.3.3 Database Procedures and Packages

These are program segments as described in section 2.3.1 above but stored in the database server side. These types of program segments are useful to minimize the
network load created during execution of program modules that require frequent database access.
CHAPTER THREE
MAINTENANCE

3.1 Installation and Configuration of Application Servers

WB currently runs three application servers: Forms Server, Web Server, and Reports Server whose functionalities were outlined in sections 2.1.2 and 2.1.3.

3.1.1 Forms Server Installation and Configuration

The windows version of Oracle Forms server 6i is utilized by WB. The installation and configuration direction recommended by Oracle Corporation is followed during this phase. However, the following configurations are made during installation.

3.1.1.1 Configuration of Web Server. WB utilizes the webDB as a listener for Forms Server. The following virtual directory mappings are set.

- /dev60temp/: - is mapped to the directory where temporary report files, etc are generated: <Oracle_HOME>\web60\temp
- /forms60java/: - is mapped to the directory where java files are located <Oracle_HOME>\Forms60\java
- /jinitiator/: - is mapped to the directory where jinit.exe resides
- /wpicons/: - is mapped to the directory where the icons for the tree-structured menu are located.
- /doc/: - points to the directory where application specific HTML files are located
- /wpconfig/: - mapped to the directory containing the application-specific.DAT file.

3.1.1.2 Environment Variable Configuration (NT).

Required environment variables are generated automatically upon installation. However, the following application specific environment variables are added in the registry file.

- FORMS60_PATH - updated to include the forms developer application path
- GRAPHICS60_PATH - added to include the path for graphics files
- REPORTS60_PATH - added to include the path for reports file location

3.1.1.3 Forms Server Configuration File and HTML Setup. The Forms configuration file formsweb.cfg and the
two HTML files (base.html and basejini.html) located in
<oracle_HOME>\Forms_server directory are used to launch
web forms java applet that WB uses. The Base.html file
utilizes the browser’s native JVM whereas basejini.html
uses Oracle’s JInitiator JVM.

The formsweb.cfg file contains CGI and servlet
parameters for variables defined in the base.html files. When a user inputs the WB URL, the base.html file reads
corresponding variable values from the formsweb.cfg file.
The complete configuration details of both the base.html
and formsweb.cfg files are shown in Appendix C.

3.1.1.4 Forms Startup Parameters Configuration.
Forms Service utilizes four parameters during start up.
These are Port Parameter, Mode Parameter, Pool Parameter,
and Log Parameter.

The Port Parameter defines the port on which the
forms process is started. This parameter must match with
the server port number specified in the configuration
file or the port number passed as argument on the URL.
This Parameter is set as 9001 (default value) for WB.

The Mode Parameter indicates the network connection
mode. It can have a socket mode that uses direct socket
connection, HTTP mode that can traverse firewalls, or
HTTPS mode that traverses firewall and SSL server authentications and message encryptions features. WB is set to have HTTPS Mode.

Pool Parameter indicates the number of available spare connection for users. This parameter is configured to value 100 in WB.

The Log Parameter generates a log file in the given path. This is set to the directory /WB/LOG in WB.

3.2 Secure Shell Applets and Secured File Transfer Protocol

The MindTerm Secure Shell (SSH) applet from AppGate is incorporated in WB to access and/or transfer file remotely to other servers. Particularly, this feature is integrated with the existing SSH server installed in the Computer Science Department, CSUSB to enable access the servers in the department. This feature will allow students to get direct access to respective accounts to work on and transfer course work on-line without the need to install the client version of the SSH on client machines. The SSH and SFTP applets reside in the web server directory.
CHAPTER FOUR

USERS MANUAL

4.1 Login Interface

The login screen shown in Figure 6 is activated when WB is started up. Users are required to input user name, password, and the database name to connect to. With valid login information supplied, the Main Interface is displayed.

![Login Interface](image)

Figure 6. Login Interface

4.2 Main Interface

The Main Interface is the driving window containing a tree-like structured menu which users can click on to get access to the course work information. As shown in Figure 7, the menu is grouped by course name, coursework type, and coursework details. Users having a student role will see all the courses they are registered in the menu if the course is available in WB. Users having a
professor role will have the list of courses they are teaching.

