Faculty of Computer Science and Information Technology

Nurul Sufiani Ahmad
WEK010219

Title: PBL Facilitator

Supervisor: Mrs Kiran Kaur a/p Gurmit Singh

Moderator: Prof. Madya Salimah Mokhtar

2003/2004

Abstract

The traditional learning method, lecture-based learning that's still dominating learning method not only Malaysia, but almost everywhere in the world, can no longer ensure student's career stability in real world after education despite high academic achievement. In lecture-based learning or LBL, students usually tend to memorize information that the teachers feed to them for examination purpose without really understanding the meaning of the facts. This scenario won't guarantee students' readiness to face real problems in real world, as they won't be able to apply their knowledge in a more real-life context.

Problem-based learning (PBL) however encourages students to learn how to solve real-life problems. It's a learning method where students work in small groups with minimal help from facilitators and collaborate to analyse a given scenario and analyse it to identify the problems and then provide the solutions. It's a much better way to make students really understand what they learn, why they learn and how to apply what they learn in real life.

PBL discussions however, are always deterred or not fully effective because of different factors, such as uncomfortable environment like hot temperature, crowded space, too noisy, uncomfortable chairs and etcetera. Even if the discussions are held outside class hours, then there's the matter of adjusting members' schedule, additional costs in terms of energy and time, external distractions like chats and many others.

As it is, the building of a web-based system that manages online PBL classroom will solve this problem. Named PBL Facilitator, it allows students to have a comfortable discussion environment, at students' and lecturers' choosing and cost.

Acknowledgement

So many people have helped in doing my final project, from the first of doing WXES3181, till the end of WXES3182. And I am eternally grateful for all the role they played inmaking this project worked.

First of all, I would like to thank my family, especially my parents for their undivided confidence in me,and for all their love, support and encouragement.

To the most important person in doing this project, my supervisor Mrs. Kiran Kaur Gurmit Singh, thank you for all your patience, and the guidance and advice you gave me. And not forgetting my moderator, Prof. Madya Salimah Mokhtar, thank you for your positive encouragement. Thank you both for believing in meto build this project and make it a success.

Not forgetting to all my friends, to whom that I could voice my concerns and anxieties and help me in solving them. Thank you all for the time we spent together discussing our works and all the useful opinions.

I love you all for all that you've done for me. Thank you all and god bless.

Contents

Contents **Page**

List of Figures

Lists of Tables

Table **Page**

Chapter 1: Introduction

Most educational institutions now are applying lecture-based learning (LBL) during classes, tutorials or lectures, as it's a traditional method of learning that's applied all over this country since decades ago. This situation involves all educational levels; primary schools, secondary schools, colleges and universities.

Lecture-based learning is a learning method where the teacher/lecturer teaches in front of the class and the students listen. It may be a one-way learning, where the teacher talks and students listen or two-way learning where teacher teaches with active participation from students. Either way, there's no guarantee that all students will take part in the class activities or even 'present' to the class' situation. It's not uncommon to see students whom are involved in their own personal activities like sleeping, chatting, daydreaming and etcetera during class.

Some teachers may able to acquire full attention from his or her students; either by intimidation, discreet threats, being funny, or develops a fun and exciting learning environment. And some lecturers just plain ignore the uninterested students and focus on the interested ones. But the important thing here is not lecturer's effort that must be relied on, but the students' themselves. However, it's better if the lecturers and schools' management could create an environment that will indiscreetly force the students to participate in class' activities.

One way to achieve that kind of environment is by implementing problem-based learning (PBL) in class activities. PBL is the opposite of (LBL). Where the lecturers are the major player in LBL, in PBL, each student plays the major roles whereas the lecturers act only as a second role player.

1.1 Project Overview: Introduction to Problem-based Learning (PBL) Facilitator

Online PBL Classroom Management System is a web-based system that manages online classrooms that applies PBL in its learning environment. Courses registered in this system will have their own respective "classroom". Students will enter an online classroom to participate in PBL discussions in their own respective group's discussion rooms, not just to download lecture notes, links and read announcements and messages from facilitators and other students, while facilitators will provide the students with announcements on the classroom's notice board, give feedbacks, upload lecture notes, add links to resources and monitor students' activities.

The system shall manage all the tasks that are involved in running the classroom. The system should be able to handle the process of registering the students, teachers and courses. All members need a username and a password to have access to the system. This feature is to ensure privacy to system and members' activities. The system also manages all the processes of allocating courses to its respective classrooms, assigning students into groups, managing PBL discussion rooms, evaluating students' performance, and many other tasks.

A student can only enter an online classroom if she or he is already registered for that class by the facilitator.

Students can interact with each other freely in the system via the classroom's message board. There are no chat rooms in a classroom, but students and facilitators are to have their discussions in message boards that are provided in assigned discussion rooms. Students can only participate discussions in their own discussion groups' message boards and they can only enter other groups' discussion rooms as guest. This feature is to ensure privacy and to avoid interruptions during group's

discussion. A facilitator however, have access and can participate in all group's discussions while monitoring students activities. This system enables facilitators to monitor students' communication more effectively.

The system can also evaluate students' performance based on their response during a discussion. Students will have immediate feedback of their performance from the system. This feature can prompts students to be more active during a discussion. Facilitators will also receive the students' evaluation result from the system, and they will also personally evaluate each student's performance based on the transcripts of students' discussions that the system recorded.

There are several drawbacks of the system. The drawbacks are:

1. the system is not intelligent enough. The system evaluates a student's performance by the number of responses that the student made, not the contents of the responses. The system won't be able to recognize whether a response by a student is related to the discussion or just rubbish statement. The consolation of this fact is that the facilitator can personally evaluate a student's performance based on the recorded discussions transcripts.
2. non members of PBL online classrooms are not authorized to enter the system, as the system requires password-access log in. Because of this, students or teachers whom doesn't take courses that offers PBL won't have the opportunity to experience an online PBL environment. Thus, they are not aware of the advantage of studying PBL online.

1.2 Motivation to Develop Online PBL Classroom

The system was developed as it's realized that current PBL classroom that's held during or outside class hours are lacking and faulty in several important facts:

1. Uncomfortable and noisy environment
   i. During a PBL session during class hour, all students will gather in their own groups. As all groups start to discuss, the class' volume tends to increase, as most students will be talking.
   ii. Groups usually seat in circles; it could be very uncomfortable, especially if there is less space in the classroom than necessary to accommodate all the groups.
2. Since there are many groups, there's no guarantee that the facilitators can monitor all students' activities. And students would have to wait in turns to have facilitator's attention.
3. Sometimes, the presence of facilitator while monitoring the groups could intimidate the students from discussing freely and more openly.
4. Sometimes groups will extend their discussion outside class hours if their discussion hasn't been satisfactory. This also could bring forth several dilemmas:
   i. It's hard to set a meeting place that's agreeable to all members.
   ii. It's not easy to set a time of discussion as members usually have their own schedule.
   iii. Most of the discussion time is probably spent with idle chatter.
   iv. It's not easy to find a discussion venue with suitable environment that's within reach of each member. Discussions are better suited if held at calm, non-crowded, peaceful places.
   v. Costly, from several aspects: time, money and energy.

vi. It's important to have face-to-face communication during discussion. If members of a group are shy of each other, especially if they have never known each other before, the discussion won't be fully effective.

1.3 The Objectives to Develop the System

The reasons the system is developed are:

1. To develop a web-based PBL learning system that's user-friendly, easy to use, educational, and interesting with non-formal environment.
2. To provide an easier method for students to have discussions at their own cost and choosing.
3. To provide a comfortable discussion environment.
4. Enable students to discuss freely and openly without interruptions from other groups or people around them.
5. Helps students to better acquaint themselves with technology-based learning environment.

1.4 Scope of the System

This system was built to manage online classrooms so as to enable PBL discussions in more comfortable environment. The number of courses allowed to register on the system depends on the amount of space in the server. And for each course, the number of discussion groups depends on the size of group that the facilitator set.

The system is divided into 8 separate modules:

1. Log In module
2. Group Management module
3. Evaluation module
4. Discussion Rooms module
5. Administrator module
6. Facilitator module
7. Student module
8. User Register module

Below is a brief description of each module's functions:

Log in module

This module manages members' log in whenever classroom members wants to enter the system and logging out whenever students wants to exit the system. It will also handles the situation whenever a member forgotten or lost a password or a username.

This module requires a password and a username from a class member to grant access into the system. Whenever a user logs on to the system, the system will also record the time, period and date of entry for security purposes.

Group Management module:

This module manages the process of assigning students randomly into groups after a course has been registered, and create discussion room for each group.

Evaluation module:

This module is responsible for capturing group members' response during a discussion. The module will then evaluate each member's performance based on

number of responses by a particular student and post the evaluation to facilitator of that class and to the student himself. The result of the evaluation will be displayed on the system after the discussion session has ended. Each member in a group will also evaluate each other after a discussion session.

The module will also record the transcript of the discussion and send it to the facilitator for further personal evaluation by the facilitator.

Discussion Rooms module:

This module manages the discussion rooms. The module will identify class members that enter a discussion room and their status and level of privileges. Their user ID and status will be displayed so their presence is known to all present. Group members and facilitators will be able to post responses in the room's board.

Administrator module

This module allows the administrator to manage the registration of facilitators. Administrator will also be in charge of the updating and editing of the data in the system, and he/she can posts announcements on the system to the system users. Administrator holds the power of granting the registration of a facilitator and can also terminates a classroom or the membership of a user if found necessary.

Facilitator module

Facilitator module allows a facilitator to register courses. Facilitator(s) will assign case studies/scenarios to the groups, post messages on the classroom's notice boards, upload all the case studies, answers or other information such as notes for students'

use on the system, add useful links, give feedback to discussion groups and terminate a student's membership if found necessary.

Student module

In this module, students can post messages on the classrooms' message board. Students will also be able to upload the solutions of their group's assigned case into the system directly to the facilitator.

User Register

This module enables user to confirm their registration. Administrator registers administrators and facilitators, and facilitators register students. Users need to input their details and change the password given in order to confirm their registration.

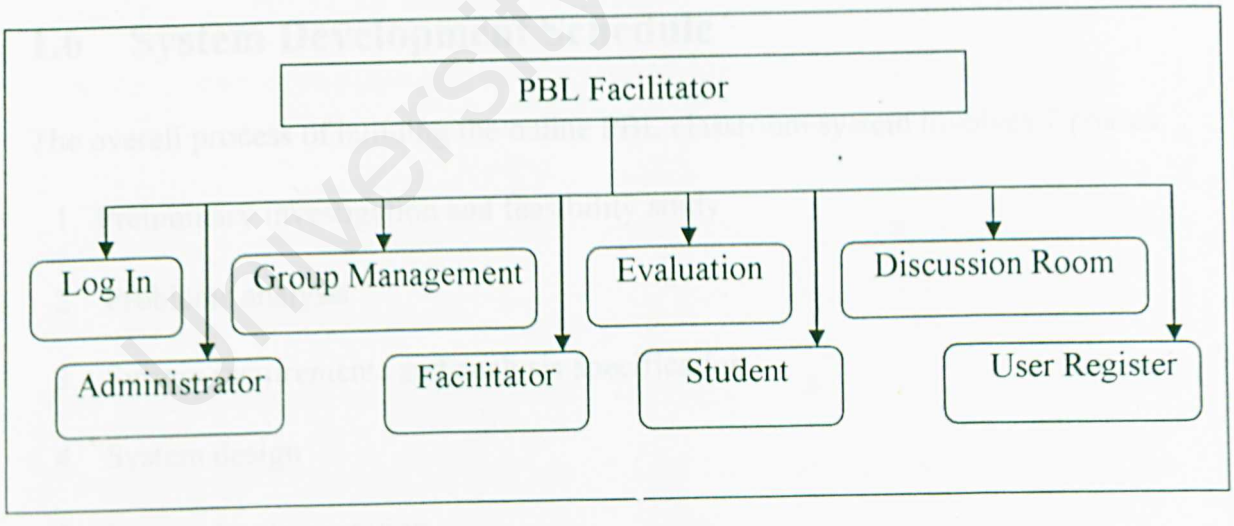


Figure 1.1: Modules in PBL Facilitator

1.5 User Target

The system is developed for the use of higher levels education institutions that applies PBL in their courses. The institutions may have been using PBL methods before or are just now starting to integrate PBL in their learning environment. It is of small matter as this system is suited to both situation and easy to use, as long as the users realize the full meaning of PBL to make full use of the online PBL classroom. The system can also be used to replace the existing usual PBL classroom (non-online) or for the purpose of new PBL course. Students and lecturers/teachers/tutors of secondary schools, colleges and universities are most encouraged to use this type of system since they are now are more into PBL and more classes are applying PBL activities than LBL.

1.6 System Development Schedule

The overall process of building the online PBL classroom system involves 7 phases:

1. Preliminary investigation and feasibility study
2. Problems analysis
3. System requirements and analysis specification
4. System design
5. System Implementation
6. Documentation
7. Presentation

The first four phases are performed during the 3rd semester of 2003/2004 sessions. The time span of doing the phases are about 3 months.

Preliminary Investigation and feasibility study:

The purpose of this phase is to identify whether the system is worth to develop or not. It's also conducted to establish the scope, preliminary requirements and constraints and schedule of the project, and to determine the feasibility of the project.

Problems Analysis:

During this phase, existing online PBL online classrooms and any other online systems that are related to the project scope is looked at and analyzed. The main purpose of this phase is system improvement objectives where the criterias that the new system will be evaluated is defined.

System requirement and analysis specification phase:

During this phase, all requirements are defined and then prioritized. This task concern on defining what the system should do, not how the system should work.

Design phase:

Design phase involves transforming the requirements statements from the requirements analysis phase into design specifications for system construction.

System implementation and documentation are conducted during the first semester of 2004/2005 sessions in 4 months:

System Implementation (Construction, Testing and Maintenance):

The process of constructing (coding) and testing the system is conducted during this phase. Any corrections and improvements towards the system (maintenance) can be done after system testing or simultaneously.

Documentation:

The task of documenting the facts and specifications of the system actually starts since the beginning of project initiation.

Presentation:

The system is to be presented in report form and presentation during the last week of project schedule.

The project schedule is outlined using a project management tool that enables a project manager to stay in control of the project and keep the project performed under budget and in schedule. It enables project scheduling, resources maintenance and activities tracking to keep track of project's progress. The tool that was used for this purpose is Microsoft Project.

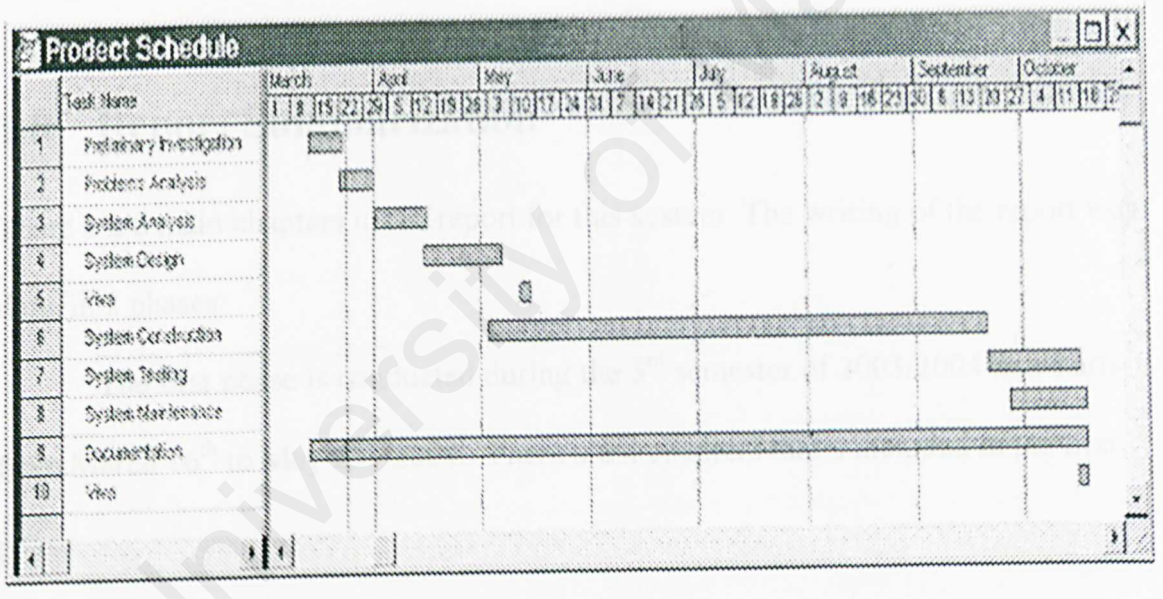


Figure 1.2: Gannt Chart for PBL Facilitator

1.7 Expected Output

During the time that was spent researching, designing and building the system, the final output for PBL Facilitator has been determined to perfect these criteria:

1. the system produced should be an interesting, simple, user-friendly and

educational PBL online classroom.

2. the system should be able to provide online discussion classrooms that can heightens students' interest and participations towards PBL activities.
3. the system should be able to provide comfortable environment for active discussions.
4. a system that enables facilitators to monitor groups' activities without interruptions more easily.
5. a system that should be able to heighten students' awareness and interest towards web-based learning environment.

1.8 Report Summarization

There are 8 main chapters in the report for this system. The writing of the report was done in 2 phases.

The first phase is conducted during the 3rd semester of 2003/2004 that starts from March 16th to May 23rd 2004. There are 4 chapters that's included in the first phase:

1. Introduction.
2. Literature Survey
3. Methodology
4. System Analysis

The second phase starts from 28th May till 3rd September 2004 during 1st semester of 2004/2005. The remaining 4 chapters are performed during this phase. The chapters are:

5. System design

6. System development
7. System testing
8. Conclusion

The process of designing and building the system is to start simultaneously with the second phase of the report writing. However, it's much more wiser to start building scratches of the system since the beginning of report writing from the ideas, imaginations or whatever insights that comes to mind, so that students can actually have a starting point and continue to build a real system with proper designs and requirements later after all the necessary researches and planning has been completed.

Below are listed all the chapters to be included in this report and brief explanations for each of them.

Chapter 1: Introduction

The first chapter defines the problem that's being researched on, which in this case concerns the PBL and online classroom management system. Introduction also defines the objectives and scopes of the project, states the motivation for doing this project and for whom the system is intended for. The project schedule is shown in a Gannt Chart.

Chapter 2: Literature Survey

This chapter records the result of the researches that is conducted concerning the problem of this paper before the project is undergone. The research involves researches and analysis on existing PBL online classrooms, e-forum and web sites that have interactive online message boards and previous paperwork about online

PBL classes. It also records the findings on techniques and methodologies to be used on the system and all the domains that are to be covered in order to build the system.

Chapter 3: Methodology

Chapter three explains the techniques and researches methods that are used in solving the project's problems in detail.

Chapter 4: System Analysis

System Analysis explains about all the requirements that are necessary in building the system. Examples of the requirements are functional requirements, nonfunctional requirement and hardware and software requirements. All the software and hardware requirements are compared to identify which are most suitable to be used in the system.

Chapter 5: System design

System Design chapter explains about the designing of user interface, process design, database design, the data flows and all the modules in the system. All these components and various more will then be integrated to form the complete online PBL classroom system. Each component has its own specific functions necessary to ensure that the system will work properly.

Chapter 6: System Implementation

This chapter explains the process of converting all the modules and algorithms into programming languages commands. These commands will enable computers to understand the structure and working of the system, and thus to run the system. All

the programming languages chosen for this purpose will also be explain in this chapter.

Chapter 7: System Testing

System Testing explains whether the system works, as it should according to the requirements and specifications defined. It will identify whether the final system satisfies the objectives of developing the system. The system shall first be tested in real life and the result of the experiment will be recorded in this chapter.

Chapter 8: Conclusion

The final chapter shall discuss in whole about the final output of the system project. In this chapter, are included the results of the system (its performance, rating, etcetera), all the problems concerning the system and its solutions, advantages and disadvantages of the developed system, additions that can perfected the system, suggestions to improve the system and the overall conclusion of online PBL classroom management system.

Chapter 2 Literature Review

2.1 Introduction to Problem-based Learning (PBL)

PBL is an educational strategy that uses a scenario to drive the learning and promotes active learning. It is a method of instruction that uses scenarios or triggers as a context for students to acquire problem solving skills and basic knowledge (Banta *et al*, 2000). It encourages active interactions among students in small groups as they are required to discuss a given scenario or ill-structured problem which reflect the real world with minimal assist from lecturers. Problems are in fact a vehicle for developing students' problem solving skills. New information is required of the students as they self-directed themselves in learning. Lecturers are present only to facilitate or guide the students' activities and assist them in how to reach a solving, not to help them solving the problems. PBL is a much more effective way to make students understand what they learn and how to apply them (Diana Jones, 1996).

The ability to solve problems is more than just accumulating knowledge and rules; it is the development of flexible, cognitive strategies that help analyse unanticipated, ill-structured situations to produce meaningful solutions. Though many of today's complex issues are within student understanding, the skills needed to tackle these problems are often missing. Typical problem solving taught in schools often tends to be situation specific with well-defined problem parameters that lead to predetermined outcomes with one correct answer. In these situations, it is often the procedures required to solve the problem that are the focus of instruction. Unfortunately, students skilled in this method are not adequately prepared when they encounter problems in which they need to transfer their learning to new domains.

According to an article by Diana Jones of San Diego State University (1996), real-life problems seldom parallel well-structured problems; hence, the ability to solve traditional school-based problems does little to increase relevant, critical thinking skills students need to interact with real life. Well-structured problems with their sterile environments in which there is only one right answer simply teach students about problem solving, not how to problem solve. In real life, we seldom repeat exactly the same steps to solve problems; therefore, the lockstep solution sequence taught in well-structured classroom problems is seldom transferable. Instead, real-life problems present an ever-changing variety of goals, contexts, contents, obstacles, and unknowns which influence how each problem should be approached. Students need to practice solving ill-structured problems that reflect real life to be successful in their career. This skill is the goal of problem-based learning.

Novice learners require more structure and cues while solving problems while more experienced students are self-directed learners. According to Jones (1996), the best format for problems is unorganized, unsynthesized, and open-ended because this allows for student processing. The problems presented must have these criteria:

1. common situation to serve as a prototype for other situations
2. significant
3. prevention is possible
4. interdisciplinary
5. cover objectives
6. task oriented
7. and complex enough to incorporate prior knowledge

2.1.1 Benefits of PBL

Listed below are several of the benefits and the reason of implementing PBL:

1. PBL prompts motivation

   Students will engage more in learning, as they will feel empowered to have an impact on the outcome of the investigation.

2. PBL provides relevance and context

   It makes student realize the importance of learning the particular course, why do they need to learn it and how to apply the knowledge they gained in real life situation.

3. Encourages student to learn, use, apply and enhance higher-order thinking skills.

   PBL persuades students to learn how to learn. It is a self-directed learning, as they must learn how to do all the processes involve in solving problem and to generate their own strategies with minimal guidance from facilitator.

There are more benefits of implementing PBL. Steve Gibbs (2003), while presenting his paperwork: Problem-based Learning with the Net, stated the benefits of PBL. Among them are:

1. PBL allows creativity, critical thinking and self-direction
2. PBL allows more than one answer
3. PBL fosters group collaboration
4. PBL deals with controversy as well as facts. This is because the learning is more important than the final product
5. PBL teaches life-long learning skills.
6. In PBL, students must dig deeper into the WHY of things.

2.1.2 Disadvantages and advantages between PBL and LBL

Results of a study by Samford University (Banta *et al*, 2000) stated that though the amount of time learning by PBL and non-PBL students are the same, PBL students have better achievement and are more satisfied with their learning experience and education atmosphere. They also rate their experience higher in terms of meaningfulness, flexibility, emotional climate, and nurturance and students interaction. The study further shows that PBL implementation increase class attendance and decrease student's distress or mood.

Table 2.1: Differences between LBL and PBL

LBL	PBL
Students likely to describe their studies as "irrelevant, passive and boring"	Students describe their studies as "engaging, difficult and useful"
Teacher lectures while students take notes; knowledge is transferred from teacher student	Teachers serves as facilitator while students collaborate with each other and teacher to develop personal understanding of content and information is conveyed to groups of students.
Students inclined to use faculty selected, reading material	Students make use of library; journals, online researches and self-selected material
Teachers organize content into lectures based on context of discipline.	Teachers design course based on "ill-structured" problems and select concepts that will allow students to transfer knowledge and increase students' motivation by providing real life problems and understanding students' problems.
Teachers introduces new information and suggests implications of those information	Teachers and students introduce new information and both offer interpretations and implications
Students absorb, transcribe, memorize and repeat information for tests and examinations.	Students engage in meaningful learning to understand what they learn for examination and real life purpose
Individualistic and competitive learning.	Students learn in a collaborative and supportive environment.

Students seek correct answer to finish tutorials, tests or examinations.	Students learn to frame questions, formulate problems, explore alternatives and make effective answers.
Roles of teacher and student are regimented: teacher disseminates knowledge and the student reflects that information	Roles of teacher and students are dynamic: teacher and students are a community of learners. Teacher serves as coach and mentor; students become active participants in learning
Grading is summative, and the facilitator is the only evaluator.	Students evaluate their own contributions as well as other members and the entire group (peer evaluation).

Source: (Banta *et al*, 2000 ; LeJeune, 2001)

Stated in the table above are several differences between LBL and PBL. Several of the advantages over LBL are (Jones, 1996):

1. PBL emphasize on meaning, not facts

Memorization is a common occurrence in LBL. And even then, most students actually use little of what they memorized. PBL however, engages students in structuring solutions to real life, relevant and contextualized problems. Students become actively engaged in meaningful learning by replacing lectures with discussion forums, faculty mentoring and collaborative research.

When PBL discussions are done online, students will have better opportunity in understanding what they learn. Since there are many learners gathering and discussing, more opinions will be issued. This is also helped by the fact that all the users are "anonymous", because some students are afraid to voice their opinions for fear of being ridiculed or just being plain shy.

2. PBL increase self direction

Students tend to assume greater responsibility for their learning as they pursue solutions to their problems. PBL students use self-selected resources (e.g. journals, online or other library resources, discussions, etcetera) more often than LBL students, and they view their facilitator as a resource to solving relevant problems rather than as a source of answers. These, together with learning skills help students become more competent in information seeking skills than LBL students.

When discussions are online, students must encourage themselves to go online and participate in their groups' discussion, as there will be no one to make them to. They also must always be ready with information and materials for discussion purposes, as they can't simply just rely on anyone else, and lack of physical contact doesn't allow them to share, especially printed materials. Otherwise, they'll be left out and won't benefit from the discussion.

3. PBL ensures higher comprehension and better skill development

Skill and comprehension development occurs when the problems to be solved are engaging, difficult and useful. Real world contexts and consequences allow learning to become more profound and durable, and increase the transferability of skills and knowledge from the classroom to work, as students are able to practice the knowledge and skills in a functional context, allowing students to better imagine what it will be like using the knowledge and skills on the job.

Online discussion has wider participations. And with different sort of people, with different skills, opinions and understanding, students will

definitely have more chance in practicing their skills and better comprehension.

4. PBL encourages interpersonal skills and teamwork

PBL incorporates collaborative teams in solving relevant problems, as social interaction is an important aspect in work life. This methodology promotes student interaction and teamwork, thereby enhancing students' interpersonal skills such as working with group dynamic, peer evaluation, and how to present and defend their plans.

Online discussion is just like a meeting of anonymous surfers. User interacts without physical recognition and reaction. Thus, students must learn to perfect their interpersonal skills to make the discussions more effective. This in turn, will further increase their teamwork skills.

There are more advantages of PBL in students' learning. With PBL, students can train themselves in critical and proactive thinking, improve ability to work in a team, improve one's computer skills, improve ability to find and use appropriate resources and many others.

2.1.3 Assessment of students' achievement in PBL

As there are multiple purposes of PBL, it's important to consider various ways of evaluating student's achievements. Several of evaluation techniques are (Jones, 1996; Banta *et al*, 2000):

1. Written examinations
2. Practical examinations – These examinations are used to ensure that students are

able to apply skills learned during the course.

3. Concept maps - Students can determine their cognitive growth by depicting their knowledge via generating concept maps.
4. Peer assessment - Members of a group can assess each other as they are required to work together, provided they are given an evaluation rubric to guide them in the evaluating process.

1. Self assessment - This task helps students identify gaps in their knowledge base in order for more meaningful learning to result.
2. Facilitators/tutor assessment - A tutor's feedback should encourage the students to explore different ideas.
3. Oral Presentations - Oral presentation provides students an opportunity to practice their communication skills.
4. Reports
5. Multi-choice questions (MCQs)
6. Short-answer questions (SAQs)
7. Modified essay questions
8. Triple jump exercise and extended matching items

2.1.4 Barriers to Implement PBL and Steps to Overcome the Problems

Jones (1996) stated several barriers while implementing PBL:

1. School will need to change the curriculum, something that's usually viewed negatively as additional work without benefits as they need to develop a new one and

they have to change their lifetime experiences and requires an adjustment period. They also needed to know what to be expected and how to adjust to the changes.

2. PBL requires more students' time, independencies and responsibilities. Facilitators should make the transition smooth for students via effective communication and orientation.

3. In PBL, preparation to teach problems takes more time than to presenting lectures. Student-teacher contact hours are also reduced weekly as PBL involves small groups of preferably 4 to 5 students, while lectures include the entire class. Thus, more teachers would be required to teach the same content in a PBL curriculum.

4. Changing curriculum requires costs. PBL also requires numerous small rooms equipped for teaching and adequate copies of current library resources. The school will be burden with the wonder whether the changes are worthwhile in terms of benefits in learning effectiveness or efficiency.

5. Faculty lack extrinsic rewards for PBL teaching. Professors receive more rewards for research, publication and fund raising than teaching. The reward system may actually create disincentives.

Listed below are several of the steps can be taken to improvise on the problems (Jones, 1996):

1. Overcome resistance from faculty members

2. Reward faculty members for PBL participation

3. Find time and money to develop PBL

4. Train facilitators to handle PBL courses

5. Improvise strategy in holding faculty members' interest in PBL

2.1.5 PBL in Schools

There are hundreds universities and colleges that has applied PBL in their learning environment. A few of them are (PBL Insight, 2000):

1. University of Delaware, Baltimore
2. Central Queensland University, Australia
3. University of Newcastle, Australia
4. Aalborg University, Aalborg, Denmark
5. Newcastle University, Newcastle
6. Samford University, Birmingham
7. University of Hong Kong, Hong Kong
8. University of Malaya, Kuala Lumpur

According to a research (Salimah M. and Zaitun A. B., 2003), University of Malaya has been implementing PBL since 2001. Though at that time, it seems none of UM staff or faculty members are knowledgeable about PBL, much seen the implementation of PBL. Faculty members and lecturers were sent to workshops, conferences and training to understand more about PBL.

At Faculty of Science Computer and Information Technology, in 2003/2004 there are up to 14 courses that actually took part in including PBL in the syllabus. Most of the courses applied both lecture-based and problem-based learning. And several actually fully implements PBL during class hours.

There been universities that conduct their PBL classrooms online. Among them are Central Queensland University and Anaheim University, Australia. However, there are also protected PBL discussion classrooms for non-professional groups on the Internet.

2.2 A PBL Case Study and Methods for Facilitators and Students

This case study and its method of solution is an example suitable for use in a first year college biology lecture theatre, but the method is applicable to any class size and educational level (Peter Ommundsen, 2003).

2.6.1 Case Study: A Case of Deadly Exertion

A fitness test of applicants to a fire department resulted in 32 hospitalizations with back pain, muscle pain, and reduced urine output. One person died. [MMWR 39:751-6.]

The students will at some point address muscle physiology. What happens when muscle cells break during exertion? What are the consequences of hyperkalemia on the heart? Where does all the potassium originate? What are the effects of myoglobin on the kidneys? What is the impact of oxygen free radicals produced by damaged muscles?

2.2.1 Method for Facilitators

1. Form Small Groups

PBL may be a part of a class session or all of it. But, during that time, students must form small work groups of 3-5 people. Facilitators may assign the groups themselves or assign them randomly.

2. Present the Problem

Present the students with a brief problem statement (preferably on a printed work sheet), e.g., "A 28-year-old man appears to have osteoporosis." In some cases a video clip or specimen might be used as a trigger. Emphasize to the students that they are dealing with an authentic case history. Bizarre problems work best. Prior to class facilitators should review the case history and arm themselves with data that can be released incrementally (progressive disclosure) as the case proceeds. There is a comprehensive data set for the osteoporosis problem in the New England Journal of Medicine, 1994, 331:1056-61; 1088-9. Needless to say, the students should not be given the reference, as the objective is to solve a problem, not read a solution.

