Deanship of Graduate Studies
AL-Quds University

Collaborative Working Environments for Web-Based Remote Experimentation

Eiman Mohammad Mahmoud Abdelghani

M.Sc. Master

Jerusalem – Palestine

1430-2009

Collaborative Working Environments for Web-Based Remote Experimentation

Prepared By:
Eiman Mohammad Mahmoud Abdelghani

B.Sc. Omar-Almoktar University / Libya

Supervisor: Dr. Salaheddin Odeh

A thesis submitted in partial fulfillment of requirement for the degree of master of computer and electronic engineering - Faculty of Engineering - Graduate Studies
Al-Quds University

1430-2009

Al-Quds University
Deanship of Graduate Studies
Electronic and Computer Engineering Master Program

Thesis Approval

Collaborative Working Environments for Web-Based Remote Experimentation

Prepared by: Eiman Mohammad Mahmoud Abdelghani
Registration No: 20510260

Supervisor: Dr. Salaheddin Odeh

Master thesis submitted and accepted, date: …………………………….
The names and signatures of the examining committee members as follows:

1. **Head of committee: Dr. Salaheddin Odeh signature…………………**
2. **Internal examiner : Dr. Labib Arafeh signature…………………**
3. **External examiner: Dr. Radwan Tahboub signature…………………**

Jerusalem-Palestine

1430-2009

Dedication

I am eternally grateful to my parents for everything they have done for me, to my sister, to my brothers

Eiman Mohammad Mahmoud Abdelghani

Declaration:

I Certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed:

Eiman Mohammad Mahmoud Abdelghani

Date:

Acknowledgements

I am indebted to my supervisor, Dr. Salaheddin Odeh, for assisting me during the preparation of this work and motivation to reach new heights. Thank you very much.

Thanks also to all staff members of the Department of Computer and Electronic Engineering / Graduate Studies / Al-Quds University.

A special thank goes to my family, without their support this work would not have been possible, and for everybody who was supported me.

Collaborative Working Environments for Web-Based Remote Experimentation

Prepared by: Eiman Abdelghani

Supervisor: Dr. Salaheddin Odeh

Abstract

Collaborative working environments for distance education can be considered as a more generic form of contemporary remote labs. Nowadays, the majority of existing real laboratories are not constructed to allow the involved participants to collaborate in real time. To make this revolutionary learning environment possible we must allow the different participants to communicate and to coordinate with each other in a productive manner. Production rate and efficiency gained through synchronous communication and coordination between the different partners, which mean that users can carry out an experiment simultaneously. Within this process, coordination can be accomplished by chat tools. In recent times, multi-user environments are successfully applied in many applications such as air traffic control systems, team-oriented military systems, chat-text tools, and multi-player games. Thus, understanding the ideas and techniques behind these systems could be of great importance of the contribution of ideas to our e-learning environment for collaborative working.

In this thesis, collaborative working environments from theoretical and practical perspectives are considered to build an effective collaborative real laboratory, which allows two students or more to conduct remote experiments at the same time as a team. In order to achieve this goal, we have proposed distributed system architecture to enable the collaborative work among students. This system is organized into subsystems and components and allows the end-users (tutor and students) to interact with the system through a web-based user-interface, enabling them to complete their tasks; however, this architecture have been built around an existing laboratory components and equipments in a general university/ engineering college. Unfortunately, there are many problems that still exist in remote experiments of engineering studies, such as not finding the proper assistance while performing an experiment, distracting students working on an experiment because of not finding a proper feedback from other students, and inequality in task division between them. Therefore, the students will lose motivation and the experiment could fail, leading to ineffective learning.

The architecture applied to this system follows the simplest client-server architecture, where an application is organized as a server and a set of clients. Enabling students to obtain an automated help by either a human tutor or a rule-based e-tutor for the purpose of student support in remote experiment environments, we had to facilitate a synchronous interaction between students and tutor achieved by chat tool. Effective collaboration between students is obtained by using color coding to distinguish their contributions from each other.

The software-technical implementation of the system have been realized by using the powerful Microsoft .NET, which offers a complete integrated developmental environment (IDE) with a wide collection of products and technologies. As a result of the software and hardware development process was a web-based collaborative working environment, which can be interactively used by students through a web-based user-interface. This GUI facilitates the remote control and access of various instruments and experiment setups. Groups of students are now independently able to coordinate and to implement the experiment at any time and from anywhere, including regulating the work positively.

Table of contents

List of Tables

List of Figures

List of Appendices

Publications

1. Salaheddin Odeh and Eiman Ketaneh,*" Collaborative Working e-Learning Environments Supported by Rule-Based e-Tutor,"* iJOE International Journal of Online Engineering – www.i-joe.org, Volume 3, Issue 4, 2007.

Chapter 1

Introduction and problem statements

1.1 Introduction

Recently, the university can go to students and not vice versa; this idea is what is designated as distance learning. In this thesis, we will argue how two or more students can execute an experiment in real time, at the same time and on different places. People collaborate because doing so is satisfying or productive. Human connections have prompted millions of users to join by chat rooms peppered with humor. Without collaboration, many computer users have felt introversive and isolated, but as in any human community, there is also controversy and slander. In a collaborative environment, goals of collaboration are to allow two or more students in conjunction with a tutor to communicate with each other at the same time, where they are distributed in space. Before beginning to discuss how we can improve and develop the field of collaborative working e-learning, it is very necessary to study and analyze other socio-technical systems, where collaboration is successfully realized such as:

- Air traffic control systems [1].
- Team-oriented military systems [2].
- Chat tools [3], [4].
- Share desktop and share applications [5].

1.2 Problem statement

Complex problem solving tasks without instructional support will often demand too much from the students and will lead to ineffective learning [6]. As long as students need instructions and help in solving scientific problems, the learning environment shall provide knowledge for solving these problems. Instructional support is an important element especially in web-based learning settings; therefore, remote laboratories should provide support for students. In remote laboratories, a tele-tutor communicates via synchronous or asynchronous communication tools with his students, resulting as a central role regarding instructional support.

One form of communication between students and tutors is asynchronous; this kind of communication has several disadvantages [6]:

- Students will not get any support from the tutor during the laboratory session; therefore, students might not able to find answers and solutions for their questions and problems in the experiment.
- Students have to accept time delays for getting answers to their questions, and therefore, if time delays are too long, they could lose their motivations.

By contrast, the synchronous communication bypasses the above mentioned problems:

- The synchronous communication restricts the time flexibility of the students.
- Upcoming problems can then be solved immediately by using chat tools and rule-based approach (see chapter 4).

Hands-on lab enables the students to discuss problems, where each student complements the other to achieve solutions easily. Although remote e-labs have several advantages (see chapter 2), distributed laboratories including these with collaborative characteristics suffer from some problems such as [7]:

- Inequality in task division between students.
- Students feel isolated because of lack of feed-back from other students.
- Groups without supervision will face a lot of difficult problems left unsolved.

According to previous studies, the majority of current existing real labs are not constructed to allow the participants to collaborate in real time and are not designed to support students through an automated help in any time [8], [9],[10].

From this point, our contribution in this thesis will be about the development of education through the use of distributed e-laboratories for collaborative working; so after our previous discussion, the main problem caused by the asynchronous approach is that if the student groups are not under enough control by their teachers, they could have a time delay for getting the answers to their questions.
Once the delay becomes too long, the students could lose their motivation, and could face many problems left unsolved. One way the discussed problems can be solved is by making a

remote human tutor to support students available over a distance via synchronous communication tools, such as chat tools that allows quicker assistance. However, a remote human tutor can not support students at any time and will not be available throughout the day.
Therefore, it is intended to implement an automated helping system in the form of a rule-based e-tutor for user support in remote experiment environments. A rule-based system embraces a rule-base including stored knowledge about the correct experiment configuration, (see chapter 4). The rules determine what should be done in different situations and are initially designed by a human experiment, where each rule has two parts: conditions and action.
Because of inequality in task division while executing an electronic experiment between students, color coding is used to distinguish between collaborated students regarding what connectors are added by a specific student; each student would be ordered a certain color to distinguish his/her work from others.

A collaborative lab is an open laboratory spanning multiple geographical areas where collaborators interact via electronic means [11]; however, distributed e-laboratories for collaborative working have designed to encourage closer relationships between students in an implemented lab's experiment remotely.

1.3 Objectives

The implementation of the developed e-lab was possible after having suggested a well-known architectural model, the client-server architecture, which fits into the designing of collaborative working e-learning environments [9], [12]. It facilitates what is called multi-user e-learning environment, where more than one student and a tutor have access to an experiment at the same time and from different places (see chapter 4). During an experiment, collaborated students are not only supported by a human tutor, but by a rule-based e-tutor as well.

One widely used technique to establish the synchronous communication in our collaborative working environment is chat tools in addition to visualization techniques; they make it possible to establish an effective interaction between students and tutor.

Hence, after we have implemented a proposed architecture to support collaborative working, the objectives we had are:

1- Providing the students with an automated help at any time.
2- Putting students in a real environment lab.
3- Distinguishing students’ contributions from each other.
4- Targeting the system at a real interaction between students and tutor.
5- Accessing the lab from any place and at any time.

Chapter 2

E-Learning Environments: Electronic Laboratory 'E-Laboratory'

2.1 Introduction

In most times, we gain knowledge on certain subjects or fields through the theoretical approach. In the field of engineering and science, lecturers deliver the knowledge of specific subjects for students through learning ideas and theories through the regular classroom; usually the concepts had done by examples and explanation of exercises.

In addition, the students are trying to increase their knowledge of what they learned in class room from related literature.

However, all the information theory that the students had learned difficult to understand as well that students have few opportunities to apply knowledge learned in class-contact lectures. In view of this, laboratory experiments play a vital and active role in the completion of theories and ideas learned in lectures. Where it is impossible to hide the give of knowledge from the lecturers for the students to be more effective with the help of experiments or what we call the practical approach [13], [14].

Development and the rapid growth of Web technologies and the wide use of the Internet could make teaching and learning through the Internet to be more effective in these days. Many universities have set up portals to offer an e-learning environment either as teaching aids to support conventional teaching approach or as a teaching medium for long-distance or off-campus programs and more orientation is towards the development of e-laboratories [13], [15], since the laboratory equipment is preparing to be controlled through the Internet.

The setting up of e-laboratories fits perfectly in an e-learning environment especially for engineering and scientific programs which could provide means for students to carry out experiments via the internet [13], [16], and [17]. In scientific and engineering education, the learning theories or concepts through practical approach or applications is a conclusive step of the learning process [18].

In most universities, identifying a particular time for the students to attend the laboratories for the implementation of practical experience with supervision by the lecturers gives the students few times to do their experiments.

Moreover, the students were restricted to specific time periods for access to the laboratory. For this reason, access to laboratories, are only allowed during week days and working hours. This issue, becomes immaterial if the laboratory equipment is linked to the internet and the students can remotely monitor and control the equipment from somewhere via the internet. The students can then perform the laboratory experiments at any preferred time, even during midnight or weekends [13], [19].

In the laboratory, instruments and equipment are inadequate when compared to the increasing number of students; in any case, it is very expensive to provide laboratory equipment to meet the increase in the number of students [20]. Moreover, many universities nowadays offer off-campus and distance learning programs such that the laboratory equipments or instruments are geographically located away from the student.

As the student needs to travel every day, to the laboratory, to implement the practical experiences using the devices there. This is impractical because of cost and time constraints. A lot of universities, published laboratory equipment on the Web page where it can be accessed from anywhere in the world [13].

In the e-laboratory, the key interest that it had used as effective tools and assistance in the teaching process, where they are processed through the Internet is to give practical examples and application through the laboratory experiments carried out by the lecturers. Applications of these experiments have found that they could be very effective operation to understand the student subjects learned. However, most of the lectures include a large number of engineering students per class. Thus, this hinders lecturers to clarify and provide assistance for each student [20]. So remote laboratories solve these problems, ability to offer effective solutions for teaching approaches, where the real physical equipments are actually being controlled and monitored.

Another advantage of an e-laboratory is that, universities around the world can share their laboratory resources with each other at minimum costs. Lecturers could offer a larger variety of practical demonstrations covering of different types of processes to their students without the need for their universities to spend large amount of money in buying these processes.

Students would be able to benefit from observing and taking part in laboratory experiments on a variety of different fields not available in their university laboratories [13], [18].

2.2 Internet as a tool for interactive learning

There is an already fact that the need for educational experiences is rapidly growing, and at the same time, the development of technology is rapidly changing the stage of education to which teaching and learning are taking new dimensions. Therefore, the flexibility provided by the Internet technology becomes important and interesting.

With the fast growth of the Internet, many educational universities began to use web as a new medium to assist the teaching and learning activities. Internet technology allow teachers and students continue with their minds. It let them try their ideas, either the Internet provides learning and teaching which involve interactions, either with students, teachers, the environment, or the learning material [21], [22].

Using the Internet in teaching makes the communication between the students and the instructor more easy and interesting. Particularly, using web in teaching makes it available for the students who prefer or require learning outside the classroom to study at their convenient time and place. The web has already been one of the most popular mediums for the deliverance of the course information [23].

Internet will help students develop their own cultural. In this way, it could help continuing self-education. It facilitates collaboration and interaction between students and teachers. Such collaboration efforts promote the participation of information and help to integrate learning experiences.

2.3 Instructional reasons for the use of the Internet

The Internet becomes increasingly important as a learning environment, and internet technology is rapidly being adopted in engineering education as a tool for enhancing the educational experience. Electronic learning abbreviated as e-learning has made possible by advances in technology, where it is to presume that the speed of technological change is set to continue over the next decade and beyond. The web puts a huge number of learning resources within reach of anyone with Internet access [24].

The use of the Internet to link individuals with others sharing common interests provides the learners for building communities that offer support, solidarity, information and social groups. Although the Internet provides exposure to diverse groups and ideas, people are most strongly drawn to online groups that share their interests and concerns [25].

In the past, university instructors always gave the students a lot of handouts. Nowadays, many instructors use web as a good mean to help them to organize course information by delivering information to the students. They gather course information, including schedule, assignment, study guides, and syllabus and put them on the course website. The communication between the students and the instructor can occur either synchronously or asynchronously. Both instructors and students will not need to be linked to time; the students don't need to be required to meet with teachers in a specific time. They can learn anytime of the day or night. The instructor can communicate with the students using email, and on-line chat at any convenient time. Also, students and instructors need not be in place bound, access may be where there Contact the Internet [13].

Web-based education can help people who don't have the energy or natural ability to go to a university campus to study and the use of technology make it easier for instructors to organize and refreshes the course information.

When we learn or teach, there are four basic skills explicit [13]: listeners, speaking, reading and writing. Internet provides the students a lot of material to practice the skills such as:

- There are materials in the form of text suitable for reading practices.
- It is also available for students to listen to the news from a country's language on the Internet since the goal of a large amount of audio files accessible.
- If the instructor creates some interactive video material based on the language skill of students and puts them on the web, the learning will be more interesting more than ever as the learning becomes more efficient.

- If the students use a specific program, such as chat tool (instant messaging), this method requires a mastery of the skill of writing by students, as well as this tool provides a voice conference to discuss a group of students through the microphones, which supports the verbal text (speaking practice).

The biggest difference between the web and textbook is that the students can find only the knowledge written in the book, but from the web, students can see information they are really interested in.

For learners and teachers, a source of great web where they could find a different type information around the world. Also, students can use the web as a source of information delivery as well. Hence writing for the web allows students to write for a specific audience, rather than a teacher or other students.

As we know, all people need to contact with others [25]. If students find that they can communicate with others on the Internet, this satisfaction will be a great incentive for them to learn how to use the Internet. Thus, the learning process would be very nice and interesting. A combination of class-contact teaching and web material will be an ideal combination for Interactive learning [13].

Web is a tool that can improve and enhance classroom teaching [25], but instructors should be certain what they will do with the web before they incorporate the web into their teaching. Instructors must contact others who use the web in their education to gain experience, learn from others. In the meantime, instructors should consult technical experts to discover any offer technical assistance and support in education. Instructors must remember "the instructor can't just drop technology into a class on short notice. That is ,it takes preparation and experience to be sure that technology enhances rather than distorts the process of the course" [13].

2.4 The general concept of e-Learning education

E-learning is an approach that complements traditional learning methods and gives a more effective experience to the learner because e-learning uses technology to support the learning process. Fundamentally, it is about putting the learner first by placing resources at the learner's fingertips [24].

E-learning is much more than online courses. It is about outreach. It gives access to online information, networks with whom individuals can share learning problem solving and online tutors and mentors who can offer advice and guidance. The learning process should always drive the use of technology [24].

2.5 E-Learning strategy in engineering education

Learning is not just a teaching or studying, based on brain activity; it is also a social interaction with people, learning from each other. Learning is actually something we do every hour of the day. People like not only to exchange information and ideas, but also prefer to support and to help each other as well; this could achieved electronically, which could be as a part of an e-learning strategy for establishing a fundament to support learning communities or networks of selected groups of learners in electronic selected topics.

E-learning brings together industry-leading tools that support learning. In every subject area, there is a set of community tools that allow users to come together in groups to share information. There are also discussion groups led by moderators who guide discussions and answer specific questions. By using basic network meeting tools, such as web cams, online chat, students are able to carry out learning activities in the workplace or from any position, while also obtaining instant remote supervision and feedback from tutors.

2.6 Advantages and disadvantages of e-Learning in engineering education

Some learners do not respond well to conventional methods of education, there was a lack of choice about when, where and what to learn. Many learners say: they felt stupid at school if they had to ask for the same thing to be explained more than once. Others found lessons went too quickly or too slowly and their interest and concentration is damping.

In contrast, supported e-learning has the potential to provide a very different and flexible learning environment that can adapt to individual needs. Online students are in charge of their own learning, responsible for the time-table and the school bell. They can choose what to learn, when to learn and how long to learn [24].

E-learning suitable for learners because it provides them with immediate comments, there is no need to wait for students in a class or waiting for books to obtain any comments, so the course can be taken at their own pace, repeating or spending more time on areas of difficulty.

2.6.1 Advantages of e-Learning in engineering education:

In the realm of e-education, there are clear benefits that can be derived from e-learning as follows [24]:

- It supports and facilitates team-orientated collaborations and expands the ease of access to engineering education across geographical and cultural boundaries, among others.
- Class notes and materials are posted on the Internet and students can access the sites from anywhere in the world, where a student is given course materials and reads on his/her own.
- E-learning is interactive. That is, the software permits the student to communicate, not only with the lecturer, but also with classmates. It enriches and supplements the classroom experience by engaging the web.
- E-learning has the ability to communicate consistently to learners by providing the same concepts and information – unlike classroom learning, where different instructors may not follow the same curriculum or teach different things within the curriculum.
- E-learning is cost effective in terms of learners per instructor. In addition, it saves classroom time and this is very significant for learners who are employed on a fulltime basis.
- Students, instructors and evaluators can track learning outcomes more easily.

2.6.2 Disadvantages of e-Learning in engineering education:

Online learning does have its own invisibility problems, where online learning can seem far away and isolated. For the e-learner, as with all distance learning participants, e-learning relies on self-motivation. With no enforced discipline or deadlines, it is easy for the learner to be distracted and put off work for a distant tomorrow. With no human presence, it is also impossible for a learner with a problem to obtain help easily. Thus, personal interaction between the instructor and students is either absent or else very different from traditional face-to face learning [24].

Other subtle disadvantages of e-learning include the ability to read text from the computer screen. Moreover, linear text is often difficult for people to read from the computer**.** Hence, learners often have to reformat the text and print it out for reading, requiring the need for a printer.

The problems of invisibility, and isolation can be managed in several ways, the key solution is communication to learners. The first is a help where learners can talk to someone to give them advice on their course. Second, many e-learners are also assigned an online tutor who checks on their advancement and can be contacted by e-mail with any queries or worries. Third, Online chat rooms provide learners with the means to contact and talk to other people doing the same course, exchanging advice, ideas and encouragement.

2.7 The state-of-the-art e-laboratories

Engineering courses normally include the laboratory components, which are essential to the learning process. Where using technology and the Internet could be for real laboratories or simulated laboratories to enhance the learning/laboratory concept and theory, remote laboratory is suitable for engineering courses offered within distance learning and real laboratory environment. It differs from simulation approach which employs programming code to simulate the result of the laboratory experiment such as Pspice, Electronic Work Bench and Labview [26].

Distance learning has taken on many forms and meanings. Early techniques for distance learning involved printed materials, including tutorials, assignments, and exams, passed between student and educators through the mail in order to complete course. Over time, this experience was enhanced through the use of other media including radios, television, audio and video tapes, and the telephone. Current practices are incorporating the internet to bring multimedia interaction to students everywhere, anytime, in real time [27].

The Internet has become a widespread tool for teaching and learning because of the facts that it enables more flexible delivery (anytime), distance education (anyplace), new visualization possibilities (interactivity), and cost reduction. The Internet becomes increasingly important as a learning environment, and internet-based technology is rapidly being adopted in engineering education as a tool for enhancing the educational experience enabling access to remote laboratories and performing real experiments.

The recent trends and advancements in technology shows that there is a need of user friendly environment to increase the efficiency of the learn process. The basic problem faced by the engineering student during the laboratory classes is that they have to perform the experiments in the laboratory in a group of three or four students. Such as normally gives negative support to some of the students and hence create a lack of interest. In order to rectify such problems students try to copy the results of their batch mates which degrades the student's performance. Such problems can be rectified if the student is allowed to perform the experiments throughout the day by either being physically present in the laboratory or by performing the experiments via Internet [28]. Educators create these online laboratories to help students acquire hands-on laboratory experience without requiring physical access to a building with specific experimental equipment [29].

2.8 Categories of electronic laboratories (E-laboratories)

Remote and virtual laboratories are two effective techniques for the use of the internet in engineering education, previously introduced as computer-assisted instruction and computer-assisted experimenting. Specifically, the software involved in creating such online laboratories either allows a user to interact with an experimental setup located in another geographical location (i.e. a remote laboratory) or uses numerical simulation tools to emulate the behavior of experimental system (i.e. a virtual laboratory) [29].

2.8.1 Remote laboratories (R-laboratories):

R-laboratories (sometimes called "web-based control"), offer remote access to real laboratory equipment and instruments in real time [26], [30]. It is the experiment/laboratory which is conducted and controlled remotely through the Internet and the experiments use the real components or instrumentation at a different location from where it is controlled or conducted, R-laboratories are the realness of systems learners work on [31].

2.8.2 Virtual laboratories (V-laboratories):

V-laboratories are based on simulations of real systems [31]. Where a simulation commonly replaces the real system, virtual laboratories typically resort to simulation software such as Matlab [32] or LabView [33] or specific applications. These simulations can be run directly on a client host (learner computer) such as with Easy Java Simulation [34], but we could envisage server-side simulations when specific software or calculus power is required and not easily available on client side. The main drawback is that it is not a real system, but it is a relatively realistic model of a real system.

We can find remote (or virtual) laboratory experiments in various scientific and technical topics such as automatic control ([35], [36], [37]), electronics, chemicals and mechanicals ([17], [38], [39]), and in robotics [40].

2.9 Advantages of remote laboratories

The design and use of a remote laboratory in a faculty of engineering has the following clear advantages [19], [29]:

- Better performance for the laboratory equipment since they are available to students during 24 hours a day. .
- Organization of laboratories: It is not necessary to keep the laboratories open at all times.
- Remote laboratories promote collaborative work.

- Organization of work for the student: with remote laboratories both students and lecturers can better organize their own time, including class time schedules.
- Autonomous learning: remote laboratories promote autonomous work.
- Distance courses: remote laboratories permit the organization of engineering courses without the need to have the students present, avoiding many of the current problems, as we have mentioned previously.
- Integration of handicapped students: since all the hardware equipment is controlled by a computer, they may be used by handicapped students with software / hardware techniques specially designed for their particular needs.

Their main drawbacks are based on a loss of observability (no direct system view: only through a webcam) and command ability (keyboard or mouse tools to drive remote system).

2.10 Different remote laboratory technologies

The use of technology in engineering education enables learners to develop life-long learning skills, problem solving skills, learning with technology skills, and technology skills (using tools and environments) [19], [41]. Generally, a remote laboratory is composed of different and clearly defined parts, for example:

- A server that will have the laboratory equipment connected. This computer will be in charge of programming the devices and controlling it.
- A website (i.e. a set of web pages on a web server) offering Internet access to the people interested in using the laboratory remotely.
- A set of client computers connecting to the remote laboratory server through the remote laboratory site.
- A webcam attached to the remote laboratory server, providing images though the remote laboratory site in order to show the evolution of the practical case.
- A custom made card that will establish the communication between the server and the laboratory equipment.

2.11 Styles of e-laboratory courses differ from most of the existing courses by a number of innovative features:

E-laboratory courses complement traditional courses and practice teachers to innovate their courses [42]; these are:

- Introducing learners to the methodology through simple, yet practical, examples to stimulate their interest in engineering before exposing them to rigorous theory and advanced mathematics.
- Giving learners a better feel for the topic by problem through on-line simulation, graphical visualization, and interactive virtual experiments.
- Allowing different target groups to select individual paths through the course tailor-made to their actual needs and respecting their background.
- Allowing both for self-study and remote tutoring with investigative and collaborative modes of learning.
- Integrating computers into the course curriculum consistently and giving learners a hands-on opportunity to acquire the necessary skills.

The need to increase a collaborative learning in engineering education by means of delivering experimental data to people in different parts of the world has become a priority among students. Methods such as email do provide such a facility but they lack the ability to provide this data in a real time environment. Therefore, it is necessary to probe another approaches such as chat tools to provide data in a real time [28].

Chapter 3

Collaborative Working Environments

3.1 Introduction

The term “collaborate” means “to work”. Literally, to collaborate means to “co-work” or work together. Light et al. [43] define collaboration more precisely as: “Collaboration is the process of two or more people working together toward a common purpose or goal, where the participants are committed and interdependent and work in a common context, physically co-located and using shared resources”.

Collaboration is a process by which individuals and/or groups work together on a practical endeavor. Collaborative working is a fundamental feature of organizations and is increasingly being supported by technology [43].

As people often work with others and with the increasing importance of computers in our work and everyday lives, it is natural to expect computers to play an important role in facilitating collaborative working [44].

Electronic collaboration (e-collaboration) is the computer mediated process of two or more (dislocated) people working together on a common purpose or goal, where the participants are , on the one hand, committed and interdependent and, on the other, work in a common context using shared resources, which supported by (web-based) electronic tools [43], [45]. From the previously mentioned, we can set the goals of any Collaborative Working as:

- The group members each bring knowledge to the group.
- Share that knowledge between them.
- Each person contributing ideas and desired work to be combined into a final result.