4.2.1 Main Menu

The tree-like Menu utilizes iconic Menu items to graphically indicate the status of the course work. The following iconic images are used to provide information:

4.2.1.1 Question Mark. This iconic image indicates that work needs to be done on the course work pointed at by the mouse button. When USER type is student, this icon indicates different information depending on the level at which it is shown in the menu:

1. Course level - indicates that all course work posted are not yet submitted.
2. Course work level - Indicates all exercises in that particular course work type are not yet submitted.

3. Exercise level - indicates this particular exercise is not yet submitted.

When user type is professor, the icon has similar meaning at all levels but indicates that all course works are not yet posted for students to work on.

4.2.1.2 Right Arrow. This iconic image is only available to professors. It indicates that there is course work submitted but is not yet graded. The icon will have the following meanings at different levels of the menu:

1. Course level - indicates that all course work created under the respective course name have been submitted but not graded.

2. Course work level - indicates that all exercises defined under the respective course work are submitted.

3. Exercise level - indicates that there exists at least one exercise that is submitted but not graded.
4.2.1.3 Up Arrow. This icon is defined at all levels of the menu and is available only to professors. It indicates that the respective exercise is posted when displayed at exercise level. When displayed at course work level, this icon indicates that all exercises created under the respective course work are posted. At course level of the menu, this icon indicates that all course work defined in the respective course are posted.

4.2.1.4 Tic Mark. This icon is also defined at all levels of the menu indicating grading of the course works. It is defined for all user types.

4.2.1.5 Access Denied Mark. This icon is only available to student user type. It indicates that user has submitted an exercise when it is displayed at the exercise level. When all exercises under a specific course work are submitted, this icon will be displayed at course work level. At course level, this icon indicates that all respective course work are submitted.

Users with student user type can only view the work performed at this level until the course work are graded.

4.2.2 Pop-Up Menu

Pop up menu is activated when users right-click on the menu items. The pop up menu lists allowed functions
that can be performed at the current menu item depending on the user type and status of the course work details.

Figure 8 shows the pop-up menu having all functionalities for student user type while Figure 9 shows pop-up menu for professor user type.

Figure 8. Pop-Up Menu I
Figure 9. Pop-Up Menu II

The following actions are performed when a user selects the respective popup menu items:

1. WORK_ON - Displays the WORK_ON interface to enable users with students to work on exercise. It is available only to students and is only available at exercise level of the Main Menu.

2. Check Grade - This initiates a grade report. It is available at all levels of the Main Menu. When initiated at course level, it displays a report showing summary of grades at course level. Details of each exercise grades are
displayed when initiated at exercise level. It is visible for all user types and if there exists a graded exercise.

3. Check Solution - Initiates a SOLUTION_BLK interface with Query-Only status. It is visible for all graded exercises and is available at the exercise level of the Main Menu.

4. Grade Exercise - Initiates SOLUTION_BLK interface with update status. It is only available to professor user type and for submitted exercises. It is visible at the exercise level of the Main Menu.

5. Create Exercise - This pop up menu item is only available to professor user type. It invokes Create_Exercise interface. It is visible at the COURSE work level of the main menu.

6. Add Course Work - Invokes Register_Exercise interface. It is only visible at the course level of the main menu and to professor user type.

7. Add Question - Is available to professor user type and initiates the Add_Question interface.
It is visible at the course level of the Main Menu.

8. Set Due Date - Initiates the Set_DueDate interface and is visible at exercise level of the Main Menu. It is visible for professor user type.

9. Add User - Initiates the Add_User interface. It is only accessible by professor user type. It is visible at course level of the Main Menu.

10. Add Exercise - Invokes Add_Exerno interface. It is visible to professor user type and at course work level of the Main Menu.

