The Design and Implementation of a Virtual Engineering Technology Laboratory Experiment

GRADUATE PROJECT TECHNICAL REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science

by

Sae Hwang
Fall, 2002

Committee Members

Dr. David Thomas
Committee Chairperson

Dr. Ray Bachnak
Committee Member

Dr. Mario Garcia
Committee Member
ABSTRACT

Hands-on activities are very important for teaching engineering concepts and techniques. While institutions are moving toward distance education opportunities, they are facing the problem of offering quality laboratory experiences to their students. As a result, a number of web-based systems that rely on software development programs, such as CGI, Perl, Java, and C++, have been developed. This approach makes developing, maintaining, and upgrading laboratory experiments very expensive and a time consuming process. This project involves the design and implementation of a virtual laboratory that enables engineering technology students to perform laboratory experiments online. The laboratory uses LabVIEW to provide data acquisition, storage, and processing capabilities and real-time viewing of the laboratory environment using a digital camera. LabVIEW is a powerful programming toolkit that accommodates data acquisition, instrument control, data processing, and network communication. It also enhances streaming of live data as well as interaction among various system modules. Experimental results show that the system functions as expected.
# TABLE OF CONTENTS

Abstract........................................................................................................................................................................ii

Table of Contents........................................................................................................................................................iii

List of Figures ..............................................................................................................................................................vii

1. Introduction and Background .................................................................................................................................1

   1.1 Data Acquisition using LabVIEW ......................................................................................................................1

   1.2 Distance Learning System .................................................................................................................................2

   1.3 Access through a Firewall .................................................................................................................................3

2. Virtual Engineering Technology Laboratory Experiment ..........................................................................................4

   2.1 Overview .............................................................................................................................................................4

   2.2 Student Home Page ..........................................................................................................................................6

      2.2.1 Login Page .................................................................................................................................................6

      2.2.2 Student Page .............................................................................................................................................6

   2.3 Instructor Home Page .......................................................................................................................................10

      2.3.1 Login Page ..............................................................................................................................................10

      2.3.2 Lab Pages ................................................................................................................................................11

      2.3.3 User Info and Add New User Pages .......................................................................................................13

      2.3.4 Announcement Page ...............................................................................................................................14

3. System Design ............................................................................................................................................................15

   3.1 Hardware Overview for Acquisition System ........................................................................................................15

   3.2 Data Acquisition System ..................................................................................................................................15

      3.2.1 The Transducer ....................................................................................................................................17

      3.2.2 The Signal Conditioner ........................................................................................................................17
3.2.2.1 Chassis (SCXI 1000) ........................................................ 18
3.2.2.2 Analog Input Module (SCXI 1102C) ......................... 18
3.2.2.3 Input Terminal Block (SCXI 1303) ......................... 19
3.2.2.4 Analog output Module (SCXI 1124) ......................... 19
3.2.2.5 Output Terminal Block (SCXI 1325) ......................... 20

3.2.3 The DAQ Board................................................................. 20

3.2.4 Configuration of Data Acquisition System ......................... 21

3.3 Image Acquisition System....................................................... 22
3.3.1 Video Camera ................................................................. 23
3.3.2 Image Acquisition Board .................................................. 24
3.3.3 Driver and Sub-Program.................................................. 24
3.3.4 Configuration of Camera and Image acquisition Board ....... 26

3.4 Software Design...................................................................... 26
3.4.1 LabVIEW as a Programming Language ......................... 27
3.4.1.1 Graphical Programming .......................................... 27
3.4.1.2 Reusability ............................................................... 29
3.4.1.3 Maintainability ......................................................... 31
3.4.1.4 Extendibility ............................................................. 32
3.4.2 LabVIEW Application of the Temperature System ......... 33
3.4.2.1 Front Panel of the Temperature System ................. 33
3.4.2.2 Block Diagram of the Temperature System ............. 36
3.4.3 Web Application .............................................................. 39
3.4.3.1 Programming Languages ......................................... 41
8. Add New Lab Page .............................................................................................................. 61

9. User Info Page .................................................................................................................. 62

10. Add New User Page ....................................................................................................... 64

11. Announcement Page ..................................................................................................... 65
LIST OF FIGURES

Figure 2.1 Virtual Engineering Technology Laboratory Experiment ........................................ 4
Figure 2.2 Building Cooling System .................................................................................. 5
Figure 2.3 Login Page ......................................................................................................... 6
Figure 2.4 Student Page ...................................................................................................... 7
Figure 2.5 The Front Panel of Temperature System ........................................................... 7
Figure 2.6 Request Control of VI Menu on the Temperature System .............................. 8
Figure 2.7 Temperature System Experiment ...................................................................... 9
Figure 2.8 Release Control of VI Menu ............................................................................ 10
Figure 2.9 Lab Info Page .................................................................................................. 11
Figure 2.10 Add New Lab Page ........................................................................................ 12
Figure 2.11 Modify / Delete Lab Page .............................................................................. 12
Figure 2.12 User Info Page ............................................................................................... 13
Figure 2.13 Add New User Page ...................................................................................... 13
Figure 2.14 Announcement Page ...................................................................................... 14
Figure 3.1 Process of Data Acquisition and Instrument Control ....................................... 16
Figure 3.2 Data Acquisition System for Temperature System .......................................... 16
Figure 3.3 SCXI 1000 chassis ........................................................................................... 18
Figure 3.4 SCXI 1102C .................................................................................................... 19
Figure 3.5 SCXI 1303 ....................................................................................................... 19
Figure 3.6 SCXI 1124 ....................................................................................................... 20
Figure 3.7 SCXI 1325 ....................................................................................................... 20
Figure 3.8 PCI-MIO-16E-4 Data Acquisition Board ........................................................ 21
Figure 3.9 MAX for Configuring Data Acquisition System............................................. 22
Figure 3.10 Process of Image Acquisition........................................................................ 23
Figure 3.11 Sony DFW-V500 IEEE-1394 Camera and Image Acquisition Board........... 24
Figure 3.12 Architecture of NI-IMAQ for 1394 Software Package ................................. 25
Figure 3.13 MAX for Configuring Image Acquisition System......................................... 26
Figure 3.14 Front Panel Constructing Graphical User Interface ..................................... 28
Figure 3.15 Block Diagram Corresponding to Front Panel ............................................. 28
Figure 3.16 Block Diagram Constructing an Executable Program.................................... 29
Figure 3.17 Front Panel and Block Diagram of ave.vi ................................................. 30
Figure 3.18 Front Panel and Block Diagram of New Program using ave.vi as Subroutine ................................................................. 30
Figure 3.19 ave.vi after Changing from Arithmetic Ave to Geometric Ave ................... 31
Figure 3.20 VI using the Changed ave.vi as Subroutine ................................................ 32
Figure 3.21 Block Diagram Adding the New Function Checking if Two Input are Equal ................................................................. 32
Figure 3.22 Front Panel Adding the New Function Checking if Two Input are Equal ... 33
Figure 3.23 Front Panel of Temperature System........................................................... 33
Figure 3.24 Acquisition Loop Module of Data Acquisition........................................ 37
Figure 3.25 Analysis Structure Module of Data Acquisition ........................................ 38
Figure 3.26 Block Diagram of Image Acquisition Module ........................................... 39
Figure 3.27 Level 1 Transition Diagram of the Web application .................................... 40
Figure 3.28 Level 2 Transition Diagram of the Instructor Pages ................................. 40
1. Introduction and Background