In engineering education, concepts taught through lectures are often complemented by laboratory experimentation. Students can observe phenomena that are often difficult to explain by written material. Furthermore, interactive experimentation on real world improves the motivation of the students and also develops an engineering approach to solve realistic problems. This contribution presents a collaborative working environment for a remote laboratory. Students have access to the remote laboratory via Internet from anywhere at any time. The collaborative environment allows the experimentation in a team, where the group is able to interact and to discuss the results of their work.

Before we begin to establish and development of the collaborative working of the e-laboratory, it is necessary to study and analyze the technical systems in various social fields, and benefit the study of these systems, in terms of means of collaboration such as coordination and communication between the collaborators, etc.; however, collaboration is successfully in many fields as illustrated in the next section.

3.2 Applications of collaborative working

1. An airport control tower [1]:
   - The function of the control tower at an airport is to provide information and instructions, in the form of clearances, to aircraft in certain areas of the airport and the airspace immediately around it in order to ensure the safe and efficient movement of traffic.
   - Fig.(3.1) below shows the section of the tower control room, where most of the control work takes place as:
     - The tower controller is responsible for controlling aircraft on the runway.
     - The ground controller is responsible for certain other areas of the airport.
     - The planner (or coordinator) typically maintains contact, by telephone, with other air traffic control facilities that manage the airspace around the airport.

Communication between different parts of the airport as illustrated in Fig. (3.2), where a controller looks at the radar screen, which shows traffic in the region of airspace surrounding the airport.

Coordination is also clear from the collection of flight strips indicating information about departing flights are relayed by a closed-circuit television link to the offices that providing services in the airport as shown in Fig. (3.3).

In other words, it is clear that the activity of the airport is a highly distributed activity.

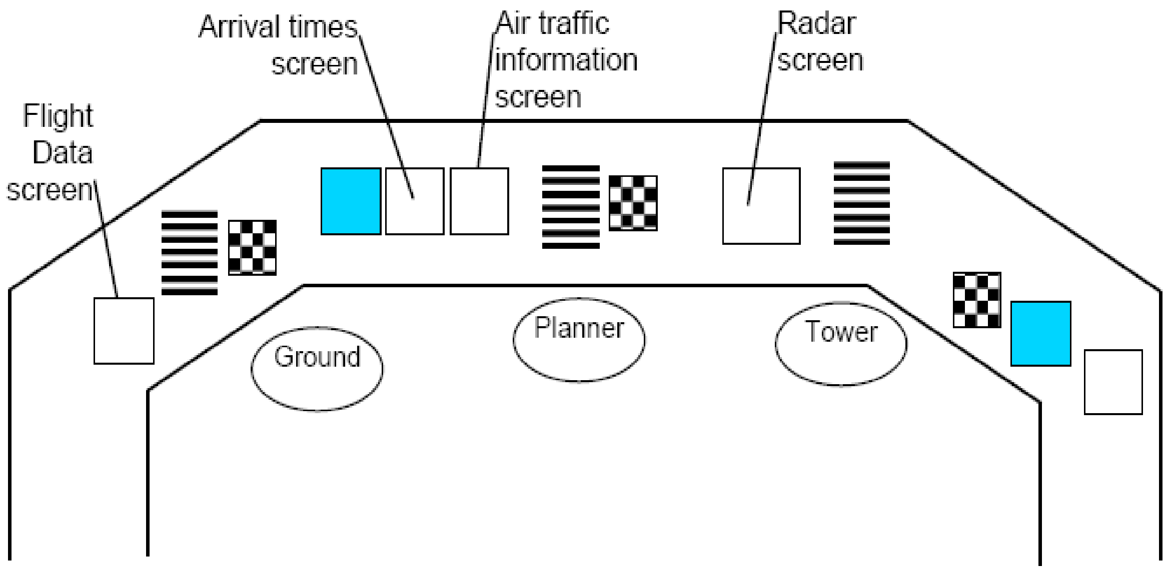


Figure 3.1: Layout of the tower [1].

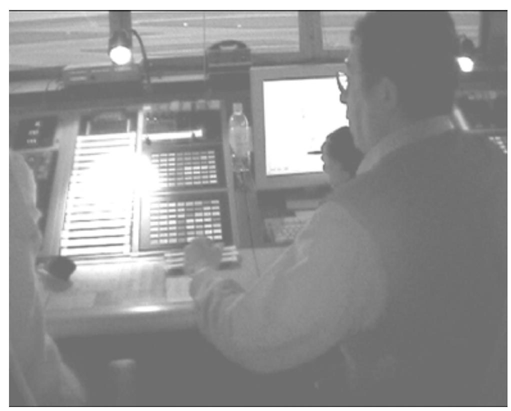


Figure 3.2: Telephone calls and radio conversations [1].

2. Team military systems [2], [46]

Advanced technologies are used to facilitate military team work collaboration through communication and information exchange for enabling remote operations. Collaborative technologies currently used in remote military operations (show Fig.(3.4)), such as email, and desktop conferencing, assist explicit communications between distributed team members around the world as part of physically distributed teams.

Figure 3.3: Flight strips about departing flights are relayed [1].

3. Chat tools [3], [4]

Millions of people meet online to chat, to find like-minded people, to debate topical issues, to play games, to give or ask for information, to find support, to shop, or just to coupling with others. They go to chat rooms, discussion groups to participate by sending short messages in the form of text, as commonly used in the world.

The system provides a chat facility students can use to communicate in real time with each other. The chat of every course provides students with a generic channel on which they can talk together. Beside the conversation space, a list of the users currently on the channel is shown.

It is to note, that chat tools are an integral part of many collaborative environments as they mostly sample instant messenger and allow chatting with all participants as illustrated in Fig. (3.5).

4. Share desktop and share applications [5]

Share desktop displays what you see on your desktop to all other participants and can both give control to others and take back control. Whereas share applications in addition to their ability to allow participants to see your application on their screen, they make it possible for a participant to control someone's application such as PowerPoint and Word. That is, other participants can make their own edits and updates. Principles and techniques suggested in share desktop and share applications have been successfully applied in web conferencing (Fig. (3.6)).

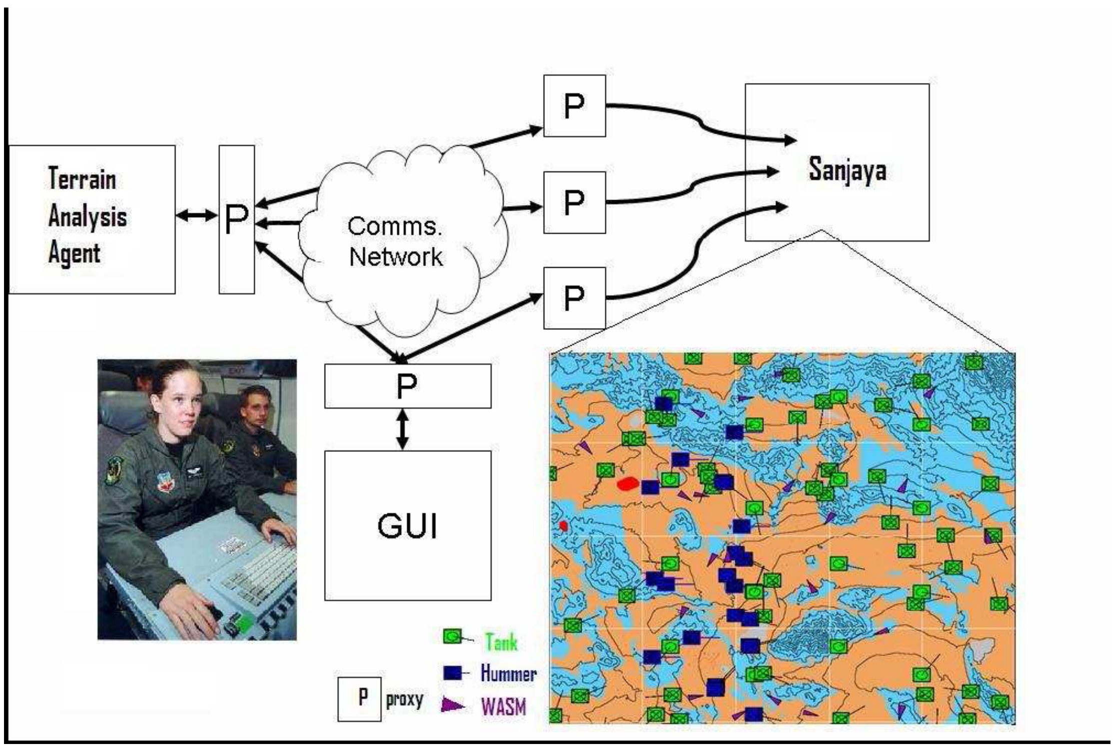


Figure 3.4: Team military systems [2].

Through the previous discussion, we can find the following shared features for any system supporting collaborative working:

- Communication (between partners).
- Coordination (roles and task attribution).
- Production (the task itself).

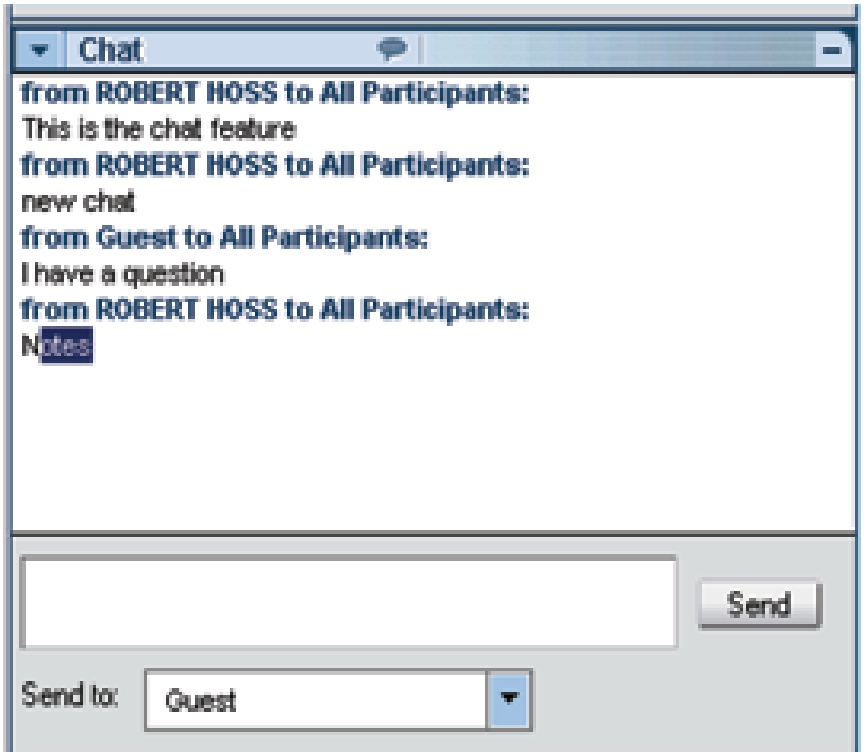


Figure 3.5: Chat tool [4].

3.3 Collaborative learning

Collaboration between learners can have a positive impact in a learning session only if learners can exchange efficiently. Discussions, advices given by a co-learner are good means to help a learner in knowledge understanding.
Web-based collaborative environments are a special category of e-learning tools that support a group of learners in achieving a common learning goal.

In local laboratory experiment, students usually work together in groups of two or more. This learning paradigm is often called collaborative learning. Collaborative learning develops skills for solving problems in a team. The premise of collaborative learning is based upon consensus building through collaboration by group members. Members of the learning group will usually organize their activities themselves and decide upon the roles of the different members via consultation and negotiation. With the rapid expansion and availability of communication and information technologies, collaborative learning can also be done effectively in a remote environment at different places. Collaborative working environments bring together users, which are geographically distributed, but connected via a network [47].

Students in the computer lab were able to see each other and talk to each other. Each student uses a computer adjacent to his or her group members. Usually, face-to-face students first rearranged their chairs so they formed a circle with members of their group in order to utilize their time in discussing a specific problem. Thus, the process was similar to communicating with a group of individuals in a chat group in the same room such as the computer lab. In many aspects, the face-to-face chat samples the computer-based chat. For example, students go back to their computers and answer questions, working with their group via the specific system software. The students are able to talk with the members of their group, but enter all their answers into the computer [48].

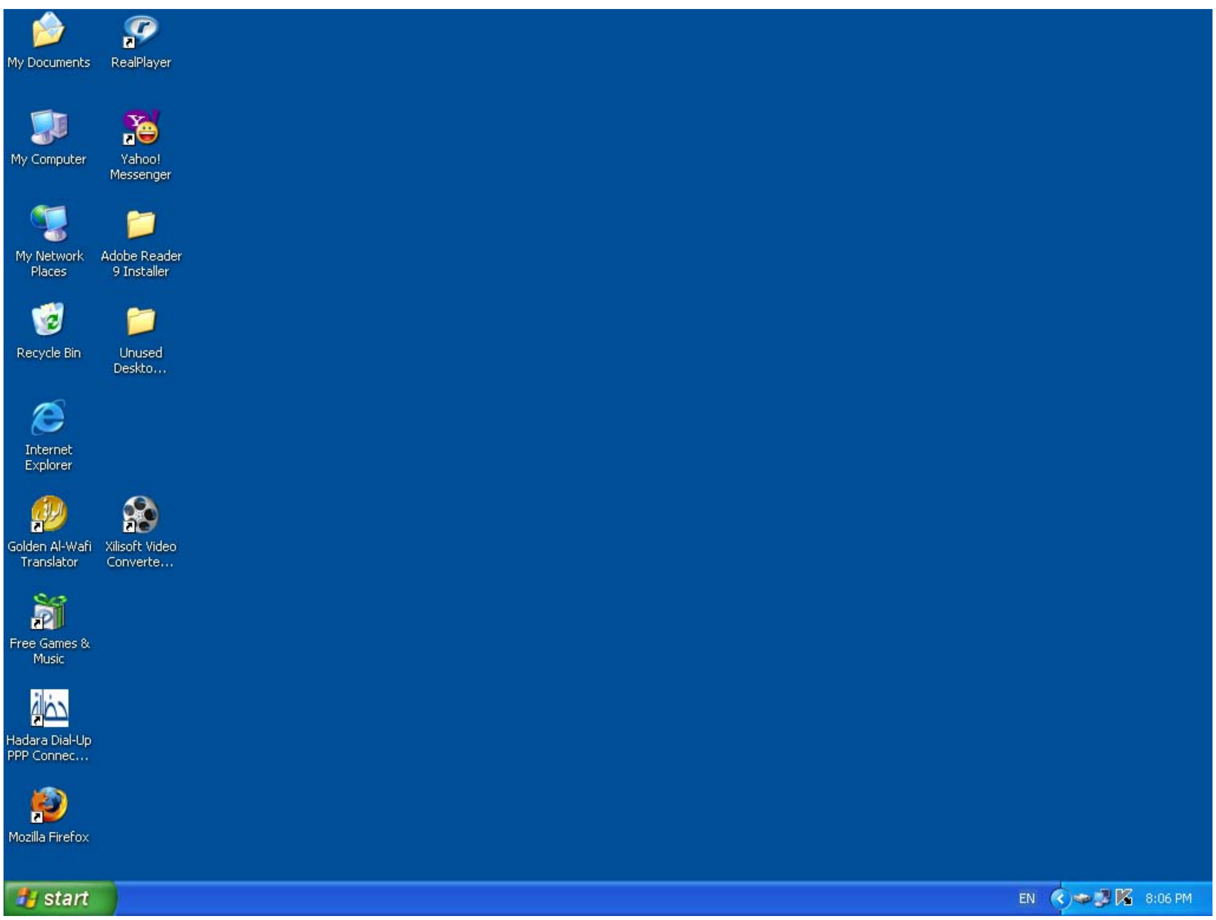


Figure 3.6: Examples of share desktop and share application [5]

3.3.1 Effects of collaborative learning:

Various theories have been devised to explain the collaborative learning effect. Among those we consider most useful are [49]:

1. The group environment provides more "air time" for students. That is, it provides more opportunities for students to ask questions and, thus, acquire new information.
2. During collaborative learning, students make public estimates about their knowledge. The feedback from other students helps group members interchange their ideas, particularly, if the discussion involves some degree of disagreement.
3. The social necessity to communicate their ideas requires students to articulate and elaborate their knowledge to others. The acts of articulation and elaboration encourage the active use of the content and, thereby, increase learning.
4. Students in collaborative groups exhibit helping to each other, tutoring, sharing notes, etc. leading to increase learning.
5. Collaborative learning leads to increased speed of learning by increasing motivation and attention.

3.3.2 Social and psychological aspects of collaborative learning:

Different social and psychological aspects deal with collaboration in learning processes [50], [51]:

a) Cognitive Learning: Learning produces sustainable results when external information or the requirements of a task can be embedded in already existing cognitive structure. In other words, it must serve as confirmation, modification or contradiction of the learner's existing knowledge. It is to note that cognitive science is an interdisciplinary science that draws on many fields (as psychology, artificial intelligence, linguistics, and philosophy) in developing theories about human perception, thinking, and learning.

b) Motivation: The learning process will be better accepted and will lead to sustainable knowledge when learning can be experienced as the result of one's own activity, not as a mere adaptation to the knowledge of other people.

c) Social construction: The construction of the three social elements: understanding, knowledge acquisition and production are mainly based on collaborative knowledge-sharing interaction with others.

Two basic aspects of collaboration might be important [52]: the first one involves the relationship among students; in other words, students work together as peers, applying their combined knowledge to the solution of a problem. The dialogue that results from this combined effort provides students with the opportunity to test and refine their understanding in an ongoing process. The second aspect of collaboration involves the role of the teacher as a moderator during the learning process by supporting students and providing direction when students are having difficulties.

Apart from the psychological learning aspects, collaborative learning is important in engineering education for the following reasons [52]:

a. Students acquire various skills, such as the ability to work in teams and to achieve objectives in collaboration with others.

b. Students learn to communicate with each other using technical expressions that are specific of their professional engineering domain.

c. Students learn to integrate the know-how of others in order to accomplish a given work task.

d. Students acquire remote collaboration skills, when the teamwork is carried out from several locations.

3.3.3 Collaborative Courses:

Collaborative courses belong to distance learning. They are a mix of static content, which can be in any form such as textbooks, videocassettes and web pages as well as synchronous and/or asynchronous instructor interaction through a communications medium including, for example, e-mail, videoconference, chat tools, etc. [53].

The web makes it possible to integrate synchronous and asynchronous technologies so that students can benefit from both. These combinations of technologies and the web site provide a richer basis for collaborative course. However, technologies choices must ensure that all students will be able to use these technologies, and that software is straightforward, and pleasant to use [25].

3.3.3.1 Asynchronous courses:

The majority of distance learning courses make use of only asynchronous technologies. These technologies are usually the most stable, affordable, and accessible, and they have the added benefit of being the most available to learners in a wide variety of time zones environments.
In courses that are delivered asynchronously, learners can each make use of the communications medium at a time and place most convenient to them; because of this, these courses are available to others who have difficulty scheduling attendance in traditional classroom environments. In addition, asynchronous classes can easily be made available to a collaborative learning of learners who can share information and experiences with one another without regard to geographic boundaries or time limitations. In the simplest configurations, asynchronous courses can make use of extremely communications media such as e-mail [53].

However, some instructors and learners are not suitable to collaborate in asynchronous interactions. For one thing, asynchronous communications are largely text-based, which can limit the effectiveness of discussions delivered through this medium. Students who are not native speakers of the language used for course delivery face not only special difficulties, but cultural and language barriers can more easily lead to miscommunication and misunderstandings in text-based communications [53].

Discussions in a text-based medium can be difficult, and collaboration in such a discussion can take a lot of time for the learners. The delays in response in asynchronous classes often causes students to become distracted with other responsibilities rather than remaining focused on class participation, so the time required for the learner to respond to individual learner questions and problems is much greater than the time typically spent by teachers in traditional classroom settings [53], [6].

3.3.3.2 Synchronous courses:

Synchronous courses that employ two-way video or audio technologies can allow instructors and learners to collaborate demonstration and observation of practices for a variety of skills-based learning activities in a way that it would not be possible in other forms of distance classes because of the fact that teachers and learners can engage in nearly all traditional learning activities through voice, text, and video.
In these instances, the online classroom can begin to seem more like a traditional classroom in many respects, and teachers can begin to employ many of the same approaches used to teach face-to-face classes, as well as being able to respond to visual and verbal forms from students [53], [6].

However, access to high-quality video and audio is still beyond the reach of most online course developers due to cost or bandwidth issues, and most synchronous online courses are therefore still delivered through text or text and audio only [53].

3.4 Learner and instructor interaction

In traditional classroom environments, the immediate two-way interaction that takes place between learners and instructors allows for a very flexibility type of teaching. A classroom instructor can change the learning content and interaction immediately based on feedback from the class. For example, by initiating a new activity, starting an open discussion, or finding alternate ways of explaining or demonstrating a difficult concept to a learner who is having trouble understanding.

In traditional classroom settings, teaching can take many forms, can occur within or outside of a designated learning environment, and can be prescriptive or reactive, planned or constructed. The instructor can adapt materials to meet individual student needs, and flaws and deficiencies in both course design and the instructional materials used to support teaching can usually be compensated for during the face-to-face teacher-learner interaction in the classroom. The traditional classroom environment also supports a wide variety of instructional methods including, for example, discussions, brainstorming, collaborative learning, etc. that create a highly interactive and rich learning environment. In contrast, the use of electronic and other distance learning technologies requires a carefully planned approach to instructional development, as communication between learners and instructors. In most cases, the instructor and learners are each working in isolation, communicating through any form of media [53].

Many of the instructional methods used in a traditional classroom environment are implemented with both verbal and nonverbal forms of communication that are difficult to replicate in the types of media commonly used for distance learning. The four types of communications and interactions that may occur in distance learning courses include [53]:

- Learner-content interactions: Used to obtain intellectual information being in the form of text, images, simulations, etc. from the instructional media.

- Learner-instructor interactions: Two-way dialog to motivate learning, clarify concepts, and provide feedback.

- Learner-learner interactions: Provides an exchange of information, ideas and dialog between students, with or without instructor involvement.

- Learner-interface: Used to access and participate in instruction and communicate with instructors and other learners. The type of interface (whether it is a book design, interactive computer program, or other interface) is dependent on the technologies and media used to deliver learning. Good learner interface interaction allows the learner to concentrate on the other three types of learning interactions. If poorly designed, the learner-interface interaction distracts the learner from accessing content and communicating with others.

3.4.1 Learning with instructional support:

Complex problem solving tasks without instructional support will often demand too much from the students and will lead to ineffective learning. The ability to engage in two-way communication with an instructor and other learners allows distance learners in facilitated courses to solve in more complex learning problems.
As long as students need instructions and help in solving scientific problems, the learning environment shall provide knowledge for solving these problems. Instructional support is an important element especially in problem-based learning settings; therefore, remote laboratories should provide support for students.

Table 3.1: Time-space matrix as a traditional way to decompose collaborative systems

Time / Place	Same Time	Different Times
Same Place	**Face to Face**	**Asynchronous Interaction**
Different Places	**Synchronous distributed**	**Asynchronous distributed**

In remote laboratories, a tele-tutor communicates via synchronous or asynchronous communication tools with his students, resulting as a central role regarding instructional support.

Table (3.1) classifies collaborative environments using a time-space matrix, as illustrated in Fig. (3.7). in the following, we are going to discuss some advantages and disadvantages of the different approaches. One form of communication between students and tutors is asynchronous or synchronous as discussed in next section.

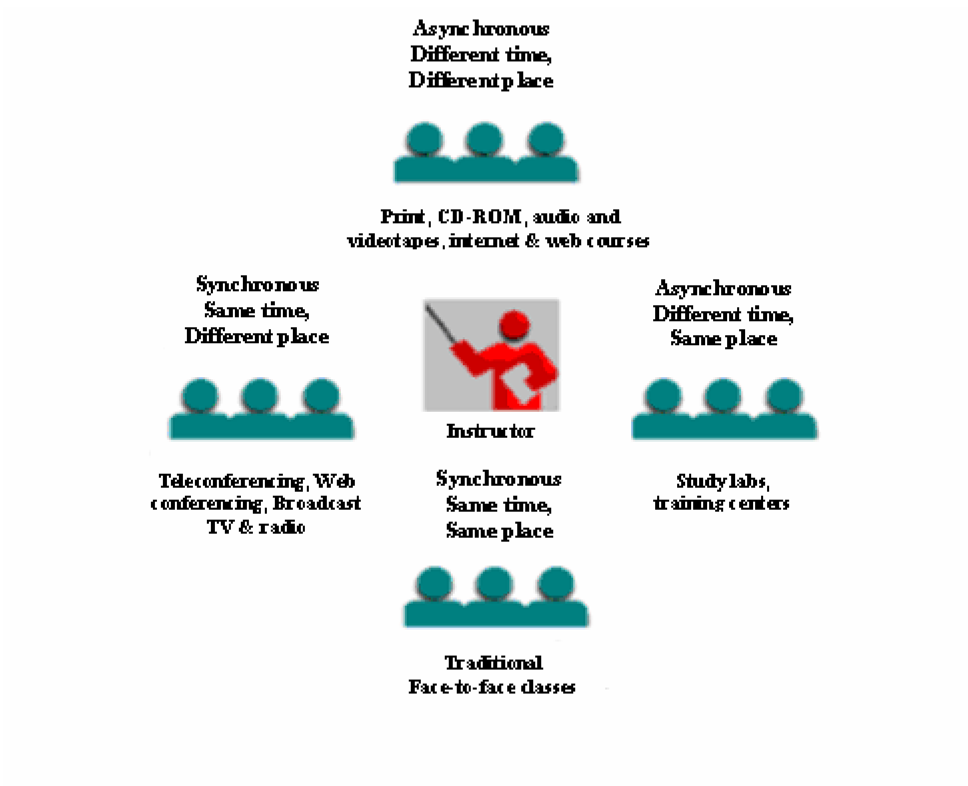


Figure 3.7: Time-Space Matrix as a traditional way to decompose collaborative systems [53]

3.4.1.1 Communication services:

Distance teams are distributed in time and space using various communication media carrying voice, video and text data. There are two common kinds of communication modes:

- **Synchronous:**

  It restricts the time flexibility of the students, upcoming problems can then be solved immediately by using chat tool [54], application sharing such as LogMeIn [55] and video-conferencing tools [43], [54], [56] .

- **Asynchronous:**

  This mode of communication is most suitable in situations where students are widely dispersed in time zones. Traditionally, e-mail and web pages have been the most widely used systems for asynchronous communications. Students will not get any support from the tutor during the laboratory session; therefore, students might not able to find answers and solutions for their questions and problems in the experiment. So students have to accept time delays for getting answers to their questions, and therefore, if time delays are too long, they could lose their motivations [43], [54], [56].

