PENINSULA TECHNIKON
FACULTY OF ENGINEERING

The design and implementation of an Assignment Management Module to function within the architectural constraints of an existing Open Source Software (OSS) Learning Management System (to meet target-user requirements)

Prepared by: Ke Sun (200306101)

Internal Supervisor: Bennett Alexander

Dissertation submitted to the Higher Degrees Committee of Peninsula Technikon in Fulfilment of the Requirements for the Master of Technology Qualification in the Department of Information Technology at the Peninsula Technikon

2003/2004

Declaration

Hereby I, Ke sun, declare that this study project is my own original work and that all sources have been accurately reported and acknowledged, and that this document has not previously in its entirety or part been submitted at any university in order to obtain an academic qualification.

Ke Sun

Acknowledgements

I wish to express my sincere acknowledgement and appreciation to the following people:

Mr Bennett Alexander, my supervisor, for all his help, enthusiasm and guidance and for the encouragement to pursue this project.

All other Peninsula Technikon colleagues, for their friendship, support and motivation.

My parents, for encouragement, motivation and support.

Liyan, my wife, for endless patience, encouragement and love.

Glory to God without whom this study would not have been possible.

Abstract

This document describes the design and implementation of an Assignment Management Module (AMM) to function within the architectural constraints of an existing open-source software (OSS) Learning Management System (LMS). The project is established for Information Technology Department Master Degree Project of the Peninsula Technikon.

The Assignment Management Module will be constructed in orders to make it easier to create, mark, and manage assignments and record individual student performances. The design entailed work on different function blocks like a user's authorisation, files upload/download and mailing reminder unit as well as writing and testing of the application code on the internet/intranet. The development process of the project to explore how to fulfill software engineering methodology in an open-source environment, also presents details of the design architecture and technologies to be used, as well as being mindful of its future directions.

The target audience of this document is anyone with an interest in an open-source software project in general, and in a Learning Management System in particular. If the reader has also been a contributor to Learning Management System, and especially to the management of assignments, this document may provide additional value, in that it strives to present a new approach to the understanding of such a module.

Abbreviations

AI: Artificial Intelligence

AMM: Assignment Management Module

ASP: Active Sever Pages

AVOIR: African Virtual Open Initiatives and Resources

CRUD: Create, Read, Update, Delete

CVS: Concurrent Versions System

FTP: File Transfer Protocol

GCC: GNU Compiler Collection

GPL: Gnu Public Licence

HOD: Head Of Department

IRC: Internet Relay Chat

IT: Information Technology

ITMPT: Information Technology Department Master Degree Project of the Peninsula Technikon

ITMPTG: ITMPT Programming Group

KEWL: Knowledge Environment for Web-based Learning

KEWL.NextGen: The Next Generation of KEWL

LDAP: Lightweight Directory Access Protocol

LMS: Learning Management System

MVC: Model View Controller design pattern

NTP: Network Time Protocol

OOP: Object-Oriented Programming

OSS: Open-Source Software

PEAR : PHP Extension and Application Repository

PHP: Pre-Hypertext Processing Language

RDMS: Relational Database Management System

SAD: Software Architecture Design of AMM

SDLC: System Development Life Cycle

SMTP: Simple Mail Transfer Protocol

SPMP: Software Project Management Plan of AMM

SQL: Structured Query Language

SRS: Software Requirements Specification of AMM

STC: System Test Chapter of AMM

URL: Uniform Resource Locator

WBS: Work Breakdown Schedule

Contents

List of tables

List of figures

CHAPTER 1 INTRODUCTION TO STUDY

1.1 TITLE

The design and implementation of an AMM to function within the architectural constraints of an existing OSS LMS to meet target user requirements.

1.2 OUTLINE OF THE PAPER

As the concepts of open-source project, LMS and AMM software may be unknown to most of the readers of this paper. It has been divided into five parts. The first part (Chapter 1 and Chapter 2) is an extensive background on open-source and LMS terminology. The second part (Chapters 3 to 6) is a presentation of the whole development process of AMM. It establishes the blueprint and arithmetic model as well as an introduction to lifecycle paradigm software engineering methodology exercised in the project lifecycle. Then follows a presentation of the case study (Chapter 7)—the design and implementation AMM architecture in KEWL.NextGen. Thereafter follows a presentation of the conclusions (chapter 8) and short recommendations.

1.3 STATEMENT OF THE PROBLEMS

There is a need for an AMM in the existing open-source LMS, and making it easier to create, mark, and manage assignments and record individual student performance in one application module.

The sub-problems arising from this are:

- what kind of software engineering methodology should be adopted?
- what kind of design pattern should be used in developing the AMM?
- how can the AMM be embedded into the existing LMS?

1.4 PURPOSE OF THE STUDY

- understanding technology about Linux System, PHP, and MySQL.;
- through literature study, research the academic approaches to meet the actual software engineering methodology and design pattern problems in an open-source environment;
- practice and examine software engineering methodologies through the development process of AMM in open-source environment;
- comprehend and practice object-oriented design pattern of PHP in case study;
- to provide an easy to use tool that offers lecturers the ability to create and administrate customized assignments and a friendly online assignments environment for students; and
- enhance instructional design features, making online study easier and more convenient.

1.5 HYPOTHESIS

The software engineering methodology and design pattern used in LMS in an open-source environment is similar to widely known software development methods, such as the waterfall paradigm, prototyping paradigm and spiral paradigm.

- an existing open-source learning management project that will open developing environment and offer relative documentation;
- education is a social issue and communities will form partnerships to resolve educational problems; and
- some software and hardware companies will encourage academic research and offer necessary technical support.

1.6 SCOPE OF RESEARCH

This study deals with the development of an AMM in LMS, and will practice by means of a development process how to fulfil engineering development methodology that is in the open-source environment, and that is not in a normal software development environment.

Open-source projects are mainly an Internet phenomenon—but certainly not limited to the Internet as a communications medium. In this study, it focuses only on an open-source LMS developed on the Internet/intranet.

When this document uses the word "methodology", it refers to the overall development process of a LMS further focusing on a specific action, that is the process of developing an AMM.

As a result of time-constraints, a lack of support by the software company and the hardware company and a lack of practical experience, some of the technical issues such as text processing and AI may be abandoned during the software implantation process.

1.7 SCIENTIFIC METHOD

As the purpose of this project is the design and implementation of an AMM to function within the architectural constraints of an existing LMS in open-source environment, the experimental method was chosen.

The follow diagram gives the research logic model towards a solution in a research environment.

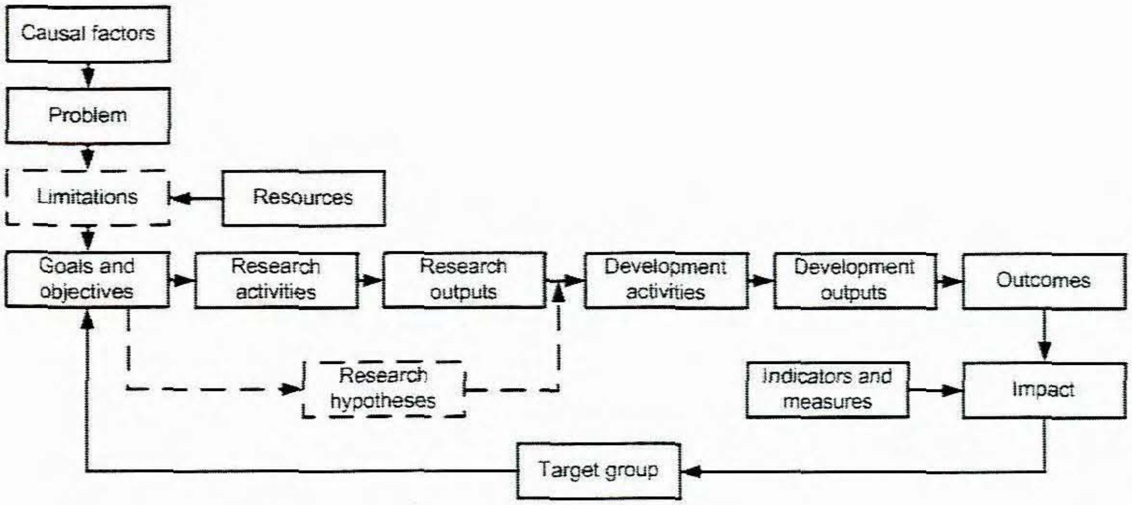


Figure 1.1. Logic module (Johan, 2002)

1.8 BACKGROUND

Open-source software is one of the most active and growing areas of IT and deserves substantial attention and consideration. Not only software specialists but also economists are studying this new trend and many national and international public institutions are interested in open source software. IT theory-based learning and teaching is becoming a pervasive technology in public and private institutions, from old computer-based training applications to modern life long learning concepts, as well as from CDROM-based individual learning to network-based collaboration, and from static contents to multimedia dynamic applications. The evolution of learning paradigms, technologies and concepts has grown very rapidly, and the point of convergence of these trends is what we now identify with as LMS.

All software is built with a source code. "Open-source means the code can be seen and changed. With this power comes control" (Internet source 1). The concepts regarding open-source software will be explored in more depth later.

Modern educational methodologies make considerable use of a network (Internet or intranet) for implementing many forms of collaborative learning and integrated teaching. Practice communities, virtual classrooms, discussion groups, and assignments are only a few examples of management collaborative knowledge. All these forms of highly interactive learning use a software supporting tool, the so-called LMS platform, which provides the different services to teachers and students, for example courseware delivery, communication, cooperation, monitoring of the students activity, assignment management, etc.

Development of LMS platforms, either commercial or open-source, is a very active area in the software industry nowadays. These days, LMS programs are designed to integrate with other delivery methods to create a powerful blended approach to people development. Far from taking away the human touch, LMS can dramatically increase the effectiveness of classroom training, by enabling delegates to prepare fully for such a course in advance.

1.8.1 Typical LMS

LMS is a Web-based software solution to simplify the administration of learning programs. It creates efficient processes for both learners and administrators. LMS is in use today in many organisations (universities, schools, and corporations). Traditionally LMS provides a content repository for course materials as well as facilities for student (trainee) tracking and management. Additionally, some LMSs provide authoring tools, assignment tools and communication tools, such as email and discussion groups. Detail information refer to Appendix C.

1.8.1 The background of the existing LMS-KEWL

AVOIR background

The UWC initiated collaboration with other South African institutions to harness the potential within African countries to create a group of open-source software developments known as AVOIR. These developers aimed at creating both educational and business opportunities within the African continent, through their software development. The focus of this project is to further human development and job-creation by the formation of international alliances both within and outside Africa.

KEWL 1.X

"Knowledge Environment for KEWL is an advanced LMS that was developed in ASP and which runs on Microsoft servers with an MS-SQL server as the database" (Derek, 2004:4). Even though KEWL utilizes a Microsoft platform, the code is open-source, and can be download from the internet (http://cvs.uwc.ac.za/).

KEWL has most of the features common to commercial Learning Management Systems and is ready for use to deliver online courses. All activity within KEWL is based around two main objects: user and course. When a user logs in to a KEWL site, permissions are established and the appropriate links become available. The user is first in a 'lobby' area, but once a user enters a course, then all activity takes place in relation to that course. If the user moves to another course, then all activity is related to that course, thus simplifying the actions that are needed to access different tools.

The KEWL interface has to be as simple as possible because of the low bandwidth environment found in most developing countries, including Africa. Because content will vary from educator to educator, bandwidth requirements will depend on the content a particular educator stipulates.

An advantage of the 1.2 version is that it has multilingual and translation facilities, which can translate the interface into other languages. A second advantage is the scope for personalisation and accommodation of learners with special needs, with the modifiable skins for the interface.

KEWL.NextGen

KEWL.NextGen aims to run on GNU/LINUX and other systems by porting KEWL to PHP and MySQL/PostgreSQL. The research supporting this forms part of the AVOIR project and also utilises the contributions of international open-source developers. "The approach to this conversion is to base the code base on the functionality that has worked well in KEWL, but to re-engineer with a modern, modular architecture" (Derek, 2004:6). This project development uses the skills of computer scientists, programmers, educationists and users.

1.8.2 The background of the existing LMS-MOODLE

"Moodle is a software package for producing internet-based courses and Web sites. It's an ongoing development project designed to support a social constructionist of education" (Internet source 2).

Moodle is provided freely as open-source software (under the GNU Public License). This means that Moodle is copyrighted, but that users have additional freedoms. They are allowed to copy, use and modify Moodle provided that they agree to provide the source to others; not modify or remove the original license and copyrights, and apply this same license to any derivative work.

1.8.3 Assignment Management Module (AMM)

The AMM is an integral part of LMS. An AMM will be constructed, to make it easier to create, mark and manage assignments and record individual student performance. Fewer clicks and a more intuitive workflow will increase lecturer productivity, reducing the amount of time spent on organising papers, sorting out floppy disks, or downloading files from disparate sources.

The core functionalities of the AMM are to:

- to collect and organise electronic assignments through the AMM;
- to submit assignments, through corresponding course content areas related to the assignments, thereby linking lessons to student deliverables and results; and
- the return of and comment on corrected files, as well as store of private notes related to student work in the AMM.

1.9 CONCEPTUAL UNDERPINNING FOR THE STUDY

1.9.1 What is open-source

"Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. copyleft guarantees that every user has freedom" (Stallman, 1998:12).

"Software distributed under such licence terms as copyleft is always distributed in source-code form and often, but not always, in binary form. Most copyleft licences are designed to not only permit modifications, but also to encourage people to help the code evolve. It is such licences we refer to when using the expression open-source software" (Internet source 3).

The idea of open-source software development is to allow and encourage people on the Internet to contribute to the evolvement of a software project. This form of software development has many advantages. For example, a company may create some kind of utility to solve problems experienced in a network administration, which cannot be solved in any standard way. "Provided the software is of interest to other organizations and individuals, it will evolve—bugs being corrected and new features being added, without great effort by the company that created it in the first place" (Raymond, 2003: 4).

"The concept 'open-source' expression was introduced early in 1998 as a less confrontational way of referring to free software" (Raymond, 2003: 1). It has been argued that the word "free" is detracting commercial interests from adopting the idea of open software development, as the word "free" often is interpreted as "free of charge" rather than in the sense of "freedom". Since its introduction, the concept has been adopted by media and developers, and even by major software companies such as Netscape Communications, Corp. and Corel Computer Corp.

1.9.2 What is LAMP development environment?

"The implementation followed a model for open-source deployments called 'LAMP' that includes the Linux operating system, Apache Web server, MySQL database and any of three development languages—PHP, Perl or Python." (Internet source 3).

Many successful organisations use LAMP *for intranet, extranet, and customer* website applications. According to a Merrill-Lynch study, "...over one third of corporate *CIOs (Chief Information Officers) plan on using Linux systems this* year. More than five million domains on the Web incorporate PHP support in the servers that power them. MySQL continues to win awards and enthusiastic reviews" (Internet source 4).

All of the components of LAMP can be downloaded and used for free. Whether a user is using it for website with over 100 visitors per month, or a company-wide intranet application, the software costs nothing. This can result in extensive cost savings, which is why so many large organisations are moving to Linux and other open-source solutions for a variety of projects.

There is an extensive support network for all LAMP components. There are literally millions of LAMP users who congregate in online communities, such as http://www.onlamp.com, to help each other get the most out of LAMP. Training is available from a wide array of providers, and many consulting firms offer advanced capabilities for those businesses that require sophisticated LAMP development.

Perhaps most importantly, LAMP works well. Its component technologies offer tremendous flexibility, rapid development, high performance and fewer security and stability problems than many far costlier proprietary solutions.

1.9.3 What is Linux?

The Linux operating system will be used as a platform to run a website server and database server in the present research environment.

Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that would exceed the Minix standards. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The current full-featured version is 2.4 (released January 2001) and development continues.

Linux is developed under the GNU General Public License and its source code is freely available to everyone. This however, does not mean that Linux and its assorted distributions are free—companies and developers may charge money for it as long as the source code remains available. Linux may be used for a wide variety of purposes including networking, software development, and as an end-user platform. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems.

Because of the very nature of Linux's functionality and availability, "...it has become quite popular worldwide and a vast number of software programmers have taken Linux's source code and adapted it to meet their individual needs. At this time, there are dozens of ongoing projects for porting Linux to various hardware configurations and purposes" (Internet source 5).

1.9.4 What is Apache?

"Apache is an open-source HTTP server for Unix, Windows NT, and other platforms. Apache was developed early in 1995, based on code and ideas found in the most popular HTTP server of the time, NCSA httpd 1.3. It has since evolved to rival (and probably surpass) almost any other Unix-based HTTP server in terms of functionality and speed. Since April 1996 Apache has been the most popular HTTP server on the Internet, In May 1999 it was running on 57% of all Web servers" (Internet source 6).

1.9.5 What is MySQL?

MySQL is a Relational Database Management System (RDMS). A relational database adds speed and flexibility by storing data in separate tables rather than putting all the data in one area. These tables are linked by defined relations and make it possible to combine data from several tables upon request. Using a RDMS means it is possible to add, access, and process the data stored in a database. SQL stands for "Structured Query Language"—the most common standardised language used to access databases. MySQL is very fast, reliable and easy to use. MySQL also has a very practical set of features developed in close cooperation with its users. It is also open source and is therefore freely accessible at www.mysql.com.

As a result of its connectivity, speed and security, MySQL is used to access databases on the Internet. "It was originally developed to manage large databases at a much faster speed than the solutions that previously existed. MySQL has for several years, been thriving in the challenging areas of production " (Internet source 7).

1.9.6 What is PHP?

The PHP Hypertext Processor is an open-source server-side scripting language for Web servers, which can be embedded inside HTML as a clever means of providing dynamic Web pages. Dynamic Web pages are pages which interact with the user, so that each user visiting the page sees customised information—which may vary each time and which may be based on a form they have just filled in, or on information extracted from a database or some other external source. Typical applications include e-commerce, online newspapers, visitors' books, ticketing systems, project management, and other groupware projects. The traditional way to produce this type of dynamic page is via Common Gateway Interface (CGI) scripts, but these are separate programs which must be executed as a new process for each page hit, so they scale badly and rapidly become memory and processor hogs as server load increases.

PHP solves this problem by becoming part of the Web server, essentially extending the functionality of the server itself, so that the server can do the processing without having to spawn extra processes. It is not alone in doing this, but unlike most other scripting languages for Web page development, PHP also offers excellent connectivity to most of the databases in use today. Perhaps the greatest advantage of PHP, when compared to other scripting languages such as ASP or Cold Fusion, is that it is open-source and cross-platform. The natural home of PHP on Linux servers running Apache server software, but it runs equally well on any other UNIX or Windows platform, and can be used with other Web servers.

PHP started its life as a Practical Extraction and Reporting Language (Perl) program written by Rasmus Lerdorf to track visitors to his online résumé. It was then rewritten in C and was extended to include support for database access. From these simple beginnings the open-source community has expanded and developed PHP into a powerful server-side scripting language.

1.10 DEFINITIONS

Asynchronous: "Asynchronous communication that is not synchronized; that is, not occurring at a fixed time or place. The term 'asynchronous' is usually used to describe communications such as e-mail and bulletin board systems. Its opposite is synchronous" (Internet source 8).

CGI: "Common Gateway Interface is a set of rules that describe how a Web server communicates with another piece of software on the same machine, and how the other piece of software (the CGI program) talks to the Web server. Any piece of software can be a CGI program if it handles input and output according to the CGI standard" (Internet source8). Usually a CGI program is a small program that takes data from a Web server and does something with it, such as putting the content of a form into an e-mail message, or turning the data into a database query. CGI "scripts" are just scripts which use CGI. CGI is often confused with Perl, which is a programming language, while CGI is an interface to the server from a particular program. Perl is an application of CGI, as well as Python, PHP, and other scripting languages.

Class: "A 'class' is the prototype for an object in an object-oriented language and is analogous to a derived type in a procedural language. A class may also be considered to be a set of objects which share a common structure and behaviour. The structure of a class is determined by the class variables which represent the state of an object of that class, and the behaviour is given by a set of methods associated with the class" (Internet source 9).

Classes are related in a class hierarchy. One class may be a specialisation (a "subclass") of another (one of its "superclasses") or it may be composed of other classes, or it may use other classes in a client-server relationship. A class may be an abstract class or a concrete class.

Collaborative learning: "Collaborative learning is a style of teaching and learning where students work in teams toward a common goal" (Internet source 8). In some online courses, collaborative learning teams are used where a student is asked to work cooperatively with his/her classmates. The idea is that students learn from each other.

Download: "To copy a file from another source to your computer. The opposite is upload" (Internet source 8).

FTP: "FTP is the abbreviation for File Transfer Protocol, the protocol, or code, used on the Internet for sending files to and from a remote server (Internet source 8).

Function: "Computing usage of the word 'function' derives from the mathematical term but is much less strict. In programming (except in functional programming), a function may return different values each time it is called with the same argument values and may have side effects.

A procedure is a function which returns no value but has only side-effects. The C language, for example, has no procedures, only functions. ANSI C even defines a type, void, for the result of a function that has no result" (Internet source 9).

GUI: "GUI is the abbreviation of Graphical User Interface, an interface that is graphical (such as Windows), rather than text-based (such as DOS). In a GUI, are executed commands are executed by clicking on icons, whereas in DOS commands are executed by typing text commands on the command line" (Internet source 8).

GNU: "GNU is a recursive acronym for "GNU's Not UNIX" is a project that aims to create a completely free unix-compatible software system" (Internet source 10). Nowadays this project has produced amongst others, widely used compilers and debuggers. The project is run by the free software foundation.

HTTP: "Hypertext Transfer Protocol is the protocol that supports the exchange of Web pages over the Internet" (Internet source 1).

Hypertext:" Hypertext is a special method of formatting text and graphics so that links can be created among the objects. On the World Wide Web, text and graphics that are "hyperlinks" are usually underlined in blue, indicating that they can be clicked on to link to related information" (Internet source 8).

Plug-In:" Some Web sites and courses require additional software programs for a Web browser called "plug-ins". They enable a browser to display various audio, video, or graphics. Some of these utilities include:

- Adobe Acrobat Reader (for PDF files)
- QuickTime digital media player
- RealPlayer digital media player
- Macromedia Shockwave Player
- WSFTP Free Version" (Internet source 8).

Perl: PERL is the abbreviation of 'Practical Extraction and Reporting Language', a robust programming language is frequently used for creating CGI programs on Web servers because it is faster than UNIX shell script programs. It can read and write binary files, and it can process very large files (Internet source 8).

PPP: "Point-to-Point Protocol is a method of connecting a computer to the Internet. Point-to-Point Protocol is a standard for directly connecting computers to the Internet via dialup telephone connections" (Internet source

8).

Search engine: "A Web site that allows a user to search the Internet by keyword. Examples are Yahoo, Google and Lycos" (Internet source 8).

Server: "A 'sever' refers to a computer or device on a network that stores and manages network resources. The server "serves" information to other computers (clients). For example, all Web sites are stored on Web servers. When a Web site is accessed, the accessor is a client being served files from that the hosting server" (Internet source 8).

Session: "A session refers to a lasting connection between a user (or user agent) and a peer, typically a server, usually involving the exchange of many packets between the user's computer and the server. A session is typically implemented as a layer in a network protocol (e.g. Telnet, FTP).

In the case of protocols where there is no concept of a session layer (e.g. UDP) or where sessions at the session layer are generally very short-lived (e.g. HTTP), virtual sessions are implemented by having each exchange between the user and the remote host include some form of cookie which stores state (e.g. a unique session ID, information about the user's preferences or authorisation level, etc.)" (Internet source 9).

Synchronous: "Synchronous is the opposite of asynchronous, and refers to communication that occurs in real-time such as Chat" (Internet source 8).

Telnet: "Telnet is a terminal emulation program that connects a computer to a server on a network so that the user can enter commands as if he/she were at a terminal directly connected to the network he/she is accessing. Telnet access is commonly used to search library databases" (Internet source 8).

URL: "Uniform Resource Locator (URL) is the Internet "address" of a Web page. Each Web page on the World Wide Web has a unique URL" (Internet source 3).

1.11 SUMMARY

Free software is gaining popularity and media attention. Based on open-source, some of LMSs are kind of free software. They are not only in binary, but also in source code form. Traditionally LMS provides a content repository for course materials as well as facilities for student (trainee) tracking and management. The AMM is a basic module of an LMS. This first chapter introduces our study purpose, scientific method and relative aspects of our study, and provides a useful description of the fundamental background of open-source software and LMS.

Chapter 2 will explore in greater depths and reviews, the software developing methodologies in open-source software project, and some terminologies of LMS and technical issues.

CHAPTER 2 REVIEW OF RELATED LITERATURE

2.1 INTRODUCTION

This section reviews the technologies of the open-source project and the current status of LMS based on the open-source and relative technical issues.

First, an insider's view of engineering software in an open software project and the relative software engineering methodology will be discussed. Second, the current LMS based on the open-source will be presented. Following that, an emerging industry standard of LMS will be discussed. Fourthly, it looks at current LMS delivery technologies and future delivery technologies. Finally, the cooperant development tools used in developing LMS content is discussed.

2.2 OPEN-SOURCE PROJECT

An open-source project may take various forms. Raymond (2003: 6) identifies two development styles and considers them to dominate the open-source development area namely the 'Cathedral' style and the 'Bazaar' style of software development.

- **The Cathedral style of software development**

"Software developed in the 'Cathedral' style is typically crafted by a single programmer or a small isolated group of programmers" (Raymond, 2003:28). Although contributions of source code may be accepted from people not associated with the project, it is not the primary development style. Design and implementation decisions are made exclusively by the authors, as is the release cycle. In many cases there are very few releases of the software until it is considered usable for a larger user community, thus mailing lists make it harder for programmers to contribute code to the project.

- **The Bazaar style of software development**

"Projects executed in 'Bazaar' style basically adhere to the rule "release early; release often". Although many projects are run by a small group of programmers, the software is released as often as possible to encourage contributions from other programmers" (Raymond, 2003:47). Design decisions are often discussed intensely on mailing lists, newsgroups and IRC. Often there is only one central code repository, or source tree, to which contributions are added. When developers disagree on implementation techniques, the source tree may split, as the repository manager may chose to incorporate only one of the proposals into the source tree. This way the original project may deviate into several related but rather different projects, even though this is quite uncommon.

These fundamental development styles are very useful in describing the how an open-source development project is managed, but tell us very little about the specifics of the analysis, specification and design phases found in most software engineering methods.

2.2.1 Project management and organization

The Free Software Foundation is making an effort to create a free implementation of the open step specification as published by NeXT Software Inc. and SunSoft Inc. in 1994. This project, called GNUstep, is divided into several sub-projects, which are managed rather independently from each other. Of particular note is that these projects did not produce any code in the early stages, creating design specifications and requirements instead. When the coding started very few developers did a majority of the work, "and even though the project begun in 1994, the first public beta was only released in 1998" (Raymond, 2003:64).

In 1979, Eric Allman at the University of California at Berkeley wrote the program now known as Sendmail. Sendmail, as of 1998, is the dominating electronic mail routing implementation, and has received major contributions not only from individuals on the Internet, but also by Sun Microsystems Inc. and the Hewlett-Packard Company. The product evolves continuously, mainly in maintenance aspects such as bug fixes, but new functionality is also incorporated.

Other free development efforts have split up, creating new applications. The compiler created by the Free Software Foundation called GCC has spawned a new project called EGCS, executed by Cygnus Solutions, adding features not incorporated in the original source-tree. Another project aiming to create a high-end image processing application called GIMP has spawned an effort to create a user interface library in parts similar to the OSF Motif library—and that effort is in turn the basis of the GNOME project, which like the KDE project, is developing an integrated user environment with attractive developer support libraries.

Of vital importance to all of the development projects mentioned above, is communication between the contributors. This is accomplished in many different ways, and often there is no single preferred communication channel in an open-source project. The use of newsgroups and mailing list dominates, and the Web is almost always used as a tool to create a central source of information.

Distribution of source code is done in various ways. Many projects use version control systems, of which Concurrent Versions System, CVS, is dominating. A common way of distribution of source-code to developers is to allow anonymous access to the source-tree using a CVS client. It is also common to place snapshots from such a CVS source tree on FTP sites that are mirrored around the world. The version control systems are fundamental to the project management of these projects, as they aid in keeping track of changes in the source-tree, making it possible to reverse changes done to the software. In most cases it is easy for developers to gain full access to the CVS server, and the contributions made by those who only have anonymous, read-only access are sent to those who have full access.

Some of the projects we have examined, such as the GNU compiler project are more centralised than others. Although all projects utilise some kind of central coordinating organisation, the influence this organisation has on the contributions to the actual software differs.

2.2.2 The Developers and the users

The users of open-source software are not always developers. There is, for example, a vast number of Sendmail and Linux users that are not software developers. Many projects actually strive to widen this "ordinary" user-base, although a software developer is also a potential contributor to the project. This is because most users in fact are co-developers of the project, helping it evolve in other ways than contributing code. An example of this is the process of bug-correction often encountered in these projects. It is common that bugs are reported, identified in the source code and corrected by different individuals. The "ordinary" users that have no programming experience are often prone to report bugs, and as there are many users that are also programmers, chances are that at least one programmer will take the time to identify the error in the actual source code – and as the error is identified, someone else often takes to time to correct it.

Other kinds of co-development efforts, such as introducing ideas for new features, translating a program to another language and writing documentation is often done by users that are not programmers. These tasks are almost always done incrementally and given a large user-base. Much work in these areas will be done by many individuals making quite small contributions.

2.2.3 Will and duty

The software projects studied during this research project, evolve for various reasons. Moodle, the LMS, has strong commercial interests. As an open-source project grows, gaining a bigger user-base, there will be a greater need for the software to evolve. As the need grows, so does the pressure on the core developers in the project. In this situation one of two things usually happens: either the project is split up into two or more smaller projects or the number of developers that make very large contributions increases

The latter of these is very much due to the fact that many users are developers and, as most users are aware that there are no guarantees given by anyone that the software will evolve at a certain rate, one has to give in order to gain. The former is a direct consequence of the principle of copyleft—instead of switching to a new software package, a group of skilled enthusiasts are legally allowed to reuse parts of the original code and extend it in order to satisfy the needs that are not met by the original development project.

2.3 SOFTWARE ENGINEERING METHODOLOGY

Software development projects usually obey some kind of structure, following given timelines and entering different stages during the development process, such as analysis, specification, design and implementation phases. "The use of a software engineering methodology is fundamental to most application developers today. The methodology is an important tool to control such things as resource management, product design and quality assurance, in order to produce 'well-engineered software'" (Pressma,1992:64).

2.3.1 Attributes of software engineering methodology

Sommerville (1999:38) argues that a well-engineered software system possesses four key attributes, assuming that the software provides the required functionality, namely maintainability, reliability, efficiency and an appropriate user interface.

- **Maintainability**

A software with a long life-time is subject to regular change, and has to be written and documented so as to make changes possible to the code without undue costs.

- **Reliability**

A software should perform as expected by users, and should not fail more often than is allowed for in its specification.

- **Efficiency**

A software should not make wasteful use of system resources, such as memory and CPU.

- **An appropriate user interface**

To allow users take full advantage of a software system, the user interface has to be designed with the capabilities and the background of the user in mind. Different software development methodologies take on different approaches in order to satisfy these attributes—in particular, users are given different roles in the development process and the design and requirement specifications are made in different ways.

2.3.2 Typical Softer Engineering Methodologies

"Given unlimited resources, the majority of software problems can probably be solved but the challenge for software engineers is to produce high-quality software with a finite amount of resources and to a predicted schedule" (Sommerville 1999:45).

Developing a software system is usually not done in a single day of hard work, but is a complex and time-consuming process. In order to control this process, reducing the complexity and uncertainties surrounding the developing software system, engineers try to adhere to some kind of framework that introduces certain degrees of structure to the overall development process.

"Software engineering methodologies are the framework that tells us how we should go about developing our software systems. These frameworks define different phases of the development process, such as planning, requirements analysis, design, testing and maintenance" (Pressman, 1992:59).

The most popular methodologies for software engineering are sometimes referred to as "software engineering paradigms". The choice of which methodology to use in a development project is closely related to the size of the software system and the environment in which it is supposed to function. The environment in itself constitutes a larger system—though most often not a computer software system, but rather some kind of organisation. The different paradigms presented in this chapter all spring from this view of the world as a system of systems.

2.3.2.1 The life-cycle paradigm

The life-cycle paradigm of software engineering is sometimes called the "waterfall model", as it demands a sequential approach to the development process. The work is started at the system level and passes through phases of analysis, design, coding, testing and maintenance. Six activities constitute the overall development process (see Figure 2.1 The life-cycle paradigm)

(Mathiassen, 1998:26).

- **System engineering and analysis**

This activity is characterised by system-level requirements being gathered for all system elements, which then are reduced to a subset of requirements that are relevant to the software system being developed. Overall design and analysis tasks are also executed in order to understand the full system, of which the software system being developed is only a part.

- ***Software requirement analysis***

This activity is usually executed together with the customer, as the goal is to document all functions, performance and interfacing requirements for the software.

- **Design**

When creating the design of the software system, the requirements are transformed into a representation of software that can be assessed for quality before the actual coding begins. Data structures, architecture, procedural detail and interface characterisation are outlined and documented in a design specification.

- **Coding**

This activity is the transition of the design specification into a software program.

- **Testing**

This activity has to be executed using documented test methods in order to ensure that as many errors as possible are unveiled, and that the software is in accordance with the requirements.

- **Maintenance**

Rather than being an atomic activity, maintenance reapplies all other activities, as new requirements are defined in order to adapt the software to the changes occurring in its environment and to correct errors encountered after the software has been deployed.

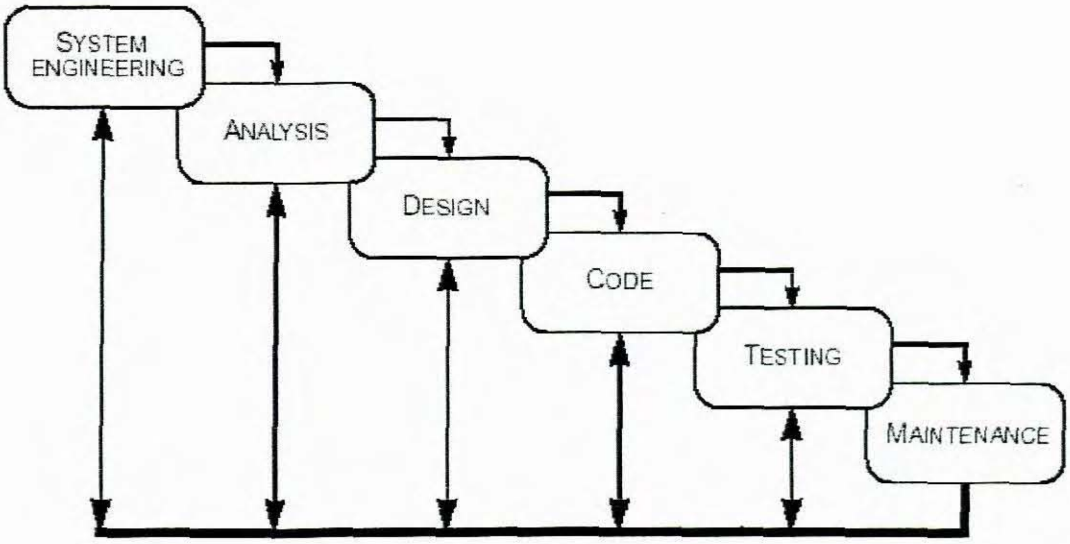


Figure 2.1. The life-cycle paradigm

2.3.2.2 *The prototyping paradigm*

Contrary to the static, procedural approach offered by the life-cycle paradigm, the prototyping paradigm can be used. Similarly to the life-cycle paradigm, this process begins by gathering requirements of the system. The developers meet with customers, determine the overall objectives of the software and identify any known requirements. A quick design then occurs, focusing on areas visible to the users, such as user interface and basic functionality. The design model is then used to implement a first prototype, which may take one of three forms: the interactions prototype, and existing program. (see Figure 2.2 Prototyping paradigm).

- **Interactions prototype**

This is a paper or computer software prototype, which makes it possible for users to understand how to interact with the software system.

- **Subset function prototype**

This is a working software program that implements a subset of the required functionality.

- **Existing program**

This is an existing program that implements most or all of the required functionality, but has features that should be improved in a later development effort. "When the prototype is created, the customer reviews it. Typically this review gives feedback to the developers that help remove uncertainties in the requirements of the software system, and starts an iteration of refinement in order to further clarify requirements by improving the prototype, or by building new prototypes. This process will result in one of two kinds of complete prototypes" (Pressman 1992:12), a throw-away prototype or a prototype to refine and deliver.

- **A throw-away prototype**

A throw-away prototype may be usable software program, but is not suitable as the final software product for various reasons, such as poor performance, maintainability or overall quality.

- **A prototype to refine and deliver**

A prototype to refine and deliver is enhanced and possibly reworked in various areas so that it is suitable to deliver as the final software product.

It should be noted that there are generally more reasons to throw away a prototype than there are not to—Brooks expresses this view like this: "The basis of this argument is that the first system built usually is too slow, too big or too awkward in use -- and that the prototype serves a better purpose of refining requirements than refining code" (Pressman 1992:8).

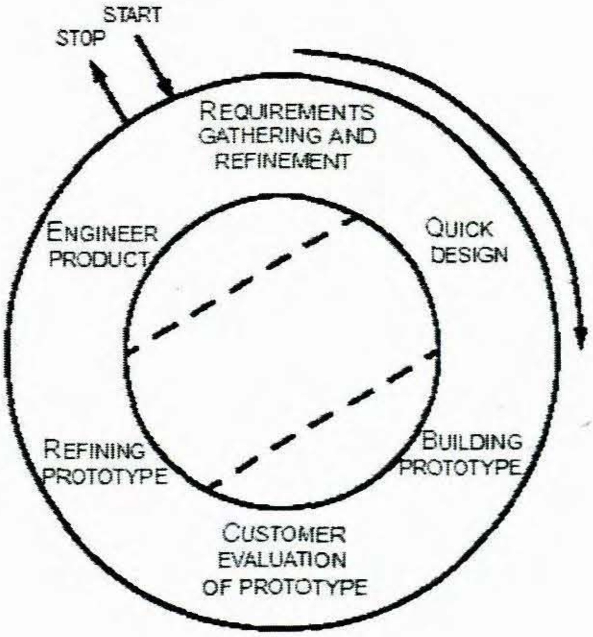


Figure 2.2. Prototyping paradigm

2.3.2.3 The Spiral model paradigm

"In addition to some of the aspects of the life-cycle and prototyping paradigms, the spiral model adds an element of risk analysis to the development process. The model is presented as a spiral (Figure 2.3), in which each iteration is represented by a circuit around four major activities planning, risk analysis, engineering and customer evaluation. (see Figure 2.3 Spiral paradigm)

- **Planning**

Planning determines the objectives and constraints of the project, and defines the alternatives.

- **Risk analysis**

Risk analysis allows for the analysis of alternatives, and identification/resolution of risks.

- **Engineering**

Engineering refers to the development of the "next-level" product.

- **Customer evaluation**

Customer evaluation involves the evaluation of the product engineered.

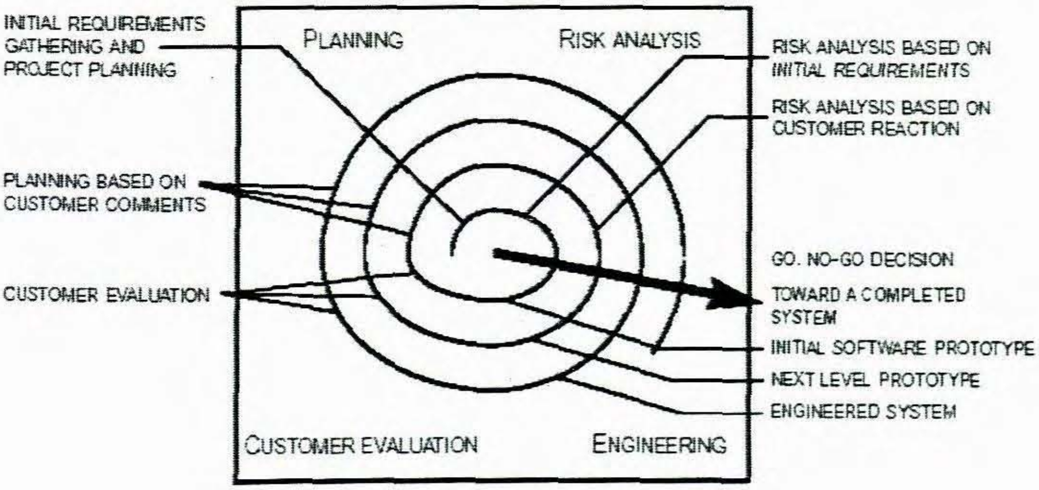

INITIAL REQUIREMENTS GATHERING AND PROJECT PLANNING
PLANNING
RISK ANALYSIS
RISK ANALYSIS BASED ON INITIAL REQUIREMENTS
RISK ANALYSIS BASED ON CUSTOMER REACTION
PLANNING BASED ON CUSTOMER COMMENTS
GO. NO-GO DECISION
CUSTOMER EVALUATION
TOWARD A COMPLETED SYSTEM
INITIAL SOFTWARE PROTOTYPE
NEXT LEVEL PROTOTYPE
ENGINEERED SYSTEM
CUSTOMER EVALUATION
ENGINEERING

Figure 2.3. Spiral paradigm

In each iteration the requirements are refined and a more complete prototype is produced, either by building on the prototype created in the first iteration, or by creating a new one. The risk analysis sometimes ends in a no-go decision, at which point the project may be terminated if risks are considered too great. The engineering that is done at each iteration may be executed in life-cycle form, as well as prototyping, depending on the certainty of the requirements (Pressman 1992:58).

As the developer and the customer are given an opportunity to react to risks at each iteration in the model, it can be considered to be evolutionary. It allows the developer to use the prototyping approach at any stage in the development, still maintaining the stepwise systematic approach of the life-cycle paradigm.

2.4 CURRENT LMS BASED ON OPEN-SOURCE

Today a number of educational institutions are offering their courses through the Internet, either partially or entirely. Given that much of the online learning work is focusing on the Web as the medium of interaction and learning, a plethora of Web-based course management tools have sprung up. These tools facilitate creation, administration and monitoring of courses conducted over the Internet, and are known as an LMS. There are a number of LMS products available today, such as WebCT, Blackboard, eCollege, Saba, Learning Space, and Topclass (Internet source 4).

Through the literature review, we know that when a developing team is going to develop a Web-based course for the first time, many questions arise concerning the evaluation and a choice of a LMS platform that can be adopted easily and quickly in order to provide all advantages of distance learning. Obviously the commercial LMS products are generic solutions, which can usually be implemented without conflicting with current systems. Nevertheless, the developers of a LMS are motivated to build their own platforms because of many disadvantages of existing software platforms.

