

INFORMATION WEBSITE
A CASE STUDY OF KAMPALA INTERNATIONAL UNIVERSITY WEBSITE
AND
ONLINE REGISTRATION

BY
NGONZI MONIC KASYATA

PDC/42307/91/DU

PGDCS (KIU)

A PROJECT REPORT SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF POST GRADUATE DIPLOMA IN COMPUTER SCIENCE
OF KAMPALA INTERNATIONAL
UNIVERSITY

AUGUST 2010

DECLARATION:

I declare that this research report is from my own work and has never been produced by anyone else for the same award in this or any other university or institution.

Name: NGONZI MONIC KASYABA

Signature: [Signature]

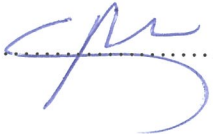
Date: 22/10/2020

APPROVAL

This is my research report and is submitted to the School of Postgraduate Studies with the approval of my supervisor.

Names Professor Vicente Gonzalez

Date. 22/10/2010

Signature.....

DEDICATION:

This research report is dedicated to my beloved husband Mr. Nyiiro Saul Kabi Robert and Son Isabirye Jordan.

ACKNOWLEDGEMENT.

I must first acknowledge the effort of my supervisor Professor Vicente Gonzalez For the excellent work in reviewing and guiding me while writing the manuscripts. Credit goes to him for the steady encouragement and keen insight throughout the writing process which have been of enormous value. I also extend my sincere gratitude to my family members for their staunch support emotionally, spiritually and above all for believing in me that I could write this project. Above all things Gods guidance as a driver in my life while writing this research and always cannot be forgotten for he makes us heads and not tails always and indeed we are winners.

ABBREVIATIONS

CSS - Cascading Style Sheets

DBMS - Database Management System

SDLC - System Development Life Cycle

UML – Unified Modeling Language

B2B - Business-to-business

LAN - Local Area Network

WAN- Wide area network

HTTP- Hypertext Transfer Protocol

HTML- Hypertext Markup Language

Table of contents

ACKNOWLEDGEMENT	5
CHAPTER ONE	10
Introduction	10
1.0 Background	10
1.2 Problem statement.....	12
1.3 Objectives	12
1.3.0 General objective	12
1.3.1 Specific objectives.....	13
1.4 Research questions	13
1.5 Justification.....	14
CHAPTER TWO.....	16
2.0 LITERATURE REVIEW	16
2.1 The internet	17
2.2 Browsing the Web.....	18
2.3 Creating web pages	19
CHAPTER THREE	20
3.0 METHODOLOGY.....	20
3.1 SIGNIFICANCE	20
3.3 Data collection.....	21
3.3.0 Type and Sources of Data.....	21
3.3.1 Data analysis	21
3.4 Ethical issues	22
3.5 Web Scripting with PHP	22
3.6 Database Management Systems	23
3.6.0 Database	25
3.6.1 REASONS AS TO WHY WE USE DBMS	25
3.6.2 Examples of when to use a DBMS.....	26
3.6.3 Examples of when not to use a DBMS	27

3.7 Hypertext Transfer Protocol.....	27
3.7.0 HTTP example	28
3.8 Web Servers	30
3.8.0 URLs	30
3.9 Types of Web Sites	31
3.9.0 Information	31
3.9.1 Portal.....	31
3.9.2 News.....	31
3.9.3 Business/Marketing	31
3.9.4 Educational:	31
3.9.5 Entertainment:.....	31
3.9.6 Advocacy	32
3.9.7 Blog	32
3.9.8 Personal	32
CHAPTER FOUR.....	33
4.0 Data presentation analysis and interpretation of findings	33
4.1 Introduction Data Analysis.....	33
4.1.1 Response Analysis	33
4.1.2 Summary of the response rate.	33
4.2 Cost of Services Offered	34
4.3 Focus Strategies	34
CHAPTER FIVE	35
5.0 Introduction	35
5.1 System study.....	35
5.1.1 Overview of the existing system processes.....	35
5.1.3 Overview of the new system.....	35
5.2 Requirements	35
5.2.2 Functional System Requirements	36
5.2.3 Non Functional Requirements	36
5.2.4 System Requirements	36
5.3 System Design	37

5.3.1 High Level Application Design	37
5.3.2 Relational Database Design	37
5.3.3 Actual System Design features and Interface design.....	37
5.4 Diagram Designs of the system.....	38
5.5 Physical design.....	39
5.6 Implementation	40
5.6.1 Sample codes implementation	40
5.7 Documentation.....	55
5.7.1 Installation.....	55
5.8 User training.....	56
5.9 System conversion	56
5.10 Testing Plan.....	56
5.11 Testing Schedule	57
5.11.1 Code Walk-through.....	57
5.11.2 Component testing	57
5.11.3 Bottom-up integration test.....	57
5.11.4 System Test.....	58
5.12 CONCLUSION	58
References	60

CHAPTER ONE

Introduction

Web design is the skill of creating presentations of content (usually hypertext or hypermedia) that are delivered to an end-user through the World Wide Web, using a Web browser or other Web-enabled software . The intent of web design is to create a website—a collection of electronic documents and applications that reside on a Web server/servers. The website may include text, images, sounds and other content, and may be interactive.

1.0 Background

KIU is a private University in Uganda which has builds a name in Uganda's education sector with remarkable resilience. The University is chartered by the Government of Uganda as a recognized and acknowledgement of its ability to and competence to offer relevant Education and is a member of the pre-eminent bodies which is commonwealth Universities, Association of African University and Internet University Council of East Africa.

Kampala international website was published and is considered to be the first website in 1998. Berners-Lee (1991) was the first to combine Internet communication (which had been carrying email and the Usenet for decades) with hypertext (which had also been around for decades, but limited to browsing information stored on a single computer, such as interactive CD-ROM design). Websites are written in a markup language called HTML, and early versions of HTML were very basic, only giving a website's basic structure (headings and paragraphs), and the ability to link using hypertext. This was new and different from existing forms of communication - users could easily navigate to other pages by following hyperlinks from page to page.