Chapter 4

System Architecture, Design and Implementation of the Collaborative Working E-learning Environment

4.1 Introduction

Remote laboratories can be considered as a well established teaching structure and learning environment for developing of skills required for the efficient collaborative learning environment to achieve collaboration and communication between students [57].
The concept of the learning services and their deployment through technologies are excellent means to integrate real laboratories into collaborative e-learning environments for engineering education [58].

This chapter describes in-depth a suggested architecture based on .NET technologies to support the software development process of web-based collaborative working environment, in addition to an interactive Graphical User Interface (GUI) environment. This GUI facilitates the remote control and access of various instruments and experiment setups.

Our proposals have emphasized on the collaboration aspects between two students or more. Collaboration can be achieved when students work together. It will be shown how we have implemented an e-collaborative environment for remote experimentation supplemented by a rule-based e-tutor, based on Microsoft Visual Basic 2005.NET framework [59] that gives support in order to facilitate the implementation of the Web-based collaborative working.

A suggested architecture can be built around an existing laboratory components and equipments in a general university/ engineering college without much difficulty. It is required to design an experiment set-up and to write software control code for interfacing it with both a lab server and the internet. The web interface browser was written using the programming language “vb.net”. It was necessary to program controls to interface our experiment with the lab server and the internet. The laboratory experiment set-up is attached to a server, which provides the web interface to the remote clients. Multiple clients are able to get connected through remote login into the system.

4.2 System structure for collaborative working

System that was designed, developed and implemented for the purpose of remote access to laboratory equipments and to enable experimental collaborative work for on-line support of e-learning is called e-collaborative environment for remote experimentation. It is prototype system that is intended to support experiments in the area of electronics, but its structure poses no limitations on other types of experiments also, such as in physics, mechanical engineering, or some similar fields of engineering or science.

The original system was developed for at presence (local) learning, where each student would be provided with the circuit board, along with a set of laboratory experimentation notes.
The purpose of the work described in this thesis was to consider the ability to provide a remote collaborative learning facility via an internet link as an alternative approach for the course delivery. In this case, electronic circuit arrangement hardware was modified for connection to a remote collaborative laboratory.

In order to achieve this case, we have decided to propose a new system to achieve the collaborative work among students. To propose a new system, we have to study and analyze the old systems, where these systems include one or more technical systems, also include knowledge of how the system should be used to achieve some broader objective, as well as include people as inherent parts of the system where these systems called social technical systems. By contrast, an air traffic system, a team military system include thousands of hardware and software components plus human users (collaborators) who make decisions based on information from the computer system, include the utilization and benefit the study of these systems in terms of coordination and communication between the collaborators to make a decision.

In laboratory experiments of the faculty of engineering, solving tasks without instructional support will often demand too much from the students and will lead to ineffective learning [6]. As long as students need instructions and help in solving scientific problems, the learning environment shall provide knowledge for solving these problems. Instructional support is an important element especially in problem-based learning settings; therefore, remote laboratories should provide support for students. In remote laboratories, a tele-tutor communicates via synchronous or asynchronous communication tools with his students, resulting as a central role regarding instructional support.

As mentioned in the previous chapter, the main problem caused by the asynchronous approach is that if the student groups are not under enough control by their teachers, they will have a time delay for getting the answers to their questions. Once the delay becomes too long, the students could lose their motivation, and face many problems left unsolved. One way the discussed problems can be solved is enabling a remote human tutor to support students over a distance via synchronous communication tools such as chat tools that allows quicker assistance. However, a remote human tutor can not support students at any time and will not be available throughout the day.

Therefore, it is intended to implement an automated helping system in the form of a rule-based e-tutor for user support in complex remote experiment environments. A rule-based system embraces a rule-base including stored knowledge about the correct experiment configuration, (see section 4.6.4). In so doing, we have found the solution to the problem of supervision of the instructor to give feedback directly to students, as well as provision of dialog box text messages (developed by vb.net) to be feedback to the students who did not find any information feedback from the other students. In order to solve the problem inequality in task division between students by using color coding (see section 4.6.3)

4.3 Distributed system architecture for collaborative e-learning

Fig.(4.1) illustrates a suggested system architecture accommodated to the special needs of this research for establishing a collaborative working e-learning environment, which is not only supported by a human tutor, but by a rule-based e-tutor as well (see section , Rule-based e-tutor).

The architecture applied to this system follows the simplest client-server architecture, where an application is organized as a server and a set of clients [61]. This architecture enables the different users such as students and tutors to:

- Perform on real (physical) experiments remotely whenever they want and anywhere they are.
- Collaborate by two students or more and attribute the tasks between them.

- Development of a remote experiment and hardware facility based on an existing course structure and requirements to facilitate the collaborative work remote learning scenario.
- Synchronous interaction between students and tutor.
- Provide the students with an automated help at any time.

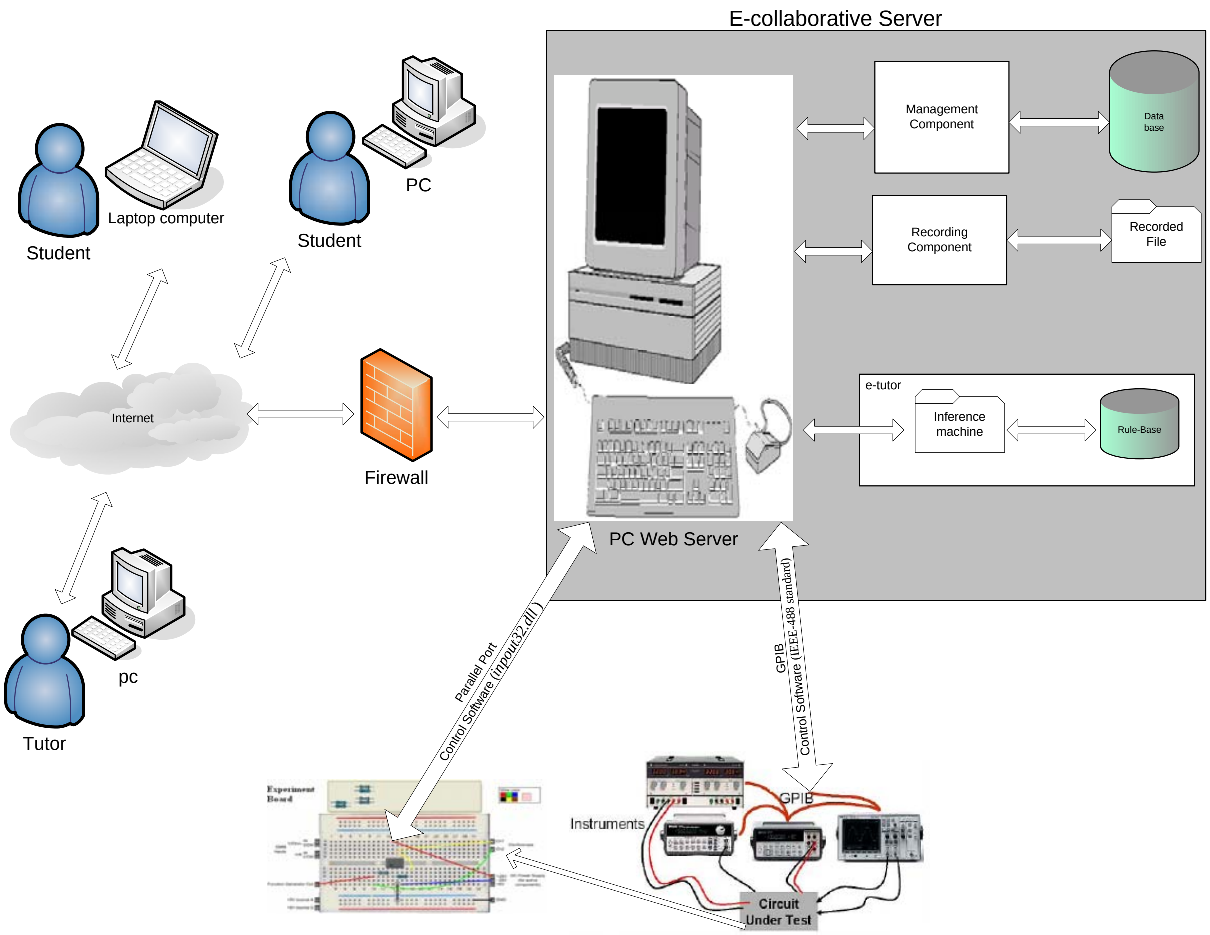


Figure 4.1: System architecture of the suggested e-collaborative environment for remote experimentation supplemented by a rule-based e-tutor

The overall architecture of the system is determined. Fig.(4.1) illustrates the system architecture of the e-collaborative environment for remote experimentation supplemented by a rule-based e-tutor suggested as a possible solution to achieve effective collaboration between students and a tutor in remote labs. This system is organized into subsystems and components. It also includes the end users of the system and the user-interfaces through which the users can interact with the system in order to complete tasks. In brief, overall decisions are made about actors /sub-systems interaction, data storage and implementation.

4.3.1 Hardware Architecture:

A general scheme of the system that has been designed using client-server architecture is shown in Fig.(4.1). The remote users (collaborators) can login to the e-collaborative server using TCP/IP link over the Internet and can select and perform an experiment.

At the server end, hardware setup of all the laboratory instruments and the experiments are attached to the server, instruments and devices are also connected with the programmable instruments through a PCI General Purpose Interface Bus (GPIB) card and GPIB cables. An experimental board containing different components and electronic elements are connected to the parallel port of the server to view real-time experimental setup, see Fig.(4.1).

4.3.2 Software Architecture:

The software design for the e-collaborative environment for remote experimentation system focuses on a client-server type software model whose primary functionality is to interface the remote users (collaborators) with the server and controlling of the actual laboratory experiments/instruments.

The GUI for the client side has been developed using Microsoft Visual Basic VB.NET. The underlying protocol for communication between server and client side is TCP/IP.

The server side controls the server process as well as to modify the configurations of the instruments. It handles all the tasks of communication to and from the instruments using the instrument controller (a PCI GPIB card). Standard Command for language library has been used in vb.net environment to send commands to and receive data from the instrument driver, which uses the GPIB IEEE 488.2 standard protocol to drive the instruments.

The clients' side control the experiment interface by sending command to the parallel port for selection of different points related to the experiment board.

The client side GUI allows the student to control various functions of the instruments associated with the experiment in real-time.

For each instrument as well as experiment at the server end, corresponding GUI has been designed at the client end using vb.net control toolbox. The client's command generator for instrument issues commands according to the parameter set specified by the student for a particular instrument and transmits them to the server. The experiment results checked by e-tutor then sent back by the server are then handled and displayed in the client GUI.

4.4 The components of the proposed architecture

The architecture proposes that our distributed application should be made of the following components:

4.4.1 Web server:

The web server provides several services for the clients through a web browser. In addition to its role as a middleware to communicate with the clients and the other system components via the internet, it represents the central unit of the collaborative e-learning and functions as a coordinator between the various components.

4.4.2 Web-based user-interface:

All clients of the distributed e-collaborative system, students and tutors, use the same web-interface. In our case, the clients will be mediated by a conventional web browser such as the Microsoft Internet Explorer.

When the web-based user interfaces was designed, several human-computer interaction rules for user-interface design had to be taken into account such as consistency of data display (labeling and graphic conventions), minimal memory load on user, flexibility for user control of data display, presentation of information graphically where appropriate, standardized abbreviations, presentation of digital values only where knowledge of numerical value is necessary and useful, minimal surprise (users should never be surprised by the behavior of a system), user guidance (the interface should provide meaningful feedback when errors occur and provide user help facilities), user familiarity (the interface should use terms and concepts drawn from the experience of the people who will make most use of the system) [63].

4.4.3 Management and schedule component:

The schedule data of accessible time is stored in a schedule data base. When a student group tries to access to the experiment, the management and the schedule component examines whether they are allowed to do this according to a schedule time-table. Moreover, this component manages the registration procedures of the students for enabling them to execute the experiment. The entered data will be temporarily stored and after its verification by the experiment administrator, it will be stored in the database permanently. The system informs the students about the failure or the success of their registration attempts by user name, password or e-mail confirmations.

4.4.4 Rule-based e-tutor:

Main goal of this research is helping students to execute experiments remotely at anytime and from anywhere. This goal can be fulfilled when the synchronous human tutor is continually available. Realistically, a remote human tutor will not be online all day. Therefore, we want to implement an automated help system (e-tutor) for user support in complex remote experiment environments. Hayes-Roth [64] notes that rule-based systems automate problem-solving know-how and providing a means for capturing human expertise.

Rule-based systems share certain key properties: incorporating practical human knowledge in conditional if-then rules, increasing their skill at a rate proportional to the enlargement of their knowledge bases, being able to solve a wide range of possibly complex problems by selecting relevant rules and then combining the results in appropriate ways, determining the best sequence of rules to execute adaptively, and explaining their conclusions by retracing their actual lines of reasoning and translating the logic of each rule employed into natural language.

In our research, the e-tutor is realized as a simplified rule-based system because our focus is mainly on collaborative systems and not on knowledge-based systems. Knowledge-based systems, which represent an important class of intelligent systems, can be especially useful for solving complex problems in cases where purely algorithmic or mathematical solutions are either unknown or demonstrably inefficient. They employ human knowledge captured in a computer to solve problems that ordinarily require human expertise. Moreover, they stored knowledge about the field of interest (in natural language-like formalism), i.e. knowledge-base systems are: facts and rules. Rules are probably the most popular choice for building knowledge-based systems [65], [66].

We will realize a simplified rule-based system whose rules consist of two parts: conditions and action where the knowledge contains the domain knowledge useful for problem solving. In a rule-based system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure. When the condition part of a rule is satisfied, the rule is said to fire and the action part is executed and an inference machine. The inference machine is a mechanism, which plays the role of reasoning and searchs a strategy for a solution.

The purpose of the inference machine is to seek information and relationships from the knowledge and to provide answers, predictions, and suggestions in the way a human expert would. The inference machine must find the right facts, interpretations, and rules and then assemble them correctly; the database includes a set of facts used to match against the IF (condition) parts of rules stored in the knowledge [67].

The e-tutor embraces a rule-base including stored knowledge about the correct experiment configuration. Evaluating the rules periodically causes the system to react in accordance to how students deal with the experiment. The experiment designer in cooperation with the experiment tutor maintains the e-tutor rule-base with plausible rules so that they verify and validate the fired rules after every experiment session.

While carrying out an experiment, the recording component described in the next section, archives the students' interactions with the experiment along with the e-tutor reactions.

4.4.5 Recording component:

It is important to have a component that records the students' interactions with the user-interface.

The importance of this component includes:

- Storing the work of each student during the course of the experiment, and storage to be corrected by the teacher and whether the connection is true or not.
- If the instructor was not found on line and the e-tutor could not correct the connections of the students through the experiment, sometimes the human tutor is an assessment or correction after the students had finished their work, for this reason we have needed for recording of experiment at work.
- And, can also record the experiment as a video and store it in the format “Movi file”, which would be beneficial to students for reference when needed.

4.4.6 The experiment:

The remote experiment can be any one of an engineering laboratory covering topic related to electric circuits or electronics and so on. Fortunately, most of the current instrumentations such as oscilloscopes and multi-meters are provided with control through PCI GPIB (General Purpose Interface Bus) card and GPIB cable [68], [69] or through parallel port [70].

4.5 User-Interface design for remotely collaborating students

For designing the user-interface, it is to suggest the usage of simple screen sketches and key-screen prototyping [61]. The user-interface concept can be illustrated by simple screen sketches aimed at conveying the system concept to non-technical users. That is, key-screen prototypes show users the design of the proposed system and allow them to evaluate it; they can be used for usability testing.

Before discussing the prototype of the web-based user-interface for the clients shown in Fig.(4.2), it will be of great significance if we clarify some term definitions related to human factors that must be taken into account when we have designed the user-interface.

The prototype is the basis for a web-based user-interface for the clients: the students and the tutor. Ergonomically, we have to distinguish between aspects of perceptive and cognitive ergonomics [71]. The cognitive ergonomics relates to reasoning, memory and knowledge [72]. Here, we are more concerned with perceptive ergonomics focusing on designing issues such as color, shape form, dimension and allocation, highlighting and so on. From Fig.4.2, the desktop of the web-based user interface is divided into several windows for representing different functionalities. The windows are:

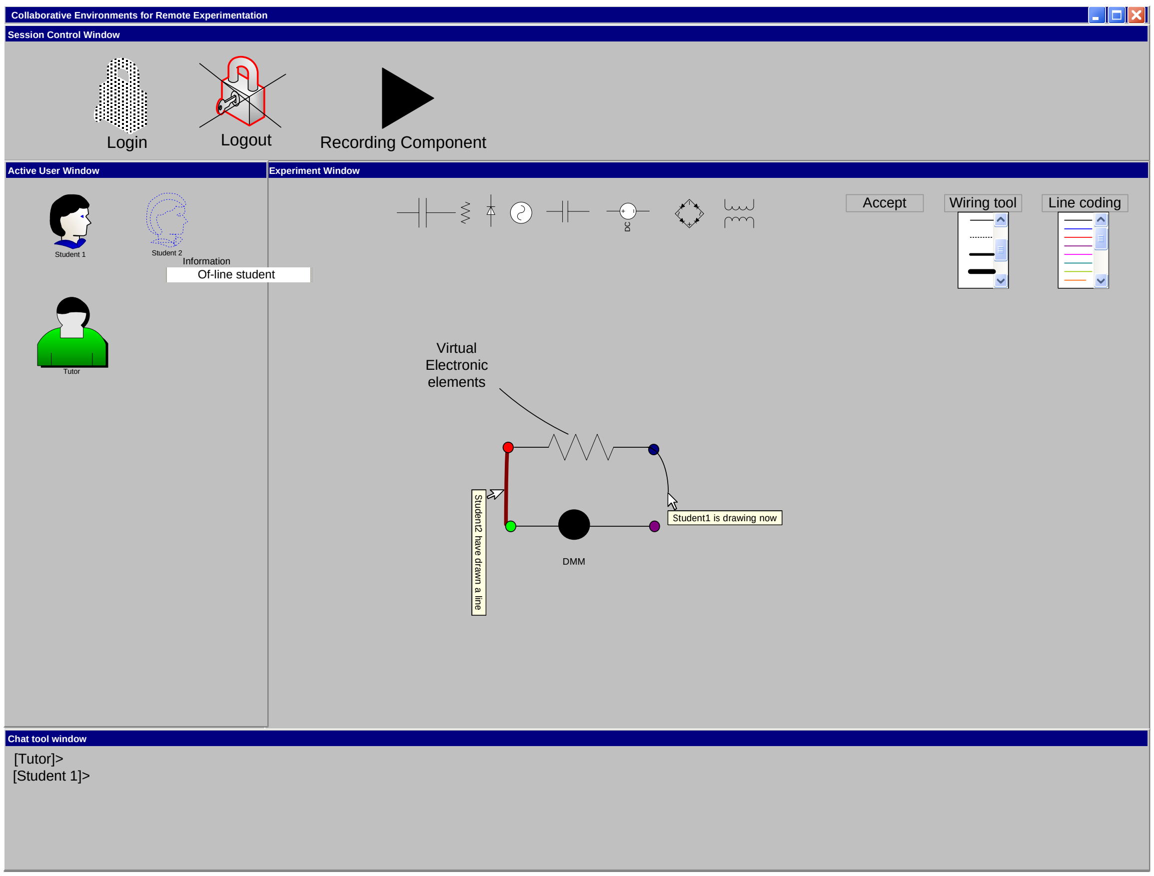


Figure 4.2: A key-screen prototype of the web-based user-interface for the clients

4.5.1 Experiment window:

The experiment window represents the remote tool kit and the instrumentation necessary for that experiment, which will be virtually visualized. By using the tools of the interaction configuration area, the student can select the suitable wiring tool to connect the electronic elements together. For a better visualization, she/he can adjust the wire color according to the function of the wire, i.e. black for ground, blue for positive VCC etc. The instrumentation area includes the equipments such as oscilloscopes to pursue the signals after building the circuit; as well as various function generators to feed the circuit with input signals. The students are able to connect the inputs and outputs of a virtual oscilloscope with other electronic elements. After every building step the students can select the accept button, the user-interface sends the circuit configuration to the server, where the e-tutor initially carries out a consistence check to the sent circuit data. If these data passes the check, this data are used to control the real experiment.

4.5.2 Chat tool window:

As previously discussed, chat tools have a central role in e-collaborative environments. Through the chat text window, a user can send a text message to a particular user or to all users at once.

4.5.3 Active user window:

An active user window shows the users who are logged into the experiment by means of an icon titled with the student's name. These icons are selectable as pop-up menus leading to usable user-interface. The menu item "highlighting" helps the users know what configurations on the experiment are done. The highlighting tool frames all electronic elements such as resistors, wires, ICs etc., with a dashed line whose color corresponds to the color code assigned to the student who has done the changes. The menu item "information" causes the system to display information about the student, how long the student is being logged-in etc.

4.5.4 Session control window:

This window consists of buttons with meaningful icons for controlling session concerns such as logging-in and logging-out, session recording or playing back etc. When the web-based user-interface has been developed, several human-computer interaction issues such as visual thinking and icons [63] must be taken into consideration. An icon is an image, picture, or symbol representing a concept.

The development of the system follows the prototyping approach because of its appropriateness in helping resolve requirements and design uncertainties as it is the situation in most of research projects [61], [62]. In the iterative process, the stages: specification, design, development, and testing are not chained, but rather interleaved and concurrent.

4.6 System development and implementation

The development of the system follows the prototyping approach that we have proposed previously. For developing the user interfaces and architecture of a suggested system, it is recommendable to use an interactive development system like Microsoft Visual Web Developer 2005 Express Edition in addition to these tools: Microsoft Visual Basic 2005 Express Edition [59], Web Server: IIS (Internet Information Service). The first two tools are based on the .NET technology, allowing quick creation through drawing and placing of graphical objects on the user display. .Net technology includes graphical user interface tools with rich libraries for user interface components, enabling data to be displayed in many forms.

4.6.1 Session control window for management component:

This window consists of buttons such as create a new user, logging-in and logging–out, session recording. The home page in our website implemented as “Login.aspx” allows students to log into their accounts and to enter a distance experiment window (see Fig. (4.3)). Through the website represented in Fig. (4.4), a new user can be registered. In this case, the student enters for the first time to a distant collaboration experiment to contribute in the work group.

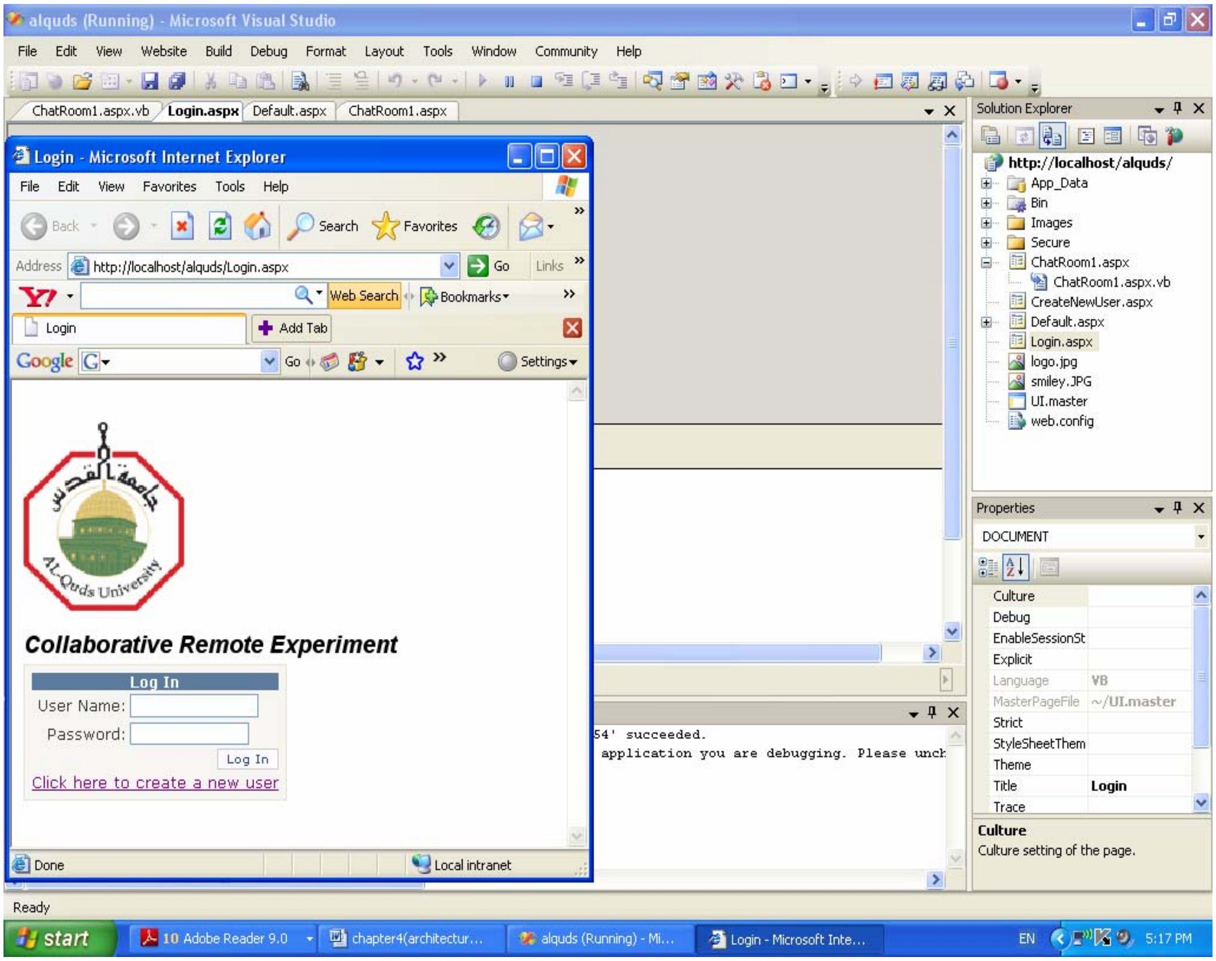


Figure 4.3: A Login page using aspx technology

Creating a new student demands the student to fill-in a registration form. After finishing this step, a user account is created. In Asp.net, database using SQL servers can be easily created with the help of this database, named here as ASPNETDB.MDF, in which user names, passwords and other account information are stored. On the form, the student must fill-in every field including two passwords and they must match what is stored in the student's account in the database. The output in Fig.(4.5 (a)) demonstrates successfully the creation of a user account. Fig. (4.5 (b)) illustrates the error message that appears when you attempt to create a second user account with the same user name. ASP.NET requires that each user name be unique. Then, after the user enters a user name and password, he must click the Log In button. ASP.NET determines whether the information provided match those of an account in the database; if they match, the student is authenticated (i.e., authenticating a student), and the browser redirects to the page specified by the property “DestinationPageUrl”. We have put this property to the chatwindow.aspx page. If the student’s identity cannot be confirmed (i.e., the student is not authenticated), the login page displays an error message. When the user clicks the Logging Out button, see Fig. (4.6), he will be redirect to the first page, (see Fig. (4.19) (part A)).

