



uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES



FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES

Yuxiang Zhang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGRÉ

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Analysis and Improvement of Teaching Processes and Environments with Internet-related Tools

TITRE DE LA THÈSE / TITLE OF THESIS

Professor John C. Nash

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

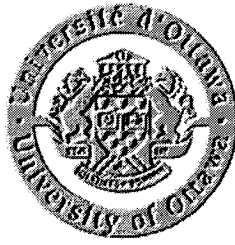
EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Professor Aline Germain-Rutherford

Professor Tony Quon

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies



**University of Ottawa
School of Management
Master's Program in Systems Science**

M.Sc. Thesis

**Analysis and improvement of teaching processes
and environments with Internet-related tools**

Student Name: Yuxiang Zhang

Thesis Supervisor: John C. Nash, D. Phil. (Oxon.)

School of Management

University of Ottawa

Date: May 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-18486-8
Our file *Notre référence*
ISBN: 978-0-494-18486-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

Nowadays, universities and colleges seek e-learning solutions to permit students to learn at different places and times from the usual classes, and e-teaching tools to assist teachers to handle their increasing numbers of students. Inexpensive and platform-independent tools are useful to professors who want to cope with their teaching without more work and with limited resources such as people, machinery, and Internet bandwidth. To improve work processes in the educational environment, particularly in teaching quantitative, problem-solving subjects, the development, dissemination and use of Internet-related tools would be welcomed.

Several Internet-related tools or projects are ongoing under Professor John C. Nash in the School of Management at the University of Ottawa. The E-tutor project to date has shown the merit of the concept of a "watch the student work" teaching method that can be (in part, at least) implemented using WWW-type tools running at low bandwidth with more than 100 students. These tools are, however, not yet organized perfectly. It is also important that their advantages and weaknesses be analyzed and placed in context with other solutions that are being developed and implemented.

Acknowledgements

I would like to thank Dr. John C. Nash, my thesis supervisor, for his time and energy, prompt and kind support, and good advice during the whole thesis work.

I would like to thank Professor Tony Quon and Professor Aline Germain-Rutherford, my thesis examiners, for their time, expert comments and suggestions, and helps.

Also, I would like to thank my parents, my older brother and my wife for their constant encouragement and support.

Table of Contents

1. INTRODUCTION	1
2.1 INTERNET, E-EDUCATION AND TEACHING TOOLS	2
2.2 DEMING MANAGEMENT METHOD ABOUT QUALITY CONTROL.....	4
2.2.1 <i>Introduction of Dr. Deming's Ideas</i>	4
2.2.2 <i>Application in the field of education</i>	7
2.2.3 <i>Biography of Dr. Deming and his success in Japan</i>	8
2.3 TEACHING-RELATED SOFTWARE SYSTEMS	9
2.3.1 <i>E-tutor</i>	10
2.3.2 <i>Moodle Course Management System</i>	10
2.3.3 <i>Manhattan Virtual Classroom</i>	13
2.3.4 <i>Claroline</i>	14
3. ANALYSIS OF TEACHING PROCESSES AND ENVIRONMENTS WITH INTERNET-RELATED TOOLS	17
3.1 ETPRL/ETPHP	17
3.1.1 <i>ETPRL</i>	18
3.1.2 <i>ETPHP</i>	21
3.1.3 <i>ETPRL and ETPHP</i>	21
3.2 PICKSEAT.....	22
3.3 ET-GRASED.....	23
3.4 FOTOHTML.....	24
3.5 CO4	25
3.6 E-TUTOR AND OTHER COURSE SYSTEMS	26
4. IMPROVEMENT OF TEACHING PROCESSES AND ENVIRONMENTS WITH INTERNET-RELATED TOOLS	28
4.1 IMPROVEMENT OF ETPRL	28
4.1.1 <i>Motivation</i>	28
4.1.2 <i>Procedure of the Improvement</i>	29
4.2 IMPROVEMENT OF ETPHP	34
4.2.1 <i>Motivation</i>	34
4.2.2 <i>Procedure of the Improvement</i>	34
4.3 CONTACT FORM SYSTEM	46
4.3.1 <i>Motivation</i>	46
4.3.2 <i>Procedure of implementation</i>	47
4.3.3 <i>New Approach</i>	57
4.4 BACKUP ISSUE	59
4.4.1 <i>Motivation</i>	59
4.4.2 <i>Procedure of implementation</i>	59
5. ASSESSMENT, RECOMMENDATION AND CONCLUSION	67
5.1 CONTACT FORM SYSTEM, CO4 AND PHPBB.....	67
5.2 DEMING'S IDEAS AND E-TUTOR PROJECT	69
5.3 MY CONTRIBUTIONS AND EVALUATION	71
5.4 OVERALL SCORE FOR DIFFERENT COMPONENTS	75
5.5 LIST OF MOST COMPUTER TECHNOLOGIES OR TOOLS INVOLVED	76
6. FUTURE WORK	78
7. BIBLIOGRAPHY AND INTERNET REFERENCES	79

8. APPENDIX (PROGRAMS)	85
7.1 PROGRAMS OF ETPRL.....	85
7.1.A <i>Test.pl</i>	85
7.1.B <i>achprpw.pl</i>	85
7.1.C <i>astorpw.pl</i>	89
7.1.D <i>pstorpw.pl</i>	91
7.1.E <i>sgetfile.pl</i>	93
7.1.E <i>aNewPsw.pl</i>	101
7.1.F <i>INSTAL.In.WinX.Notes.txt</i>	103
7.1.G <i>quicklogin.htm</i>	104
7.1.H <i>zz-abset.pl</i>	104
7.1.I <i>admin.lst</i>	115
7.2 PROGRAMS OF ETPHP.....	116
7.2.A <i>Test.php</i>	116
7.2.B <i>Setup.php</i>	116
7.2.C <i>etphpxt2.php</i>	120
7.2.D <i>extract2.php</i>	121
7.2.E <i>exlogin.php</i>	124
7.2.F <i>indexEx.php</i>	125
7.3 PROGRAMS OF CONTACT FROM SYSTEM.....	139
7.3.A <i>feedback.html</i>	139
7.3.B <i>EditMsg.php</i>	141
7.3.C <i>CmtIndex.php</i>	142
7.3.D <i>action.php</i>	145
7.3.E <i>handlefd.php</i>	147
7.3.F <i>Install.txt</i>	152
7.3.G <i>readme.txt</i>	153
7.3.H <i>SaveMsg.php</i>	154
7.3.I <i>ManageIB.php</i>	155

List of Figures

Figure 2-1: DEMING Management Method	7
Figure 2-2: Manhattan virtual classroom	14
Figure 2-3: Claroline e-learning platform	16
Figure 4-1: ETPHP – File Extractor Screenshot 1	42
Figure 4-2 ETPHP – File Extractor Screenshot 2	43
Figure 4-3: ETPHP - Exlogin - System View	44
Figure 4-4: ETPHP – External Login GUI	45
Figure 4-5: System View of the Contact Form System	47
Figure 4-6: Modules of the Contact Form System	48
Figure 4-7: Flow Chart of the Contact Form System	50
Figure 4-8: Contact Form System Graphic User Interface 1	51
Figure 4-9 Contact Form System Graphic User Interface 2	52
Figure 4-10: Contact Form System Graphic User Interface 3	53
Figure 4-11: Contact Form System Graphic User Interface 4	54
Figure 4-12: Flow Chart Of the Testing Of the Contact Form System	56
Figure 4-13: Graphic view of new approach - PhpBB	58
Figure 4-14: Backup by the "pull" way	61
Figure 4-15 Demo for the "pull" way	63
Figure 4-16: Backup by the "push" way	64
Figure 4-17 Demo for the "push" way	65
Figure 5-1: List of computer technologies and tools involved	77

1. Introduction

In the 21st century, the Internet is becoming a powerful method of communication. As it is global and growing rapidly, the Internet is making the world smaller and more interconnected. Information can be transmitted at a rather high speed. Internet technology now affects people's daily life and work. Every day, many people access the Internet to send emails, chat with friends, read news, search for information, and so on. In the field of education, the Internet has been found to greatly facilitate the process of teaching and learning (Rumble, 2002). With this new technology, it is possible for more individuals than ever before to access knowledge and to learn in new and different ways.

In recent years, many universities and colleges are facing educational challenges such as growing enrollments, shortages of qualified teachers, overcrowded and old buildings, and demands for higher educational standards, etc. In order to address these problems, many universities and colleges provide e-learning solutions based on Internet technology. These approaches overcome constraints of time and distance as they provide access to knowledge that was once difficult to obtain without dedicated and expensive staff. Basically, e-learning solutions not only allow students to learn in different places and times than their usual classes, but also provide students different ways to interact with other people to improve their learnings..

E-teaching tools are introduced to help professors with their teaching processes, or to maintain the quality of their teaching while the number of students is increasing. During the development and usage of e-teaching tools, students and professors who have limited resources in terms of machinery or Internet bandwidth usually welcome inexpensive and platform-independent tools.

In order to simplify the workflow, and to save time and energy for professors and students, it is necessary to analyze existing internet-related tools and to make some improvements. This is the goal of the proposed thesis.

2. Background

2.1 Internet, E-education and Teaching Tools

2.1.1 The Popular Internet

When the Internet was introduced as a small military experiment many years ago (Segaller, Stephen 1998), nobody knew it would develop so rapidly and would become one of the most popular forms of communication.

The Internet is becoming more popular than ever. It links people together via computers and telephone lines, or, in some cases, via wireless radio connections. Wherever users are, the Internet allows them to communicate with each other. Up to the year 2002, there were 655 million users around the world and the number continuously increased to 945 million in the year 2004 (CyberAtlas staff 2005). Why did the Internet become so popular?

Every day a great deal of timely, useful and informative news and knowledge are updated on different web sites, so people like to use the Internet more often than before. They read the news, communicate with friends or coworkers by checking their email, and they surf the Internet. More and more corporations or governments come to know the importance of the Internet. They often encourage communication between their employees by email, which not only facilitates their work, but also enhances the popularity of the Internet.

2.1.2 Challenges and Opportunities

In 2003, the final year of Ontario's high school program was eliminated. Grade 13 students graduated with those in Grade 12 at the same time and many of them wanted to continue to study in Ontario colleges and universities. According to the publications from the Ontario Universities' Application Centre (OUAC) in 2003, there were 101,668 applications to Ontario colleges and universities by the January 15 deadline. The number of applicants increased 46.7% from January 2002, and 69.9% from January 2001. This "double cohort" caused many other educational problems or challenges for universities and colleges such as shortages of qualified teachers, overcrowded and old buildings, and demands for higher educational standards, etc.

In order to meet these challenges and improve the productivity and effectiveness of teaching, many universities and colleges considered e-teaching as a solution, which is a new learning and teaching tool based on the Internet technology. The new solution can break the limitation of time and distance as it provides a new access method to knowledge that was once difficult to obtain. Basically, the e-teaching method permits students to learn at different places, times and paces from traditional classes. Students can learn in their own homes, whenever they want, and can customize the paces of their studies.

Although growing enrollments bring many challenges, colleges and universities obtain a great deal of business opportunities as well. From the view of the economy, when demand increases, supply increases correspondingly. The growing market eventually yields great profits to suppliers. In fact, online continuing education is a big business now and expected to be bigger in the future. In the new information economy, many organizations and companies need skilled workers to improve productivity and effectiveness. Many employees want to get continuous training to keep or do their jobs well. As they are gradually realizing the advantage of online continuing education, more and more organizations and companies are trying to use web-based training for their workers.

In conclusion, if colleges and universities can deal with and overcome these challenges successfully, they will gain a great deal of development opportunities in the era of the new information economy.

2.1.3 E-learning/teaching and assistant tools

As we know, instructors impart knowledge or skills to students through such ways as textbooks, lectures, magazines or even the Internet. In the field of e-education, there are two essential aspects: e-teaching and e-learning, which mainly refer to the process of teaching and learning by teachers or students separately.

It is very convenient for students that they can take courses without leaving the comfort of their homes and whenever they are free. They can save much travel time since they do not have to go to class according to special timetables.

It is also true that teachers can make their jobs more effective and productive by using the e-teaching/learning technology. For instance, instead of distributing class materials separately to each student, tutors can post them on one special web site so that students can access them whenever they need. Apart from course materials, professors can also post announcements, assignments, problems, solutions and the like on the web. Another advantage is that instructors can make their courses more interactive and interesting by easily adopting such collaborative tools as discussion forums.

Therefore, good web-based teaching tools enable tutors to perform such teaching related activities as distributing course documents, announcements, and assignments, and conducting group discussions on some issues or problems being taught. These teaching tools, just like electronic teacher assistants, will certainly help tutors to enhance their teaching quality.

2.2 DEMING Management Method about quality control

2.2.1 Introduction of Dr. Deming's Ideas

In Dr. Deming's book, *Out of the Crisis* (1986), he presented fourteen famous admonitions, which serve as management guidelines. By knowing and implementing these maxims, organizations may benefit from a more efficient workplace, greater profits, and increased productivity. The following are "Fourteen Points" and "Seven Deadly Diseases".

Fourteen Points: (Deming 1986)

1. "Create constancy of purpose for improvement of product and service". In other words, "create and communicate to all employees a statement of aims and purposes of the company".

2. "Adopt the new philosophy". It means, "Adapt to the new philosophy of the day because industries and economics are always changing".
3. "Cease dependence on mass inspection" or "build quality into a product throughout production".
4. "End the practice of awarding business on the basis of price tag alone". Instead, "try a long-term relationship based on established loyalty and trust".
5. "Improve constantly and forever the system of production and service" or "work to constantly improve quality and productivity".
6. "Institute on-the-job training".
7. "Institute leadership" or "teach and institute leadership to improve all job functions".
8. "Drive out fear in the work place; create trust."
9. "Break down barriers between staff areas" or "strive to reduce intradepartmental conflicts".
10. "Eliminate slogans, exhortations, and targets for the workforce" or "eliminate exhortations for the work force; instead, focus on the system and morale".
11. "Eliminate numerical quotas" or "(a) Eliminate work standard quotas for production. Substitute leadership methods for improvement; (b) Avoid numerical goals, Alternatively, learn the capabilities of processes, and how to improve them".
12. "Remove barriers that rob people of pride of workmanship".
13. "Institute a vigorous program of education and retraining" or "educate with self-improvement programs".
14. "Take action to accomplish the transformation" or "include everyone in the company to accomplish the transformation".

The Seven Deadly Diseases: (Deming 1986)

1. "Lack of constancy of purpose"
2. "Emphasis on short-term profits"
3. "Evaluation by performance, merit rating, or annual review of performance"
4. "Mobility of management"
5. "Running a company on visible figures alone"
6. "Excessive medical costs"

7. "Excessive costs of warranty, fueled by lawyers that work on contingency fee"

Some Obstacles: (Deming 1986)

1. "Neglect of long-range planning and transformation"
2. "The supposition that solving problems, automation, gadgets, and new machinery will transform industry"
3. "Search for examples (copying others without understanding can be hazardous)"
4. "Our problems are different (an excuse for resisting change)"
5. "Obsolescence in schools"
6. "Reliance on quality control departments"
7. "Blaming the workforce for problems"
8. "Quality by inspection"
9. "False starts (For process to change you must act on the change)"
10. "Meeting specifications (Meeting specifications doesn't address the goal of improved quality)"
11. "Inadequate testing of prototypes"
12. "Anyone that comes to try to help us must understand all about our business"

In Dr. Deming's other famous book, *The New Economics* (1994), he presented principles of profound knowledge. Firstly, he defines a system as "a network of interdependent components that work together to try to accomplish the aim of the system" (Deming 1994). Secondly, he states that a business organization is a man made system with an aim or purpose. To achieve its objective, the whole system needs to be optimized. According to his definition of a system, the network of interdependent components or workers needs to be optimized. It means managers should notice that communication and cooperation is more important than competition, that adversarial competition would not lead to the optimization of the system, or even would destroy the system in some cases.

Dr. Deming believes that each worker can reach nearly unlimited potential if placed in a suitable environment that provides adequate support, training, and sense of pride and responsibility. The majority of a worker's effectiveness is determined by his or her environment and only minimally by his or her own skills. To establish such an ideal

environment, a manager must have an appreciation for a system, knowledge of variation, theory of knowledge, and knowledge of psychology. As a result, workers understand their specific tasks and techniques as well as their higher values and are thus stimulated and empowered to perform better.

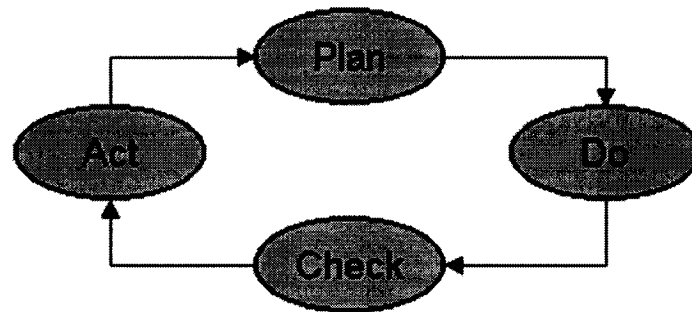


Figure 2-1: DEMING Management Method

2.2.2 Application in the field of education

While Dr. W. Edwards Deming's quality principles are widely spread around the industry field, people find it can also apply to the field of education. Some schools are accepting these ideas and are in the midst of a great transformation. The characteristic of education generally moves from effective to quality (Myron Tribus 2003).

In the 20th century, some elementary schools focus on how to help children prepare for the challenges of middle school in the classroom. However, after Deming's quality principles are introduced, schools would focus on assisting children to learn to set goals that are personal and lead to their success (Myron Tribus 2003).

After Deming's management concepts with those of the traditional school are compared, some great differences can be found. Considering his first point: "create a constancy of purpose", this constancy of purpose focused on the old effective school model rather than on the continuous quality improvement model. In addition, "Adopt the new philosophy" and "cease dependency on inspection" were missing. In the old system, schools promoted inspection-type systems for students and teachers. Students were evaluated by their curricula and end results such as standardized tests and examinations. They experienced a

direct teacher model: teachers directed the classroom. Teachers also suffered with measurable outcomes every year. Besides, schools did little to "break through the barriers" and "drive out fear in the work place", the process of decision-making was not clearly defined, and teachers needed to express their dissatisfaction with this model of leadership (Myron Tribus 2003).

How to successfully transform from a good effective school to a quality learning organization is critical. Firstly, some concepts need to be emphasized and accepted by schools. For example, individuals are valued, parents are partners, decision-making is shared, and teachers are professional educators and team members. Secondly, schools have to change the educational environment. The classroom should be a place where children get encouraged and assisted in obtaining skills and strategies for learning. Schools need to give up the old methods of grading students, which are based on curricula and "end results". However, developing alternative assessments may be a challenge. Thirdly, schools have to realize the importance of a parent's role. Parents should be respected as valuable customers, suppliers and critical partners rather than barriers. In the new system, the schools' decision-making process needs parents' activities. They are invited to open houses, weekly newsletters, teacher representatives, etc. (Myron Tribus 2003).

In conclusion, after moving from an effective-oriented to a quality-oriented school, teachers' approval rate to their schools greatly increases, parents become more involved than ever in their children's education, and children get higher scores in standardized tests (Myron Tribus 2003).

2.2.3 Biography of Dr. Deming and his success in Japan

Mr. William Edwards Deming was born on October 14, 1900 in Sioux City, Iowa, United States. In 1921, He graduated from the University of Wyoming in Cheyenne. Then he obtained a master's degree from the University of Colorado in mathematics and physics.

Later, he earned his PhD degree in Physics at Yale University. In 1927, Dr. Deming was introduced to a statistician, Walter A. Shewhart, who had already developed methods to introduce processes into “statistical control”. As these two great people continued to study together, Shewhart’s quality control theories became the foundation of the famous Deming management method theories. Dr. Deming taught graduate courses in quality control and sampling at New York University (Mary Walton 2003).

As his ideas were not completely recognized in the United States, during the late forties, and early fifties, Dr. Deming started to teach quality control to the Japanese. Slowly the influence of his techniques was adopted, and by 1951 Japan established the Deming Prize, which is a medal for improved quality to be awarded to scientists for theoretical improvements in statistical theory and to companies for the application of statistical control techniques. His tactics continued to impact Japanese productivity and quality well into the sixties and seventies. During those years, the Japan economy grew much faster than most countries in the world, and even today Japan keeps its technological leadership in some fields such as automobiles, electronics, shipbuilding, etc. (Mary Walton 2003).

When United States companies found that they were losing market share to Japanese companies, Deming’s technologies, to improve efficiency and quality while reducing costs, were gradually accepted in his home country. His applications of statistical quality control, “fourteen points”, and “seven deadly diseases” became famous and are still acknowledged and adapted in industry in the United States (Mary Walton 2003).

2.3 Teaching-related software systems

The purpose of this section is to provide readers the background knowledge by introducing some selected teaching-related software systems, so that readers can get some primary common views of these systems used in the e-education field. Actually, a way to get common views or new ideas, which is used later to make some analysis and improvements of e-tutor, is one of the important goals of this thesis.

As we may know, in the field of e-education, many good e-teaching tools are introduced to assist professors with their teaching processes, to keep the quality of their teaching

while the number of their students is increasing. During the development and usage of e-teaching tools, students and professors who have limited resources such as machinery or Internet bandwidth usually welcome inexpensive and platform-independent tools. The E-tutor system, the Moodle Learning Management System, the Manhattan Virtual Classroom, and the Claroline system are excellent examples to meet this requirement. All of them are free, open source software packages designed using sound pedagogical principles and have been in use for many years. Another kind of teaching related tool, such as a spreadsheet program, has been widely used by professors and students because of its user-friendly interface and powerful functions. In the following section, most of them will be introduced.

2.3.1 E-tutor

E-tutor is a suite of open source (GPL) online course tools. These tools were designed and developed under Professor John C. Nash in the School of Management at the University of Ottawa. The E-tutor project to date has shown the merit of the concept of a "watch the student work" teaching method that can be, in part or at least, implemented using WWW-type tools running at low bandwidth with more than 100 students.

In this project, there are some sub-projects:

1. File uploading for homework submission and retrieval for marking (ETPRL/ETPHP)
2. Choosing a "volunteer" (PICKSEAT)
3. Image grab and send (ET-GRASED)
4. Image consolidation and captioning (FOTOHTML)
5. Collaborative web forum tools for sharing ideas and information (CO4)

This is the target project that will be improved in this paper. Please refer to Chapter 3 for more detailed information.

2.3.2 Moodle Course Management System

Martin Dougiamas has developed the famous course management system, Moodle. Up to September 2005, the Moodle course management system has been in use for three years

since Moodle 1.0 was released on August 20th, 2002. Besides, the Moodle website claims this system can scale from a single-teacher site to a 40,000-student University (Martin Dougiamas – 2005).

The system is fully suitable for online education as well as supplementing face-to-face learning. Some interesting characteristics of the system are listed as follows (Martin Dougiamas – 2005):

- Supporting many interface languages.
- Easy to install on almost any platform that supports PHP
- Full database abstraction supports all major brands of database
- Emphasis on strong security throughout the web, such as all forms are checked, data validated, cookies encrypted, etc.

This system is structured by three sub systems (Martin Dougiamas – 2005).

1. Site management

In this sub system, we can define one admin user. To suit local needs, the site administrator can customize site colors, fonts, layout, etc. by using different themes. In addition, there are plug-in language packages, which can translate the whole site into many different languages. To meet some special local needs, it is possible to easily modify the code of the site because the source code is written in PHP language under a GPL license.

2. User management

The objective of this subsystem is to reduce management activities to a minimum, while meeting a high level of security. Both teachers and students could use this system to facilitate their teaching or learning.

Teachers can add, modify or erase students' names from the list of registered students. To increase security, teachers could set an enrollment password to their courses to keep out non-registered students.

Each registered student has only one account. In the online society, students are encouraged to make a profile including photos and brief descriptions. They can also decide whether or not to share their email addresses with others.

3. Course management

This sub system is mainly responsible for handling the course management. There are seven modules in this section.

The assignment module allows students to upload their assignments to the server machine and allows teachers to add particular feedback to each student's assignment page. Students can check the due date and requirements of their assignments and get notified by email when their assignments are marked. Teachers can be notified if late assignments are submitted.

The choice module provides users to set options about voting about some issues, and getting responses from other users.

The forum module provides users an online place to discuss some topics.

By using the journal module, a user can receive some open questions and teacher feedback.

The quiz module provides a question database to users. It supports different kinds of questions such as multiple-choice questions, short answer questions, and true or false questions. After students finish the quiz, the grade is automatically made to notify students and teachers.

In the resource module, students can access a great deal of useful resources about courses such as flash, video, sounds or documents.

In the survey module, teachers can analyze the quality of the online classes after students take the survey of courses at the end of the term. Students can be notified the level of their studies compared to the average level of the whole class.

2.3.3 Manhattan Virtual Classroom

The Manhattan virtual classroom is also a free, open source course management system that allows students to study their courses online. The Manhattan system is recognized as a robust education system and it has been used at the Western New England College since 1997. After several years of continuous development and improvement, today it is a reliable online education platform, which supports over 300 courses per term at colleges and universities throughout the world (Steven Narmontas 2003).

In addition, Manhattan was designed to run on the Linux operating system. According to the tradition of the Unix family system, Manhattan is completely free and is released under the GNU General Public License.

Manhattan virtual classroom has provided professors and students many functions:

- Communication between students and professors by a special e-mail system in the 'virtual classroom'.
- The composition of class syllabi, lectures notes and other handouts available to students on the web.
- Professors post some interactive quizzes for students to self-evaluate their course studies.
- Host live chat and discussion sessions for the whole class.
- Post assignments, receive work from students, and provide feedback from anywhere on the Internet.
- Easily distribute grades to students, while protecting their privacy.

The following graph will be shown after users log in the Manhattan virtual classroom.

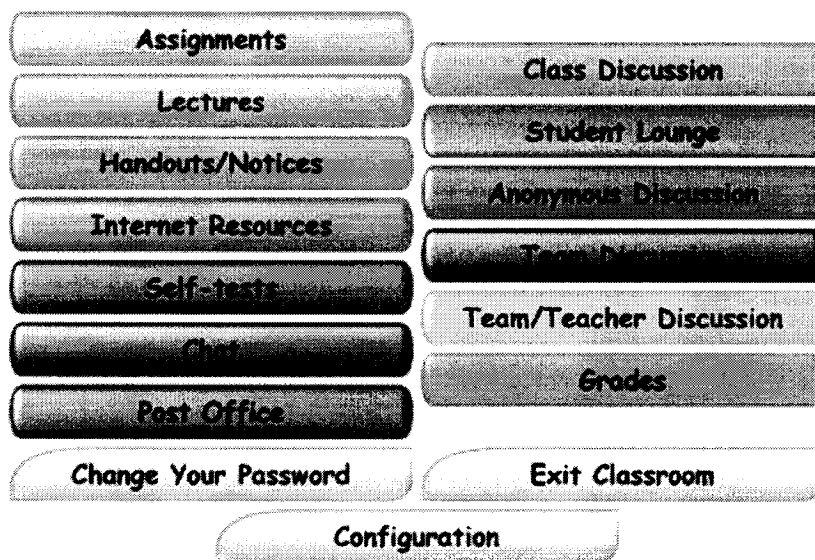


Figure 2-2: Manhattan virtual classroom

There are many available options for users. However, most options are similar to the modules in the Moodle Course management. To avoid the repeated description, this section will not introduce these options. In order to get some general ideas about the functionality of these options, please refer to the previous section: Moodle Learning Management System (section 2.3.2).

2.3.4 Claroline

As another popular course management system, Claroline has been developed as an example of an Open University and welcomed around the world.

“Some 10.000 people downloaded the last Claroline version (1.6) since its release, in April 2005. That means 100 downloads per day. Total download count of all the different Claroline versions reaches 78.000.... 451 organizations from 64 countries using or testing Claroline” (Thomas De Praetere, 2005, Claroline Home page).

Since activities involving teaching and learning depend much more on good social organization than on technologically sophisticated tools, the aim of Claroline is positioned to support good teaching and learning, not to replace it (Thomas De Praetere 2005).

Like the above two course management systems, Claroline provides some function modules for teachers and students. These modules can be activated or deactivated by administrators or professors. If modules are deactivated, students cannot use them. Some brief descriptions of these function modules are listed below: (Thomas De Praetere 2005)

- Agenda: list a course agenda with tasks and deadlines by tutors.
- Documents: publish documents in any format (Word, PDF, HTML, and Video) by tutors.
- Announcements: Make announcements (also via email) by tutors.
- Forums: students and teachers can share or exchange their view of courses
- Groups: create or manage student groups by tutors.
- Links: list course relevant links.
- Student papers: students can submit their assignments or share their files here
- Users: list names and emails of professors and students in the class.
- Exercise: students take some exercises composed by tutors.
- Course description: give the description of the course by tutors.