Currently, another approach to start with the creation of a Web-based learning system could be to attempt to implement a tool that is open-source or free software, and join to the development team and in this way to speed up the whole process of establishing the LMS.

In general, the usage of open-source software is considered as speeding up the process of developing systems from different kinds of open-source projects, and increasing the productivity of scientific investigations. Customisable and free, open-source is becoming serious competition to the commercial LMS vendors. The advantage of open-source solutions goes beyond cost savings; a LMS requires a degree of customisation, and by having access to the source code, developers can make appropriate changes in the source code to adapt it to their needs.

Over the last few years it has become evident that a remarkable degree of development of LMS platforms has occurred. Many studies pay attention to the evaluation of available platforms and describe the systems themselves as well as their advantages and disadvantages as developing environments, The most reasonable seems to be a comparison of the platforms according to the LMS models. Services necessary for comparison of the platforms according to the LMS plafform source code could be also a base for comparing them. There are many open-source packages that can be used for creating Web-based courses, as is shown in Table2.1 Main LMS platform projects based on open-source.

Platform	Author	Country	Language
Ganesha	Anemalab	France	PHP
FreeStyle	University of Muenster	Germany	Java
RearSite	University of Rennes1	France	Perl
BSCW	FIT-GM Recherche University	Germany	Python
ClassWeb	UCLA	USA	Perl
UPportal	JA-SIG,MIT	USA	Java
Acolad	University Louis Pasteur	France	PHP

Adept	SourceForge	International	PHP
Claroline	University of Louvain	Belgium	PHP
Ilias	University of Cologne	Germany	PHP
Mimerdesk	Ionstream	Finland	Perl
Eledge	University of Utah	USA	Java
FLE3	University of Art and Design of Helsinki	Finland	Zope
WBT-Master	Project Coronet-Fraunhofer IESE	Germany	Java
Moodle	Moddle	Australia	PHP
Manhattan	SourceForge	USA	C
Spaghetti Learning	Spaghetti Brain	Italy	PHP

Table2.1. Main LMS platform projects based on open-source (internet source 10)

The delivery technologies used in the current LMS that are based on open-source, can be analysed and are exhibited in Figure2.4 Proportion of used delivery technologies. The adopted PHP is the highest at 41%. These results show that PHP is popular in the open-source field. It also supports the trend of developing LMS, which are open-source.

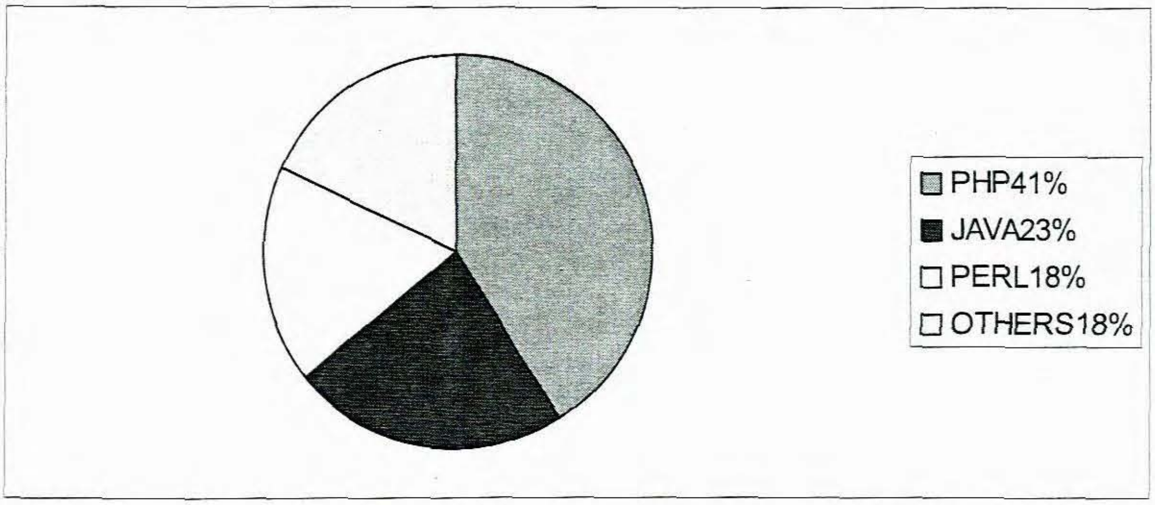


Figure2.4. Proportion of used delivery technologies

2.5 INDUSTRY STANDARDS OF LMS

"Solutions based on the specifications save customers money and reduce technical risks as the e-learning market continues to evolve" (internet source 12). A current movement in the e-learning industry involves the development and adoption of learning standards. Vendors, academics, government agencies, and industry consortia are all collaborating to define ways that will enable learning technology products to interoperate.

Closed, proprietary solutions may have worked in the past for CD-ROM, but they are unacceptable for interoperability on the Web. "The goal of the learning standards initiative is to develop open specifications. More developers are eschewing proprietary hardware and software designs that put systems at greater risk of obsolescence" (Barron, 1997:46). Ideally, the result of standardisation would be the ability to access all courses from any vendor directly from any intranet and administer these courses from any one training management system as well as from any intranet. Also, there would be compatibility across product lines. For example, a user would be able to buy the management system from one vender, authoring from another, content from a third, and expect it to all work together.

Development of the standards was started a decade ago by the Aviation Industry CBT Committee (AICC), an open forum of training professionals that develops guidelines for interoperable learning technology. The AICC has developed a Computer-Managed Instruction (CMI) specification that defines the tracking data exchanged between management systems and interactive lessons. It also defines an interchange format for course structure so that entire courses can be exchanged between management systems made by different vendors (Conner, 2000:107).

The EDUCOM Instructional Management Systems Project (IMS) is a coalition of over 225 educational institutions, training organisations, government agencies, and vendors defining a comprehensive architecture for online learning. The architecture encompasses platform independent interfaces for metadata, aggregated content, management services, user profiles and external services, such as databases. The IMS architecture anticipates the widespread availability of emerging technologies, such as XML, and provides an excellent vision for the future of online learning. IMS recently submitted a metadata specification to the IEEE Learning Technology Standards Committee (IEEE LTSC) for standardisation.

Notable is the IMS metadata specification that was generally accepted by IMS. Metadata associates descriptive information, such as author, title or subject, with content so that it can be easily located and appropriately used. The IMS Metadata specification benefits the learner looking for specific information with a meta-data-aware search tool both when the search is of Web-based resources and CD-ROM or DVD-ROM encyclopaedias (Internet source 11).

The World Wide Web Consortium (W3C) charts the future course of general-purpose Web technologies, such as HTML and XML. While the W3C does not focus on learning, it does define basic technologies that are assumed by many learning technology specifications.

The Computer Education Management Association (CedMA) is a forum whose members are education managers from companies manufacturing hardware or software products. CedMA provides a forum to discuss training and business issues of common interest to technology vendors. It is well positioned to accelerate vendor awareness and adoption of learning technology standards.

The Advanced Distributed Learning (ADL) initiative fosters collaborations between the government of the country, its academia and its industry to accelerate the advent of effective online learning. The initiative began in November 1997 under the aegis of the U.S. Department of Defense and the White House Office of Science and Technology Policy. June 1998, the ADL conducted a successful test of its Shareable Courseware Object Reference Model (SCORM), a reference model that defines a Web-based learning "content model".

SCORM incorporates IMS metadata standards, and, for the first time, allows for content from different vendors' learning management systems to be passed to other vendors' systems without any problems. At the ADL-sponsored Plugfest in June 1998, the ADL, AICC, IMS, and IEEE groups were able to meet together and discuss a unified e-learning specification that incorporates the four groups' work (Bethoney, 2000:208). Also at the event, over 90 organisations pledged support for SCORM specification.

The IEEE LTSC is an open, accredited standards body tasked to develop "real", de jure learning technology standards.

Consortia, such as IMS, ADL and the AICC increasingly acknowledge the IEEE LTSC as the single forum for turning specifications into standards. Both the AICC and IMS initiatives are furthering their goals in the IEEE LTSC. The AICC has submitted its CMI specification and IMS has jointly submitted a metadata specification with the European ARIADNE Project. The Alliance of Remote Instructional Authoring and distribution Networks for Europe (ARIADNE) is a research and technology development (RTD) project pertaining to the "Telematics for Education and Training" sector of the 4th Framework Program of the European Union. The project focused on the development of tools and methodologies for producing, managing and reusing computer-based pedagogical elements supported training curricula. Validation of the project's concepts is currently taking place in various academic and corporate sites across Europe.

Since December 1997, ARIADNE has been involved in standardisation activities performed under the auspices of the IEEE LTSC Committee. In this context, ARIADNE has agreed to collaborate with the US-funded Educause IMS Project, in view of reaching, as quickly as possible an Educational Metadata set that would be widely acceptable.

ARIADNE is also active in the standardisation activities initiated by the European Commission, scheduled to take place under the auspices of the CEN/ISSS (European Committee for Standardization/Information Society Standardization System). Work in this forum will initially concentrate on the "localisation" of the mainly English language results obtained so far at the IEEE (Richards, 1998:198).

2.6 PRESENT DELIVERY TECHNOLOGIES

At present, there are several options available for the training manager interested in implementing an LMS solution within the organisation. Though they might differ in details such as bandwidth, user interface and interactivity, the technologies used to deliver e-learning instruction have begun to converge around common technology standards and the delivery infrastructure known as the Internet. Approaches to online learning vary from those that use the Internet as little more than a distribution mechanism for simple text and graphics content to new technologies that allow live, two-way interactivity by means of dial-up connections, to science fiction-like virtual reality simulations (Peter, 1992:23). This section provides a review of these various technology systems.

Text and Graphics (HTML)

Perhaps the most basic delivery method used in Web-based LMS is through static HTML pages. Brandon Hall describes this medium as consisting mostly of text-and-graphics Web pages that utilise the basic capabilities of the World Wide Web. These courses tend to be primarily informational in nature, and represent most of the LMS courses that are currently available.

One advantage of using intranet-based training and performance support is the cross-compatibility of the HTML language. All that is needed to access LMS courses is a Web browser, and it also frees an organisation from being tied to any one supplier's proprietary system.

Text and graphics-based courses include many different types, such as courses that exist purely on email, bulletin board/online discussion forums, and static HTML Web pages that consist of text and graphics. This delivery format is often used as a supplement to traditional face-to-face instruction (Hall, 1997:197). While these courses represent the bare minimum in terms of utilising the capabilities of the Web, designing interactivity into text and graphics courses can augment their instructional effects. One common way in which this is accomplished is through the use of multimedia.

DHTML

Many LMSs are beginning to employ a Dynamic HTML (DHTML) scripting in delivery. DHTML is similar to Thin-client technology "that it allows for more engaging interactions without the need for browser plug-ins—provided the browser version is fairly recent. The difference is that DHTML is more often used for delivering asynchronous content. DHTML can create robust applications over a standard Internet browser used to deliver technologies such as multimedia, and even authoring environments for e-learning content" (Hall, 1997:126).

XML

XML, short for Extensible Markup Language, allows learning content to be labeled in detail, making it possible to customise e-learning content based on a learner's needs. "This detailed labeling of Web page content also allows for more accurate searches. The technology is on the path to replace HTML as the standard Web authoring language and is already being used by some e-learning providers as a means of providing on-the-fly customization of content "(Barron, 2000:42).

Additionally, if authoring-system vendors and courseware providers were to adopt XML as a standard, the need for third party browser plug-ins (i.e. Shockwave) would be eliminated (Filipczak, 1998:101). XML extends the advantages of inter-operability even further by both integrating content and tracking learner progress across several different providers. Finally, the richer language of XML allows for more interactive content than HTML, which leads to a more engaging experience for LMS, and which may ultimately make LMS more capability viable. XML is expected to transform the Internet radically in general and LMS in particular upon adoption of standards by the World Wide Web Consortium.

Multimedia

"Multimedia training is a type of computer-based training that uses two or more media, including text, graphics, animation, audio (sound/music), and video" (Barron, 2000:36). In practice, multimedia utilizes as many tools as is practical to produce a colorful, engaging program delivered via the computer.

A typical multimedia program allows users to control their progress and pace through the course so that each student can learn at his/her own speed. Multimedia languages, such as Java and DHTML, and plug-ins for authoring tools, such as Shockwave, are becoming increasingly used to deliver LMS courses.

However, multimedia is not without its drawbacks. Limited bandwidth presents a special problem when designing Internet-based education with multimedia. Connection speeds can be slow and downloads can be long due to factors which trainers often have little control over. Until bandwidth improves, LMS developers often need to exclude most of the "fat media" in their delivery systems, especially video, or create a hybrid design. (Barron, 2000:22) explains, "The visually rich, highly interactive medium and sophisticated authoring tools of the CD-ROM era have been replaced with the bandwidth constraints of the Internet and authoring limitations of HTML." However, experts predict that this situation will improve in the future as new technologies such as greater bandwidth, and load balance.

When discussing interactivity in LMS, it is important to ensure that uncalled for use of multimedia does not take the limelight away from the instructional strategy. To be effective, LMS needs to step beyond simple interactivity such as "Next" buttons, and move to a more engaging form of interactivity that promotes insight, skill, and the ability to reapply knowledge in numerous work contexts.

"The problem is that building sound interactivity based on business and instructional analysis is a difficult task that can drive budgets way up and deadlines way out—which is the wrong direction in the face of surging learning demands and shortening development cycles" (Horn, 2000:36).

One way to help make this manageable is by creating an infrastructure that supports collaborative learning.

Collaborative technologies

Collaborative learning networks may consist of numerous technologies, such as bulletin boards, conferencing software (i.e. Microsoft's NetMeeting), and streaming media (i.e. RealPlayer). The purpose of these technologies is to create an environment that fosters the students' learning through interactivity. In this environment, students are more apt to consider how the content they are receiving will impact them on the job (Hites, 1999:59).

In a collaborative example, instructors take on more of a facilitator role than of a lecturer role, often mentoring virtual teams as they work through problems and questions that relate to the instructional objectives for the course.

Web Collaborative learning network

Up to now, we have discussed collaborative environments that do not necessarily require specific time and place logistics. Another component of collaborative learning environments may include synchronous interaction.

Synchronous interaction is an increasingly popular delivery system used in *many e-learning environments today. Hall (2000:248) defines synchronous* delivery as "...an instructor or team of instructors (who) will present and be connected by audio or video, while at the same time the student will be online interacting with the online course." One of the appealing powers of synchronous interaction is the continuous, real-time sharing of knowledge and learning in the work situation (Karon, 2000:39). Other advantages include the immediate access to instructors and online mentors to ask questions and receive answers, similar to traditional learning environments. The disadvantage is that it requires a set date and time, contradicting the "anytime, anywhere" promise of e-learning (Hall, 2000:98). Streamed media can be used to deliver audio to multiple learners at one time.

Collaborative software packages such as Microsoft NetMeeting can enable live, multi-way interactivity by means of a dial-up connection where learners can jointly edit documents and collectively solve problems using a shared 'whiteboard' space.

Another technologies, called Thin-Clients, are emerging that provide similar functionalities without the need to install (or download) an application on each user's computer. This prevents the training department from having to take users through a software installation, rebooting, and troubleshooting before learning even begins, resulting in a lower cost for implementation (Barron, 2003:3).

Thin-client technology is fuelled by voice-over-internet-protocol (VOIP), which can deliver two-way audio, or connect learners to a phone bridge for integrated teleconferencing. One problem preventing mass adoption of thin-clients is the compatibility issues that emerge as users upgrade their Web browsers to newer versions. The thin-clients often have to be changed continuously to support compatibility (Barron, 2003:1).

2.7 PROSPECT DELIVERY TECHNOLOGIES

Virtual Reality Modeling Language (VRML) is the 3D language of the Web. It's purpose is to provide information in Web pages in a three-dimensional format.

Because objects in this environment are 3D, they can be viewed from any angle, including close-ups.

Applications for this in learning include simulations, such as in a manufacturing environment where one "moves" around the factory "operating" the machinery. (Hall,1997:111). Current problems facing the widespread adoption of VRML in LMSs are the necessity for a client-side plug-in to be installed on the learner's computer. It is hoped that this problem will be solved by the development of international standards for VRML, currently being drafted by the Web3D Consortium (Barron, 2003:12).

2.8 COOPERANT DEVELOPMENT TOOLS

Cooperant development tools are a kind of tool to organise source code in development. It is useful for large, distributed teams as well as for individual developers. The following diagram illustrates the work principium of cooperant development tools.

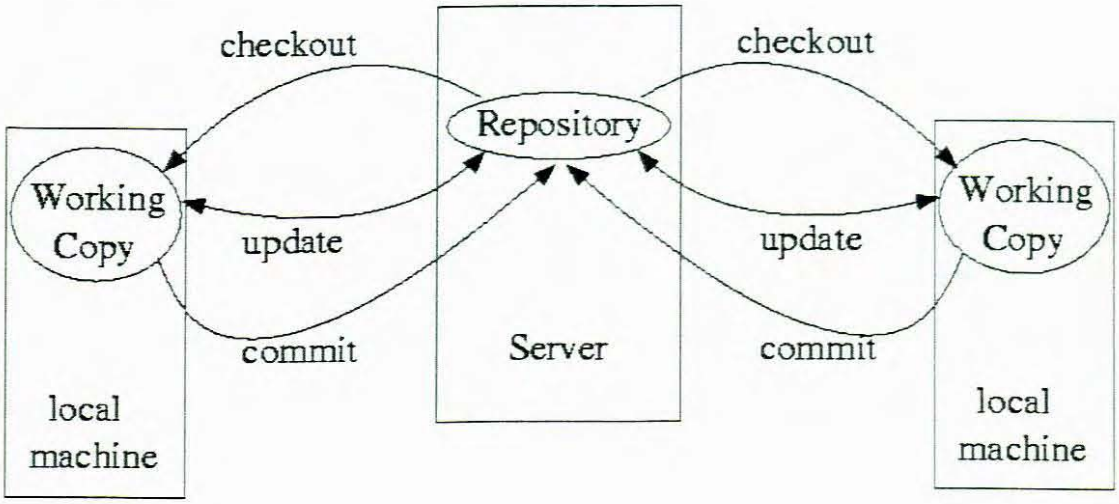


Figure 2.5. Principium of cooperant development Tools

There are also some third parties developing platforms for LMS content that are based on open-source:

2.8.1 TWiki

"TWiki is a leading-edge, Web-based collaboration platform targeting the corporate intranet/internet world. TWiki fosters information flow within an organization; lets distributed teams work together seamlessly and productively; and eliminates the one-Webmaster syndrome of outdated intranet content" (Internet resource 7).

2.8.2 CVS

"CVS is the Concurrent Versioning System. It is a commonly used way of storing source code because it keeps versions of all files so that nothing is ever lost, and usage by different people is tracked, it also provides ways to merge code if two or more people are working on the same file. All code and all versions are stored on a central server" (Internet source8).

2.9 SUMMARY

Whatever the kind of software, software development methodology is essential to the software development process.There are useful descriptions available of the fundamentals of the open-source project development process, and also of the widely know software development methodologies. Further we introduced current areas of LMS in an open-source environment, industry standard of LMS, relative delivery technologies and developing environments.

It moves forward to the AMM developing process. In the next section the life-cycle paradigm will be executed and examined, fulfilled in open-source environment.

CHAPTER3 SOFTWARE PROJECT MANAGEMENT PLAN FOR ASSIGNMENT MANAGEMENT (SPMP FOR AMM)

3.1. INTRODUCTION

This chapter defines the working relationship between the students that comprise the Information Technology Department Masters Degree Project of the Peninsula Technikon (ITMPT), and AVOIR. AVOIR is an ambitious collaboration of several African higher education institutions initiated by the University of the Western Cape. "it is about harnessing the enormous potential that exists within Africa and the African Diaspora to create a core of open-source software developers who, through software development activities, aim to create educational and business opportunities that contribute to development on the continent"(Deark, 2004:2). The application is the Assignment Management Module (AMM), which is a module of the Learning Management System (LMS). The developer, who is a member of ITMPT development Group, will present the application. All details regarding the layout and management of this Module are found in this chapter.

3.1.1. Purpose

This chapter clearly captures the internal processes used within the Assignment Management Module (AMM) to accomplish project goals. The layout and management of the entire project to be developed by ITMPT is also discussed. The reader of this Software Project Management Plan (SPMP) will know how the developer intends to control and manage the development of the proposed application to be delivered to the LMS, and what the AMM is supposed to do.

3.1.2. Scope

Details concerning the management process, standards, and procedures are given in this chapter. Details concerning the structure of ITMPTG structure, individual roles and responsibilities, project planning and tracking mechanisms, as well as risk management are included where appropriate.

3.1.3 Overview of contents of chapter

This subsection briefly describes each of the remaining sections in the chapter. The rest of the chapter is organised as follows: The project overview section identifies why a LMS needs an AMM. The project organization section describes the modeling approach and organizational structures used to develop the AMM. The section on project management and control gives a detailed breakdown of risk and change management, as well as o scheduling and issue control. The technical process section lists the methods, tools, and techniques, which the developer uses throughout the project lifecycle and the formalities which have to occur throughout the paper process. The activities, schedule and budget section gives additional information regarding time and financial constraints on the project.

3.2 PROJECT OVERVIEW

This section contains information about the project of AMM, required deliverables, and this chapter itself.

3.2.1 Project summary of AMM

AMM is the core module of the LMS. The development process is composed of multiple parts, which are in turn divided into phases. Each phase is accompanied by a major chapter which has to be created, reviewed, and approved by the Mr Bennett Alexander. Mr Bennett Alexander is the Internal Supervisor of ITMPT. The purpose of each chapter is to narrow the approach and design developers use to manage, create and implement the project throughout its lifecycle. Because the AMM is part of the LMS, the project being offered by the developer is the AMM of the LMS. In order for LMS to take full advantage of the AMM being offered, reviews between the ITMPT and developers take place bi-monthly. The purpose of these meetings is to synchronise all expectations and constraints between the two parties.

The first phase of the project lifecycle is entitled the Software Project Management Plan (SPMP) chapter. Its purpose is to capture the internal management processes of AMM clearly so as to accomplish project goals. After the SPMP is presented and approved, the next phase entitled the Software Requirement Specification (SRS) is created. The purpose of the SRS is to determine the development requirements for the intended solution. The next phase is entitled the Software Architecture Design chapter (SAD) and it describes the layout of the solution the developer intends to use for the problem at hand. Upon completion of this phase comes the software coding, which is produced concurrently with the last three phases. The System Tests and Results phases lists the test runs and results before the product is accepted to ensure it meets the requirements. Afterwards, the Case study phase covers the implementation of AMM in KEWL.NextGen. Finally, the Engineering Report phase generates the dissertation for the ITMPT.

3.2.2 Project Deliverables

The sequences of products delivered over the life of the project are identified below, in Table 3.1 Project deliverables, along with the phase in which the product is delivered.

Phase	**Deliverable**
Project Management Plan	Software Project Management Plan of AMM
Software Requirement Generation	Software Requirements Specification of AMM

Software Design	Design Chapter of AMM
Implementation	Code
System Testing System	Test Chapter of AMM
Acceptance Testing	Software Test Report of AMM
Engineering Report	Generate Dissertation for ITMPT

Table 3.1. Project deliverables

3.3 ENGINEERNING METHODOLOGY AND PROJECT ORGANISATION

This section provides the reader with information on how the ITMPTG is organised, and with the engineering methodology of AMM including the waterfall model that is used. It also gives specific information on what individual ITMPT members are responsible for.

3.3.1 Engineering methodology

A modified waterfall model is followed throughout the project lifecycle. This model is ideal for the development of the system because it provides the ability to go back to the previous phase and rework mistakes or problems. See Figure 3.1 Modified waterfall model for a visual representation of the modified waterfall model. This Software Project Management Plan (SPMP) gives information on the project for the rest of the development of the system. The concept of the project is explored in Chapter 2 (literature review). Requirements for the AMM are gathered into the Software Requirements Specification (SRS). The SRS is referred to frequently during design and testing to ensure that the system adheres to the requirements. The Software Architecture Design chapter (SAD) is the primary source of information for the system code.

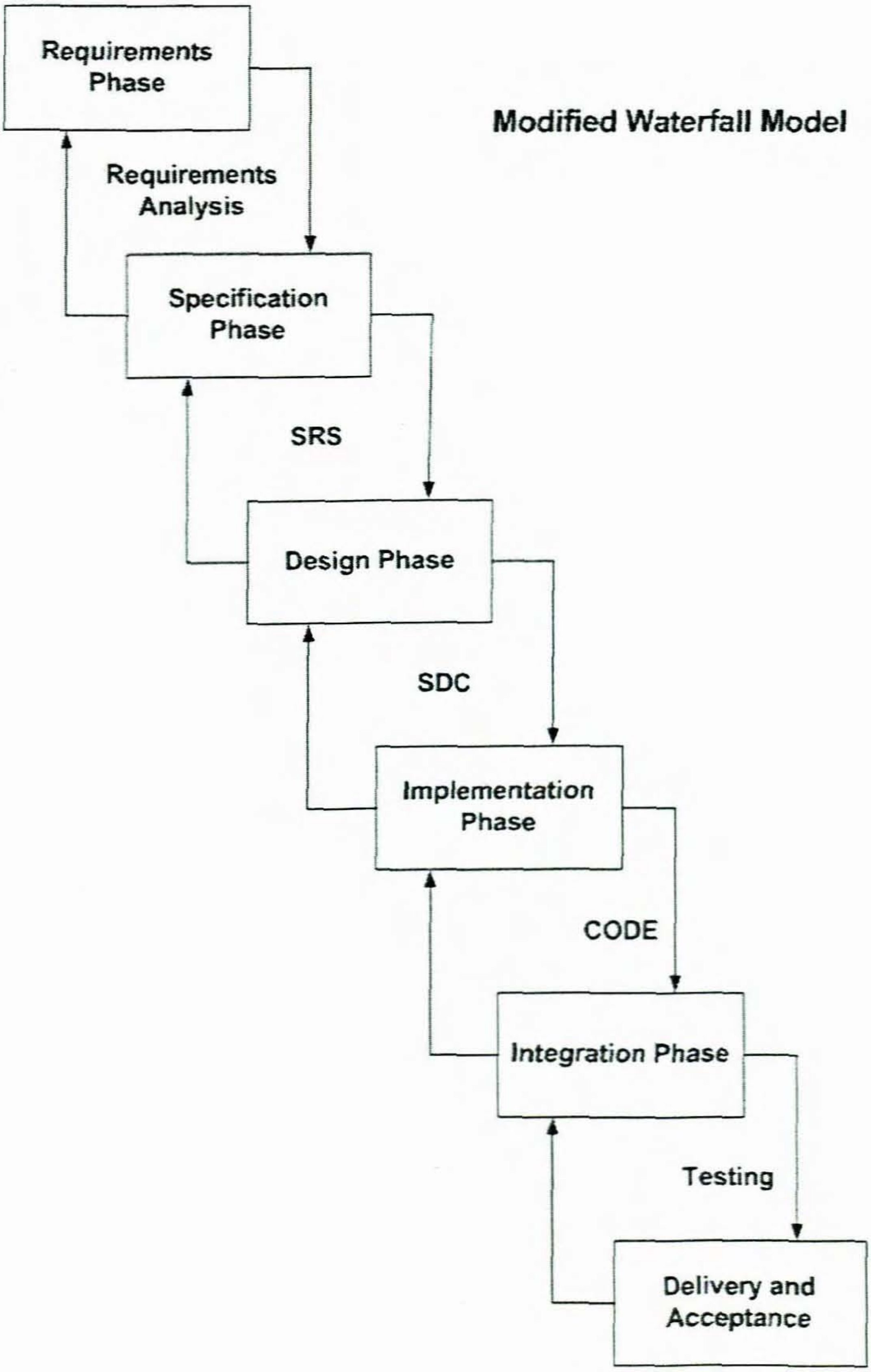


Figure 3.1: Modified waterfall model

3.3.2. Organisational structure and Interfaces

ITMPT Development Group is made up of seven members. Five of them are Masters students. They participate in the ITMPT, and have varying degrees of programming and technical writing experience. They are in charge of developing different modules of the LMS. Ke sun is a member of this group and he is in charge of developing an AMM. Team members have been assigned a role according to their personal strengths.

Currently acting as the Team's Project Manager is Mr. Bennett Alexander. His duties include organizing tasks, delegating parts to the team members who can best accommodate the workload, and holding Team members accountable for the work they promise to perform. Acting as Executive Secretary for ITMPT is Reneat. The roles that Reneat is responsible for include taking notes at peer review meetings, IT Department supervisor meetings, and developers meetings. These notes are posted to a website and individual members of ITMPT can access the information, so that follow-ups can be performed for any task delegated or discussed in previous meetings.

ITMPT has made it a high priority to be on time with all project phases, and therefore the schedule arrangement by developers is crucial. It is Ke sun's responsibility to analyse the time required for an AMM phase and to outline an allotted time in which the development process should start and end. In peer review meetings, this Project Schedule for an AMM is often referred to, to measure progress and to offer suggestions in order to add more benefit to the efficiency with which the developer performs. Further, it is also Ke sun's role to assimilate individual hours worked each week on the AMM and to present them to the Internal Supervisor. These hours are used to measure overall work performed on the project for the AMM so far. Each position is important in its own right. The developer is responsible for the overall look, feel, and readability of all chapters presented to the Internal Supervisor. The technical content supervisor, Mr Bennett Alexander, is responsible for the feasibility of all Project design structures and implementation schemes. Mr Bennett Alexander is also responsible for implementing open-source packages and linking them to each other in the testing environment. The following table, Table 3.2 Roles *of major activities, displays the roles of each phases.*

Major Activities	**AUTHOR**	**SUPERVISOR**
Software System Proposal Of AMM	Ke Sun	Mr.Benett Alexander
Software Project Management Plan Of AMM	Ke Sun	Mr.Benett Alexander
Software Requirements Specification Of AMM	Ke Sun	Mr.Benett Alexander
Software Design of AMM	Ke Sun	Mr.Benett Alexander
Code	Ke Sun	Mr.Benett Alexander
System Test Chapter Of AMM	Ke Sun	Mr.Benett Alexander
System Test Report Of AMM	Ke Sun	Mr.Benett Alexander
Dissertation For ITMPT	Ke Sun	Mr.Benett Alexander

Table 3.2. Roles of major activities

3.4 PROJECT MANAGEMENT AND CONTROL OF AMM

This section describes in detail how the plan is kept current, how the project is managed, how progress is measured, how schedules are tracked, what specific methodology is used for software development, and how verification and validation are conducted. It also discusses the method the developer intends to use to deliver the system.

3.4.1 How the plan is kept current

If the need arises to update the Management Plan, or any other sections, a change request is given to the developer. The developer makes the necessary changes, and appends information about the change to the end of the chapter.

3.4.2. How the project is managed

ITMPT members rely on daily communication to ensure that the project is being well-managed. At least once a week the internal supervisor keeps the developer up-to-date on current progress and upcoming deadlines. The Project Manager delegates tasks to an appropriate developer based on the schedule, the developer is then held accountable for completing these tasks on time. Figure 3.2 AMM project management plan depicts the project management cycle that is used within the development process of AMM. Mr Bennett Alexander currently assumes the ITMPT project management role and tasks are delegated to the appropriate members according to job title and emphasis strengths. As these tasks are completed, a peer review between group members occurs where suggestions and special details are given. When all tasks are completed, the chapter is synchronised as regards grammar usage and reading flow. Once the chapter is in its first draft form, it is presented to the internal supervisor and returned with further suggestions for revision. The developer resynchronises the chapter after the revisions have been completed and a second draft is presented. The department supervisor may return minor revisions and the chapter is presented to the users after these revisions have been addressed. A chapter is considered to be in the baseline stage once it has been signed by the internal supervisor. Once a chapter has reached the baseline stage, the developer starts working on the next chapter.

AMM Project Management Plan

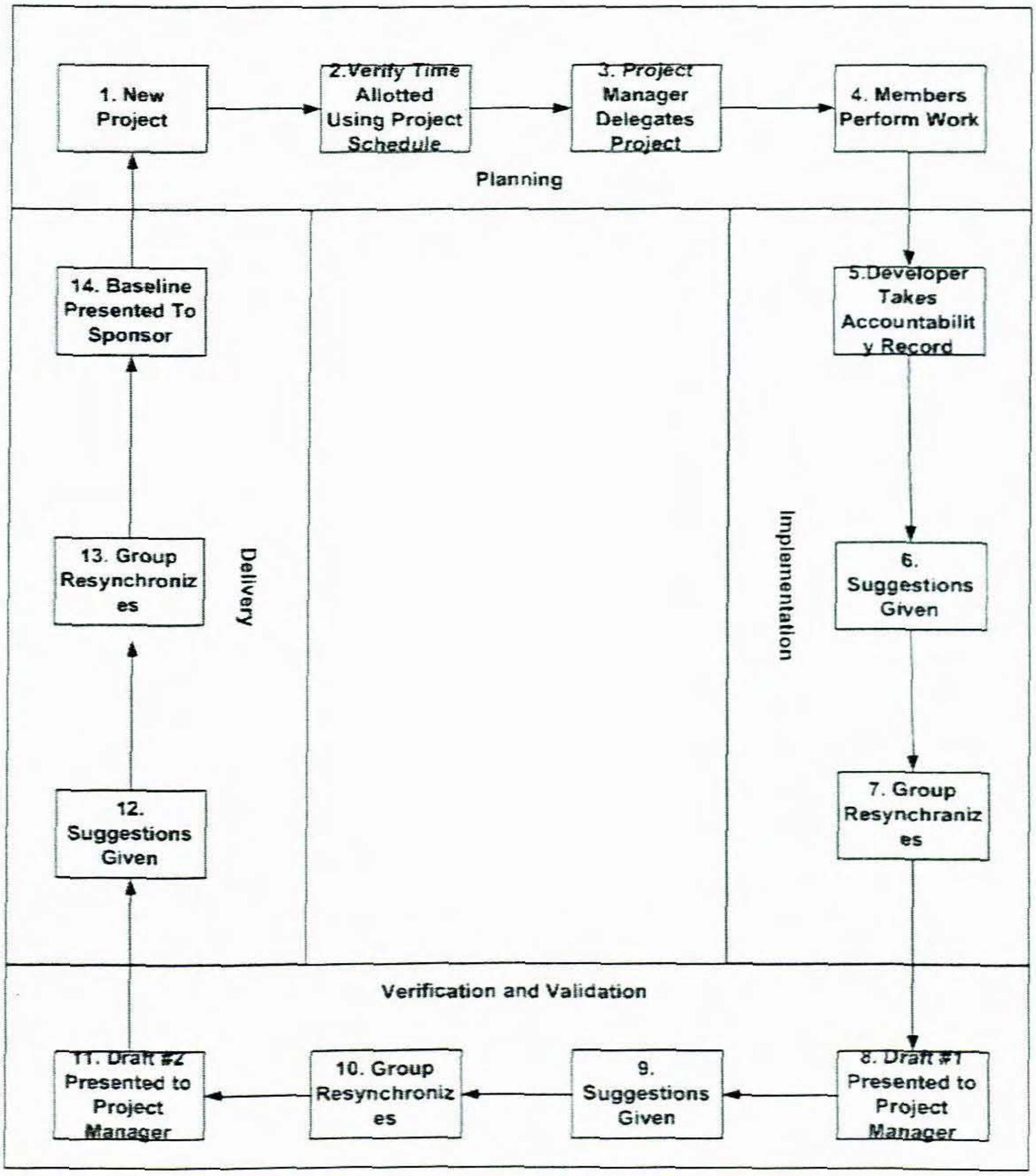

1. New Project
2.Verify Time Allotted Using Project Schedule
3. Project Manager Delegates Project
4. Members Perform Work
Planning
5.Developer Takes Accountability Record
6. Suggestions Given
7. Group Resynchranizes
Implementation
8. Draft #1 Presented to Project Manager
9. Suggestions Given
10. Group Resynchronizes
11. Draft #2 Presented to Project Manager
Verification and Validation
12. Suggestions Given
13. Group Resynchronizes
14. Baseline Presented To Sponsor
Delivery

Figure 3.2. AMM project management plan

3.4.3 How progress is measured

A critical aspect of the SPMP concerns the completion of work products. Progress is measured as a function of parts completed for the deliverable in question. These elements are delegated by the internal supervisor and are reviewed, refined and revised in department meetings. The Project Schedule tracks where start and deadline dates have been suggested for each phase and is used to ensure the developer completes tasks before they are to be delivered. The date on which a work product is deemed complete is termed a milestone. In order to determine whether a work product has indeed reached a milestone, it first has to pass a series of reviews performed by ITMPT Team members, the developers and the internal supervisor. A typical milestone is the date on which the design is completed and passes review. Once a work product has been reviewed and agreed upon, it becomes a baseline and can be changed only through formal procedures. After the HOD of the IT Department has signed the chapters for baseline validation, the developer moves into the next required stage of the process cycle.

3.4.4 How schedules are tracked

To ensure that deadlines are being met, the developer of AMM will keep schedules current and up-to-date by recording the progress of the development in each phase.

3.4.5 Specific methodology used for software development

The following is a collection of techniques adopted by the ITMPT throughout the complete lifecycle of the AMM:

- discover the customer's needs and desires;
- define flexible functions to accommodate these needs;
- propose these solutions to the customer through formal communication chapters and contracts;

- remain open for customer feedback, comments, and revisions;
- design the functions that fall within the scope of AMM to be integrated inside LMS;
- present a testing version to be tested by the customer for functionality and approval;
- revise any components that need additional attention; and
- present the finished product with all required documentation.

3.4.6 Conducting verification and validation

As seen in Figure 3.4.2, verification and validation occurs as a series of steps involving peer reviews, developers, and eventually an internal supervisor. Each phase may occur multiple times before the final version is created and all parties have come to a mutual agreement.

Verification of the project's conformance to users' requirements is a high priority in user reviews. Each need is re-evaluated to assure that features and functionality of the system cover the users' requirements. Amendments to this SRS are made if any additional functions or features are needed, after which they are agreed upon by the users and the developer.

The developer is responsible for holding the users accountable to suggest whether these needs are covered or whether additional functions should be implemented to cover any special cases the users may have.

3.4.7. Delivery plan of an AMM

I. Deliver and install module on LMS.

II. Test module by itself.

III. Test module after integration with the rest of the system.

IV. Have users verify that the module adheres to its requirements.

V. Go back to step #1 until the entire system has been integrated, tested and accepted.

3.4.8 Project management objectives and priorities

The goals of this project are to learn as much as possible about the technology used in the project and about software engineering in general. In order to do this the developer works hard to plan ahead by scheduling meetings that make effective use of Team members' time. The developer also practices and examines waterfall engineering methodology through the development process of an AMM.

Meetings are seen as a high priority. They are scheduled in the following manner:

- regularly scheduled meetings take place on Saturdays;
- online Teleconferencing meetings are scheduled as needed;
- phone bridge meetings are conducted at the onset of any new phases, and as needed thereafter;
- late night school meetings occur before the final draft of any phases is due;
- the developer meets with supervisor Mr Bennett Alexander every Saturday between 09:00 and 12:00 to learn about software engineering, and to be briefed on details of the work we are doing;
- once a week the developer reviews the progress of the AMM in order to get tips on how to improve the phases;
- the agenda for the meetings is set at the preceding meeting when possible, but time is always set aside to work on any new issues that may arise. The agenda is posted in the 'minutes' section of the ITMPT website; and
- the developer will keep track of the hours he spends on the AMM, These hours are then submitted to Mr Bennett Alexander at the end of the week.

3.4.9 Assumptions, dependencies, and constraints of an AMM

The developer assumes that the need for a developing a document of existing LMS really exists, and that because of this need, existing LMS will work in a cooperative effort to help the developer complete all aspects of the requirements of the AMM. Additionally, the developer is assuming that MySql is a viable option for a database. The developer also assumes that users of the LMS use common Web browsers, such as Netscape Navigator and Microsoft Internet Explorer. Furthers, that users of a LMS have an existing email system that the developer could use to send Simple Mail Transfer Protocol (SMTP) mail. Finally, the developer assumes the AMM is intended for a group of 100 users or less.

The developer uses open-source code when deemed valuable to extend the functionality of the application. The developer is required to create original code and configuration parameters in order to make the AMM system unique and to meet the user's specific needs. The system may require substantial time to learn and install these packages; there would be greater limitations to the complexity of the features included in the AMM System. LMS utilize open source products whenever feasible and develop and integrate the AMM product into their existing systems.

It is also critical that the AMM system should not be dependent on a product where there could be problems integrating and using the software.

Neither ITMPT nor the developer assumes that the implementation of software is required. The team will acquire a server for use during the development of the product of an AMM.

At the time of delivery, a LMS will provide a server environment where the product and any other required software are installed.

The developer will provide detailed documentation of the AMM to show how database schemas can be easily be created inside the delivered source code, and how each portion of the system is used. Once ITMPT completes the primary deliverables, the team will also include documentation to describe how to extend the capabilities of the AMM.

3.4.10. Risk management of AMM

Table 3.3 Project risk management summarises the possible risks and solutions that could be encountered in the development project.

Risk:	**Solution:**	**Severity:**
Loss of ITMPT Member of AMM	Role Switching	Serious
Loss of Data/Work	Backup Plan	Minor
Sponsor not available	Continue the Project using current assumptions	Moderate
Deficiency in one or more of the other programs that we require (i.e. Electric Pen-basic Mark)	Seek an alternate software solution, if this fails, negotiate a change in the requirements with the Sponsor	Serious
Loss of Sponsor	Seek an alternative Sponsor that can use as much of the work that we have already done on a new Project	Serious
Hardware failure	Set up a new box running Linux if we need to Moderate	Moderate
Missing requirements after the SRS had been baselined	Consult with the Supervisor and Sponsor, and update the SRS with a change form.	Moderate

baselined		

Table 3.3. Risk management of AMM

The ITMPT takes extreme precautionary measures to ensure the safety of all data produced of AMM for the developer. Through the incorporation of an automated process, redundant versions of all files are created on three physically separated servers. A batch process has been created which has two jobs running off the developer's computer every night. The first job, which starts at 03:00AM nightly, authenticates automatically into the group's remote file transfer protocol (FTP), 10.56.1.28/staff, and fully backs up all data to the developer. At 04:00AM the developer's computer synchronises with the group's practice implementation environment Linux Sever computer so that this third machine becomes aware of any changes made to the 10.56.1.28/staff directories. Through this method, three redundant fail-safe versions of ITMPT's data are created nightly.

3.4.11 Schedule control of AMM

Current progress of the task at hand is compared against the baseline for the particular task. As development progresses, the team makes estimates about the time needed to finish the task. As the team gets closer to the baseline, these estimates will govern the pace needed to meet the baseline. Once a task meets the baseline in the estimated time, the task is marked finished and development moves on to the next task. However, in the event the team needing more time than the baseline, the rollover will affect the baselines for the remaining tasks.