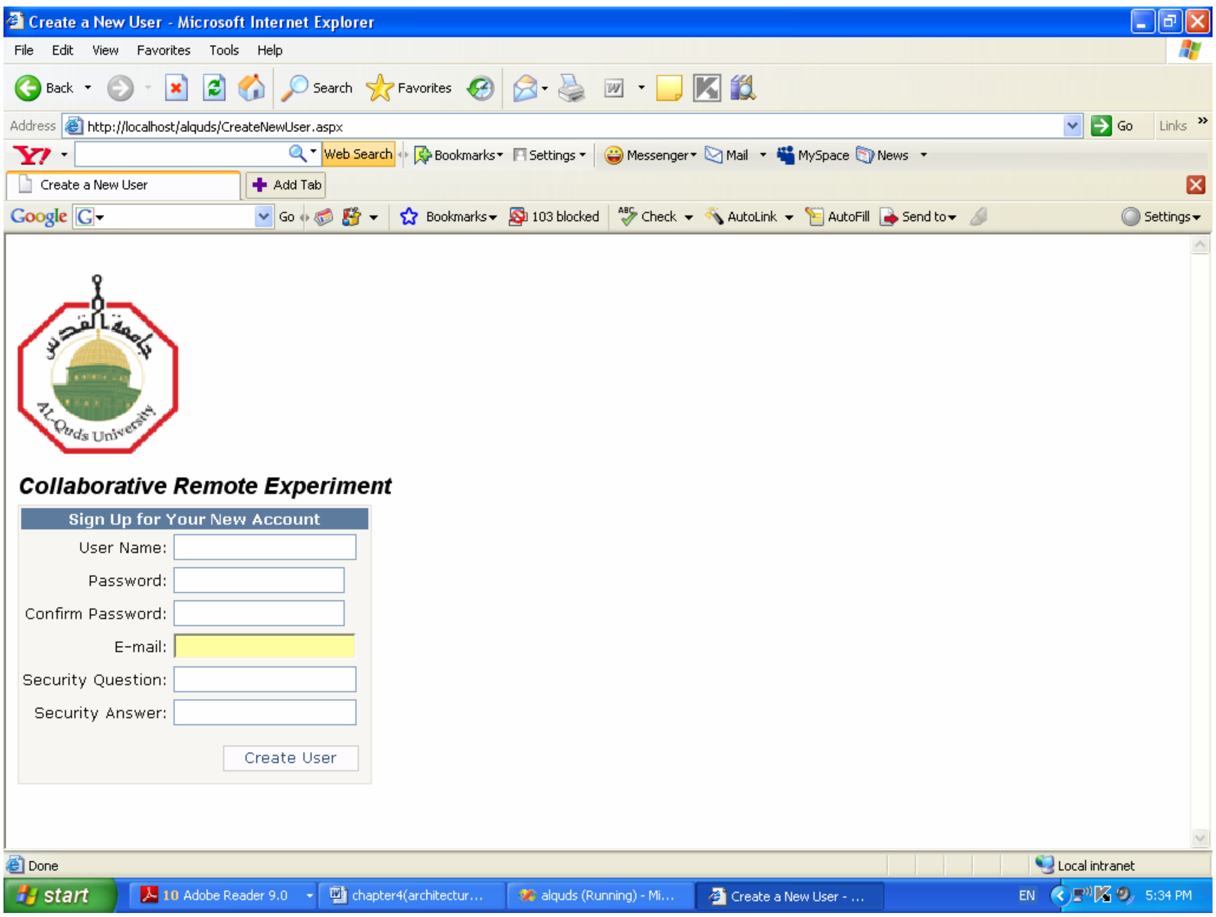


Figure 4.4: A web page called “CreateNew Student.aspx”, which provides a student with a registration form

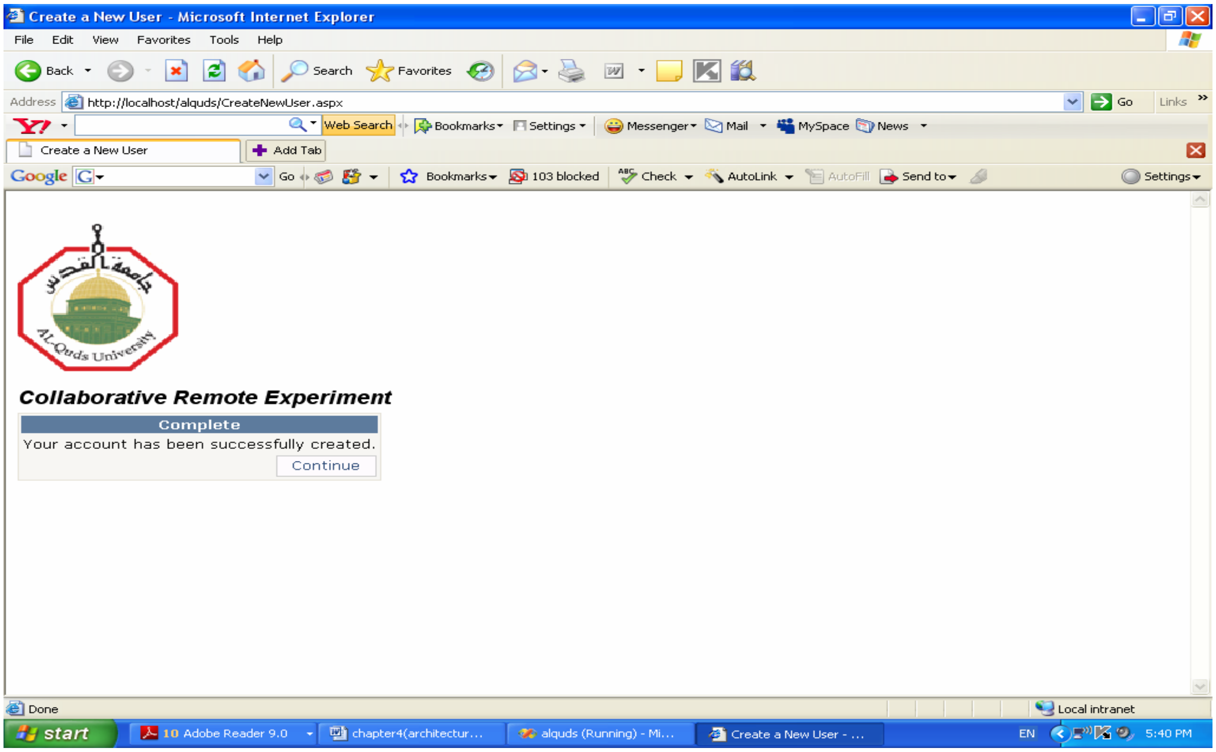


Figure 4.5 (a): A web page, which appears after successfully creating a new student account

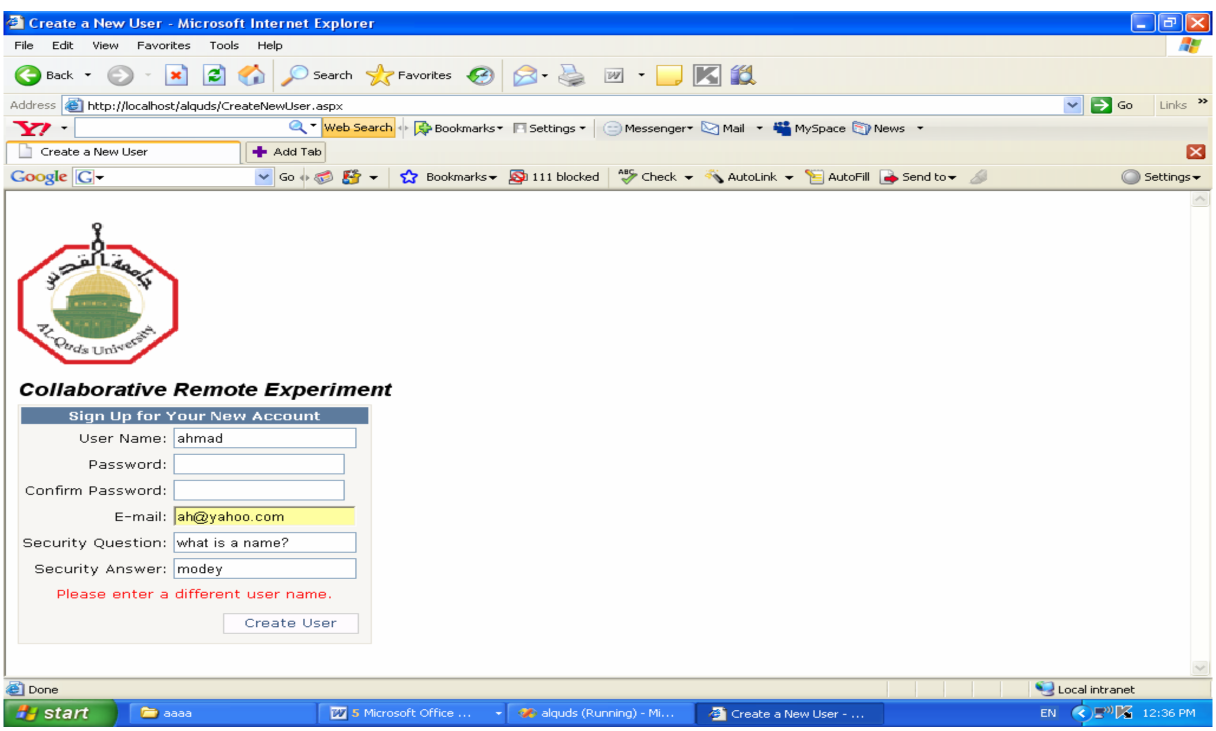


Figure 4.5 (b): Showing an error message when trying to create a new user that already exists

4.6.2 Chat tool:

Recently, more and more sites benefit from chat tools to help people socialize or sort out important issues. However, providing our system with a chat tool stimulates collaboration among students while executing an experiment. After starting the chat application in our implementation, a web page based on "aspx" technology will be started, which appears once a student clicks either the continue button (illustrated in Fig. (4.5 (a)) or a Log In button (see Fig. (4.3)).

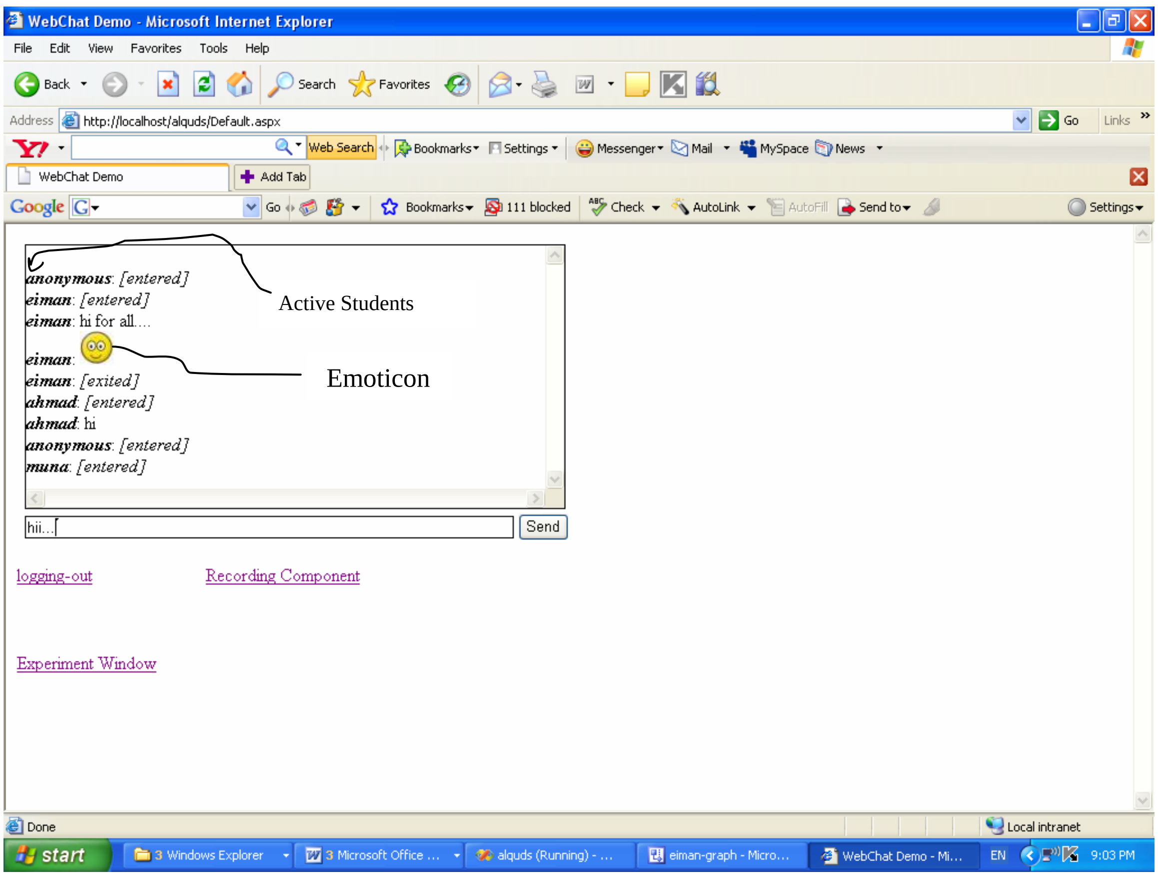


Figure 4.6: Chat tool window

After that, the student enters his/her name to be used in conversations between students. The code responsible for this part of functionality is listed in appendix a (p.79).

Then, the student clicks an "Enter chat" button, to enter a chat window as shown in Fig. (4.6). By means of the chat-text window, a student can send a text message to a particular student or to all students at once. As shown in Fig. (4.6), it is possible for students to exchange messages textually, supported with emoticons that are symbols or combination of symbols used to convey emotional content in written or message form. Some examples of text-based emoticons include: :-) :-(;-) (see Fig. (4.6)). (The complete code of this part of program is listed in appendix a (p.80)), (see Fig. (4.19), part B).

4.6.3 Circuit-wiring electronic:

The normal procedure for performing a single experiment includes the following steps [73]:

- Every student team wires the circuit specified in the instruction manual using a voltage source, the breadboard, and some of the components provided. At least one of the instruments must be connected to test points in the circuit in order to collect experimental data.
- The instructor checks each circuit formed to avoid possible damage. If the circuit is safe, the student team is allowed to continue by activating the source. Once the source is activated, they begin to deliver the required experiment.
- The students read the instruments. Then the students record an experiment as shown in the following section, where this will be carried out with or without the support of the instructor.

It is not possible for students to manipulate the components and wire a circuit with their fingers in the remote laboratory. A type of circuit-wiring, e.g., a switching point (see Fig.(4.11)) must be used. The complexity of such a switching point increases with the number of circuit points provided.

By means of the button “Experiment Window” shown in Fig. (4.6), the student can conduct a small experiment dealing with electronic circuit. The remote experiment is one of an engineering lab which covers the topic ohm’s low.

A snapshot during an experiment using the remote laboratory configuration is shown in Fig. (4.7).

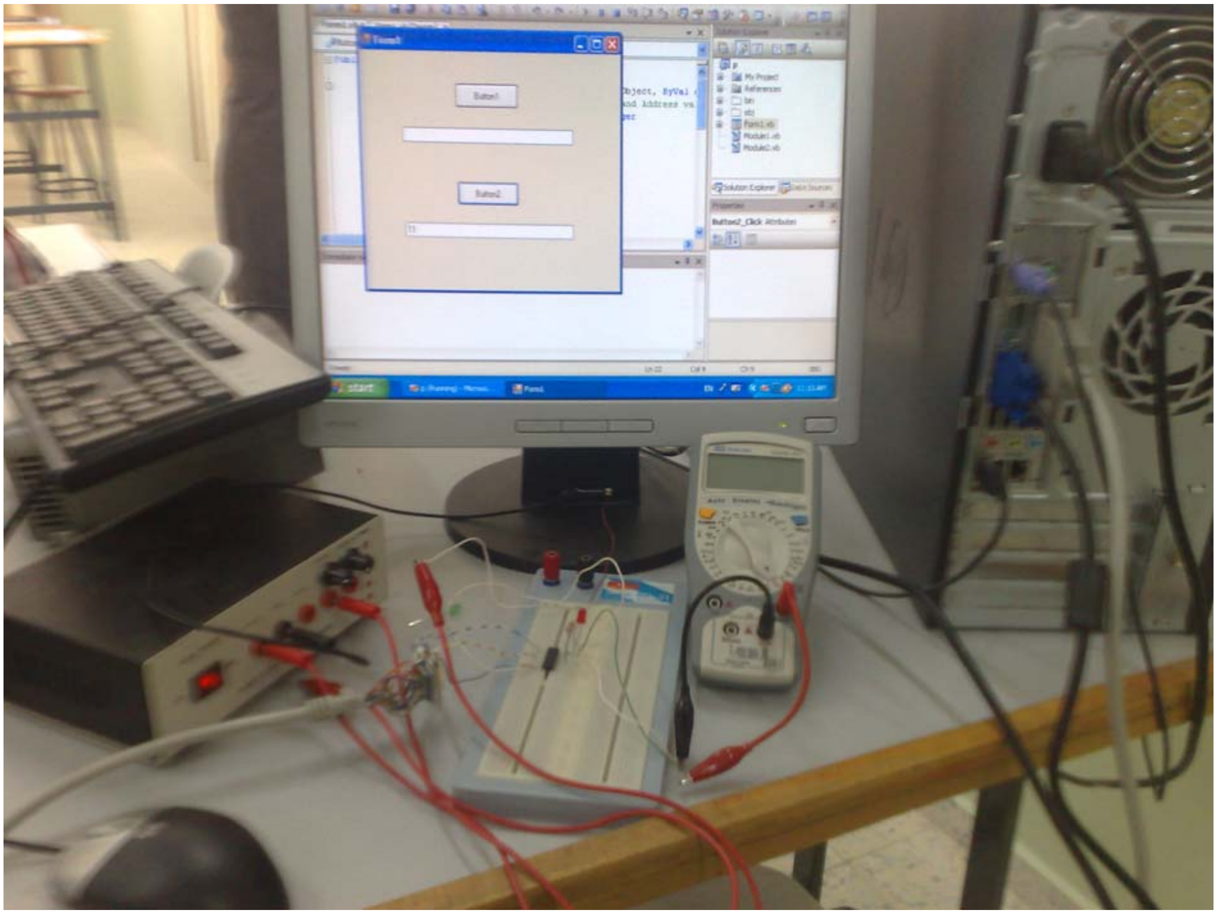


Figure 4.7: The remote lab configuration

For designing the user-interface of experiment window, the virtual representation of the experiment is shown in Fig. (4.8). For wiring the electronic elements together and connecting them with the instrumentations, a wiring and connecting tool realized as a pop-up menu can be used. The pop-up menu offers various types of wires and cables. The color and width properties are changeable (see Fig. (4.9)). From a pop-up menu, every student can choose its own color and width of a line. This distinguishes a student's work from others. This shows the collaborative working between students and the equal distribution of work between them.

On the client user-interface, virtual components as shown in Fig.(4.8) are used to be connected with other electronic elements and instrumentations to achieve a certain circuit configuration. Each element or device on the bread board is represented graphically, see Fig. (4.11), and their terminals are marked by highlighted circles as shown in Fig. (4.8). When a student has connected two points to complete a close circle, he/she connects terminals marked by highlighted circles as shown in Fig. (4.8). While a student draws a line between two terminals, he/she has to position the mouse cursor on the specified circles of the terminals. The code for this functionality was written in vb.net, which stores the corresponding x y coordinates of that point in a text file (see program listing in appendix a (p.81)), (see Fig.(4.19), part C). The equivalent pseudo code is listed in the following segment:

Initialize x-y coordinates for each line when MouseDown event occur

Initialize x-y coordinates for each line when MouseMove event occur

Initialize x-y coordinates for each line when MouseUp event occur

Draw lines // in vb.net we use a library, which supports several object graphics

Save coordinates of lines in a text file

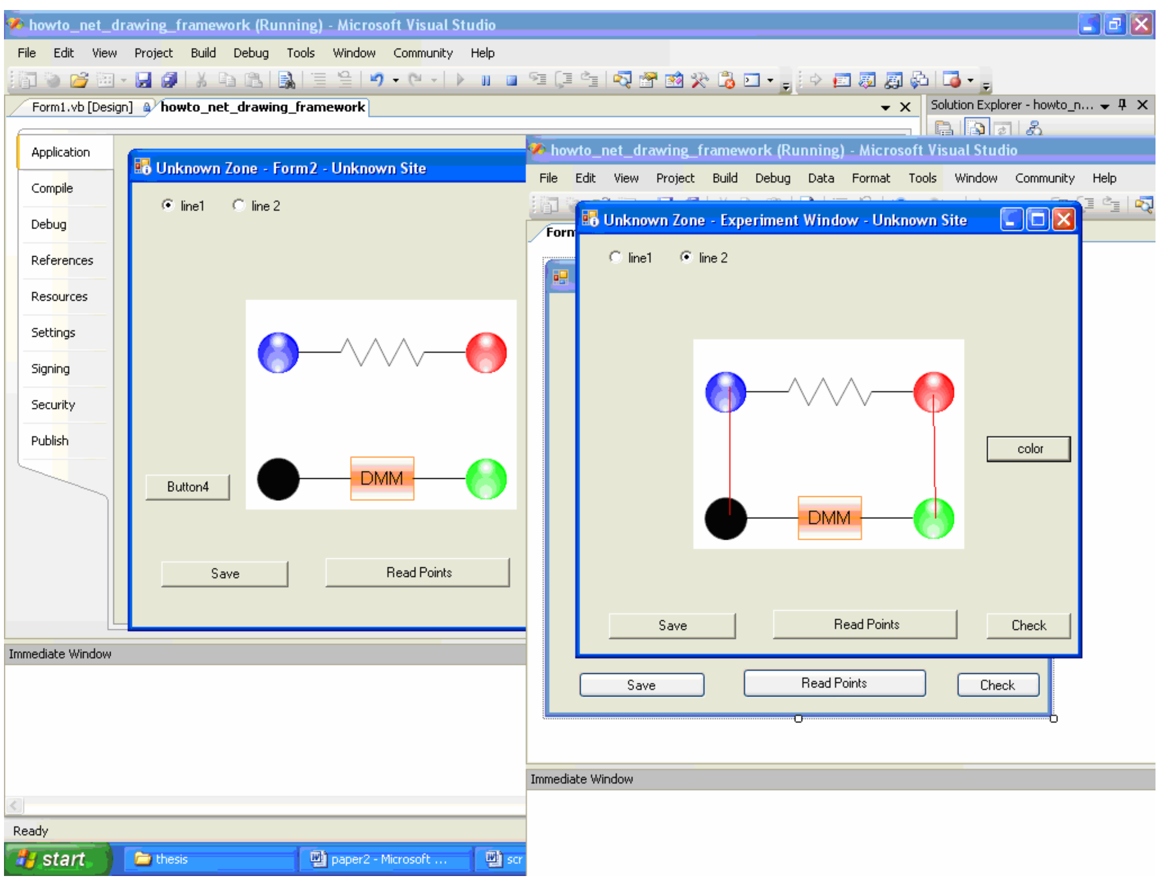


Figure 4.8: Virtual components on the user-interface (Experiment Window)

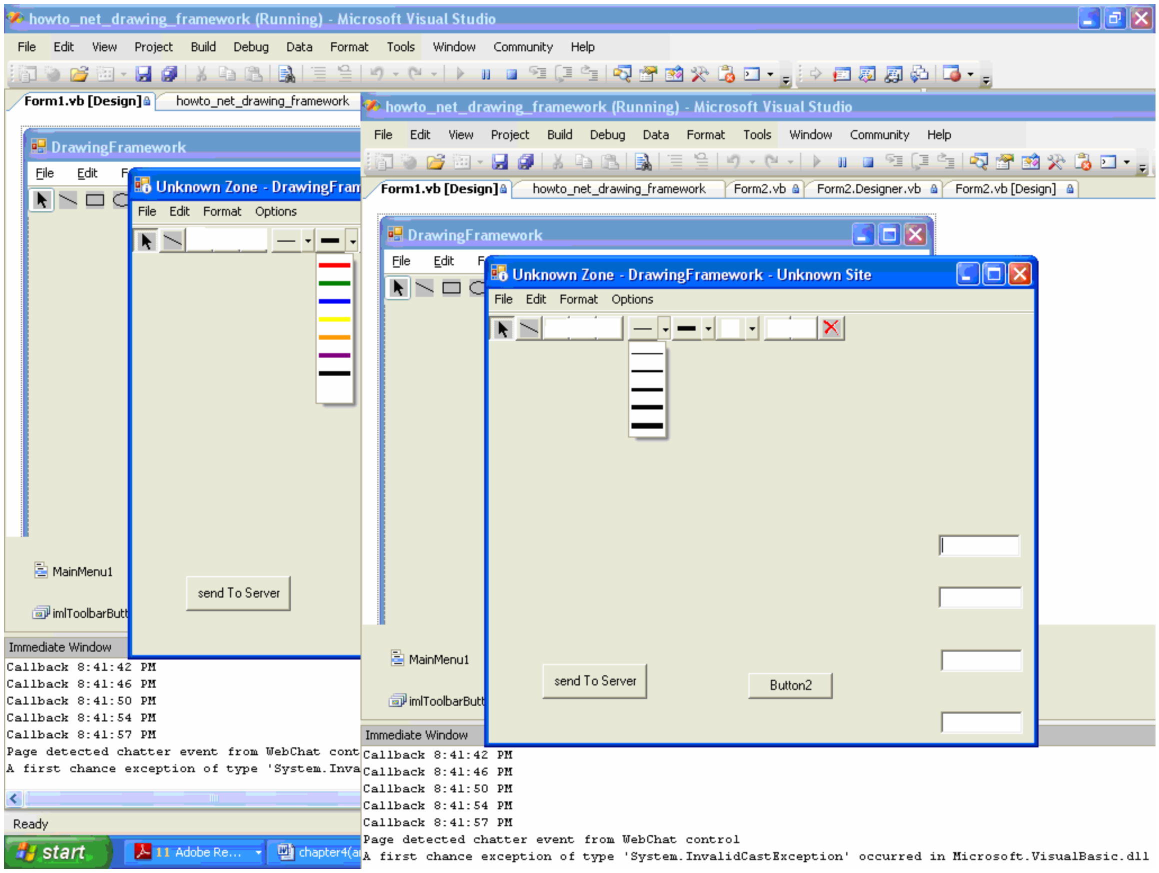


Figure 4.9: Color and width properties on the user-interface (Experiment Window)

Fig.(4.10) shows the collaborative work division between two active students working on a certain experiment, in which the value of resistance using the DMM will be measured. The collaborative work division is achieved through coding the different activities of the students on the virtual bread board with different line colors and thicknesses.

