# WebGE – WEB-BASED GROUPWARE FOR EDUCATION

**FOONG YEW HONG @ FOONG YEW YEW**

**FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY**
**UNIVERSITY OF MALAYA**
**KUALA LUMPUR**

**2004**

# WebGE – WEB-BASED GROUPWARE FOR EDUCATION

**FOONG YEW HONG @ FOONG YEW YEW**

**DISSERTATION SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE**

**FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR**

**DECEMBER 2004**

# Abstract

While group learning methods such as group assignments, forum and problem-based learning are increasingly being implemented in the education field, a variety of groupware tools such as email, online-forum and real-time chatting are used to replace or supplement the conventional face-to-face group learning processes. However, these tools are not particularly designed for education purpose and have some limitations. This study presents an integrated, full-fledge system, namely Web-based Groupware for Education (WebGE), for supporting the group learning processes. In this system, virtual groups are formed and information is shared within the group. Besides providing the commonly used groupware functions such as online discussion, real-time chatting, emailing, photos posting and files sharing, the course teachers also can use the specifically designed features such as assignment management, self-test, member progress, events management, polling, course material management and so forth. In simple words, WebGE is a complete online group learning system. WebGE was built based on the famous Unified Process, which uses object-oriented paradigm as the basis of software development. The implemented technologies are the state-of-the-art Microsoft ASP.Net with VB.Net and also the Microsoft SQL Server 2000. By using the latest technologies, the reliability and performance of WebGE can well be guaranteed.

# Acknowledgement

My first appreciation goes to my project supervisor, Associate Professor Pn. Raja Noor Ainon Zabariah Raja Zainal Abidin for guiding, supervising and helping me. I am grateful to have a supervisor who is friendly and patience during the project discussion.

I also want to express my gratitude to my family members who have always been supporting me during the course. Without their supports, I would not be able to complete this project on time.

Lastly, special thanks to my course mates and friends who have offered ideas and help in the project.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1 – Introduction

## 1.1 Overview

Working in group is something inevitable in our life. From a small group of friends until a big team in organisation, human always work in groups. In order to work in groups, usually, people gather together at a particular place and time. However, with the advent of computer and network technology, the style or method of working in groups has changed. Nowadays, people are able to work in groups despite the barrier of time and distance.

The software that facilitates people to work in group is called "Groupware". (Definition of groupware is further discussed in chapter two). With the use of groupware, traditional paperwork, such as scheduling, coordination of tasks and collaboration among members, are complemented or replaced with computer-based works, which is easier and more efficient. Realizing the benefits of groupware, more companies, especially the large scale team-driven companies, are going to implement or already have been implementing groupware technology for quite some time.

Nowadays, group learning methods such as problem-based learning, forum, group assignment is widely used and becoming increasingly important, particularly in tertiary education. In this case, groupware can be considered as a great tool for group learning as it enables people from different places, even different countries, to share their knowledge or information in an efficient and economic way.

In general, web-based system has the advantages such as ease of operation, accessible through any computer that is connected to Internet, easy to maintain and so forth. Also, groupware that operate through World Wide Web offer significant benefits (Wheeler, *et al.*, 1999). Therefore, it is suitable for web-based system to be used in education field, which the students are geographically dispersed. In this study, a web-based groupware for education (WebGE) is proposed.

## 1.2 Project Motivation

The driving force behind the decision to come out with the web-based groupware system is the successes and advantages of groupware in business world. The following shows several tangible benefits of company gains from the use of groupware (Opper and Fersko-Weiss, 1992).

- Reduce Face-to-face Meeting Frequency. Although people agree that meetings are essential in business, most will also freely admit that meetings waste a lot of time. Generally, at present, no electronic gathering, such as groupware, can 100% replaces the face-to-face meetings. However, with the use of groupware, people not always need to gather together since some of the problems can be solved through groupware. Opper and Fersko-Weiss (1992) pointed out that one out of three meetings can be eliminated by using groupware. Cutting down on meetings by this much saves not only the meeting time, but also travel time, and the disruption to work flow.

- Reduces Meeting Time. When face-to-face meeting is required, groupware facilitates the process of setting up the meeting. The conventional method might take days of phone calls and requires some luck on timing. With groupware, it might be done in minutes. Furthermore, during the meeting, some questions can be directed to continue the discussion electronically. Therefore, less meeting time with the use of groupware.
- Conduct Discussion Anywhere. This is especially important to the international company, where employees often travel abroad. With groupware, employees can obtain company's information from anywhere through internet. Also, the cost of communication is less expensive than phone-call or other methods.
- Curtailed Missed Communications. The most infamous form of missed communication is telephone tag. With the use of groupware, the messages sent by each other can be recorded, and thus, people can retrieve back the previous record for verification.
- Reduces the Physical Transfer of Information. With groupware, people can send documents to others in soft-copy. Thus, it is more efficient in terms of delivery time. Moreover, with the use of shared document tools, the information can be changed and molded in a more secure and efficient way.

Analogous to the business world, in education field, especially tertiary education, students are required to carry out meeting, projects, assignments, presentation, forum and discussion in groups. Therefore, it can be anticipated that groupware can greatly enhance or improve the learning processes in education

## 1.3 Problem Statement

As group learning is part of education, students are required and encouraged to work in groups (such as group study, discussion, project and so on). However, several problems arise when working in group as listed below:

1) The difficulties to gather all the group members for a group meeting – Since most students are geographically dispersed and have different free time, they might face difficulty to gather together.
2) Group meeting is always time-consuming and tedious – If more group members are contributing ideas and opinion in the group meeting, the meeting duration becomes longer. Furthermore, the members who are passive or inactive in the meeting might feel bored if the meeting time becomes too long.
3) Missed communication might occur – Missed communication is commonplace as people usually omit some words they hear during a conversation. Furthermore, communication through hand phone and telephone, where voice quality is worse than normal face-to-face conversation, has greater probability of missed-communication to occur.

To overcome the above problems, people can implement computer technology to support their group. Generally, email service is one of the great solutions, which has the advantages such as fast, low-cost, access anywhere through Internet, and so forth. Yet, it still has some inadequacies, such as the files and mail are only stored at the recipient (it might be deleted), the file being sent is not guaranteed to be received

(mailbox full or other reasons), not a real time conversation tool, and so forth. Therefore, people tend to integrate different tools, such as internet chatting tool, email, file transfer tool, scheduling tool and so on. Nevertheless, it requires some time to learn the operation of such tools, and this might hinder or deter people from using such tools. For these reasons, it is imaginable that an education sector needs to have its own groupware system, which is centralised, easy-to-operate, and incorporates most of the groupware features, for supporting the group learning process.

## 1.4 Objectives of Study

This study aims to develop a web-based groupware system that facilitates the group learning processes. This system is named as Web-based Groupware for Education (WebGE). It will provide an effective and efficient way for carrying out group coordination, collaboration and communication processes. Furthermore, it should enable people to use this facility through any computer that is connected to internet.

The objectives of this research are outlined as follows:

1) To investigate the use and requirements of groupware system in education field.
2) To design and build a web-based groupware system for education.
3) To provide the commonly used groupware functions, such as forum, files sharing, information sharing, scheduling, polling and so forth.

4) To provide the specific groupware functions for education, such as reports generation for members' visitation frequency; group page generation and deletion; and so forth.

The system should fulfil the above goals through the following sub-systems:

1) Group formation – The system should provide an easy and automated way in forming a particular group. This includes generating invitation email and sign-up form to the people who invited to join the group.
2) Group management – The system should empower the group founder in an easy administration of group specifications.
3) Data management – The posted data are properly stored. Furthermore, the members are able to easily access the group database over the Internet.
4) Administration – The system should provides administration package which is used by the system administrator for housekeeping purposes

## 1.5 Scopes of Study

Because groupware is a large field of study, which covers all computer-based system that supports working in group, some limitations of study are made so that it covers only the necessary parts of groupware for education. The following shows the scopes of this study.

1) The proposed system mainly is developed for education purpose. Hence, it does not cater for business and social needs although some of the supported features can be used for such purposes.
2) The system does not provide video conferencing feature because of the following reasons:
    a. Video conferencing requires high bandwidth network connection. Although the use of high band-width network is increasing, most of the students are still using dial-up network, which is not suitable for video conferencing.
    b. Video conferencing requires a video camera attaches to the computer. Analogous to the first reason, most of the students do not have such facility.
    c. Video conferencing is more suitable to be used in Local Area Network (LAN), which the data transferring rate is high enough for transferring voice and video data. However, this system is aimed for students who are geographically dispersed. Furthermore, it is not sensible that students will use the video conferencing to communicate, whereas they can meet face-to-face at somewhere inside the campus.
3) Voice conferencing function is also not incorporated in the system because of the following reasons.
    a. Although voice conferencing requires lower bandwidth network connection compared to video conferencing, it still consumes quite high data transmission rate compared to text-based communication.

b. The quality of voice conferencing through internet is even worse than telephone conferencing and most probably would lead to missed communication which might cause casualty to the group.

4) The system also does not support group writing or shared document editing. These tools should be developed as another system.
5) The system will be designed using Microsoft ASP.Net, which can be operated through IIS (Internet Information Server). To date, it can be installed on Microsoft Windows 2000, XP, and 2003.
6) The system can cater for common collaboration needs, which should incorporate the real-time text-based chatting, online polling, file sharing, photo sharing, and online forum.

## 1.6 Report Organization

This project report comprises eight chapters. It is structured as follows:

- Chapter 1: Introduction – Background information of the study.
- Chapter 2: Literature Review – Review on definition and classification of groupware; discuss on the group learning topic and also related terminologies; and analysis on existing system.
- Chapter 3: Methodology – Justification of the use of Unified Process as the software development process, and investigation on the use of unified process in this project.

- Chapter 4: System Analysis – Details of the requirements capture, which presented through use case diagrams; and the requirements analysis, which presented through collaboration diagram.
- Chapter 5: System Design – Details of the architecture developed, in the area of object-oriented design, data modelling, database design, and user interface.
- Chapter 6: System Implementation – The construction of the system into final system.
- Chapter 7: Testing – Details of various testing and debugging performed.
- Chapter 8: Discussion & Conclusion – Discussion and conclusion of the study

# Chapter 2 – Literature Review

## 2.1 Introduction

In this chapter, the literature part of this research study are presented and discussed. Primarily, the literature review in this chapter can be divided into three main parts as follows.

1) The general review on groupware, which includes groupware definition and groupware classification.
2) Reviews on the group learning and related terminologies.
3) Review on the existing groupware for education systems.

## 2.2 Definition of Groupware

> "'Groupware' is one of those mysterious and undefinable terms that have the ability to affect all our lives." (Coleman and Khanna, 1995)

Nowadays, the term of "Groupware" is used widely and loosely. In fact, "groupware" does not really have a specific definition for agreeable by all. In the past decade, many researchers have put efforts to define it and even there are some articles that

focus mainly on defining the "groupware" term (Allen, 1990; Finley, 1995; Johnsen-Lenz & Johnson-Lenz, 1998). In this session, various definitions of groupware which given by researchers are examined. However, this thesis would not focus mainly on defining the groupware term. Instead, much effort is put on investigating the use of groupware in education field.

The following shows several definitions for groupware.

- Computer-based systems that help two or more people working together are called groupware. (Ensor, 1990)
- Groupware is simply a tool that helps people work together more easily or more effectively. (Hills, 1997)
- Groupware is software for enabling collaboration within and between companies, and *computer supported cooperative work* (CSCW) is the term used by academics researching this area. (Chaffey, 1998)
- Groupware is software that supports the creation, flow, and tracking of non-structured information in direct support of collaborative group activity. (Orfali, *et al.*, 1999)
- Groupware is special software that allows members of a work team to coordinate their activities and communications around specific shared projects. (Coleman & Anderson, 2000)
- Groupware is computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment. (Ellis, *et al.*, 1991)

All of the above definitions define groupware as software or system that supports group. The forth definition addressed the term more specifically by adding the groupware activities: creation, flow and tracking of non-structured information. The fifth one added a keyword – project, which means groupware should be used with a specific goal and time. Finally, the sixth definition addressed the term as a computer-based system (instead of "software") and also explained that the users should have a common goal and a shared system interface. Apart from the abovementioned definitions, there are also other definitions coined by other researchers. Anyway, definition of groupware is not the main concern in this project but how groupware can be used for education purpose is more important.

In general, groupware is used to support the three "C"s – communications, collaboration, and coordination (Coleman & Anderson, 2000; Chaffey, 1998). Listed below are the definitions for three "C"s:

- Communication – to share information among the team members.
- Collaboration – to support and consolidate the team to work together.
- Coordination – to support coordination of each member role with each other.

## 2.3 Classification of groupware

Groupware is a broad term which covers lots of software that support group. Therefore, it is beneficial to classify these tools in order to have a better understanding about groupware. In this Section, two classification schemes, based on

Ellis, *et al.*, (1991), are presented. The first is based upon notions of time and space, and the second is based on application-level functionality.

### 2.3.1 Time and Space Classification Scheme

This classification scheme, which based on notions of time and space, is the best known classification scheme and has been presented in many articles (Johansen, 1989; Coleman & Khanna, 1995; Finley, 1995; Chaffey, 1998). Table 2.1 summarizes the four categories of groupware which make up by the time and space notion. Details and explanations of each category are presented in this Section.

**Table 2.1: Different uses of groupware classified in time and space**

| | ***Synchronous/Same Time*** | ***Asynchronous/Different Time*** |
|---|---|---|
| Same location | Same time, same place<br>Example: meeting support software | Different time, same place<br>Example: workflow systems |
| Different location | Same time, different place<br>Example: video-conferencing | Different time, different place<br>Example: e-mail and discussion groups |

(Source: Chaffey, 1998)

- Same Time/Same Place – When holding a conventional face-to-face meeting, groupware can be used as a supportive tool. For example, the game show response systems, such as the America's Funniest Home Videos. During this

game show, the studio audience used voting keypads which is installed at their chairs to vote on the best video (Finley, 1995). Another example is the group decision support systems, which can be used for brainstorming, issue analysis, prioritizing, policy formation, stakeholder identification and co-creation of document (Finley, 1995). Other Same Time/Same Place tools are such as low-cost copyboards, overhead projectors and specially designed team rooms. Further details about these tools can be obtained at Johansen (1989). There are many other tools in this category. Anyway, all of the groupware in this category serve one common purpose – to help teams to get the information in order.

- Same Time/Different Place – There are many times when face-to-face meetings are not possible to be held, particularly in this globalised world, where people often travel around. As mentioned in the first chapter, groupware can be used despite the distance barrier. Examples of groupware in this category are real-time video-conferencing, live text-based chatting boards, shared drawing pad and live voice-conferencing. Same time communication enables immediate feedbacks from team members. Thus, more information can be transmitted within a short period. However, live voice or video conferencing often transmits large amount of data, and thus requires higher band width of transmission medium (e.g. LAN, ADSL and so on).
- Different Time/Same Place – This category is most difficult to be understood (Johansen, 1989). In this category, "same place" is the key factor that needs to be discussed. Johansen (1989) addressed three different definitions for this key. The first is the exact same physical place, such as the public kiosk that can be used by different people at different times. The second is the same

metropolitan area, such as groupware systems which used by a team that is spread around on a campus location or in different offices in the same metropolitan area. The last definition for "same place" is a team room, assigned at the beginning of a team's life. Such rooms are often served as a focal point for the team. Inside the room, groupware tools are used for shared filing, coordination of tasks and information sharing.

- Different Time/Different Place – There are some times when team members are unable to get together in person and even cannot arrange for same time/different place meetings. In this case, groupware in this category are useful as a way for communication. For instance, email can be sent and received at anytime and anyplace. Another example is the online forum website, which enables message to be posted and viewed by all the team members. Other different time/different place tools are such as voice mail, group writing, and tools that support joint software development. All these tools can even be used for real-time communication since the messages are sent immediately. Anyway, it is not particularly designed to do so.

### 2.3.2 Application-level Classification Scheme

The second classification scheme is based on application-level functionality. Ellis, *et al.* (1991) listed six categories of groupware in this scheme. This classification scheme is not meant to be comprehensive and many of the defined categories are overlapped (Ellis, *et al.*, 1991). In this Section, six categories of groupware, namely Message Systems, Multiuser Editors, Group Decision Support Systems and Electronic

Meeting Rooms, Computer Conferencing, Intelligent Agents, and Coordination Systems, are explained.

1. Message Systems

   This category of groupware provides asynchronous exchange of textual messages between groups of users. Groupware which fall in this category often do not require high-end hardware and can be used at most of the computer. Examples of groupware in this category are email and online forum/discussion board. Sometimes, the system is incorporated with "intelligence" feature in order to mange the information overload problem. For example, the email spam filtering feature and also the email automatic classification features.

2. Multiuser Editors

   There are many cases in organisations that many people work together to create, review, and revise a single reports or presentations. In these cases, multiuser editors are very useful by providing the group writing features. Such tools can create a log file on tracking who and when the document is modified. Moreover, the system is also able to differentiate between the comments of various reviewers.

   Apart from that, some tools provide real-time group editing feature, which allow several people to edit the same object at the same time. In most cases, all the group members can concurrently read access to any segment of the object, but only one user can write to a particular segment of the object. The locking and synchronization of the segment in the object is managed by the editor transparently. Besides group writing, the same thing is also applied to

software development. This is crucial because software development is so complex that no programmer can create good software with just two hands and one mind (Yong, 2004).

3. Group Decision Support Systems and Electronic Meeting Rooms

   The main goal of this type of groupware is to improve the productivity of decision-making meetings, either by speeding up the decision-making process or by improving the quality of the resulting decisions (Ellis, *et al.*, 1991). In general, such tools are installed at every seat in the meeting room and it is used for brainstorming, stakeholder identification and issue analysis. In this case, large amount of cost is required since such system is specially designed for a particular room. Another example is the online voting tools, which enable users to vote online within a certain period.

4. Computer Conferencing

   As mentioned in the first classification scheme, computer conferencing enables people to communicate despite the distance barrier. Basically, computer conferencing can be divided into three approaches, namely real time computer conferencing, computer teleconferencing, and desktop conferencing (Ellis, *et al.*, 1991).

   The first approach (real time computer conferencing) only permits voice data to be transferred to each user. The application can either is built on top of an existing application or start from ground up. Therefore, the system has other features, such as sending text message or graphic data, together with the voice conferencing.

   Computer teleconferencing enables users to communicate through voice and video. However, the limitations are it requires special rooms and sometimes

trained operators. In most cases, the communication is only between two users. Furthermore, the users are unable to communicate through text and graphic messages.

Considering the limitations of the teleconferencing and real-time conferencing, desktop conferencing offers a better option by combining the advantages of both approaches while mitigating their drawbacks. Desktop conferencing offers voice and video capabilities, and also permits shared application to be run.

5. Intelligent Agents

   In an electronic meeting, the participants might consist of people and artificial participants. Such artificial participants are called "intelligent agents". Intelligent agents are rare to be found in most groupware due to its difficulties and complexity of development.

6. Coordination Systems

   Commonly, in a project, each member is assigned with different tasks at different time. In this case, coordination system provides an easy and convenient way to coordinate each member's roles. The system permits individuals to monitor each member's action, and also trigger users' actions by informing them through automatic reminders or alerts.

## 2.4 Group Learning

Group learning has long been implemented and is becoming increasingly important in education field. In tertiary education, group assignments, discussion and presentation

are almost inevitable in every subject of study. Furthermore, problem-based-learning (PBL) (another group learning method) is also becoming increasingly important in universities or colleges.

Indeed, group learning offers many advantages if compare to the conventional teaching method. In general, the reasons for using groups in education can be summarized as motivational, educational and ideological (Reynolds, 1994). Details of each abovementioned category are described below:

- Motivational – Group learning is motivational because people usually learn more and enjoy when they are involved and act as one of the teacher who contributes ideas. Besides, taking part in group learning involves not only the mind, values and feelings also brought to play. With deeper involvement in a group, people also become more memorable on the knowledge learned in the group. However, the limitation is the group members have to be active within the group.
- Educational – In conventional teaching method, students obtain knowledge mostly from the tutor. However, group learning enables students to become one of the teachers, who contribute their findings and ideas. Thus, students not only can obtain knowledge from their tutor, but from other students as well. Besides, group learning also offers other advantages, such as helping student to develop problem solving and communication skills, which are essential in the working world.
- Ideological – The process of collective enquiry in group learning also prepares people for a society based on democratic principles (Reynolds, 1994). This

method of learning helps develop individuals to be able to live and work in the social world.

This project is aims to develop a web-based groupware for education. In this case, the learning method must base on group. Therefore, it is essential to review and study on the various terminologies and methods which related to group learning in order to gain deeper understanding of group learning. In this Section, four topics, namely Collaborative Learning, Cooperative Learning, Problem-based-learning (PBL) and Computer Supported Collaborative Learning (CSCL), are presented. With such understanding, better product can be delivered to support the group learning processes.

### 2.4.1 Collaborative Learning

Collaborative Learning is an instruction method in which students work in groups towards a common academic goal (Gokhale, 1997). In Collaborative Learning, the students are responsible for one another's learning as well as their own. Thus, the success of one student helps other students to be successful. Some examples of Collaborative Learning are such as group discussion, forum, and so on.

Basically, Collaborative Learning is different from the conventional learning method, where most of the time students sit passively listen to the teacher's teaching. Tinzmann, *et al.* (1990) pointed out four general characteristics of Collaborative Learning. Brief description of each characteristic is presented as follows.

- Shared knowledge among teachers and students – In traditional classrooms, teacher always act as the information giver and knowledge flows only one way from teacher to student. In contrast, the students and teachers are the information giver in Collaborative Learning. In this case, the teacher has vital knowledge on the course content, skills, instruction and still provides information to students as usual. However, the teacher also value and build upon the knowledge, personal experiences, language, strategies and culture brought by the students to the learning situation.
- Shared authority among teachers and students – In Collaborative Learning, teachers share authority with students in very specific ways. For example, each student can express their findings and ideas in group discussion. Also, different opinions can be given within the group in order to find out the best answer. In this way, students are trained to be more critical-thinking.
- Teachers as mediators – In Collaborative Learning, the teacher's role is emphasizes more as a mediator. The teacher should help students to connect new information to their experiences or other field of knowledge, helps students figure out what to do when they are stumped, and directs them to the right way to obtain knowledge.
- Heterogeneous group of students – This is a critical characteristic of Collaborative Learning because this will ensure knowledge is more evenly shared among all students despite their ability, achievement or interests. Segregation would seriously weaken collaboration, and impoverishes the classroom by depriving all students' opportunities to learn from and with each other. Furthermore, heterogeneous groupings of students also enable students

to experience the real working world environment, where the group members consist of people from different background.

### 2.4.2 Cooperative Learning

Cooperative Learning is the instructional use of small groups so that students work together to maximize their own and each others' learning (Johnson, *et al.*, 1991). Both Cooperation Learning and Collaboration Learning are very similar and often used interchangeably. In general, Collaborative Learning is a broad and general term while Cooperative Learning is a more specific term and can be recognised as a specific type of Collaborative Learning. However, in fact, there are some differences between both terms.

Rockwood (1995) pointed some differences between Collaborative and Cooperative Learning. First, Cooperative Learning is the methodology of choice for foundational knowledge while Collaborative Learning is connected to the social constructionist's view that knowledge is a social construct. Second, in Cooperative Learning, the instructor is the centre of authority in the class, with group tasks usually more closed-ended and often having specific answers. In contrast, with Collaborative Learning, the instructor's authority and empowers are reduced, and the students are often given more open-ended, complex tasks.

### 2.4.3 Problem-based-learning (PBL)

Problem-based learning (PBL) is a way of constructing and teaching courses using problems as the stimulus and focus for student activity (Boud & Feletti, 1991). In general, PBL can be considered as an approach to learning where the problem comes first and the knowledge is developed as a consequence of trying to solve the problem. In this case, students work in small learning teams, and learning begins with a problem to be solved rather than content to be mastered. Furthermore, the PBL approach uses complex, real-world problems, and this motivates students to identify and research concepts and principles which they need to know in order to work through those problems.

Ultimately, PBL is used to train students to have the following abilities (Duch, *et al.*, 2001):

- Critical Thinking
- The ability to find, evaluate and use appropriate learning resources
- Cooperative working skills
- Both verbal and written communication skills
- Become continual learners

### 2.4.4 Computer Supported Collaborative Learning (CSCL)

Computer Supported Collaborative/Cooperative Learning, as its name implies, is Collaborative Learning using computer as the medium. CSCL is similar to CSCW (Computer Supported Cooperative Work). The differences between CSCW and CSCL

are as follows: CSCW tends to focus on communication techniques themselves whereas CSCL focuses on what is being communicated; CSCW is used mainly in the business setting whereas CSCL is used in the educational setting; the purpose of CSCW is to facilitate group communication and productivity whereas the purpose of CSCL is to scaffold or support students in learning together effectively (Hsiao, 1999).

If compare to the conventional Collaborative Learning, CSCL offers some advantages, such as students no need to always meet in person, learning and communication can be done at anywhere and anytime, all interaction records are logged for further reference, and so on. As more people realising the benefits of CSCL, CSCL is increasingly being implemented in the education field in order to support the conventional Collaborative Learning.

In most cases, the CSCL system is a web-based system, which implements the internet and World-Wide-Web technology. Since it is a web-based system, the students or users can access the system through a standard web browser at any computer which is connected to the internet. Generally, the system should supports four different types of users, which are (1) learner, (2) tutor, (3) author, and (4) administrator (Xue, *et al.*, 2001). The roles of each type of user are summarized as follows:

- Learner – access the registered course material, submits assignments, discuss problems with their tutor and classmates through chat rooms or bulletin board, coordinate group activities, and so on.

- Tutor – monitor and support the learner as they proceed to the course material; add additional course material; access, grade and add comments to the learner's works; and so on.
- Author – write and upload the course material to the corresponding course page (the process of authoring and uploading course material can be done easily and supported by the system), design the exercises with corresponding choices of answer, determine the grading scheme, and so on.
- Administrator – registration of new users; manage the learner, author, tutor and administrator database; install or remove course pages; and so on.

As conclusion, the CSCL is similar to the topic of this study – Web-based Groupware for Education. However, in most times, CSCL is referred to as a field of study whereas the Web-based Groupware for education is referred to as a tangible object or software.

## 2.5 Review on existing system

At present, there are quite a number of groupware for education system that has been developed. Some of these groupware are commercial systems, which are available in the market; but most of them are non-commercial, which are developed by certain universities. It is important to review these systems in order to capture the requirements for this project. In this Section, reviews on two groupware for education system, namely AulaNet and WebCT, are given.

As mentioned, the main objective of doing this review is to capture as many user requirements as possible from the existing groupware for education systems. These systems are real world working systems that have been tested and implemented for quite some times. This implies that most of the user requirements for this project can be obtained by reviewing these systems. However every system has its own limitations and weaknesses. Therefore, the second objective of this review is to make comparison among these systems, and comment on their strengths, weaknesses and limitations. With this information, improvement can be made for this project.

### 2.5.1 AulaNet

AulaNet is a non-commercial web-based groupware system, developed in the Software Engineering Lab (LES) of the Information Systems Department at the Catholic University of Rio de Janeiro (PUC-Rio), for administration, creation, maintenance and participation in distance learning courses (AulaNet Student Manual, 2004). Its development has been carried out since June 1997, with doctorate, master's degree and undergraduate students who maintain the code and improve it with topics from their research. To date, the newest version is AulaNet v2.0.

As mentioned, AulaNet is a web-based system, where the users can access it via any computer with internet browser installed and internet connection available. However, in order to fully access its contents, it is recommended to install the following plug-ins: Adobe Acrobat Reader, Microsoft PowerPoint Animation Player, Real Player, Macromedia Shockwave and QuickTime. A screenshot of the AulaNet interface is shown at Figure 2.1.


**Figure 2.1: A screenshot of AulaNet interface**

AulaNet is based on group learning, where users must share ideas (or communicate), be in tune with other participants of the groups (coordinate), and carry out tasks in a satisfactory manner (cooperate) (Fuks, *et al.*, 2002). Realising this principle, the AulaNet's services are divided into communication, coordination and cooperation services. The details of these services are summarized in Table 2.2, 2.3 and 2.4.

**Table 2.2: AulaNet Communication Services**

| Services | Sender-Receiver | Type | Description |
|---|---|---|---|
| Message to Teachers | one-teachers | asynchronous | The messages are sent through email to the instructors or coordinators, and are kept a copy in the system for subsequent consultation. |
| Discussion Group | one-group | asynchronous | It acts like a mailing list. The messages are sent through email to all members, and are kept a copy in the system. |
| Interest Group | one-group | asynchronous | A conferencing system or on-line forums which enable users to reply the posted topic. The messages are displayed in order. |
| Debate | one-group | synchronous | Real-time conversation through text chat. |
| Contacting Participants | one-anyone | synchronous | Lets members who are simultaneously connected to the system contact each other through messages that open up in new windows. |

**Table 2.3: AulaNet Coordination Services**

| Services | Description |
|---|---|
| Notices | ▪ Notification tool.<br>▪ Basic scheduling tools (calendar management). |
| Lesson Plan | ▪ A tool for the basic coordination on the flow of the course work.<br>▪ The teacher structures the course's educational content and separating them into classes. Then, upload it into this service page.<br>▪ It also allows learners to take private notes on a class that remain on file for their personal viewing, allowing them to save, for each content, doubts, observation, comments, pending tasks, etc. |
| Tasks | ▪ Assessment tools.<br>▪ It is used to assign work to learners.<br>▪ It manages task resolution file submissions and let the instructor make assessments and comments.<br>▪ Can be configured whether a learner's task resolution is visible to the others. |
| Follow-Up Reports | ▪ A tool for monitoring group participation.<br>▪ Learner participation can be quantify or qualify.<br>▪ The grade interval for asynchronous events are good, regular, weak and very bad; whereas for synchronous events are very active, active, low active and indifferent.<br>▪ It offer reports about average concept of learners, effective |

| Services | Description |
|---|---|
| | contributions, frequency of participation in debates, quantity of contributions per service and detailed information of each service. |

**Table 2.4: AulaNet Cooperation Services**

| Services | Description |
|---|---|
| Teacher Co-Authorship | ▪ Allow another teacher to assist the course teacher in the creation and maintenance of the course. |
| Learner Co-Authorship | ▪ Learner supply new contents, which need to be checked by the teacher. |
| Extra Contents | ▪ Such as references to textbooks (Bibliography) and Internet pages (Webliography). |

AulaNet has five principal user types, namely Administrator, Student, Coordinator, Co-author Teacher and Mediator. Details of each user type's role are shown as follows:

- Administrator – is the facilitator of the teacher/course/student integration, dealing with purely operational issues, such as the registration of teachers, admission of students in courses, etc.
- Student – is the course's end-user, representing the target audience to whom the course is designed.
- Coordinator – is the course's creator, participating in it since its initial description to the input of contents. You can choose whether or not the Coordinator will have the help of a co-author Teacher.

- Co-author Teacher (optional) – is responsible for helping the Coordinator in the creation and supply of educational contents to a course.
- Mediator – is the one responsible for giving the course.

(Source: AulaNet 2.0 Student's manual, 2004)

Strengths of AulaNet

- AulaNet supports three languages: Portuguese, English and Spanish.
- AulaNet offers most of the basic groupware features.
- AulaNet implements Access (*.mdb) file type of database, which is easier to be installed and backup. Furthermore, it is cost-saving because database server, such as Oracle or MS SQL Server, is not needed to be installed.
- AulaNet enables other teacher to assist the course teacher in creation and maintenances of the course.