4.3 WORK_ON Interface

The WORK_ON interface enables student user type users to access course work details and work on the respective exercises.
<table>
<thead>
<tr>
<th>NO.</th>
<th>Questions</th>
<th>Due date</th>
<th>Total Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>write a C++ code to display a message &quot;HELLO WORLD&quot;</td>
<td>27/03/2003</td>
<td></td>
</tr>
</tbody>
</table>

Points
15

Figure 10. WORK_ON Interface

As shown in Figure 10, it has the following elements:

1. **Problem Item** - Displays the questions set for an exercise.
2. **Answer Item** - Text field for answer input. When question is of a multiple-choice type, it is hidden and the choices are displayed.
3. **No. Item** - Displays the current question number.
4. **Point Item** - Displays the associated points for the current question.
5. Due_date item - Displays the due-date set for the current exercise.

6. Total Points item - Displays the total points achievable for the current exercise.

7. BACK/NEXT button - These buttons enable users to navigate to the previous/next record if it exists. At first record, the BACK button is disabled while at last record the NEXT button is disabled.

8. Remote Login Button - Enables users to have a remote connection to servers. Currently it is configured to remotely connect to the Computer Science Server.

9. Save Button - When pressed, this button saves all the work performed.

10. Submit Button - When pressed, the SUBMIT button saves and submits the work performed. Once pressed, users cannot edit.

11. Quit Button - Quits the session and closes the WORK_ON interface.

The Save, Submit, and Quit button all initiate an alert to users when pressed allowing users to confirm their selections.
4.4 SOLUTION_BLK Interface

The SOLUTION_BLK interface enables users to view attempted answers to exercises and the corresponding correct answers. The interface is shown in Figure 11.

![SOLUTION_BLK Interface](image)

Figure 11. SOLUTION_BLK Interface

The SOLUTION_BLK interface has two states depending on the type of user. When user type is student, this interface is not updateable. It enables users to view solution, professors' comment, and respective points achieved. When User type is professor, the SOLUTION_BLK interface is updateable allowing instructors to grade exercises and post comments. The SOLUTION_BLK interface has the following elements:
1. **Problem Item** - Displays the questions set for an exercise.

2. **Answer Item** - Text field for answer input. When question is of a multiple-choice type, it is hidden and the choices are displayed. This field is not updateable. Moreover, when Question type is multiple-choice and the attempted answer is wrong, the correct choice will be shown with a red foreground color.

3. **Your Answer Item** - This item displays the attempted answer. When question is a multiple-choice type, this item will be hidden.

4. **Point Item** - Displays the associated point for the current question

5. **Earned Point Item** - Indicated the achieved point for the current question. When user is of student type, this item will not be updateable. When User type is professor, it will have updateable feature allowing insertion of grade for the current question. When question type is multiple-choice and attempted answer is correct, this item will have the same value as
Point Item. However, if not correct it will have a value equal to zero.

6. Earned Total Points Item - Indicates the total arithmetic sum of the earned point for the current question.

7. Student Name List - This list will hold names of students who have submitted the current exercise but have not yet been graded. When user type is student, this list will only have the current user's name. Once Graded, This list will be updated automatically removing the graded exercises.

8. Back/Next Button - These buttons enable users to navigate to the previous/next record if it exists. At first record, the back button is disabled while at last record the next button is disabled.

9. Save Button - When pressed, this button saves all the work performed. This button is only enabled to professor and TA user types.

10. Submit Buttons - When pressed, the submit button saves and submits the work performed.
This button is only enabled to professor and TA user types.

11. Quit Button - Quits the session and closes the SOLUTION_BLK interface.

12. Comment Button - When this button is pressed, it will initiate a comment block to enable insert/view a comment attached to the current question. This button is not enabled if the current question has no comment.

4.5 Comment Interface

The Comment Interface is shown in Figure 12.

![prof_comment]

Figure 12. Comment Interface

This interface enables professor type users to attach comments at the time of grading exercises. When user type is student, this interface is initiated when
the comment button in the SOLUTION_Blk interface is pressed. The following items are incorporated in the comment interface:

1. PROF_COMMENT Item- When user type is student, this text area displays the comment attached to the current question if any and will be not be updateable. When user type is professor, it will enable professors to insert or edit comments pertaining to the current question.

