DESIGN AND IMPLEMENTATION OF A MAILING LIST MANAGEMENT SYSTEM

CASE STUDY: FACULTY OF COMPUTER SCIENCE.

BY

JACKLINE JOHN RICHARD

BBC/10013/81/DF

ACHOL JACOB GENG

BBC/10014/81/DF

A PROJECT REPORT SUBMITTED TO THE SCHOOL OF COMPUTER STUDIES IN PARTIAL FULLFILMENT OF THE REQUIREMENT FOR THE AWARD BACHELORS IN COMPUTER SCIENCE OF KAMPALA INTERNATIONAL UNIVERSITY

UGANDA

MAY 2011

Approval

This is to certify that this research proposal entitled "**mailing list management system**" has been conducted under my supervision and guidance and has been submitted to the school of computer studies with my approval.

Signature ……………………………

Mr. KASAWULI

Date 21st……………/May……………/2011………

Table of contents

CHAPTER ONE

INTRODUCTION

1.1. Background of the study:

This project describes the design approach and solution techniques developed for a mailing list manager at the faculty of computer studies of Kampala international university. The problem considered here is the design of a system to assist with the means of communication at the faculty. The design however, anticipates expanding the scope of the problem to accommodate other departments.

The solution to the mailing list problem is developed here using hypertext pre-processor (PHP), programming language which is a respected technology for web-based applications and allows developers to reach far beyond simple HTML application. We employ an object model which allow for the project to be written efficiently, using standard object-oriented methodologies. The internet has spawned a new need for database that can access via the web. The product developed here takes advantage of this technology.

1.1 Statement of the problem

There is need for a system that allows faculty and department members to communicate with each other via mailing lists in order to exchange information and for administrators to provide updates on all activities that may be taking place at the faculty to their lecturers in a cheap and efficient manner. The system currently used has a numbers of problems including the fact that the communicator is not guaranteed that the recipient will indeed receive the sent message.

1.2 Objectives of the Study

1.2.1 General Objective

The objective of this project is to design and describe the processes involved in the development of a mailing list manager with an outline of the fundamental concepts, theories and principles of the computer systems including software, hardware and communication.

1.2.2 Specific Objective of the study

i. Develop a system that allows members to access information about the university in a short period of time.

ii. To enforce a high level of security by ensuring that the service is accessed only by those with proper permissions and passwords.

iii. To provide an administrator with an interface that allows her to create mail, create lists, view and sent newsletters to lists.

iv. Setting up a database in MSQL and an interface to it using PHP and the increasingly popular cascade style sheets technology.

1.3 Scope of the study

The system has a database of lists, subscribers and archived newsletters uploading newsletters that have been created and sending mail.

This project is concerned with mailing list problems. A mailing problem has various constraints specific to the requirements of the department. In this project, we only described a model of the mailing list problem and apply a solution approach that has the flexibility of adding other specific changes in requirements.

There possibly exist a number of models that have been developed by other researchers to solve this problem, our aim however was not to beat their results but to demonstrate that our approach can operate across a wide range of mailing list instances. We present a graphical user interface that is easy to use and allows even the less experienced computer users masters in no time.

This project provide for a low- tech solution since the subscriber ship is accepted to be stable.

1.4 Significance of the study

This project will be beneficial to the system users in that they will be able to obtain more skills related to the use of computer technologies.

The system will be able to ease communication among members with the same interest at very low cost high speeds.

A number of researchers will be able to work on the same projects without the need for face to face interactions.

This project has provided us with a chance to put to practice some of the concepts from our courses into practice. Knowledge about system surfaces alone is insufficient in computer science; what is needed is an understanding of contents therefore this was a chance for us to show that we understand the ways and methods in which programs are represented and interpreted. Information can now be transmitted in a timely and efficient manner to all users concerned.

A mailing list manager that is easy to use, robust, expandable, workable has been developed to cater for the varying communication needs of the department

CHAPTER TWO

LITERITURE REVIEW

2.1 Introduction

This chapter involved looking at some literature related to the development of Mailing lists to ensure that no duplication of earlier work is carried on here. The process involved looking at document resources such as text books and online publications relating to our research problem.

We describe some of the most important characteristics of database systems, benefits and their impact on the development of mailing lists.

Organizations the world over have witnessed the benefits of fast access to information and we thought this would be a database system would be ideal in helping develop this system.

2.2 Mailing Lists

Mailing lists is a list of names and addresses that may be kept by an organization for sending publications to its members (Kabir, 2003). It consists of email addresses, subscribers receiving the mail at those addresses, the publications sent to those addresses and the reflector which is a single email address that when designated at the recipient of a message, will send a copy of that message to all the subscribers. Electronic mailing lists are usually fully or partially automated through the use of special mailing list software and a reflector address that are set up on server capable of receiving e-mail (welling 2001). Incoming messages sent to the reflector address are processed by the software and, depending their content, are acted upon internally or are distributed to all e-mail addresses subscribed to the mailing list. Depending on the software, additional addresses may be set up for the purposes of sending commands. Many electronic Mailing list servers have a special email address in which subscribers can send commands to the server to perform such tasks as subscribing and unsubscribing, temporarily halting the sending of messages to them, or changing available preferences. The common format for sending these commands is to send an email that contains simply the command followed by the name of the electronic mailing list the command pertains to. Some list servers allow users to subscribe, unsubscribe and change preferences via a website. Electronic mailing list servers can set to forward massages to subscribers of a particular mailing list either individually as they are received by the list server or digest form in which all messages received on a particular day by the list server are combined into one email that is sent once per day to subscribers. Some mailing lists allow individual subscribers to decide how they prefer to receive messages from the list server (individual or digest).

2.2.1 Types of Mailing lists

One type of electronic mailing list is an announcement list, which is used primarily as a one –way conduit of information and can only be "posted to" by selected people (maslaskowski, 2002).

Another type of electronic mailing list is a discussion list, in which any subscribe may post. On a discussion list, a subscriber uses the mailing list to send massages to all other subscribers, who may answer in similar fashion. Thus, actual usually topic oriented for example, politics, scientific discussions, joke contests, and the topics can range from extremely narrow to "whatever you think could interest us". In this they are similar to Usenet news groups, and share the same aversion to off-topic messages. The term discussion group encompasses both these types of list and newsgroups (www.evoknow.com).

On some discussion lists, every message must be approved by a moderator before being sent to the rest of the subscribers. Moderator approval is usually employed to keep a high average

Quality of posts and weed out spam. Some mailing lists are open to any one who wants to join them, while others require an approval from the list owner before one can join. Joining a mailing list is called "subscribing" and leaving a list is called "unsubscribing".

Mailing list services free web-based services offering an easy way to run and maintain such lists were popular in the late 1990s, but many of these were taken over, so that the only popular provider is now is Yahoo! Groups. This is used by a wide range of groups, including organizations that might at first glance to consider rivals to Yahoo! MSN Groups appear to be pushing hard to catch to Yahoo!

Innercircle.cc is designed to email a personal group of inner circle of friends, family members, soccer team, tennis buddies and other groups. Using a single email, it has an easy to use interface to create and manage multiple personal groups such as friends, family or group working on a similar project.

Free lists .org is a technology-related, web service using all free software, though it may be more difficult for some users to setup. The new version of Google groups includes free mailing lists services as well as access to Unsent.

2.2.2 Archives

A mailing list archive is a collection of past messages from one or more electronic mailing lists (Hernandez, 1999). Such archives often include searching and indexing functionality. Many archives are directly associated with the mailing list, but some organizations like Gmane collect archives from multiple mailing lists hosted a different organizations- thus, one message sent on one popular mailing list can end up in many different archives.

2.2.3 Simple PC-Based Mailing List Management

You can create simple mailing list in most e-mail clients. Microsoft Outlook Express calls it a group; Outlook tags it as a distribution list. A mailing list managed by your email client is easy to setup, doesn't cost anything, and can work effectively for small lists of up to a few dozen or so email addresses, however, sending email directly from your PC is not suitable for mailing lists with hundreds of recipients. Some ISPs limit the number of addresses in each message, or restrict the number of outgoing emailing messages to as little as 200 per day. in addition, the larger your mailing list, the more time it will take to send your message, which will keep you from using your email client. Inevitably, some mailing list messages will bounce back, clogging your inbox.

2.3.4 Mass Mailing Using a Server

A better approach to tackling mass emailing is to offload the processing from your PC to a server. You may be able to use your own email server, which is often the same as your Web server if you use a Web hosting service. Alternatively, you can outsource your mailings to a third-party service that specializes in email list management (smith, 2002). Using your own email server can give you more control and many costless than outsourcing to an e-mail list service. Mailing list mangers Majordomo and Mailman are two free and popular solutions.