Weaknesses of AulaNet

- AulaNet only can be installed on Microsoft operating system (e.g. MS Win2000, MS Win2003). Sun Solaris or Unix-based operating systems are not supported.
- AulaNet store data in Microsoft Access (*.mdb) format. However, Microsoft Access is not suitable to be used to store large amount of data because the performance begins to breaks down for large databases, and for many simultaneous users. Furthermore, using Microsoft Access as server-side databases for even small databases will yield poor performance, since it does not have a server component, which means the client application always has to read the entire table from the file server (Kauffman, *et al.*, 2002).

- The remote control panel of the system is awkward to be used because it might sometimes obstruct the viewing of the main contents.
- Some features, such as shared drawing pad and online voting tools, are not provided.
- Students are not allowed to form their own group. This is inflexible and limits the students to interact with their course mates only.

### 2.5.2 WebCT

WebCT (World-Wide-Web Course Tool) developed by the University of British Columbia is a web based course management system that designed to deliver online learning. WebCT development was started by Murray W. Goldberg in the year of 1995 (Goldberg, *et al.*, 1996). Initially, it had been distributed without any cost. However, with the set up of WebCT Education Technologies Corporation in the year of 1997, the WebCT became commercialize thereafter (Goldberg, *et al.*, 2000). At present, WebCT is a well-known web-based learning tool and is presently being used by over 2,000 universities and colleges all over the world.

Although the "groupware" term is not mentioned in the WebCT website, its idea of development is based on groupware, which the students join a virtual community and share knowledge within the community (WebCT, 2004). Basically, a community is created for each course and the students are not allowed to create their own community in WebCT. Apart from that, courses materials are published at the course community webpage to supplement the existing courses.

Similar to AulaNet, WebCT provides a login page for security purpose. With the appropriate username and password, the user can access to the course material. A basic course page includes a welcome message and the links to the activities that the instructor has chosen for that class. The interface of a basic course page of WebCT is shown at Figure 2.2.

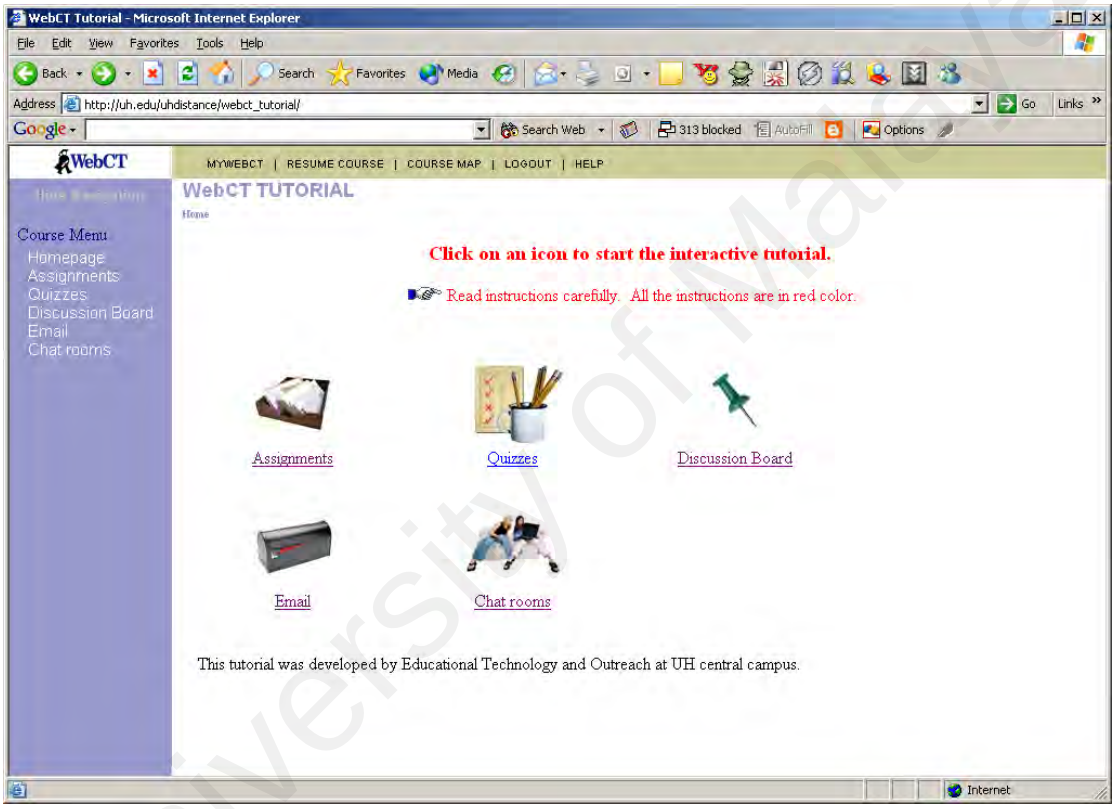


**Figure 2.2: Screenshot of WebCT's basic course page.**

To support group learning, WebCT provides many groupware features. Most of them are similar to other groupware learning system, such as AulaNet. The features that are built in the WebCT are given in Table 2.5.

**Table 2.5: WebCT's features**

| Features | Descriptions |
|---|---|
| Chat Room | ▪ With the Chat tool, members of a class can have real-time, text-based conversations (similar, but less sophisticated than other chat tools such as MSN messenger or ICQ).<br>▪ Each WebCT course web site includes 4 chat rooms which can be used for different discussions or topics.<br>▪ When someone enters a chat room, their WebCT user id appears on a list viewable to those in the room. New messages are displayed at the bottom of the message list and scroll up and then off the screen as other messages are entered.<br>▪ All messages are stored in chat logs located in the Manage Files tool.<br>▪ Instructors get the best use out of the tool when chats are scheduled, structured around a topic, and moderated. |
| Discussion board | ▪ The Discussions tool allows students and teachers to communicate by posting, reading and responding to messages to an online bulletin board.<br>▪ The tool contains three discussion threads (named: All, Main, and Notes) open to all the students enrolled in the class, but teachers can customize |

| Features | Descriptions |
|---|---|
| | discussions by:<br>o adding custom topics pertinent to their course material<br>o restricting access to student subgroups<br>o allowing anonymous or non-anonymous postings<br>▪ Messages can be displayed in the chronological order (unthreaded) or indented beneath the message they are in response to (threaded).<br>▪ A handy search tool gives users the power to search for messages by user id, name, subject, etc.; it even lists messages which contain specified words or phrases unique to the body of a message. |
| Mail | ▪ It is similar to normal email system, but there are some limitations.<br>▪ With real e-mail, you can send to just about anyone with an Internet connection, but WebCT's mail can only be used to send messages between members of the class the WebCT site has been set for. |
| Calendar tool | ▪ The Calendar is like a daily planner, telling you about course events.<br>▪ Calendar entries include a heading, a start and end time, a summary and even a link to a URL.<br>▪ The designer can make public entries (for all course |

| Features | Descriptions |
|---|---|
| | participants to view) and both the designer and students can make private entries (entries only visible to the person who makes the entry). |
| White board | ▪ The Whiteboard is a simple graphics tool that people can use simultaneously from different computers.<br>▪ When one person "draws" on the Whiteboard, anyone who is using it from another computer will be able to see and manipulate the image as soon as it is finished being drawn. It's like a chat room that uses images instead words. |
| Student Presentations | ▪ With the Student Presentations tool, a group of students can be given access to an area in the WebCT course to upload and view group files. |
| Course Material | ▪ The contents are course syllabus, assignments, and course materials. The course material might include text, complex equations, images, video, and audio. |
| Glossary | ▪ Provides a place to put course-related jargon, terms, even pictures, in an alphabetically-ordered format. |
| Online bibliographies | ▪ Provides a place to put links to other sites. |
| Assignments | ▪ The Assignments tool allows instructors to specify assignment details, provide related files to be downloaded and set up online "drop boxes" for assignments so students can submit them electronically. |

| Features | Descriptions |
|---|---|
| Grades | ▪ Students use this tool to securely access their grades that instructors have entered into WebCT's Student Management grade book tool. |
| Online quizzes/test | ▪ where students can take quizzes and test online and also get an immediate feedback and scoring or it can be graded individually by the faculty |
| Progress | ▪ Details of what services have been used by students and how often they have used it. |
| Change password | ▪ A place where the user can change the password |

There are four classes of user in WebCT, namely Administrator, Designer, Markers and Students. When signing onto a course, WebCT provides a different view of the course depending on the class of user who signing in. The designer and marker views are supersets of the student view. Details of each class of user are listed below:

- Administrator – There is only one administrator account. This person does not sign on to any individual course, but instead signs on to the WebCT administration page. Once the administrator can initialize and delete courses, and change the passwords of course designers. This person does not actually configure or add any content to a course, but simply initializes a course and hands over the new empty course to a designer. The administrator username cannot be changed. However, the administrator can change his/her own password.

- Designer – Each course has one designer account. Normally, the designer is the instructor of the course. The designer can manipulate the course in any way, create quizzes, alter grades, check student progress, define student presentation groups, manipulate student accounts, and so on. The designer account name cannot be changed. The designer password can be changed by the administrator.
- Markers – Each course can have any number of markers. A marker has the same privileges as the student, but can also grade quizzes and manipulate student grades. The course designer creates the marker accounts.
- Students – Each course can have any number of students. Students cannot manipulate the course content (other than in the student presentation areas as defined by the designer). Students can change their own password at the discretion of the designer. The course designer creates the students accounts.

Strengths of WebCT

- WebCT is undeniably an excellent web-based distance learning system. Except some of the minor and value-added features, WebCT contains all the necessary groupware for education functions.
- WebCT provides greater flexibility by enabling teacher to determine the layout or interface of the course page. Besides, teacher also can determine the functions availability for each of the course page.
- WebCT offers appealing and yet easy-to-navigate graphical user interfaces.
- WebCT supports 12 languages: Arabic, Simplified Chinese, Dutch, Finnish, French, German, Italian, Japanese, Portuguese, Spanish, Swedish, and UK English. Traditional Chinese, Farsi, Galician, and Catalan.

- WebCT can be installed at most of the platforms, including MS Win2000, Sun Solaris and Unix.

Weaknesses of WebCT

- WebCT does not allow students to form their own group. Each community is created for one course. This is inflexible and limits students to interact with their course mates only.
- The use of Oracle as database server is costly.
- Online voting tool is not provided.

### 2.5.3 Blackboard

Similar to WebCT and AulaNet, Blackboard is web-based application which used to make and host course web sites. Nonetheless, unlike WebCT and AulaNet, Blackboard was initialised with commercial purpose.

Blackboard was developed by Blackboard Inc. which founded in the year of 1997 (Blackboard, 2004). Since the first release of Blackboard, the Blackboard continued to evolve and gradually became famous in a variety of institutions. Nowadays, it is used by many institutions around the world and often being compared with WebCT as an on-line course management solution.

Blackboard encourage interaction between students and faculty, and as well as among students themselves. Similar to WebCT and AulaNet, the interactions among the students is done via the set up of a course community. All students who registered the same course will interact at the course web site which designed by the course lecturer.

In order to facilitate group learning, blackboard course web site are incorporated with groupware tools, such as email, discussion boards, chat rooms, calendar, document sharing and so forth.

The interface of Blackboard is more of less the same as WebCT, which there are login page, courses selection page and also a few links to the groupware tools. The major difference is the Blackboard provides the search tool which is quite useful. Figure 2.3 shows the screenshot of the Blackboard's course main page.

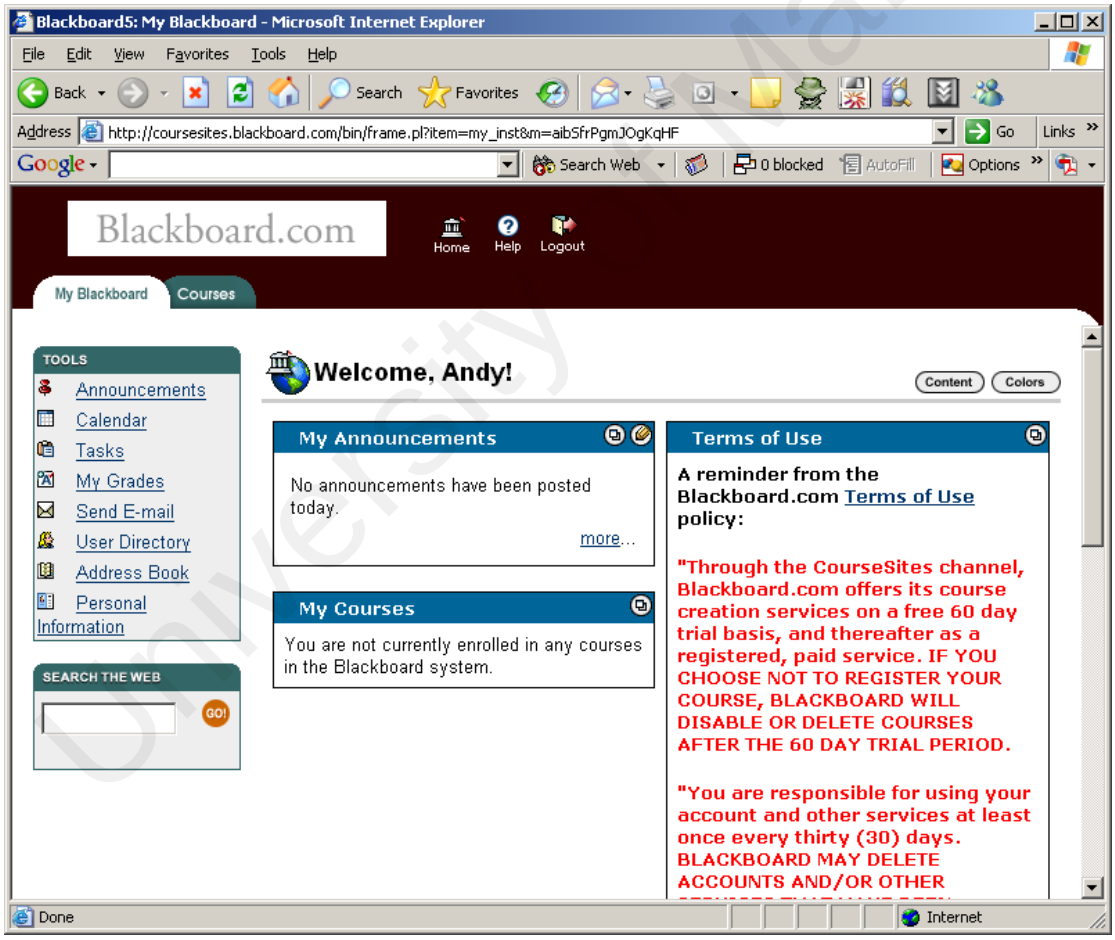


**Figure 2.3: Screenshot of the Blackboard's course main page**

The features provided by Blackboard are the same as the WebCT. However, some of the naming of these features is different. For example, the document sharing tool is

called “Drop Box” in Blackboard, whereas the WebCT named it as “Student Presentations”. Please refer to section 2.5.4 for details on WebCT features.

Similar to WebCT and AulaNet, each user is assigned to a specific user role which determines the authorisation for accessing the system features. There are live main user roles in Blackboard, namely Instructor, Teaching Assistant, Course Builder, Grader and Student. These roles are course-specific, which means a single user can be assigned the Instructor role in one course community, but be assigned the Student role in another course community. Descriptions of each user role are listed below:

- Instructor – This is the highest level of privilege within a course. Typically, the course lecturer is assigned to this user role and able to access everything within the course pages.
- Teaching Assistant – As its name imply, the teaching assistant role is normally assigned to someone who will be assisting the course instructor. Typically, this would be the course assistant. The teaching assistant user role has most of the instructor user role’s privilege except for certain user management capabilities (For example, the ability to remove an instructor from the course).
- Course Builder - The Course Builder role has access to the course's Control Panel, but only to the Content Areas section and the User Management section. An Instructor would assign someone the Course Builder role so that person could assist the Instructor in the creation of course content and some of the course management. As with the Teaching Assistant role, access within the User Management section is limited to prevent the Course Builder from removing Instructors, modifying user properties, and so on.

- Grader - The Grader role has access to the course's Control Panel, but only to the Assessment section. A Grader would assist the Instructor in the creation, management, delivery, and grading of online assessments delivered through Blackboard.
- Student - The Student role is the default course role. A user with the Course User role of Student has no access to the course's Control Panel.

Strengths of Blackboard

- The system contains all the necessary groupware for education functions.
- The system can run on Microsoft, Linux and Sun Solaris platform.
- Unlike WebCT, which supports only Oracle database, Blackboard provides options of using either MS SQL Server or Oracle as the database server.
- Blackboard supports many languages. The system interface can be changed to different language by applying the corresponding plug-in.
- The interface is intuitive and user friendly.

Weaknesses of Blackboard

- Similar to WebCT, the system does not allow students to form their own group.
- Online voting tool is not provided.

### 2.5.4 Summary of the Existing System

After the review on the existing groupware for education systems, this section focus on providing the summary of these three systems and the descriptions on the differences of WebGE from the reviewed systems.

From the review of these three existing systems, it can be seen that the WebCT and Blackboard outrange AulaNet. Nonetheless, the AulaNet has some features which are not available at WebCT and Blackboard. However, there is some trade-off for having such features. For example, the AulaNet stores the data in Access file, which is free and easy to be installed. On the flip side, the use of Access file for storing data will yields very poor performance.

The WebCT and Blackboard are identical in terms of features and user interface. However, there are some differences in term of system requirements. In this case, the Blackboard outrange than WebCT by supporting both Oracle and MS SQL Server as the database server.

After the reviews of the existing systems, WebGE's requirements were generated based on the aims to consolidate as much as possible the strengths and features of these three existing systems. Therefore, all the necessary groupware functions were included as the requirements. Apart from that, it also should minimize as much as possible the weaknesses. For instance, WebGE provides some additional features such as online voting tool and student group formation. Mote details regarding the strengths of WebGE can be obtained at section 8.1 - System Strengths. Nonetheless, due to the time constraints, some strengths of the existing system, such as the ability to support different languages, were not included as the WebGE's requirement.

## 2.6 Chapter Summary

This chapter has described the literature review conducted in this study prior to the development of the system, which forms the foundation of the study. This includes the review on the groupware definition and classification, and also some description about group learning. Lastly, three existing systems, namely AulaNet, WebCT and Blackboard, were examined.

# Chapter 3 – Methodology

## 3.1 Introduction

The WebGE project development is based on Unified Process (UP) framework, which is a famous object oriented framework. However, some modifications were made to the original Unified Process framework in order to suit this project. Further explanations of the changes made are given in Section 3.9 – Applying the Unified Process.

In this chapter, most of the contents explained the Unified Process in general. The contents in this chapter are the overview, four phrases and five workflows of the unified process; the Unified Modeling Language (UML); and the Rational Rose Modeling tool. After the explanations of Unified Process, the last part of this chapter is about how the Unified Process was tailored and applied to this project.

## 3.2 Overview of the Unified Process (UP)

The Unified Process is a software development process. More specifically, it is a generic process framework, which can be specialized for a very large class of software systems, for different application areas, different types of organizations, different competence levels, and different project sizes (Jacobson, *et al.*, 1999).

Nowadays, the Unified Process has been widely used in many software development projects. In fact, the Unified Process is more practical and complete if compared to the traditional software development process models, such as waterfall model and V-model. Such traditional models normally only specify the phases which involved in a software development process and the interactions among these phrases. On the other hand, the Unified Process shows its completeness by including the four phases, five workflows, the interactions of these phases and workflows, deliverables of each workflow and the visual modeling technique (UML). Figure 3.1 illustrates the development cycle for the Unified Process.

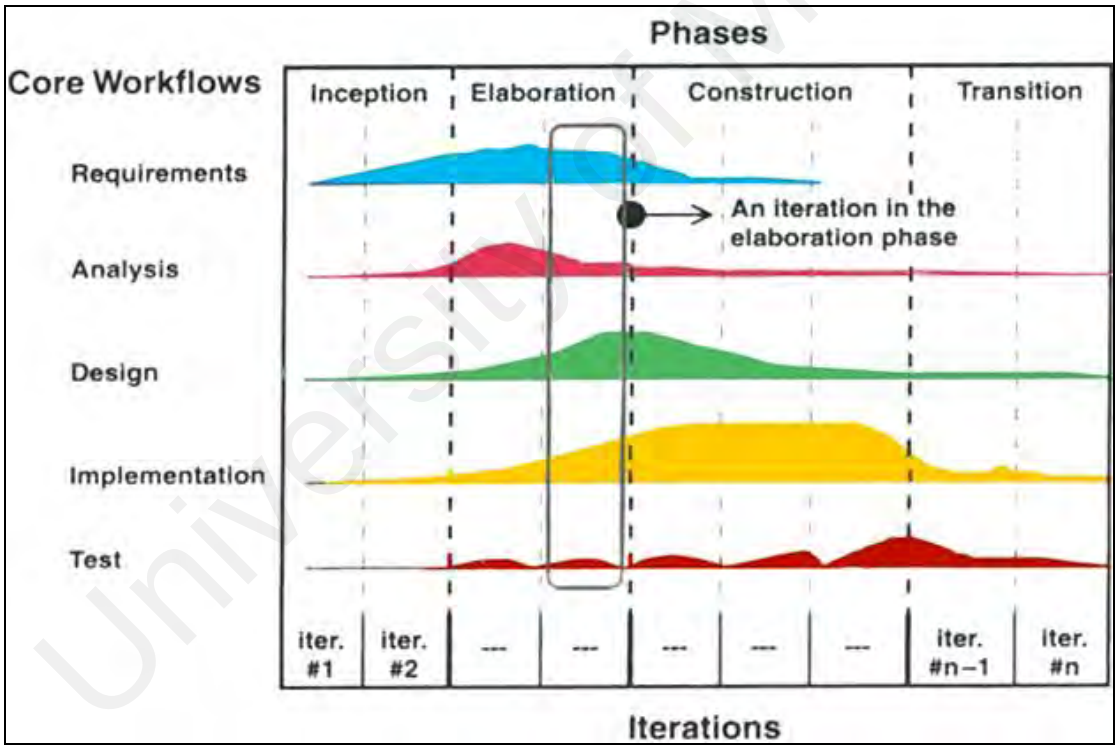


**Figure 3.1: The Unified Process Development Cycle (Jacobson, *et al.*, 1999)**

As shown in Figure 3.1, the phases in Unified Process are inception, elaboration, construction, and transition; whereas the workflows are requirements, analysis, design, implementation and test. Typically, a project has to go through all the four

phases. Each phase is consists of many iterations, which a typical iteration goes through all the five workflows as shown for an iteration in the elaboration phase in Figure 3.1. All the five workflows are iterated until the project comes to a certain major milestone of the phase.

Recall that a typical iteration goes through all the five workflows. However, the workloads of each workflow within an iteration are not even. For example, the first iteration in the inception phase has more workloads on the requirements phase, and most probably will not involve the implementation and test workflow. The approximate extent to which the workflows are carried out in each phase is shown by the curves in Figure 3.1.

As mentioned, the iterations within each phase stops in a major milestone. By specifying a major milestone for each phase, the iterations of workflows within each phase can be done with a clear and realistic objective (this objective is known as the minor milestone), and hence these iterations can be planned in a systematic way. Besides, the major milestone enables managers to make crucial decisions before work proceeds into the next phase and also to monitor the progress of work in each of the four phases.

### 3.2.1 Characteristics of the Unified Process

The Unified Process has three important characteristics, namely (1) use case driven; (2) architecture-centric; and (3) iterative and incremental. These three characteristics are equally important. It is essential to study these characteristics in order to understand the Unified Process. This section explains these characteristics in general.

### 3.2.2 The Unified Process Is Use Case Driven

In order to understand this characteristic, two primary terms, namely actor and use case, need to be addressed first. Basically, a use case is a sequence of actions, performed by one or more actors (people or non-human entities of the system) and by the system itself, that produces one or more results of value to one or more of the actors (Scott, 2002). An example of use case is a human uses ATM machine to withdraw money. In this case, the person who uses the ATM Machine is the actor. Apart from that, there are other use cases in this scenario, such as check account balance, transfer account money and so forth. In short, use cases are used to capture all the functional requirements of a system, and the grouping of many use cases forms the use case model.

One of the key aspects of the unified Process is its use of use cases as a driving force for all development work, from initial gathering and negotiation of requirements through code (Scott, 2002). Furthermore, all of the workflows of a project development are initiated from the use cases. Figure 3.2 depicts the dependencies of the use case model with the other models – analysis, design, deployment, implementation and test model.

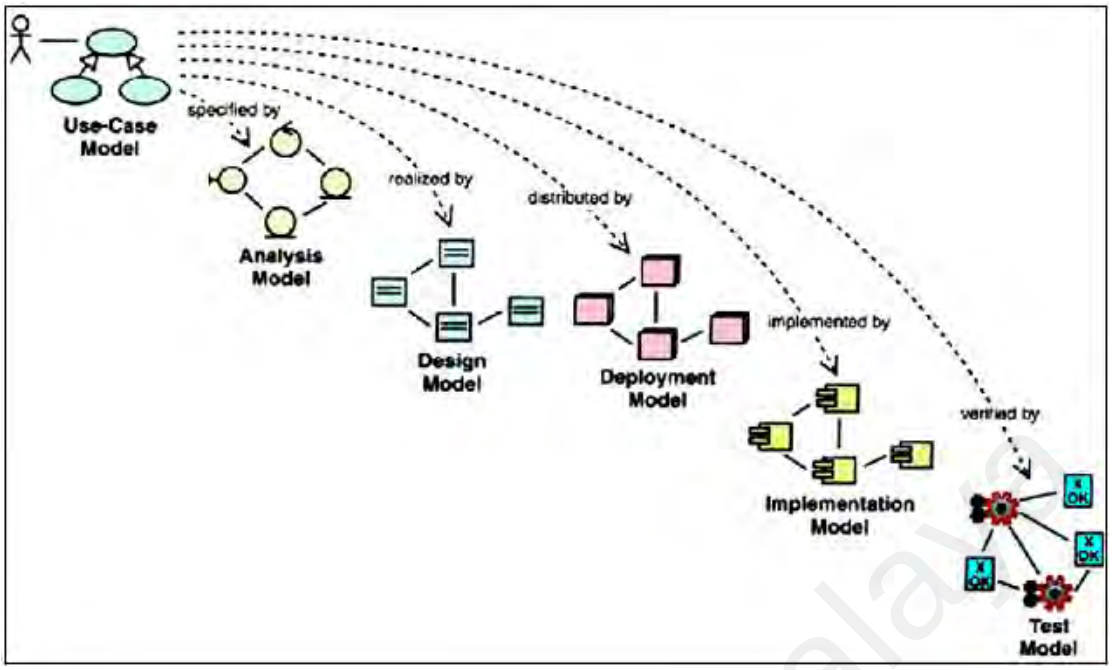


**Figure 3.2: The dependencies between the use-case model and the other models (Jacobson, *et al.*, 1999)**

### 3.2.3 The Unified Process Is Architecture-Centric

Architecture is a view of the whole design with the important characteristics made more visible by leaving details aside (Jacobsen, *et al.*, 1999). The Unified Process is architecture-centric, and provides several advantages as listed below (Scott, 2002):

- Understand the big picture – The architecture descriptions facilitate the understanding of the system being built. Thus, the development progress is monitored and will not go astray.
- Organizing the development effort – The use of architecture patterns help shape the development effort on various levels. By using this aspect of architecture effectively, the project team can increase the chances that communication across sub-teams.

- Evolving the system – The architecture offers a set of essential reference points on which future development work can rely. Thus enhances team members' ability to evolve the system effectively and efficiently.
- Guiding the use cases – As mentioned, the Unified process is use-case driven, and it drives the architecture of a software system. However, on the other hand, the architecture also guides the selections and exploration of the use cases. With this mutual relation, both the architecture and the use cases evolve in parallel.

### 3.2.4 The Unified Process Is Iterative and Incremental

Nowadays, the demand for high performing and sophisticated software is increasing. Software development, particularly the commercial software, usually requires several months to possibly a year or more to be completed. In this case, the development work is usually divided into smaller slices or mini-projects. Each mini-project is an iteration, which results in a version of the system that will be release internally or externally. Each new version of release, which is the product resulted from the new iteration, is suppose to offer incremental improvement over the previous version.

Similar to the second characteristics, the iterative and incremental characteristic of the Unified Process also offers several advantages, as listed below (Scott, 2002):

- Logical progress toward a robust architecture – As mentioned above, each iteration results incremental improvement over the previous version. This also means that the architecture is evolving to become more robust. Furthermore, the iterative and incremental characteristic of Unified Process also enables the

team to make necessary major changes early at considerably less cost than they would inflict later in the project.

- Dealing with ongoing changes in requirements – The Unified Process does not require the full requirements be specified first. Basically, each iteration produces a small part of the whole system. With the experience obtained from each iteration, the project team and the customers can negotiate requirements on an ongoing basis, thus reducing the risk associated with trying to specify all of the requirements up front.
- Greater flexibility to change the plan – The success or failure of each iteration can be used as a guide to decide whether the project plan should be changed.
- Continuous Integration – The Unified Process divides the project into many mini-projects, and each mini-project produces a part of the whole system. This enables the progress of the project to be measured toward specific goals, rather than toward more abstract and general requirements. Furthermore, the problems of the whole system are also isolated, and can be addressed in ways that don't disrupt the integrity of the whole working system.
- Early Understanding – The iteration of development also enables the project developers to gain experiences from each iteration. Besides, the impacts of the mistakes made by team member are reduced since those mistakes are isolated.
- Ongoing focus on risk – Since the project is divided into many iterations, the risks to the project can be addressed earlier. In this way, appropriate actions can be taken, and thus mitigate the risks to the greatest extent possible.

## 3.3 The Four Phases of the Unified Process

As mentioned in Section 3.2, there are four phases, namely inception, elaboration, construction and transition, in the Unified Process. Within each phase, there is a major milestone, which used to indicate the completion of the phase. It is crucial to understand each of these phases in order to define the major-milestones and also to monitor the project progress. This Section gives a brief description of each phase, which includes the descriptions of its goal and tasks.

### 3.3.1 Inception

The primary goal of the inception phase is make the business case for the viability of the proposed system. In other words, the outcome of this phase should be something viable and agreed by all the major stakeholders so that the project can continues to the next stage. Basically, the following tasks are performed during the inception phase:

- Define the scope of the system.
- Outline candidate architecture.
- Identify critical risks and address the strategies to overcome or minimize it.
- Based on the initial estimates of available resources, start to make the business case.

The inception phase ends when it reaches its major milestone, which called Life-Cycle Objectives. The indications that the milestone has been achieved are:

- The scope of the proposed system is agreed by all the major stakeholders.
- A set of critical high-level requirements are addressed in the candidate architecture.

- The business case for the project is viable and agreed by the stakeholders for continued development.

### 3.3.2 Elaboration

The main focus of the elaboration phase is to formulate the architecture baseline. The architecture baseline is the system architecture which contains the expanded versions of the six models initialized during the inception phase. The following shows the tasks performed in elaboration phase:

- Capture majority of the remaining functional requirements.
- Expand the candidate architecture into a full architecture baseline.
- Address significant risks.
- Provides sufficient details to guide the next phase of the project.

The major milestone of the elaboration phase is called Lifecycle Architecture. The following shows the indications that the project has reached this milestone:

- Most of the functional requirements (approximately 80%) have been captured in the use case model.
- The architecture baseline has been produced.
- The project is ready to continue to the next phase.