3. Activate the Groups

Ask the groups to brainstorm possible causes of the osteoporosis. Each group will have to discuss, review, or investigate the biology of bone, including the role of osteoblasts, diet, vitamin D, parathyroid hormone, growth hormone, calcitonin, kidney function, etc. This is when much learning occurs, as the students help each other understand the basic biology. PBL students must reflect upon biological mechanisms rather than just memorize facts (as might occur in some traditional

lecture-only courses). The facilitator circulates among the groups, providing assistance but not solutions. The groups may well explore avenues unanticipated by the facilitator. This is highly desirable and should not be discouraged. The facilitator should avoid controlling the agenda of the groups. Each group ranks its hypotheses in order of priority and prepares requests for more data. (E.g., for calcium deficiency hypothesis -- "What did he usually eat?")

4. Provide Feedbacks

Ask that a representative from each group place their top priority hypothesis or data request on the chalkboard (if already entered by another group, place their second choice, etc.). If this is not practical, ask for oral suggestions from the groups when the small group work is halted and the class is reconvened. Student suggestions may include:

- Low calcium diet
- Immobility
- Low density of vitamin D receptors
- Calcitonin deficiency
- Excessive PTH
- Chronic acidosis buffered by salts mobilized from bone

The small group work can be stopped and the instructor can briefly discuss the ideas with the entire class. It is important to value every contribution, to assist the students in analysis of the biology involved, and to provide further information [he was not immobile, he had a normal diet, etc.]. The students can be prompted for data

requests: "If you could ask for just three test results from examination of this man, what would they be?"

It is not likely that the students will solve a problem on the first pass, and the feedback from the facilitators motivates the next round of small group work. The students could now be told that the man's lumbar spine density is 3.1 standard deviations below the average age-matched healthy female (osteoporosis = 2.5+ SD), his height is 204 cm, his left middle finger is 10 cm, and knee films show open epiphyses. (The students should now be able to figure out that the man may still be growing at age 28). The cycle of small group work and instructor feedback can be continued during the current class session or on future occasions. *The key to managing a PBL session is providing continual feedback to maintain student enthusiasm while simultaneously prolonging the resolution of the problem to ensure that adequate learning occurs.*

5. Ask for a Solution

At this point in our example, the groups will likely focus on the hormones required for epiphyseal closure and bone mineralization. They may ask facilitators for serum estrogen levels (high), which will suggest estrogen-resistance. Were estrogen receptors defective? (Yes.) When a reasonable number of groups have solved the problem, a student might request a brief written analysis from each group describing the biology involved in the case. Students may be asked to include certain key words in their reports. If the facilitator wishes to further pursue this case at a later date he/she could tackle the genetics of the defect. (C to T transition in the estrogen receptor gene in both alleles causing a premature stop codon; both parents heterozygous with consanguinity in the pedigree.)

2.2.2 Method for Students

1. Define the Problem Carefully

What exactly are the students trying to determine? Does the problem have several components? If several, state them separately. Does everyone in the group agree with the way the problem has been framed? Ask group members to "think out loud," as that slow down their reasoning and enables people to check for errors of understanding.

2. Explore Possible Solutions

Brainstorm ideas that may contribute to a solution. A group member must justify their ideas to the others. Clarify for them the biology involved. Have them paraphrase the ideas. Listen carefully to the ideas of other group members and give positive feedback. Make a list of learning issues. What do we know? What don't we know? Is this problem analagous to any past problem? What core biological concepts may apply to this problem? Assign research tasks within the group.

3. Narrow Your Choices

After developing a list of hypotheses, sort them, weed them, and rank them. List the type of data required to test each hypothesis. Give priority to the simplest, least costly tests. It is easier to get information on the diet of a subject than it is to do sophisticated biochemical tests.

4. Test Your Solution

Seek from the facilitator the data that is needed to test the ideas. If all the possible solutions are eliminated, begin the cycle again: define, explore, narrow, test. When the students encounter data that confirm one of the hypotheses, they may be asked to write a biological explanation of the solution and justify it using the available evidence.

2.3 Introduction to Online Learning

Online learning refers to learning and other supportive resources that are available via a computer that displays material in response to a learner's request. The computer prompts the learner for more information and presents appropriate material based on the learner's response (Carliner, 2002).

An essential element of online learning definition is the use of computing and telecommunication technologies to deliver or receive education. Online learning utilises the Internet or videoconferencing to create communities of learners. Course materials are provided on a website and occasionally it is linked to a CD ROM. Email, bulletin boards, forums and chat rooms are used to interact with other students and teachers.

Online learning offers many varieties and flexibilities. In contrast with simple online learning, online learning might consist of a course that includes regularly scheduled lectures by video conference on the Internet, a web page with several supplemental materials including videotapes of previous lessons, discussions outside

of class by electronic mail by lists and online tests whose results are automatically recorded in student records. Online learning may take place in the classroom or workplace, alongside other activities, or may be undertaken at home or at online access centres in the community. (Department of Education Tasmania, 2003)

According to Saul Carliner (2002), there are several forms of online learning, such as computer-based training, web-based training, computer-based instruction, and technology-based instruction.

These terms has specific meaning:

- **Computer-based training** refers to courses presented on a computer. The course does not provide links to learning resources outside of the course. The course is usually computer-based training that is studied on a computer that is not connected to a network.
- **Web-based training** is a form of computer-based training, refers to courses available on a network that is linked to learning resources outside of the course, such as references, electronic mail and discussions, and videoconferencing.
- **Technology-based instruction** refers to training through media other than the classroom. That includes computers, and also television, audiotape, videotape, and print.

PBL is most suited for web-based training. Combined with collaboration learning, it's a perfect learning environment for PBL. Students and teachers may have their discussions in forums and they can exchange information either by email or on the web site for their 'online PBL classroom'.

The advantages of online PBL classrooms compared to usual classrooms are:

1. students and teachers are not restricted to close space.
2. there's no worry about class not having enough space to accommodate all students.
3. students can choose to join the class/discussion wherever he felt comfortable.
4. online classroom ensures that students won't be disturbed by other students or other factors like class temperature and etcetera.
5. online classrooms reduce costs; energy, money and classroom maintenance.
6. teachers can monitor students more easily.
7. aside from class session rules, students and teachers are not restricted to any regulations like dressing codes and can be more carefree.

An online classroom also has its disadvantages. For instance, instead of concentrating to class session, students may very well playing computer games without teacher's knowledge. However, what really important is students' willingness to participate in class activities and learn. Because however good a technology is, it would turn to waste, or worse, contribute to bad cause if the users won't realize its full purpose and benefits, and the same goes with online classrooms.

The Technology

There are two types of technology needed to support online learning (Carliner, 2002):

1. Learning programs development: this included all the hardwares and softwares needed to create and preparing learning materials.

2. Learning program deliverance: includes at least the specialized computer hardware and software that allows learners use the learning materials.

2.4 Collaborative learning

Collaborative learning is a principle in PBL. Carliner (2002) refers to collaboration is a situation in which people work together where learning occurs in the process of doing so. It is useful in tutoring and coaching and also promotes learning between people interaction.

Online collaboration occurs in several ways:

- collaboration occurs when people from different locations work together via electronic mail and exchange of work files.
- people from different locations "meet" online to work together. This situation depends on the technology available at the time. The participants might have online "chat", a discussion that occurs online when all of the participants are simultaneously signed on or they can have a videoconference or simply "meet" on an online forum.
- since all of the people working together need to be online at the same time, collaboration is said to be synchronous. Software called groupware is designed to promote synchronous and asynchronous online collaboration.

Technology needed for collaborative learning

Minimum: 1. Connection to a computer network (modem, software)

2. E-mail account and software

3. Software that allows file exchanges

Maximum: 1. Connection to a high-speed network that allows video and audio transitions.

2. Camera and related software to capture and transmit images

3. Microphone and related software to capture and transmit sounds

4. Software for viewing and transmitting video and audio images

5. Software that allows image display and image viewing.

6. Software that allows image editing and enables a user from different location to view the changes.

2.5 Existing Systems

2.5.1. PHP Builder Community Forums (http://www.phpbuilder.com/board)

PHP Builder is a web site that host information and latest news about PHP and also provides forums for discussions about PHP. The community forums in PHP Builder are very active. There are many forums available, from the forums for PHP beginners, forums on PHP installations to the forums for the experts in PHP.

Internet users whom want to participate in the forums must register for a forum membership first. Membership is free. Members can posts messages and queries and can also answers other members' questions. Non-member users can also view the message boards. Like registered members, they can view all the discussions displayed on the web site and take advantage of the knowledge from the result of the discussion, but they cannot posts any messages.

Figure 2.3 is a view of the main page of the forum 'General Help'. It is a forum that discuss generally about everything concerning PHP. The forums in PHP Builder are so active that if a user sent an inquiry, and if it's worth looking at, the person will get a reply in a matter of minutes. Sometimes, a member has to check whether a similar question has been issued before. Otherwise, that person probably won't get a reply and actually waste his and other members' time. Anyone can start a discussion or thread. Basically it starts when a member issues a question and other members will help in solving the problem. Once a discussion has already reached a solution, it's advised for the discussion starter to put a 'Resolved' status on the discussion title, as can be seen from the picture.

The user interface for PBP Builder is simple and attractive. It doesn't use too many colours and its navigation is simple, so users won't lose their orientation easily.

There are several features that were found interesting about PHP Builder Community Forums. Among them are profile editing, forum calendars, FAQs, private messaging, and moderators can terminate a message and also boot out the issuer from the forum if the message is found offensive and if a moderator received a complaint from other members.

2.5.2 WebmasterWorld (www.webmasterworld.com)

WebmasterWorld is a web site that hosts news and forums topics surrounding the design, production and maintenance of a website for web professionals. There are two types of forums in WebmasterWorld; private and public. Like PHP Builder Community, Internet user can view the forums freely, but registration is necessary for discussion participation. Users can have the beast threads by subscribing to the WebmasterWorld, with monthly payment. Most of the best posts are displayed on its main page. It also provides glossary, search engine, and indexed contents, among many other features. The site has easy navigation, user friendly with not too complex design and not too many items on its pages. However, the font property is too large. The user interface is not attractive, and the design is too simple.

Figure 2.1: PBP Builder Community Forums

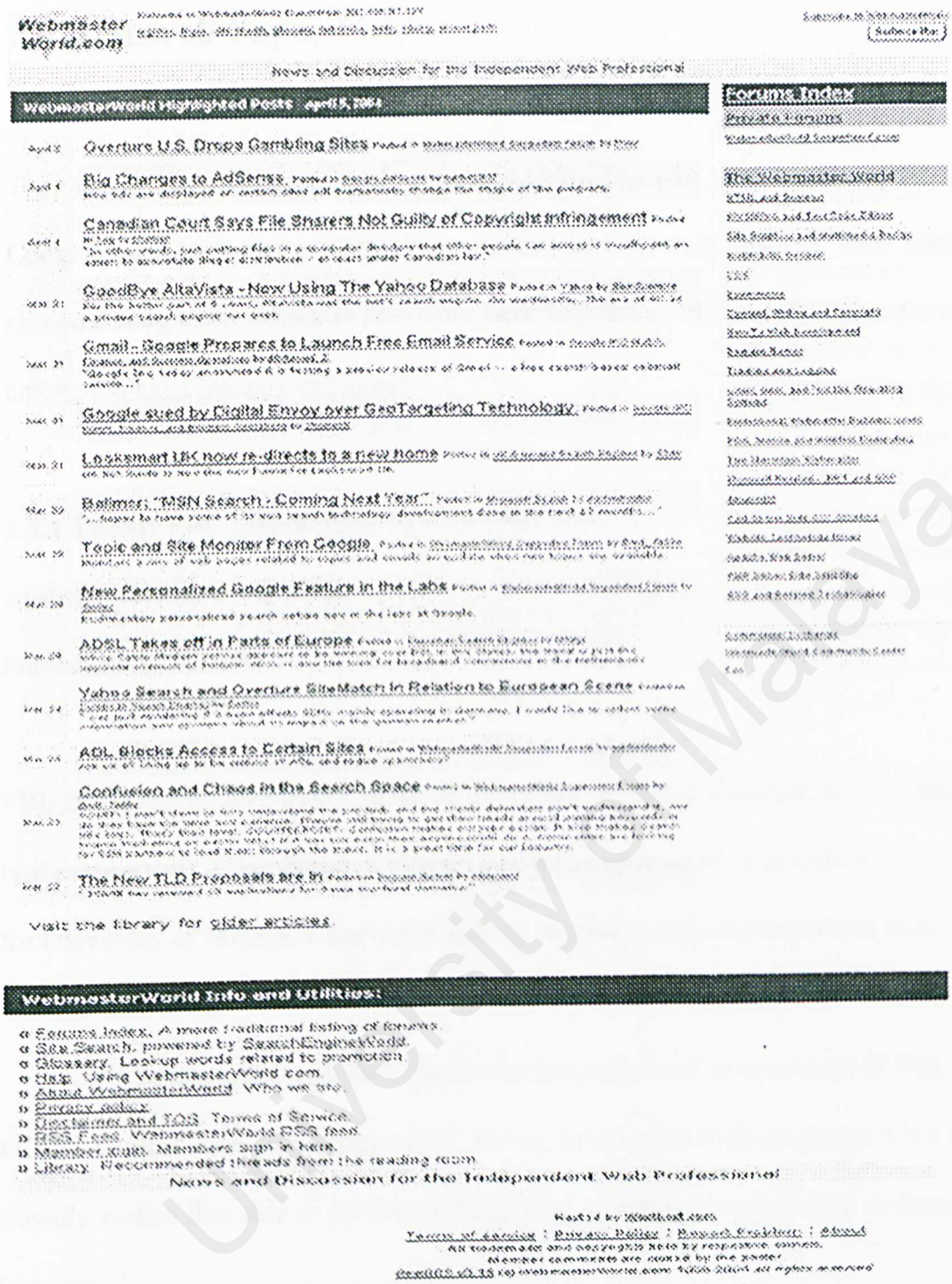


Figure 2.2: www.WebmasterWorld.com main page

2.6 Research Paper

The three theses were researched on were published by past year Faculty of Science Computer and Information Technology students at University Malaya. They were chosen among many others as they were most relevant to PBL Facilitator and were among the most recently published.

2.6.1 Thesis: PBL Information System (PBLIS)

Author: Nicholas Barau Eddy

Published: 2003/2004

PBL IS a dynamic web-based client/server system intended to create an information rich environment to be shared in interactions between students and lecturers. It's built for University of Malaya usage and it able to facilitate implementation and data management of PBL.

It's built mainly with the objective to automate data management of PBL in FCSIT. Created with technology of NT Server, MySql and PHP language, it's a user-friendly system that able to generate strong level of interactions between students and lecturers.

The system is like a management system for a course that applies PBL. Its functions consider the whole management from the dean to the students with different levels of privileges. There are links to PBL journals and resources. There are many interesting features like creation of journals, journal templates, journal submission, and etcetera. Course and peer evaluation is also available and members can create forums for discussions.

PBL IS has a logical flow of interface. Its strength lies in its portability, error detection, internal search engine, security, and plug-in module addition capability. The system however, lacks of database backup, has no mailing function, and requires full user validation and no graph and chart creation.

2.6.2 Thesis: e-Forum

Author: Suresh Kaliadas

Published: 2003/2004

E-Forum is a web-based application intended for use of professional/non-professional groups, education institutions, advertising company and many others. It was developed using PHP and MySql technologies on Windows 200 platform and Apache server. It can be viewed either on Internet Explorer or Netscape browser.

This system boasts of its friendliness and easy GUI. It provides authorization as part of security control and provides search engine to enable user to look up for other members, subjects and topics. The contents are upgradeable from time to time and administrators can bane sensitive words from the message board. The system can be accessed on any platform without any restriction on any hardware or software requirements.

Non-registered users can access the system and view all the forums, however they aren't allowed to participate in any of it. e-Forum members can form their own group discussion for any topics, do maintenance on the group they created and join existing groups. However, they need to get administration approval before they can join a new discussion topic. The discussion boards are provided with security control.

Registered users can mail each other by U2U function. Members can look up for other online members via who's online function. Privacy of members is protected.

It also provides other interesting features like private messaging. e-Forum however, lacks of any user help function like forgotten password and FAQ. There's also no uploading function in this system

2.6.3 Thesis: Online Meeting System

Author: Ong Yu Kheng

Published: 2003/2004

Online Meeting System is a web-based system that enables organization to conduct meeting or conference online. It is developed using Java and Microsoft SQL technology to reduce unnecessary operating cost and wasting time and resource for face-to-face meeting. With this system, team people in need for communicating and cooperating for certain purpose can solve their dilemma easily. The users targeted are team members of virtual and traditional organizations, and academic societies.

The system has user-friendly GUI and provides multiple user environments as according to level of access by users. The system is simple with straightforward linking with point and click mechanism. Administrator and moderator will find it easy to maintain the system as it provides easy and functional system management and room management.

Its disadvantages however are lack real time environment, limited meeting room tools, and tedious information entry for registration.

2.7 Tools

The tools that will be explained included the programming languages and database that will be used to build PBL Facilitator, and also the development and graphic software.

2.7.1 Web programming

2.7.1.1 PHP

PHP is an open-source server side scripting language that is imbedded into HTML. PHP can perform tasks that any CGI program can do. Its strength is in its compatibility and simplicity with many types of database like Oracle, MySql, ODBC and many others, to generate content dynamically for very powerful application. It allows integration with external libraries like FPDFLib that generates PDF documents and other programming languages like C++, JavaScript and the like.

According to Zend Technologies, there are 8 reasons among many why PHP is a good solution to create dynamic sites. The reasons are:

1. short development time, which means fewer programmers are required.
2. PHP is easy to maintain and update compared to Perl or ASP.
3. PHP scripts works on any Internet servers including Apache, Microsoft and Netscape service solutions.
4. PHP allows accesses to commercial and non-commercial databases.
5. There are more than 6 million domains that use PHP.
6. PHP is open source, and there is definitely no hidden cost.
7. PHP is open API, which means that modules can be added into the script to.
8. PHP is successful on mission-critical sites and application

2.7.2 Programming Language

2.7.2.1 HTML

According to Ian Graham (1998) HTML or HyperText Markup Language is designed to specify the logical organisation of a document with important hypertext extensions, and not to be the language of a WYSIWYG word processor such as Word or WordPerfect. Different browsers of different abilities can view a HTML document.

2.7.2.2 JavaScript

According to webopedia , JavaScript is designed by Sun Microsystems and Netscape to help programmers design interactive websites. Javascript is an adjunction interpreted language or script language to Java and can shares the features and structures of the full Java language. It was however developed independently. Javascript can be added to HTML source code and interpreted by the web browser to create a dynamic environment and it has considerable use in creating interactive web-based forms.

An online article by Steve Littler (JavaScript Defined) states that Javascript can do several things such as:

- Changed a formatted date on a page automatically
- Cause text or graphic image to change during mouse roll-over
- Make a pop-window appear

2.7.2.3 C++

C++ is defined by webopedia(2002) as a superset of the C language. It is a high level programming language developed by Bjarne Stroustroup at Bell Labs. C++

is an object-oriented programming language that's considered as the best language for creating large-scale applications and is most popular for graphical applications that runs in Windows and Macintosh environments.

2.7.2.4 VBScript

cnet.com defines Visual Basic Scripting Edition (VBScript) as a programming language developed by Microsoft for creating scripts (miniprograms) that can be embedded in HTML Web pages for viewing with Internet Explorer. These scripts can make Web pages more interactive. VBScript also works with Microsoft ActiveX Controls, allowing Web site developers to create forms, interactive multimedia, games, and other Web-based programs. VBScript is similar in functionality to JavaScript and is a subset of the widely used Microsoft Visual Basic programming language.

2.7.3 Database

2.7.3.1 MySql

According to definition by TechTarget, MySql is an open source relational database management system (RDBMS) that uses structured query language (SQL)for adding, accessing and processing data in a database. As it's open source, it is free under the terms of General Public License (GPL) or free without bound to the terms. MySql can be downloaded from its official web site www.mysql.org.

MySql provides APIs for C, C++, Eiffel, Java, Perl, PHP and Phyton. Mysql is most commonly used for web applications and embedded applications because of its speed and reliability. It can run on UNIX, Windows and Mac OS.

2.7.4 Web Server

2.7.4.1 IIS

According to TechTarget definition, IIS is short for Internet Information Server. IIS is a Microsoft's web server that runs on Windows NT platforms. IIS is easy to handle as it is tightly integrated with the operating system. Unlike Apache and Netscape's Web servers, IIS can only run Windows NT platforms.

IIS is a group of Internet servers with additional capabilities for Microsoft's Windows 2000 and above operating systems software. It provides both FTP server and web server capability. Computers with IIS can create web pages using Microsoft's Front Page and other tools.

2.7.5 Web Development Software

2.7.5.1 Macromedia Dreamweaver MX

With Macromedia Dreamweaver and its WYSIWYG (What You See Is What You Get) environment, there was no more head scratching for those who weren't trained programmers; no more writing HTML by hand for beginners; and no more puzzling over cryptic lines of code. The average Web user could design pages even though they had little or no code knowledge. It's intuitive and have flexible visual authoring environment, Roundtrip HTML, and support for Dynamic HTML.

Dreamweaver MX, available for Macintosh and Windows, was released in summer 2002. And for the first time, designers, developers and programmers can work within one environment to easily create and manage any professional Web site,

whether it's build with HTML, XHTML, XML, web services, ColdFusion, ASP.NET, ASP, JSP or PHP. In Dreamweaver MX, Macromedia has combined the functionality of what in the past had been three separate products: the visual layout tools of Dreamweaver; the rapid web application features of Dreamweaver UltraDev; and the extensive code editing support of HomeSite to provide a complete solution for developing websites.

Chapter 3 Methodology

The model that had chosen for this project is waterfall with rapid prototyping model. This model is a combination result of waterfall model and rapid throwaway prototyping model.

3.3 Waterfall with prototyping model

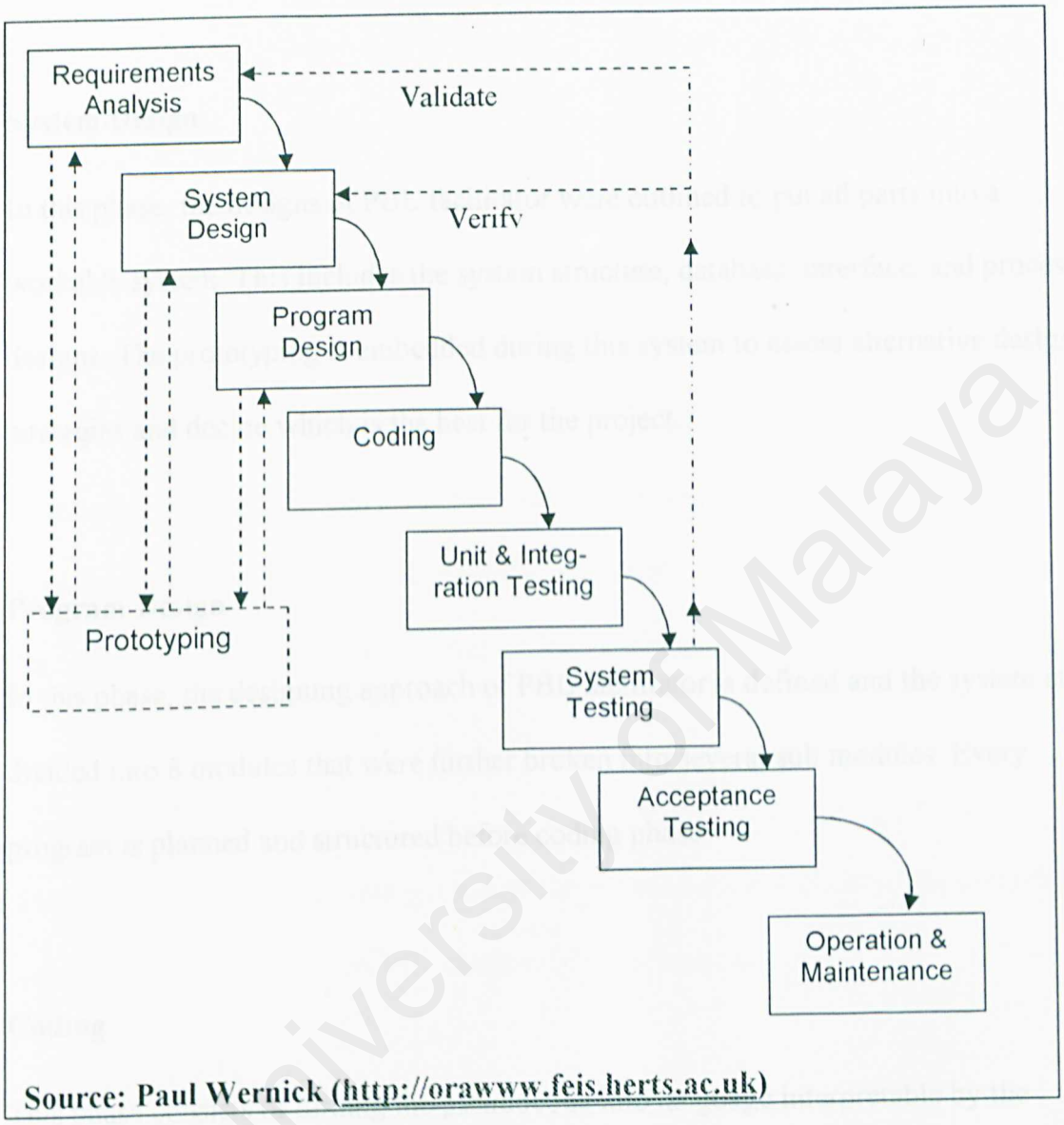
Waterfall with prototyping model is depicted in Figure 3.1. The requirements, system design and program design in waterfall model is integrated with prototyping to improve system understanding. A prototype is sort of a toy version of the system. It's developed in a low cost and short time to explore possibilities and confirm the requirements. Aside from enabling developers to capture accurate requirements, this will also allow client to have an early view on the system (Sellappan, 2000; Sommerville, 2001).

Many other models are embellished from this chosen model, as it provides:

- simplicity of explanation
- systematic and organized
- makes explicit which intermediate products are necessary to begin the next phase

According to Paul Wernick, with this model, the requirements are prototyped, and developers will conduct a fast design and implementation, followed by a quick test. The clients then can use the prototype as an experiment. During this experiment, observations are made and comments from the clients are captured for further

references. The information will be added to the requirements, and perhaps process iteration will be done. The prototype then will have to be thrown away.



Source: Paul Wernick (http://orawww.feis.herts.ac.uk)

Figure 3.1: Waterfall with prototyping model

Requirement Analysis

During this phase, all the requirements of the system are gathered and prototyped. This is to ensure that the requirements gathered are practical and feasible. The requirements for the PBL facilitator are collected via the side interviews from several students of FSKTM, most of whom have experienced in participating PBL discussions. If the requirements were proven inadequate or not practical, revisions will be made where the information needed were gathered through researches on

previous paper works and web sites that are relevant to the system and definitely more interviews.

System Design

In this phase, the designs of PBL facilitator were outlined to put all parts into a workable system. This includes the system structure, database, interface, and process designs. The prototyping is embedded during this system to assess alternative design strategies and decide which is the best for the project.

Program Design

In this phase, the designing approach of PBL facilitator is defined and the system is divided into 8 modules that were further broken into several sub modules. Every program is planned and structured before coding phase.

Coding

This phase consists of turning the pseudocode into language interpretable by the machine. The programming languages used are PHP, HTML, Javascript, C++ and Mysql query language. The coding is generated and modified using Macromedia Dreamweaver MX.

Unit and Integration Testing

In this phase, all the coded programs were evaluated and tested. This was to ensure intractability between all the programs and subprograms. All the subprograms are also

tested for its efficiencies and stabilities. All errors and faulty were discovered during this period.

System Testing

The PBL facilitator was tested as a whole system in this phase. The system was tested with sample data and then verification and validation processes was conducted. The system was validated to ensure that the system has implemented all the requirements; so that each system function can be traced back to a particular requirement in the specification. This is to ensure that the developer is building the right product according to the specifications. The verification to the system is conducted to ensure that each function works correctly and to determine the quality of the implementation.

Acceptance Testing

In acceptance testing, the users got to test the system with actual data. The users then evaluated and commented on PBL facilitator's performance.

Operation and Maintenance

As this system is fairly simple to use, majority of users of the system; student weren't given training. Facilitators however, were given simple briefing (written or oral) about the system, while administrators were trained on how to use and maintain the system. User manual were also produced and from this stage onward, it's system maintenance until the system is terminated.

3.2 Requirements Gathering

As stated during requirements analysis, users' requirements are collected via interviews to students of Faculty of Science Computer and Information Technology, University of Malaya. Most of them are final year students whom have experience in PBL courses.

From the interviews, what the users wants can basically compiled into 5 criteria:

1. The system must be user friendly, interesting and easy to use.
2. Facilitator must be around to supervisor and response to student's inquiries, though not necessarily in real time.
3. The system must allow performance evaluation during discussion sessions.
4. The system must be secure.
5. The system must have fast response time.

3.4 Fact Finding

To do this project, a lot of information is needed as references and project's contents. The information is gathered via several methods:

1. Internet searching
2. Reading material
3. Researches
4. Discussion with friends
5. Discussion with supervisor

Internet searching

There is information about anything you can think of on the Internet. There are more than millions of web site on the Internet that contain information in various formats; text, images, video, audio and animation. Almost every topic that has been discussed in this report was referenced on the articles and documents that were found on the Net. The search engine that is used as the information retrieval tool for this project is Yahoo!

Yahoo! can be accessed at http://www.yahoo.com

However, it is most important to know the correct way in retrieving information via the tool that's being used. A user has to use the correct combination of keyword and operator to get the right information. Otherwise, the information retrieved would be out of the desired scope with broader scope of information, or it will return the wrong information altogether.

Several of the keywords that were used in order to look up for information used in this project were:

1. "waterfall with prototyping model" - this query returns lists of links to documents and web sites that are relevant to waterfall with prototyping model.
2. "Apache" - this query returns lists of web sites that contains information regarding Apache web server.
3. "PBL" - this query returns lists of web sites that contain information about problem-based learning.

Existing online systems that are related to this project were also accessed. The purpose is to identify common features and characteristics of good web sites. Web sites that provide e-forums and active online message boards were analyzed to

identify the factors that interest users and characteristics of well-managed online discussions.

Several of the systems are:

1. http://www.phpbuilder.com - this web site hosts the latest information about php and owns a very active community that interacts via forums provided to discuss problems and solutions about php on everything.
2. http://www.webmasterworld.com - this web site hosts public and private forums for independent system web developers for news and discussion.

Reading material

The reading materials referred for this project consists of the reference books that were borrowed from the UM Library and also the developer's. Several of the books were:

1. Software Engineering by Dr. P. Sellappan (2001)
2. System Analysis and Design Methods by Jeffrey L. Whitten, Lonnie D. Bentley and Kevin C. Dittman
3. Professional PHP Programming by Jesus Castagnetto, Harish Rawat, Sascha Schumann, Chris Scollo and Deepak Veliath.

These books were important, as they were sorely needed in helping developer to understand and also to provide insight in how to conduct system development, and helping in the coding and database querying.

More reference were done based on the notes of the courses that the developer's had taken in previous semesters, such as Database and Information Resource Management courses. But the most important material was the Guide on WXES3181

report writing that can be downloaded from the faculty's web site, as it provides the basic and important guideline in writing the report.

Researches

Researches were conducted on projects and thesis by past year FSKTM's students. All the reports are available in the Document Room at the faculty. The purpose of the researches is to have examples, guidelines and ideas on building a good online PBL classrooms and fro the report writing. Most of the thesis that were referred to were the ones that are relevant to the PBL facilitator system.

There were three theses from many that were found most relevant:

1. PBL Information System by Nicholas Barau Eddy (2003/2004)
2. e-Forum by Suresh Kaliadas (2003/2004)
3. Online Meeting System by Ong Yu Kheng (2003/2004)

Although the scope and definition of the theses are not the same as PBL facilitator, they provide useful information related to the requirements and designs for the system. For example, they provide good examples on the modules that are necessary in web sites that enable communication and collaborative discussions.

Discussion with Friends

Discussions with friends were often held about reports and projects. Weaknesses and errors were pointed to each other and advices were freely given. Discussions were mostly done via email, telephone or face to face at the faculty.

Discussion with Supervisor

Discussions with the project supervisor, Mrs. Kiran Kaur a/p Gurmit Singh was mostly held at her office at the faculty once a week, otherwise via email. Everything about this project, from the report writing, to the system development was discussed. From her, the clearest information about how to successfully built the system was gathered. She guided and gave advice on what is best for the project, and corrects whenever there's error. She was the most important person throughout this project.

Chapter 4 System Analysis

4.1 Requirements Analysis

According to Sommerville (2001) requirements of a system are descriptions of services that are required and constraints on operation and development of the system. Requirements analysis answers the question, "What do users need and want from the new system?" not on the how (Whitten *et al*, 2002). The requirements are established after consultation with user through requirement engineering process. Requirements engineering is a process of finding out, analysing, documenting and checking the requirements. It is a very important phase as errors at requirements analysis phase could cause problems in the system design and implementation phase.