Majordomo is a tried and true mailing solution that has been used for more then decade. Unfortunately, Majordomo can be confusing for a non-techie to set up. Furthermore, its aging command-line-based user interface isn't intuitive for a new mailing list administrators-though a Web-based interface add- on is available.

Mailman has a more polished look and a more accessible menu interface. It automatically processes bounced messages, with the ability to retain temporarily unavailable addresses while removing the truly dead addresses from your mailing list.

Both mailing list managers can work well in a shared hosting environment for lists of up to 500 or so addresses. Firing off a message to thousands of e-mail addresses at once can bring a shared hosting server to its knees. It is recommended that you check with your service provider to determine what size configuration of mailing list your server can handle. You may be asked to stagger messages for larger lists, configuring the software to send a batch of ten messages every five seconds or so. You may need to upgrade your account to one with more bandwidth (data transmission capacity) – and a bigger price tag, most likely –in order to send frequent mailings to a large list (Buther, 2000).

2.3 Database system

For most systems to work satisfactorily, a database most be designed to enable efficient storage and retrieval of data. Given this fact, it is there fore necessary to provide a detailed account of what databases are.

A database is a shared collection of logically related data organized to meet the needs of an organization (Weier, 2000). A database management system is software used to create, manage and organize and organization data. A DBMS works with the operating system to modify data and make data accessible in a variety of meaningful and authorized ways. When new data is created and old data deleted, fragmentation occurs. One of the main tasks during maintenance is to monitor the usage and growth of the data base, and keep fragmentation to a minimum. It is also important to tune the database after it has been implemented to attempt to improve the overall performance for the end users. Change that might occur includes addition or deletion of database tables, columns or indexes. Database maintenance shall be performed by the database administrator.

2.3.1 Advantages of Database

i. Transaction support is provided. Having a logical unit of work typically involving several database operations in particular, several updates operation. For example transferring money from account A to account B clearly to updates are required, one to withdraw cash from account A and the other to deposit it to account B. if the user has started that the two updates are part of the same transaction then by the system can effectively guarantee that either both of them or neither is even half way the process.

ii. Integrity maintenances. To ensure that the database is correct, is controlled centrally by permitting the data administrator define and implement integrity constraints known as business rules (Estes, 2001).

iii. Enforcing security. Data access is restricted to those with passwords and proper permissions (Estes, 2001).

iv. Balancing conflicting requirements. The database administrator with instructions from the data administrator can structure the system so as to provide an overall service that is best for the enterprise to avoid conflicting requirements issues (Estes, 2001).

v. Enforcement of standards. (Estes, 2001) states that the database administrator under the data administrator's instructions ensures that all applicable standards may include any or all of the following departmental installation, international standards. Standardizing data representation is particularly desirable as an in instrument to aid interchange or movement between systems. In each organization there is normally a department that is given the responsibility of data and information processing together with the task of ensuring smooth flow of information between organizations departments. Database has gained momentum at data management for any organization to accomplish its requirements. Today, the success of an organization depends on its ability to acquire accurate and timely data about its operation.

2.3.2 Database maintenance.

Database within an organization should be continually updated to include new business transactions. Other changes must also be made to insure accuracy of data in database. The tasks mentioned above are achieved through transaction processing programs and end user applications which closely collaborate with the DBMS.

The activities performed here include:

i. Deletion of out dated data.

ii. Expansion of database capacity, so that if accommodate larger data volumes for enhanced application requirements.

iii. Addition of new files to the database.

iv. Incorporation of fields onto existing records in the database.

v. Linking up all the items in the database logically.

2.4 Application architecture and Development Process.

The application architecture and development lifecycle are closely linked and have a great impact on the quality and maintainability of the resulting application (Hernandez, 1999). The application consists of several components that allow the user to manipulate records in the main database only using the components they require. This modular approach allows for a staged development of the most important and high-risk components, these separate components share functionality because they will manipulate the same database and hence having the same data structure. This shared functionality affords itself to the analysis classes to be shared among several components. The steps to approach would have a function in each component that would handle the database connection. If there is a change in the way this component functions, each application has to be modified and recompiled.

The advantage of an object-oriented approach is that change can easily be managed if the system is designed well. This results in easy maintenance and cost saving.

2.5 Database Management Systems.

According to(Ramez, 2001) , a DBMS is a complex software system that consists many components or modules for implementing the catalogue query language, interface processors, data access, concurrency control, recovery and security. A

DBMS is software that handles all requests to the database (Data, 2001), below is what typically happens.

i. A user issues an access request using some language.

ii. The database intercepts and analyses that request.

iii. The database inspects, in turn the external schema/conceptual mapping, conceptual schema, the internal mapping and the storage structure.

iv. The DBMS executes the necessary operations on the stored database.

2.5.1 Advantages of database Management systems.

i. Controlling redundancy. Since views of the different user groups are integrated during the database design, hence this does not permit inconsistency thus saving space therefore leading to control of files being repeated and in turn performance of queries improves (Ramez, 2001).

ii. Restricting unauthorized access. Data access is granted to only those with proper permissions and passwords. For example only the DBA's staff may be allowed to use certain privileged software, such as software for creating new accounts. Ramez, (2001).

iii. Representing complex relationships among data. ADB may include several data interrelated in many ways therefore representing it in a complex relationship as well as retrieve and update related data easily and efficiently (Ramez,2001).

iv. Enforcing integrity constraints. This is to ensure that the data entered into the system is valid. This is achieved through the rules which must remain true for the database to preserve integrity. These rules are specified at the time of creating the database (Ramez, 2001).

v. Backup and recovery. Despite the fact that the DB is protected from unauthorized access as well as invalid entries, there is always a danger of data in the DB being lost at any time due to hardware problems or system crashes. To guard data from this, RDBMS have in built backup and recovery techniques that the DB is protected from these kinds of fatalities (Elmasri, 2001).

2.5.2 Limitation of Database Management System.

i. Complexity. A DBMS is extremely complex software. Therefore, database designers and developers, DAs, DBAs and end users need to understand its functionality before taking advantage of it.

ii. The DBMS is a large piece of software occupying large MB or GB of disk space therefore requiring substantial memory space to run efficiently.

iii. The cost of the DBMS varies greatly depending on the functionality provided and the environment in which it works.

iv. Additional hardware costs. The disk storage requirements for the DBMS may necessitate the purchase of additional hardware, to achieve the desired performance.

v. The cost of the DBMS and extra hardware is insignificant compared to the cost of converting the existing application to run on the new hardware and the DBMS. This consists of staff training, employing specialists to help with the conversion and running of the system.

vi. Performance. Typically, a file based system is designed to serve a specific application, as a result, the performance is always good but for a DBMS that is designed to serve several applications, the applications may not run as fast as is desired.

vii. Higher impact of failure. The centralization of resources increases the vulnerability of the system. Since all users and applications depend on the availability of the DBMS, failure of any component may bring operations to a stand still.

The DBMS must be able to produce forms and reports.

Forms are the screen display of the DBMS that a user can fill out for a particular entity.

Reports are printed documents.

2.5.3 Characteristics of Database Management systems

i. DBMS contains a database itself which in turn consists of well defined constraints.

ii. Data independence. In file based systems, the structure of the file may require changing all the programs that access the file. By contrast, DBMS access programs do not require such changes. The structure of the data file is stored in the DBMS catalogue separately from the access programs.

iii. DBMS incorporate views of different users. A database usually has many different users who may require different views of the database. A view is a virtual table which may not necessarily exist in the DB be may be produced upon request by a particular user.

iv. Support for many users. A multi-user database allows multiple users to access the database at the same time. The DBMS include concurrent control to ensure that several users can access the same data at the same time.

2.6 Summary

A database system can be defined as an electronic collection of several records that are organized to serve many applications at the same time by organizing data so that it can appear to be in one location rather than storing data in separate files that can easily be misplaced or damaged.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This section describes the data collection techniques that have been employed to collect the data necessary for the completion of this project, which is the development of a mailing list manger. We also look at the system development methods. This section also describes the process of replacing existing systems with new systems.

The chapter also explains how the specification document is broken down into components and how they relate with each other to provide the overall functionality of the system.

3.2 The system Development Life Cycle (SDLC).

SDLC is a well defined process by which a system is conceived, developed and implemented. Development started by identifying the need to develop the new system.

Requirement analysis involved formally defining the functional needs of the system in terms of data, system performance, and security and maintainability requirements for the system are clearly understood by everyone involved in the project. This process requires extended collaboration between the developers and the client. At this point, functional and non functional requirements are clearly defined. The requirement specification document which is a natural language statement about what the system is intended to do is produced. All requirements were defined to a level of detail sufficient to start design of the system.