3.4.12 Issue resolution of AMM

One-on-one (personal) issues should be dealt with privately. If those problems cannot be solved, the problem is brought to the group's attention and the group works on solving it. Technical issues need to be agreed upon by the developer and the supervisor if necessary. The Project Manager holds one-on-one conferences with the members of ITMPT to discuss any issues that may come up, and to ensure that all members are working to the best of their abilities. ITMPT intends to strive towards mutually beneficial decisions for all parties involved in the outcomes.

3.5 TECHNICAL PROCESS OF AMM

This section defines the general technical issues of the project, and describes the tools and techniques that the developer needs to develop the system. It specifies the development methodologies and programming languages, and also identifies what tools and techniques are used to specify, design, build, test, integrate and deliver the project's work products of the AMM.

3.5.1 Formal project chapters of AMM

All the formal project chapters of AMM (SRS, SAD, etc.) are written in and managed using Microsoft Office 2000. Chapters with low complexity are written in Word. More complex chapters are written in Word plus other Microsoft applications, and are combined using Microsoft Binder. The change control features within Office 2000 are utilized to assist the group members in identifying and tracking changes. However, the security features are not utilised; passwords within chapters are not set because the group members felt that this would not enhance productivity.

The Project chapters are stored on the group server at IP address10.56.1.28. A new directory is created to contain each formal chapter. All files, which are required to build the formal chapter, are stored in this directory. Filenames are of the form "DOCID_REV.doc" DOCID is the acronym of the chapter (e.g. SPMP) followed by an underscore and a two-digit revision number.

All group members are required to contribute to each formal chapter. This presents a problem with managing change control. If two group members download, modify and upload a change to a chapter, there is a chance that one person's work may be lost. To minimise the chances of this occurring, each group member must follow this process:

I. inform other group members using both email and ICQ that you intend on working on a chapter;
II. download the largest revision number file;
III. make changes;
IV. make certain that someone else has not uploaded a larger numbered file;
V. upload changed file; and
VI. inform other group members, again, using email and ICQ, that a new version has been posted.

3.5.2 Software package configuration files of AMM

Configuration files control the behavior of the software modules, which make up the AMM. For example, the configuration file, which controls the Apache server, is named httpd.conf. It is managed with the Concurrent Versions System (CVS). The process that group members use with CVS, is as follows:

I. check out a local copy of the file. Either use the cvs checkout <directory> command from a LINUX workstation, or use the Windows WinCVS (http://www.wincvs.org/) to check it out;
II. make the necessary changes to the local copy using an approved editor;
III. before checking in the changes, run the cvs update to examine any changes have been committed by other team members; and
IV. commit the changes with the cvs commit command.

3.5.3 Web based help files of AMM

The Web based help files are written using Microsoft Word 2000. They are managed just like the formal chapters with respect to version control. They are kept on the ftp: 10.56.1.28//ITMPT.homepage.com server and copied to the Linux server.

3.5.4 Web template files of AMM

The Web template files are used to create a consistent look-and-feel to all the AMM Web pages and forms. The template files are treated in the same ways to source code with respect to version control. They are edited with a text editor and managed with CVS.

3.5.5 AMM Software Including inc and PHP driven scripts

As users access the AMM system, they activate the PHP processing scripts, which are to be written in PHP. Additionally, there are some regularly public Class or Function files created, which are also activated by PHP. These files are written in PHP as .inc scripts. Both the PHP scripts and the inc scripts are managed using CVS as described in the "Software Package" section above.

3.5.6 Development script of AMM

The software documentation of AMM is kept in the same servers and directories as the source code itself. It is managed using the same tools as the source code. In fact, the software documentation will become the code. The HTML template files contain comments, which are the software documentation. The technical chapter manager provides several template files that include examples. The templates are specialised for specific types of tasks. For example template-facetemp.inc is a template for building interface based PHP programs. Another example would be template-db.inc, which is a connection to the MySQL database server. Included in the template are sections and examples for documentation, including .php and .inc. Each template contains instructions regarding how to re-name the template, check it into CVS, and how to make changes. The dates on which the design was reviewed, who reviewed it, and when it was baseline approved appear within each file. Any changes beyond the baseline must also be reviewed and approved.

3.6 ACTIVITIES, SCHEDULE, AND BUDGET OF AMM

This section outlines the tasks that need to be performed throughout the project of an AMM, and how the developer manages the budget and schedule of the AMM project.

3.6.1 Activities and tasks of AMM

The sequence of work activities to be performed over the life of the Project of an AMM is identified below in Table 3.4 Tasks to be performed. Each phase contains a set of tasks that lead to the completion and approval of a baseline product for that phase. For a graphical representation of the project schedule see Figure 3.3. Project Gantt chart.

Phase	**Task**	**Deliverable**
Conceptual study and literature review of AMM	Identify project. Prepare project proposal	Software background and literature review chapter
Project Management Plan of AMM	Plan and prepare SPMP	Software project management plan
Software requirement Generation of AMM	Complete requirement analysis Prepare technical specification Prepare SRS	Software Requirements Specification
Software Design of an AMM	Prepare architectural design	Software Design Chapter

	Prepare detailed design. Prepare SDD	
Implementation of AMM	Write code. Perform unit testing. Perform integration testing	Code
System Testing of AMM	Prepare system tests. Perform system testing. Prepare STD and STR	System Test Chapter, Results
Engineering Report	Write whole document for project of AMM	Dissertation for ITMPT

Table 3.4. Tasks to be performed

3.6.2 Resource requirements of AMM

The ITMPT student on the AMM project (Ke sun) spends an average of 20 hours a week on the project, making a total of 3200 man-hours by the time of the completion of the project. He also needs access to a high quality printer and paper to print the deliverables required throughout the project of the AMM. On the days ITMPT convenes with our internal supervisor Mr Bennett Alexander, meetings generate unseen costs incurred through Mr Bennett Alexander, which results in an OSS of effective productivity time. Potential miscellaneous unseen costs may also occur due to meeting room occupation and scheduling.

3.6.3 Budget of AMM

Table 3.5 defines potential real costs to ITMPT through the development of the Software AMM for an existing LMS.

Direct Costs	Labour costs: 32 weeks * 20 hours/week *	R19 200,00

	R30/hour engineers	
Indirect Costs	Transportation cost: 30 miles * 2 ways * 1 cars * 12 month	R360,00
Equipment Costs	Tape Backup Device/Media: Software: Open source code Hardware: Linux server lease Printed Chapters at 16 user	R 18 000,00
Total Potential Incurred Costs		R37 560,00.

Table 3.5. Budget for an AMM

3.6.4. Schedule

See Figure 3.6.4 for the project Gantt chart, which shows the schedule breakdown of the project in a graphical format.

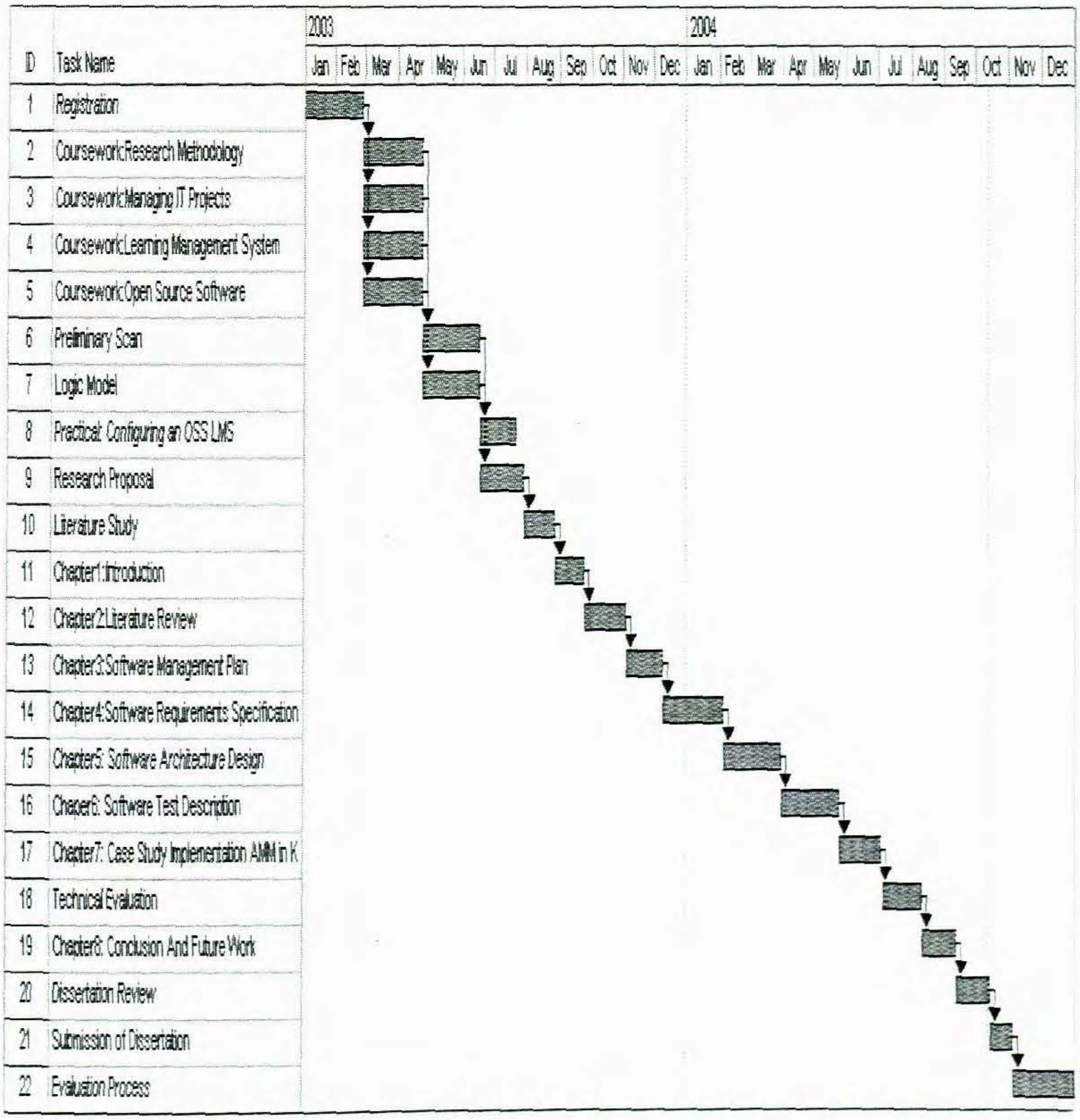


Figure3.3. Project Gantt chart

3.7 SUMMARY

This chapter introduced and provided background information on the project management plan for the application development process. A project overview was given, software engineering methodology and project organisation were discussed, and project management and control of the AMM were described in detail. Further areas of discussion included technical processes of the AMM as well as activities, schedule and budget of the project.

CHAPTER 4 SOFTWARE REQUIREMENTS SPECIFICATION FOR AMM (SRS FOR AMM)

4.1 INTRODUCTION

This chapter defines the software requirements specified by the users of LMS and the developer. The developer must use these specifications to create the AMM application for the LMS in open-source environment, which serves as the ITMPG for the Peninsula Technikon. The developer of ITMPG and users of LMS are provided with a specific implementation of AMM, which is referred to as the AMM. Throughout this chapter, specifications given are met by this AMM implementation. All details regarding requirements and functionality of this project are found in this chapter.

4.1.1 Purpose

The requirement analysis is the first phase of waterfall methodology. This chapter explores how this is a fulfillment process in AMM, and captures what functions and features are required for AMM. The developer must use these specifications when making decisions regarding how to implement features so that project requirements are met.

4.1.2 Scope

Any required features or functions must be clearly noted throughout this chapter. Included in the Software Requirements Specification (SRS) is some discussion regarding implementation details, although that is not the primary purpose of this chapter.

4.1.3 Overview of contents of chapter

This subsection briefly describes each of the remaining sections in the chapter.

The rest of the chapter is organised as follows: The general description section gives an explanation of application functions and the constraints, assumptions, and dependencies upon which these functions are based. A specific requirements section is composed of an external Interface requirements subsection describing the user, hardware, and software interfaces. Following that a functional requirements subsection describes all the design modules and functions, which will constitute the operational features of AMM. A performance requirements subsection then describes the operational environment placed on both the software and the users of the software. A design constraints subsection describes the constraints imposed by standards and hardware limitations. Following this, a quality attributes subsection scopes the integrity and security features of AMM. Finally, an other requirements subsection describes requirements that do not fall under the 'Specific Requirements' subsection.

4.2 GENERAL DESCRIPTION

This section contains information that provides insight into the specific requirements of Section three. The information provided describes the general factors affecting the application and its requirements. AMM is a software system that streamlines the processing of the administrative assignment required when new lecturer-users or student-users working on their assignments using the LMS. It allows an initial user set of assignments to be associated with a new user and it tracks these assignments through to their completion.

4.2.1 Application perspective

AMM is a network-based application. It requires that the computer on which it is installed be connected to a Transmission Control Protocol/Internet Protocol (TCP/IP) network. It also requires that there be several network services available. It depends on the operating system to provide several services and it depends on several software packages to provide key functionality. In the following sections of this chapter the subsystems and their dependencies are itemised and explained.

4.2.1.1 Networking

AMM uses a TCP/IP based network for users to access it. Users access AMM with Web browsers. In addition to Web access, AMM sends email messages to the Users via the Simple Mail Transfer Protocol (SMTP). Therefore, it is required that the computer on which AMM runs be connected to a TCP/IP network.

4.2.1.2 Network services

The AMM application does not directly make use of any network services.

4.2.1.3 Operating system

AMM has been designed to run on a computer running the Linux operating system, specifically on a computer running Apache version 2.0 and MySql version 3.10.

4.2.1.4 Software applications

AMM requires that a Structured Query Language (SQL) database be accessible and that AMM be granted sufficient privileges within the SQL database to operate. It is required that a Web server application (capable of binding to PHP) be installed on the computer on which the AMM is installed. The LMS implementation requires the use of version and authentication controls to provide secure authentication of user information. All code dealing with authentication is encapsulated into one module. Future versions of AMM need only make changes in this authentication module to alter the authentication method.

4.2.2. Application Functions

Due to the dynamic nature of each implementation of AMM, Table 4.1 Functionality requirements of AMM represents possible events and responses appropriate to the LMS implementation of AMM. This table does not limit the possible functionality of the LMS, but notes those functions that should be delivered by developer.

Functional requirement
Login function
Different user operation interfaces (administrators, lecturers and students)
Lecturer users and student users have access to their own preferences in the course.
Creating an assignment function
Assignment types include the online activity model and the upload-file model.
The lecturers can choose to allow resubmission of assignments before assignments are marked.
Assignments can be specified with a due date and a maximum mark.
Lecturers can login like students to preview the assignment.
Distributing the assignment function
Users can distribute their finished assignments by URL. The assignments are date-stamped.
Students can upload their assignments (any file format) to the server. They are also date-stamped.
Late assignments are not allowed, and are indicated clearly to the student as such.
Provide function to prevent multiple submissions.
Assignment feed-back function
Lecturer feedback information is appended to the assignment page for each student.
For each particular assignment, the whole course can be assessed (mark and comment) on one page in one form.
Offer electrical-pen mark-up tools to simulate normal pen. Lecturers are given help online to correct students' uploaded essays.

Use statistical presentation to display the mark of the assignment

Table 4.1. Functionality requirements of an AMM

4.2.3 User characteristics

The AMM system is implemented through a Web interface and will expect the user to be familiar with Web browsers, email uploading and hyperlinks. Online help should be available to assist users and administrators in the LMS, and further help is available via the paperbound software provided by the developer. Users must have the ability to control the frequency of application reminders regarding certain items so that LMS does not become a nuisance to individuals.

Based on the user's permission level, AMM will provide the respective views of the system. For example, a student-user view of the system will differ from that of a lecturer-user. Administrators of LMS have access to all assignments and direct access to the database that AMM facilitates in issuing queries. Groups of users can be created to facilitate some of the security features of AMM. A default user is linked to each Course that is created and can be modified through the administrative setup interface. Users have access to their own user preferences, and can create their own assignments; however they cannot delete or modify assignments created by an individual from another course.

4.2.4 General constraints

Environmental (or operational) limitations:

- AMM is designed for a single server implementation on the LMS. There is no backward compatibility with the earlier versions nor will the application be functional without the required external modules.

Hardware limitations:

- The AMM system should be installed on a computer running the Linux operating system. This computer should have at least 64 Mb of Random Access Memory (RAM) and at least 200 megabytes of free hard drive space to store the AMM application and the LMS. The existing computer is physically located at the ITMPT office building which houses a 100 Base-T switched Ethernet network and a Digital Subscriber Line (DSL) to the Internet.

Required interfaces to other applications include the following:

- AMM should communicate with the Web server via CGI;
- AMM should be able to interface with any relational database management system that is accessible via PHP and uses standard SQL;
- AMM should be written in PHP version 4. PHP is a scripting engine that binds to a Web server and creates the dynamic Web pages on which AMM is founded upon. PHP operates on the server side and allows for easy intervention from the SQL database; and
- The developer implementation of AMM should facilitate secure authentication of users. AMM should obey the LMS secure authentication principle because it is parts of the LMS.

Parallel operations:

- Parallel operations are limited to the abilities of the required software. MySQL executes multiple queries over threads rather than processes. By facilitating this modern operating system technique, this allows MySQL to operate on different tables simultaneously. Apache Web server supports a definable number of simultaneous connections and each individual connection has support for SSL and PHP.

GUI requirements:

- The GUI design used for AMM is consistent with the look and feel of the ITMPT Intranet. It is the desire of the developer to present all data, fields, and reports in a manner which is easy to read and understand. Options within the AMM application are accessible via mouse clicks and keyboard tabbing, as would any Web-based page or application. Screens will be organised to facilitate all functionality of AMM without having to go more than four levels deep in the menu-system. All customisable fields in AMM, including menu option names, button names, error messages and page titles will be easily modified through one file. All hard-coded value or priority levels will also be chaptered and customizable through this file. This will allow simple implementations of the AMM to be created in a short amount of time. For easy administrator intervention, an open-source GUI DBDesigner4 is used to interface with the MySQL database.

Compliance with certain specified standards:

- The following are standards which ITMPG has made an intentional effort to adhere by: HTTP/1.1, HTML 4, PHP 4.x, PERL 5.x, and MySQL 3.x.

Language requirements:

- The developer will use languages that are easily maintained by KEWL. The design of AMM is based upon the functionality of PHP. Use of the PHP language will allow easy functionality of Web-based queries and dynamic Web pages.

Protocols:

- The authentication mechanism is provided through the Lightweight Directory Access Protocol (LDAP). This will provide or deny users access according to the System User Database. Other protocols used are .inc PHP scripts to provide some basic functionality of AMM, SMTP for the email facility, SSL to provide security, and the Web browser uses HTTP.

Security:

- The LMS Apache Web server has an SSL certificate installed, which encrypts all packets communicating with AMM. It is the developer's intent to facilitate such a certificate to provide a safe means of security to all users of the AMM system. The LMS will authenticate valid AMM users. Neither data entered by users, nor data sent from AMM to the user will be stored on the client system. To prevent unauthorized access to AMM while a User is still logged in, all pages beyond the login page will expire within a configurable time period. This time is determined at the Administrator's discretion and is set in the same overall configuration file as the menu option names, button names, and error messages. The server will validate all data sent from the client. The data will also be guarded against tainting, and will not be sent directly to any system calls.

4.2.5 Assumptions and Dependencies

In Table 4.2, it is shown that application assumptions and dependencies details.

Assumptions	**Dependencies**
Parallel accessing is limited to the abilities of MySQL and the Apache Web server.	Administrator must have skill with PHP
MySQL is adequate as a database	Administrator should know basic SQL
Application can be implemented in PHP	Reasonable Version of PHP
All software applications perform as expected	A SQL based database system
	An LDAP compliant protocol

	An HTTP/1.1 compliant Web server
	200 MB of available hard drive space
	Existing Ethernet Network Connection in Place

Table 4.2. Application assumptions and Dependencies

4.3 SPECIFIC REQUIREMENTS

The design and approach of the LMS implementation (AMM) is specified for circumstances that require specific fields to be defined. In cases where hardware or external dependencies are involved, the AMM foundation is specified.

4.3.1 External interface requirements

This section describes in detail the requirements for how the AMM system interfaces with all outside entities. This includes users, hardware, software, communications media and protocols.

4.3.1.1 User Interfaces

The two main types of users for the AMM implementation include administrators, and all other users. In general, administrators must have the same interface as all other users with a few exceptions.

- administrators have access to all assignments, and direct access to the database; and
- users have access to their own preferences, and can perform only the assignment that the Administrator gives them permission to.

The output screen will have the following general layout:

Header

- Course name and logo
- Menu Options

Body

- Output of functions
- Fields will appear here

Footer

- The LMS logo and version
- Copyright information

The output screen will have the following format:

- Adhere to look and feel of course
- Reports are User defined
- Users must not see underlying SQL string
- Netscape and Internet Explorer compatible
- Straightforward approach to creating assignment
- Default options are provided in all appropriate circumstances

Relative timing associated with I/O:

- Querying attempts are dependent upon hardware and network traffic
- Results must appear within five seconds when users are working within the Intranet

Required Menu options:

- Assign the assignment
- User preferences

- Grade book, reporting and feed back information from teacher
- Setup

4.3.1.2 Hardware Interfaces

All hardware is accessed indirectly via the operating system or other software. In no cases does the software code communicate directly with any hardware devices, because of a lack of sufficient technical support from the hardware company (e.g. electrical-pen that is for assignment mark-up).

4.3.1.3. Software Interfaces

In Table 4.3, there is a summary of the software interfaces detailing the software interfaces employed.

Name of Application:	Version Number	Vendor Identification
Apache Web Server	2.0.48	Apache Software Foundation
MySQL	4.0.17	Community Developed
PHP	4.3.4+PEAR	Community Developed

Table 4.3. Software interfaces

4.3.1.4 Communications interfaces

The AMM system will use the following standard protocols:

- SQL—for connecting with the data stores
- LDAP—for authentication only.

4.3.2 Functional requirements

4.3.2.1 Context diagram

LMS AND AMM
Context Diagram
LMS: Learning Management System
AMM: Assignment Management Module

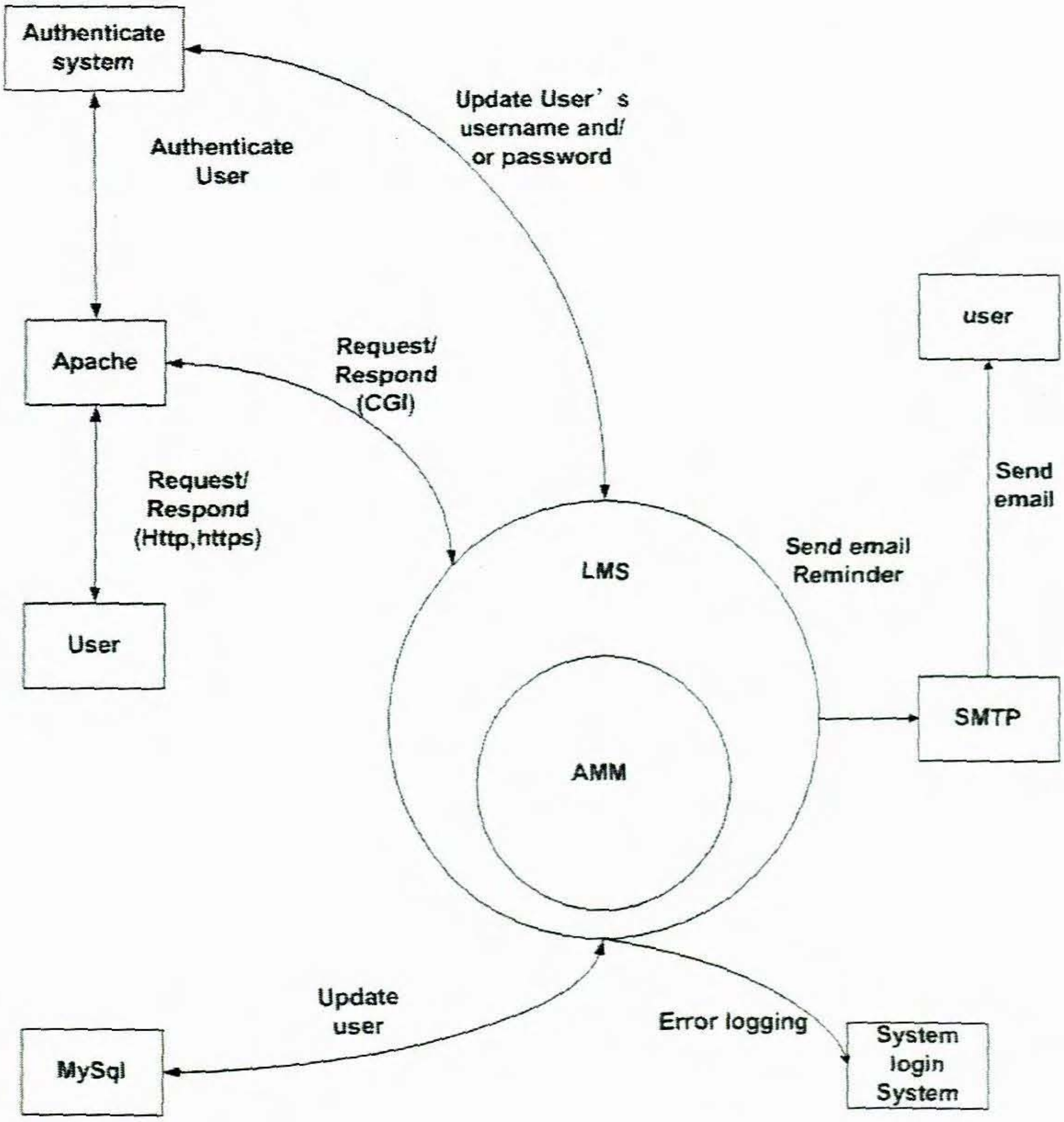


Figure 4.1. Context

4.3.2.2. Overview Diagram

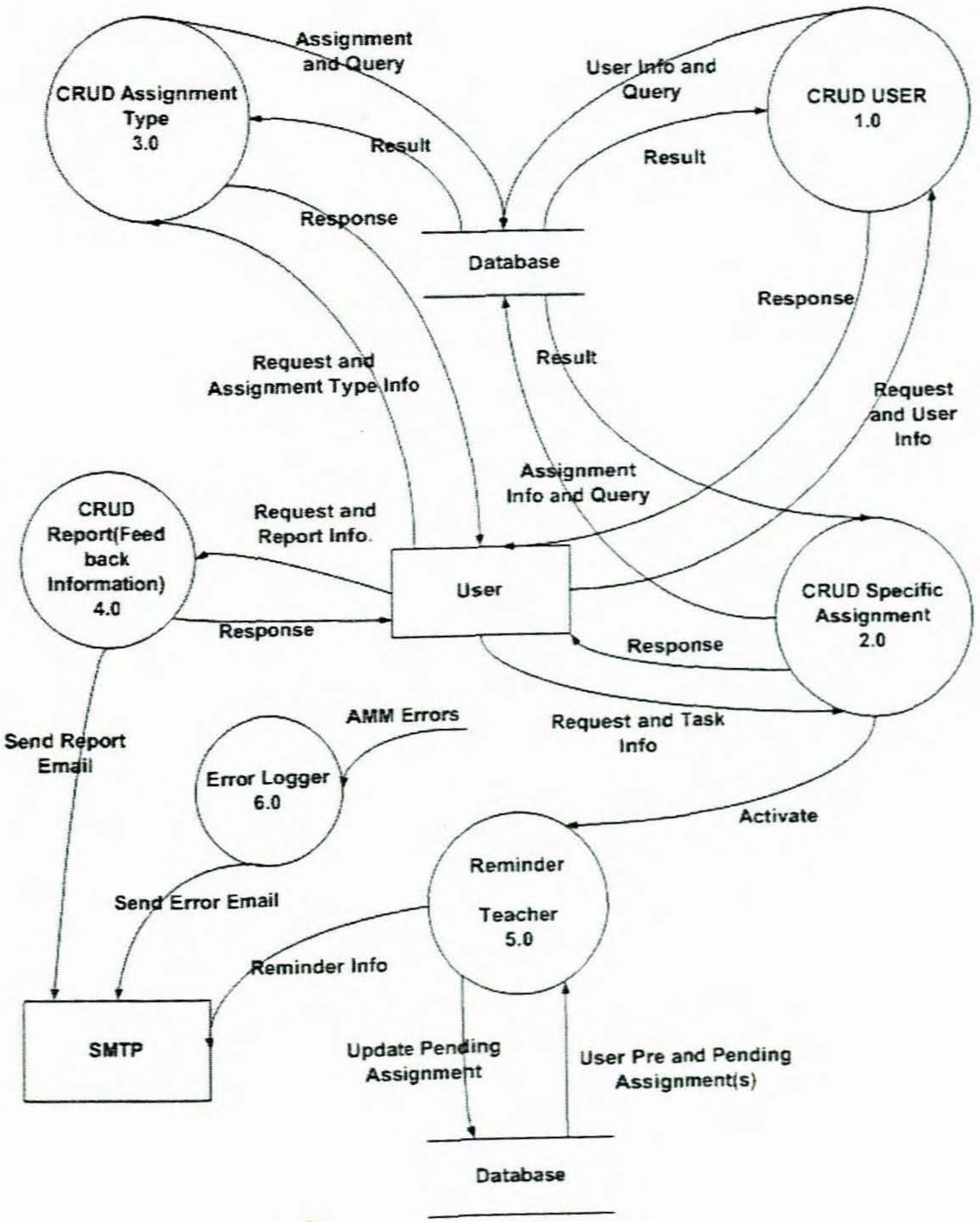
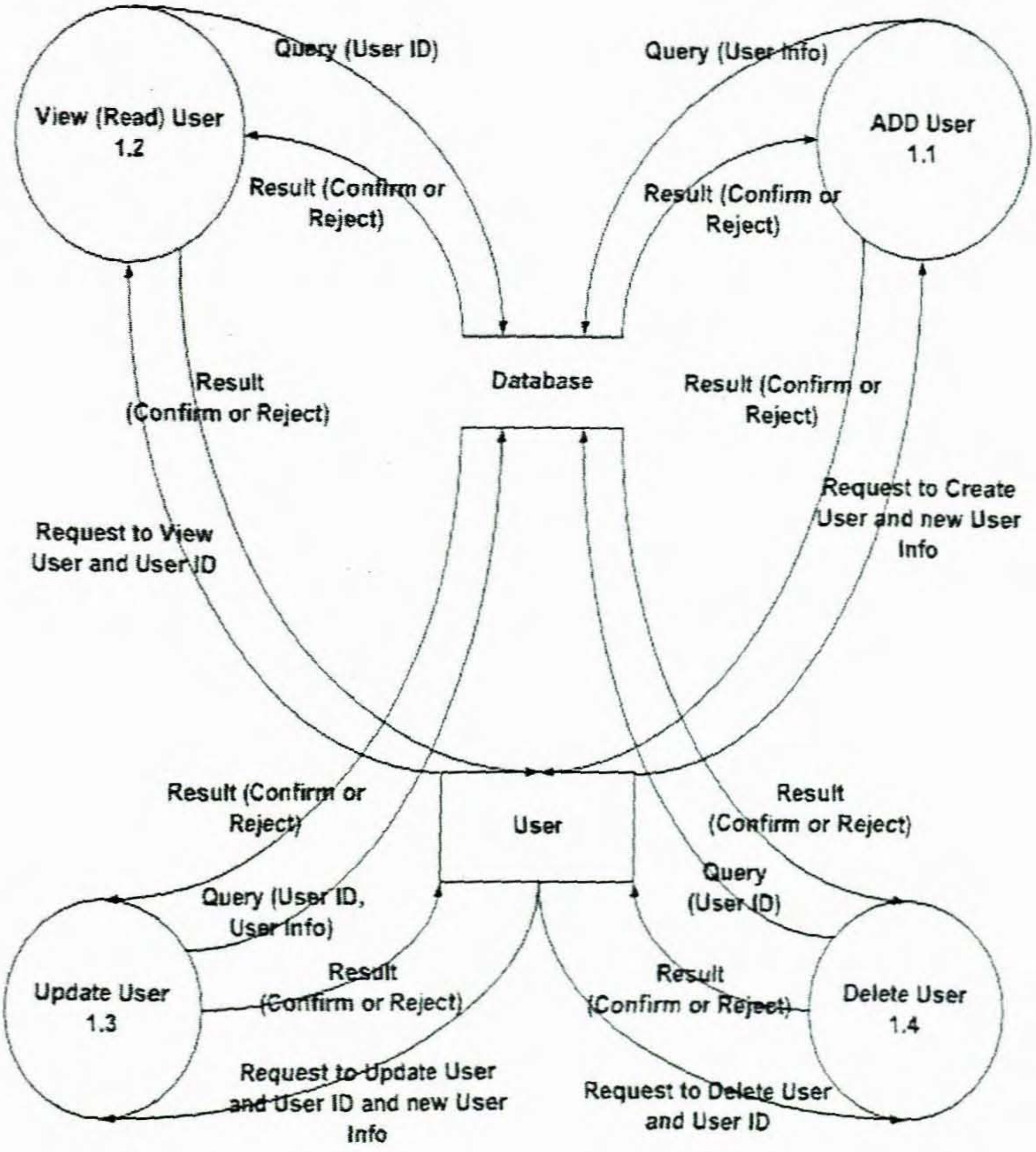


Figure 4.2. AMM level overview

Process 1.0—Users management



Query (User ID)
Query (User Info)
View (Read) User
1.2
ADD User
1.1
Result (Confirm or Reject)
Result (Confirm or Reject)
Database
Result (Confirm or Reject)
Result (Confirm or Reject)
Request to Create User and new User Info
Request to View User and User ID
Result (Confirm or Reject)
User
Result (Confirm or Reject)
Query (User ID, User Info)
Query (User ID)
Update User
1.3
Result (Confirm or Reject)
Result (Confirm or Reject)
Delete User
1.4
Request to Update User and User ID and new User Info
Request to Delete User and User ID

Figure 4.3. Users management

Process description 1.1 add user

When an administrator or user with the required permission level requests to add a user into the AMM, they must specify the new user's name, password, and group level. Add user will then perform an SQL query to add this user to the database, and display the results of the action to the user.

Purpose of add user

It is contended that AMM needs to be accessible to many users with different access (permission) levels, and should be able to support the addition of new users into the system.

Inputs of add user

The purpose of the function "Inputs of add user" is the first step in the field of the user management, and it represents the identity of the user. It could be categorized up four parts as below:

I. The first of all is New user's user name;

II. new user's password; and

III. new user's group/permission level.

Outputs of add user

The outputs of add user stands for the feedback of the system from the database. When a user does not commit all the categories of inputs of add user, and the database would perform unsuccessfully. The two operations are shown as below:

I. SQL query sent to the database to add the new user; and

II. results of the operation sent back to the user.

Process definition of add user

In order to add a new user entry into the database, add user must first check the current user's permission level to verify that they can legally perform the add. If the current user doesn't have permission to perform the add action, they will get an appropriate error screen describing why they couldn't perform the add action. If the user has permission to perform the add, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 1.2 view (read) user

When the current user requests to view the definition of a user in the AMM, they must specify the user's name, or user_ID. Read user will then perform an SQL query to return this user's information from the database, and display the results of the action to the user.

Purpose of read user

In the situation of the validation, users or administrators of AMM need to check out some information about a specific user or general users in the system.

Inputs of read user

user_ID and user name are necessary to obtain meaningful insight into read user. Each user has a unique user_id that does not mix up the results from the database, in case of the system meet the same user name. It is maintained that the user_id always be the primary key in the database.

Outputs of read user

If the SQL query send to the database for viewing a user or users, the function of outputs of read user can accept parameters, process some logic, and then return the results of the operation that are sent back to the current user.

Process definition of read user

In order to view a user entry in the database, read user must first check the current user's permission level to verify that they can legally view the user's information. If the current user doesn't have permission to view the user's information, they will get an appropriate error screen describing why they couldn't perform the read. If the user has permission to perform the read, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 1.3 update user

When the current user requests to update the definition of a user in the AMM, they must specify the user's name, or user_ID, and the new data to modify the user with (password, and/or group level). Update user will then perform an SQL query to Update this user's information in the database, and display the results of the action to the user.

Purpose of update user

Users or Administrators of AMM need to update the information on a specific user in the system. There takes a situation for example, they change the position from one department to another. It is a necessary condition to update them.

Inputs of update user

If a user wants to update the information, user_ID or user name must be supplied, and commit new user data which includes password, group level. The primary key that user_id could not be changed. Results from the database feedback immediately.

Outputs of update user

When a user commits the SQL query to the database in updating the user information, the system process some logical estimation, and then returns the results of the operation to the current user.

Process definition of update user

In order to update a user entry in the database, update user must first check the current user's permission level to verify they can legally update the user's information. If the current user doesn't have permission to update the user's information, they will get an appropriate error screen describing why they couldn't perform the update. If the user has permission to perform the update, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 1.4 delete user

When the current user requests to delete the definition of a user in the AMM, they must specify the user's name, or user_ID. Delete user will then perform an SQL query to delete this user's information from the database, and display the results of the action to the user. When a user is deleted, the system will delete all the relative information of his/her assignments from the database.

Purpose of delete user

Users or administrators of AMM need to delete specific users in the system when they should no longer be allowed access to AMM. This situation would be happened at a student graduated or a teacher retired.

Inputs of delete user

When users give some reasons to depart their positions away, only an administrator can delete the users' information, and results would be indicated by the Database.

Outputs of delete user

To delete a user, the SQL query would be sent to the database. When the database received this query, it depends on whether the operation was sent by an administrator or not. If it were an administrator's operation, the confirmation would be sent back, otherwise it would be not.

Process definition of delete user

In order to delete a user entry in the database, Delete user must first check the current user's permission level to verify they can legally delete the user. If the current user doesn't have permission to delete the user, they will get an appropriate error screen describing why they couldn't perform the delete. If the user has permission to perform the delete, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation. When a user is deleted, the system will delete all the relative information of his/her assignments from the database.

Process 2.0—Assignments management

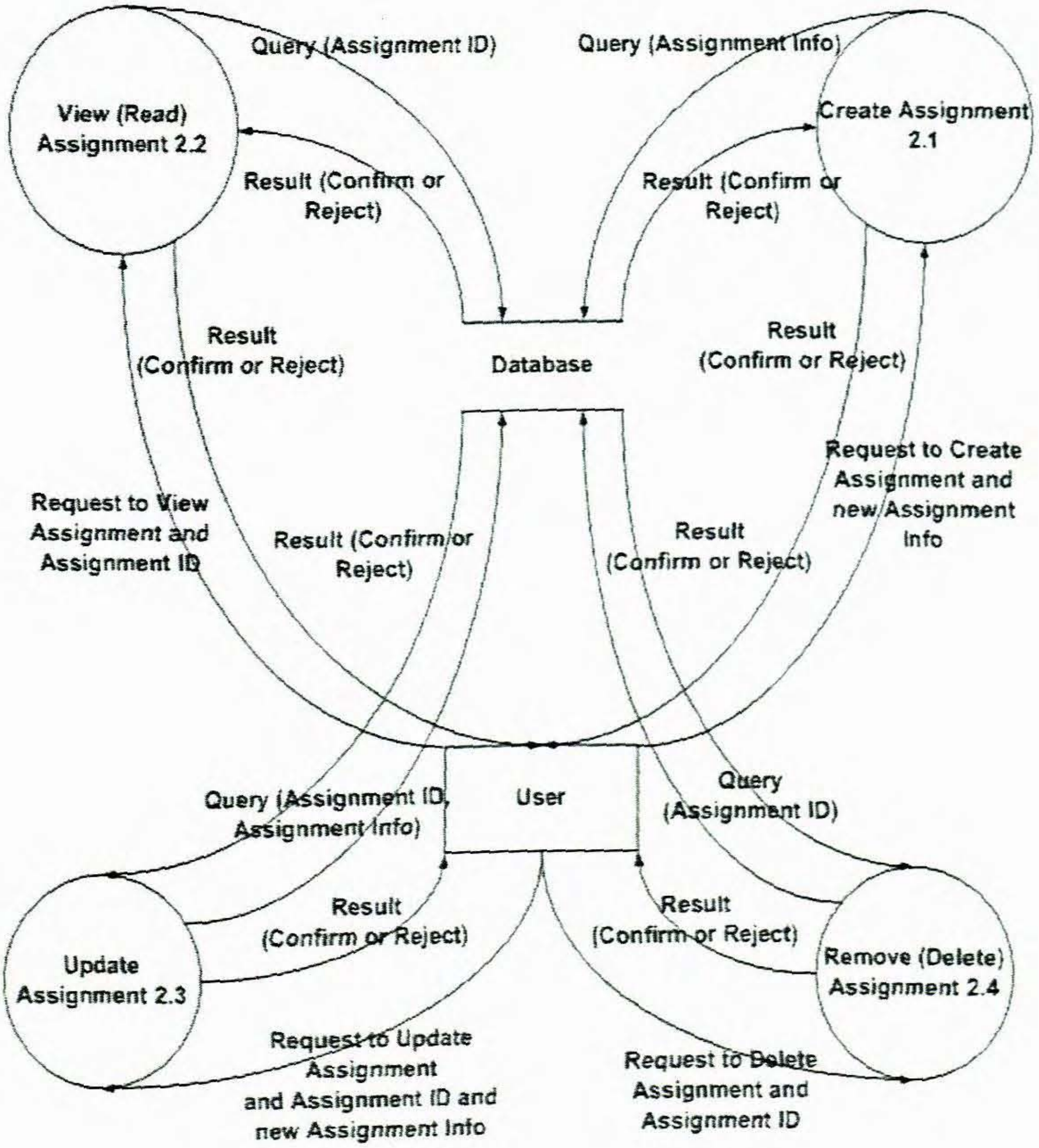

View (Read) Assignment 2.2
Create Assignment 2.1
Update Assignment 2.3
Remove (Delete) Assignment 2.4
Database
User
Query (Assignment ID)
Query (Assignment Info)
Result (Confirm or Reject)
Result (Confirm or Reject)
Result (Confirm or Reject)
Result (Confirm or Reject)
Request to View Assignment and Assignment ID
Result (Confirm or Reject)
Result (Confirm or Reject)
Request to Create Assignment and new Assignment Info
Query (Assignment ID, Assignment Info)
Query (Assignment ID)
Result (Confirm or Reject)
Result (Confirm or Reject)
Request to Update Assignment and Assignment ID and new Assignment Info
Request to Delete Assignment and Assignment ID

Figure 4.4. Assignments management

Process description 2.1 create an assignment

When an administrator or user with the required permission level requests to create a new assignment into the AMM, they must specify the new assignment information to be tracked. create an assignment will then perform an SQL query to add this new an assignment to the database, and display the results of the action to the user.