The process of coordination and consultation between two students is carried out through the chat tool. Simultaneously, each student can see the screen of the other student and that is what is known as share desktop (see section 4.6.6).

Once a student draws a line to connect two terminals of the circuit, the points of each straight line (x-y coordinate) will be stored and then sent to the e-tutor (see section 4.6.4)

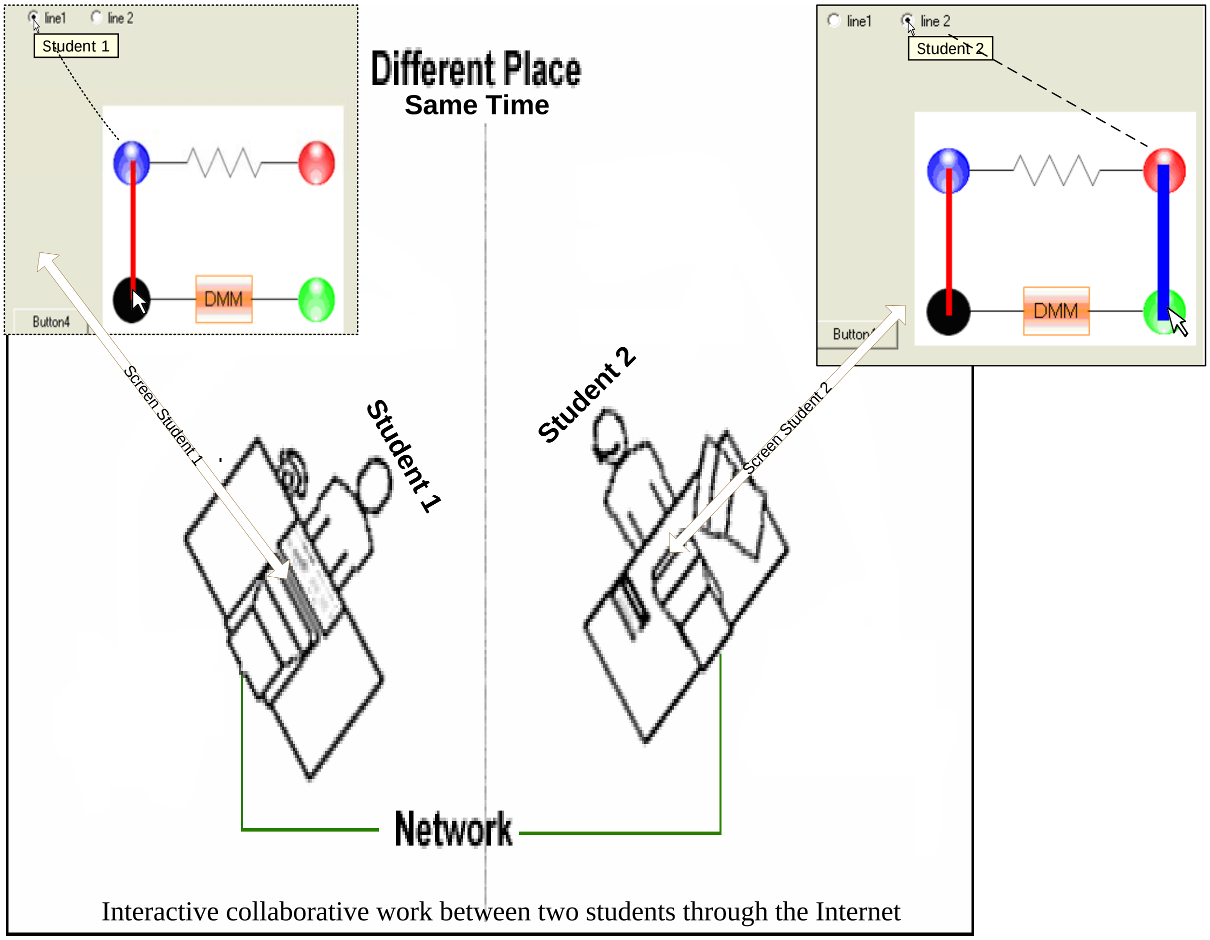


Figure 4.10: Interactive collaborative work between two students

To build an electronic circuit, we have used a breadboard unit allowing students to assemble electronic parts without needing to solder anything (see Fig. (4.11)). The breadboard shown in Fig.(4.11) is large enough to accommodate circuits with many components.

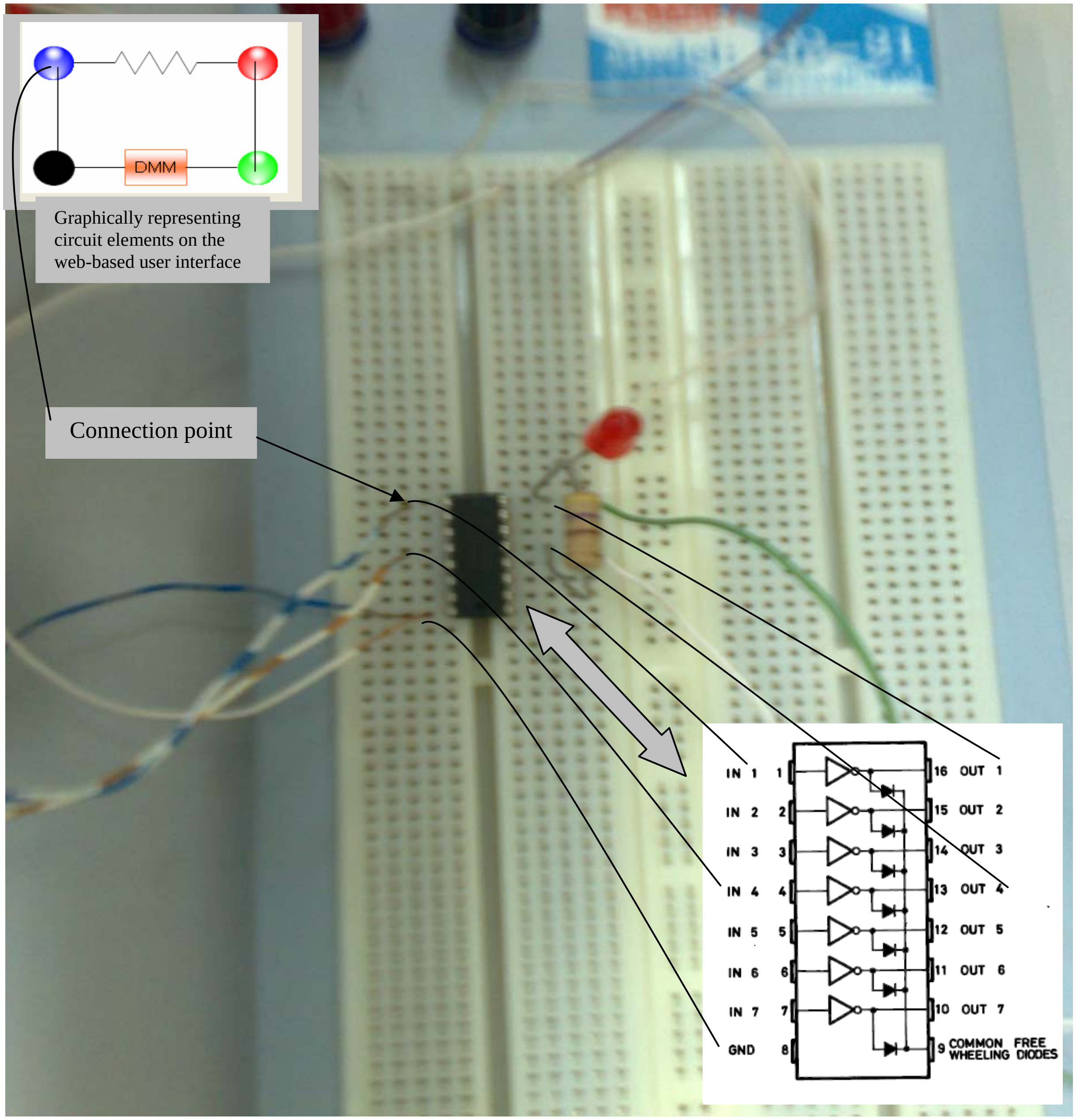


Figure 4.11: Pin connections of dip 16 ULN2003 on the breadboard

After having connected the circuit elements on the virtual user-interface, a text file containing the connections of the circuit elements with each other will be sent to the server. When the server receives the information, the e-tutor proves the validity of these connections (see section 4.6.4 Rule-based e-tutor); if the connections are correct, the server connects the equivalent switches. In other words, it translates the virtual connections carried out on the user-interface to real connections through closing the equivalent physical switches in the remote experimental kit as shown in Fig.(4.11). It is to note the parallel port is used as an interface between the lab server and the experiment (see Fig.(4.12)). The parallel port is one of the easiest interfaces available to students to communicate with hardware devices outside the PC [14].

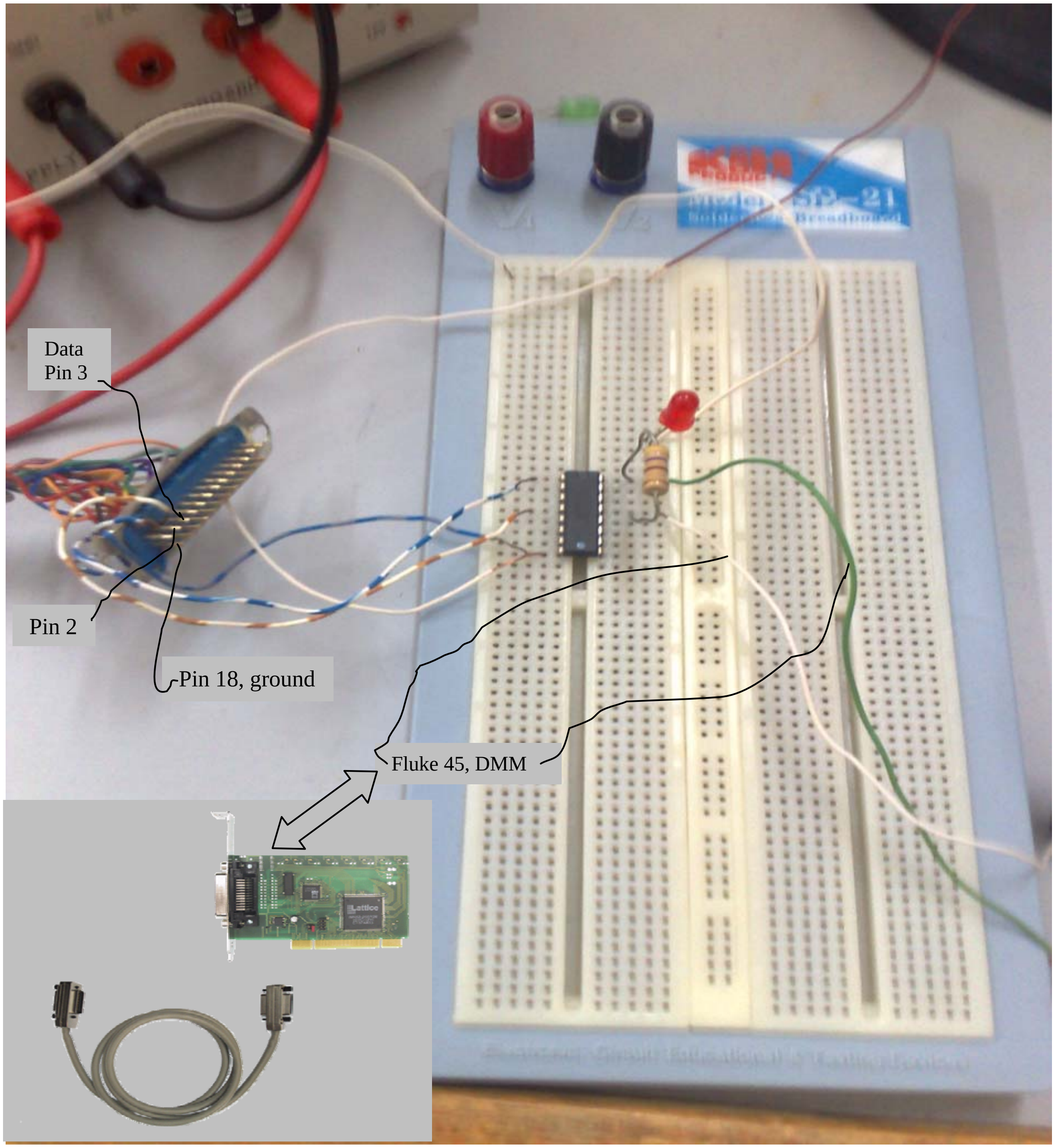


Figure 4.12: The circuit configuration of the experiment kit

It contains a set of signal lines, namely (8-bit parallel interface), so we have eight bits available there. Since each data bit can be set as either “0” (“turned off”) or “1” (“turned on”), we can directly turn on or off up to eight devices in order to send or receive data with other components.

We have used a third party tool in the form of a software library called “inpout32.dll” to connect to ports available in our PC. In the library “inpout32.dll”, all of the system drivers and small tools needed to activate and to run that system driver are included. Additionally, we have used the driver “UserPort.SYS”, which makes it possible to easily access hardware directly.

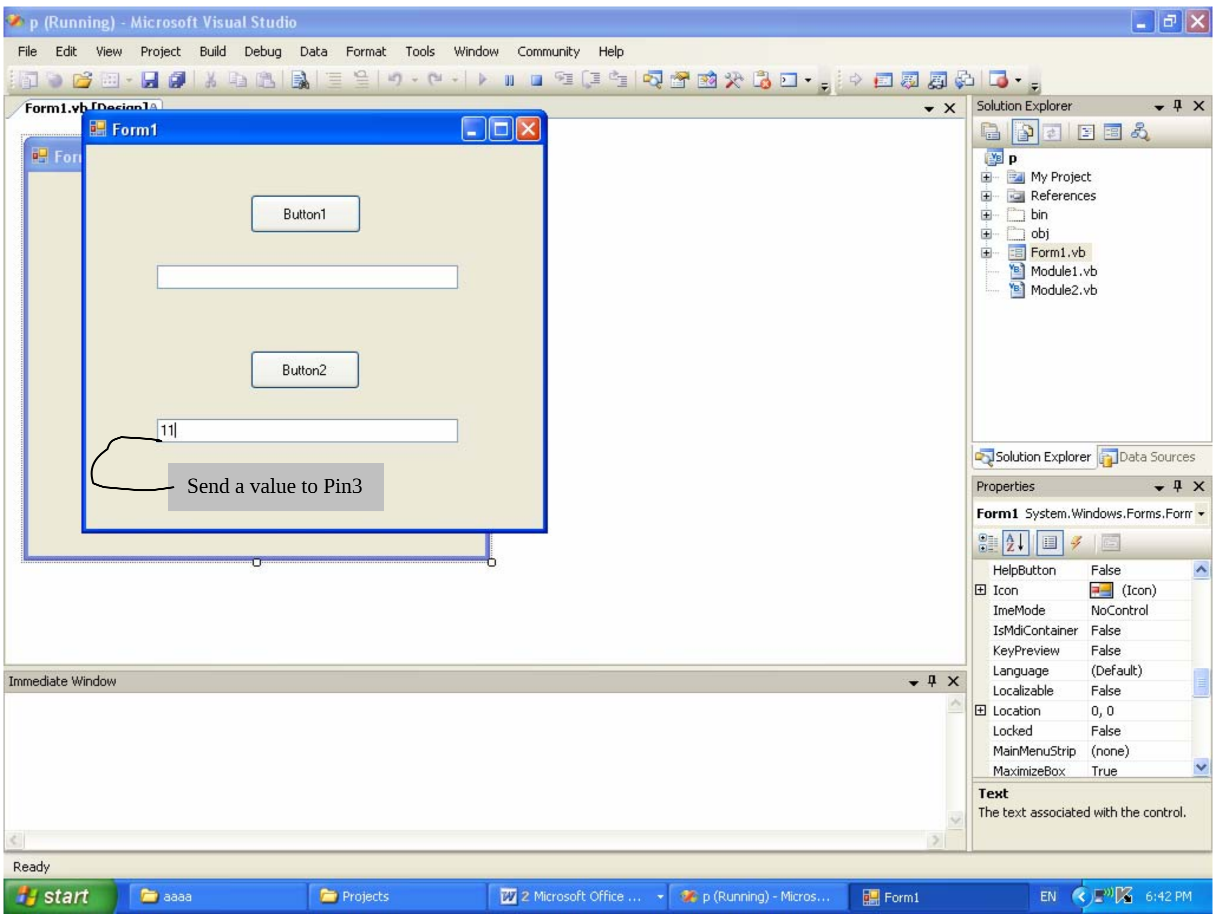


Figure 4.13: User-Interface illustrating the process of sending data to the parallel port

We have used a DLL (dynamic link library) file to communicate with the hardware (connection point). For example, in the program source code, we have to add the following DLL entry to read or write values from the parallel port (the complete program source code is listed in the appendix a (p. 82)):

```
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal
PortAddress As Short, ByVal Value As Short)
```

The Out32 function is specifically used to write values to data ports. In the example (see Fig. (4.13)), we have sent the value to Pin2 and Pin3 (data pins) to activate the intended switch as shown Fig. (4.11). An appropriate switch system for connecting terminals of points in circuits, is ULN2003 IC, which includes a seven Darlington drivers that are enabled to power up devices with 500mA [74] (see Fig. (4.11)).

By contrast, to measure the value of a resistor in our experiment, we can use GPIB (General Purpose Interface Bus) card and GPIB cable (see Fig. (4.12)),instead of using the parallel port. The main purpose of the general purpose-interface bus (GPIB) is to send information between two or more devices. Before any data is sent, the devices must be configured to send the data in the proper order and according to the proper protocol. The electrical specifications as well as the cables, connectors, control protocol, and messages required to allow information transfer between devices are defined by the IEEE-488 standard [13]. IEEE-488 supports data transfer at up to 1 Mbytes/sec. In addition to simple data transfers, the IEEE-488 standard defines a number of specialized commands for interface programming in the form of subroutines available as programming libraries for different programming languages such as C, Pascal, C#,VB, etc.

It is possible to read values measured by an instrumentation equipment, such as the Fluke 45 if such devices are equipped with hardware based on the IEEE-488 standard. GPIB cable allows us to connect the IEEE-488 based Fluke 45 with lab server, which must be expanded with PCI GPIB card. In the source code responsible for reading data from the Fluke 45, we have to add the following piece of program:

```
TextRead.Text = Reader.GetOhms ()
```

4.6.4 Rule-based e-tutor:

While carrying out an experiment, the recording component archives the students' interactions with the experiment, along with the e-tutor reactions. As previously noticed, the e-tutor proves the validity of the connections in an interactive manner. That is, it marks the unaccepted connections with a different color, and informs interactively the students about their failures. The client stores these connections in a text file in order to send them to the e-collaborative server. The way the e-tutor validates the wired electronic components on breadboard, works as follows:

- For every experiment, the e-tutor knows the lengths between the circuit elements.
- While a student connects the circuit terminals (highlighted circles) with each other, the client stores the lengths of the new drawn lines in a new text file (see Fig. (4.14)).
- If a student confirms the new changes on the circuit, the client sends the file containing the lengths to the laboratory server for comparison purposes; the comparison is made by if-then rules. The complete code for this part of program is listed in appendix a (p.83).
- The result of this comparison indicates the correctness of the manipulated connections. Accordingly, the lab server sends the client the new result to inform the student about his success or failure.
- If the connections are valid, the server connects the corresponding controllable switches in the experiment.

The pseudo code representation of the above discussed procedure is listed below:

```
Read coordinates from a text file
Set coordinates in an array
If the length of lines equal the length of lines that have been previously stored in the database
        Print “ok, the connection is valid”
Else
        Print “no, the connection is invalid”
```

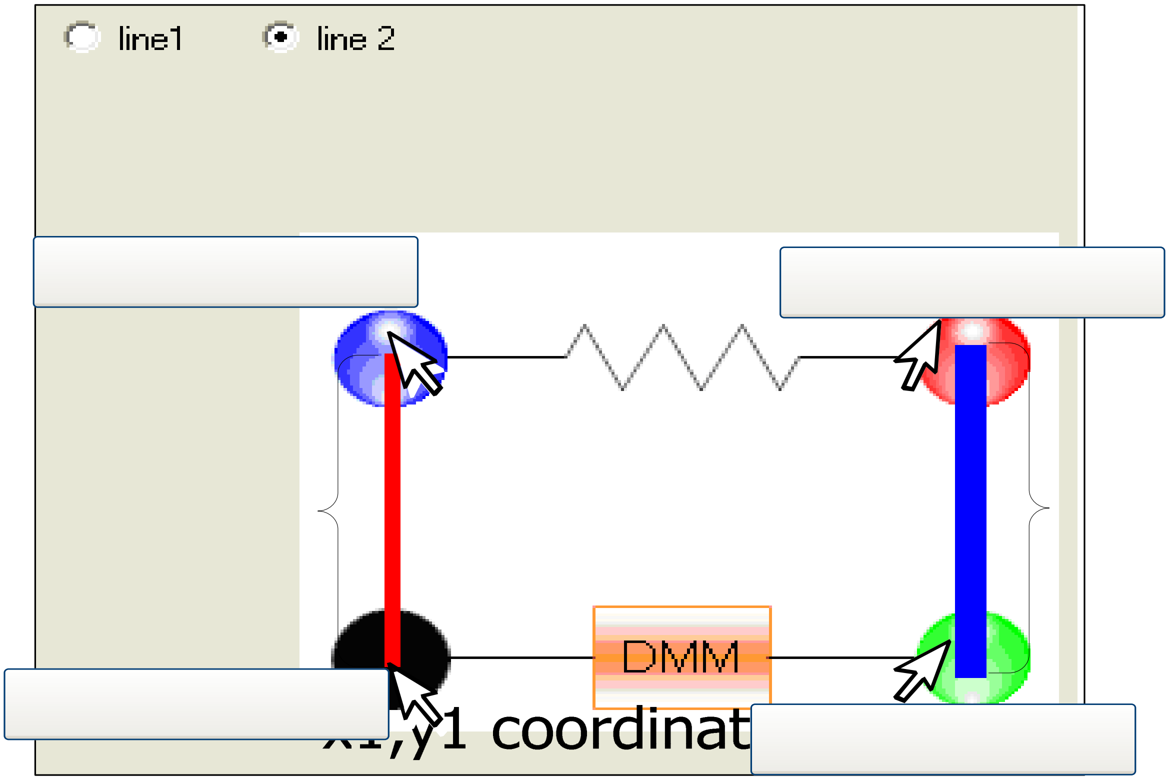


Figure 4.14: rule-based e-tutor set-up

After checking the points that have been manipulated by the student on the virtual breadboard by the e-tutor, a message is sent as feedback to the student about his/her work as shown in Fig.(4.15), (see Fig. (4.19), part D).

Length of straight line 1

x2,y2 coordinates of line1

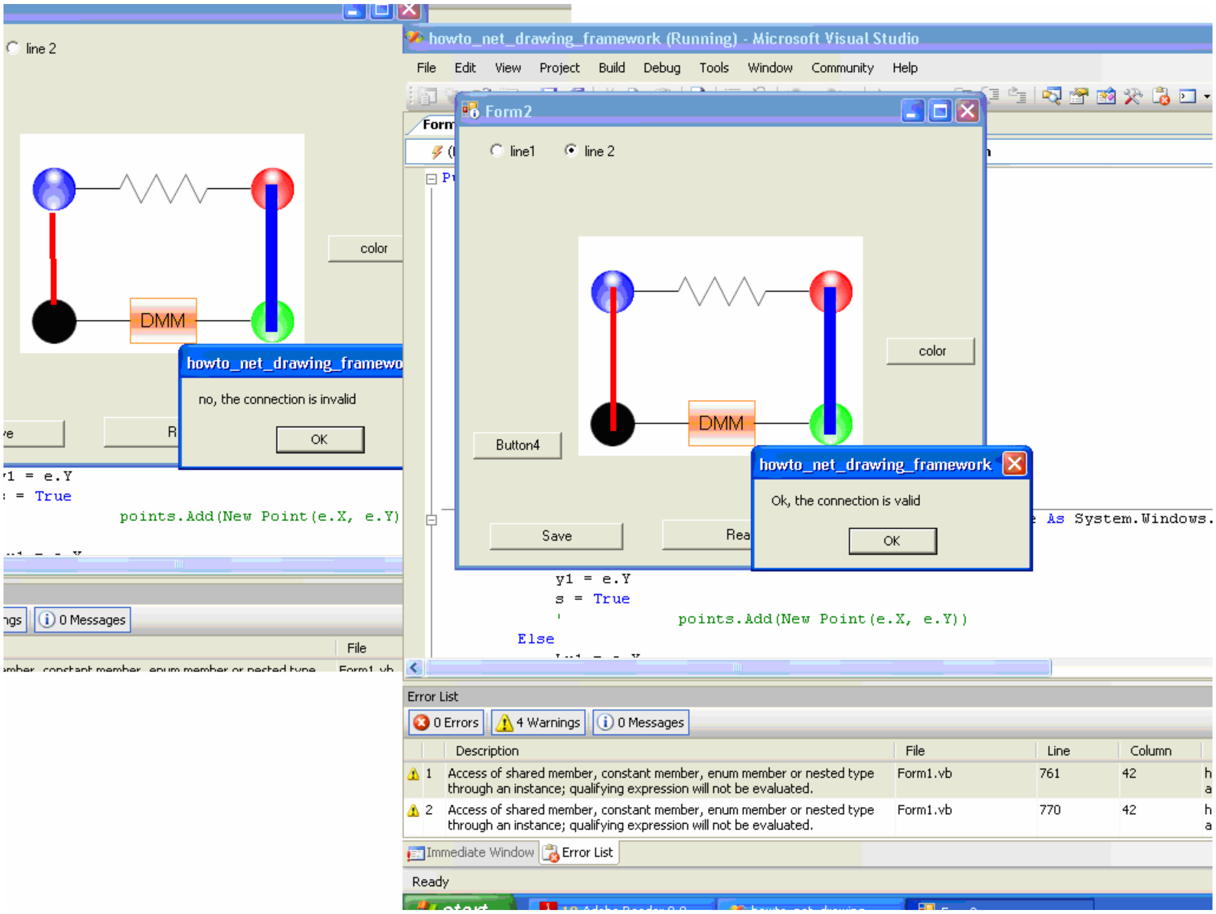


Figure 4.15: dialog box message as a feedback check of an e-tutor

4.6.5 Recording component:

The component for recording the students' activities has been implemented in vb.net to record the students' interaction with the user-interface, (see Fig. (4.19), part E). The following solution bypasses the problems mentioned in section (4.4.5):

- Capturing a student's screen

  While students are working on an electronic experiment, the recording component captures images every 1000 milliseconds. This is achieved through passing the screen area to the method "ScreenShot.CaptureImage". The complete listing of this code is given in appendix a (p.84). Captured images are saved as bitmaps (see Fig. (4.16)).