System design System design is the process of planning a new business system or one to replace or compliment an existing system. After producing the requirement specification document undergoes two consecutive design processes.

The first is the logical design in which the requirements are partitioned into components. These results into the architectural design document which describe

What each component does how and how each must interact with the other to provide the functionality required from the whole system. Each component is then designed in turn; this is referred to as detailed or physical design. This phase produces a detailed design document which describes how each component does what it is required to do and thus how the describes how each component does what it is required to do and thus how the whole system does what it is required to do.

System analysis is the process of gathering and interpreting facts, diagnosing problems and using that information to recommend improvements to the systems.

System implementation and testing: This phase was initiated after the system was tested and accepted by the user. This phase continues until the system is operating in accordance with the defined user requirements.

This is the physical realization of the system and the process of exhaustive elimination of errors that may exist in the system and also check that the system produces the desired results under known condition.

System validation The system is monitored for continued performance in accordance with the user requirements

The ezycom system has been tested and results have been obtained under known condition and careful investigation has carried out to confirm that the system satisfies the desired needs and function.

3.3 Data collection methods

In this section we have described the major data techniques we used in the collection of that data required to present right answer to the research question that is the designing of the mailing list manager.

Following the above information we thereof present the description of the techniques we employed and why we think they were convenient for our project.

3.3.1 Questionnaire

It's a technique in which a number of printed questions will be used for obtaining data. After filling up the questioners they are returned to us. Through questionnaires we were able to address the specific objective of our research problem. With this technique we were able to follow two board categories of the question that are closed ended and open ended in the designing our questions. Short question were used so that they are easy to understand and answer.

Advantages

a. the method was relatively cheap and expeditious

b. The information obtained was being expected to valid and reliable.

c. It ensured anonymity

d. It placed less pressure to respondent and gave then more time responds to questions.

Disadvantages

In spite of all the good of this technique this does not out the other side of unpleasantness therefore they are below;

I. It does not apply to those who are illiterate.

II. It caused us problem for the cases of incomplete entries.

III. We were uncertain whether all people would respond to our cell of answering the questionnaires'

IV. The information that was needed in the study was unable to be catered for.

V. It also presented confusion to the respondent as to the nature of information we exactly required.

VI. Questionnaires take considerable amount of time design and deliver to the respondents.

3.3.2 Interviews

Frey and oishi (1995:01) define an interview as purpose confiscation in which one person ask prefer questions (interviewer) and another answers them (respondent). This is done to gain information on a particular area to be researched. Interviews are useful tool which can lead to further research using other methodologies such as observations and experiments. Interviews can have one of two basic structures. Structured (closed interview style) or unstructured (open interview style).

Open-ended or unstructured interviews are defined by Nichols (1991:131)as" informal interview not structured by standard questions. Fields workers are free to deal with topics of interest in any order and phrase their question as they think best". This type of structure uses broad range of question asking them in any order according to how the interviewer develops. Open-ended question allow the interview, if they wish, to probe deeper into initial responses of the respondent to gain more a more detailed answer to the question. The richness of the data is therefore entirely dependant on the interviewer. They themselves, must judge how much or how little they should probe or say themselves.

- This type of structure is useful as a pilot study, to test out peoples responses would be to a particular issue. It may throw a completely different light on an issue that the interviewer that had previously never considered
- Freedom for the respondent to answer how they wish is important in giving them a feeling of control in the interview situation.

The major drawback of this approach

It takes a considerable amount of time to collect and analyse the responses. Due to the varied nature of the responses, it is necessary to use contained analysis technique to analyse them which take time.

Open question used in this unstructured interview approach can cause confusion either because of lack of understanding of the question by the format or lack of understanding of the respondent's answer by the interviewer.

Despite some of the limitation open-ended questions are very important

Closed or structured interview are defined by the Nichols (1991:131) as a social survey where "the rage of possible answer to each questions is known in advance. Often possible answer are listed on the form so that the interviewer simply makers the appropriate reply in each case". This approach is much more standardized using a prearranged list of answer for the respondents to choose from. There is little freedom for flexibility, due to fixed question order. Each person is given the same question therefore been uniform.

Advantages:

1) The information is easily quantifiable and the responses can be compared

2) It is convenient for such small organization like school of computer studies

3) They help in obtaining fast hand information since you actually talk to the right people who are meant to provide the information you need.

4) It gives adequate opportunity for the interview to forward varying opinions and suggestions about a particular topic.

5) It makes the interviewee feel at home that he/she is part of the changing system being suggested.

6) It is a quick way of collecting information since questions asked get to be answered there and then before proceeding to another.

Disadvantages

a) There is little freedom for flexibility due to the fixed question order. There is little room for unanticipated discoveries. Respondents may feel that their response does not fit any of the designed answer.

b) Respondents often lie due to feelings of inadequacy, luck of knowledge on the topic, nervousness or confusion

c) They will disrupt work at the department and this reason some people may refuse to be interviewed.

d) it wastes time because we have to interview each individual at a time of their choosing which isn't exactly what we desire and therefore require lots of patience.

e) They might not bring out the potential in the respondents.

Using this two structure, there two basic types of interviews used in every day research. The first of these is known as one to one interviews, personal interviews or intensive interviews. These types of interviews use a small sample averaging 30 people. The interview lasts up to several hours. It focuses on the use of open-ended questions allowing the respondent to answer freely. Questions that follow are basically are entirely based on how the respondents answer leads the interview. The questions therefore are standardized. It is a suitable way of dealing with sensitive or taboo topics yet has a whole requires a very good rapport to be establish between interviewers and respondents. This interview approach is flexible, providing large amount of details. It is clear that the answers are solely those of the person being questioned. The intensive situation that the interview is conducted in, May itself to be gained without directly asking for it. This approach however is very time consuming and costly. Interviewer bias is also quite a problem. The non standardization of the questions in these methods means it is difficult to generalize it on a larger scale. The second type of interview techniques is a group interview or focus group study. This approach as defined by Wimmer and Dominick (1997:97) is "A research strategy for understanding audience/consumer attitude and behaviour". The members of a focus group should feel very much at ease with each other before conducting the interviews; ideally they should perhaps know each other already. The members of the group should be of the same sex and share similar backgrounds in order to rule out any confounding variables (Nichols 1991:14). Conversation in a focus group can be either structured or unstructured and last up to two hours. Focus groups, like one to one interviews allow for the collection of preliminary information about a topic. They may be used in pilot studies to detect ideas that will be further investigating using another research method. Focus group interviewing is much cheaper and quicker to run than intensive one to one interviews and the responses tend to be more complete and less inhibited (Wimmer and Dominck 1997:97). The disadvantage here is that one person may consistently undermine the others, dominating the conversation. Some researchers claim that focus groups are not a good research methodology because of the potential influence of one or two respondents on the remaining members of the group. A dominant respondent can negatively affect the outcome of the group and that group pressures may influence comments made by individuals. Although this could occur, it may also be that other respondent's ideas spark new ideas with others, creating a snowball effect. The use of what is known as an extended focus group method, whereby each respondent fills out a questionnaire prior to the focus group discussion expressing their own personal views. This limits the problem of them being unwilling to express their opinion in front of the rest of the group.

General Advantages of Interview Methods

1 Interviews allowed questioning to be guide as we wanted and we could clarify points that needed to be made clearer much more easily than in something like a mailed questionnaire. The techniques however, relied on the willingness of the

respondent to give accurate and complete answers. Respondents often provided elaborate answers in an attempt to figure out the purpose of the study.

2 It was a good method because the group was small and not scattered geographically.

3 The responses were fast and it therefore saved us a lot of time.

4 The respondents were provided with the chance to express their opinions.

5 The respondents could easily obtain clarity on questions they did not understand.

3.4 Programming Languages

The language of choice we propose to use for our project is PHP. PHP is a server-side scripting language designed specifically for the web. The PHP code is interpreted by the web server and generates HTML or other output. PHP is open source and that makes it easy to obtain, use and redistribute. The current major version of PHP is 4.

PHP is a versatile scripting language in that it is available for Microsoft Windows, many versions of UNIX and has a fully functional web server. When PHP teams up a database such MYSQL, the result a dynamic application that allows us to produce customizable a applications that contain real-time information. PHP's major strength includes the following:

High performance: it is very efficient. Using an inexpensive server, you can perform millions of hits per day.

Acts as an interface to many different database systems: PHP has native connections available to many database systems. In addition to MYSQL, you can connect to postgreSQL, dbm, Oracle, fileProl, hyper wave, Informix and Sybase databases among others. Using the open database that provides an ODBC driver. Have built in libraries for many common web tasks.

