PENINSULA TECHNIKON
FACULTY OF ENGINEERING

The design and implementation of an Assessment 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: Na Zhang (200206084)

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

PENINSULA TECHNIKON
FACULTY OF ENGINEERING

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

Prepared by: Na Zhang (200206084)

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

I, the undersigned, hereby declare that the work contained in this dissertation, is my own original work and has not previously in its entirety, or in part been submitted at any university for a degree.

ZHANG Na.

Acknowledgements

This study would not have been possible without the contributions of many people and organisations. First I am appreciative of all the developer members of Knowledge Environment for Web-based Learning Next Generation (KEWL.NextGen) of the University of the Western Cape (UWC). My thanks to all who welcomed me into their meetings, and especially to those who gave me technical support and who offered me opportunities to listen and learn. I am forever grateful for their generous sharing of their most valuable resources and time to help me move forward.

Secondly, I would like to thank to my mentor, Bennett Alexander, Head of Department, Information Technology, Peninsula Technikon, who provided intellectual support, constructive criticism, and encouragement as I moved through this research.

Finally, a special word of thanks my parents and family members for support while I have been in South Africa to study.

致谢

我十分感谢我的导师，Alexander 先生，感谢他给予我大力的支持和鼓励，以及对论文提出的建设性批评和意见。同时感谢我们的项目合作伙伴，南非西开普大学的各位开发人员，感谢他们给予的技术支持和帮助。我还要感谢所有在南非帮助过我的人们。最后，我要感谢在中国的父亲、母亲以及亲人、朋友，感谢他们的支持和鼓励。

Abstract

This dissertation was written in fulfillment of the requirements for the degree Master of Technology in the Faculty of Information Technology at the Peninsula Technikon in South Africa.

The dissertation covers the introduction of the study; the reviews of the case, Free and Open Source Software; the discussion of methodology of software project management in terms of software engineering; the analysis of assessment methods; the techniques of using PHP, MySQL, Apache and CVS; and the implementation of a test module.

The research represents the consideration of the problem, namely that there are few free assessment manager programmes that can be used for the quick and efficient deployment of academic assessments. An experimental research method was used to with the aim of solving the above problem. In other words, a test management system module had to be designed and implemented to function within the architectural constraints of a developing Open Source Software (OSS) Learning Management System (LMS), which is called Knowledge Environment for Web-based Learning Next Generation (KEWL.NextGen).

The test management system is a module developed on Object-Oriented Programming (OOP) in PHP and MySQL application. The scripts of this test module are written under the KEWL.NextGen' application framework, which is based on a close approximation of model, view and controller (MVC) design

pattern. Now the entire source coding of the test module has been uploaded to Web site http://kngforge.uwc.ac.za.

This test module can be used for the creation and management of an online test. It provides facilities to create a new test, and to preview, edit, delete and mark an existing test. Three question types have been achieved, namely multiple-choice, true/false and short answer.

Abbreviations

OSS: Open Source Software

IT: Information Technology

LMS: Learning Management System

KEWL: Knowledge Environment for Web-based Learning

KEWL.NextGen: Knowledge Environment for Web-based Learning Next Generation

UWC: University of the Western Cape

MVC: Model-View-Control

ASP: Active Server Pages

GPL: Gnu Public Licence

XAMPP: X window Apache MySQL PHP PEAR

CVS: Concurrent Versions System

RDMS: Relational Database Management System

COL: Commonwealth Of Learning

ICT: Information and Communication Technologies

MIT: Massachusetts Institute of Technology

OSI: Open Source Initiative

FSF: Free Software Foundation

SDLC: System Development Life Cycle

SMEs: Subject Matter Experts

UML: Unified Modelling Language

FTP: File Transfer Protocol

OOP: Object-Oriented Programmeming

AVOIR: African Virtual Open Initiatives and Resources

PEAR - PHP Extension and Application Repository

HTTP: HyperText Transfer Protocol

DNS: Domain Name System

URL: Uniform Resource Locator

HTML: HyperText Markup Language

Table of Contents

CHAPTER 1 INTRODUCTION TO THE STUDY 1

1.1 Overview 1

1.2 Background 3

1.2.1 E-learning executive summary 4

1.2.2 LMS capability 8

1.2.3 Free and open source software 9

1.2.4 GNU public licence (GPL) 11

1.2.5 Linux, Apache, PHP and MySQL 12

1.3 Statement of the Problem 15

1.3.1 Awareness of the problem 15

1.3.2 The sub problems 15

1.4 Assessment 16

1.4.1 Why assess 16

1.4.2 Computer applications within assessment 17

1.4.3 Types of assessment 17

1.4.4 The key principles of assessment 18

1.5 The Research Logic Model (Towards Solution in a Research Environment) 19

1.5.1 The significance of the research (objectives, goals, outputs, outcomes) 20

1.5.2 Research activities 22

1.5.3 Target group 22

1.5.4 Research resources 22

1.5.5 Possible limiting factors 23

1.6 Assumptions and Delimitation of the Research 23

1.6.1 The assumptions 23

1.6.2 Delimitation of the research 23

CHAPTER 2 LITERATURE REVIEW 24

2.1 Summary 24
2.2 Free and open source software 25
2.3 Driving forces 26
2.4 The current situation 27
2.5 Why free and open source software 29
2.6 Open standards 29
2.7 Using free and open source software 30
2.8 Financial and legal aspects 31
2.9 Conclusion 32
CHAPTER 3 RESEARCH PROJECT PLAN, METHODOLOGY AND MANAGEMENT 33
3.1 Software project management plan introduction 34
3.1.1 Project overview 34
3.1.2 Project organisation methodology 34
3.1.3 Management process 35
3.1.4 Technical process 35
3.1.5 Work packages, schedules, and budget 36
3.2 Methodology 37
3.2.1 Executive summary 37
3.2.2 Waterfall methodology summary 38
3.2.3 Waterfall methodology strength 40
3.2.4 Waterfall methodology weaknesses 41
3.2.5 Iterative methodology summary 42
3.2.6 Iterative methodology strength 43
3.2.7 Iterative methodology weaknesses 45
3.2.8 Methodology application 46
3.3 The management processes of this project using waterfall methodology 46
3.3.1 The specifications of management processes 47
3.3.2 Analysis 48

3.3.3 Conceptual design 49
3.3.4 Interface design 49
3.3.5 Testing, Evaluation and Delivery 49
CHAPTER 4 ASSESSMENT METHOD ANALYSIS 50
Introduction 50
Objective 51
4.1 Assessment practise principles 51
4.2 Assessments design 51
4.3 Assessment implementation 53
4.4 Research questions 54
4.4.1 What is to be gained from formative assessment? 54
4.4.2 Why is formative assessment so important to inquiry learning? 55
4.4.3 What does formative assessment involve? 56
4.5 Formative assessment cycle 56
4.6 How to improve test questions 58
4.6.1 Choosing between objective and subjective test items 58
4.6.2 When to use objective or essay tests? 58
4.6.3 Multiple-choice test items 59
4.6.4 True/False test items 65
4.6.5 Matching test items 68
4.6.6 Completion (Fill-in blank) test items 70
4.7 Two methods for assessing test item quality 72
4.8 Exposition of a test module 75
4.8.1 System overview 75
4.8.2 Teachers' shell 75
4.8.3 Students' shell 76
4.8.4 Question bank 76
4.8.5 User records 77
CHAPTER 5 OPEN SOURCE SOFTWARE TECHNOLOGY APPLICATION 78

5.1 Apache 78

5.1.1 Using Apache with Microsoft Windows 78
5.1.2 Running apache as a service 80
5.1.3 Testing the installation 81

5.2 MySQL 82

5.2.1 Choose which distribution to install. 82
5.2.2 Download the distribution that needs to be installed 82
5.2.3 Install the distribution 82
5.2.4 Preparing the windows MySQL environment 83
5.2.5 Perform any necessary post-installation setup 84
5.2.6 If one wants to run the MySQL benchmark scripts, Perl support for MySQL must be available 85

5.3 XAMPP (software package install apache distribution containing MySQL, PHP and Perl) 86

5.3.1 Installation 87
5.3.2 Apache notes 90
5.3.3 MySQL notes 90

5.4 Using CVS 92

5.4.1 TortoiseCVS: a Windows CVS client 94
5.4.2 Checking out the KEWL module 94
5.4.3 Updating local copy 96
5.4.4 Committing changed code back to the repository 96
5.4.5 Adding and removing files 97
5.4.6 Use of CVS for KEWL.NextGen 97

5.5 Introduction to object-oriented programming 98

5.5.1 Programming 98
5.5.2 Object-oriented terminology 99
5.5.3 Object 100
5.5.4 Class 101
5.5.5 Object vs. Class 102
5.5.6 Inheritance 102

5.6 Object-oriented programming 104

5.6.1 Object-oriented programming in PHP 104

CHAPTER 6 RESEARCH PROJECT IMPLEMENTATION IN KEWL.NEXTGEN 108

6.1 Executive summary of the existing LMS KEWL 108

6.1.1 What is KEWL? 109

6.1.2 The assessments current feature of KEWL 1.2 111

6.2 Design pattern 114

6.2.1 What are design patterns? 114

6.2.2 Why use design patterns? 115

6.3.3 Model –View- Controller design pattern and framework 115

6.3 The KEWL.NextGen framework 117

6.3.1 Objects and instances of objects and their naming 120

6.3.2 Database abstraction 121

6.3.3 Normalisation and referential integrity 121

6.3.4 Making a ktest module with the KEWL.NextGen application framework 122

6.4 Use of UML for KEWL.NextGen 123

6.4.1 What is the UML? 124

6.4.2 Designing 124

6.4.3 Documenting 125

6.5 Report on test module application in KEWL.NextGen 125

6.5.1 Functionality of test module 125

6.5.2 The test module application use cases 125

6.5.3 Architectural representation - activity diagrams 132

6.5.4 Implementation 135

6.5.5 Process view 137

6.5.6 Architectural constraints 139

6.5.7 Enhancement 140

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS 142

7.1 Conclusions 142
7.1.1 The primary purpose of this research project 142
7.1.2 The objectives that have been achieved 143
7.1.3 The functionalities that have been achieved as per design 144
7.1.4 Measuring the quality of the test module 145
7.1.5 The specific technology frameworks that have been used to develop this project 145
7.1.6 Lessons that were learned in the project management of this project 146
7.2 Recommendations 146
LIST OF REFERENCES 148
Web resources: 148
Book resources: 149
Appendix A: An evaluation of the development of a test management module against the Information Management Body of Knowledge (IMBOK) framework 150
Appendix B: KEWL.NextGen Learning Management System 159
Appendix C: Glossary 164
Appendix D: Code of the test module 165

List of Figures

Figure 1-1: Typical LMS Communications 9
Figure 1-2: Research logic model 19
Figure 3-1: Waterfall model sequence 39
Figure 3-2: Waterfall model deliverables 39
Figure 3-3: Project work to be undertaken 48
Figure 4-1: Formative assessment cycle diagram 57
Figure 4-2: A Test module overview 75
Figure 4-3: Question bank 76
Figure 5-1: CVS checkout 94
Figure 5-2: CVS checkout module 95
Figure 5-3: Checkout finished 96
Figure 6-1: Current KEWL 1.2 assessment module architecture diagram 113
Figure 6-2: Model-View-Controller pattern 117
Figure 6-3: MVC approach to the application framework 119
Figure 6-4: Location of files 122
Figure 6-5: The test module application use case diagram 126
Figure 6-6: The test module activity diagram 134
Figure 6-7: Main page of test module 137
Figure 6-8: Edit page of test module 138
Figure 6-9: Result page of test module 139

List of tables

Table 1-1: Assessment Outcomes 16
Table 1- 2: Computer applications within assessment 17
Table 5-1: CVS repository for KEWL.NextGen 98
Table 5- 2: object-oriented programming terms 99
Code list 5-1: PHP Class 105
Code list 5-2: Create an object 105
Code list 5-3: Inheritance 106
Code list 5-4: constructors 107
Code list 6-1: the constructor 118
Code list 6-2: controller main file 118
Code list 6-3: InnoDb handler 122
Table 6-1: UML 124
Table 6-2: A use case specification 127
Table 6-3: Test module physical structure 1 136
Table 6-4: Test module physical structure 2 136
Table 6-5: Test module physical structure 3 136

CHAPTER 1 INTRODUCTION TO THE STUDY

1.1 Overview

"In I.T. projects, there are still a number of unique problems, related mainly to the unclear nature of our objects. It is very uncommon at the start of a systems project to know exactly what the required system should be like. In this respect, systems projects are much more like research projects. Fortunately, techniques are now evolving to help control these "fuzzy" project"(Mcleod & Smith, 2001:3).

The first part of this research will be a literature study, a technology study, and an Internet search to find out what the best practises and user-requirements are. These will be used to design, implement and develop an assessment module and its type and mode. The next part is implementation of the system in an existing Open Source Software (OSS) Learning Management System (LMS) as a test module. This system has been developed by the University of the Western Cape (UWC) and is called KEWL (Knowledge Environment for Web-based Learning). The existing open source application will be converted entirely to PHP, and open source programming language. The test module of Kewl.NextGen (KEWL Next Generation) will be the research project. The dissertation structure is shown below.

In chapter 1, introduction to the study, the background of a learning management system was provided, and open source software and applications were explained. The research problem was stated and aims and objectives of the research were described. The end users were defined.

In chapter 2, literature review, one typical article, "Free and Open Source Software" was chosen to study and analyse current open source environment, developing foreground and its application. The literature review brought the researcher into contact with new ideas and approaches, and revealed unknown data sources.

In chapter 3, which is the research project plan, methodology and management, two methodologies were discussed, namely waterfall and iterative-based on the comparison. Waterfall methodology was deemed suitable for this research project in terms of software engineering. Next the management process of the research project was determined. Software project management is most important to any project, and certainly in this research project.

In chapter 4, the assessment method analysis, assessment practice principles, implementation and methods were analysed. Methods by which to improve the quality of test questions were provided. This analysis will impact great on design and implementation of the test module.

In chapter 5, a concrete explanation was given of the knowledge resources and technical skills that were used. Object-oriented programming application in PHP was also described. At the same time open sources were studied and researched. Furthermore, it was exphasised that a learning process is a prerequisite for implementation of a test module.

In chapter 6, which covers the research project implementation in KEWL.NextGen, KEWL.NextGen project background, design pattern MVC (Model-View-Control), application framework of MVC and code standards were introduced. These offered the unification of standards for the KEWL.NextGen project. The latter section of this chapter includes a report that contains the information on the design of the completed test module and its architecture. The design included the functionality, management, and advanced features of the test module. The architecture sections covered logical and physical architecture. This chapter explains the whole process of the test module implementation and management in Kewl.NextGen and what happened in the test module itself.

In the final chapter, encapsulating the conclusions and recommendations, three aspects of the research project emerged: in terms of the test module itself, functionality; secondly, implementation of the technology that was used;

thirdly, the project methodology. Special recommendations were made to the partners, and meaningful recommendations regarding further research were put forward.

1.2 Background

The evolution of the Internet is one of the most exciting technological advances in recent history. One of the most interesting aspects of this evolution is the emergence of electronic learning (e-learning) as an innovative educational approach. Education is on the brink of a new era based on the telecommunication revolution and the explosion in the use of the Internet. This information highway has made participation in e-Learning a reality for anyone. E-learning is defined as the process of using electronic technology to map the traditional teaching and learning activities in an educational process where the instructor and the students are geographically separated. What is important to the understanding of e-learning is not the evolution of the hardware or underlying network but the evolution of the usage of the Internet, and the role that it plays in the lives of users. Clearly, as bandwidth increases and higher speed network access reaches users, these factors play a role in user adoption of e-learning. Such types of systems provide the accessing and searching for updated material and also increases system flexibility and availability. This leads to an expected growth of participation level in the education process.

The learning landscape today is very different from that of 50, 20, even five years go. Not very long ago, students had two options for post-secondary learning: college or university. Students typically got an education and then made their own way through the world. Only very specific skills required further specialised training. Today, the options are limitless. In addition to college and university programmes that have become increasingly specialised, there are specialised training schools. Students can choose from technology schools or specialised training schools. Yet these learning options are fraught with issues. There are many reports of unsuccessful programmes: from the curriculum material that is obsolete, to students attending high-priced training schools

only to come to their new job still lacking relevant skills. These issues are serious for individual learners. The organisation of learning institutions and corporate training programmes has traditionally focused on administration. But new developments in learning management have shifted the emphasis from *administration to strategic planning. This new focus represents a quantum leap* in learning management.

1.2.1 E-learning executive summary

E-learning is a wide set of applications and processes, which use all available electronic media to deliver vocational education and training. The term covers computer-based learning, Web-based learning, and the use of mobile technologies; it includes virtual classrooms and digital collaboration and uses. Drivers for e-learning continue to be a mixture of the advent of new technologies; the expectations of users growing up with new technologies and increasingly using them for productivity and pleasure; the quest for efficiencies in the delivery of training; organisational initiatives to maintain currency with developments in technology; and policy initiatives. The e-learning market place has been expanding and is predicted to grow significantly over the coming years. The advances made are now being brought into a phase of consolidation, with a focus on quality and standards that ensure access to effective learning experiences. The creation of technical specifications and the development and adoption of technical standards are key activities ultimately underpinning the success of e-learning globally. Within the multimedia and Web development industry there is clear evidence of a gradual maturing of practises, such as the adoption of appropriate and sound process models for development, the inclusion of quality goals and testing through defining requirements, and the recognition of the critical role of the Instructional Designer in e-learning development. There is an increasing recognition that successful learning requires not just quality instructional content but an appropriate context that includes its facilitation and an understanding of the learner. Educational Theory has also had an impact on using Information and Communication Technologies (ICT), as content design practitioners search for a theoretical basis to justify their designs, and as technology is seen

increasingly as an enabler of learning. The growing trend to blended learning recognises the use of ICT in the instructional process as one that augments rather than replaces face-to-face delivery, and provides unique experiences that assist in achieving desired learning goals. Continually changing demographic profiles for consumers of e-learning imply the need to adopt a user centred design process for development projects, rather than use an off-the-shelf or template solution, and underscore the importance of developing processes and skills rather than product. Access to and the ability to effectively use ICT to obtain information and services are becoming increasingly important requirements necessary to fully participate in contemporary Australian economic, political and social life. Improved access is achieved through transforming workplace practises and removing significant barriers to organisations adopting e-learning, as well as targeting policies to increase information literacy. Dramatic changes in hardware and software and the continued mainstreaming of technologies into our lives through e-commerce and entertainment are providing a powerful and unstoppable force for the growth of e-learning. The energy in the e learning debate-surrounding standards will undoubtedly subside and be replaced by broader concerns relating to quality of learning experience. Learning for life requires knowledge to be personally, socio-economically and culturally integrated. E-learning is becoming an integrated and critical component of corporate knowledge management and performance enhancement, and return on investment is measured in that context. The success of e-learning can be electronically related to business successes and more businesses will recognise e-learning's ability to build knowledge and develop skills while reducing training-related costs. Within corporate training, there is a sound understanding of how to exploit these linkages.

E-learning as a component of flexible learning is a wide set of applications and processes, which use all available electronic media to deliver vocational education and training. It includes computer-based learning, Web-based learning, virtual classrooms and digital collaboration and uses. Online or Web-based learning (learning via the Internet, intranets and extranets) is increasingly understood to be a subset of e-learning (technology supported

learning). In this dissertation e-learning is discussed with particular emphasis and relevance to online learning and Web-based technologies, as these represent the area where most of the effort and data are available.

There are many identifiable drivers for ICT enabled instruction, and these may be classified as technical innovation, organisational and business developments, and the needs and demands of the individual learner. Often it is a combination of these three components that brings about change. Although the rate of development in each of the areas differs, they are clearly not mutually exclusive, and developments are invariably led by the advent of new technologies. The capabilities of hardware and the technical infrastructure to support it move ahead rapidly and often lead to development in other areas. For example, technologies such as wireless access to high bandwidth and Web-enabled mobile phones are released to a market and initially adopted by enthusiasts. Business then experiments with possible viable models for the efficient use of the technologies and methods for integrating them into organisational practises and culture. Finally, as end users become accustomed to the technology and the demographic profile shifts, they will more readily recognise its benefit and the technology becomes a part of their lives, and the cycle is complete.

A similar cycle exists within educational technologies. Educational applications base technologies and tend to lag significantly behind in their maturity cycles, not least because of the complexity of integrating them into organisational practises and culture, and in gaining broad acceptance from their users. For an educational technology to be mature, it must firstly be a stable technology, secondly it must be integrated into the business and provide a Return On Investment, and thirdly it must conform to workflow and practises of its core users, namely teachers and learners. They are the ultimate consumers of e-learning.

E-learning is efficient because it shortens the time required to update workers on new products, methods, and processes. Proponents of e-learning suggest that it provides real-time learning of critical or just-in-time knowledge. With

state-of-the-art e-learning management systems, training costs can be traced to individual learners and costs can then be measured against results. Advocates believe that online training is better, faster, and cheaper than conventional training. E-learning is becoming an integral part of organisational training. It may be delivered via numerous electronic media, including the Internet, intranets, extranets, satellite broadcast, audio/videotape, interactive television, and CD-ROM.

Learner needs are also driving forces in e learning. In today's new economy that is characterised by industrial change, globalisation, increased intensive competition, knowledge sharing and transfer, and information technology revolution, traditional classroom education or training does not always satisfy all the needs of the new world of lifelong learning. Learner-centred learning is replacing instructor centred learning. Such learning is initiated in homes, offices, shops and factories as well as classrooms. E-learning provides people with a flexible and personalised way to learn. It offers learning-on-demand opportunities and reduces learning costs. At its best, e-learning is individual, customised learning that allows learners to choose and review material at their own pace. Technology has been a strong catalyst for educational innovation and improvement, especially where the World Wide Web is involved.

The e-learning market-place has been expanding and is predicted to grow significantly over the coming years. While the discussion of e-learning is not exclusive to any particular technology such as the Web, it is there that most of the activity and interest are centred. It is worthwhile remembering that the Web as an underpinning technology for enabling communication, and as a means of delivering learning, has been with us for only 10 years. In that time, there have been multiple enhancements of the fundamental concept of the Web.

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.

1.2.2 LMS capability

For the enterprise, a learning management system provides an online learning environment by enabling the management, delivery and tracking of blended learning (i.e. online and traditional classroom) for employees, stakeholders and customers. A robust LMS should integrate with other departments, such as human resources, accounting and e-commerce, so that administrative and supervisory tasks can be streamlined and automated and the overall cost and impact of education can be tracked and quantified. Furthermore, a 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, a 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.

For education, 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, a LMS tracks their progress through a programme of study or provides a forum for collaboration with peers. A LMS provides a self-serve function that simplifies the clerical requirements of participation in a learning programme. It makes registration, enrollment, purchase, and payment more efficient. It provides catalogue for courseware and learning materials, it offers notification options and the ability to collaborate online with instructors or fellow students. For administrators, using a 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.

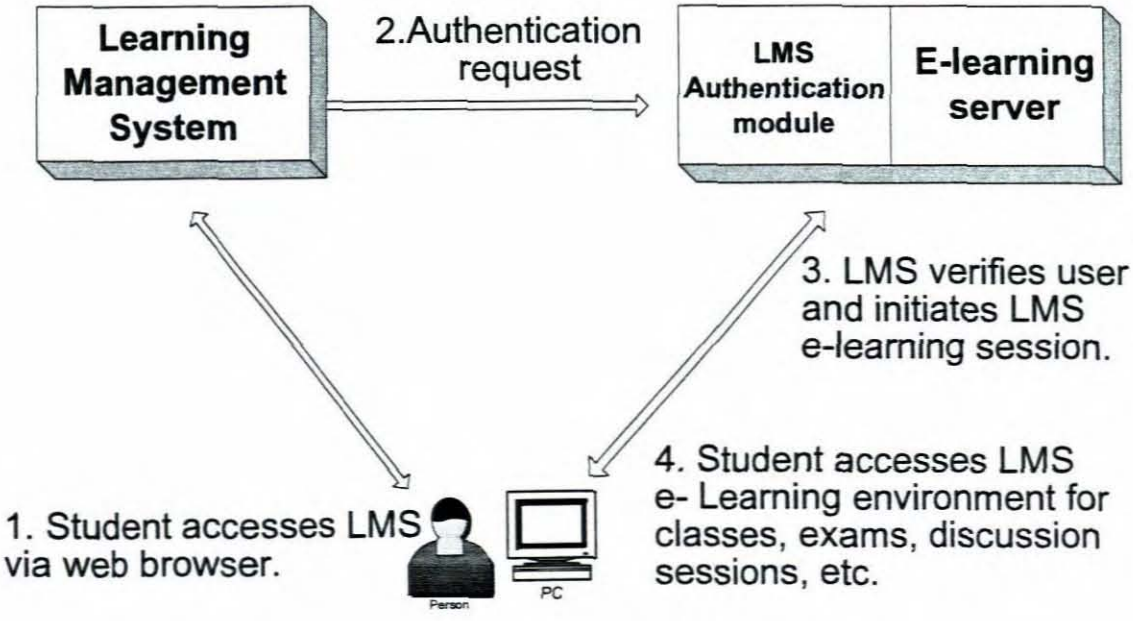


Figure 1-1: Typical LMS Communications

1.2.3 Free and open source software

At the beginning of the 1980s, licensing conditions were changed for Unix. Prices increased and the possibilities for sharing source code were restricted substantially.

In 1984 Richard M. Stallman started a project called GNU with the idea of recreating the open environment that he had experienced as a member of the application development team at the Massachusetts Institute of Technology (MIT). But this time his ambition was greater. He wanted to create an operating system that was completely open and free. He also created a special type of software licence called the GNU licence (GNU GPL – General Public Licence) in order to guarantee that software developed within the GNU project would continue to be open and free for all to use.

GNU GPL became the object of wide attention, but not everyone was attracted to the puristic and ideological spirit in which Stallman worked. Bruce Perens, together with Eric S. Raymond, therefore started the Open Source Initiative (OSI) in 1997. They provided their own definition of open source code, "The Open Source Definition". This meant that held themselves to the guidelines set forth in the definition could be called Open Source Software. The main difference between GNU GPL and OSI is that OSI allows commercial use and sales of the software. Proprietary software is sold without any access to source code and can therefore not be changed, improved or further distributed. A licence for proprietary software entitles a user only to use the software under certain conditions.

Free and open source software gives the user the right to use, copy, distribute, examine, change and improve the software. These rights are stipulated in the licences, which apply to free and open source software. GNU GPL is the most widespread licence, but also the one that places the most demands on openness and lack of restrictions.

The licensing model in GPL is very much different from other software licences, mainly due to the intentions of the licensing agreement and its protective clauses. GNU GPL is based on the freedom to make an agreement or contract and on the fact that the originator has the right to make the software available on his/her own conditions. Copy left is a mechanism, in GPL licences, which stipulates that free software remains free, even when modified or changed. Copy left does not allow someone who further distributes the software, in

original or modified form, to add any restrictions to the licence. This means that all copies of the software, even modified, continue to be free.

It is this mechanism, which precludes reuse of GPL licenced software in proprietary software, which Microsoft has heatedly opposed.

1.2.4 GNU public licence (GPL)

The GNU General Public Licence is often called the GNU GPL for short; it is used by most GNU programmes, and by more than half of all Free Software packages. The licences for most software are designed to take away one's freedom to share and change it. By contrast, the GNU General Public Licence is intended to guarantee one's freedom to share and change free software-to make sure the software is free for all its users. This GPL applies to most of the Free Software Foundation's software and to any other programme whose authors commit to using it. A user can apply it to his/her programmes.

The term "free software" is sometimes misunderstood-it has nothing to do with price, but refers to the free software can be described as free software, for a particular user, if he/she has the freedom to run the programme, for any purpose. This means that the user has the freedom to modify the programme to suit his/her needs. To make this freedom effective in practise, the user must have access to the source code, since making changes in a programme without having the source code is exceedingly difficult. The user has the freedom to redistribute copies, either gratis or for a fee; to distribute modified versions of the programme, so that the community can benefit from his/her improvements. Since "free" refers to freedom, not to price, there is no contradiction between selling copies and free software. In fact, the freedom to sell copies is crucial: collections of free software sold on CD-ROMs are important for the community, and selling them is an important way to raise funds for free software development. Therefore, a programme that may not be included on these collections is not free software. Because of the ambiguity of "free", people have long looked for alternatives, but no one has found a suitable alternative. The rich English language with its immense variety of

words and nuances lacks a simple, unambiguous, word that means "free," as in freedom-"unfettered" being the word that comes closest in meaning. Alternatives such as "liberated" and "open" have either the wrong meaning or some other disadvantage.

The GNU Project was launched in 1984 to develop a complete Unix-like operating system that is free software: the GNU system. (GNU is a recursive acronym for "GNU's Not Unix"; it is pronounced "guh-noo".) Variants of the GNU operating system, which use the kernel Linux, are now widely used. Though these systems are often referred to as "Linux", they are more accurately called GNU/Linux systems. (http:// www.gnu.org)

That is also the Web site of the Free Software Foundation (FSF). FSF is the principal organisational sponsor of the GNU Project. FSF receives very little funding from corporations or grant-making foundations. They rely on support from individuals who support the FSF mission to preserve, protect and promote the freedom to use, study, and copy, modify, and redistribute computer software, and to defend the rights of Free Software users. Free Software can be a valuable resource in education. Not only can it be technically or pedagogically superior to proprietary alternatives, but it can also promote the values of the GNU project in the schools: Freedom and Cooperation. Today, many students and schools are not aware of Free Software or the great advantages it brings. At the same time, software knowledge ability is required for more and more tasks, and Free Software has a distinct advantage in this field.

1.2.5 Linux, Apache, PHP and MySQL

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 exceeded the Minix standards. He worked from 1991 when he released version 0.02 and 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 was developed under the GNU General Public Licence 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.

Due to the very nature of Linux's functionality and availability, it has become quite popular worldwide and a vast number of software programmers have taken the Linux 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 (http://www.linux.org).

The name 'Apache' was chosen from respect for the Native American Indian tribe of Apache, well known for their superior skills in warfare strategy and their inexhaustible endurance. It is also an effective pun on a patchy Web server - a server made from a series of patches - but this was not its origin. The group of developers who released this new software soon started to call themselves the Apache Group. Between 1995 and 1999, the Apache HTTPD Web Server created by the Apache Group became the leader of the market (and currently still is, with more than 65% of the Web sites in the world powered by it). But as the Web is growing bigger, economic interests started to grow, and the Apache Web site hosted new sister projects (such as the mod_ perl project, the PHP project, the Java Apache project). The need for a more coherent and structured organisation that would shield individuals from potential legal attacks is felt to be more and more necessary.

The PHP Hypertext Processor is an open source server-side scripting language for Web servers that can be embedded inside HTML as a clever means 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 CGI scripts, but these are separate programmes which must be executed as a new process for each page hit, thus they scale badly and rapidly become memory and processor hogs as server load increases.

PHP solves this problem by becoming a 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. PHP's natural home is 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 life as a Perl programme 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.

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 a relation, which makes 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. Structured Query Language (SQL) is the most common standardised language used to access databases.

MySQL is Open Source software and is freely available at www.mysql.com. Open Source software means that the source code can easily be manipulated and modified by anyone. It is very simple to use. MySQL is very fast, reliable, and easy to use. It also has a very practical set of features developed in close cooperation with its users. MySQL is used to access databases on the Internet due to its connectivity, speed and security. 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.

1.3 Statement of the Problem

There is no assessment management system in the existing Open Source Software Learning Management System. There are few free assessment manager programmes that can be used to deploy academic assessments quickly and efficiently.

1.3.1 Awareness of the problem

Assessment is an integral part of teaching. Through the assessment it is possible to improve learning, enhance course development and measure student performance. It was deemed necessary to explore the assessment management system module, because:

- very little assessment management module programmes are written in PHP and are using MySQL in Open Source environment;
- therefore, it is not cheap to buy an assessment manager system;
- some academics never use online assessment for students;
- some students do consequently not know that they can do this; and
- even if there was a programme exited, its extent would not be comprehensive.

1.3.2 The sub problems

The following questions arose:

- Could the assessment management system module be embedded into

the KEWL?

- How much would the assessment management system module?
- What would the speed and reliability of the assessment management system module be?
- Was enough data available about mastering technology?

1.4 Assessment

1.4.1 Why assess

Assessment is an integral part of teaching. Through assessment it is possible to:

- improve learning
- enhance course development
- measure student performance

Table 1-1 highlights outcomes from assessment in these three areas. The main focus is on assessment and the improvement of learning.

Assessment and learning	**Assessment and course development**	**Assessment and student performance**
Provides feedback to the learner	Provides feedback to tutor	Provides a measure of achievement
Identifies where students need to develop skills	Allows evaluation of teaching efficacy	Ranks students
Increases student motivation	Promotes course development through curriculum review	Enables student progression

Table 1-1: Assessment Outcomes

1.4.2 Computer applications within assessment

Application	Benefits
Online coursework submission	Reduced administrative load
Recording and tracking of assessment marks	Quick and easy collation and monitoring
Online learning logs	Easy to monitor, search, sort and read
Providing online assessment materials	Low cost, easy to organise
Assessment notice board	Immediate access
Online test demo	Acquaintance test environment
Online test	Quick and easy to administer
Computer-aided assessment	Fast marking time and provide immediate feedback

Table 1- 2: Computer applications within assessment

1.4.3 Types of assessment

Formative Assessment

Black and Wiliam (1998b) define assessment broadly to include all activities that teachers and students undertake to get information that can be used diagnostically to alter teaching and learning. Under this definition, assessment encompasses teacher observation, classroom discussion, and analysis of student work, including homework and tests. Assessments become formative when the information is used to adapt teaching and learning to meet student needs. (http://edresearch.org/pare/getvn.asp? v=8&n=9).

Formative assessment is used to give students an indication of how they are progressing in terms of their skills, knowledge, attitudes and understanding in a subject. Students can use this type of assessment as a diagnostic tool to identify and improve areas of weakness and as a means of practising a skill.

Formative assessment items are often called self-assessment questions or activities and are not usually given formal grades. They aim to improve the quality of students' learning.

Summative assessment:

Summative assessment is the attempt to summarise student learning at some point, e.g. at the end of a course. This type of assessment is not designed to provide the immediate, contextualized feedback useful for helping teacher and student during the learning process.
（http://www.fairtest.org/examarts/winter99/k-forma3.html）

By contrast, formative assessment provides feedback to students during the course so that they have opportunities to improve. The process used by teachers and students recognises and responds to students' learning in order to enhance that learning, during the learning.

1.4.4 The key principles of assessment

- Assessment and learning are integrated.
- Assessment methods must be aligned to the learning outcomes.
- Appropriate criteria must be matched to the methods.
- Frequent appropriate feedback is required.
- Marking must be reliable and valid.

The following steps are to be followed in order to achieve effective assessment:

- Review the learning outcomes.
- Identify performance criteria for each outcome.
- Look at alternative assessment methods.
- Decide which methods will best the essential intended learning outcomes.
- Determine what tasks the students will undertake.
- Determine what outputs will be produced.

- Develop a method of grading the outputs.
- Develop an assessment schedule.
- Produce an assignment brief.

1.5 The Research Logic Model (Towards Solution in a Research Environment)

The logic model is an effective tool that helps to lay out the groundwork for the evaluation of the outcome of an intervention, which in development and in action research, is an important factor in determining future success.

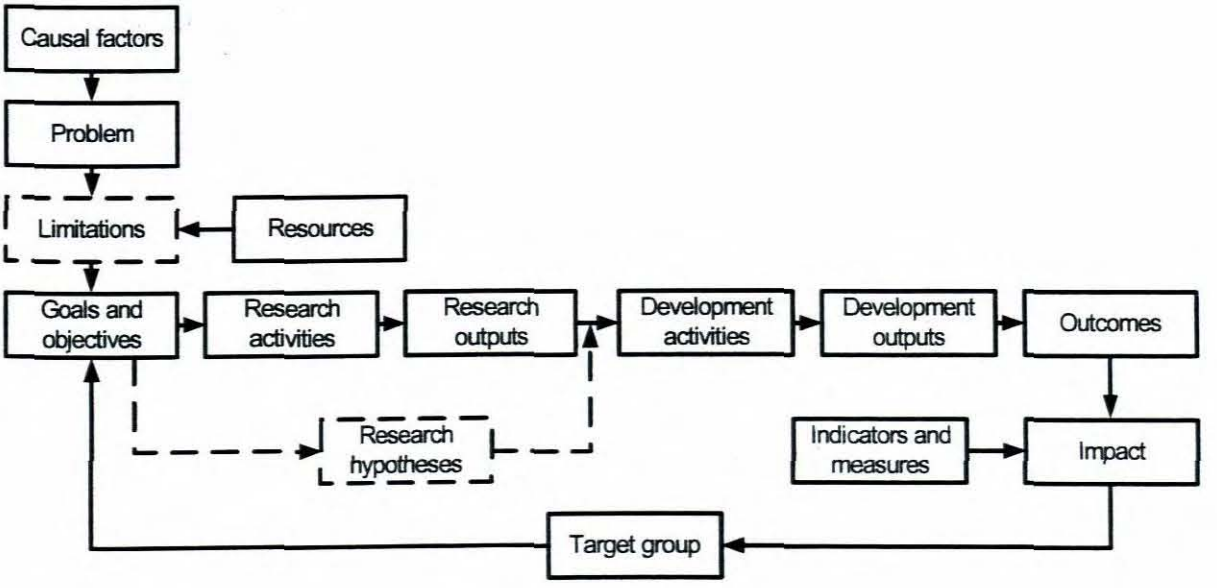


Figure 1-2: Research logic model

Figure 1-2 shows the main features of the simple logic model. There are six components of particular interest that are largely self-evident. The detailed components of the model are discussed in the section that follows.