As the Web and Web design progressed, the markup language changed to become more complex and flexible, giving the ability to add objects like images and tables to a page. Features like tables, which were originally intended to be used to display tabular information, were soon subverted for use as invisible layout devices. With the advent of Cascading Style Sheets (CSS), table-based layout is commonly regarded as outdated. Database integration technologies such as server-side scripting and design standards like W3C further changed and enhanced the way the Web is made. As times change, websites are changing the code on the inside and visual design on the outside with ever-evolving programs and utilities.

With the progression of the Web, tens of thousands of Web design companies have been established around the world to serve the growing demand for such work. As with much of the information technology industry, many Web design companies have been established in technology parks in the developing world as well as many Western design companies setting up offices in countries such as India, Romania, and Russia to take advantage of the relatively lower labor rates found in such countries.

1.1 LOCATION

Kampala International University (main campus) is located along Ggaba road in Kansanga, a rapidly growing suburb of Kampala, 3 km from the city centre and 2 km from the shores of Lake Victoria, the second largest fresh water lake in the world. The location of the university, near Uganda's commercial capital, provides students with opportunities for social and cultural enrichment. Kampala International University (western campus) is located in Ishaka, Bushenyi district. The beauty of the country side environment of the western campus provides an ambience for comfort and concentration.

Kampala International University (KIU), commenced operation as a private University in October 2001. The university offers both undergraduate and post-graduate programs on full-time as well as part-time basis.

All KIU academic programs are conducted on a semester basis except the Access Program, Health Sciences, In-service Education courses, and the MBA (Executive), which follow separate schedules

From the diversity of KIU members, there is a need for easy access to reliable information by any KIU and prospective members about these accommodation services provided to members of KIU. This is not currently the case, especially for new individuals who have never been to Kampala city or Uganda.

KIU has institution web site with information about its academic programs and news. This project can be linked to KIU web site so as to provide those requiring services to access this information via link. This web site can eventually owned and customized by KIU if it decide to expand their registration services to gather for students on line registration service.

Kampala International University online registration will benefit KIU and non-KIU members seeking information about the courses they want to undertake in the university.

1.2 Problem statement

Information on availability of vacancies on any given faculty is currently not available online and informal posters that cannot be located with ease nor is its content reliable. Most students have to travel from different countries to Uganda seeking for admission forms.

The following problems have to be encountered in seeking admission in KIU:

- i. Resources are wasted including expenses of taxi fare and time spend moving from one country to another
- ii. Lack of location site map to guide individuals who have been to Kampala city before, or those who are not well vast with the area.
- iii. Unavailability of online contact details of service providers that can be used to make prior inquiry

1.3 Objectives

1.3.0 General objective

The research project is aimed at establishing a web site that will provide invaluable information to individuals of KIU so as to locate and acquire admission wherever they are in the world within

shortest time possible and use of less resources and also it is aimed at bringing together the Web and databases. The Kampala international university Online registration, application has many components of a typical web database application, including:

- Web pages populated with data from a database
- User-driven querying and browsing, in which the user provides the parameters that limit the searching or browsing of the database
- Data entry and validation. HTML <form> widgets collect data, and JavaScript client-side scripts and PHP server-side scripts perform validation.
- User tracking; that is, session management techniques that add state to HTTP
- User authentication and management
- Reporting

Most web database applications do this through three layers of application logic. At the base are a database management system (DBMS) and a database. At the top is the client web browser used as an interface to the application. Between the two lies most of the application logic, usually developed with a web server-side scripting language that can interact with the DBMS, and can decode and produce HTML used for presentation in the client web browser.

1.3.1 Specific objectives

The specific objectives of this project are:

- i. Come up with strategies used by the institution to register students online.
- ii. Develop cost leadership strategies being used to overcome increased competition in business.
- iii. Develop strategies that can be used by the institution to increase number of students.

1.4 Research questions

In order to answer the above stated research purpose this study will seek to answer the following research questions

- (i) What are the cost leadership strategies being used by the web designers due to increased competition in business?
- (ii) What are the differentiation strategies web designers are using due to increased competition?
- (iii) Which are the web designers focus strategies used due to increased competition?
- (iv) Which are the promotional strategies being used by the institution due to increased competition?

1.5 Justification

Nowadays website becomes an essential part of every profession. Website makes strong impact on the image of an institution like KIU. It is very important when you are more concern about exploring your business worldwide. Also With the growth of the Web over the past decade, there has been a similar growth in services that are accessible over the Web. Many new services are web sites that are driven from data stored in databases. Examples of web database applications include news services that provide access to large data repositories, e-commerce applications such as online stores, and business-to-business (B2B) support products.

Database applications have been around for over 30 years, and many have been deployed using network technology long before the Web existed. The point-of-service systems used by bank tellers are obvious examples of early networked database applications. Terminals are installed in bank branches, and access to the bank's central database application is provided through a wide area network. These early applications were limited to organizations that could afford the specialized terminal equipment and, in some cases, to build and own the network infrastructure.

The Web provides cheap, ubiquitous networking. It has an existing user base with standardized web browser software that runs on a variety of ordinary computers. For developers, web server software is freely available that can respond to requests for both documents and programs. Several scripting

languages have been adapted or designed to develop programs to use with web servers and web protocols.

If you are into the business, an owner of small or medium or big company, or a professional or free lancer you need to have your own website for many reasons.

There are approximately 970 million Internet users, 15% of the world's population. Thus it is a most efficient way to get potential customers from all over the world through internet marketing.

Websites have no boundaries as far as place or region or country is concern. It is international. If you have a website, then you have something international that overcomes the limits between countries and continents.

Today, many businesses depend on their website. They gained vast exposure and have increased their sales revenue by developing professional website design and search engine friendly website.