Easy to learn and use: The syntax of PHP is based on other programming language, primarily C and Perl. So if you are familiar with other C-like languages such as C++ and java, production is almost immediate.

Low cost: PHP is free. The latest version can be downloaded free of change.

Portability: PHP is available in many operating systems. PHP code can be written on the free Unix- like operating systems such as Linux and FreeBSD, commercial UNIX versions such as solaris and IRIX or on different versions of windows.

Availability of source code. We have access to the PHP code. Unlike commercial closed – source products, if there is some think you want modified or added to the language, you are free to do this. PHP works together with a databases system, and we have chosen MSQL because of its following features:

Low costs: MYSQL is available free of charge under an open source license.

Ease of use: Most of the modern databases use SQL. If you have used another RDBMS, there should be no trouble adapting to this one.

Portability: MSQL can be used on many different UNIX systems as well as Windows.

Sources code: we can easily obtain and modify MSQL code easily.

The major components of the mailing list manager are as follows:

1 The web service:

This component will provide an interface through which:

a. System administrators may edit user and group privileges.

b. users may view their account information.

2 Authentication service:

a Establishes connection to the database on behalf of valid users.

Use case name	Use case description	Users administrator
Show-all-lists	Shows a list of available mailing lists.	Participating actors and roles.
Show archive	Displays archived news letters for a particular list	Users, administrator
Accounting settings	Displays users account settings.	Logged in user.
Show other lists	Displays mailing list to which the user is not subscribed.	Logged in user.
Show my lists	Displays information to which the user is subscribed	Logged in user.
Subscribe	Describes the event of a	Potential member.

	potential member requesting to join a list by subscribing.	
Unsubscribe	This describes the event of a member cancelling their membership.	Member.
Change password	Describes the event of a member that wants to change to a new password.	Member, administrator.
Create mail	This displays a form for uploading newsletters.	Administrator.
Create lists	Displays the form that allows for the creation of lists.	Administrator.
Store list.	Stores mailing lists details to the database.	Administrator.
Send	Sends newsletters to subscribers.	administrators
Login	Gives the user a login form	Member, administrator.
Logout	Ends a user's session	Member, administrator.
New account	Creates a new account for the user.	Member.
Store account	Stores account details for the user.	Member.

Database service.

This component will store account information for each user as well as their privileges as specified by the administrator. The components will interact as follows:

1 The web service provides web pages for users and administrators. The service web service also stores specification made by the administrators with the database service.

The accounting and authentication service identifies the privileges and specification for each user by querying the database service.

when a user initiates an action, the accounting and authentication service will establish a connection to the database service using the specification associated with that user.

The accounting and authentication service will log information about the user's activity using that connection.

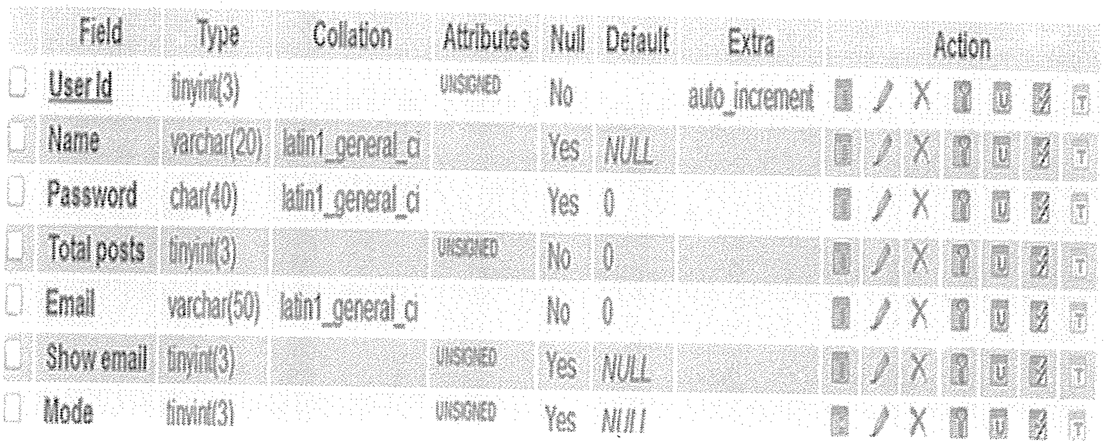
The table below shows use cases and actors for the proposed system.

3.5 Local Design

We determined the details required to create logical data models for each functional process, and for the project as a whole. This phase provided a clear, accurate description of data requirements that was used to design the database.

This describes the components of the new system and how these components work together to realize what is required to do.

user table



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	User Id	tinyint(3)		UNSIGNED	No		auto_increment	▮ / X ▮ U ▨ T
□	Name	varchar(20)	latin1_general_ci		Yes	NULL		▮ / X ▮ U ▨ T
□	Password	char(40)	latin1_general_ci		Yes	0		▮ / X ▮ U ▨ T
□	Total posts	tinyint(3)		UNSIGNED	No	0		▮ / X ▮ U ▨ T
□	Email	varchar(50)	latin1_general_ci		No	0		▮ / X ▮ U ▨ T
□	Show email	tinyint(3)		UNSIGNED	Yes	NULL		▮ / X ▮ U ▨ T
□	Mode	tinyint(3)		UNSIGNED	Yes	NULL		▮ / X ▮ U ▨ T

The screen shot above show the design of the user's table that will be used to store information about the users of our system. The design has seven (7) columns. The userID field is an auto incrementing field which uniquely identifies each user in our system.

The name field stores the name of a particular user

The password field stores the password that the user will use to log in to the system.

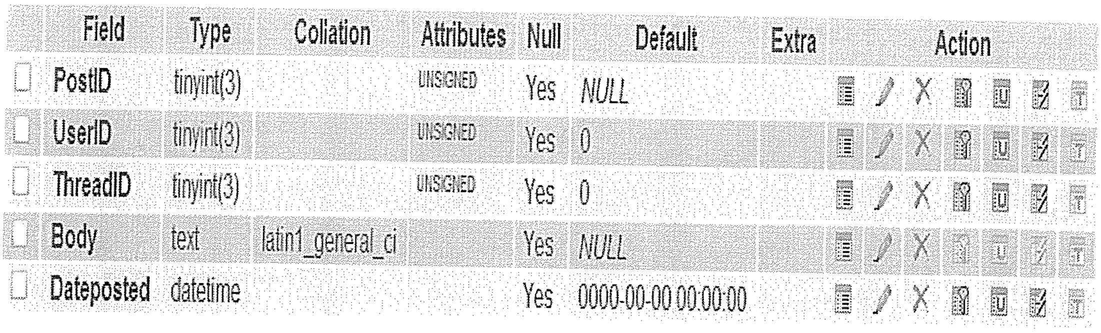
The total post fields keep track of the numbers of post a particular has made with this system.

The email field store the email address of that user.

The show mail field allows the system to determine whether that user's email address should be shown to other user or not.

The mode fields store an integer value to inform the system whether that user is the moderator or not.

Posts table

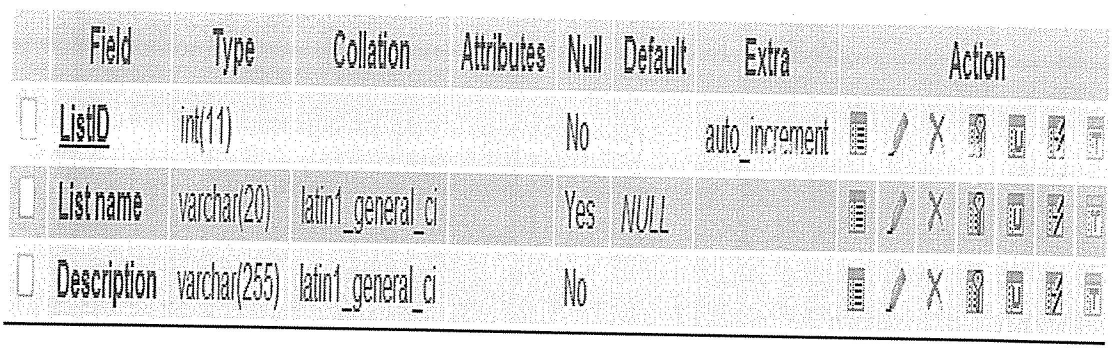


	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	PostID	tinyint(3)		UNSIGNED	Yes	*NULL*		
□	UserID	tinyint(3)		UNSIGNED	Yes	0		
□	ThreadID	tinyint(3)		UNSIGNED	Yes	0		
□	Body	text	latin1_general_ci		Yes	*NULL*		
□	Dateposted	datetime			Yes	0000-00-00 00:00:00		

The screen above shows an illustration of the table designed for the post that are made by normal users of the system. The table has five columns;

The postID field is an out-incrementing field that uniquely identifies each post made by a user. The UserID fields maps the post to the user who made the post. The threadID field is used to store the identifier for the message send by the user. The body field contains text that is the actual message sent by a user. The data posted field stores date and time information about when that post was made.