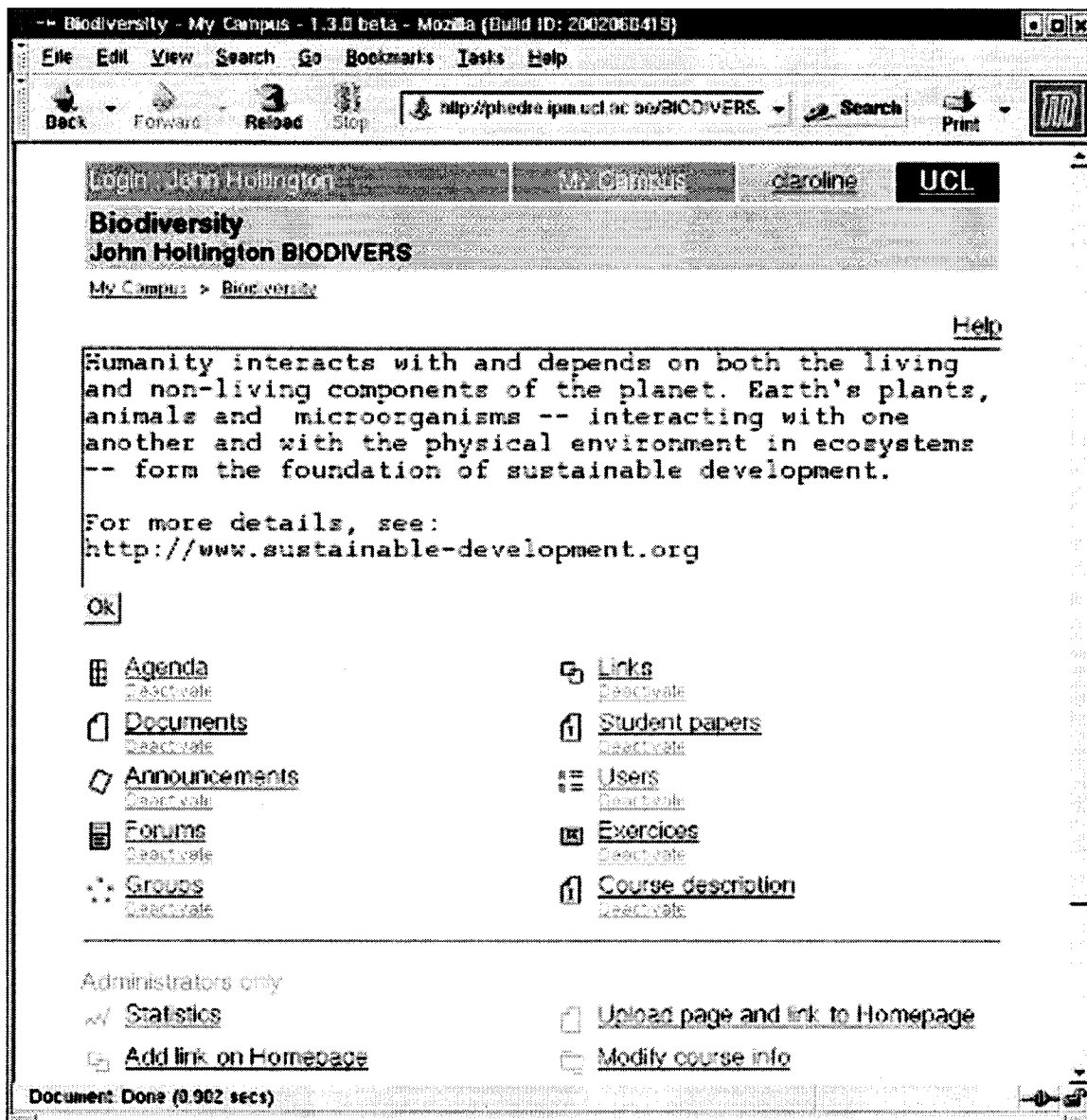


Figure 2-3: Claroline e-learning platform

3. Analysis of teaching processes and environments with Internet-related tools

The E-tutor project, which was intended as a phased development of an interactive tutorial support system with usable interim components, is directed by Professor John C. Nash, and the early software was developed by Mr. Pat Suwalski and Mr. Marcus Svilans. Despite being undergraduate students, these two people had considerable software experience, and the resulting programs indicate that they have made a number of good design decisions and implementation choices to enhance the quality of the E-tutor project.

This chapter mainly focuses on current existing tools to discuss and analyze teaching processes and environments.

The E-tutor project is a web-based tutoring system. Goals of this system are to prove the concept that a “watch the student work” teaching method can be in part implemented by using WWW-type tools which run at low bandwidth, yet still accommodate 100 students or more.

Until now, E-tutor system contains five sub-systems, including ETPRL/ETPHP, Pickseat, ET-Grased, Fotohtml, and CO4. Let us explore them one by one.

3.1 ETPRL/ETPHP

Goals of the subproject are to solve the following problems: (1) file uploading by students for homework submission, (2) downloading or retrieval by professors or teaching assistants, and (3) relevant software setup and class list input steps for either professors or administrators.

Up to now, two versions of homework submission tools have been developed. The first version of the tool was ETPRL, which is a tool to handle just one assignment or task, implemented as simple scripts in the Perl programming language. The second version of the tool was ETPHP, which is programmed by the PHP programming language and based on the MySQL database system.

Although both versions of the subproject are designed to run on the Linux operating system, some programming code and software testing is made under both the Microsoft Windows operating system and the Novell network. The reason is that it is easier to find better and more popular software to use in Windows than in Linux. We have to use the mature software to shorten the duration of the project development. Moreover, it is easy to transfer the Microsoft Windows version to the Linux version, because the programming language is recognized by both operating systems, and most code can be left unchanged. Next we further discuss and analyze these tools.

3.1.1 ETPRL

ETPRL is a small and handy tool. The main function is to upload files to the Internet. Students submit their homework in electronic format, while professors return students' homework without encountering paper. In this version, there are two structures:

Server-side: This structure is usually designed for administrator or professor or teaching assistant users, who have the right to access the server-side. Therefore, there are an `install.pl` script, an `index.html` file as the initial target of the browser, several Perl CGI scripts, and a data file named as `students.dat`, which contains the list of all students.

User-side: This structure is usually designed for the student users, who can use web browsers to submit their homework to the server-side. In each user-side, one cookie needs to be stored in user-side's machine, however, the problem is that cookies do not have to be generally enabled because all students may not have the same preference in options of their web browsers.

Here is the brief explanation of the scripts of ETPRL (John Nash, 2003):

Install.pl -- installation (needs root or similar privileges)

Some files in the path of Server Root /praname:e

index.html -- calls login.cgi and so forth

style.css -- use of style sheet avoids issues of screen size

uologo.jpg -- a logo to appear on the initial page

Some files in the path of CGIroot/projname:

getfile.cgi -- This script uploads the requested file

inc_login.cgi -- Checks is login cookie set. If not calls up login screen

login.cgi -- login script

prefs.cgi -- holds the directory names etc. for user machine & courses

showlogin.cgi -- script that displays login boxes

Some files in the path of CGIroot/projname/admin:

users.cgi -- allows user to add/delete/modify student list (students.dat)

download.cgi -- user can download student work file-by-file or as zipped archive

.htaccess -- access controls for apache httpd daemon

.htpasswd -- list of passwords for apache httpd daemon

retrieve.cgi -- downloads a single file (called by download.cgi)

Some files in the path of /var/projname:

students.dat -- tab delimited text file of student names and numbers

During the distribution and maintenance of the first version, some important problems were found. Solutions are needed because they actually affect the performance of the project.

Firstly, the code of ETPRL is not easy to be maintained as we thought. I came across two major problems. 1. There are only a few comments or documents for the source code, especially about installation script. Even with Professor Nash's help, I still spent much time for the installation of ETPRL. Regarding part of the original source code, sometimes it is really difficult to understand their meaning because of the lack of documentation. 2. The PERL programming language is new to me. Occasionally the language itself gives

me much trouble. Because one of the characteristics of the language is to have many unclear and shortcut ways to implement the same function, it may be easily written by an original programmer, but not be favorable to others. For example, before we modify or enhance some functions, we have to spend some time to understand the meaning of the source code. Even if we made the code by ourselves several weeks or months ago, it may still be difficult to recall it without good documentation. As a result, if we cannot understand the code, one straightforward way is to redo this code or function completely. Obviously that is not a good systematical method. Hence, the lack of documentation is the first thing we have to think about.

Secondly, this PERL programming language is sensitive to the directory structure of Linux. Because different users have their own preferences, the directory of Linux may be different. And the first version has some problems if the directory structure of Linux is changed. The problem causes some difficulties to the development and maintenance of the first version. However, if we want this version run on a certain server, we need to change some source code, mainly about the file folder, in this version somewhere. Of course, we can avoid this problem right now if we give some warnings and limits to users of this version.

Besides, ETPRL is awkward to adjust to multiple courses taken by a single student, multiple markers and sections, and other special conditions. Actually, all of these may happen in the University. Of course, a complex situation could be seen as a combination of several simple situations. During the development of an approach, we should be concerned about and figure out the simple one firstly, then we can consider the complex one based on the simple ones. According to the systematic method, this problem should be firstly simplified to a simple case, a single course taken by a single student, a single marker and a single section, etc. After the basic case is solved, we can go further to solve the complex problem gradually.

In conclusion, the first version of homework submission tools, ETPRL, has some problems involving programming, development, distribution, maintenance, etc. It is

necessary for us to make the upgrade, enhancement and optimization of this subproject. The new version should solve or avoid some of these problems.

3.1.2 ETPHP

ETPHP is an enhanced version of the above subproject, which is structured on the MySQL database technology, and using the PHP programming language. It also allows students to submit their homework and professors to manage students' homework. Comparing to ETPERL, ETPHP uses two new technologies, PHP and MySQL, which are widely used by many programmers in many countries. Around the world, many genius software developers keep working to make these two open source tools perfect. Therefore, we could spend less money and time to take care of problems of programming language and database management. In addition, because PHP and MySQL are open source and free of charge, it should be easy to distribute on campus because students usually only have limited funds.

ETPHP has been in use during some study terms in the University of Ottawa. Based on the security of MySQL and the powerful functions of PHP, it works very well and provides good security to users, including professors, teaching assistants, students and administrator. It has been exhibited on an exchange show in Carleton University. In conclusion, there are very limited spaces to upgrade this mature tool unless we get some new designs or new requirements.

3.1.3 ETPRL and ETPHP

Basically, both ETPRL and ETPHP provide the same functions for users. Students can use them to submit and get back their homework. TAs or professors can get students' homework, and send back to this system after they are marked.

Compared to ETPRL, ETPHP needs less time or work to set up for each "task", has a more integrated graphic user interface and uses the support of a database that permits simpler logins for users. In ETPRL, the administrator or professor users need to import or enter the student users' information for each section and each assignment or quiz. The

student users need to provide the login information for each “task” before they are able to upload their homework files.

However, the backup of ETPRL is easier to manage than ETPHP. With the support of the database technology, ETPHP puts all information in a single large “glob” of up to several hundred megabytes sometimes, which is vulnerable to the disk failure. However, ETPRL uses the simple files technology to store information, which allows us to use a flexible backup and data management strategy. For example, we can separate the whole data into small pieces and backup them separately to avoid some old data being copied again and again. Due to the limited resources we have, this kind of cost-effective strategies may be very welcome.

Therefore, for some courses that have multiple assignments and sections, ETPHP is preferred. However, for the courses that have a single assignment and section, ETPRL can work very well.

3.2 PICKSEAT

This objective of this subproject is to choose a student randomly in a classroom. For example, when a tutor raises a question for students, a random student may be chosen to answer. One advantage of this method is that we have the same probability of choosing each seat in a classroom. Then we can avoid favoritism or victimization.

The seat arrangement for each classroom is represented as a simple character graph, that is, every seat number in a classroom is found in a simple text file. If a simple character is selected, a specific seat can be mapped according to the corresponding relationship. If there is not a student sitting on the chosen seat, this tool will find a nonempty seat by some specific algorithm such as it will proceed to the right and then forward row by row until a student is found. If necessary, this process of choosing a volunteer student can be shown on the map of the classroom.

The functionality could be implemented more sufficiently in the MS-DOS system than in Microsoft Windows, since we just need some simple characters to represent classroom layout and results. Of course, if we have adequate requirements from students, we could make a Windows version, which will contain a graphic user interface to show the layout of the classroom, the process of selection, etc.

3.3 ET-GRASED

Students can use this tool to grab images and send them to their professor or classmate. This tool is constructed with two main parts: server program and client program. The server program is a CGI-BIN script in PHP for the Linux operating system, which mainly aims to store and display images. The client program is programmed in Microsoft Visual Basic, and it is intended to run in Microsoft Windows only mainly to grab and send images.

Allowing images to be sent around a classroom is very critical for effective communication within a classroom. As we all know, some particular situations can be hardly described by words, however, if relevant pictures are shown, then the situation can be clearly stated and understood by students.

Besides, when students are taking an online class, they may sit in their homes and face computers rather than other students and professors. That is very different from the traditional educational method. In this case, effective communication between professors and students mainly depends on the multimedia technology. If there are not voices and images to be presented during a class, it is very hard to make teaching processes successful through the Internet.

In conclusion, the image presentation is a necessary method for good communication between teachers and students, especially for online class. That is the primary motivation for designers who are concerned to develop the tool.

3.4 FOTOHTML

The software tool was programmed to make one or some HTML format files to show some captions and photos or images which users prefer to put together. Because this program is designed to run in the MS-DOS environment, there should be an input file to tell many user-defined parameters to the FOTOHTML program. If the input file is called *source.txt*, the program will use “source” as the root of destination file names. Actually, the functionality of input file here is similar to the one of a dialog window, just asking users to provide some necessary information.

After users provide all defined information, a set of HTML files are created if necessary. The program creates a new HTML page with a limit of 10 photo images. The rule of name files is as follows: the first file is *source.html*, from the second one, the files' names are numbered sequentially. For example, *source.html*, *source1.html*, *source2.html*, ... *source10.html*... and every html page has a “previous” and “next” button except for the first or last page.

The program defines every image as an 800*600 resolution. Therefore most images could be fit perfectly on the normal display screen. Images shrink or zoom automatically and may lose their actual sizes because of the specific resolution. However, that is the popular resolution when images are dealt within the HTML file, of course, the scaling could be changed in the source code.

The major advantage of this program is that it could check the existence of files. Because some pictures may be deleted or moved to other directories, missing or wrong image files are often reported after these HTML files are created. To deal with this situation, such errors are notified to users and recorded in the file *Missing.txt* in the second revision.

Until now, the program has the MS-DOS version, and it could run as a command-line program in the most popular Microsoft Windows operating systems. It sounds very good. However, one of main problems of this program cannot be avoided when it is used

widely. When users need to change some defined information in the input file, which was modified a long time ago, they need to refer to specific rules of input files. Unfortunately, rules may be a little bit sophisticated to most new users. Actually, that is a common troublesome problem of input files for most MS-DOS version programs. If a software program has GUI, it will be much easier to use for users. Therefore, the development of a version in Microsoft Windows could be one of future jobs if we have adequate requirements.

3.5 CO4

This tool performs like a collaborative web forum, which is programmed in the PERL/CGI language. The primary objective of this software is to share ideas and information between students and professors.

The security of this web forum is a little higher than usual ones in some ways, such as the procedure of login to the forum. A user needs to provide some necessary information including the name of the fileset owner and the fileset, user name and user password sequentially. If any one of these four parts is given wrong, the user will get an error message and not be allowed to login to the web forum. The user could use different names of filesets to login to different forums. For example, we can assign the name of fileset as mss-tech or mss-english. They are two different forums, which focus separately on the technology or English issue for the students who study in the master's program in Systems Science.

Like other normal forums, this system is also designed to provide another method to login, which only needs user's identity and password. This method is usually designed for guests or unregistered users with some limited access rights. For example, they can only read limited discussion topics, cannot post any messages, etc. However that functionality sometimes cannot work properly.

During the login procedure of the web forum, a cookie is created by the system to temporarily store the login information. To increase security, the cookie expires after a few minutes, therefore users will then have to log in again.

Setting up a class forum, which is a popular method to enhance the communication in the class, brings a great deal of benefit and conveniences for professors and students.

On the one hand, it provides effective communication within students, teaching groups and developers. It facilitates professors teaching process because the tool can save a great deal of time. They can avoid answering the same questions again and again by posting course FAQ, announcements and especial instructions, etc. to the forum. It is also good for students because they can get general information and answers to common questions before they begin to study. In addition, if most users post messages to the forum instead of sending email, they can avoid a great deal of email to burden their limited-size email boxes. Some special forums can be set up for different groups such as software developers, professors and teaching assistants because sometimes they need an exclusive place to share information and experiences.

On the other hand, forums can be a modest online place to share professors' and students' favorite collections such as beautiful photo albums, English dance files, etc. These interesting collections usually increase users' interests of the web forum, which enhance the timely and wide communication between users.

3.6 E-tutor and Other Course Systems

There are many other course systems, such as Moodle Course Management System, Manhattan Virtual Classroom, Claroline, and WebCT, etc. Why not just use one of them to replace some functionality of E-tutor or itself?

Basically, these course systems are much bigger and more complicated than E-tutor. That is the reason we have Section 2.3 to give an overview of them. Most of them not only

have the contact form and homework transfer functions in E-tutor, but also provide other powerful functions. For example, most of them have many interface languages, the internal forums and e-mail systems, etc. However, these functions may not be necessary or suitable to the School of Management at the University of Ottawa. And due to these extra functions in these systems, the difficulty of the installation and maintenance of these systems is increased, moreover, high-bandwidth connections and high performance computers may be required. Therefore, we prefer a customized tool, E-tutor, to meet the needs and wants in our situation.

Strictly speaking, this discussion is out of scope of the thesis. This thesis is intended to make some improvements of E-tutor by analyzing existing internet-related tools, not to study to use other tools instead. However, because these systems, such as WebCT, etc., are being used on campus, this issue has been raised during the thesis work.

4. Improvement of teaching processes and environments with Internet-related tools

As mentioned in the previous chapter, the goals of the E-Tutor system are to prove the concept that a “watch the student work” teaching method can be in part implemented by using WWW-type tools which run at low bandwidth, yet still accommodate 100 students or more.

Because of the double cohort, the focus of improvements is now on workflow and easing the clerical burden on professors and teaching assistants while possibly also simplifying students’ routine work.

Therefore, the first and biggest motivation of any improvement or implement below in this chapter is to try to simplify workflow and reach the goals of the E-Tutor system. To avoid mentioning this in each of the following improvement parts, generally this common point is not restated.

4.1 Improvement of ETPRL

4.1.1 Motivation

First of all, this tool or some functions in it may still be used in future. We need to upgrade this project by adding or modifying some functions.

Secondly, there are some obvious and potential problems after we use and test this tool. We need to fix them in order to release this project.

Thirdly, we also need to compose and upgrade some documents regarding the source code, installation and configuration of ETPRL. These documents are not only detailed notes of our work but also helpful guides or manuals for others who want to know or use

ETPRL. There are many tricks and tips during the installation and configuration in different operating systems. It is very worthy and helpful to document these.

4.1.2 Procedure of the Improvement

4.1.2.1 Installation and Testing

ETPRL has existed and been used before I came across it. Therefore, the first thing I need to do is to install and test it in order to make some improvements,

During the improvement, it is more convenient and easier for me to build a development environment in my own computer, a local machine. As I can test and develop the project without using other computers. Of course, the local machine is only for the development and the primary testing. We also need to use other computers and operating systems to make further tests after the primary test is passed.

Here are some procedures of how and what I installed in the local machine. In my computer, there was a windows operating system (Windows 98 or Windows 2000). Hence, what we need to install is an abyss web server (abssy1222), PERL language interpreter (ActivePerl-5.8.0.804-MSWin32-x86). After these applications are installed, we need to make sure the web server can work with the PERL interpreter by adding the following line in the configuration of ActivePerl.

```
cgiinterpreter "c:\perl\bin\perl.exe" cgi pl
```

If the location of perl.exe is different than c:\perl\bin, it has to be changed to a corresponding path.

Then, we could use any web browser to try this URL: <http://localhost/test.pl>, the file is listed in the Appendix 7.1. If it prints some information about system environment, the settings of the abyss web server are successful.

```
#!c:\perl\bin\perl.exe  
# filename: test.pl  
# Added by zyx
```

```

print "Content-type: text/plain\n\n";
foreach $var (sort(keys(%ENV))) {
    $val = $ENV{$var};
    $val =~ s|\n|\n|g;
    $val =~ s|\"|\"|g;
    print "${var}=\"${val}\"\\n";
}

```

After the abyss web server works well in the local machine, we need to install the ETPRL project. Firstly we have to copy all files to c:\Program Files\Abyss Web Server\cgi-bin\etpsetup or a corresponding directory. Then, we could get an install graphic user interface when we open the URL: <http://localhost/cgi-bin/etpsetup/etsuab.htm>. By clicking submit button, ETPRL would be installed in the local machine.

There are three kinds of login interfaces for different level users:

1. ETPRL login for the admin users: <http://127.0.0.1/cgi-bin/t/admin/alogin.pl>

After an admin user creates professor users

2. ETPRL login for the professor users: <http://127.0.0.1/cgi-bin/t/prof/plogin.pl>

After a professor user creates professor's tasks for the student users,

3. ETPRL login for the student users: http://127.0.0.1/cgi-bin/t/slogin.pl?profid=nash&taskname=mid_term or corresponding links

When this project has been uploaded to the Macnash server (Linux) and used by professors and students in the university, there are similar links for these 3 different level users.

1. ETPRL login for the admin users:

<http://macnash.admin.uottawa.ca/cgi-bin/etprl/admin/alogin.pl>

2. ETPRL login for the professor users

<http://macnash.admin.uottawa.ca/cgi-bin/etprl/prof/plogin.pl>

3. ETPRL login for the student users:

http://macnash.admin.uottawa.ca/cgi-bin/etprl/slogin.pl?profid=nash&taskname=mid_term or corresponding links

4.1.2.1 Upgrades

After we had installed and tested ETPRL for a while, we found some problems or issues that need to be solved or upgraded. Details of what I did are listed below.

1. Modified old functions to fix some problems and added a new function

a) Uploading problems

Sometimes, when students submit or upload their homework files, it takes a very long time to finish. Or sometimes, it just cannot work. After we had tested relevant software models, we found the system would slow down when the name of a file, which needs to be uploaded, contains some special characters or is empty. Although we could tell students to standardize their filenames, it is still good for us to fix this glitch, which might prevent some potential security problems. A relevant file is `sgetfile.pl`, which can be found in the Appendix 7.1.

b) Password changing problem

When admin or professor users change their own passwords, if they want to change to an empty password, or leave their passwords empty and click “OK” to change, their passwords will be changed to random characters generated by a computer. This may be a small and big problem. For developers, it is a small issue. We just need to add functions to verify the character before we submit to change the password. However, it might be a big issue to admin or professor users, as no new professors can be added because admin users can not login to maintain the system; no new assignments for students can be added because professor users do not know their passwords, etc.

Relevant files are `astorpw.pl`, `achprpw.pl`, and `pstorpw.pl`, which can be found in the Appendix 7.1.

c) Password retriever for admin users.

This new tool has been developed to retrieve passwords in case admin users do not know or forget their passwords. Password retriever resets their passwords and gives a default username and password. Actually one important reason we want to develop password retriever is to fix the “password changing problem” above. But after that problem has been fixed, we find it is still useful when admin users really forget or do not know their passwords. Of course, some security problems should be noticed when this script is used. For example, ones who use this tool would know the passwords of admin users. In Linux, we could make this script file and the password file readable and writable to nobody but admin users if we do not need to use this script. Only when admin users want to use this tool to get back their passwords, they could temporarily set these two files readable and writable to everyone (`chmod 666 aNewPsw.pl` and `chmod 666 adminpw`). After they get their new passwords, they should immediately set these two files back, only readable and writable to admin users (`chmod 600 aNewPsw.pl` and `chmod 600 adminpw`).

An affected file is `aNewPsw.pl`, which can be found in the Appendix 7.1.

The link on the Macnash server is

<http://macnash.admin.uottawa.ca/cgi-bin/etprl/admin/aNewPsw.pl>

Or in the local machine is

<http://127.0.0.1/cgi-bin/t/admin/aNewPsw.pl>

2. Local test and Remote test

After I finished modifying, debugging and testing of the models in ETPRL in the local machine (Windows Operating System), I have to upload ETPRL to the Macnash server (Linux Operating System) to run and test it again.

During the remote test, we also found some specific problems, mainly about the location or links of script files. We fixed them by replacing wrong links with right ones, and then tested the corresponding modules again.

In a word, this procedure went on very smoothly because a great deal of testing and problems had been fixed in the local machine.

3. Documentation

When I studied source code of this project, I added many comments in the program although some helpful comments existed from the original programmers. From my point of view, comments are very helpful, especially for software programmers. Sometimes, software programmers, even exceptional ones, may not understand source code, which were composed long time ago. Hence, I also prefer to put some comments to make the whole program easily understandable as I browsed these files. There are many files that are affected that could be found in the latest version. I do not list all of them in the Appendix 7.1.

Another kind of documentations of this project I composed is a guide of installation and usage of ETPRL.

Some relevant files are `INSTAL.In.WinX.Notes.txt`, and `quicklogin.html`, etc., which can be found in the Appendix 7.1.

4. Integration and backup

The last component is integration and backup. I have to put all modified or updated files together and backup different version during the period of development. This is such an important step, which makes sure our new work or effort would not be lost, and also goes with the latest version.

Some relevant files are `zz-abset.pl`, and `admin.lst`, etc., which can be found in the Appendix 7.1.

4.2 Improvement of ETPHP

4.2.1 Motivation

ETPHP is a well-designed, well-established, and fairly easy-to-use tool. Original developers, Dr. John C. Nash, Mr. Pat Suwalski and Mr. Marcus Svilans, have made a number of good design decisions and implementations. Thanks to their efforts and contributions, the ETPHP project has been in use successfully by many students and teachers for many study terms in the School of Management at the University of Ottawa. That means there is less improvement needed for the future. However, new requirements always come out with time passing by. When we plan to integrate this tool into other systems, we need to sort out the login module and make an interface to show others. We also need some documents or guides about how to set up or install the environment and use ETPHP for developers who have limited knowledge about Linux (like me at that time).

4.2.2 Procedure of the Improvement

4.2.2.1 Installation Problems and Example Steps

ETPHP has existed and been used before I came across it. Hence, the first thing I should do is to install and test it in order to make some improvements.

During the improvement of the program, it is more convenient and easier for me to build a development environment in my own computer, a local machine. As I can test and develop the project without using other computers. Of course, the local machine is only for the development and primary testing. We also need to use other computers and operating systems to make further tests after the primary test is passed.

The implementation of the development environment of ETPHP is much more complicated than that of ETPRL. Below I listed detailed procedures of how and what I installed in the local machine.

4.2.2.1.1 Installation of Xandros Operating System

ETPHP is designed to run in a Unix/Linux Operating System. Therefore, we first need to install a Unix/Linux OS. However, there are many different Unix or Linux versions in the world, and if developers are new to Unix/Linux, but are very familiar with the Windows operating system, like me at that time, what version of Unix or Linux are easy for them to use? At that time, Professor Nash suggested that I can install Xandros, which makes the transition from Windows to Linux easy. (Jim Lynch, PCMagazine.com, 2005)

As my experience indicates, the installation procedure is very easy. Here we go to start the installation.

1. Download the Xandros installation files (Desktop OS Open Circulation Edition) from <http://www.xandros.com/about/downloads.html>. I selected to download Open Circulation by BitTorrent without any charge rather than by HTTP, which costs US\$ 10. Then we may need to use one download tool, BitTorrent, which could be downloaded from the Xandros website, <http://www.xandros.com/torrent/BitTornado-0.3.7-w32install.exe>

The time of the whole procedure depends on computers, BitTorrent seeds, and the bandwidth of Internet. For instance, it took me around two days to complete the download.

2. After getting the installation file, we need to burn it to an installation CD.
3. After the installation CD is ready, half of the installation procedure is complete. The rest is very easy. Just go to the CMOS configuration of a computer, let the computer root from the CDROM, or let it read root information from the CDROM first. Then put the installation CD in the CDROM and restart the computer. Even if there are many different Operating Systems in the same computer, the smart installation program detects that, and prompts us to select one disk to install. Here, I suggest to reserve one partition of one hard disk for Xandros and backup important data somewhere in different hard disks and computers, or burn them to CDs. During the installation, the program shows several windows to tell us the progress; usually we do not need to change any settings during this period, just like the installation of a Windows Operating System.

4. After Xandros has been installed, we'd better go to Xandros Networks to upgrade the system. It is similar to "Windows Updates" in a Windows Operating System. This is also a time-consuming procedure, which took me around 5 hours to finish. But it is still worthy because the system becomes stronger, much more effective, and has more functions than ever before.

4.2.2.1.2 Installation of XAMPP

Before I became familiar with XAMPP to set up the environment of ETPHP, I found that it was not easy to install the Apache web server, MySQL, PHP and PERL, etc. Besides, it was even more difficult to make them work together in Unix/Linux. According to my experience, many installation trouble or difficulties are common to users in Linux/Linux, and troubleshooting is time-consuming and costly.

I am not alone. My professor, Dr. John C. Nash, has had many problems with the installations as well. The first sentence of the homepage of XAMPP is: "Many people know from their own experience that it is not easy to install an Apache web server and it gets harder if you want to add MySQL, PHP and PERL."

There are similar experiences in three of eleven reviews for the Apache Friends XAMPP (2004~2005): "... I had very little experience of apache, mysql and php and needed an easy way to test PHPNuke systems locally ... I tried manual setup of all the various packages necessary and failed miserably...", and "... I have tried for the last 3 weeks to configure Apache, PHP, and MySQL on a Windows XP system without success. There are many tutorials out there but is very difficult to do UNTIL I found XAMPP....", and "In December, 2004, I was a complete novice in all aspects of web servers. I attempted and failed to install apache, php, perl from their sources"

And the procedure of troubleshooting is time-consuming. Here are Results of "Survey: Moving From Windows To Linux Isn't Cheap" (2004), "... come from 1,000 IT administrators and C-level executives worldwide. Of the 300 large companies--those with

10,000 or more users--that it surveyed, 90% indicated that a significant or total switch from Windows to Linux would be expensive, complex, and time-consuming ..."

Even though I followed some installation guides step by step. I always came across problems or errors. Sometimes, I needed to install one component before the Apache installation. Sometimes, I needed to change setting files to make the installation successful. Sometimes, the same errors or problems persisted to show up even if I read the whole installation guide, and did whatever the guide said; the installation program just says it cannot finish the installation properly. I have spent a great deal of time, around two weeks or so, to try to finish the setting of the environment (Apache+MySQL+PHP) in Linux. That is really a terrible and despairing experience. In contrast, the similar installation process in Windows usually only takes me around one hour or so.

But is there a good installation program to be used in Unix/Linux as easily as the one in Windows? Of course, the answer is yes. And it gives me an idea, if I could read or find corresponding articles or solutions right after difficulties come out, I would have saved so much precious time, two weeks or so. Hence, knowing where and how to find solutions, or sharing others' experiences is a clever, practical and effective way to save our time and stamina. This is a good lesson for me and other similar developers.

During the period of seeking to solve this installation problem, I have asked many other experienced Unix/Linux users or developers for their advice. Finally, I have found a wonderful, user-friendly program to complete the setting of the environment. It is XAMPP, which is very suitable for users who want to set up a development environment (Apache+MySQL+PHP) in Linux as soon as possible, or for whoever is new to Unix/Linux. Right now I strongly suggest using this program to finish this difficult part very quickly. If we have time or interest about it in the future, we could go back to install Apache, MySQL, PHP and PERL one by one. By this method, we could save a great deal of time, energy, and avoid wasting many futile efforts we will make.

Here are some procedures to set up the development environment of ETPHP.

1. Download the installation files from <http://www.apachefriends.org/en/xampp-linux.html#374>. On my computer, I use XAMPP Linux 1.4.14, which contains Apache 2.0.54, MySQL 4.1.12, PHP 5.0.4, phpMyAdmin 2.6.2-pl1, and other components. This version is fine for ETPHP.

2. Installation

Firstly we could login to Xandros Operating System as a system administrator root. Then extract the downloaded archive file to /opt by the following command.

```
tar xvfz xampp-linux-1.4.14.tar.gz -C /opt
```

Then XAMPP should be installed in the /opt/lampp directory. We could check by Xandros File Manager.