2. OK Button - When pressed, this button closes the comment interface.

4.6 CREATE_QUESTION Interface

The CREATE_QUESTION interface enables professor type users to create and store exercise questions for the current course. It is initiated from the CREATE_QUESTION pop-up menu item.
As shown in Figure 13, this interface is composed of the following items:

1. Question Item - Is a multiple line text field that holds question text.

2. Answer Item - Is a multiple line text field to input correct answer for the current question.

3. Points Item - Holds possible achievable point associated to a correct answer.

4. From List - This list holds numerical consecutive numbers to select from indicating
the chapter the current question is referring to.

5. ATTACH_GRAPHICS Checkbox – When selected, this checkbox initiates a FILE_PATH interface to select graphics files and stores the file name along with the question in the database.

6. MULTI_CHOICE Checkbox – This checkbox indicates the type of question that is being created. When selected, it indicates that the question is of multiple-choice type and initiates the choice fields for insertion. When not selected, it indicates the question to be of an essay type.

7. Back/Next Button – These buttons enable users to navigate to the previous/next record if it exists. At first record, the back button is disabled while at last record the next button is disabled.

8. Submit Button – Stores the question to be added in the database. This button is not enabled unless the one of following condition is met:
i. When MULTI_CHOICE checkbox is selected, at least two choices must be inserted along with the correct choice box selected.

ii. When MULTI_CHOICE checkbox is not selected, question text item should not be null or empty.

9. UPLOAD Button - This Button enables questions to be uploaded from a text file conforming to the format shown in APPENDIX D. When this button is pressed it will initiate a Dialog window for file upload/transfer. Once a file is selected, LOAD_QUESTION interface will be initiated.

10. Cancel Button - Cancels all edits and exit CREATE_QUESTION interface.

11. NEW_EXERCISE Button - When selected, this button enables to produce exercises directly from the questions that are being created. This button will trigger the REGISTER_EXERCISE interface.
4.7 Load_Question Interface

Load_Question interface is initiated when the upload button in CREATE_QUESTION interface is selected. This interface enables loading of question from an OS text file. All the items defined in this interface have the same functionalities as those defined in CREATE_QUESTION interface.

4.8 Register_Exercise Interface

This interface is shown in Figure 14.

![Figure 14. Register_Exercise Interface](image)

The Register_Exercise interface is initiated either from the Create_Coursework popup menu item or when the NEW_EXERCISE button in the Load_Question interface is pressed. It incorporates the following items:
1. Exercise List - This list displays all the course work types that are stored in the database and are not currently defined for the current course.

2. Percentage Item - Input area for the total percentile point associated with the new course work type.

3. Add_New Button - When pressed, it displays text fields to input course work description and the corresponding course work indicator.

4. OK Button - Saves the new coursework type that is created.

5. Cancel Button - Resets the percentage field.

6. Close Button - Closes and exits the Register_Exercise interface without saving changes.

4.9 Add_Exerno Interface

The Add_Exerno interface is initiated when Add_Exercise popup menu item is selected. It is also initiated when creating a new exercise from Create_Question interface. This interface is shown in Figure 15.
These elements are associated with the ADD_EXERNO interface:

1. **Exer_No Item** - This item indicates the new exercise number that is being created. WB automatically inserts the new value by referencing the last exercise number that is stored and increments the numerical value by one.

2. **Percentile Item** - This holds a portion of percentile value that is to be assigned for the new exercise number for a given course work.

3. **Due_Date Item** - Indicates the due date set for the current exercise number being created. By
default this will have a current system date. It can be edited by clicking the desired date on the calendar attached.

4. Calendar - Facilitates selection of the due date. Calendar of months starting from the current month and date is displayed. Navigation between months is achievable by clicking on the left and right arrows. Clicking a date updates the DUEDATE field.

5. OK Button - Saves the new insertion and updates the database.

6. Cancel Button - Resets the percentile field.

7. Close Button - Exits the ADD_EXERNO interface.

The Ok and Close buttons initiate an alert to confirm the selection.