The project will be conducted in the following stages:

- Data collection
- Web site design and development
- Web site delivery and maintenances
- Documentation

CHAPTER TWO

2.0 LITERATURE REVIEW

In mid-1960s, Ted Nelson was trying to devise a computer system that store literary documents, link them according to logical relationships, and allow reader to comment and annotate what they read (June Jamrich Parsons et al (2000)). He envisioned a set of documents, which he called a “Hypertext”, connected by a set of “hypertext links” that could be navigated to view additional material related to a topic.

Web site development involves developing web pages and linking them in logical manner. It involves relevant content such as graphics, text, among others depending on the type of web site. Web sites can viewed over the computer networks.

Computer networks

A computer network is an interconnection of computers and computing equipment using either wires or radio waves over small or large geographical areas (Curt M. white).

A computer networks network that use radio waves are termed wireless and can involve broadcast radio, microwaves, or satellite transmissions.

Networks that are small in geographic size-spanning a room, a building or a campus-are called Local Area Network (LAN).

Networks that serve an area of 3 to 30 miles-approximately the area of a typical city-are called Metropolitan Area Networks (MAN).

Large networks encompassing parts of states, multiple states, counties, and the world are called Wide Area Network (WAN).

When a computer connects to a network, it is online. Networks allow computers to share resources, such as hardware, software, data, and information. Sharing resources saves time and money. In many networks, one or more computers act as a server. The server controls access to the resources on the network. The other computers on the network, each called a client or workstation request resources from the server. The server ordinarily has more power, more storage space, and expanded

communication capabilities than the client computers. The world's largest computer network is the Internet.

KIU presently has six computer labs, all networked. Four of these computer labs are connected to the internet while the rest have local area connection (LAN). Wireless connection is also available within the University

2.1 The internet

The Internet is a worldwide collection of computer networks-sharing digital information via a common set of networking and software protocols.

The internet began in 1960s as an experiment in design of a robust computer networks. The goal was to construct a network of computers that should withstand the loss of several machines without compromising the ability of the remaining ones to communicate. Funding came from U.S. department of Defense, which had vested interest in building information networks that could withstand nuclear attack.

The resulting network was a marvelous technical success it was limited in size and scope for the most part, only defense contractors could gain access to what was then known as the ARPAnet.

With the advent of high-speed modems for digital communication over common phone lines began connecting and taking advantages of the network's advance and global communications. Nonetheless, it was not until 1993 that the Internet really took off.

The Internet was a tough to hoe back then. Getting connected and using the various software tools, if they were even available for their computers, presented an insurmountable technology barrier for most people. And most available information was plain-vanilla ASCII text about academic subjects. The internet was just too disorganized, and, outside of the government and academia, few people had the knowledge or interest to learn how to use the arcane software or the time to spend rummaging through the documents looking for one of interest.

Businesses, called access providers, offer user and company's access to the Internet free or for a fee. By subscribing to an access provider, you can use a computer and a communication device, such as modem, to connect to many services on the Internet. The web, short for the World Wide Web, is one of the more popular services on the Internet.

At about the same time the internet opened up for business, some physicist at CERN, the European Particle Physics Laboratory, released an authoring language and distribution system they developed for creating and sharing multimedia-enabled, integrated electronic documents over the Internet. And so was born HTML, browser software, and the web.

KIU fraternity presently enjoys internet service through internet service provider (ISP) Uganda Telecommunication Limited (UTL).

2.2 Browsing the Web

A web browser, or browser, is application software that allows users to access and view web pages. There are many browser products available, and each browser product has different features. The two most popular windowing-based browsers are Netscape and Internet Explorer. The features of web browsers which have common basic set include:

- All web browsers are HTTP clients that send requests and display responses from web servers (usually in a graphical environment).
- All browsers interpret pages marked up with HTML when rendering a page; that is, they present the headings, images, hypertext links, and so on to the user.
- Some browsers display images, play movies and sounds, and render other types of objects.
- Many browsers can run JavaScript that is embedded in HTML pages. JavaScript is used, for example, to validate a <form> or change how a page is presented based on user actions.
- Selected web browsers can run components developed in the Java or ActiveX programming languages. These components often provide additional animation, tools that can't be implemented in HTML, or other, more complex features.
- Several browsers can apply Cascading Style Sheets (CSS) to HTML pages to control the presentation of HTML elements.

Web browsers are the most obvious example of a user agent, a software client that requests resources from a web server. Other user agents include web spiders—automated software that crawls the Web and retrieves web pages—and proxy caches, software systems that retrieve and locally store web pages on behalf of many other user agents.

2.3 Creating web pages

Web pages are created using Hypertext Markup Language (HTML) generated manually with a text-based editor such as Notepad or by using a web page authoring tool.

When we use the term "the Web," we mean three major, distinct standards and the tools based on these standards: the Hypertext Markup Language (HTML), the Hypertext Transfer Protocol (HTTP), and the TCP/IP networking protocol suite. HTML works well for structuring and presenting information using a web browser application. TCP/IP is an effective networking protocol that transfers data between applications over the Internet and has little impact on web database application developers. The problem in building web database applications is interfacing traditional database applications to the Web using HTTP. This is where the complex application logic is needed.