3. Start to run this environment

After we simply type the following command in a shell window: "/opt/lampp/lampp start", then the XAMPP service starts, which means Apache, MySQL, PHP, PERL, etc. are running now and ready to be used.

4. Testing

The last thing is to check that everything really works. We could just try the following URL at our favourite web browser: <http://localhost>. If we get the welcome page of XAMPP, then it means everything is working and the installation is successful.

This is an official way to conduct the testing. We could also do it by our own way:

1. Make a test.php, which only contains one following line

```
<? phpinfo() ?>
```
2. Copy test.php to /opt/lampp/htdocs/
3. Try <http://localhost/test.php> in a web browser. If it shows some information about the Apache web server and PHP parameters, we have completed the installation successfully.

This is a general way, which is suitable to any web server that runs in various Operating Systems.

5. Setting about the security

To make our XXAMP more secure, we could simply run the following command:

```
/opt/lampp/lampp security
```

After the above five steps are finished, the environment for the development of ETPHP is ready, and we could go to the next step to install ETPHP. We could get more information about the XXAMP from this link, <http://www.apachefriends.org/en/xampp-linux.html>.

4.2.2.1.3 Installation of ETPHP

After we set up Xandros and XAMPP properly, we can start to install the ETPHP project.

There are a few notes to be noticed when we perform the installation.

1. We could get the ETPHP installation file from the link:

<http://prdownloads.sourceforge.net/etutor/etphp-1.0alpha.tar.gz?download>. After getting the zipped file, we need to unzip it and copy files to an appropriate directory. Suppose we are not familiar with corresponding Unix/Linux commands (tar, cp, etc.), we could make it by the Xandros File Manager, just right click the zipped file and select unzip from the popup menu, then copy the unzipped files to `/opt/lampp/htdocs/et`

2. We need to login as an administrator user. Then we could get a setup form of the ETPHP program via <http://localhost/et/setup.php> in a web browser. In the installation form, an admin password of MySQL needs to be provided and the default user name and password of admin is the same: "root". We could use default variables if we changed nothing when the XAMPP has been installed.
3. If in one computer we want to reinstall ETPHP or it has been installed before, we need to drop the "etutor" database via phpmyadmin, <http://localhost/phpmyadmin>. Just make sure the "etutor" database is useless, otherwise, we may need to rename

and backup that. After the setup form has been filled up, the installation program should configure the ETPHP program by itself.

4.2.2.1.3 Use and Test ETPHP

After the ETPHP system is installed successfully, we could login as an administrator to use and test it. According to the documentation of ETPHP, here are some standard procedures:

- Administrator logs in
- Administrator sets system-wide preferences
- Administrator creates the professor users
- The professor users log in
- The professor users add their courses
- The professor users add their students to these courses, including TAs, teaching assistants
- The professor users edit the TA users to give them TA rights
- The professor users create assignments for their courses
- The student users log in
- The student users upload assignments
- The TA or professor users log in and download assignments
- The student users download the marked assignments

Please check README file for more detailed information in the latest version.

4.2.2.2 Upgrades

After we had installed and tested ETPHP for a while, we found some problems or issues, which need to be solved or upgraded. Details of what I did are listed below.

1. Setup problems

The old version worked pretty well before, but when MySQL comes to version 4.0 or later, some functions, used in ETPHP, have been removed and replaced by

new functions. When I tried to set up ETPHP, the installation program did not work well in MySQL 4.0+. After tracing, I found one obsolete function, which needs to be replaced. The old function, `mysql_create_db`, is to create a database. However, in the new version, MySQL has stopped supporting this function and suggests `mysql_query ("CREATE DATABASE new_basebase")` instead. The affected file is `setup.php`, which can be found in the Appendix 7.2.

2. File Extractor

File Extractor, which has been in use as an assistant tool with ETPHP, is directed and programmed by Professor John C. Nash. Before I started to use this tool, the main functions work very well, including GUI, backup, etc, but there was not helpful information for users.

Basically, we have used File Extractor to backup data in the ETPHP system regularly, especially about the assignments data students submit.

The backup of these data usually has been done very often to prepare for some accidents. However, every time before we run the backup tool, we need to know two critical parameters, “Starting Assmt ID” and “Ending Assmt ID”, which can be gotten by another tool (`phpmyadmin`). Of course, it is easy for us to get these parameters by File Extractor itself.

Hence, that’s a point to be upgraded, and a new function is designed to show helpful information only when it is required. When any of critical parameters given by users is wrong, the tool will show a helpful table, which contains these two parameters.

Relevant files are `etphpxt2.php` and `extract2.php`, which can be found in the Appendix 7.2. Here are some screenshots of File Extractor.

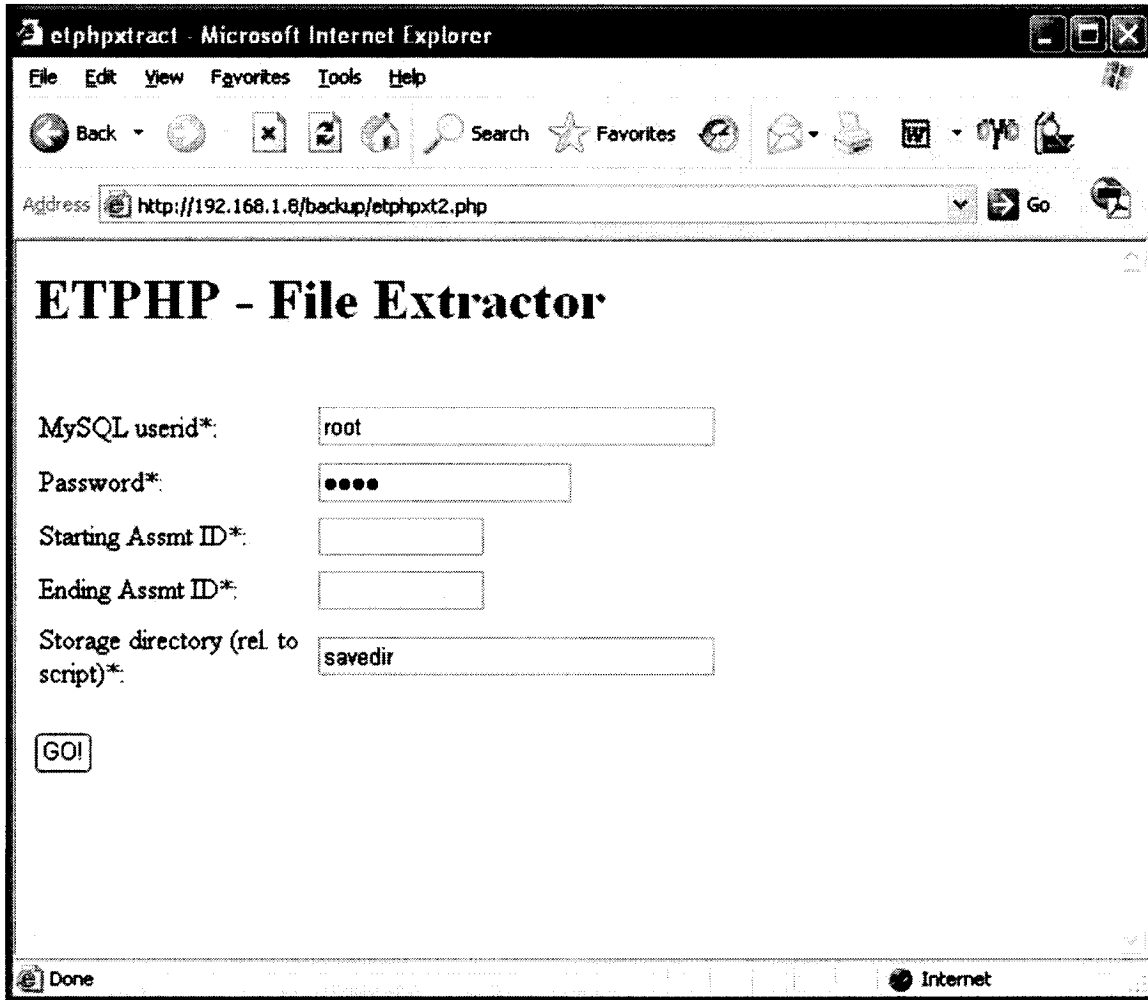


Figure 4-1: ETPHP – File Extractor Screenshot 1

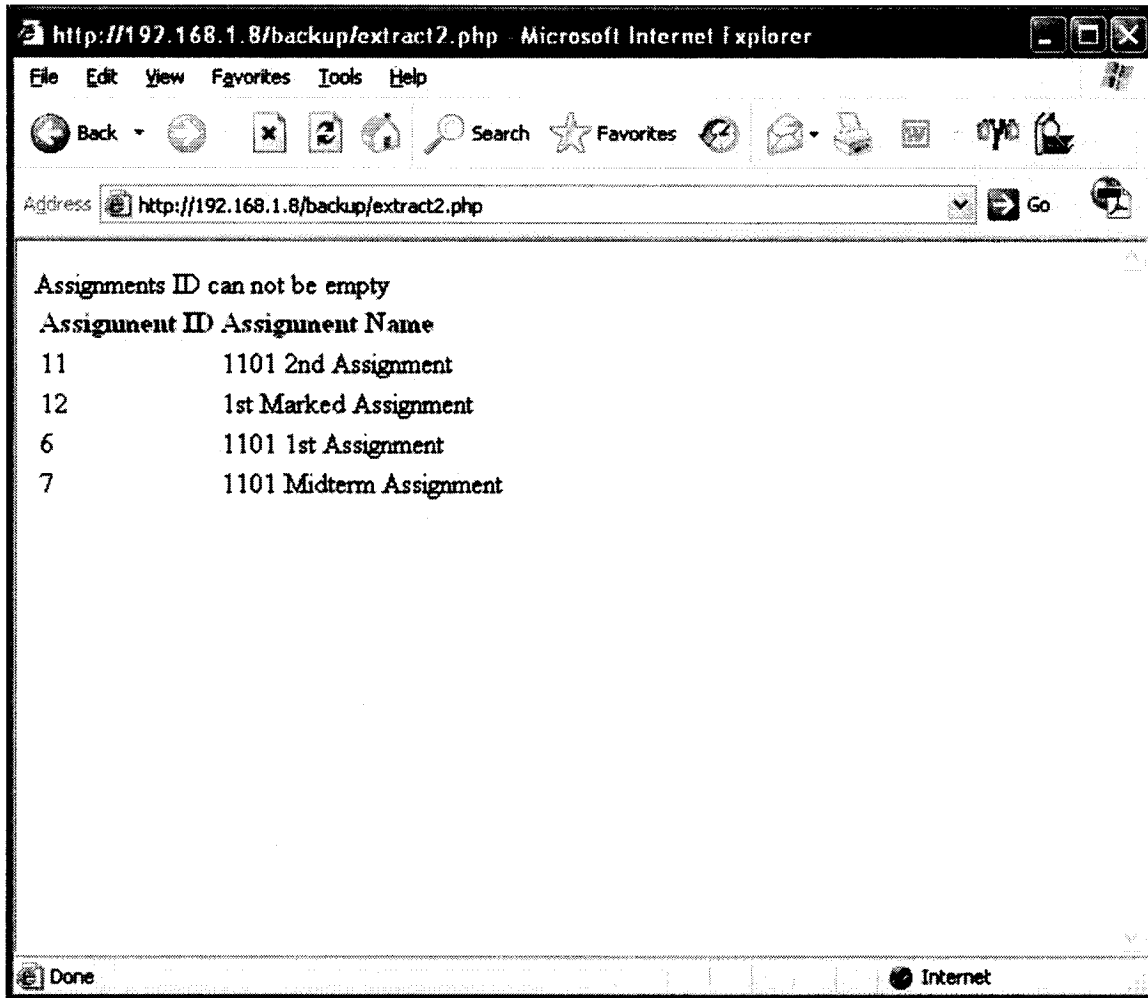


Figure 4-2 ETPHP – File Extractor Screenshot 2

3. External login

In the future, we probably need to integrate ETPHP into other systems. Therefore, we need to develop a graphic user interface or function interface to show the login module could be separated. This is just a demo and if we really need to integrate ETPHP into other systems, we still have a long way to go, to modify some other relevant program code, such as the way to access the database and changing the graphic user interface to match a new system.

Besides, we expect PHP and HTML would keep introducing new versions. Some functions or characters, which worked well in the old version, may not be supported by the newer version. During the test, we found some login problems. For example, a PHP script file cannot send parameters to itself, which did not exist, in the old version.

Here is a diagram to show how it works.

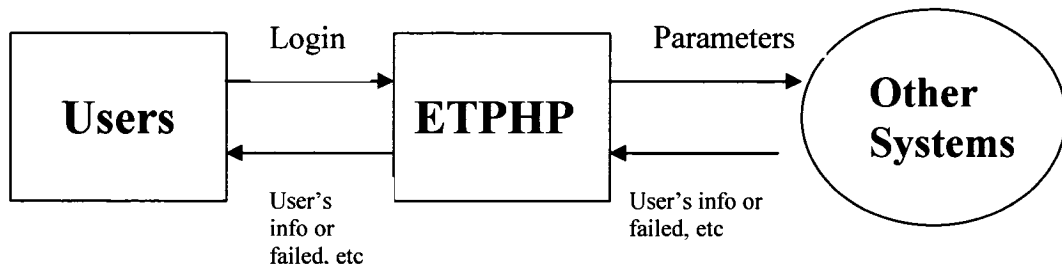


Figure 4-3: ETPHP - Exlogin - System View

When a user wants to login to ETPHP, we will automatically redirect it to one specific system. The user has to provide a username and password to log in that system. After a while, the system will tell ETPHP whether the user logs in successfully or not. If the login is successful, we will continue to get other information about this user such as student number, valid course information, section or other things we need.

The most important thing is ETPHP will not store any information by itself in its own system. All these data will be in other systems, which will be responsible for the security of them. One more thing we need to care about is the security of the transaction between ETPHP and other systems. We could use some technologies to encrypt information, such as SSL, etc.

And this is just a proposal, still in the phase of estimation and design. Of course, there will be a great deal of things to be done after the system, with which ETPHP

integrates, has been specialized. At that time, most of the work should be about systems' interface, database access, and graphic user interface. Here we only make a very simple demo. Corresponding files are exlogin.php and indexEx.php, which can be found in the Appendix 7.2.

Here is a screenshot of the demo we made.

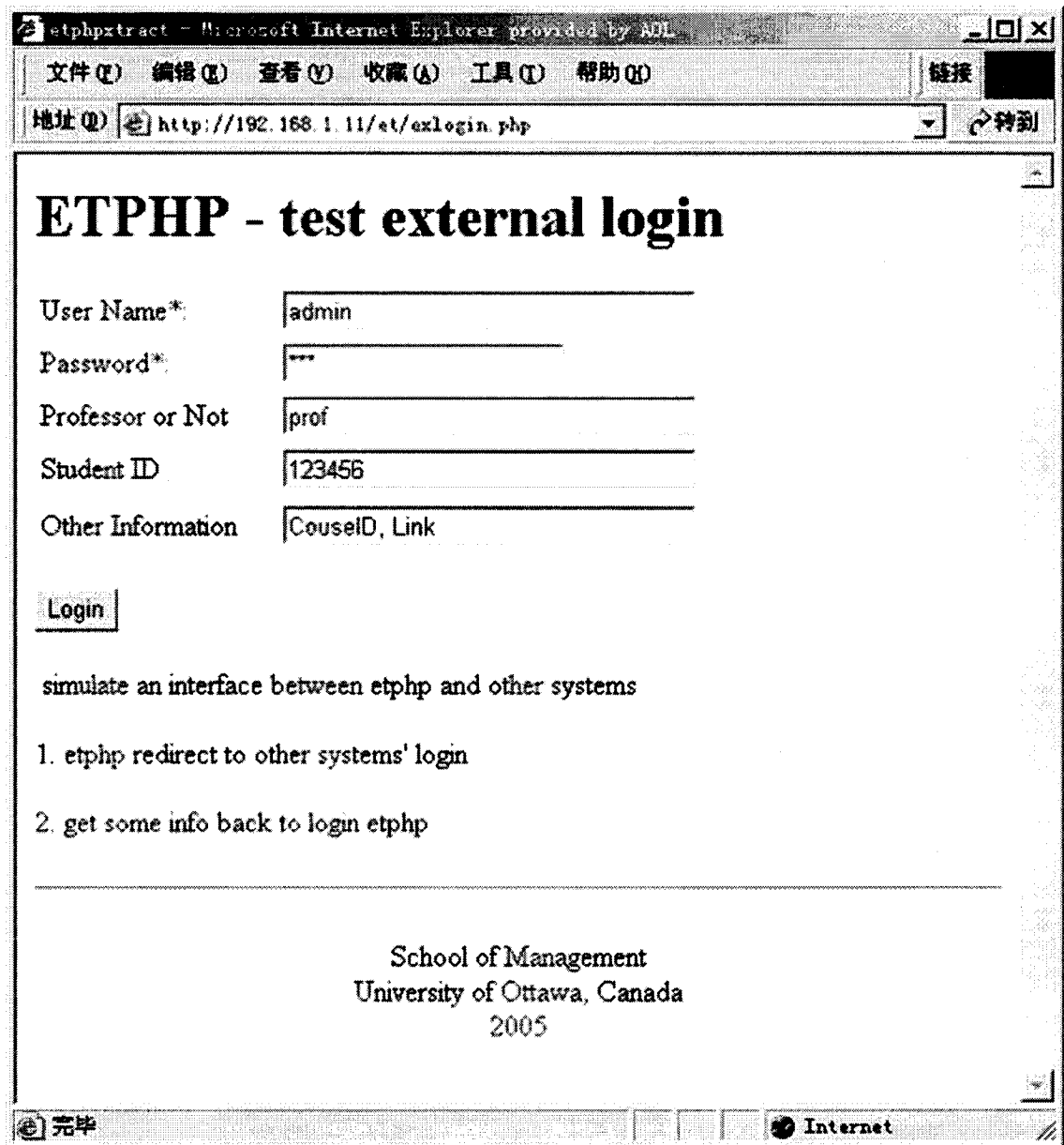


Figure 4-4: ETPHP – External Login GUI

4.3 Contact Form System

4.3.1 Motivation

When students have some questions or concerns, they could send emails directly to staff, such as professors or teaching assistants, but that's not a good way for staff to manage.

Sometimes staff wishes not to get emails directly for following reasons:

1. There may be several teaching assistants who could answer students' questions. However, students may collect all of their email addresses and send the same questions to all of them or only some of them who happen to be away or could not response for a while. Hence, it possibly causes these troubles: the time of multiple staff are required, students request are not responded quickly, bandwidth are wasted.
2. Some students could not clearly and briefly state their questions. They may put much irrelevant information in their questions, which takes email recipients longer time to get their points. It is better to give one form to let them fill out and help them to express clearly and briefly.
3. Staff wishes students can share or get answers from others. They hope to list answers to their frequent ask questions (FAQ) to reduce repetition of questions

Therefore, we need a contact form system to deal with these problems. Students could send their questions or concerns to professors or teaching assistants by a standard form. And staff could attach their comments, reply and manage these messages. Besides, the staff's email addresses are not exposed, which avoid some potential junk emails. What's more, professors and teaching assistants could access or share messages from students to know what they want and need.

In all, this system could effectively ease the clerical burden on professors and teaching assistants while possibly also simplifying students' workflow. Meanwhile, it could be helpful to show the goals of the E-tutor system, which is to prove the concept that a

“watch the student work” teaching method can be in part implemented by using WWW-type tools which run at low bandwidth, yet still accommodate 100 students or more.

4.3.2 Procedure of implementation

According to software engineering, there are several steps in a procedure for the development of a new software system, including design, programming, optimization, test and maintenance. We give a brief description of them in different parts.

4.3.2.1 Analysis and Design

This first step is to analyze this contact form system. Actually, we have described some details in the “Motivation” section (4.3.1). According to systems science, it is very crucial to get components of a system and their relationship. However, there are often some sub systems in one system. If we only consider users of the contact form system as a system, we will get the following diagram.

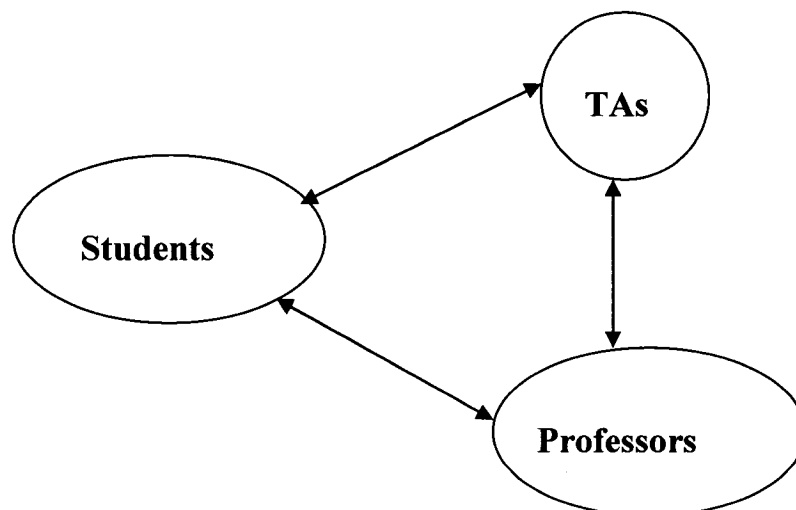


Figure 4-5: System View of the Contact Form System

Then, we have to know what customers or clients want and need, and to specify what functions we should have. For example, students want to send messages to professors or teaching assistants. Professors or teaching assistants need to manage messages, such as attaching their comments, and moving messages to different folders. Therefore, there are at least 3 functions for them. Here is a diagram to describe them.

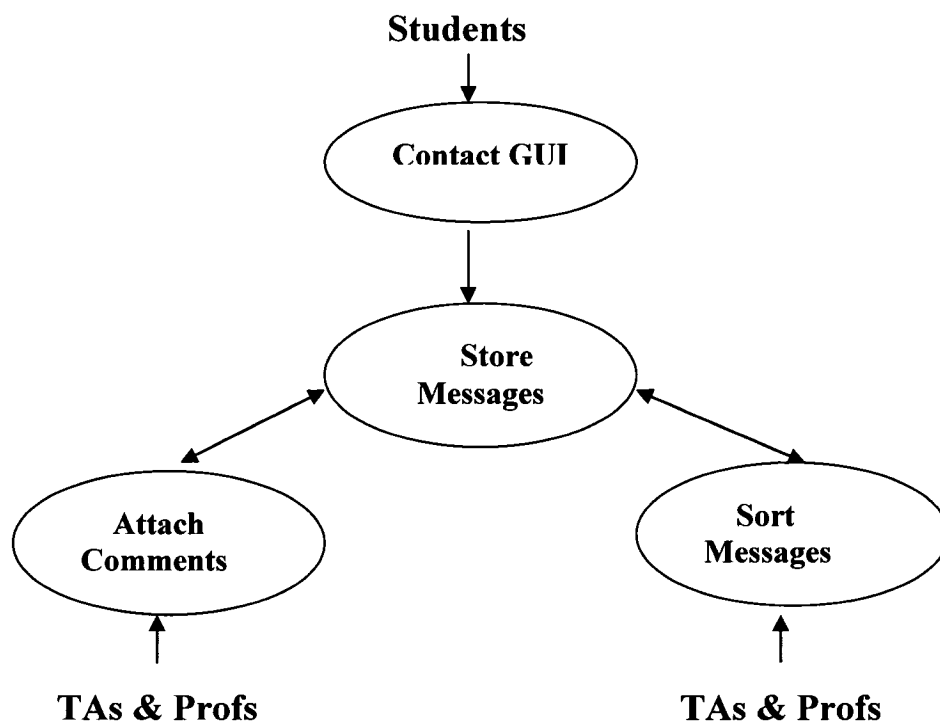


Figure 4-6: Modules of the Contact Form System

4.3.2.2 Programming

To finish these functions, the knowledge of some programming languages such as PHP and HTML is necessary in our situation.

- **Contact GUI and Store Message**

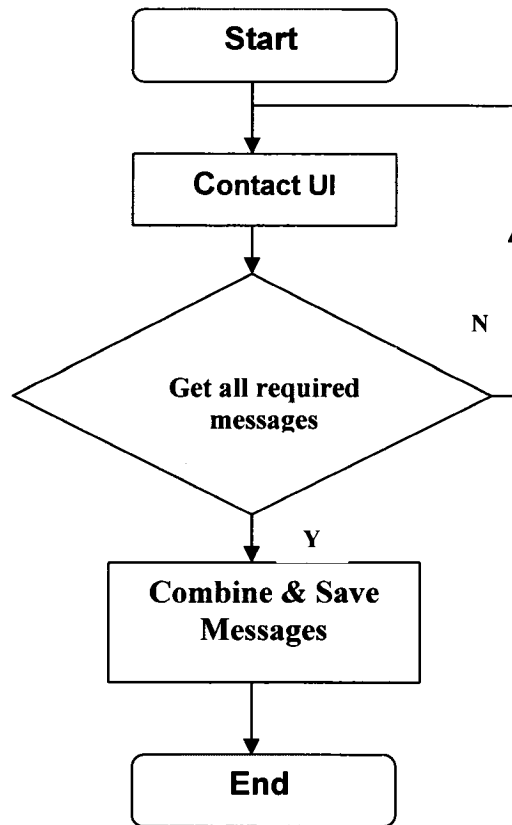
This form is used for the student users, and it works not only as a messenger but also helps students to clarify and shorten their questions.

The interesting thing is the background colour of this form. The original colour is grey. But when it is used in the doc-depot server, the original grey background colour is mixed with the one provided by the doc-depot server. And it comes to another colour, which looks pretty good.

Relevant files are `feedback.html` and `handlefd.php`, which can be found in the Appendix 7.3.

Here are a flow chart and a screenshot of the contact GUI.

Figure 4-7: Flow Chart of the Contact Form System



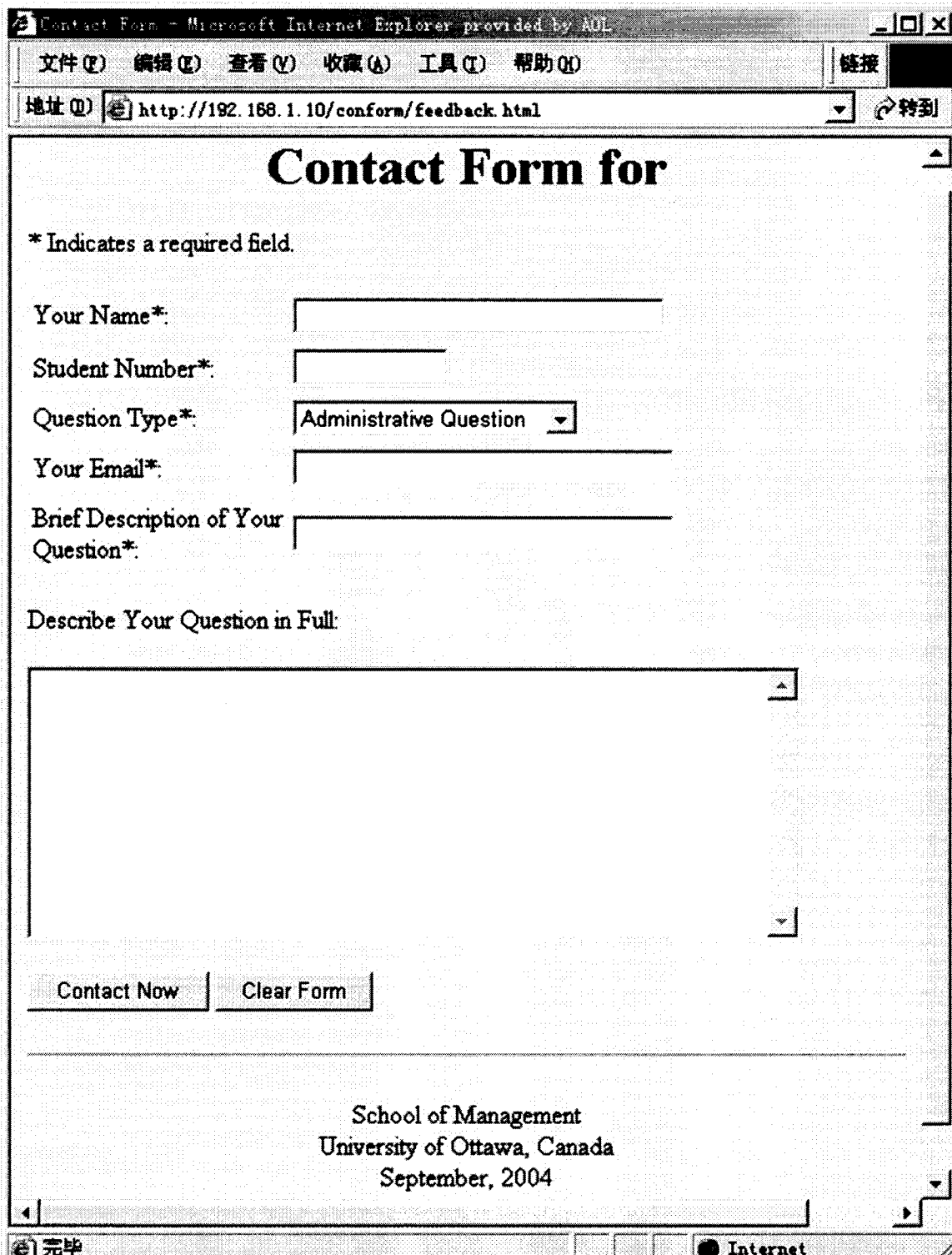


Figure 4-8: Contact Form System Graphic User Interface 1

- **Attach Comments**

The teaching assistant and professor users can select one specific student message to read from the index of all students' messages.