4.10 Register_User Interface

This interface is shown in Figure 16. It is available to PROFESSOR user type to give access to users to utilize WB.
The following items are incorporated with this interface:

1. Last Name field - To input last name of the user
2. First Name field - To input first name of the user
3. User Id field - To input a unique user identification number.
4. File Button - To read user information from a text file conforming to the format.
5. Ok Button - Saves the input
6. Next Button - Saves the current input and clears the interface for next input.
7. Close Button - Exits from the REGISTER_USER interface.

Figure 16. Register_User Interface
The Ok and Close Buttons initiate an alert to Confirm selection.

4.11 Control Interface

The Control interface displays the current course name, course work, and exercise number to indicate to users the details of the current course being visited. It is shown in Figure 17. below.

```
| cs330 | Home Work | HMK_1 |
```

Figure 17. Control Interface
CHAPTER FIVE

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusion

The WB project shows a successful implementation for coursework preparation and submission features. Designed and implemented using a component based software engineering methodology, WB can be used as a client/server application or as a Web-based application with minimal configuration setup.

WB gives instructors a variety of options to create and manage coursework details. In addition, WB delivers a powerful feature that enables instructors to share question information.

WB can be accessed anywhere by any java-enabled web browser. This makes it possible for WB to be used for online courses.

WB consists of various state-of-the-art Oracle technologies. This project has provided me with the experience in installing and configuring Database systems and Application servers, designing, coding, testing, and debugging real world applications.
5.2 Future Directions

WB system implements features that are common in educational activities as described in Chapter One. WB may have the following additional functionalities:

5.2.1 Grade Availability through Wireless Technology

Oracle application server 9iAS is equipped with additional features that support wireless technology. Using this inherent facility, WB can have grade availability through wireless equipments such as, cell phones and personal digital assistant (PDA).

5.2.2 Artificial Intelligent Capability

Currently, WB handles automatic grading feature for multiple choice type questions. Functionalities of WB can be enhanced more through the implementation of automatic grading system for essay type questions using AI agents.

5.2.3 Rich Text Questions

Currently, WB allows questions to be expressed using ASCII text. Future versions can add HTML, RTF, and XML formatted text at the professors’ choice. HTML and RTF will allow many graphics to be embedded in questions and answers.
5.2.4 Answer Statistics

The system can summarize students responses highlighting questions where many did poorly. This is useful for faculty. A question where all fail indicates something is wrong.
APPENDIX A

DATABASE TABLES
<table>
<thead>
<tr>
<th>STUDENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student id</td>
<td>Last_Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Section no</td>
<td>Quarter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUD SEC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>StudId</td>
<td>Secno</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEC_GRADE_SCALE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Secno</td>
<td>From Point</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Course no</td>
<td>Course_Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXERCISE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exer Type</td>
<td>Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dept Code</td>
<td>Dept_Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROFESSOR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fac Id</td>
<td>Fac_Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROF SEC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof_Id</td>
<td>Secno</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEC_EXER_TYPE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Secno</td>
<td>Exerno</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Question no</td>
<td>Courseno</td>
</tr>
</tbody>
</table>

|          | Topic | Complexity_level | Access_Mode |
### EXER_QUESTION
| Exerno | Secno | Questno |