CHAPTER THREE

3.0 METHODOLOGY

3.1 SIGNIFICANCE

Life in the 21st century is full of technological advancement and in this technological age it is very difficult for a business to survive without a website. Businesses without a website in today's date tend to lose large number of its potential customers. Each and every individual from a kid to an elderly person prefer the internet to look out for their desired products. The number of search engines is increasing day by day and if your business is not on these engines you tend to loose your customers to your competitors. Every businessman has the same question on his mind; I have fared well without a website so far, why would I need one now? The answer is simple, for advertising. Advertising plays a very important role in luring customers to buy your products thus enhancing your business. As times have changed the media of advertising have modified, starting from print media to televisions and radios it has now reached to computers. The business has to change as per the latest trends to remain flourishing in the market. As it is said "Something that is permanent in this world is 'Change' "

A website has now become the means of generating more business than any other medium of advertising. Internet today is being used 24 / 7 and is accessible from every house. People are highly dependant on the internet for small and inconspicuous things like a pin or a paper. Today everything you need is achieved at the click of a button. Online shopping trends have totally changed the traditional way of shopping. The million dollar question is, Are you smart enough to spend some money on building an attractive website to enhance your business? The answer should be yes! Having a website in today's date is equivalent to having a record of your company in yellow pages (back in 10- 15 yrs). A website for your business can serve several different purposes. It can give your customers a detailed description about your company and its products as well as it can be used as a brochure to advertise your products. And the biggest advantage is it is available to view 24 / 7 for 365 days a year. Existing customers can also keep in touch with the latest offers or products etc through the website. Newsletters can be sent to the customers who have signed up on the website for the same so they

get the updates regularly. In fact having a news section on the website is also an option which can be updated periodically, when it's not possible to send newsletters. Thus it's ultimately beneficial to both your existing as well as new customers. In short to brief up all, a website for your business is essential to increase your profits by generating new business and also keeping your existing customers up-to-date with any changes or product ranges.

3.3 Data collection

The method used to collect data in this study was questioning and observation. The questioning involved designing and distributing questionnaires to the respondents and interviewing the management staffs. This is because clarification was easily done where the questions were not well understood. The questionnaires were also sent out to the respondent as this allowed them to express themselves freely at their own convenient time without fear of intimidation. In interviewing the main instruments used was interview guide and the researcher recorded the information as the interviewee respond. In questioning both structured and unstructured questions were used, so as to ensure that the respondents are free to express their opinion on the competitive strategies used within the organization due to increased competition. The study employed verbal and non-verbal method to collect data. Verbal method included questioning and interviews whereas non-verbal methods involved written response to questionnaires.

The procedure of data collection involved designing and distributing questionnaires, arranging for interviews and observing the systems as well as the employees as they work.

3.3.0 Type and Sources of Data

Primary data was collected from the respondents through interviews and questionnaires and secondary data was obtained from the previous reports of the web designers.

3.3.1 Data analysis

The data collected from the organization was presented using data entry template using statistical package for social scientist (spss). The data was run and checked for any entry error and was interpreted and presented using, tables, bar graphs, pie charts and percentages, to analyze the result of using the various strategies due to increased competition.

3.4 Ethical issues

The researcher made use of introduction letter from the university which gave a background and purpose of the study and the researcher sought permission from the management of the Kampala international university to allow collection of data from the organization. And the researcher will ensure confidentiality of the information received and will later communicate to the organization by handing to them a copy of study.

3.5 Web Scripting with PHP

PHP has emerged as a component of many medium- and large-scale web database applications. This isn't to say that other scripting languages don't have excellent features. However, there are many reasons that make PHP a good choice, including:

- PHP is open source, meaning it is entirely free. As such, community efforts to maintain and improve it are unconstrained by commercial imperatives.
- One or more PHP scripts can be embedded into static HTML files and this makes client-tier integration easy.
- There are over 15 libraries for native, fast access to the database tier.
- Fast execution of scripts. With the new innovations in the Zend engine for script processing, execution is fast, and all components run within the main memory space of PHP (in contrast to other scripting frameworks, in which components are in distinct modules). Empirical evidence suggests that for tasks of at least moderate complexity, PHP is faster than other popular scripting tools.
- Platform and operating-system flexibility. Apache runs on many different platforms and under selected operating systems; PHP runs on all these and more when integrated with other web servers.
- PHP is suited to complex systems development. It is a fully featured programming language, with more than 50 function libraries.

When a user agent makes a request to the web server for a PHP script, six steps occur:

1. The web server passes the request to the Zend engine's web server interface.
2. The web server interface calls the Zend engine and passes parameters to the engine.
3. The PHP script is retrieved from disk by the engine.

4. The script is compiled by the runtime compiler.
5. The compiled code is run by the engine's executor and may include calls to function modules. The output of the executor is returned to the web server interface.
6. The web server interface returns output to the web server (which, in turn, returns the output as an HTTP response to the user agent).

How the PHP scripting engine is managed and run depends on how the PHP module is included in the Apache web server installation process. The PHP module library is statically linked with the Apache http binary executable. This means that the PHP scripting engine is loaded into main memory when Apache runs, making the PHP engine run faster. The drawbacks are that Apache with a static PHP library consumes more memory than if the module is loaded dynamically, and that the module upgrade process is less flexible.

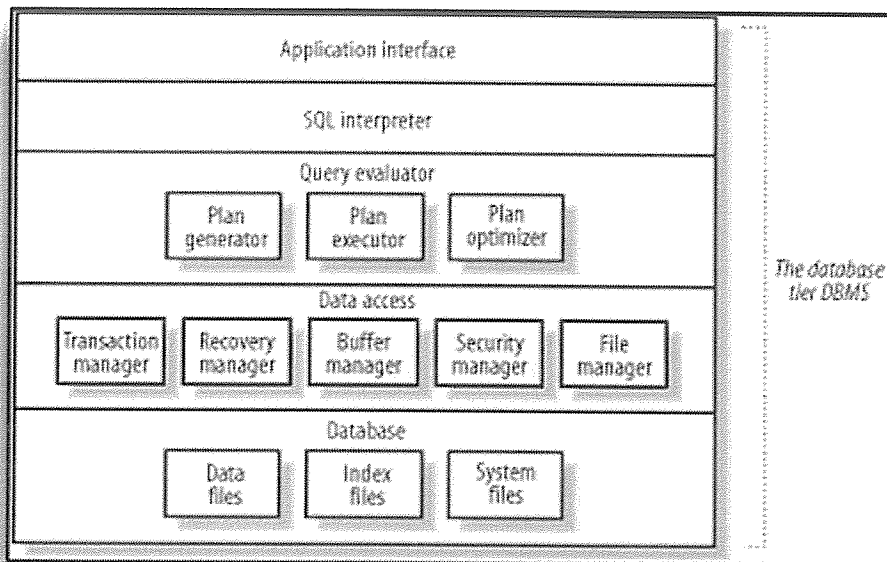
3.6 Database Management Systems

A database management system stores, searches, and manages data.