- **Goals and objectives** are aimed or desired results; what one hopes to achieve.
- **Outputs** are what the activities produce.
- **Outcomes** are what will be expected to happen with the available outputs.
- **Activities** are what are to be done in order to achieve the goals and objectives

- **The target group** is the community or population of people (or organisations) that are the intended beneficiaries.
- **Resources** are what are needed to carry out the activities.

1.5.1 The significance of the research (objectives, goals, outputs, outcomes)

The following **objectives** were determined for the research:

- To determine the actual assessment problems through literature study or focus group, or questionnaire.
- To resolve the problems those were identified.
- To implement an assessment management logic model.
- To develop a flexible assessment management system module in an open source environment.
- To expand assessment functions.
- To enhance instructional design features, making online assessment easier and more convenient.
- To design specific assessment tool utilities.
- To understand the technology used in Linux System, PHP, and MySQL.
- To create an assessment management system module programme.
- To provide an easy-to-use tool that offers teachers the ability to create and administer customised assessment and to create a friendly assessment environment for students.
- To allow students to simulate the actual examination experience.
- To offer other business firms the opportunity to use it, alter it, and redistribute it all without charge for employee training purposes.

The following **goals** were established:

- To develop an assessment management system module that provides flexible design (interoperability) for future adaptability in an OSS environment.
- To develop an assessment management system module that provides adequate user support provisions.

- To develop an assessment management system module with appropriate software functionality and features.
- To develop an assessment management system module that meets architectural constraints of an existing LMS.
- To develop an assessment management system module that meets target users' requirements.

The following **outputs** were:

- Administrative processes for assessment
- Promotional Web and promotional material to the wider portion of the chosen target group
- Develop and run the Assessment Management System Module
- Series of working papers on learning to install Linux, using PHP and MySQL
- An Assessment Management System Module programme
- *Business partner agreement to support work*

The following **outcomes** were determined:

- Provide a free assessment management system module programme
- Increase efficiency of assessment management
- Increase reliability and improve judgment of assessment
- Develop job skills
- Improve assessment management system module programme status
- Establish more effective distribution of assessment resources
- Equip graduates to handle information according to organisational needs
- Increase self-esteem

1.5.2 Research activities

- Establish laboratory, install Linux, PHP, MySQL and server
- Allocate the necessary equipment for the laboratory
- Workshops
- Training
- Study Linux, PHP, and MySQL and other such technologies
- List a questionnaire for teachers and students
- Get information from the target users
- Research the assessment method
- Identify software project plan
- Determine software project management methodology
- Create storyboard to meet target-user requirements
- Create preliminary programme
- Debug and test prototype programme
- Deliver the PHP scripting language and the whole programme online
- Evaluate programme
- Create a more complete programme and write a report

1.5.3 Target group

- High school
- College
- University
- Other academic
- Company training

1.5.4 Research resources

- Linux operating system

- PHP and MySQL software
- Free software and other publications
- Collaborators
- Self-directed
- Computers
- Internet
- Funds
- Some related books, printer, scanner, etc.

1.5.5 *Possible limiting factors*

- Organisations have no adequate management tools with which to assess the costs and benefits of investments.
- Not enough self-directed ability available.
- The thought of assessment manager type and mode are a bit imperfect.

1.6 Assumptions and Delimitation of the Research

1.6.1 The assumptions

- Education is a social issue and communities will form partnerships to resolve education problems.
- The education society will encourage communities to provide and fund support.
- The academics will support free Internet access for students to improve their ability to study independently.

1.6.2 Delimitation of the research

The Assessment Management System Module is used to determine whether a learner has achieved an adequate level of skill or mastery of some subject content. Formative assessment involves providing feedback and an opportunity for improvement to take place. This research project just is

intended to develop online test functions only, rather than including real time communication, a discussion forum, or an essays file.

CHAPTER 2 LITERATURE REVIEW

The purpose of this literature review is to provide the reader with general overview of "free and open source software". This article is as research example.

2.1 Summary

Free and open source software gives the user the freedom to use, copy,

distribute, examine, change and improve the software. These rights are stipulated in licences. Development takes place in the form of projects, and those who participate are mostly professional software developers who either take part in paid time with the permission of their employer, or in their own time. Many large suppliers (IBM, HP, Sun, etc.) use free and open source software in their own products and solutions and also contribute to the various projects with their own resources.

Free and open source software entails a new kind of competition, separated from that of traditional business in that the product generally is not owned by any single company and therefore cannot be purchased on the market. Furthermore, the software itself is not constricted by any cost or fee. It can be obtained free of charge on the Internet.

2.2 Free and open source software

One could illustrate the idea of free and open source software as compared to proprietary software with a situation where a person buys a house. Upon purchasing the house he also receives technical drawings and blueprints so that he can make changes himself to the structure. One alternative would be for the seller to retain ownership. All drawings would remain in his possession and therefore all changes to the house would have to be made by the same vendor or seller.

Another case could be the purchase of a car, where only the dealer's repair shop retains all car manuals and repair guides for exclusive rights to repair or make any changes in the performance of the car.

Richard M Stallman, founder of Free Software Foundation, uses the illustration of cookery and food. How would we experience the world around us if recipes were not freely available or free to change and modify? What would it be like if we committed a crime every time we made a copy of the recipe or gave it to someone else?

Free and open source software means that the source code is freely accessible, that the software can be freely used, changed, improved, copied and distributed by all who wish to do so.

Free and open source software does not have to be free of charge. Besides being able to construct business models around the software based on commercial aspects, a company can receive direct payment by the use of a large number of licensing schemes and models. These models can also be included in the overall definition of what we mean by free and open source software. What is important here is that the source code is available to the customer.

2.3 Driving forces

There are several studies about how free and open source software is developed, who takes part in the development and their motives for developing this kind of software. One observation made is the high level of creativity in development projects for free and open source software. Another observation is that most of those taking part in projects are themselves professional developers who either take part in their own time or with the permission of their employer on paid, company time.

Characteristic for those taking part is a strong feeling of solidarity with the developer community. Intellectual stimulation and the chance to develop one's own competency is also an important impetus. Participating members of the community want to learn from each other by contributing themselves, and being able to learn from the knowledge of other developers. Professional developers, who create both proprietary and free and open software, often prefer the latter model because of more room for creativity. There are also not the same time restraints with scheduled deadlines. One often receives quick feedback on one's contributions. It is also important to "be seen". This also contributes to the wish to arrive at a good result. In a closed environment, where an individual programmer's work is not noticed in the same way, commitment on the part of the programmer can slack and can even contribute

to mediocre programme code.

A typical community consists of a core group of dedicated and knowledgeable individuals, sometimes taking part on their own initiative and sometimes through an elective process. This group structures the on-going work and chooses the best contributions. Anyone can submit a contribution, both companies and individuals, but it is the core group that decides the design of, for example, system architecture and also which contributions are to be used. Linus Torvalds is an example of a very successful leader of such a community –Linux.

Thus, the community is responsible for development, further development and maintenance of the product. Almost all development projects put up their own portal on the Internet, a Web site for the project, where others outside the community can follow ongoing development, read documentation, download new versions (both stable versions for use and beta versions for testing), comment on ongoing work and contribute themselves to the project.

2.4 The current situation

Free and open source software enjoyed a strong upswing in Europe during the year 2002. A number of initiatives are currently underway, both on a national level and on an EU level, in order to contribute to the spreading and use of free and open software in public administration and management. Interchange of data between Administrations, a programme under the auspices of the European Commission, arranged during the latter part of February 2001, under the guidance of Erkki Liikanen, a symposium in Brussels on the topic of free and open software in the public sector. The symposium came to the following conclusions:

- There is extensive experience in the use of free and open software in the public sector in Europe.
- Current projects include both server installations (back office) and client computers in the workplace.
- Free and open source software is used because of adaptable

functionality, lower overall costs, vendor independence and adherence to open standards, interoperability and security.

A number of measures were suggested:

- Upon procuring IT services, more importance should be placed on the use of open standards, including standards for document formats and exchange.
- Authorities should supply information on possible solutions based on free and open software.
- A forum for the exchange of best practise in the use of free and open source software should be established.
- A software pool, i.e. a model for enabling exchange of software developed by members, should be established.

At this time there are at least 20 countries all over the world where governments have taken a position in favour of free and open source software. A number of these countries in South America, with Peru at the forefront, have chosen to make new laws on the use of free and open source software, while other countries, such as Great Britain and South Africa, have taken a more pragmatic approach concerning recommendations and policies for free and open source software.

Germany is perhaps the most active country in Europe, with a number of pilot projects for the introduction and use of free and open source software. The Netherlands has recently put forward a three-year plan of action for open standards and free and open source software in public administration. In the United States, the Department of Defense is a long-standing proponent of free and open source software.

The most interesting countries at the present time are India and China, countries with large populations, which can influence the future course of events on the software arena. Both of these countries have a considerably positive attitude towards the use of free and open source software, first and foremost in order to stimulate local business initiatives and to minimise the

outflow of currency from the countries, but national independence also plays a vital part. In 2003, the strongest growth for free and open source software will probably take place in Asia.

2.5 Why free and open source software

One of the strongest arguments for using free and open source software is the opportunity to arrive at a higher degree of independence regarding price and licensing conditions. In a situation with economic restraint, new and more rigorous licensing conditions, and with software that becomes replaced by newer versions more often than before, the software environment becomes increasingly more expensive.

Free and open source software enjoys a significant market share in many areas. Notably, Apache is used for more than 65% of all Internet Web servers, often with Linux as operating system. Free and open source software often has higher dependability, and in many cases better performance when directly compared to its proprietary counterpart. Scalability and flexibility within the model for the development of free and open source software enables it to be developed for a large number of platforms and environments. An area difficult to measure and compare is security, but it has been found that it is just as good, if not better, than proprietary alternatives. Free and open source software is less prone to attacks and virus over the Internet. As far as costs are concerned, it is to the advantage of free and open source software, especially if one looks exclusively at direct costs.

2.6 Open standards

Free and open source software preferably makes use of open standards, but there is nothing explicitly said about this in the various types of licences for the software. The standards that are of interest in this context are definitions of file formats and standards for the exchange of information, i.e. protocols and formats for communication between different applications and systems.

The term standard means an accepted norm for a certain activity, event or

occurrence, or a variation thereof, which is commonly used or accepted. There is, on the other hand, no clear definition of the term open standard, but generally a standard is considered open if:

- anyone can use it or take part in work on creating the standard;
- it is arrived at in generally accepted and open environment;
- development takes place according to a model of consensus, and is not led by one single player;
- the specification is published freely and is available without cost or only at prime production cost;
- documentation is published freely and is available without cost or only at prime production cost.

2.7 Using free and open source software

Several different studies show that Linux is becoming more and more common as an operating system within companies and government agencies all over the world. In the first place, Linux replaces proprietary UNIX and Windows platforms on servers. Studies show that Linux is the fastest growing operating system on the market today.

Major systems suppliers and integrators, such as Hewlett-Packard and IBM, put Linux on an even par with earlier traditional operating systems and offers support and consultancy services at the same level. IBM goes as far as choosing Linux as a UNIX-based operating system for certain system solutions instead of its own AIX. Most often a sort of partnership is established with one of the leading Linux distributors such as RedHat or MandrakeSoft and tests are carried out to ensure that applications and hardware are compatible and work together. Both IBM and HP dedicate considerable resources in the development of Linux.

Even other application providers develop Linux-based versions of their own software. Oracle is one such example. At Oracle it is considered just as important to make available software versions for Linux as any other operating system.

In some areas, Linux dominates the market for Internet servers: Web servers and large server clusters with stringent requirements for calculation capabilities. Recent developments in better user interfaces and office programmes such as Open Office have made Linux a viable alternative even for desktop computers in the workplace.

2.8 Financial and legal aspects

An interesting question is whether or not it is possible to make a profit on free and open source software. Since access to the workings of the software itself, the source code, is free of charge, any business model must be geared towards value-added services and products.

Successful business models based on free and open source software emanate from one or more of the following areas:

- Software distributions: the sale of a packaged product based on free and open source software.
- Development and sales of in-house developed product.
- Added-value sales: free and open source software is used in order to support the sale of one's own supplementary products, such as other applications and hardware.
- Services such as support, training, consulting.
- Accessories such as literature.

It is relatively easy to compare the costs for obtaining and upgrading licences between free and open source software and commercial software. When one considers other factors, one sees that a comparison becomes more complicated because each installation is generally the result of special conditions. If one has a well-working and stable environment – whether it is based on open source or commercial products – it is generally cheaper to remain in that environment. But as soon as changes are made, for example when upgrading an existing product to a new version, various costs arise.

This area is complicated and a more in-depth study of the legal aspects, copyright, immaterial rights, intellectual property rights, etc. for certain types of licences in light of Swedish legislation is recommended.

One misunderstanding concerning free and open source software is that the software is not protected by copyright. Free and open source software is regulated by licensing conditions whereof GPL is one of a number of types of licences based on copyright protection. Mixing commercial use and distribution of free and open source software together with proprietary products is fully possible – with certain exceptions for GPL.

2.9 Conclusion

The conclusion is that free and open source software in many ways, both functionally and qualitatively, is a quite equivalent to – or better than – proprietary product. Free and open source software should therefore be judged on an even par with proprietary software in a procurement process in order to establish better market competition. It is also necessary to place demands on open standards and file formats in order to achieve interoperability between different systems.

Proprietary software is mainly sold nowadays without any guarantees, and producers exclude responsibility for, more or less, all errors and bugs in the product. The producer dictates conditions for changes and corrections, and the lifetime of a product is usually short before new versions are released. If the software producer goes out of business, all development and support disappear and the user has to look for alternative solutions. With free and open source software these dependencies can be avoided.

The concept of free and open source software has existed for almost 20 years, and Linux has existed for about 10. Many of today's Internet functions are almost completely based on free and open source software, for example functions such as e-mail and the translation of computer names to IP addresses, i.e. DNS. More than 65% of all Web servers on the Internet are

based on open source software.

Open standards and formats along with free and open source software are important factors in order to be able to arrive at:

- increased competition;
- improved interoperability;
- reduced costs.

For administration in the public sector, free and open source software is not any makeshift phenomenon, but instead a fully adequate and dependable competitor to existing proprietary products and solutions.

CHAPTER 3 RESEARCH PROJECT PLAN, METHODOLOGY AND MANAGEMENT

The former section described a general set of guidelines about software project management. It would provide the directions for this project. Then based on a comparison between two different methodologies, the waterfall methodology was identified as being more suitable for this project. Thus a specific management method of this software project was determined.

3.1 Software project management plan introduction

3.1.1 Project overview

A project overview describes the relationship of a project to the target users and other projects. It provides a concise summary of the project objectives, the product to be delivered, major work activities, major work products, major milestones, required resources, and master high-level schedules and budget requirements.

3.1.2 Project organisation methodology

An organisation methodology specifies the process model for the project (system development life cycle (SDLC)), describes the project organisational structure, identifies organisational boundaries, and defines individual responsibilities for the various project elements.

Process model

A process model outlines the relationships among major project functions and activities by specifying the timing of major milestones, baselines, reviews, work products, project deliverables, and sign-off that span the project. The process model or SDLC approach is described using a combination of graphical and textual notations. The process model must include project initiation and project closeout activities.

Organisational structure

An organizational structure describes the internal management structure of the project. Graphical devices such as hierarchical organisation charts or matrix diagrams are used to depict the lines of authority, responsibility, and lines of communications within the project.

Project responsibilities

The responsibilities of project team members are described here. A matrix of functions and activities versus responsible individuals may be used to depict

project responsibilities.

3.1.3 Management process

This section describes management objectives and priorities, project assumptions, dependencies, and constraints, risk management techniques, monitoring and control practises, and the staffing plan.

Monitoring and control mechanisms

This section explains the reporting mechanisms, report formats, information flows, review and audit mechanisms, and other tools and techniques to be used in monitoring, tracking, and controlling adherence to the Software Project Management Plan. The relationship of monitoring and control mechanisms to project support functions shall be delineated in this section.

3.1.4 Technical process

The technical methods, tools, and techniques to be used on the project are described here. In addition, the plan for software documentation shall be specified, and plans for project support functions such as quality assurance, configuration management, and verification and validation shall be identified.

Methods, tools, and techniques

This section specifies the computing system, development methodology, team structure, programming language, and other notations, tools, techniques, and methods to be used to specify, design, build, test, integrate, document, deliver, modify, or maintain the project deliverables. In addition, the technical standards, policies, procedures, and guidelines governing development shall be included or referred to via other documents.

Software documentation

The documentation plan for a software project is specified directly, or by reference. The documentation plan shall specify configuration management and version control requirements along with storage and media requirements.

3.1.5 Work packages, schedules, and budget

This section of the Software Project Management Plan will specify the work packages, identify the dependency relationships among them, state the project resource requirements, provide the allocation of budget and resources to work packages, and establish a project schedule.

Work packages

This section will define the work packages for the activities and tasks that must be completed in order to satisfy the project agreement. Each work package will be uniquely identified; identification may be based on a numbering scheme and descriptive title. A diagram depicting the breakdown of activities is used to depict a hierarchical relationship among work packages.

Dependencies

This section will state the ordering relations among work packages to account for interdependencies among them and dependencies on external events. Techniques such as dependency lists, activity networks, and the critical path method are used to depict dependencies among work packages.

Resource requirements

This section identifies, as a function of time, estimates of total resources required to complete the project. Numbers and types of personnel, computer time, hardware, software, office facilities, travel, training, and maintenance requirements are typical resources that are specified.

Budget requirements

Estimates of total budget required to complete a project are identified, as a function of time.

Budgets and Resource Allocation

The allocation of budget and resources to the various project functions,

activities and tasks are specified.

Schedule

In this section specify the schedule for the various project functions, activities, and tasks is specified, taking into account the precedent relations and the required milestone dates. Schedules are expressed in absolute calendar time or in increments relative to key project milestones.

3.2 Methodology

3.2.1 Executive summary

Software development organisations are always asked what methodology they use. The implication is that one particular methodology must be superior to all others. This is particularly common among consulting companies, who usually claim some sort of competitive advantage based on their unique methodology. Other software tool companies or academics promote their own methodology as the best choice for achieving the highest quality, lowest cost, and fastest development time. Sometimes, proponents will argue for the benefits of their own particular choice with almost religious fervour. However, most development methodologies can be basically categorised as either a waterfall or iterative.

The waterfall approach emphasises a structured progression between defined phases. Each phase consists of a definite set of activities and deliverables that must be accomplished before the following phase can begin. In addition, different people are usually involved during each phase. Strengths of the waterfall approach include the ease of analysing potential changes, the ability to coordinate large distributed teams, predictable dollar budgets, and the relatively small amount of time required from subject matter experts. On the other hand, the weaknesses of the waterfall methodologies include: lack of flexibility, difficulty in predicting actual needs for the software, the loss of intangible knowledge between phases, discouragement of team cohesion, and the tendency not to discover design flaws until the testing phase.

Iterative methodologies include Extreme Programming and Rapid Application Development. The emphasis is on building a highly skilled and tightly knit team that stays with the project from beginning to end. The only formal project deliverables are the actual working software and the essential system documentation that is completed at the end of the project. This approach delivers various benefits: rapid feedback from users that increases the usability and quality of the application, early discovery of design flaws, an ability to easily roll-out functionality in stages, a more motivated and productive team, and finally knowledge retention for the duration of the project. Offsetting the strengths are the following drawbacks to iterative methodologies: difficulty in coordinating large projects, the possibility that the project will never end, a tendency not to thoroughly document the system after completion, and the difficulty in predicting exactly what features will be possible within a fixed time or dollar budget.

The trade-offs between the two basic methodologies suggests that no cure-all exists and in choosing a methodology, one should consider the particular goals and constraints to determine whether just one or a mix will be the best given situation.

3.2.2 Waterfall methodology summary

Waterfall model is the classical SDLC model first described by Royce in 1970, illustrated in figure 3 -1. It begins with requirements definition, followed by specifications (what must be done/delivered), then design (how the specifications will be met); next implementation (building the product); integration (making sure all componets work together) and finally operations (when the product is deployed into its working environment). These stages are assumed to be discrete, with one completing before the next stage commences. Thus the specification would be signed off before design would begin. In practise, it is almost impossible to get each stage's output 100 percent right before proceeding, so there is provision for feedback, to allow errors detected in latter stages to cause corrections to output produced in

earlier phases. (Mcleod & Smith 2001:116)

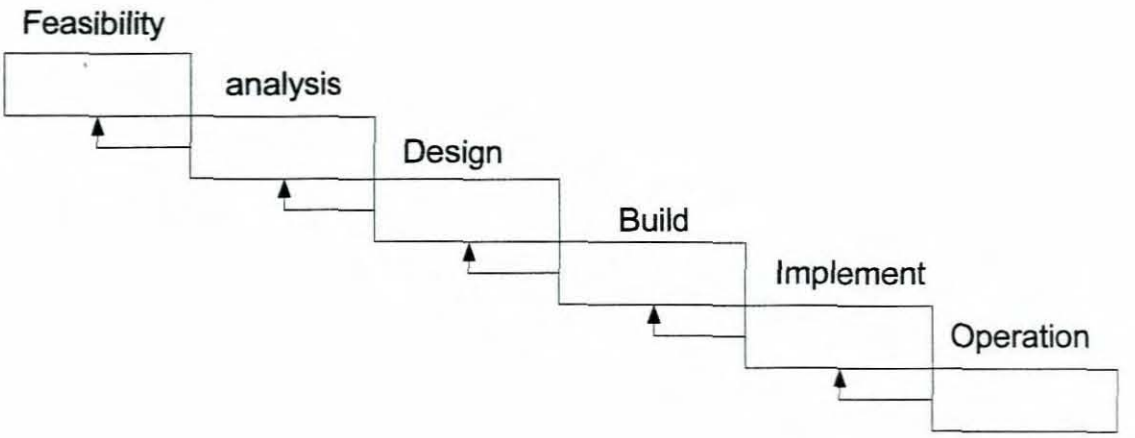


Figure 3-1: Waterfall model sequence

Typically, waterfall methodologies result in a project schedule with 20-40% of the time budgeted for the first two phases, 30-40% of the time to the programmeming, and the rest allocated to testing and implementation time, illustrated in figure 3 – 2. The actual project organisation tends to be highly structured.

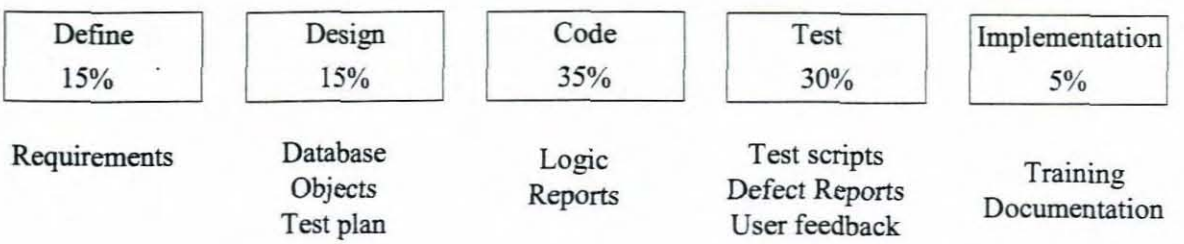


Figure 3-2: Waterfall model deliverables

Most medium to large sized projects will include a rigidly detailed set of procedures and controls to cover everything from the types of communications to use in various situations, to authorising and tracking change orders, to the specific ways that defects are logged, communicated, resolved, and re-tested. Perhaps most importantly, waterfall methodologies also call for an evolution of

project staffing throughout the various phases. While typical consulting companies will refer to the differences in staffing simply as "roles," which imply that the same people could remain on the project and simply switch roles, the reality is that the project staffs constantly change as the project progresses. Reasons for the change include economics, mentoring, and expertise - economics in the sense that the project budget encourages the replacement of a relatively highly paid architect with a lower paid staff programmer as soon as possible. On the other hand, an architect with a particular skill set or an analyst with valuable subject area knowledge may be demanded on another project. A fundamental assumption is that the extensive project documentation and control procedures enable relatively easy knowledge transfer to new project staff.

3.2.3 Waterfall methodology strength

Most of the benefits from using a waterfall methodology are directly related to its underlying principles of structure. These strengths include:

- Ease in analysing potential changes;
- Ability to coordinate larger teams, even if geographically distributed;
- Ability to enable precise budget
- Less total time required from subject matter experts (Subject Matter Experts)

Because the requirements and design documents contain an abstract of the complete system, the project manager can relatively quickly analyse what impact a change will have on the entire system. For example, if one developer wanted to modify the fields in a database table or a class, the project manager could look up what other components of the system rely on that particular table or class and determine what side effects the change may have.

The same documents that take so much time to assemble at the front-end also make dividing up and subsequently coordinating the work easier. Because the design produced in a waterfall approach is so detailed, it is easier to ensure that the pieces will be easier to integrate when the project nears the end of the programming phase. Even with a waterfall approach, the only way to ensure a

precise up-front budget cost is to have an external vendor submit a fixed bid for the remaining phases of the project after one or two of the initial phases have been completed. In this way, financial risk is contained even if the project takes longer than expected. Perhaps one of the most compelling reasons for using the waterfall approach is simply the relatively small amount of time required of the subject matter experts. Because the SMEs in a project typically have other primary responsibilities, their time is limited. The waterfall approach ensures that a significant involvement from them is only required during the initial requirements phase as well as in part of the design, testing, and implementation phases. They have very little involvement during the entire programming phase.

3.2.4 Waterfall methodology weaknesses

While waterfall has advantages, its highly structured approach also leads to disadvantages such as the following:

- Lack of flexibility;
- Difficulty in predicting all needs in advance;
- Intangible knowledge lost between hand-offs;
- Lack of team cohesion;
- Design flaws not discovered until the testing phase.

Even though the impact of changes can be analysed more effectively using a waterfall approach, the time required to analyse and implement each change can be significant. This is simply due to the structured nature of the project, and this is particularly acute when the changes are frequent or large. Even with improvements in providing wire frame screen mock-ups and more detailed flowcharts, the front-end planning process has a hard time effectively predicting the most effective system design on the front-end. One of the most challenging factors that cause this difficulty is the unfamiliarity most SMEs are with formal system design techniques. Even more disturbing is the inevitable loss of knowledge between the planning and programming phases. Even with the most detailed documents, the analysts and architects always have an implicit understanding of the project needs that are very hard to transfer via

paper documents. The information loss is particularly harmful to the project when it is developing a relatively new system as opposed to modifying an existing system. Closely linked to the knowledge loss effect, is the fact that the waterfall methodology discourages team cohesion. Many studies have found that truly effective teams begin a project with a common goal and stay together to the end. The tendency to switch out project staff from phase to phase weakens this overall team cohesion. In fact, it is common for the project manager be the only person that sees a project from beginning to end. The effect on team productivity is very hard to quantify.

The most significant weakness is the possibility that a poor design choice will not be discovered until the final phases of testing or implementation. The risk of this occurring increase as project size and duration goes up. Even dedicated and competent people make simple mistakes. In the context of the rigid waterfall timetable, mistakes made in the master design may not be discovered until six or nine months of programming have been completed and the entire system is being tested.

3.2.5 Iterative methodology summary

The iterative family of methodologies shares a common emphasis on highly concentrated teams with minimal structure and access to constant feedback from the SMEs. Typically, a project will begin with the integrated project team being assembled and briefed on the project objectives. The team will consist of all of the essential roles from the very beginning and each member may actually play multiple roles. Rather than a distinct progression through phases, the iterative methodology emphasises creating a series of working prototypes for evaluation by the SMEs until the objectives are accomplished and the system is ready for final release. During the process, it is critical for the actual project leader as well as the senior members of the team to balance the SME requests against the overall system constraints and platform limitations to ensure that quality and performance objectives can be met. It is the development team's responsibility to offer constructive feedback to the SMEs in order to suggest alternatives and work together to the best mutual solution.

Where possible, individual team members will be given complete ownership of a particular component and charged with ensuring its usability, quality, and performance. The senior team members are responsible for enforcing quality and consistency standards. Even on large projects, the initial planning will consist of only a broad outline of the business objectives and will only create a framework for the overall project components. In some cases, a set of core features for the entire system will be built initially and subsequent features added as the project progresses. In other cases, just certain modules will be built entirely during the early part of the project and other components added over time.

3.2.6 Iterative methodology strength

Many of the strengths of the iterative system are listed below:

- Rapid feedback from actual users;
- Flexibility to address evolving requirements;
- Design flaws discovered quickly;
- Easy to roll out new functionality in stages;
- Higher motivation and great productivity;
- Very little knowledge loss between phases.

Feedback from the SMEs or users can be based on an actual working prototype of the system relatively early in the project life cycle. This enables the SME to base his or her feedback on actually working with a limited version of the final product. Much as it is easier to decide if a product meets target-user needs if we can examine it in the shop than if someone were to just describe it to us, the SME is instantly able to identify potential problems with the application as the developer is interpreting his/her requirements before too much time has passed. Since the development team receives feedback at early stages in the overall development process, changes in requirements can be more easily incorporated into the finished product. More importantly, if the SME determines that a feature would not be as valuable, it can be omitted before too much development time has been spent on integrating the particular component into the overall system. In a similar way, since the team is

deploying actual working prototype versions of the application along the way, a flaw in the design should become more apparent earlier in the project schedule. Instead of discovering a potential problem only after the system goes to full-scale testing, more design flaws can be addressed before they impact other features and require significant effort to correct. Because iteration actually functions (sometimes to a limited degree), deploying parts of the system in a staged roll-out becomes much easier. Using an iterative methodology, the team simply stabilizes an earlier iteration of the component, collaborates with the SME to ensure it is stable, and rolls it out. Another advantage of doing a staged roll-out in this way is that actual production use will generate more improvement suggestions to be incorporated in subsequent iterations of the same component and/or other components.

The team approach stressed in the iterative methodology increases overall motivation and productivity. Because the same people are involved from beginning to end, they know that the design choices made will ultimately affect their ability to complete the project successfully. Productivity will be enhanced because of the sense of ownership the project team has in the eventual result. While it may seem like an "empowerment" fad, many studies have found that a team charged with a common goal tends to be much more productive than groups of people with individual incentives and shifting assignments. The fact that an integrated team maintains a thorough understanding of the project is a more tangible benefit. This effect arises simply by having the same individuals involved from the very beginning and listening first hand to the SMEs describe their needs and objectives. The subsequent feedback during iteration of the project builds upon the initial understanding. Since the same person is listening to the needs and writing the code, less time needs to be spent authoring documents to describe those requirements for eventual hand-off. This translates into more time spent writing and testing the actual software.

3.2.7 Iterative methodology weaknesses

The drawbacks to using an iterative approach are worth considering and should be weighed carefully when deciding on a methodology for a new project. Some of the more serious weaknesses include:

- Difficulty coordinating larger teams;
- Can result in a never-ending project if not managed properly;
- Tendency to not document thoroughly;
- Predicting the precise features to be accomplished in a fixed time/budget.

Iterative projects tend to be most effective with small, highly skilled teams. It is much more difficult to ensure that the components mesh together smoothly across larger, geographically distributed projects. While steps can be taken to minimise the chances of failure, coordinating large iterative development efforts is typically very hard to accomplish effectively because of the lack of detailed planning documents. Because there are no specific cut-off milestones for new features, an iterative project runs the risk of continuing into perpetuity. Even though one of the strengths is the ability to react to changing target-user needs, the project leader must determine when the major target-user needs have been met. Otherwise, the project will continue to adapt to changing target-user needs and the software will never end. This will result in never really deploying a finished product to full production use. This is a risk even in a staged roll-out situation because there are always improvements possible to any software. In any software project, there is always the tendency to borrow time from the final system documentation tasks to resolve more defects or polish certain features more. This risk increases on iterative projects because there is usually no scheduled documentation period. The result is a system that is very hard to maintain or enhance. In a similar way, in an iterative project it is much easier to fix a definite project schedule or budget than to determine exactly what features will be able to be built within that timeline. This is simply due to the fact that the features change based on user feedback and the evolution of design.

3.2.8 Methodology application

In this brief section, it has clearly presented the tradeoffs between two basic approaches to software development in order to show that no methodology is universally superior. Instead, the approach should depend on its particular needs and the constraints that one has to work with. Based on the issues discussed, a few basic guidelines that may help point a team in the right decision are listed below.

- The iterative methodology is usually better for new concepts.
- Waterfall is usually better for modifications to existing systems or building large-scale systems after proof-of concept prototypes have been established
- However, some situations will require a hybrid approach

So, in this research project the design and implementation modules to function within the architectural constraints of an existing open source software LMS in order to meet target-user requirements, it is best to choose the waterfall methodology.

3.3 The management processes of this project using waterfall methodology

Software project management is the collection of techniques used to develop computer systems and deliver various types of software products. This developing discipline traditionally includes technical issues such as the choice of software development model, how to estimate project size and schedule, how to ensure safety, what resources to reuse and which programming environment to use for the development. The discipline also includes management issues such as when to train personnel, what are the risks to the project success, and how to keep the project on schedule.

Questions that are asked initially include:

- How must one organise the development team?
- What are the indicators and measures of the product's quality?

- How must one employ a certain set of development practises?
- How can a software development organisation be transferred a to new modelling and/or development paradigm?
- How can one create and maintain a good relationship with the customers and end-users?
- What remedial actions should be taken when something goes wrong in the course of the project?

3.3.1 The specifications of management processes

Software development is a complex process involving such activities as domain analysis, requirements specification, communication with the customers and end-users, designing and producing different artifacts, adopting new paradigms and technologies, evaluating and testing software products, installing and organising end-users' training, envisioning potential upgrades and negotiating about them with the customers, and many more. In order to keep everything under control, eliminate delays, always stay within the budget, and prevent project runaways, i.e. situations in which cost and time exceed what was planned, software project managers must exercise control and guidance over the development team throughout the lifecycle of the project. In doing so, they apply a number of tools of both economic and managerial nature. The first category of tools includes budgeting, periodic budget monitoring, user chargeback mechanisms, continuous cost/benefit analysis, and budget deviation analysis. The managerial toolbox includes both long-range and short-term planning, schedule monitoring, feasibility analysis, software quality assurance, organising project steering committees, and the like. All of these activities and tools help manage a number of important issues in the process of software development. The diagram (figure 3-3) is a specific illustration of this project. It describes the whole management method of this project.