### STUD_EXER_ATTEMPT
| StudId | Secno | Exerno | Questno | Attemp Answer | Earned pt | Prof_Comment |

### STUD_EXER_STATUS
| StudId | Secno | Exerno | Status |

### EXER_STATUS
| Secno | Exerno | Status |

### RANDOM_EXERCISE
| Secno | Exerno | Simple | Medium | Complex | Topic |

### CHOICE
| Questno | Choiceno | Choice |
APPENDIX B

DATA DICTIONARY
<table>
<thead>
<tr>
<th>TABLE NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHOICES</td>
<td>The CHOICE table stores choice number and choice description for each question that are of multiple choice type.</td>
</tr>
<tr>
<td>COURSE</td>
<td>Information pertaining to a course given is maintained by the COURSE table. This includes course no, course description and department number</td>
</tr>
<tr>
<td>DEPARTMENT</td>
<td>The DEPARTMENT table contains relevant attributes of a department which includes department number and department name</td>
</tr>
<tr>
<td>EXER_QUESTIONS</td>
<td>EXER_QUESTIONS table contains information which describes the composition of each exercise types by storing the exercise number with corresponding section number and the detail questions each exercise refers from the main EXERCISE table</td>
</tr>
<tr>
<td>EXER_STATUS</td>
<td>The EXER_STATUS table stores the status of each created exercises. It can have POSTED, GRADED, or NULL values.</td>
</tr>
<tr>
<td>EXERCISE</td>
<td>EXERCISE table holds exercise type and exercise description.</td>
</tr>
<tr>
<td>PROF_SEC</td>
<td>PROF_SEC table describes the professor-section relationship by storing the professor's id and the section number the professor is assigned</td>
</tr>
<tr>
<td>TABLE NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PROFESSOR</td>
<td>Instructors' information is contained in the PROFESSOR table. Faculty id, name, department code, and Prof_flag are contained in this table. The Prof_Flag field describes if the instructor is a faculty or a Graduate Assistant.</td>
</tr>
<tr>
<td>QUESTION</td>
<td>QUESTION table is a warehouse for all approved questions segregated by course number. It contains valuable information like, course number, questions and respective answers, possible point set for the question, which chapter a question refers, owner, and for what exercise type (LAB, EXAM, PROJECT, etc.) the question is intended.</td>
</tr>
<tr>
<td>RANDOM_EXERCISE</td>
<td>The RANDOM_EXERCISE table holds information pertaining to those exercises set to be randomly generated at the time of course work creation. It holds the number of questions that are to be randomly generated for the selected complexity level and topic.</td>
</tr>
<tr>
<td>SEC_EXERCISE</td>
<td>The total share point (from 100%) of each exercise type for a given section is stored in SEC_EXERCISE table.</td>
</tr>
<tr>
<td>TABLE NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SEC_EXERTYPE</td>
<td>SEC_EXERTYPE table holds information pertaining to exercise set for a specific section of a course. It stores exercise number, section number, total number of questions set, Due date, status of the exercise, and the total point given for each exercise numbers. The STATUS field indicates if a specific exercise is SUBMITTED, GRADED, or POSTED.</td>
</tr>
<tr>
<td>SEC_GRADE_SCALE</td>
<td>SEC_GRADE_SCALE table holds grading information set for each section numbers. This includes information like, section_no, range of point (From_point,To_point) and the respective letter grade for that specific range of point.</td>
</tr>
<tr>
<td>SECTION</td>
<td>This table holds section information for all courses registered in the system. This Information includes Section number, Quarter, and course no.</td>
</tr>
<tr>
<td>STUD_EXER_ATTEMPT</td>
<td>The works performed by students are kept in STUD_EXER_ATTEMPT table. This table stores student id, section number, exercise no, question number, attempted answer, earned point, and comment by professor for each question in a given exercise number.</td>
</tr>
<tr>
<td>STUD_EXER_STATUS</td>
<td>This table holds the status of each exercise pertaining to students. The status may have SAVED, SUBMITTED or NULL values. - Work_on!</td>
</tr>
<tr>
<td>TABLE NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>STUD_SEC</td>
<td>STUD_SEC table stores Student_id and Section_no information - Registration</td>
</tr>
<tr>
<td>STUDENT</td>
<td>The STUDENT table maintains student information like student Id, last name, and first name</td>
</tr>
</tbody>
</table>
APPENDIX C

CONFIGURATION FILES
; Forms Web CGI Configuration File
; -----------------------------------------------
; This file defines parameter values used by the Forms
Web CGI
; **********************************************
; PARAMETER VALUES USED BY DEFAULT
; **********************************************

; SYSTEM PARAMETERS
; ---------------

baseHTML=%FORMS60%\server\base.htm
baseHTMLJInitiator=%FORMS60%\server\basejini.htm
HTMLdelimiter=%
MetricsServerPort=9020
MetricsServerErrorURL=
IE50=JInitiator

; USER PARAMETERS

form=test.fmx
userid=
otherparams=

; 2) HTML page title, attributes for the BODY tag, and
HTML to add before and
after the form:
pageTitle=Oracle Forms Server
HTMLbodyAttrs=
HTMLbeforeForm=
HTMLafterForm=