Purpose of create an assignment

The AMM needs to track the progress and completion of various assignments that users or Administrators defined.

Inputs of create an assignment

The purpose of the function "create an assignment" is the first step in the field of the assignments management. It could be categorized up two parts as below:

I. New Specific assignment Information; and

II. results from the Database.

Outputs of create an assignment

The outputs of create an assignment stands for the feedback of the system from the database. When an assignment has not been committed all the categories of inputs of create an assignment, and the database would perform unsuccessfully. The two operations are shown as below:

I. SQL query sent to the database to add the new assignment; and

II. results of the operation are sent back to the user.

Process definition of create an assignment

In order to create a new assignment entry into the database, create an assignment must first check the current user's permission level to verify that they can legally create the assignment. If the current user doesn't have permission to create the assignment, they will get an appropriate error screen describing why they couldn't create the assignment. If the user has permission to create the assignment, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 2.2 view (read) the assignment

When the current user requests to view the definition of an existing assignment in the AMM, they must specify the assignment_ID. Read Specific assignment will then perform an SQL query to return this assignment's information from the database, and display the results of the action to the user.

Purpose of read the assignment

Users or administrators of AMM need to get information on a specific assignment or assignments in the system.

Inputs of read the assignment

assignment_ID and assignment name are necessary to obtain meaningful insight into read specific assignment. Each assignment has a unique assignment_id that does not mix up the results from the database, in case of the system meet the same assignment name. It is maintained that the assignment_id always be the primary key in the database.

Outputs of read the assignment

If the SQL query send to the database for viewing an assignment or assignments, the function of outputs of read specific assignment can accept parameters, process some logic, and then return the results of the operation that are sent back to the current user.

Process definition of read the assignment

In order to view an assignment entry in the database, read the assignment must first check the current user's permission level to verify that they can legally view the assignment information. If the current user doesn't have permission to view the assignment information, they will get an appropriate error screen describing why they couldn't perform the read. If the user has permission to perform the read, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 2.3 update the assignment

When the current user requests to update the definition of a Specific assignment in the AMM, they must specify the assignment_ID, and the new data with which to modify the assignment. Update Specific assignment will then perform an SQL query to Update this assignment's information in the database, and display the results of the action to the user.

Purpose of update the assignment

Users or Administrators of AMM need to update the information on a Specific assignment in the system in order to have full control of the management of assignments.

Inputs of update the assignment

If an administrator or a teacher wants to update the information, assignment_id or assignment name must be supplied, and commit new data. The primary key that assignment_id could not be changed. Results from the database feedback immediately.

Outputs of update the assignment

When an administrator or a teacher commits the SQL query to the database in updating a specific assignment, the system process some logical estimation, and then returns the results of the operation to the current user.

Process definition of update the assignment

In order to update an assignment entry in the database, update specific assignment must first check the current user's permission level to verify they can legally update the assignment information. If the current user doesn't have permission to update the assignment information, they will get an appropriate error screen describing why they couldn't perform the update. If the user has permission to perform the update, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 2.4 delete the assignment

When the current user requests to delete the definition of an assignment in the AMM, they must specify the assignment_ID. Delete Specific assignment will then perform an SQL query to delete this assignment from the database, and display the results of the action to the user.

Purpose of delete the assignment

Users or Administrators of AMM need to delete specific assignments from the system when they should no longer be tracked by AMM.

Inputs of delete the assignment

When teacher gives some suitable reasons why an assignment would not be used, only a teacher can delete the assignment using assignment_id, and results would be indicated by the Database.

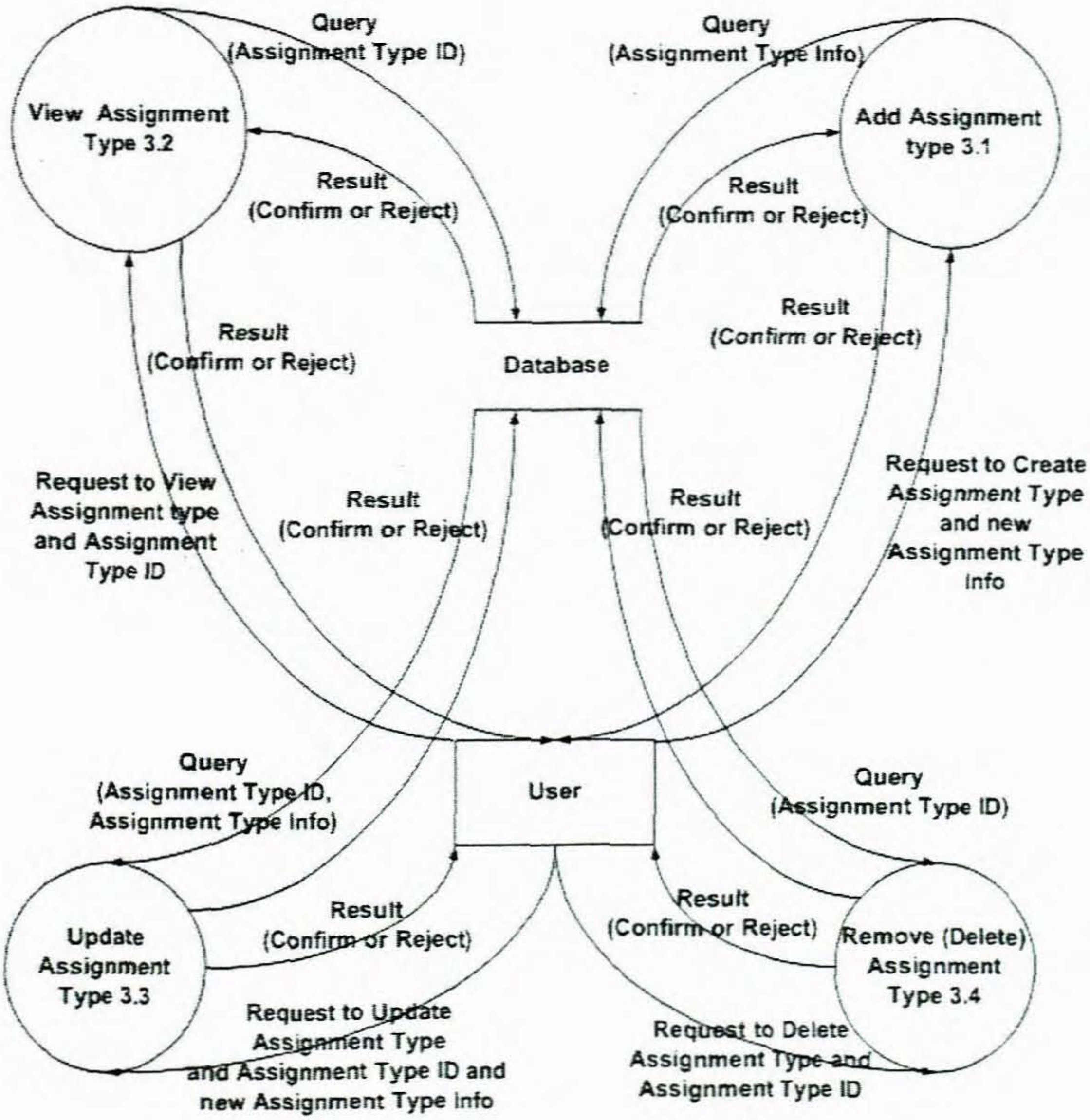
Outputs of delete the assignment

To delete a specific assignment, the SQL query would be sent to the database. When the database received this query, it depends on whether the operation was sent by a teacher or not. If it were a teacher's operation, the confirmation would be sent back, otherwise it would be not.

Process definition of delete the assignment

In order to delete an assignment entry in the database, Delete Specific assignment must first check the current user's permission level to verify they can legally delete the assignment. If the current user doesn't have permission to delete the assignment, they will get an appropriate error screen describing why they couldn't perform the delete. If the user has permission to perform the delete, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process 3.0—assignment types management



Query
(Assignment Type ID)
Query
(Assignment Type Info)
View Assignment Type 3.2
Add Assignment type 3.1
Result
(Confirm or Reject)
Result
(Confirm or Reject)
Result
(Confirm or Reject)
Result
(Confirm or Reject)
Database
Request to View Assignment type and Assignment Type ID
Result
(Confirm or Reject)
Result
(Confirm or Reject)
Request to Create Assignment Type and new Assignment Type Info
Query
(Assignment Type ID, Assignment Type Info)
User
Query
(Assignment Type ID)
Update Assignment Type 3.3
Result
(Confirm or Reject)
Result
(Confirm or Reject)
Remove (Delete) Assignment Type 3.4
Request to Update Assignment Type and Assignment Type ID and new Assignment Type Info
Request to Delete Assignment Type and Assignment Type ID

Figure 4.5. Assignments Type

Process description 3.1 add assignment type

When an administrator or user with the required permission level requests to create a new assignment type into the AMM, they must specify the new assignment type information. Create assignment type will then perform an SQL query to add this new assignment type to the database, and display the results of the action to the user.

Purpose of add assignment type

The AMM needs to store the different kinds of assignments that the AMM will track.

Inputs of add assignment type

New assignment type Info: assignment name, Associated assignments ID's, course, and due time. Results from the database: either a confirmation, or rejection of the operation.

Outputs of add assignment type

The outputs of add assignment type stands for the feedback of the system from the database. When a user does not commit all the categories of inputs of add assignment type, and the database would perform unsuccessfully. The two operations are shown as below:

I. SQL query sent to the database to add the new assignment type; and
II. results of the operation sent back to the user.

Process definition of add assignment type

In order to create a new assignment type entry into the database, create assignment type must first check the current user's permission level to verify that they can legally create the assignment type. If the current user doesn't have permission to create the assignment type, they will get an appropriate error screen describing why they couldn't create the assignment. If the user has permission to create the assignment type, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 3.2 view assignment type

When the current user requests to view the definition of an existing assignment type in the AMM, they must specify the assignment_type_ID. View assignment type will then perform an SQL query to return this assignment type's information from the database, and display the results of the action to the user.

Purpose of view assignment type

Users or Administrators of AMM need to get information on assignment types in the system.

Inputs of view assignment type

assignment_type_ID is necessary to obtain meaningful insight into view assignment type. Each assignment type has a unique assignment_type_id that does not mix up the results from the database. It is maintained that the assignment_type_id always be the primary key in the database.

Outputs of view assignment type

If the SQL query sends to the database for viewing an assignment type or assignment types, the function of outputs of view assignment type can accept parameters, process some logic, and then return the results of the operation that are sent back to the current user.

Process definition of view assignment type

In order to view an assignment type entry in the database, View assignment type must first check the current user's permission level to verify that they can legally view the assignment type information. If the current user doesn't have permission to view the assignment type information, they will get an appropriate error screen describing why they couldn't perform the read. If the user has permission to perform the read, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 3.3 update assignment type

When the current user requests to update the definition of an assignment type in the AMM, they must specify the assignment_type_ID, and the new data with which to modify the assignment type. Update assignment type will then perform an SQL query to Update this assignment type's information in the database, and display the results of the action to the user.

Purpose of update assignment type

Users or Administrators of AMM need to update the information on assignment types in the system in order to change the kind of assignments the system tracks.

Inputs of update assignment type

If an administrator or a teacher wants to update the information, assignment_type_id must be supplied, and commit new assignment type data. The primary key that assignment_type_id could not be changed. Results from the database feedback immediately.

Outputs of update assignment type

When an administrator or a teacher commits the SQL query to the database in updating an assignment type, the system process some logical estimation, and then returns the results of the operation to the current user.

Process definition of update assignment type

In order to update an assignment type entry in the database, update assignment type must first check the current user's permission level to verify that they can legally update the assignment type information. If the current user doesn't have permission to update the assignment type information, they will get an appropriate error screen describing why they couldn't perform the update. If the user has permission to perform the update, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 3.4 remove assignment type

When the current user requests to delete the definition of an assignment type in the AMM, they must specify the assignment_type_ID. Delete assignment type will then perform an SQL query to delete this assignment type from the database, and display the results of the action to the user.

Purpose of remove assignment type

Users or administrators of AMM need to delete assignment types from the system when they should no longer be tracked by AMM.

Inputs of remove assignment type

When teacher gives some suitable reasons why an assignment type would not be used, Only a teacher can delete the assignment type using assignment_type_id, and results would be indicated by the Database.

Outputs of remove assignment type

To delete an assignment type, the SQL query would be sent to the database. When the database received this query, it depends on whether the operation was sent by a teacher or not. If it were a teacher's operation, the confirmation would be sent back, otherwise it would be not.

Process definition of remove assignment type

In order to delete an assignment type entry in the database, Delete assignment type must first check the current user's permission level to verify that they can legally delete the assignment type. If the current user doesn't have permission to delete the assignment type, they will get an appropriate error screen describing why they couldn't perform the delete. If the user has permission to perform the delete, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation. If the user does not have permission, then an appropriate error message is sent to the user.

Process 4.0—Mark report management

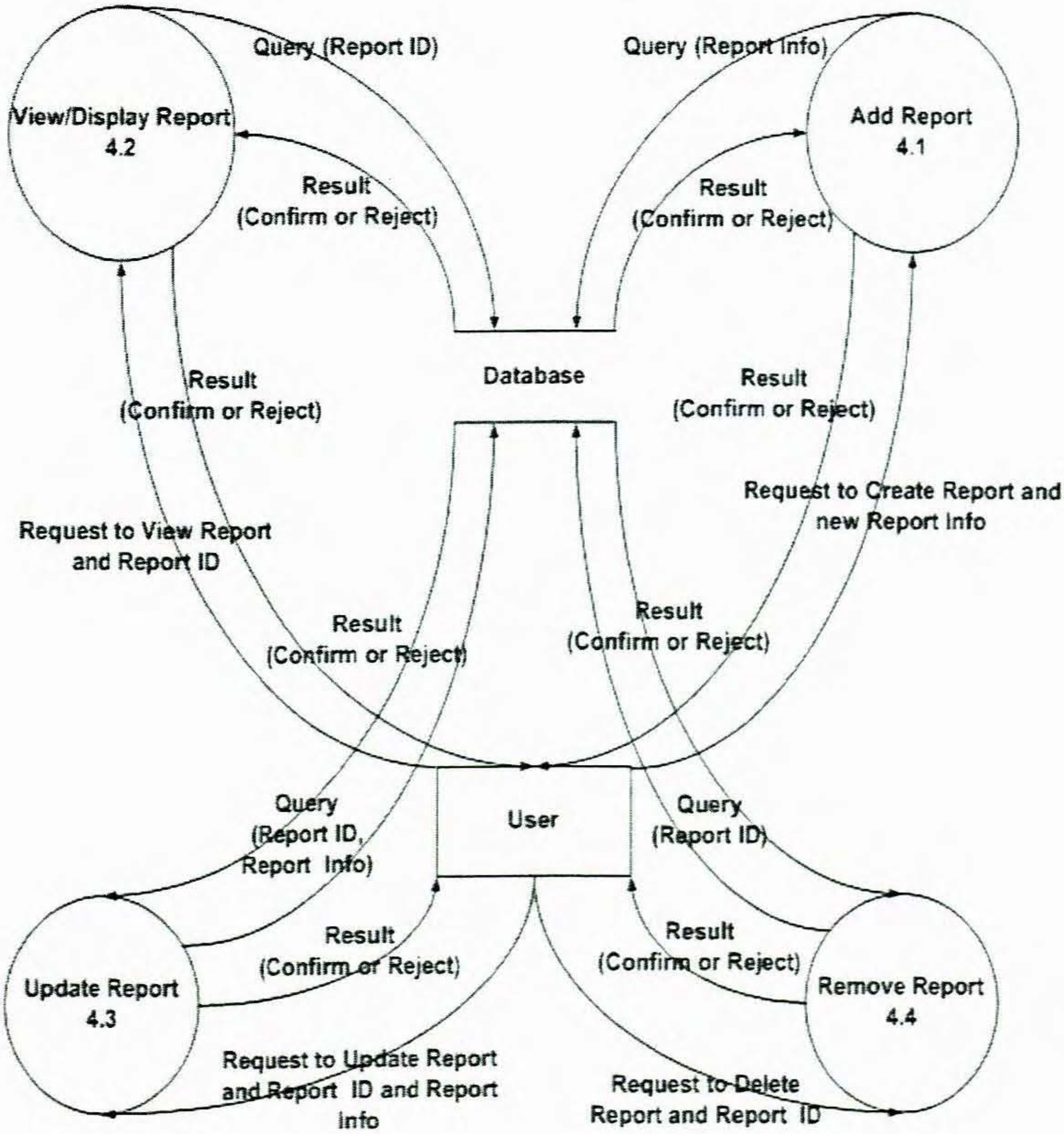


Figure 4.6. Mark report management

Process description 4.1 add report

When an Administrator or user with the required permission level requests to create a new report into the AMM, they must specify the new report information. Add report will then perform an SQL query to add this new report to the database, and display the results of the action to the user.

Purpose of add report

The AMM needs to store the different Mark-reports so that users of AMM can view them

Inputs of add report

The purpose of the function "inputs of add report" is the first step in the field of the mark report management. It could be categorized up two parts as below:

I. new report Information; and
II. Results from the database.

Outputs of add report

The outputs of add report stands for the feedback of the system from the database. When a user does not commit all the categories of inputs of add report, and the database would perform unsuccessfully. The two operations are shown as below:

I. SQL query sent to the database to add the new report; and
II. and results of the operation sent back to the user.

Process definition of add report

In order to create a new report in the database, Add report must first check the current user's permission level to verify that they can legally create the report. If the current user doesn't have permission to create the report, they will get an appropriate error screen describing why they couldn't create the report. If the user has permission to create the report, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 4.2 view/display report

When the current user requests to view the definition of an existing report in the AMM, they must specify the report_ID. Display report will then perform an SQL query to return this report's information from the database, and display the results of the action to the user.

Purpose of display report

Users or Administrators of AMM need to get general information in the form of reports from the system.

Inputs of display report

report_ID is necessary to obtain meaningful insight into inputs of display report. Each report has a unique report_id that does not mix up the results from the database. It is maintained that the report_id always be the primary key in the database.

Outputs of display report

If the SQL query sends to the database for viewing a report, the function of outputs of display report can accept parameters, process some logic, and then return the results of the operation that are sent back to the current user.

Process definition of display report

In order to view a report entry in the database, Display report must first check the current user's permission level to verify that they can legally view the report information. If the current user doesn't have permission to view the report information, they will get an appropriate error screen describing why they couldn't perform the read. If the user has permission to perform the read, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 4.3 update report

When the current user requests to update the definition of a report in the AMM, they must specify the report_ID, and the new data with which to modify the report. Update report will then perform an SQL query to Update this Assignment type's information in the database, and display the results of the action to the user.

Purpose of update report

Users or Administrators of AMM need to update the information on reports in the system in order to change the kind of reports users of the system can view.

Inputs of update report

If a user wants to update the information, report_id must be supplied, and commit new report data. The primary key that report_id could not be changed. Results from the database feedback immediately.

Outputs of update report

When a user commits the SQL query to the database in updating a report, the system process some logical estimation, and then returns the results of the operation to the current user.

Process definition of update report

In order to update a report entry in the database, Update report must first check the current user's permission level to verify that they can legally update the report information. If the current user doesn't have permission to update the report information, they will get an appropriate error screen describing why they couldn't perform the update. If the user has permission to perform the update, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process description 4.4 remove report

When the current user requests to delete the definition of a report in the AMM, they must specify the report_ID. Delete report will then perform an SQL query to delete this report from the database, and display the results of the action to the user.

Purpose of remove report

Users or Administrators of AMM need to delete reports from the system when they should no longer be viewed by users of AMM.

Inputs of remove report

When a user gives some suitable reasons why an assignment type would not be used, a user can delete the assignment type using report_id, and results would be indicated by the Database.

Outputs of remove report

To remove a report, the SQL query would be sent to the database. When the database received this query and processed it, then the results of the operation sent back to the current user

Process definition of remove report

In order to delete a report entry in the database, Delete report must first check the current user's permission level to verify that they can legally delete the report. If the current user doesn't have permission to delete the report, they will get an appropriate error screen describing why they couldn't perform the delete. If the user has permission to perform the delete, then the SQL query corresponding to the user's request is sent to the database, and the user is shown the results of the operation.

Process 5.0—Assignment reminder management

AMM
Diagram 5.0:Reminder manager

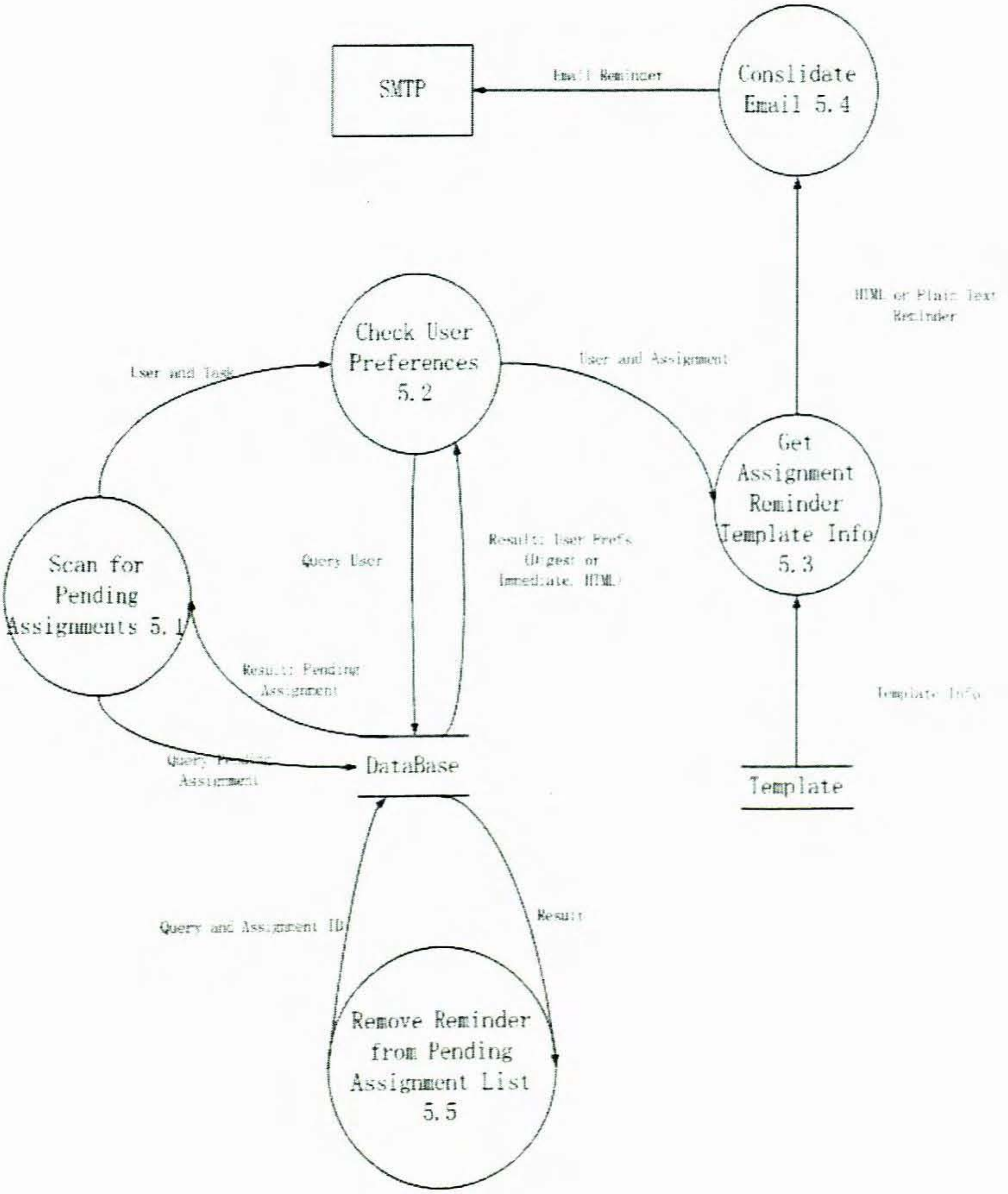

SMTP
Email Reminder
Conslidate Email 5.4
HTML or Plain Text Reminder
Check User Preferences 5.2
User and Task
User and Assignment
Get Assignment Reminder Template Info 5.3
Scan for Pending Assignments 5.1
Query User
Result: User Prefs (Digest or Immediate, HTML)
Result: Pending Assignment
Query Pending Assignment
DataBase
Template Info
Template
Query and Assignment ID
Result
Remove Reminder from Pending Assignment List 5.5

Figure 4.7. Reminder Manager

Process description 5.1 scan for pending assignments

Periodically, the LMS will request that AMM scan the database for pending assignments that are needed to be submitted or marked. Scan for Pending assignments will then perform an SQL query to find out which users currently need to be reminded about pending assignments. For each pending assignment found, Scan for Pending assignments will send information (of the user responsible for the assignment, and information showing that it needs to be submitted or marked.) to Check user Preferences.

Purpose of scan for pending assignments

The AMM needs to send email reminders to users based on pending assignments in AMM.

Inputs of Scan for pending assignments

The purpose of the function "inputs of Scan for Pending assignment" is the first step in the field of the reminder manager. It is pending assignments from database.

Outputs of Scan for pending assignments

The outputs of scan for pending assignment stand for the feedback of the system from the database. User Information sent to Check user Preferences.

Process definition of scan for pending assignments

In order to determine the assignments that still need to be completed, Scan for Pending assignments must first scan the database for currently pending assignments. For every pending assignment found in the database, Check user Preferences sends information regarding the user responsible for the assignment.

Process description 5.2 check user preferences

When AMM determines that an email reminder must be sent, Check user Preferences must query the database to determine whether the user prefers to receive immediate or consolidated emails.

Purpose of check user preferences

Some users of AMM prefer to receive emails as soon as AMM finds a pending assignment for that user, and other users prefer to receive digest emails of all their pending assignments.

Inputs of check user preferences

The function of inputs of check user preferences needs user Information. It can get user preference results from database.

Outputs of check user preferences

The SQL query sent to database to determine user's Preferences. assignment information sent to Get assignment Reminder Template Information.

Process Definition of Check user Preferences

In order to determine whether a user prefers to receive digest or immediate email reminders, AMM must send an SQL query to the database. Once the user's preferences have been determined, the applicable assignment information is sent to Get assignment Reminder Template Info.

Process description 5.3 get assignment reminder template Information

In order to format the email reminder correctly, Get assignment Reminder Template Info must read the appropriate email reminder template file.

Purpose of get assignment reminder template Information

Users of AMM must receive correctly formatted HTML or plain text email reminders from AMM.

Inputs of get assignment reminder template Info

Assignment Information and template information are necessary to obtain meaningful insight into inputs of get assignment reminder template information.

Outputs of get assignment reminder template Information

The outputs of get assignment reminder template information need HTML or Plain Text Reminder.

Process definition of get assignment reminder template Information

To correctly format an email reminder for users of AMM, Get assignment Reminder Template Info must get the necessary Template information from the template file, and format the email in either HTML or plain text.

Process description 5.4 consolidate emails

If a user prefers to receive email reminders in a consolidated form instead of immediately, consolidated emails must store all email reminders for the current session to be sent out in same format email.

Purpose of consolidate emails

Users of AMM might not want to be bombarded with an email for each pending assignment he/she must perform.

Inputs of consolidate emails

The function of inputs of consolidate Email required HTML or Plain Text Reminder.

Outputs of consolidate emails

The outputs of consolidate Email is backed up by Email Reminder to be sent by SMTP

Process definition of consolidate emails

In order to send a user a digest email instead of individual emails for each pending assignment, Consolidate Emails must compile email reminders sent to it into a singe large email reminder. This large digest email will then be sent to SMTP to send to the appropriate user.

Process description 5.5 remove reminder from pending assignment list

When a pending assignment has been completed or no longer needs to be completed, AMM must remove the assignment from the pending assignment list by sending the appropriate SQL query to the database.

Purpose of remove reminder from pending assignment list

When a pending assignment has been completed or no longer needs to be completed, AMM must remove the assignment from the pending assignment list.

Inputs of remove reminder from pending assignment list

Only an administrator can remove the inputs of remove reminder from pending assignment list, and the results would get from the Database

Outputs of remove reminder from pending assignment List

To remove a pending assignment, the SQL query would be sent to the database.

Process Definition of Remove Reminder from Pending assignment List

In order to delete a pending assignment from the pending assignments list, Remove Reminder from Pending assignment List must send an SQL query to the database to update the status of the Pending assignment.

Process description 5.1 scan for pending assignments

Periodically, CRON will request that AMM scan the Database for pending assignments. Scan for Pending assignments will then perform an SQL query to find out which users currently need to be reminded about pending assignments. For each pending assignment found, Scan for Pending assignments will send information (of the user responsible for the assignment) to Check user Preferences.

4.3.2 Performance Requirements

The number of workstations supported depends on the Apache Web server's capacity for multiple TCP/IP connections. Simultaneous database access depends on MySQL user handling capabilities, however, the software chapteration suggests that multiple users are supported in simultaneous queries.

4.3.3 Design Constraints

Report formats, data naming, coding and chapterations will follow IEEE standards. In order for the application to work, there are some hardware limitations:

- An existing Ethernet network environment.
- 200 MB of hard disk space to load the modules of the LMS.
- 64 MB of minimum RAM
- A Linux compatible computer

The numbers of users of AMM is limited to the configurations of Apache Web server and MySQL.

4.3.4 Quality Attributes

The software is developed using a consistent standard that all ITMPG members will adhere to. It's developer's intent to comply with the set standards and deliver a uniform application. This approach is taken to facilitate future expansions and easy maintainability.

4.3.4.1 Reliability

It is expected that AMM is a stable application capable of going for months without intervention due to external errors. Such errors may include unforeseen power outages and hardware failure, both of which may require an Administrator to restart services related to the AMM system.

4.3.4.2 Maintainability

The basic functionality of AMM is expanded through an administrative interface that can be used by the Sponsor to edit the structure of the database. An Administrator is able to connect directly through Linux server client to the MySQL database from which the AMM application creates and generates queries. AMM must be modularized to allow system expansion for the specific purposes of the Sponsor. Errors caught by AMM must be recorded in a log file.

4.3.4.3 Program Quality Attributes

The software is written using the standard conventions that include:

- Proper indenting of code under each function
- Functions and variables are named to reflect their use
- Modules are properly spaced for easy readability
- On-line comments show structure, flow, and functionality of the source code.
- The software application is designed to handle unexpected data and

appropriately log error messages

- Well-tested algorithms are used throughout the project life-cycle
- Limitations are imposed by language and hardware implementation

4.3.4.4 Security

The system can only be accessed though a user name and password. Based on the information provided the user is able to access the various function of the system. The system has safeguards against unauthorized access. For instance, at the Student level a user cannot access the function of Teacher level. Accordingly, usage of fictitious user names is blocked. There is no limitation on allowed access to a valid user. The system (including data) must be backed up often to protect against any kind of loss. Password access is a safeguard against unauthorized use of the software. The Apache server offers some virus protection and other commercial virus protection software are used to provide added security.

Security is an extremely important issue to the developers of AMM. It is the intent of ITMPG for the system to be implemented behind a firewall with all port 80 requests being sent to the Web server on which AMM is hosted. All data transfers are encrypted through Secure Socket Layer (SSL) encryption. Internal security to the AMM application has been considered throughout the design process and will be carried out through proper design measures. All potential data corruption related to viruses or intentional deletion of data are not addressed in the design or scope of the software application but can be avoided through proper Intranet and physical security measures. Should data be accidentally deleted, ITMPG's backup mechanisms can be used to restore the data to its previous state.

4.3.4.5 Transferability/Conversion

It is the intentions of the developer to have this application qualify for easy system transferability and platform conversion. However, the AMM implementation is designed to using Apache Web server, MySQL database, and STMP mail services. All modules external to the AMM application (including all the aforementioned) are required to be installed before the AMM application can be fully operational. Except for Apache Toolbox, no other scripts or batch processes are included in the delivery of the software application, which will ease in the installation of the required software components. The AMM system will be installed manually according to the directions given in the user Manual (or Maintenance Manual).

4.3.4.6 Operational quality attributes

The HTML code produced by the AMM system will imitate the look and feel of the ITMPT Intranet environment. All screen layouts are designed with a moderately Web-experienced end user in mind. Menus and buttons, which are currently facilitated in most corporate Web pages, may be used to allow end users to make defined queries. The user interface will present email embedded HTTP links, which will help the AMM system account for completed Assignments. users are required to understand the necessity of activating these links upon completion of a specified Assignment. Online help is available to users and Administrators of the AMM system at all times and will make reference to all major Assignments that the application is responsible for.

4.4 SUMMARY

This chapter provided a detailed description of the software requirements of the application. A general description of the application perspective, functions, characteristics, constraints and assumptions and dependencies was given.

Specific requirements, namely external interface requirements, functional requirements, performance requirements, design constraints and quality attributes were described in detail. All processes of the functional requirements were further elaborated upon.

CHAPTER 5 SOFTWARE ARTCHITECTURE DESIGN (SAD FOR AMM)

5.1 INTRODUCTION

This chapter on Software Architecture Design (SAD) is a formal architectural blueprint for the AMM. The functions that constitute the design of the AMM are discussed in detail within this chapter. The layout is structured to help the reader recognise the requirements imposed upon the system and how these requirements are being met.

5.1.1 Purpose

This chapter explores the fulfillment process of the engineering methodology design phase, and describes all the standards and the design of AMM to be used by the developers during in the implementation phase. The SAD is an important reference not only for developers, but also for the testing programmers and editors of documentation. With the help of the standards and specification described here, the developer can be more efficient and successful in implementing the application.

5.1.2 Scope

AMM is a Linux compatible application that manages assignments at websites. This application will help keep track of the assignments made for different users, as they will get reminders as long as the assignment is pending. It considerably reduces the amount of paperwork needed for such assignments. The goal of the developers is to implement this application by adhering to the specifications, execution environment, naming standards, coding standards, software architecture, software design, software modules (including both interface and Program Design Language), design analysis, and packaging of Version Viewer provided by SAD. In addition, all requirements documented in the SRS shall be implemented and addressed in the SAD.

5.1.3 Overview of contents of chapter

This subsection briefly describes each of the remaining sections in the chapter as well as the contents of each diagram. The rest of the chapter is divided into the steps as described. The next section architectural design specifies which processes will be assigned to which processors, where the data will be stored, and how much communication is required between processors. This section provides a road map of the software. It facilitates the understanding of the scope, concepts, logical organisation of the software and interaction between each of the software components. The following section, external interface design, provides insight into the look, feel and behavior of the portion of the system that is visible to the user. Section four, database design, provides the translation of the requirements model contained in the SRS into a relational database. The next section, internal components design, details the description for the design of the software. Included in this section are the software processes, software interactions and where applicable, interaction and logic diagrams depict the relationships modules and functions of the required by the project. This section provides sufficient detail about the software that would be difficult to uncover by reading the code. The sixth section component identifier specifies the naming rules to identify various components. The section performance analysis details any performance issues or constraints, and resolutions generated by the author. The section on feasibility and resources estimates summarizes computer resources required to build, operate and maintain the software.

5.2 ARCHITECTURAL DESIGN

The following details specify which processes will be assigned to which processors, where the data will be stored, and how much communication is required between processors. Details provide insight into scope, concepts, logical organisation of the software and interaction between each of the software components.

- **The scope of the software**

AMM is strictly responsible for providing a way to perform assignment management via the user's Web browser. Specifically, AMM provides the following functionality: user management, specific assignment management, mark assignments, information regarding management of assignments submitted and assignment feedback management the ability to send email reminders, and error reporting.

- **The concepts used to develop the software**

  I. AMM is written in PHP, an open-source, server-side scripting language.
  II. Apache is the Web server used to allow users access to AMM.
  III. MySQL is the open-source relational database used to store the majority of AMM data.

- **The logical software organization**

AMM is an open-source system for the management of assignments.

- **The pedagogy procedures contained in the software**

Upon installation, the Administrator(s) will create all processes unique to the implementation environment of LMS. AMM is designed to suite the general needs of many elicitation environments, and for this reason no specific pedagogy procedures have been hard coded.

Interactions between each of the software components

I. The "login" module is responsible for making sure that only authenticated users are authenticated to AMM pages.

II. The "logout" module is responsible for clearing all AMM cookies and variables, and logging the user out of the system.

III. The "Control.php" file gives AMM a uniform Look and Feel using Cascading Style Sheets and the "header" and "footer" modules give AMM a standard top and bottom for each page that is obey the same outlet style as another module of LMS.

IV. The "menu" module defines the structure of the navigation menu in AMM.

V. The "myprefs" function allows AMM users to change basic email and user preferences.

VI. The "error.inc" module supplies AMM implementers with the option for users error messages and corresponding codes.

VII. The "db.inc" module provides a standard library of database functions that work with just about any Relational Database on the market.

VIII. The modules:"report_php", "user_php", "assignment_php", "assignment_type_php", and all of their subordinate modules provide the main functionality of AMM.

5.2.1 Client/Server hardware tiers

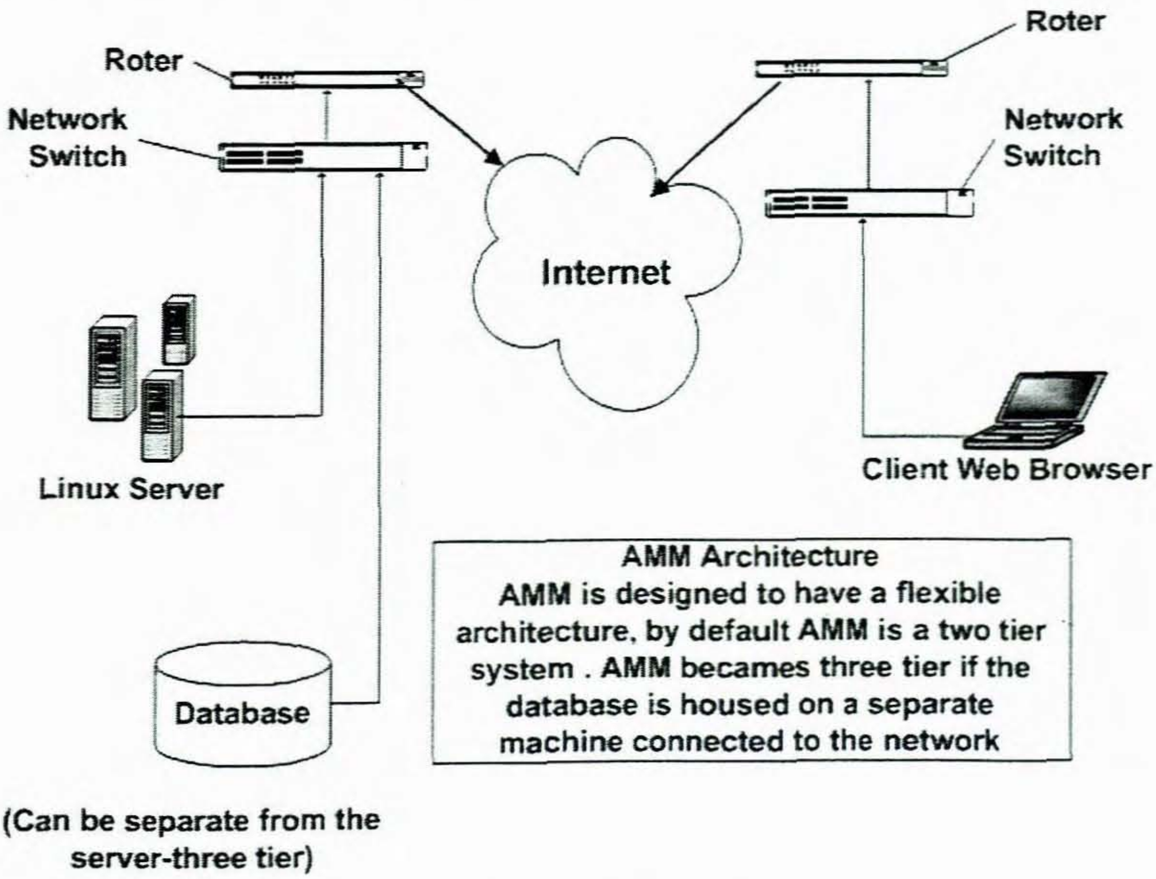

Roter
Network Switch
Roter
Network Switch
Internet
Linux Server
Client Web Browser
AMM Architecture
AMM is designed to have a flexible architecture, by default AMM is a two tier system . AMM becames three tier if the database is housed on a separate machine connected to the network
Database
(Can be separate from the server-three tier)

Figure 5.1. Client/Sever hardware tiers

Two-tier architecture

Under this implementation, the SQL server will be installed on the same machine as the Apache Web-server. All SQL queries will be sent to the local host and will therefore not require going over the network. Under this implementation, user post to the database should be faster, thereby decreasing Web server response time.

Three-tier architecture

Under this implementation, the database will be stored on a separate machine other than the Web server. All SQL queries will be directed to a machine name and logical port address. This schema may increase the latency of a CGI

POST.

5.2.2 Client/Server Software Layers

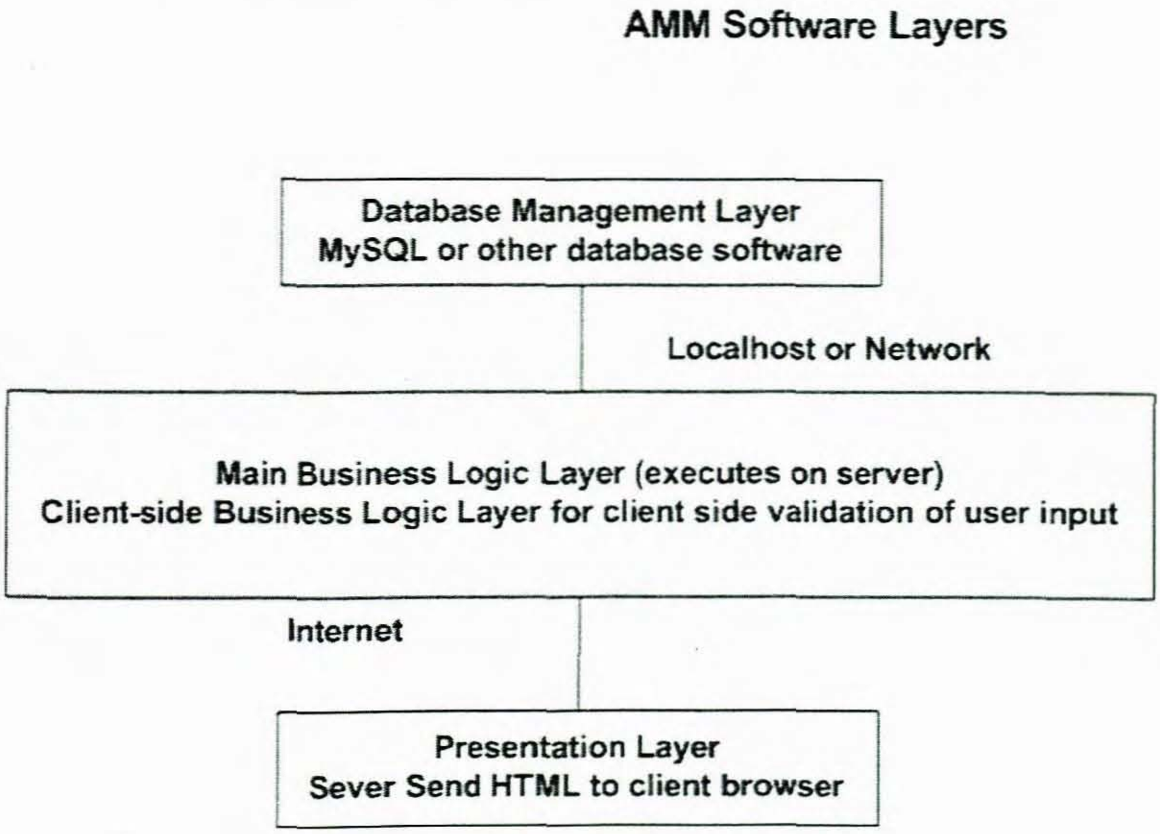

AMM Software Layers
Database Management Layer
MySQL or other database software
Localhost or Network
Main Business Logic Layer (executes on server)
Client-side Business Logic Layer for client side validation of user input
Internet
Presentation Layer
Sever Send HTML to client browser

Figure 5.2. Client / Server Software Layers

Presentation Layer

This layer resides at the "edge" of the software system. Its job is to capture external event requests and to perform some degree of editing of incoming data. It is also charged with presenting the event responses to the outside world (e.g. the screens and reports). This layer is usually allocated to the client machine or, in the case of a Web based application, to the Web server. It is important to note that what is included in this general description of the presentation layer is the graphical user interface (GUI).