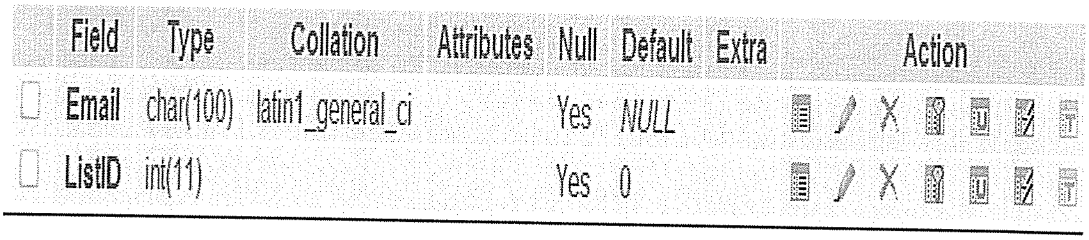
list table



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	ListID	int(11)			No		auto_increment	▤ ✎ X ▦ ▦ ▦ ▦
□	Listname	varchar(20)	latin1_general_ci		Yes	NULL		▤ ✎ X ▦ ▦ ▦ ▦
□	Description	varchar(255)	latin1_general_ci		No			▤ ✎ X ▦ ▦ ▦ ▦

The_diagram illiterates the list design table. This table stores information about the listID available. System users can subscribe to some or all of these lists in order to receive mail. This table consists of three columns. The list id is a field that uniquely identifies the lists within this table. List name is the name of the list. Description is a detailed description of what the list is about.

Sublists table



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	Email	char(100)	latin1_general_ci		Yes	NULL		▤ ✎ X ▦ ▦ ▦ ▦
□	ListID	int(11)			Yes	0		▤ ✎ X ▦ ▦ ▦ ▦

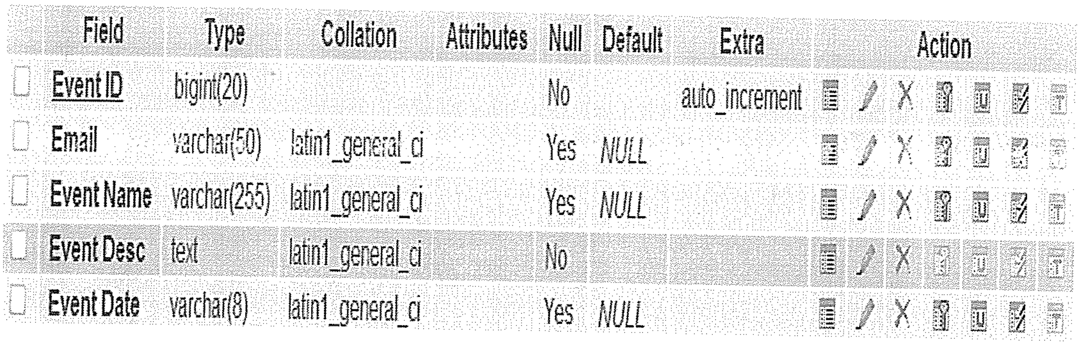
The diagram illustrates the design of the sub lists table; this table store information about the users who are subscribed to the list. The table has to columns; the email fields stores the email address of the person who he subscribed to a particular list. The listID fields contains UN integer that maps this table to lists table.

Threads table

The screen above shows the design of threads table. The table keeps information about the messages that have been posted system users. The table has two columns: thread ID, which is auto-incrementing field that uniquely identifies the message that have been made by the users.

Title: field which stores the title of each thread that is made by a system user

Events table

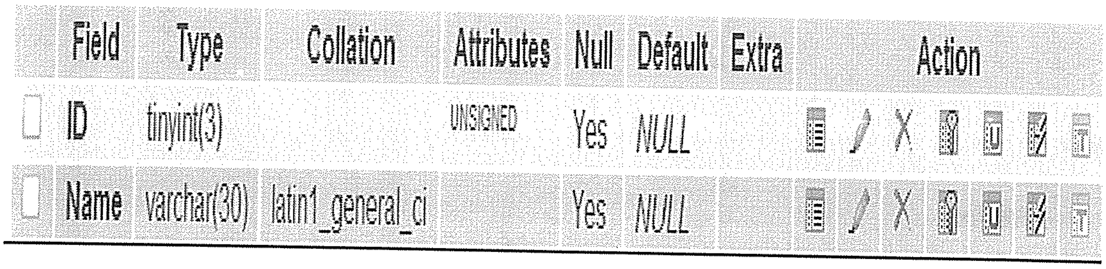


	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	Event ID	bigint(20)			No		auto_increment	▤ / X ▨ ▨ ▨ ▨
□	Email	varchar(50)	latin1_general_ci		Yes	*NULL*		▤ / X ▨ ▨ ▨ ▨
□	Event Name	varchar(255)	latin1_general_ci		Yes	*NULL*		▤ / X ▨ ▨ ▨ ▨
□	Event Desc	text	latin1_general_ci		No			▤ / X ▨ ▨ ▨ ▨
□	Event Date	varchar(8)	latin1_general_ci		Yes	*NULL*		▤ / X ▨ ▨ ▨ ▨

The screen above shows the illustration of the design of the events table which allows an administrator to set up personal reminders. The table consist of five fields

Event ID; this is an auto-incrementing field that uniquely identifies the events in this table. Email; this field stores the email address of the person that stored this event. Event name; this field stores the name of a particular event. Event Desc; this field stores the description of the event. Event date; this stores the date when the event is to happen.

Help table



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	ID	tinyint(3)		UNSIGNED	Yes	*NULL*		▤ / X ▨ ▨ ▨ ▨
□	Name	varchar(30)	latin1_general_ci		Yes	*NULL*		▤ / X ▨ ▨ ▨ ▨

The screen shows the design of the help table that contains information that provides more detailed of how to use the system.

The table contains two columns. The ID field is an auto incrementing field to uniquely identify each of the pages of the system.

3.5.1 Detailed Design

Detailed design considers the out requirements such as storage requirements, system control and backup.

3.5.2 Output Requirements

The product has been produced as both a soft and a hard copy. These are parts that translate information executed by the computer into a form that people comprehend.

3.5.3 Hard copy

It's a printout of the information stored in a computer. It is called hard because it's physically on paper. The hard copy will be printed.

3.5.4 Soft Copy

This is the computer version of the project. It provides a graphical user interface via which users interact with the system.

Its advantages include

1. It is easy to use.
2. Very easy to learn.
3. It provides an easy interface for fast use.
4. It provides for multitasking.

Data shall be input and administered centrally by the system administrator. The keyboard and mouse will be used as the input data into the system.

3.6 Processing Requirements

The system will require the following to perform efficiently and operate optimally.

3.7 Operating System

A Microsoft windows operating system with a PHP server such as apache installed. Microsoft windows XP, windows 2000 are the examples of the operating systems that can be used for this system. This is because windows new technology supports New Technology File system (NTFS) which offers adequate security among other features. It offers not only user level security, but also data and file level security. Stability, ease of use, easy to learn and an easy Graphical interface make it competitive in comparison to other operating systems.

3.8 Database Management System

The system has been designed with MSQL database server; however the database can be exported for use with other popular DBMS such as Oracle, Informix, and Sybase. The DBMS must be installed for the system to run.

3.9 Anti Virus Software.

There is need to have an anti virus software program to detect and disinfect viruses that may affect the database records.

3.10 Documentation

After the software specification satisfies the client's need, it's documented. The boundaries of documentation are:

A guide for people who will maintain the code in future in case the product needs to be updated. A soft copy of the source code which is easy to understand is made available. This documentation can also be used by people that want to learn how to use or have an idea of how software is developed. The documentation also acts as a guide that provides detailed explanations of what the software is about and how to use it.

3.11 Summary.

We have explored the methodologies which were employed to collect data do during this development phase in the collection of data. We have discussed the merits and demerits of the programming language that was used to develop the system as well as the use cases and actors for the developed system.

System design calls for a fore sight on the truth of system performance. It is at this level that the physical and conceptual aspects of system design are discussed. This section has specified the output design, input design rules and guidelines as well as the database including information on software and hardware requirements.