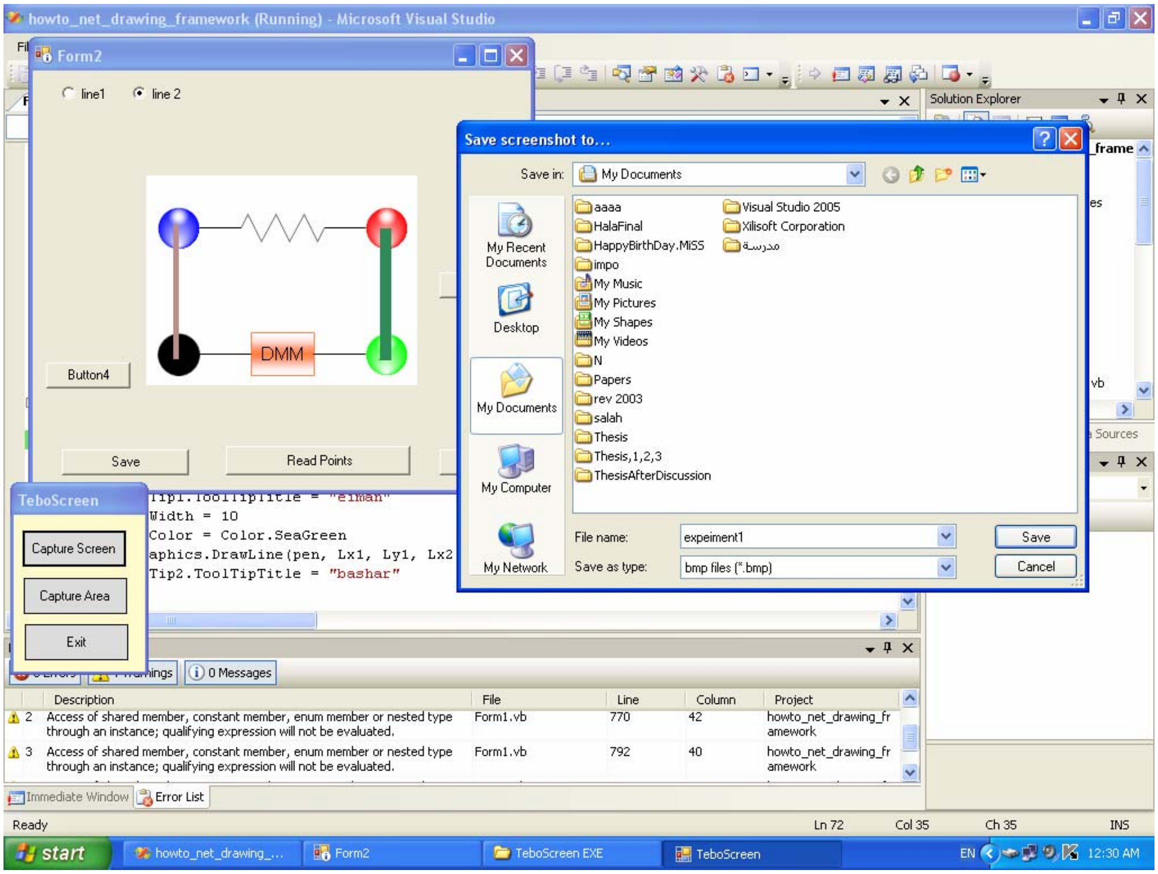


Figure 4.16: Example of a capture screen

- Capture Area

  While students are working on an electronic experiment, the recording component captures a specific area of a screen between starting point and end point of the selected area, and, then sends it to the method “SaveBitmap( )”, which will save the content into a bitmap file, the complete listing of this code is given in appendix a (p.84).See Fig.(4.17).

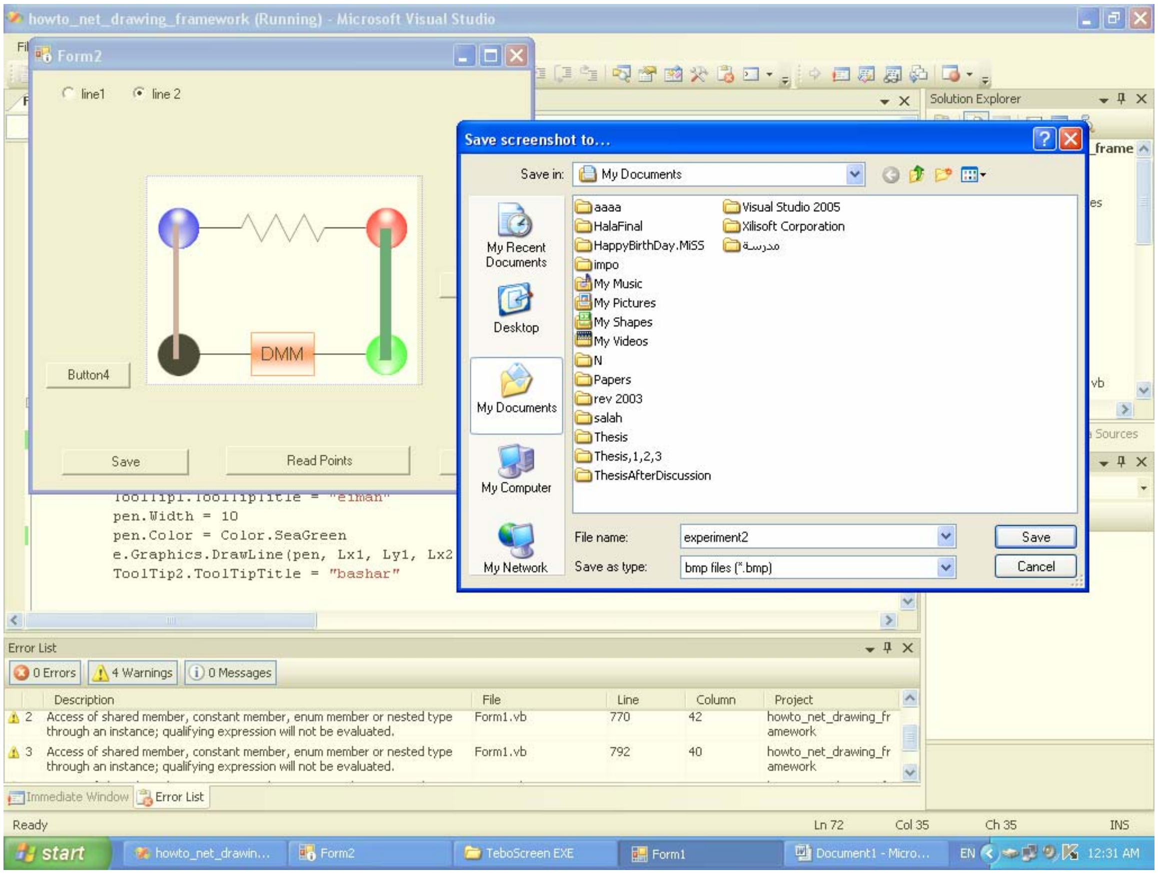


Figure 4.17: Example of a capture area

- Recording an experiment session in a movie file (.Avi File) using Microsoft Windows Media player, which will be needed for a future reference (see Fig. (4.18)). The complete listing of this code is given in appendix a (p.91).

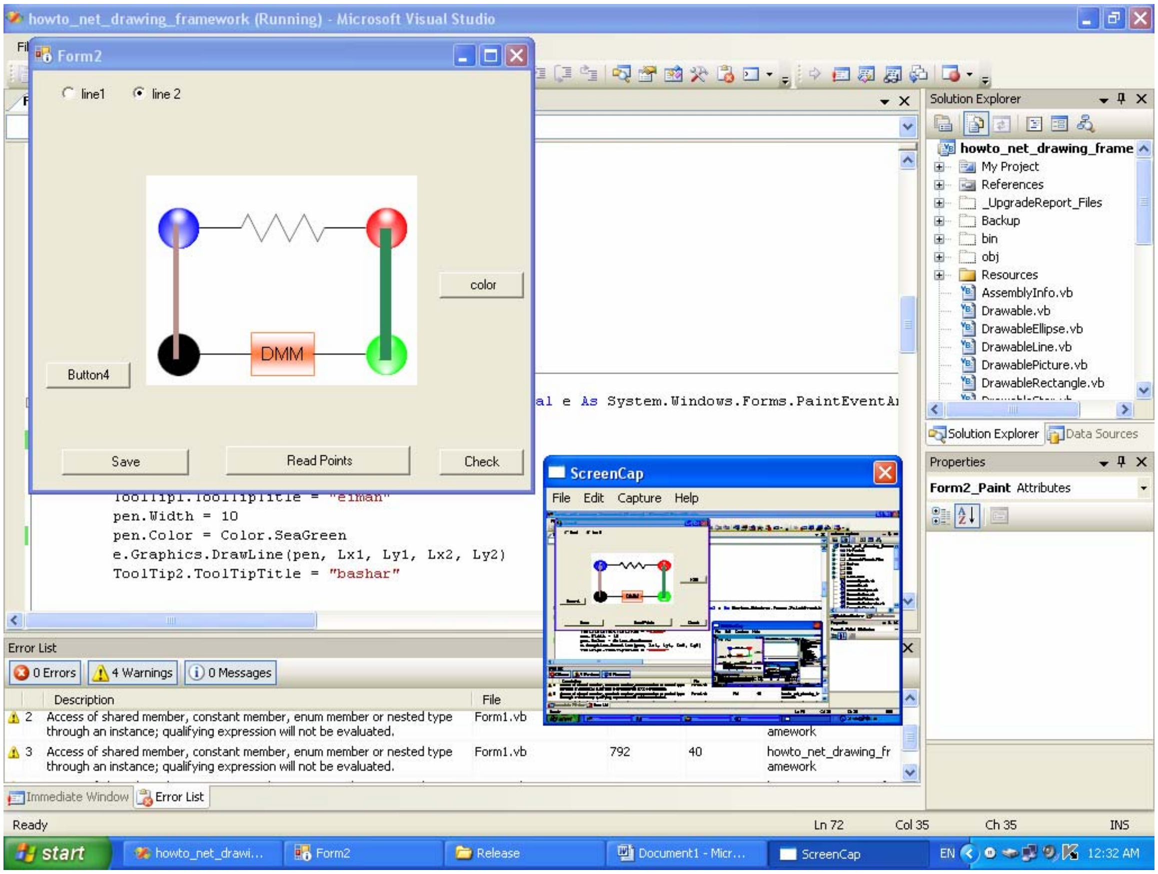
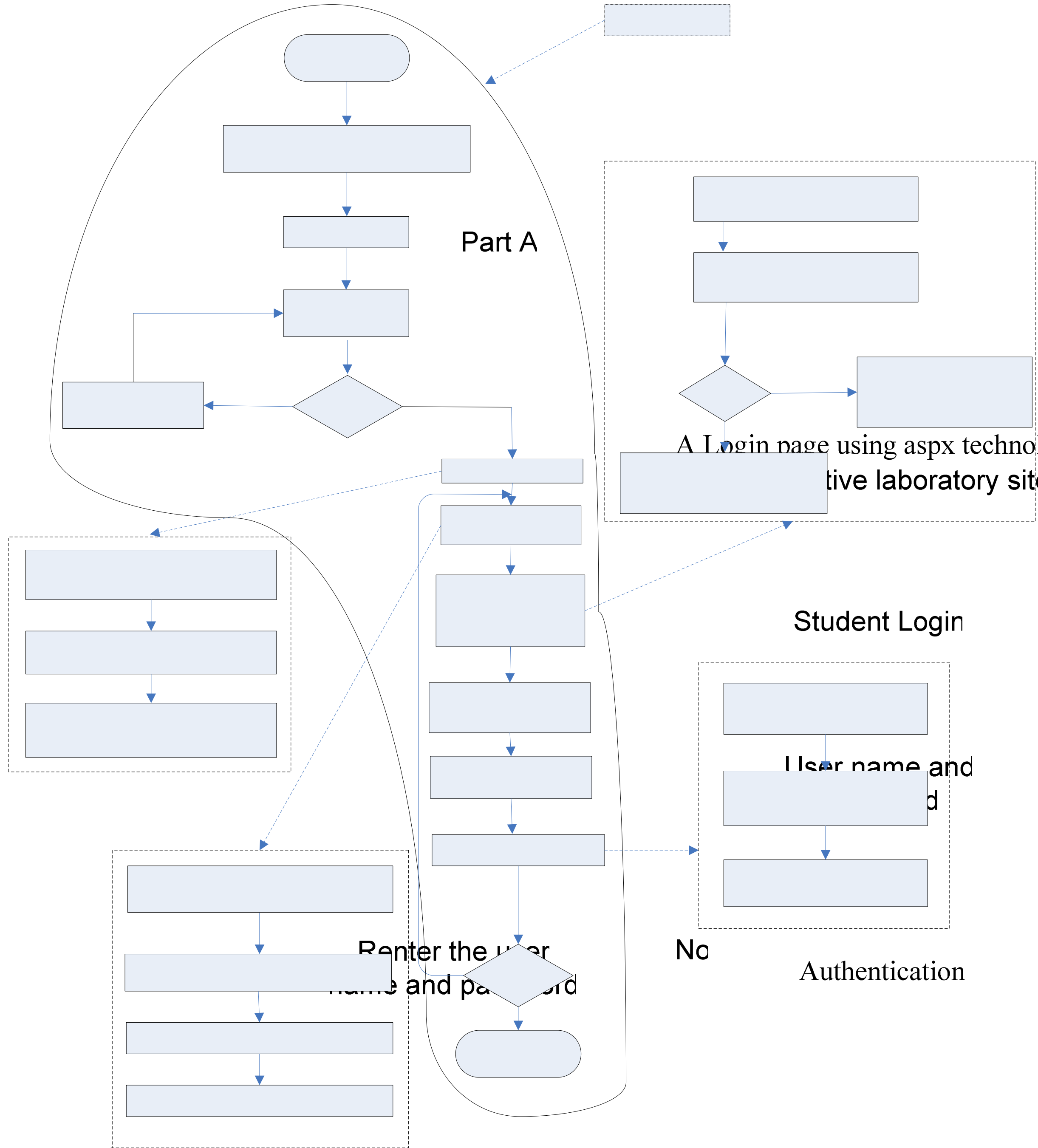


Figure 4.18: Example for recording an experiment by Movie File

4.6.6 Share desktop:

Using the same web-interface by a Share desktop, where a share desktop displays what you see on your desktop to all other participants. We used a third party tool software called "LogMeIn" [55], which is shown in Fig.(4.19). It is a suite of software services that provides remote access to computers over the internet. It displays what you see on your desktop to all students executing the experiment remotely, and gives you the flexibility to access and control your computers from anywhere, and then you can beam into that PC from anywhere using just a web browser.



Activate c

Circui
elec

Part B

Figure 4.19: Flowchart of the software part of the system

Enter student's name to be used in
conversations between students

Send the c
to the la
se

Send a text message to a particular student
or to all students at once.

The server
correspondin
switches in th

Exchange messages textually, supported
with emoticons

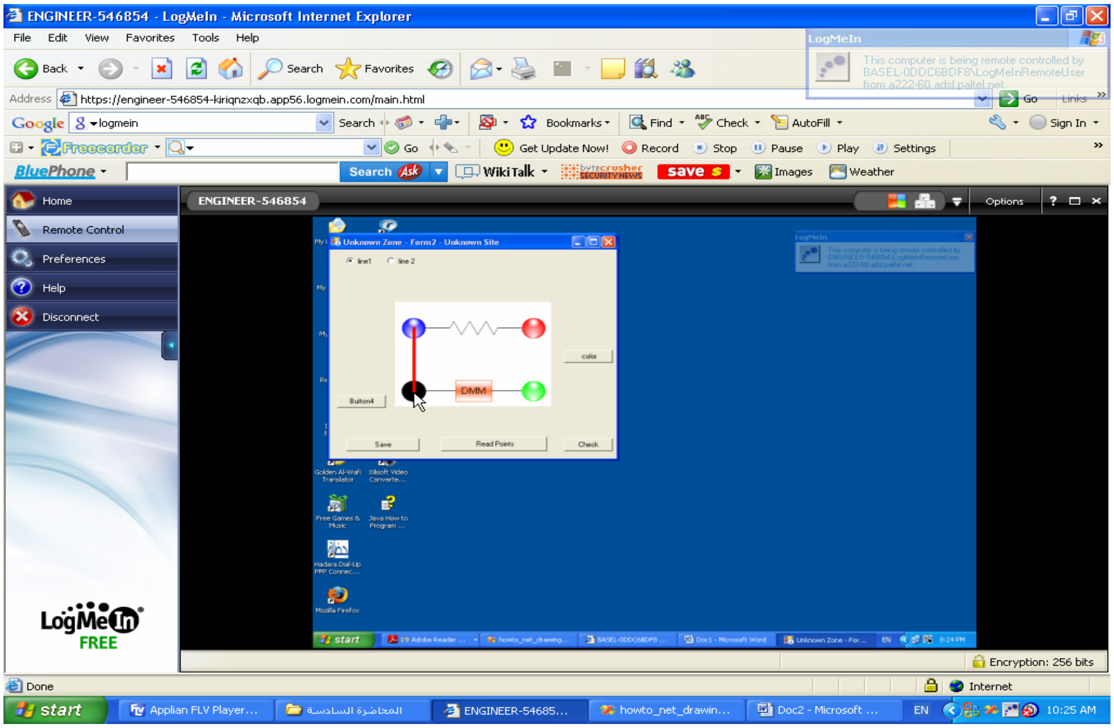


Figure 4.20 (a): Remote access to a screen of Student1 over the internet using the tool “LogMeIn”

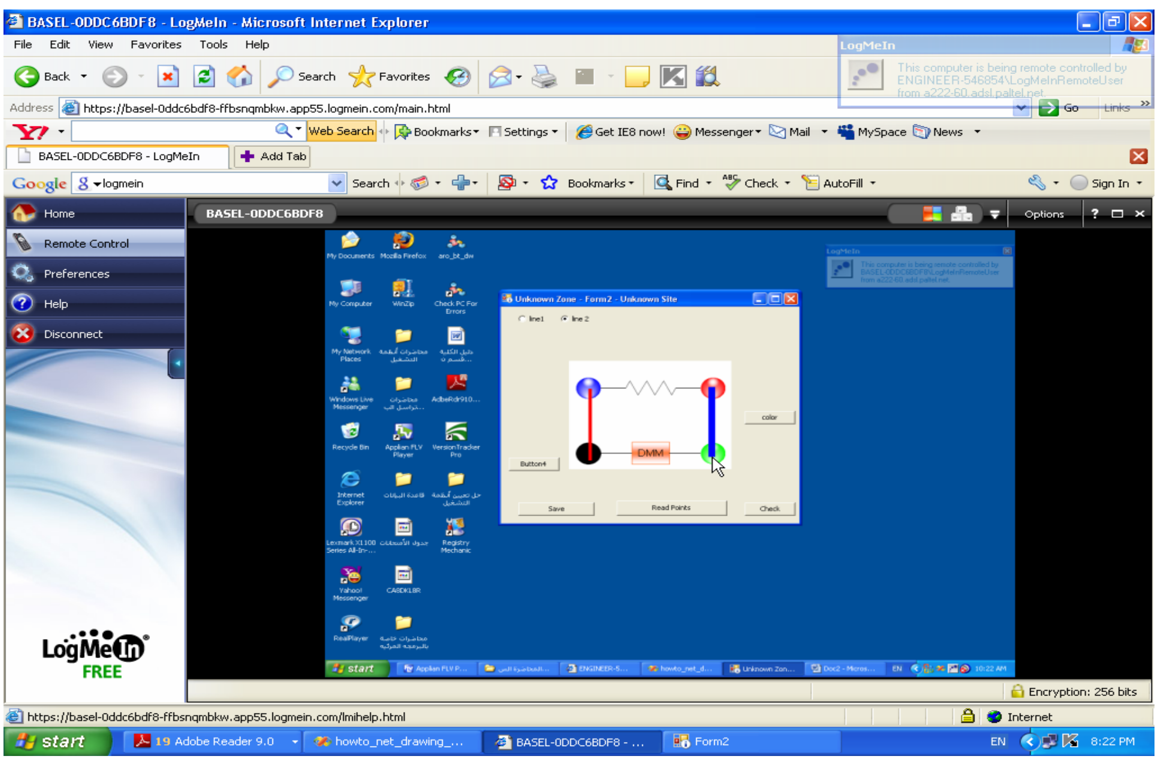


Figure 4.20 (b): Remote access to a screen of Student2 over the internet using the tool “LogMeIn”

Chapter 5

Conclusion and Future Work

Laboratory experiments are vital and cannot be separated from science and engineering education. In most cases, students were and still are coming to the laboratory to test the theory they learnt during lectures or had gained from different knowledge sources. In most cases, they work in groups and, ideally, they learn how to collaborate as a team. In some cases, they just follow instructions from their supervisors. Collaboration in a remote laboratory, which allows more students to work simultaneously, can be very similar to the real one except that one entity could be missing; namely, the supervisor might not be on hand. However in some cases, the supervisor’s help is still available via the communication channel.

5.1 Benefits of a distributed system architecture for collaborative e-learning with a tutor support

There are several benefits achieved through introducing distributed collaborative e-learning systems:

- Students can perform on real (physical) experiments remotely whenever they want and from anywhere.
- Two students or more supported by an experiment tutor are enabled to collaborate together, leading to facilitate the collaborative work of the remote learning scenario.
- Synchronus interaction between students and tutor is possible.
- Students are always provided with an intelligent automated help.

5.2 Drawbacks of distributed system architecture for collaborative e-learning

Nowadays, most of the drawbacks resulted from a distributed collaborative e-learning system are technological, for example,

- There are certain constraints such as bandwidth requirement, Internet time lag during data transfer, Internet traffic management and technical architectural needs. All of these problems need to be addressed.
- Time delay according to electronic components that used such as switches.

- Dot net framework for client (web interfaces) and server sides demands powerful computer systems, because this software takes a considerable amount of time to load the dot net framework at the client side over a low bandwidth network link.

5.3 Discussion

One of the problems faced by engineering students during the laboratory classes is that they have to perform the experiments in the laboratory in groups of three or four students. This sometimes gives negative support to some of the students, inequality in task division between them, and hence creates a lack of interest. In order to rectify such problems, students try to copy the results of their batch mates what might degrade the student's performance. Demonstrating a difficult concept to a learner who is having trouble understanding in traditional classroom, the instructor can adapt materials to meet individual student needs, so that flaws and deficiencies in both course design and the instructional materials used to support teaching can usually be compensated. Sometimes, if the number of students in the classroom is large, there is no opportunity to answer all questions and queries from asked by the students to their instructor; therefore, students do not receive feedback from their instructor. All the problems we have talked about previously do not only occur within the traditional classroom, but also occur in e-laboratories for collaborative working as well.

Such problems we have addressed so far can be rectified if the students are allowed to perform the experiments throughout the day via the Internet. These online laboratories help students to acquire experience achieved by conventional hands-on laboratories, but requiring a physical access to a building. Therefore, we have proposed in this thesis distributed system architecture for collaborative e-learning. The implementation of the system is discussed in chapter four.

While having explored and experimented collaborative working environments for remote experimentation, we found that it was necessary to consider other collaborative systems successfully applied in other fields such as air traffic and the military. This architecture puts students in a real environment for manipulating their experiments via the internet. Our distributed e-collaborative system could be made of the following components: a server, web-based user-interfaces, a management component, a rule-based system (e-tutor), a recording component, and the experiment itself. The web-based user-interfaces play a central role within the distributed e-collaborative system as it represents the window to the experiment. Their design takes various human-computer interaction rules for user interface into account. The web-based interface, which is realized as an integrated desktop, includes windows for displaying the experiment, a chat tool, a window for active users, and a dialog frame for session control.

In an e-laboratory, the interaction between instructor and students are indirect and the communication between them takes place through any form of media. Therefore, we proposed the use of a chat tool, which was implemented in vb.net for allowing effective interaction between students and instructor as it is the case in real environments. Due to this, the students can receive feedback or any assistance needed, sampling studying in traditional classrooms. Since a remote human tutor can't be online every time to support students, it was necessary to implement an automated helping system (e-tutor) to support students, and to provide a feedback to them when they need clarification on their work.

In this thesis, the e-tutor is realized as simplified rule-based system because our focus is mainly on collaborative systems and not on knowledge-based systems. The rules of the rule-based system consist of two parts: conditions and action. The e-tutor functions as an observer for students' actions manipulated on the experiment. Once the connections of the electronic elements on the virtual experiment are incorrect, the e-tutor displays a warning message and prevents contacting elements to be connected on the remote experiment board.

According to previous studies, the majority of current existing real laboratories for collaborative working are not constructed to allow the participants to collaborate in real time and are not designed to support students through an automated help at any time [8],[9] and [10].

Table (5.1) demonstrates a comparison between the methodologies and techniques of other collaborative e-laboratories compared with our thesis contribution, showing the advantages and disadvantages of each philosophy.

Table 5.1: A comparison between our thesis contributions with other collaborative e-laboratories

Property	Automated help (e-tutor)	Share desktop	Coordination (roles attribution)	Synchronous/ Asynchronous Communication	Real/Virtual Environment laboratory
[8]	none	none	none	Synchronous (chat tool)	Virtual (Simulation results)
[9]	none	none	none	Asynchronous	Virtual (Simulation results)
[10]	none	none	none	Synchronous (chat tool & video conference)	Virtual (Simulation results)
Our distribution	Rule-based e-tutor	Available “logmein”	Color coding	Synchronous (chat tool)	Real results

A wide variety of social factors can play a large part in online collaborative e-laboratory, namely; affecting the learner’s motivation, the ability to participate in an experiment, and communicate and coordinate with colleagues. Some of these factors include: cultural differences, language barriers, specially for students who are not native speakers of the language, facility with written communication such as chat tools (synchronous), social presence such as feeling of connectedness and participation, relationship among students and the dialogue that results between students, providing students with the opportunity to test and refine their understanding in an ongoing process. Once all of these factors are stimulated, the productivity of the students will be increased, leading to provide a positive learning and exchange of information between students. Accordingly, the learning process (environment) will also become more efficient. That is, an e-collaborative system environment is characterized as efficient when it enables the collaborating partners to exchange information through a coordinated communication. Concluding that for collaborative e-learning systems, the synchronous approach is superior to the asynchronous one regarding instructional support. Table (5.2) shows the differences between the conventional collaborative and internet-based e-collaborative learning environments.