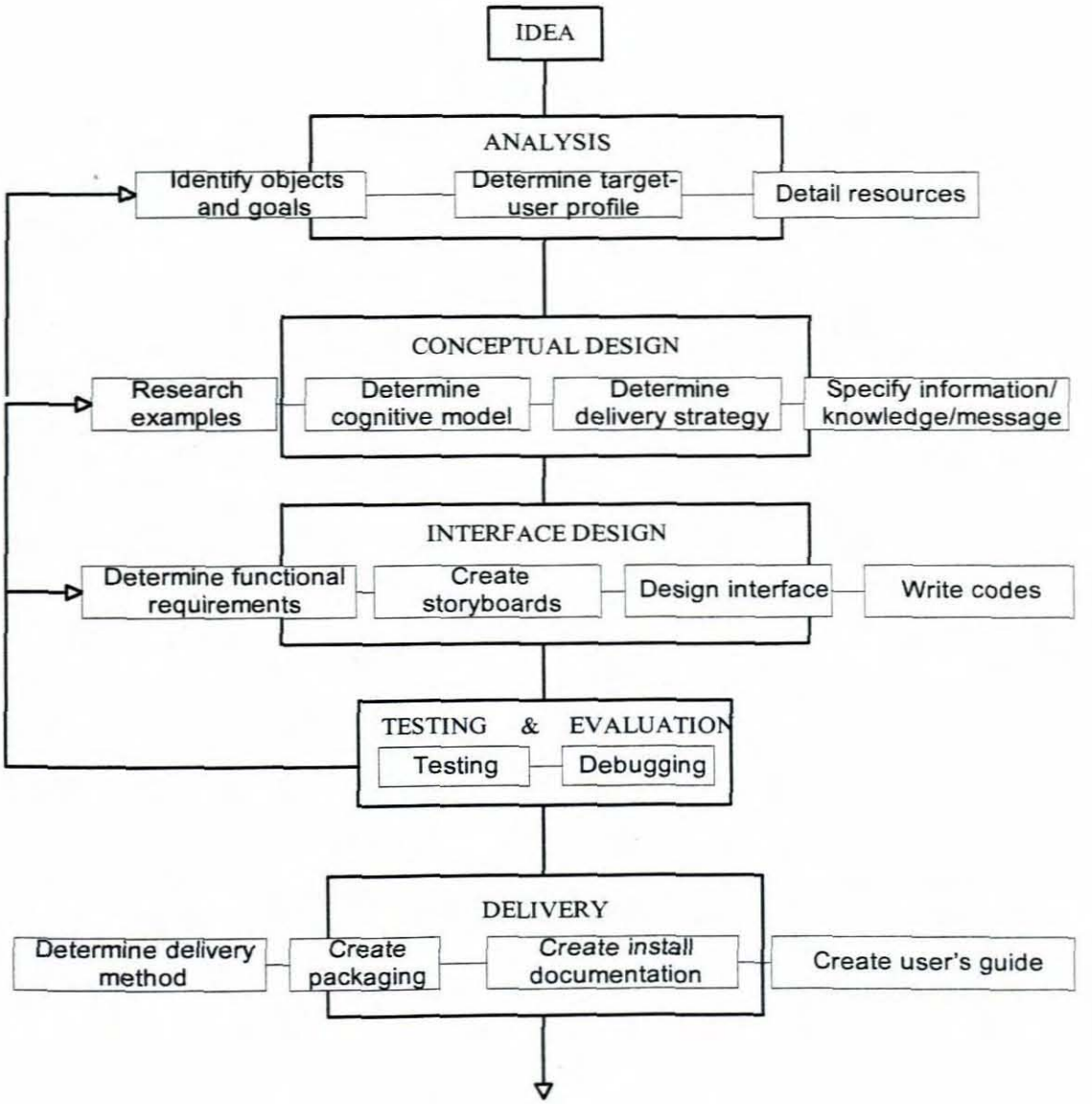


Figure 3-3: Project work to be undertaken

The checklist on the next pages is related to the diagram (figure 3-3). It explains the processes of this project briefly.

3.3.2 Analysis

Identify objectives and goals: Describe what the project is trying to achieve, and what the project is aimed at.

Determine the target-user profile: Who is the end user?

Detail resources: authoritative knowledge resources/ hardware, equipment / software/ technical skills.

These have been discussed in chapter 1.

3.3.3 Conceptual design

Research example: Examine similar existing projects for design and approach. This has been discussed in chapter 2.
Determine cognitive model: Identify the design pattern. Design patterns are basically a general method that can be used in solving a variety of different problems. This will be discussed in chapter 4.
Determine delivery strategy: Introduce how the team members communicate with each other and with target-users.
Specify information /knowledge/ message: This covers all technology application. The above two sections will be included in chapter 5.

3.3.4 Interface design

Determine the functional requirements; create storyboards; design interface and write codes. All will be included in chapter 6.

3.3.5 Testing, Evaluation and Delivery

Alpha testing of the test module should be carried out constantly throughout the process on the local host. The beta testing involves the other developer of UWC using the prototype in a variety system on Kngforge Website (http://kngforge.uwc.ac.za/). The problems encountered and suggestions made will inform a further process of refinement.

Understanding the requirements of the end users involves identifying an appropriate delivery system. For education institutions, where software and system maintenance is available, the prototype may be sited on a powerful server by a trained technician. Additional resources could be available from an accompanying Internet site, which might contain user feedback facilities, relevant links, and an online version with functionality and interactivity. The delivery will be implemented after achievement and testing of the whole system.

CHAPTER 4 ASSESSMENT METHOD ANALYSIS

This section will explore the assessment issues, including assessment principles, design and implementation. Three questions will be considered: what is to be gained from formative assessment? Why is formative assessment so important to inquiry learning? What does formative assessment involve? Then some examples will be given to explain how to improve test questions. Thus it will provide a set of suggestions for four types of questions. At last, a test module is assumed and model built.

Introduction

Teaching, tutoring and assessing students become radically different in a virtual learning environment than in a traditional classroom (Harasim, Hiltz etal, 1995; Hiltz and Wellman, 1997). The more activities that are 'Web-based,' the more technology-based supports are required for the educator. Computers free educators to spend less time on direct instruction, but do not diminish the role of the educator. Instead, they can change the role towards guiding and helping students to put information into context. Laurillard (1993) and Ramsden (1992) concur with this view in their discussion of the role of the educator. The process of guiding students is an ongoing struggle and requires thorough understanding of the learning process. For tutors and teaching assistants, Web-based learning environments introduce a whole new situation.

Learning and assessment are integrated in the doing process, which is expanded to cover the whole course. Formative assessment focuses on the learning process. Compare this to summative assessment, which is the grading of students' performance in relation to predefined goals. Formative assessment, on the other hand, is dialogue and feedback. Loss of interaction among students and educators is a serious potential risk, especially if the educators have no, or limited, feedback about the students' learning processes. Timely and individualised feedback is crucial for the learning process. Instructional software, to some extent, provides individualised support by making more advanced functionality available as the learner develops

expertise. Web-based learning environments have inherent characteristics, which affect the awareness of the learning processes.

Objective

On completion of this module, it will be possible:

- to understand the importance of research and development as a management decision-making tool,
- to analyse the assessment method, implementation and criteria,
- to understand the complete scope of the research methodology,
- to apply the research methodology in planning, conducting and implementing research projects.

4.1 Assessment practise principles

The following assessment principles must be adhered to:

- Assessment will be a central part of curriculum design for each module;
- Assessment will be designed to encourage meaningful learning by students;
- Assessment will be clearly and appropriately aligned with learning outcomes;
- Assessment design and practise will be appropriate to the diversity of students;
- Students will receive prompt, useful and clear feedback on their performance;
- Assessment design and practise will aim for fairness and consistency;
- Assessment practise and grading will be based on criteria;
- The nature of the assessment process, the assessment criteria and the way in which feedback will be given must be made clear to students at the start of each module;
- Assessment will be designed to avoid an overload on students and staff.

4.2 Assessments design

Assessments that are designed to contribute to grades (summative

assessments) will always include an element of formative feedback to students. Purely formative assessments of students' progress, which do not contribute to grades, will provide additional opportunities for feedback at appropriate points during a module.

Assessment will be based on identified learning outcomes, which express threshold levels of achievement. The actual levels of student achievement of the outcomes will be differentiated and judged by appropriate criteria, drawn up by the relevant teaching staff, in relation to the institution's grade criteria guide.

Assessment is linked to student learning. The way in which students approach their learning is determined by the means that will be used to assess them. Therefore assessment procedures should encourage learning and not simply measure it.

- The design of the instrument (test, written assignment, portfolio, practical demonstration and tasks, observation, interviews, combination of tools, etc.) is appropriate (meets the outcomes as specified in the standards, are at the right level, etc).
- The design of the instrument is based on information taken from relevant source documents (unit standards, and any other documents which prescribe what must be assessed and the criteria on which judgments will be made).
- The design of the instrument is linked to an appropriate assessment strategy (e.g. taking into account opportunities for integrated assessment, or for gathering naturally occurring evidence).
- Instructions to learners are clear and unambiguous.
- Time given for gathering and presentation of evidence (whether in one sitting or over time) is sufficient to allow an average learner to demonstrate competence.
- There is a relationship between course methodology and content and the assessment.
- Grading design (assessment criteria, issues of weighting, format of

judgments, etc.) is done concurrently with instrument design, and is compatible with the instrument.

- Assessment design includes the development of an assessment guide laying out details and instructions for the assessment activity. Explicit grading instructions are developed (for marking, recording on observation sheets, or evaluating a product such as a lesson plan or training event, etc.).
- The design makes provision for special needs without compromising the validity of the assessment.
- The assessment is implemental within any reasonable site costs and time requirements.

4.3 Assessment implementation

- Teachers are prepared for assessment, in relation to understanding what is expected of them, and an evaluation of their readiness.
- The application of the instrument (exam, sittings, interviews, observation sessions, portfolio evaluations, etc.) is carried out appropriately and fairly;
- The assessment environment is conducive to fair, valid and reliable assessment (e.g. it is physically acceptable; it is in line with recognised codes of practise and learning or work site standard operating procedures; where applicable, security measures such as invigilation procedures are in place to prevent irregularities).
- Adequate resources are provided (e.g. workplace equipment, resources such as chalk boards/ flip charts etc. for classroom-based demonstrations, and any other necessary technical resources as specified by the assessment requirements in the standards or qualifications).
- Where language may be a barrier to accurate assessment of the relevant competence, support procedures such as interpretation and translation should be in place. This includes provision for sign language and resources for non-sighted learners.
- Provision is made for accommodating any other special needs without compromising the validity of the assessment.

4.4 Research questions

4.4.1 What is to be gained from formative assessment?

Formative assessment is familiar in daily life outside the classroom. Those who are responsible for the work performance of others make on-the-spot judgments all the time, often by asking employees to propose a course of action or explain their thinking. In response, they make adjustments to the amount of support or training they provide. This is an ongoing process, not a formal assessment; improvement in performance is the goal. The same thing happens in schools: Teachers constantly have to make judgments about how well students are doing. Formative assessment provides a more formal structure for making these judgments, ensuring that decisions are based on carefully interpreted evidence. There are other benefits, as well. For instance:

- Formative assessment, which takes place within regular learning activities, takes into account a full range of skills and attitudes without undue dependence on reading and writing skills when these skills are not the ones being assessed. This is in contrast to most classroom tests, which focus on a narrow range of easily tested knowledge and which inevitably depend upon students' reading and writing abilities. Moreover, tests are rarely free of bias in relation to gender, language, and culture, as well as to test sophistication and test coaching.
- As already noted, improved formative assessment has the greatest effect on increasing the learning of lower-achieving students. Not only does this help in providing equal learning opportunities to all sections of the community, but it reduces special need placements as well.
- In tune with the practise of formative assessment, widely accepted theories of learning emphasise the role of the student in actively constructing understanding. An important part of the process of learning with understanding is linking new experiences, and the ideas used to make sense of them, to previous experiences. The integrated knowledge that results can then be applied in new situations. In contrast, knowledge that is not linked to a general framework of understandable concepts is

not readily applied beyond those situations in which it was learned and practised.

4.4.2 Why is formative assessment so important to inquiry learning?

The evidence and arguments just described show that, in a number of subjects, formative assessment can increase the attainment of all students and bring particular gains to lower-achieving students. In the case of learning science, however, there is an added reason for taking seriously the case for improving formative assessment practise, which follows from considering what students need to learn through inquiry.

It is the nature of learning through inquiry that understanding is built from existing ideas and experience. Inquiry teaching leads students to build their understanding of fundamental scientific ideas through direct experience with materials, by consulting existing resources, consulting with experts, and interacting and debating among themselves (National Science Foundation, Foundations: The Challenge and Promise of K–8 Science Education Reform, 1997). In order to promote the construction of understanding through inquiry, it is clear that students need the following:

- to have experiences that are within reach of their existing ideas and ways of thinking
- to have experiences that link to previous ones and to the ideas that help understanding
- to make their own ideas explicit to themselves as well as to the teacher
- to have access to other ideas through books, social interaction, teacher guidance, and the media
- to gather evidence by using inquiry skills to test their own and others' ideas
- to be in control of making sense of new experiences
- to reflect on how their ideas and skills have changed to engage in activities that they see as relevant, important, stimulating, and valued for themselves, rather than simply for their usefulness in passing tests and examinations.

4.4.3 What does formative assessment involve?

All assessment involves gathering evidence, interpreting it, and using the result in some way. How these things are done depends on the purpose of the assessment. Two of the main purposes are to help learning (assessment for learning) and to report on the learning of individual students (assessment of learning). These are the main concerns within this context, but the use of data about the performance of groups of students for a third purpose cannot be ignored-that of evaluating teaching and school target-setting, since this has an impact on classroom work.

4.5 Formative assessment cycle

Assessment for learning, or formative assessment, is a procedure for regulating teaching so that the pace of moving toward a goal is adjusted to ensure the active participation of the students. As with all regulated processes, feedback into the system is an important mechanism for ensuring effective operation. Thus feedback of information about learning helps ensure that new experiences are not too difficult or too easy for students. In the case of teaching, the feedback is given to both the teacher and the students. Figure 4 -1 describes the formative assessment cycle processes.

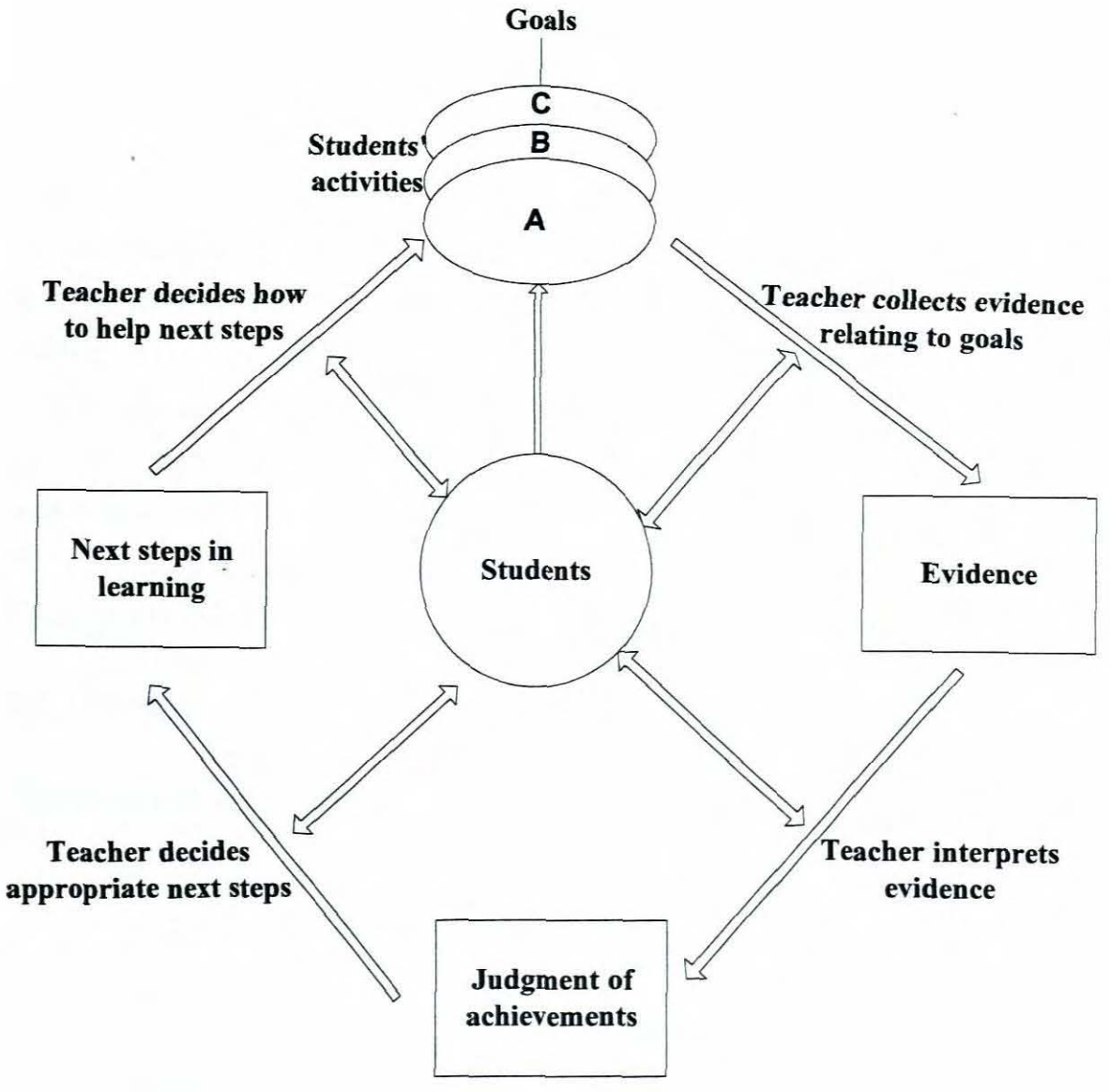


Figure 4-1: Formative assessment cycle diagram

The formative assessment cycle involves:

- using evidence for feedback in teaching;
- collecting evidence of students' thinking;
- gathering evidence from students' writing;
- discussing the purpose of activities and using examples;
- demonstrating what has been learned.

4.6 How to improve test questions

4.6.1 Choosing between objective and subjective test items

There are two general categories of test items: (1) objective items that require students to select the correct response from several alternatives or to supply a word or short phrase to answer a question or complete a statement; and (2) subjective or essay items which permit the student to organise and present an original answer. Objective items include multiple-choice, true/false, matching and completion, while subjective items include short-answer essay, extended-response essay, and problem solving and performance test items. For some instructional purposes one or the other item types may prove more efficient and appropriate.

4.6.2 When to use objective or essay tests?

Objective tests are especially appropriate when:

- the group to be tested is large and the test may be reused;
- highly reliable test scores must be obtained as efficiently as possible;
- impartiality of evaluation, absolute fairness, and freedom from possible test scoring influences (e.g., fatigue, lack of anonymity) are essential;
- teachers are more confident of their ability to express objective test items clearly than of their ability to judge essay test answers correctly;
- there is more pressure for speedy reporting of scores than for speedy test preparation.

Essay tests are especially appropriate when:

- the group to be tested is small and the test is not to be reused;
- it is appropriate to encourage and reward the development of student skill in writing;
- the educator is more interested in exploring the student's attitudes than in measuring his/her achievement;
- the teacher has greater ability as a critical and fair reader than as an imaginative writer of good objective test items.

Either essay or objective tests can be used to:

- measure almost any important educational achievement a written test can measure;
- test understanding and ability to apply principles;
- test ability to think critically;
- test ability to solve problems;
- test ability to select relevant facts and principles and to integrate them toward the solution of complex problems.

In addition to the preceding suggestions, it is important to realise that certain item types are better suited than others for measuring particular learning objectives. For example, learning objectives that require the student to demonstrate or to show may be better measured by performance test items, whereas objectives that require the student to explain or to describe may be better measured by essay test items. The matching of learning objective expectations with certain item types can help teachers select an appropriate kind of test item for their classroom exam examination, and provide a higher degree of test validity (i.e. testing what is supposed to be tested). To further illustrate this point, several sample learning objectives and appropriate test items are provided on the following page.

After deciding to use an objective, essay or both objective and essay examination, the next step is to select the kind(s) of objective or essay items that teachers wish to include in the examination. To help teachers make such a choice, the different kinds of objective and essay items are presented in the following section of this dissertation. The various kinds of items are briefly described and compared to one another in terms of their advantages and limitations for use. Also presented is a set of general suggestions for the construction of each item variation.

4.6.3 Multiple-choice test items

A "multiple choice question" is a question in which students are asked to select

one alternative from a given list of alternatives in response to a "question stem".

The multiple-choice item consists of two parts: (a) the stem, which identifies the question or problem and (b) the response alternatives. Students are asked to select the one alternative that best completes the statement or answers the question. For example,

Sample multiple-choice item

(a) Item stem: Which of the following is a chemical change?

(b) Response alternatives:

a. Evaporation of alcohol
b. Freezing of water
*c. Burning of oil
d. Melting of wax

*Correct response

Advantages in using multiple-choice items

Multiple-choice items can provide:

- versatility in measuring all levels of cognitive ability
- highly reliable test scores
- scoring efficiency and accuracy
- objective measurement of student achievement or ability
- a wide sampling of content or objectives
- a reduced guessing factor when compared to true-false items
- different response alternatives that can provide diagnostic feedback.

Limitations in Using Multiple-Choice Items

There are certain limitations in using multiple-choice questions. Multiple-choice items

- are difficult and time-consuming to construct;
- lead an instructor to favour simple recall of facts;

- place a high degree of dependence on the student's reading ability and the instructor's writing ability.

Suggestions for writing multiple-choice test items

I. When possible, state the stem as a direct question rather than as an incomplete statement. Present a definite, explicit and singular question or problem in the stem.

Undesirable:
Alloys are ordinarily produced by ...
Desirable:
How are allows ordinarily produced?

II. Eliminate excessive verbiage or irrelevant information from the stem.

Undesirable:
While ironing her dress, Jane burned her hand accidentally on the hot iron. This was due to a transfer of heat be ...
Desirable:
Which of the following ways of heat transfer explains why Jane's hand was burned after she touched a hot iron?

III. Include in the stem any word(s) that might otherwise be repeated in each alternative.

Undesirable:
In national elections in the United States the President is officially
a. chosen by the people.
b. chosen by members of Congress.
c. chosen by the House of Representatives.
*d. chosen by the Electoral College.
Desirable:

In national elections in the United States the President is officially chosen by

a. the people.

b. members of Congress.

c. the House of Representatives.

*d. the Electoral college.

IV. Use negatively stated stems sparingly. When used, underline and/or capitalise the negative word.

Undesirable:

Which of the following is not cited as an accomplishment of the Kennedy administration?

Desirable:

Which of the following is NOT cited as an accomplishment of the Kennedy administration?

V. Make the alternatives grammatically parallel with each other, and consistent with the stem.

Undesirable:

What would do most to advance the application of atomic discoveries to medicine?

*a. Standardised techniques for treatment of patients.

b Train the average doctor to apply radioactive treatments.

c. Remove the restriction on the use of radioactive substances.

d. Establishing hospitals staffed by highly trained radioactive therapy specialists.

Desirable:

What would do most to advance the application of atomic discoveries to medicine?

*a. Development of standardised techniques for treatment of patients.

b. Training of the average doctor in application of radioactive treatments.

What would do most to advance the application of atomic discoveries to medicine?

*a. Development of standardised techniques for treatment of patients.

b. Training of the average doctor in application of radioactive treatments.

c. Removal of restriction on the use of radioactive substances.

d. Addition of trained radioactive therapy specialists to hospital staffs.

VI. Make the alternatives mutually exclusive.

Undesirable:

The daily minimum required amount of milk that a 10-year-old child should drink is

a. 1-2 glasses

*b. 2-3 glasses

*c. 3-4 glasses

d. at least 4 glasses

Desirable:

What is the daily minimum required amount of milk a 10-years-old child should drink?

a. 1 glass

b. 2 glasses

*c. 3 glasses

d. 4 glasses

VII. Be sure there is only one correct or best response to the item.

Undesirable:

The two most desired characteristics in a classroom test are validity and

a. Precision

*b. Reliability

c. Objectivity

c. Objectivity

*d. consistency

Desirable:

The two most desired characteristics in a classroom test are validity and

a. Precision

*b. Reliability

c. Objectivity

d. standardisation

VIII. Make alternatives approximately equal in length.

Undesirable:

The most general cause of low individual incomes in the United States is

*a. a lack of valuable productive services to sell.

b. unwillingness to work.

c. automation.

d. inflation.

Desirable:

What is the most general cause of low individual incomes in the United States?

*a. A lack of valuable productive services to sell.

b. the population's overall unwillingness to work.

c. the nation's increased reliance on automation.

d. an increasing national level of inflation.

IX. Avoid irrelevant clues such as grammatical structure, well-known verbal associations or connections between stem and answer.

X. Use at least four alternatives for each item to lower the probability of getting the item correct by guessing.

XI. Randomly distribute the correct response among the alternative positions throughout the test having approximately the same proportion of alternatives a, b, c, d and e as the correct response.

XII. Use the alternatives "none of the above" and "all of the above" sparingly. When used, such alternatives should occasionally be used as the correct response.

4.6.4 True/False test items

A true/false item can be written in one of three forms: simple, complex, or compound. Answers can consist of only two choices (simple), more than two choices (complex), or two choices plus a conditional completion response (compound). An example of each type of true-false item follows:

Sample true-false item:

The acquisition of morality is a developmental process. True False

Advantages in using true/false Items

True/False items can provide:

- the widest sampling of content or objectives per unit of testing time.
- scoring efficiency and accuracy.
- versatility in measuring all levels of cognitive ability.
- highly reliable test scores.
- an objective measurement of student achievement or ability.

Limitations in using true-false items

There are various limitations involved when true/false items are employed as test questions. The following should be taken into account:

- They incorporate an extremely high guessing factor. For simple true-false items, each student has a 50/50 chance of correctly answering the item without any knowledge of the item's content.
- They can often lead an instructor to write ambiguous statements, as it is very difficult to write statements that are unequivocally true or false.

other item types.

- They can often include more irrelevant clues than do other item types.
- They can often lead an instructor to favour testing of trivial knowledge.

Suggestions for writing true/false test items

I. Base true/false items upon statements that are absolutely true or false, *without qualifications or exceptions.*

Undesirable:
Nearsightedness is hereditary in origin.
Desirable:
Geneticists and eye specialists believe that the predisposition to nearsightedness is hereditary.

II. Express the item statement as simply and as clearly as possible.

Undesirable:
When you see a highway with a marker that reads, "Interstate 80" you know that the construction and upkeep of that road is built and maintained by the state and federal government.
Desirable:
Both state and federal governments provide the construction and maintenance of interstate highways.

III. Express a single idea in each test item.

Undesirable:
Water will boil at a higher temperature if the atmospheric pressure on its surface is increased and more heat is applied to the container.
Desirable:
Water will boil at a higher temperature if the atmospheric pressure on its surface is increased.
And/or

Water will boil at a higher temperature if more heat is applied to the container.

IV. Include enough background information and qualifications so that the ability to respond correctly to the item does not depend on some special, uncommon knowledge.

Undesirable:
The second principle of education is that the individual gathers knowledge.
Desirable:
According to John Dewey, the second principle of education is that the individual gathers knowledge.

V. Avoid lifting statements from the text, lecture or other materials so that *memory alone will not permit a correct answer.*

Undesirable:
For every action there is an opposite and equal reaction.
Desirable:
If you were to stand in a canoe and throw a life jacket forward to another canoe, chances are your canoe would jerk backward.

VI. Avoid using negatively stated item statements.

Undesirable:
The supreme court is not composed of nine justices.
Desirable:
The supreme court is composed of nine justices.

VII. Avoid the use of unfamiliar vocabulary

Undesirable:
According to some politicians, the raison d'etre for capital punishment is retribution.

Desirable:

According to some politicians, justification for the existence of capital punishment is retribution.

VIII. Avoid the use of specific determiners that would permit a test-wise but unprepared examinee to respond correctly. Specific determiners refer to sweeping terms like "all," "always," "none," "never," "impossible," "inevitable," etc. Statements including such terms are likely to be false. On the other hand, statements using qualifying determiners such as "usually," "sometimes," "often," etc., are likely to be true. When statements do require the use of specific determiners, make sure they appear in both true and false items.

IX. False items tend to discriminate more highly than true items. Therefore, use more false items than true items (but no more than 15% additional false items).

4.6.5 Matching test items

In general, matching items consist of a column of stimuli presented on the left side of the examination page and a column of responses placed on the right side of the page. Students are required to match the response associated with a given stimulus.

Advantages in using matching items

- They require short periods of reading and response time, allowing to cover more content.
- They provide objective measurement of student achievement or ability.
- They provide highly reliable test scores.
- They provide scoring efficiency and accuracy.

Limitations in using matching items

- It is difficult to measure learning objectives requiring more than simple recall of information.
- They are difficult to construct due to the problem of selecting a

common set of stimuli and responses.

Suggestions for writing matching test items

I. Include directions that clearly state the basis for matching the stimuli with the responses. Explain whether or not a response can be used more than once and indicate where to write the answer.

Undesirable:

Directions - Match the following.

Desirable:

Directions - On the line to the left of each identifying location and characteristics in Column I, write the letter of the country in Column II that is best defined. Each country in Column II may be used more than once.

II. Use only homogeneous material in matching items.

Undesirable:

Directions: - Match the following.

1. Water	A. NaCl
2. Discovered Radium	B. Fermi
3. Salt	C. NH3
4. Year of the 1st Nuclear Fission by Man	D. H_2O
5. Ammonia	E. 1942
	F. Curie

Desirable:

Directions - On the line to the left of each compound in Column I, write the letter of the compound's formula presented in Column II. Use each formula only once.

Column I	Column II
1. Water	A. H_2SO_4
2. Salt	B. HCl

3. Ammonia C NaCLI

4. Sulphuric Acid D. H_2O

III. Avoid grammatical or other clues to the correct response.

Undesirable:

Directions - Match the following in order to complete the sentences on the left.

1. igneous rocks are formed A. a hardness of 7.
2. the formation of coal requires B. with crystalline rock.
3. a geode is filled C. A metamorphic rock.
4. feldspar is classified as D. heat and pressure.
E. through the solidification of molten lava.

Desirable:

Avoid sentence completion due to grammatical clues.

IV. Keep matching items brief, limiting the list of stimuli to fewer than 10.

V. Include more responses than stimuli to help prevent answering through the process of elimination.

VI. When possible, reduce the amount of reading time by including only short phrases or single words in the response list.

4.6.6 Completion (Fill-in blank) test items

The completion item requires the student to answer a question or to finish an incomplete statement by filling in a blank with the correct word or phrase. Examples are supplied below.

Sample completion item

According to Freud, personality is made up of three major systems, the _____, the _____ and the _____.

Advantages in using completion items

- They can provide a wide sampling of content.

- o They can efficiently measure lower levels of cognitive ability.
- o They can minimise guessing as compared to multiple-choice or true-false items.
- o They can usually provide an objective measure of student achievement or ability.

Limitations in using completion items

Completion items

- o are difficult to construct so that the desired response is clearly indicated;
- o have difficulty measuring learning objectives requiring more than simple recall of information;
- o can often include more irrelevant clues than do other item types;
- o are probably more time-consuming when compared to multiple-choice or true-false items;
- o are more difficult to score since more than one answer may have to be considered correct if the item was not properly prepared.

Suggestions for writing completion test items

I. Omit only significant words from the statement.

Undesirable:

Every atom has a central (core) called a nucleus.

Desirable:

Every atom has a central core called a (n) (nucleus).

II. Do not omit so many words from the statement that the intended meaning is lost.

Undesirable:

The _____were to Egypt as the_____were to Persia and as _____were to the early tribes of Israel.

Desirable:

The Pharaohs were to Egypt as the_____were to Persia and as _____were to the early tribes of Israel.

III. Avoid grammatical or other clues to the correct response.

Undesirable:
Most of the United States' libraries are organised according to the (Dewey) decimal system.
Desirable:
Which organisational system is used by most of the United States' libraries? (Dewey decimal)

IV. Be sure there is only one correct response.

Undesirable:
Trees that shed their leaves annually are seed-bearing, common).
Desirable:
Trees that shed their leaves annually are called (deciduous).

V. When possible, delete words at the end of the statement after the student has been presented a clearly defined problem.

Undesirable:
(122.5) is the molecular weight of $KClO_3$.
Desirable:
The molecular weight of $KClO_3$ is (122.5).

VI. Avoid lifting statements directly from the text, lecture or other sources.
VII. Limit the required response to a single word or phrase.

4.7 Two methods for assessing test item quality

This section presents two methods for collecting feedback on the quality of test items. The two methods include using self-review checklists and student

evaluation of test item quality. Teachers can use the information gathered from either method to identify strengths and weaknesses in item writing. Evaluate teachers test items by checking the suggestions that follow.

Multiple-choice test items

- When possible, state the stem as a direct question rather than as an incomplete statement.
- Present a definite, explicit and singular question or problem in the stem.
- Eliminate excessive verbiage or irrelevant information from the stem.
- Include in the stem any word(s) that might otherwise have been repeated in each alternative.
- Use negatively stated stems sparingly. When used, underline and/or capitalised the negative word(s).
- Make all alternatives plausible and attractive to the less knowledgeable or less skilful student.
- Make the alternatives grammatically parallel with each other, and consistent with the stem.
- Make the alternatives mutually exclusive.
- When possible, present alternatives in some logical order (e.g. chronologically, most to least).
- Make sure there was only one correct or best response per item.
- Make alternatives approximately equal in length.
- Avoid irrelevant clues such as grammatical structure, well-known verbal associations or connections between stem and answer.
- Use at least four alternatives for each item.
- Randomly distribute the correct response among the alternative positions throughout the test having approximately the same proportion of alternatives a, b, c, d, and e as the correct response.

True-False test items

- Base true-false items upon statements that are absolutely true or false, without qualifications or exceptions.

- Express the item statement as simply and as clearly as possible.
- Express a single idea in each test item.
- Include enough background information and qualifications so that the ability to respond correctly does not depend on some special, uncommon knowledge.
- Avoid lifting statements from the text, lecture or other materials.
- Avoid using negatively stated item statements.
- Avoid the use of unfamiliar language.
- Avoid the use of specific determiners such as "all," "always," "none," "never," etc., and qualifying determiners such as "usually," "sometimes," "often," etc.
- Use more false items than true items (but not more than 15% additional false items).

Matching test items

- Include directions that clearly state the basis for matching the stimuli with the response.
- Explain whether or not a response could be used more than once and indicate where to write the answer.
- Use only homogeneous material.
- When possible, arrange the list of responses in some systematic order (e.g. chronologically, alphabetically).
- Avoid grammatical or other clues to the correct response.
- Keep items brief (limit the list of stimuli to under 10).
- Include more responses than stimuli.
- When possible, reduce the amount of reading time by including only short phrases or single words in the response list.

Completion test items

- Omit only significant words from the statement.
- Do not omit so many words from the statement that the intended *meaning* is lost.
- Avoid grammatical or other clues to the correct response.

- Include only one correct response per item.
- Make the blanks of equal length.
- When possible, delete the words at the end of the statement after the student has been presented with a clearly defined problem.

4.8 Exposition of a test module

4.8.1 System overview

Students' shell — Students would use this to access the system.

Teachers' shell — Teachers would use this to access the system.

Question bank — All questions should be stored here.

User records — The user information and students' scores should be stored here.

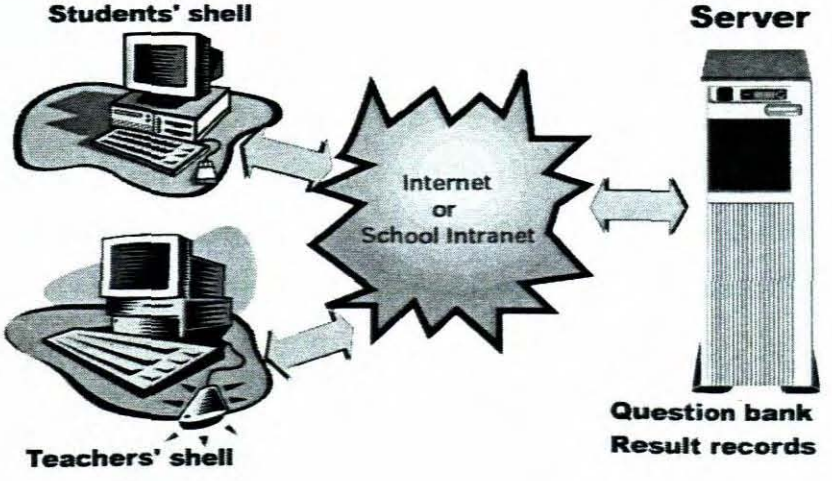


Figure 4-2: A Test module overview

4.8.2 Teachers' shell

A test module should allow teachers to create and manage online tests with full account access to the test module. Teachers could use the system on the Web. Once a teacher has logged into the system, a list of courses the teachers was teaching would be displayed. By choosing a course and a unit in the chosen course, he or she could create new exercise questions or modify existing questions. The Teacher could also ask the system to generate reports on the students' performance. The system should allow for constantly providing

feedback to students on their marks. The full reports on the students' performances should be kept in a text file for grading purposes or future reference. The programme should also keep track of the time remaining for the test, and would stop the test once the time was up.

4.8.3 Students' shell

A test module system should also provide some services for students. They could do the exercises at any time and anywhere as long as students were connected to the Web. They would see a list of all exercises from all courses they are learning. They could choose which exercise to do first. The exercise questions should be interactive and interesting. Hints might appear if their answer was not correct. They could see their score and know their performance.

4.8.4 *Question bank*

The exercise questions could be stored in the question bank. They could be organised into courses. Each course should be divided into a number of units. There could be four types of questions. A question stored in the question bank could be used in many papers. Each paper might contain questions from a single unit, or contain questions from many units.