CHAPTER FOUR

SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

Having clearly understood the problem, collected and analyzed data and hence identified the system requirements, the next important phase is all the key issues identified previously are carefully considered. Detailed system analysis is then carried out on these issues to examine the system in terms of inputs, outputs, and the processes involved.

Since most of the errors emanate from this area, a well planned system design should be carried out to eliminate future system failures. This chapter therefore entails the various steps involved in system design during the system design and the system development.

4.2 Document Review

This was done through reviewing the institution's current records pertaining results management and getting related literature to broaden the researchers' knowledge and experience in order to assign system requirements and design to solve the record maintenance related problems like students daily inquiries.

The current system at the institution is a manual system. All the transactions performed at the institution are noted down on files which are then archived together in a safe place.

4.2.1 Strength of the current system

- It cannot be affected by power failures.
- All transactions performed can be retrieved relatively easy.
- It is very easy to implement.

4.2.2 Weakness of the current system

- A lot of paper work is needed hence higher maintenance cost.
- The manual system is relatively slow and cannot satisfy the growing number of population in the institution.
- Inconsistency of data as it is moved from one location to another.

- Possibility of data loss if the original files are misplaced since there are no backup copies.

4.3 Analysis of the proposed system

Due to the numerous losses of resources and the time wasted while serving students there was need to design and develop a system which could curb this weakness.

The proposed system will run a database management system (DBMS) accessible from a well designed graphical user interface. It will furnish a genetic solution to the problems of the persistent data storage, data integrity, security, and the overall speed of operations. It will provide backup facility to facilitate recovery.

4.3.1 Benefits of the proposed system

- It will be relatively cheap to maintain since no other hardware or equipment needs to be bought after it has been implemented. Although the initial price of implementation may be high because of buying new hardware, the overall price of maintenance is minimal.
- It operates at a much faster rate than the manual system since all the data is stored under one location and accessing it using the computer is fast.
- Since all the data is stored under one database, data inconsistency is eliminated when moving the information because you will only have to move the database
- The new system will support backup of its data. This will increase the systems reliability by enabling data recovery in case of a disaster.

4.3.2 Requirement analysis

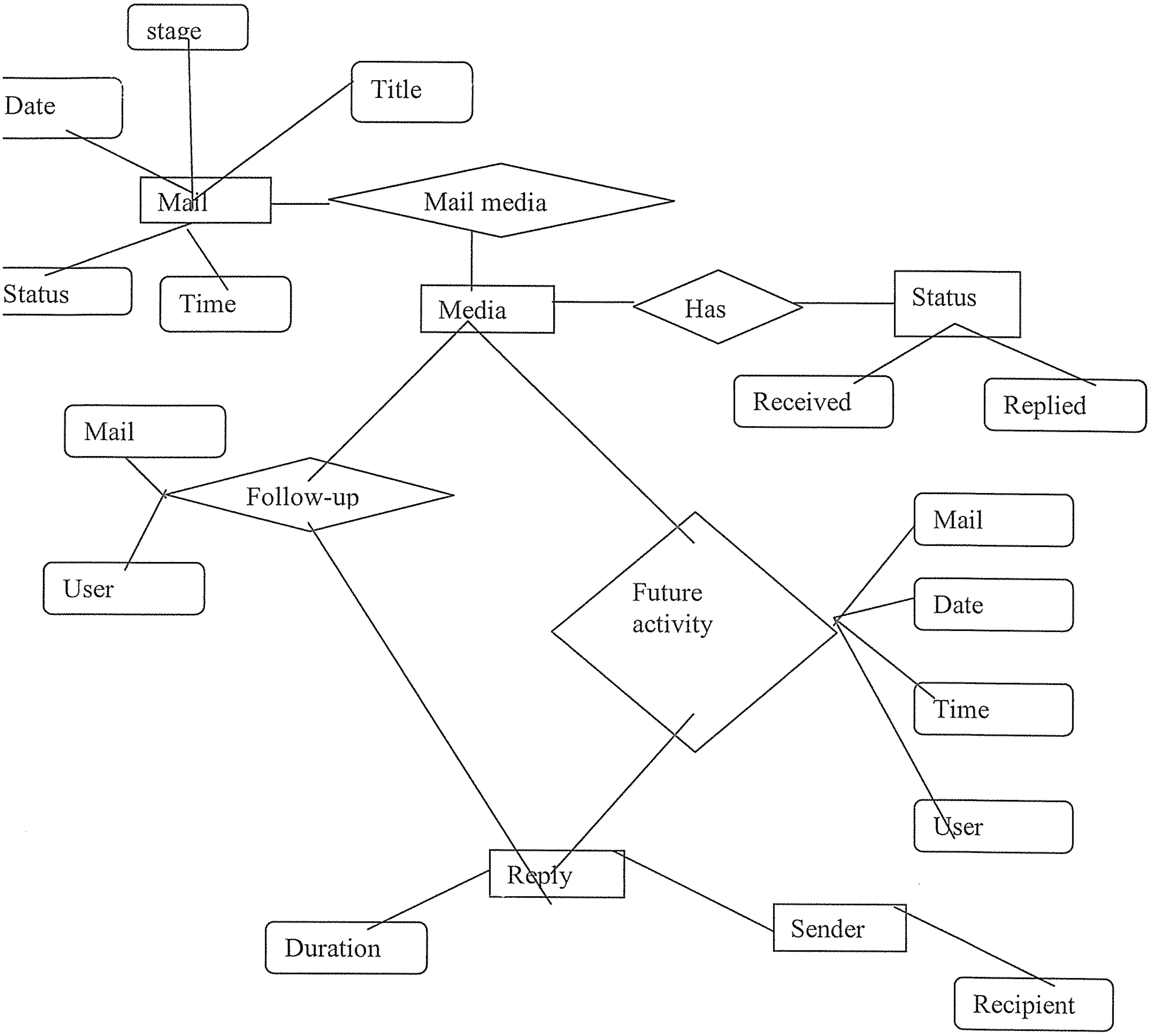
A record database needs to store information pertaining to its users, students and the media it contains e.g. inquiries, users, students and the institution itself. Records will be identified by their title, and the users by their login id or staff ID. In order to allow multiple copies of the same mail, each media item will have a unique ID number so that they can be stored under the same address.

The system users will be required to provide their personal information including their name, sex, phone number, e-mail address when signing up for a mailing list system.

4.4 System design

4.4.1 E-R model

It is clear that the physical objects from the previous section- the users and media details correspond to entities in the entity-relationship model, and the operations to be done on those entities-holds, follow-ups and so on- correspond to relationships. However, a good design will minimize redundancy and attempt to store all the required information in as small space as possible. To achieve this, a detailed ER diagram was designed as shown below.



E.R Diagram

NB/ Notice that the information about records has been separated from the media equity. This allows the database to store multiple copies of the same item without redundancy. The status entity has also been separated from media in order to save space. The hold relationship stores the entry's place in line (denoted by "queue"); it can be used to create a waiting list of different mails. The user must have a mail associated with it. The system users will therefore be responsible for performing all the administrative functions, such as checking media in and out and updating the mail details.

4.4.2 Flow charts

This is a diagram that represents the sequence of operations in a process. There are different ways of specifying processes in a system including.

- Structural English
- Flow charts

The researcher however, preferred the use of flow charts over the other methods. This is mainly due to the simplicity of flow charts.

This system consists of several individual processes that when integrated together, they form a coherent system. Below are some of the processes that are found in this project.

4.5 Database design

This system is expected to provide a suitable communication link between the user and the database. To ensure for an efficient system, it is important to design a database that is free from data redundancy and errors.

Since the system is to be used in a school institution, it must be able to store the media available in the school and also information pertaining to the department users. This makes it necessary to create separate tables to accommodate this information.

4.5.1 4.1.0 Relational database design

After coming up with an entity-relationship model to describe the system, we took advantage of the special relationships found in the design, and condensed the information to tables. This new design combines some entities and relationships into common tables to form a compact and efficient database representing the whole system.

Database schemas

Physical database design

4.6 GUI Design

The first step in designing the GUI was to choose a means of accessing the databases. In addition to simplifying operations on the database, it also makes extending the functionality of our system easier. When adding a new feature or improving and existing one, we will not need to change the entire database; it will only be necessary to modify the user interface portion of the code associated with it.