The requirements are sometimes not easy to verify. This is because user sometimes cannot determine what they really wants, and sometimes they are not consistent in their demands.

There are two types of requirements that will be covered in this chapter:

1. Functional requirements
2. Non-functional requirements

4.1.1 Functional Requirements

Functional requirements of the system describe the services and activities PBL Facilitator must provide, the way the system should react to certain inputs and the way the system should behave in certain situations. The requirements should be complete, where the supposed services are all defined and consistent, which means

that that the definitions should not be contradictory (Sommerville, 2001; Whitten *et al* 2002).

The functional requirements of the system are:

Log In Module

Log In module enable user to log in into the system, and logging out of the system. The system will perform authentication process to determine authorization status before admitting a user (facilitator or student) into the system. Whenever a user logs on to the system, the system will record the time, period and date of entry. This module also enable user to change password and inquire for password should a user forgotten or lost his password.

1. **User Log In**
    a. The system should enable user to log in using a user ID and a password.
    b. The system should be able to identify in which course(s) a student or a facilitator belongs and authorized to access.
        i. Students and facilitators are only authorized to access into the classrooms in which they are registered.
    c. The system should prompt a message if user input wrong user ID, password and course match.

2. **Change Password**
    a. The system should allow user to change password for security purpose.

3. **Password Enquiry**

    a. The system should enable user to enquiry for their password should they forgotten their password.

    b. The system should be able to immediately post the password to the enquirer automatically via email.

4. **Log In Entry**

    a. The system should be able to record the date and time whenever a user access into the system.

Group Management Module

This module enables the system to automatically assign students randomly into groups of fixed size after all students of any classrooms have been registered.

1. **Group Assignment**

    a. The system should be able to automatically assign students into groups of fixed size in random.

2. **Create Discussion Rooms**

    a. The system should be able to create discussion rooms for registered courses.

        i. The number of discussion rooms for a course is according to the number of groups in the course.

Evaluation Module

This module enables students' performance evaluation during discussions. The system will capture students' responses in discussions and do evaluation based on the information captured and posts the transcript and the evaluation result to the facilitator(s) of the course in charge. Discussion group members are also able to evaluate each other. Student can view the evaluation result after a discussion session has ended.

1. **System Evaluation**
    a. The system should be able to capture the number of response of each discussion member during a discussion session and evaluate each student's performance based on the number of response of that particular student.
    b. The system should be able to post the evaluation result to the facilitator(s) of the course in charge.
    c. The system should instantly display the evaluation result to the student after the discussion session has ended.

2. **Transcript Record**
    a. The system should be able to record transcript of discussion and post it to the facilitator(s) of the course

3. **Peer Evaluation**
    a. The system should enable members of a discussion to evaluate each other after a discussion session has ended.

b. The system should allow the members of a discussion to view the peer evaluation.

Discussion Rooms Module

This module handles groups' discussion rooms. The module will enable the system to identify which students belong to which groups and identify which students are allowed to enter a discussion room as guest and which students are members of a discussion room and can participate in the discussion. Facilitator(s) are also able to interfere and participate in discussions in a classroom that the facilitator is in charge of.

1. **Message board**
   a. The system should be able to allow members of a discussion room to posts messages on the room's message board during a discussion session.

2. **Discussion Room**
   a. The system should be able to display user ID of members, facilitators and guests that enter a discussion room.

3. **Access authorization**
   a. The system should be able to identify the members of a particular discussion rooms.

i. The students should be allowed to enter the room and participate in any discussion held in the room and labelled as 'Member'.

b. The system should be able to identify students that are not members of a particular discussion room.

i. The students should be allowed to enter the discussion room labelled as 'Guest' and can only view the discussion held but cannot participate.

c. The system should be able to identify the facilitators and labelled them as 'Facilitator'.

i. The facilitators are allowed to enter all the discussion rooms and interfere.

Administrator module

This module enables administrator(s) to manage the registration of facilitators. The administrator will be able to update the data in the system's database and posts notices on the system to the system users. The module also enable administrator to terminates a classroom or the membership of a class member if found necessary. Administrator can also register new administrators. However, new administrators do not need to confirm their registration.

1. **Administrator's Tools**

a. The system should be able to allow administrator to update the data in the system's database.

b. The system should enable administrator to posts notices on the system

a. the system should be able to allow facilitators to assign case studies to groups.
b. the system should be able to allow facilitators to post messages on classroom's notice boards.
c. the system should be able to allow facilitators to upload documents, feedbacks and links on the system.
d. the system should enable facilitators to terminate a student's membership if found necessary.

Student Module

This module allows students to posts messages on classroom's main message board and uploads documents like question's answers on the system.

1. **Student's Tools**
   a. the system should be able to allow students to post messages on the classroom's main message board.
   b. the system should be able to allow students to upload documents on the system.

User Register

This module enables user to confirm their registration. Administrator registers administrators and facilitators, and facilitators register students. Users need to input their details and change the password given in order to confirm their registration.

1. **User Register**

   a. the system should be able to allow user to confirm user's registration on the system.

      i. The system should allow user to create a username and change the password given by the system.

      ii. The system should allow user to input their details.

4.1.2 Non-functional Requirements

Non-functional requirements are requirements that are not directly concerned to the services PBL Facilitator provides. Instead, they describe on other features, characteristics, and constraints that define the system (Whitten *et al*, 2002).

The non-functional requirements of the system are defined as followed:

User friendliness

The system must be easy to understand and user friendly, without appearing too formal an environment. This is very important, as one of the purposes of the system is to provide an alternative environment for discussion. And the system mustn't be intimidating in order to encourage students to actively participate in the system.

The system should also be able to handle errors such as wrong input from users. The system must be able to accommodate any level of user, from the secondary to the universities student.

Ease of Use

The system must be very easy to understand that minimal trainings are required for the administrators and possibly no trainings at all for the users.

Language

The language must be written in a way that can be understood by all non-technical users. The best choice of language for PBL Facilitator is probably English. The information on the system that is provided by the system such as instructions mustn't include any jargons to avoid any possible confusion among users.

Availability

The system must be reliable and not cause unnecessary and unplanned downtime.

Security

The system must also be secure from unauthorized access. And the security of the data must be confirmed.

Error Detection

The system must be able to detect whenever an error occur and prompt a message to alert the user or the administrator.

Efficiency

The system must be efficient in its function.

Response Time

All the transactions must be done in a reasonable period of time. The transactions include the registration process, message posting, file uploading and several others.

Expendability

The system must be able to be upgraded and modified in order to accommodate the changes in future development.

Scalability

The system must be able to provide the capability to migrate as a client or a server to a machine of lesser or grater power depending upon the requirements, with little or no changes at all to the components.

Link

The must provide only active links.

4.2 Hardware Requirements

4.2.1 Developer

Table 4.1: Developer's and user's hardware requirement

Minimum requirements	Suggested requirements
Intel Pentium I Processor (166MHz)	Intel Pentium 4 Processor (2.8GHz)
64 MB SDRAM	256 MB SDRAM
5 GB hard disk	40 GB hard disk
Windows 98	Windows XP

4.2.2 User

Minimum requirements:

- Intel Pentium 2 Processor with 300 MHz or above
- 64 MB SDRAM or above
- 100 MB free hard disk

4.3 Software Requirements

4.3.1 Developer

Minimum requirements:

- PHP
- MySql
- Development tools like Microsoft FrontPage/Macromedia Dreamweaver/Notepad
- IE 4.0 browser or above

4.3.2 User

Minimum requirements:

- IE 4.0 browser or above

Chapter 5 System Design

The system design was built on the knowledge obtained from the analysis phase. It focused on the logical and physical aspects of the system (Sellappan, 2000).

5.1 Database Design

The name of the database chosen for this system is PBL Facilitator. And there are yy basic tables in PBL Facilitator. The tables are:

1. Table name: Temporary_Facilitator

Function: Administrator approves facilitator's apply for membership. The table will only store the data sent by facilitator via the online form, and the password and username generated by the system. All the data will be stored in tis table until the application has been approved and the facilitator has confirmed his registration.

Table 5.1: Temporary_Facilitator data dictionary table

Fieldname	DataType	Size	Key	Description
userName	varchar	11		Username generated by system to be used to confirm registration.
userPword	varchar	8		Password generated by the system to be used to confirm registration.
email	varchar	120		User's email address
Institute	varchar	80		Institute where the facilitator from or teaches
Faculty	varchar	80		Faculty where the facilitator from
No	int	3	primary	sequence number of unregistered facilitator
AppStatus	char	1		The status of the application. 'Y'=approved, 'N'=not approved.
Name	varchar	75		Facilitator's full name

2. Table name: Temporary_Student

Function: Facilitator registers students. The table will store data about the student given via the online from by the facilitator. The system generate username and password that are given to the facilitator to be given to the student. The data will be stored until the student has confirmed his registration, and then they will be deleted.

Table 5.2: Temporary_Student data dictionary table

Fieldname	DataType	Size	Key	Description
userName	varchar	11		Username generated by system to be used to confirm registration.
userPword	varchar	9		Password generated by the system to be used to confirm registration.
CourseID	varchar	9		Code of the course registered to.
GroupNo	int	3		Student's assigned group number.
FacultyID	varchar	80		Faculty's ID where the student from

3. Table name: LogIn_Entry

Function: store date and time each time a user logs into the system.

Table 5.3: LogIn_Entry data dictionary table

Fieldname	DataType	Size	Key	Description
userID	varchar	9		User's user ID
DateEntry	date			The date of the entry.
TimeIn	time			The time when the student logs in.
TimeOut	date			The time when the student logs out. The defaul value is: 00:00:00
No	int	3	primary	The number of each wntry in sequence.

4. Table name: Administrator

Function: stores data about the system's administrator.

Table 5.4: Administrator data dictionary table

Fieldname	DataType	Size	Key	Description
AdmID	varchar	9	primary	Administrator's user ID
AdmEmail	varchar	120		Administrator's email address
AdmAltEmail	varchar	120		Administrator's alternate email address
AdmName	varchar	75		Administrator's full name.

5. Table name: Facilitator

Function: store data about the registered facilitators in the system.

Table 5.5: Facilitator data dictionary table

Fieldname	DataType	Size	Key	Description
FID	varchar	9		Facilitator's user ID
FName	varchar	75		Facilitator's full name.
FEmail	varchar	120		Facilitator's emaill address.
FAltEmail	varchar	120		Facilitator's alternative email address
FacultyID	int	3		Faculty's ID where the facilitator from.

6. Table name: Student

Function: store data about the registered students in the system.

Table 5.6: Student data dictionary table

Fieldname	DataType	Size	Key	Description
StID	varchar	9	primary	Student's user ID
StName	varchar	75		Student's full name.
StEmail	varchar	120		Student's emaill address.
StAltEmail	varchar	120		Student's alternative email address
FacultyID	int	3		Faculty's ID where the student from.
Matric_Num	varchar	9		Student's matric number.

7. Table name: Uaser_Password

Function: store all users' username and password.

Table 5.7: User_Password data dictionary table

Fieldname	DataType	Size	Key	Description
UserID	varchar	9	primary	User's user ID
UserPw	varchar	8		User's password
Status	char	1		User's status. 'A'=administrator, 'F'=Facilitator, 'S'=student.
LogStatus	char	1		User's status whether currently logged in or not. 'Y'=yes, 'N'=not. Default value is 'N'.

8. Table name: User_Course

Function: store data which students registered to which course.

Table 5.8: User_Course data dictionary table

Fieldname	DataType	Size	Key	Description
UserID	varchar	9		User's user ID
CourseID	varchar	9		The course the student registered to.
UserStatus	varchar	1		User's status. 'A'=administrator, 'F'=Facilitator, 'S'=student.

9. Table name: Course

Function: store data about the courses registered in PBL Facilitator.

Table 5.9: Course data dictionary table

Fieldname	DataType	Size	Key	Description
CourseID	varchar	9		The course the student registered to.
NumOfSt	int	3		Number of students registered into the class.
NumOfF	int	3		Number of facilitators registered into the class.
DateCreated	date			The date the class was registered.
CourseName	varchar	75		The full name of the course.

FacultID	int	3		Faculty's ID where the course's members from.
MaxPGroup	int	2		Maximum of student in a group.
Creator	varchar	9		Facilitator's ID whom registered the course.

10. Table name: Faculty

Function: store data about the faculties that the system's users are from.

Table 5.10: Faculty data dictionary table

Fieldname	DataType	Size	Key	Description
FacultyID	int	3	primary	Faculty's ID.
Faculty	varchar	80		The name of the faculty.
Institute	varchar	80		The name of the institute.

11. Table name: Adm_Noticeboard

Function: store the annoncements that the administrator posted on the syste's interface.

Table 5.11: Adm_Noticeboard data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3	primary	The number of announcements in sequence.
Message	text			The announcement.
Location	char	1		The location of the announcement. 'L'=login page, 'F'=facilitator's page, 'C'=main page.
LastUpdated	date			The date when the announcement was last updated.
TimeUpdated	time			The time it was last updated.

12. Table name: Adm_EmailReport

Function: store the comments/queries that the administrator received from users.

Table 5.12: Adm_EmailReport data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3	primary	The number of posts in sequence.
DateSend	date			The date the post was posted.
TimeSend	time			The time it was sent.
Subject	varchar	25		The subject of the post.
FromAddress	varchar	120		User's email address.

13. Table name: <courseID>_disctranscript

Function: store data about the discussions held by the students.

Each course will have this table. Example: course wxes3181 will have a wxes3181_disctranscript table.

Table 5.13: <courseID>_disctranscript data dictionary table

Fieldname	DataType	Size	Key	Description
GroupNo	int	2		The number of the group.
DatePosted	date			The date the discussion was held.
TimePosted	time			The time the discussion was started.
DiscNo	int	3		The number of discussion.
Pdf	char	1		The status of the transcript, whether it has beenconverted into pdf document. 'Y'=yes, 'N'=no.
Link	varchar	75		The link to the pdf document (if it has been converted)

14. Table name: <courseID>_evaluation

Function: store data about the evaluations of a discussion.

Each course will have this table. Example: course wxes3181 will have a wxes3181_evaluation table.

Table 5.14: <courseID>_evaluation data dictionary table

Fieldname	DataType	Size	Key	Description

DiscNo	int	2		The number of the discussion.
UserID	varchar	8		Student's user ID
Facs_Eva	int	2		Facilitator's evaluation result.
Syst_Eva	int	2		System's evaluation result.

15. Table name: <courseID>_groupassignment

Function: store data about the uploaded documents relating to students' assignment.

Each course will have this table. Example: course wxes3181 will have a

wxes3181_groupassignment table.

Table 5.13: <courseID>_groupassignment data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	2		The number of the document.
GroupNo	int	2		The number of the group.
FileTitle	varchar	75		The given title of the document.
FileLink	varchar	75		The link to the document.
TypeOfDoc	char	1		The category of the document. 'A'=solution, 'Q'=situation, 'G'=group's solution, 'F'=feedback.
UserID	varchar	8		The user whom uploaded the file.
OptName	varchar	20		Optional name for the file.

16. Table name: <courseID>_Links

Function: store data about the links on the course's classroom.

Each course will have this table. Example: course wxes3181 will have a

wxes3181_Links table.

Table 5.16: <courseID>_Links data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3		The number of the link.
FID	varchar	8		facilitator's ID whom addedd the link
LinkTitle	varchar	25		The title of the link.
Link	varchar	75		The link (url).
LinkType	char	1		The category of the link: 'W'=web

				sites, 'D'=documents or files.
Descrpt	text			Description of the link.

17. Table name: <courseID>_listgroup

Function: store list of groups in a course.

Each course will have this table. Example: course wxes3181 will have a wxes3181_listgroup table.

Table 5.17: <courseID>_listgroup data dictionary table

Fieldname	DataType	Size	Key	Description
GroupNo	int	2		The number of the group.
DiscStatus	char	1		The status of the group. Whether in discussion or not. 'Y'=yes,'N'=no.
TotalMember	int	1		Number of students in the group. Default value=0.

18. Table name: <courseID>_messageboard

Function: store data about the messages posted on the classroom's message board.

Each course will have this table. Example: course wxes3181 will have a wxes3181_messageboard table.

Table 5.18: <courseID>_messageboard data dictionary table

Fieldname	DataType	Size	Key	Description
Message	text			The message posted.
MemberID	varchar	8		The user ID of the poster.
TimePosted	time			The time it was posted.
DatePosted	date			The date it was posted.
No	float	3,1		The number of the message.
MssgTitle	varchar	75		The title of the post.
Replies	int	3		The number of replies for the posts. Default value=0
CorrspmdTo	float	3,1		The number of the post threat that it

				corresponds to. If it itself is a thread, then the value=0.0.

19. Table name: <courseID>_noticeboard

Function: store the announcements posted by the facilitators on the classroom noticeboard.

Each course will have this table. Example: course wxes3181 will have a wxes3181_noticeboard table.

Table 5.19: <courseID>_noticeboard data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3		The number of the notice.
Message	text			The notice.
FID	varchar	8		Facilitator's ID whom posted the post
DatePosted	date			The date it was posted.
TimePosted	timw			The time it was posted.
Location	char	1		The location it was posted. 'M'=the main page of the classroom's page, or 'x'= the Discussion Room 'x' (x=one of the group numbers)

20. Table name: <courseID>_complaint

Function: store complaints/comments from student to facilitator.

Each course will have this table. Example: course wxes3181 will have a wxes3181_complaint table.

Table 5.20: <courseID>_complaint data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3		The number of the complaint.
Complaint	text			The complaint.
ComplaintTilte	varchar	25		The title of the complaint.
DatePosted	date			the date it was posted.

TimePosted	time			The time it was posted.
StID	varchar	8		Student's ID whom posted the complaint.

21. Table name: <courseID>_peerevaluation

Function: store peers' evaluation for each discussion session.

Each course will have this table. Example: course wxes3181 will have a wxes3181_peerevaluation table.

Table 5.21 <courseID>_peerevaluation data dictionary table

Fieldname	DataType	Size	Key	Description
DiscNo	int	3		The number of the discussion.
Members	varchar	9		UserID whom did the evaluation.
Member1	char	2		The student's ID that was evaluated. This field depends on the number of students per group of a course. If the number is 3, then there are 2 more fields; Member2 and Member3.

22. Table name: <courseID>_<TheNumberOfDiscussion>

Function: store data transcripts recorded during a discussion session. If administrator converted the transcripted into .pdf document, then this table will be automaticaly deleted.

Each discussion will have this table. Example: course wxes3181 and discussion number 8, will have a wxes3181_8 table.

Table 5.22: <courseID>_grouopassignment data dictionary table

Fieldname	DataType	Size	Key	Description
No	int	3		The sequence of responses.
Member	varchar	8		The user's ID whom posted the response.
Input	text			The response

5.2 System Structure

As described in Introduction and System Analysis, PBL Facilitator is divided into 8 modules. Each module is divided into one or more functions as can be seen from the structures below.

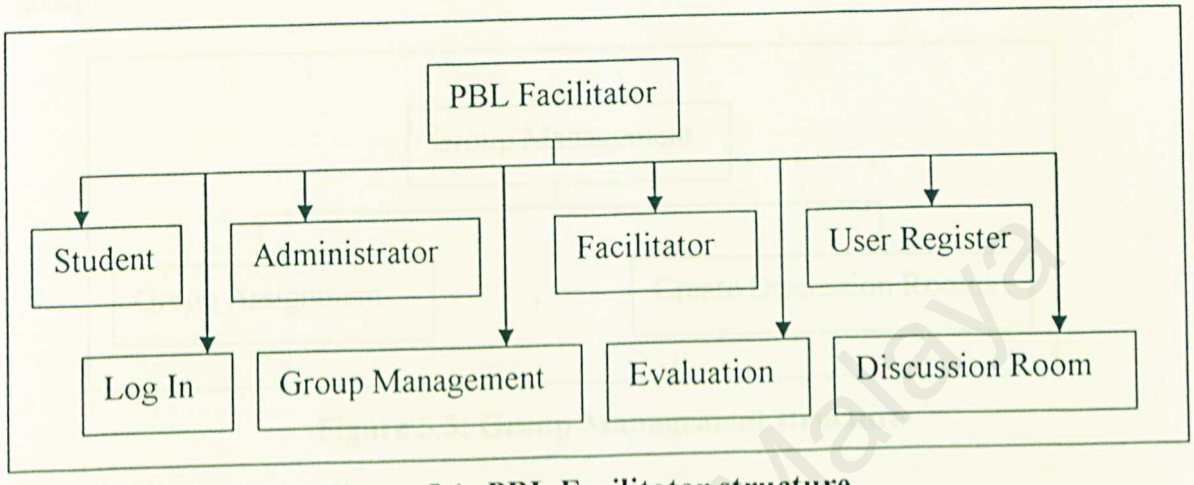


Figure 5.1: PBL Facilitator structure

The Log In module enable user to log in and logging out of PBL Facilitator. The system performs authentication process to determine authorization status before admitting a user (facilitator or student) into the system. This module also enable user to change password for security purposes and inquire administrator for password retrieval should a user forgotten or lost his password. And the system will record the time and the date whenever a user log in into the system.

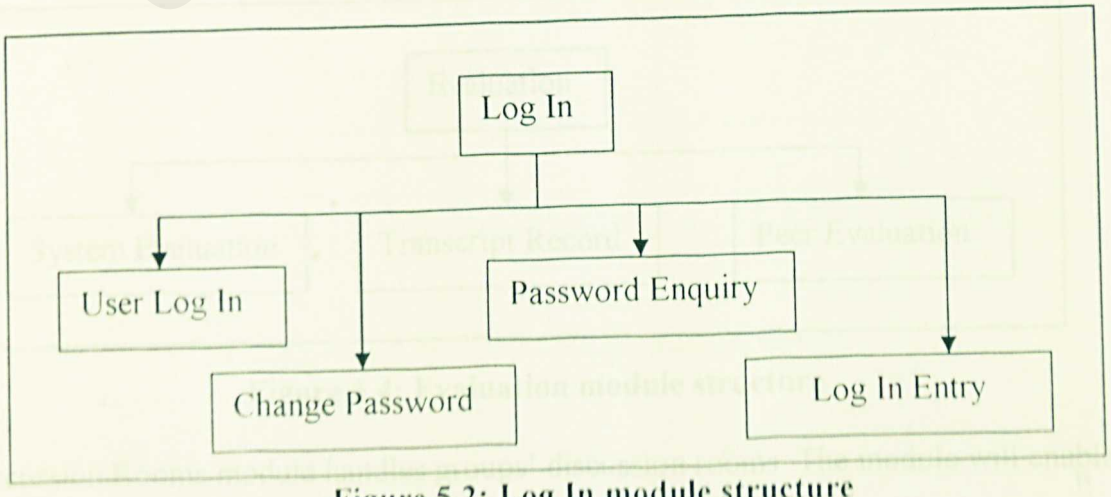


Figure 5.2: Log In module structure

Group Management module is divided into two main functions. The sub module Group Assign is responsible for automatically assigning students of a course into groups of fixed size by the facilitator after a facilitator registered a course. Another sub module as in Figure 5.3 will automatically generate a discussion room for each group.

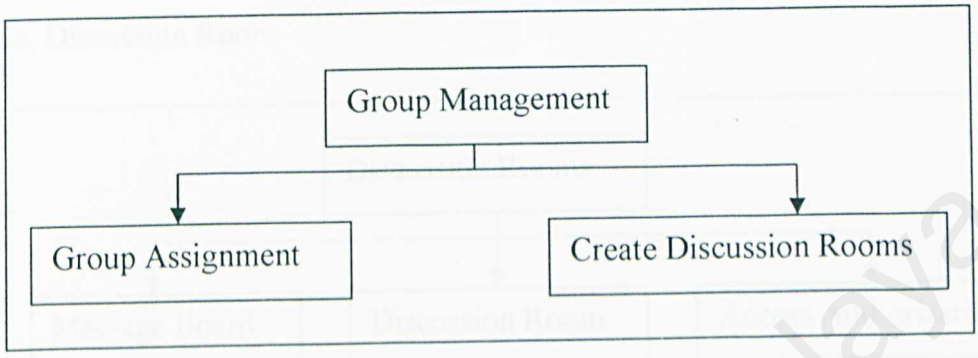


Figure 5.3: Group Management structure

Evaluation module is responsible in evaluating student's performance after a discussion. The system captures students' responses in discussions and does evaluation based on the numbers of responses and posts the evaluation result to student and also facilitator. The discussion transcript is recorded and saved for facilitator's personal evaluation. Student can view the evaluation result after a discussion session has ended and group members are also able to evaluate each other.

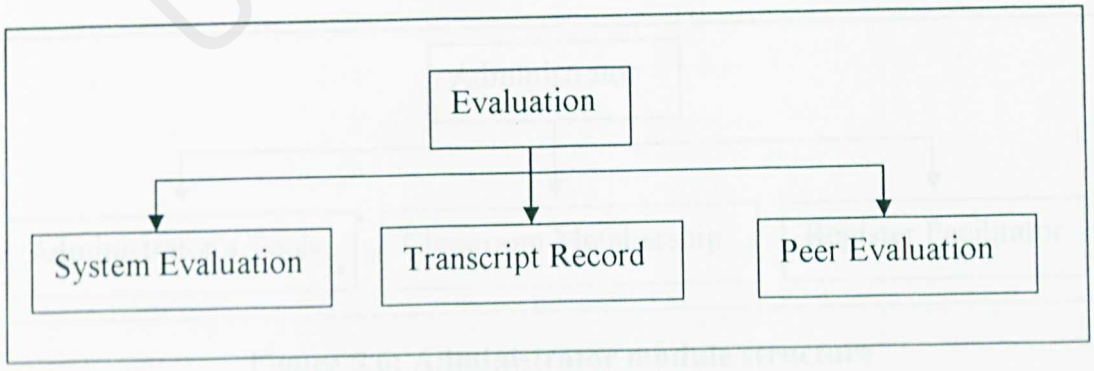


Figure 5.4: Evaluation module structure

Discussion Rooms module handles groups' discussion rooms. The module will enable the system to identify which students belong to which groups and identify which

students are allowed to enter a discussion room as guest and which students are members of a discussion room via the Access Authorization sub module. Message Board sub module allows message posting in the room. Facilitator(s) are also able to interfere and participate in discussions in a classroom that the facilitator is in charge of. And users in a discussion room can also see other users in the room via the last sub module, Discussion Room.

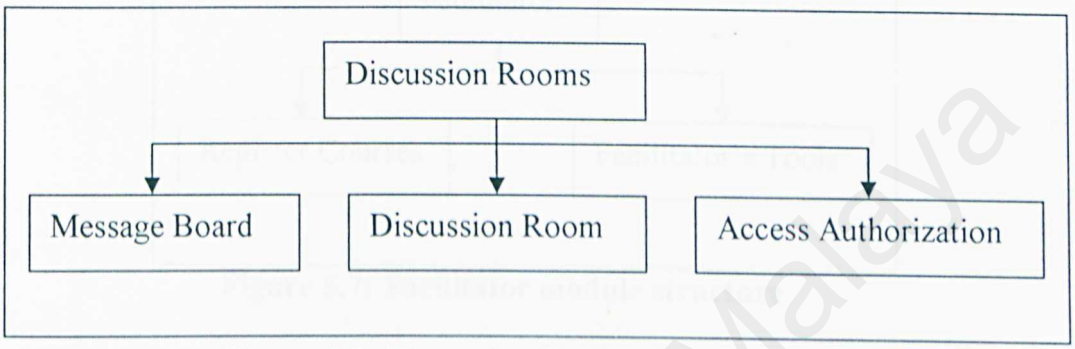


Figure 5.5: Discussion Rooms structure

Administrator module consists of three sub modules. With Register Facilitator, administrator(s) are able to manage the register facilitators. The administrator will be able to update the data in the system and posts notices on the system to the system users with Administrator's Tools. The last sub module as in Figure 5.6 below also enable administrator to terminates a classroom or the membership of a class member if found necessary.

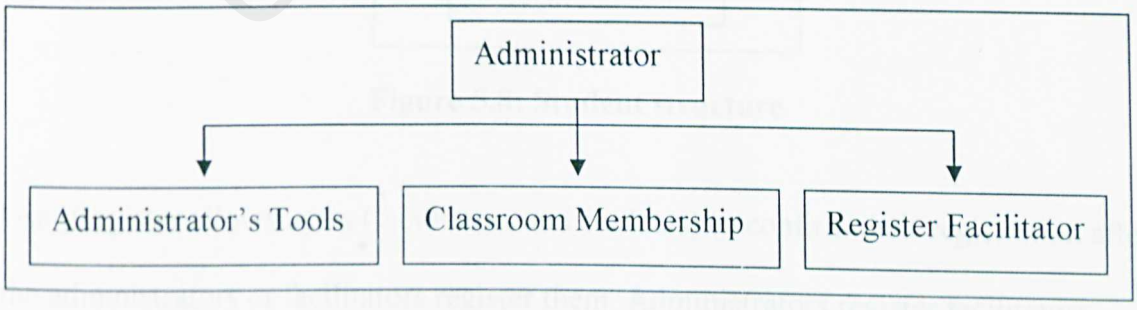


Figure 5.6: Administrator module structure

Facilitator module allows facilitator(s) to register courses and students with the Register Courses sub module. The second module (Facilitator's Tools) enable facilitator to assign question to groups, post messages on classroom notice boards, upload documents or links on the system and terminate a student's membership if found necessary.

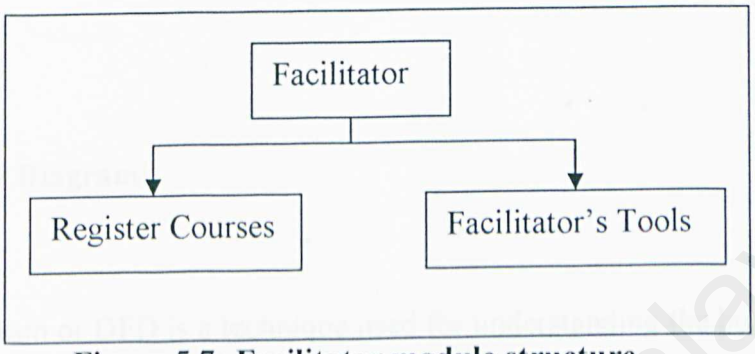


Figure 5.7: Facilitator module structure

There's only one sub module in the Student module. The Student's Tools are allows students to post messages on classroom's main message board and upload documents like question's answers on the system.

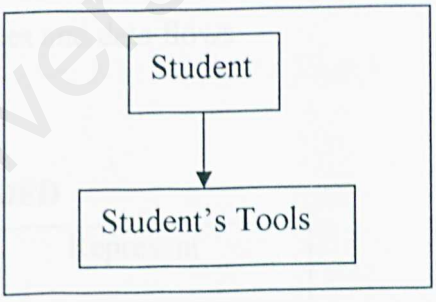


Figure 5.8: Student structure

User Register allows users (facilitators and students) to confirm their registration after the administrators or facilitators register them. Administrators register facilitators while students are by facilitators. To confirm their registration, users need to access the registration function on the system with the password given by the system and then create a username and new password and input necessary details.

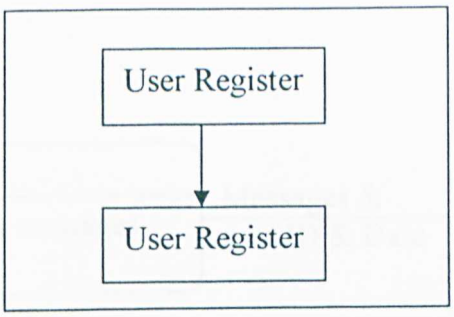


Figure 5.9: User Register structure

5.2.1 Data Flow Diagram

Data Flow Diagram or DFD is a technique used for understanding the business logic and for communication with others (users, programmers and system analysts). It models the logical or conceptual functions needed to perform the various business functions. It is more on what needs to be done, rather than how it will be done.

There are four basic symbols for drawing a DFD. The symbols represent external entities, data stores, processes and data flows.