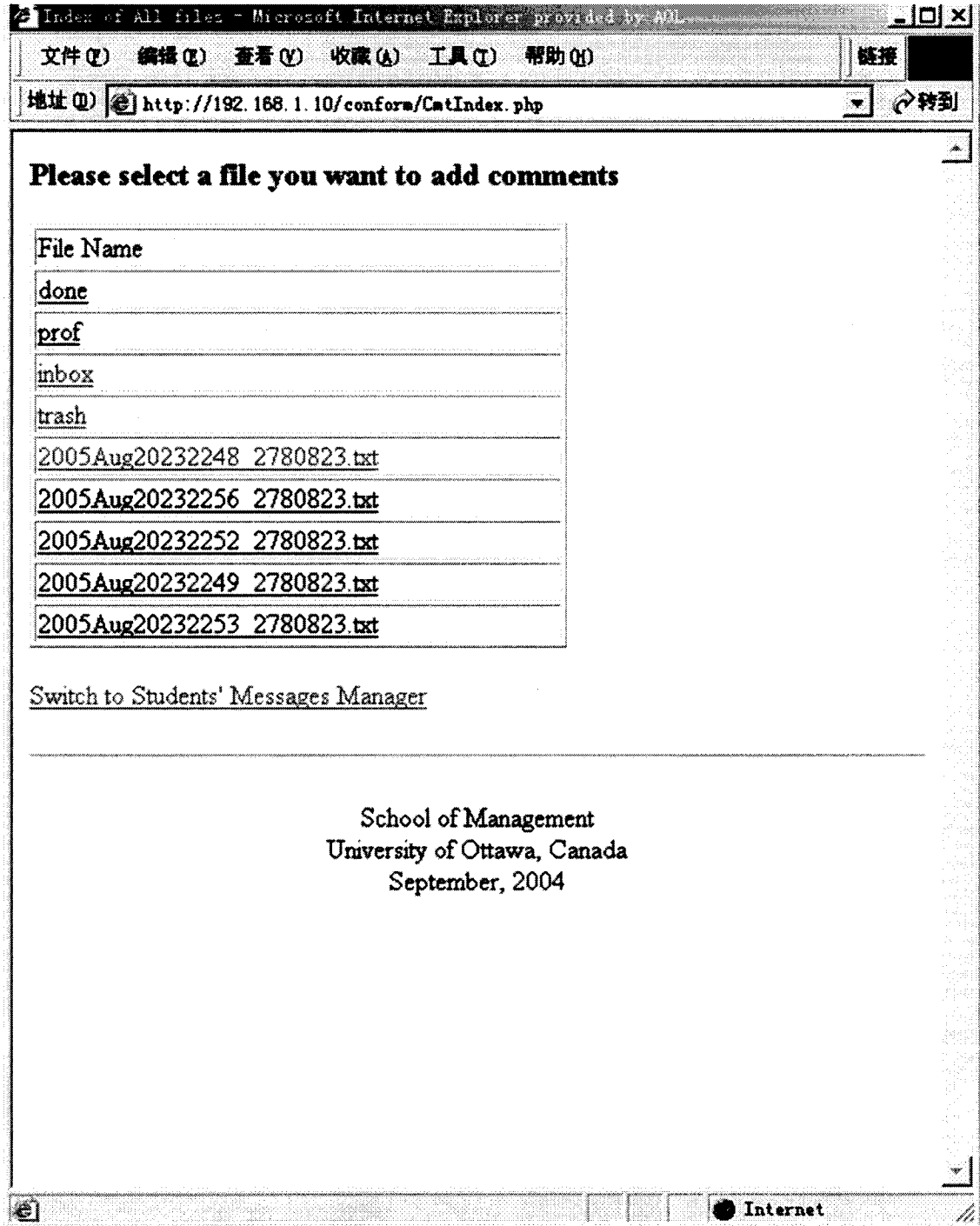


Figure 4-9 Contact Form System Graphic User Interface 2

Then they can read one student's message in the upper text box and add their comments in the lower. Comments will be attached with the student's original message after the "save now" button is clicked. The program puts comments and the corresponding student's message together in one file.

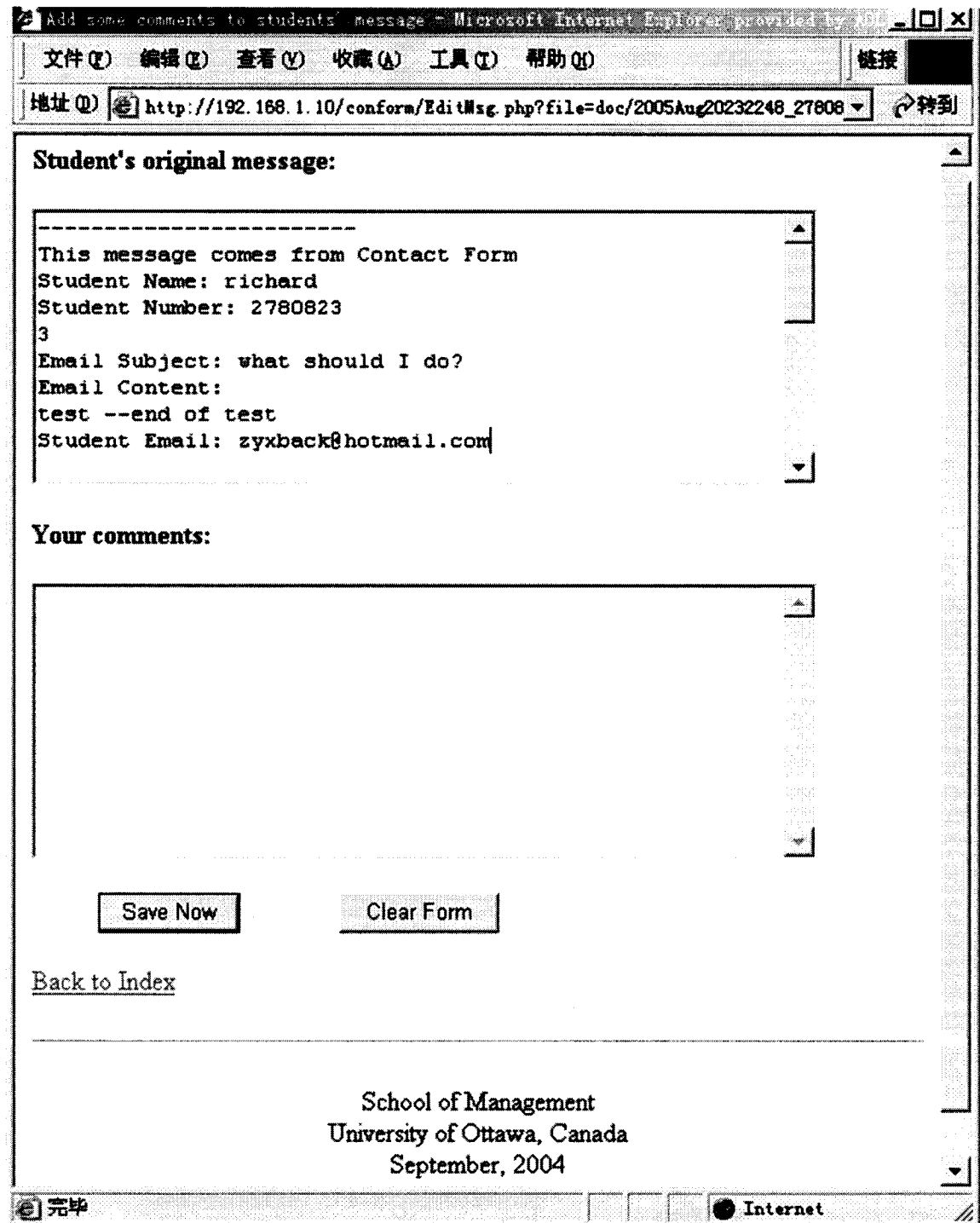


Figure 4-10: Contact Form System Graphic User Interface 3

- **Students' Messages Manager**

When one student's message has been commented or handled, it needs to be put it away, such as sending to professor, completing it or sending to garbage. Here is a screenshot.

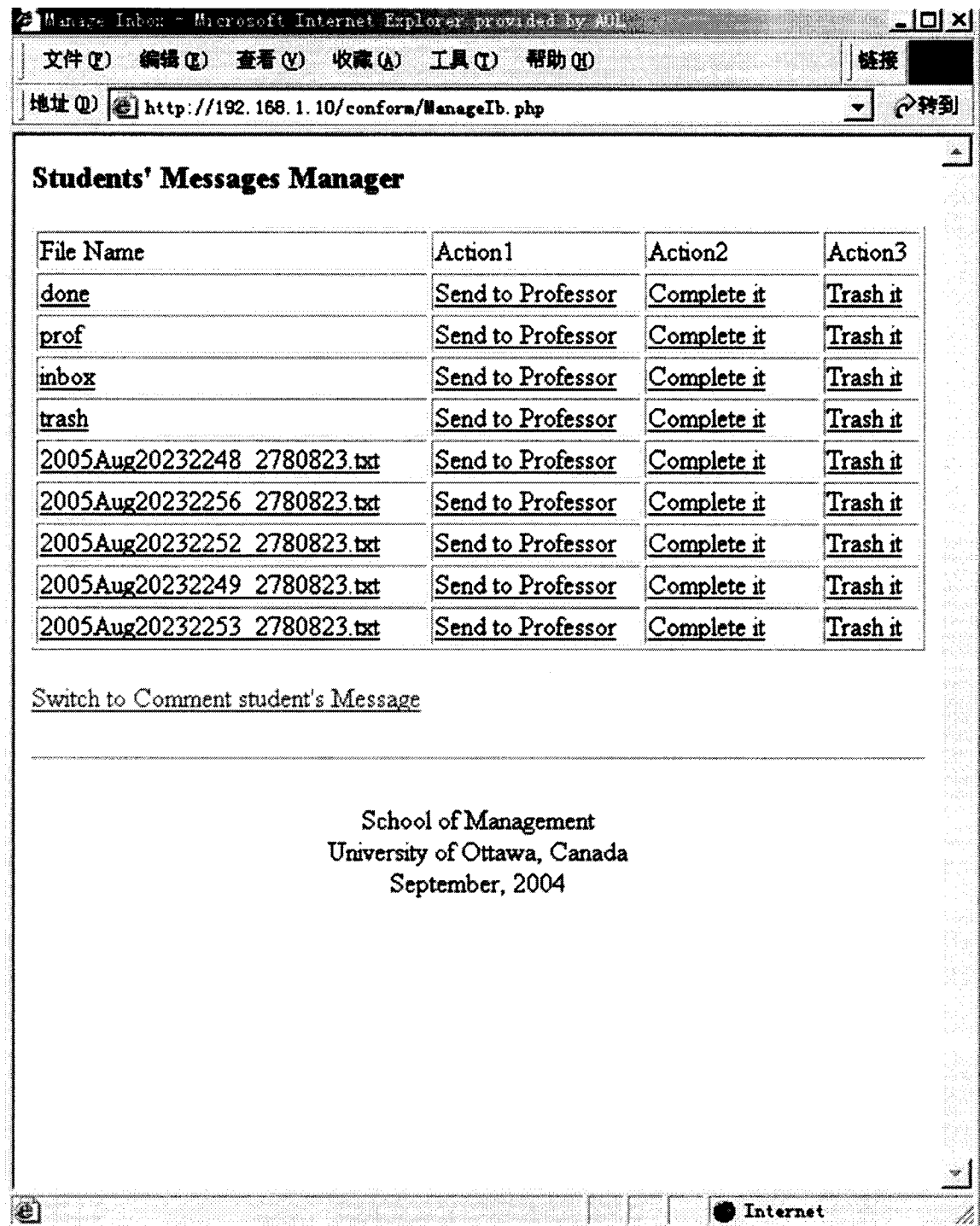


Figure 4-11: Contact Form System Graphic User Interface 4

4.3.2.3 Quality Assessment

We have to test the program on the different computers, operating systems and network environments such as a local machine and target machine, Windows and Linux/Unix, and LAN and Internet.

First, we made a test on the local machine, the system of this computer was Windows and the Apache web server. This part went very smooth and almost no part of the program needed to be changed because most of test had been done during the development of the program.

Next, we tested it on two computers through the local network. We set up another computer, which run as a web server. The system of the server computer was Xandros and the Apache web server. We tested the program on the client computer, which has a Windows operation system. It worked pretty well.

Thirdly, we tested it via the Internet. We uploaded the program to the Macnash server. The system of that server was Linux/Unix and the Apache web server. The program also worked pretty well.

The last step, we tested it on the target server. We uploaded the program to the doc-depot server, which is for the students and professors in the School of Management at the University of Ottawa. We came across some problems with the URL because the doc-depot server adopts some technologies to hide their real links. After we contacted the administrator of the doc-depot server, Mr. Paul Massue-Monat, we got many helpful ideas, including links, security, etc. Then we made and tested a special version for the target server, in which the corresponding source code have been modified for the URL problems. At last, it worked properly and had much more security than ever before.

Here is a flow chart to describe the procedure of the testing.

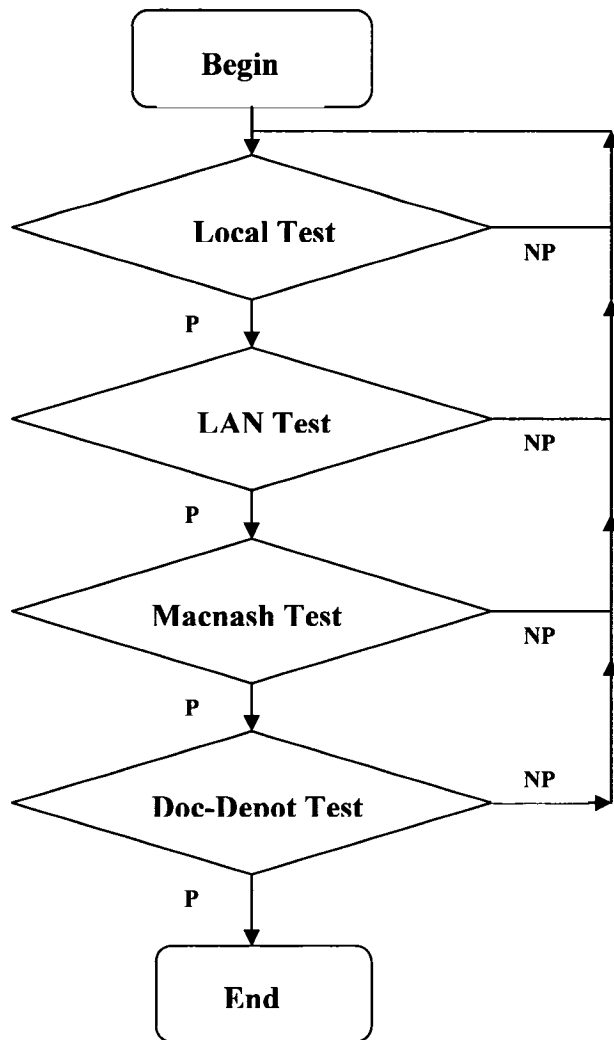


Figure 4-12: Flow Chart Of the Testing Of the Contact Form System

4.3.2.4 Documentation

This is an important phase for the development. During the phase of programming, we have kept composing documentation such as comments on the code, and issues we came across, etc. After completed the testing phase, we spent much more time to compose necessary documents such as the introduction and installation. The affected files are `readme.txt`, and `install.txt`, etc. which can be found in the Appendix 7.3.

4.3.3 New Approach

After this contact form system had been used for two study terms, the 2004 fall term and 2005-winter term in the School of Management, we found another approach, phpBB, which could achieve the same goal. This is a new communication system, which has essential contact functions and is much more open to the student, teaching assistant and professor users.

PhpBB, which is to create communities including students, teaching assistants and professors, is free software and can be downloaded from www.phpbb.com. We have used this new approach in some courses such as Statistics for Management I (ADM2303), <http://macnash.admin.uottawa.ca/forum2303/>, and Statistics for Management II (ADM2304), <http://macnash.admin.uottawa.ca/forum2304/>.

After using it for a while, we find some new issues had come up, such as the admin or professor users need longer time to get familiar with it and the student users may start some irrelevant topics about courses, etc. However, generally speaking, this approach worked well and communications between students, teaching assistants and professors were enhanced.

The following is a screenshot in the ADM2303 course forum

ADM2303 View Forum - Administrative Issues - Microsoft Internet Explorer provided by AOL

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H) 链接

地址(A) http://macnash.admin.uottawa.ca/forum2303/viewforum.php?f=5&sid=c6c75efae49a0332c639ef4

ADM2303

Welcome to the Forum of Statistics for Management I

[FAQ](#)
[Search](#)
[Memberlist](#)
[Usergroups](#)
[Register](#)
[Profile](#)
[Log in to check your private messages](#)
[Log in](#)

Administrative Issues

Moderators: [aashish](#), [phansalker](#)

Users browsing this forum: None

[newtopic](#) [ADM2303 Forum Index -> Administrative Issues](#) [Mark all topics read](#)

Topics	Replies	Author	Views	Last Post
Announcement: Final Marks List	2	aashish	126	Wed Aug 17, 2005 3:01 pm Guest
Announcement: Assignment 2 Marked	1	aashish	41	Tue Jul 12, 2005 6:11 pm mrmutez
Announcement: Assignment 1 and Midterm Marks	1	aashish	103	Mon Jun 27, 2005 4:52 am Guest
Announcement: Assignment 1 Marked	2	aashish	128	Tue Jun 21, 2005 10:12 pm a2303
Announcement: Mid-Term Examination	1	phansalker	198	Mon Jun 13, 2005 10:45 pm Guest
Marks are up on Infoweb!!!	0	Guest	3	Wed Aug 24, 2005 3:48 pm Guest
unofficial final marks	1	Guest	24	Wed Aug 24, 2005 3:38 am mrmutez
STUDENT ID!!!	0	Guest	20	Mon Aug 08, 2005 3:21 am Guest
Fina Exam	1	Guest	42	Mon Aug 08, 2005 12:18 am aashish
Assignment 3	2	mrmutez	44	Fri Jul 29, 2005 4:08 am Guest
assignment # 2 marks	2	Guest	44	Tue Jul 12, 2005 4:21 am

Figure 4-13: Graphic view of new approach - PhpBB

4.4 Backup Issue

4.4.1 Motivation

After we use a system such as ETPRL, ETPHP, and phpBB for a long time, we will definitely get some important data including homework files students uploaded, and system configuration files, etc. Of course, we do not want to lose them during a certain period.

Therefore, it is critical for us to backup these data regularly because computer corruption may happen any time, such as hard drives may fail, motherboards may short out, and mistakes by users may erase data. This kind of tragedy, a significant data loss, has happened to the Macnash server during the time of this thesis. A great deal of important data stored in that server disappeared. Fortunately, we found some backups later somewhere else.

This is one motivation for us to do some research about the backup issue and make a good backup strategy.

Usually, for security reasons, professors may be responsible to perform the backup, which should be a specific regular work of a network administrator. In our situation, Professor John C. Nash is responsible for the regular backup works. To simplify professors' workflow and ease their burden is another motivation to find a good and possible backup method or strategy.

4.4.2 Procedure of implementation

4.4.2.1 Analysis

There are many methods for backing up the data. We could backup them manually or automatically. We also could send backups to different hard disks in different computers, or to CDs, floppy disks, or even tapes. In a word, a good backup strategy should consider

separation of backup media storage, method and time, and it takes human less or no time after the strategy is set.

Automatic backup of the data to different hard disk drives on different computers via the Intranet or Internet in different time is a suitable backup strategy for us. We have two or three computers. Each of them has a Linux or windows operating system installed. These computers are connected by the Intranet or Internet. We want to backup or copy the important data stored in one computer to another computer daily, such as 1:00 AM every day. One solution is to use the Cron script, SSH and Rsync technologies provided by Linux. Below we discuss the details of this solution, which could be a suitable backup strategy for us.

4.4.2.2 Implementation

We need one Rsync server and one Rsync client. The data that needs to be backed up is stored in the server. The backups go to the client. There are two different methods to implement it, including the “pull” and “push” ways. The “pull” way means the client sends a backup request to the server to finish a backup job. Similarly, the “push” way means the server sends a backup request to the client to finish a backup job.

Here is a diagram of the “pull” way and how to implement it step by step.

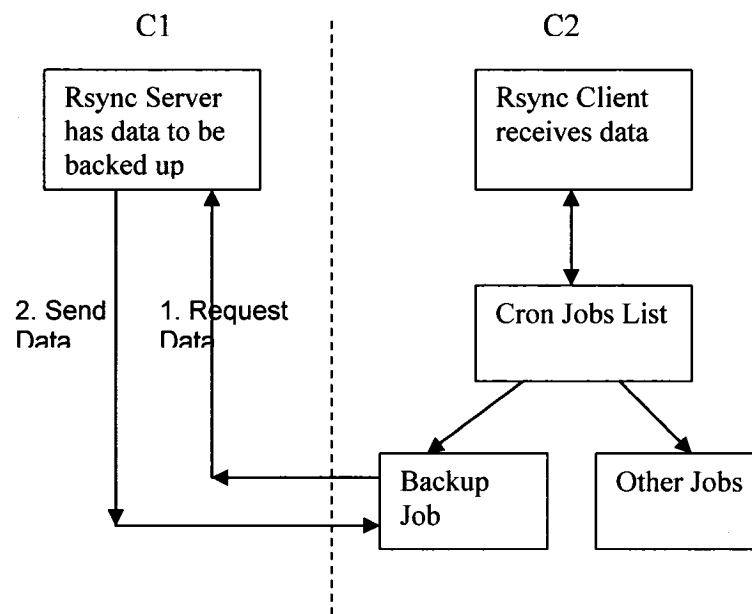


Figure 4-14: Backup by the "pull" way

On the server computer, we need to finish the following steps:

1. Modify or create `/etc/rsyncd.conf`, here is the content of this file.

motd file = `/etc/rsyncd.motd`

log file = `/var/log/rsyncd.log`

pid file = `/var/run/rsyncd.pid`

```
lock file = /var/run/rsync.lock
```

```
[delzyx]
```

```
path = /home/root/wwwdata  
comment = need to be backed up  
uid = root  
read only = yes  
list = yes  
auth users = root  
secrets file = /home/root/ root.pas
```

root.pas is a secret file, which contains the password of a “root” user. Make sure this file is only readable and writable to the “root” user by this command: `chmod 600 root.pas`

When we type `ls -l`, the following information should be listed.

```
root.pas          -rw----- 1 root  root
```

And the format of the content of root.pas is: `root:rootpassword`.

2. Start the Rsync server by typing `/usr/local/bin/rsync --daemon`
That’s all for the configuration in the Rsync server.

On the Rsync client computer, we need to type the following command in a shell window:

```
rsync -vzrtopg --progress --delete root@192.168.1.3::delzyx .\backup\
```

Then we could copy the data in the `/home/root/wwwdata` on the server computer to the backup directory on the client computer.

Here are some important notes for this command.

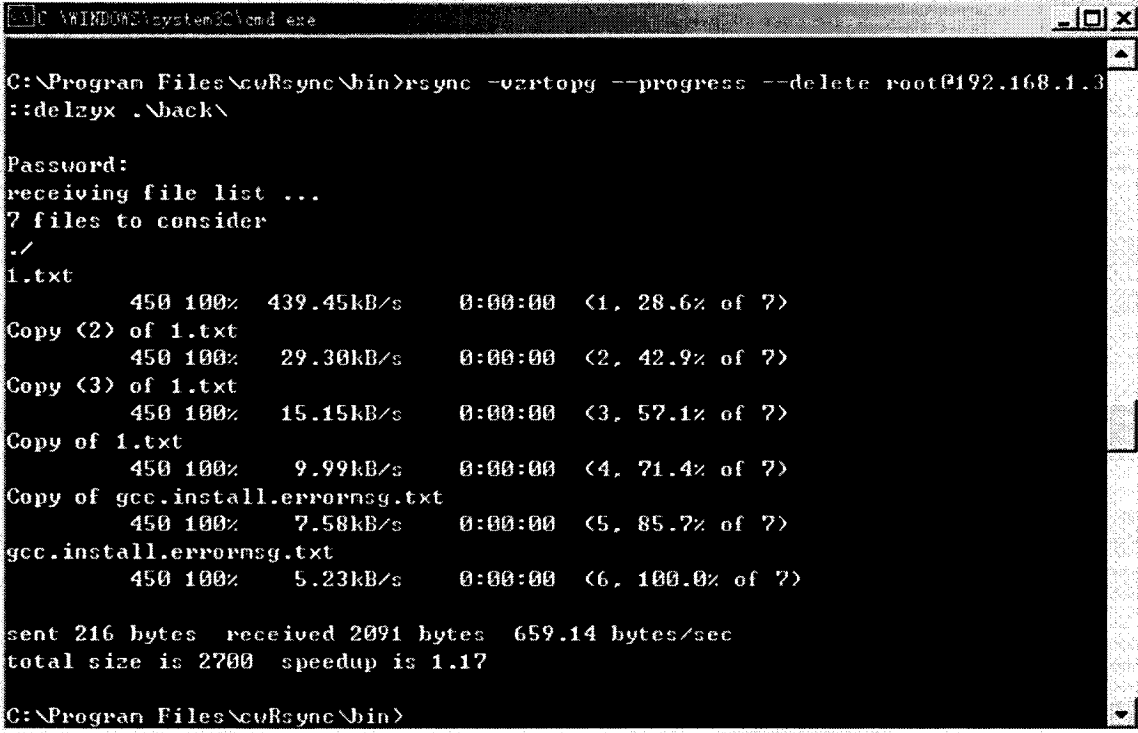
`--progress` means it shows the progress of copy and other useful information.

Root is a user account in the server computer.

192.168.1.3 is the IP address of the server computer.

Delzyx should match the label in rsyncd.conf in the server computer.

The backup directory must exist before this command is executed.



```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\cvRsync\bin>rsync -uzrtopg --progress --delete root@192.168.1.3::delzyx .\back\
Password:
receiving file list ...
7 files to consider
./
1.txt
  450 100% 439.45kB/s   0:00:00 <1, 28.6% of 7>
Copy (2) of 1.txt
  450 100%  29.30kB/s   0:00:00 <2, 42.9% of 7>
Copy (3) of 1.txt
  450 100%  15.15kB/s   0:00:00 <3, 57.1% of 7>
Copy of 1.txt
  450 100%   9.99kB/s   0:00:00 <4, 71.4% of 7>
Copy of gcc.install.errormsg.txt
  450 100%   7.58kB/s   0:00:00 <5, 85.7% of 7>
gcc.install.errormsg.txt
  450 100%   5.23kB/s   0:00:00 <6, 100.0% of 7>

sent 216 bytes  received 2091 bytes  659.14 bytes/sec
total size is 2700  speedup is 1.17
C:\Program Files\cvRsync\bin>
```

Figure 4-15 Demo for the "pull" way

Here is a diagram of the “push” way and how to implement it step by step.

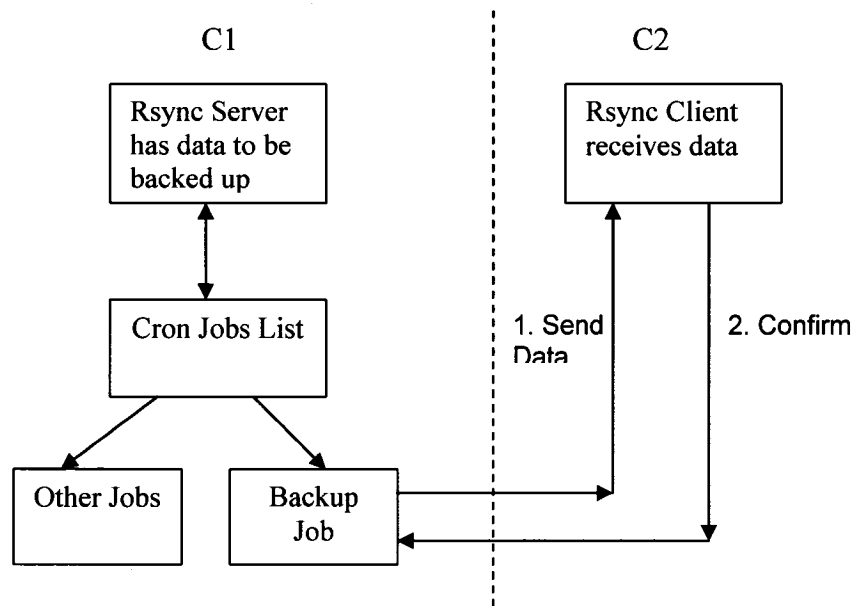


Figure 4-16: Backup by the "push" way

On a client computer, which is the Macnash server, we need to complete the following steps:

1. Create a user “rzhang” and grant this user the "ssh" and "backup" rights.
2. Make sure we have the `\home\rzhang\backup` directory, to which we backup data from a server computer. And grant user “rzhang” the read and write permissions to this folder.
3. Make sure the client computer has Rsync and SSH installed.

On the server computer, some steps below are needed.

1. Run one command below.

```
rsync -vzrtopg --progress --delete \home\rzhang\wwwdata  
rzhang@macnash.admin.uottawa.ca:backup
```

“rzhang” is the user name on the macnash.admin.uottawa.ca server.

We could backup all files in the `\home\rzhang\wwwwdata` folder on the server computer to the `\home\rzhang\backup` folder on the client computer, which is `macnash.admin.uottawa.ca`.