A database is a collection of related data. The data stored can be a few entries, or rows, that make up a simple address book of names, addresses, and phone numbers. In contrast, the database can also contain millions of records that describe the catalog, purchases, orders, and payroll of a large company. The database behind the project, Kampala international university online registration, is an example of a medium-sized database that falls between these two extremes.

A DBMS is a set of components for defining, constructing, and manipulating a database. When we refer to a database management system, we generally mean a relational DBMS or RDBMS. Relational databases store and manage relationships between data—for example, students placing registration requirements, customer orders containing line items.

The figure below shows the simplified architecture of a typical DBMS.



A DBMS consists of several components:

Applications interface

Libraries for communicating with the DBMS. Most DBMSs have a simple command-line interpreter that often uses these libraries to relay requests typed from the keyboard to the DBMS and to display responses. In a web database application, the command-line interpreter is usually replaced by a function library that is part of the middle-tier scripting language.

SQL interpreter

A parser that checks the syntax of incoming query statements and translates these into an internal representation.

Query evaluator

Generates different plans for evaluating a query by considering database statistics and properties, selects one of these plans, and translates the plan into low-level actions that are executed.

Data access

The modules that manage access to the data stored on disk, including a transaction manager, a recovery manager, the main-memory buffer manager, data security manager, and the file and access method manager.

3.6.0 Database

The physical data itself stored in data files. The data also contains index files for fast access to data, and database and system summary statistics primarily used for query plan generation and optimization.

The important components for web database application developers are the database and applications interface. For all but large-scale applications, understanding and configuring the other components of a DBMS is usually unnecessary.

3.6.1 REASONS AS TO WHY WE USE DBMS

A question that is often asked is: why use a complex DBMS to manage data? There are several reasons that can be explained by contrasting a database with a spreadsheet, a simple text file, or a custom-built method of storing data.

Take spreadsheets as an example. Spreadsheet worksheets are typically designed for a specific application. If two users store names and addresses, they are likely to organize data in a different way—depending on their needs—and develop custom methods to move around and summarize the data. In this scheme, the program and the data aren't independent: moving a column might mean rewriting a macro or formula, while exchanging data between the two users' applications might be complex. In contrast, a DBMS and a database provide data-program independence, where the method for storing the data, the order of the stored information, and how the data is managed on disk are independent of the software that accesses it.

Managing complex relationships is difficult in a spreadsheet or text file. For example, consider our online registration: if we want to store information about students, we might allocate a few spreadsheet columns to store each student residential address. If we were to add student physical addresses and postal addresses, we'd need more columns and complex processing to, for example, process a mail-out to students. If we want to store information about the fee payment of student, the

spreadsheet becomes wider still, and problems start to emerge. For example, it is difficult to determine the maximum number of columns needed to store orders and to design a method to process these for reporting.

Spreadsheets or text files don't work well when there are associations or relationships between stored data items. In contrast, DBMSs are designed to manage complex relational data. DBMSs are also a complete solution: if you use a DBMS, you don't need to design a custom spreadsheet or file solution. The methods that access the data—most often the query language SQL—are independent of how the data is physically stored and actually processed.

A DBMS usually permits multiuser transactions. Medium- and large-scale DBMSs include features that control the writing of data by multiple users in a methodical way. In contrast, a spreadsheet should be opened and written only by one user; if another user opens the spreadsheet, she won't see any updates being made at the same time by the first user. At best, a shared spreadsheet or text file permits very limited concurrent access.

An additional benefit of a DBMS is its speed. It isn't totally true to say that a database provides faster searching of data than a spreadsheet or a custom file system. In many cases, searching a spreadsheet or a special-purpose file might be perfectly acceptable, or even faster if it is designed carefully and the volume of data is small. However, for managing large amounts of related information, the underlying search structures in a DBMS can permit fast searching, and if information needs are complex, a DBMS should optimize the method of retrieving the data.

There are also other advantages of DBMSs, including data-oriented and user-oriented security, administration software, and data recovery support. A practical benefit is reduced application development time: the system is already built; it needs only data and queries to access the data.

3.6.2 Examples of when to use a DBMS

In any of these situations, a DBMS should probably be used to manage data:

- There is more than one user who needs to access the data at the same time.
- There is at least a moderate amount of data. For example, we may need to maintain information about a few hundred students.

- There are relationships between the stored data items. For example, students may have any number of related inquiries orders.
- There is more than one kind of data record. For example, there might be information about students, registration, inventory, and other data in an online store.
- There are constraints that must be rigidly enforced on the data, such as field lengths, field types, uniqueness of customer numbers, and so on.
- New or consolidated information must be produced from basic, related information; that is, the data must be queried to produce reports or results.
- There is a large amount of data that must be searched quickly.
- Security is important. There is a need to enforce rules as to who can access the data.
- Adding, deleting, or modifying data is a complex process.

3.6.3 Examples of when not to use a DBMS

There are some situations where a relational DBMS is probably unnecessary or unsuitable. Here are some examples:

- There is one type of data item, and the data isn't searched. For example, if a log entry is written when a user logs in and logs out, appending the entry to the end of a simple text file may be sufficient.
- The data-management task is trivial. In this case, the data might be coded into a web script in the middle tier, rather than adding the overhead of a database access each time the data is needed.
- The data requires complex analysis. For analysis, a spreadsheet package or statistical software may be more appropriate.