Table 5.17: Symbols in a DFD

Symbol Name	Represent
Facilitator	External Entity
PBL Facilitator	Data Storage
Register Course	Process
status →	Data Flow

5.2.1.1 Student's DFD

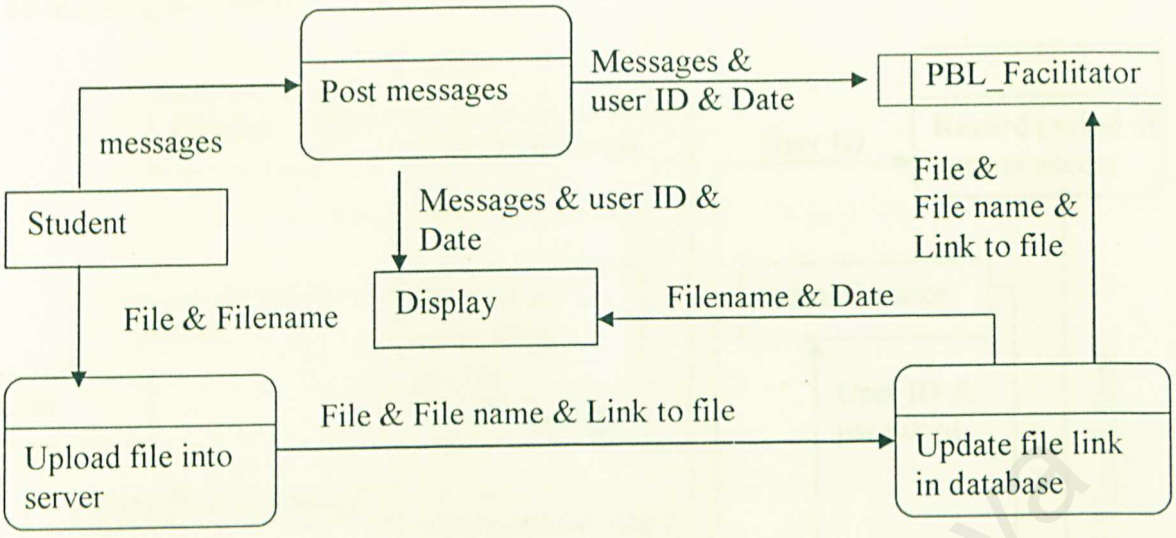


Figure 5.10: Student DFD

5.2.1.2 User Register's DFD

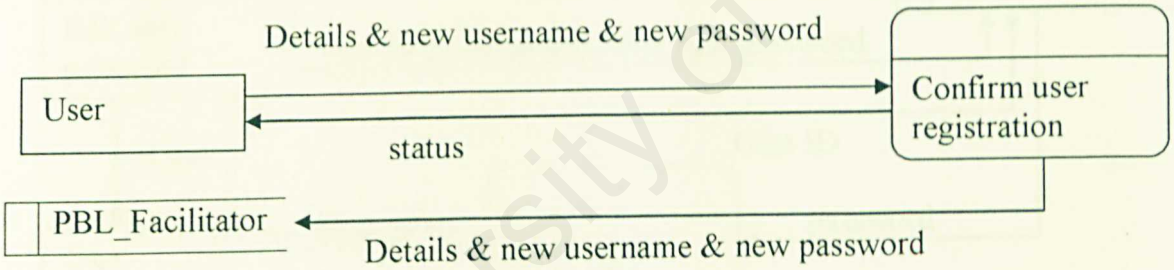


Figure 5.11: User Register DFD

5.2.1.3 Group Management

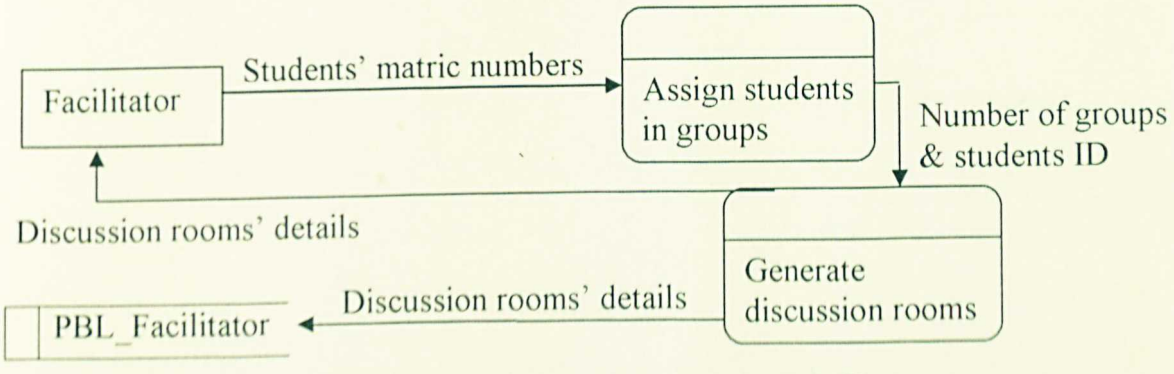


Figure 5.12: Group Management DFD

5.2.1.4 Log In's DFD

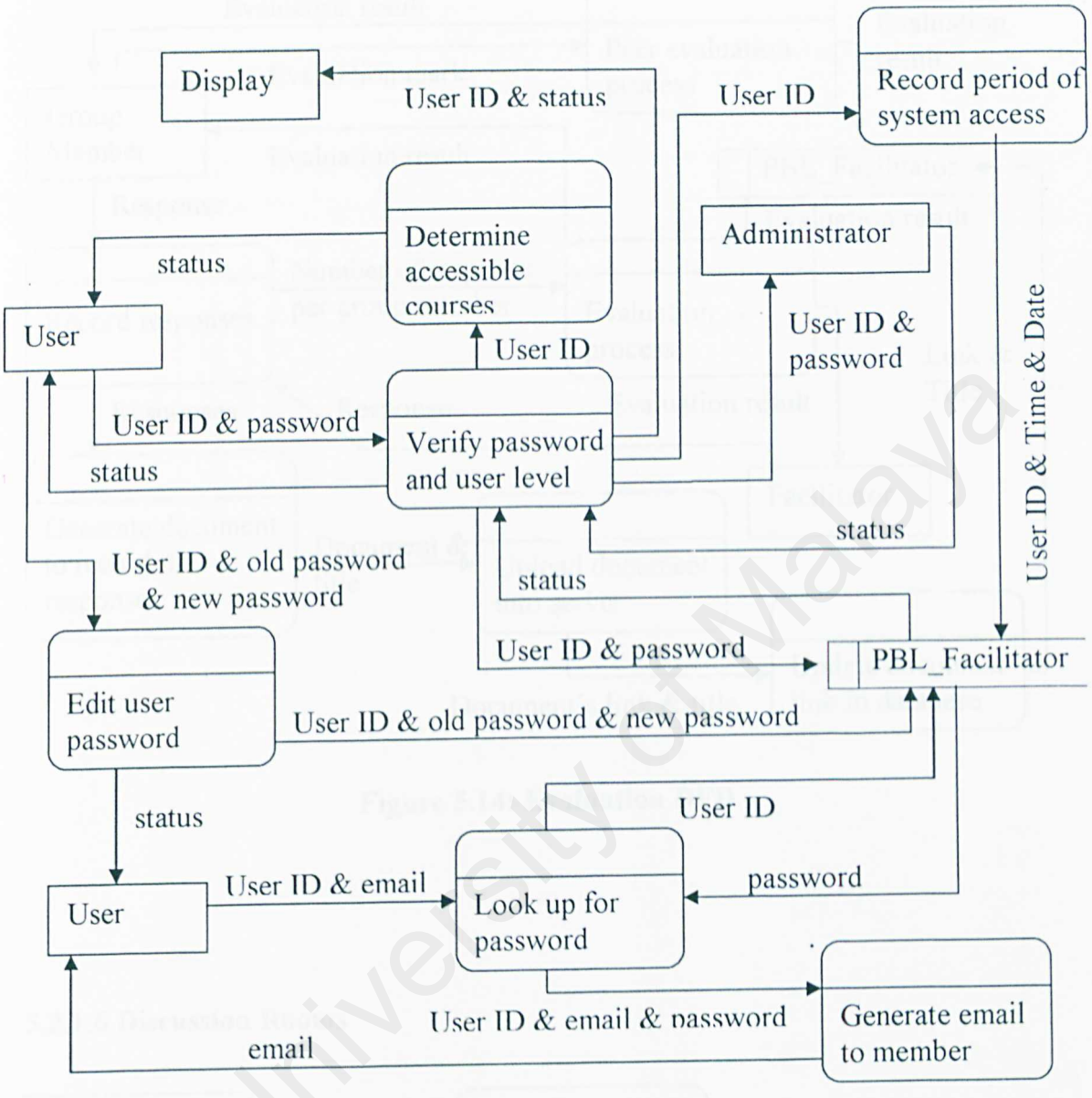


Figure 5.13: Log In DFD

5.2.1.5 Evaluation

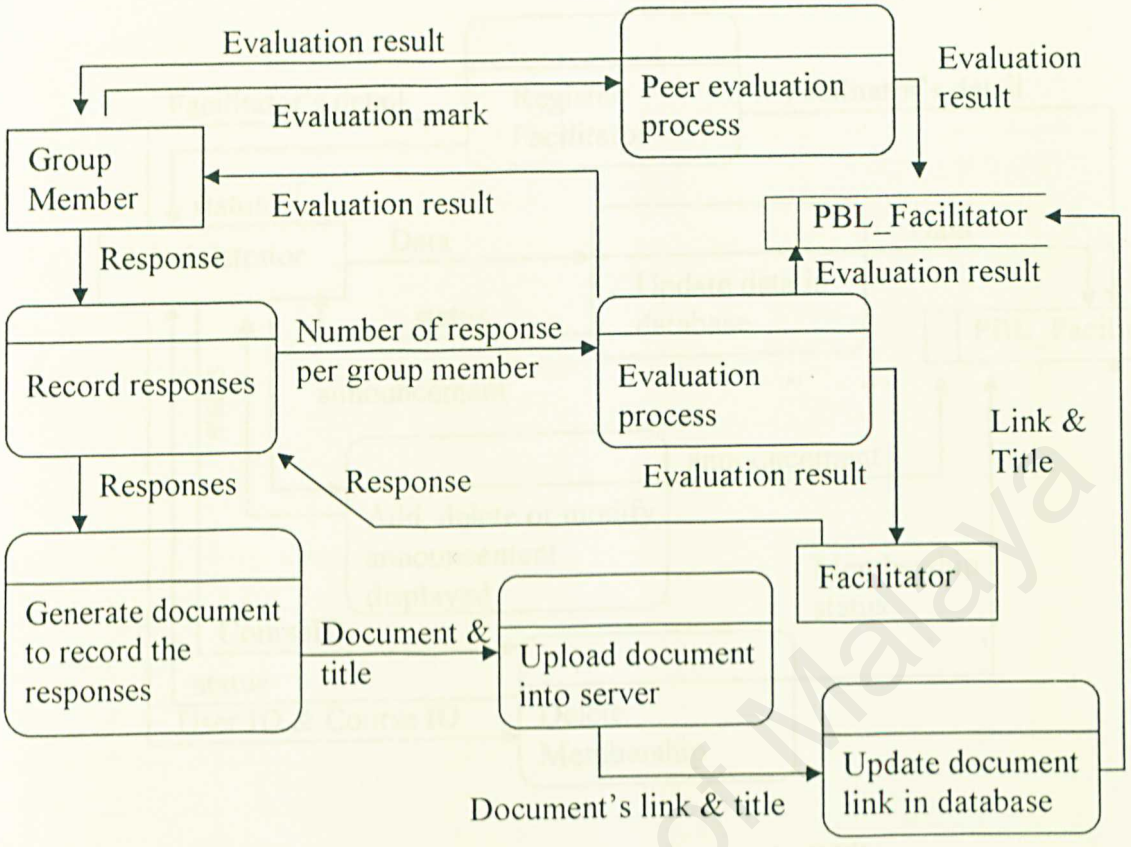


Figure 5.14: Evaluation DFD

5.2.1.6 Discussion Rooms

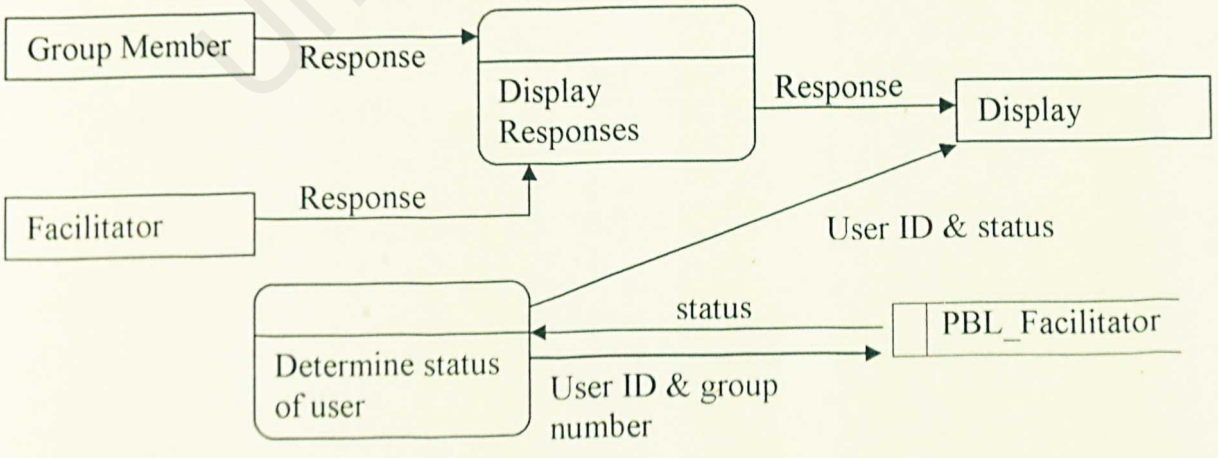


Figure 5.15: Discussion Rooms DFD

5.2.1.7 Administrator's DFD

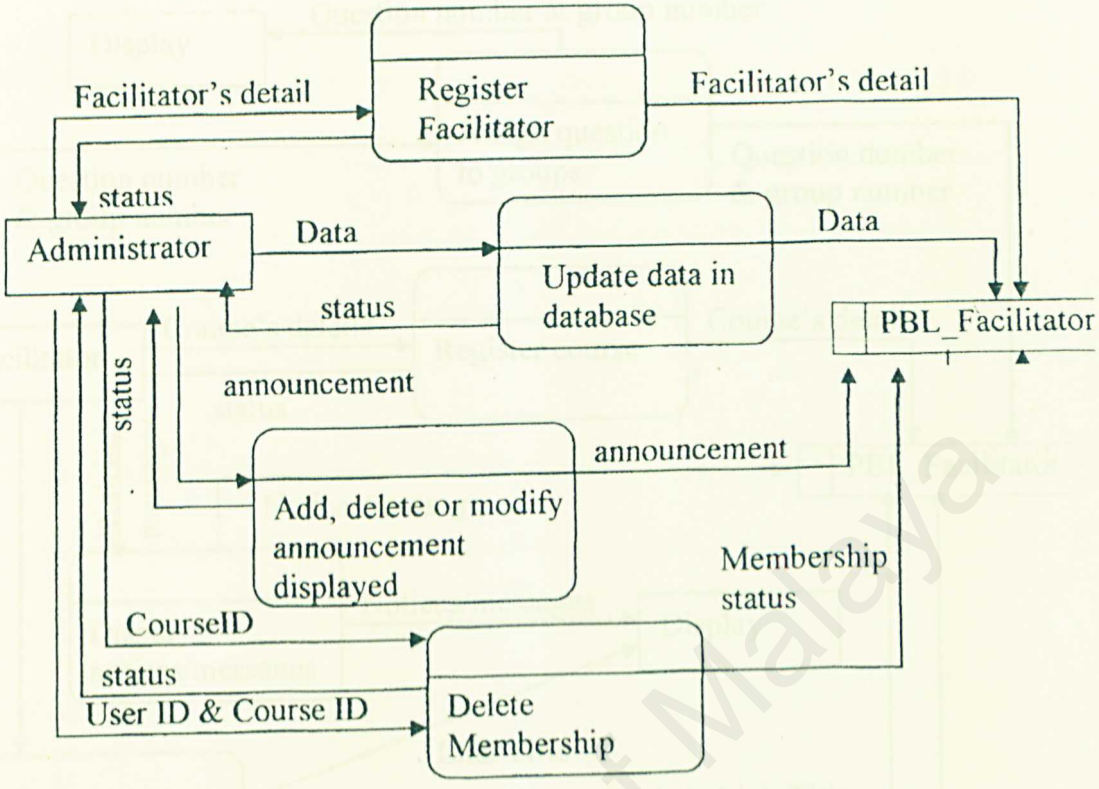


Figure 5.16: Administrator's DFD

5.2.1.8 Facilitator's DFD

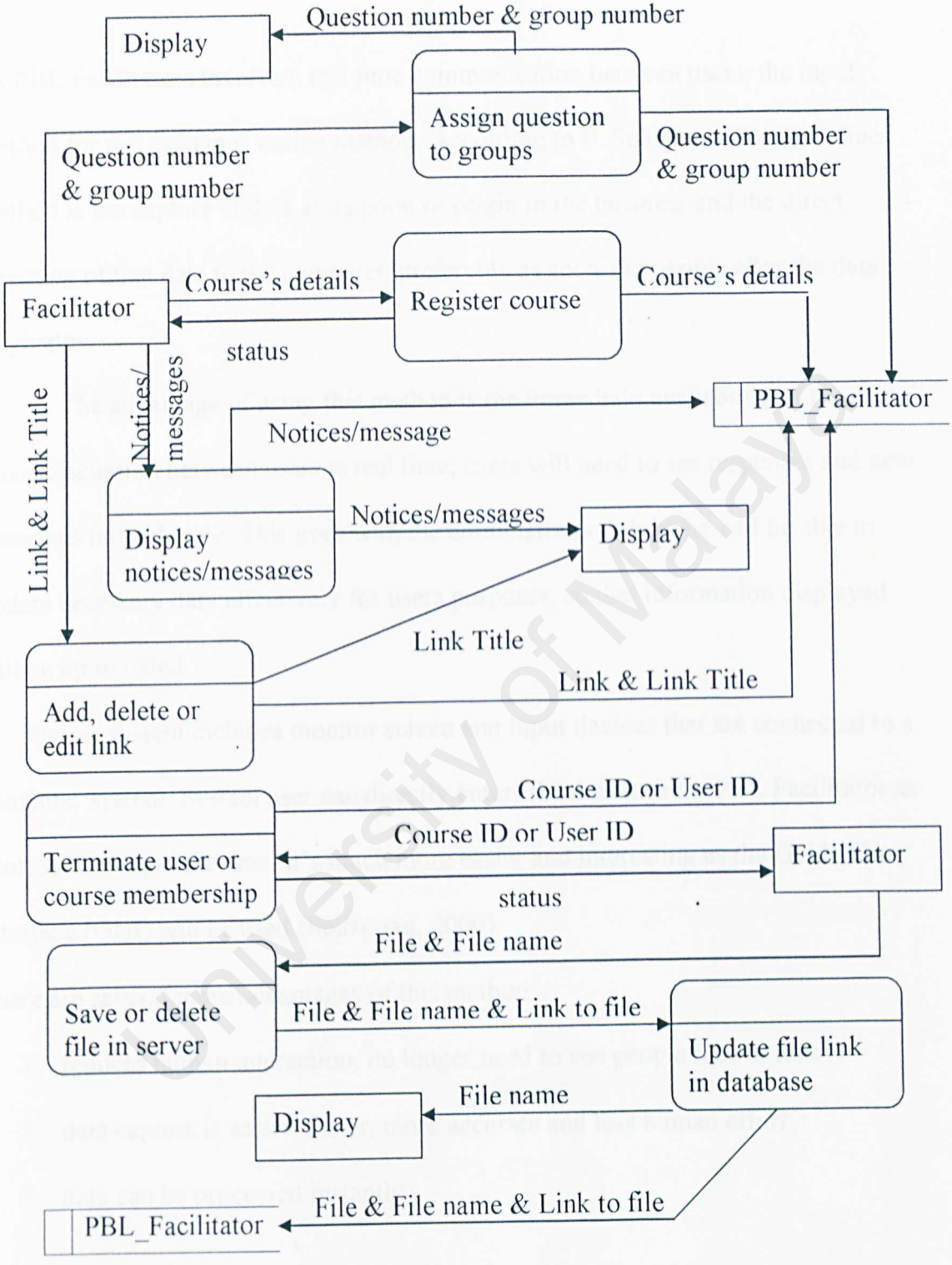


Figure 5.17: Facilitator DFD

5.3 Input/Output Design

As PBL Facilitators involve a real time communication between users, the input method for this system is online method. According to P. Sellappan (2000), online method is the capture of data at its point of origin in the business and the direct inputting of that data to the computer, preferably as soon as possible after the data originates.

The advantage of using this method is the immediate availability of the data. Since discussion between users is real time, users will need to see responses and new messages immediately. This goes with the administrator, where he will be able to update necessary data effectively for users purposes, so that information displayed will be up to dated.

Online system includes monitor screen and input devices that are connected to a computer system. System user can directly enters the data into the PBL Facilitator as soon as user input the data. It's much more easier and interesting as the Guide User Interface (GUI) will be used (Sellappan, 2000).

There are several more advantages of this method:

1. reduces human interaction: no longer need to see people face to face
2. data capture is easier, faster, more accurate and less human effort
3. data can be processed instantly

5.3.1 Input Design GUI

The input GUI must fulfils all these criteria:

1. easy to understand and user friendly.

2. can avoid unnecessary repetitive input
3. minimize human effort (avoid long sentences, etc)
4. input only necessary data
5. if possible, use codes instead of full name for attributes
6. instructions for inputting data must be clear and precise
7. data to be input must be sequenced

Several of the interfaces for data input in PBL Facilitator are depicted in Figure 5.18 and Figure 5.19. The screen-based controls used are:

1. text box
2. radio button
3. check box
4. list box
5. drop-down list
6. combo box

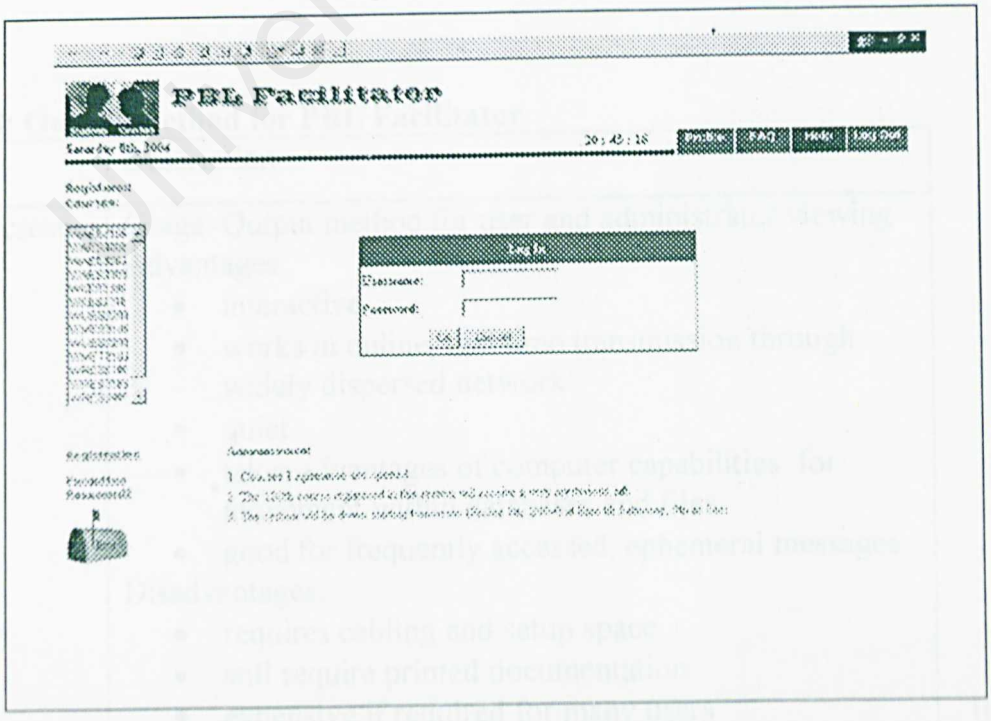


Figure 5.18: User Interface for Log In page

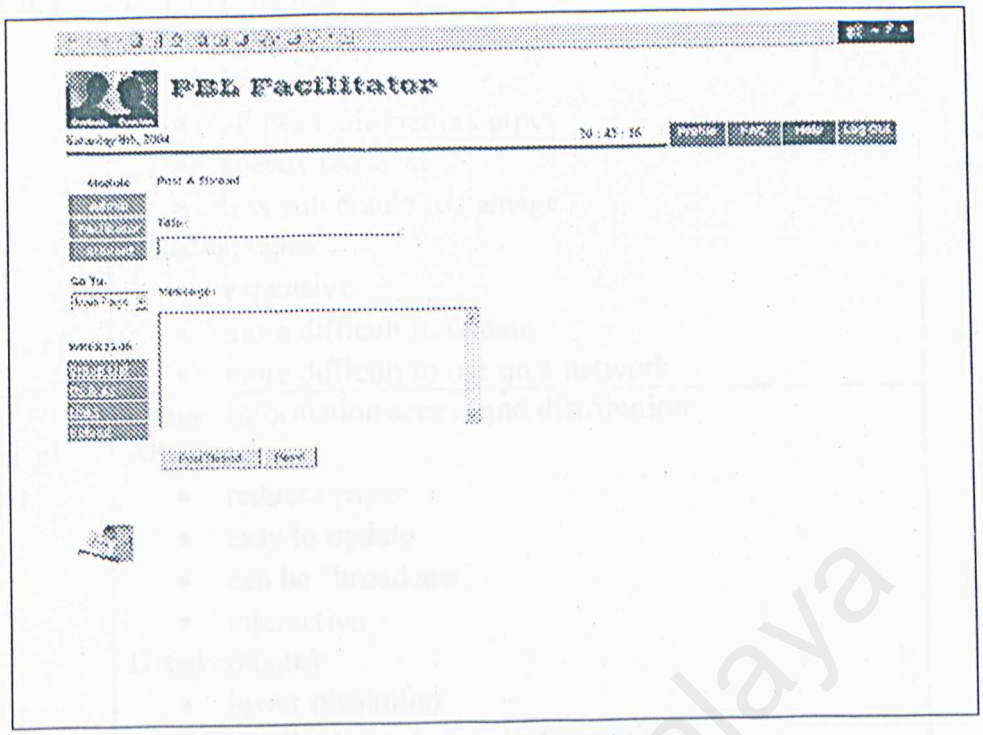


Figure 5.19: Interface for posting messages

5.3.2 Output Design

The output for the system is presented using the display screen, CD-ROM or DVD and Electronic Output (examples: email and web pages).

Table 5.18: Output method for PBL Facilitator

Method	Description
Display Screen	Usage: Output method for user and administrator viewing Advantages: • interactive • works in online, real time transmission through widely dispersed network • quiet • takes advantages of computer capabilities for movement within databases and files • good for frequently accessed, ephemeral messages Disadvantages: • requires cabling and setup space • still require printed documentation • expensive if required for many users

CD-ROM and DVD	Usage: Documentation Advantages: • large capacity • allows multimedia output • speedy retrieval • less vulnerable to damage Disadvantages: • expensive • more difficult to update • more difficult to use on a network
Electronic Output (email, web pages)	Usage: Information access and distribution Advantages: • reduces paper • easy to update • can be "broadcast" • interactive Disadvantages: • lower resolution • email not conducive to formatting • difficult to convey context of messages in email • need diligent maintenance (web site)

Source: P. Sellappan (2000)

Example of GUI for the output design in PBL Facilitator:

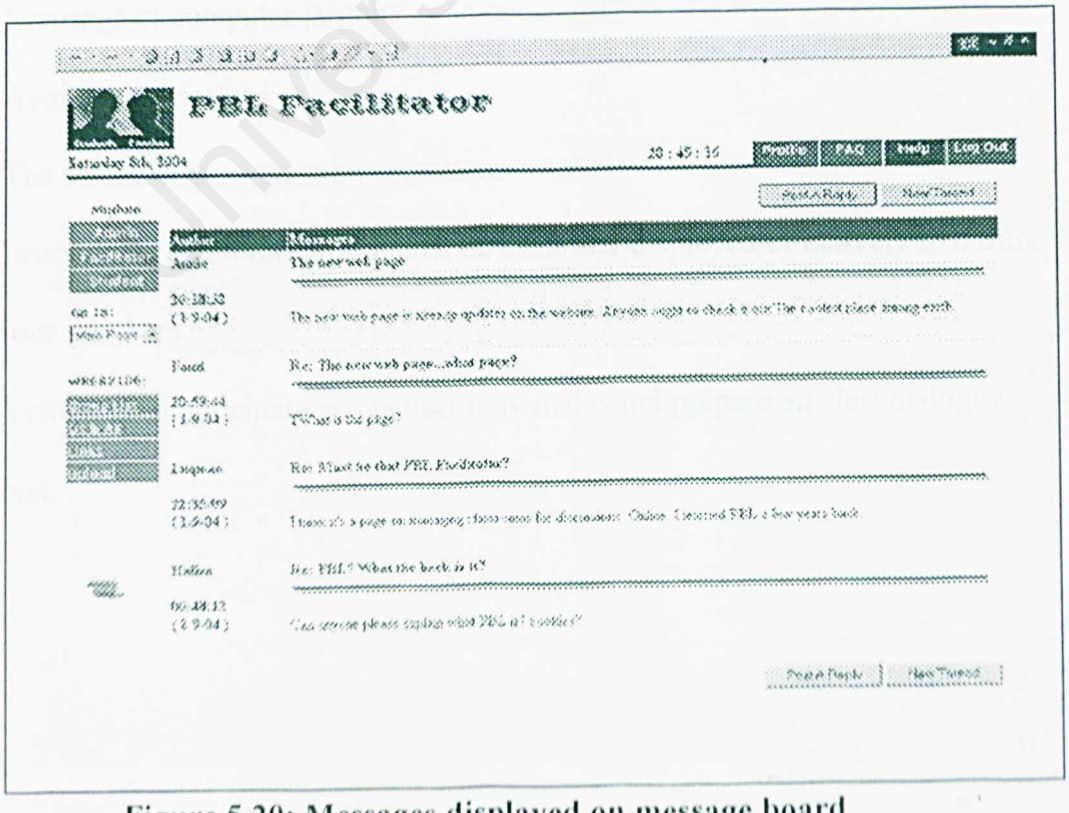


Figure 5.20: Messages displayed on message board

5.4 Interface Design

There are several types of user interface. GUI or Graphical User Interface is chosen for PBL Facilitator as it allows users to communicate easily with the system. GUI uses graphics like menus, boxes and windows which can be directly manipulated using input device like keyboard and mouse (Sellappan, 2000).

According to P. Sellappan (2000), user interface must have these criteria:

1. User friendly and simple
2. System navigation must not be too complex and have too many branches
3. The language must be correct and easy to understand
4. Terms used must be simple and consistent
5. System must not be intimidating or insulting to user (in words, graphics, etcetera)
6. System must be informative so user should know what to do next
7. No usage of computer jargons
8. Avoid abbreviations
9. The screen display must be consistent
10. Instructions and information must be clear and displayed effectively to ensure user see them
11. System must anticipate errors user may make and prepare an alert dialogue box.