### 3.3.3 Construction

The main goal of the construction phase is to build a system capable for initial operation in the user environment. The produced system in this phase can be considered as "beta release". In this phase, the system is built iteratively and incrementally, until a workable beta version of the system is produced. The produced

system should cover all the identified use cases. However, it may contain some defeats or bugs which will be fixed at the next phase. The completion of the construction phase is indicated by the achievement of the major milestone which called Initial Operational Capability.

### 3.3.4 Transition

The main goal of transition phase is to roll out the fully functional system to customers. The main task of this phase is to correct the defects and problems of the previously produced system. Besides, some modifications also can be made in order to improve the system. After performing all these tasks, the final release version of the system is produced at the end. The major milestone in this phase is called Product Release.

## 3.4 The Five Workflows of the Unified Process

As mentioned in Section 3.2, the five workflows, namely requirements, analysis, design, implementation and test, cut across the set of four phases. Basically, these work flows are performed iteratively in every phases. It is important to understand the details of each workflow in order to perform it correctly. The following sub-session give a brief explanation of these workflows.

### 3.4.1 Requirements

The primary goal of this workflow is to build the use case model, which means to capture the system functional requirements (Please refer to Section 3.5 for explanation

about use case model). Since the Unified Process is iterative, the requirements are capture through the project cycle. However, different phases require different level of requirements to be captured. The following explains the extent to which the requirements workflow is carried out in each phases (Jacobson, *et al.*, 1999).

- During the inception phase, the use cases are identified in order to delimit the system and scope the project. Approximately less than 10% of requirements are captured in this phase.
- During the elaboration phase, the remaining requirements are captured so that the size of the development effort can be estimated. Approximately 80% of requirements are captured in this phase, and about 5 to 10% of those requirements should be implemented into the architecture baseline.
- Along with the system development process in construction phase, the remaining requirements are captured.
- Almost no requirements are captured in this phase except the changing requirements requested by the customers.

### 3.4.2 Analysis

The analysis workflow is aimed at building the analysis model (Please refer to Section 3.5 for explanation about analysis model). The activities involved in this workflow are refining and structuring the requirements which captured from the requirements workflow.

Similar to the requirements workflow, the workload of this workflow is different at each phase, as explained below:

- During the inception phase, critical high level requirements are described and the analysis workflow starts to increase.
- During the elaboration phase, much effort is put on the requirements and elaboration workflow. The majority of analysis model is created in this phase.
- During the construction phase, the analysis model is normally completed.
- During the transition phase, the analysis model is fine tuned if necessary.

### 3.4.3 Design

The primary goal of the design workflow is to produce the design model (Please refer to Section 3.5 for explanation about design model). The activities of design workflow involve describing the physical realization of use cases. The outcome of this activity is the design model, which serves as an abstraction of the system's implementation.

The different extents to which the design workflow is carried out in each phase are described below:

- During the inception phase, the design model is roughly drafted as an effort to realize critical high level requirements.
- During the elaboration phase, the significant use cases are addressed within the design model. The deployment model might also starts evolving during the elaboration phase if the system to be developed has a high level of physical distribution.
- During the construction phase, majority of the design model and deployment model are built.
- During the transition phase, the design model and deployment model are fine tuned if necessary.

### 3.4.4 Implementation

The aim of the implementation workflow is to build the implementation model (Please refer to Section 3.5 for explanation about implementation model). In this workflow, how the elements of the design model are packaged into software components, such as source code files and dynamic link library (DLL), are described.

The different extent to which the implementation workflow is carried out in each phase is described below:

- During the inception phase, the implementation model (if exists) normally takes the form of an executable prototype.
- During the elaboration phase, the implementation model addresses the architecturally significant use cases.
- During the construction phase, the implementation model is normally completed.
- During the transition phase, the implementation model is fine tuned.

### 3.4.5 Test

The test workflow is aims to produce the test model (Please refer to Section 3.5 for explanation about test model). In this workflow, how integration and system tests will exercise executable components from the implementation model is described. Besides, how the team will perform the integration, system and unit test is described as well.

The different extents to which the test workflow is carried out in each phase are described as follows:

- Normally, there are no test model exists during the inception phase. However, there are some rare cases, which the test planning focuses on executable prototype.
- During the elaboration phase, the test model is concerned with the architecturally significant use cases.
- During the construction phase, majority of the test model which is concerned with unit, integration and system testing is built.
- During the transition phase, the test model is fine tuned while ongoing testing activities are carried put to detect flaws and errors.

## 3.5 The Primary Models of the Unified Process

A model is an abstraction of a system from a particular viewpoint. Different models provide advantage of allow people to view the system differently at various viewpoints. Furthermore, the model also able to hide the details of which some people not interested. For example, most end users only interested to the functions provided by the system but not interested to the development process or technical details of the system. In this case, the use case model suits their needs.

There are six primary models, namely use case, analysis, design, deployment, implementation and test model, in Unified Process. Each of these models has different role and also provides different viewpoint to the system. The following sub-Sections present a brief description on these models. Further details about these models can be obtained from the book – *The Unified Software Development Process* by Ivar

Jacobson, Grady Booch and James Rumbaugh (Please refer the references Section in this dissertation).

### 3.5.1 Use Case Model

The following describe the use case model:

- The use case model consists of use cases and actors. Each use case in use case model defines the interaction between the actors and the system, and also the task that can be performed by the system.
- The use case model records the system functional requirements.
- It is built during the requirements workflow.
- Most of the use cases are constructed during the inception and elaboration phases.
- The use case model is easy to be understood by most people and serve as a communication medium among the customer, users, and system developers.
- The use case model also can serve as an agreed contract and as a guideline for the client to validate the system functionalities and also for the developer to build the expected system.

### 3.5.2 Analysis Model

The following describe the analysis model:

- The analysis model describes and structures the use cases.
- The analysis model provides a rough sketch of the system, which will be further refined in design model.
- It is structured by stereotypical classes and packages.

- The analysis model consists of class and interaction diagrams. If necessary, it might contain activity and statechart diagrams.
- The analysis model is built during the analysis workflow.

### 3.5.3 Design Model

The following describe the design model:

- The design model describes the physical realization of use cases by focusing on how the functional and non-functional requirements, together with other constraints related to the implementation environment, impact the system under consideration (Jacobson, *et al.*, 1999)
- The design model serves as the blueprint of the implementation.
- The design model consists of a set of collaborations of model elements that provide the behaviour of the system.
- The design model takes more efforts to be developed, approximate 5:1 ration to analysis model (Jacobson, *et al.*, 1999)
- The diagrams involved in the analysis model are class, interaction and activity diagrams. However, the class and interaction diagrams in design model provide more details than the similar diagrams in analysis model.
- The design model is built during the design workflow.

### 3.5.4 Deployment Model

The following describe the deployment model:

- The deployment model describes the physical distribution of the system in terms of computational nodes. Each node represents a computational resource, often a processor or similar hardware device (Jacobson, *et al.*, 1999).

- The deployment model depicts the mapping between the software architecture and the hardware architecture.
- The deployment model is composed of one or many deployment diagrams.
- The deployment model is built during the implementation workflow.

### 3.5.5 Implementation Model

The following describe the implementation model:

- The implementation model describes how elements in the design model, such as design class, are implemented in terms of components such as source code files, executables, and so on (Jacobson, *et al.*, 1999). For example, a class in design view can be implemented as a physical file with extension ".vb" in Microsoft Visual Basic.Net.
- The implementation model is composed of one or more component diagrams.
- The implementation model is built during the implementation workflow.

### 3.5.6 Test Model

The test model primarily describes two things – 1) what will be tested; and 2) how it will be tested. The test model is composed of a collection of test cases, test procedures, and test components. The following explains the test case, test procedure, and test component in general.

- Test case – A document which specifies test objective, test data, execution conditions, and expected test results. Test cases can be derived from use cases, design documents, or the software code. A test case can be implemented by one or more test procedures.

- Test Procedure – A set of detailed instructions for the setup, execution, and evaluation of test results for test cases. One or more test cases are implemented by a test procedure.
- Test component – It is used to automate one or several test procedures. It can be developed using a scripting language or a programming language.

## 3.6 The Unified Modeling Language (UML) Diagrams

The Unified Modeling Language (UML) is a standard modeling language which used in describing and designing software system. The UML is useful by enables the software developers to visualize, specify and document the structure and behaviour of the system. Basically, the UML is open standard (controlled by the Object Management Group) and has a very rich graphical notations collection which used to model all or most aspects of the system. However, it also allows users to extend the language itself by using the three constructs of stereotypes, tagged values and constraints.

In the Unified Process, the UML graphical notations are used to construct all the diagrams. Then, one or more diagrams combine to form a model. For example, the use case model is composed of a set of use case diagrams. The UML diagrams involved in the Unified Process are Use Case Diagram, Interaction Diagram (composed of the Sequence diagram and the Collaboration diagram), statechart diagram, deployment diagram and component diagram. These UML diagrams are described briefly in the following text. Further information regarding these diagrams

can be found from the book – UML distilled: A brief guide to the standard object modeling language, authored by Cris Kobryn, Grady Booch, Ivar Jacobson and Jim Rumbaugh (Please refer the references Section in this dissertation).

### 3.6.1 Use Case Diagram

The use case diagram contains use cases and actors, and also the association between them. It is the first diagram that needed to be produced before other diagrams can be drawn. The use case diagram serves as a communication medium between the customer and the system developer. Therefore, it should be written in plain English, which can be understood by non-technical people (the client or end user).

The use cases can be created in different level of details. In general, the less-detail one is called business use case, which mainly focuses on business processes.

### 3.6.2 Collaboration Diagram

The collaboration diagram, or communication diagram in UML 2.0, is interaction diagram which describes the data links between the various participations interaction. In collaboration diagram, the participants are freely placed and the participants are connected through lines drawn among them. Furthermore, the sequence of messages is shown through numbering.

### 3.6.3 Sequence Diagram

The sequence diagram is the most commonly used interaction diagram, which focuses on the time ordering of messages between objects. Typically, it captures the behaviour

of a single scenario or use case. In sequence diagram, the objects or participants are placed horizontally whereas the sequence of messages is shown in vertical direction.

Both sequence diagram and collaboration diagram provides same amount of information. In fact, there are some case tools which allow the conversion between both diagrams. The decision of which interaction diagram to be used is actually based on the system analyst's preference. On the whole, most people prefer sequence diagram (Kobryn, *et al.*, 2003).

### 3.6.4 Class Diagram

The class diagram shows classes of the system and various relationships that exist among them. It depicts the static view of the system. There are basically two types of class diagram, namely Analysis Class Diagram and Design Class Diagram. In general, the design class diagram provides more detail than the analysis class diagram. Besides the ability to show the relationships among the classes, the class diagram also serves as the database model when the system entities are mapped into tables in relational database.

### 3.6.5 Activity Diagram

The activity diagram is used to describe the work flow, business process and procedural logic. The activity diagram contains many nodes, which each node represent a single action to be performed. The nodes in activity diagram are connected. There are other elements, such as fork, join, decision and merge, exist in activity diagram in order to show the parallel actions and the conditional behaviour.

The usage of activity diagram is fairly unrestricted. It can be used to represent flow of activities in a major business process or it can also be used to represent the algorithm in a class's method implementation.

### 3.6.6 Statechart Diagram

The statechart diagram, or called as state diagram, documents the sequence of states that a class goes through during its lifetime and the events that cause a state transition. For instance, a member object in the library may have a few states during its lifetime, such as 'created', 'suspended', 'renewed' and 'expelled'. Besides, the events that trigger the state transition are also shown in statechart diagram.

### 3.6.7 Deployment Diagram

The deployment diagram describes the components or hardware used in the system and the relationships between these hardware components. It basically depicts the network architecture of the system distribution. For example, a web-based system may have the web server, application server and database server.

### 3.6.8 Component Diagram

The component diagram shows a set of components and the dependencies between them. A component is a physical and replaceable part of a system that offers reusable services. Examples of component are such as executable (EXE) file, dynamic link library (DLL) file and database table.

## 3.7 Rational Rose Modeling Tool

In this study, Rational Rose Modeling tool was used to produce all of the UML diagrams, which discussed in Section 3.7. Rational Rose is a powerful case tool developed by Rational Software Corporation. It is well-known and has been widely used in the software development industry. The following shows some of the key features of Rational Rose:

- It supports team development by offers a single united tool, language and notation for the whole software development team.
- It provides a complete set of UML notations and stereotypes, which can be used to build all the UML diagrams.
- It has a single easy-to-navigate Integrated Development Environment (IDE) for all the UML models. The list of available models is presented in a tree form at the left of the screen while the contents of each model are presented in a spacious workspace at the centre of the screen.
- The reporting utility is provided in order to generate a textual documentation based on the UML models which have been built. The report is generated as a Microsoft Word document.
- It enables traceability between requirements and other model elements.
- It supports for round trip engineering, which enables code generation from model – forward engineering; and model generation from code – reverse engineering.
- It provides a model checking utility that checks for errors and inconsistency in all the models that have been created.

## 3.8 Why Unified Process was Chosen?

The reasons why the Unified Process was chosen as the software development process framework for this project are shown below:

- It is a complete process framework that provides every piece of knowledge which is needed for software development. For example, it specifies how iterations are done throughout the development process and also what are the artefacts and deliverables that need to be produced in each phase and workflow. Since all this information is packaged into a single framework, the ability and trustworthiness of the Unified Process in software development should never be a question.
- The Unified Process comes along with a highly rated Unified Modeling Language (UML). The richness of notations in UML means that every model in software development can be constructed using a single language. Using a single modeling language for analysis, design, implementation, and testing guarantees a clearer and traceable set of models (Bahrami, 1999).
- The Unified Process encourages use of visual modeling, especially in Unified Modeling Language (UML), to visualize, specify, construct and document software systems. The models created serve as the software blueprint and provide a common understanding within the development team and with other stakeholders.
- The UML is open standard and provides highly-descriptive notations, which are easy to understand and able to model most system requirements and specifications. Besides providing the standard notations, UML allows software

developer to extend the language by introducing stereotypes, constraints and tagged values.

- The Unified Process is component-based, which provides the advantages such as components reusability and data abstraction.
- Modern software development is iterative in nature because of the uncertainty factor. Changes to user requirements or mistakes in the development process simply mean that the software development process will have to be iterated. The Unified Process is iterative and incremental in nature and thus it is very suitable for developing software.
- There are lots of UML modeling tool in the market, such as Rational Rose, SmartDraw, MagicDraw and so on. Therefore, the user can choose the most suited modeling tool to be used according to the features of modeling tool and the user's budget.

## 3.9 Applying the Unified Process

The Unified Process framework was applied in this project. Mainly, there are six iterations – one iteration in the inception phase; two iterations in the elaboration phase; two iterations in the construction phase; and the final iteration in the transition phase. In team development, each team member's roles or responsibilities has to be specified in order to ensure the development work is organized in a systematic manner. However, since this project is handled solely by the author, the specification of which person handles a certain task is not given.

Table 3.1 shows the completion level of each of the five workflows in each iteration of phase for this project. The percentage figure denotes amount of work accomplished for a particular workflow during a particular iteration in a phase. A precise look at the table shows that how the work is distributed among the four phases and the five workflows is a bit different from what is actually planned in the Unified Process (please refer to Figure 3.1).

Table 3.2 - 3.7 show the work breakdown structure of this project, and the deliverables of each workflow of iteration. Finally, the deliverables or artefacts that collected at the end of each workflow are given in Table 3.8. All these deliverables are also presented in the coming chapters. For the purpose of this writing, the deliverables for the requirements workflow are presented together with deliverables for the analysis workflow.

**Table 3.1: Completion level of each workflow after each iteration**

| | **Inception** | **Elaboration** | | **Construction** | | **Transition** |
|---|---|---|---|---|---|---|
| ***Iteration*** | #1 | #1 | #2 | #1 | #2 | #1 |
| **Requirements** | 15% | 75% | 80% | 100% | 100% | 100% |
| **Analysis** | 10% | 60% | 80% | 100% | 100% | 100% |
| **Design** | 5% | 25% | 50% | 100% | 100% | 100% |
| **Implementation** | 0% | 0% | 10% | 60% | 95% | 100% |
| **Test** | 0% | 0% | 15% | 40% | 60% | 100% |

**Table 3.2: Work breakdown structure during the inception phase**

| Workflow | | Inception (single iteration) |
|---|---|---|
| Requirements | Activities | ▪ Define system scope and objectives.<br>▪ Plan project schedule.<br>▪ Identify the most critical use cases with actors.<br>▪ Identify any non-functional requirements that are specific to each of the identified use case.<br>▪ Identify all generic non-functional requirements, such as network architecture, software and hardware requirements, etc. |
| | Deliverables | ▪ Initial use case model with most critical use cases.<br>▪ Initial non-functional requirements description for these use cases.<br>▪ Specify hardware and software requirement constraints.<br>▪ Project scope, objectives and schedule |

| Workflow | | Inception (single iteration) |
|---|---|---|
| Analysis | Activities | ▪ Construct the collaboration diagram to show the realization of use cases that have been identified during the requirements workflow.<br>▪ Construct the class diagram to show classes and their relationships that participate in the realization of these use cases. |
| | Deliverables | ▪ Initial analysis model. |
| Design | Activities | ▪ Minimal design work is done because the architecture of the system is not revealed yet. Plot a rough sketch of design.<br>▪ Construct the initial high-level version of the deployment model. |
| | Deliverables | ▪ Initial design model.<br>▪ Initial deployment model. |
| Implementation | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Test | Activities | ▪ No work is done here |
| | Deliverables | ▪ No deliverables here. |

| Workflow | | Inception (single iteration) |
|---|---|---|
| | Summary of deliverables for the inception phase | ▪ Project scope and objectives.<br>▪ Project schedule.<br>▪ Candidate architecture which comprises the initial versions of the six models.<br>▪ Initial non-functional requirements description. |

**Table 3.3: Work breakdown structure during the first iteration of the elaboration phase**

| Workflow | | Elaboration (first iteration) |
|---|---|---|
| Requirements | Activities | ▪ Identify other important use cases besides those that have been identified during the inception phase.<br>▪ Create user interface prototypes to see whether there are important use cases that are left out.<br>▪ The total use cases that have been identified should be about 80%.<br>▪ Identify any non-functional requirements that are specific to each of the identified use case.<br>▪ Restructure the use case model. Some of the use cases that have been identified earlier might have to be split up.<br>▪ Reprioritize the use cases. This is important to produce the architecture of the system. |
| | Deliverables | ▪ 80% completed use case model.<br>▪ Non-functional requirements description.<br>▪ User interface prototypes. |

| Workflow | | Elaboration (first iteration) |
|---|---|---|
| Analysis | Activities | ▪ Analyze all the identified use cases in great detail. Build the analysis class diagram and collaboration diagram.<br>▪ Determine precisely the responsibilities of each analysis class. |
| | Deliverables | ▪ 60% completed analysis model. |
| Design | Activities | ▪ Design classes (add in attributes and operations).<br>▪ Design the realizations for important use cases only (based on the priority). |
| | Deliverables | ▪ 25% completed design model. |
| Implementation | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Test | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |

**Table 3.4: Work breakdown structure during the second iteration of the elaboration phase**

| Workflow | | Elaboration (second iteration) |
|---|---|---|
| Requirements | Activities | ▪ Refine the use case model that has been built so far (if necessary). |
| | Deliverables | ▪ 80% completed refined use case model. |

| Workflow | | Elaboration (second iteration) |
|---|---|---|
| Analysis | Activities | ▪ Refine the analysis model that has been built so far. |
| | Deliverables | ▪ 80% completed analysis model. |
| Design | Activities | ▪ Refine the design model that has been built so far. |
| | Deliverables | ▪ 50% completed design model. |
| Implementation | Activities | ▪ Implement design classes in terms of file components.<br>▪ Group the file components according to subsystems.<br>▪ Build an initial deployment model.<br>▪ Implement (involves coding) what that has been designed so far. |
| | Deliverables | ▪ 20% completed implementation model.<br>▪ 20% completed deployment model. |
| Test | Activities | ▪ Plan test cases and test procedures for components and subsystems that have been identified in this iteration.<br>▪ Perform the component and integration testing on what have been implemented in this iteration. |
| | Deliverables | ▪ 15% completed test model. |

| Workflow | | Elaboration (second iteration) |
|---|---|---|
| | Summary of deliverables for the elaboration phase | ▪ 80% completed use case model.<br>▪ 80% completed analysis model.<br>▪ 50% completed design model.<br>▪ 20% completed deployment model.<br>▪ 20% completed implementation model.<br>▪ 15% completed test model.<br>▪ User interface prototypes. |

**Table 3.5: Work breakdown structure during the first iteration of the construction phase**

| Workflow | | Construction (first iteration) |
|---|---|---|
| Requirements | Activities | ▪ Capture remaining use cases that are required but not really important.<br>▪ Identify any use case specific non-functional requirements.<br>▪ Restructure the use case model if necessary. |
| | Deliverables | ▪ Complete use case model. |
| Analysis | Activities | ▪ Analyze all the remaining use cases. |
| | Deliverables | ▪ Complete analysis model. |

| Workflow | | Construction (first iteration) |
|---|---|---|
| Design | Activities | ▪ Design the realization of all use cases that have not been realized yet.<br>▪ Make sure that all classes have attributes and operations that can be mapped easily into implementation.<br>▪ Design database schema.<br>▪ Design user interfaces. |
| | Deliverables | ▪ Complete design model.<br>▪ Complete database schema design.<br>▪ Complete user interfaces design. |
| Implementation | Activities | ▪ Map all the design classes into components in component diagram.<br>▪ Complete the deployment model.<br>▪ Implement up to 60% of design model. |
| | Deliverables | ▪ Complete implementation model.<br>▪ Complete deployment model.<br>▪ 60% of source codes. |
| Test | Activities | ▪ Plan test cases and test procedures for what that have been implemented in this iteration.<br>▪ Perform the component and integration testing on what have been implemented in this iteration. |

| Workflow | | Construction (first iteration) |
|---|---|---|
| | Deliverables | ▪ 50% completed test model. |

**Table 3.6: Work breakdown structure during the second iteration of the construction phase**

| Workflow | | Construction (second iteration) |
|---|---|---|
| Requirements | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Analysis | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Design | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Implementation | Activities | ▪ Implement the remaining items in design model. |
| | Deliverables | ▪ Complete source codes. |
| Test | Activities | ▪ Plan test cases and test procedures for what that have been implemented in this iteration.<br>▪ Perform the component and integration testing on what have been implemented so far. |
| | Deliverables | ▪ 80% completed test model. |

| Workflow | | Construction (second iteration) |
|---|---|---|
| | Summary of deliverables for the construction phase | ▪ Complete use case model.<br>▪ Complete analysis model.<br>▪ Complete design model.<br>▪ Complete implementation model.<br>▪ Deployment model with deployment diagram.<br>▪ 80% completed test model.<br>▪ Complete database schema design.<br>▪ Complete user interfaces design.<br>▪ Complete set of source codes. |

**Table 3.7: Work breakdown structure during the transition phase**

| Workflow | | Transition (single iteration) |
|---|---|---|
| Requirements | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Analysis | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Design | Activities | ▪ No work is done here. |
| | Deliverables | ▪ No deliverables here. |
| Implementation | Activities | ▪ Install the beta release<br>▪ If errors found during test, repair the source codes. |

| Workflow | | Transition (single iteration) |
|---|---|---|
| | Deliverables | ▪ Full product release. |
| Test | Activities | ▪ Plan test cases and test procedures for the whole complete system<br>▪ Perform the component, integration and system testing on the whole system. |
| | Deliverables | ▪ Complete test model |
| | Summary of deliverables for the construction phase | ▪ Complete test model.<br>▪ Full product release. |

**Table 3.8: Summary of deliverables of each workflow**

| Workflow | Deliverables |
|---|---|
| Requirements | ▪ Project scope, objectives and schedule.<br>▪ Use case model with use case diagrams and non-functional requirements description.<br>▪ User interface prototypes (temporary). |
| Analysis | ▪ Analysis model with class diagrams and collaboration diagrams. |
| Design | ▪ Design model with relevant diagrams.<br>▪ Database design. |

| Workflow | Deliverables |
| --- | --- |
| | ▪ User interfaces design. |
| Implementation | ▪ Deployment model with deployment diagram.<br>▪ Implementation model with component diagram.<br>▪ Source codes, database implementation and other relevant component files. |
| Test | ▪ Test model with test cases and test procedures. |

# Chapter 4 – System Analysis

## 4.1 Introduction

System analysis involves investigating a system to fully understand the problems and requirements of the system. This corresponds to earlier iterations in Elaboration phase of Unified Process.

This chapter is composed of two main topics – Requirements Capture and Requirements Analysis. The Requirements Capture topic documents all the system requirements of this project while the realization of the functional requirements is presented in the Requirement Analysis topic.

## 4.2 Requirements Capture

There are two types of requirements, namely functional requirements and non-functional requirements.

In general, functional requirements specify the actions which the software must perform whereas non-functional requirements specify properties which the software must exhibit. Both the functional and non-functional requirements are specified in this section.

Before the specific requirement to be captured, the higher level of system requirements are identified as follows:

- WebGE is based on groupware approach. It permits two types of group, namely Course Group and Student Group, to be formed.
- The Course Group is a more formal type of group, which contains more functions that used to support an education course. Furthermore, only the administrator has the authority to create Course Group.
- The Student Group, on the other hand, is less formal, contains less functionality and can be created by any members of the WebGE system.

Subsequently, three main user groups or actors for WebGE, namely Administrator, Course Group Users and Student Group Users, are identified. Both Course Group Users and Student Group Users are further divided into three types of users: Course Group users are consists of Instructor, Assistant, and Students; whereas Student Group users are consists of Founder, Manager, and Members. All the actors for WebGE and its corresponding description are summarized in Table 4.1 and 4.2.

**Table 4.1: Actors for WebGE**

| 1) Administrator | 2) Course Group Users | 3) Student Group Users |
|---|---|---|
| - | Instructor | Founder |
| - | Assistant | Manager |
| - | Students | Members |

**Table 4.2: Description of each Actor**

| Actor | Description |
|---|---|
| **Administrator** | The system administrator for WebGE, who mainly responsible to initialize and remove Course Groups. |
| **Instructor** | The Course Group Instructor. In most cases, the Course Instructor is the lecturer of that particular course. |
| **Assistant** | The course assistant is responsible for helping the Course Instructor in the creation and supply of educational contents to a Course Group. |
| **Students** | The student is the Course Group's end users, representing the target audience to whom the Course Group is designed. |
| **Founder** | The founder of the Student Group, who has the most privilege in the Student Group. |
| **Manager** | The manager of the Student Group is assigned by the course founder and has some privilege over the Student Group Member. The Student Group Manager is responsible for helping the founder in administration of the Student Group. |
| **Member** | The Student Group Member has the lowest privilege in the Student Group and serves as the Student Group's end users. |

Figure 4.1 depicts the interaction between the WebGE actors and the main subsystem in the WebGE.

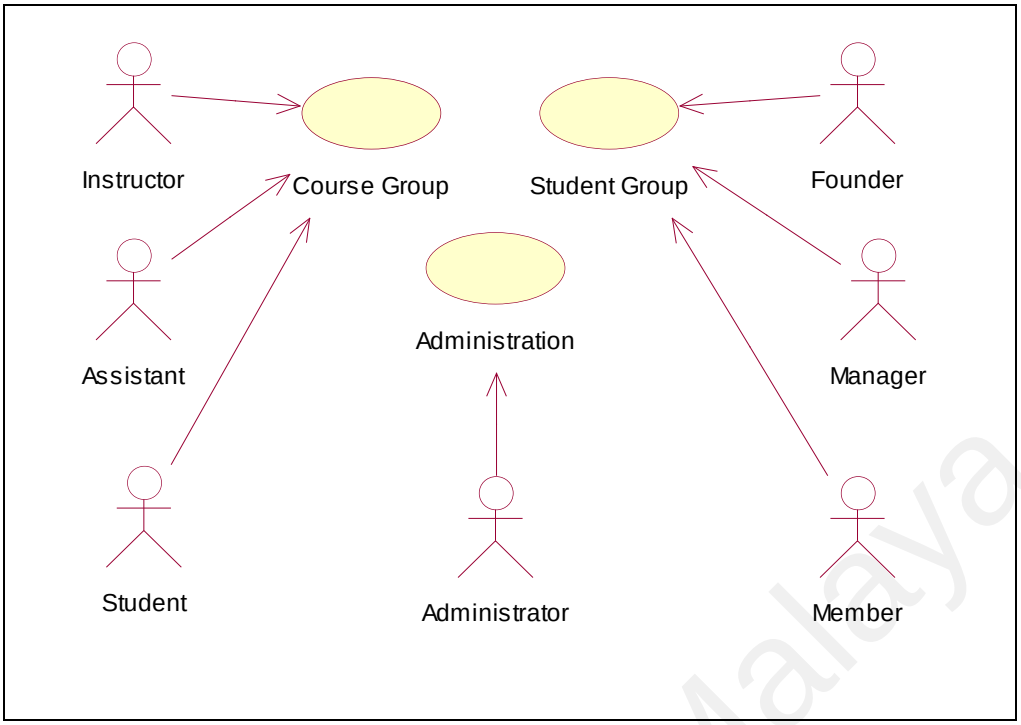


**Figure 4.1: WebGE top level use case diagram**

### 4.2.1 Functional Requirements

The functional requirements of this project are modeled using the UML use case diagram. Since there are plenty of functional requirements, they are grouped into sixteen different use case packages as listed below:

1. Admin
2. Announcement
3. Assignment
4. Chat
5. Discussion
6. Event
7. File
8. Group
9. Group Registration

10. Material
11. Member
12. Noteboard
13. Photo
14. Poll
15. Reference
16. Self-test

All the use cases of the WebGE are shown in the following use case diagrams, according to the use case packages. In order to reduce the diagram complexities, some actors who interact with the same use cases are combined and displayed as an actor (e.g. instructor/assistant/student). Apart from that, these diagrams are self-explanatory but descriptions are provided for some of the use cases which need to be explained.