3.7 Hypertext Transfer Protocol

The three-tier architecture provides a conceptual framework for web database applications. The Web itself provides the protocols and network that connect the client and middle tiers of the application; that is, it provides the connection between the web browser and the web server. HTTP is one component that binds together the three-tier architecture. it's important to understand the

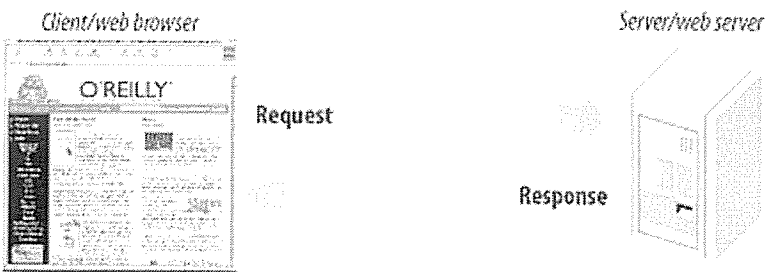
problems HTTP presents for web database applications. The HTTP protocol is used by web browsers to request resources from web servers, and for web servers to return responses.

HTTP allows resources to be communicated and shared over the Web. From a network perspective, HTTP is an applications-layer protocol that is built on top of the TCP/IP networking protocol suite. Most web servers and web browsers communicate using the current version, HTTP/1.1. Some browsers and servers use the previous version, HTTP/1.0, but most HTTP/1.1 software is backward-compatible with HTTP/1.0.

HTTP communications dominate Internet network traffic. In 1997, HTTP accounted for about 75% of all traffic. I speculate that this percentage is now even higher due to the growth in the number and popularity of HTTP-based applications such as free email services.

3.7.0 HTTP example

HTTP is conceptually simple: a client web browser sends a request for a resource to a web server, and the web server sends back a response. The HTTP response carries the resource—the HTML document, image, or output of a program—back to the web browser as its payload. This simple request-response model is shown below



A web browser makes a request and the web server responds with the resource

An HTTP request is a textual description of a resource and additional header information. Consider the following example request:

```
GET /index.html HTTP/1.0
From: hugh@computer.org (Hugh Williams)
User-agent: Hugh-fake-browser/version-1.0
```

Accept: text/plain, text/html

This example uses a GET method to request an HTML page index.html with HTTP/1.0. In this example, three additional header lines identify the user and the web browser and define what data types can be accepted by the browser. A request is normally made by a web browser and may include other headers; the previous example was created manually by typing the request into Telnet software.

An HTTP response has a response code and message, additional headers, and usually the resource that has been requested. An example response to the request for index.html is as follows:

```
HTTP/1.0 200 OK
Date: Sat, 21 Jul 2002 03:44:25 GMT
Server: Apache/1.3.20
Content-type: text/html
Content-length: 88
Last-modified: Fri, 1 Feb 2002 03:40:03 GMT
```

```
<html><head>
<title>Test Page</title></head>
<body>
<h1>It Worked!</h1>
</body></html>
```

The first line of the response agrees to use HTTP/1.0 and confirms that the request succeeded by reporting the response code 200 and the message OK; another common response is 404 Not Found. In this example, five lines of additional headers identify the current date and time, the web server software, the data type, the length of the response, and when the resource was last modified. After a blank line, the resource itself follows. In this example the resource is the requested HTML document, index.html.

HTTP is stateless. Statelessness means that any interaction between a web browser and a web server is independent of any other interaction. Each HTTP request from a web browser includes the

same header information, such as the security credentials of the user, the types of pages the browser can accept, and instructions on how to format the response. Statelessness has benefits: the most significant are the resource savings from not having to maintain information at the web server to track a user, and the flexibility to allow users to move between unrelated pages or resources.

Because HTTP is stateless, it is difficult to develop stateful web database applications. What is needed is a method to maintain state in HTTP so that information flows and structure can be imposed. A common solution is to exchange a token between a web browser and a web server that uniquely identifies the user and her session. Each time a browser requests a resource, it presents the token, and each time the web server responds, it returns the token to the web browser. The token is used by the middle-tier software to restore information about a user from her previous request, such as which menu in the application she last accessed. Exchanging tokens allows stateful structure such as menus, steps, and workflow processes to be added to the application.

3.8 Web Servers

Web servers are often referred to as HTTP servers. The term "HTTP server" is a good summary of their function: their basic task is to listen for HTTP requests on a network, receive HTTP requests made by user agents (usually web browsers), serve the requests, and return HTTP responses that contain the requested resources.

There are essentially two types of request made to a web server: the first asks for a file—often a static HTML web page or an image—to be returned, and the second asks for a program to be run and its output to be returned to the user agent. Requests for web scripts that access a database are examples of HTTP requests that require a server to run a program. The HTTP requests are for PHP script resources, which require that the PHP Zend engine be run, a script retrieved and processed, and the script output captured.

3.8.0 URLs

A web site is composed of a series of web pages that contain explanation. Each page is stored as a separate file and referred to by a unique URL. Like a domain name, URL (Uniform Resource Locator) is an Internet address. A URL, however, is the address of a document on a computer, whereas a domain name represents the IP address of a computer. The components of a URL include

the web protocol, the web server name, the folder in which the page is stored, and the filename of the page.

Web page URLs begin with “http://”. The HTTP is a protocol, a communication system that allows Web browser to communicate with the web servers. The next part of URL is the web server name. A web server is a computer that uses special software to transmit web pages over the Internet. Many web server names are prefixed with www. By entering the web server name, you enter the site’s home page.

A home page is similar to the title page in a book. It identifies the site and contains links to other pages at the site (June Jamrich Parsons et al (2000))

3.9 Types of Web Sites

3.9.0 Information: Contain factual information. Government agencies have information web sites providing information such census data, tax codes, and budgets. Other organization provide information such as public transport such as public transportation schedules and published research findings.

3.9.1 Portal: Is a Web site that a variety of Internet services from a single convenient location. Most portals offer free services such as; search engines, news, sports and weather, free web publishing services, among others.

3.9.2 News: Contains news worthy materials including stories and articles relating to current events, life, money, sports, and the weather. Newspapers, Television, and Radio stations are some of the media that maintain news websites.