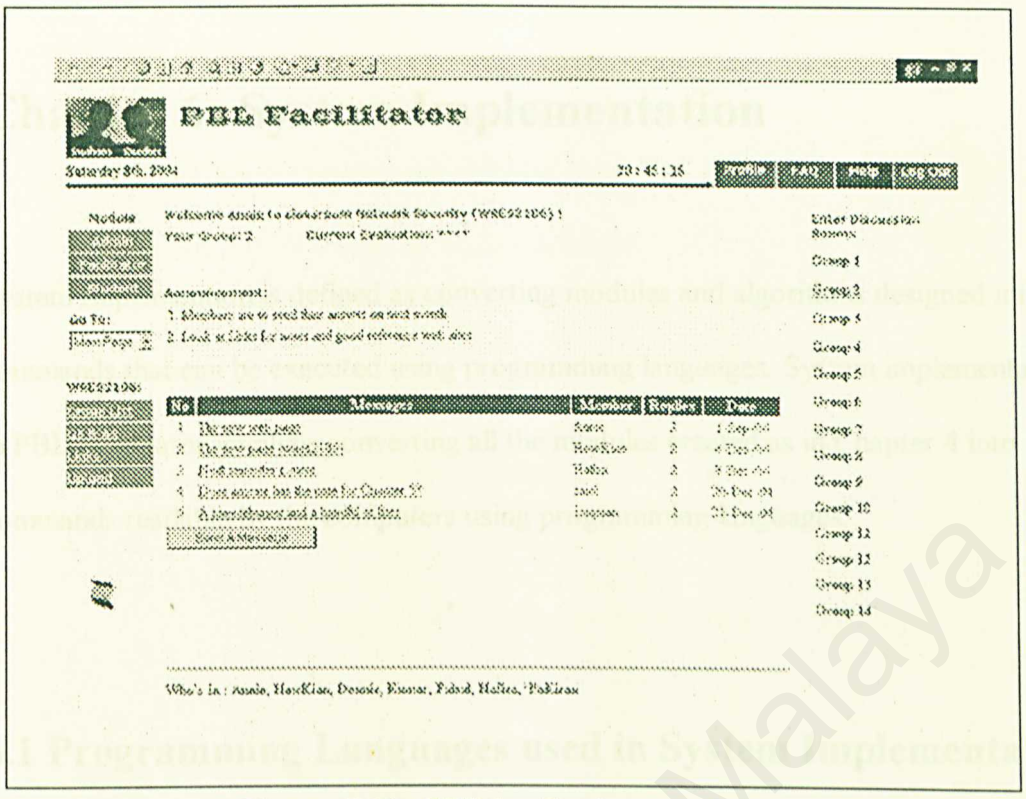


Figure 5.21: GUI for Student module

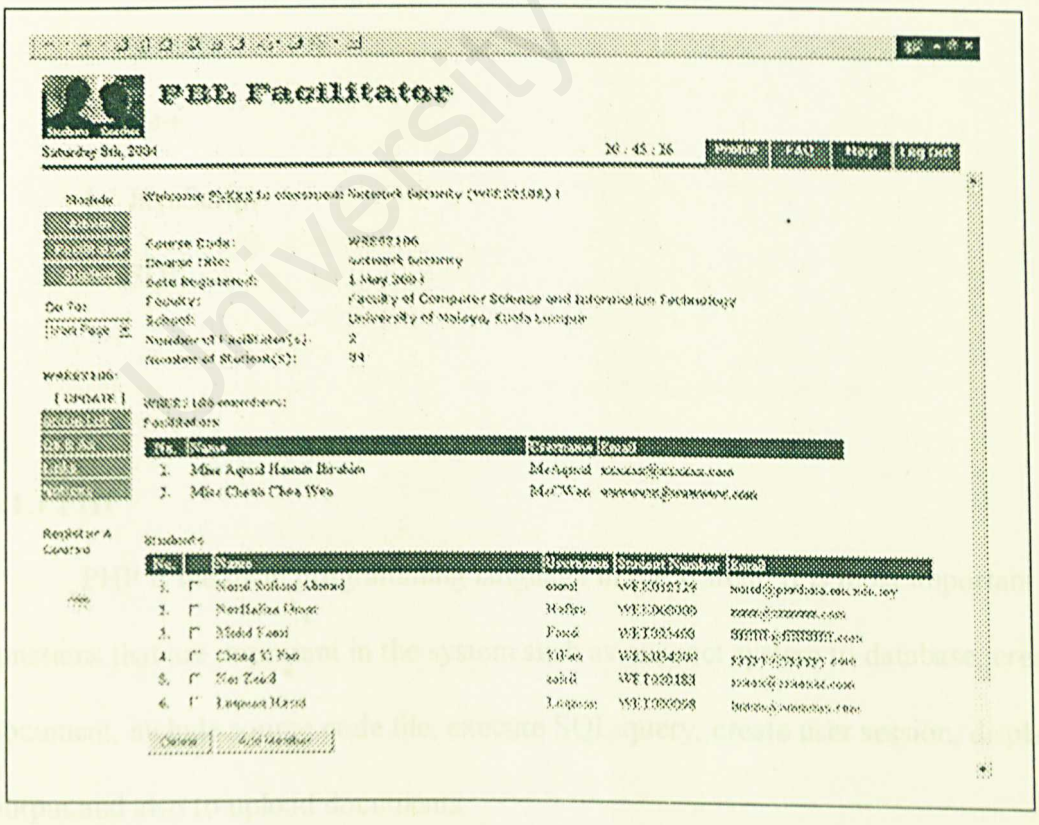


Figure 5.22: GUI for Facilitator module

Chapter 6: System Implementation

System Implemention is defined as converting modules and algorithms designed into commands that can be executed using programming languages. System implementation in PBL Facilitator involves converting all the modules created as in Chapter 4 into commands readable by the computers using programming languages.

6.1 Programming Languages used in System Implementation

The programming languages used in PBL Facilitator are:

1. PHP
2. HTML
3. C++
4. JavaScript
5. SQL

6.1.1 PHP

PHP is the main programming language in the system. It is most important in functions that are important in the system such as connect system to database, create pdf document, include source code file, execute SQL query, create user session, display output and also to upload documents

a. connect system to database

- PHP enables connection towards database. This is important as many important data of the system resides in MySql database.

```
<?php
$dbconn = mysql_connect('localhost', 'root', 'newsystem') or
        die ("<p>Connection to database is not successful !");
mysql_select_db('pblfacilitator',$dbconn) or
        die ("<p>Access to database is not successful !");
?>
```

Figure 6.1 code for connecting system to database

b. create pdf documents

- discussion transcripts are required to be converted into pdf documents. This is to help in lessing the size of database and also provides better view of the transcripts.

```
<?php
define('FPDF_FONTPATH','font/');
require('../pdflibrary/fpdf.php'); //opening fpdf function
$CourseID = $_POST['CourseID']; // retrieve data from form
$pdf=new FPDF();
$pdf->AddPage();                    // create new pdf document
//Set font
$pdf->SetTextColor(0,0,160);
$pdf->SetFont('Times','B',10);
//Move to 8 cm to the right
$pdf->Cell(100,6,"Course : ".$CourseID);        // display CourseID
$pdf->Output();                     // display the new document

// the whole coding has been edited for example purpose        ?>
```

Figure 6.2 Code for converting data in database into a .pdf document

c. include source code files

- sometimes there are too many recurring data in a web page. So, to lessen the amount of lines in a page, the data can be saved as an include files whether as a .php, .html, .inc or many other types of web page. This lessen the programmer`s task to write/copy the same code numerous times whether in a same page or in many different pages.It is also helpful in hiding important data from users` view. For example, dbinfo.php has the code for connecting system to database. It has the username and password to access the database and also dbinfo.php is used in almost all the pages in the system. All files that are included in other files are stored in a folder named `incfiles`.

```
<?php
include("../incfiles/dbinfo.php");   // open connection to
                                     //  database
?>
```

Figure 6.3 the code for including dbinfo.php

d. execute SQL queries

- SQL queries in PBL Facilitator involves data input from user. So, php provides great help in collaborating the datas and the queries in order to make them work.

```
<?php
include("../incfiles/dbinfo.php");
$FileNo = $_POST['FileNo'];                    // retrieve data from form
$CourseID = $_POST['CourseID'];
if (isset($_POST['del'])) {                    // action to delete chosen data
  $table = $CourseID."_Links";
  $query = "delete from $table where No='$FileNo'";   // query database to
  $result = mysql_query($query);                      // delete data based on
  $query1 = "select No from $table order by No";      // retrieved data
  $result1 = mysql_query($query1);
  $number1 = mysql_num_rows($result1);
  if($number1!=0) {                            // resorting data in ascending order
     $j=1;                                     // after deletion complete
     for($i=0; $i<$number1; $i++) {
            $No[$i] = mysql_result($result1,$i,'No');
            $query2 = "update $table set No='$j' where No='$No[$i]'";
            $result2 = mysql_query($query2);
            $j++;
            }
        }
     }
?>
```

Figure 6.4 code for deleting data in database

e. create user session

- creating user session is important for system`s integrity and security. A session for a user will be created each time a user logs into the system. The system will also be destroyed whenever the user logs out the system.

```
<?php
session_start();
$userID=$_POST['userID'];
unsetLogStatus($userID);
session_unset($userID);
session_unset($userPw);
session_unregister("userID");                  // delete session
session_unregister("userPw");
```

```
if(!isset($userID) ){                    // check whether a user is logged in or
not
  logIn($k);
  exit;
  }

else {
  session_register("userID","userPw");    // register session
  $userdata = checkuserID($userID,$userPw);  // check whether
  $username = $userdata['UserID'];              // username and password
  $userstatus = $userdata['Status'];           // is valid
  if(!$username) {
    session_unregister("userID");
    session_unregister("userPw");
    $k=2;
     logIn($k);
     exit;
     }
   else {
         mainPage($userstatus,$username);        }
  }
?>
```

Figure 6.4 code for creating user session

f. display output

- the most common usage of PHP is to display output, especially the value of a defined variable.

```
<?php
for($i=0; $i<$num3; $i++) {                    // retrieving data from database
  $AdmName[$i] = mysql_result($res3,$i,'AdmName');
  $AdmID[$i] = mysql_result($res3,$i,'AdmID');
   // table coding
   .
   .
  echo "<td>".$AdmID[$i]."</td>";
  echo "<td>".$AdmName[$i]."</td>";
  echo "</tr>";
  .
  .     // end table coding
  }
?>
```

Figure 6.6 An example of outputting data retrieved from database. The code has been edited (shortened) for this example.

g. upload documents

- users are allowed to upload documents for work purposes. Students in PBL Facilitator can upload their solutions for situations given by the facilitator (also uploaded) in the web site.

```
<?php

$realname = "Group".$GroupNo."_".$HTTP_POST_FILES['userfile']['name'];
$path="../Assgnmt_GroupSol/".$realname;
include("../incfiles/dbinfo.php");

if (is_uploaded_file($HTTP_POST_FILES['userfile']['tmp_name'])) {
  copy($HTTP_POST_FILES['userfile']['tmp_name'], C:/PBL_FileLink/$realname");    }
else {
  echo "Possible file upload attack. Filename: ".HTTP_POST_FILES['userfile']['name']; }
move_uploaded_file($HTTP_POST_FILES['userfile']['tmp_name'],
"../Assgnmt_GroupSol/$realname");

.

.
// further action: save document path and name into database
?>
```

Figure 6.7 An example of uploading and copying file into specified folder. The code has been edited (shortened) for example purpose.

6.1.2 HTML

Aside from displaying output, in PBL Facilitator, HTML is mainly used for interface design and interaction medium between user and system, such as forms.

a. Interface

- all of the pages in PBL Facilitator are constructed using 3 main tables: top table, left table and main table(middle). All three tables actually reside in a bigger table. Below is an edited example to show the basic layout of the pages.

```
<html>
<head>
<title>FAQs</title>
<?php include("../incfiles/DatenTime.php"); ?>  <!--output the dynamic time and date-->
</head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body background="../images/arctile.jpg">
<table width="980" height=100% border="0" cellspacing="0">  <!-- the main table -->
 <tr>
  <td colspan=2 width=980 height=119 valign=top>
       <?php include("../incfiles/Header.php"); ?></td>        <!-- the top table -->
 </tr>
 <tr>
  <!--below is the second (left) table. It is mainly for navigation bars -->
   <td width=108 valign=top><?php include("../incfiles/ClassroomLeft.php"); ?></td>
   <td width=872 valign=top>

            .
            .          <!--this is where the middle table resides, it doesn't necessarily -->
            .          <!--have to be a table, however it's much more easier to maintain -->
                       <!—if every main part is designed as tables                        -->
   </td>
 </tr>
</table>
</body>
</html>
```

Figure 6.8 basic pages layout

b. forms

- the tool used for this system is Macromedia Dreamweaver MX. When creating designs or forms, the coding is automatically generated. The forms in the system are programmed either using PHP or HTML. However, it's better if form designs stay in HTML coding, as HTML coding can be viewed, where as PHP can't, and thus making maintenance work much easier.

```
<?php // table coding
echo "<form action=ContactAdmin.php method=post>"; ?>
 <td colspan="2">
      <input type="text" name="recipient" size="60" maxlength="50"></td>
 </tr>
 <tr>
  <td width="99" height="50"> Subject: </td>
  <td colspan="2"><input type="text" name="subject" size="60" maxlength="50">
  </td>
 </tr>
 <tr>
  <td width="99" height="30" valign="top"> Message:</td>
  <td colspan="2"><textarea name="message" cols="60" rows="8" wrap="soft">
   </textarea></td>
 </tr>
 <tr height="30">
  <td height="40" align="right"><input type="reset" value=" Reset "> </td>
  <td width="114" align="center"><input type="submit" name="send" value="Send">
  </td>
 </tr>
<?php echo "</form>"; ?>
```

Figure 6.9 an edited example of form design

6.1.3 C++

PHP enables C++ to be embedded in web pages. C++ is useful for looping purposes: while, if-else and also for.

```
<?php
$table = $CourseID."_complaint";
$que1 = "select * from $table";
$res1 = mysql_query($que1);
$num1 = mysql_num_rows($res1);

for($i=0; $i<$num1; $i++) {
  $Complaint[$i] = mysql_result($res1,$i,'Complaint');
  $ComplaintTitle[$i] = mysql_result($res1,$i,'ComplaintTitle');
  $No[$i] = mysql_result($res1,$i,'No');
  $DatePosted[$i] = mysql_result($res1,$i,'DatePosted');
  $TimePosted[$i] = mysql_result($res1,$i,'TimePosted');
  $StID[$i] = mysql_result($res1,$i,'StID');
  }
// above is an example from Complaint.php . It shows that all the rows in the
//  selected table is retrieved. And this requires each data in every column in
// every row to be treated as array. And this can be achieved using loop for.
?>
```

Figure 6.10 an example for C++ usage in PBL Facilitator

6.1.4 JavaScript

JavaScript is mainly used to give feedback to users` action, especially action concerning forms. For example, a facilitator may try to start a private discussion rooms by clicking a button, and the system will generate an alert to give warning to the facilitator. Another example is when a new user fills up a form to register, after the form is filled and sent, the system will notify the user that the system has received the data from the form.

It is considered necessary as it`ll give users clue of the rights and wrongs of their action.

Apart from generating alerts, JavaScript is also used in the system to display dynamic date and time (DatenTime.php) and to display drop-down menu.

```
<?php

if(isset($_POST['del'])) {
  if($userStatus=='A') {  ?>
        <script language="JavaScript">
        <!--
        alert("For facilitators' purpose only!");
        -->
        </script>          <?php
         }
   else {
        .
        .  // SQL queries for deleting data from a table
       }
}
/* this code is extracted from Complaint.php. It stated the flow of action if
 * an administrator tries to delete a data from a page in  facilitator's
 * module.
 * if proven the user is administrator, an alert will be displayed   */
?>
```

Figure 6.11 an example of JavaScript usage

6.1.5 SQL

MySql uses SQL queries for querying data. SQL queries can do the following tasks :

a. select data from tables

b. delete data in tables

c. update tables (data/design)

d. insert data into tables

e. create/drop tables or databases

a. select data from tables

- whenever data from databases are to be viewed.

```
<?php
include("../incfiles/dbinfo.php");
$que5 = "select CourseID from course order by CourseID asc";
$res5 = mysql_query($que5);
$num5 = mysql_num_rows($res5);

for($i=0; $i<$num5; $i++) {
        $CourseList[$i] = mysql_result($res5,$i,'CourseID'); }

mysql_close($dbconn);

// the query is to select all code of courses registered in the system
// and display them on the left table in Administrator.php
?>
```

Figure 6.12 selecting all code of courses for displaying purpose

b. delete data in tables

```
<?php
$MsgNo = $_POST['MsgNo'];
include("../incfiles/dbinfo.php");
$grtable = $CourseID."_Messageboard";

if (isset($_POST['del'])) {
 $query = "delete from $grtable where CorrspndTo='$MsgNo' or No='$MsgNo'";
 $result = mysql_query($query);
 }
// the row of data that has a field No value of '$MsgNo'
// is deleted from the selected table
mysql_close($dbconn);
?>
```

Figure 6.13 an edited example of deleting data in a table

c. update data in tables

```
<?php
$noticeloc = $_POST['noticeloc'];
$newnotice = $_POST['newnotice'];
$FileNo = $_POST['FileNo'];

include("../incfiles/dbinfo.php");

 if (isset($_POST['edit'])) {
   $table = $CourseID."_Noticeboard";
   $query = "update $table set Message='$newnotice', FID='$userID',
             DatePosted=curdate(), TimePosted=curtime(),
             Location='$noticeloc' where No='$FileNo'";
   $result = mysql_query($query);
   }
mysql_close($dbconn);

// the row of data that has a field No value of `$FileNo` is updated with
// the retrieved from the form
?>
```

Figure 6.14 an edited example of updating data in a table

d. insert data into tables

```
<?php
$noticeloc = $_POST['noticeloc'];
$newnotice = $_POST['newnotice'];
$FileNo = $_POST['FileNo'];

include("../incfiles/dbinfo.php");
if($add) {
 $table = $CourseID."_Noticeboard";
 $query = "select count(No) from $table order by No";  // count number of row in
 $result = mysql_query($query);                        // table
 $number = mysql_result($result,0);
 $number++;                                  // the number of new data to be inserted
 $query1 = "insert into $table set No='$number', Message='$newnotice',
            FID='$userID', DatePosted=curdate(), TimePosted=curtime(),
            Location='$noticeloc'";
  $result1 = mysql_query($query1);
 }
```

```
mysql_close($dbconn);

// a new row of data is inserted into the table
?>
```

Figure 6.14 an edited example of inserting data in a table

e. create/drop tables or databases

```
<?php
$courseCode = $_POST['courseCode'];
include("../incfiles/dbinfo.php");
$table = $courseCode."_disctranscript";
$query4 = "create table $table (
                    GroupNo int(2),
                    DatePosted date,
                    TimePosted time,
                    DiscNo int(3),
                    Pdf char(1),
                    Link varchar(75))";
$result4 = mysql_query($query4);
mysql_close($dbconn);

// a new table is created
```

Figure 6.15 an edited example of creating new table

Chapter 7: System Testing

7.1 System Testing

After the coding phase is complete, the system (PBL Facilitator) will undergo the testing phase in order to identify whether it fulfills the requirements and specifications as decided in Chapter 4. During this phase errors and bugs are found and then fixed. There are four types of testing executed during these phase :

1. unit testing
2. integration testing
3. system testing
4. acceptance

7.1.1 Unit testing

A unit is the smallest component in a system. In PBL Facilitator, the smallest unit is the sub-submodule. During unit testing, each individual unit is tested separately. All functions in a unit were examined carefully to check for errors. This was to ensure that all corresponding functions can interact normally inside the unit.

After that, the source code was compiled and then runned on the browser. The browser then notified any errors or warnings and based on the notification, the source then reviewed and corrections were made.

- it is important however to remember that in most circumstances, the browser won't display any notification. But with using PHP, user can request the browser to display the notification by changing the 'display_error=On' in the php setting in php.ini

Test Input	System Result
Invalid password or Invalid username	Output: system prompts error message. Result: system responded correctly
Valid password and username	Output: user is transferred into the next page. If the user is an administrator, he can access all the modules: administrator, facilitator, and classroom. If a facilitator, he can access both the facilitator and classroom module, but only in the courses that he registered into. If a student, he can only access the classroom module in courses that he registered into. Result: system responded correctly.

Table 7.1 Example of unit testing and how system corresponds to the test

Table 7.1 above is an example of unit testing. When a user logs into the system, system will access the database and check the validity of both the username and password. If one or both of them prove to be wrong, the system will prompt a message stating the error. But else, the user will be brought to the next page, and system will determine their level of access from the database.

7.1.2 Integration testing

In integration testing, more than one module was integrated and tested to see whether they work properly. The system was tested using the bottom-up testing, where the testing starts from the lowest sub modules to the top sub modules.

Test Input	System Result
Correct combination of generated username and password and user's particulars	Output: the system checked whether the user already confirmed his registration. If yes, then an error message was prompted. If not, then the user was brought to the next registration page to finish the confirmation task. Result: system responded correctly
All submit buttons in Facilitator module were clicked.	Output: as both administrators and facilitators can access to the Facilitator module, system will check whether the user's status was administrator or facilitator. If the status was administrator, then error message confirming denial access to task was prompted. But if the result was Facilitator, then the system will proceed to the next steps prior to clicking the button. Result: system responded correctly.

Table 7.2 Example of integration testing and how system corresponds to the test

Table 7.2 above provides examples of integration testing. The first example is what occurs in User Register module when users (both student and facilitators) need to confirm their registration. The module then checked with Log In module, in order to identify whether that particular user has already registered. If the user has already registered, then he would be prompted with an error message. Else, he would be taken to the next page where he can finish confirming his registration.

7.1.3 System testing

All the main sub modules or sub systems were combined and integrated to function as a complete system. The system then was run on the browser and relationships

between the modules and sub-modules ware recognized as successful or not. This was determined by recognizing errors that occurs while running the system.

Test Input	System Result
All navigational buttons are clicked	Output: the page goes to the intentioned page depending on the button clicked by the user. Result: system responded correctly
Submit incomplete forms	Output: the system prompts error message and request user to fill in the necessary fields. Result: system responded correctly.

Table 7.3 Example of system testing and how system corresponds to the test

In the first example in Table 7.3 above, users clicked all the navigational buttons and determine whether the page turned into the correct page. The result proved successful when the browser went to the correct page. And in the second example, users filled in the forms and left out several fields, including the necessary ones. The test proved successful when the system prompted error message and user were brought back to the form to be finished.

7.1.4 Acceptance testing

The acceptance testing involved real data and end user. During this testing, several colleagues were involved in trying out the system, but the most important end user happened to be the moderator of this project, Prof. Salimah Mokhtar, whom was also

involved in approving the requirements of this system. Real data such as the name of courses and actual students` names in University Malaya were used.

The result of the testing proved quite satisfactorily, as many of the requirements were proven met and the implementation were successful. However, certain requirements were lacking, such as the `peer evaluation`. As such, the system was reviewed and additional function had to be made, and the implementation and testing began all over again.

7.2 System Maintenance

After the system has been fully developed, the next important step is maintaining the system. System maintenance keeps the system in good condition and up-to-date. Maintaining system involves updating system based on current environment and changes in users` requirements.

7.3 Documentation

As written in the sub chapter 7.2, maintaining system is a very important task. As it is, it`s very important that the programmer made the future maintaining task easy. At the very least, provide the system documentation.

There are two kinds of documentation prepared for this system; source code documentation and user manual.

1. user manual

- user manual is created so that administrator can understand how to use the system. For PBL Facilitator, user manuals are created all three types of users. The manuals however, are only supplied for administrators' purpose. Facilitators can request for the manuals from the administrator if they need the manual. Studentsdo not need to have manual as the facilitators will be able to brief for them.

2. source code documentation

- source code of documentation are comments in source code files that explains what is the function and purpose of the codes. This type of documentation makes it easier for future programmer to understand the codes and maintain them.

Chapter 8: System Evaluation

The system's evaluation task involves analysing problems encountered while developing the system and solutions taken to overcome them. Limitations and strengths of the system are also gathered, where they can later be analyzed to define possible future enhancements to make the system more efficient suitable for tomorrows' changing reqirements.

8.1 Problems and solutions

There are many factors that determine whether a system proves to be a success or not. Numerous problems were encountered while developing this system. The problems had to be solved in order to make the system works. If they couldn't be, compromisation was necessary and other alternatives had to be used. The main problems encountered and the solutions are as listed below:

1. lack of experience and knowledge in system development

- this problem is very alarming as it costs the time spent in learning and understanding the processes in developing the system. As the time frame of developing the system is not more than four months or so, this proved to be quite a heldback in the system's progress. This problem also caused system builder to focus on the wrong part of the system development, and sometimes result in negletion of important requirements.

Solution: the solution to this problem is to take quality time to analyse the past projects by the senior students. This, even though didn't actually solve the problem, it

minimized the effect and provided good guide on what's important, what to be focused and foremost, and also tips for better developemnt process.

2. time-consuming database

- the MySql database used in this system doesn't have its' own interface like Microsoft Access or any other database. Instead user have to enter the dos window and type the queries to use the database. The task in using the database this way, is very tedious, especially since PBL Facilitator involves a lot of database queries.

Solution: the solution is very simple, download tools from the Internet that provide user interface for the MySql database. Most of the software on the Net (probably all) allows internet user to download the software for trial for a few days (without all privileges), mostly for a month, and then user must purchase the license before he can continue using the tool. However, a tool called MyISql (can be downloaded at www.download.com) allows trial for unlimited period of days. The drawback however, a user that uses unlicesed MyISql will be prompted every few minutes with a reminder to purchase the license, a very annoying interruptions. The tool however, is very easy to use compared to others researched, and it eventually is used for the PBLFacilitator database interface.

8.2 System strengths

The strengths of the system are assessed by checking on the users` non-functional requirements, whether they are successfully met or not. And below are listed all the strength of the PBL Facilitator.

1. user-friendly

The system is easy to understand and easy to use, with no necessary trainings for the users. It have informal environment that isn't intimidating. It can accommodate all three kinds of user (administrator, facilitator, student) and differentiate their privileges without difficulties.

2. security

The system is secured with user identification and authentication to avoid unauthorized access.

3. error catching

The system can detect errors that occurs (run-time) and alert user of the error.

4. expendable

The system can be upgraded and modified in future development to a ccommodate possible necessary changes.

5. logical interface flow

The system`s navigational flow is very easy to understand and it doesn`t have deep and many branches. Users have no problems differentiating the modules he`s in and how to move from modules to modules.

8.3 System Limitations

There are several limitations of the system. The limitations are:

1. As there`s actually no need for more than one administrator, there`s no function on the system that enables administrator to add new administrator. Administrator will have to add a new one by going directly into the database and ad the necessary data in the table Administrator and User_Password.

2. The system is not intelligent enough. After a session of private discussion, the system will then evaluate each student`s performance based on the number of responses by each student. The limitation here, is that the system can not differentiate whether each response is relevant to the discussionor not. And thus, all the responses will be taken into account, even if the response is `Hi`.

8.5 Future Enhancement

The system is open to more upgrading and enhancing to make it run more smoothly and effectively in its purpose. The future enhancements are:

1. The system will be intelligent enough to be able to differentiate between relevant responses and meaningless responses.
2. The system will be able to allow administrator to add more administrator directly using the system, without having to access the database.
3. The whole process of registering student will be made faster. Possibly without facilitator`s involvement.

8.6 Knowledge Gained from Experience

The experience of buiding a complete, functional useful web-based system really provided the builder with priceless knowledge useful for one's self especially for working life later.

Several of the knowledge gained from the whole experience are:

1. Going into a project with relevant knowledge prepared is very important to avoid being idle and wasting precious time learning and practicing at the same time.
2. Always there are others who know more about programming better than one`s self. Don`t hesitate to ask.
3. Be confortable with one`s boss (in this case, the supervisor). Their guidance and advice is very important. Do not ever cross them.
4. Always do your back-up.
5. Last minute project is hazardous.

User Manual

Contents

Contents Page

Lists of Figures

Figure **Page**

A. About the User Manual

The purpose of PBL Facilitator's user manual is to provide easy means to understand how to use system. The manual explains the functions of every page in the system, and also included clips of the user interface.

Overall, the manual is divided into 5 main parts; the installation guide, and the navigational flow in the system starting from ; system log in, system main page, the administrator module, the facilitator module and the classroom module.

The installation guide will guide the administrator on how to properly install PHP and MySql to enable the system to run properly.

The manual shows the usage of the system based on the navigational flow. As the system's user interface is divided into 3 main interface;

1. solely for administrator's purpose (Administrator module)
2. for facilitators' purpose (Facilitator module) - administrator can access but can't use the functions.
3. for students' and facilitators' purpose (Classroom module) - administrator can access but can't use the functions.

it is easier if user understands what the user can or can't do based on the user's status.

B. PBL Classroom Installation

Step 1:

Below are the requirements (suggested) necessary to run the system:

1. Intel Pentium 4 Processor (2.8 GHz)
2. 256MB SDRAM
3. 40 GB hard disk
4. Windows XP
5. IIS
6. Input devices: mouse and keyboards
7. Driver: CD-ROM and floppy disc
8. Development tools: Macromedia Dreamweaver MX
9. Browser: IE 4.0 or above

Step 2:

Administrator need to install PHP and MySql database into the intended server. Both can be downloaded respectively at http://www.php.net and http://www.mysql.com for free.

After installing PHP, open the php.ini file and set display_error to 'Off':

display_error='Off'.

Next, in the Mail function, set the smtp_server and smtp_send so that the server enable the system to relay emails automatically via the server. Consult the server administrator for this purpose. Example:

smtp_server=202.185.11.22

smtp_send=myexample@mydomain.com

for further information, refer to the installation manual provided while downloading php from the website.

** if MySql already installed, ask the database administrator for the password and username to access MySql**

Step 3:

i. Copy the PBLClassroom folder from the installation disc into wwwroot folder.

ii.Open file 'dbinfo.php' in folder 'PBLClassroom/incfiles/'.

iii.Look at the third line : *$dbconn = mysql_connect('localhost', 'root', 'password')*

iv.Change *password* to the password that you use to access to the MySql database.

root is the default username to access the database. However, you can change it to the username that you created while installing MySql.

Step 4:

Access into MySql and create a database. name the database 'PBLFacilitator'.

you can download tools that provides GUI for MySql to make database maintenance easier

Step 5:

Open your browser and run the url: http://localhost/PBLClassroom/PreInstall.php

The system installation is completed.

To access the system, run the url: http://localhost/PBLClassroom/LogIn/LogIn.php

Login as administrator with the username: *default* and password: *default* (both default value). Then change the username and password in the user 's profile.

C. System LogIn

All links in the LogIn.php is accesssible to all users (members or non-members).

All registered users log into the system via the same page using the Log In form in the middle of the page (Figure 9.1). If the user inputted correct username and password, then the system will open to the Main Page(Figure 9.11). Otherwise the system will prompt an error message as exampled in Figure 9.2.

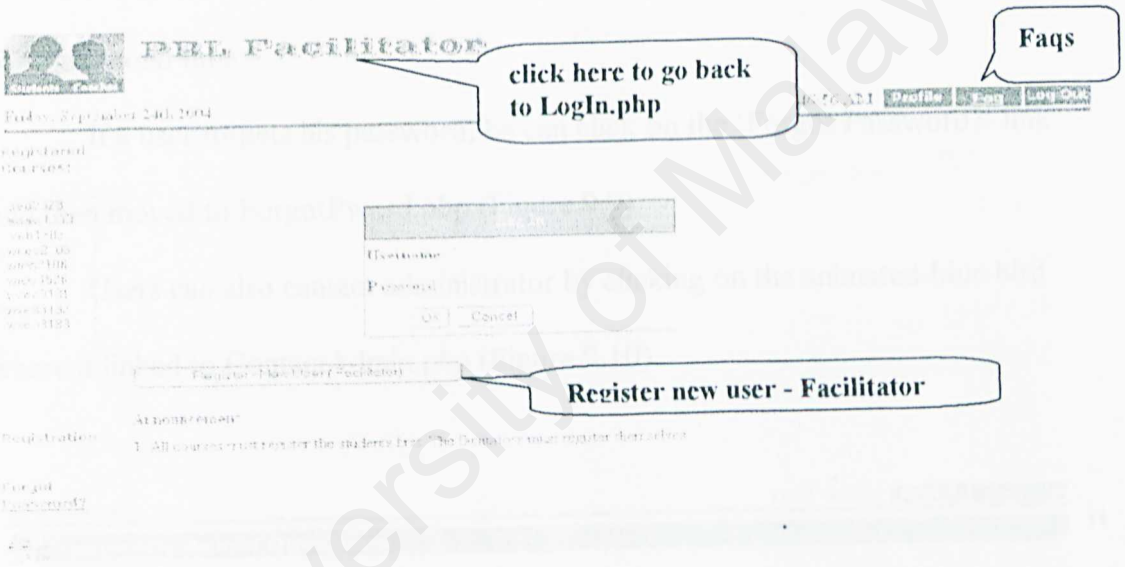


Figure 9.1: The front page for LogIn.php

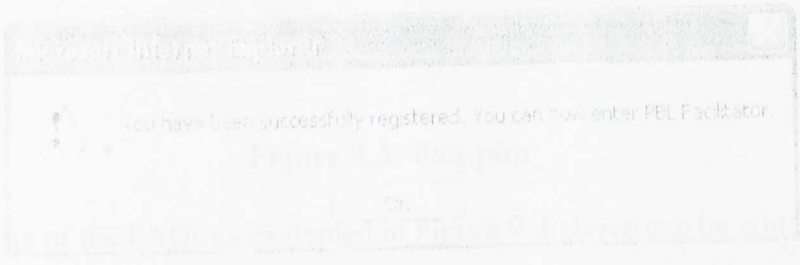


Figure 9.2: Example of prompted message

The Faqs link at the top right corner link to Faq.php that shows the lists of of frequently asked questions. (Refer to Figure 9.3). The other two links: Profile and Log Out will only be active after users has logged into the system.

The button below the log in form goes to Register_NewUser.php (Figure 9.4) for facilitators that wanted to register into the system.

When the curser moves over the 'Registration' link, a two-listed menu : 'Facilitator' and 'Student' is pulled down. User can choose either one as it serves two purpose, to confirm facilitators' registration (Figure 9.5) and also to confirm students' registration (Figure 9.7). The menu will disapper if the cursor moves over it for the second time.

If a user forgets his password, he can click on the 'Forgot Password?' link and then moved to ForgotPword.php (Figure 9.9).

Users can also cantact administrator by clicking on the animated-blue bird where it linked to ContactAdmin.php (Figure 9.10).

Figure 9.3: Faq.php

The lists of the FAQs as exampled in Figure 9.3 above can be edited by administrator by opening the file PBLClassroom/incfiles/faq.php using any web development tool.