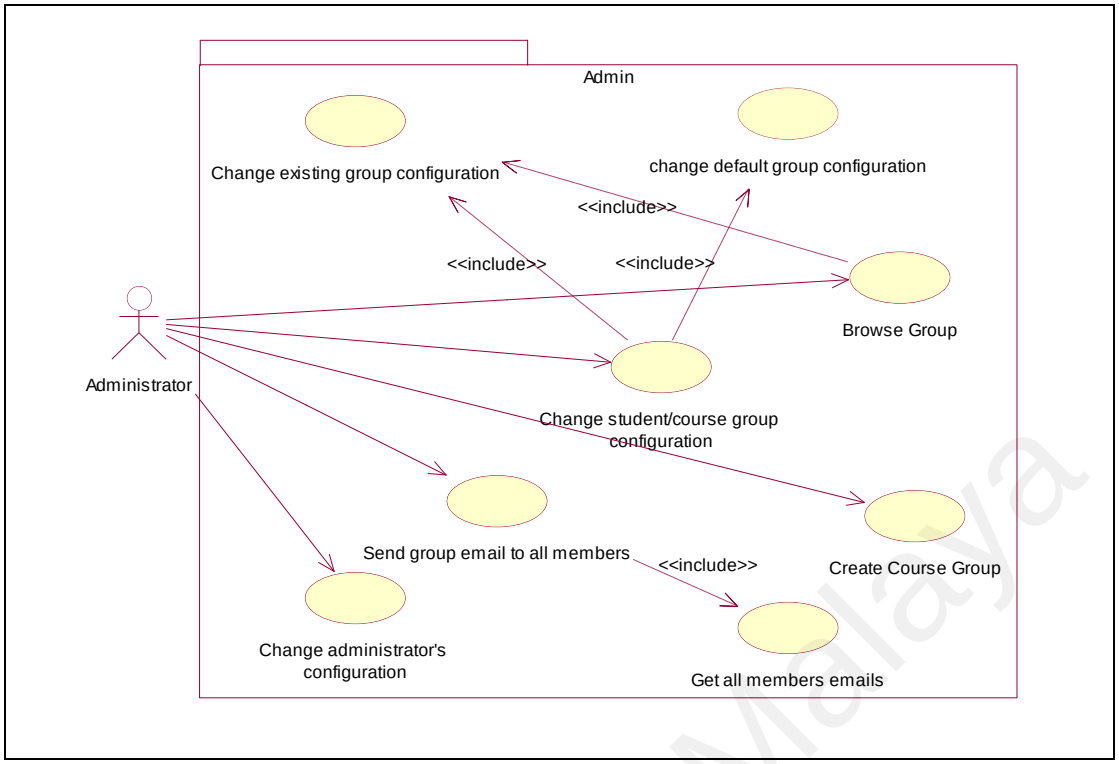


**Figure 4.2: Use case diagram of the Administration package**

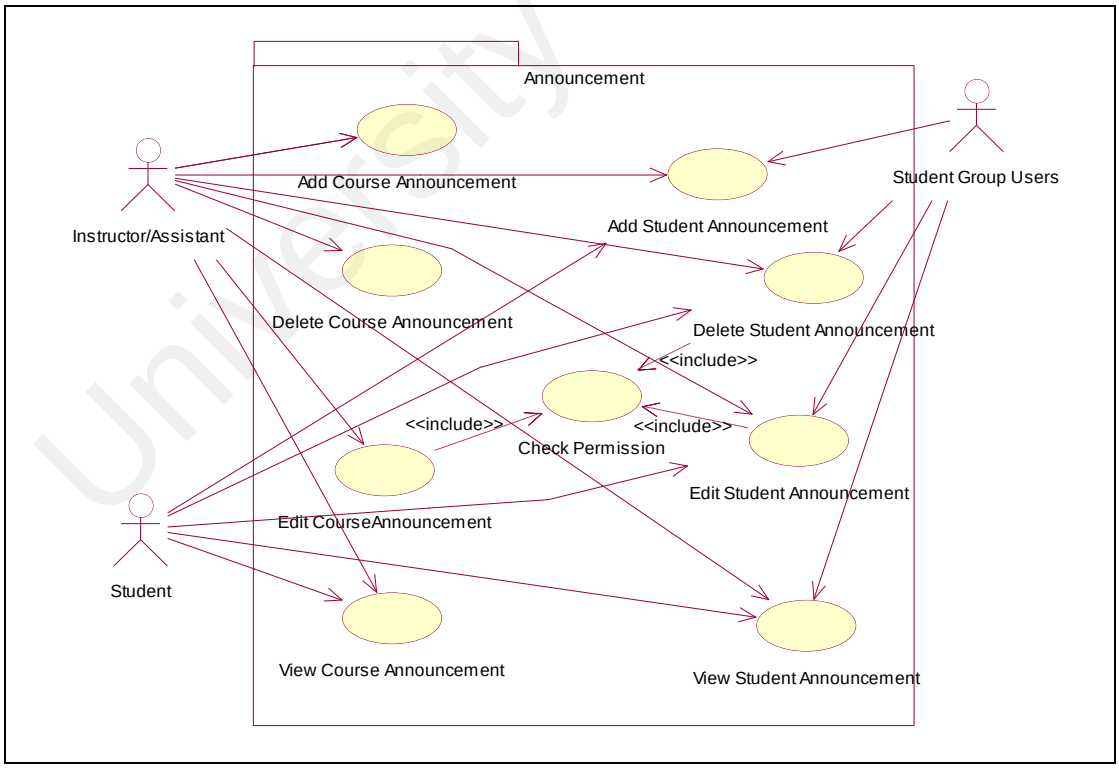


**Figure 4.3: Use case diagram of the Announcement package**

The “delete student announcement”, “edit student announcement” and “edit course announcement” use cases in Figure 4.3 require validation in order to perform the corresponding action. The validation or “check permission” use case is performed according to the following rules:

- The person who posted a record can delete that particular record.
- The Course Group Instructor and Assistant can delete all records posted in the Course Group, which include the course record and student record.
- The Student Group Founder and Manager can delete all records posted in the Student Group.
- Only the person who posted a record can edit that particular record.

The above rules are applied at all other use cases that have the “check permission” use case.

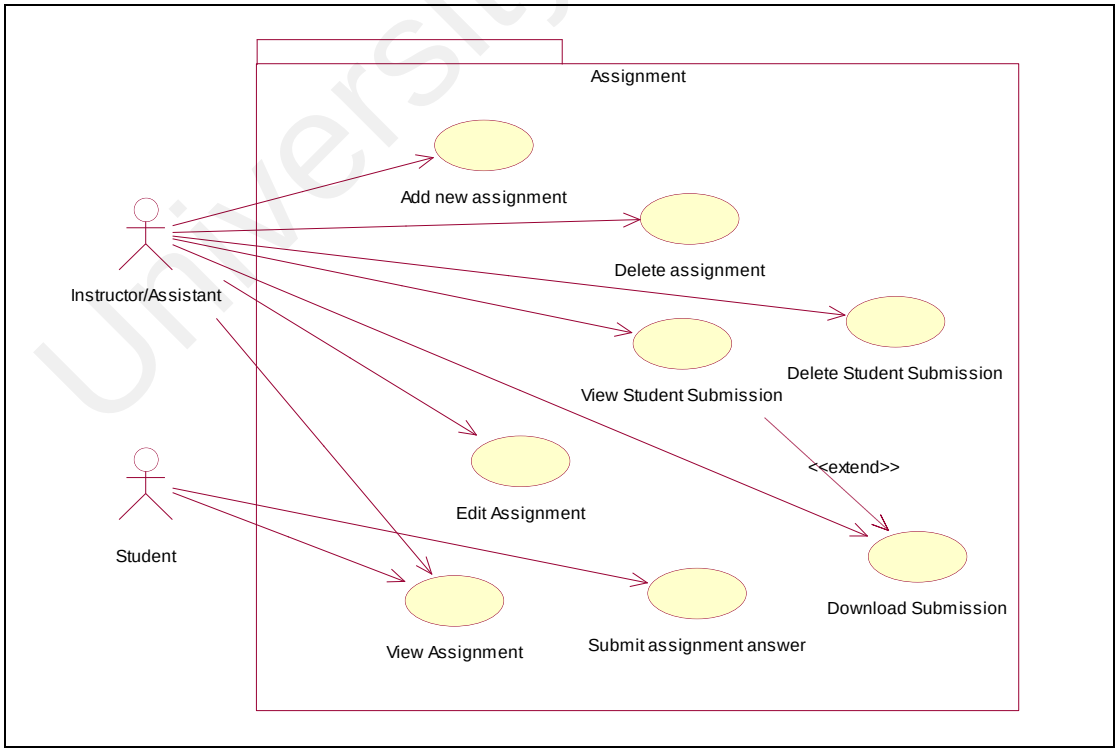


**Figure 4.4: Use case diagram of the Assignment package**

In Figure 4.4, it is shown that the "view student submission" use case can proceed to download the students' uploaded assignment answer files.

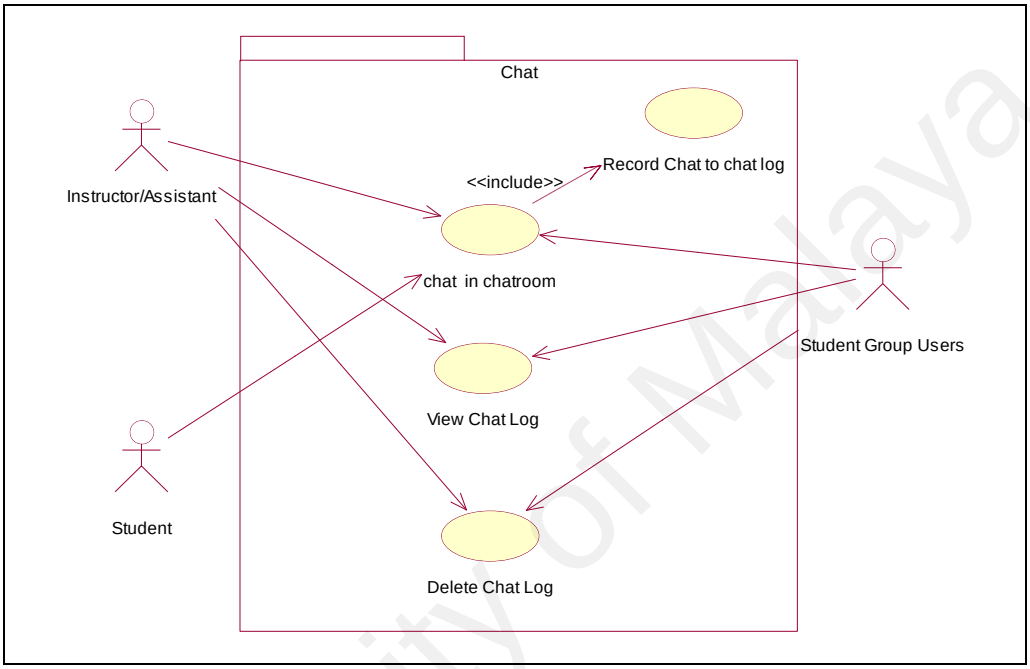


**Figure 4.5: Use case diagram of the Chat package**

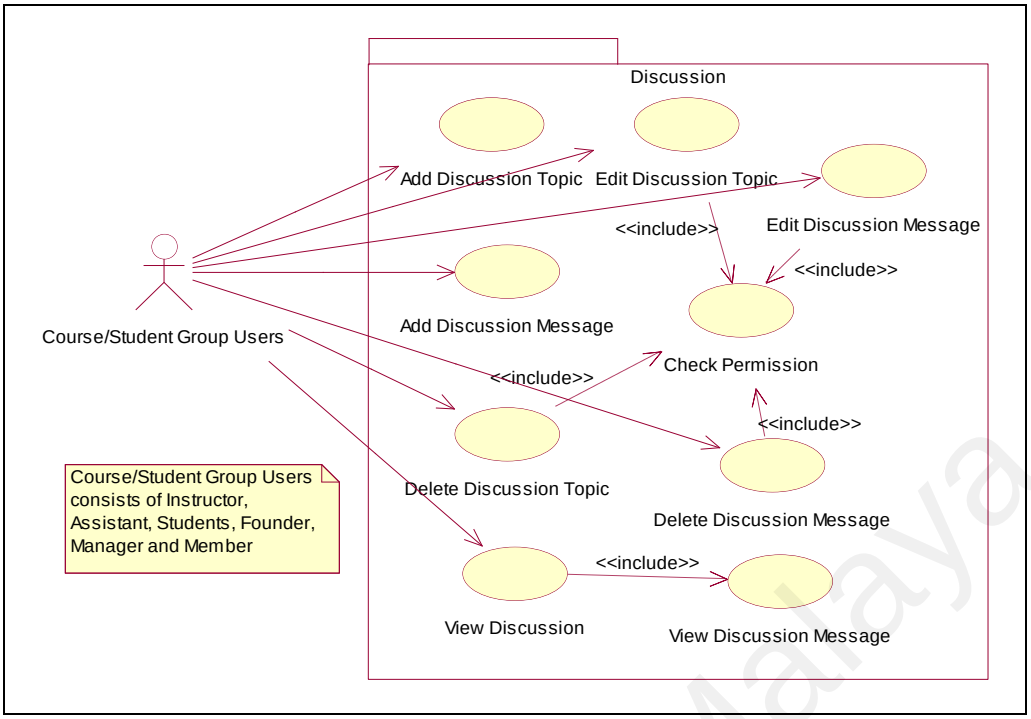


**Figure 4.6: Use case diagram of the Discussion package**

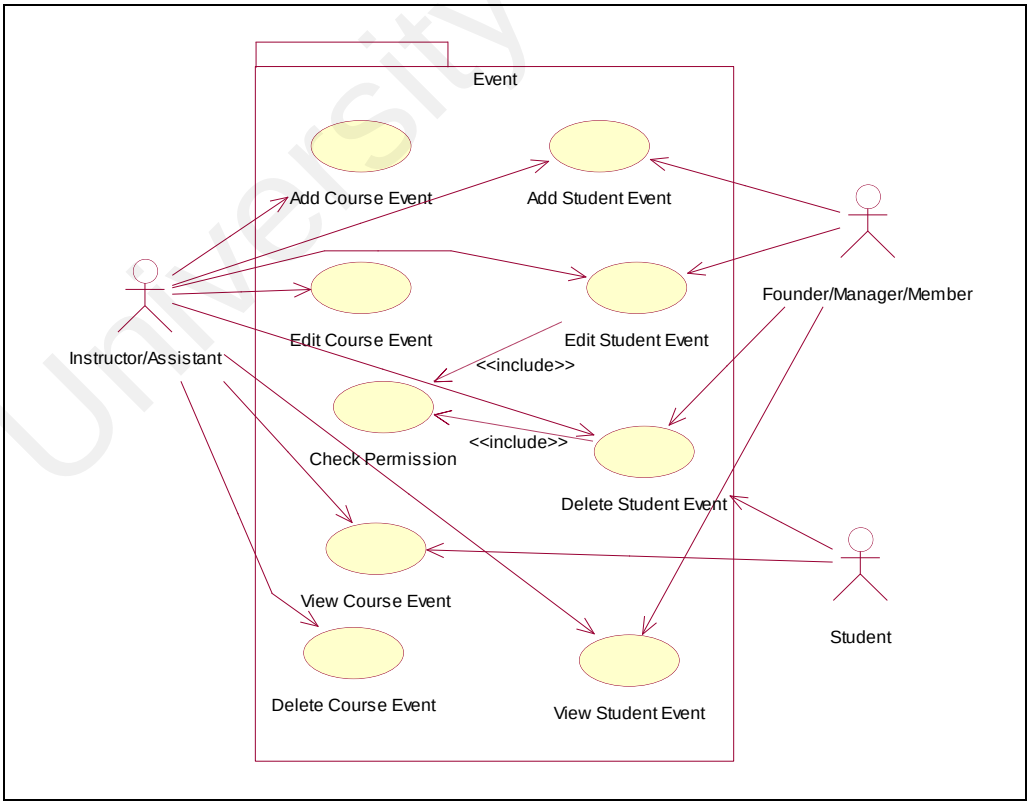


**Figure 4.7: Use case diagram of the Event package**

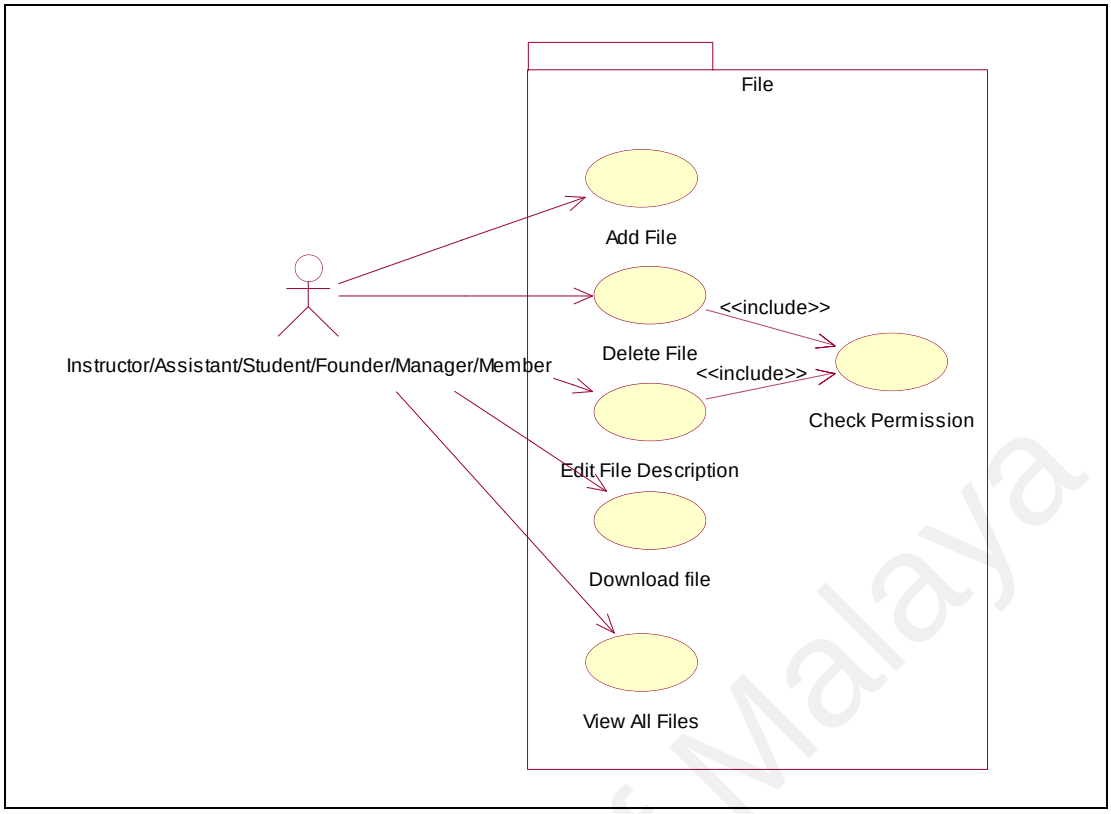


**Figure 4.8: Use case diagram of the File package**

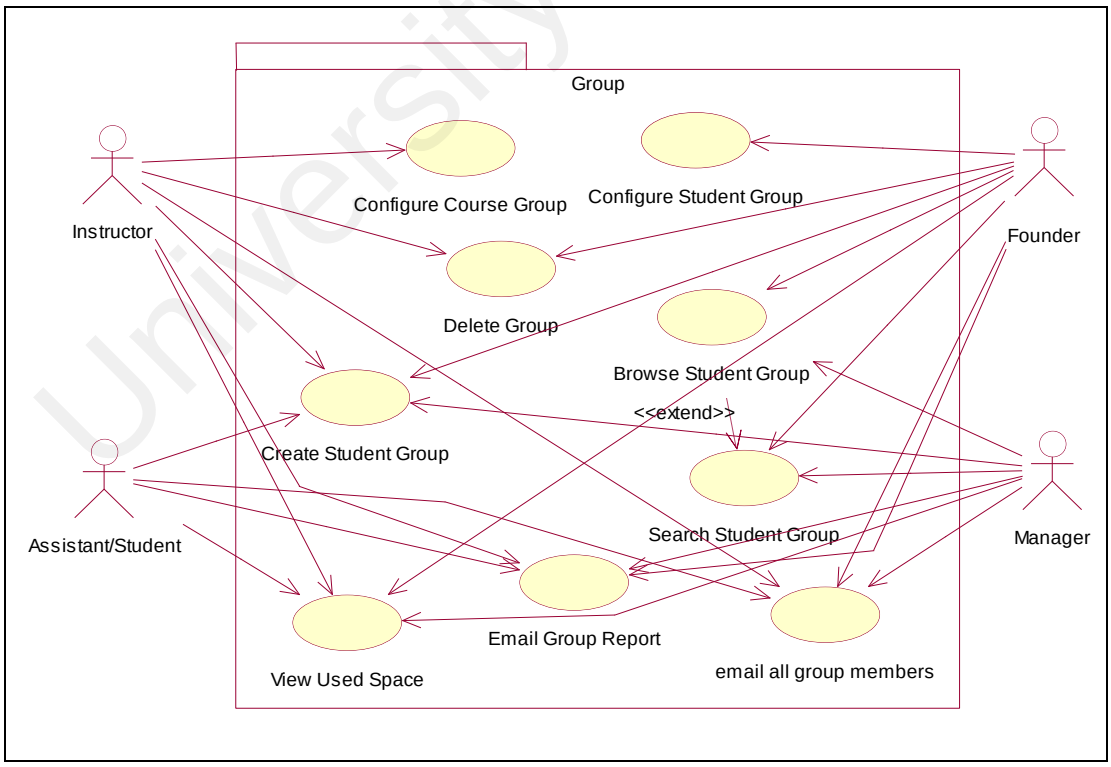


**Figure 4.9: Use case diagram of the Group package**

As shown in Figure 4.9, users are allowed to create any number of Student Groups. However, only the administrator is allowed to create Course Group, as shown in Figure 4.2. The main difference between Course Group and Student Group is the Course Group contains more functions, which particularly used to support a course.

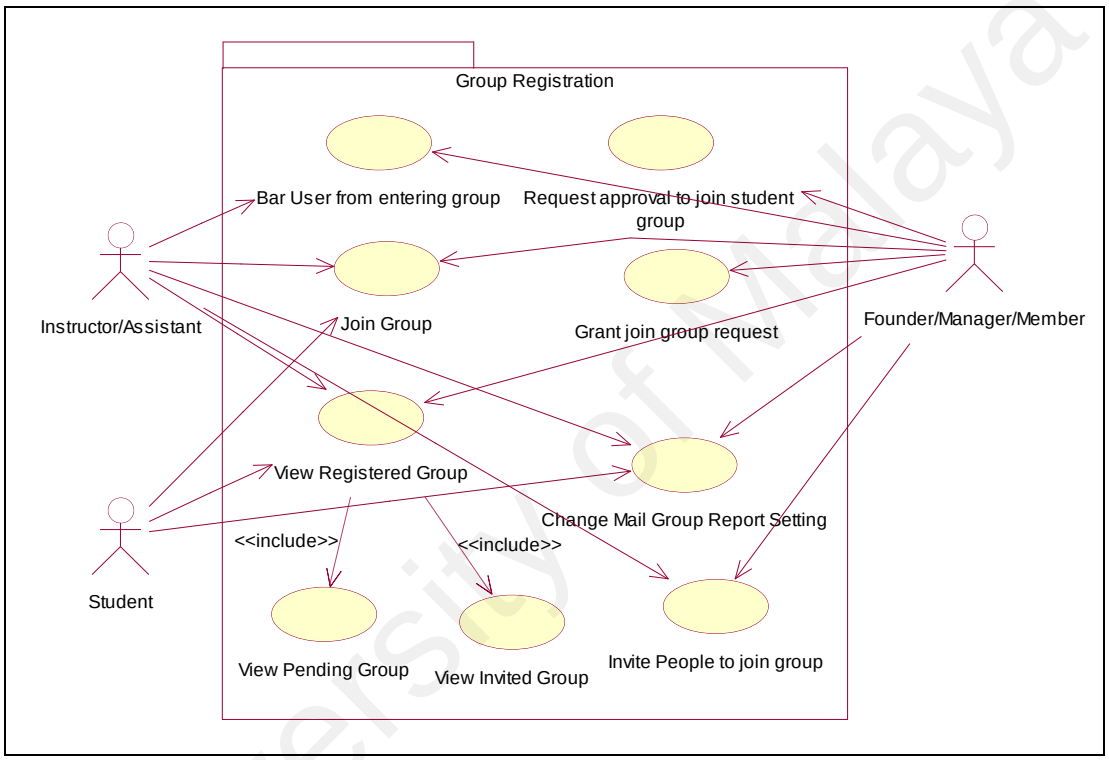


**Figure 4.10: Use case diagram of the Group Registration package**

Figure 4.10 shows that all the Student Group users – founder, manager and member, are allowed to invite and approve people to join their group. However, in Course Group, only the Course Group Instructor and Assistant have the authority to assign people as the Course Group members. Besides, people also are not allowed to send requests to join any Course Group.

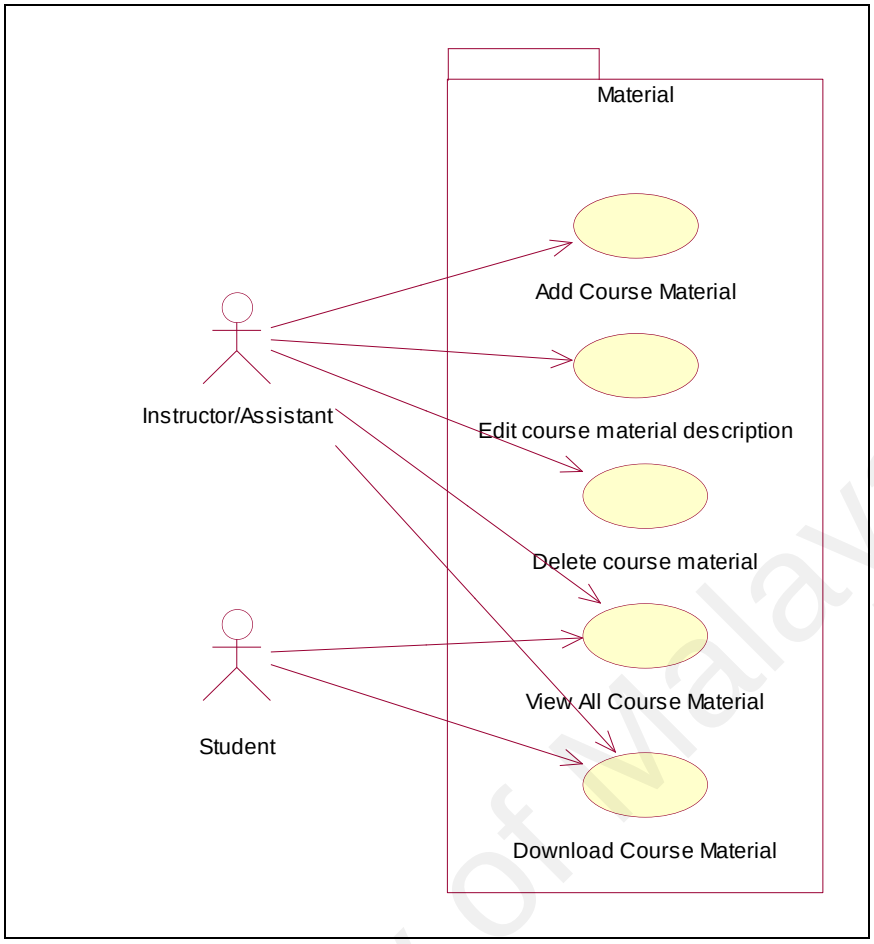


**Figure 4.11: Use case diagram of the Material package**

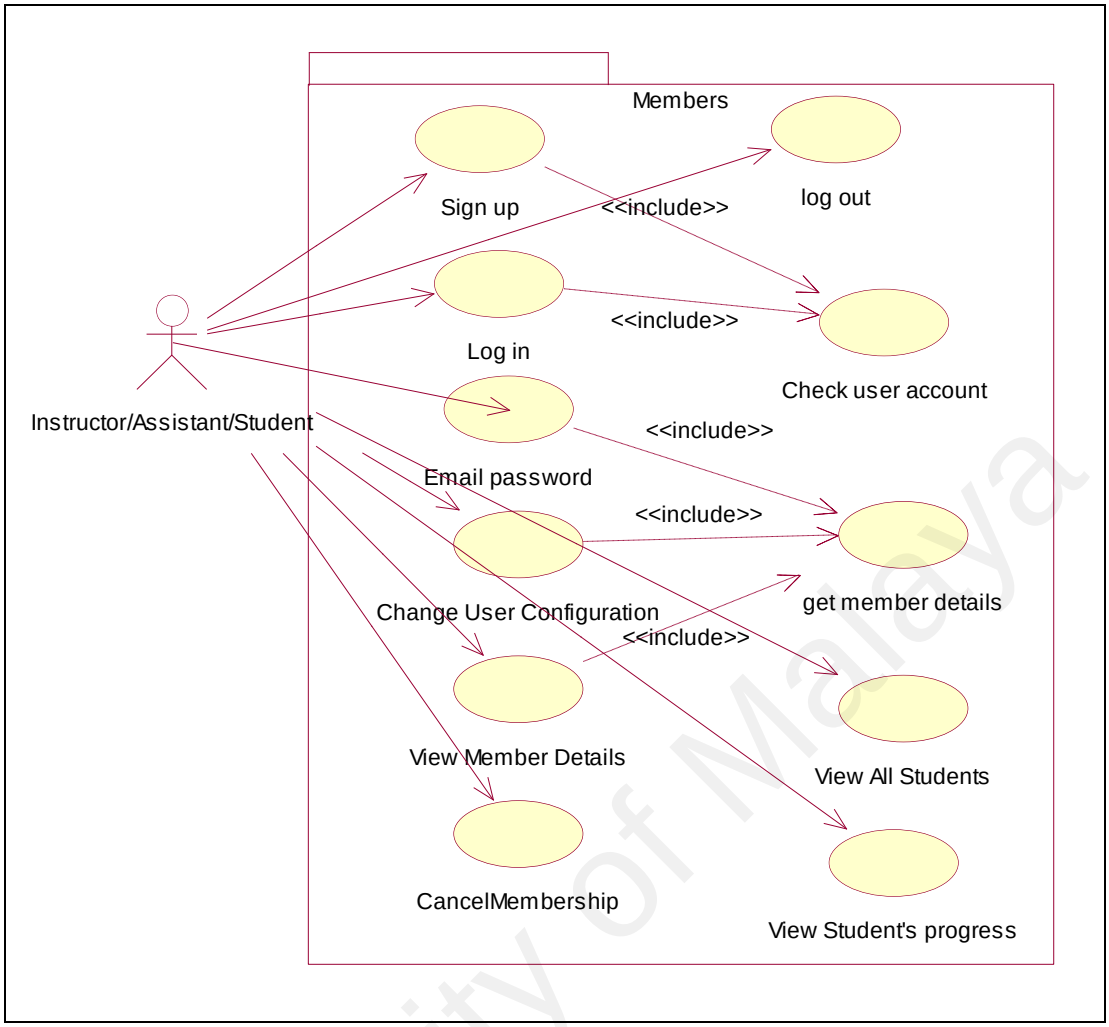


**Figure 4.12: Use case diagram of the Member package**

The "email password" use case in Figure 4.12 is used for sending an email which contains the password information to the corresponding user. This feature is useful in case the user has forgotten his password. Before the user sign up an account at WebGE, the system performs a checking in order to disallow duplicate email or nickname.