Laboratory experiments in engineering technology courses play a significant role in supporting the learning process by providing hands-on experiences so that students can form solutions and gain insight for real-life problems. A number of constraints, however, prevent the broad-based integration of experiments into the curriculum [Deliwala 1997]. Laboratory facilities, operating hours, safety issues, distance education, and software development difficulties are a few of the examples of the limiting factors. As a result, a number of web-based systems that rely on software development programs, such as CGI, Perl, Java, and C++, have been developed. This approach makes developing, maintaining, and upgrading laboratory experiments very expensive and a time consuming process.

A distance laboratory system has to provide efficient interaction with multi-user operations, machine and platform independence, secure operations, graphical user interface compatibility, high processing bandwidth with real-time data streaming capabilities, low cost maintenance, and intranet fusibility [Egarievwe 2000]. This report describes the design and implementation of a virtual laboratory system that enables engineering technology students to perform laboratory experiments online. The laboratory uses LabVIEW to provide data acquisition, storage, and processing capabilities and real-time viewing of the laboratory environment using a digital camera.

1.1 Data Acquisition using LabVIEW

A data acquisition (DAQ) system needs to get real-world signals into the computer. These signals come from a diverse range of instruments and sensors, and each type of signal needs special consideration [Bishop 2001]. Hardware without software is
of little use and the hardware can be very difficult to program when there are no proper controls. In addition, it is very difficult to find the appropriate application for a specific purpose. LabVIEW provides an easy way to accomplish the configuration and control of components within several boards [Bishop 2001]. This makes LabVIEW a good choice for creating specific applications at relatively low cost.

1.2 Distance Learning System

With the advancement of computers and the advent of the Internet, distance learning and computer based self-tutoring serve well as complimentary tools in education. Most existing approaches rely on achieving the improvement of students learning by developing simulation application that illustrate the basic concepts of engineering using traditional programming languages, such as Java, CGI, Visual Basic, etc. [Karweit 2000]. However, simulation applications do not allow students to experiment with real time equipment. While it is possible to develop real time measurements with traditional programming languages [RMSI 1998], such an approach requires the professional knowledge of both programming languages and hardware. This makes developing applications a very time consuming process.

LabVIEW has several advantages over other languages since it was specifically developed for high-performance scientific and engineering applications [Egarievwe 2000]. To enlarge the advantage of LabVIEW, the communication tool between instructors and students is required. For instance, LabVIEW applications might be executed at several computers and might be developed by several programmers. This
project uses LabVIEW to develop a distance learning system. The system includes a video camera that transmits images of the laboratory equipment.

1.3 Access through a Firewall

The access through the Internet is limited because the engineering technology laboratory at TAMUCC is behind a firewall. A firewall is a set of programs that protects the resources of a private network from users of other networks [Mankell 2002]. There are several ways to construct a firewall such as Network Address Translation (NAT), Proxy Sever, Packet Filtering and the combination of other methods [Minasi 2000]. The NAT technology is used for the firewall at TAMU-CC. The computers on the local network behind a NAT firewall use IP addresses that are reserved for use on internal networks only [Minasi 2000]. Those IP addresses of computers on a local network (at the inside of a NAT) will not show up on the Internet so the computers at the outside of a NAT cannot access the computers at the inside of the NAT. This virtual laboratory is available within the firewall at TAMU-CC.
2. Virtual Engineering Technology Laboratory Experiment

2.1 Overview

The Virtual Engineering Technology Laboratory is a Web-based application for helping students at TAMU-CC improve their learning experiences through interactive processes that are not possible with textbooks. Figure 2.1 shows a diagram representing a virtual experiment.

Figure 2.1 Virtual Engineering Technology Laboratory Experiment
LabVIEW applications are usually executed on LabVIEW Server computers. Clients or students access and perform the experiments through a web browser such as Internet Explorer or Netscape. Instructors can control students’ access and add or delete experiments. Instructors can also make experiments unavailable to students by hiding them. A Web application can provide the following.

- Serves as a communication tool between students and instructors
- Control the secure access of students for experiments
- Control the view of Web pages for students

The virtual Engineering Technology Laboratory is a temperature monitoring and processing system that offers students the opportunity to experiment with data acquisition and measurement. Such a circuit can be used for several industrial or research purposes, including fire alarm system, cooling system, and weather monitoring system. As an example, Figure 2.2 shows an example of a Building Cooling.

Figure 2.2. Building Cooling System
2.2 Student Home Page

The **Student Page** allows students to perform lab assignments. Students need to have proper user IDs and passwords to access the experiments.

2.2.1 Login Page

At the login prompt, students enter their user IDs and passwords as assigned by the instructor. See Figure 2.3.

![Login Page](image)

**Figure 2.3 Login Page**

2.2.2 Student Page

After a successful login, students can see the list of available laboratory experiments and a description of each. Announcements may also be displayed at the top of this page (Figure 2.4). When students click the button of an experiment, they can see the graphical user interface of the LabVIEW application in a new pop-up window. In this demonstration, Students can see the information for the Temperature System.
Figure 2.4. Student Page

Figure 2.5 shows the graphical user interface (called the *Front Panel*) of the Temperature system.

![Figure 2.5 The Front Panel of the Temperature System](image)
This system supports multi-user access. This means that all students who access this application see the same Front Panel. When a student wishes to control the lab experiment through the Front Panel, he or she must request control permission. If there is someone who already has control permission, other students can not have permission until control permission is released.

To request control permission, a student should click the right button of the mouse on the Front Panel and choose the Request Control of VI from the menu shown in Figure 2.6.

![Request Control of VI Menu on the Temperature System](image)

After a student is granted control, he or she can start the application by clicking the arrow at the top and can have access to all control functions. While the student manipulates some functions available on the temperature system, other students can only watch the changes on the front panel. Figure 2.7 shows the Low Limit of 75°. If the measured temperature is below the Low Limit, the temperature system turns on the LED and displays a warning message on the Front Panel as shown in Figure 2.7.
Students must release control when they have finished their experiment to give control permission to other students who want to do their experiment. To release control permission, the student selects the **Release Control of VI** (Figure 2.8).
2.3 Instructor Home Page

The instructor can control the view of Student Page, give the student access permission by assigning user IDs and passwords, and make announcements. The instructor can also add the link of new experiments and delete links for unavailable ones.