Table 5.2: Differences between conventional collaborative and internet-based e-collaborative learning environments

conventional collaborative environments	e-collaborative learning environments
Instructor-centered	Learner-centered
Lecture-oriented	Collaborative and discussion-based
While teachers act as experts, students are perceived as novices	Student participates in team learning through learning from other learners and collaborates with the facilitator to create the learning process
Learning content is static	Content is dynamic
Emphasis is focused on evaluation and testing	Emphasis is focused on performance

In this research, several contributions to the field of collaborative e-learning environments are achieved, for example: the students are always provided with an automated help by an e-tutor, are placed in a real environment laboratory through a user-centered user-interface, are enabled to communicate with each other and with the tutor synchronously by a chat tool, and can access the laboratory from any place and at any time.

5.4 Future work

5.4.1 Further studies of interest:

Based on the results achieved in this thesis, more complicated engineering and science experiments could be investigated. As previously mentioned, our intention was to implement a simple experiment instead of a complicated one because we were interested to focus our discussion on the ideas of collaboration work, the main goal of this thesis. Access to the implemented experiment from any place by more than one student is possible enabled by the basic configuration of the proposed architecture. It would be interesting if we can test our ideas with more complicated experiments. Complicated engineering experiments with many electronic devices and equipments can be easily implemented through scaling by adding new programmable devices, electronic components and PC’s. One way the complexity of experiments can be reduced is the usage of programmable devices, which can manage a large number of experiments. For new experiments, new programmable devices must be installed and used.

It would be of great interest to measure how the introduction of complex experiments affects the interaction between students, the human tutor and the system; as well as whether the feedback returned from the simplified rule-based e-tutor to the students is satisfied and correct according to their mistakes manipulated on complex experiment circuits. Another problem that might appear through monitoring of complicated experiments is the clearness regarding inequality in task division between students solved in this thesis by means of color coding. Accordingly, a better version of the e-tutor based on knowledge-based techniques could be implemented to serve as an automated help system for user support in complex remote experimentation environments.

5.4.2 Obstacles:

While having developed and implemented the system, our implementation of the components of the proposed system have met several problems to be solved, for example, it was difficult to implement advanced window features by using web forms realized by asp dot net. Therefore, we were enforced to solve this problem through substituting conventional web-forms with windows. As obvious, the implementation of several visual interaction forms was mainly based on windows. To combine web-based forms with locally implanted windows, we were enforced to couple the experiment user–interface implemented as a window with the chat-tool implemented as an aspx page by means of links, the so-called HyperLink controls. Unfortunately, this solution is not ideal because the interaction will be negative influenced caused by integration problems. In addition to that, we have used a third party tool "LogMeIn" to function as a share application, with which we have difficulties to implement within vb.net.

References:

[1] B. Fields, P. Amaldi, A. Tassi, *Representing collaborative work: The Airport as Common Information Space*, Technical Report: IDC-TR-2003-003, Interaction Design Centre, School of Computing Science, Middlesex University, UK, 2003.

[2] S.D. Scott, M.L. Cummings, D.A. Graeber, W.T. Nelson, R.S. Bolia, "Collaboration Technology in Military Team Operations: Lessons Learned from the Corporate Domain," Proceedings of CCRTS 2006: the Command and Control Research and Technology Symposium, San Diego, CA, USA, 2006.

[3] J. Lonchamp, "*A Structured Chat Framework for Distributed Educational Settings*," 6th International Conference on Computer Supported Collaborative Learning - CSCL'05, Taipeh, Taiwan, 2005.

[4] "*Collaborative Tool Demonstration*," Peace and stability education, Workshop report, the United States Army Peacekeeping and Stability Operations Institute at Carlisle Barracks, Pennsylvania, USA, pp. 189-205, 2006.

[5] Y. Cao, G. Sharifi, Y. Upadrashta, J. Vassileva, "*A Case Study on Social Network in a Computer Game*," Proceedings of the second international joint conference on Autonomous agents and multiagent systems, Melbourne, Australia, pp. 954 – 955, 2003.

[6] A. Böhne, N. Faltin, B. Wagner, "*Synchronous tele-tutorial support in a remote laboratory for process control*," INNOVATIONS 2004: World Innovations in Engineering Education and Research, edited by W. Aung, R. Altenkirch, T. Cermak, R. W. King, L. M. Sanchez Ruiz, iNEER, USA, 2004.

[7] O. C. Santos, A. Rodríguez, E. Gaudioso. J. G. Boticario, "*Helping the tutor to manage a collaborative task in a web-based learning environment*," 11th international conference on artificial intelligence in education, Sydney, Australia, 2003.

[8] MJ. Callaghan, J. Harkin, TM. McGinnity, LP. Maguire, "*Client-Server Architecture for Collaborative Lecture-Led Remote Experimentation,*" 7th Distance Learning and the Internet Conference APRU DLI 2006, Japan, 2006.

[9] A. Bischoff and C. Rohrig, *"Remote Experimentation in a Collaborative Virtual Environment,"* Department of Electrical Engineering University of Hagen D-58084 Hagen, Germany.

[10] C. Schmid, "*Grid Supported Virtual Laboratories with Collaboration in Engineering Education,*" Ruhr-Universität Bochum, Bochum, Germany, iJOE International Journal of Online Engineering – www.i-joe.org– Volume 4, Special Issue 1: REV2008, July 2008.

[11] N. Peric et al, *"Virtual laboratory for automatic control and supervision challenges and opportunities,"* E-mail: nedjeljko.peric@fer.hr URL: http://www.rasip.fer.hr/peric.

[12] M. Cefalo, L. Lanari, G. Oriolo, M. Vendittelli, "The REAL Lab: Remote experiments for active learning," XLI AICA Annual Congress, Trento, IT, 2003.

[13] A.KManir, "*The Internet as a tool for interactive learning, teaching and research: Nigerian experience,"* Bayero University, Kano. Department of Library and Information Science Kano, Nigeria.

[14] I. Gustavsson, Johan Zackrisson, Henrik Akesson, Lars Hakansson, Ingvar Claesson, Thomas Lagoe," *Remote Operation and Control of Traditional Laboratory Equipment,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 1, 2006.

[15] M. Plaisent, L. Maguiraga, Prosper Bernard and Samir Larhrib," *Evaluating e-labs' Experimentation,*" University of Quebec in Montreal, Canada, michel.plaisent@uqam.ca, lassana.maguiraga@uqam.ca , prosper1@compuserve.com , mslarhrib@newconsul.com, http://www.ejel.org ©Academic Conferences Limited, Electronic Journal on e-Learning Volume 2 Issue 1 (February 2004) 195-202.

[16] M. Cooper,"*Remote laboratories in teaching and learning – issues impinging on widespread adoption in science and engineering education,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 1, Issue 1, 2005.

[17] Vladimir M Cvjetkovic, Yevgeniya S Sulema," *Remote Laboratory for Supporting e-Studies in Electronics,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 1, 2006.

[18] Sanjib Das, L. N. Sharma, A. K. Gogoi, "*Remote Communication Engineering Experiments Through Internet,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 1, 2006.

[19] J. Garcia-Zubia, D. Lopez-de-Ipina, P. Orduña," *Evolving towards better architectures for remote laboratories: a practical case,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 1, Issue 2, 2005.

[20] B. Akin," *Application of a Remote Lab: Single Phase PFC circuit,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 2, 2006.

[21] A. Hossien (1996) Interactive Education: Impact of the Internet on Learning & Teaching available at www. Altavista .com accessed 26/10/2006.

[22] E. Babulak, *"Interdisciplinary and ubiquitous internet technologies in support of automation,"* University of Quebec, Canada Staffordshire University, United Kingdom, E-mail: babulak@ieee.org.

[23] K.Zubrinić, D.Kalpic, *"The Web as Personal Learning Environment,"* University of Dubrovnik, Dubrovnik, Croatia, University of Zagreb, Zagreb, Croatia, 27 June 2008.

[24] Paul W. Magoha, Andrew W. Otieno, *"The global perspectives of transitioning to e-learning in engineering education," World Transactions on Engineering and Technology Education, Volume 3, No.2, 2004.*

[25] J. Preece & D. Maloney-Krichmar, " *Online Communities: Focusing on sociability and usability,*" University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250 , preece@umbc.edu , dkrichmar@erols.com, 14 January 2003.

[26] H. Hasnim, M. Z. Abdullah," *Remote Lab Generator (RLGen): A software tool using auto-generating technique to develop a remote lab,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 3, No. 4, 2007.

[27] A. LELEVE et al, *"Generic e-lab platforms and elearning standards,"* 20 avenue Albert EINSTEIN, 69621 VILLEURBANNE Cedex, FRANCE (arnaud.leleve@insa-lyon.fr) Laboratoire ICTT - INSA de LYON.

[28] A. Taneja, A. Kushwah, A. Gupta, Vipin B Vats," *TeleLab–A Remote Monitoring and Control System,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 3, Issue 1, 2007.

[29] Raymond C Almgren, James A Cahow," *Evolving Technologies and Trends for Innovative Online Delivery of Engineering Curriculum*", iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 1, Issue 1, 2005.

[30] N. Aliane, *"LABNET: A Remote Control Engineering Laboratory,"* Universidad Europea de Madrid, DACA, Madrid, 28671 (Spain), 17 November, 2006.

[31] P. Coquard, M. Guillemot, A. Leleve, D. Noterman, H. Benmohamed," *AIP-Primeca RAO Remote Laboratories in Automation,"* iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 4, Issue 1, February 2008.

[32] K. Zakova and M. Sedlak, *"Remote Control of Experiments via Matlab,"* Faculty of Electrical Engineering and Information Technology Slovak University of Technology, Bratislava, Slovakia, 18 July 2006.

[33] E. Stefan Brenner*," Virtual Instruments for the Circuit Analysis with Interactive Parameter Modification,"* iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 3, 2006.

[34] Hatem H. Elrefaei," *Online Library of Scientific Models, A New Way to Teach, Learn, and Share Learning Experience,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 4, Issue 2, May 2008.

[35] S. Uran, D. Hercog, K. Jezernik," *Remote Lab Experiment RC Oscillator for Learning of Control,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 4, 2006.

[36] O.F. Toader," *Remote Data acquisition and Instrument Control Using Labview and Appletview,"* Name Nuclear Engineering and Radiological Sciences, University of Michigan, Ann Arbor, MI USA.

[37] Sarosh H Patel, Tarek m Sobh," *Online Automation & Control: An Experiment in Distance Engineering Education,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 3, 2006.

[38] I. Gustavsson, J. Zackrisson, K. Nilsson, J. Garcia-Zubia, L. Hakansson, I. Claesson, T. Lago, " *A Flexible Electronics Laboratory with Local and Remote Workbenches in a Grid,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 4, Issue 2, May 2008.

[39] D. Geoffroy, T. Zimmer, M. Billaud,"*A remote laboratory for electrical engineering education,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 3, 2006.

[40] F. A. Candelas Herías, C. A. Jara Bravo, F. Torres Medina, "Flexible *virtual and remote laboratory for teaching Robotics,*" Current Developments in Technology-Assisted Education (2006).

[41] C. Mergl,"*Comparison of Remote Labs in Different Technologies,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 4, 2006.

[42] H. Mann," *Learning Dynamics and Control Using Remotely Tutored Simulation and Virtual Experiments,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 2, Issue 2, 2006.

[43] Michelle W. L. Fong, "*e-collaboration and virtual organizations,*" Victoria University, Australia, Copyright © 2005 by IRM Press, ISBN 1-59140-285-9.

[44] J. begole, M. beth rosson, and C. A. shaffer, "*Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing Systems,*" Virginia Polytechnic Institute, ACM Transactions on Computer-Human Interaction, Volume 6, No. 2, June 1999, Pages 95–132.

[45] John Wiley & Sons Ltd, Patterns for Computer-Mediated Interaction, West Sussex PO19 8SQ, England, Copyright © 2007.

[46] Curtis J. Bonk, *et al*, "*Applying Collaborative and e-Learning Tools to Military Distance Learning: A Research Framework,*" September 2000, United States Army Research Institute for the Behavioral and Social Sciences.

[47] A. Bischoff and C. Rohrig," *Streaming Audio/Video and Multiuser Virtual Reality based Environment for Collaborative Remote Experimentation,*" Department of Electrical Engineering, and Information Technology, University of Hagen, D-58084 Hagen, Germany.

[48] Jaacquelliine A. Burkke," *Collaborative Accounting Problem Solving Via Group Support Systems in a Face-to-Face Versus Distant Learning Environment,*" Information Technology, Learning, and Performance Journal, Volume 19, No. 2, Fall 2001.

[49] Michael J. Sipusic, Robert L. Pannoni, Randall B. Smith, John Dutra, James F. Gibbons, and William R. Sutherland, "*Virtual Collaborative Learning,*" © 1999 Sun Microsystems, Inc. All rights reserved. The SML Technical Report Series is published by Sun Microsystems Laboratories, of Sun Microsystems, Inc. Printed in U.S.A, SMLI TR-99-72 January 1999.

[50] R. Kuhlen, J. Griesbaum, T. Jiang, J. Konig, A. Lenich, P. Meier,T. Schütz, W. Semar, "*K3 – an e-Learning Forum with Elaborated Discourse Functions for Collaborative*

Knowledge Management," Department of Computer and Information Science University of Konstanz Germany, rainer.kuhlen@uni-konstanz.de.

[51] I. Hamburg, et al, Germany, e-mail: hamburg@iatge.de, Christiane Lindecke, Arbeitszeitberatung Dr. Lindecke, Gelsenkirchen, Germany Herbert ten Thij, I.E.R., Eindhoven, Holland, "*Social aspects of e-learning and blending learning methods,*" 4-th European conference-comm-line 2003, Bucharest , September 25-26 , 2003.

[52] Jose M. M. Ferreira, D. Mueller," *Online labs and the MARVEL experience,*" iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 1, Issue 1, 2005.

[53] NinthBridge, *"Distance Learning for Nonprofit Organizations,"* New York, NY 10001 212-561-8523.

[54] S. H. Godar, S. Pixy Ferris,"virtual and collaborative teams," William Paterson University, USA, October 2003.

[55] https://secure.logmein.com/home.asp?lang=en

[56] F. Gignac, "*Building Successful Virtual Teams,*" © 2005 ARTECH HOUSE, INC. 685 Canton Street, Norwood, MA 02062.

[57] J. Machotka and Z. Nedic, *"From the Collaborative Environment of the Remote Laboratory NetLab to the Global Collaboration,"* University of South Australia, Adelaide, Australia, iJOE International Journal of Online Engineering – www.i-joe.org, – Volume 4, Special Issue 1: REV2008, July 2008.

[58] C. Schmid, *"Grid Technologies for Virtual Laboratories in Engineering Education,"* Ruhr-Universität Bochum, Bochum, Germany, iJOE International Journal of Online Engineering – www.i-joe.org,–Volume 4, Issue 1, February 2008

[59] Deitel Associates, *Visual Basic 2005 How to Program*, Prentice Hall, 2006.

[60] R. Menchaca, *"Software Engineering, HCI Techniques and Java Technologies Joined to Develop Web-Based 3DCollaborative Virtual Environments,"* CLIHC'05, October 23-26, 2005, Cuernavaca, México. Copyright is held by the author(s). ACM 1-59593-2240.

[61] I. Sommerville, *Software Engineering* (8th Edition). Addison Wesley, 2007.

[62] T. Faison, *Component-Based Development with Visual C#*, John Wiley & Sons, 2002.

[63] B. Shneiderman, and C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition). Addison Wesley Longman, 2004.

[64] F. Hayes-Roth, "Rule-based systems," *Communications of the ACM*, vol. 28, Nr. 9, pp. 921-932, 1985.

[65] G. J. Nalepa and I. Wojnicki, *"Knowledge-Based Approach to the Executable Design Concept,"* Institute of Automatics,AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland gjn@agh.edu.pl, wojnicki@agh.edu.pl, 2007.

[66] M. Jarke, *"KBMS Requirements of Knowledge-Based Systems,"* In: J. W. Schmidt and C. Thanos (eds.): Foundations of Knowledge Base Management: Contributions from Logic, Databases, and Artificial Intelligence. Berlin: Springer, pp. 391-195.

[67] A. Abraham, *"Rule-based Expert Systems,"* Oklahoma State University, Stillwater, OK, USA, 2005 John Wiley & Sons, Ltd. ISBN: 0-470-02143-8.

[68] CEC, *CEC 488 programming and reference*, Part number 370966A-01, 2003.

[69] A. J. Caristi. *Ieee-488: General purpose instrumentation bus manual (professional and technical series)*, Academic Press, 1989.

[70] J. Axelson, *"parallel port complete,"* Programming, Interfacing, & Using the PC's Parallel Printer Port.

[71] G. Johannsen. Mensch-Maschine-Systeme. Berlin: Springer, 1993.

[72] J. R. Anderson, *Cognitive psychology and its implications* (5th ed.), New York: Worth, 2000.

[73] I. Gustavsson, Blekinge Institute of Technology, SE-372 25 Ronneby Sweden," *Traditional laboratory exercises and remote experiments in electrical engineering Education,"* International Conference on Engineering Education, July 21–25, 2003, Valencia, Spain.

[74] Seven darlington arrays, February 2002, © 2002 STMicroelectronics – Printed in Italy – All Rights Reserved STMicroelectronics group of companies.

Appendix A

Code implementation in the programming language “vb.net “

1. A web page called “CreateNew Student.aspx”:

```
<%@ Page Language="VB" MasterPageFile="~/UI.master" Title="Create a New User" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent" Runat="Server">
    <asp:CreateUserWizard ID="CreateUserWizard1" runat="server" BackColor="#F7F6F3"
BorderColor="#E6E2D8"
        BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em">
        <WizardSteps>
            <asp:CreateUserWizardStep runat="server">
            </asp:CreateUserWizardStep>
            <asp:CompleteWizardStep runat="server">
            </asp:CompleteWizardStep>
        </WizardSteps>
        <SideBarStyle BackColor="#5D7B9D" BorderWidth="0px" Font-Size="0.9em"
VerticalAlign="Top" />
        <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
        <SideBarButtonStyle BorderWidth="0px" Font-Names="Verdana" ForeColor="White" />
        <NavigationButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC" BorderStyle="Solid"
            BorderWidth="1px" Font-Names="Verdana" ForeColor="#284775" />
        <HeaderStyle BackColor="#5D7B9D" BorderStyle="Solid" Font-Bold="True" Font-Size="0.9em"
            ForeColor="White" HorizontalAlign="Center" />
        <CreateUserButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC" BorderStyle="Solid"
            BorderWidth="1px" Font-Names="Verdana" ForeColor="#284775" />
        <ContinueButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC" BorderStyle="Solid"
            BorderWidth="1px" Font-Names="Verdana" ForeColor="#284775" />
        <StepStyle BorderWidth="0px" />
    </asp:CreateUserWizard>
</asp:Content>
```

2. A Login page using aspx:

```
<%@ Page Language="VB" MasterPageFile="~/UI.master" Title="Login" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent" Runat="Server">
    <asp:Login ID="Login1" runat="server" BackColor="#F7F6F3" BorderColor="#E6E2D8"
BorderPadding="4"
        BorderStyle="Solid" BorderWidth="1px" CreateUserText="Click here to create  a new user"
        CreateUserUrl="~/CreateNewUser.aspx" DisplayRememberMe="False" Font-Names="Verdana"
        Font-Size="0.8em" ForeColor="#333333" DestinationPageUrl="~/ChatRoom1.aspx">
        <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True" Font-Size="0.9em"
ForeColor="White" />
        <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
        <TextBoxStyle Font-Size="0.8em" />
        <LoginButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC" BorderStyle="Solid"
BorderWidth="1px"
            Font-Names="Verdana" Font-Size="0.8em" ForeColor="#284775" />
    </asp:Login>
</asp:Content>
```

3. Chat tool window using aspx:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default"
validateRequest="false" %>

<%@ Register Assembly="ChatControl" Namespace="ChatControl" TagPrefix="cc1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Chat</title>
</head>
<body>
    <form id="form1" runat="server">
        Your Name: <asp:TextBox ID="txtUsername" runat="server">Anonymous</asp:TextBox>
        <br />
        <br />
    <div>
          
```

```
        <asp:Button ID="btnEnterChat" runat="server" Text="Enter Chat Room" />
        <br />
    </form>
</body>
</html>

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub btnEnterChat_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnEnterChat.Click
        Context.Items("UserName") = txtUsername.Text
        Server.Transfer("Chatwindow.aspx")
    End Sub
End Class


<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ChatRoom1.aspx.vb" Inherits="ChatRoom1" validateRequest="false" %>

<%@ Register Assembly="ChatControl" Namespace="ChatControl" TagPrefix="cc1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>WebChat </title>
</head>
<body>

    <form id="form1" runat="server">
    <div>
         <cc1:WebChat ID="WebChat2" runat="server" CallBackInterval="4" ChatTopic="Cooking" HistoryCapacity="20" Height="245px" Width="490px" />
                    
        <br />
        <br />
        <br />
        <asp:LoginStatus ID="LoginStatus1" runat="server" LogoutAction="RedirectToLoginPage"
            LogoutText="Exit" EnableTheming="True" Height="40px" LogoutPageUrl="~/Default.aspx" Width="72px" LogoutImageUrl="~/logo.jpg" LoginText="logging-out" />
                              
        <asp:HyperLink ID="HyperLink2" runat="server" Height="40px" NavigateUrl="~/Default.aspx"
            Width="144px">Recording Component</asp:HyperLink><br />
        <br />
        <br />
        <asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Default.aspx">Experiment Window</asp:HyperLink>
           
    </div>
    </form>
</body>
</html>



Partial Class Chatwindow
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If Not Me.IsPostBack Then
            Dim sName As String = "anonymous"
            On Error Resume Next
            sName = Context.Items("UserName").ToString
            WebChat2.UserName = sName
        End If
    End Sub
```

```
    Protected Sub WebChat1_Chatter(ByRef ChatText As String) _
    Handles WebChat2.Chatter
        System.Diagnostics.Debug.WriteLine("Page detected chatter event from WebChat control")

                'replace bad words...
        ChatText = ChatText.Replace("shit", "sh**")

' support emoticons
        ChatText = ChatText.Replace(":)", _
            "<img src='smiley.jpg' title=':)' />")
    End Sub

    Protected Sub LoginStatus1_LoggingOut(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.LoginCancelEventArgs) Handles LoginStatus1.LoggingOut

    End Sub
End Class
```

4. Store x-y coordinates in text file:

```
Private Sub Form2_MouseDown(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseDown
        If L1.Checked = True Then
            x1 = e.X
            y1 = e.Y
            s = True
        Else
            Lx1 = e.X
            Ly1 = e.Y
            s = True
        End If
        points.Add(New Point(e.X, e.Y))
    End Sub

    Private Sub Form2_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseMove
        If s = False Then Exit Sub

        If L1.Checked = True Then
            x2 = e.X
            y2 = e.Y
            Me.Invalidate()
        Else
            Lx2 = e.X
            Ly2 = e.Y
            Me.Invalidate()
        End If

    End Sub

    Private Sub Form2_MouseUp(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles Me.MouseUp
        If L1.Checked = True Then
            x2 = e.X
            y2 = e.Y
            Me.Invalidate()
            s = False
        Else
            Lx2 = e.X
            Ly2 = e.Y
            Me.Invalidate()
            s = False
        End If
        points.Add(New Point(e.X, e.Y))
    End Sub

    Private Sub Form2_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Me.Paint
        pen.Width = 5
        pen.Color = Color.Red
```

```
        e.Graphics.DrawLine(pen, x1, y1, x2, y2)

        ToolTip1.ToolTipTitle = "student1"
        pen.Width = 10
        pen.Color = Color.Blue
        e.Graphics.DrawLine(pen, Lx1, Ly1, Lx2, Ly2)
        ToolTip2.ToolTipTitle = "student2"

    End Sub
‘ select a color
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
        Dim c As New ColorDialog()
        Dim result As DialogResult = c.ShowDialog()
        If result = Windows.Forms.DialogResult.Cancel Then
            Return
        End If
        pen.Color = c.Color
        Me.Invalidate()
    End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        points.Clear()
        Me.Invalidate()
    End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Dim f As New System.IO.StreamWriter("e.txt", True)
        f.WriteLine(x1)
        f.WriteLine(y1)
        f.WriteLine(x2)
        f.WriteLine(y2)
        f.WriteLine(Lx1)
        f.WriteLine(Ly1)
        f.WriteLine(Lx2)
        f.WriteLine(Ly2)
        f.Close()
    End Sub
```

5. read or write values from the parallel port:

```
Module Module1
    Public Declare Function Inp Lib "inpout32.dll" _
        Alias "Inp32" (ByVal PortAddress As Integer) As Integer

End Module
Module Module2
    Public Declare Sub Out Lib "inpout32.dll" _
       Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

End Module



Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        'Declare variables to store read information and Address values
        Dim intAddress As Integer, intReadVal As Integer

        'Get the address and assign
        intAddress = 889

        'Read the corresponding value and store
        intReadVal = Module1.Inp(intAddress)

        txtStatus.Text = intReadVal.ToString()
```

```
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Dim intVal2Write As Integer

        'Read the corresponding value and store

        intVal2Write = Convert.ToString(txtWriteVal.Text)