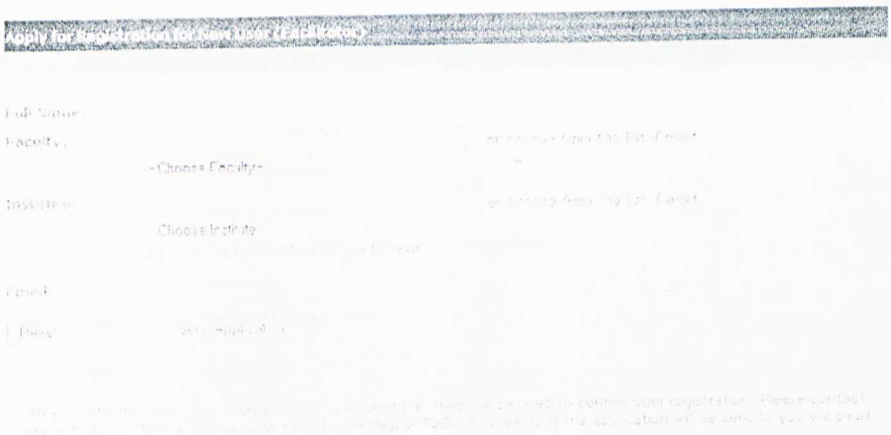


Figure 9.4: Register_NewUser.php

New user must filled in all the fields before sending the form by clicking on the 'Send Application' button. If any field is empty, then user will be prompted an error message, else, user will be prompted with acknowledgement message.

Figure 9.5: Register_Facilitator.php

In Figure 9.5, user has to fill the UserName and Password fields with the ones that were generated by the system and sent to user via email. If all the data matches with the data in database, system then will proceed to the next page (Figure 9.6). If not, then error message will be prompted.

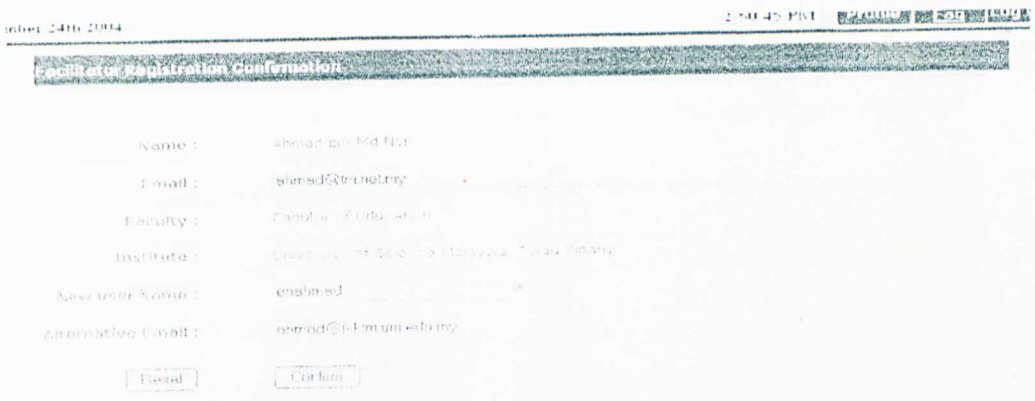


Figure 9.6: Register_FacilitatorNext.php

The field which has red asterisk beside it is a compulsary field. It means that user must fill in the field before it can be fully registered as a member. An error will be prompted if it's left empty. When the user clicked the 'Confirm' button, then the user is officially a registered member of PBL Facilitator.

Figure 9.7 and 9.8 are the web pages for confirming students' registration. The flow and process involved are basically the same as the facilitators'. Accept that the username and password to be filled in Figure 9.7 are the ones that the system automatically generate and then sent via email to the student's facilitator, whom then gave them to the students.

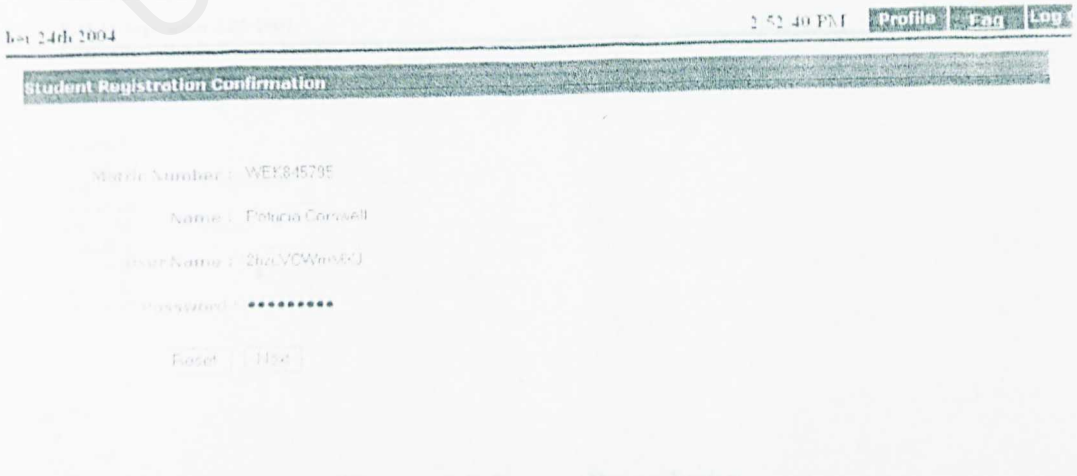


Figure 9.7: Register_Student.php

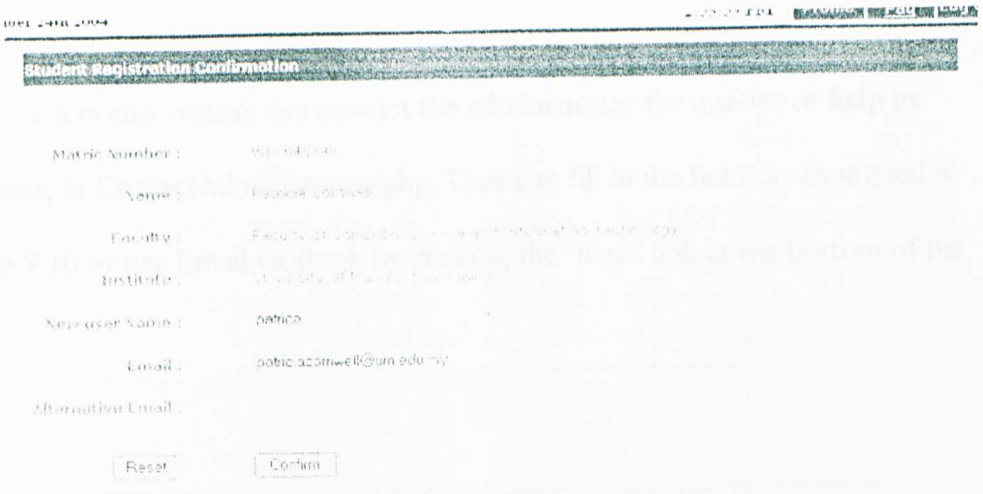


Figure 9.8: RegisterStudent_Next.php

User can go to ForgotPword.php if he forgot his password. In Figure 9.9, user can input his username and email address in the form fields. The email address must be one of the addresses registered under the user's profile. If the username and email address matches the ones in database, the system then will look up for the password in the database and automatically generate an email containing the password to user's email address.

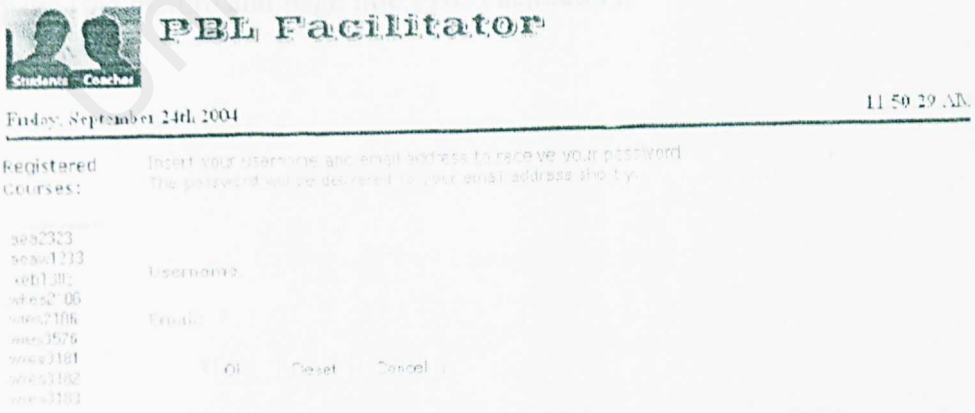


Figure 9.9 ForgotPword.php

Users and visitors can contact the administrator for queries or help by accessing to ContactAdministrator.php. User can fill in the fields as exampled in Figure 9.10 or use Email Outlook by clicking the 'here' link at the bottom of the page.

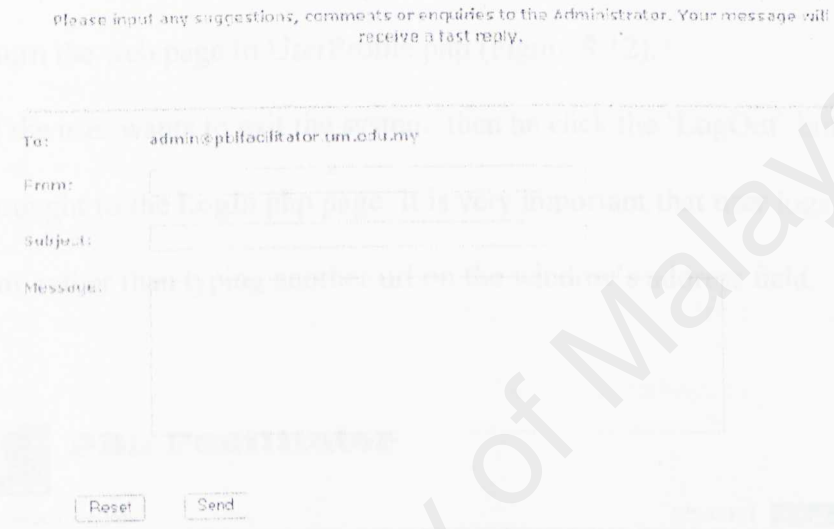


Figure 9.10: ContactAdmin.php

A user can always go back to the first page (LogIn.php) by clicking the head of the page (the picture and page title:PBL Facilitator).

D. Main Page

The first page that user comes into after logging in is the MainPage.php.

MainPage.php is actually the first page of Classroom module.

The 'Profile' and 'LogOut' links at the top right corner of the page are now actives. The 'Faq' link is the same as in the LogIn.php page. Clicking the 'Profile' link will turn the web page to UserProfile.php (Figure 9.12).

If the user wants to exit the system, then he click the 'LogOut' button and then be brought to the LogIn.php page. It is very important that user logs out to exit the system, rather than typing another url on the window's address field.

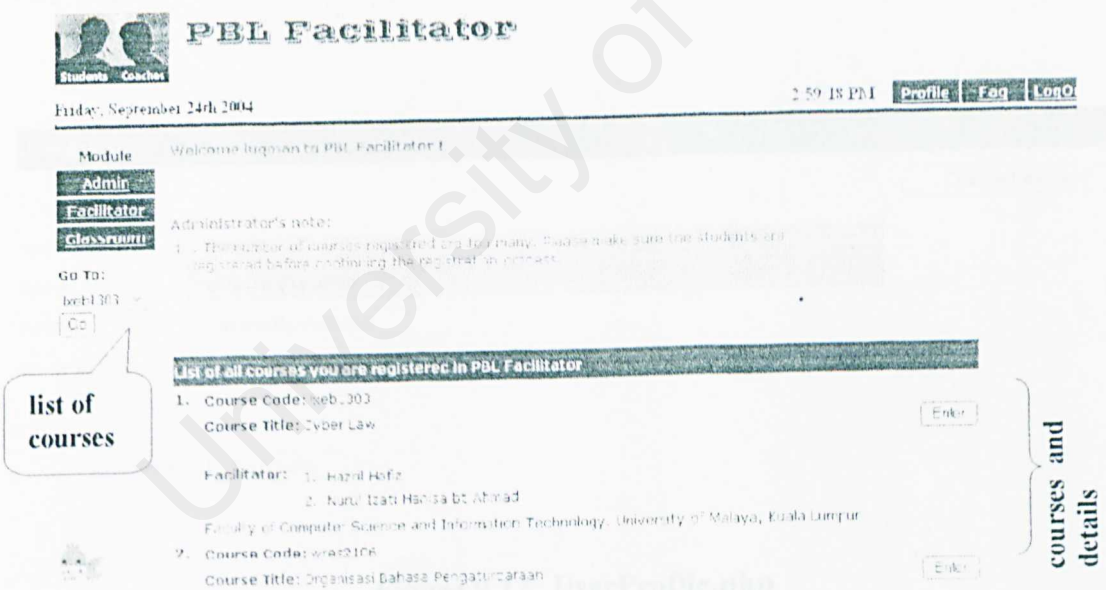


Figure 9.11: MainPage.php

At the left side of the web page, are 3 columns in blue-black color. The columns are : 1. Admin Module (administrator's access only)

- links to Administrator.php (Figure 9.14)

2. Facilitator Module (administrator and facilitator's access only)

- links to Facilitator.php (Figure 9.24)

3. Classroom Module (all user)

- links to MainPage.php (Figure 9.12)

Below the colums, there is a drop-down list of courses. And the right side of the page lists courses and their details. The lists of courses displayed depends on the status of the user. If the user is an administrator, then all the courses registered will be displayed. And he can enter any of the course's web page either by choosing from the drop-down list and click the button 'Go' or scroll down through all the courses and the details and click the button 'Enter'.

The same goes for facilitators and students. The only difference is that the list only display the courses that the facilitator or student registered to.

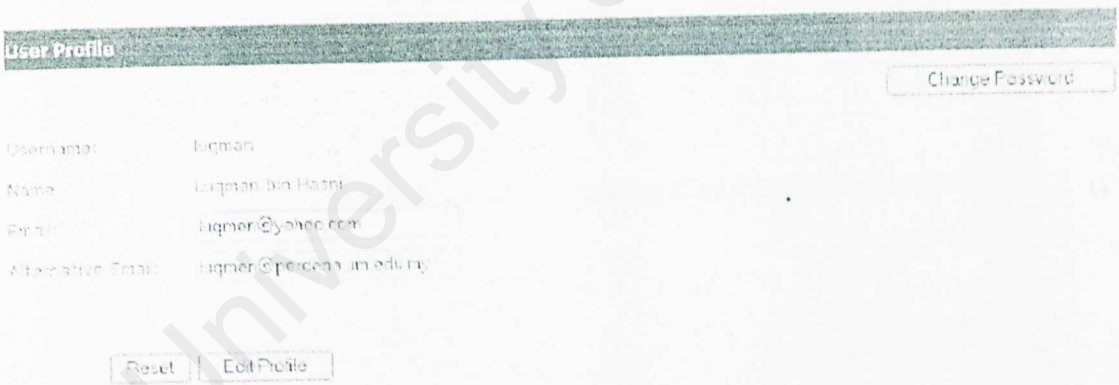


Figure 9.12: UserProfile.php

In the Figure 9.12 above, user can only edit his email and alternative email. If he wants to change his password, he can click the button 'Change Password' to go to ChangePword.php (Figure 9.13).

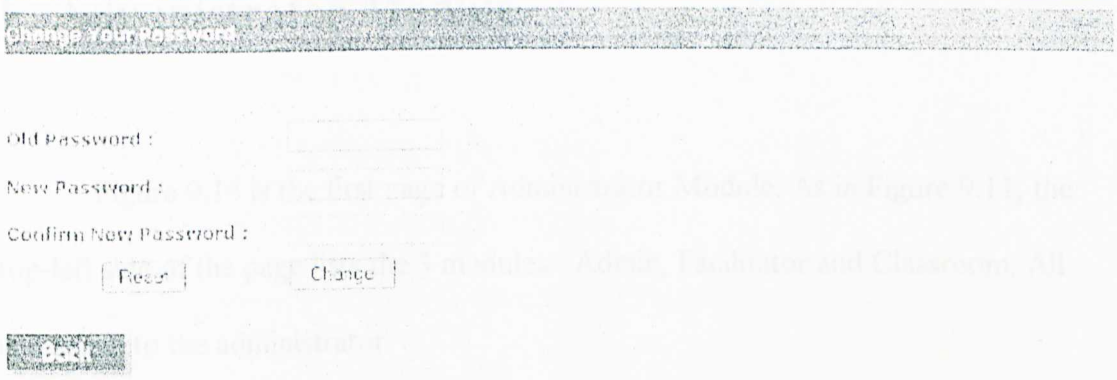


Figure 9.13: ChangePword.php

The 'Back' button will bring user back to the previous page, the UserProfile.php.

E. Administrator Module

Figure 9.14 is the first page of Administrator Module. As in Figure 9.11, the top-left side of the page lists the 3 modules : Admin, Facilitator and Classroom. All accessible to the administrator.

Directly below the columns is the drop-down list of all the courses registered in PBL Facilitator. Administrator can view general information regarding the course by choosing a course code and clicking the button 'Go'. The button links to the CourseInfo.php (Figure 9.15).

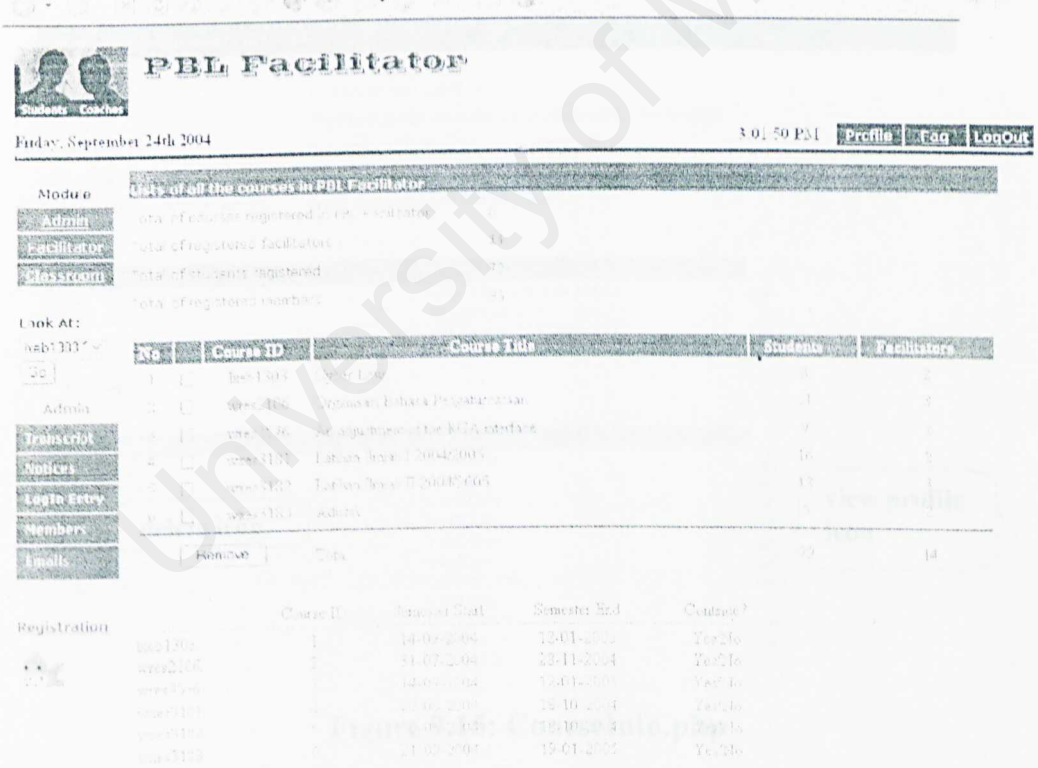


Figure 9.14: Administrator.php

And below the list, are 5 columns that each links to different page.

Column Transcript links to Evaluation.php (Figure 9.16).

Column Notices links to Notices.php (Figure 9.18).

Column LogIn Entry links to LogInEntry.php (Figure 9.19).

Column Members links to Members.php (Figure 9.21).

Column Emails links to EmailReport.php (Figure 9.22).

The 'Registration' link below the columns links to RegisterApp.php (Figure 9.23).

In the Administrator.php, administrator can view the total of registered courses, facilitators and students, and the starting and ending semester for each course. Administrator can also unregister a course and delete all it's details by checking a checkbox and click the 'Remove' button.

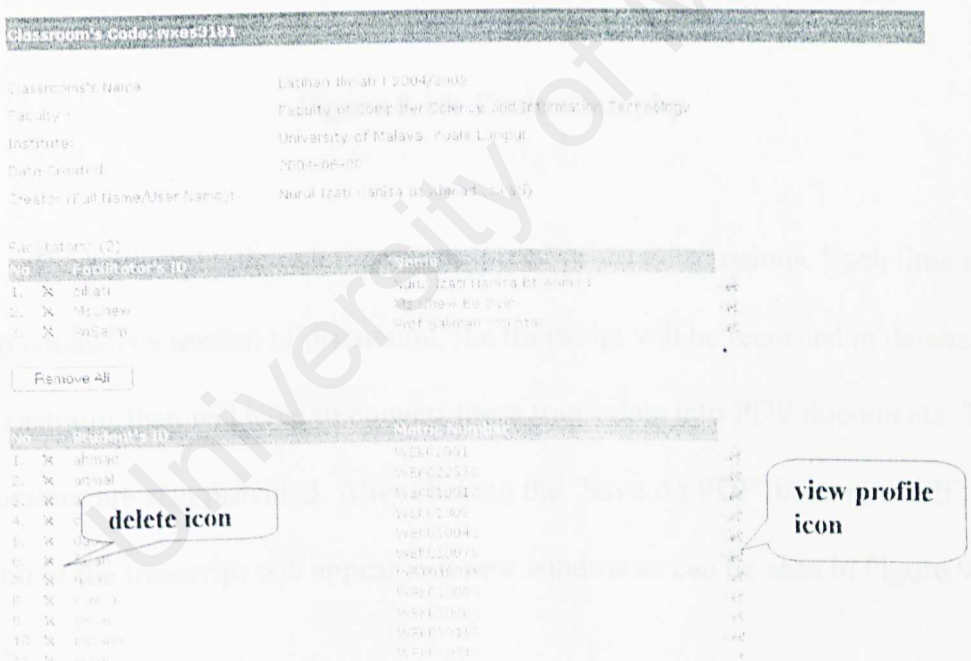


Figure 9.15: CourseInfo.php

CourseInfo.php displays the general detail of a course. Administrator can remove a student or facilitator from a course if necessary and it can be done by clicking the delete button before the Facilitator's or Student's ID that's intended to

be removed. The 'Remove All' button will delete all facilitators' or students' membership from the course.

Details of each member can be view by clicking on the view profile icons.

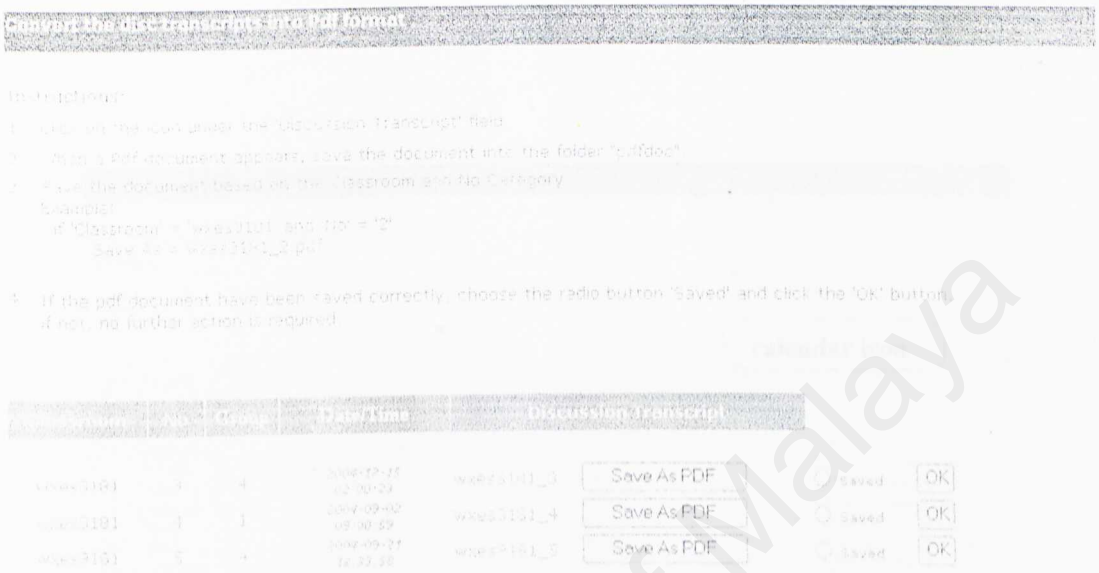


Figure 9.16: Evaluation.php

Evaluation.php lists all the transcripts of private discussions. Each time a group conducts a session of discussion, the transcript will be recorded in database. Administrator then will have to convert these transcripts into PDF documents. The instructions are also provided. After clicking the 'Save As PDF' button, a .pdf version of the transcript will appear in a new window as can be seen in Figure 9.17 below.

Notices.php displays all the administrator's notes on the LogIn.php, Classroom.php and also Facilitator.php. The notices can be deleted by clicking the 'Delete All' button (all notices) or the delete icon (one by one). They can also be edited by clicking the edit icon. And new notices can be added with selections on the location of the new notice.

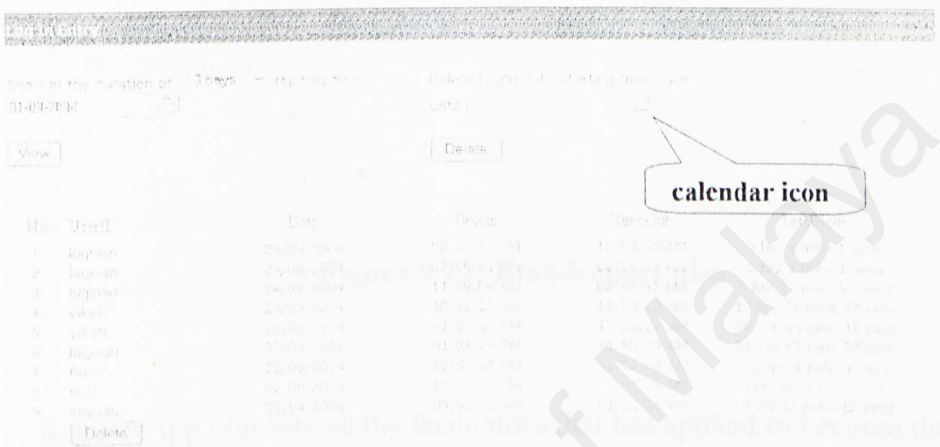


Figure 9.19: LogInEntry.php

LogInEntry.php displays the recorded date and times (in and out) each time a user logs into the system. Administrator can view and delete the entries based on the number of days required. The dates can be choosen by clicking on the calendar icon to open calendar.html (Figure 9.20).

Figure 9.20: calendar.html

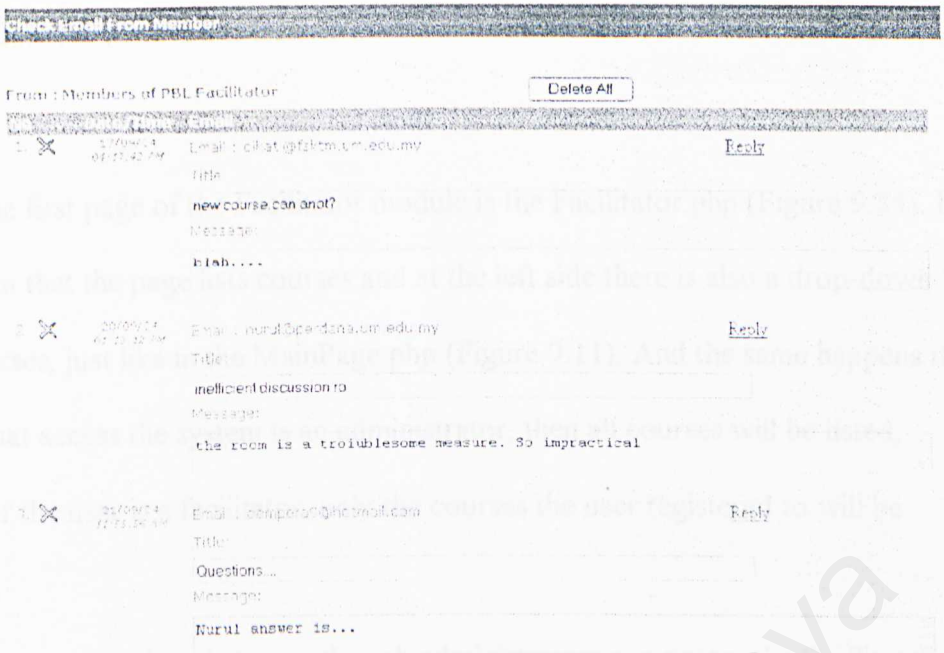


Figure 9.22: Emailreport.php

RegisterApp.php lists all the facilitators that has applied to become the new user of PBLFacilitator. If administrator approves of the application, administrator will

check on the checkbox and click the button 'Accept Application'. The system then will create a username and password and automatically generate an email containing the username and password to the person's email address. If the 'Reject' button is chosen, then the user's data indatabase will be deleted and the system will generate an email informing the person of the rejection.

F. Facilitator Module

The first page of the Facilitator module is the Facilitator.php (Figure 9.24). It can be seen that the page lists courses and at the left side there is also a drop-down list of courses, just like in the MainPage.php (Figure 9.11). And the same happens if the user that access the system is an administrator, then all courses will be listed, where as if the user is a facilitator, only the courses the user registered to will be displayed.

Please remember that even though administrators can access the Facilitator module, he can only view the pages without provileges to all the functions available.

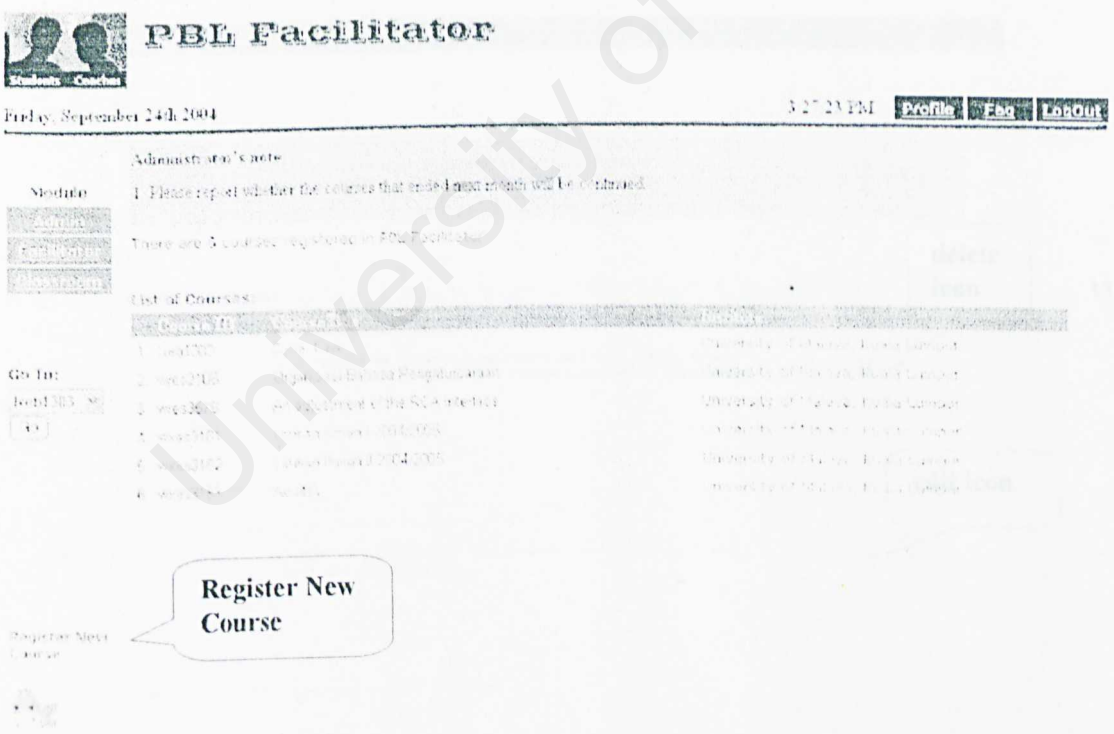


Figure 9.24: Facilitator.php

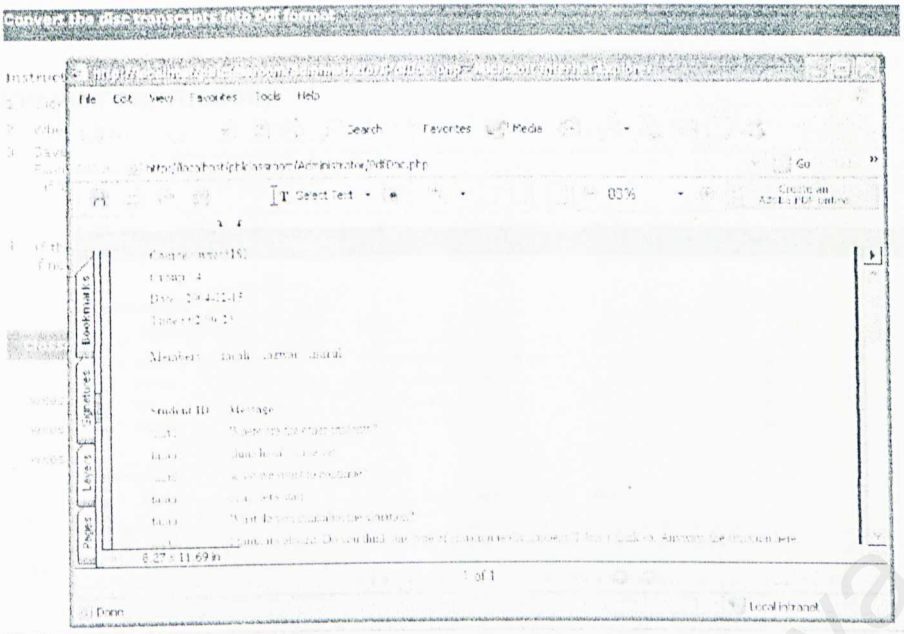


Figure 9.17: PdfDoc.php

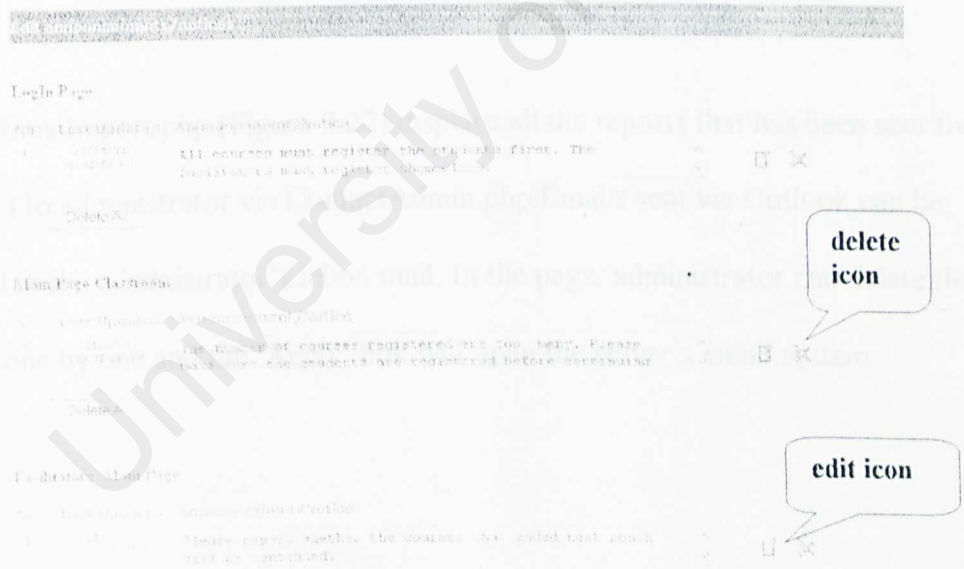


Figure 9.18: Notices.php

Members.php lists all the registered and unregistered (students whom haven't yet confirm their registration).