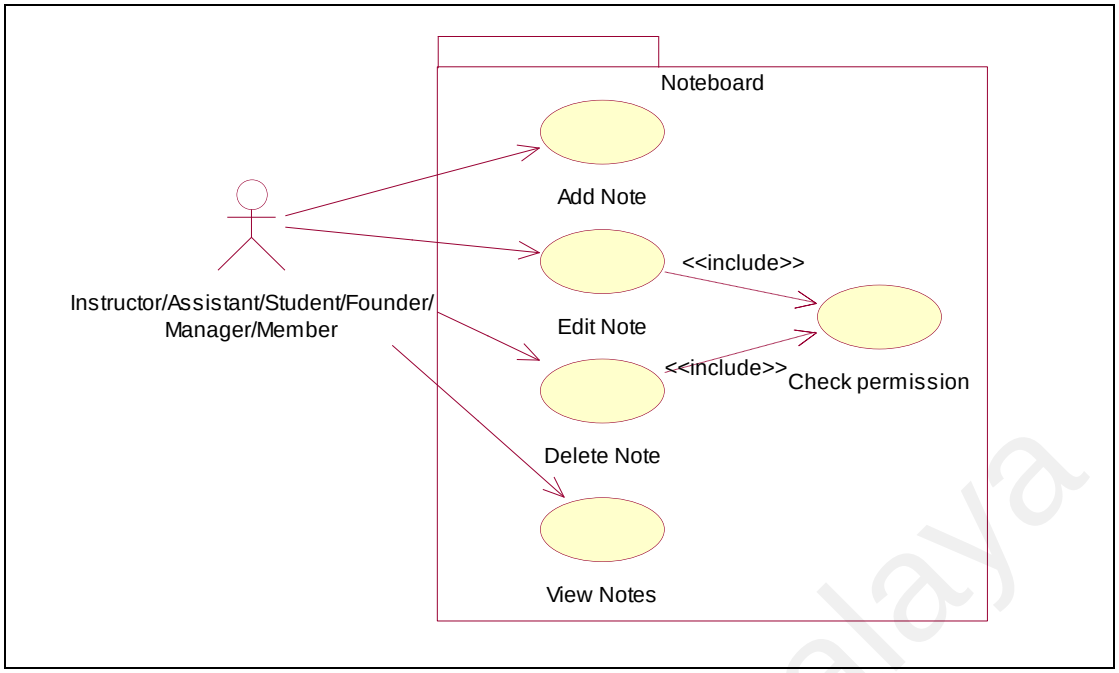


**Figure 4.13: Use case diagram of the Noteboard package**

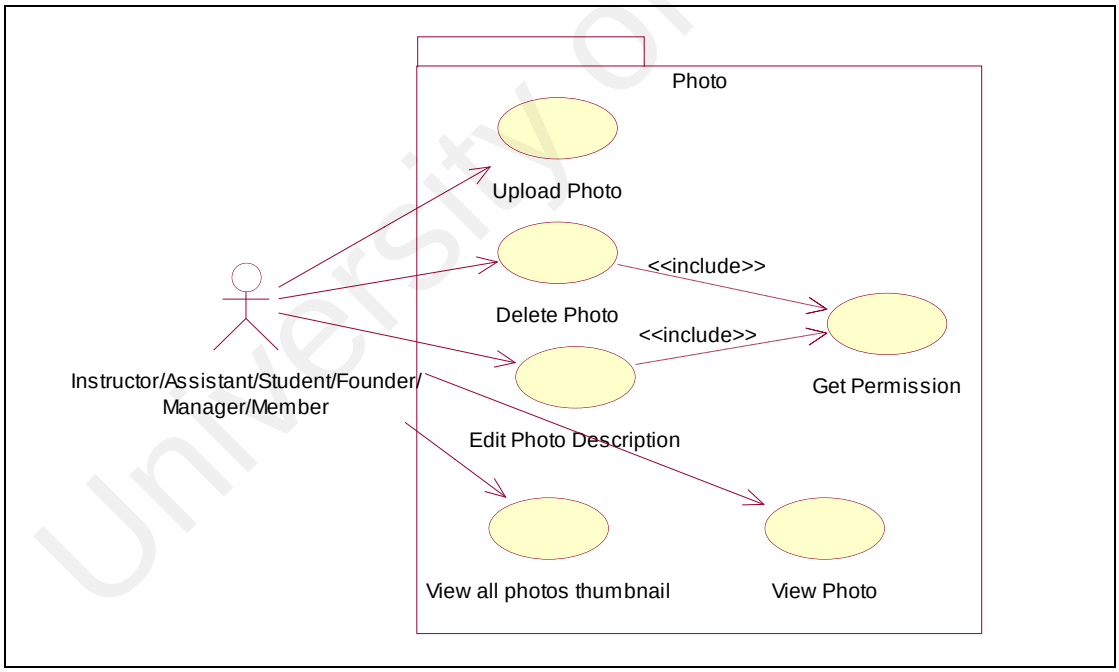


**Figure 4.14: Use case diagram of the Photo package**

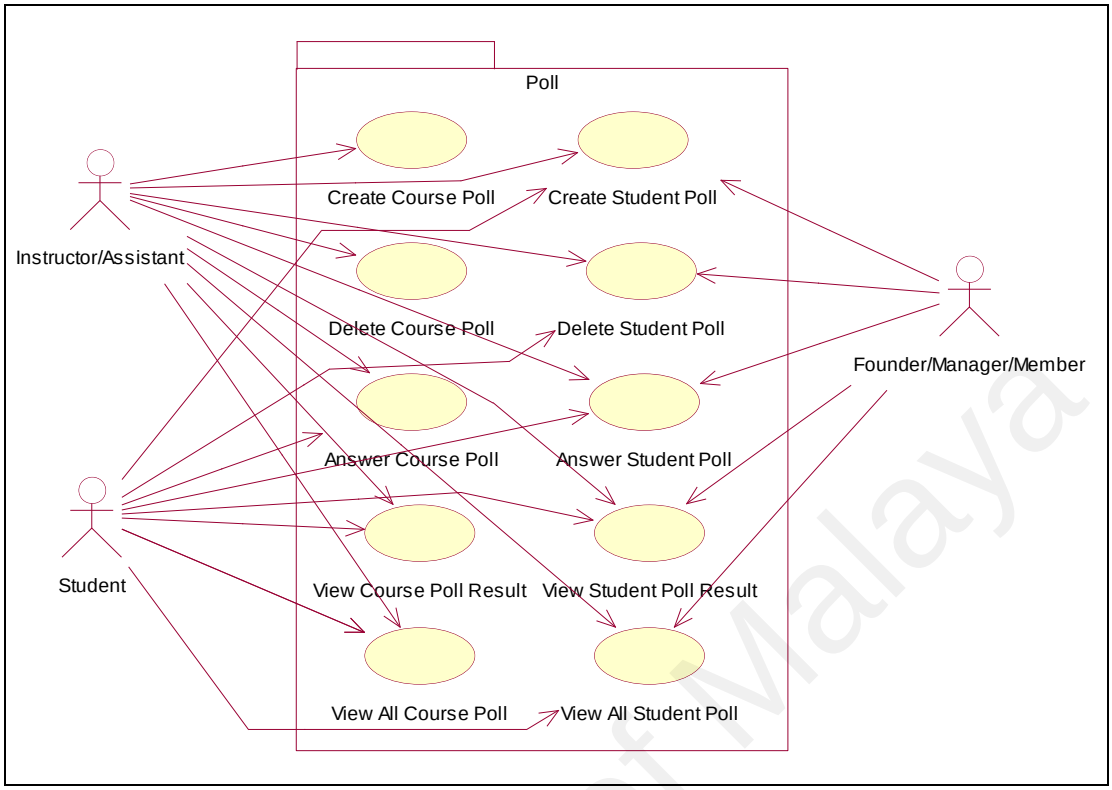


**Figure 4.15: Use case diagram of the Poll package**

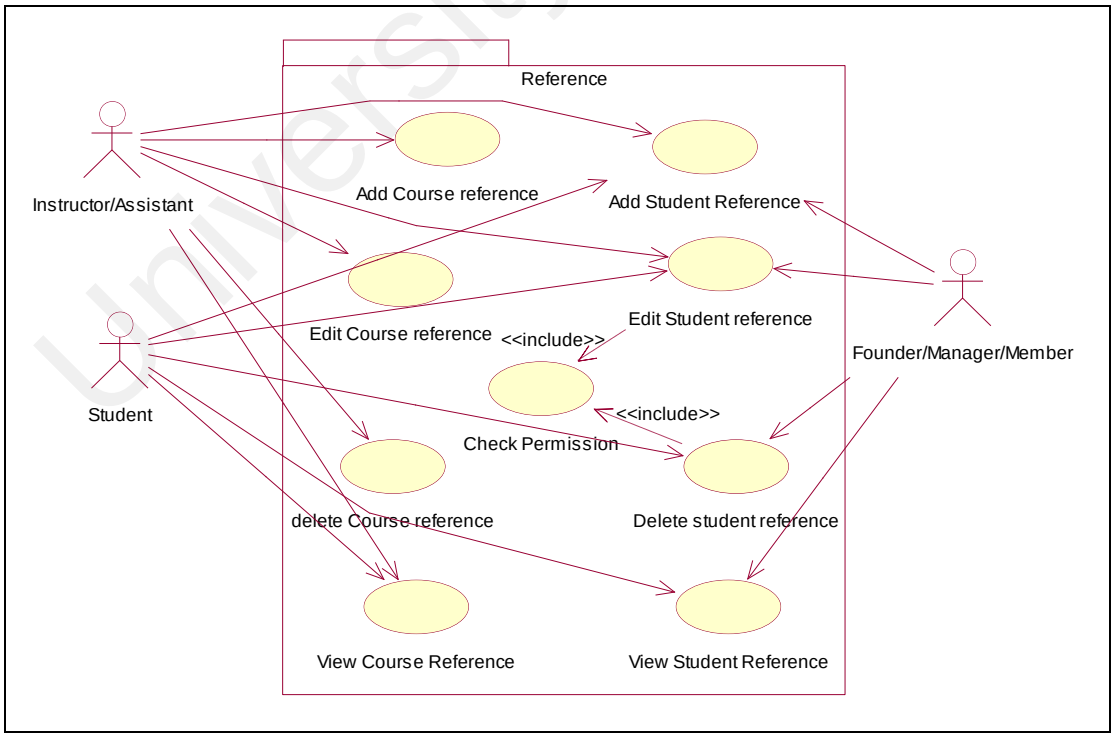


**Figure 4.16: Use case diagram of the Reference package**

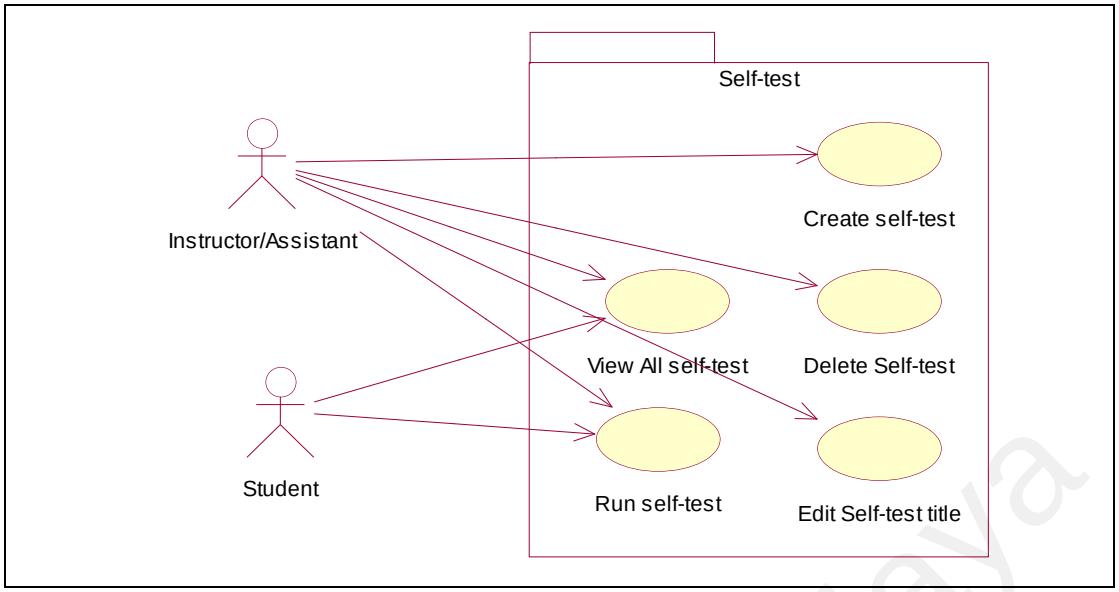


**Figure 4.17: Use case diagram of Self-test package**

### 4.2.2 Non-functional Requirements

Four types of non-functional requirements, namely usability, reliability, performance and supportability, are identified in this study. The following gives a brief description of these non-functional requirements.

- Usability – The ease with which the system can be used.
- Reliability – The extent to which the system is dependable.
- Performance – Covers areas such as response time, throughput, capacity and degradation modes.
- Supportability – The extent to which the system can be easily modified to accommodate enhancements and repairs.

All four types of non-functional requirements for WebGE are presented in Table 4.3.

**Table 4.3: Non-Functional Requirements**

| Types | Requirements |
|---|---|
| Usability | Ease of use<br>▪ Intuitive and easy to use:<br>▪ Minimal training required for members |
| | Instructions<br>▪ Easy-to-follow instructions are available. |
| | Feedback<br>▪ Instant feedback (both positive confirmation and error) is always given. |
| | Delete confirmation message<br>▪ Delete confirmation message is always displayed whenever a delete request is made by the users. |
| | Language<br>▪ Language used in the system should be understood unambiguously by non-technical users. Wherever the use of jargons is necessary, the terms shall be explained to the users. |
| | Navigation<br>▪ Navigation should be fast and easy:<br>▪ A global navigation bar shall exist on all screens.<br>▪ Navigation from one place to another in the system should take a minimal and reasonable number clicks. |

| Types | Requirements |
|---|---|
| | Compatibility<br>▪ The system should work under Internet Explorer v5.0 and above. |
| | Link Verification<br>▪ No dead hyperlink shall exist in the system |
| Reliability | Security<br>▪ The system shall be secure from unauthorized access.<br>▪ All data shall be kept away from unauthorized users. |
| | Accuracy<br>▪ All posted data should be free of errors. |
| Performance | Response time<br>▪ All transactions should take less than one second to process. |
| | Scalability<br>▪ Throughput: The system can process up to 100 transactions per second.<br>▪ Capacity: The system can accommodate up to 100 users and respondents at a time. (Scalability test is performed using Microsoft Application Centre Test, which is provided in the Visual Studio.Net 2003 package) |
| | Download time<br>▪ The download time for each page should take at most one second. |

| Types | Requirements |
|---|---|
| Supportability | Supportability<br>▪ The system should be easy to be upgraded and modified to accommodate future enhancements. |

Apart from identifying the above non-functional requirements, the hardware and software requirements for WebGE are also needed to be specified. The minimum hardware and software requirements for WebGE are shown in Table 4.4 and 4.5.

**Table 4.4: Hardware requirements**

| | Server | Client |
|---|---|---|
| Processor | Intel Pentium III or compatible 450 MHz. | Intel Pentium or compatible 166Mhz. |
| Memory | 128MB (for SQL Server 2000 Enterprise Edition)<br>64MB (for SQL Server 2000 Standard Edition) | 64MB |
| Disk Space | 10MB (for installation of WebGE only) | - |
| Display | 800 X 600 VGA | 800 X 600 VGA |
| Keyboard & Mouse | Yes | Yes |

**Table 4.5: Software Requirements**

| Server | Client |
|---|---|
| Microsoft Windows 2000 Server | Microsoft Windows 98 or later |
| Microsoft .NET Framework | - |
| Internet Information Server v5.0 | Internet Explorer 5.0 |
| Microsoft SQL Server Enterprise Edition / Standard Edition | - |

## 4.3 Requirement Analysis

The requirement analysis is concerned with the realization of all the use cases that have been identified. The diagram used for requirement analysis is interaction diagram, which can be either collaboration diagram or sequence diagram.

Both sequence and collaboration diagrams show the same information. However, depending on the requirements of stakeholders and developers, either one or both may be generated to document the data flow within the system. In this study, only the collaboration diagrams are generated. Figure 4.18 to Figure 4.33 shows the collaboration diagrams for some of the more complex use cases that have been identified. These diagrams are self-explanatory and hence, no further descriptions are provided.