"Business" logic layer

This layer contains the code, which executes and enforces the policy of the business. The business rules are contained in the "business" logic layer, hence the name. This software layer may be deployed on the client machines, on the server, or any machine on the network.

Data management layer

This layer provides access to the stored data. It manages concurrent requests to read and write to the database. In the case where data is distributed throughout the system, this layer would handle the synchronisation of these distributed data elements. In almost all cases, this software layer is defined by the data base management system used.

5.3 APPLICATION DESIGN PATTERN

The AMM application framework is based on a close approximation of the model, view, controller (MVC) design pattern, which separates an application's data model, user interface and control logic into three separate components. This enables changes to be made to one component without impacting on the others. Since the model is usually stable, separating it from the view and controller logic that a changed often during development leads to more robust applications that are easier to maintain. (see Figure 5.3 MVC Model).

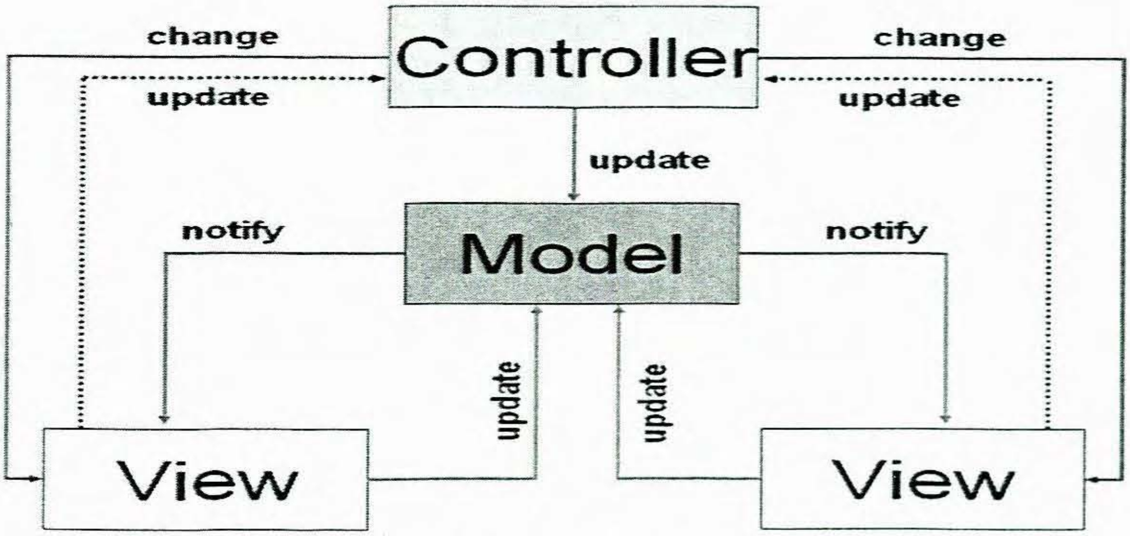


Figure 5.3. MVC Model

5.4 SYSTEM INTERFACE DESIGN

The requirements that have been imposed on AMM regarding the look and feel are clarified in the following subsections.

5.4.1 System overview

AMM is intended to automate some of the LMS aspects of assignment management. Functions such as assignment assign, assignment mark and assignment submit compose a bulk of what makes AMM a useful system. AMM should be applicable to any dynamic LMS environment where users are not involved in a routine structure, but where assignments are changing or where users are meeting with different clients.

5.4.2 Application overview

AMM is a dynamic Web-based application with a database backend. users can access AMM without having to install any additional software onto a standard operating system. This is advantageous because it can be accessed from any computer with access to the Internet and can be used and updated in a moments notice. AMM supports concurrent connections so that many users can be updating the database simultaneously and view the updates of each other almost instantly.

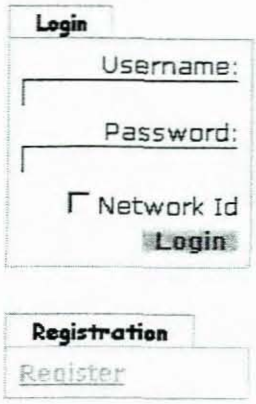
5.4.3 Window navigation diagram

AMM is based on an intuitive Graphical user Interface (GUI) and can easily be navigated through a navigation menu that directs users to the different options that are offered through AMM. Based upon user privilege level, certain functionalities may or may not be available to users. This prevents unauthorised users from making global changes to the structure of AMM. Only administrators of AMM have permission to make changes to the users that can access AMM as well as manipulate the assignment functions upon which the program is based.

5.4.4 Window layout

All windows are browser based and therefore are viewable in the same manner as normal Web pages. Dependant on certain computer resolution settings, the scrolling features that are offered in Web browsers or wheel mice may need to be incorporated in order to view the whole Web page at any time. The look and feel of AMM is meant to reflect the Intranet environment of the project sponsor ITMPTG. The following graphics illustrate the look and feel of AMM:

The user login screen:



Login

Username:

Password:

□ Network Id

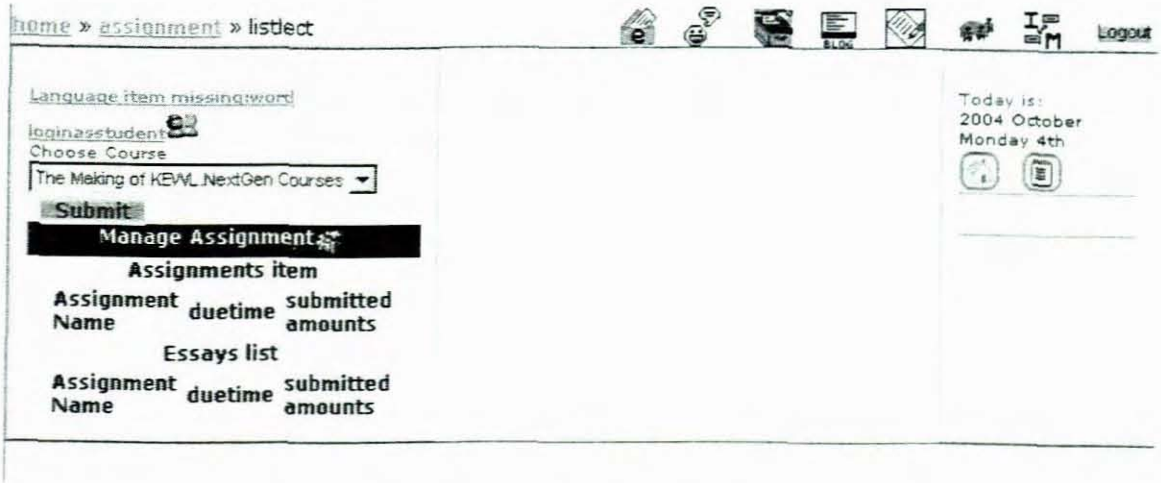
Login

Registration

Register

Figure 5.4. Login screen

The homepage screen



home » assignment » listlect

Logout

Language item missing:word

loginasstudent

Choose Course

The Making of KEWL.NextGen Courses

Submit

Manage Assignment

Assignments item

Assignment Name	duetime	submitted amounts

Essays list

Assignment Name	duetime	submitted amounts

Today is:
2004 October
Monday 4th

Figure 5.4.4-2. Home screen

5.4.5 Window specification

Windows must be organised in such a way as to allow the users of AMM to navigate quickly through the system so that updates or modifications can be *done without having to click more than five times from the login screen to get to* any option offered by AMM. Also screens should be organised in a way so as to be optimally viewed at the resolution of 800x600 pixels.

5.4.6 Window Description

The navigation menu will be reachable regardless of where a user may be in the AMM at any time. The menu will be on the left side of the screen and will run from the top down. Submenus will appear as links under the main navigation categories so that a user may jump to submenus without having to click an additional amount of times to get to a certain place in the program. A global footer will also appear at all times in the browser which will display any additional notes such as public licensing information.

5.4.7 Window Mini-specification

A link that says 'AMM' will be at the top left corner of the page, and will bring users to the LMS upon activation.

5.4.8 Field specification

All screens should be organised to allow a sufficient number of characters to be displayed in each input field. In pages that allow input of a large amount of data at one time, such as the "Create New Assignment" section, the format should be easily viewable at the screen resolution of 800x600 pixels.

5.5 DATABASE DESIGN

This section provides the translation of the informational model contained in the SRS into a relational database. The relational database will be comprised of tables. Each table will be companied of a series of columns which represent individual data elements. The data records in the table form the rows. Each table has a primary key. Tables are related to each other by embedding the primary key from one table into another as a foreign key to implement the relationship. Foreign keys enable the relational database management system to enforce referential integrity. Referential integrity insures that no row in a" parent" table can be deleted if it is still referenced in a row of a "child" table.

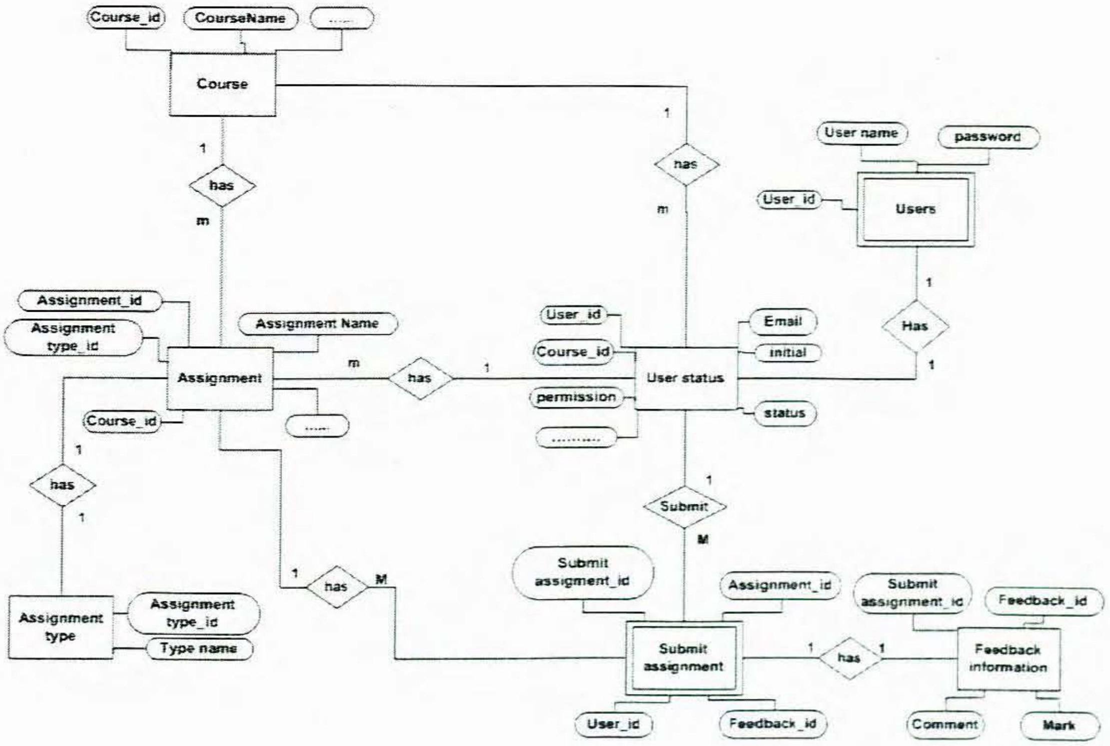


Figure 5.5. AMM entity relationship diagram

5.6 INTERNAL COMPONENT DESIGN

This section provides a detailed description for the design of the software. Included are software processes, software interactions and, where applicable, interaction and logic diagrams depicting the relationships of modules and functions as required by the project. The purpose of this section is to provide sufficient detail about the software that would be difficult to uncover by reading the code. This will be accomplished by identifying processes and illustrating processing dependencies, describing key algorithms and data structures within each process, and by illustrating the message interaction between processes.

5.7 PERFORMANCE ANALYSIS

The number of workstations supported depends on the capacity of the Apache Web server for multiple TCP/IP connections. Simultaneous database access depends on MySQL user handling capabilities, however the software documentation suggests that multiple users are supported in simultaneous queries.

Relative timing associated with I/O

Results should appear within five seconds when users are working within the Intranet.

Apache Web server documentation denotes an expected response timeframe of five seconds for any CGI POST made to the Web server through moderate local area network traffic.

Results should appear within 20 seconds when users are working over the Internet.

Apache Web server documentation also denotes an expected response timeframe of 10 seconds for any CGI POST made to the Web server through moderate Internet traffic.

Querying attempts are dependent upon hardware and network traffic.

Even under extreme network traffic, all requests made to the Apache Web server should be fulfilled within the required time specified by the project sponsor. AMM is built with the assumption that the hosting Web server has at least a 10Mbs Ethernet backbone to an Internet router.

Under normal circumstances the Apache Web server should be able to handle a realistic number of concurrent connections and still fall within this time restraint.

5.8 FEASIBILITY AND RESOURCE ESTIMATES

This section should contain a summary of the computer resources required to build, operate and maintain the software. (See SRS Section 4.3.3).

Client

IE 5+ or NS 4+

Cookies must be turned on, and JavaScript enabled

Connection to the internet/intranet (access to the Server described below)

Server

Hard Drive Space: 200 MB

Ram: 64 MB

Operating System: BSDI 4.2

Domain name to be able to send SMTP based emails from

TCP/IP Network Connection:

Supporting Software:

PHP v.4

MySQLv4

LDAP

SSL

Web Server (recommend Apache)

5.9 SUMMARY

This chapter provided a detailed description of the architectural design, system interface design, database design and internal component design of the application. Performance analysis and feasibility and resource estimates were also discussed.

CHAPTER 6 SOFTWARE TEST DESCRIPTION FOR ASSIGNMENT MANAGEMENT MODULE (STD FOR AMM)

6.1 INTRODUCTION

This chapter defines the Software Test Description (STD), which is developed to verify the integrity of the application to be created for a LMS. The developers at ITMPG should use these specifications to create the AMM software application, which serves as the ITMPT. Throughout this chapter, testing procedures are given to verify the stability of this software implementation. All details regarding testing descriptions and functionality of this project are found in this chapter.

6.1.1 Purpose

This chapter captures the testing procedures and descriptions that are required in AMM. The developer should use these specifications when making decisions regarding the stability of all software features so that project requirements are met. It also informs the LMS of what to expect from the application to be delivered.

6.1.2 Scope

Any required testing procedures and descriptions should be clearly noted throughout this chapter. Included in the STD is some discussion regarding verification details.

6.1.3. Overview of Contents in Chapter

This subsection briefly describes each of the remaining sections in the chapter as well as the contents of each diagram. The rest of the chapter is divided into the steps as described. The next section test Plan Description section gives an explanation of testing procedure and the constraints, assumptions, and dependencies that must be verified to assure software integrity. A test design specification section is composed of testing approach subsections including features and combinations to be tested and those which are not to be tested. Finally, a test specification subsection describes all the testing modules and criteria that will certify the operational features of AMM.

6.2 TEST PLAN DESCRIPTION

The developer understands the role of quality assurance to be a superset of testing. Through the formalities described below, the mission of the developers is to help minimise the risk of project failure.

6.2.1 Test items

The following files makeup the bulk of the code for AMM. Each page will be thoroughly tested to check for data integrity and any security vulnerabilities that may arise.

Here follows is a list of source code file names

indel.php controller.php register.php error.inc db_course.sql
db_assignment.sql db_assignment_submit.sql db_user.sql db_course.php
db_assignment.php db_assignmentsub.php db_user.php
assignmentAdd_tpl.php assignmentEdit_tpl.php assignmentUpdate_tpl.php
assignmentMark_tpl.php deleteConfirm_tpl.php assignmentView_tpl.php
assignmentEmail.php assignAdmin_tpl.php assignLecturer_tpl.php
assignStudent_tpl.php

6.2.2 Management Approach

The primary goal of ITMPTG during the testing process is to monitor application quality. The management of quality is broken into individual sets that will be followed during the testing procedure. These sets are outlined as indicated below.

6.2.2.1 Quality planning

At this stage, the developer will have explored and evolved a clear idea of the requirements for the delivered application. The ITMPT will also play a vital part in ensuring that all industry standard techniques have been utilised to create a well-developed system. Frequent communication between group members and the users will serve as a continual review of the system requirements.

6.2.2.2 Quality assessment

During the development of the functionality and usability of the AMM, the developer will compare the application with the quality standard of open-source code available on the Internet. During the application lifecycle, these open-source applications will be insightful towards the functionality and usability development of AMM. The developer will adopt any formalities used throughout the open source community, as an effort to enhance the use and configurability of the system.

6.2.2.3 Quality reporting

The developer will be informed of any system errors through a formal bug tracking procedure. When a bug is discovered, the developer manually updates this formal tracking mechanism. Before updates to the current version of code can be made, the developer will look to see if any pending bugs have been reported. Once the bug has been resolved, the reference to it will be removed from the tracking mechanism.

6.2.3 Test deliverables

All codes developed by the developer for the AMM system are tested independently and collectively. System functionality is tested for all potential stresses that have been denoted in the SRS. For instance, the system has been specified for use on a platform of 100 users or less. These realistic stresses, along with hypothetical scenarios, such as the intuitiveness of the interface, have been considered during the testing procedure. Each module of code was checked for internal integrity, to ensure that data sent and received would perform as expected. These individual modules were then collectively tested between each other to verify that the flow of data was also performing as expected.

6.2.4. Testing Tasks

Due to the non-deterministic paths that can be taken throughout the system, a standard has been established to ensure that the important facets of AMM have been considered during every test. These testing tasks include: authenticating on every instance of a test, fulfilling that particular test requirement, and then logging out of the system. This ensures that all full ranges of functionality had been considered on every potential branch of the system. It also ensures that user permissions are operative at every step of the testing process.

6.2.5 Responsibilities

It is the developer's responsibility to correct any debilitations that have been discovered through the testing process, and to align the functionality of the system to the specifications of the SRS. Any non-functional appendages to the system which are external to the scope of the SRS need not be fully working at the time of delivery.

6.2.6 Schedule

The developer has coordinated the testing schedule to coincide with other activities in the project. These external activities include the user manual and any other required document that may be necessary in completing the project. Planning and status meetings will also be held concurrent to the testing of all code.

Due to the importance of the testing process, the developer will begin testing codes the moment that an application is available. A mechanism of logging bug reports has been implemented and will help manage and classify each bug according to its potential risk to the delivered application. The developer will continue enhancing the code up to the delivery date. The developer will insist on careful change control to assure that no reckless modifications are made to the code base during the testing cycle.

6.2.7 Risks and contingencies

A careful process of version control has been established by existing LMS to protect validated code from non-tested enhancements of the original code. Every version of code contains all modules, and therefore are not merely increment updates of certain files. If an enhancement or change to a version proves to be overly buggy, ITMPTG will simply rework the changes from a previously posted version. The contingencies that should be followed include team wide notification that work on the code will occur, and afterwards team wide notification that a new version of code has been posted to the CVS site. It is through this process that team members prevent creating concurrent updates to the code, which may be incompatible to each other.

6.3 TEST DESIGN SPECIFICATION

The developer has determined the following guidelines to be strictly adhered during the testing process. By aligning the software process to these specifications, full validation of sponsor requirements can be expected.

6.3.1 Testing approach

The testing approach is broken down into sets, which will be followed before and during the testing phase. These sets are denoted by the following guidelines:

- **Perform an appropriate test process**

Adherence to the step-wise process defined in section 4.3.2 fulfils the guideline of an appropriate test process. Any bugs or deviations from the expected performance will be formally reported in a bug tracking system.

- **Test planning**

The developer will use two forms of software testing. These forms include white-box and black-box testing techniques. The white-box method will be used to validate the internal functions inside each page of code. The black-box method will be used to evaluate how the pages of code interact with each other.

- **Application analysis**

Throughout the lifecycle process, the developer has constructed a clear definition of the functionality and borders of the system. Therefore the developer understands how AMM is expected to work and will recognise a lack of functionality through the testing process.

- **Application coverage**

All aspects of AMM will undergo thorough testing, ranging from core application functions to software security. A priority level will be in effect throughout the testing process. This priority scheme will be based on overall risk to project delivery, and hence any feature impeding this delivery will receive the highest attention.

- **Test design**

The testing plan used by the developer will aim first at the individual modules of the AMM system. Once these individual modules have been validated, the system will be tested as a component of parts. The developer will concentrate on major functionalities until they have been approved; afterwards items of lesser importance will undergo robust testing. The developer realizes that the delivery date has to be met, and therefore should weigh the attention level each component receives against the remaining time until project delivery. This balance between time incurred in testing each module or set of modules versus other aspects of the system, is directly proportional to how the overall impact of a fault in these modules would affect the entire project.

- **Test execution**

A test process is appropriate if it provides an acceptable level of confidence in the quality assessment. The less sure a developer need to be about his/her quality assessment, the less extensive the test process should be. "Don't insist on extensive testing for low risk components. But, if you need to ship a high quality application, than you need to work hard at the tasks above" (Raymond,2003:73).

6.3.2 Environmental needs

ITMPTG required the use of a duplicate environment for the project in order to validate the testing process fully. A computer running Apache, PHP, MySQL, LDAP and SSL was facilitated in the testing process to ensure that all aspects of the code performed as expected. To test the security parameters of AMM, ITMPTG made use of a network protocol analyzer to validate that all traffic sent to and from the system was done using encrypted data packets.

6.3.3 Pass/Fail criteria

In order for code to pass fully, it had to fulfil the intent of the specification in the SRS. If the requirement specified had minor technical details, which could be interpreted in different ways, a majority vote determined the methods and thresholds to be used as a validating measurement. If the software fell short of this degree of quality and alignment to the original intent, the code was considered insufficient for validation, and underwent revisions until the measurement proved it to be a baseline work.

6.3.4 Suspension/Resumption criteria

Testing on an area of the system will be suspended if major revisions have to be implemented due to a lack of requirement adherence. Once this module, or set of modules, has been revised to meet the original software requirement specifications, testing will resume. In the meantime, the developer will delegate team members to continue testing other aspects of the system while the code revision takes place.

6.4 TEST SPECIFICATION

ITMPT has established the following specifications to formalise the testing process of AMM. Adherence to these guidelines will ensure that team members focus adequate attention on the aspects of the code that are the highest priority, and then move to code that creates a smaller chance of project failure. Team members will communicate formally with each other to report findings, and collaborate on a streamlined method of correcting software errors.

6.4.1 Test procedures

The developer will follow the testing process outlined, and will record the actual output of the system in the column specified. If the output is not aligned with the expectation, the code will undergo a revision until it receives a full validation by the entire team.

6.4.2 Test procedure conventions

The formal testing breakdown used by the developer is represented by a test matrix spreadsheet. Columns are divided by Requirements, the Execution, which *must occur to prepare for module(s) being tested, Input that must be* given through the testing process, the expected output which should occur, and the actual output that the system produced. Rows in the test matrix spreadsheet represent each point specified in the Requirements Traceability Matrix. *These requirements compose the total requirements imposed upon the* system for technical stability and by request of the project users.

6.5 SUMMARY

In this chapter, Software Test Description was discussed. Specifically, the requirements of the test plan description, the test design specifications and test specifications, including testing procedures, schedules and protocols, were described. This discussion is important, as it contributes to the verification of the integrity of the application.

CHAPTER 7 CASE STUDY RESEARCH PROJECT IMPLEMENTATION IN KEWL.NEXTGEN

7.1 INTRODUCTION

This chapter introduces the implementation process of the AMM for KEWL.NextGen. The AMM will be constructed to make it easier to create, mark and manage assignments as well as record individual student performance.

7.1.1 Purpose

The AMM is the core module of LMS. This chapter clearly captures the internal development processes of AMM to accomplish project goals. The reader of this document will know the developer's design ideas and architecture of AMM, and how the developer intends to control and manage the development of the proposed application to be delivered to KEWL.NextGen, as well as what the application is supposed to do.

7.1.2 Scope

Details concerning the project management process, development standards, and procedures all obey the framework of KEWL.NextGen. The whitepaper can be downloaded from Kngforce.uwc.ac.za.

7.2 ORGANISATION OF DEVELOPMENT AND METHODOLOGY

7.2.1 Management organization

The development organisation of AMM as a module of KEWL.NextGen takes the following forms: The projects use version control systems (CVS) to adhere to the rule "release early—release often". Although many modules are run by a small group of programmers or individual developers, the software is released as often as possible to encourage contributions from other programmers. Design decisions, standards and technical skills are often discussed intensely on mailing lists and published on the website, and also there is also central code repository (kngforge.uwc.ac.za), to which contributions are added. The version control systems are fundamental to the project management of the project. These fundamental methods are very useful regarding how the AMM application is managed.

7.2.2 Software development methodology

"Whatever the kind of software, software development methodology is essential to the software engineering process" (Stephen,1999). During the development process of the Assignment Management Module, the developers used KEWL version 1.2 as a reference, so it was decided to adopt the prototyping paradigm (existing program form) as the development methodology.

7.3 REQUIREMENT

7.3.1 Existing application KEWL version 1.2.

In KEWL version 1.2, assignment functionalities plug into the assessment module, and the existing module that implement a part of the required functionalities. It is convenient for lecturers and students to operate the system. The developers splitted them from the assessment module. Assignment functionalities have features that should be improved in a later development effort.

7.3.2 Requirements for AMM in KEWL.NextGen

The AMM requirements can be divided into two aspects: functional and non-functional requirements. Both functional and non-functional lists are listed in Table 7.1, list of requirements to which can be referred to during application development and functional testing. The initial requirements will be covered in the application implementation, and advanced requirements may not be covered.

In table, '**I**' represents initial requirement, '**A**' stands for the advanced requirement

Functional requirement	
Login function	
Different users operation interface (administrators, lecturers and students)	I
Lecturer-users and student-users have access to their own preferences in the course.	I
Creating assignment function	
Assignment types include online activity model and upload-file model.	I
The lecturers can choose to allow resubmission of assignments before these are marked.	I
Assignments can be specified with a due date and a maximum mark.	I
Lecturers can login like students to preview the assignment.	I
Distributing assignment function	
users can distribute their finished assignments by URL. The assignments are date-stamped.	I
Students can upload their assignments (any file format) to the server. These assignments are also date-stamped.	I
Late assignments are not allowed, and are shown clearly to the student.	I
Provide function to prevent multiple submissions.	**A**
Assignment feedback function	
Lecturer feedback information is appended to the assignment page for each student.	I
For each particular assignment, the whole course can be assessed (mark and comment) on one page in one form.	I
Offer electrical-pen mark-up tools to simulate normal pen. Lecturers are given help online to correct students' uploaded essays.	**A**
Use statistical presentation to display the mark of the assignment	**A**

Non-functional requirement	
All non-functional requirements obey KEWL.NextGen *framework*	

Table 7.1. Requirements list

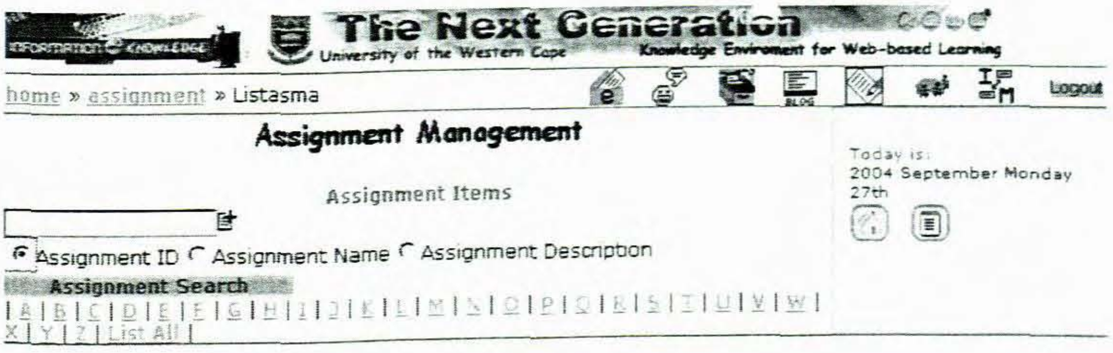
7.4 APPLICATION ARCHITECTURE

The design is classified as follows: interface design, process view and data architecture. The architecture is represented in Rational Unified process UML specification.

7.4.1 The user interface architecture

The three main types of users for the AMM implementation include administrators, lecturers and students. Administrators have their own style interface. *They also have access to all assignments, and operate all sections* of the databases (Read, Create, Edit and Delete).

Administrators have authorisation functionalities to control users and access to AMM.

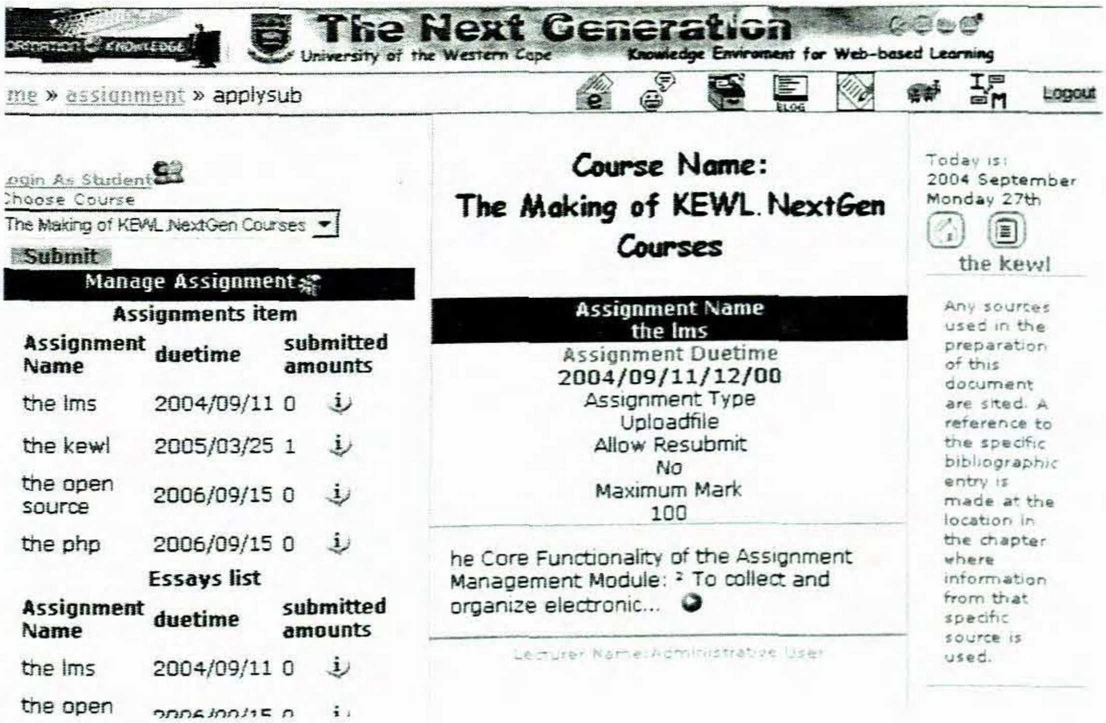

INFORMATION KNOWLEDGE
The Next Generation
University of the Western Cape
Knowledge Enviroment for Web-based Learning
home » assignment » Listasma
e
BLOG
Logout
Assignment Management
Assignment Items
Today is:
2004 September Monday 27th
Assignment Search
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | List All |

● Assignment ID ○ Assignment Name ○ Assignment Description

Figure 7.1. Administrator-user main interface

In general, lecturers should have a similar interface to students, but get a few extra abilities. Lecturers can access all assignments in the course, but only obtain the permission to operate assignments that they set.

Students have access to their own preferences in the course, and can view only the assignment feedback information that the lecturer gives them permission to.



The Next Generation
University of the Western Cape · Knowledge Enviroment for Web-based Learning

me » assignment » applysub · Logout

Login As Student
Choose Course
The Making of KEWL.NextGen Courses
Submit

Manage Assignment

Assignments item

Assignment Name	duetime	submitted amounts
the lms	2004/09/11	0
the kewl	2005/03/25	1
the open source	2006/09/15	0
the php	2006/09/15	0

Essays list

Assignment Name	duetime	submitted amounts
the lms	2004/09/11	0
the open	2006/09/15	0

Course Name:
The Making of KEWL.NextGen Courses

Assignment Name
the lms
Assignment Duetime
2004/09/11/12/00
Assignment Type
Uploadfile
Allow Resubmit
No
Maximum Mark
100

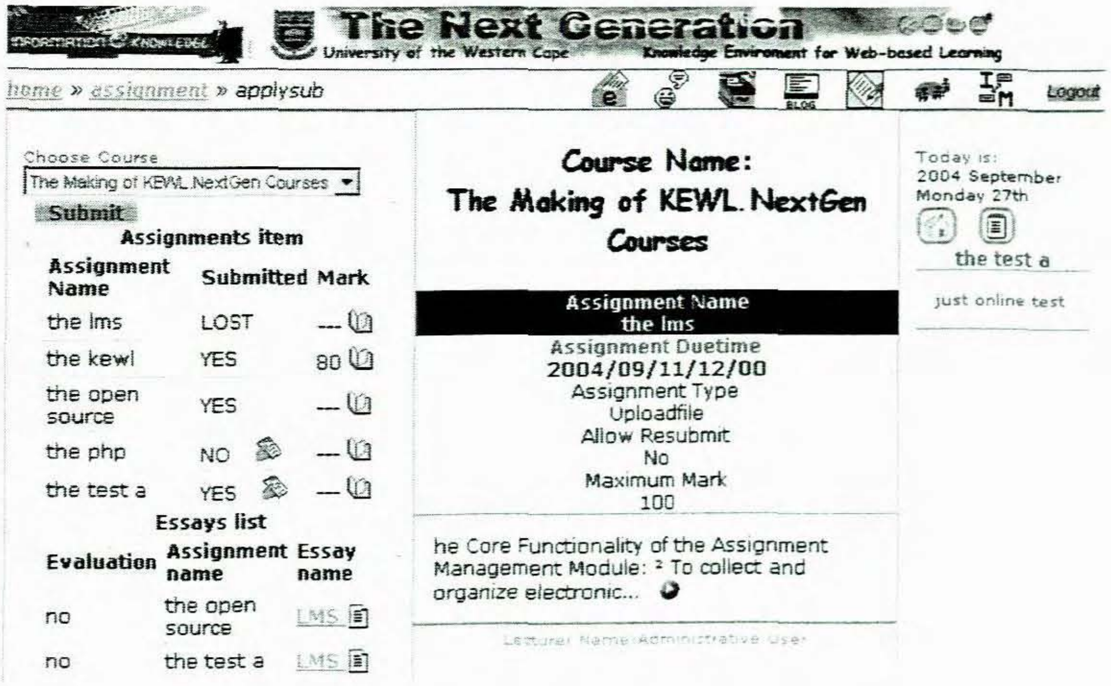
he Core Functionality of the Assignment Management Module: ² To collect and organize electronic...

Lecturer Name:Administrative User

Today is:
2004 September
Monday 27th
the kewl

Any sources used in the preparation of this document are sited. A reference to the specific bibliographic entry is made at the location in the chapter where information from that specific source is used.

Figure 7.2. Lecturer-user main interface


The Next Generation
University of the Western Cape
Knowledge Environment for Web-based Learning
home » assignment » applysub
BLOG
Logout

Choose Course
The Making of KEWL.NextGen Courses
Submit

Assignments item

Assignment Name	Submitted	Mark
the lms	LOST	---
the kewl	YES	80
the open source	YES	---
the php	NO	---
the test a	YES	---

Essays list

Evaluation	Assignment name	Essay name
no	the open source	LMS
no	the test a	LMS

Course Name:
The Making of KEWL.NextGen Courses

Assignment Name
the lms
Assignment Duetime
2004/09/11/12/00
Assignment Type
Uploadfile
Allow Resubmit
No
Maximum Mark
100

he Core Functionality of the Assignment Management Module: ² To collect and organize electronic...

Lecturer Name:Administrative User

Today is:
2004 September
Monday 27th
the test a
just online test

Figure7. 3. Student-user main interface

The output screen will have the following general layout:

Left frame

- Course login link to choose course
- Assignments general information
- Upload essays list
- Assignments operation menu

Middle frame

- Individual assignment detail information
- Assignment operation link

Right frame

- KEWL.NextGen homepage link
- Assignment main-page link

- Assignment title and description

7.4.2 Process view

This section explains the process of assignment management. Access to the assignment page is defined by the user's level. Administrator-users can manage all assignments, the lecturer-users can manage their own assignments, and students can view their assignments and perform them. This process needs a user registered in the course.

To manage assignments

a. Enter the administration page or click on the "manage assignments" icon

To create new assignment

i . Click on "add" icon.

ii . Choose the course.

iii. Enter name and description.

iv. Choose the assignment type.

v . Choose whether assignment can be resubmitted.

vi. Enter the maximum mark.

vii. Enter the due time.

viii. Click on "save" button.

To edit an existing assignment

i . Click on the “edit” icon corresponding to the assignment to edit.

ii . Do the necessary editing and press “save” button.

To delete an assignment

i . Click on the “delete” icon corresponding to the assignment to delete.

ii . Click on “confirm” to finally delete it.

To preview an assignment

i . Press “login as student” icon.

To mark students’ assignments (online model type)

a. Click on the mark icon corresponding to the assignment in the left area of the Webpage, or click on the mark icon corresponding to the student’s name in the middle area of the Webpage.

i . Enter the mark and comment.

ii . Press “save” button.

To mark students’ assignments (upload-essay type)

a. Click on the “info” icon corresponding to the assignment on the left of the Webpage, or click on the mark icon corresponding to the student’s name in the middle area of the Webpage.

i . Click on the “essay” icon to download the essay.

ii . Enter the mark and comment.

iii. Press “save” button.

To view assignments

a. Users can view assignments' general information in the left area of Webpage or click on corresponding the assignment to view detailed information.

To submit assignments

a. Click on the "submit" icon corresponding to the assignment on the left area of the Webpage or in the middle area of the Webpage.

i . Enter the online assignments (online model type) or upload an essay (upload-essay type).

ii . Press submit button.

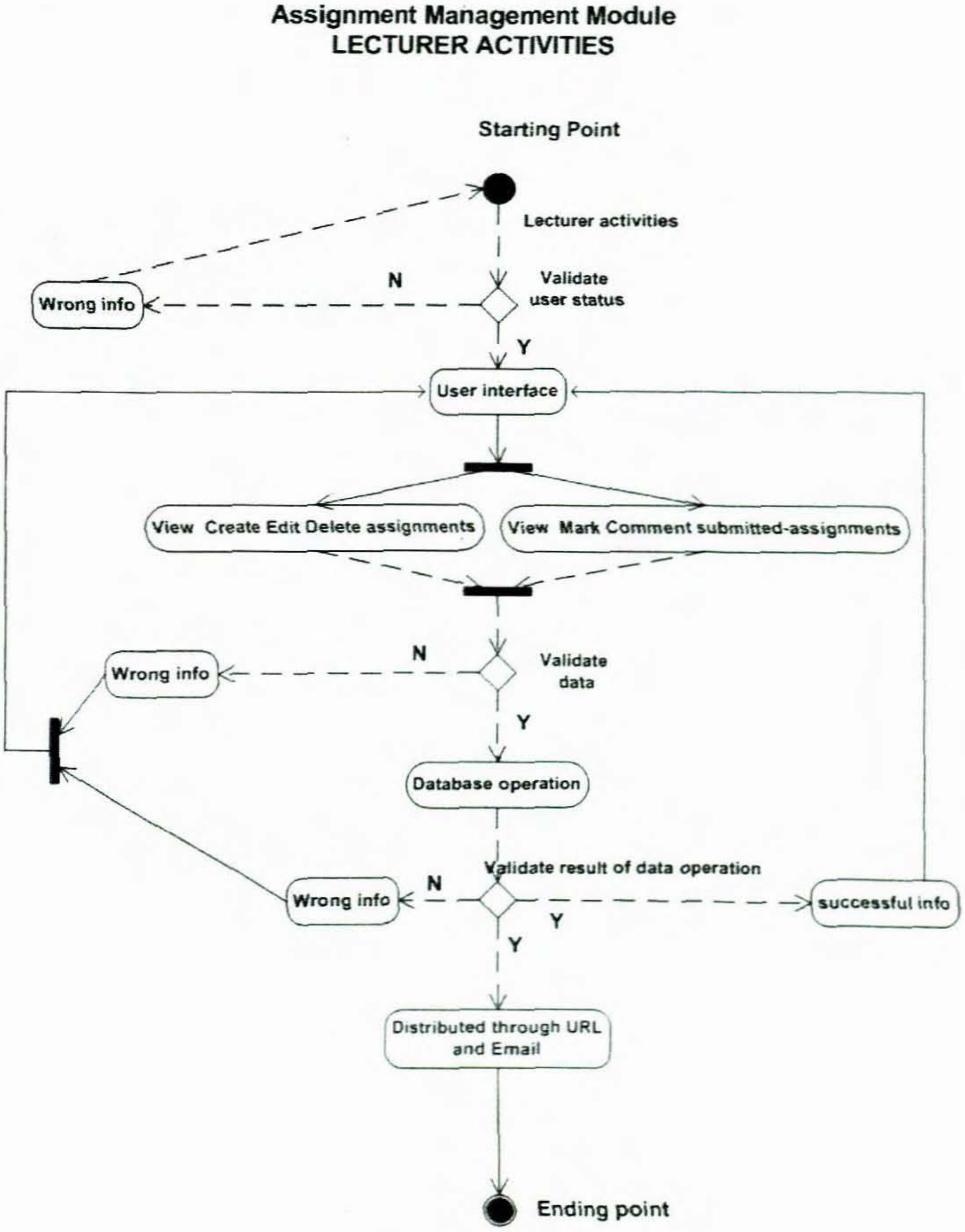
To view assignments feedback information

a. Student-users can view assignment marks in the left area of the Webpage.

b. Click on "submitted-view" icon of the corresponding assignment to view comment information in the left area of Webpage or click on corresponding assignment in the middle area of Webpage.