CHAPTER V

IMPLEMENTATION TESTING AND EVALUATION

5.1 Introduction

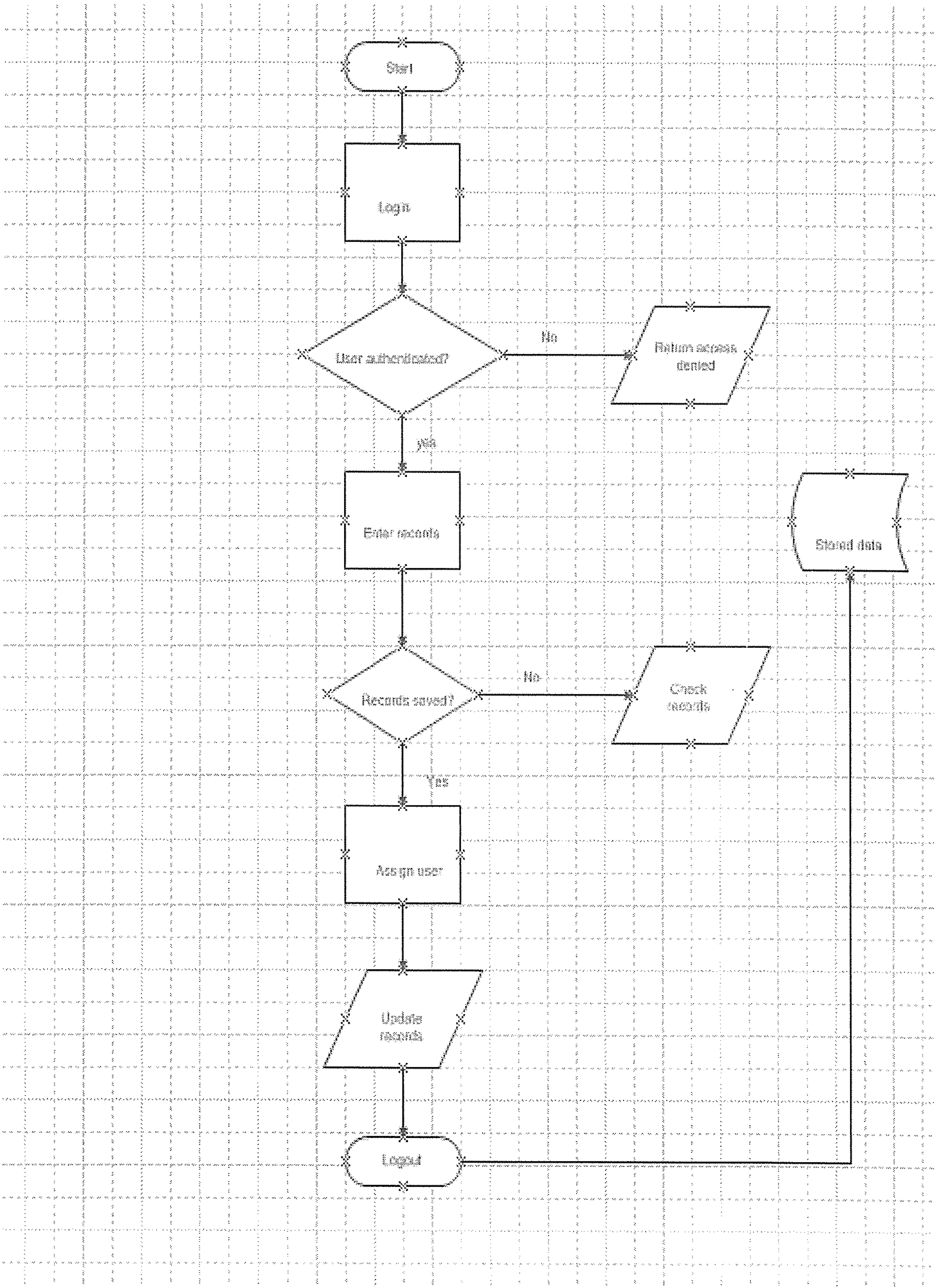
Once the design was complete, the development entailed two broad phase: back-end and user interface. The back-end comprised database application whereas the user interface comprised the application that the users would use to interact with the database. Mysql server was used while dream weaver 8 was used to develop the user interface.

This chapter generally describes the step involved in the implementation process. That is, development and implementation of a suitable user interface, program testing, then finally, a description of the system change over.

5.1.1 Graphical user interface

For a user to use the system, he must first register. First he/she has to provide all the necessary information that is obligatory. If a user provides the correct information at registration time, then the system allows him/her to create an account otherwise, an error is generated and the user can not be registered as a new user.

5.2 Login system



5.2.1 Mails

All data in the mailing list Management Software is organized in the so called mails. The mail is a comparatively independent mails introduced to the department centre and managed by its experts. The mail can include deferent activities, but they are always addressed towards one aim – solving the student's problems.

The mail is the main logical unit of the program. It could contain the following details:

- Description related to its origin
- Current status
- persons (physical and juridical persons) associated to it
- planned or done activities related to the mail (e.g. inquiries and questions)
- Reports
- Attached documents etc.
- Date
- Time

The mail is a process which undergoes development. All activities associated with a certain mail records like status changes, additional events etc are recorded in the "Follow-ups" section. The history of the mail records could be tracked from this section from the moment of its registration in the system on.

5.2.2 New mail record

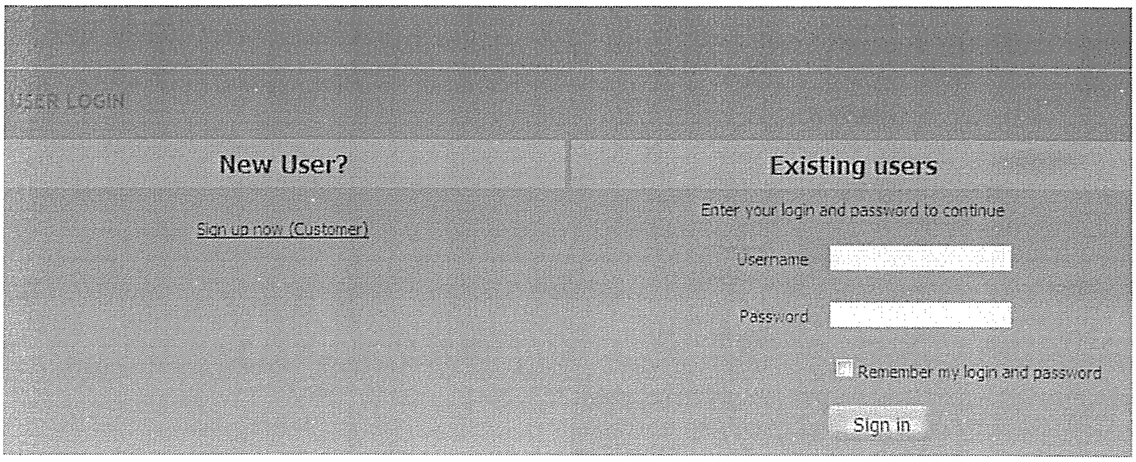
After entering the system through a username and password, you will get on the page called: "students support center". Here the user can now start entering a new record concerning a his or her question or inquiry.

The interface includes the following features:

- ✓ A unique personal mail
- ✓ Creation date
- ✓ Title of the mail

- ✓ Status of the mail
- ✓ The type of the question and the environment

The interface below gives the user ability to register a new record and saves it in the master file.



USER LOGIN

New User?

Sign up now (Customer)