2.3.1 Login page

Like the student Login Page (Figure 2.3), the Login prompt will be seen at the start of the Web application. The instructor should enter his user ID and password at the Login prompt.
2.3.2 Lab Pages

The instructor can control the view of the student’s pages using instructor Lab pages. The lab pages consist of three pages: Lab Info Page (Figure 2.9), Add New Lab Page (Figure 2.10) and Modify/Delete Lab Page (Figure 2.11). The Lab Info Page allows the instructor to see the list of labs and the announcement made before. The view of Lab Info Page is very similar with the view of Student Page.

The instructor can add new lab information into the lab list by filling in Title prompt, Description prompt and the URL prompt to identify the LabVIEW Server computer on the Add New Lab Page (Figure 2.10). If the instructor wants to delete and modify the lab information, the instructor must select the lab number button at the Change Info list of Lab Info Page (Figure 2.9) that makes a user move to the Modify/Delete Lab Page (Figure 2.11).

![Figure 2.9. Lab Info Page](image-url)
Figure 2.10 Add New Lab Page

Figure 2.11 Modify/Delete Lab Page
2.3.3 User Info and Add New User Pages

The User Info Page (Figure 2.12) shows the list of user IDs and the status of the user. This page also allows the instructor to modify the user information and to delete the user by clicking the button labeled Change or Delete.

![Figure 2.12. User Info Page](image)

When the instructor wants to add a new user, he should enter the user ID, password and status at the three prompts labeled User ID, Password and Status, respectively on the Add New User Page.

![Figure 2.13. Add New User Page](image)
2.3.4 Announcement Page

There is one text area labeled *Announcement* and a submit button on the Announcement Page (Figure 2.14). When the instructor needs to notify students of some announcement, he must fill up the Announcement prompt and submit it. Students can see that notice immediately after logging in.

Figure 2.14. Announcement Page
3. System Design

3.1 Hardware Overview for Acquisition System

In the application of data acquisition and equipment control, hardware requirements and configuration is a very important issue due to the fact that software depends on the hardware specification [Bishop 2001]. For instance, as the thermocouple of the temperature system transmits analog signals, the acquisition system connected with the thermocouple has to manipulate analog signals and the LabVIEW application also must understand and process that signal [NI 2000].

This project involves two main hardware systems: Data Acquisition System and Image Acquisition System. The Data Acquisition System reads the temperature and sends the signal to an LED when the temperature measured is lower or higher than a certain range. The Image Acquisition System reads the image data from the camera and broadcasts the live video data on the Web to show the results of the Temperature System.

The main considerations for choosing hardware are the following [NI 2000].

- Compatibility with LabVIEW
- Cost Effectiveness
- Compatibility with other hardware for later use or update

3.2 Data Acquisition System

The basic flow of the Data Acquisition System begins with a reading from a set of transducers or instruments. The reading from the transducers is then passed to a Signal
Conditioner that transmits the information to the data acquisition board attached to the computer. Finally, the data are sent to the users through the Internet.

The overall process is summarized in the diagram shown in Figure 3.1 [Edwards 1999].

The Data Acquisition System available in the engineering technology laboratory can be broken down into three parts: Transducer, Signal Conditioner, and DAQ Board as shown in Figure 3.2. Detailed descriptions of the components follow.
3.2.1 The Transducer

A transducer is a device or instrument that converts a physical phenomenon (such as temperature or force) to a measurable electrical signal (such as voltage or current) [Bishop 2001]. Common examples of transducers include sensors for temperature, battery voltage, flow rate, pressure, fluid level and strain gauge. When measuring physical phenomena, a transducer must convert these phenomena into a measurable electrical signal such as voltage or current [Bishop 2001]. In the temperature system, the thermocouple is used for the input channel and the LED is used for the output channel. Both the input and output in this case are analog signals.

3.2.2 The Signal Conditioner

All transducers output an electrical signal. This is often not suitable for direct measurement by the DAQ system. For example, the output voltage of most thermocouples is very small and susceptible to noise and often needs to be amplified and filtered before measuring. Another example is when the signal being monitored contains large voltage spikes that could damage a computer or harm a person. Therefore, the signal can not be directly connected to the DAQ board [Bishop 2001]. Intermediate devices are needed to solve these problems. In this system, SCXI specifications (Signal Conditioning eXtensions for Instrumentation made by National Instruments, Inc) are used for a highly expandable signal conditioning system [NI 2000]. The Signal Conditioner used for this demonstration consists of 5 components: Chassis (SCXI 1000), Analog Input Module (SCXI 1102C), Input Terminal Block (SCXI 1303), Analog Output Module (SCXI 1124), and Output Terminal Block (SCXI 1325).
3.2.2.1 Chassis (SCXI 1000)

Figure 2.3 shows a 4-slot chassis known as the SCXI 1000. It houses all of the components and power modules used in this demonstration. The SCXI 1000 chassis has the following specifications [NI1999].

- 120 VAC
- Low noise environment for signal conditioning
- Rugged, compact chassis
- Forced air cooling
- Chassis to house all SXCI modules

3.2.2.2 Analog Input Module (SCXI 1124)

The component in Figure 3.4 is the SCXI 1102C Analog Input Module. It is ideal for higher bandwidth analog signals. Each channel can be configured to allow any combination of thermocouples to be used. This module has the following specifications [NI 1997].

- 32 channels, up to 333 kHz scanning rates for each channel
- For low-bandwidth millivolt, volt, and current inputs
- Over voltage protection to +/- 42V
- Programmable instrumentation amplifier on every channel

![Figure 3.4. SCXI 1102C](image)

### 3.2.2.3 Input Terminal Block (SCXI 1303)

The SCXI 1303 (Figure 3.5) is a terminal block for use with the SCXI 1102C modules and is designed especially for high-accuracy thermocouple measurements. The SCXI 1303 includes isothermal construction that minimizes errors caused by thermal gradients between terminals and the cold-junction sensor [NI 1997].

![Figure 3.5. SCXI 1303](image)

### 3.2.2.4 Analog Output Module (SCXI 1124)

The SCXI 1124 is the 12-bit digital-to-analog-converter with six independently isolated channels for DC voltage or current signals so it is used to convert the digital signals from the computer to the analog signals for the output transducer like LED. The
SCXI 1124 is ideal for applications requiring isolated voltage or current outputs to control a process [NI 1999].

![SCXI 1124](image)

**Figure 3.6. SCXI 1124**

### 3.2.2.5 Output Terminal Block (SCXI 1325)

The SCXI-1325 is a terminal block with 26 screw terminals for signal connections to the SCXI-1124 module [NI 1997].

![SCXI 1325](image)

**Figure 3.7. SCXI 1325**

### 3.2.3 The DAQ Board

The data acquisition board (DAQ board) serves as the interface between the computer and the transducers. The control of instruments and acquisition of data are achieved through a DAQ board [Bishop 2001].