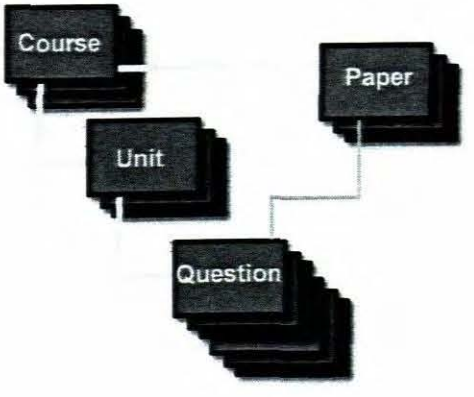


Figure 4-3: Question bank

A test module system should have two kinds of papers:

- Exercises — students could attempt questions in an exercise as many

times as they liked. The system would give feedback to tell the student whether the answer was correct. No record would be kept by the system.

- Tests — The system would not tell the student whether the answer was correct. It could keep the record of individual students.

4.8.5 User records

User records should contain information about individual users, classes and groups, as well as the scores of the students. The system should classify the users into three kinds: students, teachers and administrators. Each student would belong to a class, and he or she could not be in more than one class. One or more teachers could be designated to a class. One teacher might be assigned to more than one class. A user group could be created for a special purpose, for example, giving extra attention to weak students. A group could take students from any class. One or more teachers could be assigned to a group.

CHAPTER 5 OPEN SOURCE SOFTWARE TECHNOLOGY APPLICATION

This section will exactly describe how to use open source software - Apache, PHP, MySQL and XAMPP, as well as how the team members coordinated code changes using CVS. Finally, Object-Oriented Programming in PHP will be introduced. This will provide foundation for writing coding of the test module.

5.1 Apache

5.1.1 Using Apache with Microsoft Windows

Information on the latest versions of Apache can be found on the Web-site of the Apache Web server at http:// apr.apache.org/download.cgi. The current release, as well as more recent alpha or beta test versions, and a list of HTTP and FTP mirrors can be found there. The Apache Web server can be downloaded from there.

For Windows installations the version of Apache for Windows with the *.msi* extension should be downloaded. This is a single Microsoft Installer file, which contains a ready-to-run version of Apache. There is a separate .zip file, which contains only the source code. Developers can compile Apache themselves with the Microsoft Visual C++ (Visual Studio) tools. The Apache *.msi* file downloaded above must be run. The installation will ask for the following:

I. Network Domain. Enter the DNS domain in which server is or will be registered. For example, if server's full DNS name is *server.mydomain.net*, type *mydomain.net* here.
II. Server Name. Server's full DNS name. From the example above, type *server.mydomain.net* here.
III. Administrator's e-mail address. Enter the server administrator's or Webmaster's e-mail address here. This address will be displayed along with error messages to the client by default.

IV. For whom to install Apache Select for All Users, on Port 80, as a Service - Recommended if new Apache to listen at port 80 for incoming traffic. It will run as a service (that is, Apache will run even if no one is logged in on the server at the moment) Select only for the Current User, on Port 8080, when started Manually if Apache is to be installed for personal experimenting or if another WWW server is already running on port 80.

V. The installation type. Select Typical for everything except the source code and libraries for module development. With custom one can specify what to install. A full install will require about 13 megabytes of free disk space. This does *not* include the size of the Web site(s).

VI. Where to install. The default path is *C:\Programme Files\Apache Group* under which a directory called Apache2 will be created by default.

During the installation, Apache will configure the files in the *conf* subdirectory to reflect the chosen installation directory. However, if any of the configuration files in this directory already exist, they will not be overwritten. Instead, the new copy of the corresponding file will be left with the extension *.default*. So, for example, if *conf\httpd.conf* already exists, it will be renamed as *conf\httpd.conf.default*. After the installation one should manually check to see what new settings are in the *.default* file, and if necessary, update existing configuration file. Also, if there is already a file called *htdocs\index.html*, it will not be overwritten (and no *index.html.default* will be installed either). This means it should be safe to install Apache over an existing installation, although one would have to stop the existing running server before doing the installation, and then start the new one after the installation is finished.

After installing Apache, one must edit the configuration files in the *conf* subdirectory as required. These files will be configured during the installation so that Apache is ready to be run from the directory it was installed into, with the documents served from the subdirectory *htdocs*. There are many of other options which we should set before we really start using Apache. However, to get started quickly, the files should work as installed.

5.1.2 Running apache as a service

One can install Apache as a service automatically during the installation. If one chooses to install for all users, the installation will create an Apache service. If one specifies to install for oneself only, one can manually register Apache as a service after the installation. One has to be a member of the Administrators group for the service installation to succeed. Apache comes with a utility called the Apache Service Monitor. With it one can see and manage the state of all installed Apache services on any machine on the network. To be able to manage an Apache service with the monitor, one first has to install the service (either automatically via the installation or manually). One can install Apache as a Windows NT service as follows from the command prompt at the Apache *bin* subdirectory:

apache -k install

If one needs to specify the name of the service one wants to install, the following command can be used. This has to be done if there are several different service installations of Apache on the computer.

apache -k install -n "MyServiceName"

If specifically named configuration files for different services are required, this must be used:

apache -k install -n "MyServiceName" -f "c:\files\my.conf"

If one uses the first command without any special parameters except -k install, the service will be called Apache2 and the configuration will be assumed to be *conf\httpd.conf*.

Removing an Apache service is easy. Just use:

apache -k uninstall

The specific Apache service to be uninstalled can be specified by using:

apache -k uninstall -n "MyServiceName"

Normal starting, restarting and shutting down of an Apache service is usually done via the Apache Service Monitor, by using commands like *NET START Apache2* and *NET STOP Apache2* or via normal Windows service management. Before starting Apache as a service by any means, the service's configuration file should be tested by using:

apache -n "MyServiceName" -t

One can control an Apache service by its command line switches as well. To start an installed Apache service one would use this:

apache -k start

To stop an Apache service via the command line switches, one would use this:

apache -k stop

or

apache -k shutdown

One can also restart a running service and force it to reread its configuration file by using:

apache -k restart

By default, all Apache services are registered to run as the system user (the LocalSystem account). The LocalSystem account has no privileges to our network via any Windows-secured mechanism, including the file system, named pipes, DCOM, or secure RPC. It has, however, wide privileges locally.

5.1.3 Testing the installation

After starting Apache, it will be listening on port 80. To connect to the server and access the default page, launch a browser and enter this URL:

http://localhost/

Apache should respond with a welcome page and a link to the Apache manual. If nothing happens or an error is indicated, look in the *error.log* file in the *logs* subdirectory. If the host is not connected to the net, or if there are serious problems with the DNS (Domain Name Service) configuration, One may have to use this URL:

http://127.0.0.1/

Once basic installation is working, one should configure it properly by editing the files in the *conf* subdirectory. Again, if one changes the configuration of the Windows NT service for Apache, one should first attempt to start it from the command line to make sure that the service starts with no errors.Because Apache cannot share the same port with another TCP/IP application, one may need to stop, uninstall or reconfigure certain other services before running Apache. These conflicting services include other WWW servers and some firewall implementations. Develop Zone: http://www.apache.org/

5.2 MySQL

5.2.1 Choose which distribution to install.

Several versions of MySQL are available, and most are available in several distribution formats. One can choose from pre-packaged distributions containing binary programmes or source code. When in doubt, use a binary distribution. MySQL development occurs in several release series, and one can pick the one that best fits one's needs. After deciding which version to install, one can choose a distribution format. Releases are available in binary or source format.

MySQL 5.0 is the newest development release series and is under very active development for new features. Until recently it was available only in preview form from the BitKeeper source repository. An early alpha release has now been issued to allow more widespread testing. MySQL 4.1 is a development release series to which major new features have been added. It is still at beta status. Sources and binaries are available for use and testing on development systems. MySQL 4.0 is the current stable (production-quality) release series. New releases are issued for bugfixes. No new features are added that could diminish the code stability. MySQL 3.23 is the old stable (production-quality) release series. This series is retired, so new releases are issued only to fix critical bugs.

5.2.2 Download the distribution that needs to be installed

Check the MySQL home page (http://www.mysql.com/) for information about the current version and for downloading instructions. For a complete up-to-date list of MySQL download mirror sites, see http://dev.mysql.com/downloads/mirrors.html.

5.2.3 Install the distribution

The installation process for MySQL on Windows has the following steps: Obtain and install the distribution; set up an option file if necessary; select the

server that will be used; start the server; assign passwords to the initial MySQL accounts. MySQL for Windows is available in two distribution formats:

- *The binary distribution contains a setup programme that installs* everything so that the server can be started immediately.
- The source distribution contains all the code and support files for building the executables using the VC++ 6.0 compiler.

Generally speaking, one should use the binary distribution. It is simpler, and one needs no additional tools to get MySQL up and running. Windows System Requirements:To run MySQL on Windows, One needs the following:

- A 32-bit Windows operating system such as 9x, Me, NT, 2000, or XP. The NT family (Windows NT, 2000, and XP) permits to run the MySQL server as a service.
- TCP/IP protocol support.
- A copy of the MySQL binary distribution for Windows.
- WinZip or other Windows tool that can read `.zip' files, to unpack the distribution file.
- Enough space on the hard drive to unpack, install, and create the databases in accordance with one's requirements.

To install MySQL on Windows using a binary distribution, follow this procedure: If working on a Windows 2000, make sure that one has logged in as a user with administrator privileges.

- Exit the *WinMySQLAdmin* programme if it is running.
- Unzip the distribution file to a temporary directory.
- Run the *setup.exe* programme to begin the installation process.
- Finish the installation process.

5.2.4 Preparing the windows MySQL environment

If one needs to specify startup options when one runs the server, one can indicate them on the command line or place them in an option file. For options that will be used every time the server starts, one will find it most convenient to use an option file to specify the MySQL configuration. This is true particularly

under the following circumstances:

- The installation or data directory locations are different from the default locations (*`C:\mysql'* and *`C:\mysql\data'*).
- the server settings have to be tuned.

On Windows, the MySQL installer places the data directory directly under the directory where one installs MySQL. When the MySQL server starts on Windows, it looks for options in two files: the `my.ini' file in the Windows directory, and the *`C:\my.cnf'* file. The Windows directory typically is named something like *`C:\WINDOWS'* or *`C:\WinNT'*. MySQL looks for options first in the *`my.ini'* file, then in the *`my.cnf'* file. However, to avoid confusion, it is best if only one file is used. An option file can be created and modified with any text editor, such as the Notepad programme. Another way to manage an option file is to use the *WinMySQLAdmin* tool. One can find WinMySQLAdmin in the *`bin'* directory of MySQL installation, as well as a help file containing instructions for using it. *WinMySQLAdmin* has the capability of editing option file. However, note these points:

- *WinMySQLAdmin* uses only the *`my.ini' file*.
- If *WinMySQLAdmin* finds a *`C:\my.cnf'* file, it will in fact rename it to *`C:\my_cnf.bak'* to disable it.

Now one is ready to start the server.

Note:If winmysqladmin does not diplay the server status, then one has to change at *mysql.ini*:

[WinMySQLadmin]

user=myadminuser (e.g. root)

password=myadminpassword

The user needs root-access to mysql-server.

5.2.5 Perform any necessary post-installation setup

After installing MySQL, there are some issues that need to be addressed. For example, on Unix, one should initialise the data directory and create the MySQL grant tables. On all platforms, an important security concern is that the initial accounts in the grant tables have no passwords. One should assign

passwords to prevent unauthorised access to the MySQL server. The problems start the server. Some possible steps are the following:

- Specify any special options needed by the storage engines one is using.
- Make sure that the server knows where to find the data directory.
- Make sure the server can use the data directory. The ownership and permissions of the data directory and its contents must be set such that the server can access and modify them.
- Check the error log to see why the server does not start.
- Verify that the network interfaces the server wants to use are available.

5.2.6 If one wants to run the MySQL benchmark scripts, Perl support for MySQL must be available

Perl support for MySQL is provided by means of the *DBI/DBD* client interface. The interface requires Perl Version 5.6.0 or later. An older version of Perl will not work . Using the MySQL 4.1 client library, *DBD::mysql* 2.9003 or newer must be used. On Windows, the following should be done to install the MySQL DBD module with ActiveState Perl:

- Get ActiveState Perl from http://www.activestate.com/Products/ActivePerl/ and install it.
- Open a console window (a ``DOS window").
- If required, set the HTTP_proxy variable.

  set HTTP_proxy=my.proxy.com:3128
- Start the PPM programme:

  C:\> C:\perl\bin\ppm.pl
- If not already done, install DBI:

  ppm> install DBI
- If this succeeds, run the following command:

  *install *

  ftp://ftp.de.uu.net/pub/CPAN/authors/id/JWIED/DBD-mysql-1.2212.x86.ppd

This procedure should work at least with ActiveState Perl Version 5.6. Develop Zone: http://dev.mysql.com

5.3 XAMPP (software package install apache distribution containing MySQL, PHP and Perl)

As previously stated, the own experience has shown that it is not easy to install an Apache Web server and it becomes more difficult if one wants to add MySQL, PHP and Perl. This is the general experience of many people. After a great of research, Apache Friends was found. Apache Friends is a non-profit project to promote the Apache Web server. Kai 'Oswald' Seidler and Kay Vogelgesang founded it in the spring of 2002. They contributed toward promotions the Web server and related technologies. Apache Friends started as a project concerned solely with XAMPP. XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is very easy to install and to use - just download, extract and start. There are no changes to the Windows registry and it is not necessary to edit any configuration files. The philosophy behind XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers XAMPP is configured with all features turned on. The default configuration is not good from a security point of view and it is not secure enough for a production environment. XAMPP is a compilation of free software (comparable to a Linux distribution), free of charge and free to copy under the terms of the GNU General Public Licence. It is only the compilation of XAMPP that is published under GPL.

The distribution for Windows 98, NT, 2000 and XP contains: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System for Win32 and NetWare Systems v3.32, JpGraph, FileZilla FTP Server, mcrypt, Turck MMCache, SQLite, and WEB-DAV + mod_auth_mysql. Download XAMPP:
http://www.apachefriends.org/en/xampp-windows.html

Apache Friends XAMPP for windows Version 1.3 install

+ Apache 2.0.48
+ MySQL 4.0.17
+ PHP 4.3.4 + PEAR

+ mod_php 4.3.4
+ Perl 5.8.2
+ mod_perl 1.99_12
+ mod_ssl 2.0.48
+ openssl 0.9.7c
+ SQLite 2.8.11 + mod_auth_mysql (only experimental)

+ PHPMyAdmin 2.5.5 pl1
+ Webalizer 2.01-10
+ Mercury Mail Transport System for Win32 and NetWare Systems v3.32
+ JpGraph 1.14 (only for non-commercial, open-source and educational use)
+ FileZilla FTP Server 0.8.8
+ Oswalds CD Collection v0.2 (!)
+ PHP mcrypt() 2.4
+ Turck MMCache 2.4.6 (PHP accelerator, optimiser, encoder and dynamic content cache for PHP)
+ ADODB 4.04
+ PHPBlender 0.9 (only NT systems PHP compiler)

System Requirements:

+ 64 MB RAM (recommended)
+ 175 MB free Fixed Disk
+ Windows 98, ME, XP Home
+ Windows NT, 2000, XP Professional (Recommended)

5.3.1 Installation

Step 1: Unpack the package into a directory of choice(C:\).

Step 2: Please start "setup_xampp.bat". Enter the number 1 to begin the installation with MOD_PERL. Without MOD_PERL use number2. MOD_PERL in this Configuration delaying START and STOP of the Apache Server.

Step 3: If installation ends successfully, start the Apache 2 with

"apache_start".bat", MySQL with "mysql_start".bat". Stop the MySQL Server with "mysql_stop.bat". For shutdown the Apache HTTPD, only close the Apache Command (CMD).

Step 4: Start browser and type http://127.0.0.1 or http://localhost in the location bar. One should see their pre-made start page with certain examples and test screens.

Step 5: PHP (with mod_php, as *.php, *.php4, *.php3, *.phtml), Perl by default with *.cgi, SSI with *.shtml are all located in

=> \...*xampp\htdocs*\.

Examples:

=> \...*xampp\htdocs\test.php => http://localhost/test.php*

=> \...*xampp\myhome\test.php => http://localhost/myhome/test.php*

Step 6: Perl with mod_perl will execute in

=> \...*xampp\htdocs\modperl*

Test is with http://localhost/perl/modperl.pl

Perl:ASP will execute in

=> \...*xampp\htdocs\modperlasp*

Test is with http://localhost/perlasp/loop.asp

Step 7: Start Mercury Mail Server with

=> \...*xampp\mercury_start.bat*

Step 8: Configure FileZilla FTP Server with

=> \...*xampp\filezilla_setup.bat*

In the "FileZilla Server.xml", please change all relative paths into absolute paths

=> \...*xampp\FileZillaFTP\FileZilla Server.xml*

Step 9: XAMPP UNINSTALL? Simply remove the "xampp" Directory.

Passwords

1) MySQL

User: root

Password: (means no password!)

2) FileZilla FTP

User: newuser

Password: wampp

User: anonymous

Password: some@mail.net

3) Mercury:

Postmaster: postmaster (postmaster@localhost) und Admin (Admin@localhost)

Testuser: newsuser

Password: wampp

4) WEBDAV:

User: wampp

Password: xampp

ONLY FOR NT SYSTEMS (NT4 | windows 2000 | windows xp professional)

\...\xampp\apache\apache_installservice.bat

==> Install Apache 2 as service

\...\xampp\apache\apache_uninstallservice.bat

==> Uninstall Apache 2 as service

\...\xampp\apache\mysql_installservice.bat

==> Install MySQL as service

Before, please copy the my.cnf with absolute paths to c:\ (c:\my.cnf)!

\...\xampp\apache\mysql_uninstallservice.bat

==> Uninstall MySQL as service

\...\xampp\filezilla_setup.bat

==> Install or uninstall FileZilla FTP Server as service

==> After all Service (un)installations, it is better to restart the system.

5.3.2 Apache notes

(1) In contrast to apache 1.x, the apache2 cannot be stopped with the command "apache -k shutdown". These functions are only for installations that are serviced by NT systems. Simply close the Apache START command for shutdown.

(2) To use the experimental version of Mod_Dav load the *Modules mod_dav.so + mod_dav_fs.so* in the *httpd.conf* by removing the # on the beginning of their lines. Then try http://127.0.0.1:81 (not for FrontPage, but for Dreamweaver).

(3) To use the experimental version of mod_auth_mysql remove the # in the *httpd.conf.* Detailed information about this topic can be found on the left menu of xampp.

5.3.3 MySQL notes

(1) The MySQL server can be started by double-clicking (executing) mysql_start.bat. This file can be found in the same folder we installed xampp in; most likely this will be C:\xampp\. The exact path to this file is X:\xampp\mysql_start.bat, where "X" indicates the letter of the drive us unpacked xampp into. This batch file starts the MySQL server in console mode. The first initialization might take a few minutes.

To stop the server, use mysql_shutdown.bat, which is located in the same directory.

(2) To use the MySQL Daemon with innodb for better performance, edit the "my" (or my.cnf) file in the /xampp/mysql/bin directory or for services the c:\my.cnf for windows NT/2000. In there, activate the innodb_data_file_path=ibdata1:30M statement. Attention, innodb is not

recommended for 95/98/ME/XP Home.

To use MySQL as Service for NT/2000/XP Professional, simply copy the "my" / my.cnf file to C:\my, or C:\my.cnf. Note that this file has to be placed in C:\ (root), other locations are not permitted. Then execute the mysql_installservice.bat in the mysql folder.

(3) MySQL starts with standard values for the user id and the password. The preset user id is root, the password is "" (= no password). To access MySQL via PHP with the preset values, have to use the following syntax:

mysql_connect("localhost","root","");

If one wants to set a password for MySQL access, one must use of mysqladmin. To set the password "secret" for the user "root", type the following:

C:\xammpp\mysql\bin\mysqladmin –u root password secret

After changing the password one has to reconfigure PHPMyAdmin to use the new password, otherwise it will not be able to access the databases. To do that, open the file config.inc.php in \xampp\phpmyadmin\ and edit the following lines:

```
$cfg['Servers'][$i]['user']          = 'root';    // MySQL user
$cfg['Servers'][$i]['auth_type']     = 'http';    // HTTP authentificate
```

First the MySQL server queries the 'root' password before PHPMyAdmin may access.

The PHPMyAdmin is a MySQL database administration tool. It is a tool written in PHP intended to handle the administration of MySQL over the Web. Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage keys on fields, manage privileges, export data into various formats and is available in 47 languages, GLP Licence information.

The software is very useful. Extra release windows XAMPP 1.4.6 with PHP 5.0.1 was published on August 16th 2004. See PHPMyAdmin at: http://www.phpmyadmin.net/phpMyAdmin/

5.4 Using CVS

CVS is the Concurrent Versions System. Using it, a developer can record the history of source files. For example, bugs sometimes creep in when software is modified, and the developer might not detect the bug until a long time after the modification has been made. With CVS, he/she can easily retrieve old versions to see exactly which change caused the bug. This can sometimes be a big help. CVS is the dominant open-source network-transparent version control system. It is useful for everyone from individual developers to large, distributed teams:

- Its client-server access method allows developers to access the latest code from anywhere there is an Internet connection.
- Its unreserved checkout model to version control avoids artificial conflicts common with the exclusive checkout model.
- Its client tools are available on most platforms.

CVS maintains a history of a source tree, in terms of a series of changes. It stamps each change with the time it was made and the user name of the person who made it. Usually, the person provides some text describing why the change was made. Given that information, CVS can help developers answer questions like:

- Who made a given change?
- When did they make it?
- Why did they make it?
- What other changes did they make at the same time?

A version control system keeps a history of the changes made to a set of files. For a developer, that means being able to keep track of all the changes he/she has made to a programme during the entire time he/she has been developing it.

Code central station: Individual developers who want the safety net of a version control system can run one on their local machines. Development teams, however, need a central server all members can access to serve as the repository for their code. In an office, that dose not present a problem, as the repository can be placed on a server on the local network. For open source projects, it is still no problem, thanks to the Internet. CVS has built-in client-server access methods so that any developer who can connect to the Internet can access files on a CVS server.

Coordinating code: In traditional version control systems, a developer checks out a file, modifies it, then checks it back in. The developer who checks out a file has exclusive rights to modify it. No other developer can check out the file - and only the developer who checked out the file can check in modifications. CVS solves this problem with its unreserved checkout model. Checking out a file does not give a developer exclusive rights to that file. Other developers can also check it out, make their own modifications, and check it back in. CVS detects when multiple developers make changes to the same file and automatically merges those changes. CVS is cautious and will merge automatically only as long as the changes are not made to the same lines of code. If CVS cannot safely resolve the changes, the developer will have to merge them manually.

From now the development of the UWC code including the KEWL application, will be managed with CVS. This means that in order to effect changes to the KEWL code base, those changes must be made on the own local copy of the code, and committed back to the central code repository. These changes will not become active on the live site until the site has its copy of the code updated to the latest version. For the kewlforge.uwc.ac.za site this should happen on a regular basis so that changes can be tested, for the KEWL site (and other deployed installations) this will only occur when the next stable release of the code is made.

5.4.1 TortoiseCVS: a Windows CVS client

In order to access the CVS repository from our workstation we will need to install a CVS client. We use TortoiseCVS client on Windows – it integrates very neatly into Windows Explorer so that its functionality is available on the right-click menu for files and folders in Explorer. An installer for the latest version can be downloaded from http://cvs.uwc.ac.za/TortoiseCVS.exe.

5.4.2 Checking out the KEWL module

When it has been downloaded and run, some new options on the menu will appear when one right-clicks within Windows Explorer. The first step should be to 'checkout' a copy of the KEWL source code to local machine. It will need a folder on local machine that holds local copies of the checked out source code. Navigate to this folder in Windows Explorer and right-click on the background – A menu somewhat like that shown below should appear. Choose 'CVS Checkout':

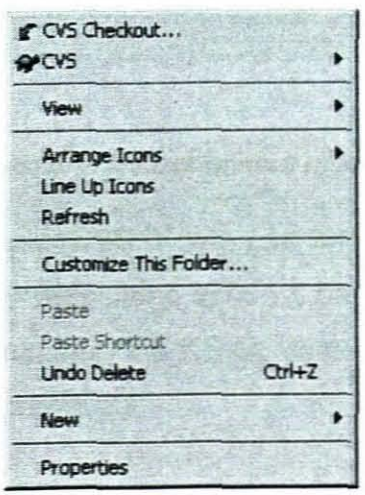


Figure 5-1: CVS checkout

Then a dialog like that below (Figure 5-2) will be shown. Fill it out as shown using user name. Server: 'cvs.uwc.ac.za', Repository directory: '/home/cvs', Module 'kewl'.

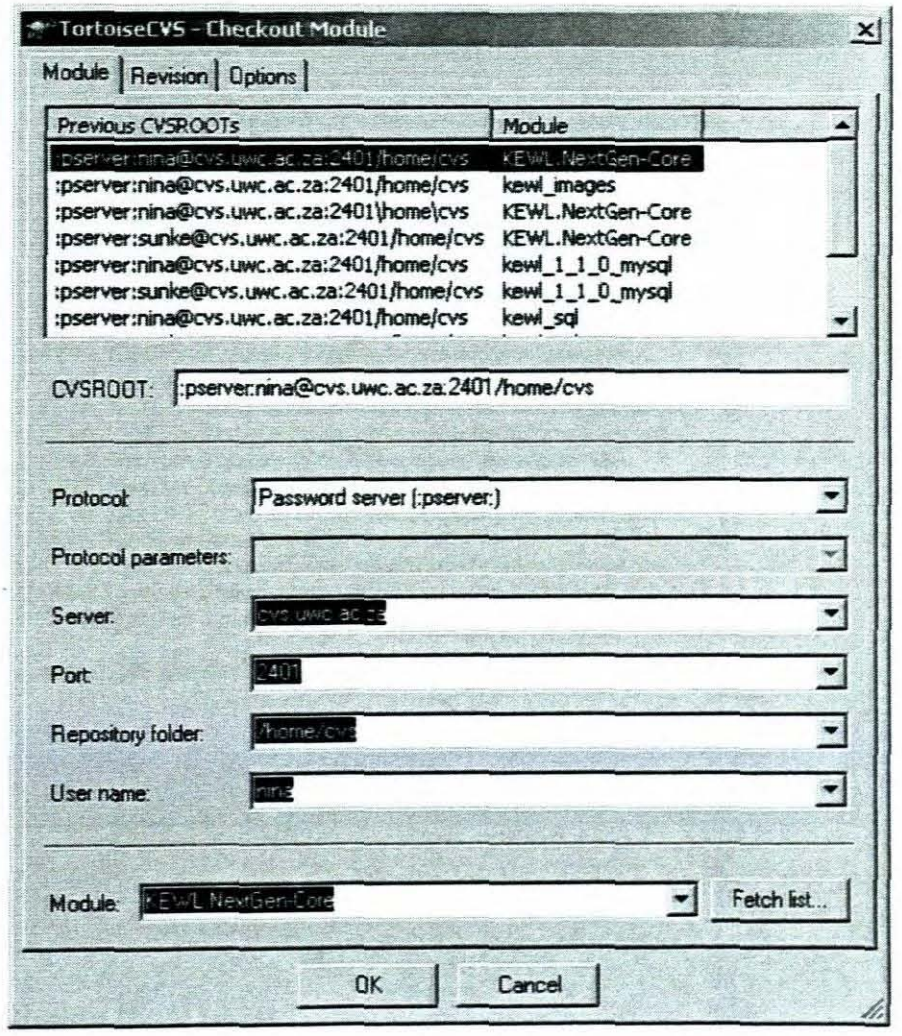


Figure 5-2: CVS checkout module

When one clicks 'OK' TortoiseCVS, it will ask for password and then begin retrieving the code from the repository. When it has finished, there should be new folder called 'kewl' containing the KEWL code. The progress window will look somewhat like this (Figure 5-3):

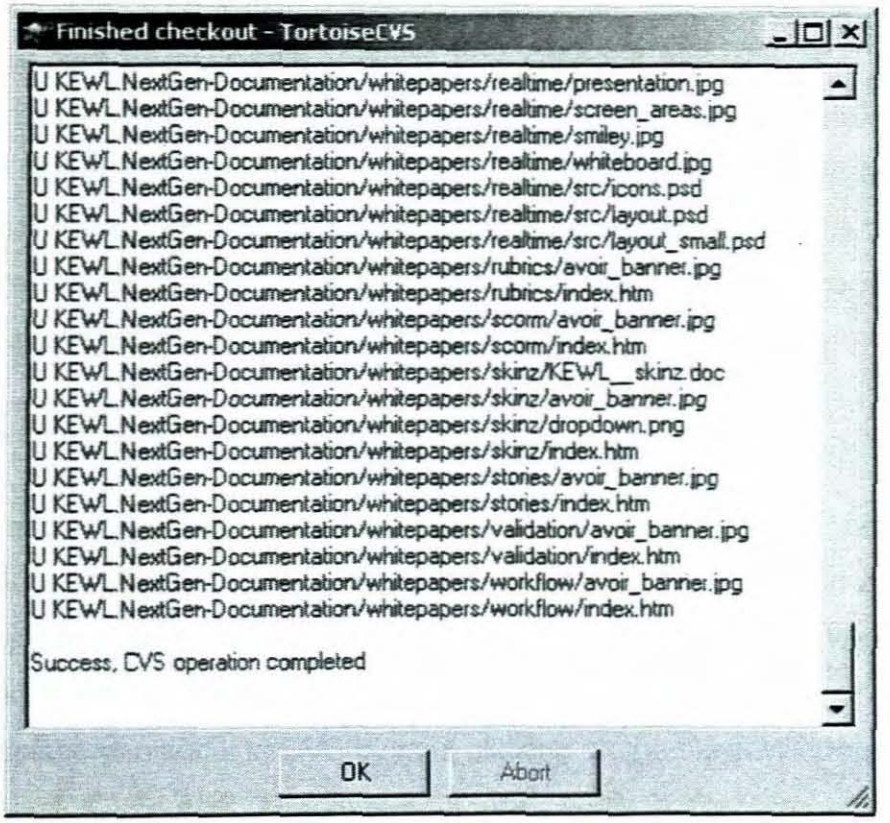


Figure 5-3: Checkout finished

5.4.3 Updating local copy

From time to time the local copy of the code must be updated to reflect the latest changes made to the repository. This will help ensure that work we do does not conflict with changes others are making. Updating our copy is as simple as right clicking on the folder icon for the 'kewl' directory created above and selecting 'CVS Update'.

5.4.4 Committing changed code back to the repository

When one is satisfied with the changes one has made to the code one should commit the changes back into the repository. One does this by right clicking on the folder icon for the 'kewl' directory and selecting 'CVS Commit'. One will be asked for a log message – one has briefly describe the changes one has made so that others can get a general sense of what one has done without looking at the code itself.

If the local copy of one of the files one is trying to commit is not up-to-date with

the latest changes from the repository, the commit will fail. Particular attention must be paid to messages showing that changes between files have been merged, or that a conflict has occurred. Those files must be double-checked to ensure that everything is still consistent. Proceed with the commit again. Note that under no circumstances will one lose one's own modifications to the file – they are preserved when changes are merged, and the original copy of the file is also saved with a filename beginning '. #'.

5.4.5 Adding and removing files

If one adds a new file to the code base, or remove an old one, he/she must make sure to inform CVS that he/she has done so. It will not automatically pick up new files, as CVS does not know whether one actually want this file included in the central repository or not. If one has added a file to the KEWL code, he/she can add it to CVS by right clicking on the file in Windows Explorer and select 'CVS Add'. Conversely, if one wants a file removed from the repository, right-click on the file and select 'CVS Remove' (obviously one must do this before deleting the file, otherwise the file will be deleted in the process). The changes will not actually be reflected in the repository until one commits these changes.

5.4.6 Use of CVS for KEWL.NextGen

All KEWL.NextGen code will reside in the CVS repositories at http://cvs.uwc.ac.za. All repositories should contain a readme.txt file explaining who created the repository and what it contains. CVS repositories for KEWL.NextGen will be structured as follow (Table 5 -1):

Repository	Contents
KEWL.NextGen_Core	The code for KEWL.NextGen that is needed for all projects
KEWL.NextGen_Module_Modname	KEWL.NextGen modules where Modname is the name of the module. For example, the test module would reside in KEWL.NextGenModule_Test.
KEWL.NextGen_SQL	Used to hold the SQL for creating and modifying KEWL.NextGen database tables.
KEWL.NextGen_ImageSRC	Used to store the source images for all KEWL.NextGen images, in original format (e.g. Gimp, Photoshop, Paint shop). Photoshop is the preferred format for image source. Image source files should be kept in logically named folders, and each folder should have a readme.txt file that explains what images are present and where they are used. No images should be used in KEWL.NextGen unless their source images are in this repository or they come from pre-designed collections.
KEWL.NextGen_Documentation	This repository should be used to store all KEWL.NextGen documentation, including code documentation, white papers, and user guides.

Table 5-1: CVS repository for KEWL.NextGen

5.5 Introduction to object-oriented programming

5.5.1 Programming

- Implementing data structures and algorithms

- Writing instructions for machines
- Implementing client specifications
- Coding and debugging
- Plugging together software components
- Specification and design
- Testing
- Maintenance

5.5.2 Object-oriented terminology

Table 5-2 shows the main object-oriented programming terms:

Object:	An object is an element (or instance) of a class; objects have the behaviours of their class. Thus, an object is a software bundle of related variables and methods. Software objects are often used to model real-world objects. The object is the actual component of programmes, while the class specifies how instances are created and how they behave.
Class:	A class describes the contents of the objects that belong to it: it describes an aggregate of data fields (called instance variables), and defines the operations (called methods).
Method:	A method is an action that an object is able to perform.
Message:	Sending a message to an object means asking the object to execute or invoke one of its methods.
Inheritance:	A class inherits state and behaviour from its super class. Inheritance provides a powerful and natural mechanism for organising and structuring software programmes.
Interface:	An interface is a contract in the form of a collection of method and constant declarations. When a class implements an interface, it promises to implement all of the methods declared in that interface.

Table 5- 2: object-oriented programming terms

5.5.3 Object

In everyday life, an object is anything that is identifiably a single material item. An object can be a car, a house, a book or a document. For programme purposes, that concept is extended somewhat. One must think of an object as anything that is a single item that a developer might want to represent in a programme. Therefore living objects, such as a person, an employee, or a customer, as well as more abstract objects, such as a company, a database, or a country will be included. Thinking about objects in this way not only enables one to write code that models the real world; it also enables one to break up a large programme into smaller, more manageable units. The idea really comes from the concept of a black box. The idea of a black box is that there are many objects in life that people are able to use but of which people do not understand the mechanism, for example a car radio. Most people do not know exactly how a car radio works; however, they do know what it does and how to operate it. Furthermore, they can take out the radio, plug in a different one and it will do basically the same thing, even though its internal workings might be completely different. Black boxes formalise the idea that there is a difference between what something does and how it works, and that two objects can do the same thing but work differently on the inside. Replacing one object with another does have some subtle effects. Car radios might have different knobs and switches, and they might project different sound qualities, but the basic function is unchanged. Another important point is that the basic user interface is unchanged - people plug one car stereo into the slot in much the same way as the other person would another.

If all that is understood, then one can basically understand OOP, because OOP is about applying these same concepts to computer programming. If, in other areas of our lives, we use objects that have a well-designed interface that we are familiar with, and we know how to use them, but do not care how they work, why not do the same thing in programmes? In other words, break each programme into many units and design each unit to perform a clearly specified role within the programme. Likewise, this module is just a unit in whole KEWL project. That is basically what an object is. It makes it easier to

design the programmes if one understands this concept. The architecture of the programmes becomes more intuitive and easier to understand because it more closely reflects whatever it is that the programme is abstracting from real life. It becomes easier for multiple developers to work together, since they can work on different objects in the code; all they need to know is what an object can do and how to interface with it. They do not have to worry about the details of how the underlying code works. In programming, it is necessary to distinguish between a class and an object. A class is the generic definition of what an object is - a template. For example, a class could be car radio - the abstract idea of a car radio. The class specifies what properties an object must have to qualify as a car radio.

5.5.4 Class

In the real world, people often have many objects of the same kind. For example, the bicycle is just one of many bicycles in the world. Using object-oriented terminology, we say that the bicycle object is an instance of the class of objects known as bicycles. Bicycles have some state (current gear, current cadence, two wheels) and behaviour (change gears, brake) in common. However, each bicycle's state is independent of and can be different from that of other bicycles. When building bicycles, manufacturers take advantage of the fact that bicycles share characteristics, building many bicycles from the same blueprint. It would be very ineffective to produce a new blueprint for every individual bicycle manufactured. In object-oriented software, it is also possible to have many objects of the same kind that share characteristics: rectangles, employee records, video clips, and so on. Like the bicycle manufacturers, people can take advantage of the fact that objects of the same kind are similar and people can create a blueprint for those objects. A software blueprint for objects is called a class. Namely, a class is a blueprint, or prototype, that defines the variables and the methods common to all objects of a certain kind.