7.4.3 The user activities diagram

7.4.3.1 Lecturer activities



Assignment Management Module
LECTURER ACTIVITIES

Figure 7.4. Lecturer activities

7.4.3.2 Student activities

Assignment Management Module
STUDENT ACTIVITIES

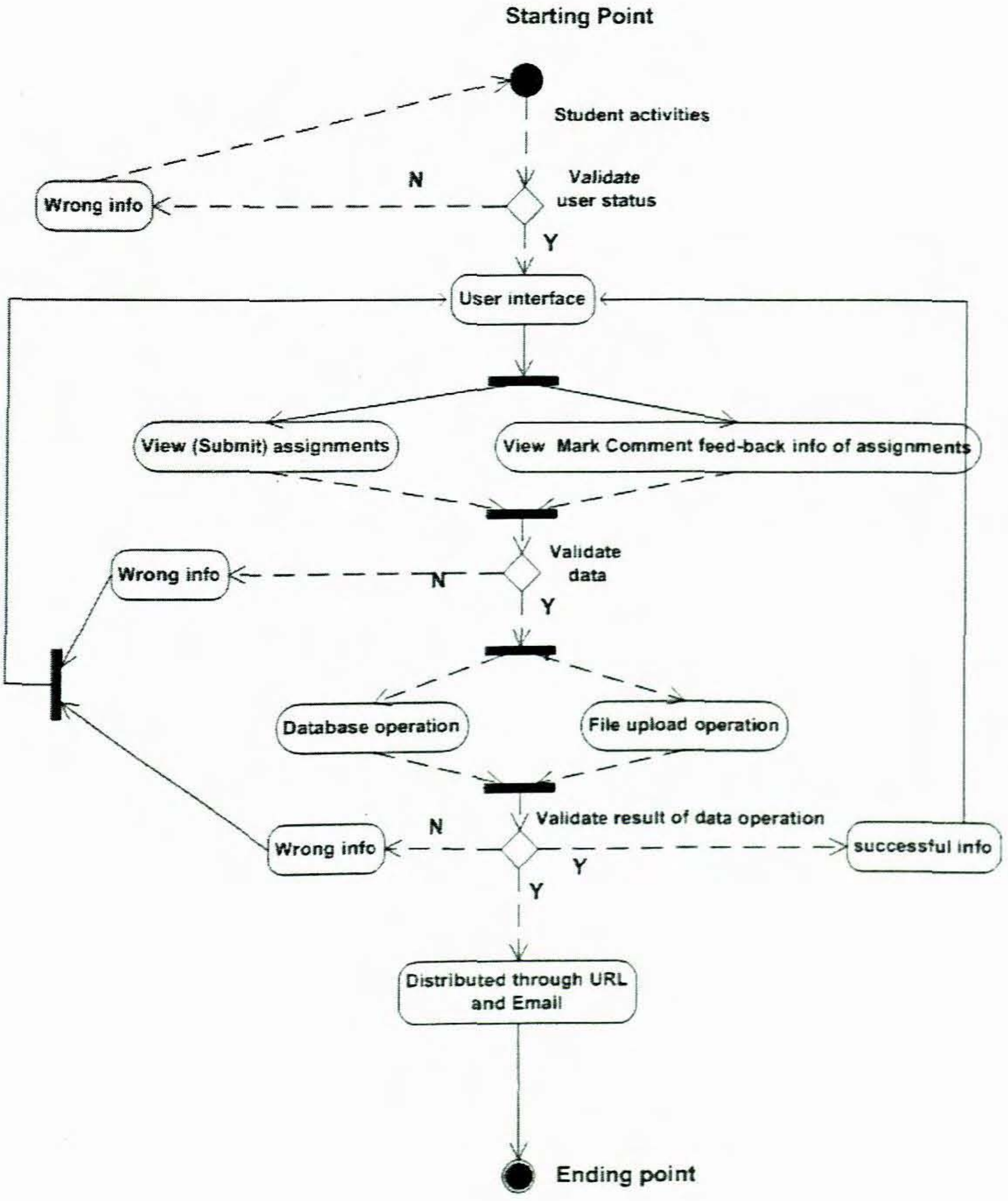


Figure 7.5. Student activities

7.4.4 Data architecture

There are two database tables in AMM, and Diagram 4.3 illustrates the table relationship.

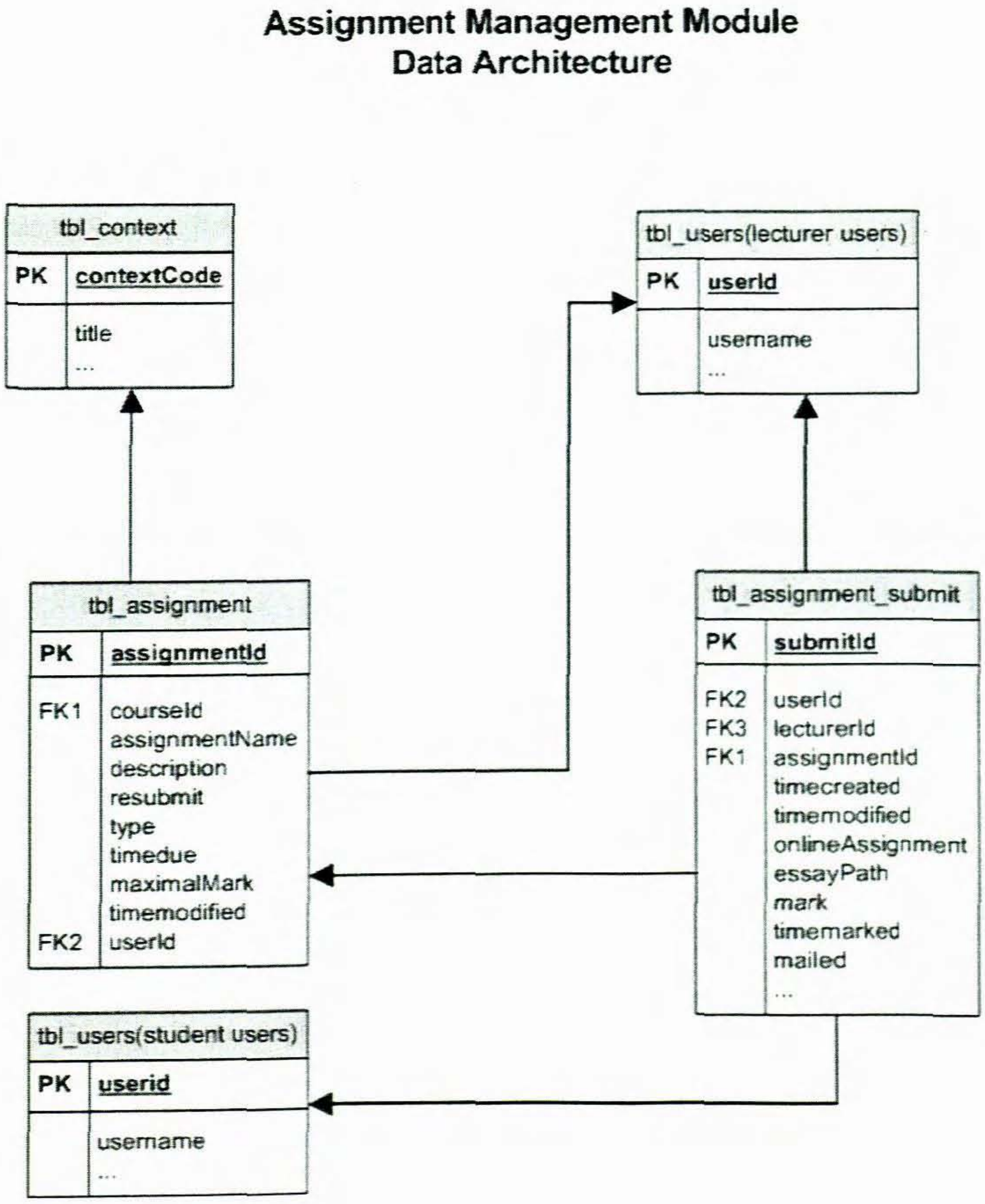


Figure 7.6. Data architecture

7.5 IMPLEMENTATION

The framework of KEWL.NextGen applied to the MVC (model-view-controller) design pattern. The MVC paradigm is a way of breaking an application into three parts: the model, the view, and the controller. MVC was originally developed to map the traditional input, processing, output roles into the GUI realm:

Input --> Processing --> Output

Controller --> Model --> View.

7.5.1 Source-code files architecture

The following files in Table 7.2, source code files list make up the bulk of the code for AMM.

Source code files name	**Repository**	**Function description**
controller.php	/assignment..	Application logic file
register.php	/assignment..	Application register file
tbl_assignment.sql	/assignment..	Database table register file
tbl_assignment_submit.sql	/assignment..	Database table register file
dbarcourses_class_inc.php	/assignment/classes	Data operation class file
dbassignment_class_inc.php	/assignment/classes	Data operation class file
dbassignmentsub_class_inc.php	/assignment/classes	Data operation class file

tabledisplay_class_inc.php	/assignment/classes	Form table display control class file
asadd_tpl.php	/assignment/ templates/content	Assignment creation page
asaddt_tpl.php	/assignment/ templates/content	Assignment creation page for specific course
asedit_tpl.php	/assignment/ templates/content	Assignment edition page
assubinsert_tpl.php	/assignment/ templates/content	Assignment submitted page
assubmark_tpl.php	/assignment/ templates/content	Assignment mark page
assubupdate_tpl.php	/assignment/ templates/content	Assignment resubmitted page
assubview_tpl.php	/assignment/ templates/content	Display specific assignment info
confirmdelete_tpl.php	/assignment/ templates/content	Data deletion confirm page
coursech_tpl.php	/assignment/ templates/content	Temporary page for choosing course
error_tpl.php	/assignment/ templates/content	Operating data failed info
list_as_tpl.php	/assignment/ templates/content	Assignments list page
list_asma_tpl.php	/assignment/ templates/content	Temporary page for selecting user status
mainadmin_tpl.php	/assignment/ templates/content	Main operation interface of super

		users
mainlec_tpl.php	/assignment/ templates/content	Main operation interface of lecturer users
mainstu_tpl.php	/assignment/ templates/content	Main operation interface of student users
okay_tpl.php	/assignment/ templates/content	Operating data success info

Table 7.2. Source code files list

7.5.2 Classes architecture

Table 7.3 Classes architecture provides the class file names with their corresponding purposes and methods

Class files name	Purpose	Method

Controller.php	It is responsible for mapping end-user action to application response	init()
		dispatch()
		show4addt()
		show4add()
		check4add()
		applayAdd()
		show4edit()

		check4edit()
		applyEdit()
		applyDelete()
		applySubmit()
		putBlogStats()
		rightFrame()
		leftFrame()
		putAsFooter()

dbassignment_class_inc.php	Operate data of table tbl_assignment	init()
		getAssignments()
		geAsAcc()
		getAsAcc2()

dbassignmentsub_class_inc.php	Operate date of table tbl_assignment_submit	init()
		getSubAssig()
		asAcount()
		countAll4stu()
		recoAll4stu()
		essayAll4stu()
		asCountLec()

		asGetSubRecord()
		getAllAs4Lec()

tabledisplay_class_inc.php	Format Webpage output	init()
		listAssigments()

Table 7.3. Classes architecture

The applications source code indicated Appendix A,B.

7.6 APPLICATION TEST

The overall testing for AMM is divided into functionality tests, usability test and performance testing. The present functional requirements analysis results are in the Table 7.4, list of functional test results.

In the follow Table, "**U**" represents uncompleted, "**C**" stands for the completed

Functional requirement	**Result**
Login function	
Different users operation interface (administrators, lecturers and students)	C
Lecturer users and student users have access to their own preferences in the course.	C
Creating assignment function	
Assignment types include online activity model and upload-file model.	C
The lecturers can choose to allow resubmission of assignments before they are marked.	C
Assignments can be specified with a due date and a maximum grade.	C
Lecturers can login like student to preview the assignment.	C
Distributing assignment function	

user can distribute their finished assignment by URL. The assignments are date-stamped.	C
Students can upload their assignments (any file format) to the server. These assignments are also date-stamped.	C
Late assignments are not allowed, and are shown clearly to the student.	C
Provide function to prevent multiple submissions.	C
Assignment feedback function	
Lecturer feedback information is appended to the assignment page for each student.	C
Student can down load feedback essay corresponding specific assignment	C
For each particular assignment, the whole course can be assessed (mark and comment) on one page in one form.	C
Offer electrical-pen mark-up tools to simulate normal pen. Lecturer give help online to correct students' uploaded essays.	U
Use statistical presentation to display the mark of the assignment	C

Table 7.4. List of functional test result

7.7 SUMMARY

In this chapter, a useful description was given by the development process of the AMM from requirement analysis to functional test. The developer successful developed the proposed application and delivered it to KEWL.NextGen.

CHAPTER 8 CONCLUSIONS AND FUTURE WORKS

8.1 CONCLUSIONS

8.1.1 The primary purpose of this research project

The chief goal for the project was to develop an Assignment Management Module (AMM) in an open-source environment with appropriate software functionality and features to meet target users requirements. The aim was to produce a tool making it easier to create, mark and manage assignments and record individual student performance in one application module.

8.1.2 The functionalities that have been achieved as per design

The developer has placed a high priority on the requirements imposed upon the AMM system throughout the project lifecycle. The AMM was implemented successfully and delivered seamlessly to the KEWL.NextGen. As regards functionalities, the AMM already has the following features.

Assignment Management Module offers a multilingual operation interface. It improves the usability and extensive user scope of the system.

Assignments can be specified with a due date and a maximum mark. During the education environment it can advantage lecturer-users evaluation performance with specific assignment. It also helps to overview a specific assignment effect of the whole class.

Assignment types include the online activity model and the upload-file model. These assignment types can enrich pedagogy methods in LMS, and help lecturer-users to implement their intentions successfully.

Students can upload their assignments (in any file format) to the server. The assignments are date-stamped. AMM can objectively reflect attributes of a specific assignment. It makes education actions more efficient and liberates users from daily menial tasks, for example collection of assignment papers.

The due date for an assignment is shown clearly to the student-users and late assignments are not allowed. This helps to encourage students and to remind them to submit assignments on time, which in turn lightens the excess load for the lecturer.

For each particular assignment, the whole course can be assessed (mark and comment) on one page. It is convenient for lecturer-users to operate an AMM. The application also improves the efficiency of result tabulation of lecturer-users.

Teacher feedback information is appended to the assignment page for each student. Student-users can get clear and exact specific assignment feedback information.

The lecturer can choose to allow resubmission of assignments before they are marked. This helps students to treat assignments more seriously and it encourages them improve the quality of their assignment.

8.1.2 The specific software engineering methodology that have been used to develop project.

In conclusion, the choice of which engineering methodology to use in a development project is closely related to the size of the software system and the environment in which it is supposed to function.

In open-source project environment, many developers are involved, often working in parallel. The combination of technical and nontechnical tasks that must occur (on time) to produce a high-quality application represents a challenge for any developers in the group. In order to avoid confusion, frustration, and failure, planning must occur, risks must be considered, a schedule must be established and tracked, and controls must be defined. These are the core activities in the software engineering methodology. Through the AMM project lifecycle, the author adopted the waterfall engineering methodology to control the development process and to organise the project procedure, and this mechanism ensured that project goals were achieved.

8.1.3 The design pattern that have been used to develop this project

Because KEWL.NextGen is open-source project, and adopted "Bazaar" style development origination—the software is released as often as possible to encourage contributions from the developers. During the case study phases, in the collaborate development environment of KEWL.NextGen, the AMM was developed in parallel with the basic function classes of KEWL.NextGen. As the business logic class and database operation classes are inherited from basic function classes, there are dependency relationships with AMM and system function classes. AMM also has interaction relationships with other functional modules, for example the "Course Management Module". These modules were also in the development process. During that period, there existed many indeterminate and variational factors in the development environment.

There are many ways to look at a problem to be solved using a software-based solution. One widely used approach to problem solving takes an object-oriented viewpoint. The problem domain is characterised as a set of objects that have specific attributes and behaviors. The objects are manipulated with a collection of functions and communicate with one another through a messaging protocol. Work was based on the Object-oriented development principle, flexibly using the MVC design pattern to establish interaction interface functions and data interface functions. The application only has sample parameters that transfer relationships with the interaction interface functions or data interface functions. If the environment changes, the correlative parameters just need to be changed. It also means that system function classes, other modules and presentation layer changes can be made freely without altering the AMM logic class. These methods are used to deal successfully with challenges as well as to improve source code maintainability, reusability and reliability.

8.1.4 Found and lesson that have been met in developing the project

During the developing process of AMM, the developers found that object-oriented concepts are the peak of the software development approaches to develop the project. Object technologies lead to reuse, and reuse leads to faster software development and higher-quality programs. Object-oriented software is easier to maintain because its structure is inherently decoupled. This leads to fewer side effects when changes have to be made a less frustration for the developers. In addition, object-oriented systems are easier to adapt and easier to scale.

It is important that an object encapsulates both data and the processing that is applied to the data. This characteristic enables classes of objects to be built and inherently leads to libraries of reusable classes and objects. Because reuse is a critically important attribute of modern software engineering, the object-oriented principles are attractive to many software development organizations. At each stage of object-oriented work, the project is reviewed for clarity, correctness, completeness, and consistency with the framework of LMS.

8.2 RECOMMENDATIONS

During the development process of AMM and due to time and environment limitations as well as lack of hardware company support, electrical-pen tools were not developed. A future project should try to get some hardware company support (get the I/O parameters, offer the product sample, etc.) and to further developing.

The function should be achieved in the next step. Further, rich assignment types are also necessary for example an offline-activity model is needed, as well as added assignments with artificial intelligence assessment (AI) functions.

The target audience of this document is anyone with an interest in an open-source software project in general, and in a Learning Management System in particular. If the reader has also been a contributor to Learning Management System, and especially to the management of assignments, this document may provide additional value, in that it strives to present a new approach to the understanding of such a module.

REFERENCES

[1] Barron, T. (1997).*The future of digital learning*. E-learning 8 Vol. 1, No.2, pp. 7-46.

[2] Barron, T. (2000) *The revolution of e-learning*. E-learning 11 Vol. 1, No.1, pp. 18-42.

[3] Bassi, L. (1997). *Training industry trends*.MIT, The McGraw-Hill.

[4] Bethoney, R (2000). *Modern Learning*. Microsoft Press

[5] Cann, A. (1999) *Approaches to the evaluation of online learning materials*. International Journal of Educational Telecommunications, 2(2/3), 10-15.

[6] Chee, Y. S. (1996). *Mind bridges: A distributed multimedia learning environment for collaborative knowledge building*. International Journal of Educational Telecommunications, 2(2/3), 137-154.

[7] Conner, R. B. (2000). *Computer-assisted learning*. McGraw-Hill Book Company.

[8] Derek Keats (2004) *KEWL.NextGen Manual for Developers*. UWC, 2004.

[9] Driscoll, M. (1998). *Web-based training American*. Journal of Distance Education 6(2), 89-102.

[10] David Ruble（1997）.*Practical Analysis and Design for Client / Server and GUISystems*. Prentice Hall PTR.

[11] Filipczak, Bob (1998). *XML: meta-language for the Web?* Training, v. 35 no. 6, pp. 14-113.

[12] Galagan, P. (2000). *Getting started with e-learning*. Training and Development54 (4), p. 62-64. Prentice Hall PTR.

[13] Ganzel, R. (1999). *What price online learning?* Training, 36(2), 50-54.

[14] Gibson, C. (1996). *Toward an understanding of self-concept in distance education*. American Journal of Distance Education, 10(1), 23-36.

[15] Hall, B. (2000). *New study seeks to benchmark enterprises*. Washington, DC: Wolf Publishing company inc.

[16] Hites, J. M. (1999). Design and delivery of training for international trainees: A case study. Performance Improvement Quarterly, 9(2), 57-74.

[17] Johan M.(2002) http://www.uwc.ac.za/ems/is/hicte/

[18] Levy, S. (2001).*Hackers:Heroes of the Computer Revolution*. Penguin Books.

[19] Karon, R. L. (2000). *Bankers go online: Illinois banking company learns benefits of e-training*. E-learning, 1 (1) 38-40.

[20] Mathiassen, j.(1998).*prototyping and specifying:Principles and*

Practices of a Mixed Approach. Scandinavian Journal of Information Systems, vol7, no.1,4, 20-69.

[21] McCain, M. (1999,). *Scaling the Great Wall.* Training and Development,7.38-42.

[22] Peter, E. (1992). *Categories of Free and Non-Free Softwar.* eBantam Books.

[23] Pressman, R.S. (1992).*Software engineering: Practitioner's approach*. McGraw-Hill Book Company.

[24] Raymond, J. J. (2003). *Handbook of Open-source project (3rd ed.)*. Houston, TX: Gulf Publishing.

[25] Richards, A. C. (1998). *Assessing distance learners*. Peachpit Press.

[26] Sommerville, I. (1999). *Software engineering*. Addison-Wesley Publishing Company Inc.

[27] Stallman, R. M. (1998). *What is copyleft?* Bantam Books.

[28] Steve, M. (1998). *Software Project Survival Guide*, Microsoft Press.

[29] Wellins, R. (2000). *The growing pains of globalizing HR*, E-learning, 11 (2) 65-71.

[30] Johan M.(2002) http://www.uwc.ac.za/ems/is/hicte/

INTERNET SOURCE

[1] (Internet source 1), (iet.ucdavis.edu/glossary/). Access time: 2003.04.25.

[2] (Internet source 2), (http://moodle.org/doc/). Access time: 2003.04.25.

[3] (Internet source 3), (http://www.eweek.com/article2/0,3959,808852,00.asp). Access time: 2003.05.09.

[4] (Internet source 4). (Www.onlamp.com/pub//lamp .html). Access time: 2003.05.18.

[5] (Internet source 5), (http://www.lunix.org) Access time: 2003.07.06.

[6] (Internet source 6), (perdictionary.com/computing/apache) Access time: 2003.10.16.

[7] (Internet source7), (www.mysql.com/introduction.html) Access time: 2004.06.27.

[8] (internet ource 8),(http:// www.distancelearning.org / glossary.html) Access time: 2004.07.30.

[9] (Internet source 9), (computing- dictionary. thefreedictionary. com/class). Access time: 2004.08.07.

[10] (Internet source 10), (http://www.fsf.org). Access time: 2004.08.10.

[11] (Internet source 11), (ming.ncsa.uiuc.edu /elearntr /standards.html). Access time: 2004.08.13.

[12] (Internet source 12), (http://www.imsproject.org/) Access time: 2004.09.03.

APPENDIX A: AN EVALUATION OF THE DEVELOPMENT OF AN ASSIGNMENT MANAGEMENT MODULE (AMM) AGAINST THE INFORMATION MANAGEMENT BODY OF KNOWLEDGE (IMBOK) FRAMEWORK

Abstract

Scientific research and development have been conducted in the area of Information Management for more that a decade. An immense amount of research has been done - covering the development and future perspectives of the discipline.

This paper presents an evaluation of the development of an Assignment Management Module (AMM) within the KEWL.Nextgen Learning Management System (LMS) environment against the Information Management Body of Knowledge (IMBOK) framework. The KEWL.Nextgen LMS (Knowledge Environment for Web-based Learning) is an Open Source Software LMS.

Keywords

Information Management, Learning Management System, Open Source Software, IMBOK, KEWL.Nextgen, AMM

Introduction

The generation and usage of information is an essential facet of modern business. It has become critical to effectively and efficiently manage information, enabling businesses to maximise the impact of the deployment of both information and other resources.

IMBOK offers a simple framework that defines an almost intuitive Information Technology (IT) to Business Strategy continuum. The framework facilitates the efficacy analysis of IT-driven business processes. The structured approach to the analysis facilitates the identification and isolation of problem areas, as well as opportunities to improve the system – reaching from consideration of raw technologies right through to issues of business practice and business strategy (Bytheway, 2004).

The Framework of IMBOK is presented as five areas of knowledge and four processes, as detailed in Figure 1 below:

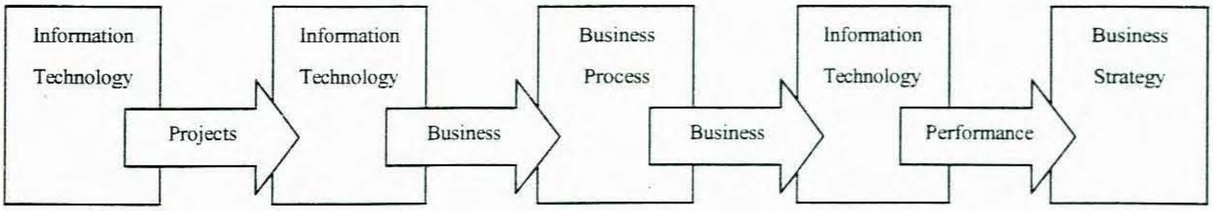


Figure 1: The IMBOK Framework

Moreover, IMBOK is intended to provide a reference framework for those who are concerned to bridge any actual or perceived divides between information technology specialists and business generalists (Bytheway, 2004). Accordingly, this framework is useful for the student to organize ideas further to evaluate development works for Information projects.

The evaluation process of the AMM will be divided into two steps. The first step is a general view of evaluation. It is a synthesis to show the evaluation process of AMM. The following step is a specific evaluation process of AMM. It categorizes detailed information of development AMM. The knowledge areas of the IMBOK framework - it includes information technology, information system, business process, business benefit, and business strategy.

General View of Evaluation AMM against IMBOK

The follow diagram, adapted from Transforming information technology into business benefits, shows the general view of evaluation AMM project in a graphical format.

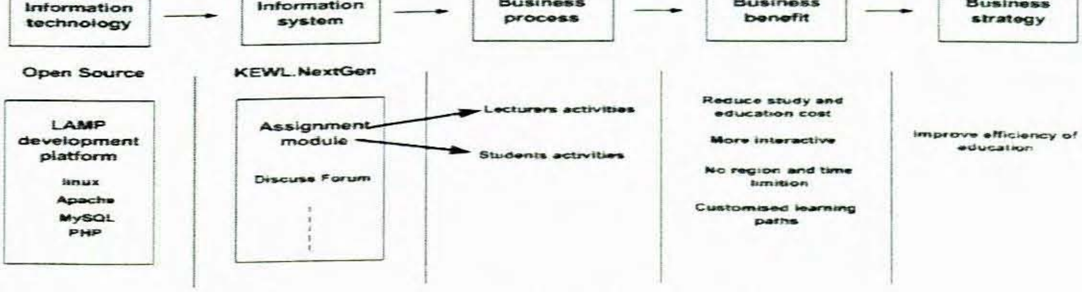


Figure 2: Evaluation of AMM Project

Evaluation for AMM against IMBOK Framework

The Information Technology

Bytheway (2004) refers to Information Technology as the specific technical components, normally organised as hardware, software and communications, which are used to make up an information system.

AMM is one of the functional modules of KEWL.NextGen and it is web-based built on PHP sand MySQL. The main technologies used include UML for design, PHP as the cross-platform scripting language and PostgreSQL or MySQL as the primary back end database.

Software	Hardware	Communications
Linux (Operate system)	Web server	Internet
Apache (Web server)	Document server	Mailing list
MySQL (Database)	Database server	Discuss forum
PHP (Development Language)	Work station	
CVS (Cooperate Development Tool)	Printer	

Table 1: The AMM Information Technologies

The Linux operating system, Apache Web server, MySQL database and PHP, comprise the "LAMP" development environment. All the components of "LAMP" can be downloaded and used for free. These features can maintain the open-source benefits for the AMM. IMBOK makes it very clear that these technology components need to be engineered into usable and useful systems that serve a real need.

Managerial issues concerning Information Technology

Bytheway (2004) clearly categorizes managerial issues as comprising of five aspects - managing suppliers, acquiring technology, managing the technology portfolio, technology competency and budget management. These issues must be dealt with, in the domain of information technology to ensure it is used properly. The following section will explain these issues in the AMM project.

The AMM is not only a software development project but also a Master study project, which aims, through the development actions, to train the student how to practice software engineering methodologies in a collaborative development environment. Furthermore, it allows the student to practice how to organise, management and control the information development project.

In the AMM project, the developer chose a modified waterfall engineering methodology. The modified waterfall model is not the same methodology as a standard waterfall model. It uses the same phases as the pure waterfall model, but is not done on a discontinuous basis.

This enables the phases to overlap when needed (http://www.business esolutions.com) and in each phase assists the creation of corresponding documentation. Using the milestones (specific method is major activities in the AMM project) and deadlines (specific method is Project Gantt chart in the AMM project); it is easy for developer to control the project process and also help the supervisor to measure the study process.

The modified waterfall can be further modified using options such as prototyping, spiral or other methods of requirements (http://www.business esolutions.com). This feature is very important in the KEWL.NextGen development environment and will be discussed in the following section.

Managing suppliers and acquiring technology

The AMM is an integral part of KEWL.NextGen. The developers of UWC provided the KEWL.NextGen's technical framework. It is MVC pattern design, and based on LAMP development platform. It is typical open-source development environment. This software is freely available from Internet. The developers of UWC also provided some technical support about technology of framework, such as core classes' direction and PHP syntax analysis.

The functionalities requirement about AMM is very blurry and the developer seldom gets chance to meet target users. There is still a chasm of some kind between the average "IT" person and "business" person, and that must be the focus of our attention. It must be bridged (Bytheway, 2004).

In an open-source environment, the developer can get the source code of the last version software and similar module from other system. The developer has to use prototype methodology to organize the business process and deduce the functionalities for the AMM. For the afore mentioned reasons - the modified waterfall is a reasonable choice for management AMM project.

Managing the technology portfolio

As AMM was based on a collaborative development environment, a central control was needed. In this project, Concurrent Versions System (CVS) was used to balance developers' development behaviour and development process.

Technology competency

In this project, the developer of Peninsula Technikon simulated the same environment as Kngforge.uwc.ac.za Website in the intranet. This was more convenient for developing and avoided the risks associated with development affecting the original environment on the UWC website.

Budget management

The management plan of AMM includes a budget management section. AMM is in fulfilment of the requirements for the Master degree of Technology in the Faculty of Information Technology at the Peninsula Technikon. The development team of Peninsula Technikon volunteer to develop KEWL.NextGen. Peninsula Technikon offers the development environment. In the AMM project, budget management just stays in a conceptual phase.

The Information System

According to Bytheway (2004), an information system is not the same as the technology upon which it is based: it is the totality of technological and human components that work together to produce the information systems and services that a business needs, and that processes information for some organisational purpose.

Application Context

An Assignment is a task given by a teacher to a student, in order for a lesson to be learnt or prepare. AMM offers an easy tool that will be constructed in order to make it easier to create, mark and manage assignments and record individual student performance in the Learning Management System (LMS). The core functionalities of the AMM are:

- To collect and organise electronic assignments through the AMM
- To submit assignments, through corresponding course content areas related to

the assignments, thereby linking lessons to student deliverables and results, and

- The return of and comment on corrected files, as well as store of private notes related to student work in the AMM

Figure 3 below, visually explains what assignment management module is:

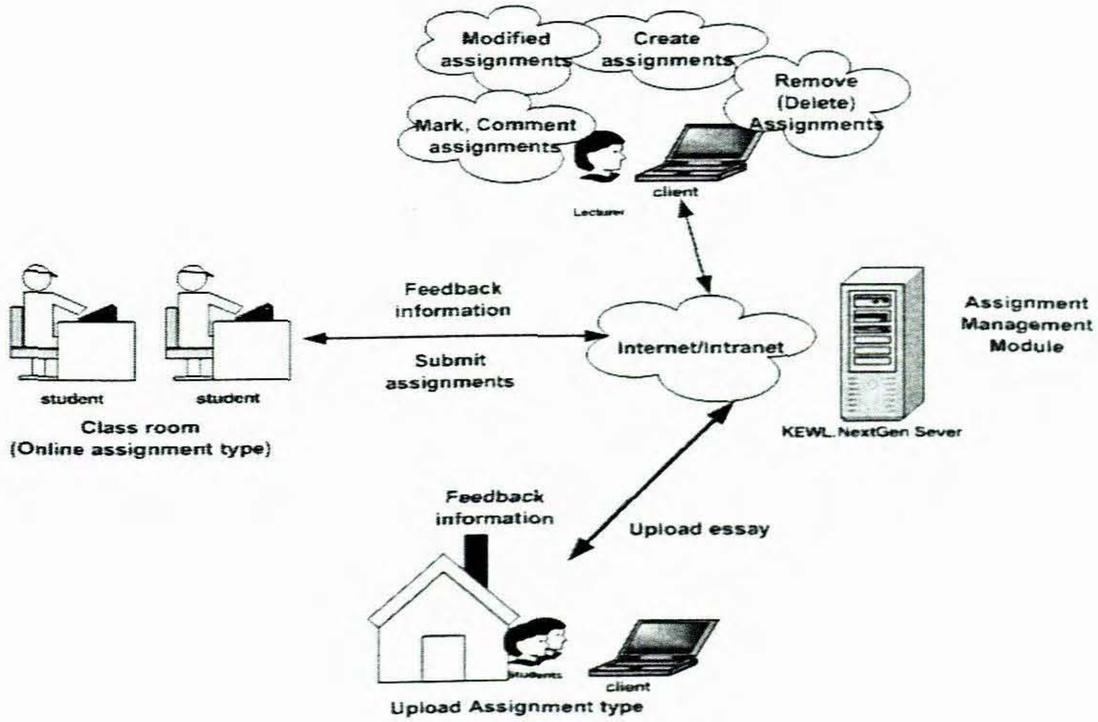


Figure 3: Context of AMM

Application Structure

In this area, the first step is to analyse the operations of users in this module. These operations fulfil the activities of users including view assignment, create assignment, edit assignments, mark assignments, and submit assignments etc. further analysis includes the complete operations to meet the target users' needs. The last step is to analyse the entity relationship, resulting in the deduction of the database structure. The developer refines the entity relationship in Figure 4 below.

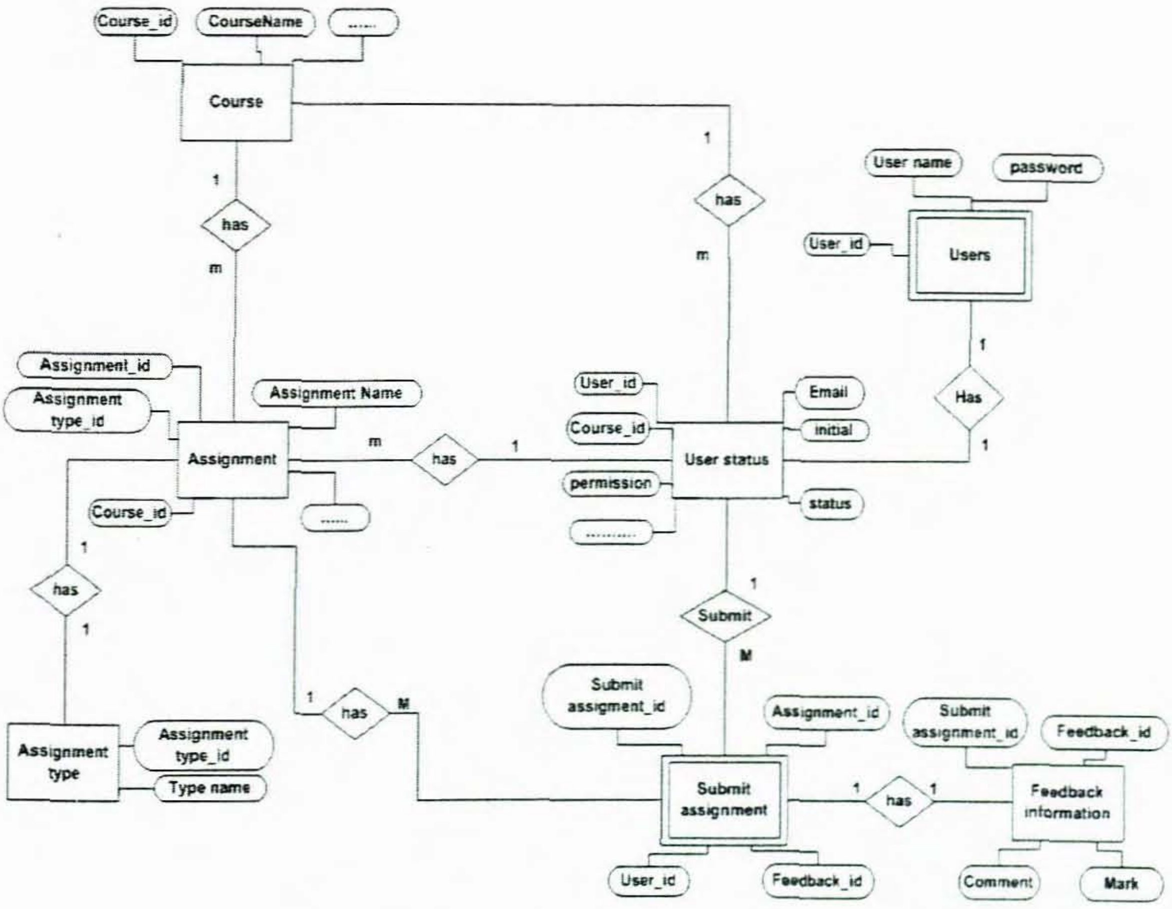


Figure 4: Entity Relationship for AMM

The Business Process

Bytheway (2004) defines a business process as a logical envelope that co-ordinates and gives purpose to business activities; generally where an activity delivers an output, a process delivers an outcome – a result that is evident to stakeholders outside the business as well as those within (Bytheway, 2004).

According to the traditional education environment and from the users' point of view, the developer organizes the users' business process as follows:

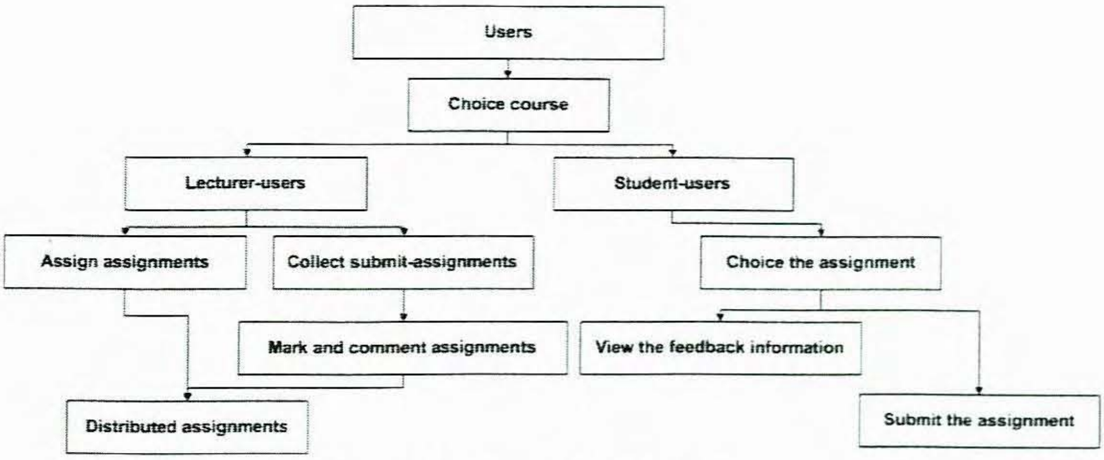


Figure 5: Users Business Process

In the AMM project, the main research activities are as follow:

- Establish laboratory, install Linux, PHP, MySQL and server
- Master Linux, PHP, and MySQL
- Requirements analysis
- Functional requirements analysis
- Non-functional requirements analysis
- Create preliminary program
- Debug and test prototype program
- Deliver the application to the existing LMS
- Debug and test prototype programme
- Evaluate program
- Create a more complete program and write a report

The Business Benefit

Bytheway (2004) refers to business benefit as – "The process of organising and managing, such that the potential benefits of an investment of time and effort are actually realized." To achieve the mentioned benefits and to fulfill at least the top-level strategy, numerous advocates had been organized to comprise an Open Source development team.

The several meetings had been held in the beginning phase to discuss the LMS business processes. The current running version of LMS KEWL was provided as a good reference in this phase. After analyzing the processes, the skeleton of LMS was described as different functionality modules, which are opened to the all Open Source advocates, which can join the open collaborative development environment.

This project is aims to bring the following benefits:

- Higher education institutions in Africa drive the production of new and innovative software for use in the higher education sector, as well as in other educational sectors, business and government;
- Improved open source advice available to education, business and government;
- Improved accessibility and enhanced local support for open source software in education, business and government
- Increase in the number of graduates trained in the application of open source principles;
- Enhanced relationships between higher education and business built around the development and support of open source software;
- Enhanced employment opportunities for graduates of higher education institutions.

The target users of AMM are the students and lecturers of higher education institutions. Nowadays, most students and lecturers have the ability to operate the computers. So conceptually, it is feasible to transform traditional education assignment action into a Web based application. Target users can accept this business change.

AMM offers a number of potential benefits that can help engage students in activities that contribute to their intellectual growth. For example, online assignments offer interactive interfaces often requiring greater reflection than in traditional in class assignments. Other benefits include the rapid capture and turnaround of knowledge, generation of customised learning paths based on individual strengths or weaknesses, no time and region limitations and at a very economical price, and it is far less expensive than traditional instructor led training.

The Business Strategy

Strategy is about change. Without change, there is no real need for strategy. With change in mind, we can argue that the simplest definition of strategy is: knowing where you are, knowing where you could choose to be, and knowing how you intend to get there (roughly speaking) (Bytheway, 2004).

Learning Management System is a broad term that is used for a wide range of systems that organise and provide access to online learning services for students, teachers, and administrators. KewlNextGen's strategy is to harness the enormous potential that exists within Africa and the African Diasporas to create a core of open source software developers who are able, through software development activities, to create educational and business opportunities that contribute to development on the continent (Derek Keats & Melisse Benn, 2003).

References

1. Andy Bytheway 2004. The Information Management Body of Knowledge "IMBOK",http://www.imbok.org
2. Derek Keats. and Melisse Benn 2003.African virtual open initiatives and resource, Kngforge.uwc.ac.za
3. Conner, R. B. (2000). Computer-assisted learning. McGraw-Hill Book Company.