The PCI-MIO-16E-4 is used in this demonstration. It has up to 16 analog inputs and two analog outputs with 12-bit resolution. The PCI-MIO-16E-4 delivers high
performance and reliable data acquisition capabilities to the entire system. It also provides the superior integration with LabVIEW [NI 1998].

![PCI-MIO-16E-4 Data Acquisition Board](image)

**Figure 3.8. PCI-MIO-16E-4 Data Acquisition Board**

### 3.2.4 Configuration of Data Acquisition System

All of the hardware used with LabVIEW is configured through the software tool named Measurement & Automation Explorer (MAX). MAX can configure the hardware and software for LabVIEW, view the devices and instruments connected in the computer system and execute the system diagnostics [NI 2000].

The Device and Interface folder shows all of the devices and instruments connected in the system and it configures them by choosing properties. Another folder, named Data Neighborhood, assign the channels and values to use throughout data acquisition. Once all of the modules and channels have been configured, they can be tested using the button called Test Panel.
3.3 Image Acquisition System

Similar to the Data Acquisition System, the basic flow of the Image Acquisition System begins with capturing images from the camera. The images captured are then passed to the Image Acquisition Board. Finally, the LabVIEW application can acquire and display the image using the driver for the Image Acquisition Board. Figure 3.10 shows the simple diagram of Image Acquisition System.
Figure 3.10. Process of Image Acquisition

The Image Acquisition System mainly consists of three parts: Video Camera, Image Acquisition Board, and Driver/Software.

3.3.1 Video Camera

There are two kinds of video cameras: Analog and Digital [NI 2001]. Digital cameras have several advantages over analog cameras. Digital video is more resistant to noise during transmission than analog video. Digital cameras also provide better accuracy, larger image sizes, faster frame rates and higher pixel resolutions than analog video cameras. Digital cameras can be further classified into parallel digital, Camera Link, and IEEE-1394. Considering the following specifications, Sony DFW-V500 IEEE-1394 Camera is chosen for the demonstration of the temperature system.

- Compatibility with LabVIEW
- Integration with NI Measurement & Automation Explorer (MAX)
- Low-cost, cross-platform solution standardized for transporting all types of digital data
- High-speed data transfer rate
• Mobility and ease of use

Figure 3.11. Sony DFW-V500 IEEE-1394 Camera and Image Acquisition Board

3.3.2 Image Acquisition Board

The image acquisition board connects between the video camera and the computer and transmits image data to the computer. Any kinds of image acquisition board supporting IEEE-1394 standard are compatible with Sony DFW-V500 Camera. Among the IEEE-1394 standard boards, Adaptec FireConnect 4300 model that supports 400Mb/s-transmission rate is used.

3.3.3 Driver and Sub-Programs

The *NI-IMAQ for 1394 software package* is needed to acquire images in LabVIEW from the IEEE-1394 Image Acquisition Board and the Video Camera [NI 2001].

The NI-IMAQ for 1394 software package consists of *IN-IMAQ driver* for IEEE-1394 and *IMAQ Vision software*. The IN-IMAQ driver for IEEE-1394 serves as the
interface between the LabVIEW application in the computer and the Image Acquisition Board. The IMAQ Vision software provides a series of sub-programs or sub-functions called *NI-IMAQ 1394 VIs*. The NI-IMAQ 1394 VIs provides the critical operational functions through the IN-IMAQ driver for IEEE-1394 so it supports the convenient way to develop the applications for manipulating the images.

Figure 3.12 illustrates the architecture of the NI-IMAQ for 1394 software package [NI 2001]. Kernel in Figure 1 shows the structure of NI-IMAQ driver for IEEE-1394 and Application Level shows how to develop the applications using IMAQ Vision software in LabVIEW.
3.3.4 Configuration of the Camera and the Image Acquisition Board

National Instruments Measurement & Automation Explorer (MAX) is available for configuring cameras (Figure 3.13). After MAX has launched, the properties of the camera can be set by right-clicking on the camera name in the IMAQ IEEE-1394 Devices folder in the MAX configuration tree. A user can select Properties from the pop-up menu and adjust the attributes of the camera and click OK to save the camera file. The Camera name information is stored in an interface (.icd) file, which the LabVIEW VIs use to select a camera and its supported attributes.

After finishing the configuration of the camera and image acquisition board, it can be tested by clicking the Snap or Grab function of MAX.

![Figure 3.13. MAX for Configuring Image Acquisition System](image)

3.4 Software Design

Virtual Engineering Lab consists of two main applications: LabVIEW Application for the Temperature System and a Web Application. The LabVIEW application acquires raw data from the equipment connected with the computer, manipulates and analyzes the data,
and sends data to the Web browser for display. The Web application allows students to
login to explore the experiments of the LabVIEW application and it also allows
instructors to control student’s access and add or delete LabVIEW applications

3.4.1 LabVIEW as a Programming Language

To understand the software design of LabVIEW application, it is necessary to
have the knowledge of the LabVIEW as a programming language. LabVIEW was
developed for high-performance scientific and engineering applications by National
Instrument Corporation. It has several great features and abilities such as graphical
programming and excellent reusability, maintainability, and extendibility to reduce the
application developing time and the cost for development or maintenance of applications.

3.4.1.1 Graphical programming

Compared to other traditional programming languages, the most obvious
difference is that LabVIEW uses graphical icons instead of lines of text to develop an
application [NI 1999]. It is easy to develop programs. Clicking any icon is enough to
create a new application There is not even one line of a text code in a program so it might
confuse the programmers that are familiar with traditional programming languages like
C, C++.

The LabVIEW consists of the front panel and the block diagram [Bishop 2001]. The
front panel is the user interface that constructs inputs and outputs. The block diagram
represents actual executable code with inputs and outputs made in the front panel. Figure
3.14 shows how inputs and outputs can be made in the front panel and Figure 3.15 shows the block diagram after the front panel was constructed.

Figure 3.14. Front Panel constructing Graphical User Interface

Figure 3.15. Block Diagram Corresponding to Front Panel
Using the inputs and outputs constructed, an executable program can be coded with functions such as +, -, while loop, for loop, etc. in the block diagram. Figure 3.16 shows how an executable program is coded in the block diagram.

![Block Diagram Constructing an Executable Program](image)

**3.4.1.2 Reusability**

Reusability of software can be defined as the ability to use a software module for purposes other than the one it was originally created for and to create new software modules by extending existing modules [Scott 2000]. LabVIEW programs are called virtual instruments (VIs) that must have at least one input and one output. Any VI can be a stand-alone program and it can be a subroutine of other VIs at the same time. Figure
3.17 shows the front panel and the block diagram of *ave.vi* when it was executed as a stand-alone program. This program takes two numbers as inputs and generate an average of two numbers as an output.

![Figure 3.17. Front Panel and Block Diagram of ave.vi](image)

The *ave.vi* can also be a subroutine of another VI. The Figure 3.18 shows the new VI that calculates the average of two inputs but uses the *ave.vi* as a subroutine.