The class for bicycle example would declare the instance variables necessary to contain the current gear, the current cadence, and so on, for each bicycle object. The class would also declare and provide implementations for the

instance methods that allow the rider to change gears, brake, and change the pedalling cadence. After people have created the bicycle class, they can create any number of bicycle objects from the class. When people create an instance of a class, the system allocates enough memory for the object and all its instance variables. Each instance gets its own copy of the entire instance variables defined in the class. In addition to instance variables, classes can define class variables. A class variable contains information that is shared by all instances of the class. For example, suppose that all bicycles had the same number of gears. In this case, defining an instance variable to hold the number of gears is inefficient; each instance would have its own copy of the variable, but the value would be the same for every instance. In such situations, people can define a class variable that contains the number of gears. All instances share this variable. If one object changes the variable, it changes for all other objects of that type. A class can also declare class method. People can invoke a class method directly from the class, whereas people must invoke instance methods on a particular instance.

5.5.5 Object vs. Class

It is obvious noticed that the illustrations of objects and classes look very similar. And indeed, the difference between classes and objects is often the source of some confusion. In the real world, it is obvious that classes are not themselves the objects they describe: a blueprint of a bicycle is not a bicycle. However, it is a little more difficult to differentiate classes and objects in software. This is partially because software objects are merely electronic models of real-world objects or abstract concepts in the first place. But it is also because the term object is sometimes used to refer to both classes and instances. In comparison, an object is shaded, indicating that the object exists and that people can use it.

5.5.6 Inheritance

Generally speaking, objects are defined in terms of classes. One can know a great deal about an object by knowing its class. Even if people do not know what a penny-farthing is, if they were told that it was a bicycle, they would

know that it had two wheels, handlebars, and pedals. Object-oriented systems take this a step further and allow classes to be defined in terms of other classes. For example, mountain bikes, racing bikes, and tandems are all kinds of bicycles. In object-oriented terminology, mountain bikes, racing bikes, and tandems are all sub-classes of the bicycle class. Similarly, the bicycle class is the super-class of mountain bikes, racing bikes, and tandems.

Each sub-class inherits state (in the form of variable declarations) from the super-class. Mountain bikes, racing bikes, and tandems share some states: cadence, speed, and the like. Also, each sub-class inherits methods from the super-class. Mountain bikes, racing bikes, and tandems share some behaviour: braking and changing pedalling speed, for example. However, sub-classes are not limited to the state and behaviours provided to them by their super-class. Sub-classes can add variables and methods to the ones they inherit from the super-class. Tandem bicycles have two seats and two sets of handle bars; some mountain bikes have an extra set of gears with a lower gear ratio. Sub-classes can also override inherited methods and provide specialised implementations for those methods. For example, if we had a mountain bike with an extra set of gears, we would override the change gears method so that the rider could use those new gears.

The inheritance tree, or class hierarchy, can be as deep as needed. Methods and variables are inherited down through the levels. In general, the farther down in the hierarchy a class appears, the more specialised its behaviour. The object class is at the top of class hierarchy, and each class is its descendant (directly or indirectly). A variable of type of object can hold a reference to any object, such as an instance of a class or an array. Through the use of inheritance, programmers can reuse the code in the super-class many times. Programmers can implement super-classes called abstract classes that define "generic" behaviours. The abstract super-class defines and may partially implement the behaviour, but much of the class is undefined and unimplemented. Other programmers fill in the details with specialised sub-classes.

5.6 Object-oriented programming

Unlike imperative programming, in which explicit sequencing of operations drives execution, or functional programming, where it is driven by the required computations, object-oriented programming can be thought of as data-driven. Using objects introduces a new organisation of programmes into classes of related objects. A class groups together data and operations. The latter, also known as methods, define the possible behaviours of an object. A method is invoked by sending a message to an object. When an object receives a message, it performs the action or the computation corresponding to the method specified by the message. This is different from applying a function to arguments because a message (which contains the method name) is sent to an object. It is up to the object itself to determine the code that will actually be executed; such a delayed binding between name and code makes behaviour more adaptable and the code easier to reuse. With object-oriented programming, relations are defined between classes. Classes also define how objects communicate through message parameters. Aggregation and inheritance relations between classes allow new kinds of application modelling. A class that inherits from another class includes all definitions from the parent's class. However, it may extend the set of data and methods and redefine inherited behaviours, provided typing constraints are respected.

5.6.1 Object-oriented programming in PHP

Object-oriented programming in any language is the use of objects to represent functional parts of an application and real life entities. For example people may have a Person object to hold the data related to a person and even provide some functionality that this person may be capable of. Object-oriented programming has long been used in games to represent the objects such as a User or an Enemy, or even a Weapon. This amazing way of programming has proven just as useful in software and Web development. Any OOP language should have the following: abstract data types and information hiding; inheritance; polymorphism. This can all be done using PHP classes.

Data members are defined in PHP using a var declaration inside the class and they have no type until they are assigned a value. A data member might be an integer, an array, an associative array or even an object. Methods are defined as functions inside the class. To access data members inside the methods one has to use $this->name, otherwise the variable is local to the method. See the code list 5 -1.

Code list 5-1: PHP Class

```
<?php
class Something {// In OOP classes are usually named
                 // starting with a capital letter.
var $x;
    function inputX($v) { // Methods start in lowercase then use
                          //lowercase to capital separate words in the
                          //method name example fetchBlog()
                $this->x=$v;
            }
    function outputX()  {
            return $this->x;
            }
    }
?>
```

People create an object using the new operator, and see the code list 5 -2.

Code list 5-2: Create an object

```
$obj=new Something;
                 // Then one can use member functions like:
$obj->input(5);
$see=$obj->outputX();
```

The inputX member function assigns 5 to the x data member in the object obj (not in the class), and then outputX returns its value 5 in this case.

Inheritance is easy in PHP using the extends keyword and see code list 5 – 3.

Code list 5-3: Inheritance

```
<?php
  class Another extends Something {
  var $y;
          function inputY($v) {// Methods start in lowercase then
                                //use uppercase initials to separate words
                                  //in the method name example
              setForForm();
              this->y=$v;
            }
          function outputY() {
               return $this->y;
            }
      }
  ?>
```

Objects of the class another now have all the data members and methods of the parent class (Something) plus its own data members and methods. People can override a method in the derived class by redefining it. If we redefine outputX in another we can no longer access method outputX in something. If people declare a data member in a derived class with the same name as a data member in a base class the derived data member hides the base class data member when it is accessed. People might define constructors in classes; constructors are methods with the same name as the class and are called when an object of the class is created, and see code list 5 – 4.

Code list 5-4: constructors

```
<?php
class Something {
var $x;
        function Something($y) {
                $this->x=$y;
            }
        function inputX($v) {
                $this->x=$v;
            }
        function outputX() {
                return $this->x;
            }
    }
 ?>
```

So people can create an object using: $obj=new Something(6); and the constructor automatically assigns 6 to the data member x. Constructors and methods are normal PHP functions so people can use default arguments. In our project, the constructor for the data connection class, and any other classes that extend the framework (Framework will be introduced in chapter 6.) is the method that must always be called *init*().

OOP code is easy to maintain, easy to understand and easy to reuse. Those are the foundations of software engineering. Applying those concepts to Web-based projects is the key to success in future Web sites.

CHAPTER 6 RESEARCH PROJECT IMPLEMENTATION IN KEWL.NEXTGEN

This section will contain all the information on design and architecture of a test module in KEWL.NextGen, Including the design pattern, framework and design standards of KEWL.NextGen. It will provide the descriptions of this test module, and support the use-case to model the system architecture. Finally, it will provide a report on test module application, which will explain in detail the implementation of the test module.

6.1 Executive summary of the existing LMS KEWL

The PHP version of KEWL next-generation LMS is part of the AVOIR project. The AVOIR project is about harnessing 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. Although world-class software will be produced within this project, this is not a software development project. Rather it is a project about human development, about capacity building, and about creating opportunities for people through the formation of transnational alliances both within Africa and out.

The aim of our team was to take an existing open source application that they developed at UWC, a sophisticated learning management system called KEWL (Knowledge Environment for Web-based Learning), and use it as the basis for building a next-generation learning management system. To achieve the next generation system, it was necessary to modularise the existing system and convert it entirely to PHP, an open source programming language. Using this tool, they will build a core of developers in African institutions, mainly universities, but also other organisational structures. This core would do the conversion of KEWL to PHP, and initiate the development of new cutting edge tools based on sound educational practises.

It was meant to be a bringing together of cutting edge educational research with cutting edge computer science to produce the world's most advanced learning management system. Once the first working code was released, other institutions and individual volunteers would be encouraged to volunteer or contribute to the project in true open source community fashion. In this way, the code would grow rapidly in relation to the original core-funded group. Best practises from commercial software engineering would be made available, and developers in Africa would have opportunities to be exposed to the latest software development techniques. Students in higher education, including those participating in short term training programmers, would be exposed to industry standard practises and learn from the source code created in this project. This would allow the talent pool in Africa to grow, and contribute to development in Africa. At the same time, the core team would be able to offer a service to government, education and business around the development and use of open source software, as well as provide a service to e-learning and other open source platforms. The KEWL component of the AVOIR project was expected to fund for three years, following which it was expected to have gained enough momentum and support to be self-sustaining.

6.1.1 What is KEWL?

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. They wanted to have a comprehensive learning management tool that they could modify and use 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. A number of courses at UWC are now run either fully online using KEWL, or KEWL is used to supplement traditional teaching-and-learning. KEWL is used for marks administration by all courses at UWC, irrespective of whether they are online or not.

Aside from its application at UWC itself, KEWL is already being used by the

NetTel@Africa project, a consortium of eight universities in Africa and four in the USA that are offering an online master's degree in ICT policy and regulation (http://www.nettelafrica.org; http://elearn.nettelafrica.org). It is also the basis for the International Ocean Institute Virtual University, an incipient global programme in Ocean Policy, Law of the Sea and Coastal Management. Experimental KEWL installations have been done in Ghana, Nigeria, Kenya, Tanzania, Turkey, China, Canada and the UK.

For testing purposes, it will run on a laptop with any Windows operating system. Aside from needing a Web server, KEWL also requires a SQL server, for example the one that is part of the Microsoft Office suite, and this may be on the same machine or on a different server on the network. Thus, KEWL will scale from a single desktop to the enterprise.

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 developers have tried to keep the KEWL interface as simple as possible, and to make it a system suitable for the low bandwidth environment that characterises most developing countries. There is no limit on the kind of content that can be contained in a KEWL course, so the actual bandwidth requirements will depend on what a particular educator decides to put into the content of a course.

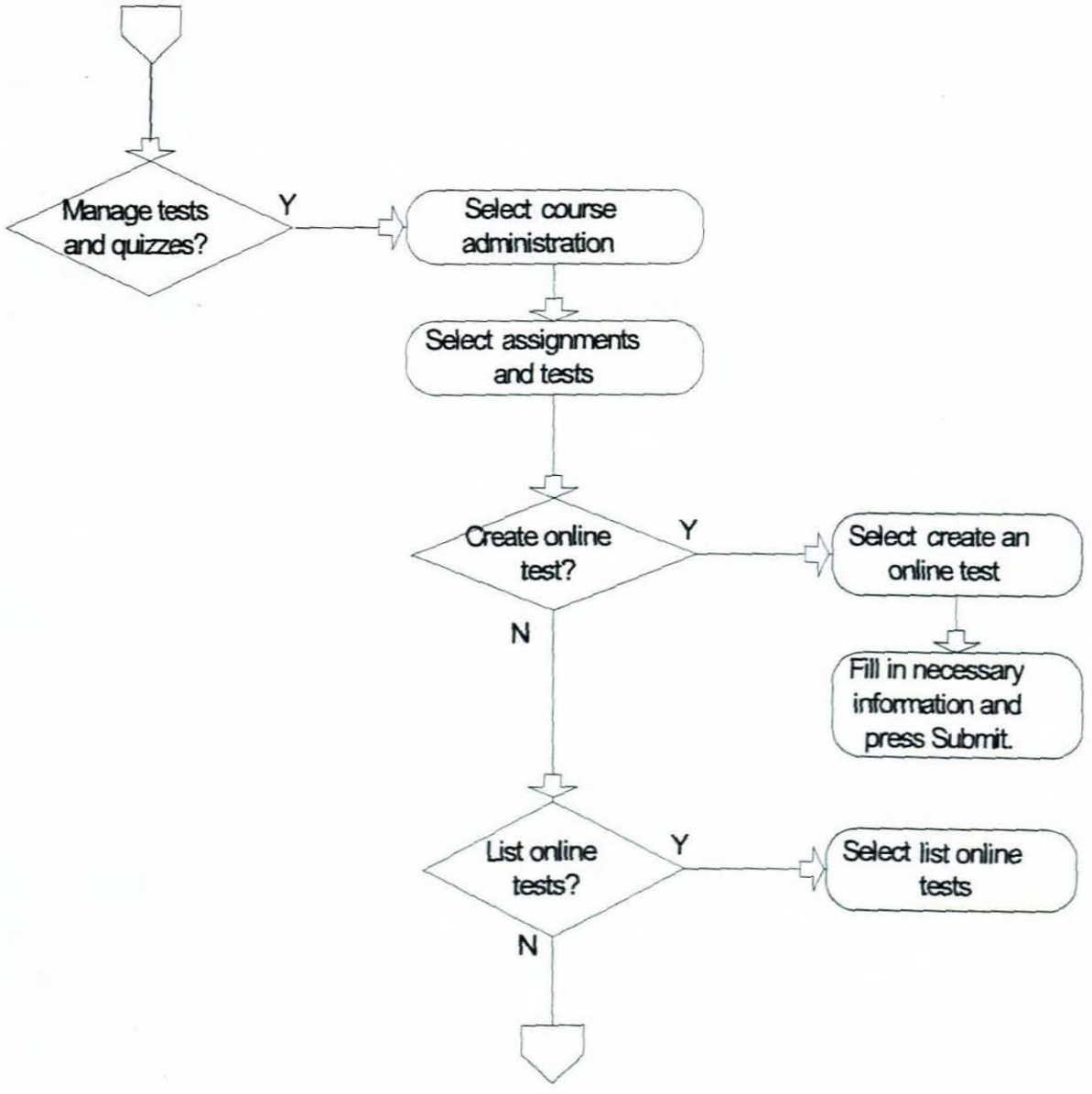
KEWL provides an interface for students to interact with educational content and resources, allows learners to access content and communicate with their teacher and classmates, and allows teachers to manage class administration and assessment. Since KEWL is available as free software under the GPL,

there are no licensing costs for using the source code, although it is suggested that any changes are made available back to the open source community.

Since KEWL was originally written as an Active Server Pages application it can only run on Microsoft servers that support ASP. Thus, while KEWL applications currently are freely available, any participating institution would have to run Microsoft Web and SQL servers and buy a licence for the underlying Microsoft software. This project therefore aims to evolve the functionality of KEWL to the next generation by upgrading and converting the code to fully open source-programming languages and cross platform environments.

6.1.2 The assessments current feature of KEWL 1.2

Figure 6-1 explains the whole process of test management of KEWL 1.2. It provides the high-level description of goals for the system architecture, supports the use-case to model the system architecture and it defines and documents the features of the software to provide guidance on how to use the software as well as to what happens in the software itself.



(Continue to next page)

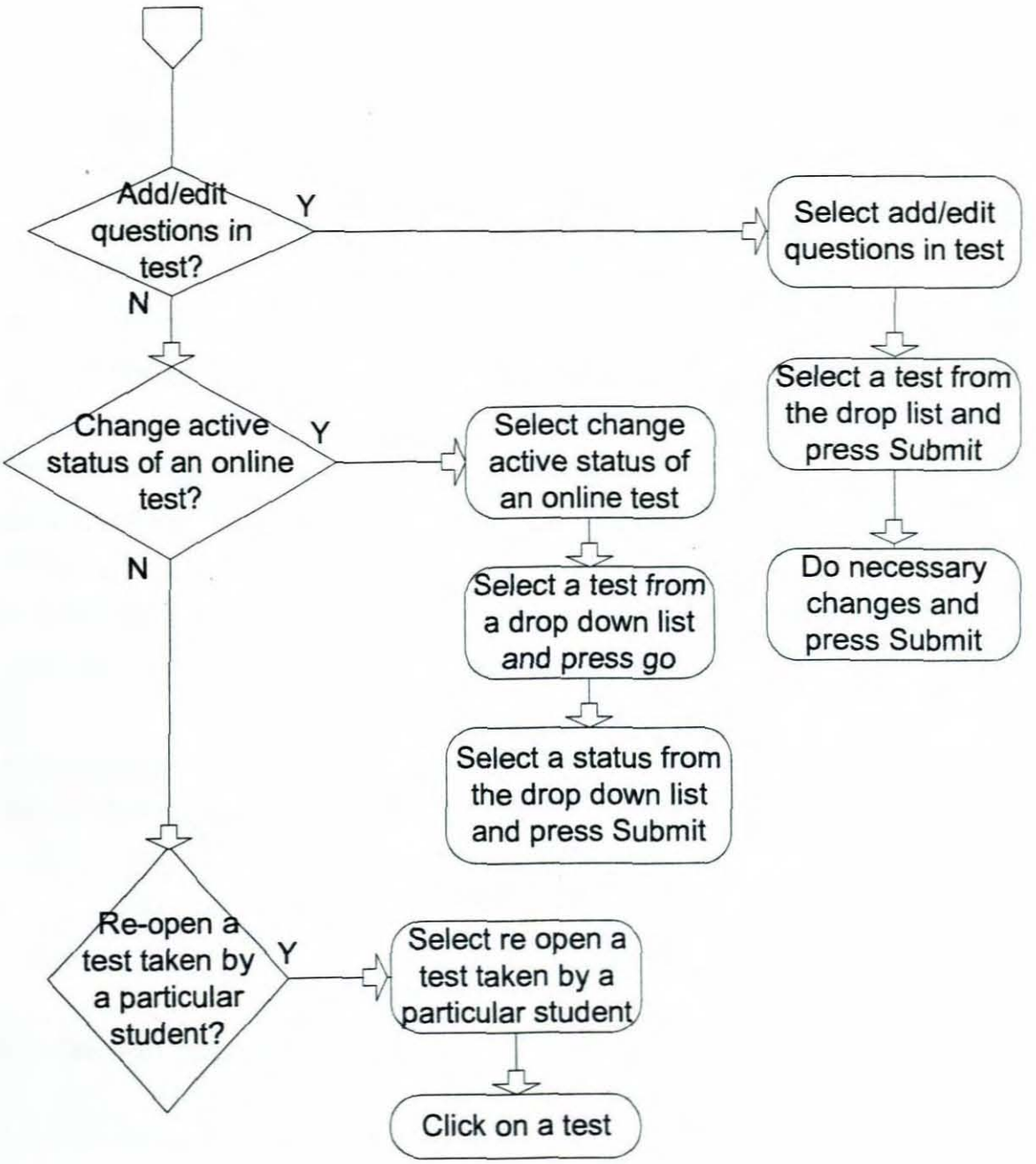


Figure 6-1: Current KEWL 1.2 assessment module architecture diagram

The process of assessment management includes the following:

- In administering marks, students view their own marks but not those of others;
- The marks administration system can communicate with a central administration database;
- It can add essays and allow essay booking on a first come first-served basis, with essays uploaded to assignments folder;

- There are online worksheets, the facility to add comments and marks, linked to marks administration tool;
- It has an upload area for assignments per course;
- Multiple choices, and computer-assessed quizzes that add marks to the marks administration system are available;
- Discussion forums can be used for assessment, with discussions sortable by students to facilitate mark assignment;
- Offline activities can be done and uploaded as assignments.

The aim for this software is to facilitate assessment management, and that means teachers should be able to put up assignments, essays, worksheets, tests, quizzes, etc. Students can then access those assessments and respond to them. Security is made possible by denying other users access to perform action on some parts or pages that do not concern them.

The multiple-choice is the only question type in the current test feature of KEWL. This project developed the three types of questions in the test module

- Multiple choices
- True or False
- Short answer

6.2 Design pattern

6.2.1 What are design patterns?

Design patterns are basically a general method that can be used in solving a variety of different problems. Design patterns are simply defined solutions to common problems. Design patterns are not created by someone who sits in a room and decides to make a design pattern; instead, design patterns are proven solutions that are implemented over and over again through different projects. This re-use of the solution itself becomes a pattern.

Design patterns help structure code, although they do not say anything about how to implement it. They solve problems before one starts to write codes

because they affect the design of programs by recognising the possible abstractions in the problem. More than one pattern may apply, and within any pattern, many ways to implement it. We typically represent patterns with modules so that their implementations are re-usable and abstract and make full use of encapsulation. The rest of the program does not need to know how the module works, and indeed, the more the program knows about the module's workings, the more of a problem one has. Patterns promote loose coupling so that their implementation does not affect the rest of the program.

6.2.2 Why use design patterns?

PHP is a scripting language usually used for simple Web development, so the question may be asked: Aren't functions are good enough? This is probably true if a developer works alone on small projects. But Design and Design Patterns are beneficial with regards to:

- Maintenance
- Documentation
- Readability
- Ease in developing large development teams
- Developing Code to be used by other than themselves

6.3.3 Model –View- Controller design pattern and framework

The Model-View-Controller (MVC) architecture pattern separates software into models representing core functionality; views that display the models to the user, and controllers that let the user change the models. Although more sophisticated architectures have since been developed, MVC is interesting to explore because its simplicity makes it more acceptable to practitioners and it is beginning to become well-known in industry.

Models

The model of an application is the domain-specific software simulation or implementation of the application's central structure. This can be as simple as an integer (as the model of a counter) or a string (as the model of a text editor), or it can be a complex object that is an instance of a sub-class.

Views

The view is the presentation to the user of information contained in the model. This usually consists of screens containing information from the model. Views deal with everything graphical; they request data from their model, and display the data. They contain not only the components needed for displaying but can also contain sub-views and be contained within super-views. The super-view provides the ability to perform graphical transformations, windowing, and clipping, between the levels of this sub-view/super-view hierarchy. Display messages are often passed from the top-level view (the standard system view of the application window) through to the sub-views (the view objects used in the sub-views of the tool view).

Controller

The controller controls the input from the user. Controllers contain the interface between their associated models and views and the input devices (keyboard, pointing device, time). Controllers also deal with scheduling interactions with other view-controller pairs: they track mouse movement between application views, and implement messages for mouse button activity and input from the input sensor. Although menus can be thought of as view-controller pairs, they are more typically considered to be input devices, and therefore are in the realm of controllers. (See figure 6-2).

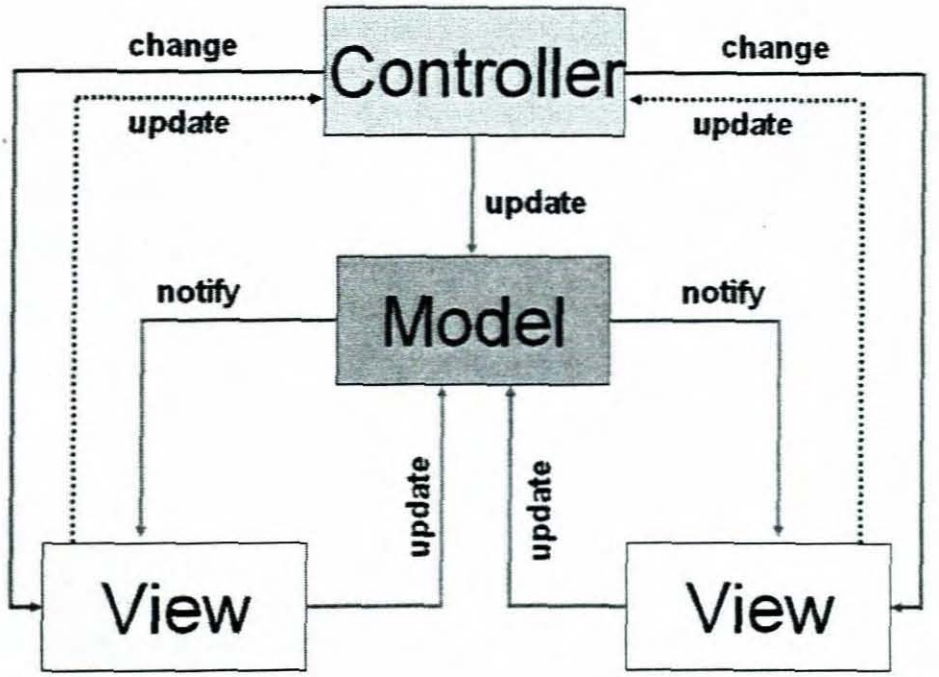


Figure 6-2: Model-View-Controller pattern

6.3 The KEWL.NextGen framework

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. 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 are changed often during development leads to more robust applications that are easier to maintain.

In the case of KEWL.NextGen, the model represents the data as the main application object that is manipulated. The framework provides a data connection base class that can be extended by table-specific module classes to act as the code for manipulation of the model (data). The data model class for a module resides in the */modules/modulename/classes/* folder, and as a naming convention should be called db*Tablename*_class_inc.php, where *Tablename* is the name of the table to which it connects. And this module name is ktest (kewl test). There will often be more than one model class used

to support a module, and one model class may connect to more than one table via SQL joins and unions.

The constructor for the data connection class, and any other classes that extend the framework, is a method that must always be called *init()*. This method executes any code that is required to be executed during construction, such as creating instances of commonly required objects. Below is an example of the init constructor from the helloworld example application:

Code list 6-1: the constructor

```
24:  function init()
25:  {
26:   $this->objUser =& $this->getObject('user', 'security');
27:   $this->objHelp=& $this->getObject('helplink','help');
28:  }
```

It instantiates the user class and the help class that will be used in the module. The controller consists of at least one file, contoller.php, that extends the controller base class. The controller resides in */modules/modulename/* and must be called controller.php. The class defined in the controller main file must be the same as the name of the module. For example, a module named ktest will have a controller.php file that contains the following code:

Code list 6-2: controller main file

```
class ktest extends controller
{
}
```

The view aspect of the MVC approach to the application framework is provided in the form of a three-level template system that uses only PHP templates and does not at this stage support a templating engine. There are three levels of templates in the KEWL.NextGen view: a page template, a layout template and a content template. These can be thought of as having a nested structure, in which the page template sets up the page, puts in a banner and closes the

page. The layout template defines any additional layout elements for the page, and the content template controls the display of the output of the controller onto the page. The framework itself provides default page and layout templates, allowing most modules to only be concerned with the content template. (See figure 6-3).

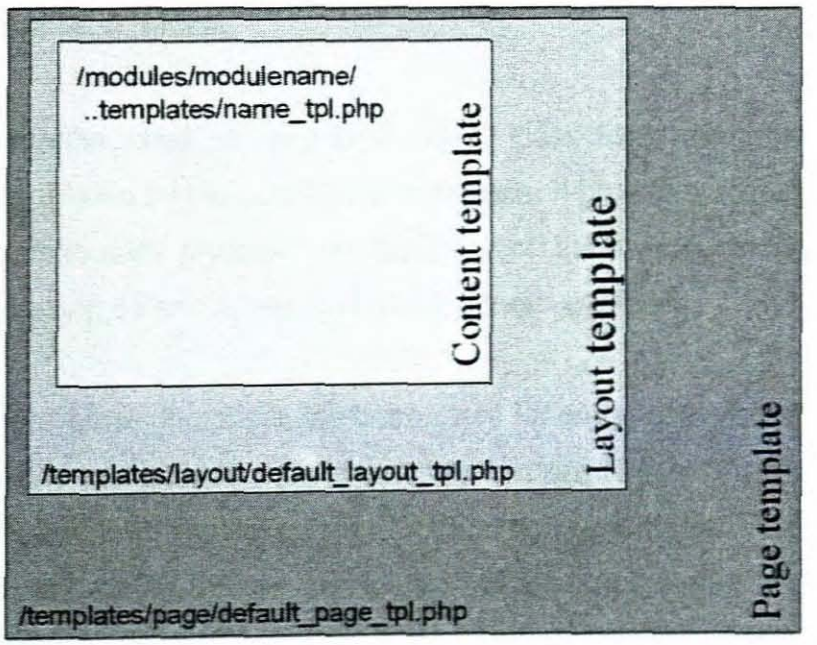


Figure 6-3: MVC approach to the application framework

The KEWL.NextGen framework provides four files that act as the glue that holds the framework together. They implement various aspects of the MVC and front-controller implemetation. These files are located in /core/classes/ and are:

controller_class_inc.php, dbtable_class_inc.php, engine_class_inc.php, and object_class_inc.php. The engine class provides core functionality needed to run modules, and other functions, and is not normally extended or referenced in a module. An understanding of the other three is, however, vital to module developers.

The file dbtable_class_inc.php contains an abstraction of the PEAR DB data access class, and is thus the base class for the data model of any

KEWL.NextGen application. It must be extended by our model classes, which should always reside in the /modules/modulename/classes folder. It is vital that its methods are used rather than raw SQL whenever possible, and always when data are being written to the database, whether new data or updates of existing data. This is vital because this is the basis for active dynamic mirroring. No code will be allowed into a KEWL.NextGen module or applciation that does not follow this requirement.

The file controller_class_inc.php is the base class for a controller, and must always be extended by the contoller of a module. It provides access to some of the core functionality provided by the core of the framework, in particular access to loading other classes and making them available.

The file object_class_inc.php is the base class for some of the other framework classes, and is available to extend classes that are not the main controller of our module, but that require access to framework elements. This class may also be extended by helper modules that are conceptually part of the framework, such as the modules that make up the htmlElements set of classes.

The framework includes modules that are always available, and that are conceptually part of the framework even though they are just a collection of modules. At the time of writing, some of these modules are still under development, so a comprehensive list is not provided. There are a set of these modules for providing HTML rendering, and they reside in the module folder /modules/htmlelements/.

6.3.1 Objects and instances of objects and their naming

All KEWL.NextGen development will follow a strictly object-oriented approach. Objects, as well as their properties and methods, begin with a lowercase letter (e.g. user) and identifiable words are separated by the use of capital letters (e.g. isLoggedIn). Instances of objects also begin with the lowercase letters and start with $obj (e.g. $objUser, $objUser->isLoggedIn(); not

Objuser.Isloggedin()). Care should be taken in designing classes for KEWL.NextGen not to try to include too much functionality in a single class. Rather build a class for core functionality, and extend it with another class when extra functionality is needed. When a new instance of a class is created, it should always be done by reference:

```
$object =& new someClass();
```

This is important, otherwise one may experience strange bugs in one's code and waste hours trying to locate them. Framework extension objects should always return a string within a show() method.

6.3.2 Database abstraction

Database abstraction will make use of the DB library of the PEAR application framework, which is further abstracted into the dbTable framework class in order to ensure that KEWL.NextGen is able to support active dynamic mirroring of sites. The dbTable layer provides database access properties and methods, and only those methods will be used to access the database. There will be no use of direct MySQL or PEAR DB calls. All database tables will have a class written for them, and this class will extend the DBTable database abstraction class. Thus, all database access that adds or modifies data must use dbTable and the methods.

6.3.3 Normalisation and referential integrity

All data tables should be properly normalised. To ensure that relationships are manageable, one must make sure that primary and foreign keys are used. One cannot create tables that will have empty data columns for any particular user, but should rather create tables that have a 1:1 relationship. One should ensure referential integrity among linked tables by using the InnoDb handler in mySQL. If one does not use the InnoDb handler, mySQL cannot enforce referential integrity. InnoDB allows one to drop any table even though that would break the foreign key constraints that reference the table. To use the InnoDb handler, make sure that the table type is set to TYPE=InnoDB as shown below.

Code list 6-3: InnoDb handler

```
<?
$sqldata[]="CREATE TABLE `tbl_test_test` (
  `testId` varchar(50) NOT NULL default '0',
  `testName` varchar(50) NOT NULL default '',
  `testOwner` varchar(50) NOT NULL default '',
  `testNote` text,
  `startDate` datetime NOT NULL default '0000-00-00 00:00:00',
  `endDate` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (`testId`)
) TYPE=INNODB";
?>
```

6.3.4 Making a ktest module with the KEWL.NextGen application framework

In developing a module for KEWL.NextGen, it is vital to understand the minimum requirements. First, however, it is useful to examine the MVC model in relation to the location of files and which base class they extend (following diagram). All base classes reside in the folder /classes/core. (See figure 6-4).

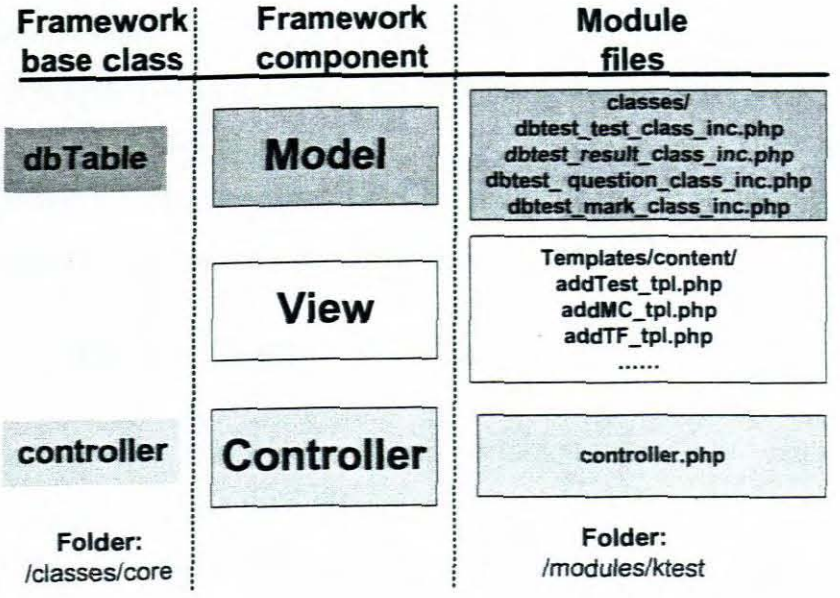


Figure 6-4: Location of files

The model is based on the database abstraction class, which is a class called dbTable that resides in the file dbtable_class_inc.php. The database connection class should be named after the table that it connects to and should reside in /modules/*ktest*/classes/ and should begin with the letters db. For example, dbtest_question_class_inc.php is the name for ktest model class file to connect to the test question database table. The class itself will thus be named dbTest_question inside the file.

As far as the view is concerned, one only needs to consider the content templates, which reside in */modules/ktest/templates/content* and can take any filename as long as the convention _tpl.php is used though the names should be meaningful. For example, content templates are called main_tpl.php, editTest_tpl.php, comfirmDelete_tpl.php, etc. The controller base class is the file controller_class_inc.php, and the controller class of the module extends this file. The controller must be a class with the same name as the module, and must reside in the file controller.php in the module's root (/modules/*ktest*/). The controller may also have other classes that help it do its work, and these classes reside with the model class in /modules/*ktest*/classes/.

Thus, there are a minimum of three files required by a KEWL.NextGen module that has a user interface, two if there is no data connection, one if there is no data connection and no output. These three represent the controller, the database connection class, and the content template as noted above.

/modules/ktest/controller.php [required]
/modules/ktest/templates/content/main_tpl.php [required]
/modules/ktest/classes/dbtest_test _class_inc.php [required]

6.4 Use of UML for KEWL.NextGen

KEWL.NextGen will adopt an object oriented, modular approach. All functionality will be contained within objects and addressed as properties and methods. Objects will be described using UML before any coding beings. Where existing KEWL objects are converted, UML diagrams will be posted on the KEWL.NextGen developers Website for comment. Developers using KDE

on the GNU/Linux platform may wish to make use of UML, which can generate PHP code (See Table 6-1).

Link	Description
http://uml.sourceforge.net	Umbrella UML Modeler is Unified Modeling Language diagram programme for KDE. ULM allows developers to create diagrams of software and other systems in a standard format.
http://www.phpedit.net/products/xmi2php	XMI2PHP is a php code generator, which takes an XMI (XMI is an XML based file format used to describe UML diagrams.) file and transform it into a file collection of php code.

Table 6-1: UML

6.4.1 What is the UML?

UML stands for Unified Modelling Language. It is commonly referred to as "the UML". It is not quite like a programming language. Neither is it quite like a programming language. It is also not another XML-based standard. The UML is a standardised graphical language to describe a system. Its vocabulary is very large, and all the graphical symbols in the UML have a well-defined meaning. The UML is not limited to designing software. It is quite possible to visualise business processes using the UML. Even though some parts of the UML are purely there for modelling object-oriented applications, it is not limited to that. There are a few particular cases where the UML can be very useful.