End-of-appendix A

APPENDIX B:KEWL.NEXTGEN LEARNING MANAGEMENT SYSTEM

Learning Management System

A Learning Management System (LMS) is defined as a learning software application or Web-based technology used to plan, implement, and assess a learning process. Typically, a learning management system provides an instructor with the tools to create learning resources, deliver content, monitor student participation, and assess student performance. Figure 1 shows typical LMS communications.

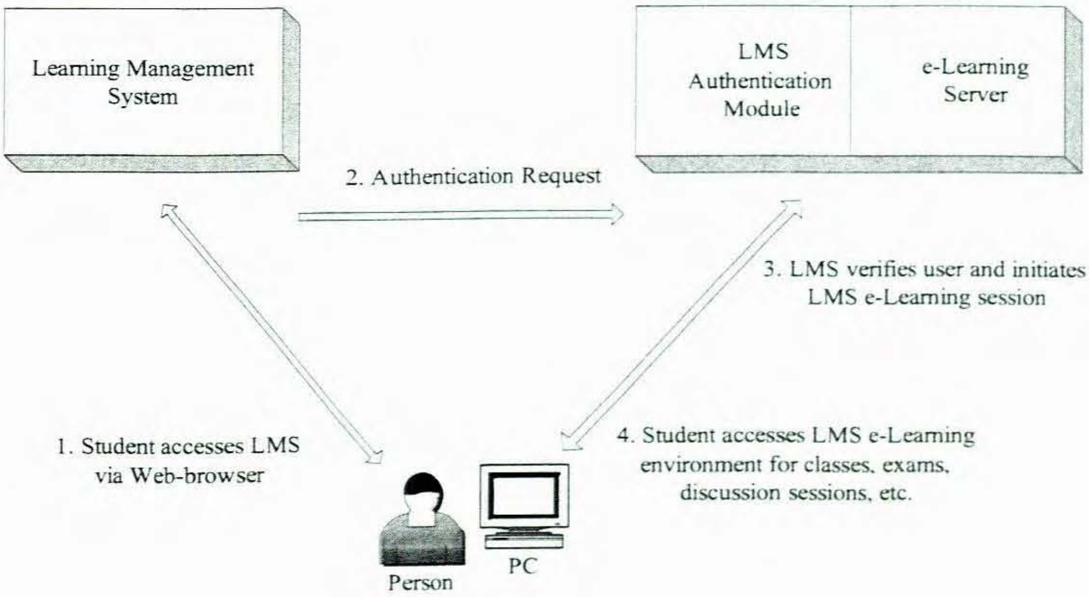

Learning Management System
LMS Authentication Module
e-Learning Server
2. Authentication Request
3. LMS verifies user and initiates LMS e-Learning session
1. Student accesses LMS via Web-browser
4. Student accesses LMS e-Learning environment for classes, exams, discussion sessions, etc.
Person
PC

Figure 1: Typical LMS Communications

A learning management system provides an online learning environment by enabling the management, delivery and tracking of learning. Furthermore, an LMS should support a collaborative learning community, offering multiple modes of learning from self-paced coursework (Web-based seminars and classes, downloadable, CD-ROM and video content) to scheduled classes (live instruction in classroom settings or online) to group learning (online forums and chats). Optimally, an LMS will consolidate mixed-media training initiatives, automate the selection and administration of courses, assemble and deliver learning content, measure learning effectiveness and integrate with other enterprise applications.

An LMS is a Web-based software solution to simplify the administration of learning programmes. It creates efficient processes for both learners and administrators. For learners, an LMS tracks their progress through a programme of study or provides a forum for collaboration with peers. An LMS provides administration functionality that simplifies the enrolment and participation in a learning programme. It provides a catalogue for courseware and learning materials, it offers notification options and the ability to collaborate online with instructors or fellow students. For administrators, using an LMS reduces or eliminates the management headaches of running a learning programme. For testing and assessment, as well as for competency certification, it standardises the process and evaluation. It tracks the success of individual students or specific courses. It integrates the marketing and accounting function to enable simple cost-benefit analysis. It controls resources – from materials to curricula, to programme offerings, to course scheduling.

Learning management Systems are used today in many organisations (universities, schools, and corporations). Traditional LMSs provide a content repository for course materials as well as facilities for student (trainee) tracking and management. Additionally, some LMSs provide authoring tools, assessment tools and communication tools, such as e-mail and discussion groups. A learning management system optimally should:

- Consolidate training initiatives on a scalable, low-cost, Web-based platform;
- Assemble and deliver learning content rapidly in multiple languages;
- Measure the effectiveness of training initiatives;
- Mix classroom and online learning;
- Integrate with other target group application solutions;
- Centralize and automate administration;
- Use self-service and self-guided services as much as possible;
- Personalise content and enable knowledge re-use.

The Knowledge Environment for Web-based Learning

The Knowledge Environment for Web-based Learning (KEWL) was developed at the University of the Western Cape to facilitate research into online learning by Derek Keats and a team of developers. The development centred on a comprehensive learning management tool that could be used to investigate the online learning process. Although it was initiated and funded as a research project, it soon became apparent that KEWL could serve the online learning needs of the University of the Western Cape.

KEWL.NextGen is a web-based training platform built on PHP and MySQL. KEWL.NextGen has most of the features common to commercial learning management systems. The following features are included: learner desktop containing information about courses visited, new mail and forum entries; learning environment with notes, tests, glossary, and search engine; course management; communication and collaboration tools including mail, forums and chat, group work systems, integrated authoring environment, support for metadata, context sensitive help; and interfaces for both learning and administrator.

The short list was systematically evaluated using hands-on testing offered through demonstration user accounts to understand the product features and also followed up by inspecting the online help, user and instructor documentation, and commentary of the user community in order to rate the short list candidates.

The following table describes the findings of a survey-style evaluation of KEWL and the other classical LMSs.

Criteria	Moodle	LON-CAPA	ILIAS	dotLRN	Atutor	KEWLNextGen	Notes
Security Features							
Encryption							None offer SSL
Authentication	*	*	*	*	*	*	Only basic login
Access Features							
Login/pwd	*	*	*	*	*	*	Password reminder exists
Roles/assignable privileges	*	*	*	*	*	*	
Browser-accessible	*	*	*	*	*	*	
Course Authorization	*	*	*	*	*	*	Instructor approves enrolment
Registration Features	*		*	*		*	Basic student contact data is retained
Course Design, Development, Integration Features							
Customize look	*	*	*	*	*	*	User can choose appearance
Both Classroom Distance Ed Support			*	*			No synchronous learning features- but content could be used to supplement classroom teaching.
Templates	*		*	*			Appearance templates, not pre-structured course skeletons
Web authoring	*		*	*		*	Simple content construction can be performed.
Multimedia support	*	*	*	*	*		Includes as links
Accessibility	*	*	*	*	*	*	
Instructional Specification Support	*		*				
Easy Navigation	*	*	*	*	*	*	Menu and icon-based
Easy Course Structuring	*	*	*	*	*	*	
Style Sheets	*	*	*	*	*	*	
Course Monitoring							

Course Listing	*	*	*	*	*	*	
Course Description	*	*	*	*	*	*	Based on author's contribution
Schedules and Availability control	*	*	*	*		*	Set course start data
Assessment Features							
Creates test question and facilitates test administration	*		*	*	*	*	Can include tests anywhere in course
Automated testing and scoring	*		*	*		*	
Learner Profile Management				*			
Self-assessment					*	*	
Online Grading	*			*	*	*	
Collaboration Features							
Messaging						*	
Email	*	*	*	*		*	
Chat	*		*	*	*	*	Integrated
Bulletin boards	*	*	*	*	*	*	Notice board
Newsgroups		*					
File exchange	*	*	*	*	*	*	Post files
Whiteboard			*				
Forums	*	*	*	*	*	*	
Productivity Features							
Bookmarks			*				
Calendar	*	*	*			*	
Orientation/Help	*	*	*		*	*	
Searching			*		*	*	Forum search

Table 1: Feature of LMSs

The KEWL.NextGen application framework is based on a close approximation of the model, view, controller (MVC) design pattern, which separates an application's data model, user interface and control logic into three separate components. This enables changes to be made to one component without impacting on the others. Since the model is usually stable, separating it from the view and controller logic that a changed often during development leads to more robust applications that are easier to maintain.

The UWC developers built KEWL.NextGen framework - see Figure 2. The main part is Classes File; it contains four classes, namely dbtable, object, engine and controller. The project work is to write modules file, which inherit attributes and methods from Classes File.

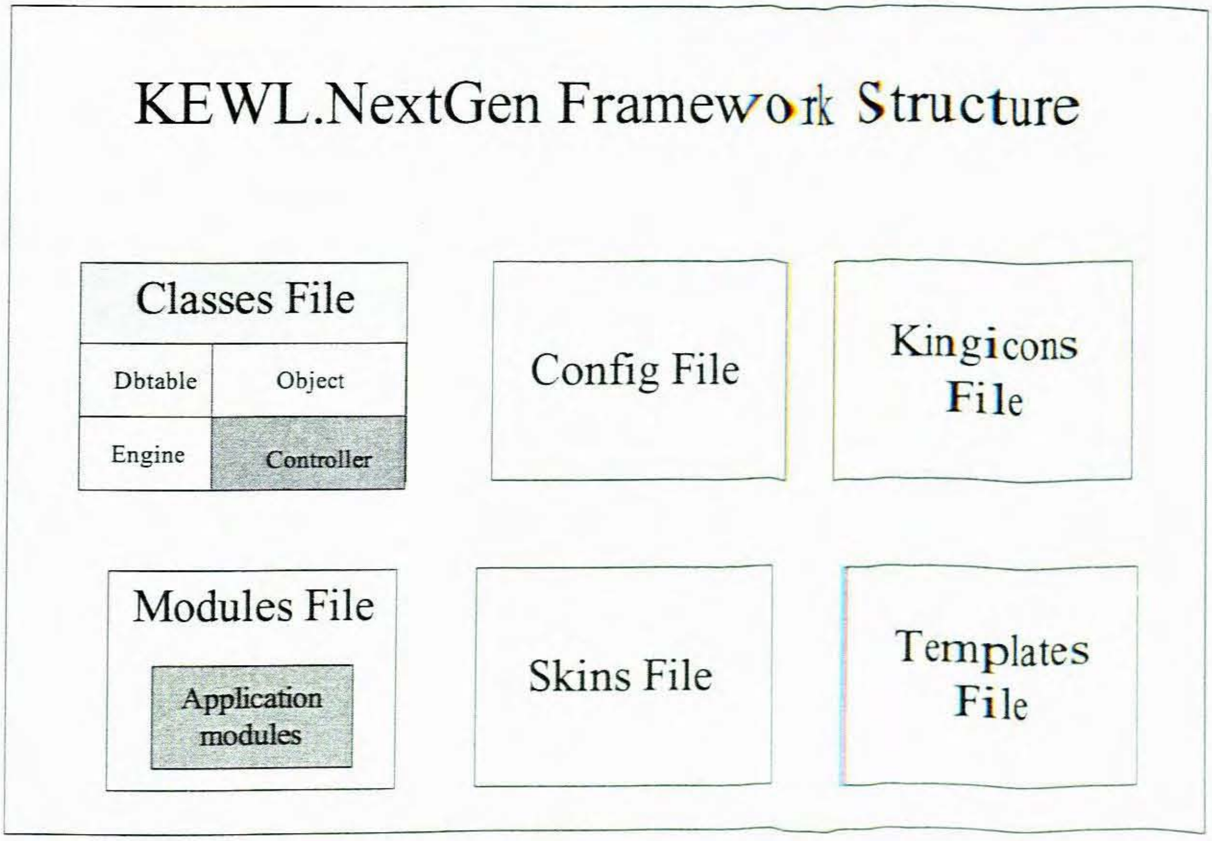


Figure 2: KEWL.Nextgen Framework Structure

Note: Application modules are developed by inheriting attributes and methods from super-class – Classes File.

Figure 3 illustrates the architectural detail of the Modules File. The Module Name File is constituted of the following three files:

- Controller File for Control Logic
- Classes File to define the application's Data Model
- Template File which includes the Content File that defines the User Interface

Application Modules are defined by unique tables, and the model represents the data as the Application Object that can be manipulated.

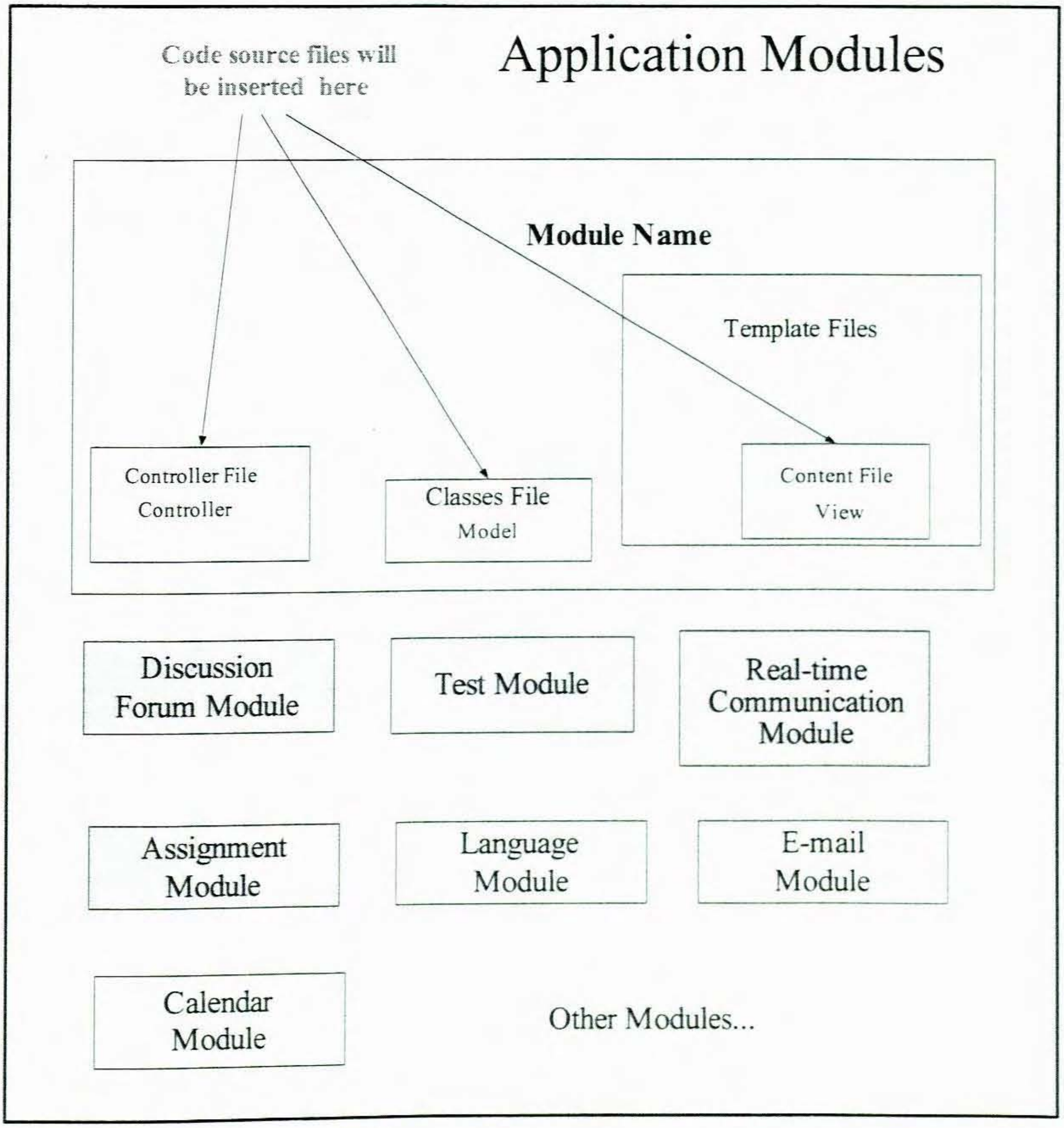


Figure 3: Application Modules Architecture

Note: The KEWL.Nextgen application framework is based on a close approximation of the model, view and controller (MVC) design pattern, which separates an application's data model, user interface and control logic into three separate components.