; 3) Values for the Forms applet parameters:
width=650
height=500
separateFrame=false
splashScreen=no
; select default background by not specifying a value
background=
lookAndFeel=Oracle
colorScheme=teal
serverApp=default
serverPort=9000
serverHost=
connectMode=Socket
archive=f60web.jar

; 4) Parameters for JInitiator
jinit_download_page=/jinitiator/us/jinit_download.htm
jinit_download_page=/jinitiator/jinit_download.htm

; Parameters related to the version of JInitiator.
jinit_classid=clsid:093501ce-d290-11d3-a3d6-00c04fa32518
jinit_exename=jinit.exe#Version=1,1,7,27
jinit_mimetype=application/x-jinit-applet;version=1.1.7.27

; *****************************************
; SPECIFIC CONFIGURATIONS
; *****************************************

[sepwin]
separateFrame=True
lookandfeel=Generic
[ie50native]
IE50=native
[applet]
baseHTMLJInitiator=

****END OF FORMSWEB.CFG****
BASEJINI.HTM File

<HTML>
<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>

%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<OBJECT classid="%jinit_classid%"
   codebase="/jinitiator/%jinit_exename%"
   WIDTH="%Width%"
   HEIGHT="%Height%"
   HSPACE="0"
   VSPACE="0">
   <PARAM NAME="TYPE" VALUE="%jinit_mimetype%">
   <PARAM NAME="CODEBASE" VALUE="/forms60java/">
   <PARAM NAME="CODE" VALUE="oracle.forms.engine.Main">
   <PARAM NAME="ARCHIVE" VALUE="%archive%">
   <PARAM NAME="serverPort" VALUE="%serverPort%">
   <PARAM NAME="serverHost" VALUE="%serverHost%">
   <PARAM NAME="connectMode" VALUE="%connectMode%">
   <PARAM NAME="serverArgs" VALUE="module=%form%
   userid=%userid%
   %otherParams%">
   <PARAM NAME="separateFrame" VALUE="%separateFrame%">
   <PARAM NAME="splashScreen" VALUE="%splashScreen%">
   <PARAM NAME="background" VALUE="%background%">
   <PARAM NAME="lookAndFeel" VALUE="%lookAndFeel%">
   <PARAM NAME="colorScheme" VALUE="%colorScheme%">
   <PARAM NAME="darkLook" VALUE="%darkLook%">
   <PARAM NAME="serverApp" VALUE="%serverApp%">
   <PARAM NAME="imageBase" VALUE="%imageBase%">
   <PARAM NAME="serverApp" VALUE="%serverApp%">
   <PARAM NAME="imageBase" VALUE="%imageBase%">
   <PARAM NAME="serverApp" VALUE="%serverApp%">
   <PARAM NAME="imageBase" VALUE="%imageBase%">
   <!-- Forms applet definition (end) -->
   <COMMENT>
   <EMBED SRC="" PLUGINSPAGE="%jinit_download_page%"
      TYPE="%jinit_mimetype%"
      java_codebase="/forms60java/"
      java_code="oracle.forms.engine.Main"
      java_archive="%archive%"
      WIDTH="%Width%"
      HEIGHT="%Height%"
      HSPACE="0"
      VSPACE="0"
   </EMBED>
</BODY>
</HTML>

64
serverPort="%serverPort%"
serverHost="%serverHost%"
connectMode="%connectMode%"
serverArgs="module=%form% userid=%userid% %otherparams%"
separateFrame="%separateFrame%"
splashScreen="%splashScreen%"
background="%background%"
lookAndFeel="%lookAndFeel%"
colorScheme="%colorScheme%"
darkLook="%darkLook%"
serverApp="%serverApp%"
imageBase="%imageBase%"
>
</NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>
APPENDIX D

TEMPLATE FOR QUESTION FILES
Here is question 1 read from file
Here is answer no 1 read from file
Here is question no 2 read from file
Here is question no 3 read from file
Here is question no 4 read from file
Here is answer no 4 read from file with extension now

NOTE:

*q*, *Q* :- flag to indicate question contents
*a*, *A* :- flag to indicate answer contents
REFERENCES