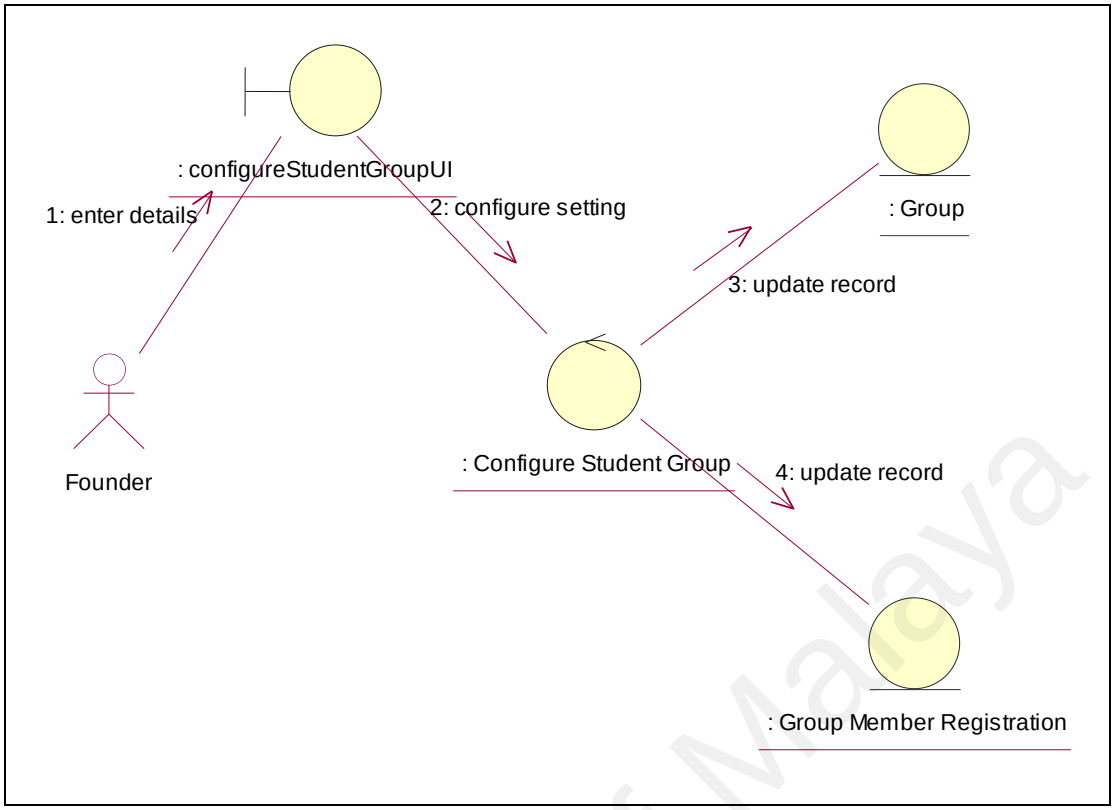


**Figure 4.18: Collaboration diagram for Configure Student Group use case**

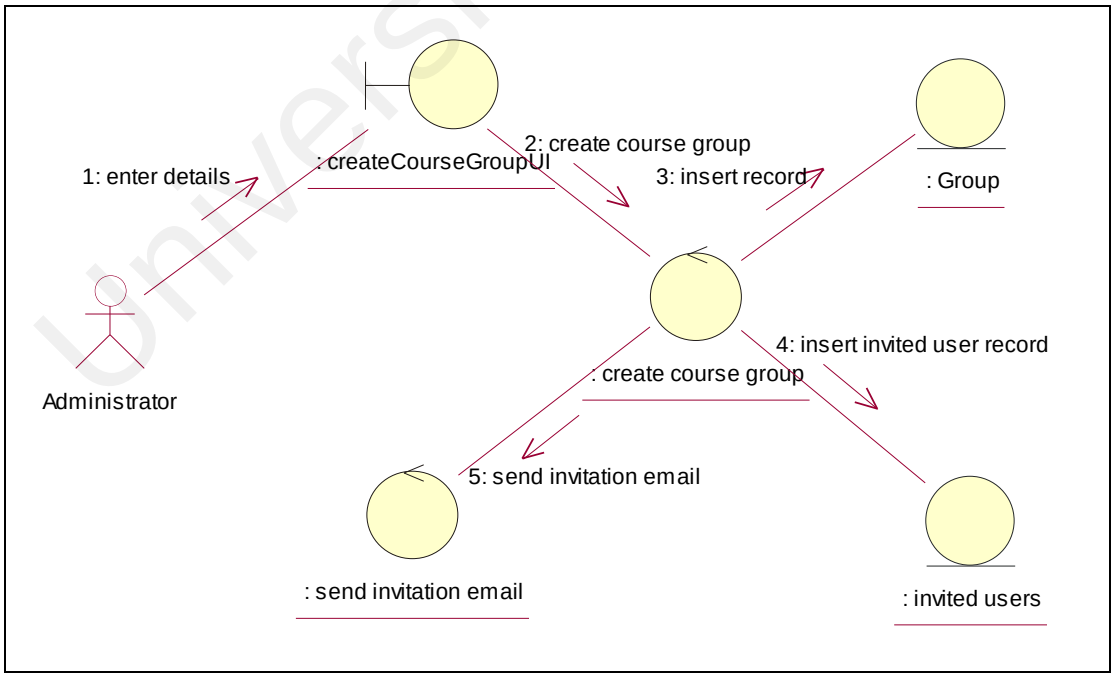


**Figure 4.19: Collaboration diagram for Create Course Group use case**

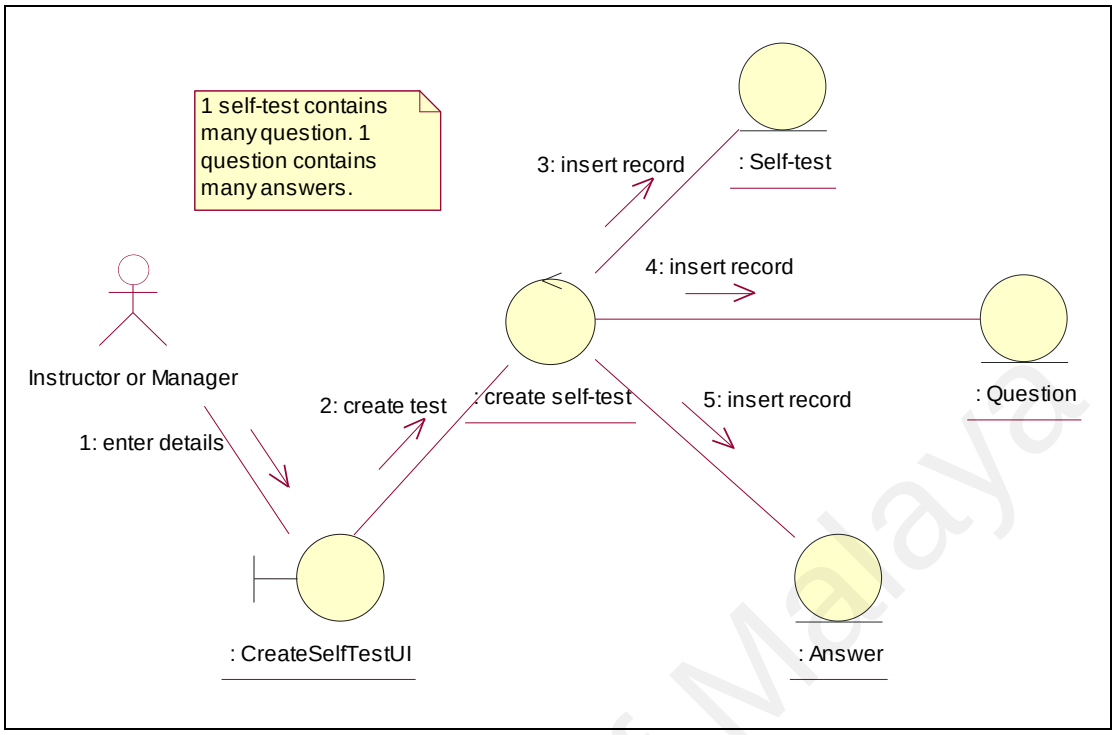


**Figure 4.20: Collaboration diagram for Create Self-test use case**

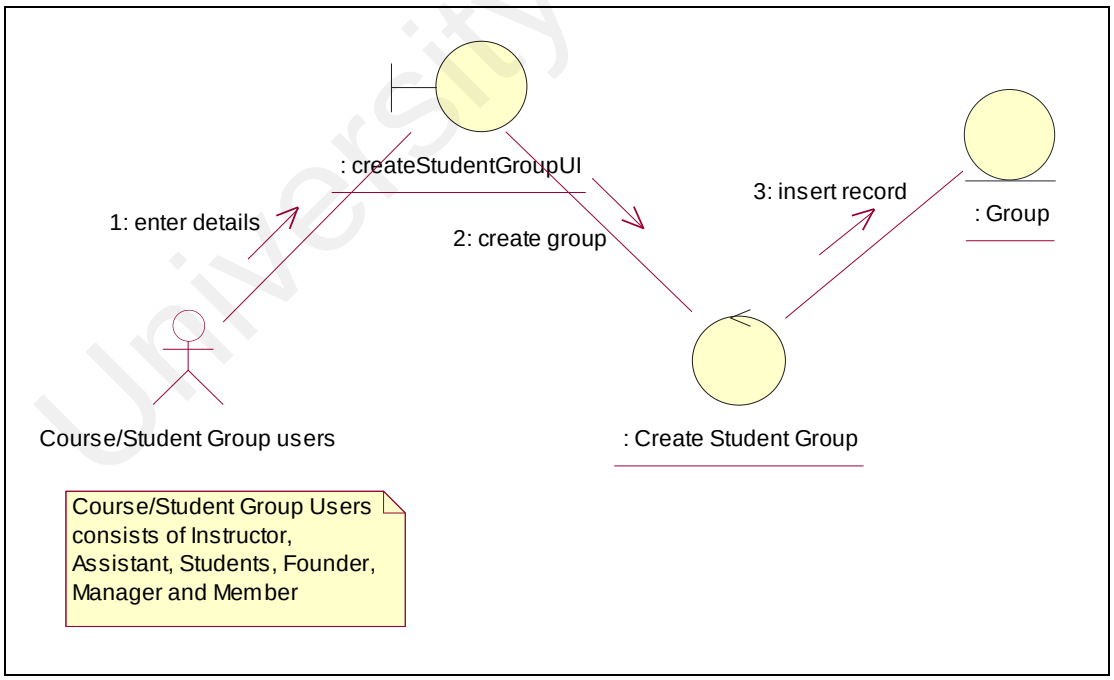


**Figure 4.21: Collaboration diagram for Create Student Group use case**

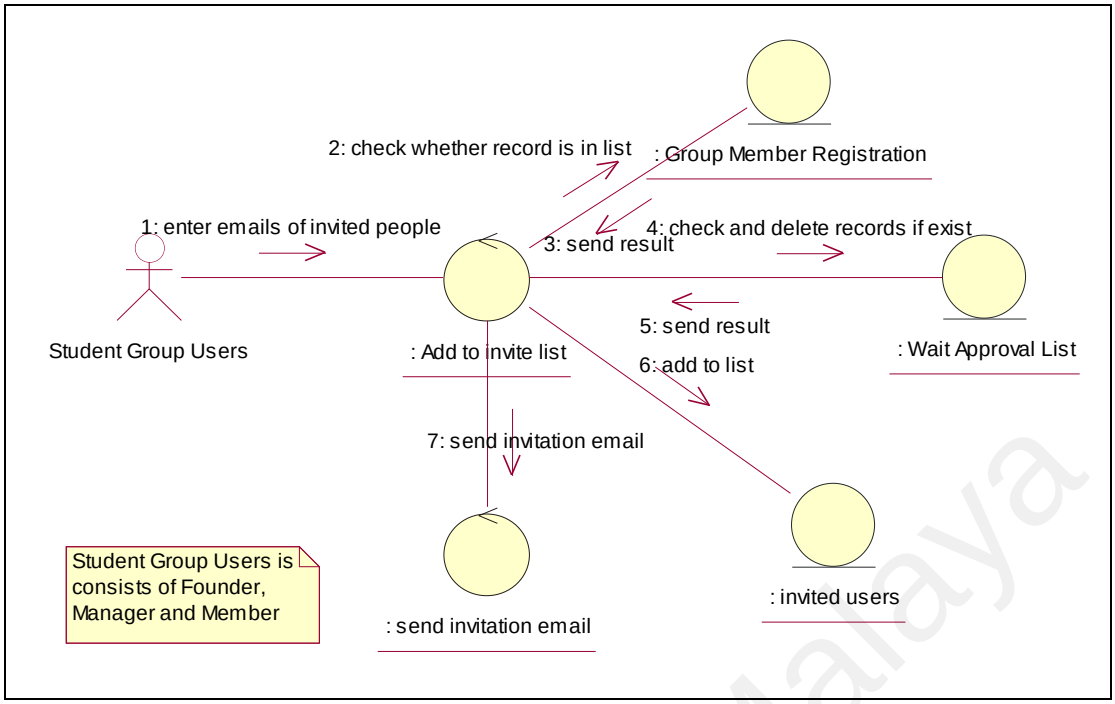


**Figure 4.22: Collaboration diagram for Invite People to Join Group use case**

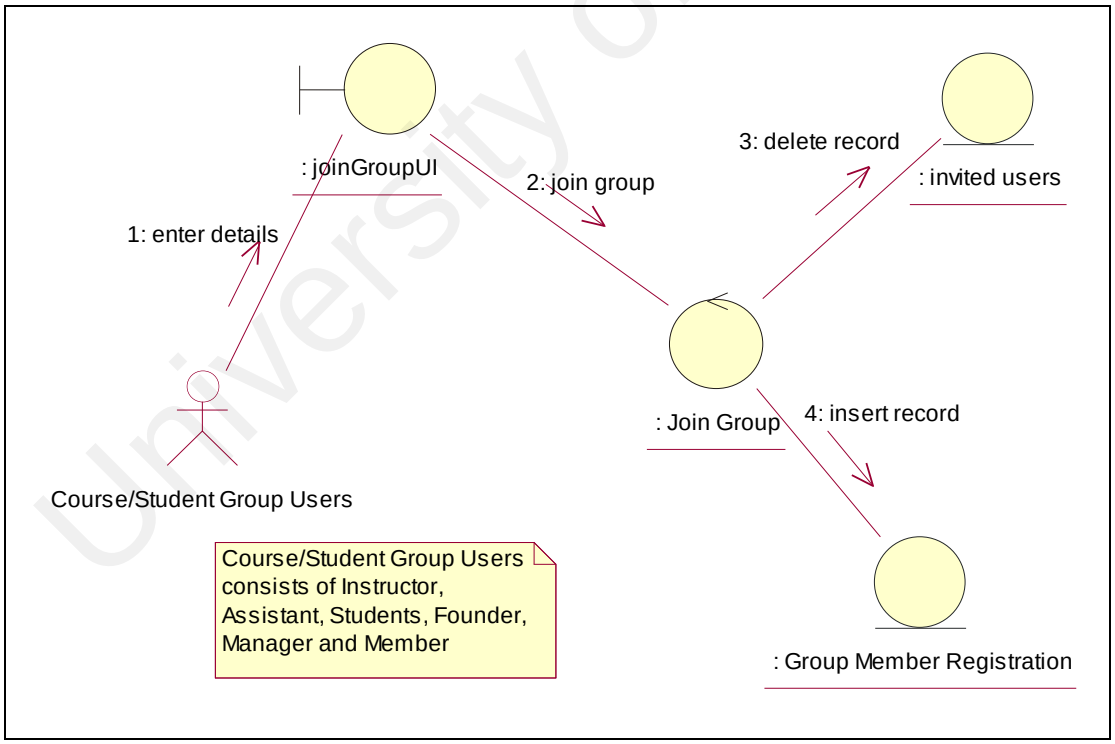


**Figure 4.23: Collaboration diagram for Join Group use case**

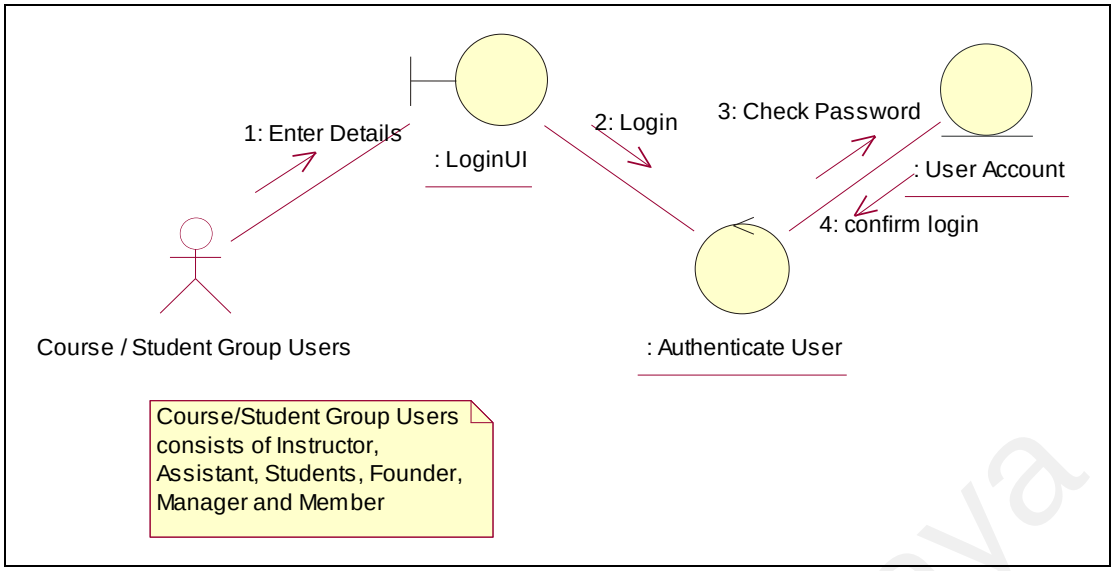


**Figure 4.24: Collaboration diagram for Log In use case**

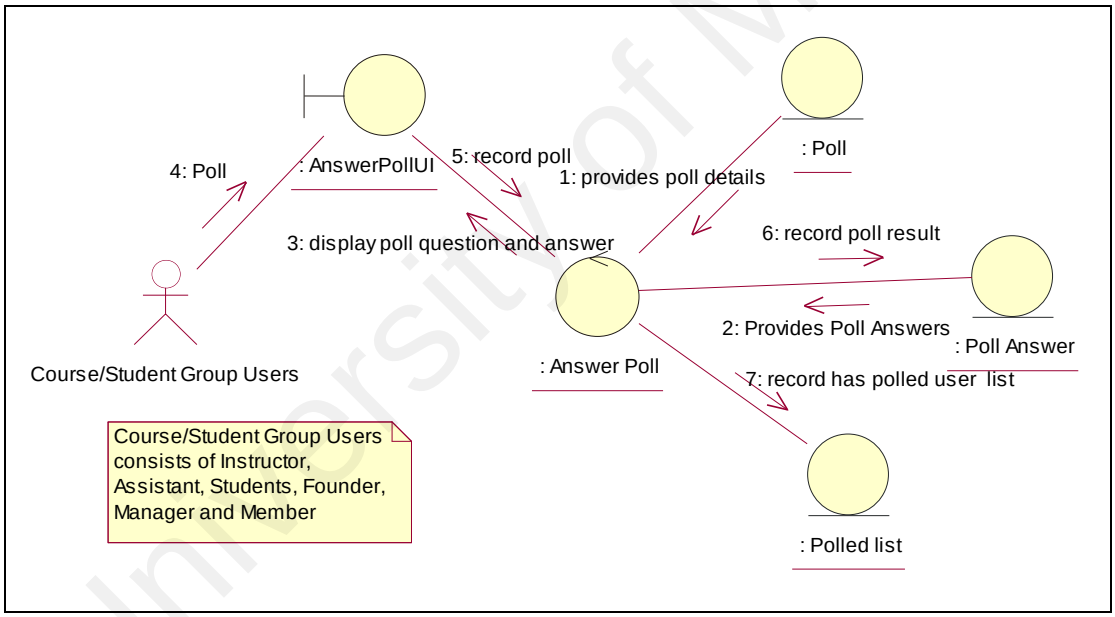


**Figure 4.25: Collaboration diagram for Answer Poll use case**

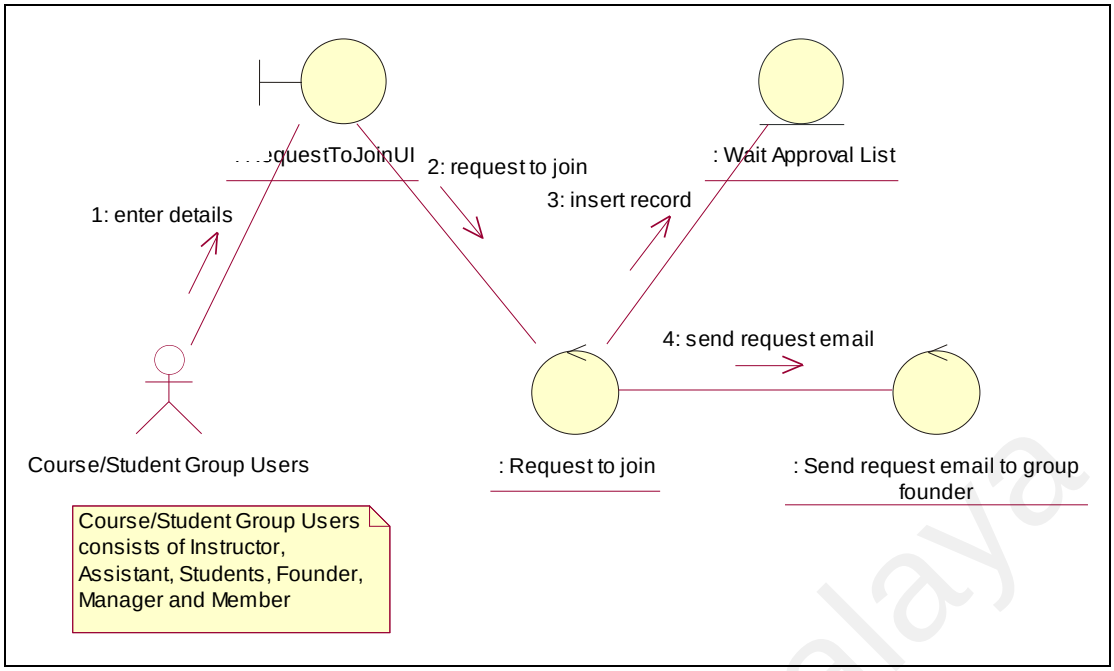


**Figure 4.26: Collaboration diagram for Request to join Student Group use case**

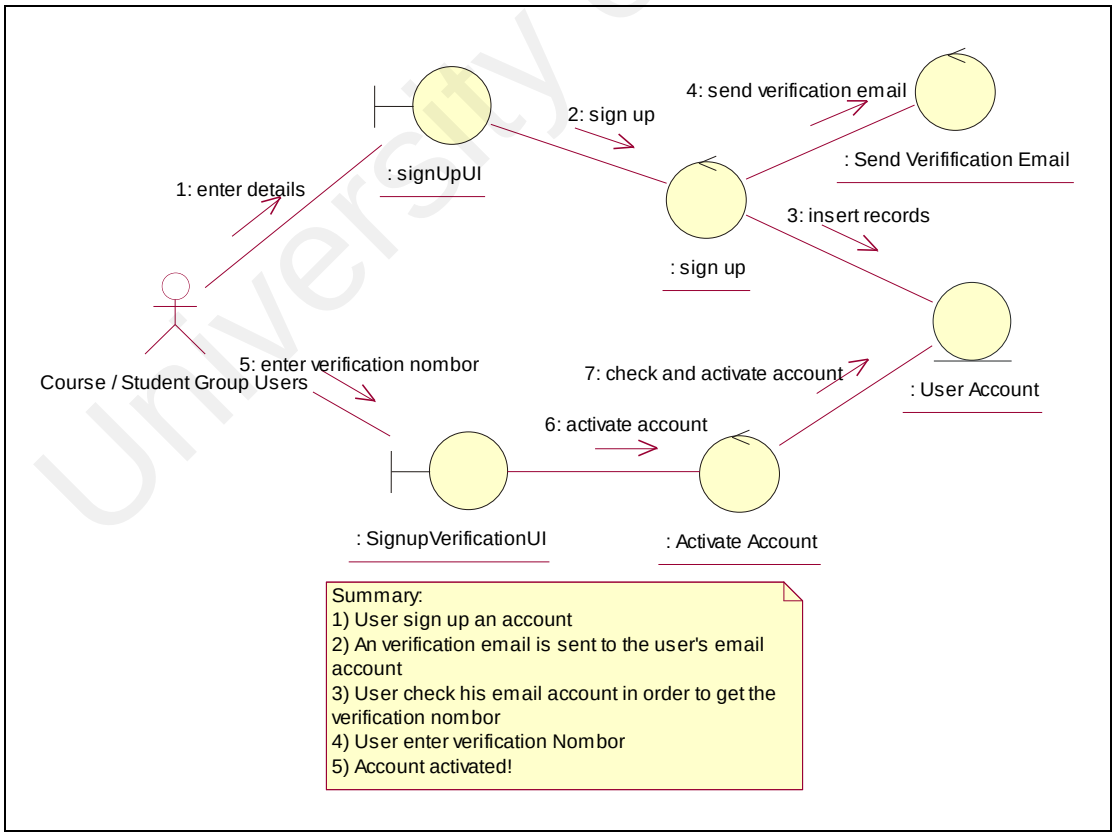


**Figure 4.27: Collaboration diagram for Sign Up Account use case**

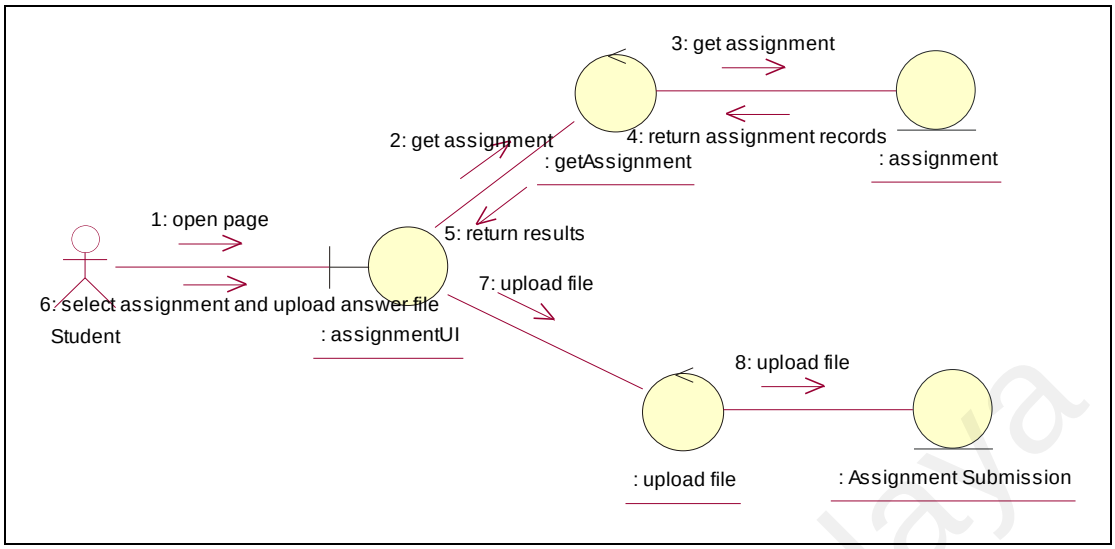


**Figure 4.28: Collaboration diagram for Submit Assignment Answer use case**

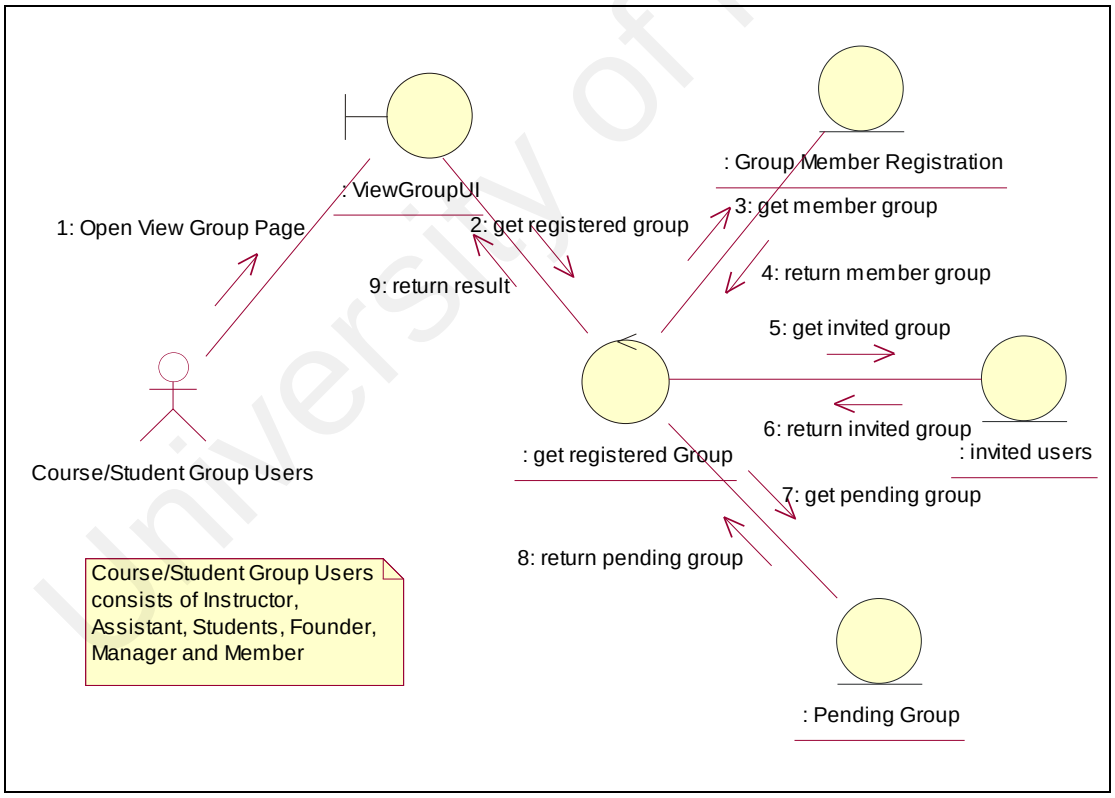


**Figure 4.29: Collaboration diagram for View Registered Group use case**

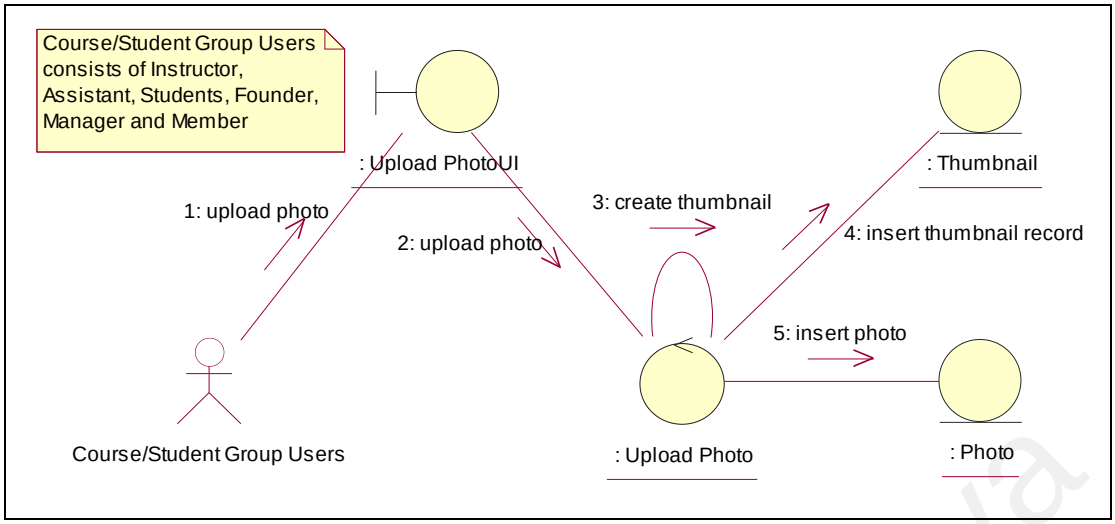


**Figure 4.30: Collaboration diagram for Upload Photo use case**

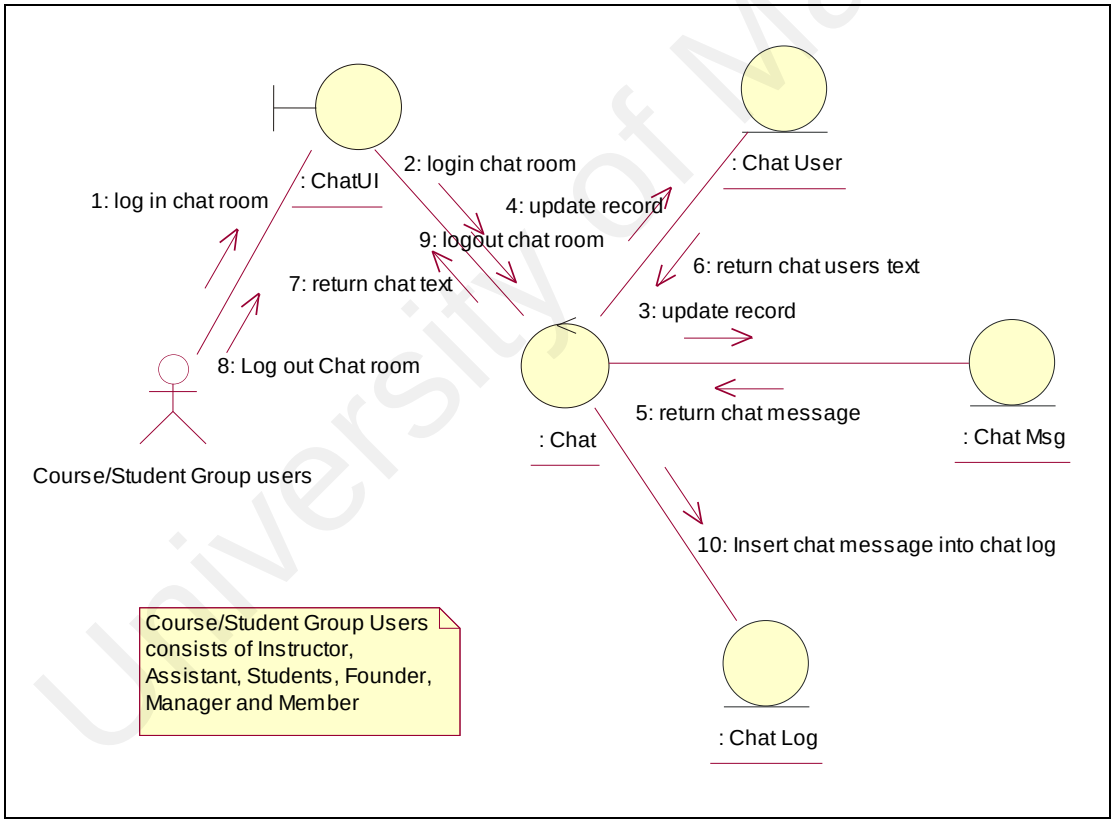


**Figure 4.31: Collaboration diagram for Chat in Chat Room use case**

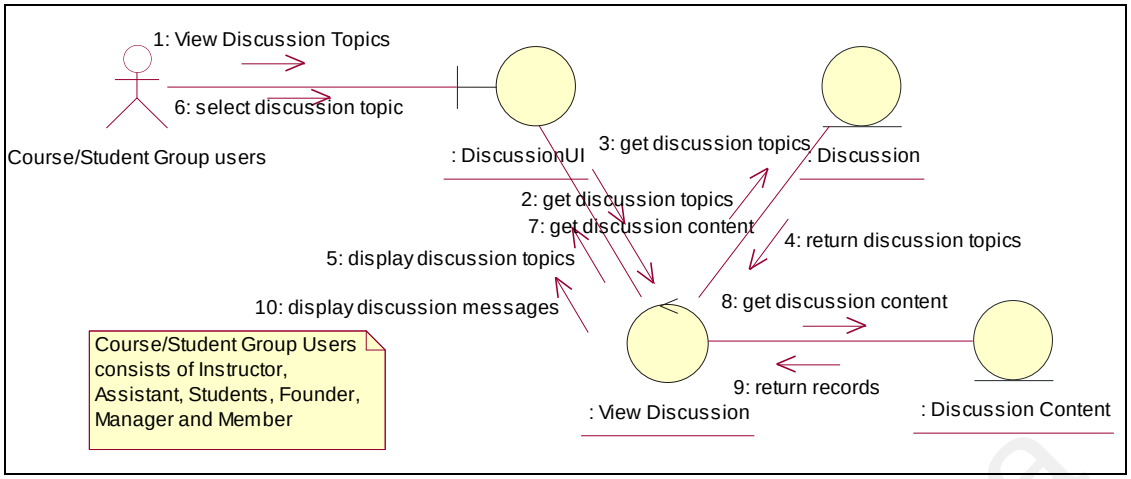


**Figure 4.32: Collaboration diagram for View Discussion use case**

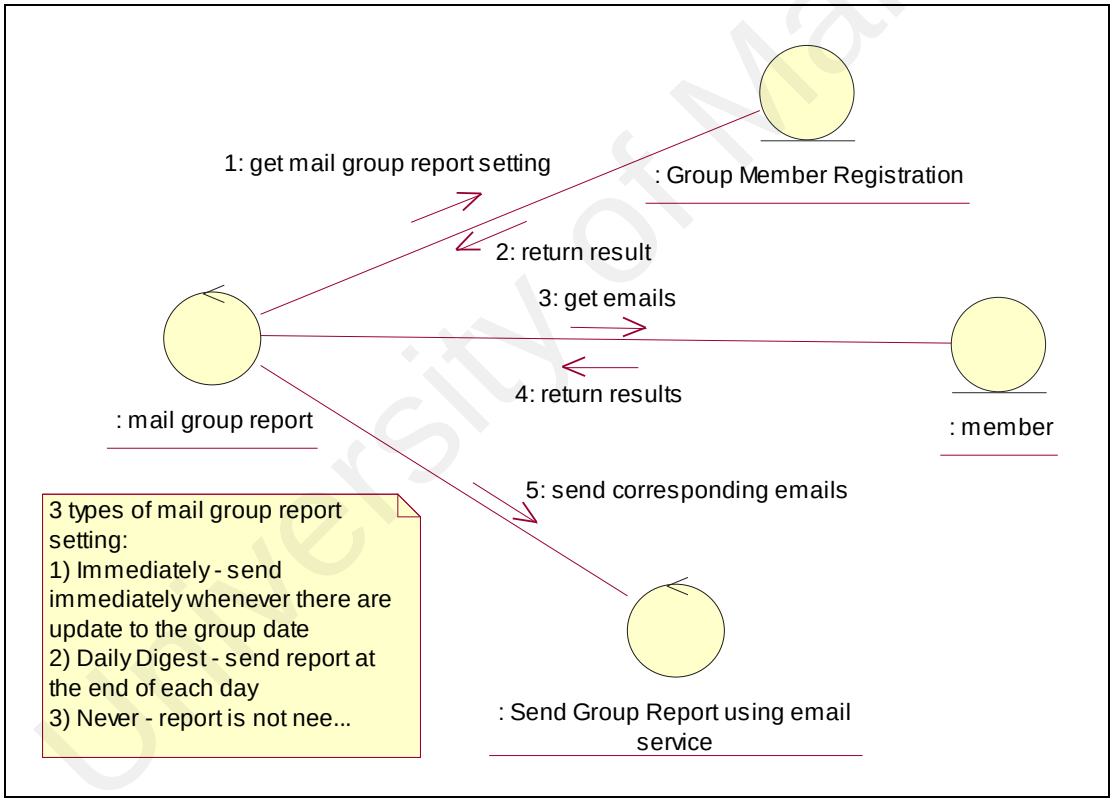


**Figure 4.33: Collaboration diagram for Email Group Report use case**

# Chapter 5 – System Design

## 5.1 Introduction

The main objective of system design is to detail the requirements that have been identified and modeled during system analysis. The outcome of system design is a design model which can be mapped easily into implementation. In this chapter, the following architectural designs are presented:

- Class Diagram
- Database Design
- User Interface Design

## 5.2 Introduction to Class Diagram

A class diagram describes the types of objects in the system and the various kinds of static relationships that exist among them (Kobryn, *et al.,* 2003). A class is a description of a set of objects that share the same attributes and behaviors. It is something that encapsulates information and behavior. Class diagrams are used to model the static design view of a system (Booch, *et al.*, 1999).

A class is mainly composed of attributes and operations. An attribute is a property of a class that describes a range of values that instances of the class may hold (Booch, *et*

*al.*, 1999). For example, a member of WebGE will have attributes such as first name, last name, email address and date of registration.

An operation refers to the implementation of a service that can be requested by other objects to behave in some way (Booch, *et al.*, 1999). For instance, a member of WebGE may have an operation called "ActivateAccount", which is used to change the account status to "Active".

In UML, a class is graphically represented by a rectangle that contains three sections. The top section holds the name of the class. The middle section holds the attributes while the bottom section contains the operations. A class can be shown with its attributes and operations visible or hidden, depending on the level of abstraction needed. Figure 5.1 shows an example of a Member class.

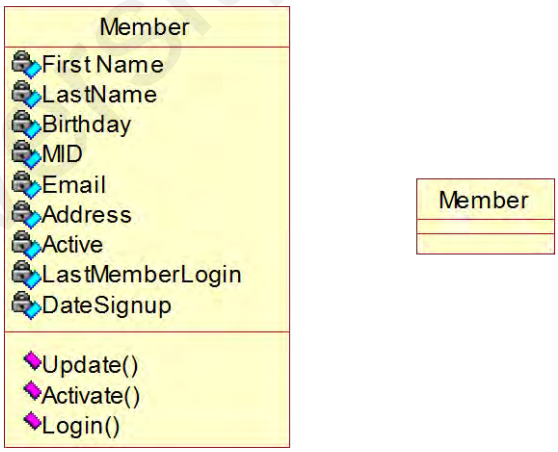


**Figure 5.1: A Member Class Example**

There are relationships among classes. Examples of classes' relationship are such as generalization, association and aggregation. The following text explains these relationships in brief.

Generalization is a relationship between a general class (super class) and a more specific kind of that class (subclass). For example, administrator and member are generalized from user class. Both administrator and member inherit all the features of user class and may override some user class method. Generalization is represented graphically at Figure 5.2.

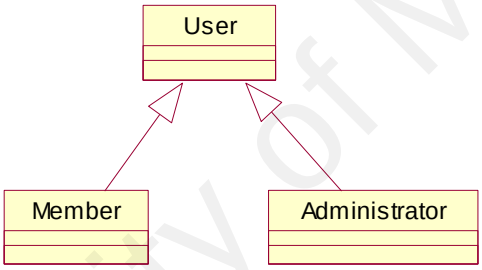


**Figure 5.2: Class Generalization Example**

An association is a structural relationship that specifies the connection between two classes. It can be used to navigate from one object of the class to another object of another class, and vice versa (Booch, *et al.*, 1999).

An association can have a name, roles and multiplicity. A name is simply a name to describe the relationship. A role describes how a class participates in a relationship. The multiplicity shows how many objects are connected to another object. The figure below shows an association between Member class and Announcement Class, in which one member posts zero or more announcement.

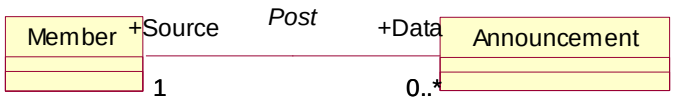


**Figure 5.3: Class Association Example**

Aggregation is a special kind of association. Aggregation associates one class to another with a "part-of" relationship. In figure 5.4, a Discussion class contains zero or more Discussion Message class (question blocks).

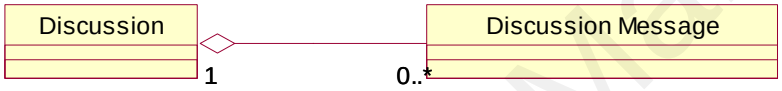


**Figure 5.4: Class Aggregation Example**

Putting all these together, a class diagram consisting of classes and their relationships therefore provide a static design view of the system.

## 5.3 Class Diagrams

This section presents the static design view of WebGE by means of class diagram.

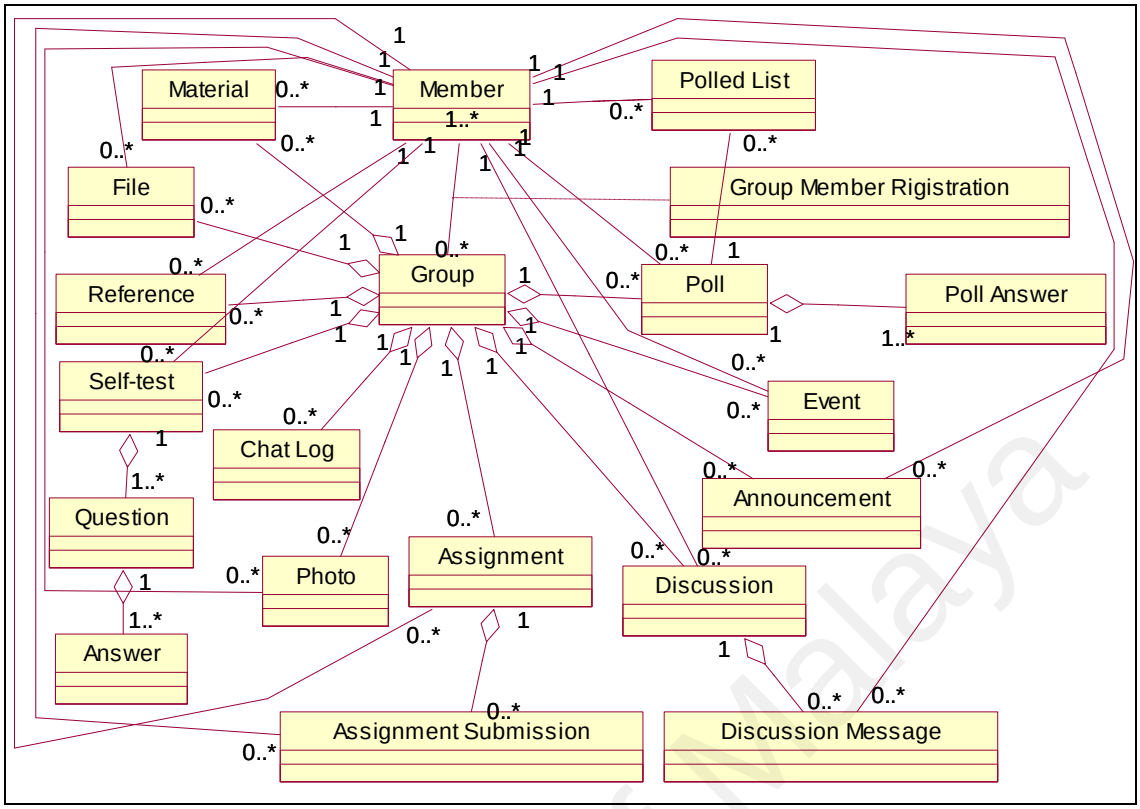


**Figure 5.5: WebGE Class Diagram**

The above class diagram is examined part by part as follows:

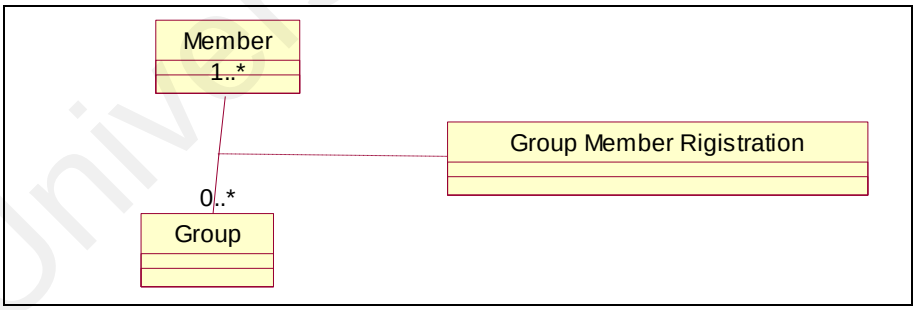


**Figure 5.6: WebGE class diagram (Part 1)**

In part 1, the diagram shows that:

- One Member class is associated to zero or more Group class (A member can have any number of groups).