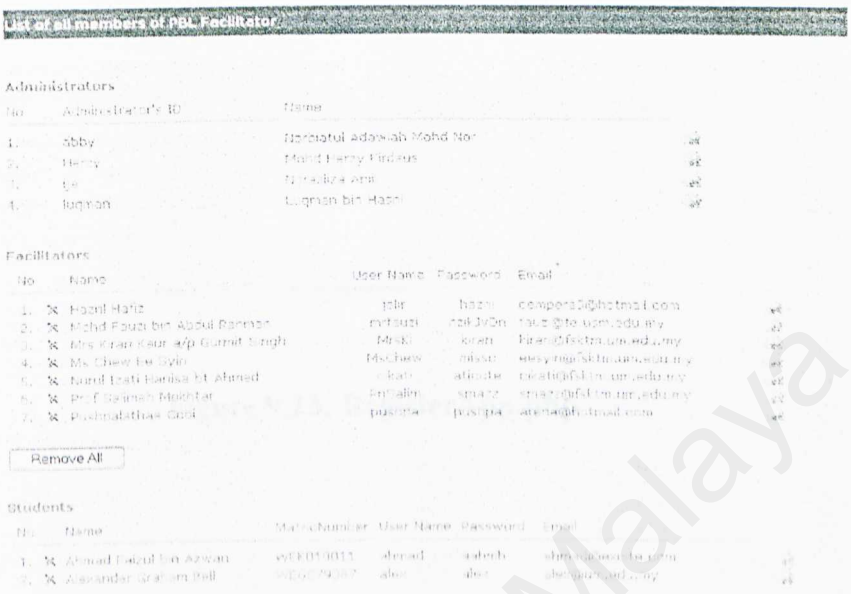


Figure 9.21: Members.php

Emailreport.php (Figure 9.22) displays all the reports that has been sent from the users to administrator via ContactAdmin.php.Emails sent via Outlook can be checked in the administrator's inbox mail. In the page, administrator can delete the reports one by one and the 'Reply' link will open the server's email system.

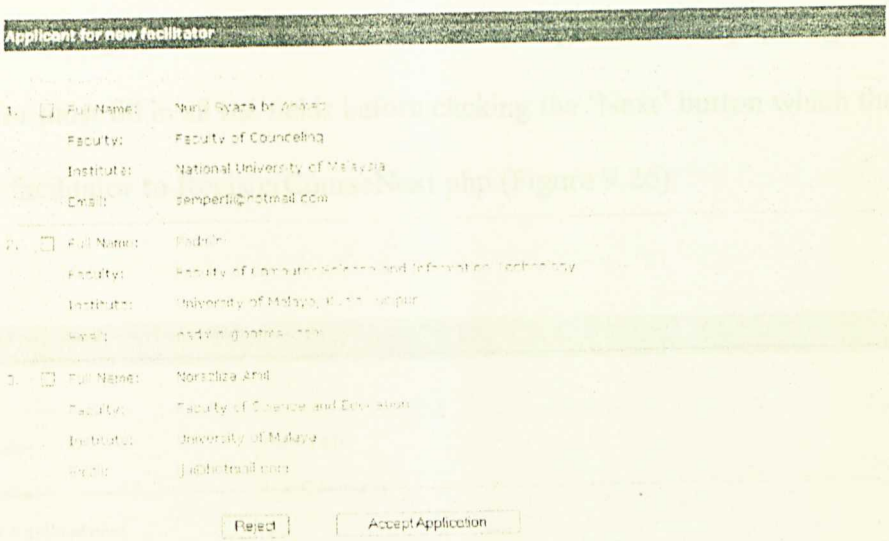


Figure 9.23: RegisterApp.php

The 'Register New Course' links user to RegisterCourse.php (Figure 9.25). Facilitator must fill in all the fields before clicking the 'Next' button which then will take the facilitator to RegisterCourseNext.php (Figure 9.26).

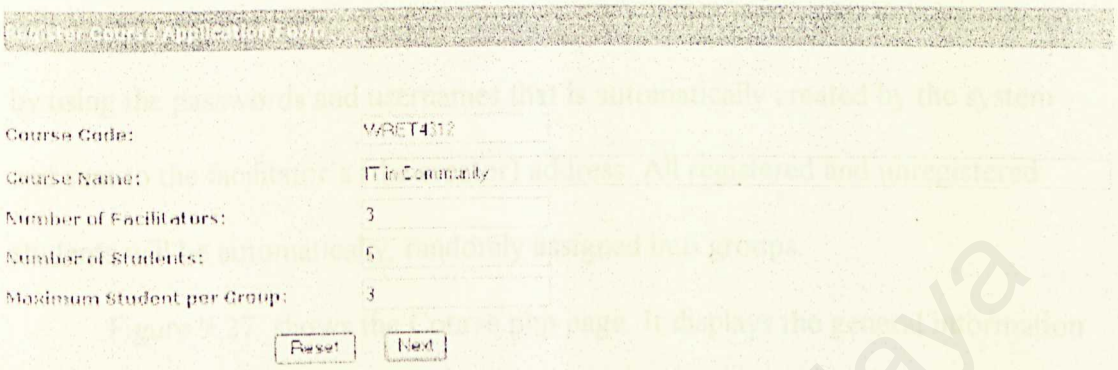


Figure 9.25: RegisterCourse.php

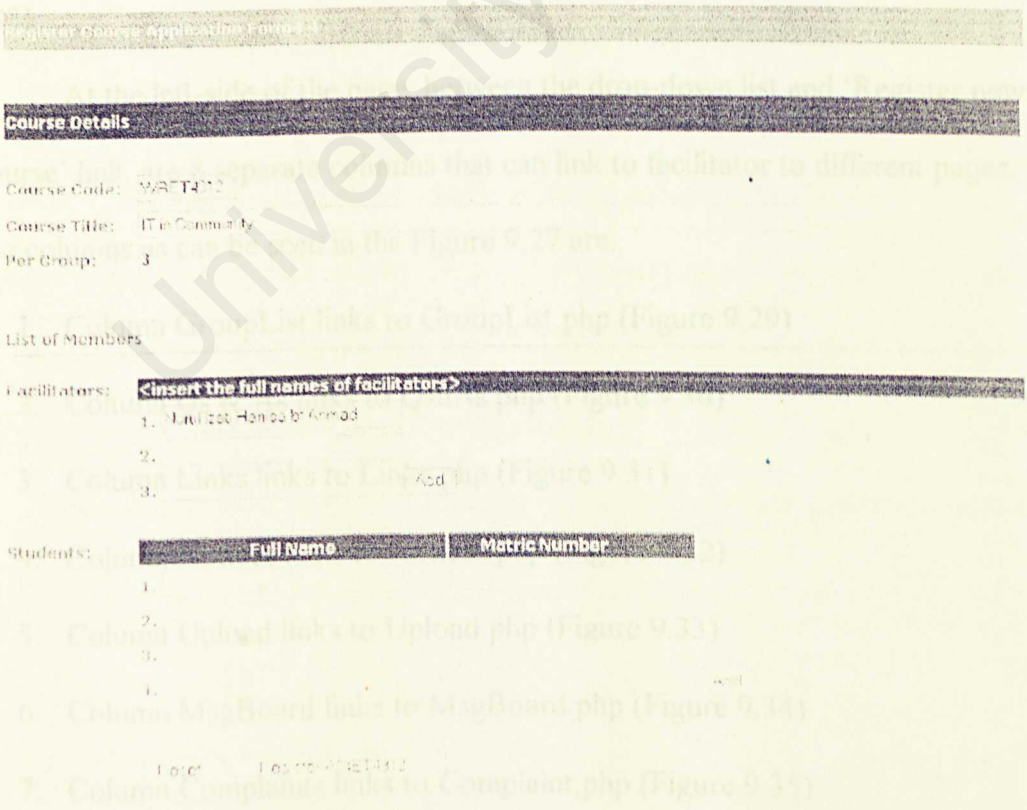


Figure 9.26: RegisterCourseNext.php

After all the fields have been filled and the 'Register CourseCode' button is clicked, the system then will create tables in the database for the new course and check which students and facilitators listed are members of PBL Facilitator.

Registered users will be recorded as members of the course, while the rest will have to register themselves (facilitator) and confirm their registration (student) by using the passwords and usernames that is automatically created by the system and sent to the facilitator's (the creator) address. All registered and unregistered students will be automatically, randomly assigned into groups.

Figure 9.27 shows the Course.php page. It displays the general information of the course and enables facilitator to remove students' membership to the classroom by clicking on the delete icon or viewing member's profile by clicking the view profile icon. Facilitator can also add new member to the course by clicking on the 'Add Member' button that'll take the facilitator to the AddMember.php (Figure 9.28).

At the left-side of the page, between the drop-down list and 'Register new Course' link, are 8 separate columns that can link to facilitator to different pages. The columns as can be seen in the Figure 9.27 are:

1. Column GroupList links to GroupList.php (Figure 9.29)
2. Column Qs & As links to QsnAs.php (Figure 9.30)
3. Column Links links to Links.php (Figure 9.31)
4. Column Notices links to Notices.php (Figure 9.32)
5. Column Upload links to Upload.php (Figure 9.33)
6. Column MsgBoard links to MsgBoard.php (Figure 9.34)
7. Column Complaints links to Complaint.php (Figure 9.35)
8. Column Evaluation links to Evaluation.php (Figure 9.36)

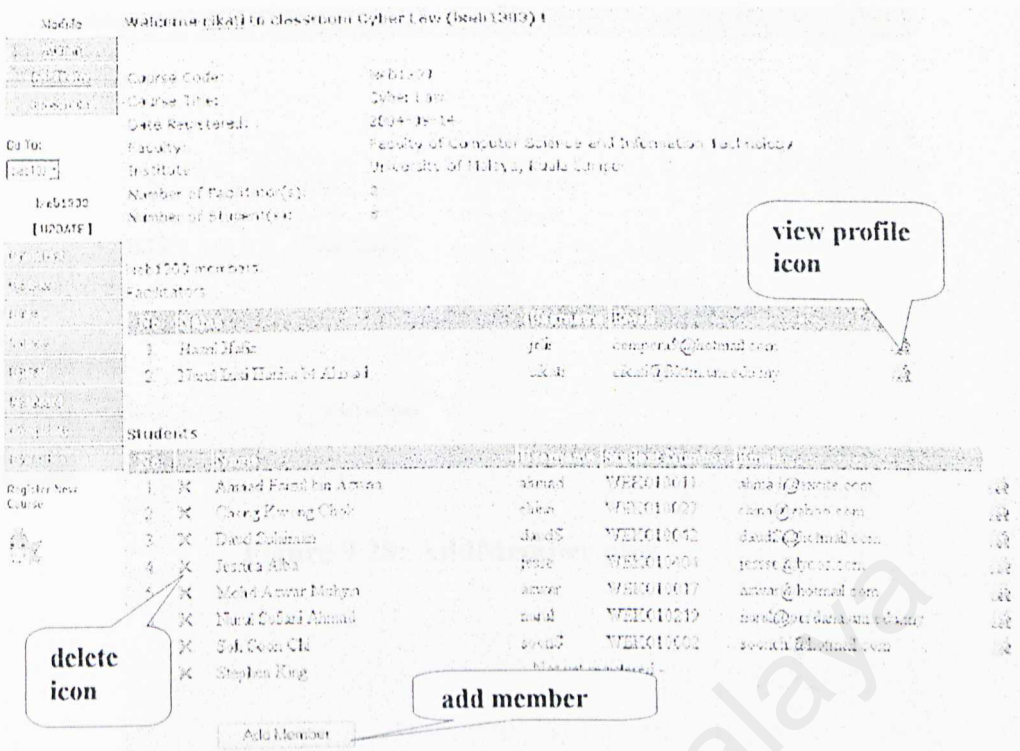


Figure 9.27: Course.php

In AddMember.php, facilitator can add new member (facilitator or student). For new facilitator, only registered facilitator can be added as new member, otherwise, the course has to wait until the person has registered themselves. While for student, non-registered student will be given username and password for confirming registration via email to the facilitator's address. Newly added student will be assigned to default group number: 0. They will then be assigned to other group by the course's facilitator in GroupList.php.

Any attemp to input data of member of the course will be prompted with a warning.

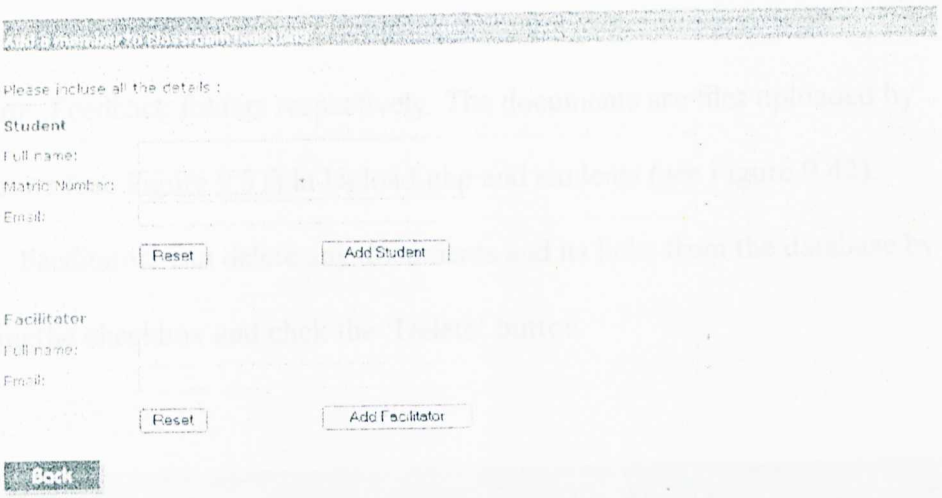


Figure 9.28: AddMember.php

GroupList.php (Figure 9.29) lists all the students by their matric numbers in their respective groups. Facilitators can change the lists of students in each group by choosing matric numbers from the drop-down matric number lists.

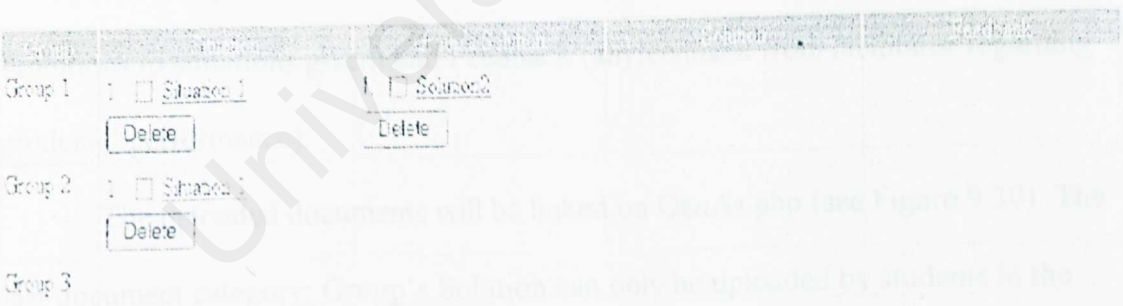


Figure 9.30: QsnAs.php

Figure 9.30 shows the QsnAs.php. The page has a table that consists of 5 columns; Group column, Situation column, Group's Solution column, Solution column and Feedback column. Each column lists links to documents stored in

Assgnmt_Problems, Assgnmt_GroupSol, Assgnmt_Solutions and Assgnmt_Feedback folders respectively. The documents are files uploaded by facilitators (see Figure 9.31) in Upload.php and students (see Figure 9.42).

Facilitators can delete any documents and its links from the database by checkingthe checkbox and click the 'Delete' button.

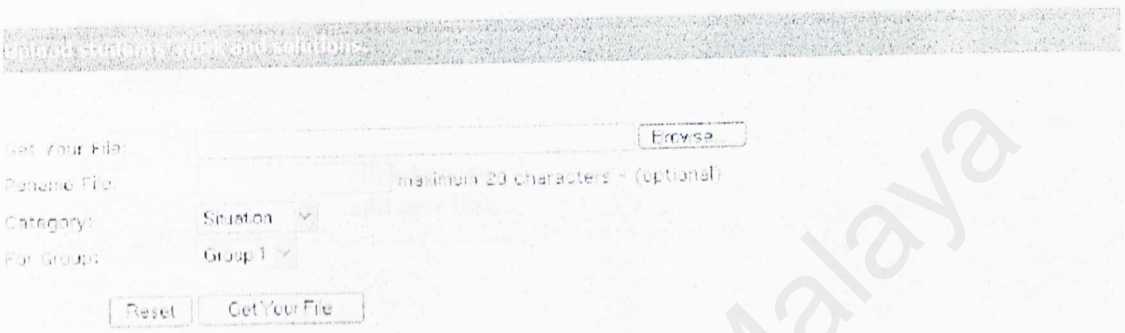


Figure 9.31: Upload.php

In Upload.php, facilitators can upload documents regarding students' assignments. There are 3 categories of the documents; Situation (e.g case study), Solution (solutions to situations given) and Feedback (anyfeedback from facilitator regarding students' performance).

The uploaded documents will be linked on QsnAs.php (see Figure 9.30). The last document category; Group's Solution can only be uploaded by students in the student module.

Figure 9.32 shows the Links.php. Facilitators can delete or edit existing links or add new links. There are 2 kinds of links; links for web sites and links for useful documents or files.

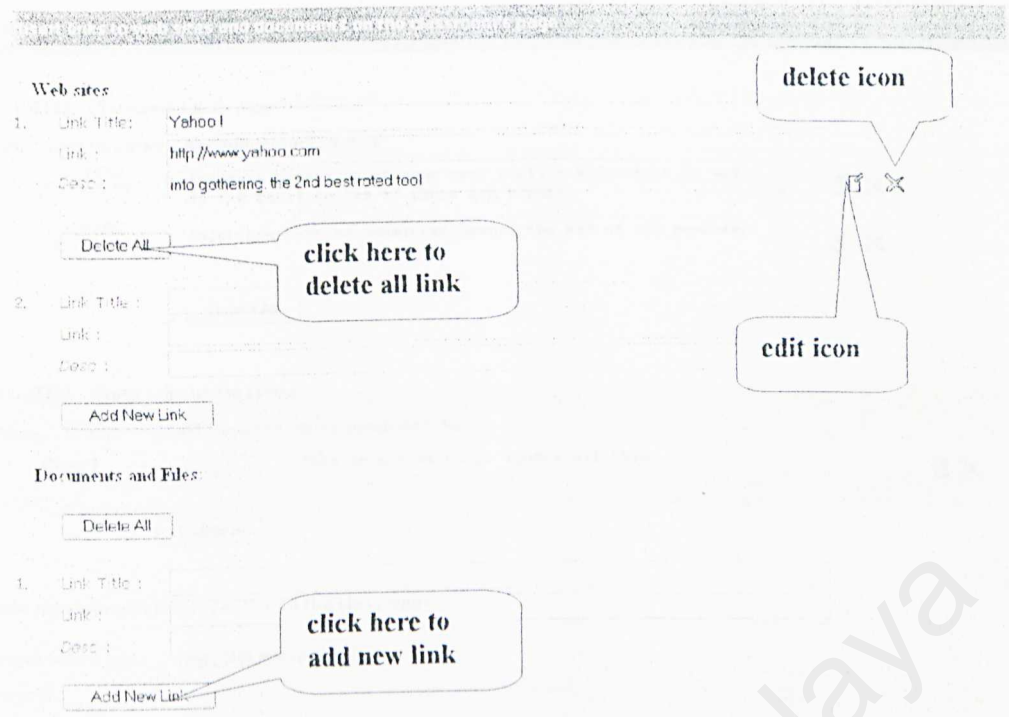


Figure 9.32: Links.php

Facilitators can add or edit all the annouincements to the course's students in Notices.php, as shown below in Figure 9.33. Like Links.php, facilitator can edit, delete or add new notices either in the course' classroom's main page (Classroom.php) or in the private discussion rooms (DiscRoom.php), both in Classroom module.

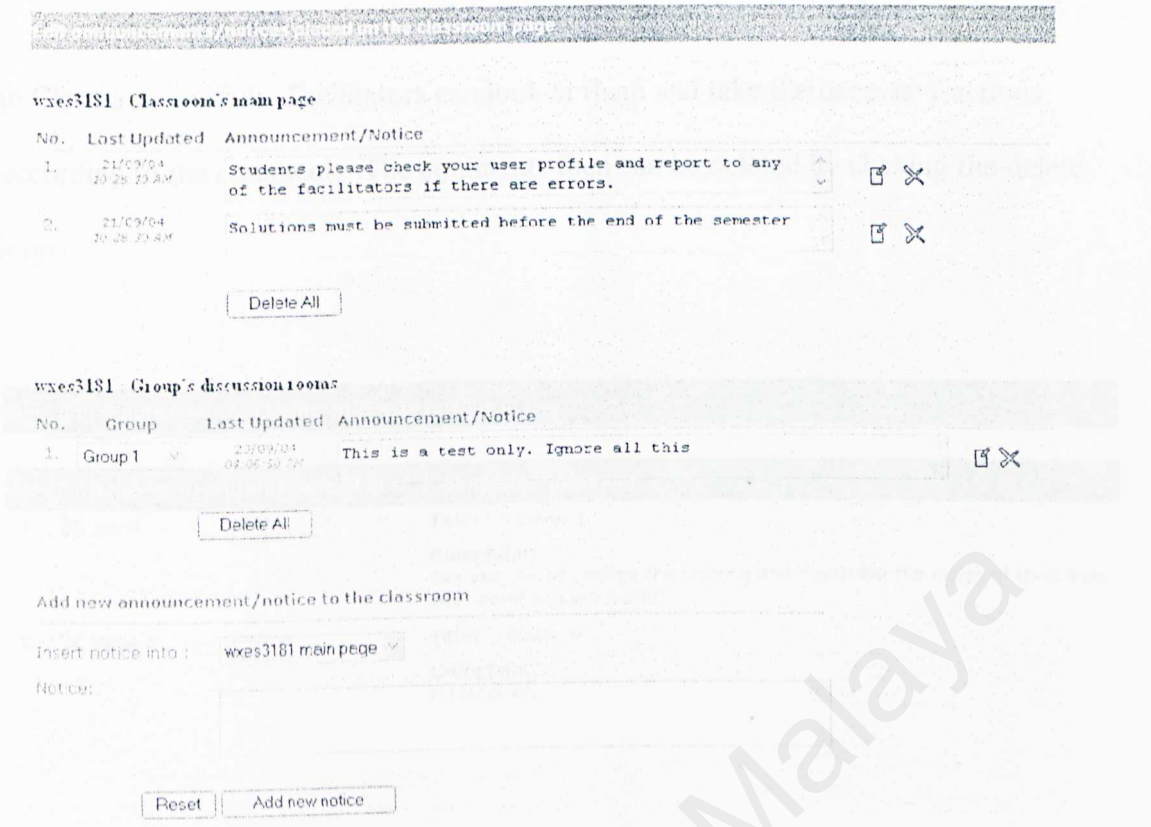


Figure 9.33: Notices.php

MsgBoard.php lists all the threads and its corresponding messages displayed in Classroom.php in Classroom module. Facilitators can view all the threads easily and delete any of the threads.

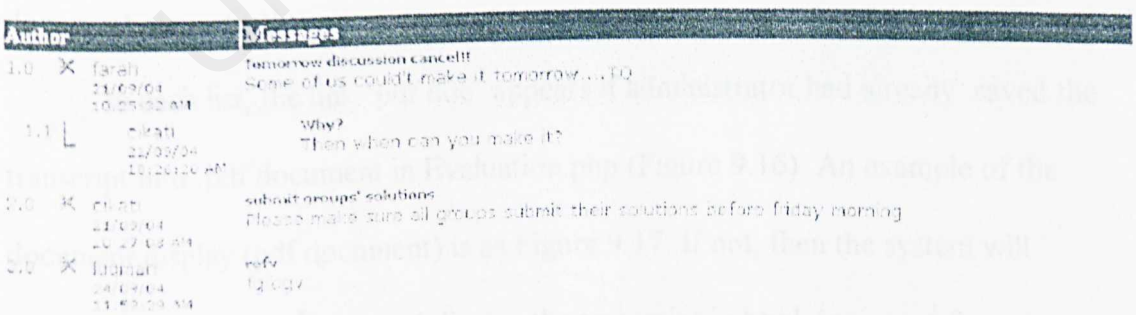


Figure 9.34: MsgBoard.php

Complaint.php displays all the comments/complaints issued by the students in Classroom module. Facilitators can look at them and take the necessary actions according to the comments. The comments then can be deleted by clicking the delete icon.

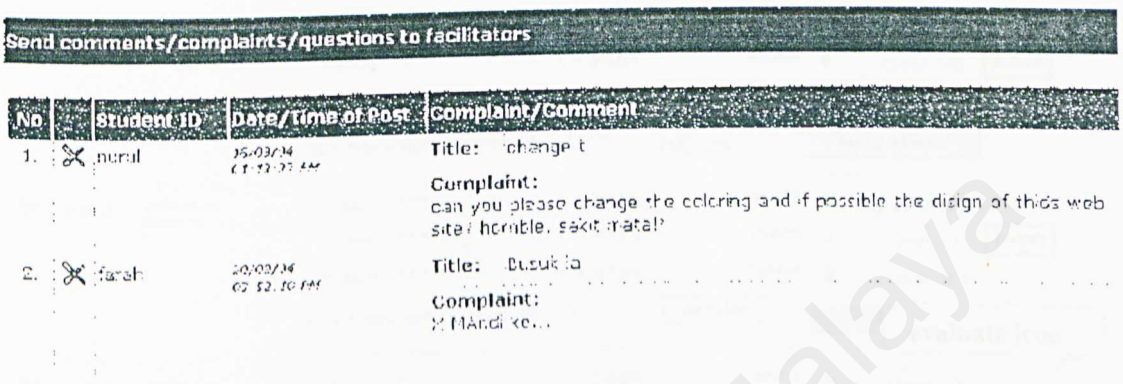


Figure 9.35: Complaints.php

Evaluation.php lists the details whenever a discussion session has been conducted. The system's evaluation result is also displayed and facilitator can view the transcript of the discussion by clicking on the button 'html doc' or the 'PDF doc' link. Either one will open a new window and display html document and pdf document respectively.

In each list, the link 'pdf doc' appears if administrator had already saved the transcript into .pdf document in Evaluation.php (Figure 9.16). An example of the document display (pdf document) is as Figure 9.17. If not, then the system will retrieve the data in database and display the transcript in html document format.

Based on the displayed transcript, facilitator can evaluate students' performance and click the evaluation icon.

Facilitator can also view the result of peer evaluation for each discussion. Clicking on the 'Peer Evaluation' button will open a new window (peerevaluation.php) that shows the result as exampled in Figure 9.37.

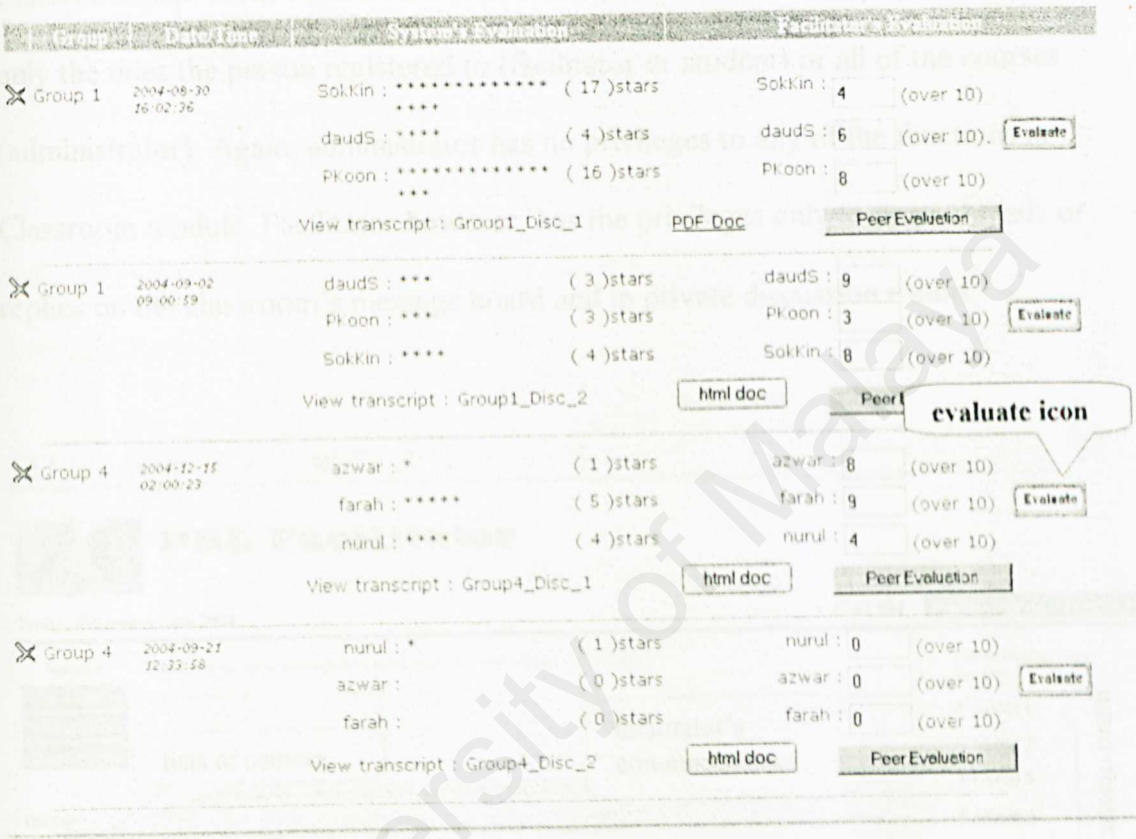


Figure 9.36: Evaluation.php .