6.4.2 Designing

The most obvious case is application design. The UML provides a large library to model virtually everything that goes on inside the system. Having a visual presentation of one's system written in UML notation makes sure that anyone

who can understand UML notation will be able to perceive what the system does. It makes it easy to discuss the application architecture and even communicate parts of the application design with clients.

6.4.3 Documenting

The UML can also be used as part of a system's documentation. Thus, when other developers are expected to change or extend an application. It can save a great deal of time if one has a UML drawing at hand when one has to change an application years later. It is generally easier to go over a graphic representation of a system than finding one's way through huge amounts of code to decipher how something was programmed.

6.5 Report on test module application in KEWL.NextGen

6.5.1 Functionality of test module

The functionality of the developed test module includes the following:

- Add a new test
- Preview an existing test paper
- Edit an existing test
    - Change test information
    - Add a question (multiple choice\true or false\short answer three question types)
    - Delete a question
- Delete an existing test
- Mark\view test results

6.5.2 The test module application use cases

Use case diagrams are often the first type of diagrams that are created after the requirements for an application from the target-user have been gathered. A use case diagram is composed of several use cases, actors and relationships between the actors and the use cases (See figure 6-5).

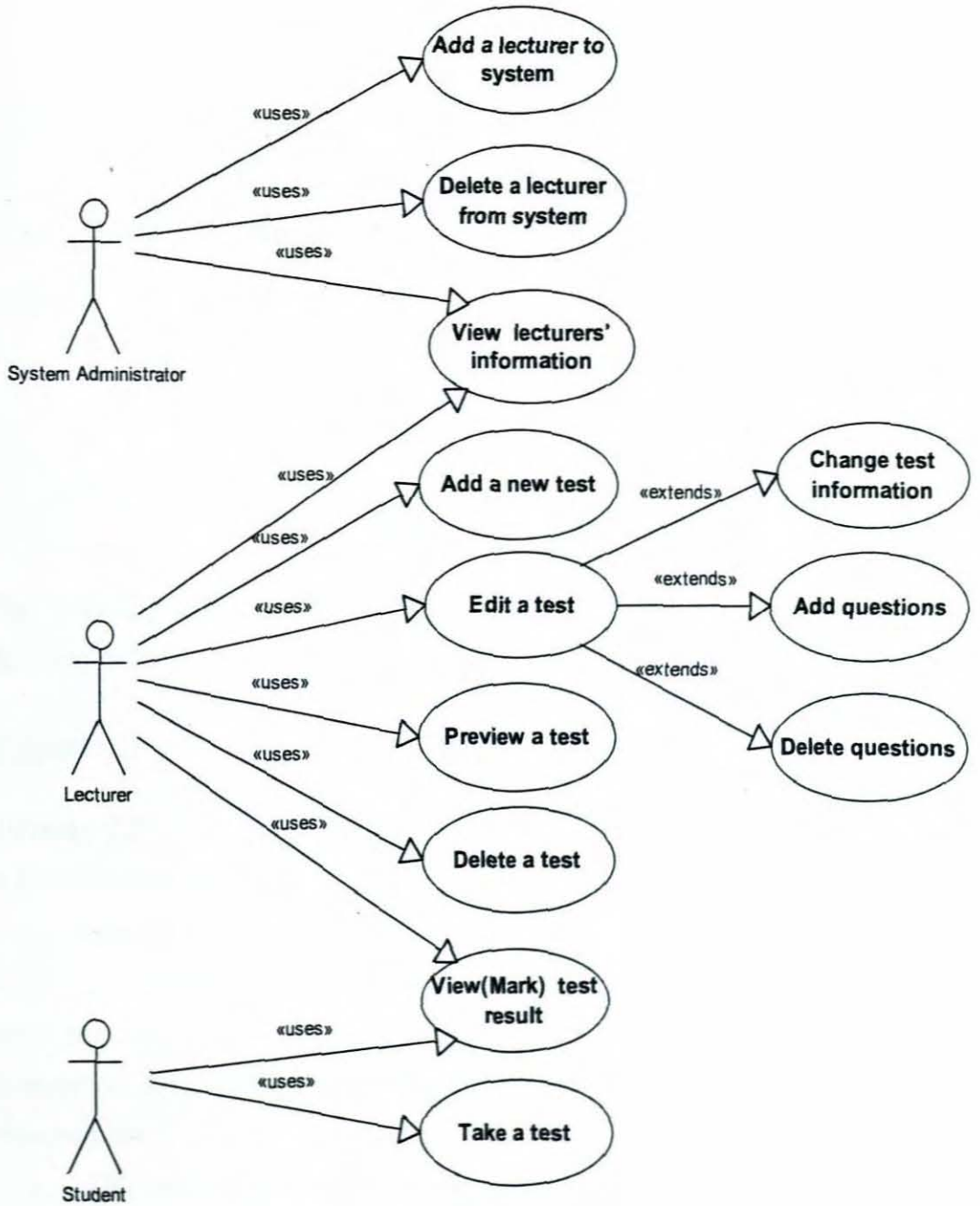


Figure 6-5: The test module application use case diagram

The use case that is actually represented by the ellipse is primarily a definite standard for writing use cases. However, there are a number of fields that commonly go into a use case specification as in table 6-2:

Use case number
Author's name and date
Use case name
Actor(s)
Use case description
Pre-conditions
Post-conditions
Event sequence

Table 6-2: A use case specification

The following are the test module use cases. They will describe the above figure in detail.

Use case 1:

Number: 2.2

Author's name: Na Zhang

Date: 2004-9-13

Name: Add a new test

Actor: teacher

Description: the use case describes the process of a teacher adding a new test

Pre-condition:

- The teacher has logged to the system.
- The teacher has a legal test ID.
- There is no duplicate test ID in existence in the database.

Post-condition:

- A new test is visible in the test list.
- The test information has been added into the database.
- The new test is in the database.

Event sequence:

- The teacher chooses to view the lists of tests in his subject.
- The teacher presses add a new test button.
- Fill in the test information.

- Create the test.

Use case 2:

Number: 2.3

Author's name: Na Zhang

Date: 2004-9-10

Name: Edit a test

Actor: teacher

Description: The use case describes the process of a teacher editing a test.

Pre-condition:

- The teacher has logged to the system.
- There is an activity test in existence in the database.
- The test is visible in the test list.
- The teacher has permission to edit this test.

Post-condition:

- The test information is changed in the database.
- The test questions are changed in the database.
- The test is still in the database.

Event sequence:

- The teacher chooses to view the list of tests in his subject.
- The teacher selects a test to edit.
- The teacher presses the edit button.
- The teacher changes the test information/ adds questions/ deletes questions.
- The teacher saves the test.

Use case 3:

Number: 2.4

Author's name: Na Zhang

Date: 2004-9-13

Name: Delete a test

Actor: teacher

Description: The use case describes the process of a teacher deleting a test.

Pre-condition:

- The teacher has logged to the system.
- There is a test in existence in the database.
- The test is visible in the test list.
- The teacher has permission to delete this test.

Post-condition:

- The test is removed from the test list.
- The test is deleted from the database.

Event sequence:

- The teacher chooses to view the list of tests in his subject.
- The teacher selects a test to delete.
- The teacher presses the delete button.
- The teacher confirms to deletion of this test.
- The teacher presses Yes/No button.

Use case 4:

Number: 2.5

Author's name: Na Zhang

Date: 2004-9-13

Name: Preview a test

Actor: teacher

Description: the use case describes the process of a teacher previewing a test.

Pre-condition:

- The teacher has logged to the system.
- There is an activity test in existence in the database.
- The test is visible in the test list.
- The teacher has permission to preview this test.

Post-condition:

- Go back to the main page.
- The test is still in the database.

Event sequence:

- The teacher chooses to view the list of tests in his subject.
- The teacher selects a test to preview.

- The teacher presses the view button.
- The teacher goes back to main page.

Use case 5:

Number: 2.6

Author's name: Na Zhang

Date: 2004-9-13

Name: View (Mark) a test result

Actor: teacher

Description: the use case describes the process of a teacher viewing (marking) a test.

Pre-condition:

- The teacher has logged to the system.
- There is an activity test in existence in the database.
- The test is visible in the test list.
- The teacher has permission to mark this test.
- The students have taken this test.
- The students' answers are in the database.
- The students' answers and correct answers are visible.

Post-condition:

- The test results have been transferred to the database.
- The scores are saved in the database.

Event sequence:

- The teacher chooses to view the list of tests in his subject.
- The teacher selects a test to mark.
- Press the result button.
- Choose one student's test paper to mark.
- Give the student a score.
- Submit.

Use case 6:

Number: 3.1

Author's name: Na Zhang

Date: 2004-9-13

Name: View a test result

Actor: student

Description: the use case describes the process of a student viewing a test result.

Pre-condition:

- The student has logged to the system.
- The student has taken a specified test.
- The student has permission to view this test result.
- The test result is in existence in the database.

Post-condition:

- *The test result is still in the database.*

Event sequence:

- The student chooses to view the test result.
- The student selects a test to view the result.
- The student goes back to his/her main page.

Use case 7:

Number: 3.2

Author's name: Na Zhang

Date: 2004-9-10

Name: take a test

Actor: student

Description: the use case describes the process of a student making an online test.

Pre-condition:

- The student has logged to the system.
- There is an activity test in existence in the database.
- The test is visible in the test list.
- The student has permission to have this test.
- The student knows the test ID.

Post-condition:

- The student's answers are transferred to the database.

- The test answers are saved in the database.

Event sequence:

- The student has logged to the system.
- The student has typed in the correct test ID.
- The student presses the submit button when he/she has finished the test.
- The student goes back to his/her main page.

6.5.3 Architectural representation - activity diagrams

The UML knows a variant of flowcharts. They are called activity diagrams. Activity diagrams are used to describe the dynamic behaviour of an application. The most common elements of an activity diagram are the following:

A starting point

A starting point indicates the start of an activity diagram and is represented by a filled circle.

Activities

An activity is a task that the application does. Activities are represented by a rectangle with rounded edges. The name of the activity goes inside the rectangle.

Decision points

A decision point represents a condition in the application. Based on whether the condition is true or false, transitions can be made to the relevant activities. A diamond in an activity diagram represents decision points. A description of the decision point appears right next to the diamond.

Transition arrows

A transition arrow is used to indicate the transition of one element in the diagram to the other. This is often a transition between two activities or an activity and a decision point. In the case of a transition from a decision point to

an activity, the transition will have the result of the usual "Yes" or "No". An arrow pointing from the starting point to the ending point represents a transition arrow.

An ending point

An ending point indicates the end of an activity diagram and is represented by an empty circle with a filled circle in it. Figure 6 - 6 represents the test module activity diagram.

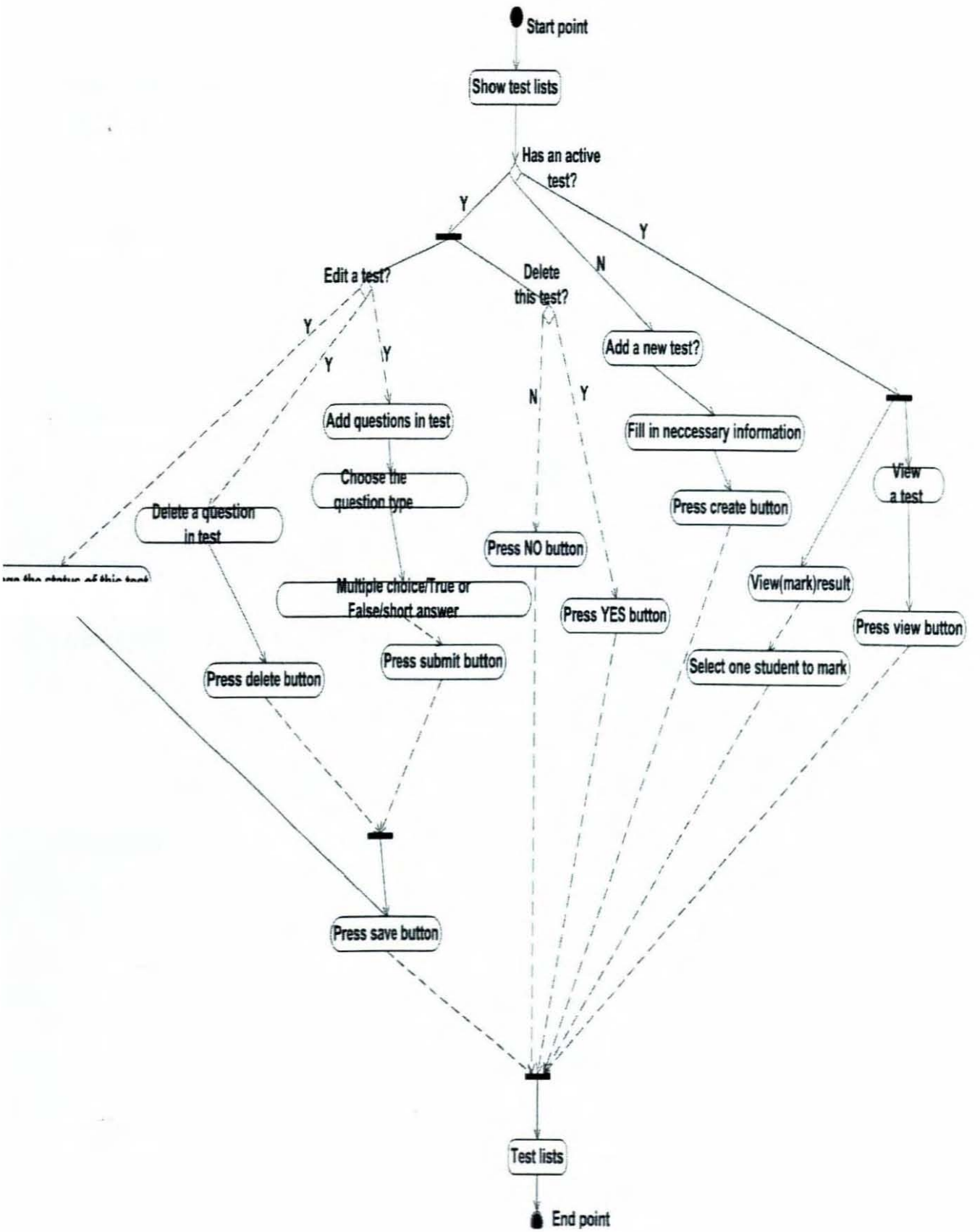


Figure 6-6: The test module activity diagram

6.5.4 Implementation

Table 6-3, Table 6-4, and Table 6-5 show the test module's physical structures

Repository (Modules\ktest\...)	Purpose	Method (function)
\classes\dbtest_test_class_inc.php	Control tests data in table tbl_test_test of dbkewl database	init () addTest () deleteTest() saveTest ()
\classes\dbtest_result_class_inc.php	Control data of students' answers in table tbl_test_stud of dbkewl database	init () addResult()
\classes\dbtest_question_class_inc.php	Control questions data in table tbl_test_question of dbkewl database	init () deleteQuestion() deleteTest() addQuestion ()
\classes\dbtest_mark_class_inc.php	Control marks data in table tbl_test_mark of dbkewl database	init () mark()
\controller.php	Extends the controller base class and controls whole test module	init () dispatch() stud() setForShow() result() testPaper() listQuestion() displayQuestion() getForEdit () listTest()

		displayData() prepForDelete() deleteItem ()

Table 6-3: Test module physical structure 1

Repository (Modules\ktest\...)	Purpose
\register.php	For registration and multilingual support
\tbl_test_test.sql	Insert table tbl_test_test into dbkewl database
\tbl_test_question.sql	Insert table tbl_test_question into dbkewl database
\tbl_test_stud.sql	Insert table tbl_test_stud into dbkewl database
\tbl_test_mark.sql	Insert table tbl_test_mark into dbkewl database

Table 6-4: Test module physical structure 2

Repository (Modules\ktest\...)	Purpose
\templates\content\addMC_tpl.php	For multiple choice question type
\templates\content\ addTest_tpl.php	Add a new test to main page
\templates\content\addTF_tpl.php	For true or false question type
\templates\content\confirmdelete_tpl.php	Confirm delete or not
\templates\content\editTest_tpl.php	Show test questions page
\templates\content\main_tpl.php	Show tests list page for super users
\templates\content\addQuiz_tpl.php	For short answer question type
\templates\content\test_tpl.php	Preview test paper page
\templates\content\result_tpl.php	Show mark test page
\templates\content\stud_tpl.php	Show test paper for students

Table 6-5: Test module physical structure 3

6.5.5 Process view

This section describes the process of test management as an administrator or teacher. Access to a test management page is defined on user level. Only the super users like administrators or teachers can manage tests and the other users can take an existing test at the moment.

Enter ktest (Kewl test) main page as in figure 6-7.

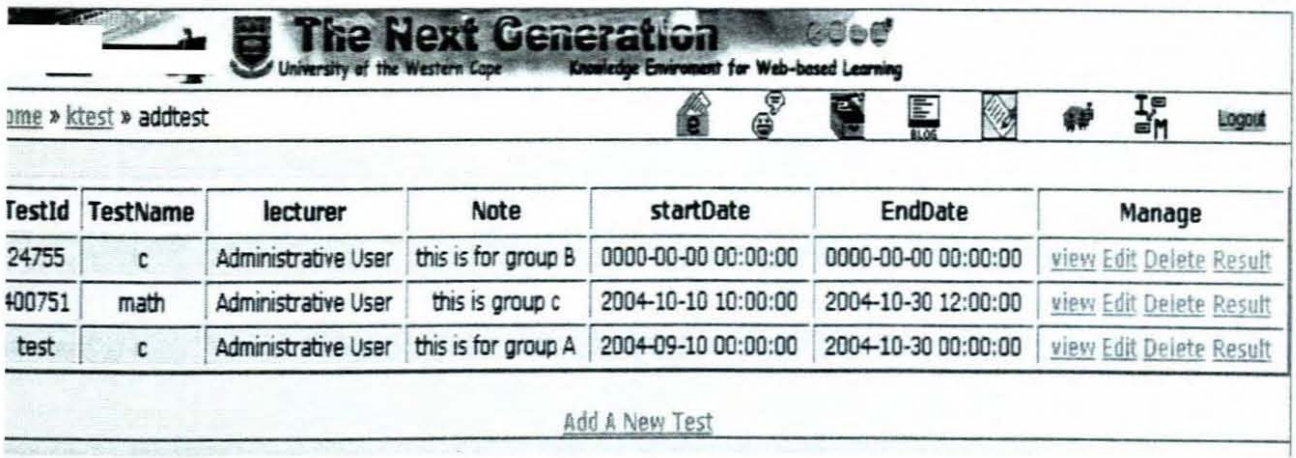


Figure 6-7: Main page of test module

To add a new test:

a) Click add a new test button
b) Fill in the necessary information. The name of the user who has logged in will appear automatically
c) Click create button
d) Return to test module main page

To view:

a) Click view button to preview test paper

To edit:

a) Click edit button. The following is the edit page figure. (See Figure 6-8).
b) It can change any information of the test, like test note, test time

c) It can add questions. Three question types, multiple choice, true or false and short answer can be chosen

d) Choose one of the three question types and click the button

e) Fill in question forms and submit

f) Return to edit page

g) It can delete the questions

h) Click save button and return to main page of test module

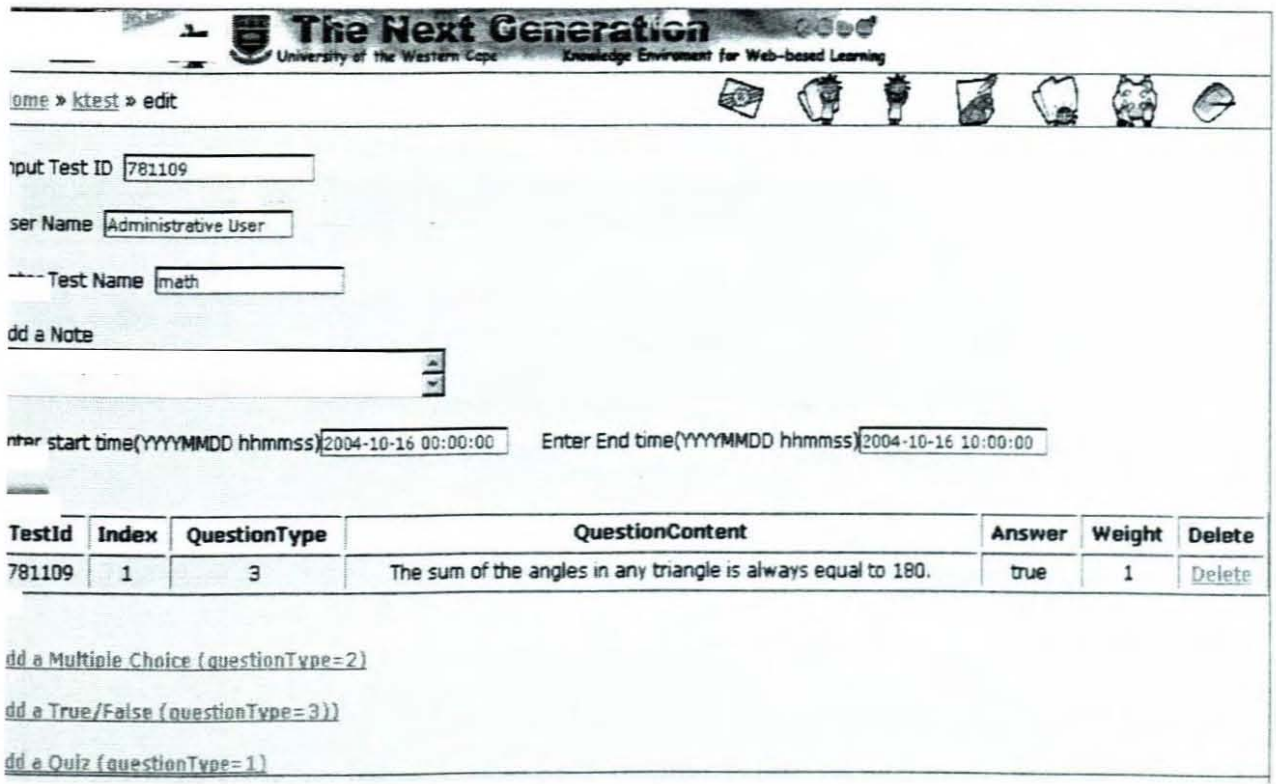


Figure 6-8: Edit page of test module

To delete:

a) Click delete button to delete the test

b) Confirm to delete or not

c) Return to test module main page

To result:

a) Click result button

b) The left part shows students' list, the following is the result page figure. (See Figure 6-9).

c) The right part shows test paper including correct answers, students' answers and additional notes (students can see it as explanation of answer when they view test result)

d) Click a student name on left side, the opposite student's number and name will appear automatically on right side

e) At the same time, the students' answers appear on right side

f) Type in score and submit

g) The score of a student will appear on left side

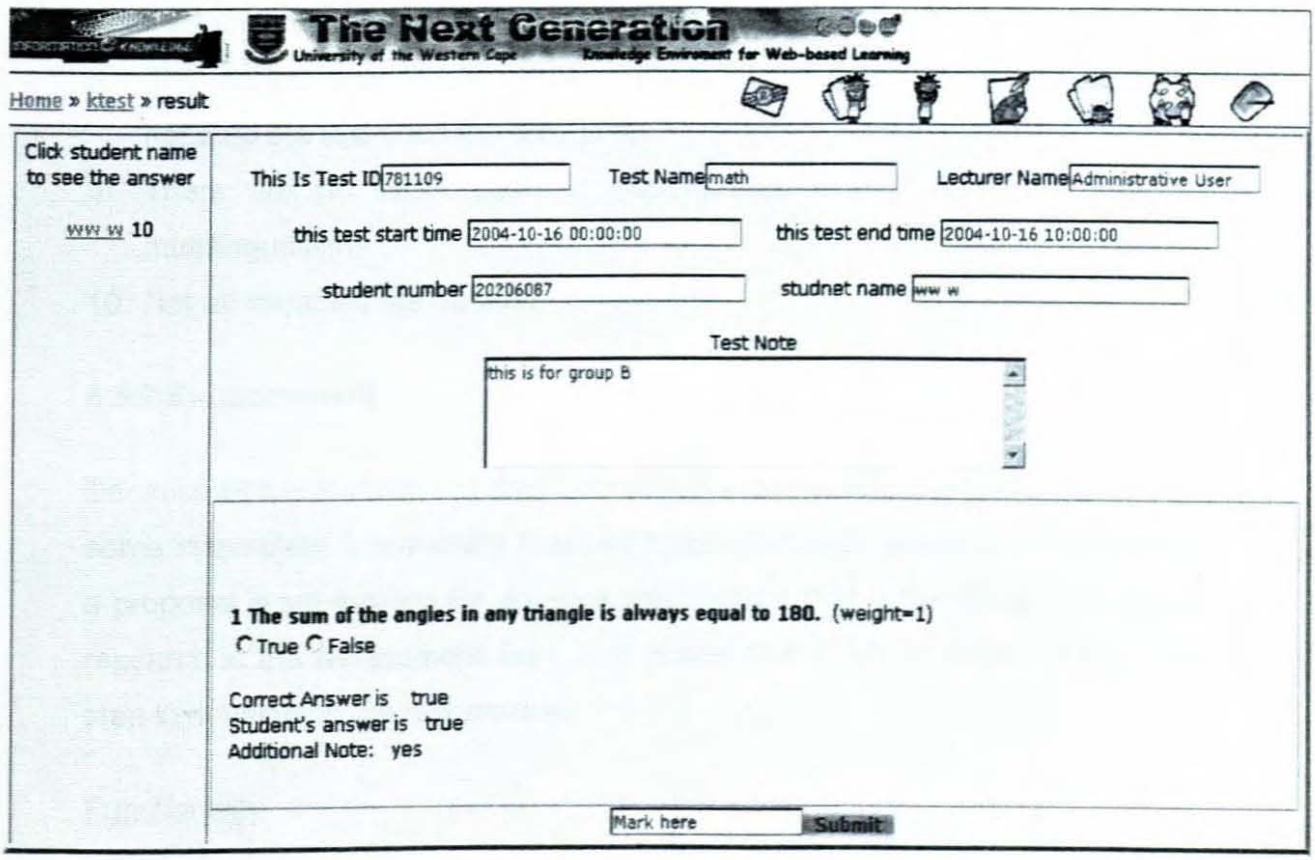


Figure 6-9: Result page of test module

6.5.6 Architectural constraints

Now the test module (name is ktest) has uploaded to http://kngforge.uwc.ac.za through CVS. There are some incomplete functions like the following:

1. No course or subject interface.

2. The interface of the student is temporary.
3. Student interface should see score, correct answer, student's answer and additional note. At the moment, the super user can do them as well as mark.
4. No student can do the same test twice, otherwise the questions and answers will conflict.
5. The teacher can mark manually. In another words, the module cannot be automatically marked, so the student cannot get a score immediately.
6. Some text areas are writable, because the team could not find attributes in the htmlelements. E.g. in test paper page, test ID, name and note, student name and number, teacher name and test time should be un-writable.
7. Test time must be filled in like (yymmdd hhmmss).
8. The module doesn't keep track of the time remaining for the test, and will not stop the test once the time is up.
9. There are no TEXT parts in register.php, thus it does not support multilingualism.
10. Not all variables are defined.

6.5.7 Enhancement

Because of the limitation of time and self-directed ability, the test module has some incomplete functionality that has been discussed above. In this section, a proposal is set forward for an advanced feature that is the result of in-depth research in the assessment field. It is hoped that it will be useful in the next step in developing the test module.

Functionality

1. Enhance the test question type:
   - Drag and drop
   - Clicking
   - Matching
2. The student can get test score online.
3. A link to calendar on preview test paper page for super users or on student interface page, will remind the users on which day the students will have

the test.

4. Fix above bugs.

Interface

A good interface is important and it should be perfected in the interface design.

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

This section will contain the conclusions and recommendations. Six questions will be answered: What is the primary purpose of this research project? What objectives have been achieved or not completely achieved in this project? Have the functionalities been achieved as per design? How is the quality of this test module measured? What specific technology frameworks have been used to develop this project? What lessons were learned in the project management of this project? Their answers will conclude this dissertation. Then special and meaningful recommendations will be made to the partners and potential users in industry and commerce, followed by recommendations for further research.

7.1.1 The primary purpose of this research project

The primary purpose of this research project was to develop a flexible test management system module in an open source environment and to create a test module program with appropriate software functionality and features to meet target users' requirements. The aim was to produce an easy-to-use tool that offers teachers the ability to create and administer customised tests and to create a friendly test environment for students. A further aim was to establish more effective distribution of assessment resources and offer other business firms the opportunity to use the program, to alter it, and to redistribute it all without charge, for employee training purposes.

KEWL is an Active Server Pages (ASP) application that runs under any Microsoft server that supports ASP, but it is open source because it is released under the Gnu Public Licence (GPL). KEWL.NextGen will convert it entirely to PHP that runs under the Lunix server. It will give users the basis from which to build the most advanced learning management system in the world, using talent largely that exists within African higher education, other organisations, businesses, and partners around the world.

7.1.2 The objectives that have been achieved

This project set out to achieve the following goals and objectives:

- To resolve the problems those were identified in this dissertation.
- To implement an assessment management logic model.
- To develop a flexible assessment management system module in an open source environment.
- To enhance instructional design features, making online assessment easier and more convenient.
- To design specific assessment tool utilities.
- To understand the technology used in Linux System, PHP, and MySQL.
- To create an assessment management system module programme.
- To provide an easy-to-use tool that offers teachers the ability to create and administer customised assessment and to create a friendly assessment environment for students.
- To develop an assessment management system module that provides flexible design (interoperability) for future adaptability in an OSS environment.
- To develop an assessment management system module that provides adequate user support provisions.
- To develop an assessment management system module with appropriate software functionality and features.
- To develop an assessment management system module that meets architectural constraints of an existing LMS.
- To develop an assessment management system module that meets target users' requirements.

These objectives were met / not completely achieved:

- To expand assessment functions.
- To determine the actual assessment problems through literature study or focus group, or questionnaire.
- To offer other business firms the opportunity to use it, alter it, and redistribute it all without charge for employee training purposes.

- To allow students to simulate the actual examination experience.

7.1.3 The functionalities that have been achieved as per design

The test module achieved its basic function. The test module allows teachers to create and manage online tests with full account access to the test module. By choosing a test on the test list, they can preview, modify, delete and mark an existing test, as well as create a new test.

The user can add a new test, and fill in some necessary information. Test notes depend on teacher who would like to give information to students about this test; as well as have the starting and ending time of the test. The name of the teacher who created this test will appear automatically. All of this information will be shown on the test paper automatically. The teacher can delete an existing test in his/her subject, but the test results are still kept in the database even after a teacher has deleted the test. It means that the students' scores will not disappear; they will be preserved for grading purpose or future reference.

The edit the test function, allows the teacher to add/delete questions and change test information. Three question types have been developed in the test module – multiple-choice, true/false, and short answer. When a teacher adds a question, he/she should input question number, question content, correct answer, additional note and weight of this question. A multiple-choice question type can have one or multiple correct answers. Also, the teacher has permission to delete questions and change test information. A teacher can preview the test paper after he/she has edited a test. It provides inspection of the test paper quality when the teacher checks this test paper.

The teacher can read all the test papers that a student has taken, using the mark function. It will show the correct answer, the students' answer and additional notes. So it will be convenient for the teacher to mark and compare correct answers and students' answers. Also it clearly shows which question has a high/low wrong or right percentage. Then the teacher will be able to emphasise an aspect of high wrong percentage of questions in future teaching.

Therefore, besides being able to have an online test, students will have the opportunity to view the results of their test papers. Correct answers, students' answers and additional notes will be shown. Additional notes will help to explain the answers. It is a useful tool in enhancing the students' quality of learning.

7.1.4 Measuring the quality of the test module

These above functionalities have been achieved and the delivery of data was normally effective. Alpha testing of the test module has been carried out constantly through the process on the local computer. It works perfectly. Entire code sources have been uploaded to kngforge Website through CVS. Kngfore Website updates every hour from the code repository in CVS. Its main purpose is for the developers to have a common platform for the testing code. Because it is updated from the code under development, it is not uncommon for this site to be broken entirely, or for some features not to be working. Once UWC developers have made the first Beta release they will have another site that will be updated only when a new version or patch is released. Therefore, the beta testing will depend on the other developers of UWC using the prototype in a variety system on the Kngforge Website. The problems encountered and suggestions made will inform a further process of refinement. (The kngforge Website is http://kngforge.uwc.ac.za/).

7.1.5 The specific technology frameworks that have been used to develop this project

Because this project is based on teamwork, a central server was needed, which all members could access to server as the repository for the code. The CVS - Concurrent Versions System has a built-in client-server access method so that any developer who can connect to Internet can access files on a CVS server. It is useful for everyone in this large distributed team. The history of the sources file can be recorded using CVS, which means the members are able to keep track of all the changes that have been made to a programme during the entire development time. A mailing list, namely

nextgen-online@cvs.uwc.ac.za is used, as communication method. Any problems, suggestions and information will be sent to nextgen-online@cvs.uwc.ac.za every day. A variety of issues are discussed through the mail.

Object-oriented programming in PHP language is used. 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. 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 are changed often during development leads to more robust applications that are easier to maintain.

7.1.6 Lessons that were learned in the project management of this project

Based on previous detailed analysis of project methodologies it was seen that the best methodology to use would change, depending on the particular project, the situation and objectives. Some situations could require a hybrid approach. Especially with regard to providing more detailed and exact estimates of long-term energy efficiency improvement potentials, technological learning over time and related costs will require an in-depth analysis and partly innovative methodological approaches.

7.2 Recommendations

It is recommended to KEWL.NextGen partners in terms of the test module that they:

- Petition the KEWL.NextGen development team to review and comment upon the test module.
- Use the above report in testing the module.
- Examine and fix the specific bugs.
- Furthermore, develop and perfect functionality, interface, even structure.

Against the background of the thematic issues discussed, there are more recommendations, namely:

- that more support of communities for education should be encouraged;
- that academics support and encourage free internet access for students;
- that a stable financial support of government and society for education should be ensured;
- that more volunteers contribute to the open source world; and
- that a more efficient project methodology be created to develop software engineering.

LIST OF REFERENCES

Web resources:

[1] Linux Homepage. Retrieved: April, 16, 2003, from http://www.linux.org

[2] GNU project Web server Homepage. Retrieved: April, 20, 2003, from http://www.gnu.org

[3] PHP Homepage. Retrieved: May, 03, 2003, from http:// www.php.net

[4] MySQL Homepage. Retrieved: May, 05, 2003, from http://www.mysql.com

[5] Apache Homepage. Retrieved: May, 20, 2003, from http://www.apache.org

[6] African Virtual Open Initiatives and Resources Homepage. Retrieved: June, 07, 2003, from http://avoir.uwc.ac.za

[7] Apache Friends Homepage. Retrieved: July, 27, 2003, from http://www.apachefriends.org

[8] Apache Friends XAMPP for Windows Version. Retrieved: July, 28, 2003, from http://www.apachefriends.org/en/xampp-windows.html

[9] Introduction to CVS. Retrieved: August, 03, 2003, from http://www.gnu.org/software/cvs/

[10] TortoiseCVS Homepage. Retrieved: August, 19, 2003, from http://www.tortoisecvs.org/

[11] MySQL Developer Zone. Retrieved: August, 30, 2003, from http://dev.mysql.com/dowloads/mirros.html

[12] Apache Portable Runtime Project download page. Retrieved: September, 07, 2003, from http:// apr.apache.org/download.cgi

[13] ActiveState Perl download page. Retrieved: October, 1, 2003, from http://www.activestate.com/Products/ActivePerlProStudio/

[14] The PHPMyAdmin Project Homepage. Retrieved: October, 19, 2003, from http://www.phpmyadmin.net

[15] NetTel @ Africa project (Network for Capacity Building and Knowledge Exchange in ICT Policy, Regulation and Applications) Homepage. Retrieved: October, 20, 2003, from http://www.nettelafrica.org/ and http://elearn.nettelafrica.org/

[16] The value of formative assessment. Retrieved: October, 21, 2003, form

http://www.fairtest.org/examarts/winter99/k-forma3.html

[17] Stanford University course work. The standard course management system. Retrieved: November, 09, 2003, from http://aboutcoursework.stanford.edu/

[18] Whiteboard Courseware System Project Homepage. Retrieved: March, 06, 2004, from http://whiteboard.sourceforge.net/

[19] Technical report. Retrieved: June, 02, 2004, from http://www.io.com/~hcexres/tcm1603/acchtml/techreps.html

Book resources:

[1] Mcleod, G. & Smith, D. 2001. Managing Information Technology Projects. South Africa. Inspired Press.