![Figure 3.18. Front Panel and Block Diagram of New Program using ave.vi as Subroutine](image)

LabVIEW provides excellent reusability to develop new programs using the feature that any VI can be a stand-alone program and subroutine at the same time.
3.4.1.3 Maintainability

Maintainability can be defined as the ability to improve software modules without destroying existing uses of the module [Scott 2000]. When a programmer uses a subroutine to develop a new program, the programmer doesn’t need to consider the detail of the subroutine. The programmer just needs to know what the inputs are and what the outputs are. Whenever the inputs and outputs are not changed, the subroutine can be executed correctly in the other programs that have called the subroutine even if the algorithm of the subroutine has been changed to improve the implementation.

LabVIEW supports good maintainability to improve existing programs. In the \textit{ave.vi} example, even though the algorithm to calculate average has been changed from arithmetic average ($a+b$) to geometric average ($\sqrt{ab}$) but the attribute and the number of inputs and output have not been changed, the \textit{ave.vi} can be still working in the other VIs using the \textit{ave.vi}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{ave.vi.png}
\caption{ave.vi after Changing from Arithmetic Ave to Geometric Ave}
\end{figure}
3.4.1.4 Extendibility

Extendibility can be defined as the ability to add new features to existing software components [Scott 2000]. It is easy to add new functions to existing VIs. Figure 3.21 and 3.22 show the easiness of the extendibility in LabVIEW when a new function checking if two inputs are equal needs to be added at the VI that calculates the average of two inputs.

Figure 3.21. Block Diagram Adding the New Function Checking if Two Inputs are Equal
3.4.2 LabVIEW Application of the Temperature System

3.4.2.1 Front Panel of the Temperature System

To develop the LabVIEW application of the temperature system, the front panel must be constructed first to decide the user interface for inputs and outputs. Figure 3.23 shows the completed front panel of the temperature system.

Figure 3.23. Front Panel of Temperature System
The list of inputs is the following:

- **Acquisition Switch**
  
  Input Type: On / Off
  
  Function: Start or stop reading the current temperature from a thermocouple

- **Update Period Input Field**
  
  Input Type: Number
  
  Function: Decide the interval of reading the current temperature

- **Analysis Switch**
  
  Input Type: On / Off
  
  Function: Start or stop the analysis functions (mean, standard deviation, temperature history, and histogram)

- **Low Limit Input Field and Dial**
  
  Input Type: Number
  
  Function: Set the low temperature limit for a warning

- **High Limit Input Field and Dial**
  
  Input Type: Number
  
  Function: Set the high temperature limit for a warning

The list of outputs is the following:

- **Thermometer and Output Field for Current Temperature**
  
  Output Type: Number
  
  Function: Display the current temperature
• **Text**

  Output Type: Text (UNDER TEMP / OVER TEMP / none)

  Function: Display a warning message if the current temperature is below the low limit or over the high limit

• **Output Field and Vertical Fill Slide for Mean**

  Output Type: Number

  Function: Display the mean of the measured temperature

• **Output Field and Vertical Fill Slide for Standard Deviation**

  Output Type: Number

  Function: Display the standard deviation of the measured temperature

• **Graph for Temperature History**

  Output Type: Graph

  Function: Display the history of the measured temperature before 50 sec from the current time

• **Graph for Histogram**

  Output Type: Graph

  Function: Display the histogram of the measured temperature

• **Picture Frame**

  Output Type: Image

  Function: Display images captured by video camera

• **Meter**

  Output Type: Number

  Function: Display the image frame rate sent from video camera
### 3.4.2.2 Block Diagram of the Temperature System

After finishing the Front Panel, the Block Diagram is made to specify how the components should work. The block diagram, which represents the executable code, consists of icons that manipulate data sent through wires.

There are two main modules for the data acquisition for temperature system. One module (Figure 3.24), labeled *Acquisition Loop*, acquires the signal from Data Acquisition System if *Acquisition Switch* on the front panel turns on, manipulates the signal to convert temperature and displays the temperature measured at the *Thermometer and Output Field for Current Temperature*. If the temperature is not in the certain range that a user defined at the *Low or High Limit Input Field and Dial* on the front panel, this module sends a signal to the LED for the warning and displays the warning message at the *Text*. 
Figure 3.24. Acquisition Loop Module of Data Acquisition

*Analysis Structure module* (Figure 3.25) calculates the mean and standard deviation of temperature measured and displays them at the **Output Field** and **Vertical Fill Slide for Mean and Standard Deviation**, respectively. This module also stores the temperature measured and displays the history and histogram of the temperature at the **Graph for Temperature History** and **Graph for Histogram** on the front panel.
Figure 3.25 shows the *Image Acquisition module* that acquires images from the video camera and display the images at the **Picture Frame** on the front panel. The image acquisition module uses the Grab acquisition type to acquire the image. The Grab is a continuous, high-speed acquisition of data to a single buffer. Two main sub-functions named IMAQ1394 Grab Setup and IMAQ1394 Grab Acquire are used for a grab acquisition in LabVIEW. IMAQ1394 Grab Setup initializes the acquisition and starts capturing the image to an internal software buffer. IMAQ1394 Grab Acquire copies the image currently stored in the internal buffer to a LabVIEW image buffer.

This module also calculate the image frame rate sent from the video camera and display it at the **Meter** on the front panel.
3.4.3 Web Application

This application is a cross-platform Web application, which is located between clients and the LabVIEW Server. The Web application is designed to serve as a communication tool between students and instructors, to control the secure access of students for experiments, and add /delete the links for LabVIEW applications.

The Level 1 Transition Diagram for the Web application (Figure 3.27) shows how the Web pages consist of and how they are related, briefly.
Figure 3.27. Level 1 Transition Diagram of the Web Application

The instructor Pages can be explained with the *Level 2 Transition Diagram* for the Instructor Pages (Figure 3.28).

Figure 3.28. Level 2 Transition Diagram of the Instructor Pages
3.4.3.1 Programming Languages

The Web application is coded with HTML, JavaScript and PHP programming languages. It can be accessed using any major browser that supports HTML 4.0 and JavaScript.

In the project, HTML 4 is used to generate the static Web pages and JavaScript is chosen to create dynamic Web pages on the client side because HTML 4 and JavaScript are compatible with many operating systems and supported by both Netscape and Internet Explorer.

PHP is used to generate server-side scripts to communicate between the Web server and the data file. Like any other CGI program, PHP [Meloni 2000] can generate content on the fly, or create a Web interface for adding, deleting, and modifying elements within the data file. It can work with just about any combination of Web server and operating system. PHP user authentication can restrict access to the Web site. The goal of the language is to allow Web developers to write dynamically generated pages quickly.

3.4.3.2 Login Page

The Login Page allows users to input User ID and Password and sends those informations to Authentication Page.