3.9.3 Business/Marketing: Contains context that promote or sell products or services. Many of these companies also allow you to purchase their products or services.

3.9.4 Educational: Offers exciting and challenging avenues for formal and informal teaching and learning.

3.9.5 Entertainment: Offer music, videos, sport, games, ongoing episodes, sweepstakes, charts and more.

3.9.6 Advocacy: Contains content that describes a course, opinions, or ideas. The purpose of an advocacy web site is to convince the reader of the validity of the cause, opinion, or idea. These web sites usually present views of a particular group or association.

3.9.7 Blog: Web blog is a web site that uses a regularly updated journal format to reflect the interest, opinions, and personalities of the author, and sometimes site visitors.

3.9.8 Personal: A private individual or family may maintain a personal web site or just a single web page, for a variety of reasons.

CHAPTER FOUR

4.0 Data presentation analysis and interpretation of findings

4.1 Introduction Data Analysis

The purpose of this chapter is to present analysis of the findings for this study, and give the interpretation of results. The results of questionnaires have been integrated with findings of interviews and observations checklists that were used to supplement and gain a deeper understanding of the findings.

4.1.1 Response Analysis

The study revealed that the response rate for men was more than women; this is an indication that men have been empowered. The management committee and employees has more men than women. The response rate showed that only 85 % of the respondents replied to the questionnaire.

Table 4.1 Gender

GENDER	NO OF RESPONDENT	PERCENTAGES
Male	25	46
Female	20	44
TOTALS	45	100

From table 4.1, the study indicated that 46 % were men while women were 44 %.

4.1.2 Summary of the response rate.

Table 4.2 showing the no of people who responded from each type of category

CATEGORIES	No of respondent	PERCENTAGE
Committee- Executive	1	3
Committee-Member	1	3

Supervisory Committee	1	3
Management –Staffs	2	6
IT Staffs	1	3
Other Staffs	16	35
Students	22	47
Totals	45	100

Source author (2007)

From table 4.2 the study indicated that, the students had the highest response rate of 47 % followed by the staffs with 35 % and management staffs with 6 % while others had a response rate of 3 % each.

4.2 Cost of Services Offered

This study shows that in order to respond to the prevailing rate of competition in the business sector, the university has tried to cut down cost of some of its services in order to encourage incoming students and to retain its members in the society. The website has cut down the cost of travelling from various countries seeking for admission, also it has eased the way old students obtain their results.

4.3 Focus Strategies

The study revealed that there are competitions in every business. This is like going for a business exhibition on internet. You cannot stop competition in any business but at least make your identity so that your potential customers are given a chance of comparing and subsequently buying your products or engaging your services with the help of professional business website.

If you own small company and you want to explore globally then website is the only effective media to connect people by creative web designing with web promotion.

CHAPTER FIVE

SYSTEM ANALYSIS, DESIGN AND IMPLEMENTATION

5.0 Introduction

We carried out an extensive study and analysis of the existing system which helped us in identifying the existing system's weaknesses.

5.1 System study

5.1.1 Overview of the existing system processes

The existing system at Kampala international university studied was carried out. The existing system is mostly paper work it consumes a lot of time and it is tiresome this system will be able to make positive contributions that would help the management of Kampala international university to make necessary improvements on students admission and improve work performance in KIU.

5.1.2 Weaknesses of existing system

From what was gathered, it was made clear that most of the problems with the old system originated from inefficiency in storing students records, retrieval and manipulation of information when required. A lot of time was normally wasted while trying to carry out all these and hence there was need for a new system that would solve problems of data redundancy, time wasting, reduce search time, allow easy data manipulation, with ability to sort information, print out reports among others.

5.1.3 Overview of the new system

The new system is a Web-Based document Management System for the Kampala international university that facilitates the staff and the administration in the storage, classification and tracking of university documents, student's bills and other information.

5.1.4 Strength of the new system

The proposed system is capable of record tracking, storing, retrieval and manipulation of information when required. It is time saving as each user knows his/her role. Data redundancy is eliminated through normalized database entities. Overall the system has clerical and attractive user interfaces that need a click to reach a specific goal.

5.2 Requirements

In order to document all the end user requirements for the system, data collected was analyzed using structured analysis approach to rigorously specify the processes. This section includes the requirements, functional and non-functional requirements as follows;

5.2.2 Functional System Requirements

It should be able accept input from the user

It must provide a way of keeping the system secure by use of a password.

The system must allow the administrator to access information entered by other users of the system.

The system should enable the user to search the database using a key word. This can be by for example entering student ID, file category or any other data.

The system must only accept unique records in the database.

It must provide a way for the user to search documents.

5.2.3 Non Functional Requirements

The system was designed to fulfill the following non-functional requirements:

The system must verify and validate all user input and users must be notified in case of errors.

The system only allows authorized users to update records.

The system administrator is the only one privileged to delete or add users.

The system must allow for expansion in the future.

The system must have a high performance level. Its level of reliability must be high.

5.2.4 System Requirements

In order for the system to perform as expected, the following system specifications for hardware and software, security and operations are required.

5.2.4.1 Cost

Cost or economic feasibility was put in place to determine whether the management of the Kampala international university had the ability to fund the project.

5.2.4.2 Technical

Technical feasibility was aimed at finding out whether the university had technical equipments to house the proposed system in coupled to whether users know how to use and integrate with the system. It was found out that the users need further training and sensitization on how to use the system and furthermore some of the technical equipments were available for implementation of the new system.

5.2.4.3 Operation

Operational feasibility was aimed at studying the environment where the current system works. This was important to ensure that the non functional requirements of the system were applicable these included; availability of space, electricity and safety of the system equipments.

5.2.4.4 Security

In this sub section it was made possible that only those users with legitimate access rights are allowed to use the system. It means that authentication was defined through use of user names and passwords; also addition of more users and elimination of users especially those who left the Parliament were defined to be performed explicitly by the administrator.