[2] Meloni, J.C. 2003. SAMS teach yourself PHP, MySQL and Apache in 24 hours. Sams Pbu.

[3] Welling, L. 2003. PHP and MySQL Web development. Indianapolis, Ind: Sams.

[4] Welling, L. & Thomson, L. 2003. PHP and MySQL Web Development. Sams Pbu.

[5] Ray, D.S. & Ray E. J. 1999. Mastering HTML 4.San Francisco: Sybex. Laurie, B. & Laurie, P. 2003. Apache The Definitive Guide

[6] Lee, J. & Ware, B. 2003. Open Source Web Development with LAMP Using Linux, Apache, MySQL, Perl, and PHP. London: Addison-Wesley.

[7] Bulger, B. 2002. MySQL/PHP database applications. New York. Chichester: Wiley

[8] Williams, H. E. 2002. Web database applications with PHP and MySQL. Cambridge [Mass]:O'Reilly.

Appendix A: An evaluation of the development of a test management module against the Information Management Body of Knowledge (IMBOK) framework

1. Summary

In this research project, a test management system module had to be designed and implemented to function within the architectural constraints of a developing Open Source Software (OSS) Learning Management System (LMS), which is called Knowledge Environment for Web-based Learning Next Generation (KEWL.NextGen).

The test system is a module developed on Object-Oriented Programming (OOP) in PHP and MySQL application. The scripts of this test module are written under the KEWL.NextGen' application framework, which is based on a close approximation of model, view and controller (MVC) design pattern. Now the entire source coding of the test module has been uploaded to Web site http://kngforge.uwc.ac.za.

Bytheway (2004) proposed the Information Management Body of Knowledge (IMBOK) model, which provides researchers a means to organise the literature about the subject and it is hoped that the fruits of present and future research will accumulate within and around the IMBOK framework. Using it, it is hoped that future students, professionals and managers will have easier access to the diverse tools and techniques that they need; furthermore, researchers will have the means to position their research ideas and to share them with others, more effectively than would otherwise have been possible. So this article is an evaluation of development of the test module against IMBOK framework.

2. Introduction to IMBOK

(Bytheway 2004, http://www.imbok.org)

2.1 What is IMBOK?

IMBOK is a living collection of ideas and tools that will assist managers who are concerned to get the best out of their information technology and information systems investments. It has emerged from research work undertaken in a partnership between the University of the Western Cape, and Cape Technikon.

IMBOK provides an easily understood framework that relates business needs to new information technology opportunities in a relatively simple, staged way. It allows researchers to position and isolate problems, and to grasp opportunities. It allows ideas to move more easily from one domain to another - from consideration of raw technologies right through to issues of business practice and business strategy.

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". So, it should be useful for: Students who are looking for a simple contextual framework within which to organise their ideas.

2.2 IMBOK framework overview

The Framework is presented as five areas of knowledge and four areas of process. See Figure 1.

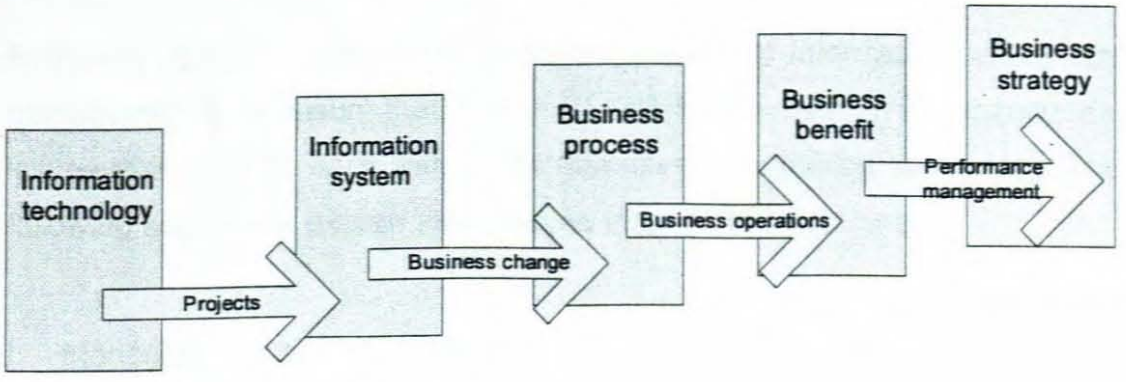


Figure 1: The IMBOK framework

3. The knowledge areas

3.1 Information technology

Bytheway (2004:18) stated that Information Technology refers to the specific technical components, normally organised as hardware, software and communications, which are used to make up an information system.

3.1.1 Nature of technology

In the case of the test module of KEWL.NextGen, information technologies were used such as listed in Table 1.

Hardware	Software	Communications
Central processing unit	Linux operating system	Internet
Disk drive	MySQL database	CVS
Printer	Apache Web server	Mailing list
Keyboard	PHP Hypertext Processor	Terminal
USB port	PHPEdit	
	PEAR	
	Concurrent Versions System	

Table 1: Technologies application in test module

3.1.2 Managerial issues concerning information technology

Bytheway stated that the most important aspect of information technology management is to ensure that IT is used well. He pointed out five managerial issues that must be dealt with in the domain of information technology. The following section will explain these issues in terms of this project.

I. Managing suppliers: the developers of UWC provided the KEWL.NextGen's technical framework, which is like the base of the building. But it didn't provide the blueprint, in another words, no requirement of test module in detail was given. The KEWL.NextGen

project was still in the very early stages, so there was no course module to be built.

II. Acquiring technology: this project is about open source software, so the most of software is provided freely from Internet. UWC provided the technical support about technology of framework, which provided the criteria of technical design, such as naming convention, structure of files and constructor, etc.

III. Managing the technology portfolio: this project is based on teamwork, a central server was needed, which all members could access to server as the repository for the code. So Concurrent Versions System (CVS) have a built-in client-server access method.

IV. Technology competency: this project uses Kngforge Website for the developers to have a common platform for the testing code.

V. Budget management: this project is in fulfilment of the requirements for the degree Master of Technology in the Faculty of Information Technology at the Peninsula Technikon. The team of Peninsula Technikon is a volunteers to develop KEWL.NextGen.

3.2 Information system

"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." (Bytheway 2004:23).

In the case of the test module system, the information system offers teachers the ability to create and administer customised tests and to create a friendly test environment for students. Figure 1 visually explains the test module system.

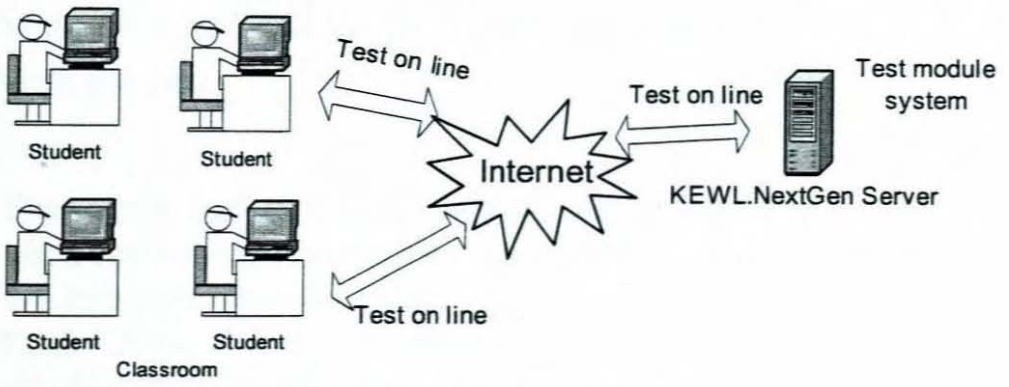


Figure 1: Test Module System

3.2.1 Applications

This test module can be used for the creation and management of an online test. It provides facilities to create a new test, and to preview, edit, delete and mark an existing test. Three question types have been achieved, namely multiple-choice, true/false and short answer. Figure 2 visually describes the functionalities of test module for teachers.

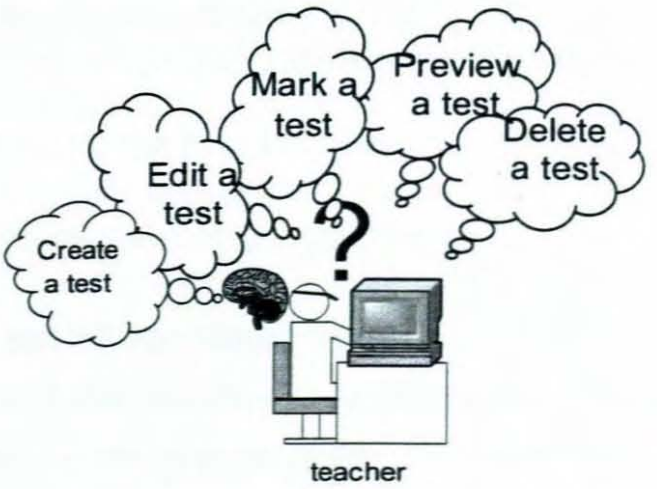


Figure 2: Applications of Test Module for Teachers

3.3 Business processes

"A business process is 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 2004:60).

In the case of test module project, the lists that follow will describe the main research activities:

- Study Linux, PHP, and MySQL and other such technologies
- List a questionnaire for teachers and students
- Get information from the UWC
- Research the assessment method
- Identify software project plan
- Create storyboard to meet UWC requirements
- Create preliminary programme
- Debug and test prototype programme

The checklist on the next section is related to above checklist. It explains outputs of activities processes briefly:

- Series of working papers on learning to install Linux, using PHP and MySQL
- Administrative processes for test
- Promotional Web and promotional material to the wider portion of the chosen target group
- Develop and run the test module system
- A test module system programme
- Business partner agreement to support work

3.3.1 Processes and the organisation chart

Figure 3 covers the KEWL.NextGen project management processes and organisation structure. The modules project was developed by sub-project.

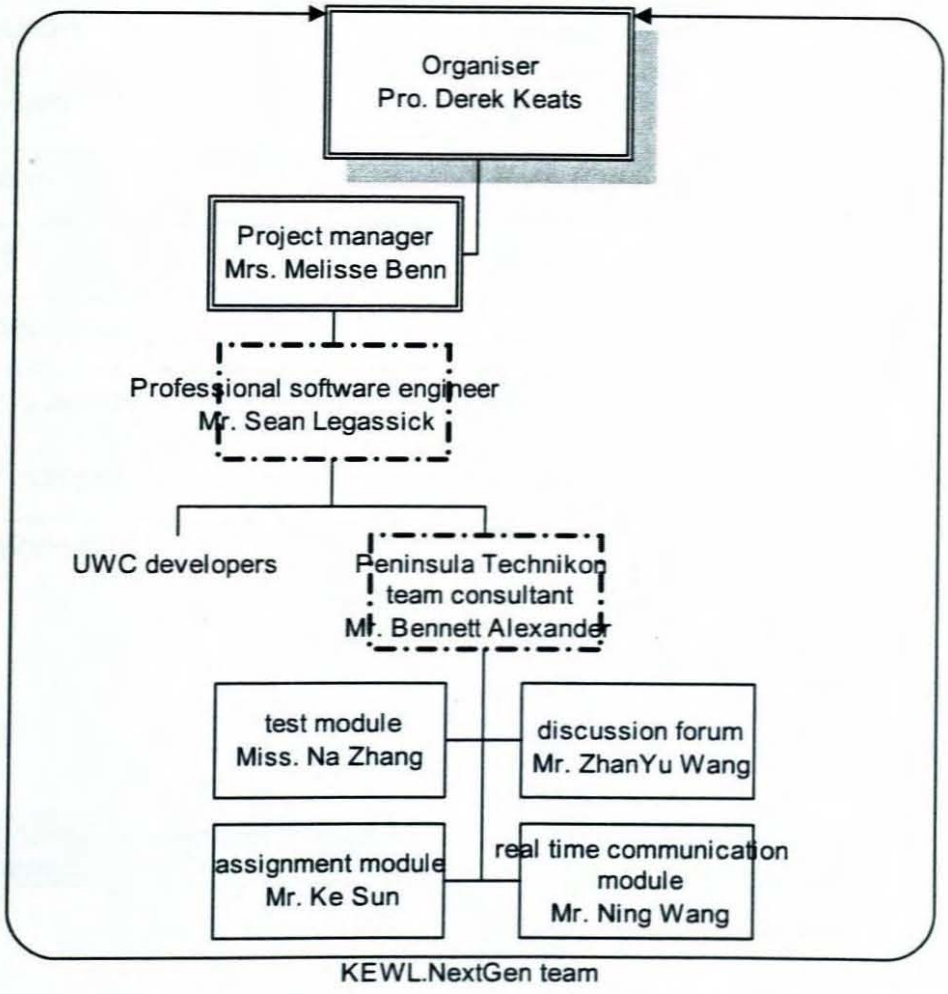


Figure 3: KEWL.NextGen Project Organization Chart

3.4 Business benefit

"The process of organising and managing, that the potential benefits of an investment of time and effort are actually." (Bytheway 2004:90).

3.4.1 Costs and Benefits

Table 2 and Figure 4 below - illustrate a simple cost-benefit analysis.

Resources	Costs
Hardware	
Software	Free (Open Source Software)
Books	
Internet	
Printing and copy	
Other licences	Free (GUN Public Licence)

- Money

Table 2: Costs of the Project

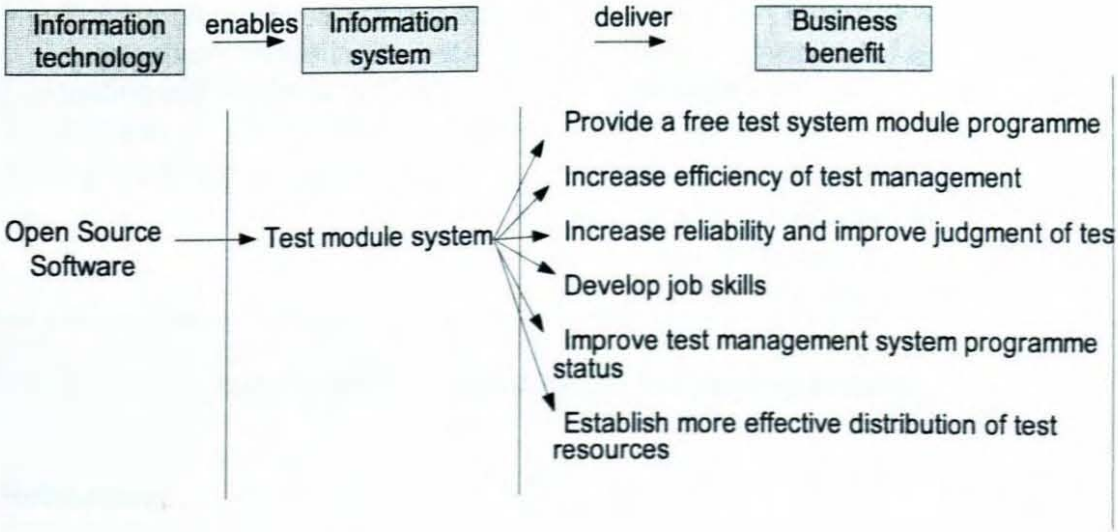


Figure 4: Transforming Information Technology into Potential Business Benefits

3.5 Business strategy

Bytheway (2004:132) stated that the analysis of Strengths, Weaknesses, Opportunities and Threats (SWOP) is one of the most straightforward and frequently deployed techniques for strategic analysis. Table 3 described simple analysis of this project using SWOP.

Strengths	Weaknesses
• Low cost options relative to the other test system • Potential for growth • Easy to maintain • Three questions types can be chosen • Easy to use • Adopt up-to-date design pattern (MVC) • comment and structures of code source are clear • Basic functionality has been achieved as per design	• The teachers mark manually • The module doesn't keep track of the time remaining for the test • Not support multilingualism • Sometime it is not easy communicate with the other developers of UWC
Opportunities	**Threats**
• Improve communication ability • The problems encountered and suggestions made in the beta testing will inform a further process of refinement • Partnerships more effectively	• Inadequate progress with technical infrastructure • Add primary key of a table have to be called "id" • Not all variables are defined in code source • No student can do the same test twice; otherwise the questions and answers will conflict.

Table 3: SWOT Analysis for the Test Module Project

References

1. Bytheway A, 2004, the Information Management Body of Knowledge, Cape Town, South Africa.

2. The Information Management Body of Knowledge, Bytheway A. Retrieved: November, 3, 2004, from http://www.imbok.org.

APPENDIX B: KEWL.NextGen Learning Management System

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.

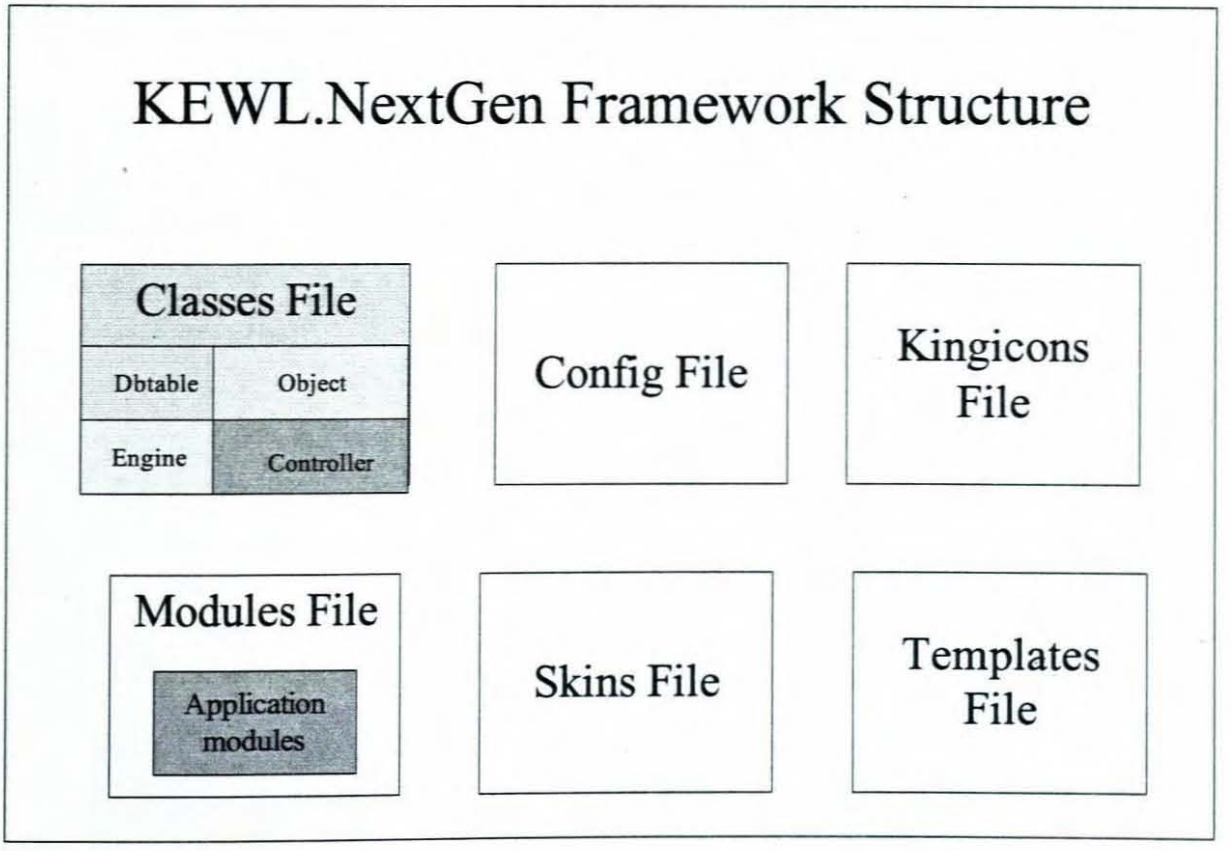


Figure 2: KEWL.Nextgen Framework Structure

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.

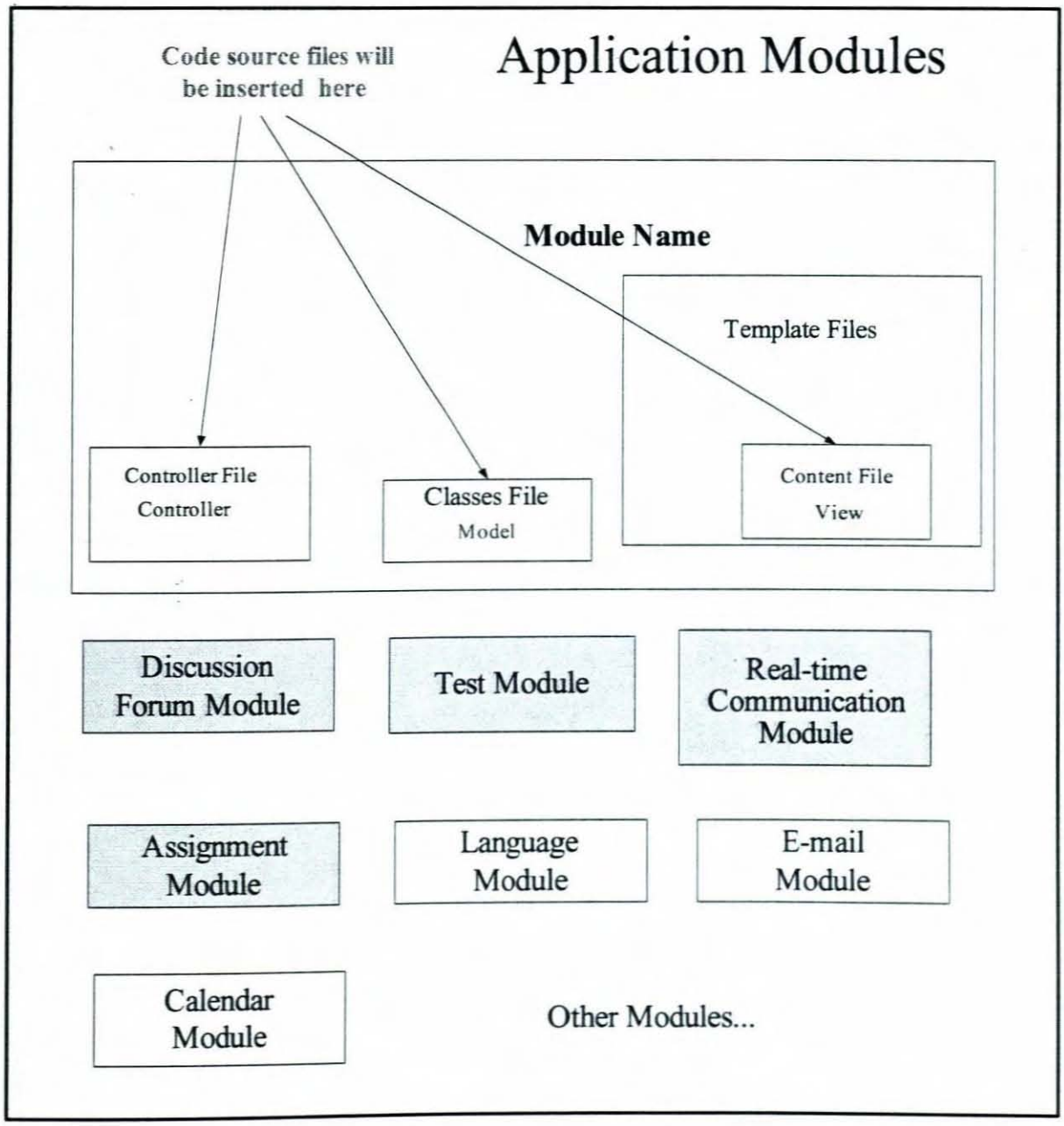


Figure 3: Application Modules Architecture

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.

APPENDIX C: Glossary

Glossary:

Linux: An operating system

PHP: Hypertext Processor, open source server-side scripting language for Web servers

MySQL: A Relational Database Management System.

Apache: Open source Web server software

PEAR: A framework and distribution system for reusable PHP components.

GNU: GNU is a recursive acronym for "GNU's Not UNIX"

SSL: A protocol that encrypts a single TCP session

KDE: A powerful Free Software graphical desktop environment for Linux and Unix workstations