3.4.3.3 Authentication Page

The Authentication Page reads userfile to obtain a list of user IDs and passwords and compare the user input from the Login Page with a list of user IDs and passwords.
When there is no matching pair, the **Authentication Page** sends an error message to the user browser. If the user ID and password from the **Login Page** is in the list of user IDs and passwords in `userfile`, the **Session** is started. Depending on the status of the user logged in, **Authentication Page** shows the **Student Page** or the **Instructor Pages**.

### 3.4.3.4 Logout Page

The **Logout Page** deletes the session ID from `sessionfile` and terminate the **Session** when users choose the Logout menu to terminate the Web application.

### 3.4.3.5 Session

After finishing authentication with the user ID and password, the **Session** is going to work to keep tracking the user and prevent unauthorized users. The **Student Page** and the Instructor Pages (**Lab Info Page, Add New Lab Page, Modify / Delete Lab Page** etc.) call the `session_varify()` function to execute the **Session** at the top of each page. The following example shows how to use `session_varify()` function.

Example:

```php
require("session.php");

$session_error = session_varify($virtualLab);

if($session_error != 0){
    echo("session error:$session_error");
    exit();
}
```
The Session obtains the list of session IDs from sessionfile and compares the session ID from the user's cookie and the list of session IDs. The Session is also checking the session time if there is any session time that is more than three hours long.

3.4.3.6 Student Page

The Student Page reads the lab information from the labfile and the announcement from the announcementfile. Using the data from labfile and announcementfile, the Student Page displays the announcement, lab ID, title, description, and button to request the lab experiment.

3.4.3.7 Lab Info Page

The Lab Info Page is for instructors to see the information of lab. The Lab Info Page reads the information from labfile and displays the lab ID, title, description, and button to move the Modify / Delete Lab Page.

3.4.3.8 Modify /Delete Lab Page

The Modify / Delete Lab Page receives the request from the Lab Info Page to display the information for a specific lab ID. The data in the labfile is changed based on the modify or delete request.

3.4.3.9 Add New Lab Page

When an instructor sends the add new lab request with title, description, and URL of LabVIEW Server, the title, description and URL will be stored in labfile.
3.4.3.10 User Info Page

The **User Info Page** reads the data from *userfile* and display the user ID and the user status and an instructor can choose the modify or delete button.

The **User Info Page** will check whether there is the same user ID in *userfile* when it receives the modify-user or delete-user request. If there is the same user ID, the information of users will be changed from *userfile* based on the request type. If not, it will send an error message to the browser of the instructor.

3.4.3.11 Add New User Page

When an instructor sends the add-user request with user ID, password and status, the **Add New User Page** checks whether the user ID is at the list of user IDs in *userfile*. If there is a matching user ID in the list, an error message will be sent to the instructor. If not, the user ID, password and status will be stored in *userfile*.

3.4.3.12 Announcement Page

The Announcement page reads the data from *announcementfile* and displays the announcement. When an instructor has changed the announcement and click the submit button, the changed announcement is changed in *announcementfile*
4. Evaluation and Results

The Virtual Engineering Technology Laboratory enables engineering technology students to perform laboratory experiments online. This project showed that LabVIEW supports an excellent ability to develop distance learning applications. Since graphical programming provides a convenient function to embody the visualization of applications, LabVIEW developers can maximize the usability of applications when they make the front panels (graphical user interfaces) without concern to the block diagrams (actual executable codes). Employing the video camera to monitor experiments can increase the learn-ability of students. In the case of the temperature system, students can also verify whether the system is working correctly or not. For instance, if a hot material is applied to the thermocouple, the temperature must increase. If the current temperature measured is over the high temperature limit, the LED must be turned on.

Web applications provide good functionality for developing a distance learning system. To test the performance of the Virtual Lab and the temperature measuring experiment, a black box testing was used to know if each function works correctly and under several different environments. Each function was able to handle the combination of several sets of inputs and response correctly to errors. For instance, when the session is over, the Web application allows users to access other Web pages and force them to move to Login Page. Virtual Engineering Technology Laboratory Experiment can be executed on both Netscape and Internet Explorer which are the major tools that students use. The temperature system experiment and the camera system produced expected results.
5. Conclusion

The Virtual Engineering Technology Laboratory demonstrates the efficiency of performing experiments online using a Web browser and a LabVIEW application. The LabVIEW application gives students access to real equipment and devices. This environment offers improved safety conditions, reduced cost, and broad base access for all students and multi-users.

This project has accomplished the following tasks:

1. Construction a Data Acquisition System for the Temperature System using LabVIEW Programming Toolkits,
2. Construction an Image Acquisition System for the Temperature System using LabVIEW Programming Toolkits,
3. Development of a Web application to control the students’ access and to provide a convenient facility for instructor and students using PHP.

With these components of the Virtual Engineering Laboratory Experiment, students can explore hands-on experience without any concern for the limitation from the use of a real engineering lab through the Intranet.
6. Future Work

The Virtual Engineering Laboratory Experiment allows students to explore computer based distance learning with one experiment, the temperature system. Other labVIEW applications need to be developed for students. Also, access from anywhere the Internet is available should be provided. This will require addressing the firewall issues described earlier in this report.
BIBLIOGRAPHY AND REFERENCES


[Deliwala, 1997] Deliwala, Siddharth. Introduction to Data Acquisition System and LabVIEW, Available from pender.ee.upenn.edu/rca/software/Labview/daqvOverview.html (Visited April 11)


APPENDICES

1. Login Page
************************************************************************
<html>
<head><title>Log in</title></head>
<body topmargin=0 leftmargin=0 marginheight=0
marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgcolor=#999966>
<tr><td width=100%><p align=center><img border=0
src=title.gif width=443
height=132></td></tr></table><br><br>
<form name="logIn" method=post action="authn.php">
<table align="center">
<tr><td><b>User ID:</b></td>
<td><input name="userID" type="text"></td></tr>
<tr><td><b>Password:</b></td>
<td><input name="password" type="password"></td></tr>
<tr><td colspan=2 align="center"><input name="enter"
type="submit" value="LOG IN"></td></tr>
</table>
</form>
<br><br><br><br><br>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgcolor=#999966>
<tr><td width=100% align=center>
<img border="0" src="logos.gif" width="49"
height="21">
<b><font color="#002800" face="Times New Roman" size=-1>Engineering Technology, Texas A&M - Corpus
Christi</b></td></tr></table><br><br>
</body>
</html>
************************************************************************

2. Authentication Page
************************************************************************
<?php
require("session.php");
if(!$file=fopen("userfile","r")){
    echo("Can't open the session file");
    exit();
}
$arr=file("userfile");
fclose($file);
for($i=1; $i<count($arr); $i++){
    $arrsplit = split(" #&\# ", trim($arr[$i]));
    for($j=0; $j<count($arrsplit); $j++){
        $user[$i][$j] = $arrsplit[$j];
    }
}