2. Make a shell file (`test.sh`), which contains the `rsync` command above.

```
#more /root/test.sh
```

```
#!/bin/sh
```

```
rsync -vzrtopg --progress --delete \home\rzhang\wwwwdata
```

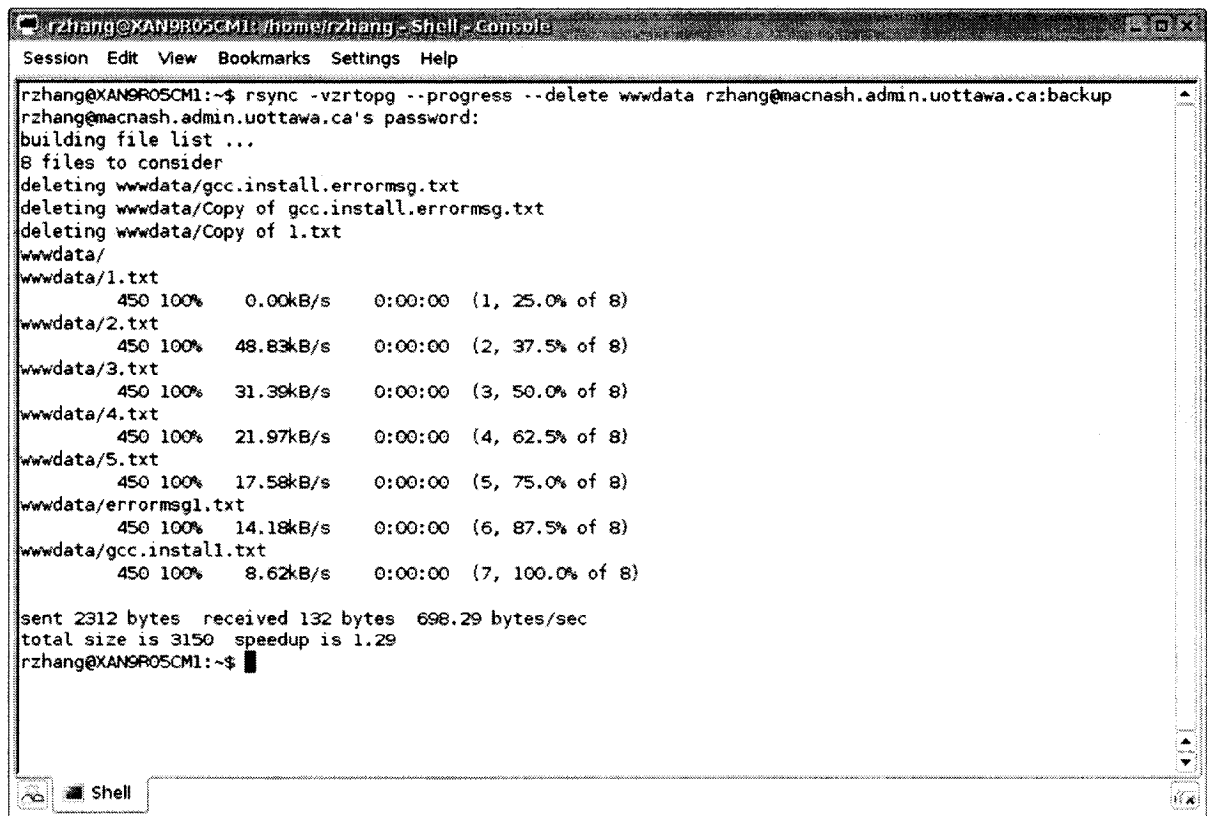
```
rzhang@macnash.admin.uottawa.ca:backup
```

3. Change the cron table and let the server finish the backup job regularly by itself.

We could use `crontab -e` to add the following line:

```
0 21 * * 1-5 /root/test.sh
```

Then the sever will run `test.sh` on 21:00 every night from Monday to Friday.



```
rzhang@XAN9R05CM1: /home/rzhang - Shell - Console
Session Edit View Bookmarks Settings Help
rzhang@XAN9R05CM1:~$ rsync -vzrtopg --progress --delete wwwdata rzhang@macnash.admin.uottawa.ca:backup
rzhang@macnash.admin.uottawa.ca's password:
building file list ...
8 files to consider
deleting wwwdata/gcc.install.errormsg.txt
deleting wwwdata/Copy of gcc.install.errormsg.txt
deleting wwwdata/Copy of 1.txt
wwwdata/
wwwdata/1.txt
  450 100%  0.00kB/s  0:00:00 (1, 25.0% of 8)
wwwdata/2.txt
  450 100%  48.83kB/s  0:00:00 (2, 37.5% of 8)
wwwdata/3.txt
  450 100%  31.39kB/s  0:00:00 (3, 50.0% of 8)
wwwdata/4.txt
  450 100%  21.97kB/s  0:00:00 (4, 62.5% of 8)
wwwdata/5.txt
  450 100%  17.58kB/s  0:00:00 (5, 75.0% of 8)
wwwdata/errormsg1.txt
  450 100%  14.18kB/s  0:00:00 (6, 87.5% of 8)
wwwdata/gcc.install.txt
  450 100%   8.62kB/s  0:00:00 (7, 100.0% of 8)

sent 2312 bytes  received 132 bytes  698.29 bytes/sec
total size is 3150  speedup is 1.29
rzhang@XAN9R05CM1:~$
```

Figure 4-17 Demo for the "push" way

The last issue is about the improvement of security. We consider using `ssh-keygen` to generate our own private/public key pair and share the public key to a remote computer. One advantage of this method is that we are able to SSH a remote computers without being prompted for a password. Let us suppose that we login as user “rzhang”. Here are steps to accomplish this.

1. Execute this command in a shell window: `ssh-keygen -t rsa`. Then computer makes the private and public keys, which should be in the `/home/rzhang/.ssh/id_rsa` and `/home/rzhang/.ssh/id_rsa.pub` files separately
2. Make sure the private key file, `/home/rzhang/.ssh/id_rsa`, is not readable by anyone else(`chmod 600 ~/.ssh/id_rsa`)
3. Copy the public key to a remote computer. For example, you could use this command: `scp ~/.ssh/idrsa.pub username@remotecomputer:~/.ssh/authorized_keys`

Then we should now be able to login to remote computers by SSH, and not being prompted for passwords. Besides, this method gives us a little higher security because the private and public keys are encrypted and stored.

5. Assessment, Recommendation and Conclusion

This chapter is to summary or present my ideas, review, assessment, recommendation and conclusion of some interesting situations related this thesis, which are hopefully helpful for the future research about this topic.

In the analysis phase, some ideas have been presented in Chapter 3. The discussion about ETPRL and ETPHP can be found in Section 3.1.3. The ideas related e-tutor and other course systems can be found in Section 3.6. After the improvement phase, more issues then can be discussed in the following sections.

5.1 Contact form system, Co4 and phpBB

Since the contact form system works well, why do we replace it by the private message capability in phpBB? Similarly, the Co4 forum turns out to be useful for some purposes such as a photo-journal, but why is it not so good for a student forum?

Basically, phpBB is a modern forum system, which contains most of the functions of the contact form system and Co4 forum, and is, moreover, suitable for use by classes involving many students.

Compared to the contact form system, phpBB does not only provide the “contact” functions, but also has some other related functions. For example, we can use it to make a public announcement to the whole class. Students can check the answers of the questions other students asked and get some helpful information, and so on. However, the contact form system is easier to use because users do not need to spend time to register and learn many non-basic functions. The contact form is easier to install because it does not require the support of database program. PhpBB, on the other hand, requires the installation of MySQL. While MySQL is stable and reliable, its installation is non-trivial. In two words, the contact form is simple and straightforward software, which only focuses on the

“contact” functions. In the particular situations of real use, that is, the ADM2303 and ADM2304 statistics courses, we prefer to provide just one system that contains both the contact and forum functions, and then students only have to become familiar with just the single application.

Compared to Co4, phpBB is easier to use in a course situation where we need multiple subject streams, i.e., multiple forums. For example, in order to access one specified forum in Co4, we need to give the fileset or forum name before the login. However, phpBB allows multiple forums and lists all forum names such as Administrative Issues, Assignment 1, Midterm Questions, and so on when a user has logged in. It is also possible for guests, ones who do not provide user name and password, to check those forums for which guest access is allowed. This kind of functionality can save much time for students. Moreover, phpBB is also able to make some forums that are only open to the registered users after they log in, make some forums that are "retired" by hiding them from all users but the forum moderators. Besides, phpBB allows self-registration of users, while Co4 requires the file owners, usually professors, to add users. What is more, the user graphic interface of Co4 is not as good as phpBB. That is reasonable because the interface of latest software usually is better than the one of older software.

Because phpBB uses the MySQL database and the PHP programming language while Co4 uses simple files and the Perl programming language, phpBB has a better data management than Co4 in some ways. Basically, a database provides more efficient queries and updates, can handle concurrent access, and crash recovery, and so on than simple files. For example, phpBB has a file locking function, which can prevent the same material being updated by two different users at the same time, while Co4 has not. Besides, phpBB has a very easy-to-use, database based mechanism to backup its data.

However, it is not trivial to view or extract data from the backup files in phpBB because they are in a particular format that is intended for internal use. On the contrary, Co4 keeps the format of files very close to the ultimate HTML that can be displayed by any

other software programs. That gives administrators a straightforward way to have a complete control of its data.

The biggest weakness of phpBB is that it does not allow users to post images, sounds, and other specific files. Users can only include the links of images or other files in external servers. However, Co4 allows users not only to include these kinds of files but also to organize these items or files in a particular order. Thus Co4 is excellent for photo albums, presentations, etc.

Given the large number of students and multiple topics in the ADM2303/2304 courses, and the limited need of graphics, phpBB is more suitable for the forum function. As we have already noted, it includes a private message function that can replace the contact form.

Therefore, the contact form system, and Co4 are not "lost" because they just have different situations in which they are appropriate. In fact, the lack of picture in phpBB is considered as a nuisance for some statistic courses, where graphs are more important for the pedagogy.

Hence, in general, phpBB is recently considered to be more suitable to students and used well in some courses for a few study terms in the School of Management at the University of Ottawa. To learn how to install and use phpBB takes time, but considering the convenience of phpBB, these efforts seem to be very worthy. The related information can be found in Section 4.3.3 New Approach.

5.2 Deming's ideas and E-tutor project

In the e-tutor project, lots of software programming works have been done successfully by applying methods, such as design, programming, testing, and so on, from software engineering to improve the e-tutor project. Why do not we just use the ideas from software engineering? Are Deming's ideas still necessary for the e-tutor project?

While there are many approaches that can lead to a satisfactory result in a project such as this, the Deming approach and perspective are particularly useful:

- The Deming philosophy looks at the whole process, not just the software. We want to save time and effort for all the parties involved.
- We want to reduce variation. In homework submission and return, this is reflected in our attempts to develop methods that permit submission and retrieval wherever and whenever, avoiding “special” arrangements when students must miss class or pickup, and especially avoiding the difficulties that attend special arrangements for regarding papers.
- For the present activities, Deming’s ideas do not provide a large benefit over other approaches. The value of the approach is in the larger, ongoing effort to improve the work processes of teaching and learning, where it provides us with direction and constancy of purpose.

During the improvement of the E-tutor project, we have been loosely following Deming's methods. Here are some examples.

All components in the E-tutor project, such as ETPRL, ETPHP, contact form, phpBB, and the others, have similar goals: to facilitate the educational tasks, to simplify workflow of professors, teaching assistants, and students, and ultimately to test the concept that a “watch the student work” teaching method can be in part implemented by using WWW-type tools which run at low bandwidth, yet still accommodate 100 students or more. These purposes affect the design, development, and improvement of these components and the resulting improvements in teaching processes and environments. This fits in very well with Deming's point: “Create constancy of purpose for improvement of product and service” (Deming 1986).

At the beginning of the maintenance and improvement of the E-tutor project, I was unfamiliar with Unix/Linux, Perl, PHP, and MySQL, etc. Professor Nash gave me helpful instructions by lending me many related books and providing some training. When I

came across some difficulties, we communicated by email, telephone or meeting. Most of problems were solved quickly. This fits the sixth of Deming's fourteen points: "Institute on-the-job training" (Deming 1986).

We keep working or searching for better tools for e-education. For example, ETPRL works well and has a place when students have only to submit a single assignment, but later we developed and used ETPHP to simplify the management of multiple assignments and sections. Co4 and the contact form system works well, but later we found and used phpBB as an approach that was easier to use for students and to manage for professors.

Besides, in the ETPRL project, we introduced a newer graphic user interface, fixed a critical uploading time-delay problem by checking the names of the uploaded files, shortened the number of clicks for students to upload their homework, shortened the process to get students' homework for TAs or professors, and shortened the process to return homework to students. Similarly, in the phpBB forum, there is now a separate forum for each assignment question and the forum name is very indicative of the purpose such as Assignment One – Q1, Assignment Two - Q3, and Mid_Term Quiz, etc. This method has simplified the workflow of professors, teaching assistants, and students to some extent by reducing multiple similar postings in different places because users could not locate the similar items. More components in E-tutor are expected to be added and enhanced in the future. This follows the fifth of Deming's fourteen points: "Improve constantly and forever the system of production and service" (Deming 1986).

5.3 My contributions and evaluation

1. ETPRL

I have enhanced the documentation of installation, configuration and program code. There was not enough documentation for users who are unfamiliar with web servers and installation. The current documents are intended to be helpful and worth reading.

Besides, I have fixed some problems below.

- **Uploading problems**

This system would slow or “hang” if there were some special characters in the name for file chosen for upload by a student. Later, we found the same problems in the batch download and upload modules.

From the view of programming, this was a very small problem. However it reduced the effectiveness in the whole system sometimes. Basically, this correction is a success in streamlining operations

- **Password changing problem**

Admin or professor users might lose their passwords when they tried to change their passwords. When they left their new passwords blank and clicked the “OK” button, their passwords would be changed to random strings.

From the view of programming, this was again a very small problem as well. However it can cause serious delays and inconvenience to users. Basically, this correction is to prevent potential problems and enhance the system.

I have also added a new function: Password Retriever for the admin users. When the admin users forget their passwords, they can use this small function to get their new passwords. Some strategies to deal with security problems of this tool are detailed in Section 4.1.2.1.

2. ETPHP

I have enhanced the documentation of installation, configuration and program code. Again the initial documentation was not sufficient for users who are unfamiliar with web servers and installation.

In addition, I have dealt with the following problems.

- **Set up problem**

In the setup module, there was an obsolete function, `mysql_create_db`, which was intended to create a database but is not supported by MySQL 4.0 or later. After I replaced it with a newer function, `mysql_query("CREATE DATABASE new_basebase")`. It works well. This correction gives us a good lesson for the upgrades in the future. With the installation and use of the newer versions of database programs or program languages, we may prepare to debug or change our old programs, which even works very well in the old environment. For more details on these, please check Section 4.1.2.2.

- Add a new function in File Extractor

For the File Extractor program, I have added a new function, which can show helpful information when users give wrong critical parameters. Please check the details in Section 4.1.2.2.

- External login

The University of Ottawa is believed to plan to institute a campus-wide common login system. Thus systems, such as ETPHP and ETPRL, need to be able to call external routines to authenticate users. The objective of the changes we have prepared is to make the login module clearer and reusable. At the moment this is only a demonstration tool to show that it is possible to link ETPHP into other systems. Please check the details in Section 4.1.2.2.

3. Contact Form System

I have finished the design and development of the contact form system with the help of Professor John C. Nash. This system was used successfully for two study terms by the students, TAs and professors in the School of Management at the University of Ottawa. Basically, it is a simple system for students to submit their questions, and for professors and TAs to manage and reply students' questions. Please check the details in Section 4.3.

4. Backup Issue

The objective is to find a good strategy to backup data for the systems we have. We found an automatic backup solution with the use of rsync and cron, which works fairly well in some machines. Please check the details in Section 4.3.

However, for some security reasons, in the MacNash and Course servers, we could not fully implement this solution. Right now we still need to initiate the backup manually rather than automatically. The issues appear to be that the cron task doesn't start and the server cannot accept passwords generated by ssh-keygen, etc. Hence, we have it working as a command and this solution is still pending.

5.4 Overall score for different components

The following table is to give the scores of the different components in the E-tutor project by some criteria according to my experience.

Tools	Set up	User Friendly	Maintenance	GUI	Popularity	Overall Score
ETPRL	Fair	Good	Easy	Good	Popular	Good
ETPHP	Fair	Excellent	Easy	Excellent	Popular	Excellent
CO4	N/A	Fair	N/A	Good	Popular	Good
Contact Form	Easy	Excellent	Easy	Good	Popular	Excellent
Backup	N/A	Good	Easy	Good	N/A	Good
PICKSEAT	N/A	Good	N/A	Good	N/A	Good
ET-GRASED	N/A	Good	N/A	Good	N/A	Good
FOTOHTML	N/A	Good	N/A	Good	N/A	Good
File Extractor	Easy	Excellent	Easy	Good	N/A	Excellent

5.5 List of most computer technologies or tools involved

Most of the technologies and tools related to this thesis are listed below, which is intended to show other developers who consider making the improvements of these projects. Given these developers are very familiar with Windows but only have limited knowledge about Unix/Linux, this table could help them to know what technologies or tools they need to know and use, and probably how much time and effects they need to spend.

Tools	Money Cost	Installation Difficulty	Learning Difficulty (Configuration, etc.)
Windows (Windows 98, Windows 2000, Windows XP)	Not Free	Easy	Easy
Linux (Xandros)	Free*	Easy	Medium
Abyss (Web Server) in Windows	Free	Easy	Medium
Apache (Web Server) in Linux	Free	Difficult	Medium
MySQL (Database) in Linux Phpmyadmin	Free	Difficult	Medium
HTML/XML (Programming Language)	Free	Easy	Medium
PERL (Programming Language) in Linux	Free	Difficult	Long time to learn
PHP (Programming Language) in Linux	Free	Difficult	Long time to learn
Java (Programming Language)	Free	Easy	Long time to learn

Winscp (to access server by SSH)	Free	Easy	Easy
XAMPP (Apache+PHP+PERL+MySQL+ phpmyadmin) in Linux	Free	Easy	Easy
Easyphp (Apache+PHP+MySQL+ phpmyadmin) in Windows	Free	Easy	Easy
Rsync, Cron, cwRsync and SSH	Free	Medium	Medium

* Linux (Xandros) is free for home edition, but not free for business edition.

Figure 5-1: List of computer technologies and tools involved

According to this table, developers who are new to Linux/Unix could select an easy way to start their work as soon as possible, which saves a great deal of time and efforts. And of course, developers, who have more experience in this field, could build their own learning curves to shorten time and efforts of the development.

Though the installation and development of programs in Unix/ Linux usually takes us more time and efforts than that in the Windows operating system, the results of saving money costs are obvious and positive. Therefore it can be a suitable alternative for those businesses or organizations that have limited budget.

6. Future Work

This thesis work has been successful in simplifying the workflow of professors, teacher assistants, and students with the e-tutor tools. However, most of the effort focused on the analysis and improvement of a few specific tools or parts thereof..

To continue the work, it will be important to consider education or “people” rather than tools themselves to improve the teaching processes and environment. In such efforts, it will be useful to apply ideas from pedagogical theory and practice to the e-tutor project. Both from a pedagogical perspective and from the point of view of “marketing”, for example, a survey of users could be helpful to enhance the e-tutor project and position it in the range of possible aids to teaching and learning..

Clearly, if there are adequate resources and requirement, some functions such as online self-tests, live chat, and online team discussion, may be good ideas for the future work in this project.

7. Bibliography and Internet references

1. John C. Nash, D. Phil. (Oxon.) (2003), **“Putting the E to work in E-teaching”**, School of Management, University of Ottawa
2. Segaller, Stephen (1998), **" A Brief History Of The Internet"**
3. Advisory Committee for Online Learning (2001), **“The e-learning evolution in colleges and universities: a pan-Canadian challenge”**
4. Ash, Charlotte (2000), **Towards a New Cost-Aware Evaluation Framework**
Educational Technology & Society 3(4) 2000, pp. 126-131
5. Bob Baxley (2003), **“Making the Web work: designing effective Web applications”**
6. Huryk, L.A. (2005), **"Enhance orientation through e-learning"**, Nursing management 36 (1), pp. 48-49
7. Mary Walton (2003), **“The Deming Management Method”**
8. John C. Nash, D. Phil. (Oxon.) (2003), **E-Tutor Home Page**, School of Management, University of Ottawa, <http://macnash.admin.uottawa.ca/~etutor/>, date accessed: December 28, 2005
9. Council of Ontario Universities (2003), **“Impressive Number of Students Apply for Fall 2003 Admission”**, <http://www.ouac.on.ca/news/dblcohort/ni-jan17press.html>, date accessed: December 28, 2005

10. Leblanc, Karen (2001) and John C. Nash, D. Phil. (Oxon.), **e-TUTOR**, University of Ottawa, School of Management, http://macnash.admin.uottawa.ca/~etutor/k_leblanc.pdf, date accessed: December 28, 2005
11. Martin Dougiamas (2005), **Moodle Learning Management System Home Page**, <http://moodle.org/>, date accessed: December 28, 2005
12. CyberAtlas staff (2005), **“Population Explosion”**, http://www.clickz.com/stats/sectors/geographics/article.php/5911_151151, date accessed: December 28, 2005
13. Myron Tribus (2003), **“TQM in Education, The Theory and How To Put It to Work”**, <http://www.mehs.educ.state.ak.us/quality/theorytqmineduc.pdf>, date accessed: December 28, 2005
14. George M. Piskurich (2003), **“The AMA handbook of E-learning: effective design, implementation, and technology solutions”**, Publisher: New York: AMACOM
15. Rumble, Greville (2002), **“Analyzing Cost/Benefits for Distance Education Programs”**, Techknowlogia, vol 4 no 2 April-June 2002,
16. Steven Narmontas (2005), **Manhattan Virtual Classroom Home Page**, <http://manhattan.sourceforge.net/>, date accessed: December 28, 2005
17. Thomas De Praetere (2005), **Claroline Home Page**, <http://www.claroline.net/>, date accessed: December 28, 2005
18. W. Edwards Deming (1986), **“Out of the Crisis”**
19. W. Edwards Deming (1994), **“The New Economics”**

20. Wang, Sudan (2002), “**Issues in adding special purpose functions to scientific or statistical free software**”, University of Ottawa, Faculty of Administration, Master of System Science thesis, December.

21. Apache Friends Team (2005), **XAMMP**,
<http://www.apachefriends.org/en/xampp.html> , date accessed: December 28, 2005

22. Apache Web Server Team (2005), **Homepage of Apache**, <http://www.apache.org/> ,
date accessed: December 28, 2005

23. MySQL Database Server Team (2005), **Homepage of MySQL database server**,
<http://www.mysql.com>, date accessed: December 28, 2005

24. PHP scripting language Team (2005), **Homepage of PHP scripting language**,
<http://www.php.net/>, date accessed: December 28, 2005

25. PERL scripting language Team (2005), **Homepage of Perl scripting language**,
<http://www.perl.com/>, date accessed: December 28, 2005

26. Xandros Operating System Team (2005), **Homepage of Xandros**,
<http://www.xandros.com/> , date accessed: December 28, 2005

27. Aprelium Technologies (2005), **Homepage of Abyss Web Server**,
<http://www.aprelium.com/abyssws/>, date accessed: December 28, 2005

28. Easyphp Team (2005), **Homepage of EasyPHP**, <http://www.easyphp.org/>, date
accessed: December 28, 2005

29. Bryan Williams (2005), **Presentations about Moodle**,
http://moodle.org/file.php/29/presentations/English/Presenters_Handout.pdf, date
accessed: December 28, 2005

30. Paris, P.G. (2004), "**E-learning: A study on secondary students' attitudes towards Online Web Assisted Learning**", International Education Journal 5 (1), pp. 98-112
31. Luke Welling and Laura Thomson (2001), "**PHP and MySQL Web development**"
32. Andrew Harris (2003), "**PHP/MySQL programming for the absolute beginner**"
33. Chuck Easttom (2004), "**Moving from Windows to Linux**"
34. Eric Rosebrock and Eric Filson (2004), "**Setting up LAMP: getting Linux, Apache, MySQL, and PHP working together**"
35. Carl Albing and Michael Schwarz (2005), "**Java application development on Linux**"
36. Rob Flickenger (2003), "**Linux server hacks**"
37. Bob Toxen (2003), "**Real world Linux security : intrusion protection, detection, and recovery**"
38. Clinton Pierce (2005), "**Sams teach yourself Perl in 24 hours**"
39. Sybex Inc. (2003), "**Perl, CGI and Javascript complete**"
40. Theo Petersen (2002), "**Web development with Apache and Perl**"
41. Elizabeth Naramore (2005), "**Beginning PHP5, Apache, and MySQL Web development**"

42. Mark Matthews, Jim Cole and Joseph D. Gradecki (2003), **“MySQL and Java developer's guide”**
43. Kevin Yank (2004), **“Build your own database driven website using PHP & MySQL”**
44. Molly E. Holzschlag (2005), **“Spring into HTML and CSS”**
46. Betsy Bruce (2004), **“Sams teach yourself Macromedia Dreamweaver MX 2004 in 24 hours”**
47. Dan Barrett and Richard Silverman(2001), **“SSH, the secure shell : the definitive guide”**
48. Daniel Jones (2005), **“Separation is Critical to Your Backup Strategy”**, <http://free-backup.info/separation-is-critical-to-your-backup-strategy.html>, date accessed: December 28, 2005
49. Jim Lynch, PCMagazine.com (2005), **“Review of Xandros Desktop OS Version 3 Deluxe Edition”**, <http://www.pcmag.com/article2/0,1895,1760197,00.asp> , date accessed: December 28, 2005
50. John C. Nash, D. Phil. (Oxon.), Pat Suwalski, Marcus Svilans (2005), **E-tutor Homepage**, <http://etutor.sourceforge.net/> , date accessed: December 28, 2005
51. Yankee Group, **“Survey: Moving From Windows To Linux Isn't Cheap (2004)”**, <http://www.informationweek.com/story/showArticle.jhtml?articleID=18900187> , date accessed: December 28, 2005
52. Hotscripts, **“Reviews for Apache Friends XAMPP (2004~2005)”**, <http://www.hotscripts.com/review/26145.html> , date accessed: December 28, 2005

53. Motschnig-Pitrik, R. (2005), "**Person-centered e-learning in action: Can technology help to manifest person-centered values in academic environments?**", *Journal of Humanistic Psychology* 45 (4), pp. 503-530
54. Santerre, C.R. (2005), "**X-Train: Teaching professional remotely**", *Journal of Nutrition* 135 (5), pp. 1248-1252
55. Brown, M.V., Galli, A.L. (2005), "**E-learning goes mainstream**", *Power* 149 (3), pp. 30-39

8. Appendix (programs)

7.1 Programs of ETPRL

7.1.A Test.pl

```
#!/c:\perl\bin\perl.exe
# filename: test.pl
# added by Richard
#####!/usr/local/bin/perl
##
## printenv -- demo CGI program which just prints its environment
##

print "Content-type: text/plain\n\n";
foreach $var (sort(keys(%ENV))) {
    $val = $ENV{$var};
    $val =~ s|\n|\n|g;
    $val =~ s|\"|\"|g;
    print "${var}=\"${val}\"\\n";
}
```

7.1.B achprpw.pl

```
#!/usr/bin/perl

## 2002-7-11 achprpw.pl -- allow administrator to overwrite prof password

# Copyright 2002 J C Nash jcnash@uottawa.ca
# Modified by Richard, 2005.8
# To fix the password change problems

use strict;

use CGI;
require "./base.pl"; # Get the cgi parameters
require "./errmsg.pl";
require "./boiler.pl";
require "./apwok.pl"; # admin password check routine

my %svar = &base();

my $query = new CGI;
my $adid = $query->param( 'adid' );
```

```

my $profid = $query->param( 'profid' );
my $action = $query->param( 'action' );

print STDERR "achprpw: adid=$adid, prof=$profid, action=$action \n";

if ( ( $adid eq "" ) or ( $profid eq "" ) )
{
    # bad input
    hmsg( "achprpw - administrator or prof id null", "You must restart properly",
    $svar{ 'bomb' }, " ", " " );
}
else
{
    # adid ok
    my $cpw = $query->cookie( 'adpwc' );

    # print STDERR "achprpw - retrieved cookie cpw = $cpw \n";
    if ( &apwok( $adid, $cpw, $svar{ 'adminhome' }, $svar{ 'relcgi' } ) ne 'true' )
    {

        # print STDERR "pw bad \n";
        hmsg( "achprpw - password NOT ok", "You must restart", $svar{ 'bomb' }, " ", " " );
    }
    else
    {
        print STDERR "achprpw -- cookie is ok \n";

        my $fpath = "$svar{ 'userhome' }" . "/" . "$profid";

        # print "Content-type: text/html\n\n";
        print $query->header();

        print qq|
<HTML>
<HEAD><TITLE>$adid achprpw</TITLE></HEAD>
<LINK REL="stylesheet" HREF="$svar{ 'relhtml' }/style.css">
<BODY>

<TABLE BORDER=0>
<TR><TD VALIGN="CENTER"><IMG SRC="$svar{ 'projectlogo' }"
HEIGHT=71>&nbsp;&nbsp;&nbsp;</TD>
<TD> <FONT SIZE=+2>
    <B>ETPRL Administrator change of professor pw for <FONT
color="red">$profid</FONT>
    <br> </FONT></TD></TR>
</TABLE>

|;

```

```

if ( $action eq 'store' )
{

#    print STDERR "achprpw -- here we would store the encrypted pw \n";
my $file = "$fpath" . "/" . "c-p-w";

#    print STDERR "filename = $file \n";
#    if (-e $file) {
#        print STDERR "prof pw file exists -- do a confirm \n";
#    }

my $pw = $query->param( 'pw' );
my $pw2 = $query->param( 'pw2' );

# added by Richard Zhang, 2005/8
# To fix this problem: when user do not enter any password, the
password still be changed.
    if ( $pw2 eq "" or $pw eq "" )
    {
my $next = "/$svar{'relogi'}/admin/achprpw.pl"; # this file!
my $action = ""; # null the action
msg( 'Passwords should not be empty', 'Try again', $next, $adid, ", " );
    }

if ( $pw2 ne $pw )
{
my $next = "/$svar{'relogi'}/admin/achprpw.pl"; # this file!
my $action = ""; # null the action
msg( 'Passwords do not match', 'Try again', $next, $adid, ", " );
}
else
{
#seem to need else!
# Moved here by Richard Zhang, 2005/8
# we should close file after open file that.
open( FILE, ">$file" ) or die ( "unabel to open $file " );
$cpw = crypt( "$pw", "ET" );

#    print STDERR "pw=$pw, cpw=$cpw \n";
$pw = '*****'; # clear for security
print FILE "$cpw";
close( FILE );

print qq|

```

<h3>password reset</h3>


```

        <!--options-->
        <TABLE bgcolor=#C0C0C0 cellspacing=0 cellpadding=0
border=0 align=center>
        <TR><TD>
        <TABLE border=0 cellspacing=1 cellpadding=2 width=50>
        <TR bgcolor=#F0F0F0>
        <TD>
        <INPUT type='hidden' name='adid' value="$adid">
        <INPUT type='hidden' name='profid' value="$profid">
        <INPUT type='hidden' name='action' value='store'>
        <INPUT type='submit' value='Done'>
        </TD>
        </TR></FONT>
        </TABLE>
        </TD></TR>
        </TABLE>
        </TD>
        </TR></TABLE>
        </FORM>

    };
}
&boiler( $svar{ 'machname' }, $svar{ 'relhtml' } );

print qq|
        </body></html>
    |;
}
}
1; # end of routine
##### end of file #####

```

7.1.C astorpw.pl

```

#!/usr/bin/perl

# Copyright 2002 J C Nash jcnash@uottawa.ca

## 2002-7-21 astorpw.pl -- store new admin password
# Modified by Richard, 2005.8
# To fix the password change problems

use strict;

print STDERR "==== astorpw ====\n";

use CGI;
require "./base.pl"; # Get the cgi parameters

```

```

require "./errmsg.pl"; #
require "./apwok.pl"; # admin password check routine

my %svar = &base();

my $query = new CGI;
my $adid = $query->param( 'adid' );
my $action = $query->param( 'action' );

if ( ( $adid eq "" ) or ( $action ne 'astorpw' ) )
{
    # bad input
    hmsg( "astorpw - admin id null or action not storeppw", "You must restart properly",
"/$svar{'relcgi'}/admin/aclrpw.pl", "", "", "" );
}
else
{
    # adid ok
    my $cpw = $query->cookie( 'adpw' );

    # print STDERR "astorpw - retrieved cookie cpw = $cpw \n";
    if ( &apwok( $adid, $cpw, $svar{ 'adminhome' }, $svar{ 'relcgi' } ) ne 'true' )
    {

        # print STDERR "pw bad \n";
        my $next = "/$svar{'relcgi'}/admin/anewapw.pl";
        hmsg( "astorpw - password NOT ok", "You must restart",
"/$svar{'relcgi'}/admin/aclrpw.pl", "", "", "" );
    }
    else
    {
        # do update choice
        my $pw = $query->param( 'adpw' );
        my $pw2 = $query->param( 'adpw2' );

        # print STDERR "pstorpw: adpw=$pw, adpw2=$pw2 \n";

        # added by Richard Zhang, 2005/8
        # To fix this problem: when user do not enter any password, the password
still be changed.
        if ( $pw2 eq "" or $pw eq "" )
        {
            my $next = "/$svar{'relcgi'}/admin/anewapw.pl";
            hmsg( "Password can not be empty", "Try again", $next, $adid, "", "" );
        }

        if ( $pw2 ne $pw )
        {
            my $next = "/$svar{'relcgi'}/admin/anewapw.pl";