Existing users

Enter your login and password to continue

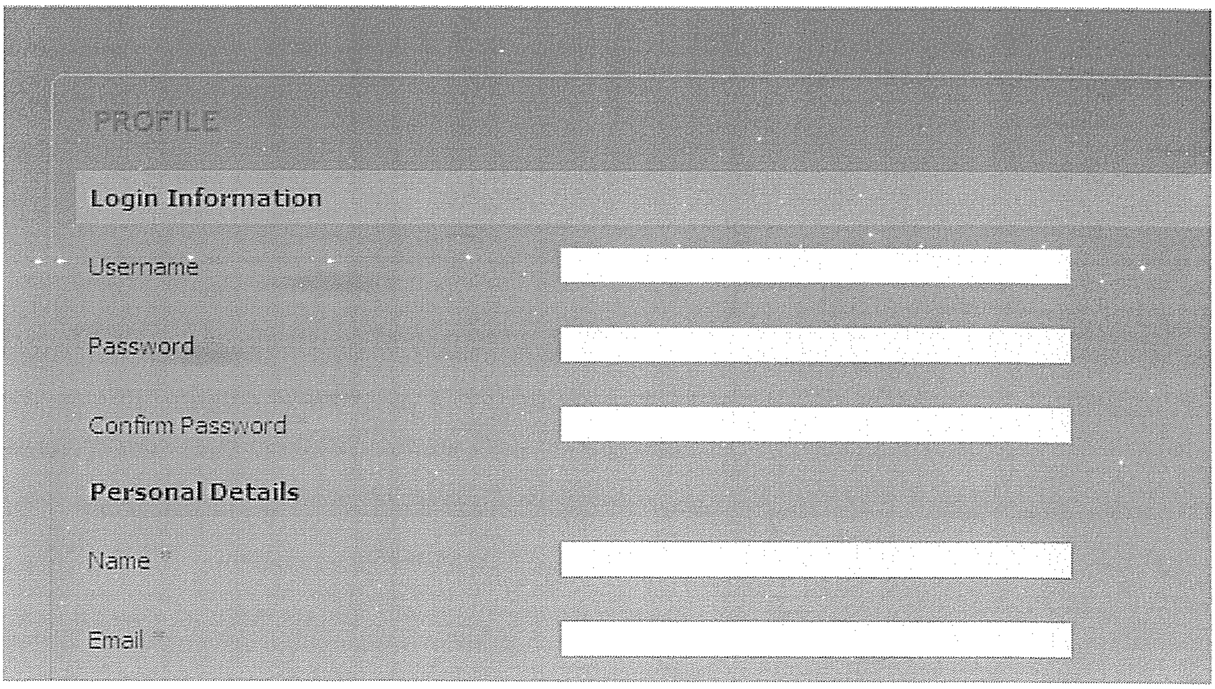
Username

Password

☐ Remember my login and password

Sign in

Here the user can sign up for his /her account in order to login tot the system.



PROFILE

Login Information

Username

Password

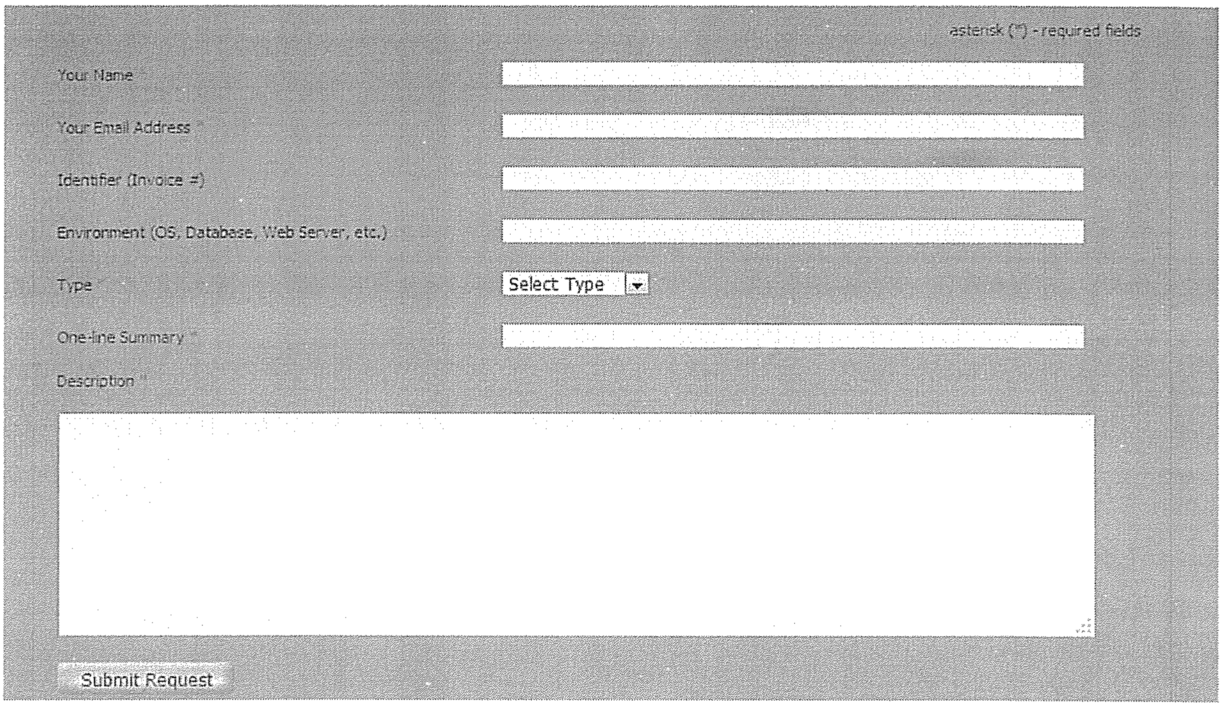
Confirm Password

Personal Details

Name *

Email *

After logging into the system the students can send their queries to their specific lecturers. This interface gives also an option of selecting or highlighting specific area on which you are enquiring from.



CHAPTER VI

DISCUSSION, RECOMMENDATION AND CONCLUSION.

6.1 Objectives and Aims

This project has turned out to be challenging in many ways. Each stage has presented its own problems to be overcome. When collecting information from industrial sources, as covered by the first objective, we expected a lack of response. Previous experience had taught me that only a small percentage of sources are likely to respond at all. We tried to compensate this by applying too many varied sources, to improve the chance of receiving varied answers, which could typify a cross-section of the educational community. What we were not prepared for, however, was the lack of ideas on how to improve the quality on the activities carried out in an institution. The general consensus of opinion of those who replied seems to be that a mailing list system is not required for the most part, as the system involves so many formalities and follow-ups which could be necessary. In hindsight, it may have been better to also apply to some personnel not directly involved with the mailing list system. Who would have less knowledge of the technicalities while still having an interest in their use? This could have highlighted more of a need for a mailing list system. The responses we have received, however, have introduced some other problems we had not envisaged, such as system crash supplied with new application. When running more than a few of these simultaneously, difficulties can arise when there is a need to move data between different software systems. Although OLE (object linking and embedding) has been designed with this in mind, it is not always possible to transfer data if the structure is foreign to that recognized by other systems.

6.1.1 Management

The Gantt chart in Appendix A details the original project plan. It lists the original topics this project had been classified under, and the expected date to start and approximate completion date. When choosing this time-scale, I had to take other considerations, such as course-work, into account. I felt it was important to have a clear plan so I would always know what could be worked on at any given time. On a study of this size it is crucial to know at the start exactly what needs to be done so the work can be paced and pre-researched accordingly.

The topics have been separated as much as possible to allow independent work units to be carried out simultaneously with little need for information from other units. The exception to this rule was the suggestions I have made as to how I believe court quality can be improved.

This required that I had information from institutional sources beforehand, as this was the main basis of the work. Furthermore, it was also essential that our research into the institution system details was carried out alongside it, so the technical knowledge was also available.

Due to the fact that we have placed ourselves under this working guideline, our project management has been quite successful. One area I should have given more time to the collection of information from industrial sources. The replies I received were not prompt. The first was returned within one month, but then I had to wait until the end of the autumn term before the others arrived.

Our plan has allowed three weeks at the end of the project with no tasks remaining. This does not mean that we expected to finish three weeks early, but that we expected some of the work to fall behind schedule. This period is the buffer by which we have been able to finish off tasks in time for the deadline. Fortunately, we have been able to make good use of this buffer to make up the time we lost with the slow responses detailed above and the difficulties encountered with the software. Further to the chart plan, we have also been attending regular meetings with our project supervisor. This has allowed us to sub-divide the work units again so each meeting had a target associated with it, ensuring the work-rate remained consistent. Whenever problems were met, we could then discuss them and come to a decision on the best method for solving it.

In our opinion, this has worked well, and has helped us maintain the flow without getting entangled in any one problem for too long.

6.2 Further Work

Certain elements in this project leave scope for further development. With almost any project which includes a software component, a list of future enhancements could be endless. In this case, we will only highlight the general areas where extra work would benefit the project. All these tests were carried out using sample data.

APPENDIX B

ctivity	Week 1					Week 2					Week 3					Week 4					Week 5		
	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W	T	F	M	T	W
roject ystem nception																							
roject ackground																							
iterature eview																							
easibility tudy																							
ystem nalysis																							
roblem lentification																							
equirements ecification	▨	▨																					
ystem Design d Coding																							
repare ogical and hysical esign	▨	▨	▨	▨	▨	▨	▨																
ustomize the chitectural sign						▨	▨	▨	▨	▨	▨	▨	▨	▨	▨								
reate the gal case stem omain											▨	▨	▨	▨	▨								
velop User														▨	▨	▨	▨	▨	▨	▨			

terfaces														▨	▨	▨	▨	▨	▨	▨			
ode the ogram UI's																		▨	▨	▨	▨	▨	▨
evelop the plication odules																							
esting and onversion																							
nplement the ogram odules																							
est each odule parately																							
tegrate the odules																							
st the stem ckage as a hole																							
oduce cumentation																							
oduction d aintenance																							
rrect any oblems																							
aluate and nitor																							