$index=1;
while(($user[$index][0] != $userID) && ($index<count($arr))) $index++;
if($index >= count($arr)){
    echo("<html><head></head>");
    echo("<body topmargin=0 leftmargin=0 marginheight=0
marginwidth=0">
    echo("<table border=0 cellPadding=0 cellSpacing=0
width=100% bgcolor=#999966>");
    echo("<tr><td width=100%><p align=center><img border=0
src=title.gif width=443
height=132></td></tr></table><br><br>
    echo("<h2 align=center>Not valid username<br>
    echo("<a href=login.php>Try
    echo("<a href=login.php>Try
    again</a>"</h2></body></html>");
    exit();
} else if($user[$index][1] != $password){
    echo("<html><head></head>");
    echo("<body topmargin=0 leftmargin=0 marginheight=0
marginwidth=0">
    echo("<table border=0 cellPadding=0 cellSpacing=0
width=100% bgcolor=#999966>");
    echo("<tr><td width=100%><p align=center><img border=0
src=title.gif width=443
height=132></td></tr></table><br><br>
    echo("<h2 align=center>Not valid password<br>
    echo("<a href=login.php>Try
    echo("<a href=login.php>Try
    again</a>"</h2></body></html>");
    exit();
} else if($user[$index][2] == "instructor"){
    session_login($userID);
    echo("<SCRIPT>
";}
<?php
require("session.php");
session_logout($virtualLab);
?>

<html><head></head><body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bcolor=#999966><tr><td width=100%><p align=center><img border=0 src=title.gif width=443 height=132></td></tr></table><br><br>
<h2 align=center>Your session is finished: Logout successfully<br>Bye !!</h2>
</body></html>

************************************************************************
3. Logout Page
************************************************************************

4. Session
************************************************************************
function session_login($uid)
{
    $current = time();
    $random = $uid.$current;
    $sid = md5($random);
    setcookie("virtualLab",$sid);

    if(!($file=fopen("sessionfile","a"))){
        echo("Can't open the session file");
        exit();
    }
    fwrite($file,$sid . "\n");
    fwrite($file,$current . "\n");
    fclose($file);

    if(!($file=fopen("sessionfile","r"))){
        echo("Can't open the session file");
        exit();
    }
    $arr=file("sessionfile");
    fclose($file);
    $index=0;
    for($i=1; $i<count($arr); $i=$i+2){
        $difference = $current - $arr[$i];
        if($difference > 86400) {
            $deleteList[$index] = $arr[$i-1];
            $index++;
        }
    }
    if($index>0){
        if(!($file=fopen("sessionfile","r"))){
            echo("Can't open the session file");
            exit();
        }
        $arrDL=file("sessionfile");
        fclose($file);
        if(!($file=fopen("sessionfile","w"))){
            echo("Can't open the session file");
            exit();
        }
        fwrite($file,"\n");
        fclose($file);
        if(!($file=fopen("sessionfile","a"))){
            echo("Can't open the session file");
            exit();
        }
    
}
exit();

for($i=0; $i<count($arrDL); $i++){
    $found = false;
    for($k=0; $k<$index; $k++){
        if(trim($deleteList[$k]) == trim($arrDL[$i])){
            $found = true;
        }
    }
    if($found) $i++;
    else fwrite($file,$arrDL[$i]);
}
fclose($file);

function session_varify($sid)
{
    if(!$file=fopen("sessionfile","r")){
        echo("Can't open the session file");
        exit();
    }
    $arr=file("sessionfile");
    fclose($file);
    $index=0;
    while(($index<count($arr)) && ($sid != trim($arr[$index]))){
        $index++;
    }
    if($index >= count($arr)){
        echo("<SCRIPT>
        echo("<!---------- JavaScript begins...
        echo("function right(){
        echo("parent.location.href="login.php"
        echo("})
        echo("//JavaScript ends ---------->
        echo("</SCRIPT>
        return 100;
    }
    else return 0;
}

function session_logout($sid)
{
    setcookie("virtualLab");
if(!$file=fopen("sessionfile","r")){
    echo("Can't open the session file");
    exit();
}
$arr=file("sessionfile");
fclose($file);

if(!$file=fopen("sessionfile","w"){
    echo("Can't open the session file");
    exit();
}
fwrite($file,"\n");
fclose($file);
if(!$file=fopen("sessionfile","a"){
    echo("Can't open the session file");
    exit();
}
for($i=0; $i<count($arr); $i++){
    if($sid == trim($arr[$i])) $i++;
    else fwrite($file,$arr[$i]);
}
fclose($file);
?
************************************************************************
5. Student page
************************************************************************
<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("session error:$session_error");
    exit();
}
if(!$file=fopen("labfile","r"){
    echo("Can't open the file");
    exit();
}
$arr=file("labfile");
fclose($file);
$numOfLab = count($arr) - 1;
for($i=1; $i<count($arr); $i++){
    $arrsplit = split(" #& #", trim($arr[$i]));
}
for($j=0; $j<count($arrsplit); $j++){
    $lab[$i][$j] = $arrsplit[$j];
}
}
}

for($i=1; $i<count($arr); $i++){  
$qury = "n" . $i;
    $quryLabID = $lab[$i][0];
    $quryURL = "http://" . $lab[$i][1];
    if(isset($$qury)){
        echo("<script>
        echo("<!--
        echo("function show_nt(){\n        echo("var ntWin;\n        
        echo("ntWin=window.open(\"$quryURL\", \"LAB\", \"width=500,height=500,scrollbars=yes,resizable=yes\")\n        echo("ntWin.focus();\n        echo("\n        "--\n        echo("</script>\n        echo("</head>\n        
        if(!$file=fopen("announcement","r")){
            echo("Can't open the file");
            exit();
        }
        
        $arrAnnounce=file("announcement");
        fclose($file);
        
exco("<table align=center>");
echo("<tr><td align=center><font color=#800000 size=+1>***
Announcement ***</font></td></tr>");
echo("<tr><td>");
for($i=0; $i<count($arrAnnounce); $i++){
    $announcement = trim($arrAnnounce[$i]);
    echo("$announcement<br>");
}
echo("</td></tr></table>");
echo("<br><br>

table align=center border=1
bordercolor=#003300>
tr><td background=tbg1.gif><b><font color=#002800
face=Verdana size=+1>Lab Name</font></b></td>
<td background=tbg1.gif><b><font color=#002800
face=Verdana size=+1>Description</font></b></td>
<td background=tbg1.gif><b><font color=#002800
face=Verdana size=+1>Go to lab</font></b></td></tr>
<form name=list action=student_page.php
method=post>
for($i=1; $i<count($arr); $i++){
    $avail = $lab[$i][3];
    $title = $lab[$i][2];
    $labID = $lab[$i][0];
    $name = "n" . $i;
    $description = $lab[$i][4];
    if($avail == "on")
        echo("<tr><td>$title</td><td>$description</td><td
align=center><input type=submit name=$name
value=$labID></td></tr>
<form name=logout action=logout.php method=post>
<p align=center><font size=+1><b>Click Log out button to
finish the lab experiment</b></font></p>
<table align=center>
<tr><td input type=submit value="Log out"></td></tr>
</table>
</form>
<br><br><br><br><br>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgroundColor=#999966>
6. Lab Info Page