```

```

    hmsg( "Password entries do not match", "Try again", $next, $adid, ", " );
}
else
{
    # do the store
    my $file = "$svar{'adminhome'}" . "/" . "$adid" . "pw";

    # print STDERR "astorpw: filename = $file \n";
    if ( -e $file )
    {

        # print STDERR "apw file exists -- do a confirm \n";
    }
    open( FILE, ">$file" ) or die ( "unabel to open $file " );
    $cpw = crypt( "$pw", "ET" );

    # print STDERR "pw=$pw, cpw=$cpw \n";
    $pw = '*****'; # clear for security
    print FILE "$cpw";
    close( FILE );
    my $next = "/$svar{'relcgi'}/admin/alogin.pl";
    $adid = ""; # null before login!!
    hmsg( "Password reset", "Continue with new login", $next, $adid, ", " );
}
}
}
1; # end of main
##### end of file #####

```

7.1.D pstorpw.pl

```

#!/usr/bin/perl

## 2002-7-21 pstorpw.pl -- store new professor password
# Modified by Richard, 2005.8
# To fix the password change problems

use strict;

print STDERR "=== pstorpw ===\n";

use CGI;
require "../base.pl";
require "../boiler.pl"; # boilerplate text
require "../errormsg.pl"; # error routine
require "../ppwok.pl"; # professor password check

# Get the cgi parameters
my %svar=&base();

```

```

my $query = new CGI;
my $profid = $query->param('profid');
my $taskname = $query->param('taskname');
my $action = $query->param('action');

print STDERR "pstorpw: profid=$profid, action=$action \n";

if (($profid eq "") or ($taskname ne "") or ($action ne 'storeppw')) { # bad input
    hmsg("pnewppw - professor id null or taskname not null or action not storeppw",
        "You must restart properly", "$svar{'relcgi'}/prof/pclrpw.pl",
        "", "");
}
else { # profid etc. ok
my $cpw = $query->cookie('profpwc');
print STDERR "pchkw - retrieved cookie cpw = $cpw \n";
if (&ppwok($svar{'userhome'}, $profid, $cpw) ne 'true') {
print STDERR "pw bad \n";
    hmsg("pstorpw - password NOT ok", "You must restart",
        "$svar{'relcgi'}/prof/pclrpw.pl", "", "");
}
}
else { # do update choice
    my $pw = $query->param('profpw');
    my $pw2 = $query->param('ppw2');
#    print STDERR "pstorpw: profpw=$pw, pw2=$pw2 \n";

# added by Richard Zhang, 2005/8
# To fix this problem: when user do not enter any password, the password still be
changed.
    if ( $pw2 eq "" or $pw eq "" )
    {
        my $next="$svar{'relcgi'}/prof/pnewppw.pl";
        hmsg("Password should not be empty", "Try again", $next, "", $profid, "");
    }
    if ($pw2 ne $pw) {
        my $next="$svar{'relcgi'}/prof/pnewppw.pl";
        hmsg("Password entries do not match", "Try again", $next, "", $profid, "");
    }
    else { # do the store
my $fpath="$svar{'userhome'}" . "/" . "$profid";
#    print STDERR "pstorpw -- here we store the encrypted pw \n";
my $file = "$fpath" . "/" . "c-p-w";
#    print STDERR "filename = $file \n";
if (-e $file) {
#    print STDERR "cpw file exists -- do a confirm \n";
}
}
}

```

```

        open(FILE,">$file") or die("unable to open $file ");
        $cpw=crypt("$pw","ET");
#       print STDERR "pw=$pw, cpw=$cpw \n";
        $pw='*****'; # clear for security
        print FILE "$cpw";
        close(FILE);
        my $next = "/$svar{'relcgi'}/prof/plogin.pl";
        $profid=""; # null before login!!
        hmsg("Password reset", "Continue with new login", $next, ", $profid, ");
    }
}
} # else profid etc.
1; # end of main
##### end of file #####

```

7.1.E sgetfile.pl

```

#!/usr/bin/perl

# Modified by Richard, 2005.8
# To fix the file upload problems

use strict;
use CGI;
use File::Copy; # 020923jn

require "./base.pl"; # get cgidir and htmldir strings
require "./boiler.pl"; # boilerplate

print STDERR "getfile started \n";

# Print out the HTML header ## CHANGED LOCATION OF style.css 020527
my %svar = &base();

my $query = new CGI;
my $profid = $query->param( 'profid' );
my $taskname = $query->param( 'taskname' );

print STDERR "profid=$profid, taskname=$taskname \n";

my ( $ok, $user, $firstn, $lastn ) = &check_login( $profid, $taskname );

if ( $ok eq 'true' )
{

    print $query->header(); # 020920 - added

    #print qq|

```

```

# <HTML>
# <HEAD>
# <TITLE>ETPRL File Upload...</TITLE>
# <LINK REL="stylesheet" HREF="$svar{'relhtml'}/style.css">
# </HEAD>
# <BODY>
#|;

# Retrieve any queries the httpd server may have retrieved from forms
# In this case, the filename

my $userfile = $query->param( 'userfile' ); #get filehandle to file uploaded by user
# added by Richard Zhang, 2005/8
# To fix this problem: when user leave the file name blank, it still be uploaded
if ( $userfile eq "" )
{
    print qq|
        <HTML>
        <HEAD>
        |;

        print("Filename should not be empty");
        print("<BR>");
        print("Please Try again");
        print("<BR>");
        print qq|
            <A
    HREF="/$svar{'relcgi'}/sgetfnam.pl?profid=$profid&taskname=$taskname">Return to
    student menu</A>
        </BODY></HTML>
        |;
    die;
}
# added by Richard Zhang, 2005/8
# To fix this problem: when user put some special character in filename, like #@ $
etc.
# I made this program before, but I can not find it now, so need to do this later
# TO DO ...

binmode( $userfile ); #switch file to binary mode

# Put the filename into 'ext', and strip everything before the extension.
# $ext now contains the extension of the filename.
my $ext = $userfile;
$ext =~ s/^\.*\./;

```

```

# Now, put the userfile (which contains full path to file on user's computer
# into $filename, and use a complex regexp to remove the path from it,
# leaving the bare filename.
# The regexp is complicated, because Windows will send a path with '\',
# Unix will send a path with '/', and Macs will send a path with ':'.
my $filename = $userfile;

print STDERR "getfile: filename=userfile = $filename \n";

$filename =~ s/(^.*(\\|\/|:))//; #get rid of path

print STDERR "filename after path removal = $filename \n";

# $user is the student's student_number as input in the login and
# passed by the inc_login.pl routine. It is pre-pended to the
# filename.
$filename = "$user-$filename";

print STDERR "getfile final filename = $filename \n";

# Strip the extension off the file. ?? This is stripping the PATH, not the extension
# This is done because things are added to the filename that need
# to go before the extension. Then, the extension that was taken
# earlier will be stuck back on.
my $bare_filename = $filename;
$bare_filename =~ s/\.*$//;

print STDERR "bare_filename = $bare_filename \n";

## make dirname here!!
# We make an "overwritten" directory that will hold files that people
# have resubmitted.
my $dir_name = $svar{ 'userhome' } . '/' . $profid . '/' . $taskname . '/files/';
my $dir_name_overwritten = $dir_name . 'overwritten/';

print STDERR "getfile dir_name = $dir_name \n";

# Make the main course directory IF it does not exist, OR print an error
# (due to permission problems, filesystems problems, etc).
# The permissions are 751 to allow people of the same group to manipulate the
# files.
if ( !-e $dir_name )
{
## Windows may need permissions removed
mkdir( $dir_name, 0751 ) || enddirerror();
}

```

```

##   mkdir($dir_name, 0751) || enddirerror();
}

# Same as above, for the 'overwritten' directory.
if ( !-e $dir_name_overwritten )
{
  mkdir( $dir_name_overwritten, 0751 ) || enddirerror();
}

# Check if the file the user uploaded already exists. This will only report
# files THIS user has uploaded, because of the prepended student number.
if ( -e "$dir_name$filename" )
{
  print STDERR "$dir_name$filename exists \n";

  # Make a filename for the overwritten file in the 'overwritten' dir.
  # Append a timestamp, so that the file can be re-uploaded many times.
  my $overwritten_filename = "$dir_name_overwritten${bare_filename}_" .
get_timestamp() . ".$sxt";
  print STDERR "ofilename=   $overwritten_filename \n";

  # Inform the user that the file has already been submitted, and that
  # the old one was archived.
  print "<FONT COLOR='Red'>You have already uploaded this file.<BR>";
  print "The last file was archived in case this upload was
accidental.</FONT><BR><BR>";

  # Rename/Move the old file to the 'overwritten' directory.
##?? does not seem to work right in Abyss
  # rename("$dir_name$filename", $overwritten_filename);
  copy( "$dir_name$filename", $overwritten_filename );
  unlink( "$dir_name$filename" );
}

# Open the local file for writing, thus the ">". Do a loop which reads the
# remote file in 16K chunks and writes into the local file. Close the file.

print STDERR "sgetfile: try writing $dir_name$filename\n";

##? try changing the slashes??

$_ = "$dir_name$filename";
## s|/|\g; #020926 try without this??
my $file = $_;
print STDERR "changed to $file \n";

```

```

open( FILE, ">$tfile" );
binmode( FILE ); # jn 020826
while ( read( $userfile, my $buffer, 16384 ) )
{
    print FILE $buffer;
}

close( FILE );

# Print a success message. No error message is needed, since if there is an
# error, the browser will inform the user.

print qq|
    Your file,<BR><B>$userfile</B><BR>has been successfully uploaded to <BR>
    the task <B>$taskname</B> of professor <B>$profid</B>.<BR><BR>
    Please refrain from uploading this file again.<BR><BR>
    <A
HREF="/$svar{'relcgi'}/sgetfnam.pl?profid=$profid&taskname=$taskname">Return to
student menu</A>
    </BODY></HTML>
|;

} # end if (check_login)
1; # end of program

# ===== subs ===== #

# This subroutine creates a timestamp in the ISO form:
# yearmonthday-hourminsec
# We make use of the perl time functions on the server.
sub get_timestamp
{
    my ( $sec, $min, $hour, $mday, $mon, $year, $yday, $isdst ) =
localtime( time() );
    $sec = "0$sec" if ( length( $sec ) < 2 );
    $min = "0$min" if ( length( $min ) < 2 );
    $hour = "0$hour" if ( length( $hour ) < 2 );
    $mon = "0$mon" if ( length( $mon ) < 2 );
    $mday = "0$mday" if ( length( $mday ) < 2 );
    $year += 1900;
    return "$year$mon$mday-$hour$min$sec";
}

sub enddirerror
{
    print "<FONT COLOR='Red'>Error: Cannot make directory: $!</FONT><BR>\n";
}

```

```

    print "</BODY></HTML>";
    die;
}

# 2001-9-23 inc_login.pl routine for etutor homework upload
## modified 2002-4-7 JN for simplification
# using simple perl scrips for Apache and compatible servers.
# Instigated by J C Nash and implemented by Pat Suwalski
# and Markus Svilans.
#
# This file is called by any perl script that is password-protected.
# It is designed verify that a login cookie was set, and if not, it
# displays the login screen. Being included at the beginning of each
# secure file, it prevents people who have not logged in from getting
# at any restricted area.

#S print STDERR "Starting inc_login.pl \n";

## subroutines

sub check_login
{
    my ( $profid, $taskname ) = @_;

##    print "Content-type: text/html\n\n";

    my ( $firstn, $lastn );

    # Get the cookie into a hash
    my $stidpw = $query->cookie( 'stidpw' );

    #    print STDERR "sgetfnam/check_login: stidpw = $stidpw\n";

    # now try to extract the bits
    my ( $user, $password ) = split ( '::', $stidpw );

    # print STDERR "sgetfnam: user = $user, password=$password, profid=$profid,
taskname=$taskname \n";

    # See if there was any information in the cookie
    if ( !$user or !$password )
    {
        &clearcookie();
        hmsg( "Missing userid or password", "Try again",
"/$svar{'relcgi'}/slogin.pl?profid=$profid&taskname=$taskname", ", $profid,
$taskname );
    }
}

```

```

}
else
{

# Open the student list and dump contents into an array
my $sfile = "$svar{'userhome'}/$profid/$taskname/students.dat";

# print STDERR "sfile = $sfile\n";
# Do we need to change for Windows?? -- see if it works!!
open( STUDENTLIST, $sfile );
my @students = <STUDENTLIST>;
close STUDENTLIST;

# print STDERR "students=@students\n";
my $ok = 'false';
$firstn = "";
$lastn = "";

# Validate the login
# Loop through the user files until the right combination of user an
# password are found... or not.
foreach my $student ( @students )
{

#           print STDERR "student=$student\n";
my @currentstudent = split ( "\t", $student );
if ( uc( $currentstudent[ 2 ] ) eq uc( $user ) )
{
if ( uc( $currentstudent[ 0 ] ) eq uc( $password ) )
{ # login was successful
## NOTE: here is where we could separate by sections
##           -- not included as at 2002-4-7 JN
$firstn = $currentstudent[ 1 ];
$lastn = $currentstudent[ 0 ];
$ok = 'true';           #could return username and section
my @clretn = ( $ok, $user, $firstn, $lastn );

#           print STDERR "sgetfnam: clretn = @clretn \n";
return @clretn;
}
}
} #foreach
# The user number does not exist, show login.
} # check on student list
} # checklogin
1;

```



```

print qq|
    <FORM ACTION="$nextscript" METHOD="POST">

        <TABLE width=600 cellspacing=0 cellpadding=2 border=0>
        <TR>
        <TD colspan=2 bgcolor="#DDDDDD"><B>ETPRL - Login</B></TD>
        </TR>
        <TR>
        <TD></TD>
        <TD><INPUT TYPE="HIDDEN" NAME="profid"
VALUE="$prof"></TD>
        </TR>
        <TR>
        <TD></TD>
        <TD><INPUT TYPE="HIDDEN" NAME="taskname"
VALUE="$task"></TD>
        </TR>
        <TR>
        <TD>Student Number:&nbsp;</TD>
        <TD><INPUT TYPE="TEXT" NAME="setuser" MAXLENGTH="8"
SIZE="20"></TD>
        </TR>
        <TR>
        <TD>Last Name:&nbsp;</TD>
        <TD><INPUT TYPE="PASSWORD" NAME="setpassword"
MAXLENGTH="25" SIZE="25"></TD>
        </TR>
        <TR>
        <TD colspan=2 align=center><BR><INPUT TYPE="SUBMIT"
VALUE="Login"></TD>
        </TR>
        </TABLE>
    </FORM>

    |;
    &boiler( $svar{ 'machname' }, $svar{ 'relhtml' } );
print qq |
    </BODY></HTML>

    |;
    exit();
}
## end of file ##

```

7.1.E aNewPsw.pl

```

# Get new admin password in case we forget this password.
# admin/test
# upgraded by Richard.Zhang 2005.8
# ETvIZQZL3PVOI

```



```

<B>ETPRL Administrator Password <br> </FONT></TD></TR>
</TABLE>
|;
    my $i = fun1();
    if($i == 1){
        print("Admin password has been reset, here is new one");
        print '<BR>';
        print("Username: admin");
        print '<BR>';
        print("Password: test");
        print '<BR>';
    }

    print '<BR>';
    print '<A HREF="http://127.0.0.1/cgi-bin/t/admin/alogin.pl">Login as an
admin</A>';

    print '</CENTER>';
    &boiler( $svar{ 'machname' }, $svar{ 'relhtml' } );
    print end_html();  ##? do we need to do anything to reload?

```

```
1;
```

```
# end main
#### end of file #####
```

7.1.F INSTAL.In.WinX.Notes.txt

Composed by Richard Zhang, 2005/8

The main steps, please refer to "INSTALL" file.

This file just keep some notes for installation under windows quickly.

Etprl is designed to run under windows + abssy, so here are some steps to install it quickly

if you are using apache, please skip this document, which is not suitable to that.

1. copy all files in this directory to abssy directory
copy etpsetup.05/*.* to c:/Program Files/Abyss Web Server/cgi-bin/etpsetup/
- 2.install etperl.
browser http://localhost/cgi-bin/etpsetup/etsuab.htm
and click submit
3. login etprl
default admin username: admin
default login password: test

Here are 3 kind of login interfaces:

1. Admin login GUI : <http://127.0.0.1/cgi-bin/t/admin/alogin.pl>
after create professor
2. Prof login GUI : <http://127.0.0.1/cgi-bin/t/prof/plogin.pl>
after create professor's task
3. Student login GUI : http://127.0.0.1/cgi-bin/t/slogin.pl?profid=nash&taskname=mid_term

Below is a tool to reset admin password, it is supposed to be only used by admin.
<http://127.0.0.1/cgi-bin/t/admin/aNewPsw.pl>

7.1.G quicklogin.htm

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Quickly login as a different level user</title>
</head>

<body>

<p align="center"><b>Etpri Quickly login as a different level user</b></p>
<p><a href="http://127.0.0.1/cgi-bin/t/admin/alogin.pl">Admin login GUI</a></p>
<p><a href="http://127.0.0.1/cgi-bin/t/prof/plogin.pl">Prof login GUI</a></p>
<p><a href="http://127.0.0.1/t/nashmid_term.htm">Student login GUI</a> or
similar link</p>
<p>Password Tool</p>
<p><a href="http://127.0.0.1/cgi-bin/t/admin/aneupsw.pl">Reset admin psw
tool</a></p>
<p>ii</p>
<p>ii</p>
<hr>
<p align="center">2005 Richard.Zhang, University of Ottawa</p>

</body>

</html>
```

7.1.H zz-abset.pl

```
# flow chart of this setup file , added by zyx
# 1. read parameters from etsuab.htm
# 2. make 5 directories
# C:\Program Files\Abyss Web Server\htdocs\t
# C:\Program Files\Abyss Web Server\htdocs\td
# C:\Program Files\Abyss Web Server\cgi-bin\t
```

```

# C:\Program Files\Abyss Web Server\cgi-bin\t\admin
# C:\Program Files\Abyss Web Server\cgi-bin\t\prof
# 3. create basenew.pl from parameteres in C:\Program Files\Abyss Web Server\cgi-bin\t
# 4. create admin.pw from parameteres in C:\Program Files\Abyss Web Server\cgi-
bin\t\admin
# 5. create startup HTML file: etprlgo.htm in current setup directory. C:\Program
Files\Abyss Web Server\cgi-bin\etpsetup
# 6. create uninstall file: etunin.htm in current setup directory.
# 7. create student upload file: supload.htm in current setup directory.
# 8. copy all files in html.lst and etprlgo.htm, etunin.htm, supload.htm
# to C:\Program Files\Abyss Web Server\htdocs\t
# 9. copy all files in cgi.lst to C:\Program Files\Abyss Web Server\cgi-bin\t
# 10. copy all files in admin.lst to C:\Program Files\Abyss Web Server\cgi-bin\t\admin
# 11. copy all files in prof.lst to C:\Program Files\Abyss Web Server\cgi-bin\t\prof
# 12. under C:\Program Files\Abyss Web Server\cgi-bin\t
# copy base.pl to base-example.pl
# copy "basenew.pl" to "base.pl"
# copy( "basenew.pl", "admin/base.pl" );
# copy( "basenew.pl", "prof/base.pl" );
# 13. try to run http://localhost/cgi-bin/t/admin/alogin.pl
#
#!/usr/bin/perl

```

```
# Copyright 2002 J C Nash jcnash@uottawa.ca
```

```
# zz-abset.pl -- to set up an etprl homework system on Abyss server
```

```
use strict;
```

```

# We assume that the zip or tar.gz module for ETPRL has been unpacked
# and we are "in" the appropriate temporary directory ON THE SERVER.
# Then load the file etprlsetup.htm into a suitable web browser and
# fill in the blanks. On submission, the form will execute this
# script and set up a system.

```

```
# Get the cgi parameters
```

```

use CGI qw(:standard); # main cgi.pm module
use File::Copy; # allows file copying
use Cwd; # current working directory

```

```
my $setupdir = cwd; # need this for copying later
```

```
# print STDERR "setupdir = $setupdir \n";
```

```
my $query = new CGI;
```

```

my $adid = $query->param( 'adid' );
my $adpw = $query->param( 'adpw' );
my $adpw2 = $query->param( 'adpw2' );

if ( $adpw ne $adpw2 )
{
#   my $next = 'etprlsetup.htm';           modified by zyx
  my $next = 'etsuab.htm';
#   print $query->header();               modified by zyx
  zmsg( "Passwords do not match", "Please try again", $next );
}

# no else -- drop through

my $machname = $query->param( 'machname' );
my $windows = $query->param( 'windows' );
my $serverroot = $query->param( 'serverroot' ); # C:\Program Files\Abyss Web
Server\htdocs
my $cgiroot = $query->param( 'cgiroot' ); # C:\Program Files\Abyss Web Server\cgi-
bin
my $relcgi = $query->param( 'relcgi' ); # t, ie. ETPRL Perl script directory relative.
my $etprldir = $query->param( 'etprldir' ); # td , ie. ETPRL professors' directory
my $zip = $query->param( 'zip' );
my $logfile = $query->param( 'logfile' );
my $logopath = $query->param( 'logopath' );
## 020826 -- convert all \ to / to avoid troubles with escape sequences
# There is a cleaner method, but ...

$serverroot = &cslash( $serverroot );
print STDERR "new serverroot = $serverroot\n";

$cgiroot = &cslash( $cgiroot );
print STDERR "new cgiroot = $cgiroot\n";

$logofile = &cslash( $logfile );
print STDERR "new logfile = $logfile\n";

$logopath = &cslash( $logopath );
print STDERR "new logopath = $logopath\n";

my $xcgi = $relcgi; # to keep value
my $retprldir = $etprldir; # relative form of dir
$relcgi = "cgi-bin/$xcgi"; # Must remember cgi-bin, but leave off slash

#   print STDERR "setup.pl: windows = $windows \n";

```

```

# stuff works with slash as normal
my $cgidir = "$cgiroot/$xcgi";          # C:\Program Files\Abyss Web Server\cgi-
bin\t
my $htmldir = "$serverroot/$xcgi"; # C:\Program Files\Abyss Web Server\htdocs\t
my $relhtml = "$xcgi";          # again no slash ??

#Note: we use same name for relative HTML and CGI-BIN directories
# for ETPRL files.
my $userhome = "$serverroot/$etprldir";
my $reluser = "$retprldir";
my $adminhome = "$cgidir/admin";
my $projectname = 'ETPRL-JN';          # added 2002-7-11
my $bomb = "http://$machname/$relhtml/etprlbad.htm"; # Error page

# ===== #
# Now make directories #

$userhome = "$serverroot/$etprldir";
my $profdir = "$cgidir/prof";# C:\Program Files\Abyss Web Server\cgi-bin\t\prof
my $admindir = "$cgidir/admin"; # C:\Program Files\Abyss Web Server\cgi-bin\t\admin
# print STDERR "About to make $userhome dir \n";
chdir( "$serverroot" );
if ( -e $etprldir ){
    zmsg( "directory $etprldir already exists!", "Stopping", "etprlbad.htm" );
}else{
# C:\Program Files\Abyss Web Server\htdocs\t
mkdir( "$etprldir", 0770 ) or die ( "cannot make $etprldir directory" );
}
if ( -e $xcgi ){
    zmsg( "directory $serverroot/$xcgi already exists!", "Stopping", "etprlbad.htm" );
# print STDERR "directory $relcgi already exists under $serverroot!\n";
}else{
# C:\Program Files\Abyss Web Server\htdocs\t
mkdir( "$xcgi", 0770 ) or die ( "cannot make $relcgi directory under $serverroot" );
}

chdir( "$cgiroot" );
if ( -e $xcgi ){
    zmsg( "directory $cgiroot/$xcgi already exists!", "Stopping", "etprlbad.htm" );
# print STDERR "directory $relcgi already exists under $cgiroot!\n";
}else{
# C:\Program Files\Abyss Web Server\cgi-bin\t
mkdir( "$xcgi", 0770 ) or die ( "cannot make $xcgi directory under $cgiroot" );
}
chdir( "$xcgi" );
if ( -e "prof" ){

```

```

# print STDERR "directory prof already exists!\n";
}else{
# C:\Program Files\Abyss Web Server\cgi-bin\t\prof
mkdir( "prof", 0770 ) or die ( "cannot make $profdir directory" );
}
if( -e "admin" ){
# print STDERR "directory admin already exists!\n";
}else{
# C:\Program Files\Abyss Web Server\cgi-bin\t\admin
mkdir( "admin", 0770 ) or die ( "cannot make $admindir directory" );
}

# ===== #
# Now make base.pl

# print STDERR "done second if \n";

my $file = "$cgidir" . "/" . "basenew.pl";
open( FILE, ">$file" ) or die ( "unabel to open $file " );
print FILE "#!/usr/bin/perl\n\n";
print FILE "# System settings for ETPRL (C) J C Nash 2002 \n\n";
print FILE "sub base {\n";
print FILE "my %sysbase = (\n";
print FILE "\"machname\" => '$machname',\n";
print FILE "\"relcgi\" => '$relcgi',\n";
print FILE "\"etprldir\" => '$etprldir',\n";
print FILE "\"zip\"=> '$zip',\n";
## print FILE "\"$logopath\"=> '$logopath',\n";
print FILE "\"cgidir\" => '$cgidir',\n";
print FILE "\"htmldir\" => '$htmldir',\n";
print FILE "\"relhtml\" => '$relhtml',\n";
print FILE "\"userhome\" => '$userhome',\n";
print FILE "\"reluser\" => '$retprldir',\n";
print FILE "\"adminhome\" => '$adminhome',\n";
print FILE "\"projectname\" => 'ETPRL-JN',\n";
print FILE "\"projectlogo\" => '$relhtml/$logfile',\n"; # leading slash rem. 020925
print FILE "\"bomb\" => 'http://$machname/$relhtml/etprlbad.htm',\n";
print FILE "\"uid\" => ,\n";
print FILE "\"gid\" => \n";
print FILE ");\n";
print FILE "#Note: we use same name for relative HTML and CGI-BIN directories\n";
print FILE "# for ETPRL files,\n";
print FILE " return %sysbase;\n";
print FILE "}\n";
print FILE "## end of base.pl ##\n";
print FILE "1;\n";

```

```

close( FILE );

# =====#
# Now make admin pw file file #
# =====#
my $fpath = "$adminhome";
$file = "$fpath" . "/" . "$adid" . "pw";
if ( -e $file ){
    # print STDERR "adminpw file exists -- do a confirm \n";
}
open( FILE, ">$file" ) or die ( "unabel to open $file " );
my $cpw = crypt( "$adpw", "ET" );
print FILE "$cpw";
close( FILE );

# =====#
# Now make startup HTML file #
# etprlgo.htm #
# =====#

chdir( "$setupdir" ); # make sure we are in the working directory
open( FILE, ">etprlgo.htm" ) or die ( "Cannot write etprlgo.htm" );
print FILE "<HTML><HEAD><TITLE>etprlgo.htm</TITLE></HEAD>\n";
print FILE "<BODY bgcolor=\"#DDF0F0\">\n";
print FILE "<h2>ETPRL - point of departure screen</h2>\n";
print FILE "<p>\n";
print FILE "<A HREF=\"http://$machname/$relcgi/admin/alogin.pl\">\n";
print FILE "Admin login</A>\n";
print FILE "<BR><BR>\n";
print FILE " <BR><HR><BR>\n";
print FILE " The ET-Perl homework/file uploader was developed as part of the\n";
print FILE " E-Tutor project of Prof. J C Nash, School of Management, University
of\n";
print FILE " Ottawa, supported by a Teaching and Learning Grant. The program
code\n";
print FILE " for ET-Perl is Copyright J C Nash, P Suwalski and M Svilans (C) 2001,\n";
print FILE " and is available under the Gnu Public Licence.<BR>\n";
print FILE " Original programming by Patrick Suwalski and Markus Svilans<BR>\n";
print FILE " Refinements and simplification to ETPRL version by J Nash.<BR>\n";
print FILE " For information contact <A\n";
print FILE " HREF=mailto:\"jcnash\@uottawa.ca\">jcnash\@uottawa.ca</A></P>\n";
print FILE "</BODY></HTML>\n";
close( FILE );

#?? probably don't need this with zz-unins.pl
# =====#

```

```

# Now make reset HTML file  #
#  etpreset.htm           #
# =====#

## chdir("$setupdir"); # make sure we are in the working directory
## open (FILE, ">etpreset.htm") or die("Cannot write etpreset.htm");
## print FILE "<HTML><HEAD><TITLE>etpreset.htm</TITLE></HEAD>\n";
## print FILE "<BODY bgcolor=\"#DDF0F0\">\n";
## print FILE "<h2>ETPRL - delete all directories</h2>\n";
## # print FILE "<P>\n";
## print FILE "<FONT color=\"red\"><h1>NOTE!! DANGEROUS!!</h1></FONT>";
## print FILE "<p>\n";
## print FILE "<A HREF=\"http://$machname/$relcgi/admin/zz-reset.pl\">\n";
## print FILE "Kill the directories!</A>\n";
## print FILE "<BR><BR>\n";
## print FILE "<BR><hr>\n";
## print FILE "</BODY></HTML>\n";
## close(FILE);

# =====#
# Now make uninstall file  #
#  etunin.htm             #
# =====#

chdir( "$setupdir" ); # make sure we are in the working directory
open( FILE, ">etunin.htm" ) or die ( "Cannot write etunin.htm" );
print FILE "<HTML><HEAD><TITLE>etunin.htm</TITLE></HEAD>\n";
print FILE "<BODY bgcolor=\"#DDF0F0\">\n";
print FILE "<h2>ETUNIN - delete all directories</h2>\n";

# print FILE "<P>\n";
print FILE "<FONT color=\"red\"><h1>NOTE!! DANGEROUS!!</h1></FONT>";
print FILE "<p>\n";
if ( $windows eq 'on' )
{
    # in Windows execute via machine
    print FILE "<A HREF=\"http://127.0.0.1/cgi-bin/t/admin/zz-unins.pl\">\n";
}
else
{
    # in Linux, presume that we can execute directly without "machine"