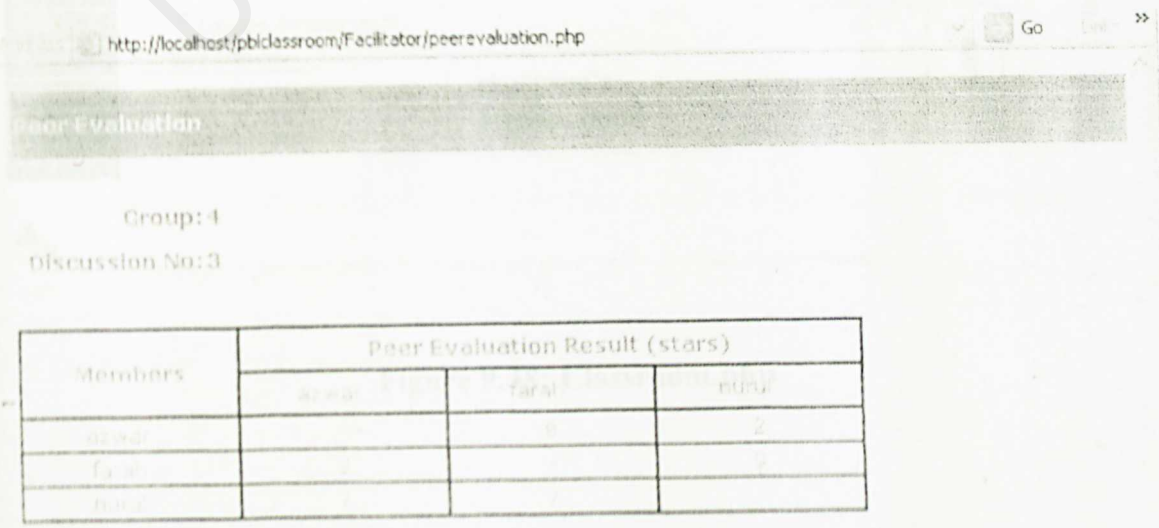


Figure 9.37: peerevaluation.php

G. Classroom Module

The second page in Classroom module (after MainPage.php) is the Classroom.php. As in Figure 9.24 (Facilitator.php), the lists of courses displayed are only the ones the person registered to (facilitator or student) or all of the courses (administrator). Again, administrator has no privileges to any of the functions in Classroom module. Facilitator however, has the privileges only to posts threads or replies on the classroom's message board and in private discussion rooms.

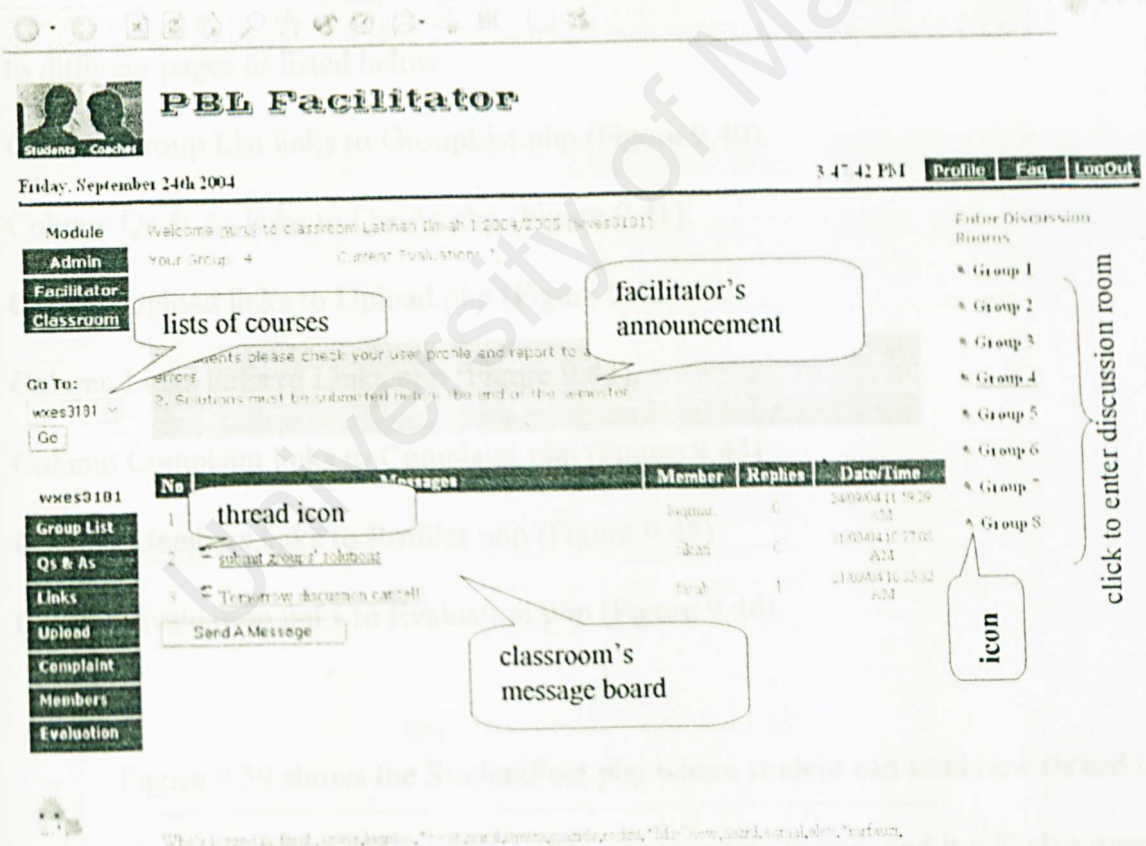


Figure 9.38: Classroom.php

User can post threads in the classroom's messageboard by clicking on the 'Send A Message' button that'll take the user to StudentPost.php (Figure 9.39). User can click on a thread icon to reply and view the thread and its correspondings

To enter any of the discussion rooms (Discroom.php, see Figure 9.47), user have to click on the small icons. Facilitators and students (member or non-members of discussion groups) can enter any of the discussion rooms and observe on-going discussions. However, only group members can start and end a discussion session, and except for non-group students, other users can posts responses during the discussion session.

There are 7 columns under the drop-down lists of courses. The columns links to different pages as listed below:

Column Group List links to GroupList.php (Figure 9.40).

Column Qs & As links to QsnAs.php (Figure 9.41).

Column Upload links to Upload.php (Figure 9.42).

Column Links links to Links.php (Figure 9.43).

Column Complaint links to Complaint.php (Figure 9.44).

Column Members links to Profiles.php (Figure 9.45).

Column Evaluation links to Evaluation.php (Figure 9.46).

Figure 9.39 shows the StudentPost.php where student can send new thread or replies. The 'New Thread' button is to let user post new thread, and it will also open the StudentPost.php.

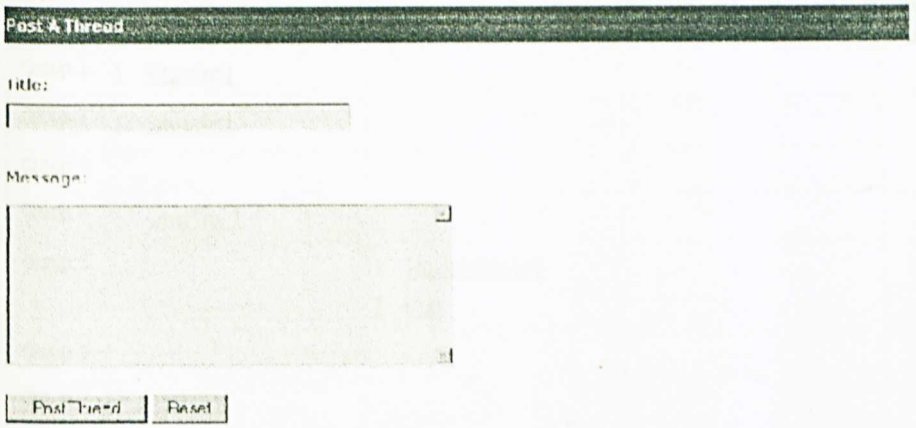


Figure 9.39: StudentPost.php

GroupList.php displays the lists of groups and members by students' matric numbers.

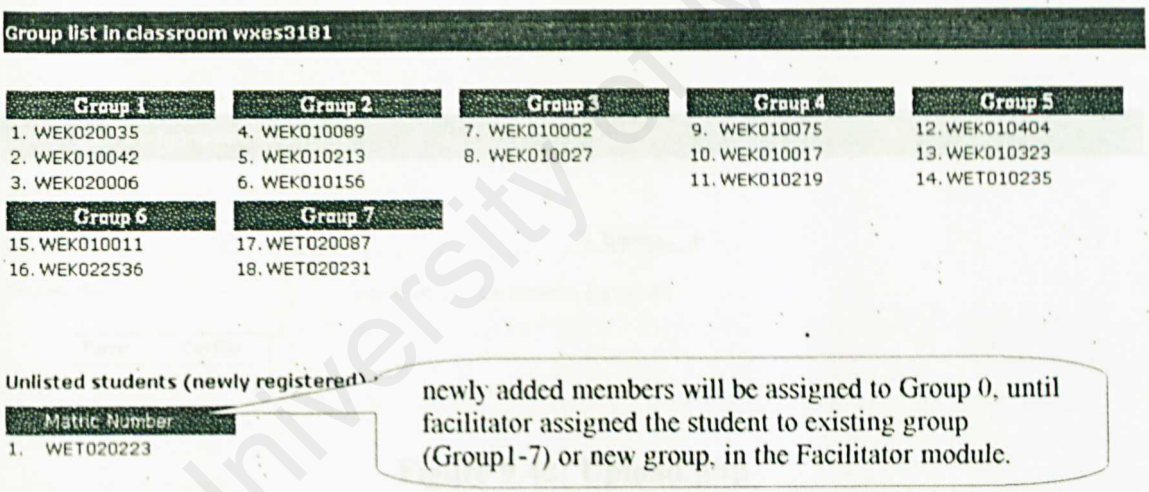


Figure 9.40: GroupList.php

QsnAs.php (Figure 9.41) displays all the links to documents uploaded by students (Figure 9.43) and facilitators (Figure 9.31). Click any of the link to open the document in new window.

Group	Situations	Group's Solutions	Solutions	Facilitator's Feedback
Group 1	1. Situation1			
Group 2	1. Situation1			
Group 3				
Group 4	1. Situation 1			
Group 5		1. Troubleshooting 2. cold		
Group 6				
Group 7				

Figure 9.41: QsnAs.php

Upload.php enables student to upload their assignment works. The uploaded documents will the be displayed under column 'Group's Solution' in QsnAs.php (Figure 9.41).

Figure 9.42: Upload.php

Links.php lists all the links that were added by facilitator in Facilitator module (see Figure 9.32).

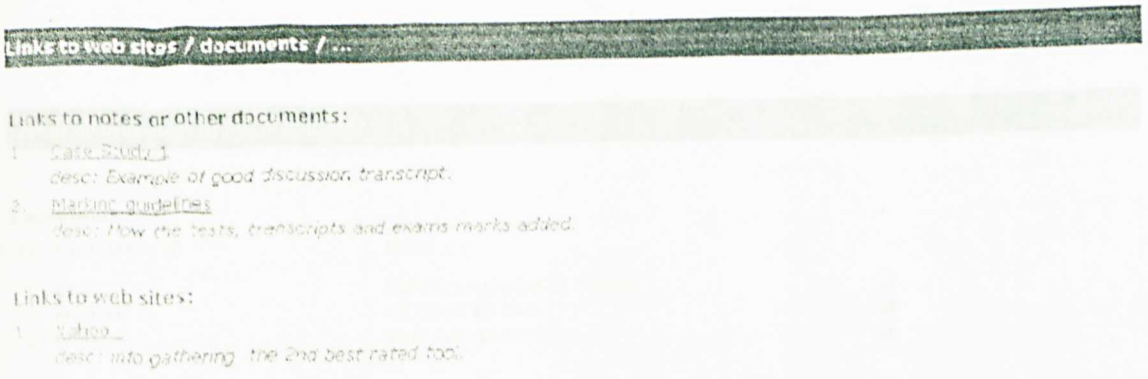


Figure 9.43: Links.php

Figure 9.44: Complaint.php

Students can complain or comment to their facilitators by filling up the Complaint form in Complaint.php. Facilitators will receive the comments in the Facilitator module.

Profiles.php (Figure 9.45) lists all the members of the classroom. User can view general profile of each member cy clicking on the view profile icon.

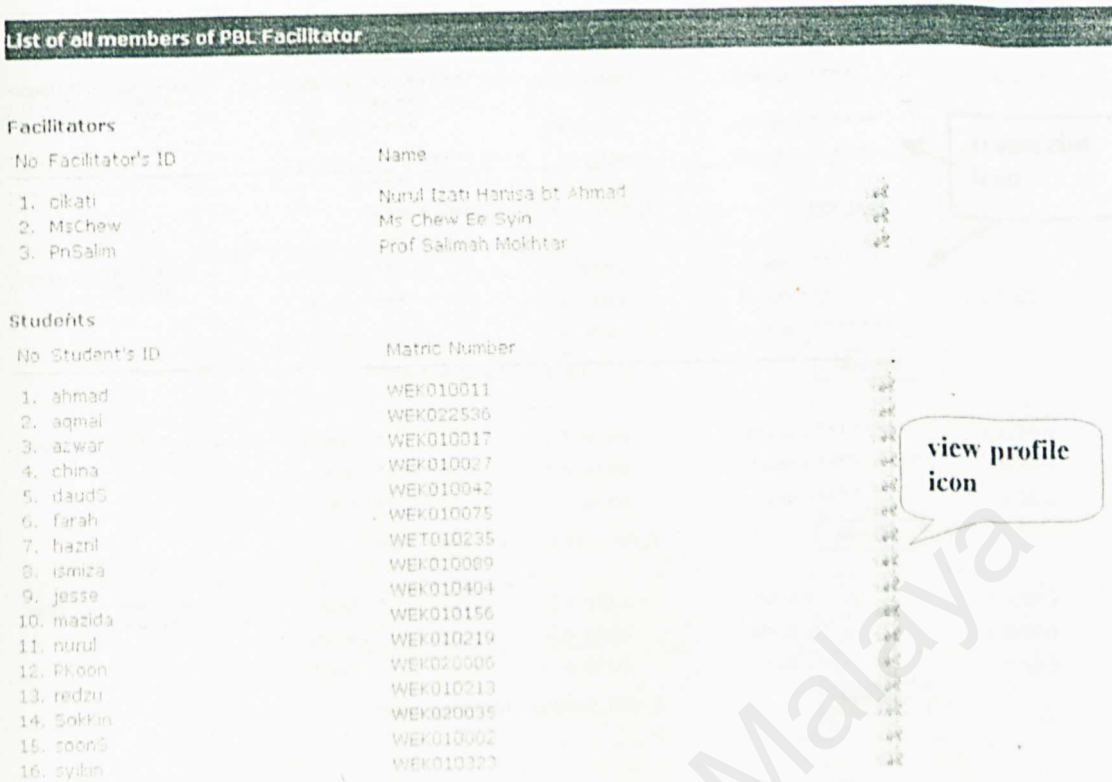


Figure 9.45: Profiles.php

Evaluation.php displays general details of each session of discussion and the system's and facilitator's evaluation result. Student can also view the transcript of the discussions by clicking the transcript icon.

The labelled 'html doc' button icon will open the transcript in new window in html document, whereas the other type of icon will open the transcript in pdf format.

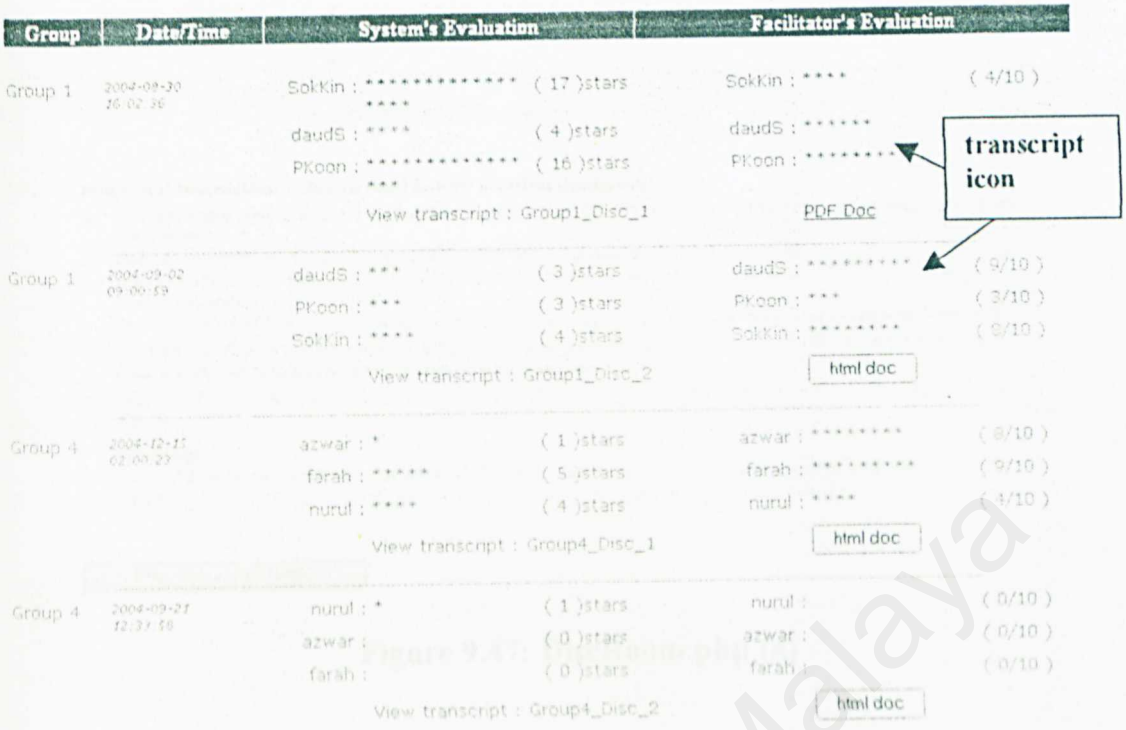


Figure 9.46: Evaluation.php

Figure 9.47 shows the DiscRoom.php. If there's no on-going session at that time, non-group members and facilitators can't enter the forum part, and so no icon to enter will be displayed. However, for group member, system will identify the user as a member and a 'Start Discussion' icon wil be displayed so that a session can be started. If, a session has already started, an 'Enter Discussion Room' icon will be displayed and all users (group members or non-group members) can enter the discussion forum.

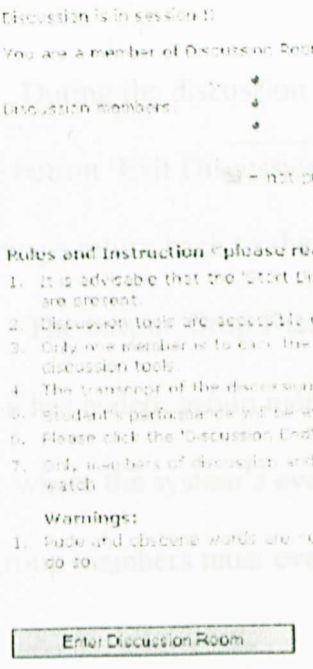


Figure 9.47: DiscRoom.php (a)

Figure 9.48 shows the interface of DiscRoom.php when a user entered the forum part. Only group-members are able to post messages, while others can only observe.

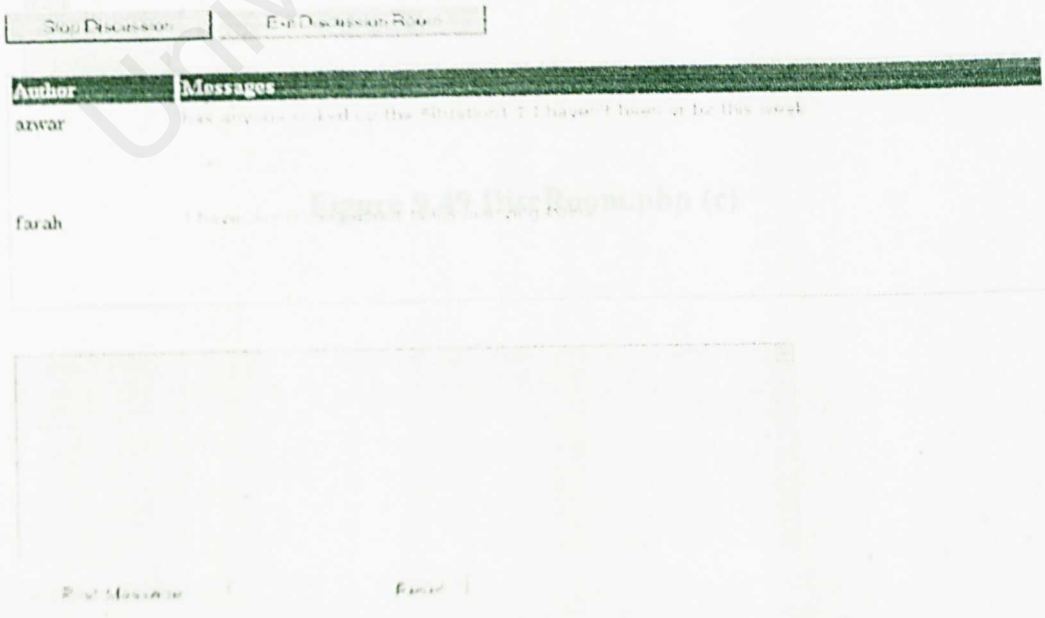


Figure 9.48: DiscRoom.php (b)

During the discussion session, users can exit the discussion room by clicking on the button 'Exit Discussion Room' that'll take them back to Classroom.php. They can always return back to the discussion room,

The button 'Stop Discussion' is only visible to group members. When a session has ended, group members must all click on this button to proceed to the next stage, where the system's evaluation of each student's perfirmance will be displayed and group members must evaluate each other. (see Figure 9.49)

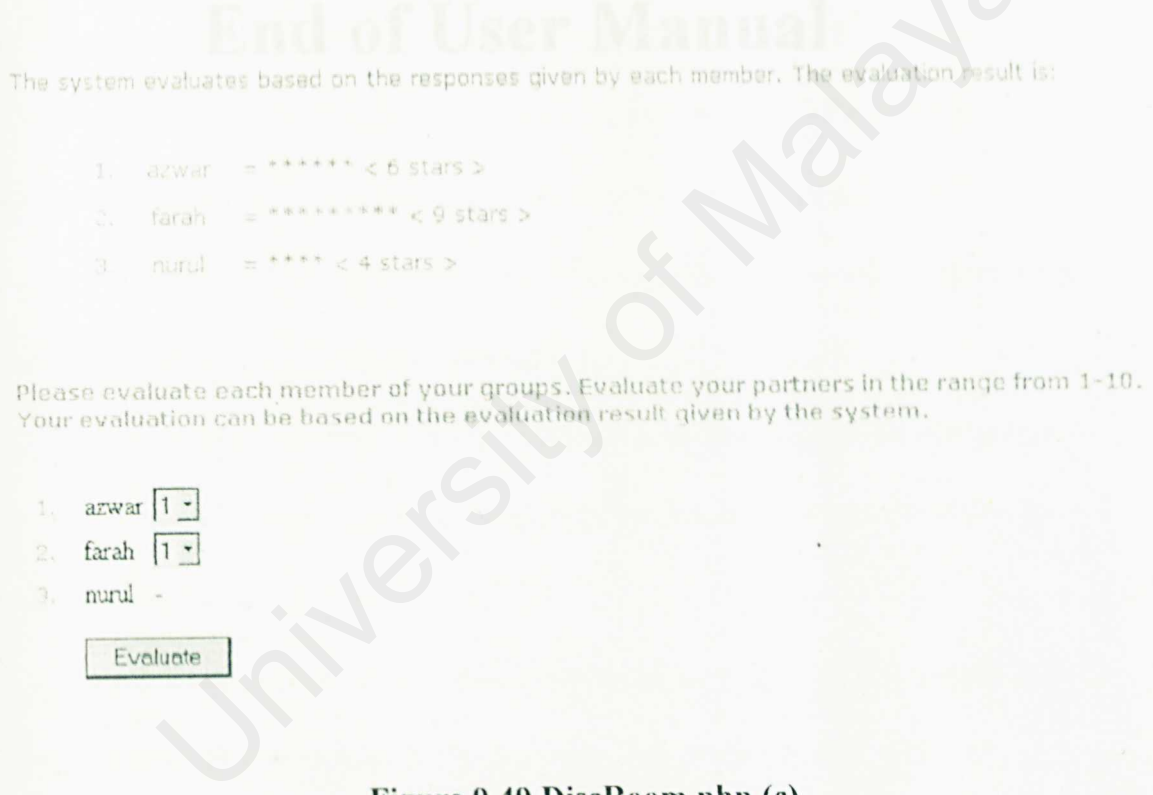


Figure 9.49 DiscRoom.php (c)

End of User Manual

Conclusion

WXES3181 is the first phase for developing PBL Facilitator. During this phase, the preliminary investigation, system's requirements analysis, and system's analysis and design were conducted to complete the phase.

The tasks stated above were completed for the purpose of creating a web-based system that manages online PBL classrooms. The final system is expected to be informal, interesting and user-friendly with easy user interface navigation and password-protected at the end of the second phase (WXES3182). The system, named PBL Facilitator, should be able to provide a better and more comfortable PBL discussion environment for both students and lecturers. And it must provide all necessary features like document uploading function, class and member registration function, message posting function and etcetera.

Each registered course will have a classroom of its own, and all discussion groups will have their own discussion room. The system is open to registration for courses in high education institutes.

The model chosen for this system is waterfall with prototyping model. PBL Facilitator is to be developed using Macromedia Dreamweaver MX. The web language chosen for the scripting is PHP with additions of C++ and JavaScript, and MySql has been chosen for its database.

The system should be applicable and fulfil users' requirements. Users targeted for PBL Facilitators are students and teachers/lecturers of universities and colleges.

References

."About PHP: PHP - Hypertext Preprocessor ".

http://www.zend.com/zend/aboutphp.php. Last access date: March 28, 2004

Banta, Black, and Kline (2000) "PBL 2000 Plenary Address Offers Evidence

For and Against Problem-Based Learning." *PBL Insight*. **3(3)**.1-11

Also available at http://www.samford.edu/pbl/

Carliner, Saul. "An Overview of Online Learning."

http://www.lakewoodconferences.com/whitepaper1.htm March 20, 2004.

Department of Education, Tasmania. "What is Online Learning?"

http://www.discover.tased.edu.au/netlearn/support/onlinelearn.htm March 20, 2004

Gibbs, Steve. "Problem-based Learning with the Net."

http://proflearn.classroom.com/ProfDev/Conferences/Conferences/handouts/SEA03/ProblemBased_Gibbs.pdf. March 28, 2004

Graham, Ian. "Introduction to HTML."

http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/html_intro.html March 23, 2004

Jones, Diana. "What is PBL" http://edweb.sdsu.edu/clrit/learningtree/Ltree.html

March 20, 2004

LeJeune, Noel. "Learner-Centered Teaching Practices Online." http://ouray.cudenver.edu/~nflejeun/doctoralweb/_Portfolio3-Final/Products/synthesis_paper_lc_online.htm. April 3, 2004.

Littler, Steve. "JavaScript Defined." http://www.afn.org/~steve/javascri.htm March 20, 2004.

List of undergraduate institutions with faculty members using PBL: *Undergraduate* Institutions with Faculty Members Using PBL (2000). *PBL Insight*. **3(3)**.14

."Macromedia Dreamweaver." http://www.sonic.net/mnitepub/pccafe/reviews/macromedia_dreamweaver_mx/dreamweaver_mx.html . March 28, 2004

Nicholas Barau Eddy (2003/2004). *PBL IS*. Bc Thesis. University of Malaya.

Ommundsen, Peter. "Problem-based Learning in Biology *with 20 Case Examples*" http://www.saltspring.com/capewest/pbl.htm. May 4, 2004.

Ong Yu Kheng (2003/2004). *Online Meeting System*. Bc Thesis. University of Malaya.

PHP Builder Community Forums can be accessed at http://www.phpbuilder.com/board

Salimah M. and Zaitun A. B.(2003) "PBL Implementation: An Experience of the Faculty of Computer Science and Information Technology, University Malaya." *Journal of Problem-based Learning*. **2(1).** 1-3

Sellappan, P (2000) *Software Engineering: Management & Methods*. Petaling Jaya: Sejana Publishing.

Suresh Kaliadas (2003/2004). *E-Forum*. Bc Thesis. University of Malaya.

Sommerville, Ian. (2001) *Software Engineering*. 6th ed. Harlow: Pearson Education Limited.

WebmasterWorld can be accessed at http:// www.webmasterworld.com

Whitten, Jeffrey L., Bentley, Lonnie D. and Dittman, Kevin C.(2002) *System Analysis and Design Methods*. 5th ed. New York: Irwin/McGraw-Hill.

Wernick, Paul. "Life Cycle Process Models: Alternatives to the Waterfall Model 1: Prototyping and Iterative Approaches".

http://orawww.feis.herts.ac.uk/~2com0047/PL2%20Prototyp.pdf

Last access date: March 20, 2004

Definitions were looked up at: http://www.webopedia.com/TERM/

http://www.cnet.com/Resources/Info/Glossary/Terms/

http://searchwebservices.techtarget.com/sDefinition/

Bibliography

."About PHP: PHP - Hypertext Preprocessor ".

http://www.zend.com/zend/aboutphp.php. Last access date: March 28, 2004

Anaheim University, Australia: http://anaheim.edu/resources/classroom.shtml

Carliner, Saul. "An Overview of Online Learning."

http://www.lakewoodconferences.com/whitepaper1.htm March 20, 2004.

Problem-based Learning at Central Queensland University. See web page at

http://pbl.cqu.edu.au

Definitions were looked up at:

http://www.cnet.com/Resources/Info/Glossary/Terms/

http://www.webopedia.com/TERM/

http://searchwebservices.techtarget.com/sDefinition/

http://searchdatabase.techtarget.com/sDefinition/

Department of Education, Tasmania. "What is Online Learning?"

http://www.discover.tased.edu.au/netlearn/support/onlinelearn.htm March 20, 2004

E-Forums Community can be accesses at http://www.e-forums.net/forums/index.php

Gibbs, Steve. "Problem-based Learning with the Net."

http://proflearn.classroom.com/ProfDev/Conferences/Conferences/handouts/SEA03/ProblemBased_Gibbs.pdf. March 28, 2004

Graham, Ian. "Introduction to HTML."
http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/html_intro.html March 23, 2004

Jones, Diana. "What is PBL" http://edweb.sdsu.edu/clrit/learningtree/Ltree.html March 20, 2004

Samford University PBL Newsletter;
Kristi L. Arndt (2000) .*PBL Insight*. **3(3)**.1-16. Also available at http://www.samford.edu/pbl/

LeJeune, Noel. "Learner-Centered Teaching Practices Online."
http://ouray.cudenver.edu/~nflejeun/doctoralweb/_Portfolio3-Final/Products/synthesis_paper_lc_online.htm. April 3, 2004.

Littler, Steve. "JavaScript Defined." http://www.afn.org/~steve/javascri.htm March 20, 2004.

."Macromedia Dreamweaver."
http://www.sonic.net/mnitepub/pccafe/reviews/macromedia_dreamweaver_mx/dreamweaver_mx.html . March 28, 2004

Nicholas Barau Eddy (2003/2004). *PBL IS*. Bc Thesis. University of Malaya.

Ommundsen, Peter. "Problem-based Learning in Biology *with 20 Case Examples*" http://www.saltspring.com/capewest/pbl.htm. May 4, 2004.

Ong Yu Kheng (2003/2004). *Online Meeting System*. Bc Thesis. University of Malaya.

Lists of Online Resources for PBL can be viewed at: http://pbl.cqu.edu.au/content/online_resources.htm

PBL Portal- e-Board at: http://www2.eboard.com/eboard/

PHP Builder Community Forums can be accessed at http://www.phpbuilder.com/board

Information about PHP can be found at http://www.zend.com/zend/aboutphp.php

Problem Based Learning Network at **http://www.ascd.org**

Queen's University at Kingston, Canada: Problem-Based Learning at the Faculty of Medicine. Access at http://meds.queensu.ca/medicine/pbl/pblhom10.htm

Salimah M. and Zaitun A. B.(2003) "PBL Implementation: An Experience of the Faculty of Computer Science and Information Technology, University Malaya." *Journal of Problem-based Learning*. **2(1).** 1-3

Sellappan, P (2000) *Software Engineering: Management & Methods.* Petaling Jaya: Sejana Publishing.

Suresh Kaliadas (2003/2004). *E-Forum*. Bc Thesis. University of Malaya.

Sommerville, Ian. (2001) *Software Engineering.* 6th ed. Harlow: Pearson Education Limited.

WebmasterWorld can be accessed at http:// www.webmasterworld.com

Whitten, Jeffrey L., Bentley, Lonnie D. and Dittman, Kevin C.(2002) *System Analysis and Design Methods*. 5th ed. New York: Irwin/McGraw-Hill.

Wernick, Paul. "Life Cycle Process Models: Alternatives to the Waterfall Model 1: Prototyping and Iterative Approaches".
http://orawww.feis.herts.ac.uk/~2com0047/PL2%20Prototyp.pdf
Last access date: March 20, 2004

End of User Manual