- One Group class is associated to one or more Member class (A group can have one or more members).
- An association class, namely Group Member Registration, describes the relationship between Member and Group class.

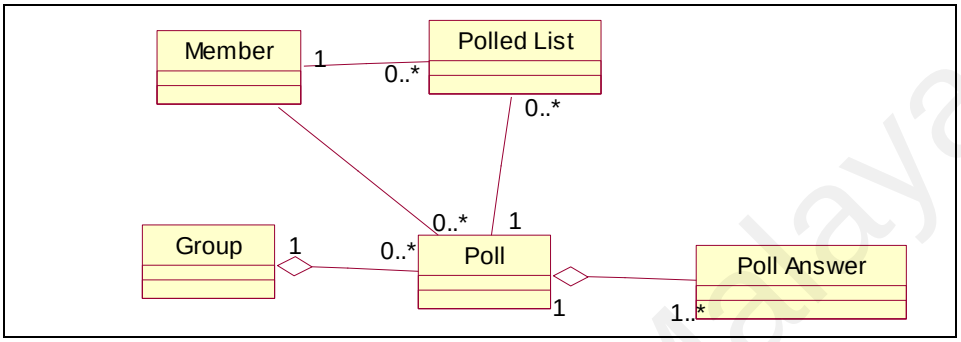


**Figure 5.7: WebGE class diagram (Part 2)**

In part 2, the diagram shows that:

- One Group class is associated to zero or more Poll class (A group can have any number of polls).
- A poll can have any number of poll answers and also any number of pulled lists. Polled list is used to keep a record of the members who has vote for the poll.
- A member can create any number of polls.

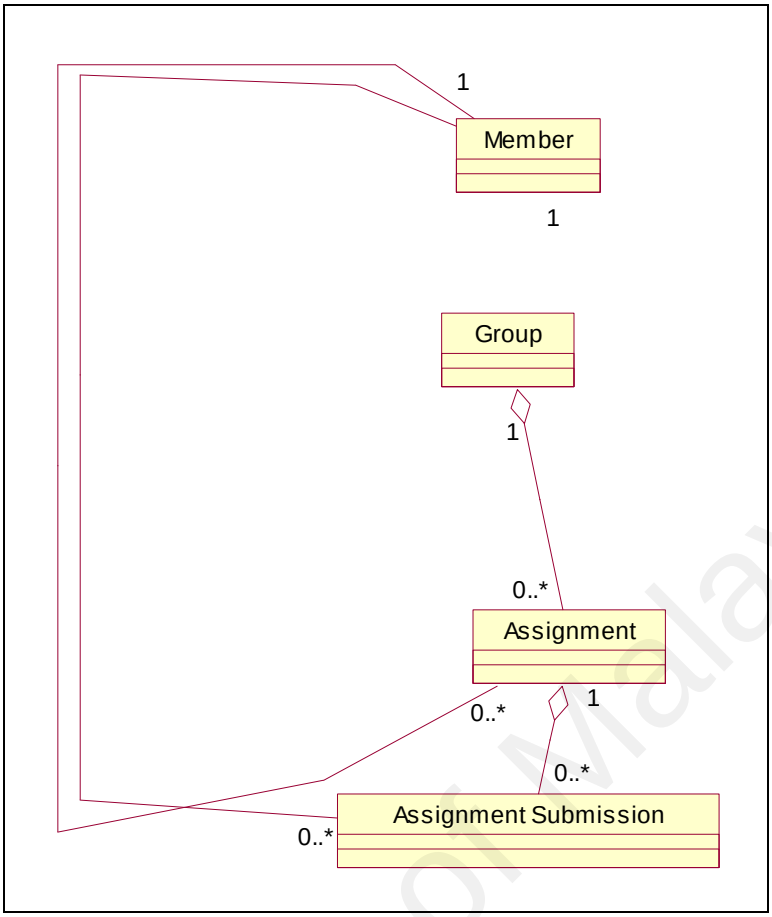


**Figure 5.8: WebGE class diagram (Part 3)**

In part 3, the diagram shows that:

- One Group class is associated to zero or more Assignment class (A group can have any number of assignment).
- A member (Course Instructor or assistant) can create any number of assignments.
- An assignment can have any number of assignment submissions. An assignment submission is an answer file which submitted by the student.

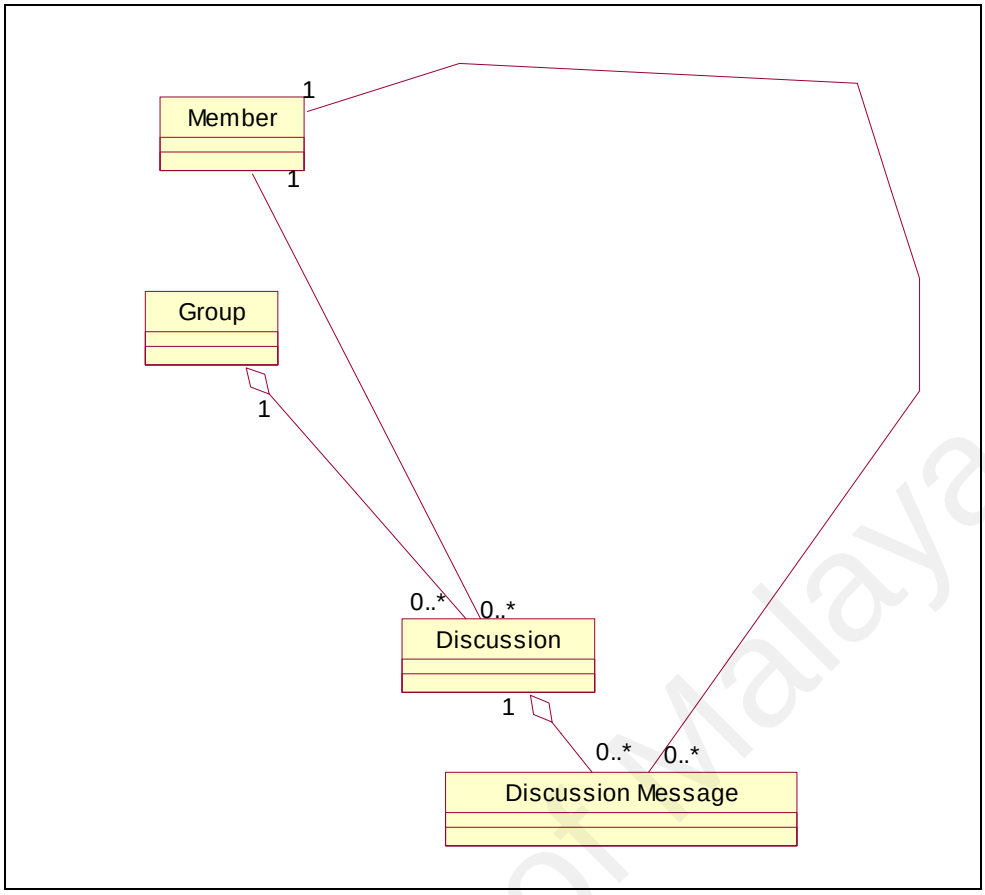


**Figure 5.9: WebGE class diagram (Part 4)**

In part 4, the diagram shows that:

- One Group class is associated to zero or more Discussion class (A group can have any number of discussion).
- A discussion can have any number of discussion messages.
- A member can create any number of discussions and discussion messages.

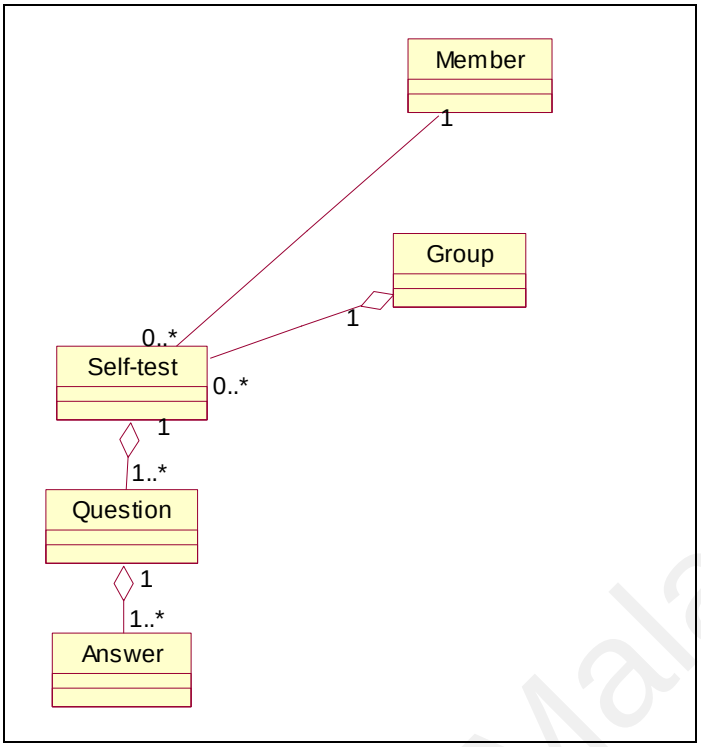


**Figure 5.10: WebGE class diagram (Part 5)**

In part 5, the diagram shows that:

- One Group class is associated to zero or more Self-test class (A group can have any number of self-tests).
- A self-test can have any number of questions.
- A question can have any number of answers
- A member (Course Instructor or assistant) can create any number of self-test.

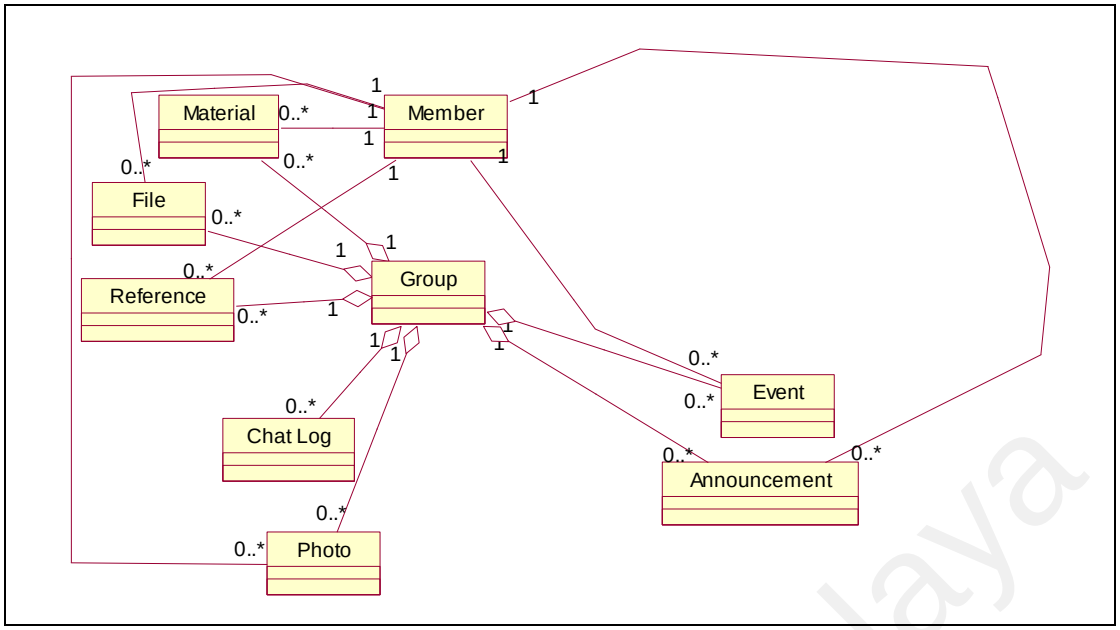


**Figure 5.11: WebGE class diagram (Part 6)**

In part 6, the diagram shows that:

- One Group class is associated to zero or more material, file, reference, chat log, photo, event and announcement classes (A group can have any number of material, file, reference, chat log, photo, event and announcement).
- A member can create any number of material, file, reference, photo, event and announcement.

## 5.4 Database Design

Most classes can be mapped directly into tables in database. Table 5.1 shows all tables found in the WebGE database.

**Table 5.1: WebGE Database Design - Tables**

| Table | Description |
|---|---|
| TblAdmin | Administration setting |
| TblAnnouncement | Announcement information |
| TblAssignment | Course assignment information |
| TblASubmission | Submitted answer files in which each file corresponds to an assignment |
| TblChat | Temporary chat messages |
| TblChatMsgLog | Chat message log |
| TblChatUser | Temporary chat users |
| TblDiscussion | Discussion topic information |
| TblDiscussionContent | Discussion Messages |
| TblEvent | Event information |
| TblFile | Files information |
| TblGroup | Group information |
| TblGroupReg | Group Member Registration |
| TblInvite | List of invitation to join group |
| TblMaterial | Course Material information |
| TblMember | Member information |
| TblNoteboard | Short note information |
| TblPhoto | Photo information |
| TblPoll | Poll information |
| TblPollAnswer | Poll Answer information |
| TblPolled | Polled list |

| Table | Description |
|---|---|
| TblQQAnswer | Self-test question answers |
| TblQQuestion | Self-test question |
| TblQuiz | Self-test information |
| TblReferences | Reference information |
| TblTempStudentAnswer | Temporary student result of a self-test |
| TblWaitApproval | List of request to join Student Group |

The specifications of all database tables in WebGE are documented at the appendix section. The fields in each table represent the attributes of the corresponding class in the class diagram as well. A primary key field in a table is shown with the abbreviation "PK" under the Key column while a foreign key field is shown with "FK". The data type for each field is based on the data types of Microsoft SQL Server 2000.

## 5.5 User Interface Design

The usability of the WebGE depends heavily on the user interface design. A good user interface design will reduce the time to learn to understand and use the system. It will also motivate the users as tasks can be performed faster with less error. Lastly, a good user interface design also enables users to remember the usage of the system even after some time.

Several principles are followed when designing user interface:

- The screens are simple without much clutter.
- Though simple, the design is still attractive.
- All pages require minimum download time.
- Certain elements are available consistently on all pages, such as page banner.
- The pages must be compatible with Internet Explorer version 5.0 and higher (most users are using this browser).
- The user interface should be goal-oriented for users.
- The screen should suit a variety of screen resolution.

The following figures show a few screenshots of WebGE.


**Figure 5.12: Screenshot of "WebGE Log In" page**

Figure 5.12 shows the WebGE log in page which is also the first page users would encounter when navigating the system. The log in page is simple with just a log in form and three other hyperlinks. Use of graphics is minimized in order to reduce the download time. Yet, it is made attractive through the use of colours. Upon loading, the cursor directly goes into the username box so minimize typing and clicking.

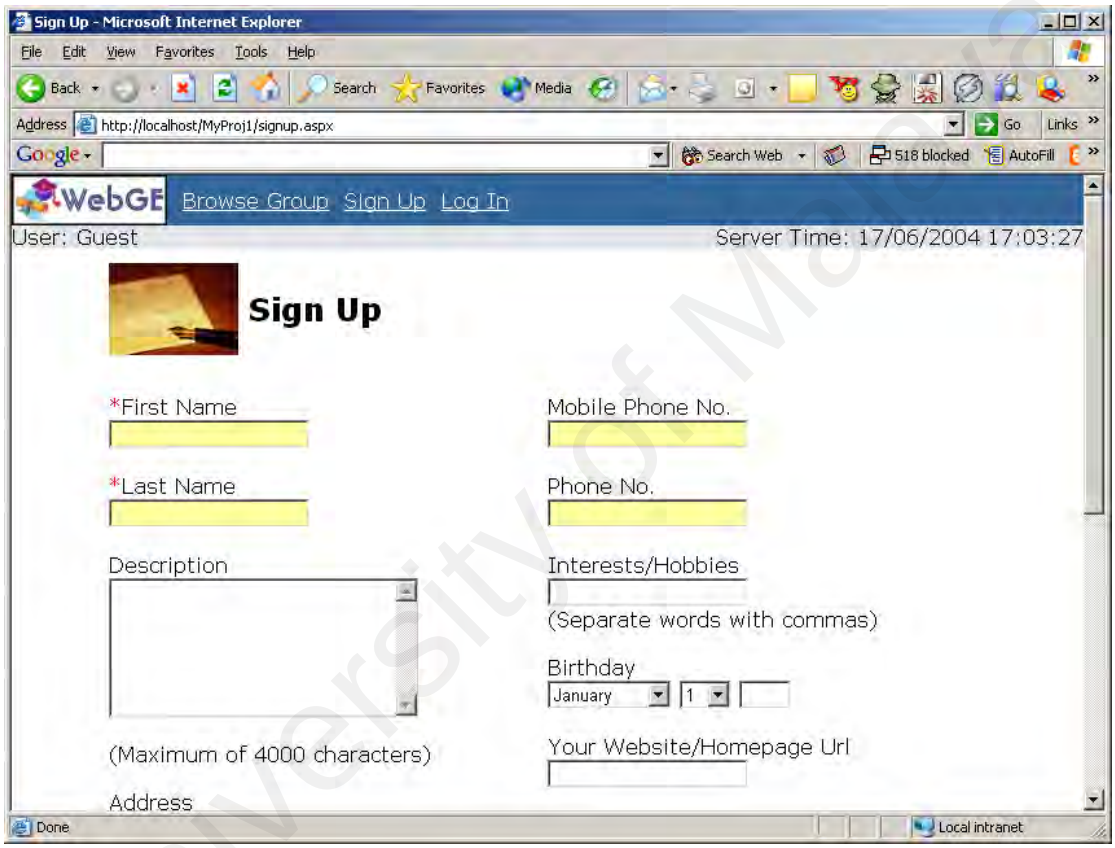


**Figure 5.13: Screenshot of "Sign Up Membership" page**

With the same principles followed, the membership registration page is simple yet attractive.

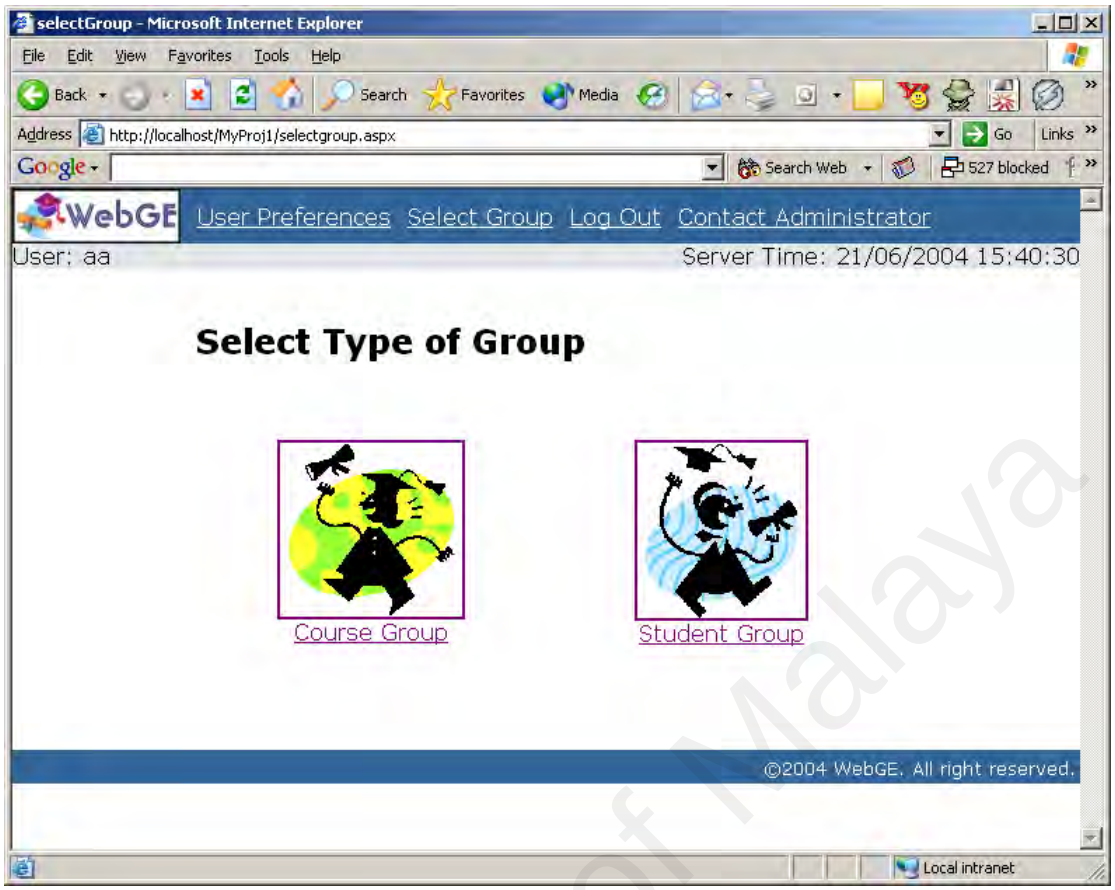


**Figure 5.14: Screenshot of "Select Type of Group" page**

Figure 5.14 shows the "Select Type of Group" page. The user can either click on the image or the text in order to enter the desired page.

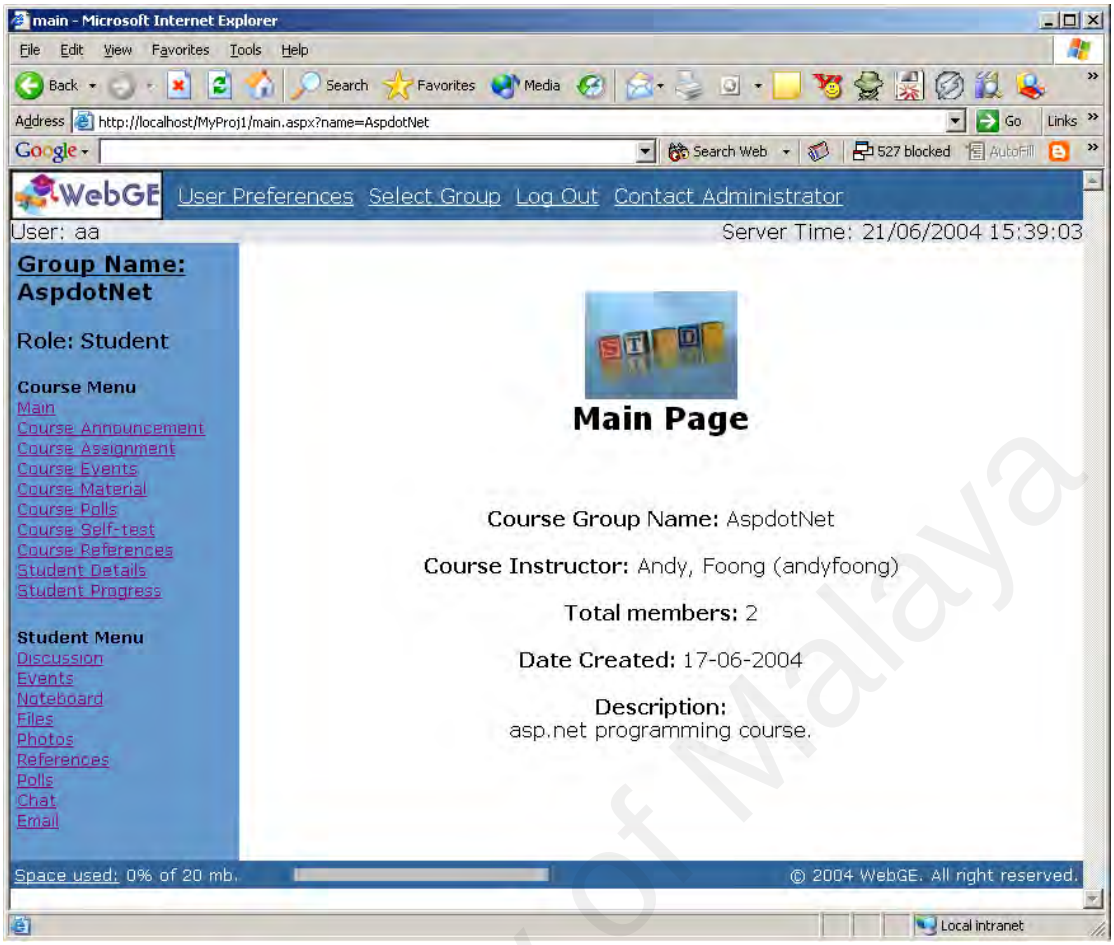


**Figure 5.15: Screenshot of Course Group main page**

Figure 5.15 shows the interface of Course Group main page. The left menu and top header are available at all pages of the group. The available features at the left menu are divided to two main parts, namely course menu and student menu. The course menu shows the links to the more formal type of information. In this case, only the Course Instructor and course assistant have the authority to post data to these pages. However, the students are allowed to view these pages and also submit answer files to the assignment page. The pages under the student menu enable students to post data and also to communicate with each other.

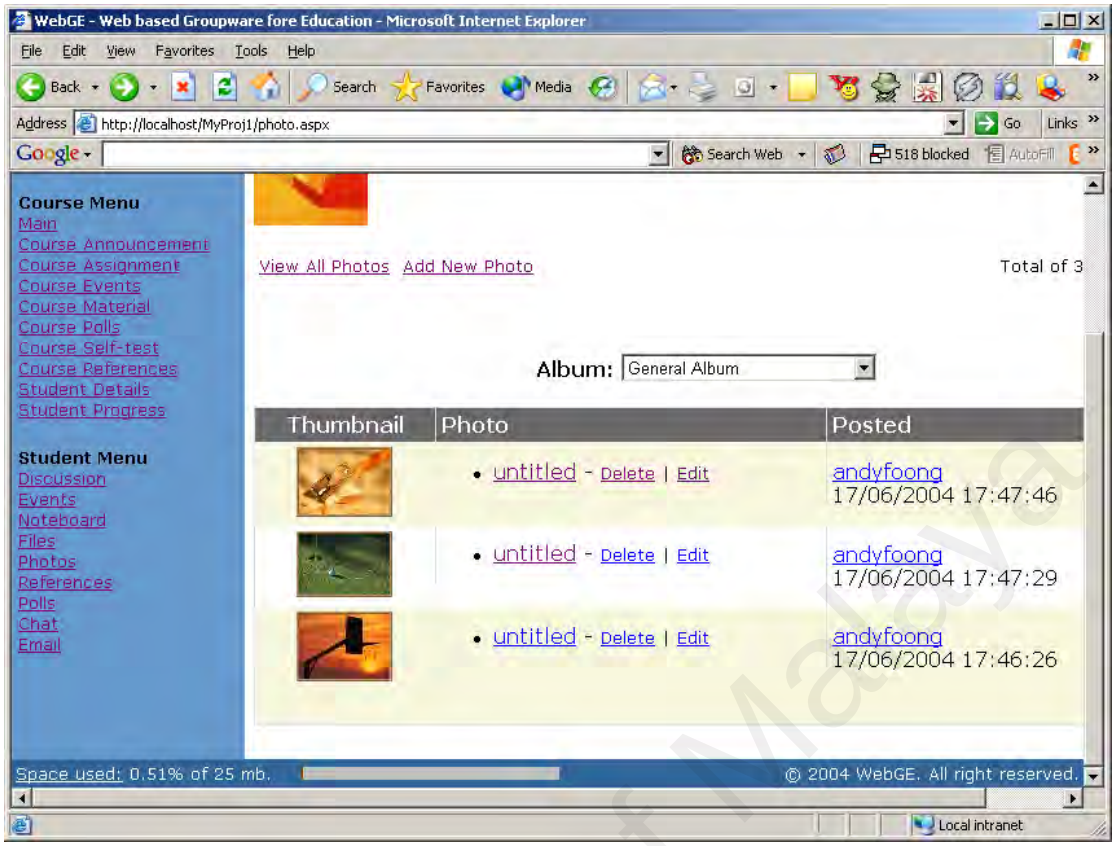


**Figure 5.16: Screenshot of view photo page**

Figure 5.16 shows the view photo page. As shown in this screenshot, a thumbnail is generated for each posted photo. The user can view this photo by clicking on the thumbnail or performs deletion and editing tasks by clicking on the corresponding text. The information of who posted the data is displayed at every record and there is always a link to view the details of that person.

At every page within a group, the same footer is displayed. The footer contains a horizontal bar showing the space that has been used for the group.

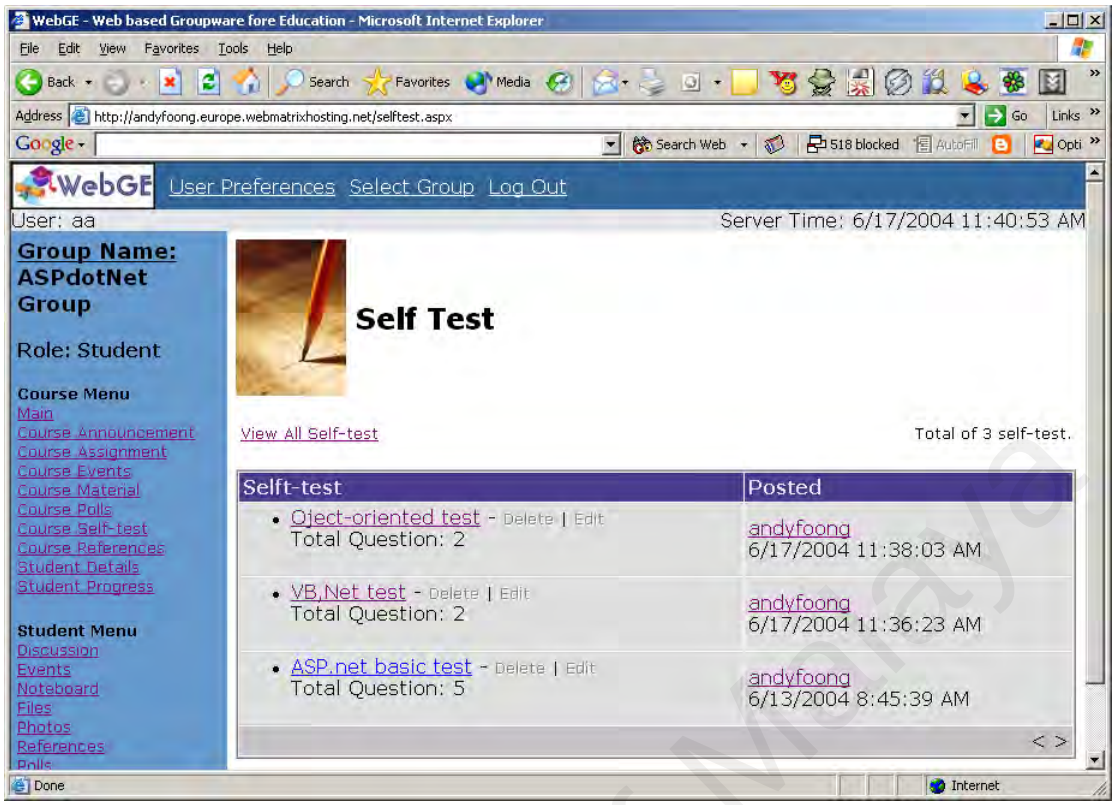


**Figure 5.17: Screenshot of self-test page**

Table 5.17 shows all the self-test records for a group. All records in WebGE are presented in table form. Each table shows up to maximum 10 records per page. If the total records exceed 10 records, the system will automatically generate pages for the records. Navigation arrow or navigation page number is shown at the bottom of each table in order to navigate to the desired record. Apart form that, each record has a hyperlink which enables user to access details information of that particular record.