        'porting is the name of the module
        Module2.Out(888, intVal2Write)
    End Sub
End Class
```

6. read a text file by e-tutor:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Dim f As New System.IO.StreamReader("e.txt")
        Dim array As Integer() = New Integer(0 To 7) {}

        x1 = CInt(f.ReadLine)
        array(0) = CInt(x1)
        y1 = CInt(f.ReadLine)
        array(1) = CInt(y1)
        x2 = CInt(f.ReadLine)
        array(2) = CInt(x2)
        y2 = CInt(f.ReadLine)
        array(3) = CInt(y2)
        Lx1 = CInt(f.ReadLine)
        array(4) = CInt(Lx1)
        Ly1 = CInt(f.ReadLine)
        array(5) = CInt(Ly1)
        Lx2 = CInt(f.ReadLine)
        array(6) = CInt(Lx2)
        Ly2 = CInt(f.ReadLine)
        array(7) = CInt(Ly2)
        f.Close()
        d1 = CInt(x2 - x1)
        d2 = CInt(y2 - y1)
        d3 = CInt(Lx2 - Lx1)
        d4 = CInt(Ly2 - Ly1)
        d11 = ((d1 ^ 2 + d2 ^ 2) ^ 0.5)
        d22 = ((d3 ^ 2 + d4 ^ 2) ^ 0.5)
        MsgBox(x1 & " " & x2 & " " & y1 & " " & y2 & " " & Lx1 & " " & Ly1 & " " & Lx2 & " " &
Ly2)
                MsgBox(d1 & " " & d2 & " " & d3 & " " & d4)
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

'IF-then Rules, where "w1, w2, w3, w4, r1, r2, r3, r4" are facts stored and compared with the
result have obtained by the student, to determine the validity of the connections.


If CBool((x1 >= w1) And (x1 <= w2) And (x2 >= w1) And (x2 <= w2) And (y1 >= r1) And (y1 <= r2)
And (y2 >= r3) And (y2 <= r4) And (Lx1 >= w3) And (Lx1 <= w4) And (Lx2 >= w3) And (Lx2 <= w4)
And (Ly1 >= r1) And (Ly1 <= r2) And (Ly2 >= r3) And (Ly2 <= r4)) Then
            MsgBox("Ok, the connection is valid")
        Else
            MsgBox("no, the connection is invalid")

        End If
    End Sub
```

7. Capture screen, capture area:

```
Module GlobalVar

    Public imgRect As Rectangle

End Module


Imports System
Imports System.Runtime.InteropServices
Imports System.Drawing
Imports System.Drawing.Imaging

Public Class ScreenCapture
    '/ <summary>
    '/ Creates an Image object containing a screen shot of the entire desktop
    '/ </summary>
    '/ <returns></returns>
    Public Function CaptureScreen(Optional ByVal theArea As Integer = 1) As Image
        'if the optional parameter theArea is not specified,get it from Area.
        Select Case theArea
            Case 1 'full screen
                Return CaptureWindow(User32.GetDesktopWindow())
            Case 2 'retangular area
                Return CaptureRectArea(User32.GetDesktopWindow(), imgRect)
            Case Else
                Return Nothing
                Stop
        End Select
    End Function 'CaptureScreen

    Public  Function  CaptureRectArea(ByVal  handle  As  IntPtr,  ByVal  capRect  As  Rectangle)  As
Image

        ' get te hDC of the target window
        Dim hdcSrc As IntPtr = User32.GetWindowDC(handle)

        ' create a device context we can copy to
        Dim hdcDest As IntPtr = GDI32.CreateCompatibleDC(hdcSrc)

        ' create a bitmap we can copy it to,
        ' using GetDeviceCaps to get the width/height
        Dim   hBitmap   As   IntPtr   =   GDI32.CreateCompatibleBitmap(hdcSrc,   capRect.Width,
capRect.Height)

        ' select the bitmap object
        Dim hOld As IntPtr = GDI32.SelectObject(hdcDest, hBitmap)

        ' bitblt over
        GDI32.BitBlt(hdcDest,   0,   0,   capRect.Width,   capRect.Height,   hdcSrc,   capRect.X,
capRect.Y, GDI32.SRCCOPY)

        ' restore selection
        GDI32.SelectObject(hdcDest, hOld)
        ' clean up
        GDI32.DeleteDC(hdcDest)
        User32.ReleaseDC(handle, hdcSrc)

        ' get a .NET image object for it
        Dim img As Image = Image.FromHbitmap(hBitmap)
        ' free up the Bitmap object
        GDI32.DeleteObject(hBitmap)

        Return img


    End Function

    '/ <summary>
    '/ Creates an Image object containing a screen shot of a specific window
```

```
    '/ </summary>
    '/ <param name="handle">The handle to the window. (In windows forms, this is obtained by
the Handle property)</param>
    '/ <returns></returns>
    Public Function CaptureWindow(ByVal handle As IntPtr) As Image

        ' get te hDC of the target window
        Dim hdcSrc As IntPtr = User32.GetWindowDC(handle)

        ' get the size
        Dim windowRect As New User32.RECT()
        User32.GetWindowRect(handle, windowRect)
        Dim width As Integer = windowRect.right - windowRect.left
        Dim height As Integer = windowRect.bottom - windowRect.top

        ' create a device context we can copy to
        Dim hdcDest As IntPtr = GDI32.CreateCompatibleDC(hdcSrc)

        ' create a bitmap we can copy it to,
        ' using GetDeviceCaps to get the width/height
        Dim hBitmap As IntPtr = GDI32.CreateCompatibleBitmap(hdcSrc, width, height)

        ' select the bitmap object
        Dim hOld As IntPtr = GDI32.SelectObject(hdcDest, hBitmap)

        ' bitblt over
        GDI32.BitBlt(hdcDest, 0, 0, width, height, hdcSrc, 0, 0, GDI32.SRCCOPY)

        ' restore selection
        GDI32.SelectObject(hdcDest, hOld)
        ' clean up
        GDI32.DeleteDC(hdcDest)
        User32.ReleaseDC(handle, hdcSrc)

        ' get a .NET image object for it
        Dim img As Image = Image.FromHbitmap(hBitmap)
        ' free up the Bitmap object
        GDI32.DeleteObject(hBitmap)
        Return img

    End Function 'CaptureWindow


    '/ <summary>
    '/ Helper class containing Gdi32 API functions
    '/ </summary>
    Private Class GDI32

        Public Shared SRCCOPY As Integer = &HCC0020
        ' BitBlt dwRop parameter
        <DllImport("gdi32.dll")> Public Shared _
     Function BitBlt(ByVal hObject As IntPtr, ByVal nXDest As Integer, ByVal nYDest As Integer,
ByVal nWidth As Integer, ByVal nHeight As Integer, ByVal hObjectSource As IntPtr, ByVal nXSrc
As Integer, ByVal nYSrc As Integer, ByVal dwRop As Integer) As Boolean
        End Function

        <DllImport("gdi32.dll")> Public Shared _
     Function CreateCompatibleBitmap(ByVal hDC As IntPtr, ByVal nWidth As Integer, ByVal
nHeight As Integer) As IntPtr
        End Function

        <DllImport("gdi32.dll")> Public Shared _
     Function CreateCompatibleDC(ByVal hDC As IntPtr) As IntPtr
        End Function

        <DllImport("gdi32.dll")> Public Shared _
     Function DeleteDC(ByVal hDC As IntPtr) As Boolean
        End Function

        <DllImport("gdi32.dll")> Public Shared _
     Function DeleteObject(ByVal hObject As IntPtr) As Boolean
```

```
        End Function

        <DllImport("gdi32.dll")> Public Shared _
        Function SelectObject(ByVal hDC As IntPtr, ByVal hObject As IntPtr) As IntPtr

        End Function
    End Class 'GDI32
       _

    '/ <summary>
    '/ Helper class containing User32 API functions
    '/ </summary>
    Private Class User32
        <StructLayout(LayoutKind.Sequential)> _
        Public Structure RECT
            Public left As Integer
            Public top As Integer
            Public right As Integer
            Public bottom As Integer
        End Structure 'RECT


        <DllImport("user32.dll")> Public Shared _
     Function GetDesktopWindow() As IntPtr
        End Function

        <DllImport("user32.dll")> Public Shared _
   Function GetForegroundWindow() As IntPtr
        End Function

        <DllImport("user32.dll")> Public Shared _
     Function GetWindowDC(ByVal hWnd As IntPtr) As IntPtr
        End Function

        <DllImport("user32.dll")> Public Shared _
        Function ReleaseDC(ByVal hWnd As IntPtr, ByVal hDC As IntPtr) As IntPtr
        End Function

        <DllImport("user32.dll")> Public Shared _
     Function GetWindowRect(ByVal hWnd As IntPtr, ByRef rect As RECT) As IntPtr
        End Function

        <DllImport("user32.dll")> Public Shared _
             Function GetClientRect(ByVal hWnd As IntPtr, ByRef rect As RECT) As IntPtr
        End Function
    End Class 'User32
End Class 'ScreenCapture



Imports System.Drawing.Graphics
Imports System.Math



Public Class inVisibleCapWin
    Dim start As Boolean = False
    Dim startX, startY As Integer
    Dim FromHotkey As Boolean
    Dim drawRect As Rectangle

    Sub PrepareCap()
        Me.Visible = True
        Me.TopMost = True
        Me.Activate()
    End Sub

    Sub FinishCap()
        Me.TopMost = False
        Me.Visible = False
```

```
        Me.Close()
    End Sub

    Private Sub inVisibleCapWin_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Me.MouseDown
        start = True
        Dim CursorPos As New Point
        CursorPos = Cursor.Position()
        startX = CursorPos.X
        startY = CursorPos.Y
        Debug.Print("start: " & startX & "," & startY)
    End Sub

    Private Sub inVisibleCapWin_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Me.MouseMove
        If start Then
            Dim CursorPos As New Point

            CursorPos = Cursor.Position()
            drawRect = makeRectangle(startX, startY, CursorPos.X, CursorPos.Y)

            'clear the screen
            Me.Invalidate(New Rectangle(0, 0, Me.Width, Me.Height))
            'draw the rectangle
            Me.Invalidate(drawRect)

        End If
    End Sub

    Private Sub inVisibleCapWin_MouseUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Me.MouseUp
        start = False
        FinishCap()
        imgRect = drawRect
        MainForm.SaveImageArea()
    End Sub

    Private Sub inVisibleCapWin_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
        Dim blackPen As New Pen(Color.Black, 2)
        ' Draw rectangle to screen.
        e.Graphics.DrawRectangle(blackPen, drawRect)
    End Sub

    Public Function makeRectangle(ByVal stX As Integer, ByVal stY As Integer, ByVal endX As
Integer, ByVal endY As Integer) As Rectangle
        If stX > endX Then
            makeRectangle.X = endX
        Else
            makeRectangle.X = stX
        End If
        If stY > endY Then
            makeRectangle.Y = endY
        Else
            makeRectangle.Y = stY
        End If
        makeRectangle.Width = Abs(endX - stX)
        makeRectangle.Height = Abs(endY - stY)
    End Function



End Class

Public Class MainForm

    'variabels

    'save path
    Dim strAutomaticPath As String
```

```
    'Automatic Capture Screen Mode
    Dim isAutomaticMode As Boolean

    'window state is Normaal(show) / Minimize
    Dim isShowClicked As Boolean

    'check when click to (CaptreScreen/Capture Area) if
    'Automatic Mode is Clicked and Cancel it
    Dim isbtnStartAutomaticMode As Boolean


    ' Panels Apperance
#Region "Panels Apperance"

    'Once Capture ...
    Private Sub btnPanel1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnPanel1.Click
        Dim sohwCurrentPanel As Boolean
        If isbtnStartAutomaticMode = True Then
            If MessageBox.Show("Automatic Cpture Mode is Running Are you want to cancel it ?",
"Cancel   Auto   Capture   Model",   MessageBoxButtons.YesNo,   MessageBoxIcon.Question)   =
Windows.Forms.DialogResult.Yes Then
                btnStop_Click(Me, e)
                ' autmatic capture mode working and we cancel it
                sohwCurrentPanel = True
            End If
        End If
        ' if autmatic capture mode working and we cancel it
        If sohwCurrentPanel = True Or isbtnStartAutomaticMode = False Then
            clearButttonsColor()
            btnPanel1.BackColor = Color.Gainsboro
            ClosePanels()
            btnExit.BringToFront()
            Panel2.Dock = DockStyle.Fill
        End If


    End Sub


    'Automatic Capture ...
    Private Sub btnPanel2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnPanel2.Click
        clearButttonsColor()
        btnPanel2.BackColor = Color.Gainsboro
        ClosePanels()
        btnExit.BringToFront()
        Panel3.Dock = DockStyle.Fill
    End Sub


    'Capture Area ...
    Private Sub btnPanel3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnPanel3.Click
        Dim sohwCurrentPanel As Boolean
        If isbtnStartAutomaticMode = True Then
            If MessageBox.Show("Automatic Cpture Mode is Running Are you want to cancel it ?",
"Cancel   Auto   Capture   Model",   MessageBoxButtons.YesNo,   MessageBoxIcon.Question)   =
Windows.Forms.DialogResult.Yes Then
                btnStop_Click(Me, e)
                sohwCurrentPanel = True
            End If
        End If
        If sohwCurrentPanel = True Or isbtnStartAutomaticMode = False Then
            clearButttonsColor()
            btnPanel3.BackColor = Color.Gainsboro
            ClosePanels()
            btnExit.BringToFront()
            Panel4.Dock = DockStyle.Fill
        End If
```

```
    End Sub



    ' For Interface ...
    Private Sub ClosePanels()
        Panel2.Dock = DockStyle.None
        Panel3.Dock = DockStyle.None
        Panel4.Dock = DockStyle.None
    End Sub
    Private Sub clearButttonsColor()

        ' LemonChiffon()
        btnPanel1.BackColor = Color.LemonChiffon
        btnPanel2.BackColor = Color.LemonChiffon
        btnPanel3.BackColor = Color.LemonChiffon
    End Sub


    'Exit Button ...
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnExit.Click
        Application.Exit()
    End Sub

    'Minimize Button ...
    Private Sub btnMinimize_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnMinimize.Click
        Me.Hide()
        CaptureScreen.ShowBalloonTip(1)  'NotifyICON
        isShowClicked = False
    End Sub

#End Region


    'Capture Buttons Events
#Region "Capture Buttons Events "
    'Capture Full screen ...
    Private Sub btnCaptureScreen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCaptureScreen.Click


        Try

            Dim clsScreen As ScreenCapture = New ScreenCapture()
            SaveFileDialog1.Filter = "Bitmap (*.bmp)|*.bmp|GIF (*.gif)|*.gif|JPEG
(*.jpg)|*.jpg|TIF(*.tif)|*.tif|PNG (*.png) |*.png"
            SaveFileDialog1.ShowDialog()
            Dim strPath As String = SaveFileDialog1.FileName

            If strPath <> "" Then
                Me.Hide()
                Threading.Thread.Sleep(1000)
                clsScreen.CaptureScreen(1).Save(strPath, Imaging.ImageFormat.Jpeg)
                Me.Show()
            Else
                MessageBox.Show("Please Enter Picture name !!")
            End If
        Catch ex As Exception
            MessageBox.Show("Unexpected Error Occurred, Please Try Again Later ")
        End Try

    End Sub

    'Capture Sscreen Area ...
    Private Sub btnCaptureArea_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCaptureArea.Click
        Try
            'to Select Area ( Select Rectangle )
```

```
            inVisibleCapWin.PrepareCap()
        Catch ex As Exception
            MessageBox.Show("Unexpected Error Occurred, Please Try Again Later ")
        End Try
    End Sub

    'Automatic Capture Full screen ...
    Private  Sub  btnStartAutomaticMode_Click(ByVal  sender  As  System.Object,  ByVal  e  As
System.EventArgs) Handles btnStartAutomaticMode.Click
        If numSeconds.Value > 0 Then
            If FolderBrowserDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
                If FolderBrowserDialog1.SelectedPath <> "" Then

                    strAutomaticPath = FolderBrowserDialog1.SelectedPath
                    isAutomaticMode = True
                    Timer1.Interval = 1000 * numSeconds.Value 'to convert time to Second
                    btnStop.Visible = True
                    btnStartAutomaticMode.Visible = False
                    Timer1.Start()
                    isbtnStartAutomaticMode = True
                    Me.Hide()
                    CaptureScreen.ShowBalloonTip(0.1) 'NotifyICON
                End If
            End If
        Else
            MessageBox.Show("Plrase Enter Captuer Screen Time !!")
        End If
    End Sub

    'Stop Capture
    Private  Sub  btnStop_Click(ByVal  sender  As  System.Object,  ByVal  e  As  System.EventArgs)
Handles btnStop.Click
        Timer1.Stop()
        btnStop.Visible = False
        btnStartAutomaticMode.Visible = True
        isbtnStartAutomaticMode = False
    End Sub

    'Save Screen Area just to wait untile user select Area ...
    Public Sub SaveImageArea()
        Try

            Dim clsScreen As ScreenCapture = New ScreenCapture()
            SaveFileDialog1.Filter    =    "Bitmap    (*.bmp)|*.bmp|GIF    (*.gif)|*.gif|JPEG
(*.jpg)|*.jpg|TIF(*.tif)|*.tif|PNG (*.png) |*.png"
            SaveFileDialog1.ShowDialog()
            Dim strPath As String = SaveFileDialog1.FileName

            If strPath <> "" Then
                Me.Hide()
                Threading.Thread.Sleep(1000)
                clsScreen.CaptureScreen(2).Save(strPath, Imaging.ImageFormat.Jpeg)
                Me.Show()
            Else
                MessageBox.Show("Please Enter Picture name !!")
            End If
        Catch ex As Exception
            MessageBox.Show("Unexpected Error Occurred, Please Try Again Later ")
        End Try
    End Sub

    'Timer Function
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick

        'take pic every ** sec
        Try
            Dim clsScreen As ScreenCapture = New ScreenCapture()
            If strAutomaticPath <> "" Then
                If Not isShowClicked Then
                    Me.Hide()
```

```
                End If
                clsScreen.CaptureScreen(1).Save(strAutomaticPath + "\" +
DateTime.Now.ToString("hh-mm-ss") + ".jpg", Imaging.ImageFormat.Jpeg)
            Else
                MessageBox.Show("Please Enter Picture name !!")
            End If
        Catch ex As Exception
            MessageBox.Show("Unexpected Error Occurred, Please Try Again Later ")
        End Try

    End Sub

#End Region


    'Menu
#Region "Menue"
    Private Sub mnuShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuShow.Click
        Me.Show()
        Me.TopMost = True
        isShowClicked = True
        Me.TopMost = False
    End Sub

    Private Sub mnuStopCapture_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuStopCapture.Click
        Timer1.Stop()
        btnStop.Visible = False
        btnStartAutomaticMode.Visible = True
        isbtnStartAutomaticMode = False
        Me.Show()

    End Sub

    Private Sub mnuMinimize_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuMinimize.Click
        Me.Hide()
        CaptureScreen.ShowBalloonTip(1)  'NotifyICON
        isShowClicked = False
    End Sub

    Private Sub MainForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Make Start Automatic mode Default
        btnPanel2_Click(Me, e)
    End Sub
#End Region

End Class
```

8. Record an experiment by Movie File (.Avi File):

```
Imports WMEncoderLib
Imports WMPREVIEWLib

Imports System.IO
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

         InitializeComponent()

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
```

```
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    Private components As System.ComponentModel.IContainer

    Friend WithEvents Button1 As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.Button1 = New System.Windows.Forms.Button
        Me.SuspendLayout()

        Me.Button1.Location = New System.Drawing.Point(48, 80)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New System.Drawing.Size(216, 23)
        Me.Button1.TabIndex = 1
        Me.Button1.Text = "Write To file and Close application"

        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(292, 266)
        Me.Controls.Add(Me.Button1)
        Me.Name = "Form1"
        Me.Text = "Form1"
        Me.ResumeLayout(False)

    End Sub

#End Region
    Dim Encoder As WMEncoder
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        ' Create a WMEncoder object.

        Encoder = New WMEncoder

        Dim SrcGrp As IWMEncSourceGroup2
        Dim SrcGrpColl As IWMEncSourceGroupCollection
        SrcGrpColl = Encoder.SourceGroupCollection
        SrcGrp = SrcGrpColl.Add("SG_1")

                Dim SrcVid As IWMEncVideoSource2
        Dim SrcAud As IWMEncAudioSource
        SrcVid = SrcGrp.AddSource(WMENC_SOURCE_TYPE.WMENC_VIDEO)
        SrcAud = SrcGrp.AddSource(WMENC_SOURCE_TYPE.WMENC_AUDIO)

        ' Identify the source files to encode.
        SrcVid.SetInput("ScreenCap://ScreenCapture1")
        SrcAud.SetInput("Device://Default_Audio_Device")

        ' Choose a profile from the collection.
        Dim ProColl As IWMEncProfileCollection
        Dim Pro As IWMEncProfile
        Dim i As Integer
        Dim lLength As Long

        ProColl = Encoder.ProfileCollection
        lLength = ProColl.Count

        For i = 0 To lLength - 1
            Pro = ProColl.Item(i)
            If Pro.Name = "Windows Media Video 8 for Local Area Network (384 Kbps)" Then
                SrcGrp.Profile = Pro
                Exit For
            End If
        Next

                Dim Descr As IWMEncDisplayInfo
        Descr = Encoder.DisplayInfo
        Descr.Author = "Armoghan Asif"
```

```
        Descr.Copyright = "Copyright information"
        Descr.Description = "Text description of encoded content"
        Descr.Rating = "Rating information"
        Descr.Title = "Title of encoded content"

                Dim Attr As IWMEncAttributes
        Attr = Encoder.Attributes
        Attr.Add("URL", "www.adnare.com")

        ' Specify a file object in which to save encoded content.
        Dim File As IWMEncFile
        File = Encoder.File
        File.LocalFileName = "C:\OutputFile.avi"

                SrcVid.CroppingBottomMargin = 2
        SrcVid.CroppingTopMargin = 2
        SrcVid.CroppingLeftMargin = 2
        SrcVid.CroppingRightMargin = 2

        ' Start the encoding process.
        Encoder.Start()

    End Sub


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If Encoder.RunState Then
            Encoder.Stop()
            Application.Exit()
        End If
    End Sub
End Class
```

بيئات العمل التعاوني للتجاربِ (في المختبرات الهندسية) عن بعد (على شبكة الإنترنتِ)

إعداد : إيمان عبد الغني

إشراف : الدكتور صلاح الدين عودة

الملخّص

هذا البحث يركز على تطوير أنظمة حاسوبية على شبكة الإنترنت لدعم العمل التعاوني عن بعد بين الطلاب المتواجدين في أماكن مختلفة أثناء قيامهم بأداء تجارب مختبراتهم الهندسية.

بيئات العمل التعاونية للتعلم عن بعد تعتبر شكل جديد مِن المختبراتِ المعاصرة التي يمكن التواصل معها عن طريق الشبكة العنكبوتية (الإنترنت). في الوقت الحاضر، أغلبية المختبراتِ التي يتحكم بها عن بعد لا تسَمح للطلاب بتنفيذ التجارب الهندسية أو العلمية بشكل تعاوني ومتزامن. لجَعْل هذه البيئةِ التَعَليّمية ممكنة ، يَجِبُ أنْ نَسْمحَ للمشاركين المختلفين للاتصال والتَنسيق مَع بعضهم البعض بطريقةٍ سلسة مُنْتِجة من خلال الإتصالِ المتزامنِ والتنسيق بين مختلف المشاركين ، مما يعني أنه يمكن للطلاب إجراء التجربة في وقت واحد. في إطار هذه العمليةِ، يُمْكِنُ تحقيق التنسيق بين الطلاب باستخدام أدواتِ الدردشةِ.

في الآونة الأخيرةِ، العديد من بيئاتَ متعددة المستخدمين التعاونية طُبقت بنجاح في العديد مِنْ الاستخدمات مثل أنظمةِ مراقبة الحركة الجوّيةِ، وأنظمة الفرق العسكرية ، وأدوات الدردشةِ، وألعاب متعددة اللاعبين. وبالتالي، فهم الأفكار والتقنيات التي تقف وراء هذه الأنظمة يمكن أن تكون ذات أهمية كبيرة في المساهمة بأفكار بناءة لبيئات العمل التعاوني للتعليم الإلكتروني.

في هذه الإطروحةِ يتم طرح مواصفات المختبر الحقيقي التعاوني، الذي يَسْمحُ لطالبين أو أكثر لإجْراء التجاربِ عن بعد في نفس الوقت كفريق متعاون.حيث تم بناءاً على ذلك إقتِراح بنية لنظام لإنْجاز العملِ التعاوني بين الطلابِ. هذا النظامِ مكون من مجموعة من الأنظمة الفرعية والمكوّناتِ التي تسمح للطلاب بالتفاعل مع بعضهم البعض لإنجاز تجاربهم المختبرية الإعتيادية الموجودة في كلية الهندسة عن بعد وبشكل متزامن. ولكن هناك مشاكل كثيرة لا تزال قائمة في التعليم التعاوني عن بعد، في الغالب، بينما يُؤدّي الطلاب التجربة عن بُعد ، لا يجدون في معظم الأحيان مساعدة لتحقيق هدفهمَ سواء من الطلاب او من معلمهم وعدم المساواة في تقسيم المهام بينهم. ولذلك ، فإن الطلاب سوف يفقدون الحافز مما يؤدي الى فشل التجربة مؤثرا ذلك سلباً على عملية التعلم.

في هذه الإطروحة، النظام المقترح والذي يستند الى مبدأ بنية الموكل-الخادم (client-server architecture) يمّكن مجموعة من الطلاب بأدآء تجاربهم ويسمح لهم بمناقشة وتنفيذ الحلول مدعومين بمساعدة إمّا معلم إنساني (human tutor) أو معلم افتراضي (e-tutor) الذي يستمد معرفته عن طريق قواعد تم تخزينها مسبقا في قاعدة بياناته. إنّ أداة الدردشة تشكل المكونة الأساسية في النظام التي تؤدي الى تسهيل التفاعل التزامني بين الطلاب أنفسهم من ناحية والمعلم من ناحية أخرى وتشعرهم بأنهم في بيئة مختبر حقيقية وليست إفتراضية .علاوة على ذلك فإنّ إستخدام ترميز الألوان (color coding) للتمييز بين مساهمات الطلابِ المُتَعاوَنين يشكل محورا اساسيا يدور حوله العمل التعاوني .

في هذه الإطروحة، قمنا ببناء مكونات النظام باستخدام بيئات تطوير تستند على تكنولوجيا (.net technology) التي تشكل بيئة تطويرية متكاملة (IDE) مدعمة بمجموعة مِنْ المُنتَجاتِ والتقنيات لتسهيل عملية تطويرِ برامجَ البيئةِ التعاونيةِ على الإنترنتِ، بالإضافة إلى واجهة المستخدم الرسومية التفاعلية (GUI). هذه الواجهة تُسهل للطالب التحكم عن بعد والوصول إلى مختلف الأدوات والاجهزة الموجودة في المختبر والتعامل معها وكأنه في بيئة مختبر حقيقي. وهذا يخلص إلى ، مجموعات من الطلابِ قادرون على تَنسيق وتَطبيق التجربةِ في أي وقت كان ومِنْ أي مكان.