    # BUT cannot execute via httpd because of security ??
    print FILE "<A HREF=\"$cgidir/admin/zz-unins.pl\">\n";
}
print FILE "Uninstall now</A>\n";
print FILE "<BR><BR>\n";
print FILE "<BR><hr>\n";

```

```

print FILE "</BODY></HTML>\n";
close( FILE );

# =====#
# Now make student upload file #
#  supload.htm      #
# =====#

chdir( "$setupdir" ); # make sure we are in the working directory
open( FILE, ">supload.htm" ) or die ( "Cannot write etunin.htm" );
print FILE "<HTML><HEAD><TITLE>supload.htm</TITLE></HEAD>\n";
print FILE "<BODY bgcolor=\"#DDF0F0\">\n";
print FILE "<h2>SUPLOAD - Links to professor/assignment upload processes</h2>\n";

# print FILE "<P>\n";
print FILE "<FONT color=\"red\"><h3>NOTE!! You must be on the student
list!</h1></FONT>";
print FILE "<p>\n";
print FILE "<UL>\n";
print FILE "</UL>\n";
print FILE "<BR>\n";
print FILE "<BR><hr>\n";
print FILE "</BODY></HTML>\n";
close( FILE );

# ===== #
# Now copy files #
# =====

chdir( $setupdir );
$file = "html.lst";
open( FILE, "<$file" ) or die ( "unable to open $file " );
my @flist = <FILE>;

# print STDERR "@flist \n";
foreach my $f ( @flist )
{
  chomp( $f );
  copy( $f, $htmldir ) or die ( "could not copy $f" );

  # print STDERR "copied $f \n";
}
copy( "etprlgo.htm", $htmldir ) or die ( "could not copy etprlgo.htm" );
# ?? copy ( "etpreset.htm", $htmldir ) or die("could not copy etpreset.htm");
copy( "etunin.htm", $htmldir );
copy( "supload.htm", $htmldir );

```

```

# ?? unlink these
## unlink("etunin.htm"):
## unlink("supload.htm"):

$_ = $logopath;
s/^s//g;
$logopath = $_;
if ( $logopath eq " )
{
    $file = $logfile;
}
else
{
    $file = "$logopath/$logfile";
}
copy( $file, $cgidir ) or die ( "unabel to open $file " );
# =====
$file = "cgi.lst";
open( FILE, "<$file" ) or die ( "unabel to open $file " );
@flist = <FILE>;
close( FILE );

#    print STDERR "@flist \n";
foreach my $f ( @flist )
{
    chomp( $f );
    copy( $f, $cgidir ) or die ( "could not copy $f" );

    #    print STDERR "copied $f \n";
}

# =====
$file = "admin.lst";
open( FILE, "<$file" ) or die ( "unabel to open $file " );
@flist = <FILE>;
close( FILE );

#    print STDERR "@flist \n";
foreach my $f ( @flist )
{
    chomp( $f );
    copy( $f, $admindir ) or die ( "could not copy $f" );

    #    print STDERR "copied $f \n";
}

```

```

# =====
$file = "prof.lst";
open( FILE, "<$file" ) or die ( "unabel to open $file " );
@flist = <FILE>;
close( FILE );

#      print STDERR "@flist \n";
foreach my $f ( @flist )
{
    chomp( $f );
    copy( $f, $profdir ) or die ( "could not copy $f" );
    #      print STDERR "copied $f \n";
}

chdir( "$cgidir" ) or die ( "could not cd to $cgidir" );
#copy( "base.pl", "base-example.pl" );
copy( "basenew.pl", "base.pl" );

# and copy into subdirs also
copy( "basenew.pl", "admin/base.pl" );
copy( "basenew.pl", "prof/base.pl" );

# $nxt = "reset.pl";
# zmsg("basenew.pl written and files copied", "try resetting?", $nxt);
my $nxt = "http://$machname/$relcgi/admin/alogin.pl";
#print $query->header();
zmsg( "basenew.pl written and files copied", "try admin login?", $nxt );

# ?? also want a sample etprldir

exit();
1;

# =====
=====#
# Objective: prints an error message
# input: title, message, next
# Output: a html contains error message
# Description: After press OK, goto "next" html;
# added by Richard Zhang.
# March 09, 2004
# =====
=====#
sub zmsg
{
# added by zyx

```

```

    print header();
    print start_html("Error Message by zyx");
# end of added

my ( $title, $msg, $nxt ) = @_; #get parameters
    # $title - message title
    # $msg - message text
my $buttons = "<INPUT type='submit' value='OK'>";
## Note: it is important to have ACTION in the FORM tag to get this to work
## post method gives badly formed request for some reason, so get used
## May be because target cannot process the posted fields in some cases.
print qq|
    <TABLE width=100% height=100% cellspacing=0 cellpadding=0
border=0>
    <TR><TD align=center>

    <FORM METHOD="get" ACTION="$nxt" ENCTYPE="multipart/form-
data">
    <TABLE bgcolor=#C0C0C0 cellspacing=0 cellpadding=0 border=0>
    <TR><TD>

    <TABLE border=0 cellspacing=1 cellpadding=2 width=300>
    <TR bgcolor=#FFF0F0>
    <TD><SPAN class=msg_title><FONT
color=red>$title</FONT></SPAN></TD>
    </TR>
    <TR bgcolor=white>
    <TD align=center><BR>$msg<BR>
    <BR>
    <CENTER>$buttons</CENTER><BR></TD>
    </TR>
    </TABLE>

    </TD>
    </TR>
    </TABLE>
    </FORM>

    </TD>
    </TR></TABLE>
|;

# print STDERR "zmsg nxt = $nxt \n";
print end_html;
# changed by zyx
die;

```

```
1;  
} # end zmsg
```

```
sub cslash  
{ # converts \ to /  
  ( my $dirstr ) = @_;  
  $_ = $dirstr;  
  s|\\|\/|g;  
  $dirstr = $_;  
  return $dirstr;  
  1;  
} # end cslash
```

```
##### end of file #####
```

7.1.I admin.lst

```
aclrpw.pl  
armfiles.pl  
armtask.pl  
aselprof.pl  
alogin.pl  
adownld.pl  
anewapw.pl  
anutask.pl  
achkpw.pl  
astorpw.pl  
achoose.pl  
admprof.pl  
aptskact.pl  
achprpw.pl  
addprof.pl  
apwok.pl  
armdir.pl  
amkbtch.pl  
armbtch.pl  
astlist.pl  
adelprof.pl  
armprof.pl  
base.pl  
boiler.pl  
errormsg.pl  
zz-mkapw.pl  
zz-reset.pl  
zz-setup.pl  
zz-unins.pl  
aNewPsw.pl
```

7.2 Programs of ETPHP

7.2.A Test.php

```
<? phpinfo() ?>
```

7.2.B Setup.php

```
<?php
```

```
// e-Tutor is licensed under the GPL. Please see the LICENSE file for more  
// information.
```

```
/*
```

```
setup.php
```

written January 2nd, 2001 by Pat

Modified August, 2005, by Richard

Function:

Setup program for e-tutor PHP.

Input Parameters:

rootname - MySQL root username

rootpw - MySQL root password

Output:

Sets up random password file and MySQL tables, displays success or error.

```
*/
```

```
?>
```

```
<HTML>
```

```
<HEAD><TITLE>e-tutor PHP Setup</TITLE></HEAD>
```

```
<LINK REL="stylesheet" HREF="style.css">
```

```
<BODY>
```

```
<?php
```

```
if (!isset($rootname)) {
```

```
?>
```

```
<P><FONT SIZE="+2"><B>e-Tutor PHP Setup</B></FONT>
```

```
<P>Before proceeding, please make certain that the directory in which the PHP  
files reside is owned by the web server. On most newer systems this can be done by  
going to that directory and typing:<BR>
```

```
<UL><FONT COLOR="Blue">chown apache:apache ./</FONT></UL>
```

If this does not work, you must find the user and group that apache belongs to, and chown for that.

<P>To complete the setup, simple follow the three steps below.
<HR>

<FORM METHOD=POST ACTION="<?php echo \$PHP_SELF; ?>">
<P><I>What is the name of your Organization?</I>

Organization: <INPUT TYPE="TEXT" SIZE="30" NAME="org"
VALUE="Some Organization">

<P><I>Choose a password for administrating e-Tutor:</I>

Admin Password: <INPUT TYPE="TEXT" MAXLENGTH="12" SIZE="16"
NAME="adminpw">

<P><I>Enter the MySQL admin user and password below.</I>

On most systems the administrator is root.

Note: This is not asking for the root password of the computer.

<FORM METHOD=POST ACTION="<?php echo \$PHP_SELF; ?>">
Root Username: <INPUT TYPE="TEXT" MAXLENGTH="16" SIZE="16"
NAME="rootname" VALUE="root">

Root Password: <INPUT TYPE="PASSWORD" MAXLENGTH="8" SIZE="16"
NAME="rootpw">

<P><INPUT TYPE="submit" VALUE="Perform Install">
</FORM>

```
<?php
}
else {
    if ($rootpw == "") {
        die('No MySQL password given!</BODY></HTML>');
    }

    if ($adminpw == "") {
        die('No administrator password given!</BODY></HTML>');
    }

    // MAKE A PASSWORD
    $random = addslashes(crypt(time()));

    // PASSWORD FILE WRITING
    if (is_file('./passwd')) {
        die('Setup already ran. Remove passwd file and MySQL database
to start a clean setup.');
```

```

    }

    $of = fopen ('./passwd', 'w') or die('Could not open password file for
writing!');
    fputs($of, $random);
    fclose($of);

    chmod('./passwd', 0600) or die('Could not change file mode. Please make
sure Apache has write permissions in this directory');

    print "Password Successfully Created...<BR>";

    // Richard
    // Connect to MySQL
    $dbcnx = @mysql_connect('localhost', $rootname, $rootpw) or die
(mysql_error());

    // MYSQL DATABASE CREATION
    if (@mysql_select_db("etutor")) {
        // Database opened, must already exist
        print "<FONT COLOR='Blue'>Note: Database already existed.
This setup may restore some preferences but not data.</FONT><BR>";
    }
    else {
        // Richard added, @mysql_create_db is too old for mysql 4.0+
        mysql_query("CREATE DATABASE etutor ") or
die(mysql_error());
        mysql_select_db("etutor") or die(mysql_error());
    }

    @mysql_query("CREATE TABLE IF NOT EXISTS user (ID INT NOT
NULL AUTO_INCREMENT PRIMARY KEY, stnum INT NOT NULL, fn CHAR(40)
NOT NULL, ln CHAR(40) NOT NULL, courseID TEXT NOT NULL, sections TEXT
NOT NULL, email TEXT, username CHAR(12) NOT NULL, password CHAR(14) NOT
NULL, account CHAR(2) NOT NULL, pwchanged BOOL NOT NULL)");
    @mysql_query("CREATE TABLE IF NOT EXISTS courses (ID INT
NOT NULL AUTO_INCREMENT PRIMARY KEY, code CHAR(12) NOT NULL,
sections TEXT NOT NULL, title TEXT NOT NULL)");
    @mysql_query("CREATE TABLE IF NOT EXISTS assignments (ID INT
NOT NULL AUTO_INCREMENT PRIMARY KEY, name TEXT NOT NULL,
courseID INT NOT NULL, section CHAR(2) NOT NULL, duedate DATETIME,
maxfilesize INT NOT NULL, maxfiles INT NOT NULL, allowlate BOOL NOT
NULL)");
    @mysql_query("CREATE TABLE IF NOT EXISTS files (ID INT NOT
NULL AUTO_INCREMENT PRIMARY KEY, courseID INT NOT NULL, section
CHAR(2) NOT NULL, assignmentID INT NOT NULL, stnum INT NOT NULL,

```

```

filename TEXT NOT NULL, date timestamp(14) NOT NULL, receipt INT NOT NULL,
mimetype CHAR(128) NOT NULL, data MEDIUMBLOB NOT NULL));
    @mysql_query("CREATE TABLE IF NOT EXISTS prefs (name
CHAR(20) PRIMARY KEY, value TEXT)");

    print "Database Successfully Created...<BR>";

    // MYSQL USER CREATION
    @mysql_query("GRANT ALL PRIVILEGES ON etutor.* TO
etutor@localhost IDENTIFIED BY '$random'") or die(mysql_error());

    print "MySQL User Successfully Created...<BR>";

    // ADMIN PASSWORD
    $adminpw = crypt($adminpw, $adminpw);
    @mysql_query("REPLACE INTO prefs SET name='adminpw',
value='$adminpw'") or die (mysql_error());

    print "Administrator Password Set...<BR>\n";

    // INITIAL PREFERENCES
    print "<BR><B>Initial Preferences...</B><BR>";
    if ($org == "") $org = 'Some Organization';
    @mysql_query("INSERT INTO prefs SET name='organization',
value='$org'") or @mysql_query("UPDATE prefs SET value='$org' WHERE
name='organization'");
    @mysql_query("INSERT INTO prefs SET name='batchorder', value=''' .
addslashes('$sn,$ln,$fn,$se,$em,$un,$pw') . '''") or print "&nbsp;&nbsp;&nbsp;<I>Batch
Order Exists... skipping</I><BR>\n";
    @mysql_query("INSERT INTO prefs SET name='loginformat',
value='namenumber'") or print "&nbsp;&nbsp;&nbsp;<I>Login Format Exists...
skipping</I><BR>\n";
    @mysql_query("INSERT INTO prefs SET name='ccformat', value='') or
print "&nbsp;&nbsp;&nbsp;<I>Course Code Format Exists... skipping</I><BR>\n";
    @mysql_query("INSERT INTO prefs SET name='forcepwchange',
value='0'") or print "&nbsp;&nbsp;&nbsp;<I>Password-Forcing Option Exists...
skipping</I><BR>\n";

    print "Initial Preferences Set...<BR><BR><BR>\n";

    print "<B>DONE</B><BR>Click <A HREF='./>here</A> to Log In.";
}
?>

</BODY>
</HTML>

```

7.2.C etphpxt2.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
  <title>etphpxtract</title>
</head>
<body>
<h1>ETPHP - File Extractor</h1>
<br>
<form method=post action="extract2.php">

<table width="596" cellspacing="1" height="172" cellpadding="2">
<tr>
<td width="201" height="25">
MySQL userid<font color="#FF0000">*</font>.&nbsp;</td>
<td width="539" height="25"><input type=text name="user" value="root" size=32
></td>
</tr>
<tr>
<td width="201" height="25">
Password<font color="#FF0000">*</font></td>
<td width="539" height="25"><input type=password name="password" value="root"
size=20></td>
</tr>
<tr>
<td width="201" height="25">
Starting Assmt ID<font color="#FF0000">*</font>:</td>
<td width="539" height="25"><input type=text name="aid1" size=10></td>
</tr>
<tr>
<td width="201" height="25">
Ending Assmt ID<font color="#FF0000">*</font>:</td>
<td width="539" height="25"><input type=text name="aid2" size=10></td>
</tr>
<tr>
<td width="201" height="25">
Storage directory (rel. to script)<font color="#FF0000">*</font>:</td>
<td width="539" height="25"><input type=text name="savedir" value="savedir"
size=32></td>
</tr>
</table>
<p><input type=submit value="GO!">
<br>
</p>
```

```
<br>
</body>
</html>
```

7.2.D extract2.php

```
<?php
```

```
mysql_connect('localhost', $_REQUEST['user'], $_REQUEST['password']);
mysql_select_db('etutor');
```

```
$a1 = $_REQUEST['aid1'];
$a2 = $_REQUEST['aid2'];
```

```
// zyx
if($a1=="" or $a2==""){
    echo "\nAssignments ID can not be empty\n";
    ShowAssignmentID();
    die;
}
if($a1 > $a2) {
    print "FAILURE -- assignments not in order <BR>";
    ShowAssignmentID();
    die;
}
```

```
$sd = $_REQUEST['savedir'];
print "Saving to $sd \n";
```

```
// echo "\nplease put a valid AssignmentID existed in the following table\n";
```

```
// debug by zyx
//mkdir($sd,0700);
//mkdir($sd."/marked",0700);
```

```
/*
$result = mysql_query("SELECT assignmentID,stnum,filename,data FROM files
WHERE assignmentID>=" . $a1 . " AND assignmentID<=" . $a2 . " ");
*/
```

```
$i = $a1;
```

```
while ($i <= $a2) {
```

```

print "<BR><BR>Assignment ID = " . $i . "<BR> \n";

// zyx
$query = "SELECT assignmentID,stnum,filename,data FROM files WHERE
assignmentID=\"" . $i . "\"";
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

if (mysql_num_rows($result)==0){
    echo "\n Cannot find any info by this assignment ID";
    echo "<br>";
    echo "\n All valid AssignmentID are listed in the following table";
    echo "<br>";
    ShowAssignmentID();
}else{
    print "<B>Files</B><BR>\n";
}

while ($r = mysql_fetch_array($result)) {
    $fn=$r['filename'];
    print 'Filename = ' . $fn . " - - - ";
    // now try to extract extension
    $nextn = strrpos( $fn, ".");
    $extn = substr($fn,$nextn);
    // print 'found hit= ' . $nextn . ' -- ' . $extn . "<br>\n";

    $fn=$r['stnum'] . "-s" . $i . $extn;
    print $fn . " <BR>\n";
    $fh = fopen($sd . '/' . $fn, 'w');
    // fwrite ($fh, $r['data']);
    fclose($fh);
}

//$result = mysql_query("SELECT assignmentID,stnum,filename,data FROM
markedfiles WHERE assignmentID=\"" . $i . "\"");

$query = "SELECT assignmentID,stnum,filename,data FROM markedfiles WHERE
assignmentID=\"" . $i . "\"";
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

print "<B>Marked Files</B><BR>\n";
while ($r = mysql_fetch_array($result)) {
    $fn=$r['filename'];

```

```

    print 'Filename = ' . $fn . " - - - ";
// now try to extract extension
    $nextn = strrpos( $fn, ".");
    $extn = substr($fn,$nextn);
// print 'found hit= ' . $nextn . ' -- ' . $extn . "<br>\n";

    $fn=$r['stnum'] . "-m" . $i . $extn;
    print $fn . " <BR>\n";
    $fh = fopen($sd."/marked/" . $fn, 'w');

//    fwrite ($fh, $r['data']);
    fclose($fh);

}

$i++;
}

print "<BR><BR><A HREF='etphpxt2.php'>Return to selector</a>\n";

// Show Assignment ID and Name in table "assignments"
// Show help for this tool after user gave wrong parameter
// Yuxiang (Richard) Zhang, Jun 19, 2005, University of Ottawa
function ShowAssignmentID()
{
// Performing SQL query
$query = 'SELECT ID, name FROM assignments';
// SELECT ID, name FROM assignments WHERE 1
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

// Printing results in HTML
echo "<table>\n";
echo "\t<tr>\n";
echo "\t\t<td><B>Assignment ID</B></td>";
echo "\t\t<td><B>Assignment Name</B></td>";
echo "\t</tr>\n";

while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

```

```
// Free resultset
mysql_free_result($result);
}
```

```
?>
```

7.2.E exlogin.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1"
  http-equiv="content-type">
  <title>etphpextract</title>
</head>

<body >
<h1>ETPHP - test external login</h1>
<form method=post action="index.php">

<table width="431" cellspacing="1" height="68" cellpadding="2">
<tr>
<td width="201" height="25">
User Name<font color="#FF0000">*:</font>&nbsp;</td>
<td width="374" height="25"><input type=text name="UserName" value="admin"
size=32 ></td>
</tr>
<tr>
<td width="201" height="1">
Password<font color="#FF0000">*:</font></td>
<td width="374" height="1"><input type=password name="password" value="111"
size=20></td>
</tr>

<tr>
<td width="201" height="25">
Professor or Not</td>
<td width="374" height="25"><input type=text name="ProfStud" value="prof" size=32
></td>
</tr>
<tr>
<td width="201" height="25">
Student ID</td>
<td width="374" height="25"><input type=text name="StudentID" value="11111"
size=32 ></td>
</tr>
</tr>
```

```

<tr>
<td width="201" height="25">
Other Information</td>
<td width="374" height="25"><input type="text" name="Others" value="11111" size=32
></td>
</tr>
</table>
<p><input type="submit" value="Login" name="LoginButton">

```

```

<br>
</p>
</form>
<p>stimulate an interface between etphp and other systems</p>
<p>1. etphp redirect to other systems' login </p>
<p>2. get some info back to login etphp </p>

```

```

<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>2005
<p align="center"> </p>

```

```

</body>
</html>

```

7.2.F indexEx.php

```

<? phpinfo() ?>
<?php
// e-Tutor is licensed under the GPL. Please see the LICENSE file for more
// information.

```

```

/*
index.php (was: login.php)

```

written January 5th, 2002 by Pat

Script:

```

    Displays login screen, then verifies login and shows main menu.
    For successful login, sets session-cookie.
*/

```

```

    // zyx added
    // echo $UserName;

```

```

//      $username = " "; //"admin";
// zyx problems here:
// we could not send username & password back to this "index.php", then have to modify
user/psw for each login

//      good for exlogin.php
$username=trim($UserName);
if( $username == ""){
    header("Location: exlogin.php");
}
$password=trim($password);

require("include.php");
doc_start('e-Tutor');
center_start();

echo $UserName;
echo $password;

/*
    $UserID = "admin";
    $UserID = "ta";
    $UserID = "students";
    $UserID = "prof"; //
*/
if (!isset($username)) {
    showLogin('1st time');
    // zyx added
    echo "Username";
    echo $UserName;
    echo "echo before psw key";
    $username = 1; //"admin";
}
else {
    // zyx added
/*
    switch($UserID){
    case "admin":
    // for admin login
        $username = "admin";
        $password = "111";
        break;
    case "prof":
    // for professor login

```

```

        $username = "jnash";
        $password = "jnash";
        break;
    case "ta":
        // for TA login
        $username = "11111";
        $password = "L1";
        break;
    case "students":
        // for Students login
        $username = "21113";
        $password = "1L3";
        break;
    default:
        echo "wrong in the switch";
    }
*/

    print "after psw key";
    $result = @mysql_query("SELECT * FROM user WHERE
username='$username'") or die ("Invalid query");
    $row = mysql_fetch_array($result);

    // Try regular login OR
    // Try student number as un and case-insensitive ln as password
    if (($row["password"] == crypt($password, $row["ln"])) || ($username ==
$row['stnum'] && $row["password"] == crypt(strtoupper($password), $row["ln"]))) {
        // The user has successfully Logged IN!

        $loggedin = 1;
        $id = $row["ID"];
        $stnum = $row["stnum"];
        $first = $row["fn"];
        $last = $row["ln"];
        $course = $row["courseID"];
        $sections = $row["sections"];
        $email = $row["email"];
        $username = $row["username"];
        $cryptpassword = $row["password"];
        $account = $row["account"];

        $user = new User($id); //re-load $user object with information for
user with user ID equal to $id

        // Do check for password change
        passChangeCheck();

```

```

        box_start("", 600);
        ?>
        e-Tutor System<BR>
        <SPAN class="title">Welcome <?php print $user->fn . " " . $user-
>ln?></SPAN><BR>
        <?php
        box_end();
        print "<BR>";

        box_start("", 600);
        $curtime = date("Y-m-d H:i:s");
        print "Current server date and time: $curtime";
        box_end();

        print "<BR>";

        switch ($row["account"]) {
            case 'st':
                showStudentMenu();
                break;
            case 'pr':
                showProfMenu();
                break;
        }
    } elseif ($username == 'admin' && adminCheckPass($password)) {
        box_start("", 600);
        ?>
        e-Tutor System<BR>
        <SPAN class="title">Welcome Administrator</SPAN><BR>
        <?php
        box_end();
        print "<BR>";

        box_start("", 600);
        $curtime = date("Y-m-d H:i:s");
        print "Current server date and time: $curtime";
        box_end();
        print "<BR>";
        showProfMenu();
    }
    else {
        // The login has FAILED
        // zyx
        // $errormessage = "<FONT COLOR='#FF0000'>Invalid
User/Password</FONT><BR>";

```

```

        echo $username;
        echo $password;
        echo "can not login";
//      showLogin('wrong psw');
        //showLogin($errormessage);
        session_unset();
    }
}

center_end();
doc_end();
// zyx
function showLogin($error) {
    global $etversion;

    // Get Organization name
    $prefresult = mysql_query("SELECT value FROM prefs WHERE
name='organization'") or die(mysql_error());
    $org = mysql_fetch_array($prefresult);
?>
// zyx
// we could not send username & password back to this "index.php", then have to modify
user/psw for each login

```

```

<from ACTION="index.php" METHOD="POST">
<?php center_start() ?>
<TABLE CELLPADDING=1 BORDER=0 ALIGN=CENTER>
<TR><TD BGCOLOR="#A0A0A0">
<TABLE BORDER=0 WIDTH=600 CELLSPACING=0>
<TR VALIGN="MIDDLE"><TD><IMG SRC="logo.png" HEIGHT=75
ALT="Logo"></TD><TD><FONT SIZE="+2" COLOR="#005EA0"><B>e-Tutor
Login</B></FONT><BR><B>at <?php echo $org['value']?></B></TD></TR>
</TABLE>
<TABLE width=600 cellpadding=2 border=0>
<TR><TD colspan=2 bgcolor="#DDDDDD"><B>e-Tutor -
Login</B></TD></TR>
<TR><TD colspan=2><B><?php echo
$error; ?>&nbsp;</B></TD></TR>
<TR>
<TD WIDTH="50%" ALIGN="RIGHT">User Name:&nbsp;</TD>
<TD WIDTH="50%"><INPUT TYPE="TEXT" NAME="username"
MAXLENGTH="35" SIZE="14"></TD>
</TR>
<TR>
<TD ALIGN="RIGHT">Password:&nbsp;</TD>

```

```

        <TD><INPUT TYPE="PASSWORD" NAME="password"
MAXLENGTH="14" SIZE="14"></TD>
    </TR>
    <TR>
        <TD colspan=2 align=center><BR><FONT SIZE="-2"><INPUT
TYPE="SUBMIT" VALUE="Login"><BR>
        <BR>
        <B>Note:</B> Your web browser must accept cookies.</FONT></TD>
    </TR>
</TABLE>
</TD></TR></TABLE>
<FONT SIZE="-2" COLOR="#A0A0A0">e-Tutor &copy;2002-2003 J.
Nash, P. Suwalski, M. Svilans.<BR>
Web: <A HREF="http://etutor.sf.net"
CLASS="hiddenlink">http://etutor.sf.net</A><BR>
Version: PHP <?php echo $tversion ?></FONT>
<?php center_end() ?>
</FORM>

```

```

<?php
}

```

```

function showStudentMenu() {
    global $user, $id, $stnum, $first, $last, $course, $sections, $email,
$username, $password, $account, $cryptpassword;
    $courses = split(',', $course);
    $sections = split(',', $sections);
    global $curtime;

    $result = @mysql_query("SELECT ID, code FROM courses") or die
(mysql_error());
    while ( $row = mysql_fetch_array($result) ) {
        $courselist[$row['ID']] = $row['code'];
    }

    box_start("Assignments", 600);

    // Generate the assignment table into $assigntable
    for ($i=0; $i<count($courses); $i++) {
        // Skip over courses where user is a TA
        if (strstr($courses[$i], '*')) {
            continue;
        }
    }
}

```

```

        $result = @mysql_query("SELECT * FROM assignments
WHERE courseID='" . $courses[$i] . "' AND section='" . $csections[$i] . "'") or die
(mysql_error());
        while ( $row = mysql_fetch_array($result) ) {
            // See if assignment is submitted already
            $afile = @mysql_query("SELECT data,filename,date,ID
FROM files WHERE stnum='$stnum' AND assignmentID='" . $row["ID"] . "'") or die
(mysql_error());

            $submitted = mysql_num_rows($afile);
            $status = "";

            if ($submitted > 0) {
                // Get the file size
                while ($fetch = mysql_fetch_array($afile)) {
                    $fieldlen = mysql_fetch_lengths($afile);
                    $status .= '<A HREF="getfile.php?fid=' .
$fetch['ID'] . "'>' . $fetch['filename'] . '</A>&nbsp;&nbsp;(' . number_format(($fieldlen[0] /
1024), 1) . "K)<BR>\n";
                }
            }
            elseif ($curtime > $row["duedate"]) {
                $status .= "<font color=#ff0000>Late</font>";
            }
            if ($submitted && $curtime <= $row["duedate"]) {
                $status .= " [

```

```

class='header'>Assignment</TD><TD class='header'>Deadline</TD><TD
class='header'>Status</TD></TR>\n";
        print $assigntable;
        print "</TABLE></TD></TR></TABLE>\n";
    }
    else {
        print "<I>You have no assignments</I>.";
    }

    box_end(); //end assignments box
    print "<P>";

    if (strstr($course, '*'))
        showTAMenu();

    require('plugins.php');
    if ($user->account == 'st' && $user->hasTACourses) {
        printPlugInIcons('ta');
    }
    else {
        printPlugInIcons($user->account);
    }

    box_start("Options", 600);
    ?>

    <TABLE width=66%>
    <TR>
    <TD align=center width=50%><A href="userprefs.php"><IMG
src="img/prefs.png" width=48 height=48 border=0><BR>User Preferences</A></TD>
    <TD align=center width=50%><A href="logout.php"><IMG
src="img/logout.png" width=48 height=48 border=0><BR>Log Out</A></TD>
    </TR>
    </TABLE>

    <?php
    box_end();
}

function showTAMenu() {
    global $user;

    box_start("Assignment Files", 600);

    ?>
    Click an assignment below to view the submitted student files.<BR>

```

```

<BR>

<?php
    if (checkadmin()) { //show all assignments to administrator
        $result = @mysql_query("SELECT * FROM assignments ORDER
BY duedate") or die (mysql_error());

        ?>
        <TABLE CELLPADDING=0>
        <TR>
        <TD BGCOLOR='#A0A0A0'>

        <TABLE WIDTH="450" CELLSPACING="1" BORDER="0"
CELLPADDING="2">
        <TR>
        <TD class='header'>Name</TD>
        <TD class='header'>Course</TD>
        <TD class='header'>Due Date</TD>
        </TR>

        <?php

        if (mysql_num_rows($result)) {
            while ($row = mysql_fetch_array($result)) {
                $temp = new Course($row["courseID"]);
                ?>

                <TR>
                <TD><A HREF='listfiles.php?aid=<?php print
$row['ID']?>'><?php print $row['name']?></A></TD>
                <TD><?php print $temp->code . " " .
$row['section']?></TD>
                <TD>Due: <?php print $row['duedate']?></TD>
                </TR>

                <?php

            }
        }
        ?>

        </TABLE>
        </TD>
        </TR>
        </TABLE>
        <?php

```

```

    } else {
        foreach ($user->courses as $courses) {
            if (($courses->isTA) || ($user->account == 'pr')) {