# Chapter 6 - System Implementation

## 6.1 Introduction

The Construction phase in Unified Process involves growing a system into production system. Traditionally this has been called the Implementation stage. In this stage, all requirements as specified in use cases are realized through design and coding the system. This chapter is concerned with the realization of these requirements into a final system.

## 6.2 Implementation Tools

This section lists various software tools used in actually implementing the system.

The web server that runs WebGE system is using:

- Windows XP
- Internet Information Services (IIS) 5.1 as web server
- Microsoft SQL Server 2000 Developer Edition

The development of the system makes use of:

- MS ASP.Net with VB.Net as programming language
- MS Visual Studio 2003 as the development tool.
- Microsoft SQL Server 2000 as database system

- JavaScript as client-side scripting language
- MS SQL Enterprise Manager as graphical database interface tool

## 6.3 Approach of System Development

The implementation of WebGE was carried out by following the modular and incremental object-oriented approach. It was modular in the way that the system was built based on the use case packages described in Chapter 4. All related objects and functions were grouped into a use case package and the system was then implemented incrementally - completing one use case package and then moving to another. Within a use case package, use cases were implemented by using the incremental approach as well, where a use case was implemented and then followed by another one, until all the use cases of the package was completed.

In order to implement a single use case, all necessary objects were created. As explained in Chapter 4 and 5, these objects include boundary objects, control objects and entity objects. The following are the general steps that had been taken in implementing a single use case in the WebGE project:

(a) Create and design all the conceivable WebGE web pages. The linking among these pages has to be considered in this process. Besides, the database specification is referred in order to clarify the data needed in realizing the use case. For example, the database specification is referred in order to create the sign up page.

(b) Create and code all the entity objects. For example, in the Create Student Group use case, the entity needed is the Group object. Basically all entity objects have at least three basic functions, which are add, update and delete.

(c) After creating all the necessary user interface objects (or better known as form objects in Visual Basic) and the entity objects, control object is created as a liaison between the former two. The actual logic of the use case is implemented through the control object.

(d) Code the linking between the form objects, control object and entity objects. This includes all the object creation procedure and also the function and subroutine calls. The use case implementation should be completed by now.

(e) Test the initially completed use case. Debug the codes if errors are detected, by using step-through debugging (will be explained later in Chapter 7).

(f) Correct the errors found.

(g) Step (e) and (f) are repeated until the codes are bugs free.

After all use cases of a use case package was completed, the whole package was tested again to check for undetected errors. For example, in the Group Registration package, the Invite People to Join Group use case and View Registered Group use case must be tested together again, to ensure that when a person is invited, the invited group would appear in the View Registered Group page.

Finally, when all the use case packages were completely implemented, they were retested as a whole system in system testing. The testing phases are elaborated in Chapter 7.

## 6.4 Implementing the Program with MS ASP.Net and VB.Net

Since the object-oriented approach is chosen as the development methodology for the WebGE project, MS ASP.Net with Visual Basic .NET (or VB .NET), which is a fully object-oriented language, is a very suitable language for the purpose. Besides, the Visual Studio .NET IDE also provides a fast mechanism in designing appealing and usable web pages. This section would elaborate on the programming styles, features and techniques that were used by the author in developing WebGE.

### 6.4.1 Programming Conventions

Before the programming work of the WebGE got started, a set of programming conventions were listed out first, so that these conventions can be used to facilitate a more systematic way of programming. It also ensures that program maintenance can be done with greater ease when bugs were detected or when enhancements were made to the current code. Most of the programming conventions focus on naming the variables and user interface controls, as well as subroutines and functions. For instance, a textbox is always named with a prefix of 'txt' while a label is always named with a prefix of 'lbl'.

### 6.4.2 Program Commenting

As usual, comments are added into program codes, so that the codes can be understood easily. In VB .NET, a line of program comment is written beginning with the sign of quote (') and the program comment is highlighted in green. Besides writing comments in between codes, the author also included a comment box at the

beginning of each module (such as class or form). The comment box basically gives a description of the name of the component, what does the component do, when is the component created, revisions made to the component, and also input and output parameters of a subroutine of function. Furthermore, all the classes were registered with the namespace "WebGE" in order to differentiate from the build-in .Net class library. Figure 6.1 shows a comment box of the Admin Class.

```
'*****************************************************************
'  Class Name  : admin
'  FileName  : Admin.vb
'  Descriptions : Contain all functions  related to administration
'  Create DateTime  : 31 May 2004 05:34PM
'  Revisions  :
'  Date Time -  Class/Function/Sub    -   Descriptions
'
'*****************************************************************
```

**Figure 6.1: Sample comment box of Admin class**

### 6.4.3 Built-in References in Visual Basic .NET

Visual Basic .NET provides a rich library of object references that can be reused by programmers, so that time is saved in developing software. In order to use these references, they must be included first under the Project -> Add Reference menu item. Some of the built-in references that were used in developing the WebGE were the System.Data object, which comprises of functions and attributes that deal with the operations of database, and also the Scripting object that deal with folder and file manipulation.

### 6.4.4 Working with Database

Since the WebGE would be using the Microsoft SQL Server 2000 as its database management system, it can therefore harness the power of the SQLClient ADO .NET object, which is a completely new database access object that is introduced in VB .NET. It uses the .NET Framework native data provider as a mean to access the database, eliminating the need to go though the OLE DB layer of the operating system, which is often practiced in the programming languages before this. By using the native data provider, data access is faster and more reliable.

In the WebGE project, all records in database are accessed using stored procedure which located at SQL Server. There is no TSQL command in the VB programming codes. There are several advantages of doing so, such as stored procedure is precompiled and faster, increase program modularity, improve database security, reduce network traffic and so forth.

### 6.4.5 Shared Functions

A class named Tools was created in this project, which it is used to store the shared and general functions/subroutines that used by other classes. For example, the "sendEmail" function, "sendGroupReport" function, "formatDate" function and so on. These functions were used throughout the whole application, eliminating the hassles of rewriting the data access codes again and again when they are needed.

### 6.4.6 Constructor Overloading

Constructor in VB .NET is implemented by using the keyword New and one of the points that make VB .NET different from its predecessor is that the constructor in VB

.NET can be parameterized and overloaded, which suggests why it is a fully object oriented language.

In the WebGE project, constructors for entity objects were normally overloaded because of the nature of the database operation. For instance, an INSERT operation would need all the attributes of an object to be initialized with values while a DELETE statement would only need the primary key attribute to be initialized. The following constructor declarations demonstrate overloaded constructors of the AP object.

```
Public Sub New(ByVal sinvoiceID As String, ByVal siAmountPaid As Single, Optional ByVal cStatus As Char = "N")

Public Sub New(ByVal sinvoiceID As String)
```

### 6.4.7 Inheritance

Inheritance is another strong point of VB .NET compared to its predecessor. Although it is not widely used in the WebGE project, it is however implemented in the Group package, involving the CourseGroup and StudentGroup class. The following statement shows how the StudentGroup class is declared to inherit from the AbstractGroup class.

```
Public Class StudentGroup : Inherits AbstractGroup
```

### 6.4.8 Polymorphism

Another feature introduced in VB .NET is the polymorphism feature. The two following function declarations of function “GetGroupDetails” demonstrate how

polymorphism is implemented in VB.NET. Both function declarations bear the same names, but with different parameters.

```
Public Overloads Shared Function GetGroupDetails(ByVal GroupName As String) As
DataTable
```

```
Public Overloads Shared Function GetGroupDetails(ByVal GID As Long) As DataTable
```

## 6.5 Deploying the WebGE

After the implementation work had completed, the WebGE would have to be deployed on the web server. There are two main files which need to be deployed to the web server, which are the database files and the web application files.

The deployment of the web application files is rather easy – just copy and paste the files (.aspx and .dll files) to a virtual directory. However, to increase the simplicity of deployment process, an executable installation package is produced using MS Visual Studio. The executable installation file is then executed at the desired web server. Apart from that, the deployment process also can be done via FTP (File Transfer Protocol) service, if the server allows web site uploading via FTP.

In order to deploy the database to the database server, the database server needs to execute a TSQL scripts. All the database tables, stored procedures and views should be created after running the script.

After both the database and application files were deployed, a connection was made between the database server and the web application, and then the web application was tested. The deployment work was considered done when everything worked properly.

# Chapter 7 – Testing

## 7.1 Introduction

As like other software, the WebGE would have to be tested rigorously in order to make sure that it fulfil the functional and non-functional requirements as well as it is bugs free. In the testing phase of the WebGE project, testing was carried out in four levels – unit, integration, system, and user acceptance test. Unit and integration testing are white box testing techniques and they are normally carried out concurrently with the implementation of the system. When any bugs were detected during the unit and integration testing, an inspection and debugging (debugging would be explained later in Section 7.5) of the code had to be done. System and user acceptance testing on the other hand are black box testing techniques, where the testing is only concerned with whether the application has fulfilled the functional and non-functional requirements as planned. When system testing uncovers any bugs or errors, unit testing and integration testing must be rerun so that the bugs can be eliminated. Lastly, the user acceptance test is concerned with system usability and also to evaluate whether the system has fulfilled the users needs and expectation.

## 7.2 Unit Testing

Unit testing is the 'deepest' among the three testing levels. It tests individual functions, subroutines and program blocks for errors. In the WebGE project, the unit testing was conducted to point out two types of errors – logic error and syntax error.

### 7.2.1 Logic Error

If the software has compiled and run successfully but the wrong results are produced, the software most probably has logic errors. Logic error is normally caused by wrong processing steps in a function or subroutine, hence producing the wrong output. The following shows a few causes to logic error.

- Forgetting a branch condition in an If-Else or Select Case statement.
- Calculating the wrong number of loop that should be taken in a "For" or "While" loop.
- Miscalculating the number of elements that an array can hold, causing overflow problem to the array.
- Extracting the wrong substring from a string.

Listed below are the steps taken in uncovering logic errors:

(a) Input variety of data combination to the function or subroutine and view the output. Check to see if the output is the desired output. If it is not, use step through debugging to find out the problem.

(b) For functions and subroutines that involve getting data from database, manually select the data from the database and compare the result with the outcome of the program.

(c) Test every branch condition in an If-Else block and determine whether statements for each branch perform correctly.

(d) Try various inputs on an If-Else block and check whether each input can be met by a branch in the If-Else block. This is to test whether all branch conditions have been considered.

(e) For a loop structure, make the loop to execute only once and also to execute a few times. Check to see if the output is correct. In many cases, error seems to occur when the loop only executes once.

(f) Step through the execution of the function or subroutine and examine how the value of each variable changes. Check whether the values of the variables change in an expected manner or the opposite.

(g) If the logic error is found at the stored procedure, debugging can be done using the MS SQL Server 2000 Query Analyzer. In this case, the same steps, which listed above, are used to debug the stored procedure.

### 7.2.2 Syntax Error

Syntax errors normally would not occur during run time because the compiler would prompt the user if there are errors while compiling the program. However there are certain syntax errors which are 'invisible' to the compiler. One of them is the syntax error occurred when calling the stored procedure. In this case, syntax error happens when the parameter passed to the stored procedure is not the desired types or not the desired name. However, such syntax errors are always displayed at the page by the ASP.Net engine. Therefore, it is rather easy to uncover such errors – simply compare the input parameter in VB.Net code with the stored procedure parameter.

## 7.3 Integration Testing

Integration testing involves the testing of interfaces between functions or subroutines. A function or subroutine might perform correctly after unit testing but when it is called by another function or subroutine, error might occur, most probably because of

the faults in the interface of the function or subroutine. A common integration error is when the variable passed into the function does not match the data type specified in the parameter of the callee function. Even if the compiler can accept, problem with casting might occur, causing precision error. Therefore integration testing is an important testing procedure in the WebGE project.

The approach of integration testing used in the WebGE project is the bottom-up approach where functions and subroutines at the lowest level are individually tested first, and then the testing would proceed with the combination of functions and subroutines at the higher level with the lowest level ones. The testing would keep moving up the hierarchy until all the functions are tested as a whole. For instance, let's assume that Function A calls Function B, and Function B calls Function C. The bottom-up approach would test C first; then, test B and C together; and finally, test A, B and C together.

Listed below are some of the elements that are tested in integration testing:

- Compatibility between the parameters data type of a function and the values that it receives.
- Compatibility between the output data type of a function and the receiving variable of the caller function.
- All possible outputs of the callee function should be captured by the caller function. For example, if the callee function were to return a character of either 'A', 'B' and 'C', then the caller function must contain codes that deal with all these three characters.

A noteworthy point that should be pointed out here is that unit testing and integration testing were actually conducted concurrently in the WebGE project as it's almost impossible to test a single function or subroutine alone because a function or subroutine would normally depend on input or output from another function and subroutine. So normally a few functions and subroutines that were related to each other were tested concurrently in order to produce the desired test results.

## 7.4 System Testing

After the unit testing and integration testing had completed and all the initial bugs had been eliminated, the WebGE was tested as a whole. The main focus was to ensure that all functional and non-functional requirements were fulfilled, besides uncovering any undetected bugs. As mentioned earlier, the unit and integration testing were repeated when new bugs were detected. Two types of system testing, namely the functional testing and installation testing, were conducted for the WebGE project. The functional testing was mostly conducted based on a test script, which is presented below.

### 7.4.1 Functional Testing

As its names suggest, the functional testing focused on functional side of the system. In simple words, it simply tests whether the system had fulfilled the functional requirements correctly. The use cases specified in 4.1.1.1 were referred to during the functional testing so that no use case would be left out from the system.

In this study, the functional testing was conducted in a systematic way based on carefully designed test scripts. The test script is shown at the tables below. However,

not all functions or use cases are included in this test script due to space constraint. Test script for other use cases would follow the styles of these test script.

**Table 7.1: Test script**

| No | Use Case | Expected Result | Result |
|---|---|---|---|
| 1 | ▪ Login to administrator account | Login successfully and the login process should not exceed 2 second. | Successful |
| 2 | ▪ Change group configuration | The group configuration setting is changed at the database | Successful |
| 3 | ▪ Create Course Group | Course Group is created and invitation emails were sent to the invited people. | Successful |
| 4 | ▪ Sign up membership | User account is created. The member details are identical with the information entered by the user. | Successful |
| 5 | ▪ Create Student Group | Student Group is created with the setting entered by the member. | Successful |

| No | Use Case | Expected Result | Result |
|---|---|---|---|
| 6 | ▪ View Registered Group | All the member group, invited group and pending group are shown. | Successful |
| 7 | ▪ Join Group | The group registration record is inserted to the database | Successful |
| 8 | ▪ Add course announcement | Only the Course Instructor or assistant can perform this task and the record is inserted without error. | Successful |
| 9 | ▪ Edit course announcement | Only the person who posted the data can perform this task and the record is updated. | Successful |
| 10 | ▪ Delete course announcement | Only the Course Instructor or assistant can perform this task and the record is deleted. | Successful |
| 11 | ▪ Bar member from entering group | Only the instructor, assistant, founder and manager can perform | The setting is changed but the corresponding member still able to |

| No | Use Case | Expected Result | Result |
|---|---|---|---|
| | | this task. After performing this task, the corresponding member will not able to enter the group. | enter the group<br>Action taken:<br>Codes are debugged and system performed as expected after debugging. |
| 12 | ▪ Student submit assignment answer file | The file is uploaded successfully and the Course Instructor or assistant is able to download it. | Successful |
| 13 | ▪ Create self-test | The self-test is created successfully and can be use by the students | Successful |
| 14 | ▪ Chat | Chatting between people is performed successfully and a chat log is recorded. | Chat is performed successfully but no chat log is recorded<br>Action taken:<br>Codes are debugged and system performed as expected after debugging. |
| 15 | ▪ Change Course Group functions availability | The setting is changed successfully. | Successful |

| No | Use Case | Expected Result | Result |
|---|---|---|---|
| | setting | | |
| 16 | ▪ Delete a group | The group record and any other records within the group are deleted. | Successful |
| 17 | ▪ Email group report | Group report is received at the member email account | Successful |
| 18 | ▪ View group used space | The used space is display in correct format and figure. | Successful |
| 19 | ▪ Request approval to join Student Group | The record is added. An email is sent to the group founder. The record is displayed at the request page. | Successful |
| 20 | ▪ Upload photo | Thumbnail is generated and displayed at the appropriate format. Uploaded photo is viewable. | Successful |
| 21 | ▪ Create Poll | Poll is created successfully and is | Successful |

| No | Use Case | Expected Result | Result |
|---|---|---|---|
| | | categorised according to its start and closed date. | |

### 7.4.2 Installation Testing

The installation testing focused on whether the system had been deployed successfully on the production site. Some of the tests conducted were:

- Database connection test
- Functional test, which include add, delete and update a record.
- Network performance test.

The main goal of the installation test is to ensure that the system works perfectly on the production site, just like how it works on the development site. In this study, the installation test was conducted at the http://europe.webmatrixhosting.net web server. The files were uploaded via FTP while the database was created using MS SQL 2000 Query Analyzer.

## 7.5 Debugging

Debugging is relatively easy in Visual Basic .NET, compared to other languages, thanks to the step through debugging feature provided in the Visual Studio .NET IDE. By using step through debugging, the execution of the program can be seen from statement to statement and what is more useful is user has the full control of moving the execution from one statement to the next statement. Whenever a statement seems

suspicious, the user can use the intermediate window to check the value of all the variables at the point of execution. This is done by using the command of a question mark (?) follow by the variable name in the intermediate window.

In the WebGE project, whenever a run-time error occurred, an error page is displayed, giving information on the source and description of the error as well as the module and submodule where the error occurred. So a debugging point would be put on the beginning of the submodule and the program is executed again. Upon reaching the debugging point, the program execution would stop, and the codes were than inspected line by line until the suspicious code was reached. The intermediate window can be used when necessary in order to ensure the correct value is assigned to the particular variable. If the debugging couldn't detect the fault, the debugging point would be moved to a point before the submodule, which normally was the caller function. This is because some errors happen when the values brought into a function from the caller function are wrong.

From the experience of developing the WebGE, all bugs were successfully eliminated by using the step-through debugging method.

## 7.6 User Acceptance Test

After all the detected bugs had been fixed and the final release of system had been uploaded into the server, the final test phase – user acceptance test was carried out. The purpose of user acceptance test is to enable the potential users to evaluate and determine whether the developed system meets their needs and expectations. As for

WebGE, a user evaluation form was generated and a group of master students which consists of 10 respondents was chosen to conduct the user acceptance test.

The evaluation form used in this test is divided into two main parts, which are system evaluation and user interface evaluation. The main purpose of the system evaluation is to test the system usability. In this part, the tester requires to rank the usability of the WebGE's functionalities according to the usability criteria of "efficient to use", "easy to remember how to use", "easy to learn", "safe to use", and "have good utility". While the first part of evaluation form is concern with the system functionality, the second part of user evaluation form is focused on the non-functional requirements, such as user interface design and performance. The user evaluation form is available at the appendix section in this thesis.

After all the questionnaires have been collected back from the testers, the results were analysed and is summarized at Table 7.2 below.

**Table 7.2: Summary of the results obtained from User Acceptance Test**

| | **Number of elements which were ranked as:** | | | | |
|---|---|---|---|---|---|
| | **1 – Very Poor** | **2 - Poor** | **3 - Moderate** | **4 - Good** | **5 – Very Good** |
| **System Features** | 3 | 26 | 91 | 102 | 28 |
| **User Interface Elements** | 2 | 8 | 40 | 59 | 11 |

According to the results obtained for the first part of questionnaire, there are average 22 out of 25 system features were ranked as moderate, good and very good. This also equal to 88% of the system features have met the testers' satisfactory. As for the user interface test, the result is slightly better. Average of 92% of the user interface elements were ranks as moderate, good and very good.

During the testing, some ideas of improvement were suggested by the testers. Basically, the suggestions can be divided into two main parts – functionalities and design. As for functionalities, a tester asked for the shared whiteboard feature and another tester ask for the voice and video conferencing feature. While the author replied that such features are complicated and people rarely use such features, the testers agreed upon. Apart from that, a tester suggested that the system should enable the users to change the page colours and layout according to their preferences. The tester said that the users might be bored with the same user interface at all the times.

In a nutshell, testers were satisfied with most (>88%) of the system features and user interface design. Some suggestions were given by the tester and such information was noted and documented in the Section 8.3 – Future enhancement of chapter 8.

# Chapter 8 – Discussion and Conclusion

Every software product generally has its strong and weak points. The WebGE is no exception either. This chapter would elaborate on the strengths and weaknesses of the WebGE as well as the future enhancements that can be made to the system, and finally the findings and knowledge learned from this study is given in the Conclusion Section.

## 8.1 System Strengths

There are two aspects to the strengths of the system, namely the product aspect and the process aspect.

The product aspect deals with the capabilities of the system. Listed below are some strengths and competitive advantages of WebGE as compared to other similar systems such as WebCT and AulaNet:

- WebGE permits two types of group to be created, namely Course Group and Student Group. The Course Group is more formal type of group which is created for a particular course whereas the Student Group is less formal and can be created by any member. This enables a small group within a course to form their own group and thus information is only dispersed among the corresponding group members. Furthermore, other peoples who are not taking the same course also can join the group.
- The available features are separated into the course features and student features. Thus, the students (end users) can differentiate between the official

information which posted by the Course Instructor and the unofficial one which posted by all members.

- WebGE has most of the needed features for education, such as assignments posting, assignment submission, discussion, photo sharing, chatting, self-test, polling, student progress and so forth. Some features such as polling and photo sharing are not provided in other system such as WebCT and AulaNet.
- All posted data (such as files, photos, records, etc.) are stored in MS SQL Server. This ensures security by forbidding unauthorised access to these data.
- WebGE is incorporated with email features. The system enables group activities report to be sent via email to the corresponding group members. Besides, invitation letter and error report can also be sent via email to the intended person.
- WebGE is a secure system. With a strong authentication system, only valid user can login to the system and login to the registered group. The different roles of members in each group also differentiate the authorities given to the member, such as the permissions to add, edit or delete a record.
- WebGE is developed using Microsoft ASP.Net and VB.Net. In this case, it is compiled and run faster if compare to other scripting language such as PHP and ASP.
- The online chatting feature is developed using ASP.Net and the output to the client side is purely HTML. Therefore, unlike other chat room that developed using Java or ActiveX, the client side need not to install the chat program (some users might not have the permission to install a program into certain computer) and the chat feature can be used it directly.

The process aspect deals with the system development process and future maintenance. A good process helps build a good product. A few pluses for the process to develop WebGE are as below:

- The adoption of Unified Process (UP), instead of waterfall approach, reflects the iterative nature of system development in the real world. Various aspects of UP, though not all, have been incorporated into the development to build a high quality product.
- The use of Unified Modeling Language (UML) to specify, visualize, construct and document the system has helped achieved simplicity, clarity and vision in developing the product.
- The system has been developed in such a way that software reuse is taken care of. Many class modules in the system, such as Member class, Group class and Admin class, are reusable in development of other system.
- The system has been developed in such a way that high maintainability is achievable. Due to the object-oriented approach, modifying the system to accommodate new requirements and future enhancements can be made relatively easily.

## 8.2 System Limitations

However, there are some limitations of the system as listed below. They have not been taken care of mainly due to limited resources and time constraints.

- WebGE does not provide shared whiteboard and voice conferencing feature. Such features are complicated and most likely should be developed in another system.

- WebGE only provides the feature to send email but does not capable to receive email. The development of email receiving feature is complex and it requires the configuration of POP3 server. Such feature should be in a standalone email system.
- WebGE is dependant on MS SQL Server 2000. In this case, additional charge is needed for database hosting. However, by the means of using MS SQL Server, the database is faster and more secure if compare to Access database.
- Although WebGE enable users to configure the available features in the Course Group page, it does not provide feature to change the page layout and colour setting.
- WebGE does not support other languages such as Malay, Chinese, Japanese and so forth. This feature is usually supported by the commercial software which its clients are from all around the world.

## 8.3 Future Enhancements

The future enhancements of WebGE could be improving the system to eliminate the system limitations mentioned in earlier section. The future enhancements are listed as follows:

- More features could be provided, such as shared whiteboard and voice conferencing.
- Email system could be incorporated to enable email receiving feature. This could be achieved by integrating the product with third party software such as ActiveMail and devMail.Net, or integrating with the existing POP3 mail server.

- The system can be developed to support different choices of database system in order to let the user to decide which database system to be used according to their requirements and financial constraints.
- Template control could be provided in order to let user to decide the page layout according to his preference.
- The software could be translated to different languages (such as Malay and Chinese) in order to suit different users' need.
- More sophisticated statistical student activity report could be generated for the Course Instructor.

## 8.4 Conclusion

At present, web-based application is increasingly gaining interests from the public. This is due to the technology advancement which leads to the increasing access of the World-Wide-Web (WWW). In this case, web based groupware for education is an interesting topic because it tends to integrate the WWW and Collaborative Learning methods together as a tool to deliver online learning and also foster knowledge sharing.

The WebGE project has achieved its objectives by developing an integrated full-fledge web-based groupware for education. Although it is particularly design for education purpose, the features provided in this system also enables it to be used for social purpose, such as forming an alumni group and so on. By using the WebGE, online knowledge sharing is encouraged and the Course Instructor is able to monitor the information that shared among the students.

Throughout the development of the WebGE, a lot of knowledge has been gained. This include working with the processes of the Unified Process, designing the system with UML models, set up the Web Server, setting up MS SQL Server database, programming with Visual Basic .NET and also SQL programming. The successful implementation of the Unified Process as the development framework and the implementation tools (ASP.Net, SQL Server) also prove that the Unified Process and the implemented tools is effective and efficient to be used for software development.

As the final word, the WebGE project is a very interesting and beneficial project to work on.

# References


Allen, C. (1990). Definitions of groupware. *Applied Groupware*, 1, 1-2.

AulaNet. (2004). *AulaNet 2.0 - Student's Manual*. Available: http://www.eduweb.com.br/ingles/home.asp (6 May 2004).

Bahrami, Ali (1999). *Object Oriented Systems Development*. Boston: McGraw-Hill.

Blackboard, Inc. (2004). *Blackboard Official Website*. Available: http://www.blackboard.com (6 May 2004)

Booch, G., Rumbaugh, R. and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Boston: Addison Wesley.

Boud, D. and Feletti, G. (1991). *The challenge of problem-based learning*. London : Kogan Page.

Chaffey, D. (1998). *Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software*. Boston: Digital Press.

Coleman, D. and Khanna, R. (1995). *Groupware: Technology and Applications*. River, N.J: Prentice Hall, Inc.

Coleman, P. and Anderson, C. (2000). *Networking complete*, 2nd edn. San Francisco : Sybex. 322-323

Duch, B.J., Groh, S.E. and Allen, D.E. (2001). *The Power of Problem-based Learning*. Sterling, Virginia: Stylus Publishing.

Ellis, C.A., Gibbs, S.J. and Rein, G. (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1). 39-58.

Ensor, R., (1990). How can we make groupware practical? *Proc. SIGCHI '90., ACM*. 87-89.

Finley, M. (1995). Toward a definition of groupware. *Managing Office Technology*, 4 (10). 35-36.

Fuks, H., Gerosa, M.A. and Pereira de Lucena, C.J. (2002). Using a groupware technology to implement cooperative learning via the internet - a case study. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. 21 -29.

Gokhale, A.A. (1997). Collaborative Learning Enhances Critical Thinking. *Journal of Technology Education*. 7(1). Available: http://scholar.lib.vt.edu/ejournals/JTE/jte-v7n1/gokhale.jte-v7n1.html (5 May 2004).

Goldberg, M.W., Salari, S. and Swoboda, P. (1996). WWW - Course tool: An Environment for Building WWW-Based Courses. *Fifth International Worl Wide Web Conference (WWW5), Paris, France*. Available: http://www5conf.inria.fr/fich_html/papers/P29/Overview.html (1 March 2004).

Goldberg, M.W. (2000). *WebCT and Trend in Educational Technologies*. Available: http://webct.media.nagoya-u.ac.jp/OHP-2000-10-25/trendsInEdTech.files/frame.htm (1 March 2004).

Hills, M. (1997). *Intranet as Groupware*. New York: John Wiley and Sons, Inc.

Hsiao, J.W.D.L. (1999). *CSCL Theories*. Available: http://www.edb.utexas.edu/csclstudent/Dhsiao/theories.htm (6 May 2004).

Jacobson, I., Booch, G. and Rumbaugh, J. (1999). *The Unified Software Development Process.* Reading, MA: Addison Wesley.

Johansen, R. (1989). Groupwise and collaborative systems-a big picture view. *IEEE Global Telecommunications Conference, and Exhibition. GLOBECOM '89.* 1217-1220.

Johnson-Lenz, P. and Johnson-Lenz, T. (1998). Groupware: coining and defining it. *ACM SIGGROUP Bulletin*, 19(2), 34.

Johnson, D.W., Johnson, R.T. and Smith, K.A. (1991). *Active Learning: Cooperation in the College Classroom*. Edina, Minn: Interaction Book Company.

Kajita, S. (2001). Current status of WebCT and future of information basis for higher education. *IEEE International Conference on Multimedia and Expo, 2001.* 637 - 640.

Kauffman, J., Ferracchiati, F.C., Matsik, B., Mintz, E.N., Narkiewicz, J.D., Tegels, K., Xie, D., West, J., Chinnathampi, J. and Greenwood, J. (2002). *Beginning ASP.NET Databases Using VB.NET*. Bermingham, U.K: Wrox Press Ltd.

Kobryn, C., Booch, G., Jacobson, I. and Rumbaugh, J. (2003). *UML Distilled*, 3rd edn. Boston: Addison Wesley.

Opper, S. and Fersko-Weiss, H. (1992). *Technology for Teams: Enhancing Productivity in Networked Organizations*. New York: Van Nostrand Reinhold. 43-58.

Orfali, R., Harkey, D. and Edwards, J. (1999). *Client/Server Survival Guide*. 3rd edn. John Wiley and Sons.

Reynolds, M. (1994). *Groupwork in Education and Training: Ideas in Practive*. London: Kogan Page Limited.

Rockwood, H. S. (1995). Cooperative and collaborative learning. *The national teaching and learning forum*, 4(6), 8-9.

Scott, K. (2002). *The Unified Process Explained*. Boston: Addison Wesley Professional.

Tinzmann, M.B., Jones, B.F., Fennimore, T.F., Bakker, J., Fine, C. and Pierce, J. (1990). *What Is the Collaborative Classroom?* Available: http://www.ncrel.org/sdrs/areas/rpl_esys/collab.htm (5 May 2004).

WebCT. (2004). *WebCT Official Website*. Available: http://www.webct.com (6 May 2004)

Wheeler, B.C., Dennis,A.R. and Press, L.I. (1999). Groupware comes to the Internet: charting a new world. *ACM SIGMIS Database*, 30(3-4), 8-21.

Xue, S., Tan, R., and Huang, H. (2001). CSCL: an Internet based education model. *The Sixth International Conference on Computer Supported Cooperative Work in Design*. 483-487.

Yong, J. (2004). How important is teamwork? *In.Tech, The Star*. 8 April 2004. 17.