Appendix D: Code of the test module

```
<?php
if (!$GLOBALS['kewl_entry_point_run']) {
    die("You cannot view this page directly");
}
/**
* ktest class for KEWL.NextGen
***/
class ktest extends controller
{
 var $objUser;
    var $objButtons;
    var $objConfig;

 function init()

 {
  $this->objButtons = & $this->getObject('navbuttons', 'navigation');
       $this->objUser = & $this->getObject('user', 'security');
       $this->objLanguage = & $this->getObject('language', 'language');
       // Create the configuration object
       $this->objConfig = & $this->getObject('config', 'config');

     $this->href = & $this->getObject('href','htmlelements');
        $this->table =$this->questionTable= &
       $this->getObject('htmltable','htmlelements');
 $this->objDbTest = & $this->getObject('dbTest_test');

 $this->objDbQuestion = & $this->getObject('dbTest_question');
 $this->objDbMark= & $this->getObject('dbTest_mark');
 $this->objDbResult = & $this->getObject('dbtest_result');
 }

 function dispatch($action)
{
 // retrieve the mode from the querystring
    $mode = $this->getParam("mode", null);
   // retrieve the sort order from the querystring
    $order = $this->getParam("order", null);
```

```
switch ($action) {
    //for defult entry can add ID checking here
 case null:
 $id=$this->objUser->userId();
 if($id==1){
 $this->setForShow($order);
      return "main_tpl.php";}
 return "stud_tpl.php";
       break;
 //to review the whole test paper
 case "view":
 $this->setvar('outStr', $this->testPaper());
 return "test_tpl.php";
 break;

 // save the students' result
 case "dotest":
    $this->objDbResult->addResult();
 $id=$this->objUser->userId();
 if($id==1){
    $this->setForShow($order);
         return "main_tpl.php";
    }
 return "stud_tpl.php";
      break;

 case "add": //main page "add"
 $this->setvar('userName', $this->objUser->fullName());
 $this->setvar('mode', 'add');
       return 'addTest_tpl.php';
       break;

case "addtest":  //create test
$testId = $this->getParam('testId',NULL);
if($testId){
 $mode = $this->getParam("mode", NULL);
        $this->objDbTest->addTest();
 }
$this->setForShow();
       return "main_tpl.php";
       break;
```

```
    //edit the test go to the edit page
 case "edit":
 $this->getForEdit("edit");
 $this->setForShow();
        return 'editTest_tpl.php';
        break;

 case "delete":
 // retrieve the confirmation code from the querystring
        $confirm=$this->getParam("confirm", "no");
        if ($confirm=="yes") {
             $this->deleteItem();
             $this->setForShow();
             return 'main_tpl.php';
           } else {
   $this->prepForDelete();
             return 'confirmdelete_tpl.php';
           }
        break;

 //add a short answer question
 case "addq":
 return 'addQuiz_tpl.php';
 break;

 //add a mutiple choice
 case "choice":
 return 'addMC_tpl.php';
 break;

 case "addt":
 return 'addTF_tpl.php';
 break;

//add the question to the database according to question type
 case "quiz":
 $this->getForEdit("edit");
 $testId = $this->getParam("testId",NULL);
 if(!$testId){
    $testId = $this->getParam('testId',NULL);
   }
```

```
 if($testId){
  $mode = $this->getParam("mode", NULL);
         $this->objDbQuestion->addQuestion();
   }
 $this->setForShow();
        return 'editTest_tpl.php';
 break;

 //save the test and question
 case "save":
 $this->objDbTest->saveTest();
        $this->setForShow();
        return "main_tpl.php";
 break;

 //dequs is for delete questions in test
 case "dequs":
 $this->objDbQuestion->deleteQuestion();
 $this->getForEdit("edit");
 $this->setForShow();
        return 'editTest_tpl.php';
 break;

/result can see all the students test answer and give the students marks
case "result":
 $studId = $this->getParam('studId',NULL);
 $this->getForEdit("edit");
 $this->setvar('outList', $this->stud());
 $this->setvar('outStr', $this->result($studId));
return "result_tpl.php";
break;

//give the selected student mark and return to the result page
case "mark":
 $this->objDbMark->mark();
 $studId = $this->getParam('studId',NULL);
 $this->getForEdit("edit");
 $this->setvar('outList', $this->stud());
 $this->setvar('outStr', $this->result($studId));
return "result_tpl.php";
break;
```

```
default:
 echo
s->objLanguage->languageText("phrase_unrecognizedaction");
 break;



list the students that have done the test and the marks
ction stud()

estId=$this->getParam('testId', NULL);
    $sql="select studId, studName, result from tbl_test_stud where
    testId='$testId' order by studName";
tud=$this->objDbResult->getArray($sql);
d=0;
tudList="Click student name<br>to see the answer<br><br>";
reach($stud as $line){

f($id!=$line['studId']){
studId=$line['studId'];
markId=$testId.$studId;
mark=$this->objDbMark->getRow('markId',$markId);

score=$mark['score'];
studName=$line['studName'];
paramArray = array('action' =>
ult','testId'=>$testId,'studId'=>$studId,'studName'=>$studName);
     $result=$line['result'];
name=$this->href->showlink($this->uri($paramArray ,
st"),$studName.$result,'');

studList.=$name.$score."<br>";
id=$studId;}

turn $studList;



repare to show the testlist and the question list
ction setForShow()

  $order = $this->getParam("order", null);
```

```
  // Define a variable to print the list
       $this->setvar('outStr', $this->listTest($order));
     $this->setvar('outQuestion', $this->listQuestion($order));
    }

  //make a reslult paper according to studentId
   function result($stud)
 {
  $testId=$this->getParam('testId', NULL);
  $sql="SELECT questionIndex, studAnswer from tbl_test_stud where
testId='$testId' and studId='$stud' Order by questionIndex";
   //get student's answer
  $ans=$this->objDbResult->getArray($sql);

  $key = "testId";
  $keyvalue = $testId;
  $inf = $this->objDbTest->getRow($key, $keyvalue);

  $this->setvar('testId', stripslashes($inf['testId']));
       $this->setvar('testName', stripslashes($inf['testName']));
       $this->setvar('testOwner', stripslashes($inf['testOwner']));
       $this->setvar('testNote', stripslashes($inf['testNote']));
       $this->setvar('startDate', stripslashes($inf['startDate']));
       $this->setvar('endDate', stripslashes($inf['endDate']));

       $sqlQues="SELECT
questionId,questionIndex,questionType,questionContent,answer,altA,alt
B,altC,altD,weight, questionNote FROM tbl_test_question Where
testId='$testId' Order by questionIndex";

  $qus=$this->objDbQuestion->getArray($sqlQues);
  //test page part
  $this->loadClass("form","htmlelements");
   $this->loadClass("button","htmlelements");
   $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
  $this->loadClass("input","htmlelements");
  $this->loadClass("htmlheading","htmlelements");
   $this->loadClass("dropdown","htmlelements");
  $this->loadClass("radio","htmlelements");
  $this->loadClass("checkbox","htmlelements");
```

```
  foreach ($qus as $paper){
   $questionIndex = $paper['questionIndex'];
   $questionType = $paper['questionType'];
   $question = nl2br(stripslashes($paper['questionContent']));

   $note=$paper['questionNote'];
   $answer=$paper['answer'];
   $a = $paper['altA'];
   $b = $paper['altB'];
   $c = $paper['altC'];
   $d = $paper['altD'];
   $weight = $paper['weight'];
   $studAnswer=$ans[$questionIndex-1]['studAnswer'];
    //display question
   $test.=
"<br><br><b>".$questionIndex." ".$question."</b>"." "." "
."("."weight=".$weight.")"."<br>";

   //according questionType to output
   if($questionType == 2){   // choice question
    $this->test = & $this->getObject('layer', 'htmlelements');

    $checkBox= new checkbox($questionIndex."A","A");
    $checkA=$checkBox->show();
    $test.= $checkA." "."A"." ".$a."<br>";

    $checkBox= new checkbox($questionIndex."B","B");
    $checkB=$checkBox->show();
    $test.= $checkB." "."B"." ".$b."<br>";

    $checkBox= new checkbox($questionIndex."C","C");
    $checkC=$checkBox->show();
    $test.= $checkC." "."C"." ".$c."<br>";

    $checkBox= new checkbox($questionIndex."D","D");
    $checkD=$checkBox->show();

    $test.= $checkD." "."D"." ".$d."<br>";
    $test.="<br>"."<font color=red>"."Correct Answer is
   ".$answer."</font>"."<font
```

```
olor=black>"."<br>Student's answer
;   ".$studAnswer."</font>"."<font
olor=black>"."<br>Additional
ote:   ".$note."</font>"."<br>"."<br>";
    }

  if($questionType ==3){   //  T/F question

    $radio= new radio($questionIndex."TrueFalse");
      $radio->addOption('1','True');
    $radio->addOption('2','False');
    $test.= $radio->show()."<br>";
    $test.="<br>"."<font color=red>"."Correct Answer is
ibsp  ".$answer."</font>"."<font
olor=black>"."<br>Student's answer
s   ".$studAnswer."</font>"."<font
olor=black>"."<br>Additional
ote:   ".$note."</font>"."<br>"."<br>";
        }

  if($questionType ==1){    //quiz

    $test.="<br>"."<font color=red>"."Correct Answer is
ibsp  ".$answer."</font>"."<font
olor=black>"."<br>Student's answer
s   ".$studAnswer."</font>"."<font
olor=black>"."<br>Additional
ote:   ".$note."</font>"."<br>"."<br>";
    }

 } //for output
     return $test;
}

//make a test page
function testPaper(){

 $testId=$this->getParam('testId', NULL);
 // test paper information
 $key = "testId";
 $keyvalue = $testId;
```

```
inf = $this->objDbTest->getRow($key, $keyvalue);


this->setvar('testId', stripslashes($inf['testId']));
    $this->setvar('testName', stripslashes($inf['testName']));
    $this->setvar('testOwner', stripslashes($inf['testOwner']));
    $this->setvar('testNote', stripslashes($inf['testNote']));
    $this->setvar('startDate', stripslashes($inf['startDate']));
    $this->setvar('endDate', stripslashes($inf['endDate']));


    $sqlQues="SELECT
stionId,questionIndex,questionType,questionContent,answer,altA,alt
ltC,altD,weight, questionNote FROM tbl_test_question Where
tId='$testId' Order by questionIndex";

qus=$this->objDbQuestion->getArray($sqlQues);
/test page part
this->loadClass("form","htmlelements");
$this->loadClass("button","htmlelements");
$this->loadClass("textarea","htmlelements");
this->loadClass("textinput","htmlelements");
this->loadClass("input","htmlelements");
this->loadClass("htmlheading","htmlelements");
$this->loadClass("dropdown","htmlelements");
this->loadClass("radio","htmlelements");
this->loadClass("checkbox","htmlelements");

oreach ($qus as $paper){

$questionIndex = $paper['questionIndex'];
$questionType = $paper['questionType'];
$question = nl2br(stripslashes($paper['questionContent']));

$a = $paper['altA'];
$b = $paper['altB'];
$c = $paper['altC'];
$d = $paper['altD'];
$weight = $paper['weight'];
//display question
$test.=
```

```
r><br><b>".$questionIndex." ".$question."</b>"." "." "
"."weight=".$weight.")"."<br>";

//according questionType to output
if($questionType == 2){   // choice question
 $this->test = & $this->getObject('layer', 'htmlelements');

 $checkBox= new checkbox($questionIndex."A","A");
 $checkA=$checkBox->show();
 $test.= $checkA." "."A"." ".$a."<br>";

 $checkBox= new checkbox($questionIndex."B","B");
 $checkB=$checkBox->show();
 $test.= $checkB." "."B"." ".$b."<br>";

 $checkBox= new checkbox($questionIndex."C","C");
 $checkC=$checkBox->show();
 $test.= $checkC." "."C"." ".$c."<br>";

 $checkBox= new checkbox($questionIndex."D","D");
 $checkD=$checkBox->show();
 $test.= $checkD." "."D"." ".$d."<br>";
 }


if($questionType ==3){   //  T/F question

 $radio= new radio($questionIndex."TrueFalse");
   $radio->addOption('1','True');
 $radio->addOption('2','False');
 $test.= $radio->show()."<br>";
     }

if($questionType ==1){    //quiz

 $textArea=new textarea($questionIndex."quiz");
     $textArea->setRows(8);
 $textArea->setColumns(50);
 $test.=$textArea->show()."<br>";
 }
```

```
    //for output
        return $test;
    }


    //make a question list in test edit page for the same test.
    function listQuestion($order=NULL)
    {
        $testId=$this->getParam('testId', NULL);

        if ($order) {
                $filter=" ORDER BY ".$order;
            }
        if($testId){
            $sql="SELECT
testId,questionId,questionIndex,questionType,questionContent,answer,a
ltA,altB,altC,altD,weight, questionNote FROM tbl_test_question Where
testId='$testId' order by questionIndex";
            }else{
        $sql="SELECT
testId,questionId,questionIndex,questionType,questionContent,answer,a
ltA,altB,altC,altD,weight, questionNote FROM tbl_test_question order by
questionIndex";
        }

        $filter=NULL;
            if ($order) {
                $filter=" ORDER BY ".$order;
            }
        $rs=$this->objDbQuestion->getArray($sql.$filter);
            return $this->disPlayQuestion($rs, "ktest", TRUE, "testId");
    }

    //to dispay question list in test edit page
    function displayQuestion($rs, $module, $allowAdmin = false, $key = null)
        {
            // Start with no keyValue for the search key
            $keyValue = null;
            // Initialize the output string to NULL
            $str = null;
```

```
  $this->questionTable->border=1;
  $this->questionTable->cellpadding=2;
  $this->questionTable->cellspacing=2;
  $this->questionTable->width="100%";

  $str = null;

      $rowcount = 0;


$head=array('TestId','Index','QuestionType','QuestionContent','Answer
','Weight','Delete');

$header=$this->questionTable->addHeader($head,"heading","align=center
");

  foreach ($rs as $line) {
   $oddOrEven = ($rowcount == 0) ? "odd" : "even";
   $questionId=$line['questionId'];
   $testId = $line['testId'];
   $questionIndex = $line['questionIndex'];
   $questionType = $line['questionType'];
   $questionContent = $line['questionContent'];
   $answer = $line['answer'];
   $altA = $line['altA'];
   $altB = $line['altB'];
   $altC = $line['altC'];
   $altD = $line['altD'];
   $weight = $line['weight'];
   $questionNote = $line['questionNote'];

   $paramArray = array('action' =>
'dequs','testId'=>$testId,'questionId'=>$questionId);

   $button=$this->href->showlink($this->uri($paramArray ,
"ktest"),"Delete",'');


 $row=array($testId,$questionIndex,$questionType,$questionContent,$an
swer,$weight,$button);
```

```
    $this->questionTable->addRow($row,$oddOrEven,"align=center");
            $rowcount = ($rowcount == 0) ? 1 : 0;

   }
    $table = $this->questionTable->show();
        //return $str;
   return $table;
   }



    //prepare for edit
  function getForEdit($mode)
  {
        $this->setvar('mode', $mode);
        $order = $this->getParam("order", NULL);
        // retrieve the group ID from the querystring
        $keyvalue=$this->getParam("testId", NULL);
        if (!$keyvalue) {
          die($this->objLanguage->languageText("modules_badkey").":
".$keyvalue);
        }
        // Get the data for edit
        $key="testId";
   $ar=$this->objDbTest->getRow($key, $keyvalue);
        $this->setvar('testId', stripslashes($ar['testId']));
        $this->setvar('testName', stripslashes($ar['testName']));
        $this->setvar('testOwner', stripslashes($ar['testOwner']));
        $this->setvar('testNote', $ar['testNote']);
        $this->setvar('startDate', $ar['startDate']);
        $this->setvar('endDate', $ar['endDate']);


   $addQAArray = array('action' => 'addquestion','testId' => $testId);
   $addQAButton = $this->objButtons->linkedButton("add",
$this->uri($addQAArray, "ktest"));
   $this->setvar('addQAButton', $addQAButton);
   $this->setvar('saveButton', $this->objButtons->putSaveButton());
  }
```

```
//make a test list in main page
function listTest($order=NULL)
{
 if ($order) {
      $filter=" ORDER BY ".$order;
      }
      $sql="SELECT
testId,testName,testOwner,testNote,startDate,endDate FROM
tbl_test_test";
      $filter=NULL;
      if ($order) {
         $filter=" ORDER BY ".$order;
      }
  $rs=$this->objDbTest->getArray($sql.$filter);
      return $this->displayData($rs, "ktest", TRUE, "testId");
 }


//to display test list in main page
function displayData($rs, $module, $allowAdmin = false, $key = null)
   {
      // Start with no keyValue for the search key
      $keyValue = null;
      // Initialize the output string to NULL
      $str = null;

    $this->table->cellpadding=2;
  $this->table->cellspacing=2;
  $this->table->width="100%";
   $this->table->border=2;

$head=array('TestId','TestName','lecturer','Note','startDate','EndDat
e','Manage');
     $this->table->addHeader($head,"heading","align=center");


      // Initialize the row to 0, the first row
      $rowcount = 0;
  foreach ($rs as $line) {
   $oddOrEven = ($rowcount == 0) ? "odd" : "even";
   $str .= "<tr>";
```

```
    $testId = $line['testId'];
    $testName = $line['testName'];
    $testOwner = $line['testOwner'];
    $testNote = $line['testNote'];
    $startDate = $line['startDate'];
    $endDate = $line['endDate'];


          if ($allowAdmin) {

              // icons in main page

       $openArray = array('action' => 'view',
                  'testId' => $testId);
     $editArray = array('action' => 'edit',
                  'testId' => $testId);
              $deleteArray = array('action' => 'delete',
                  'testId' => $testId);
     $resultArray = array('action' => 'result',
                  'testId' => $testId);

    $open=$this->href->showlink($this->uri($openArray,
'ktest"),"view",'');
    $edit=$this->href->showlink($this->uri($editArray,
'ktest"),"Edit",'');
    $delete=$this->href->showlink($this->uri($deleteArray,
ktest"),"Delete",'');
    $result=$this->href->showlink($this->uri($resultArray,
ktest"),"Result",'');
    $manange=$open.$edit.$delete.$result;
    $row=array($testId,
testName,$testOwner,$testNote,$startDate,$endDate,$manange);

    $this->table->addRow($row,$oddOrEven,"align=center");

  }
          // Set rowcount for bitwise determination of odd or even
          $rowcount = ($rowcount == 0) ? 1 : 0;
 }
```

```
        $table = $this->table->show();
       $str =$table;

    ///add a new test
    if ($allowAdmin) {
            $paramArray = array('action' => 'add');
            //$str .= " <class=\"heading\" align=\"right\"
width=\"30\">";
     $str .="<br>".$this->href->showlink($this->uri($paramArray ,
"ktest"),"Add A New Test",'');
    }
  return $str;
 }



 function prepForDelete()
  {
   $this->setvar('nobutton', $this->objButtons->button("confirm",
"delete_no.png"));
   $this->setvar('yesbutton', $this->objButtons->button("confirm",
"delete_yes.png"));
   $this->setvar('testId',$this->getParam("testId", NULL));
  }


 //for delete a test
 function deleteItem()
   {
  $testId=$this->getParam("testId", NULL);
  if (!$testId) {
      die($this->objLanguage->languageText("modules_badkey").":
.$keyvalue);
  } else {
       $this->objDbTest->deleteTest("testId", $testId);
      $this->objDbQuestion->deleteTest($testId);
  }
 }
 }
 >
```

```php
<?php
class dbTest_mark extends dbTable
{

    /**
    * Constructor method to define the table
    */
    function init() {
        parent::init('tbl_test_mark');

        $this->objUser =& $this->getObject('user', 'security');
        $this->objLanguage =& $this->getObject('language', 'language');
        $this->USE_PREPARED_STATEMENTS=True;

    }
 function mark(){

  $testId=$this->getParam('testId',NULL);
  $studId=$this->getParam('studId',NULL);
  $markId=$testId.$studId;
  $score=$this->getParam('result',NULL);
  $res=array(
    'markId'=>$markId,
    'testId'=>$testId,
    'studId'=>$studId,
    'score'=>$score,
    );
   if($this->valueExists('markId',$markId))
   {$this->update('markId',$markId,$res);}
   else{$this->insert($res);}


 }
```

```php
<?php
class dbTest_question extends dbTable
{

    /**
    * Constructor method to define the table
    */
    function init() {
        parent::init('tbl_test_question');

        $this->objUser =& $this->getObject('user', 'security');
        $this->objLanguage =& $this->getObject('language', 'language');
        $this->USE_PREPARED_STATEMENTS=True;
    }


 function deleteQuestion(){
  $questionId=$this->getParam('questionId',NULL);
  $this->delete('questionId',$questionId);
 }


    function deleteTest($testId){
  $sql="select questionId from tbl_test_question Where testId =
```

```
'$testId'";
  $rs =$this->getArray($sql);
  foreach($rs as $line){
   $questionId = $line['questionId'];
   $this->delete('questionId',$questionId);
  }
 }
    /**method to add questions*/
  function addQuestion() {

  $id = $this->getParam('testId',NULL);
  $question = $this->getParam('questionIndex',NULL);
  $questionId = $id.$question;
  $content = $this->getParam('quiz',NULL);

  /******get corret answer****/
  //define question type for quiz
  $answer = $this->getParam('quizAnswer',NULL);

  $type = 1;
  //question type is multiple choice
  if(!$answer){

   $ans = array(
        'a'=>$this->getParam('A',NULL),
        'b'=>$this->getParam('B',NULL),
        'c'=>$this->getParam('C',NULL),
        'd'=>$this->getParam('D',NULL)
        );

   if ($ans['a']){$answer.='A';}
   if ($ans['b']){$answer.='B';}
   if ($ans['c']){$answer.='C';}
   if ($ans['d']){$answer.='D';}

   $type =2;
  }
  //true & falsa type
  if(!$answer){
   $ans = $this->getParam('tureFalse');
   if($ans == 0 ){$answer='false';
```

```
    }else{$answer='true';}
    $type =3;
    }
   $a = $this->getParam('altA',NULL);
   $b = $this->getParam('altB',NULL);
   $c = $this->getParam('altC',NULL);
   $d = $this->getParam('altD',NULL);
   $weight = $this->getParam('weight',NULL);
   $note = $this->getParam('additionalNote',NULL);



  if ($this->valueExists("questionId", $questionId)) {
            $displaytext="<span class=\"error\">"
               .$this->objLanguage->languageText("error_valueexists
')
               ."</span>";
            $idclass="inputerror";
            die("Question exists and this error message needs to be
rewritten");
    }else{
    $this->insert(array(
       'questionId'=>$questionId,
     'testId'=>$id,
     'questionIndex'=>$question,
     'questionType'=>$type,
     'questionContent'=>$content,
     'answer'=>$answer,
     'altA'=>$a,
     'altB'=>$b,
     'altC'=>$c,
     'altD'=>$d,
     'weight'=>$weight,
     'questionNote'=>$note,
        )
     );
      }
  }
}
?>
```

```
<?php
class dbtest_result extends dbTable
{
    /** Constructor method to define the table */
    function init() {
        parent::init('tbl_test_stud');

        $this->objUser =& $this->getObject('user', 'security');
        $this->objLanguage =& $this->getObject('language', 'language');
        $this->USE_PREPARED_STATEMENTS=True;
   $this->objDbQuestion = & $this->getObject('dbTest_question');
    }
    function addResult() {

    $testId = $this->getParam('testId',NULL);
    $studId = $this->objUser->userId();
    $studName = $this->objUser->fullName();

    $filter="where testId='$testId'";
    $count=$this->objDbQuestion->getRecordCount($filter);

    $questionIndex=1;

    while($count--){

     $answer=NULL;

 if($this->getParam($questionIndex.'A',NULL)){$answer='A';}
     if($this->getParam($questionIndex.'B',NULL)){$answer.='B';}
     if($this->getParam($questionIndex.'C',NULL)){$answer.='C';}
     if($this->getParam($questionIndex.'D',NULL)){$answer.='D';}

 if($ans=$this->getParam($questionIndex.'quiz',NULL)){$answer=$ans;}


 if($this->getParam($questionIndex.'TrueFalse',NULL)=='1'){$answer='t
rue';}
     if(!$answer){$answer='false';}
     $this->insert(array(
          'testId'=>$testId,
          'studId'=>$studId,
```

```
            'studName'=>$studName,
            'questionIndex'=>$questionIndex,
            'studAnswer'=>$answer,
            'result'=>$result,
          ));

     $questionIndex++;

    }
  }
}
 ?>
<?php
/**
* Class for the test table in the database
*/
class dbTest_test extends dbTable
{

    /** Constructor method to define the table*/
    function init() {
        parent::init('tbl_test_test');

        $this->objUser =& $this->getObject('user', 'security');
        $this->objLanguage =& $this->getObject('language', 'language');
        $this->USE_PREPARED_STATEMENTS=True;
    }

   function addTest(){
    $id = $this->getParam('testId',NULL);
    $name = $this->getParam('testName',NULL);
    $owner = $this->objUser->fullname();
    $note = $this->getParam('testNote',NULL);
    $start = $this->getParam('startDate',NULL);;
    $end =$this->getParam('endDate',NULL);
  if ($this->valueExists("testId", $id)) {
              $displaytext="<span class=\"error\">"
                  .$this->objLanguage->languageText("error_valueexists
")
                  ."</span>";
              $idclass="inputerror";
```

```
            die("Test exists and this error message needs to be
rewritten");
    }else{
  $this->insert(array(
     'testId'=>$id,
                'testName'=>$name,
                'testOwner'=>$owner,
                'testNote'=>addslashes($note),
                'startDate'=>$start,
                'endDate'=>$end));
    }
 }

 function deleteTest($key, $keyvalue) {
      $this->delete($key, $keyvalue);
   }

   function saveTest() {
       $id = $this->getParam('testId',NULL);
    $name = $this->getParam('testName',NULL);
    $owner = $this->objUser->fullname();
    $note = $this->getParam('testNote',NULL);
    $start = $this->getParam('startDate',NULL);;
    $end =$this->getParam('endDate',NULL);

    $rsArray=array(
                'testName'=>$name,
                'testOwner'=>$owner,
                'testNote'=>addslashes($note),
                'startDate'=>$start,
                'endDate'=>$end);
          $this->update("testId", $id, $rsArray);


   }

} #end of class
?>
```

```
<?php
//******************** multiple choice question page**********//
  $this->loadClass("form","htmlelements");
  $this->loadClass("button","htmlelements");
  $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
  $this->loadClass("input","htmlelements");
  $this->loadClass("htmlheading","htmlelements");
  $this->loadClass("dropdown","htmlelements");
  $this->loadClass("radio","htmlelements");
  $this->loadClass("checkbox","htmlelements");


   $testId = $this->getParam("testId", null);
   $testOwner = $this->getParam("testOwner",null) ;
   $formInput = new form("formInput",
$this->uri(array('action'=>'quiz','testId'=>$testId,'testOwner'=>$tes
tOwner),'ktest'));

 //show test Id
    $textInput=new textinput("testId");
 $textInput->value = $testId;
 $textInput->size = 20;
 $id="<br>".$this->objH->str=("Show Test
ID")." "." ".$textInput->show();
  $formInput->addToForm($id);
 $formInput->addToForm('<br>');

 //add questio id
 $textInput=new textinput("questionIndex");
 $textInput->size = 20;
 $id="<br>".$this->objH->str=("Input Question Nember(Note:No Duplicate
Number)")." "." ".$textInput->show();
  $formInput->addToForm($id);
 $formInput->addToForm('<br>');

 //question
 $textArea=new textarea("quiz");
   $textArea->setRows(8);
 $textArea->setColumns(50);
 $res1=("Type The Question Below<br>").$textArea->show();
```

```
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

  /*****choose the correct the answer letter(A or B or C or D)multipe
answers********/
 $correctAnswer='<br>'.$this->objH->str=("Choose the correct answer
letter(one or multiple
answers)")." "." ";//$text_input->show();
  $formInput->addToForm($correctAnswer);

   $checkBox= new checkbox("A","A");
 $check=$this->objH->str=("A").$checkBox->show()." "." ";
 $formInput->addToForm($check);

 $checkBox= new checkbox("B","B");
 $check=" "." ".$this->objH->str=("B").$checkBox->show()."&nb
sp"." ";
 $formInput->addToForm($check);

 $checkBox= new checkbox("C","C");
 $check=" "." ".$this->objH->str=("C").$checkBox->show()."&nb
sp"." ";
 $formInput->addToForm($check);

 $checkBox= new checkbox("D","D");
 $check=" "." ".$this->objH->str=("D").$checkBox->show();
 $formInput->addToForm($check);

  //  add answerA
 $textArea=new textarea("altA");
   $textArea->setRows(6);
 $textArea->setColumns(50);
 $res1=("Type The Answer Below A<br>").$textArea->show();
 $str1="<br>"."<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

 //add answerB
 $textArea=new textarea("altB");
   $textArea->setRows(6);
```

```
 $textArea->setColumns(50);
 $res1=("Type The Answer Below B<br>").$textArea->show();
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

 //add answerC
 $textArea=new textarea("altC");
   $textArea->setRows(6);
 $textArea->setColumns(50);
 $res1=("Type The Answer Below C<br>").$textArea->show();
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

 //add answerD
 $textArea=new textarea("altD");
   $textArea->setRows(6);
 $textArea->setColumns(50);
 $res1=("Type The Answer Below D<br>").$textArea->show();
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

  // additional note
 $textArea=new textarea("additionalNote");
   $textArea->setRows(8);
 $textArea->setColumns(50);
 $res1=("Add an Additional Note(students can see it as explanation of
answer when they view test rusult)<br>").$textArea->show();
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

 //choose weight
 $textInput=new textinput("weight");
 $textInput->size = 20;
 $weight="<br>".$this->objH->str=("Type the weight of this
question")." ".$textInput->show();
  $formInput->addToForm($weight);
 $formInput->addToForm('<br>');
```

```
//button
$buttonSubmit=new button("submit", "Submit") ;
   $buttonSubmit->setToSubmit();
   $formInput->addToForm($buttonSubmit);
   $formInput->displayType=3;

    echo $formInput->show();

>
```

```
?php
 $this->loadClass("form","htmlelements");
 $this->loadClass("button","htmlelements");
 $this->loadClass("textarea","htmlelements");
 $this->loadClass("textinput","htmlelements");
 $this->loadClass("input","htmlelements");
 $this->loadClass("htmlheading","htmlelements");


 $testId = $this->getParam("testId", null);
 $testOwner = $this->getParam("testOwner",null) ;
 $formInput = new form("formInput",
this->uri(array('action'=>'quiz','testId'=>$testId,'testOwner'=>$tes
Owner),'ktest'));

//show test Id
   $textInput=new textinput("testId");
$testId = $this->getParam("testId", null);
$textInput->value = $testId;
$textInput->size = 20;
$id=$this->objH->str=("Show Test
D")." ".$textInput->show()."<br>";
   $formInput->addToForm($id);


//add questio id
$textInput=new textinput("questionIndex");
$textinput->size = 20;
$id="<br>".$this->objH->str=("Input Question Nember(Note:No Duplicate
umber")." ".$textInput->show()."<br>";
 $formInput->addToForm($id);


//add quiz question
   $textArea=new textarea("quiz");
   $textArea->setRows(8);
$textArea->setColumns(50);
$res1=("Add Quiz question<br>").$textArea->show()."<br>";
$str1="<br>".$this->objH->str = $res1;
$formInput->addToForm($str1);
```

```
//add quiz answer
$textArea=new textarea("quizAnswer");
  $textArea->setRows(8);
$textArea->setColumns(50);
$res1=("Add An Answer<br>").$textArea->show()."<br>";
$str1="<br>".$this->objH->str = $res1;
$formInput->addToForm($str1);


//additional notes for students to view test result(explain answer)
$textArea=new textarea("additionalNote");
  $textArea->setRows(8);
$textArea->setColumns(50);
$res1=("Add an Additional Note(students can see it as explanation of
answer when they view test rusult<br>").$textArea->show()."<br>";
$str1="<br>".$this->objH->str = $res1;
$formInput->addToForm($str1);


//choose weight
$textInput=new textinput("weight");
$textInput->size = 20;
$weight="<br>".$this->objH->str=("Type the weight of this
question")." ".$textInput->show()."<br>";
 $formInput->addToForm($weight);
$formInput->addToForm('<br>');


//button
$buttonSubmit=new button("submit", "Submit") ;
  $buttonSubmit->setToSubmit();
  $formInput->addToForm($buttonSubmit);
  $formInput->displayType=3;


 echo $formInput->show();

?>
```

```
<?php
//**************************add test on main page ***************//
   $this->loadClass("form","htmlelements");
  $this->loadClass("button","htmlelements");
  $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
  $this->loadClass("input","htmlelements");
  $this->loadClass("htmlheading","htmlelements");

    $formInput = new form("formInput",
$this->uri(array('action'=>'addtest'),'ktest'));

 //add test Id
    $textInput=new textinput("testId");
 $textInput->size = 20;
 $id=$this->objH->str=("Input Test
ID")." "." ".$textInput->show()."<br>";
  $formInput->addToForm($id);

 //show lecturer name
 $textInput=new textinput("userName");
 $textInput->value = $userName;
 $textInput->size = 20;
 $user="<br>".$this->objH->str=("User
Name")." "." ".$textInput->show()."<br>";
  $formInput->addToForm($user);


 //add test name
 $textInput=new textinput("testName");
 $textInput->size = 30;
 $name="<br>".$this->objH->str=("Enter Test
Name")." "." ".$textInput->show()."<br>";
  $formInput->addToForm($name);


 //note
   $textArea=new textarea("testNote");
   $textArea->setRows(2);
 $textArea->setColumns(50);
 $res1=("Add a Note<br>").$textArea->show()."<br>";
```

```
 $str1="<br>".$this->objH->str = $res1;
 $formInput->addToForm($str1);

 // test time
 $textInput=new textinput("startDate");
 $textInput->size = 30;
 $startDate="<br>".$this->objH->str=("Enter start time
").$textInput->show();

 $textInput=new textinput("endDate");
 $textInput->size = 30;
 $endDate=" "." "." "." ".$this->objH->str=("Enter
End time ").$textInput->show()."<br>";
  $formInput->addToForm($startDate.$endDate);
 $formInput->addToForm('<br>');

  //create test button
 $buttonSubmit=new button("addTest", "Create A New Test") ;
   $buttonSubmit->setToSubmit();
   $formInput->addToForm($buttonSubmit);
   $formInput->displayType=3;


 echo $formInput->show();

?>
```

```
<?php
//*******************true/false question page**********//
  $this->loadClass("form","htmlelements");
  $this->loadClass("button","htmlelements");
  $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
  $this->loadClass("input","htmlelements");

  $this->loadClass("dropdown","htmlelements");
  $this->loadClass("radio","htmlelements");



  $testId = $this->getParam("testId", null);
  $testOwner = $this->getParam("testOwner",null) ;
  $formInput = new form("formInput",
$this->uri(array('action'=>'quiz','testId'=>$testId,'testOwner'=>$tes
tOwner),'ktest'));

//show test Id
    $textInput=new textinput("testId");
 $testId = $this->getParam("testId", null);
 $textInput->value = $testId;
 $textInput->size = 20;
 $id="<br>".$this->objH->str=("Input Test ID  
").$textInput->show()."<br>";
  $formInput->addToForm($id);

//add questio id
 $textInput=new textinput("questionIndex");
 $textInput->size = 20;
 $id="<br>".$this->objH->str=("Input Question Nember(Note:No Duplicate
Number)")." "." ".$textInput->show()."<br>";
  $formInput->addToForm($id);

//add the question
    $textArea=new textarea("quiz");
    $textArea->setRows(8);
 $textArea->setColumns(50);
 $res1=("Type The Question Below<br>").$textArea->show()."<br>";
 $str1="<br>".$this->objH->str = $res1;
```

```
 $formInput->addToForm($str1);
    $formInput->addToForm('<br>');

//choose the correct answer(True or False)
//add "Is the answer to the above question true or false?"
 $correctAnswer=$this->objH->str=("Is the answer to the above question
true or false?   ");//.$text_input->show();
  $formInput->addToForm($correctAnswer);

 //add True\False dropdown
    $dropdown= new dropdown("tureFalse");
    $dropdown->addOption('1','True');
    $dropdown->addOption('0','False');
   $dropdown->show();
    $formInput->addToForm($dropdown);
    $formInput->addToForm("<br>"."<br>");

//additional notes for students to view test result(explain answer)
 $textArea=new textarea("additionalNote");
    $textArea->setRows(8);
 $textArea->setColumns(50);
 $res1=("Add an Additional Note(students can see it as explanation of
answer when they view test rusult<br>").$textArea->show()."<br>";
 $str1=$this->objH->str = $res1;
 $formInput->addToForm($str1);

 //choose the weight
    $textInput=new textinput("weight");
 $textinput->size = 20;
 $weight="<br>".$this->objH->str=("Type the weight of this
question").$textInput->show()."<br>";
  $formInput->addToForm($weight);

//button
 $buttonSubmit=new button("submit", "Submit") ;
    $buttonSubmit->setToSubmit();
    $formInput->addToForm($buttonSubmit);
    $formInput->displayType=3;

echo $formInput->show();
?>
```

```
Name")." "." ".$textInput->show();
  $formInput->addToForm($name);
 $formInput->addToForm('<br>');


 //note
   $textArea=new textarea("testNote");
 $textArea->value = $testNote;
   $textArea->setRows(2);
 $textArea->setColumns(50);
 $res1=("Add a Note<br>").$textArea->show();
 $str1='<br>'.$this->objH->str = $res1;
 $formInput->addToForm($str1);
   $formInput->addToForm('<br>');

   //test time
 $textInput=new textinput("startDate");
 $textInput->value = $startDate;
 $textInput->size = 20;
 $startDate='<br>'.$this->objH->str=("Enter start time(YYYYMMDD
hhmmss)").$textInput->show().' '.' '.' '.' '.' '.
' ';


 $textInput=new textinput("endDate");
 $textInput->value = $endDate;
 $textInput->size = 20;
 $endDate=$this->objH->str=("Enter End time(YYYYMMDD
hhmmss)").$textInput->show()."<br>"."<br>";
  $formInput->addToForm($startDate.$endDate);



 //button
   $buttonSubmit=new button("edittest", "save") ;
   $buttonSubmit->setToSubmit();
   $formInput->addToForm($buttonSubmit);
   $formInput->displayType=3;

 echo $formInput->show();
```

```
//to display questions

    $addM = array('action' => 'choice',
                  'testId' => $testId,'testOwner'=> $testOwner);
    $addT = array('action' => 'addt',
                  'testId' => $testId,'testOwner'=> $testOwner);
 $addQ = array('action' => 'addq',
                  'testId' => $testId,'testOwner'=> $testOwner);

 //link "Add a Multiple Choice" to mutiple choice page
    $str .=$this->href->showlink($this->uri($addM,
"ktest"),"<br><br>Add a Multiple Choice (questionType=2)",'');
 //link "Add a True/False" to add true/false page
    $str .=$this->href->showlink($this->uri($addT,
"ktest"),"<br><br>Add a True/False (questionType=3))",'');
 //link "Add a Quiz" to quiz question page
 $str.=$this->href->showlink($this->uri($addQ, "ktest"),"<br><br>Add a
Quiz (questionType=1)",'');

 echo $outQuestion; // echo question
 echo $str;  //echo add question button

 ?>
```

```
<?php

//Create the centered area for display
$this->center = $this->objConfig = & $this->getObject('layer',
'htmlelements');
$this->center->align="center";

// Add the heading to the content
$this->objH =& $this->getObject('htmlheading', 'htmlelements');
$this->objH->type=3;
//$this->objH->str=$objLanguage->languageText("mod_gwk_stories_name")
;
$this->center->addToStr($this->objH->show());


//Add the main content
$this->center->addToStr($outStr."<br />");

//Output the content to the page
echo $this->center->addToLayer();

?>
```

```
<?php
 $this->center =  & $this->getObject('layer', 'htmlelements');
 $this->center->align="left";
 $this->mark=& $this->getObject('textInput','htmlelements');
// Add the heading to the content
 $this->objH =& $this->getObject('htmlheading', 'htmlelements');
 $this->objH->type=3;

 $this->center->addToStr($this->objH->show());
 $this->table = & $this->getObject('htmltable','htmlelements');
//**********************preview test page**************//
  $this->loadClass("form","htmlelements");
  $this->loadClass("button","htmlelements");
  $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
 $this->loadClass("input","htmlelements");

  $studId=$this->getParam('studId',NULL);
   $formInput = new form("formInput",
$this->uri(array('action'=>'mark','testId'=>$testId,'studId'=>$studId
),'ktest'));

   //display test Id
   $textInput=new textinput("testId");
 $textInput->value = $testId;
 $textInput->size = 20;
 $id="<br>".$this->objH->str=(" This Is Test ID").$textInput->show();


  // dispaly test name
   $textInput=new textinput("testName");
 $textInput->value = $testName;
 $textInput->size = 20;
 $name=' '.' '.' '.' '.' '.' '.$this->objH->s
tr=(" Test Name").$textInput->show();


  //display lecture name
   $textInput=new textinput("userName");
 $textInput->value = $testOwner;
 $textInput->size = 20;
```

```
 $user=' '.' '.' '.' '.' '.' '.$this->objH->s
tr=(" Lecturer Name").$textInput->show();
  $formInput->addToForm($id.$name.$user);
    $formInput->addToForm('<br>');

  //diaplay test time
    $textInput=new textinput("startDate");
 $textInput->value = $startDate;
 $textInput->size = 30;
 $startDate='<br>'.$this->objH->str=("this test start time
").$textInput->show();

    $textInput=new textinput("endDate");
 $textInput->value = $endDate;
 $textInput->size = 30;
 $endDate=' '.' '.' '.' '.' '.' '.$this->objH
->str=("this test end time ").$textInput->show();
  $formInput->addToForm($startDate.$endDate);
 $formInput->addToForm('<br>');

  //dispaly student number
    $textInput=new textinput("studentId");
 $textInput->value = $this->getParam('studId',NULL);
 $textInput->size = 30;
 $sudnetId='<br>'.$this->objH->str=("student number
").$textInput->show();

  //display stuent name
    $textInput=new textinput("studentName");
 $textInput->value =  $this->getParam('studName',NULL);
 $textInput->size = 30;
 $studentName=' '.' '.' '.' '.' '.' '.$this->
)bjH->str=("studnet name ").$textInput->show();
  $formInput->addToForm($sudnetId.$studentName);
 $formInput->addToForm('<br>');

 //display test note
    $textArea=new textarea("note");
 $textArea->value=$testNote;
    $textArea->setRows(5);
 $textArea->setColumns(60);
```

```
$res1=("Test Note<br>").$textArea->show()."<br>";
$str1="<br>".$this->objH->str = $res1;
$formInput->addToForm($str1);

$this->center->id = "blog-content";

$this->center->addToStr($outStr);
$layer = "<br>".$this->center->addToLayer();

$formInput->addToform($layer);

$this->mark->textinput('result','Mark here');
$formInput->addToform($this->mark->show());

//create test  submit button
$buttonSubmit=new button("addTest", "Submit") ;
   $buttonSubmit->setToSubmit();
   $formInput->addToForm($buttonSubmit);
   $formInput->displayType=3;

$form=$formInput->show();

$row=array($outList,$form);
$this->table->cellpadding=2;
$this->table->cellspacing=2;
$this->table->width="100%";
$this->table->border=2;
$this->table->cell_attributes='valign=top';
$this->table->addRow($row,$oddOrEven,"align=center");

echo $this->table->show();
?>
```

```
<?php
$this->form=& $this->getObject("form","htmlelements");
$this->textInput=& $this->getObject("textinput","htmlelements");
$this->button=& $this->getObject("button","htmlelements");
$this->center = $this->objConfig = & $this->getObject('layer',
'htmlelements');

$this->center->align="center";

// Add the heading to the content
$this->objH =& $this->getObject('htmlheading', 'htmlelements');
$this->objH->type=3;

$this->center->addToStr($this->objH->show());
$this->form->form("formInput",
$this->uri(array('action'=>'view','topicId'=>$topicId),'ktest'));

$this->textInput->textinput("testId","input testId here");
$this->form->addToForm($this->textInput->show());

$this->button->button("submit", "Submit") ;
$this->button->setToSubmit();
$button=$this->button->show();
$this->form->addToForm($button);

$this->center->addToStr($this->form->show());
echo $this->center->addToLayer();

?>
```

```
<?php
 $this->center = $this->objConfig = & $this->getObject('layer',
'htmlelements');
 $this->center->align="left";

// Add the heading to the content
 $this->objH =& $this->getObject('htmlheading', 'htmlelements');
 $this->objH->type=3;

 $this->center->addToStr($this->objH->show());

//**********preview test page**********//
  $this->loadClass("form","htmlelements");
  $this->loadClass("button","htmlelements");
  $this->loadClass("textarea","htmlelements");
  $this->loadClass("textinput","htmlelements");
  $this->loadClass("input","htmlelements");

   $formInput = new form("formInput",
$this->uri(array('action'=>'dotest','testId'=>$testId,'studId'=>$stud
Id),'ktest'));

  //display test Id
   $textInput=new textinput("testId");
 $textInput->value = $testId;
 $textInput->size = 20;
 $id="<br>".$this->objH->str=(" This Is Test ID").$textInput->show();


  // dispaly test name
   $textInput=new textinput("testName");
 $textInput->value = $testName;
 $textInput->size = 20;
 $name=' '.' '.' '.' '.' '.' '.$this->objH->s
tr=(" Test Name").$textInput->show();


  //display lecture name
   $textInput=new textinput("userName");
 $textInput->value = $testOwner;
 $textInput->size = 20;
```

```
 $user=' '.' '.' '.' '.' '.' '.$this->objH->s
tr=(" Lecturer Name").$textInput->show();
  $formInput->addToForm($id.$name.$user);
    $formInput->addToForm('<br>');

  //diaplay test time
    $textInput=new textinput("startDate");
 $textInput->value = $startDate;
 $textInput->size = 30;
 $startDate='<br>'.$this->objH->str=("this test start time
").$textInput->show();

    $textInput=new textinput("endDate");
 $textInput->value = $endDate;
 $textInput->size = 30;
 $endDate=' '.' '.' '.' '.' '.' '.$this->objH
->str=("this test end time ").$textInput->show();
  $formInput->addToForm($startDate.$endDate);
 $formInput->addToForm('<br>');

  //dispaly student number
    $textInput=new textinput("studentId");
 $textInput->value = $this->objUser->userId();
 $textInput->size = 30;
 $sudnetId='<br>'.$this->objH->str=("student number
").$textInput->show();

  //display stuent name
    $textInput=new textinput("studentName");
 $textInput->value = $this->objUser->fullName();
 $textInput->size = 30;
 $studentName=' '.' '.' '.' '.' '.' '.$this->
objH->str=("studnet name ").$textInput->show();
  $formInput->addToForm($sudnetId.$studentName);
 $formInput->addToForm('<br>');

 //display test note
    $textArea=new textarea("note");
 $textArea->value=$testNote;
    $textArea->setRows(5);
 $textArea->setColumns(60);
```

```
$res1=("Test Note<br>").$textArea->show()."<br>";
$str1="<br>".$this->objH->str = $res1;
$formInput->addToForm($str1);



$this->center->id = "blog-content";

$this->center->addToStr($outStr);
$layer = "<br>".$this->center->addToLayer();

$formInput->addToform($layer);


//create test  submit button
$buttonSubmit=new button("addTest", "Submit") ;
  $buttonSubmit->setToSubmit();
  $formInput->addToForm($buttonSubmit);
  $formInput->displayType=3;

echo $formInput->show();

?>
```

```
<?
$sqldata[]="CREATE TABLE `tbl_test_mark` (
  `markId` varchar(150) NOT NULL default '',
  `testId` text NOT NULL,
  `studId` text,
  `score` text NOT NULL,
  PRIMARY KEY (`markId`)
) TYPE=INNODB";
?>


<?
$sqldata[]="CREATE TABLE `tbl_test_question` (
  `questionId` varchar(50) NOT NULL default '',
  `testId` text NOT NULL,
  `questionIndex` int(150) NOT NULL,
  `questionType` varchar(50) default NULL,
  `questionContent` text,
  `answer` text,
  `altA` text,
  `altB` text,
  `altC` text,
  `altD` text,
  `weight` varchar(50) default NULL,
  `questionNote` text,
  PRIMARY KEY (`questionId`)
) TYPE=INNODB";
?>


<?
$sqldata[]="CREATE TABLE `tbl_test_stud` (
  `testId` varchar(50) NOT NULL default '',
  `studId` varchar(50) NOT NULL default '',
  `studName` varchar(50) NOT NULL default '',
  `questionIndex` int(150) default NULL,
  `studAnswer` text,
  `result` int(15) default '0'
) TYPE=INNODB";
?>
```

```
<?
$sqldata[]="CREATE TABLE `tbl_test_test` (
  `testId` varchar(50) NOT NULL default '0',
  `testName` varchar(50) NOT NULL default '',
  `testOwner` varchar(50) NOT NULL default '',
  `testNote` text,
  `startDate` datetime NOT NULL default '0000-00-00 00:00:00',
  `endDate` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (`testId`)
) TYPE=INNODB";
?>
```