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("session error:$session_error");
    exit();
}

//read lab data
if(!$file=fopen("labfile","r")){
    echo("Can't open the file");
    exit();
}
$arr=file("labfile");
fclose($file);

$numOfLab = count($arr) - 1;
for($i=1; $i<count($arr); $i++){
    $arrsplit = split(" #&# ", trim($arr[$i]));
    for($j=0; $j<count($arrsplit); $j++){
        $lab[$i][$j] = $arrsplit[$j];
    }
}
?>

<html>
<head><title>Lab Info</title></head>
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgcolor=#999966>
<?php
if(!$file=fopen("announcement","r")){
    echo("Can't open the file");
    exit();
}
$arrAnnounce=file("announcement");
fclose($file);

echo("<table align=center>");
echo("<tr><td align=center><font color=#800000 size=+1>*** Announcement ***</font></td></tr>");
echo("<tr>");
for($i=0; $i<count($arrAnnounce); $i++){
    $announcement = trim($arrAnnounce[$i]);
    echo("$announcement<br>");
}
echo(""/	d></tr></table>");
echo(""/

<form action="labInfoChange_page.php" method="post"><?
for($i=1; $i<count($arr); $i++){
    $avail = $lab[$i][3];
    if($avail == "on")
        $avail_web = "available";
    else
        $avail_web = "not available";

    $title = $lab[$i][2];
    $url = "http://" . $lab[$i][1];
    $labID = $lab[$i][0];
    $description = $lab[$i][4];
7. Modify / Delete Lab Page

```php
<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("session error:$session_error");
    exit();
}

//read lab data
if(!$file=fopen("labfile","r"){ echo("Can't open the file"); exit();
}
$arr=file("labfile");
fclose($file);

$numOfLab = count($arr) - 1;
for($i=1; $i<count($arr); $i++){
    $arrsplit = split(" #& # ", trim($arr[$i]));
    for($j=0; $j<count($arrsplit); $j++){ $lab[$i][$j] = $arrsplit[$j];
}
}

$index = 1;
while(($lab[$index][0] != $labID) & ($index < count($arr)))
    $index++;
if($index >= count($arr)){
```
8. Add New Lab Page

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("Error: there is no such labID");
    exit();
}

$description = ereg_replace ("\\n", "<br>", $description);
    . $description . " #&# end";

if(!$file=fopen("labfile","w")){
    echo("Can't open the file");
    exit();
}
$head = "0.id #&# 1.url #&# 2.title #&# 3.available #&#
        4.short Description #&# end";
fwrite($file,$head . "\n");
fclose($file);

if(!$file=fopen("labfile","a")){
    echo("Can't open the file");
    exit();
}
for($i=1; $i<count($arr); $i++) {
    if($i == $index)
        fwrite($file, $contents . "\n");
    else
        fwrite($file, $arr[$i]);
}
fclose($file);
?>
<html><head></head>
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgcolor=#999966><tr><td width=100%><p align=center><img
border=0 src=title.gif width=443
height=132></td></tr></table><br><br>
<h2 align=center>Lab Info has been changed</h2>
</body></html>
************************************************************************
8. Add New Lab Page
************************************************************************

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){


}
9. User Info Page

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("session error:$session_error");
    exit();
}

//read lab data
if(!$file=fopen("userfile","r")){
    echo("Can't open the file");
    exit();
}
?>
exit();
}
$arr=file("userfile");
fclose($file);

$numOfLab = count($arr) - 1;
for($i=1; $i<count($arr); $i++){  
    $arrsplit = split(" #& #", trim($arr[$i]));
    for($j=0; $j<count($arrsplit); $j++){  
        $tdArr[$i][$j] = $arrsplit[$j];
    }
}
?>

<html>
<head><title>User Info</title></head>
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100%
bgcolor=#999966>
<tr><td width=100%><p align=center><img border=0
src=title.gif width=443
height=132></td></tr></table><br><br>
<table align=center border=1>
<tr><td><b>User ID</b></td>
<td><b>Status</b></td>
<td><b>New Password</b></td>
<td><b>Change Info</b></td>
<td width=20> </td>
<td><b>Delete User</b></td></tr>
<form action=userInfoChange_page.php method=post>
<?
for($i=1; $i<count($arr); $i++){  
    $userID = $tdArr[$i][0];
    $statusValue = $tdArr[$i][2];
    $newPWD = "newPWD" . $userID;
    $status = "s" . $userID;
    echo("<td>$userID</td>");
    if($statusValue == "instructor")
        echo("<td><input type=radio name=$status
value=instructor CHECKED>Instructor<br> <input type=radio
name=$status value=student>Student</td>");
    else

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
   echo("session error:$session_error");
   exit();
}
if(!$file=fopen("userfile","r")){
   echo("Can't open the file");
   exit();
}
$arr=file("userfile");
fclose($file);

$numOfLab = count($arr) - 1;
for($i=1; $i<count($arr); $i++){  // Split the row
   $arrsplit = split(" #& # ", trim($arr[$i]));
   for($j=0; $j<count($arrsplit); $j++){  // Array the split
      $stdArr[$i][$j] = $arrsplit[$j];
   }
}

$sameUserID = false;
for($i=1; $i<count($arr); $i++){  // Loop through the file
   // Check existing
   // Add the new one
}
?>
"
if($tdArr[$i][0] == $userID)
    $sameUserID = true;
}

if($sameUserID == true){
    echo("<html><head></head>");
    echo("<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>"
    echo("<table border=0 cellPadding=0 cellSpacing=0 width=100% bgcolor=#999966><tr><td width=100%><p align=center><img border=0 src=title.gif width=443 height=132></td></tr></table><br><br>
    echo("<h2 align=center>There is the same user name. Try again</h2></body></html>"));
    exit();
} else {
    $contents = $userID . " #&# " . $pwd . " #&# " . $status . " #&# end";
    if(!$file=fopen("userfile","a")){
        echo("Can't open the file");
        exit();
    }
    fwrite($file,$contents . "\n");
    fclose($file);
}

<html><head></head>
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
<table border=0 cellPadding=0 cellSpacing=0 width=100% bgcolor=#999966><tr><td width=100%><p align=center><img border=0 src=title.gif width=443 height=132></td></tr></table><br><br>
<h2 align=center>New User added</h2>
</body></html>

************************************************************************
11. Announcement Page
************************************************************************

<?php
require("session.php");

$session_error = session_varify($virtualLab);
if($session_error != 0){
    echo("session error:$session_error");
    exit();
}