End-of-appendix B

APPENDIX C: SOURCE CODE FOR AMM

```
<?php

   /****************************************************************

   *******       assignment class extends controller     **************

   ****************************************************************/


   // security check - must be included in all scripts

   if (!$GLOBALS['kewl_entry_point_run']){

     die("You cannot view this page directly");

   } // end security check


   /***************************************************************

   ******* Module class to handle management of assignments  *********

   ****************************************************************

   * @author Ke Sun                                              *

   * aknowledgements the application flow and idea come from blog and *

   * useradmin module.  Particular thanks should be given to          *

   * Derek Keats and James Scoble who have given me big help.         *

   * $Id: controller.php                                          *

   ***************************************************************/


class assignment extends controller

{

      var $objButtons;// form unit

      var $dropdown;  // form unit

      var $tablea;     // from dbassignment_clas.php

      var $tableb;     // from dbassignmentsub_class.php

      var $tablec;     // from dbarcourses_class_inc

      var $goaim;      // shows the aim webpage for after delete action;
```

```
    var $rstatus;  // shows whether a function-call did what was wanted or not
    var $rvalue;   // the return-value for the template to be used.

 function init()
 {
   // $this->objButtons=&$this->getObject('buttons','display');
    $this->objUser = &$this->getObject('user', 'security');
    $this->loadClass('htmltable', 'htmlelements');
    $this->objLanguage = &$this->getObject('language', 'language');
    $this->tablea=&$this->getObject('dbassignment');
    $this->tableb=&$this->getObject('dbassignmentsub');
    $this->tablec=&$this->getObject('dbarcourses');
    $this->tableDsplay=&$this->getObject('tabledisplay');
    //Create an instance of the trimstring object to trim the output
    $this->objTrimStr = &$this->getObject('trimstr', 'strings');
 }


function dispatch($cmd)   wrapper for Sean's code
{

   if (!isset($id))  //userId might be passed via $_GET or $_POST
   {
      $id='';
      if (isset($_POST['id']))
      {
         $id=$_POST['id'];
      }
      else if (isset($_GET['id']))
      {
        $id=$_GET['id'];

    }

 }


 if (!isset($courseid)) // $userId might be passed via $_GET or $_POST

 {

    $courseid='';

    if (isset($_POST['courseid']))

    {

       $courseid=$_POST['courseid'];

    }

    else if (isset($_GET['courseid']))

    {

       $courseid=$_GET['courseid'];

    }

}

switch ($cmd)

{

 case 'coch':

    return'coursech_tpl.php';

    break;


 case 'addt':

    $courseid=$this->getParam('courseid');

    return $this->show4addt($courseid);

    break;


 case 'add':

 case 'newassignment':

 case 'Add':
        case 'New Assignment':
            return "asadd_tpl.php";
            break;
        case 'addassignment':
        case 'add assignment':
        case 'Add Assignment':

$check=$this->check4add($this->getParam('name'),$this->getParam('description'),$this->
getParam('duedate'));
            $goaim=$this->getParam('goaim');
            if ($check=='Looks Okay'){
                return $this->applyadd($goaim);
            }else{
                $this->rstatus=$check;
                return'error_tpl.php';
            }
            break;
        case 'edit':
        case 'Edit':
            $this->setVar('admin_user',TRUE);
            return ($this->show4edit($id));
            break;
        case 'applychange':
        case 'apply changes':
        case 'Apply Changes':
            $check = $this->check4edit();
            $goaim = $this->getParam('goaim');
            if ($check == 'Looks Okay'){
                return ($this->applyedit($goaim));
            }else{
             $how=$_GET['how'];

             $match = stripslashes($_GET['searchField']);

             $asdata = $this->tableDsplay->

ListAssignments($how,$match,'TRUE','a','listassignment');

             $this->setVar('asdata',$asdata);

             return'list_as_tpl.php';

             break;

         // the main interface of assignment lecture function

        case 'listlect': // either will trigger the code

        case 'listLeAssignment': // either will trigger the code

        case 'List Lecture Assignment':

             return'mainlec_tpl.php';

             break;

         // the main interface of assignment student function

         case 'listStu': // either will trigger the code

         case 'liststu':

         case 'liststAssignment': // either will trigger the code

         case 'List Student Assignment':

             return 'mainstu_tpl.php';

             break;

         case 'AssubInsert':

         case  'assubinsert':

             $asdata=$this->tablea->getAssignments('id',$id);

             $this->setVar('asdata',$asdata);

             return 'assubinsert_tpl.php';

             break;

         case 'AssubUpdate':

         case  'assubupdate':

             $asdata=$this->tablea->getAssignments('id',$id);

             $this->setVar('asdata',$asdata);
```

```
            return'assubupdate_tpl.php';

            //$this->rvalue='assub_tpl.php';

            break;

        case 'AssubView':

        case  'assubview':

            $asdata=$this->tablea->getAssignments('id',$id);

            $this->setVar('asdata',$asdata);

            return'assubview_tpl.php';

            //$this->rvalue='assub_tpl.php';

            break;

        case 'AssubMark':

        case  'assubmark':

            $asdata=$this->tablea->getAssignments('id',$id);

            $this->setVar('asdata',$asdata);

            return'assubmark_tpl.php';

            break;


        case 'Applysub':

        case 'applysub':

            return($this->applysubmit());

            break;

        // the Interface for Lecture manage assignment item

        case 'AsLi4Lec':

        case 'asli4lec':

            $how='courseid';

            $match=$this->getParam('courseid');

            $asdata=$this->tableDsplay->

ListAssignments($how,$match,'TRUE','t','',$match);

            $this->setVar('asdata',$asdata);

            return'list_as_tpl.php';
        break;

    // the Interface for lecture view and evaluate the assignment

    case 'AsLi4LeM':

    case 'asli4lem':

        $how='assignment';

        $match=$this->getParam('id');

        $asdata=$this->tableDsplay-> ListAssignments($how,$match,'TRUE','mm');

        $this->setVar('asdata',$asdata);

        return'list_as_tpl.php';

         break;

    //this is the main interface of assignment admin

    case 'listasma':

        return 'mainadmin_tpl.php';

        break;

   default:

        return'list_asma_tpl.php';

        break;

  }


}




 /**

* method to display page for lecture adding new assignment

* @author Ke Sun

* @param string $courseid: it is for to reseach assignment in  respective course :

*/

function show4addt($courseid)

{   $r1=$this->tablec->getCourse('contentId',$courseid);
```

```
         $line=array_shift($r1);

         if (!$line){

            $this->rstatus = 'Error - no this course';

            return 'error_tpl.php';

         }else{

            $this->setvar('courseid',$courseid);

            return'asaddt_tpl.php';

         }

     } // end of function show$addt


    /**

    * method to insert assingment-info into database  (db_assignment)

    * @auther Sun Ke

    */

    function applyAdd($goaim='')

    {

       $tablename = "tbl_assignment";

       $dob=$this->getParam('duedate');

       /* it is for get timestamp from texinput unit;

       *the input formst is year/month/day/hour/minute/second(yyyy/mm/dd/hh/ii/ss).

       *

       */

       $duetime = mktime(substr($dob,11,2),substr($dob,14,2),59,substr($dob,5,
2),substr($dob,8, 2),substr($dob,0, 4));

       $modtime = time(now);

       $sdata['id'] = null;

       $sdata['courseid']=$this->getParam('courseid');

       //this userid is for assignment onwer (status:lecture or administroator).it is
dbl_assignment "userid"field

       $sdata['userid']=$this->objUser->userId();
    $sdata['name']=$this->getParam('name');

    $sdata['description']=$this->getParam('description');

    $sdata['resubmit']=$this->getParam('resubmit');

    $sdata['type']=$this->getParam('type');

    $sdata['timedue']=$duetime;

    $sdata['mark']=$this->getParam('mark');

    $sdata['timemodified']=$modtime;

    $r1=$this->tablea->insert($sdata); // calling Derek's SQL Insert function

    if (!$r1){

        $this->rstatus ="Changes have not been made. Database error.";

        return'error_tpl.php';

    }else{

        switch ($goaim){

            case 'asli4lec':

                return $this->dispatch('asli4lec');

               break;

            case '':

                return $this->dispatch('listassignment');

        }// end of switch


    }

} // end of function applyadd


/**

* method to check assingment-info before adding into database

* @author Sun Ke

* @param  string $name- the assignment name, field "name" of db_assignment

* @param  string $description- field "description" of db_assignment

* @param  string $duedate-field "timedue" of db_assignment

*/
```

```
  function check4add($name,$description,$duedate)
  {
      if ($name=="") { return("No Name Supplied!"); }
      if ($description=="") { return("No Description Supplied!"); }
      if ($duedate=="") {return "No duedate Supplied!"; }
      else if (!ereg("^([0-9]{4})/([0-9]{2})/([0-9]{2})/([0-9]{2})/([0-9]{2})$",
$duedate, $parts))
          // Check the time input format
          return "The date of duedate is not a valid date in the format YYYY/MM/DD/HH/II";

      return "Looks Okay";
   }// end of check4add function

   /**
  * method to display assignment's info for editing
  * @author Ke Sun
  * @param numeric $id - the primary key of the assignment in the database
  */
  function show4edit($id)
  {   $r1=$this->tablea->getAssignments('id',$id);
      $line=array_shift($r1);
      if (!$line){
         $this->rstatus= 'Error - no this assignment';
         return'error_tpl.php';
      }else{
         $this->setvar('asdata',$line);
         return'asedit_tpl.php';
      }
  }  // end of function show4edit
```

```
    /**

    * method to check submitted info for any illegal data

    * @author Sun Ke

    */

    function check4edit()

    {   if ($this->getParam('name')=="") { return("Assignmentname may not be blank!"); }

        if ($this->getParam('description')=="") { return "No Description of assignment
Supplied!"; }

        if ($this->getParam('duedate')=="") {return "No duedate Supplied!"; }

        else if (!ereg("^([0-9]{4})/([0-9]{2})/([0-9]{2})/([0-9]{2})/([0-9]{2})$",
$this->getParam('duedate'), $parts))

                // Check the time input format

                return "The date of duedate is not a valid date in the format YYYY/MM/DD/HH/II";

            return "Looks Okay";

    }     // end of check4eddit function


    /**

    * method to update database with assingment-info. update tbl_assignment tabel.

    * @author Ke Sun

    */

    function applyedit($goaim)

    {   $dob=$this->getParam('duedate');

        $duetime=mktime(substr($dob,11,2),substr($dob,14,2),59,substr($dob,5,
2),substr($dob,8, 2),substr($dob,0, 4));

        $modtime=time(now);

        $sdata['id']=$this->getParam('id');

        /*this userid is for assignment onwer (status:lecture or administrator).it is
dbl_assignment "userid"field

        * who edit the assignment, who is onwer?

        */
    $sdata['userid']=$this->objUser->userId();
    $sdata['name']=$this->getParam('name');
    $sdata['description']=$this->getParam('description');
    $sdata['resubmit']=$this->getParam('resubmit');
    $sdata['type']=$this->getParam('type');
    $sdata['timedue']=$duetime;
    $sdata['mark']=$this->getParam('mark');
    $sdata['timemodified']=$modtime;
    $id=$this->getParam('id');
    $r1=$this->tablea->update('id',$id,$sdata);
    if (!$r1){
        $this->rstatus="Changes have not been made. Database error.";
        return'error_tpl.php';
    }else{
        switch ($goaim){
            case 'asli4lec':
                return $this->dispatch('asli4lec');
               break;
            case 'listassignment':
                return $this->dispatch('listassignment');
        }// end of switch
    }
} // end of function applyedit

/**
* method to delete assignment data
* @author Ke Sun
* @param numeric $id - the primary key of the assignment to be deleted
*/
function applydelete($id)
```

```
    {   $this->tablea->delete('id',$id);

        $this->tableb->delete('assignment',$id);

    } //end of applydelete function



      /**

    * method to submit or mark assignment(depond on var "model" value) (insert or update
db_assignment_submit)

    * @author Ke Sun

    */

    function applysubmit()

    {

        $id=$this->getParam('id');

        $subid=$this->getParam('subid');

        $mode=$this->getParam('mode');

        $courseid=$this->getParam('courseid');

        $type=$this->getParam('type');

        $sdata['courseid']= $courseid;//this userid is orignial infor of tbl_assignment.
that is teacher id

        $sdata['assignment']=$id;

        $sdata['timecreated']=$this->getParam('timecreated');

        $sdata['timemodified']=time(now);

        $sdata['numfiles']=$type;

        $sdata['online']=$this->getParam('online');

        $sdata['patch']=$this->getParam('patch');

        switch ($mode){

            // it is for resubmit

            case'update':

                // userid is for student

                $sdata['numfiles']=$type;

                $sdata['userid']=$this->objUser->userId();
    $sdata['id']=$subid;
    $r1=$this->tableb->update('id',$subid,$sdata);
    $this->dispatch('liststu');
    break;
// it is for lecture marked the submited-assignment
case'mark':
    $sdata['id']=$subid;
    $sdata['mark']=$this->getParam('mark');
    $sdata['comment']=$this->getParam('comment');
    $sdata['teacher']=$this->objUser->userId();
    $sdata['timemarked']=time(now);
    /* set the dbl_assignment_submit field "flag" value.
     * It direcate the assignment already marked */
    $sdata['flag']='yes';
    $r1=$this->tableb->update('id',$subid,$sdata);
    break;

// for the student first submit assignment
case'insert':
    $userid=$this->objUser->userId();
    // Count the history record. It is for avoid http reload.
    $hirecord=$this->tableb->counall4stu($id,$userid);
    if ($hirecord>0) {
        //return($this->dispatch('liststu'));
       return 'mainstu_tpl.php';
      $fresh='true';
    }else{
      $sdata['userid']=$userid;
      $sdata['teacher']=$this->getParam('userid');
      $sdata['id']=null;
```

```
                $r1=$this->tableb->insert($sdata);
              }

              break;
         }//end of switch
        // check the operation result
        if (!$r1){
            // For avoid http reload
            if ($fresh!='true'){
                $this->rstatus="Assignment has not been made. Database error.";
                return 'error_tpl.php';
            }
        }else{
            if ($mode=='mark'){
                 return 'mainlec_tpl.php';

            }else{
                 return 'mainstu_tpl.php';
            }
        }
    } // end of function applysubmit



    /**
    * method to display different link of assignments submit or mark
    * with aknowledgements to  Derek Keats for the frame of follow function. The ideas also
come from blog modul same name founction.
    * @author Ke Sun
    * @param string $assignmentid - the primary key of db_assignment field "id",relating
with db_assignment_submit field "assignment". it is one2multi.
```

```
    * @param string $blf - the form unit;

    * @param string $userid- the field "userid" of db_assignment. it is for onwer in
db_assignment. it is for student in db_assignment_submit.

    * @param ini $subrecord-the number of student submited assignment for particular student.
it is unmarked.

    * @param string $assignmentname- //the value of field "name" db_assigment.

    * @param string

    * @param string

    * @param string $rsub- the indicator of allow assignment resubmit or not. "1": allow
resubmit. "0": does't allow resubmit.'

    * @param ini $subrecorda- the number of marked submited assignment for particular student.

    * @param bool $aslec- the indicator of user status. "false" is student. "ture" is lecture.

    * @param ini $subrecord- the number of unmarked submited assignment for particular lecture.

    * @param ini $subrecorda- the number of marked submited assignment for particular lecture.

    */

    function putBlogStats($assignmentid, $blf, $userId, $subrecord,

        $assignmentname,$astime,$timedue,

$rsub,$subrecorda,$aslec=false,$subrelec=0,$subreleca=0)

    {

        // Create the footer . later will add more infor .

        $name = $this->objUser->fullName($userId);

        $this->$blf = &$this->newObject('layer', 'htmlelements');

        $this->$blf->id = "blog-footer";

        $this->$blf->addToStr($dateAdded . "||" . $name);

        // view submited assignment

        if (($subrecord > 0)or($subrecorda>0) ){//if existing submit assignment

            if ($aslec!=true){ //if the user isn't lecture

                $submited= $subrecord+$subrecorda;

                $location = $this->uri(array('action' => 'assubview',

                                                'module' => 'assignment',
                                              'subrecorda' =>$subrecorda,
                                              'id' => $assignmentid));
        $subview = "<br /><a href=\""
        . $location . "\">"
        . $this->objLanguage->languageText("word_submitedview")
        . ": " . $submited . "</a>";
      }
   } else {// it does't exiting submit assignment
     $subview = "<br />" . $this->objLanguage->languageText("word_submitedview")
      . ": " . $submited;
   }
   // Create an instance of icon object
   $this->objGetIcon = $this->newObject('geticon', 'htmlelements');
   $this->$blf->addToStr($subview);
   // If the assignment due time is later than now
   if  ($timedue>time(now)){
      if ($aslec==false){ // the user is not lecture; user is student
         if (($subrecorda==0)and($subrecord==0)){// it does not existing marked and
unmarked assignment;
            // add the submit link for student
            $location = $this->uri(array('action' => 'assubinsert',
                                              'module' => 'assignment',
                                              'id' => $assignmentid));
            $this->objGetIcon->setIcon("submit2", "gif");
            $location = "&nbsp;&nbsp;&nbsp;<a href=\"" . $location . "\">" .
$this->objGetIcon->show() . "</a> ";
            $this->$blf->addToStr($location);


         }else{
               if ($rsub =='1'){ // if allow resubmit
```

```
                       if ($subrecorda==0){// doesn't existing marked assignment
                          // add the submit link for student
                          $location = $this->uri(array('action' => 'assubupdate',
                                          'module' => 'assignment',
                                          'id' => $assignmentid));
                          $this->objGetIcon->setIcon("submit2", "gif");
                          $location = "&nbsp;&nbsp;&nbsp;<a href=\"" . $location . "\">" .
$this->objGetIcon->show() . "</a> ";
                          $this->$blf->addToStr($location);
                       }
                    }
                 }
          }else{// the user is lecture
             if (($subrelec+$subreleca)!=0){// existing submited assignment
                    //get the unmarked assigment

$leasdataal=$this->tableb->getsubrecordlec($assignmentid,$this->objUser->userId(),$nee
dmark=true); //getallsubrecord($blogid,$this->objUser->userId());
                    //display the unmarked assignment
                    foreach ($leasdataal as $line)
                    {
                        $studentname=$this->objUser->fullName($line['userid']);
                        // add the mark link for lecture
                        $location = $this->uri(array('action' => 'assubmark',
                                         'suasid' =>$line['id'],
                                         'id' => $assignmentid));
                        $this->objGetIcon->setIcon("comment", "gif");
                        $location = "&nbsp;&nbsp;&nbsp;<a href=\"" . $location .
"\">" .$studentname.$this->objGetIcon->show() . "</a> ";
                        $this->$blf->addToStr("<br>".$location);
```

```
                }

          }

      }

    }// end of the if (jodge the valid due time)

    return $this->$blf->addToLayer();

}

/**

* method to display the right side page for mainlect_tpl.php and mainstu_tpl.php

* the later will add more functiniality

*/

function rightframe($status='stu',$id='')

{

    $this->loadClass('form','htmlelements');

    $this->loadClass('textarea','htmlelements');

    $this->blogCont = &$this->newObject('layer', 'htmlelements');

    $objForm = new form('rightside');

    $objForm->setAction($formaction);


    $home=$this->uri(array(), '_default'); //the home URL

    $blHome=$this->uri(array(), 'assignment'); //the blog home URL

  // $action=$this->getParam('action', Null); //Get the action

    //$blogger=$this->getParam('blogger', Null); //Get the blogger

    $this->objGetIcon =&$this->newObject('geticon', 'htmlelements');

    // The add home icon with link

    $this->objGetIcon->setIcon("blog_sitehome");

    $this->objGetIcon->align = "middle";

    $this->objGetIcon->alt = $this->objLanguage->languageText("word_home");
    $navico="<a href=\"".$home."\">"
        .$this->objGetIcon->show()."</a>";
    // The add blog home icon with link
    $this->objGetIcon->setIcon("blog_home");
    //$this->objGetIcon->align = "absmiddle";
    $this->objGetIcon->alt =
is->objLanguage->languageText("mod_assignment_assignmenthome");
    $navico .= "&nbsp;&nbsp;&nbsp;<a href=\"".$blHome."\">"
        .$this->objGetIcon->show()."</a>";
    $objForm->addToForm($navico);
    switch ($status){
        // Lectures
        case 'lec':
            //Get the name info
            $dscrb=$this->tablea->getAssignments('id',$id);
            $assname=$dscrb['0']['name'];
             //Get the description info
            $description=$dscrb['0']['description'];
            if ($id=='){
               $assname='';
               $description=$assname;
            }
            //LEAR:creat leray
            $this->$blh =&$this->newObject('layer', 'htmlelements');
            $this->$blh->id = "blheadline";
            $this->$blh->addToStr($assname);
            $this->$blh->addToLayer();
            $objForm->addToForm($this->$blh->addToLayer());
           //LEAR:creat leray
            $this->$bld =&$this->newObject('layer', 'htmlelements');
```

```
          $this->$bld->id = "blog-content";
          $this->$bld->addToStr($description);
          $this->$bld->addToLayer();
          $objForm->addToForm($this->$bld->addToLayer());
          $navBar .= "<br />".$objForm->show();
          return $navBar;
          break;
      case 'stu':
          //$objForm->addToForm();
          $navBar .= "<br />".$objForm->show();
          return $navBar;
          break;

    }

}// end of rightframe function

/**
* method to display the left side page for mainlect_tpl.php and mainstu_tpl.php
* the later will add more functiniality

*         ------------------------------------------------------------*/
function leftframe($formaction=false,$courseid='',$status='stu',$markrecord='')
{
    $this->loadClass('form','htmlelements');
    $this->loadClass('textarea','htmlelements');
    $this->blogCont = &$this->newObject('layer', 'htmlelements');

    $objForm = new form('leftside');
    $objForm->setAction($formaction);
```

```
        $student = $this->uri(array('action'=>'liststu','courseid'=>$courseid),
'assignment');
        $this->objGetIcon = $this->newObject('geticon', 'htmlelements');
        // The add people icon with link
        $this->objGetIcon->setIcon("user_user");
        $this->objGetIcon->align = "";
        $this->objGetIcon->alt = $this->objLanguage->languageText("word loginasstudent");
        $asstudent="<a href=\"".$student."\">".$this->objLanguage->languageText("word
loginasstudent")
                .$this->objGetIcon->show()."</a>";

        $objDropdown = &$this->getObject('dropdown','htmlelements');
        //$this->loadClass('htmltable', 'htmlelements');
        $courseli=$this->tablec->getAll();
        $objDropdown->dropdown('courseid');
        $objDropdown->addfromdb($courseli,"title", "contextCode", "contextCode");
        $this->loadClass('button', 'htmlelements');
        $objElement2 = new button('submit');
        $objElement2->setToSubmit();
        $objElement2->setValue($this->objLanguage->languageText("word_submit"));
        // Depend on user status give different display
        switch ($status){
            // Lectures
            case 'lec':
                // add the form unit

$objForm->addToForm($asstudent."<br>".$this->objLanguage->languageText("mod assignment
choice course")."<br>"
                .$objDropdown->show()."&nbsp;&nbsp;".$objElement2->show());
                // set the link

                $aslist=$this->uri(array('action'=>'asli4lec','courseid'=>$courseid),

  assignment');

                $this->objGetIcon = $this->newObject('geticon', 'htmlelements');

                // The add admin icon with link

                $this->objGetIcon->setIcon("admin");

               //$this->objGetIcon->align = "";

                $this->objGetIcon->alt = $this->objLanguage->languageText("word manage

 assignment");

                $aslist="<a href=\"".$aslist."\">".$this->objGetIcon->show()."</a>";

                $asmange="Manager Assignment".$aslist;

                //LEAR:creat leray

                $this->$blmv =&$this->newObject('layer', 'htmlelements');

                $this->$blmv->id = "bltitle";

                $this->$blmv->addToStr($asmange);

                $this->$blmv->addToLayer();

                $objForm->addToForm($this->$blmv->addToLayer());

                 // Get the general assignment infor for Lecture

                $asitem="<font color=black>".'Assignment item'."</font>"."<br>"

                // Call display function Listassignments

                .$this->tableDsplay->

ListAssignments('courseid',$courseid,'TRUE',$flaga='mv',$goaim='',$courseid,'userid',$

this->objUser->userId());

                //LEAR:creat leray

                $this->$blh =&$this->newObject('layer', 'htmlelements');

                $this->$blh->id = "blheadline";

                $this->$blh->addToStr($asitem);

                $this->$blh->addToLayer();

                $objForm->addToForm($this->$blh->addToLayer());
```

```
                 $asmark="<font color='black'>".'Essay list'."</font>"."<br>"
                 //Call display function Listassignments
                 .$this->tableDsplay->ListAssignments($courseid,$this->objUser->userId(),
'TRUE',$flaga='mm',$goaim='');
                 //LEAR:creat leray
                 $this->$blmm =&$this->newObject('layer', 'htmlelements');
                 $this->$blmm->id = "blheadline";
                 $this->$blmm->addToStr($asmark);
                 $this->$blmm->addToLayer();
                 $objForm->addToForm($this->$blmm->addToLayer());


                 $navBar .= "<br />".$objForm->show();


                 return $navBar;
                 break;
            case 'stu':
                 // add the form unit
                 $objForm->addToForm($this->objLanguage->languageText("mod assignment
choice course")."<br>"
                 .$objDropdown->show()."&nbsp;&nbsp;".$objElement2->show());


                  // Get the general assignment infor for Lecture
                 $asitem='Assignment item'."<br>"
                  // Call display function Listassignments
                 .$this->tableDsplay->
ListAssignments('courseid',$courseid,'TRUE',$flaga='sv');
                 //LEAR:creat leray
                 $this->$blh =&$this->newObject('layer', 'htmlelements');
                 $this->$blh->id = "blheadline";
                 $this->$blh->addToStr($asitem);
```

```
                $this->$blh->addToLayer();
                $objForm->addToForm($this->$blh->addToLayer());
                $navBar .= "<br />".$objForm->show();
                return $navBar;
                break;

        }

    } // end of leftframe function

    /**
    * method to display foot unit
    * with aknowledgements to  Derek Keats for the frame of follow function. The idear also
come from blog modul same name founction.
    * @author Ke Sun
    */
    function putAsFooter($status)
    {
        $this->footerNav = &$this->newObject('layer', 'htmlelements');
        $this->footerNav->id = "blog-outside-footer";
        $this->footerNav->str =
$this->objLanguage->languageText("mod_assignment_currentuser") . ": ";
        $this->footerNav->str .= $this->objUser->fullName();
        $this->footerNav->str .= "&nbsp;&nbsp;||&nbsp;&nbsp;"
        . $this->objLanguage->languageText("mod_assignment_userstatus") . ": ";
        $this->footerNav->str .=$status ;
        $this->footerNav->str .= $asunm;
        return $this->footerNav->addToLayer();
    }
```

```
/*-----------------------------------------------------------------------------------

--------------------*/

    // end of function command_line


} // end class assignment


?>


<?


class dbarcourses  extends dbtable


{


   function init()

   {

      parent::init('tbl_context');

   }



   /**

   * method to lookup list of course for admin functions

   * @author Ke Sun

   *

   * @param string $how - the method of searching used -  coursetitle,coursecode or contentid

   * @param string $match - the pattern to match for

   */

       function getCourse($how,$match)

       {

             //$sql="select * from tbl_archivedcourses";

             $subsqla="select * from tbl_context";

             $subsqlb="order by contextcode";
```

```
            $sql=$subsqla.$subsqlb;

            if (($how=='title')||($how='contextcode'))

           {

                 if ($match=='listall') { $match=''; }

                 $sql=$subsqla." where ".$how." like '".$match."%'".$subsqlb;

           }



           $r1=$this->getArray($sql);

           return $r1;

     }  // end of function getCourse





} // end of class sqlas


?>


<?


class dbassignment extends dbtable

{


   function init()

   {

      parent::init('tbl_assignment');

   }


   /********************---------------------------------------------------------------------------

   * method to lookup list of assignments for admin functions

   * @author Ke Sun
   *                                                                          *

   * @param string $how - the method of searching used - name, description or id  *

   * @paran string $match - the pattern to match for                               *

   ******************************************************************************/

      function getAssignments($how,$match)

      {

            //$sql="select  from tbl_assignment";

            $subsqla="select * from tbl_assignment";

            $subsqlb="order by id";

            $sql=$subsqla.$subsqlb;

            if
(($how=='name')||($how=='description')||($how='id')||($how='courseid')||($how='userid'
))

          {

                if ($match=='listall') { $match=''; }

                $sql=$subsqla." where ".$how." like '".$match."%'".$subsqlb;

          }


          $r1=$this->getArray($sql);

          return $r1;

    }  // end of function getAssignment




/*******************************************************************************

  * method to lookup accurate record  of assignments for admin functions          *

  * @author Ke Sun

  *

  * @param string $how - the method of searching used - name, description or id    *

  * @paran string $match - the pattern to match for                               *
```

```
*****************************************************************************************/

        function getAsAcc($how,$match)

        {

            //$sql="select * from tbl_users";

            $subsqla="select * from tbl_assignment";

            $subsqlb="order by id";




                $sql=$subsqla." where ".$how."= '".$match."'".$subsqlb;




           $r1=$this->getArray($sql);

           return $r1;

       } // end of function getAsAcc





/****************************************************************************************

**

   * method to lookup accurate record  of assignments for admin functions                 *

   * @author Ke Sun                                                                  *

   *                                                                              *

   * @param string $how,$how2 - the method of searching used - name, description or id    *

   * @paran string $match,$match2 - the pattern to match for                             *


*****************************************************************************************

**/

       function getAsAcc2($how,$match,$how2,$match2)

       {

           $subsqla="select * from tbl_assignment";
```

```
                $subsqlb="order by id";
                        $sql=$subsqla." where (".$how."=
'".$match."')and(".$how2."='".$match2."')".$subsqlb;
               $r1=$this->getArray($sql);
               return $r1;
        } // end of function getAsAcc2

        function getAsAcc2e($how,$match,$how2,$match2)
        {
                $subsqla="select * from tbl_assignment";
                $subsqlb="order by id";
                        $sql=$subsqla." where (".$how."=
'".$match."')and(".$how2."='".$match2."')and(type!='0')".$subsqlb;
               $r1=$this->getArray($sql);
               return $r1;
        } // end of function getAsAcc2


} // end of class sqlas

?>

<?

class dbassignmentsub extends dbtable
{
   function init()
   {
      parent::init('tbl_assignment_submit');
   }
```

```
/*******************************************************************************
    * method to lookup list of subassignments for admin functions                *
    * @author Ke Sun                                                        *
    *                                                                     *
    * @param string $how - the method of searching used - name, description or id  *
    * @paran string $match - the pattern to match for                            *

*******************************************************************************/
        function getsubAssig($how,$match)
        {
              $subsqla="select * from tbl_assignment_submit";
              $subsqlb="order by id";
              $sql=$subsqla.$subsqlb;
              if
(($how=='assignment')||($how=='teacher')||($how='id')||($how='courseid')||($how='useri
d'))
             {
                    if ($match=='listall') { $match=''; }
                    $sql=$subsqla." where ".$how." like '".$match."%'".$subsqlb;
             }

             $r1=$this->getArray($sql);
             return $r1;
        } // end of function getsubAssig


/*******************************************************************************
   * Get the number of submited for the particular assignmentid and userid;      *
   *

   *@param string  @assignmentId- the  value of field "assignment"                    *

   *@param string  @userID- the  the value of field "userId"                          *

   *@param boolean $needmark- default true count for unmark submit assignment record;  *

   *false count for marked submit assignment record;                                  *


***********************************************************************************/

   function ascount ($assignmentId='',$userId='',$needmark=true)

   {

      $sql="SELECT COUNT(assignment) AS ascount FROM tbl_assignment_submit WHERE
assignment='".$assignmentId."'";


      if (($userId!='') and ($needmark==true))

         $sql="SELECT COUNT(assignment) AS ascount from tbl_assignment_submit where
(assignment='".$assignmentId."') and (userid='".$userId."')and (flag ='no')";

      if (($userId!='') and ($needmark==false))

         $sql="SELECT COUNT(assignment) AS ascount from tbl_assignment_submit where
(assignment='".$assignmentId."') and (userid='".$userId."') and (flag ='yes')";

      $rs=$this->query($sql);

      if (!$rs) {

         return 0;

      } else {

         $line = $rs->fetchRow();

         return $line['ascount'];

      }


   } //end of ascount function


   /** ***********************************************************************

   * Get the number of submited for the particular assignmentid and userid;  *
```

```
    * It is include umark and marked assignment for student              *
    *                                                             *
    *@param string @assignmentId                                     *
    *@param string @userId                                           *
    *************************************************************************/
    function counAll4Stu ($assignmentid,$userid)
    {
        $sql="SELECT COUNT(assignment) AS ascount from tbl_assignment_submit where
(assignment='".$assignmentid."') and (userid='".$userid."')";
        $rs=$this->query($sql);
        if (!$rs) {
            return 0;
        } else {
            $line = $rs->fetchRow();
            return $line['ascount'];
        }

    } //end of counall4stu function



/*****************************************************************************
******
    * Get the info of submited for the particuler assignmentid and userid, that is for sutdnet
*
    * It is include umark and marked assignment for student                          *
    *                                                                            *
    *@param string @assignmentId                                                  *
    *@param string @userId                                                        *

*****************************************************************************
```

```
*******/
    function recoall4stu ($assignmentid,$userid)
    {

        $sql="SELECT * from tbl_assignment_submit where (assignment='".$assignmentid."')
and (userid='".$userid."')";

            $rl=$this->getArray($sql);
            return $rl;

    } //end of recoall4stu function


/*******************************************************************************
***********************
    * Get the number of submited for the particular assignmentid and userid;
*
    *
*
    *@param  boolen $needmark- default true count for unmark submit assignment record;
*
    *false count for marked submit assignment record;
*

*******************************************************************************
***********************/
   function ascountlec ($assignmentid,$userid,$needmark=true)
   {
       $sql="SELECT COUNT(assignment) AS ascount FROM tbl_assignment_submit WHERE
assignment='".$assignmentid."'";
```

```
      if (($userid!='') and ($needmark==true))

          $sql="SELECT COUNT(assignment) AS ascount from tbl_assignment_submit where
(assignment='".$assignmentid."') and (teacher='".$userid."')and (flag ='no')";

      if (($userid!='') and ($needmark==false))

          $sql="SELECT COUNT(assignment) AS ascount from tbl_assignment_submit where
(assignment='".$assignmentid."') and (teacher='".$userid."') and (flag ='yes')";

      $rs=$this->query($sql);

      if (!$rs) {

          return 0;

      } else {

          $line = $rs->fetchRow();

          return $line['ascount'];

      }


   } //end of ascount function



/**************************************************************************************
***********************

   * method to lookup list of submitassignments

*

   * @author Ke Sun

*

   *

*

   *

*

   *@param  boolen $needmark- default true count for unmark submit assignment record;

*

   false count for marked submit assignment record;
```

```
*

************************************************************************************

************************/

        function getsubrecord($assignmentid,$userid,$needmark=true)

        {

                $sql="SELECT * FROM tbl_assignment_submit WHERE
assignment='".$assignmentid."'";

                if (($userid!='') and ($needmark==true))

                   $sql="SELECT * from tbl_assignment_submit where
(assignment='".$assignmentid."') and (userid='".$userid."')and (flag ='no')";

                if (($userid!='') and ($needmark==false))

                   $sql="SELECT * from tbl_assignment_submit where
(assignment='".$assignmentid."') and (userid='".$userid."') and (flag ='yes')";


                $r1=$this->getArray($sql);

                return $r1;

       } // end of function getsubrecord




    /**

    * method to lookup list of submitassignments

    * @author Ke Sun

    *

    *

    *@param  boolen $needmark- default true count for unmark submit assignment record: false
count for marked submit assignment record;

    */
```

```
        function getsubrecordlec($assignmentid,$userid,$needmark=true)
        {
                $sql="SELECT * FROM tbl_assignment_submit WHERE
assignment='".$assignmentid."'";
                if (($userid!='') and ($needmark==true))
                    $sql="SELECT * from tbl_assignment_submit where
(assignment='".$assignmentid."') and (teacher='".$userid."')and (flag ='no')";
                if (($userid!='') and ($needmark==false))
                    $sql="SELECT * from tbl_assignment_submit where
(assignment='".$assignmentid."') and (teacher='".$userid."') and (flag ='yes')";

                $r1=$this->getArray($sql);
                return $r1;
        } // end of function getsubrecord



     /**
    * method to lookup list of submitassignments
    * @author Ke Sun
    *
    *
    *@param  boolen $needmark- default true count for unmark submit assignment record; false
count for marked submit assignment record;
    */

        function getallsubrecord($assignmentid,$teacherid)
        {
              $sql="SELECT * FROM tbl_assignment_submit WHERE
assignment='".$assignmentid."'";
```

```
            if (($teacherid!=''))

                $sql="SELECT * from tbl_assignment_submit where
(assignment='".$assignmentid."') and (teacher='".$teacherid."')";



            $r1=$this->getArray($sql);

            return $r1;

        } // end of function getsubrecord


     /**

    * method to lookup list of submitassignments

    * @author Ke Sun

    *

    *

    *@param  boolen $needmark- default true count for unmark submit assignment record; false
count for marked submit assignment record;

    */


        function getallas4lec($coursied,$teacherid)

        {

            $sql="SELECT * FROM tbl_assignment_submit WHERE courseid='".$courseid."'";

            if (($teacherid!=''))

               $sql="SELECT * from tbl_assignment_submit where
(courseid='".$courseid."') and (teacher='".$teacherid."') order by assignment and flag";



            $r1=$this->getArray($sql);

            return $r1;

        } // end of function getsubrecord
} // end of class sqsbes

?>

<?

/* -------------------- tabledisplay_class.php ----------------*/

// security check - must be included in all scripts

if (!$GLOBALS['kewl_entry_point_run'])

{

   die("You cannot view this page directly");

}

// end security check




class tabledisplay extends controller

{

   var $objButtons;

   var $dropdown;

   var $table;

   var $tablea;

   var $objLanguage;

   var $tblgroups;

   var $objUser;

   var $isAdmin;

   var $goaim; // shows the aim webpage for after delete action;

   var $rstatus; // shows whether a function-call did what was wanted or not
```

```
    var $rvalue;    // the return-value for the template to be used.


    function init()
    {
        //$this->objButtons=&$this->getObject('buttons','display');
        $this->objUser = &$this->getObject('user', 'security');
        $this->loadClass('htmltable', 'htmlelements');
        $this->objLanguage = &$this->getObject('language', 'language');
        $this->tablea=&$this->getObject('dbassignment','assignment');
        $this->tableb=&$this->getObject('dbassignmentsub','assignment');
        $this->tablec=&$this->getObject('dbarcourses','assignment');
        //Create an instance of the trimstring object to trim theoutput
        $this->objTrimStr = &$this->getObject('trimstr', 'strings');
        $this->isAdmin=$this->objUser->isAdmin();
    }


    /**
    * method to display list of assignments for admin functions
    * @author Ke Sun
    * @param string $how - the method of searching used - id, name, description or courseid
    * @param string $match - the pattern to match for
    * @param bool $adminLinks - whether to display the Add, Edit and Delete links
    * @parem string $flaga- the indication of the users staus
'a'-admindisrator,'t'-lecture,'s'-student
    * 'mv' for lecturer admin assignment item, 'mm' for lecturer mark assignment.
    *'sv'-the general assignment for student.
    * @param string $goaim- the indication of the webpage for after confirm action(confirm
delete,confirm add,confir edit...);
    */
   function
```

```
ListAssignments($how,$match,$adminLinks,$flaga='a',$goaim='',$courseid='',$how2='',$ma
tch2='')
  {
            $tblclass=$this->newObject('htmltable','htmlelements');
            $data="<div>\n<br /><table border=0 align='center'><tr>\n";
            switch ($flaga){
                case 'a':
                case 't':
                case 's':
                case 'mv':
                case 'sv':
                case 'es4lec':
                  $fieldnames=array('name');
                  break;

                case 'mm':
                  $fieldnames=array('userid');
                  break;


            }
            foreach($fieldnames as $field)
            {
                $data.="<td
class='heading'>".$this->objLanguage->languageText(strtolower($field),$field)."</td>\n
";
                switch ($flaga){
                   case 'a':
                   case 't':
                   case 's':
                   case 'mv':
```

```
                              case 'sv':
                              case 'es4lec':
                                 $field2[]='Assignment Name';
                                  break;

                             case 'mm':
                               $field2[]='USER ID';
                             break;
                        }

                  }
                      // Depend on user status to display items of form
table('a':administrator,'t':lecture,'s':student)
                      switch ($flaga){
                          case 'a':
                              $field2[]='Description';
                              $field2[]='CourseName';
                              $field2[]='TeacherName';
                              $field2[]='AssignmentType';
                              $field2[]='DueTime';
                              $field2[]='AssignTime';
                              $field2[]='AllowResubmit';
                              break;
                           case 't':
                              $field2[]='Description';
                              $field2[]='teachername';
                              $field2[]='assignmenttype';
                              $field2[]='duetime';
                              $field2[]='assigntime';
                              $field2[]='allowresubmit';
```

```
                     switch ($flaga)

                     {


                       case 'a':


$addlink=$this->uri(array('module'=>'assignment','action'=>'Add'),'assignment');

                         $this->objGetIcon = $this->newObject('geticon', 'htmlelements');

                         // The add admin icon with link

                         $this->objGetIcon->setIcon("add");

                         //$this->objGetIcon->align = "";

                         $this->objGetIcon->alt = $this->objLanguage->languageText("word

add");

                         $addbu="<a

href=\"".$addlink."\">".$this->objGetIcon->show()."</a>";

                         $data.="<td>".$addbu."</td>";

                         $field2[]= $addbu;

                         break;

                      // the teacher add link: it is limited in individual course:

                      case 't':


$addlink=$this->uri(array('module'=>'assignment','action'=>'addt','courseid'=>$coursei

d,'goaim'=>'asli4lec'),'assignment');

                         $this->objGetIcon = $this->newObject('geticon', 'htmlelements');

                         // The add admin icon with link

                         $this->objGetIcon->setIcon("add");

                         //$this->objGetIcon->align = "";

                         $this->objGetIcon->alt = $this->objLanguage->languageText("word

add");

                         $addbu="<a

href=\"".$addlink."\">".$this->objGetIcon->show()."</a>";
```

```
                    $data.="<td>".$addbu."</td>";

                    $field2[]= $addbu;


                    break;

                  // It does't give student add assignment link

                  case 's':


$field2[]=$this->objLanguage->languageText("assignmenchoicefunction_function");

                    break;

                }// end of switch


            }

            $data.="</tr>\n";

            $tblclass->width='';

            $tblclass->attributes=" align='center' border=0";

            $tblclass->cellspacing='2';

            $tblclass->cellpadding='2';

            $tblclass->addHeader($field2,'odd');

            unset($field2);

            switch ($flaga)

            {

             case 'a':

               $rl=$this->tablea->getAssignments($show,$match); // Table-derived

functions called here.

               break;

             case 't':

               $rl=$this->tablea->getAsAcc($show,$match);// to research a special

assignment for student

               break;

             case 'mv':
                $r1=$this->tablea->getAsAcc2($how,$match,$how2,$match2);// to research
a special assignment for lecture
                break;
               case 's':
                $r1=$this->tablea->getAsAcc($how,$match);// to research a special
assignments for student
                break;
               case 'mm':
                $r1=$this->tableb->getsubAssig($how,$match);// get all submited
assignments for lecture in particular course
                break;
               case 'sv':
                $r1=$this->tablea->getAsAcc($how,$match);// get all submited assignments
for lecturer in particular course
                break;
               case 'es4lec':
                $r1=$this->tableb->getalles4lec($how,$match);//get all submited essaies
for lecturer in particualr course
               break;
              }//end of switch


              $rowcount='';
              foreach ($r1 as $line)
              {
                    $rowcount=($rowcount==0) ? 1 : 0; // with aknowledgements to Derek Keats
for this idea
                    $oddOrEven=($rowcount==0) ? "odd" : "even";
                    //$oddOrEven="odd";
                    $data.="<tr>";
                    //$data.="<td
```

```
class='".$oddOrEven."'>".$line['emailAddress']."</td>\n";
                    foreach($fieldnames as $field)
                          {       // $this->objUser->fullName($line['userid']
                             switch ($flaga){
                                 case 'a':
                                 case 't':
                                 case 's':
                                 case 'mv':
                                 case 'mm':
                                 case 'sv':
                                   $data.="<td
class='".$oddOrEven."'>".$line[$field]."</td>\n";
                                   $dline[]=$line[$field];
                                  break;


                              }


                          }
                    if ($adminLinks=='TRUE'){
                         // It is depend on user status give different display items
                         switch ($flaga)
                         {
                              case 'a':
                                  // for trim the description;
                                  $data.="<td class='".$oddOrEven."'>";

$d1=$this->objTrimStr->strTrim(stripslashes($line['description']), 15);
                                  $dline[]=$d1;
                                  $data.="</td>";
                                  // for show the coursetitle
```

```
$datum=$this->tablec->getCourse('contextCode',$line['courseid']);

                                        $data.="<td class='".$oddOrEven."'>";

                                        $d1=$datum['0']['title'];

                                        if ($d1=='')

                                            $d1='null';

                                        $dline[]=$d1;

                                        $data.="</td>";

                                        // for show the assignment lecture

                                        $data.="<td class='".$oddOrEven."'>";

                                        $d1=$this->objUser->fullName($line['userid']);

                                        if ($d1=='')

                                           $d1='null';

                                        $dline[]=$d1;

                                        $data.="</td>";

                                        // for show the assignmenttype

                                        $data.="<td class='".$oddOrEven."'>";

                                         if ($line['type']=='0')


$d1=$this->objLanguage->languageText("assignmenttype_offlineactivity");

                                         if ($line['type']=='1')


$d1=$this->objLanguage->languageText("assignmenttype_uploadfile");

                                        $dline[]=$d1;

                                        $data.="</td>";

                                        // the follow is for change timestamp to normal date

formate

                                        $data.="<td class='".$oddOrEven."'>";

                                        $d1= date('Y/m/d/H/i',$line['timedue']);

                                        $dline[]=$d1;
```

```
                                    $data.="</td>";
                                    $data.="<td class='".$oddOrEven."'>";
                                    $data.="<td class='".$oddOrEven."'>";
                                    $d1= date('Y/m/d/H/i',$line['timemodified']);
                                    $dline[]=$d1;
                                    $data.="</td>";
                                    // to show it is allow resubmit assignment or not
                                    $data.="<td class='".$oddOrEven."'>";
                                    if ($line['resubmit']=='0')
                                       $d1=$this->objLanguage->LanguageText("word_no");
                                    if ($line['resubmit']=='1')
                                       $d1=$this->objLanguage->LanguageText("word_yes");
                                    $dline[]=$d1;
                                    $data.="</td>";
                                    // the follow is for the edit and delete action link

                                    $this->objGetIcon = $this->newObject('geticon',
'htmlelements');
                                    $data.="<td class='".$oddOrEven."'>";

$editLink=$this->uri(array('module'=>'assignment','action'=>'Edit','id'=>$line['id'],'
goaim'=>$goaim),'assignment');

$deleteLink=$this->uri(array('module'=>'assignment','action'=>'Delete','id'=>$line['id
'],'goaim'=>$goaim),'assignment');
                                    $this->objGetIcon->setIcon("edit");
                                    $this->objGetIcon->alt =
$this->objLanguage->languageText("word edit");
                                    $editbu="<a
href=\"".$editLink."\">".$this->objGetIcon->show()."</a>";
```

```
                                  $this->objGetIcon->setIcon("delete");
                                  $this->objGetIcon->alt =
$this->objLanguage->languageText("word delete");
                                  $deletebu="<a
href=\"".$deleteLink."\">".$this->objGetIcon->show()."</a>";
                                  $d1=$editbu.$deletebu;
                                  $dline[]=$d1;
                                  $data.="</td>";
                                  break;
                              case 't':
                                  // for trim the description;
                                  $data.="<td class='".$oddOrEven."'>";

$d1=$this->objTrimStr->strTrim(stripslashes($line['description']), 15);
                                  $dline[]=$d1;
                                  $data.="</td>";
                                  // for show the assignment lecture
                                  $data.="<td class='".$oddOrEven."'>";
                                  $d1=$this->objUser->fullName($line['userid']);
                                  if ($d1=='')
                                    $d1='null';
                                  $dline[]=$d1;
                                  $data.="</td>";
                                  // for show the assignmenttype
                                  $data.="<td class='".$oddOrEven."'>";
                                   if ($line['type']=='0')

$d1=$this->objLanguage->languageText("assignmenttype_offlineactivity");
                                   if ($line['type']=='1')
```

```
$d1=$this->objLanguage->languageText("assignmenttype_uploadfile");

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   // the follow is for change timestamp to normal date

formate

                                   $data.="<td class='".$oddOrEven."'>";

                                   $d1= date('d/m/Y',$line['timedue']);

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   $data.="<td class='".$oddOrEven."'>";

                                   $d1= date('d/m/Y',$line['timemodified']);

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   // to show it is allow resubmit assignment or not

                                   $data.="<td class='".$oddOrEven."'>";

                                   if ($line['resubmit']=='0')

                                      $d1=$this->objLanguage->LanguageText("word_no");

                                   if ($line['resubmit']=='1')

                                      $d1=$this->objLanguage->LanguageText("word_yes");

                                   $dline[]=$d1;

                                   $data.="</td>";


                                   //if the lecture is assignment onwer give the edit and

delete right;

                                   if ($line['userid'] == $this->objUser->userId()){

                                      $this->objGetIcon = $this->newObject('geticon',

'htmlelements');

                                      $data.="<td class='".$oddOrEven."'>";


$editLink=$this->uri(array('module'=>'assignment','action'=>'Edit','id'=>$line['id'],'
goaim'=>'asli4lec','courseid'=>$line['courseid']),'assignment');


$deleteLink=$this->uri(array('module'=>'assignment','action'=>'Delete','id'=>$line['id
'],'goaim'=>'asli4lec','courseid'=>$line['courseid']),'assignment');
                                    $this->objGetIcon->setIcon("edit");
                                    $this->objGetIcon->alt =
$this->objLanguage->languageText("word edit");
                                    $editbu="<a
href=\"".$editLink."\">".$this->objGetIcon->show()."</a>";
                                    $this->objGetIcon->setIcon("delete");
                                    $this->objGetIcon->alt =
$this->objLanguage->languageText("word delete");
                                    $deletebu="<a
href=\"".$deleteLink."\">".$this->objGetIcon->show()."</a>";
                                    $dl=$editbu.$deletebu;
                                    $dline[]=$dl;
                                    $data.="</td>";
                                  }
                                  break;
                              case 's':
                                  // for trim the description;
                                  $data.="<td class='".$oddOrEven."'>";

$dl=$this->objTrimStr->strTrim(stripslashes($line['description']), 15);
                                  $dline[]=$dl;
                                  $data.="</td>";
                                  // for show the assignment lecture
                                  $data.="<td class='".$oddOrEven."'>";
                                  $dl=$this->objUser->fullName($line['userid']);
                                  if ($dl=='')
```

```
                                   $d1='null';

                                 $dline[]=$d1;

                                 $data.="</td>";

                                 // for show the assignmenttype

                                 $data.="<td class='".$oddOrEven."'>";

                                  if ($line['type']=='0')


$d1=$this->objLanguage->languageText("assignmenttype_offlineactivity");

                                  if ($line['type']=='1')


$d1=$this->objLanguage->languageText("assignmenttype_uploadfile");

                                $dline[]=$d1;

                                $data.="</td>";

                                // the follow is for change timestamp to normal date

formate

                                $data.="<td class='".$oddOrEven."'>";

                                $d1= date('d/m/Y',$line['timedue']);

                                $dline[]=$d1;

                                $data.="</td>";

                                $data.="<td class='".$oddOrEven."'>";

                                $d1= date('d/m/Y',$line['timemodified']);

                                $dline[]=$d1;

                                $data.="</td>";

                                // to show it is allow resubmit assignment or not

                                $data.="<td class='".$oddOrEven."'>";

                                if ($line['resubmit']=='0')

                                   $d1=$this->objLanguage->LanguageText("word_no");

                                if ($line['resubmit']=='1')

                                   $d1=$this->objLanguage->LanguageText("word_yes");

                                $dline[]=$d1;
```

```
                                        $data.="</td>";

                                        // give the student submit link

                                        $data.="<td class='".$oddOrEven."'>";


$editLink="index.php?module=assignment&action=Assub&id=".$line['id'];


$data.=$this->objButtons->linkedButton("edit",$editLink);

                                        $d1=$this->objButtons->linkedButton("edit",$editLink);

                                        $dline[]=$d1;

                                        $data.="</td>";

                                        break;

                                case 'mv':

                                        $data.="<td class='".$oddOrEven."'>";

                                        $d1= date('Y/m/d',$line['timedue']);

                                        $dline[]=$d1;

                                        $data.="</td>";


                                         // set the link


$aslist=$this->uri(array('action'=>'asli4lem','id'=>$line['id']), 'assignment');

                                        $this->objGetIcon = $this->newObject('geticon',

'htmlelements');

                                        // The add admin icon with link

                                        $this->objGetIcon->setIcon("info");

                                        //$this->objGetIcon->align = "";

                                        $this->objGetIcon->alt =

$this->objLanguage->languageText("word view assignment");

                                        $aslist="<a

href=\"".$aslist."\">".$this->objGetIcon->show();

                                        $data.="<td class='".$oddOrEven."'>";
```

```
                                  $d1=$this->tableb->ascount($line['id']);

                                  $dline[]=$d1."&nbsp&nbsp&nbsp&nbsp".$aslist;

                                  $data.="</td>";

                                  break;

                          case 'mm':

                                   // get student name

                                   $data.="<td class='".$oddOrEven."'>";

                                   $d1=$this->objUser->fullName($line['userid']);

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   // Get the assignment name for display


$datum=$this->tablea->getAssignments('id',$line['assignment']);

                                   $data.="<td class='".$oddOrEven."'>";

                                   $d1=$datum['0']['name'];

                                   if ($d1=='')

                                      $d1='null';

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   //get the submited time

                                   $data.="<td class='".$oddOrEven."'>";

                                   $d1= date('Y/m/d/H/i',$line['timemodified']);

                                   $dline[]=$d1;

                                   $data.="</td>";

                                   // Get the mark for display

                                   $data.="<td class='".$oddOrEven."'>";

                                   if ($line['flag']=='no'){

                                     $d1="<font color=black>".'UNMARKED'."</font>";

                                   }else{

                                     $d1="<font color=red>".$line['mark']."</font>";
                                   }
                                   $dline[]=$d1;
                                   $data.="</td>";
                                   // Get the marktime for display
                                   $data.="<td class='".$oddOrEven."'>";
                                   if ($line['flag']=='no'){
                                      $d1="----";
                                   }else{
                                      $d1=date('Y/m/d/H/i',$line['timemarked']);
                                   }
                                   $dline[]=$d1;
                                   $data.="</td>";
                                   //set the link

$aslist=$this->uri(array('action'=>'assubmark','id'=>$line['assignment'],'suasid'
=>$line['id']), 'assignment');
                                   $this->objGetIcon = $this->newObject('geticon',
'htmlelements');
                                   // The add admin icon with link
                                   $this->objGetIcon->setIcon("comment");
                                   //$this->objGetIcon->align = "";
                                   $this->objGetIcon->alt =
$this->objLanguage->languageText("word manage assignment");
                                   $aslist="<a
href=\"".$aslist."\">".$this->objGetIcon->show()."</a>";
                                   $data.="<td class='".$oddOrEven."'>";
                                   $d1="<font color=red>".$line['flag']."</font>";

$d1=$d1."&nbsp&nbsp".$aslist;//$this->objButtons->linkedButton("edit",$editLink);
```

```
                                        $dline[]=$d1;

                                        $data.="</td>";



                                       break;

                                   case 'sv':

                                        // for get submitted info for the student;


$subco=$this->tableb->counall4stu($line['id'],$this->objUser->userId());


$subinfo=$this->tableb->recoall4stu($line['id'],$this->objUser->userId());

                                        // get the submit location (insert);

                                        $locIns=$this->uri(array('action' => 'assubinsert',

                                                              'module' => 'assignment',

                                                              'id' => $line['id']));

                                        $this->objGetIcon= &$this->newObject('geticon',

'htmlelements');

                                        // The add comment icon with link

                                        $this->objGetIcon->setIcon("submit2");

                                        $this->objGetIcon->align = "";

                                        $this->objGetIcon->alt =

$this->objLanguage->languageText("word assignment submit");

                                        // Add location to icon (insert);

                                        $subIns="<a

href=\"".$locIns."\">".$this->objGetIcon->show()."</a>";



                                         // add the submit link for student (update)

                                        $locUpd=$this->uri(array('action' => 'assubupdate',

                                                               'module' => 'assignment',

                                                               'id' => $line['id']));

                                        $subUpd="<a
```

```
href=\"".$locUpd."\">".$this->objGetIcon->show() . "</a> ";
                                  // Count the number of marked submit assignment by
current user

$subrecorda=$this->tableb->ascount($line['id'],$this->objUser->userId(),false);//+$sub
record;
                                  //add the assignment link for student
                                  $locview = $this->uri(array('action' => 'assubview',
                                         'module' => 'assignment',
                                         'subrecorda' =>$subrecorda,
                                         'id' => $line['id']));
                                  // add the bookopen icon to link.
                                  $this->objGetIcon->setIcon("bookopen");
                                  $this->objGetIcon->align = "";
                                  $this->objGetIcon->alt =
$this->objLanguage->languageText("word  submitted view");
                                  $subview ="&nbsp". "<a href=\""
                                           . $locview . "\">"
                                           .$this->objGetIcon->show(). "</a>";
                                  // get the submitted info
                                  if ($subco>0){
                                     $dsub="YES";
                                     if (($line['resubmit']==1) and
($line['timedue']>time(now))and($subinfo['0']['flag']=='no')){
                                     $dsub="YES"."&nbsp&nbsp&nbsp&nbsp".$subUpd;
                                     }
                                  }else{
                                     if ($line['timedue']>time(now)){
                                       $dsub="NO"."&nbsp&nbsp&nbsp&nbsp".$subIns;
                                     }else{
                                       $dsub="LOST";
                                     }
                                  }
                                  // get the mark info
                                  if ($subinfo['0']['flag']=='yes'){
                                     $dmark="<font
color=red>".$subinfo['0']['mark']."</font>";
                                  }else{
                                     $dmark='---';
                                  }
                                  $data.="<td class='".$oddOrEven."'>";
                                  $dline[]=$dsub;
                                  $data.="</td>";
                                  $data.="<td class='".$oddOrEven."'>";
                                  $dline[]=$dmark.$subview;
                                  $data.="</td>";
                                  break;
                        }//end of switch */
                    }
                    $data.="</tr>\n";
                    $tblclass->addRow($dline,$oddOrEven);
                    unset($dline);
             )
             $data.="</table>\n</div>\n";
             return $tblclass->show();
             return $data;
   } // end of function ListAssignments


} // end class assignment
```

```
?>

<?php /*************** mainadmin_tpl.php ************************
  * It is the main operation page for student                     *
  ***************************************************************/



   // LAYER: Create the layer for the blog
   $this->blogCont = &$this->newObject('layer', 'htmlelements');
   $this->blogCont->id = "";
   $count=0;
   $id=$this->getParam('id');

   $courseid=$this->getParam('courseid');

   $how='courseid';
   $match=$courseid;
   $asdata=$this->tablea->getAsAcc($how,$match);
   $asdataa=$asdata['0']['courseid'];
   $datum=$this->tablec->getCourse('contentId',$asdataa);


/*-------------------------- BEGIN Main content LAYER --------------------*/

   $this->objDbBlog = &$this->getObject('dbblog','blog');
   $this->objBlogComment = &$this->getObject('blogComment','blog');
   //$userid=$this->objUser->userId();
```

```
    //Create an instance of the htmlHeading object
    $this->objHeading = &$this->newObject('htmlheading', 'htmlelements');
    //Add the heading text to the heading object, defaults to H3
    $this->objHeading->str =
$this->objLanguage->languageText("mod_coursename:")."<br>".stripslashes($datum['0']['c
ourseTitle']); // Does it need translation?
    //Add the heading to the content string
    $this->blogCont->addToStr("<br>".$this->objHeading->show());
    //Create an instance of the URL object to link URLs in the string
    $this->objUrl = &$this->getObject('url', 'strings');
    //Create an instance of the trimstring object to trim theoutput
    $this->objTrimStr = &$this->getObject('trimstr', 'strings');
    //Create an instance of the highlight object
    $this->objHilight = &$this->getObject('highlight', 'strings');
    //Determine if the content string should be trimmed by looking at the querystring
    $trimStr=$this->getParam('trimstr', 'true');
    //Create an instance of the link object
    $this->objLink = &$this->newObject('link', 'htmlelements');
    //loop through the array of blogs returned and



    foreach ($asdata as $line) {
        $count=$count+1;
        ///$blt="blogTitle_".$count; //for the title
        ///$blh="blogHeadline_".$count;
        ///$bld="blogHeadline_".$count;
        ///$blc="blogContent_".$count; //for the content

        ///$blf="blogFooter_".$count; //for the footer                              */


        $assignmentid=$line['id']; //assignment id from recordset

        $subrecord=$this->tableb->ascount($line['id'],$this->objUser->userId(),true);

//count the number of unmark submit assignment by current user


$subrecorda=$this->tableb->ascount($line['id'],$this->objUser->userId(),false);// count

the number of marked submit assignment by current user


$assignmentName=$this->objLanguage->languageText("mod_assignment_assignmentname:")."<b

r>".stripslashes($line['name']); //title from recordset

        //Set up the link URL


        $location=$this->uri(array('action'=>'liststu',

                                    'courseid'=>$line['courseid'],'id'=>$line['id']),

'assignment');


/*------------------------later, URL will be change if $needtrim equal ture the URL is enable

else is unable----------------------------------------*/

       /*

        * Create an instance of the icon and use it to display the

        * bullet icon, linked to the item to be viewed.

        */

        $this->objVwIcon = $this->newObject('geticon', 'htmlelements');

        $this->objVwIcon->setIcon("bullet", "gif");

        $icon=$this->objVwIcon->show();

        //This need to use HTML here is something that needs looking at

        //At the very least, we should have an anchor object

        $this->objLink->href=$location;
        $this->objLink->link=$icon;
        $icon = "&nbsp;&nbsp;&nbsp;".$this->objLink->show();
        //trim  the assignment description
        $description=$this->objTrimStr->strTrim(stripslashes($line['description']), 100);
        $description = $this->objUrl->makeClickableLinks($description);
        // Add the bullet icon to the string
        $description .= $icon;
        // for show the assignmenttype
        if ($line['type']=='0'){

$type=$this->objLanguage->languageText("mod_assignment_type:")."<br>".$this->objLangua
ge->languageText("assignmenttype_offlineactivity");
        }else{ $type=$this->objLanguage->languageText("mod_assignment_type:")."<br>".$t
his->objLanguage->languageText("assignmenttype_uploadfile");
        }
        // due time of assignment
        $timedue= date('Y/m/d/H/i',$line['timedue']);

        if($line['resubmit']=='0'){

$rsub=$this->objLanguage->languageText("mod_assignment_allowresubmit:")."<br>".$this->
objLanguage->LanguageText("word_no");
        }else{$rsub=$this->objLanguage->languageText("mod_assignment_alloresubmit:")."<
br>".$this->objLanguage->LanguageText("word_yes");
        }


$outline=$this->objLanguage->languageText("mod_assignment_duetime:")."<br>"."<font
color='red'>".$timedue."</font>";
```

```
        // get maximal mark of assignment;

        $mark= $this->objLanguage->languageText("mod_assignment

maximalmark:")."<br>".$line['mark'];

        $type.="<br>".$rsub;

        $type.="<br>".$mark;

        // Get the teacherid for the foot-infor. current userid value is teacherid.

        $userId = $line['userid'];


        //LAYER: Create the layer for the title from the current record

        $this->$blt = $this->newObject('layer', 'htmlelements');

        $this->$blt->id = "bltitle";

        $this->$blt->addToStr($assignmentName);

        $this->blogCont->addToStr($this->$blt->addToLayer());


        //LAYER: Create the layer for the headline from the courrent record

        $this->$blh = $this->newObject('layer', 'htmlelements');

        $this->$blh->id = "blheadline";

        $this->$blh->addToStr($outline);

        $this->blogCont->addToStr($this->$blh->addToLayer());


         //LAYER: Create the layer for the headline from the courrent record

        $this->$blh = $this->newObject('layer', 'htmlelements');

        $this->$blh->id = "bltype";

        $this->$blh->addToStr($type);

        $this->blogCont->addToStr($this->$blh->addToLayer());




        //LAYER: Create the layer for the content from the courrent record

        $this->$blc = $this->newObject('layer', 'htmlelements');
        $this->$blc->id = "blog-content";

        $this->$blc->addToStr($description);

        $this->blogCont->addToStr($this->$blc->addToLayer());


        // get the paramte value for the putBlogStats function

        $rsub=$line['resubmit'];

        $timedue=$line['timedue'];

        $userid=$line['userid'];

        $astime=date('Y/m/d/h/i',$line['timemodified']);

        //Put in the blog stats underneath

       //  function putBlogStats($assignmentid, $blf, $userId, $subrecord,

        //$assignmentname,$astime,$timedue,

$rsub,$subrecorda,$aslec=false,$subrelec=0,$subreleca=0)


        $this->blogCont->addToStr($this->putBlogStats($assignmentid, $blf,

           $userid,$subrecord, $coursetitle,

           $astime,$timedue, $rsub,$subrecorda,$aslec=false,$subrelec,$subreleca));

//.$count);


        //Put in a blank line

        $this->blogCont->addToStr("<br>&nbsp;");

    }


    //LAYER: Create a centered layer to display the blog centred

    $this->center = &$this->newObject('layer', 'htmlelements');

    //set the align property of the layer to center

    $this->center->align="center";

    //Add all the blogs to the centred layer

    $this->center->addToStr($this->blogCont->addToLayer());

    // set the value for the current middle page;
    $mid=$this->center->addToLayer();
    /*
    * Note that we have not yet rendered any output. Everything is still
    * resident in memory as strings.
    *
    * The next step is to build the right hand column, and then add the
    * blog contents layer to the left cell of a table, and the right hand
    * content to the right cell of a table.
    *
    * I used a table here because of problems getting the layer approach
    * to work easily in all browsers given that we cannot predict the amount
    * of content and what else will need to be in the column. Layers would be
    * conceptually better, but using a  table at this stage is safer.
    */
/*------------------------------ END MAIN BLOG CONTENT AREA ---------------------*/


    /*
    * Create the right layer for the nav & info. Not that $this.rightframe() is a
    * method of the controller class. If you want to see what is going into the
    * right frame, you need to look at that method of the controller class.
    *
    * The reason for putting it outside the template is because it is used
    * in multiple other templates, and involves programme logic.
    *
    * Probably, if I was starting this over again, I would make a separate class
    * that handled this type of thing. It would be 'cleaner' that way.
    *
    */
```

```
 $left =$this->leftframe('',$courseid,'stu');

 $right = $this->rightframe('lec',$id);

 /*

 * Put the newsfeed into an IFRAME using the iframe framework extension.

 * We do this because the newsfeed comes via RSS from an external server,

 * and waiting for it to load can slow the whole site down. Therefore, we

 * put it into an IFRAME to allow the rest of the page to render without

 * being impacted by the slow loading of the newsfeed.

 */

// $this->loadClass('iframe', 'htmlelements'); //load the iframe object

 /*

 * Create a new instance of the IFRAME object.

 * Note that we can use new iframe() here because the iframe

 * class does not extend any of the framework objects. If it did

 * we would have to use $this->newObject()

 */

// $this->iframe=& new iframe();

 // Set up the location URI

// $location = $this->uri(array('action' => 'getrss',

//  'module' => 'blog'));

 //Set the SRC of the IFRAME

// $this->iframe->src=$location;

 /*

 * Set the width of the IFRAME as a percentage. It is set

 * to be the same as the table above it (set up in the controller)

 * so that when the browser is resized they appear the same width.

 */

// $this->iframe->width="91%";

 // Set the height of the iframe so it does not take up too much space

// $this->iframe->height=150;
```

```
/*
 * Use BR to move it down one blank line below the table and add
 * it to $strout using teh show method.
 */
// $strout .= "<br />".$this->iframe->show();


/*
 * Create the table for the final layout
 * Note that we are still not generating any output.
 * Everything is still resident in memory as strings.
 */
$this->rtTable = $this->newObject('htmltable', 'htmlelements');
//Begin a new table row
$this->rtTable->startRow();
/*
 * Add a cell to the table, make it 75% of the available width and add
 * the contents of the centered layer that we made earlier into it.
 */
$this->rtTable->addCell($left , "25%", "top", Null, "blog-right");
$this->rtTable->addCell($mid , "50%");
/*
 * Add a cell to the table, make it 25% of the available width and add
 * the contents of the right hand column string ($strout) into it.
 */
$this->rtTable->addCell($right, "25%", "top", Null, "blog-right");
//End the table row
$this->rtTable->endRow();


/*
```

```
    * At last put the content on the page. Note that we use the show

    * method of the table object. Remember, if you create a framework

    * extension, make sure that it has a show method that returns a

    * string. Extensions should not generally contain any code to actually

    * render output to the browser (i.e. no echo or print statements)

    */

    echo $this->rtTable->show();


    // Render the bottom template area as output to the browser


    echo $this->putAsFooter('student');

?>
```

End-of-appendix C

End-of-document