                $result = @mysql_query("SELECT * FROM
assignments WHERE courseID='$courses->ID' AND section='$courses->section'
ORDER BY duedate") or die (mysql_error());

                if (mysql_num_rows($result)) {

                    ?>

                    <TABLE CELLPADDING=0>
                    <TR>
                    <TD BGCOLOR='#A0A0A0'>

                        <TABLE WIDTH="450"
CELLSPACING="1" BORDER="0" CELLPADDING="2">
                            <TR>
                            <TD COLSPAN='2' class='header'><?php
print "$courses->code$courses->section - $courses->title" ?></TD>
                            </TR>

                                <?php
                                while ($row = mysql_fetch_array($result)) {
                                    print "<TR><TD width='50%'><A
HREF='listfiles.php?aid=" . $row['ID'] . "'>" . $row['name'] . "</A></TD><TD
width='50%'>Due: " . $row['duedate'] . "</TD></TR>\n";
                                }
                                ?>

                                </TABLE>
                                </TD>
                                </TR>
                                </TABLE>
                                <?php
                                }
                            }
                        }
                    }

                box_end();
                print '<BR>';
            }
        }
    }

```

```

function showProfMenu() {
    global $user;

    box_start("Main Menu", 600);
    ?>

    <TABLE cellspacing=0 cellpadding=0 border=0 width=100%>
    <TR>
        <TD align=center width=33%><A href="courses.php"><IMG
src="img/courses.png" width=48 height=48 border=0><BR>Manage
Courses</A></TD>
        <TD align=center width=34%><A href="users.php"><IMG
src="img/users.png" width=48 height=48 border=0><BR>Manage Users</A></TD>
        <TD align=center width=33%><A href="assignments.php"><IMG
src="img/assignments.png" width=48 height=48 border=0><BR>Course
Assignments</A></TD>
    </TR>
    <TR>
        <TD colspan=3>&nbsp;<BR>&nbsp;<BR></TD>
    </TR>
    <TR>
        <TD align=center><A href="userprefs.php"><IMG src="img/prefs.png"
width=48 height=48 border=0><BR>Preferences</A></TD>
        <TD align=center><A href="logout.php"><IMG src="img/logout.png"
width=48 height=48 border=0><BR>Log Out</A></TD>
        <TD align=center>&nbsp;</TD>
    </TR>
    </TABLE>

    <?php

    box_end();
    print "<P>";
    require('plugins.php');
    printPlugInIcons($user->account);
    showTAMenu();
}

function passChangeCheck() {
    global $user;

    // Check if the password is even to be changed
    $prefresult = mysql_query("SELECT value FROM prefs WHERE
name='forcepwchange'") or die(mysql_error());
    $pref = mysql_fetch_array($prefresult);
    if (!$pref['value']) return;
}

```

```

        // Check that the user hasn't changed their password
        $userpwresult = mysql_query("SELECT pwchanged FROM user WHERE
ID="" . $user->ID . """) or die(mysql_error());
        $userpw = mysql_fetch_array($userpwresult);
        if ($userpw['pwchanged']) return;

        // If we got this far, force password change
        box_start("Forced Password Change", 600);
        ?>
        <TABLE width="100%" cellspacing="2" cellpadding="0">
        <TR>
        <TD width="64" rowspan="4"><IMG src="img/password.png" width=48
height=48></TD>
        <TD COLSPAN=2><P>Welcome to e-Tutor. As a security measure, you
will now be asked to change your default password to one of your own choosing.
        <FORM ACTION="userprefs.php" METHOD="POST">
        <INPUT TYPE="HIDDEN" NAME="action" VALUE="changepw">
        <INPUT TYPE="HIDDEN" NAME="forcepwchange" VALUE="1">
        </TD></TR>
        <TR><TD ALIGN="RIGHT" WIDTH="40%">New
Password:</TD><TD><INPUT TYPE="PASSWORD" SIZE="8" MAXLENGTH=8
NAME="pw1 "></TD></TR>
        <TR><TD ALIGN="RIGHT">Confirm New
Password:</TD><TD><INPUT TYPE="PASSWORD" SIZE="8" MAXLENGTH=8
NAME="pw2"></TD></TR>
        <TR><TD COLSPAN="2" ALIGN=RIGHT><BR><INPUT
TYPE="SUBMIT" VALUE="Save Password"></FORM></TD></TR>
        </TABLE>
        <?php
        box_end();
        doc_end();
        exit();
        }

?>
<?php

mysql_connect('localhost', $_REQUEST['user'], $_REQUEST['password']);
mysql_select_db('etutor');

$a1 = $_REQUEST['aid1'];
$a2 = $_REQUEST['aid2'];

// zyx
if($a1=="" or $a2==""){

```

```

        echo "\nAssignments ID can not be empty\n";
        ShowAssignmentID();
        die;
    }
    if ($a1 > $a2) {
        print "FAILURE -- assignments not in order <BR>";
        ShowAssignmentID();
        die;
    }

    $sd = $_REQUEST['savedir'];
    print "Saving to $sd \n";

    // echo "\nplease put a valid AssignmentID existed in the following table\n";

    // debug by zyx
    //mkdir($sd,0700);
    //mkdir($sd."/marked",0700);

    /*
    $result = mysql_query("SELECT assignmentID,stnum,filename,data FROM files
    WHERE assignmentID>=" . $a1 . " AND assignmentID<=" . $a2 . "" );
    */

    $i = $a1;

    while ($i <= $a2) {

        print "<BR><BR>Assignment ID = " . $i . "<BR> \n";

        // zyx
        $query = "SELECT assignmentID,stnum,filename,data FROM files WHERE
        assignmentID=" . $i . """;
        $result = mysql_query($query) or die('Query failed: ' . mysql_error());

        if (mysql_num_rows($result)==0){
            echo "\n Cannot find any info by this assignment ID";
            echo "<br>";
            echo "\n All valid AssignmentID are listed in the following table";
            echo "<br>";
            ShowAssignmentID();
        }else{
            print "<B>Files</B><BR>\n";
        }
    }

```

```

while ($r = mysql_fetch_array($result)) {
    $fn=$r['filename'];
    print 'Filename = ' . $fn . " - - - ";
    // now try to extract extension
    $nextn = strrpos( $fn, ".");
    $extn = substr($fn,$nextn);
    // print 'found hit= ' . $nextn . ' -- ' . $extn . "<br>\n";

    $fn=$r['stnum'] . "-s" . $i . $extn;
    print $fn . " <BR>\n";
    $fh = fopen($sd . '/' . $fn, 'w');
    // fwrite ($fh, $r['data']);
    fclose($fh);
}

// $result = mysql_query("SELECT assignmentID,stnum,filename,data FROM
markedfiles WHERE assignmentID=" . $i . " ");

$query = "SELECT assignmentID,stnum,filename,data FROM markedfiles WHERE
assignmentID=" . $i . " ";
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

print "<B>Marked Files</B><BR>\n";
while ($r = mysql_fetch_array($result)) {
    $fn=$r['filename'];
    print 'Filename = ' . $fn . " - - - ";
    // now try to extract extension
    $nextn = strrpos( $fn, ".");
    $extn = substr($fn,$nextn);
    // print 'found hit= ' . $nextn . ' -- ' . $extn . "<br>\n";

    $fn=$r['stnum'] . "-m" . $i . $extn;
    print $fn . " <BR>\n";
    $fh = fopen($sd."/marked/" . $fn, 'w');

    // fwrite ($fh, $r['data']);
    fclose($fh);
}

$i++;

```

```

}

print "<BR><BR><A HREF='etphpxt2.php'>Return to selector</a>\n";

// Show Assignment ID and Name in table "assignments"
// Show help for this tool after user gave wrong parameter
// Yuxiang (Richard) Zhang, Jun 19, 2005, University of Ottawa
function ShowAssignmentID()
{
// Performing SQL query
$query = 'SELECT ID, name FROM assignments';
// SELECT ID, name FROM assignments WHERE 1
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

// Printing results in HTML
echo "<table>\n";
echo "\t<tr>\n";
echo "\t\t<td><B>Assignment ID</B></td>";
echo "\t\t<td><B>Assignment Name</B></td>";
echo "\t</tr>\n";

while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

// Free resultset
mysql_free_result($result);
}

?>

```

7.3 Programs of Contact From System

7.3.A feedback.html

```

<html>
<head>
    <title>Contact Form</title>
    <? require("base_conform.php") ?>
</head>
<body bgcolor="#DDDDDD">

<h1 align="center">Contact Form for <? $unit_name ?></h1>

```

```

<font color="#FF0000">* Indicates a required field.</font><p>
<!form method=post action="https://extranet.management.uottawa.ca/doc-depot/pap----
/pap1test/handlefb.php">
<form method=post action="handlefb.php">
<table width="596" cellspacing="1" height="172" cellpadding="2">
<tr>
<td width="201" height="25">
Your Name<font color="#FF0000">*</font>:&nbsp;</td>
<td width="539" height="25"><input type=text name="name" size=32></td>
</tr>
<tr>
<td width="201" height="25">
Student Number<font color="#FF0000">*</font></td>
<td width="539" height="25"><input type=text name="StudentNo" size=10></td>
</tr>
<tr>
<td width="201" height="25">Question Type<font color="#FF0000">*</font>:</td>
<td width="539" height="25"><select name="StringQuestionType">
<option value = "1">Administrative Question
<option value = "2">Course Material Question
<option value = "3">How do I ?
</select>
</td>
</tr>
<tr>
<td width="201" height="25">
Your Email<font color="#FF0000">*</font>:</td>
<td width="539" height="25"><input type=text name="email" size=33></td>
</tr>
<tr>
<td width="201" height="23">Brief Description of Your Question<font
color="#FF0000">*</font>:</td>
<td width="539" height="23">
<p align="left"><input type=text name="Subject" size=33></p>
</td>
</tr>
</table>
<br>Describe Your Question in Full:
<p><textarea name="MailContent" rows=10 cols=56>
</textarea></p>
<p><input type=submit value="Contact Now">
<input type="reset" value="Clear Form">
<br>
</p>
<hr>

```

```
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>
```

```
</body>
</html>
```

7.3.B EditMsg.php

```
<?
//      FileName: EditMsg.php
//      Objective:   GUI for adding comments
//      Made by yuxiang (richard) zhang.
//      call from CmtIndex.php

//      $file = "doc/2004Aug25132448_2780333.txt";
//      $fp = fopen($file, "r");

if (!$fp)
{
    echo "<p>can not open $file";
    exit;
}

$priority = "Normal";
$test = "";
while (!feof($fp))
{
    $Line= fgets($fp, 200);
    $test .= $Line;
}
fclose($fp);
$action1 = "SaveMsg.php?file=$file";
?>

<html>
<head>
    <title>Add some comments to students' message</title>
</head>

<body>
<form method=post action=<?=$action1?>>
<b>Student's original message:</b> <br>
<p><textarea name="MailContent1" rows=10 cols=56><?=$test?>
</textarea></p>
<b>Your comments:</b> <br>
<p><textarea name="MailContent2" rows=10 cols=56>
```



```

?>
<html>
<head>
  <title>Index of All files</title>
</head>
<body>

<?php
//      FileName: CmtIndex.php
//      Objective:   Display all Messages to add comments
//      Made by yuxiang (richard) zhang.
//      C:\Program Files\Apache Group\Apache\htdocs\feedback
// http://localhost/feedback/CmtIndex.php
// created on August 29, 2004
//      modified on August 30, 2004
//      modified on August 20, 2005

//      There are 3 places need to be changed for doc-depot
// =====
// for doc-depot only, need add.      if local host, need comment this
//      https://extranet.management.uottawa.ca/doc-depot/pap----/pap1test/
//      echo "OK".realpath("."); // should be exp:/inetpub/apache
/*
    if(realpath(".") == "exp:/inetpub/apache"){
        $_cur_dir = getcwd()."/doc-depot/pap----/pap1test/";
        if(chdir($_cur_dir)!=TRUE){
            // not exist
            echo "error to chdir pap1test folder, pls contact me";
            exit;
        }
    }
*/
// =====

    if(chdir("doc")==FALSE){
        // not exist
        echo "error to chdir doc folder, pls contact me";
        exit;
    }
    $flag = 0;
    $number =0;
    $fileName[50] = "";
    echo "<h3>Please select a file you want to add comments</h3>";
    if($handle = opendir(".")){
        while (false!=( $file=readdir($handle))){

```

```

        if($file!="." && $file!=".."){
            if($flag==0){
//
border=0><TR><TD bgcolor=#A0A0A0>\n";

                print "<table border='1' width='60%'><tr><td
width='100%'>File Name</td></tr>";
                $flag=1;
            }
        }

//

=====
// for doc-depot only, if local host, need comment this, use
another instead.

//
depot/pap----/pap1test/";
//
$ABURL= "https://extranet.management.uottawa.ca/doc-
$SendURL = $ABURL."EditMsg.php?file=doc/$file";
//

=====

// =====
// for local host
//
$SendURL = "EditMsg.php?file=doc/$file";
// =====

$stable = "<tr><td width='43%'><A
HREF=$SendURL>$file</A></td></tr>";

        print $stable;
        $fileName[$number] = $file;
        $number++;
    }
}

if($flag==1){
    print "</TABLE></TD></TR></TABLE><BR>\n";
//
    print "<input type='submit' value='Send' name='BtnSend'>";
}

    closedir($handle);
}

?>

<p><!a href="https://extranet.management.uottawa.ca/doc-depot/pap----
/pap1test/Managelb.php">

```

```
<a href="ManageIb.php">
Switch to Students' Messages Manager</a></p>
```

```
<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>
```

```
</body>
</html>
```

7.3.D action.php

```
<html>
<head>
  <title>Action Results</title>
</head>
<body>
<h3>Thank you for using this web service </h3>
<?
//      FileName: action.php
//      Objective:   Execute Action -- move files to different folders.
//      Made by yuxiang (richard) zhang.
//      C:\Program Files\Apache Group\Apache\htdocs\feedback
// created on August 26, 2004
//      modified on August 30, 2004
//      modified on August 20, 2005
```

```
function filled_out($form_vars)
{
  // test that each variable has a value
  foreach ($form_vars as $key => $value)
  {
    if (!isset($key) || ($value == ""))
      return false;
    else{
      echo "$key\n";
//      if ($key!=
    }
  }
  return true;
}
```

```
/*
if (!filled_out($_HTTP_POST_VARS))
{
```

```

        echo "You have not chosen any bookmarks to delete. Please try again.";
        exit;
    }else{
// echo "You have chosen.";

        echo "$DestinationName";
        switch ($DestinationName)
        {
            case "Nash" :
            case "Trash":
                echo "Trash";
                break;
            case "Done":
                break;
            default :
                echo "<P>some error in switch.";
                exit;
        }

        exit;
    }
*/
// =====
clearstatcache();
$Destinationfolder = "doc/"$DestinationName; //"nash";

if(!file_exists($Destinationfolder)){
// if(chdir($Destinationfolder)==FALSE){ // not exist
// chdir("../"); // back to original path
// echo "not found $Destinationfolder";
// if(mkdir($Destinationfolder,0644)==FALSE){ // rw for owner, r for
everybody else
// make the folder
echo "error during make folder: $Destinationfolder";
return false;
}
}

// =====
// same for doc-depot and localhost
// $ABURL= "https://extranet.management.uottawa.ca/doc-depot/pap----/pap1test/";
// $SourceFile = "doc/"$file; // contain folder
// echo "OK".realpath(".");

$SourceFile = "doc/"$file; // contain folder
$file_destination = $Destinationfolder."/".$file;

```

```

        if(!copy($SourceFile, $file_destination)){
            echo "fail to copy";
            exit;
        };
        unlink($SourceFile);

//      echo "$SourceFile moved to $file_destination";
        echo "$file have been moved to $Destinationfolder folder successfully";
?>
<p><a href="ManageIb.php">Go Back</a></p>
<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>
</body>
</html>

```

7.3.E handlefd.php

```

<?
//      FileName: handlefd.php
//      Objective:
//          provide students a contact form to send professor email or store mail
information to files in server.
//          if student number is in the student list, this email priority is high.
//          if not, this email priority is normal.
//      Made by yuxiang (richard) zhang. Referred to mail php manual
//      created on March 13, 2004
//      last modified on March 21, 2004
//      The php script need students.dat in the same directory. students.dat can be copied
from etprl. ie. in windows: C:\Program Files\Abyss Web Server\htdocs\td\nash\mid_term
//      March 14, 2004
//
//      when method is "Send by file",
//      a folder "doc" is created and chmod write premission to everyone else.
//      an unique filename is created to store any mail information submitted by students.
//      March 21, 2004
//
        http://localhost/feedback/processfeedback.php?name=Richard&email=zyxback@
hotmail.com&StudentNo=2780823&MailContent=dfdsfds&Subject=test
/// some fixed variable
        $toaddress = "zyxback@hotmail.com"; // the default value,
        $method = "T_File";
        $priority = "Normal";

// check input errors
        $name=trim($name);

```

```

$email=trim($email);
$feedback=trim($MailContent);
$Subject=trim($Subject);
$StudentNo=trim($StudentNo);
$toaddress=trim($toaddress);

$bError = 0;
if(strlen($name)==0) {
    echo "<p>Your name can not be empty.";
    $bError = 1;
}
if(strlen($StudentNo)==0) {
    echo "<p>Your student number can not be empty.";
    $bError = 1;
}
else{
    if (!ereg("^[0-9]{7}$", $StudentNo)){
        echo "<p>Your student number is invalid.";
        $bError = 1;
    }
}
if(strlen($Subject)==0) {
    echo "<p>The subject of your email can not be empty.\n";
    $bError = 1;
}

if(strlen($email)==0) {
    echo "<p>Your email address can not be empty.";
    $bError = 1;
}
else{
    if (!ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\-.]+\.[a-zA-Z0-9\-.]+$", $email))
    {
        echo "<p>Your email address is a invalid .";
        $bError = 1;
    }
}

if($bError){
    echo "<p>Please return to previous page and try again.\n";
    exit;
}

if(strlen($toaddress)==0) {
    $toaddress = "zyxback@hotmail.com"; // the default value,
}
else{
    if (!ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\-.]+\.[a-zA-Z0-9\-.]+$",
$toaddress))

```

```

        {
            echo "<p>2nd email address is invalid. Please return to the
previous page and try again.";
            exit;
        }
    }

    if(strlen($MailContent)==0) {
        $MailContent = "(empty)";
    }

// marked, August 25, 2004 , coz we do not need priority right now.
/*
// check students.dat file to determine the priority
$fp = fopen("students.dat", "r");

if (!$fp)
{
    echo "<p>can not open student.dat file";
    exit;
}

$priority = "Normal";

while (!feof($fp))
{
    $Line= fgets($fp, 200);
    $StudentInfo = explode( "\t", $Line);
//    $LastName = $StudentInfo[0];
//    $FirstName = $StudentInfo[1];
    $StudentID = $StudentInfo[2];
//    $Section = $StudentInfo[3];
    // if student number is in the student list, this email priority is high.
    if( $StudentNo == $StudentID){
        $priority = "High";
        break;
    }
//    echo $LastName.$FirstName.$StudentID."<p>";
}
    fclose($fp);
*/

/*
// set mail    headers priority
// high priority mail, we have to set X-Priority: 1 & X-MSMail-Priority: High ;
// if normal mail, X-Priority: 3

```

```

// if low priority mail, X-Priority: 5 & X-MSMail-Priority: Low
// I tested in windows2000 + outlook express + hotmail, March 15, 2004
switch ($priority)
{
    case "Normal" :
    case "Low":
        $bMyStudent = 0;    // 1- is in my student list, 0 - not;
        break;
    case "High":
        $bMyStudent = 1;    // 1- is in my student list, 0 - not;
        break;
    default :
        echo "<P>some error in switch.";
        exit;
        break;
}
*/
//
/*
// marked, August 25, 2004
switch ($QuestionType)
{
    case "1" :
        $StringQuestionType = "Assignment/Quiz/Exam Question";
        break;
    case "2":
        $StringQuestionType = "Class Notes Question";
        break;
    case "3":
        $StringQuestionType = "Course Timetable Question";
        break;
    default :
        echo "<P>some error in switch question type.";
        exit;
        break;
}
*/

$mailcontent = "-----\n"
                ."This message comes from Contact Form\n"
                ."Student Name: ".$name."\n"
                ."Student Number: ".$StudentNo."\n"
//          ."Priority: ".$priority."\n"
                ".$StringQuestionType."\n"
                ."Email Subject: ".$Subject."\n"
                ."Email Content: \n".$MailContent."\n"

```

```

        ."Student Email: ".$email."\n";
    //      $mailer = "webserver@macnash.admin.uottawa.ca";      // depends on
different server.

    if( $method == "T_EMail"){
        //      method is email
        $head = "";
        $head = "MIME-Version: 1.0\n";
        $head .= "Content-type: text/plain; charset=iso-8859-1\r\n";
        $head .= "Date: ". date('r'). " \r\n";
        $head .= "From: $email \r\n";
        $head .= "Reply-To: $email \r\n";
/*
        if ($bMyStudent){
            $head .= "X-Priority: 1\r\n";
            $head .= "X-MSMail-Priority: High\r\n";
        }else{
            $head .= "X-Priority: 3\r\n";
            $head .= "X-MSMail-Priority: \r\n";
        }
*/
        // send email
        mail($toaddress, $Subject, $mailcontent, $head);
    }else if ($method == "T_File"){
        $toaddress = "File Server";
        if(chdir("doc")==FALSE){
            // not exist
            if(mkdir("doc",0644)==FALSE){ // rw for owner, r for
everybody else
                // make the folder
                echo "error during make folder";
                exit;
            }else{
                chdir("doc");
            }
        }
        $fName = date("YmHis")."_".$StudentNo.".txt"; // unique file name
        $fp = fopen($fName, "w");
        if(!$fp)
        {
            echo "Error: could not create the local file";
            exit;
        }
        fwrite($fp, $mailcontent); // not priority information if send by local
file
        fclose($fp);
        chdir("../"); // back to original path
    //
    echo "OK".realpath("../");

```

```

    }
?>

<html>
<head>
  <title>Inqiry Submitted Successfully</title>
</head>
<body>
<h3>Thank you for using this web service to submit your inquiry!</h3>
Your inquiry (shown below) has been submitted successfully! Your professor will contact
you soon.
<p>
<font color="#FF0000"></font>
<p>Sent to: <? echo $toaddress; ?>
<br><? echo nl2br($mailcontent); ?> </p>

<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>

</body>
</html>

```

7.3.F Install.txt

FileNanme: Install.txt

By Yuxiang (Richard) Zhang, August 20, 2005

The installation of this tool is very easy: just copy all files to web server.

Here, we suppose you are using apache web server in Windows.

The following is the typical way to install and use this tool:

1. Copy all files to c:\Program Files\Apache Group\Apache\htdocs\conform, or a similar directory.
2. If you are using unix/linux, you need change the permission of the "doc" directory and sub-directory to others(0777). Just depends on your system, you could use FTP or a File Manager or chown to change it.
3. Go to Contact Form, <http://localhost/conform/feedback.html>. Students could submit messages to professors or file webserver.

4. Go to Comment student's Message, <http://localhost/conform/CmtIndex.php>. It displays messages or files submitted by students. After click the link of one file, you could add some comments on it. This could be used by teaching assistants and professors.

5. Go to Students' Messages Manager, <http://localhost/conform/ManageIb.php>. It provides ways to deal with the messages, including send to professor, complete it and trash it.

*Security problems

Because messages or files under "doc" may not be suitable to expose, when this tool is used in real situation, you may need to contact the system administrator of web server to set the permission of "doc" by hiding it or granting the access right to some users. And of course, It's better to grant some scripts such as CmtIndex.php and ManageIb.php to the high level users.

7.3.G readme.txt

CONFORM - a file-based client contact form implemented in PHP

J C Nash and Yuxiang (Richard) Zhang
School of Management
University of Ottawa
Ottawa, ON K1N 9B5
Canada

August 30, 2004

All files in the CONFORM system Copyright (C) J C Nash & Y Zhang 2004

The CONFORM system is intended to allow clients -- for example, customers, members, or students -- to contact key personnel in an organization through a single Web-page CONTACT FORM. In particular, we wish to avoid email for the following reasons:

- 1) the email recipient may be away or otherwise unable to respond;
- 2) some clients will collect email addresses of staff and then blanket email requests to all of them, wasting bandwidth and possibly the time of multiple staff.

The system works with a simple collection of directories. Messages come into an INBOX, and can then be directed to any one of a number of folders (sub-directories), of which there will always be TRASH and DONE. Provision is made to attach comments to messages on each transfer. In any event a timestamp and userID are attached on transfers.

Files in the CONFORM system:

feedback.html -- the contact form page where clients enter messages

base_conform.php -- a script that contains and sets the common variables of the system, in particular the base URL for the directory containing these files and the message directories, the name of the unit (e.g., course)

action.php -- this script actually carries out the message file transfer

CmtIndex.php -- this script displays all the messages in a given sub-directory

EditMsg.php -- this script creates a GUI so that users may add comments to messages before transferring them to other folders or destinations

handlefb.php -- this script reports that the message has been received

ManageIb.php -- this script is designed to present the INBOX messages to the user who is managing incoming messages

SaveMsg.php -- this script implements the save comments function.

readme.txt -- this file

7.3.H SaveMsg.php

```
<?
//   FileName: SaveMsg.php
//   get "post info" from EditMsg.php
//   MailContent2 - comments user added.
//   $file = "doc/2004Aug25132448_2780333.txt";

//echo $MailContent2;
//echo $file;
    if($MailContent2==""){
        echo "No comments need to be added";
        exit;
    }
// Dummy will be replace after we know how to get current user name
$timeStamp = "Comments added by Dummy on ".date("F j, Y, g:i a");

$mailcontent = "\n-----\n"
                ".$timeStamp.\n"
                ".$MailContent2.\n";
```

```

    $fp = fopen($file, "ab");
    if(!$fp)
    {
        echo "Error: could not create the local file";
        exit;
    }
    fwrite($fp, $mailcontent);
    fclose($fp);
?>

<html>
<head>
  <title>Comments Added Successfully</title>
</head>
<body>
<h3>Your comments(shown below) have been added, Thanks!</h3>
<font color="#FF0000"></font>
<p> FileName: <?=$file?> <p>
<p><br><? echo nl2br($mailcontent); ?> <p>

<p><a href="CmtIndex.php">Back to Index</a></p>

<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>

</body>
</html>

```

7.3.I ManageIB.php

```

<html>
<head>
  <title>Manage Inbox</title>
</head>
<body>
<h3>Students' Messages Manager</h3>
<?php
//   FileName: ManageIB.php
//   Objective:   Manage inbox
//   Made by yuxiang (richard) zhang.
//   C:\Program Files\Apache Group\Apache\htdocs\feedback
// http://localhost/feedback/ManageIb.php
// created on August 26, 2004
//   modified on August 30, 2004

```

```

//      modified on August 20, 2005

//      There are 3 places need to be changed for doc-depot
// =====
// for doc-depot only, need add.      if local host, need comment this
//      https://extranet.management.uottawa.ca/doc-depot/pap----/pap1test/
//      echo "OK".realpath("."); // should be exp:/inetpub/apache
/*
    if(realpath(".") == "exp:/inetpub/apache"){
        $_cur_dir = getcwd()"/doc-depot/pap----/pap1test/";
        if(chdir($_cur_dir)!=TRUE){
            // not exist
            echo "error to chdir pap1test folder, pls contact me";
            exit;
        }
    }
*/
// =====

    if(chdir("doc")==FALSE){
        // not exist
        echo "error to chdir doc folder, pls contact me";
        exit;
    }
    $flag = 0;
    $number = 0;
    $fileName[50] = "";
    if($handle = opendir(".")){
        while (false!==($file=readdir($handle))){
            if($file!="." && $file!=".."){
                if($flag==0){
//                    echo "$file<p>";
//                    print "<TABLE cellspacing=0 cellpadding=0
border=0><TR><TD bgcolor=#A0A0A0>\n";
//                    print "<form action='action.php' method=post>";
                    print "<table border='1' width='100%'> <tr> <td
width='45%'>File Name</td> <td width='24%'>Action1</td> <td
width='20%'>Action2</td> <td width='23%'>Action3</td> </tr>";
                    $flag=1;
                }
            }

            $checkboxName = "C".$number;
            $DestinationName = "D".$number;

//
// =====

```

```

// for doc-depot only, if local host, need comment this, use
another instead.
/*
$ABURL= "https://extranet.management.uottawa.ca/doc-
depot/pap----/pap1test/";
$SendURL[0] =
$ABURL."action.php?file=$file&DestinationName=nash";
$SendURL[1] =
$ABURL."action.php?file=$file&DestinationName=done";
$SendURL[2] =
$ABURL."action.php?file=$file&DestinationName=trash";
$SendURL[3] = $ABURL."doc/$file";

$stable = "<tr> <td width='45%'><A
HREF=$SendURL[3]>$file</A></td> <td width='23%'><a href=$SendURL[0]>Send to
Nash</a></td> <td width='20%'><a href=$SendURL[1]>Complete it</a></td><td
width='23%'><a href=$SendURL[2]>Trash it</a></td></tr>";
*/

//
=====

// =====
// for local host
$stable = "<tr> <td width='45%'><A
HREF='doc/$file'>$file</A></td> <td width='23%'><a
href='action.php?file=$file&DestinationName=prof'>Send to Professor</a></td> <td
width='20%'><a href='action.php?file=$file&DestinationName=done'>Complete
it</a></td><td width='23%'><a
href='action.php?file=$file&DestinationName=trash'>Trash it</a></td></tr>";

// this is dropboxlist, for future.
// " <tr><td width='7%'><input type='checkbox'
name='$checkboxName' value='ON'></td><td width='43%'><A
HREF='doc/$file'>$file</A></td><td width='21%'><select size='1'
name='$DestinationName'><option selected value='Nash'>Nash</option><option
value='Trash'>Trash</option><option value='Done'>Done</option></select></td><td
width='29%'><A
HREF='action.php?file=$file&DestinationName=$DestinationName'>Send</td></tr>";

// =====

print $stable;
$fileName[$number] = $file;
$number++;
}
}

```

```

        if($flag==1){
            print "</TABLE></TD></TR></TABLE><BR>\n";
//        print "<input type='submit' value='Send' name='BtnSend'>";
        }else{
            echo "No students' messages found.\n";
        }

        closedir($handle);
    }

?>
<p><!a href="https://extranet.management.uottawa.ca/doc-depot/pap----
/pap1test/CmtIndex.php">
<a href="CmtIndex.php">
Switch to Comment student's Message</a></p>

<hr>
<p align="center">
<align="center">School of Management<br><align="center">University
of Ottawa, Canada<br>September, 2004
<p align="center"> </p>

</body>
</html>

```