5.3 System Design

In this section we present the system design portion of Document repository concentrating on four areas these is; high level application design, relational database, actual system design and interface design.

5.3.1 High Level Application Design

This system that is Document repository system is intended to follow the design principles of decision support system (Carter et al 1992). The figure below shows that the Document repository system is made up of five components these are:

- ☐ A Database store of all related documents to the algorithms that generate results for user queries
- ☐ A User interface comprising all code associated with screen that process user input and display system output
- ☐ And a database management system comprising code that allows all of the system components listed previously to share data.

5.3.2 Relational Database Design

We have consolidated all system entities and attributes and these may include administration, users, research documents, reports, and department's authority. This database we designed it with the use of database management tools such as MySQL in order to come up with a system that stores the recorded information.

5.3.3 Actual System Design features and Interface design

The actual system design involves the following features:

- Document search tool - The document search tool allows staff to search and sort documents that can be uploaded by users. The user enters a title and description he will be searching for.
- Document up loader - This feature of the system allows users to upload reports, research and their profiles that users can use for research/consultation purposes. The user can search for these documents.

- Check in/Check out - This feature gives a user an interface to confirm results on the system. The user can use the check in button to see what has been uploaded. They can also check out a document for editing.

Interface design

As is true for most Information systems, the user interfaces are the most visible and most important to the user therefore we identified key interface design principles these were: integration with work flow, ease of use and consistency.

5.4 Diagram Designs of the system

The system developed is based materially on what was provided by the users as requirements which means that the diagrams were used as techniques in coming up with the actual performance flow of the system. The diagrams in this section take forms of Context Diagram (CD), Data Flow Diagrams (DFD), Flow Chart (FC), Relationships and Entity Relationships Diagram (ERD). The diagrams were so useful in describing the system diagrammatically which simplified the coding process after analyzing the performance. The following are the diagrams used:

Workspaces architecture. At the core is the Workspaces engine, which includes an XSL processor for executing locally accessed or remote steps. All documents are stored in the XML spaces repositories. The user interacts with the Workspaces engine directly and via invoked applications

Relationship Diagrams

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)		UNSIGNED	Yes	NULL	
<input type="checkbox"/>	admin	tinyint(4)			Yes	NULL	

Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> <u>id</u>	int(11)		UNSIGNED	No		auto_increment
<input type="checkbox"/> username	varchar(25)	latin1_swedish_ci		No		
<input type="checkbox"/> password	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/> department	int(11)		UNSIGNED	Yes	NULL	
<input type="checkbox"/> phone	varchar(20)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> Email	varchar(50)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> last_name	varchar(255)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> first_name	varchar(255)	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> pw_reset_code	char(32)	latin1_swedish_ci		Yes	NULL	

5.5 Physical design

This part of the system development life cycle was concerned with the actual construction of the designed system, testing to see whether it fulfilled the objectives, deliver the system into day to operation, user training and finally commissioning of the system into operation.

The system was designed in PHP, Java script and XHTML and the ease that it provides in linking with the database. The database design was implemented using MySQL. This was chosen because it is more secure compared to other DBMS.

After the system has loaded, it provides the home page which is the index which has links on which the user selects, according to the accessibility that the user was authenticated. Each user was given different access credentials where by the user first registers with the system then it generates a password which after login using the details he or she registered with. This was manually designed because different users access the system at different intervals hence sessions are provided to a particular user at a particular time

If you are new to the system, you register as a new user

The user feeds in the first name, last name, user name, phone number, example, email address and department.

5.6 Implementation

This constitutes the implementation of the system. In this part we discuss all system features that have been implemented throughout the system design.

5.6.1 Sample codes implementation

```
<?php virtual('/op/Connections/left.php'); ?>
```

```
<?php
```

```
function GetSQLValueString($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
```

```
{  
    $theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;
```

```
    switch ($theType) {
```

```
        case "text":
```

```
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
```

```
            break;
```

```
        case "long":
```

```
        case "int":
```

```
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
```

```
            break;
```

```
        case "double":
```

```
            $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
```

```
            break;
```

```
        case "date":
```

```
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
```

```
            break;
```

```
        case "defined":
```

```
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
```

```
            break;
```



```

    }

    return $theValue;

}

$editFormAction = $_SERVER['PHP_SELF'];

if (isset($_SERVER['QUERY_STRING'])) {

    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);

}

if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {

    $insertSQL = sprintf("INSERT INTO details (stu_id, f_name, l_name, reg_no, phone, email, course,
comments) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)",

        GetSQLValueString($_POST['stu_id'], "text"),

        GetSQLValueString($_POST['f_name'], "text"),

        GetSQLValueString($_POST['l_name'], "text"),

        GetSQLValueString($_POST['reg_no'], "text"),

        GetSQLValueString($_POST['phone'], "text"),

        GetSQLValueString($_POST['email'], "text"),

        GetSQLValueString($_POST['course'], "text"),

        GetSQLValueString($_POST['comments'], "text"));

    mysql_select_db($database_left, $left);

    $Result1 = mysql_query($insertSQL, $left) or die(mysql_error());

}

?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>Untitled Document</title>

<style type="text/css">

<!--

body {

    background-color: #00FF00;

}

-->

</style></head>

<body>

<form id="form1" name="form1" method="POST" action="<?php echo $editFormAction; ?>">

<table width="423" border="1" bgcolor="#FF0000">

<tr>

<th width="205" align="left" scope="col">stu_id</th>

<th width="202" scope="col"><label>

<input name="stu_id" type="text" id="stu_id" size="40" />

</label></th>

</tr>

<tr>

<td>f_name</td>

<td><input name="f_name" type="text" id="f_name" size="40" /></td>

</tr>

<tr>

<td>l_name</td>

```

```

        <td><input name="l_name" type="text" id="l_name" size="40" /></td>

</tr>

<tr>

        <td>reg_no</td>

        <td><input name="reg_no" type="text" id="reg_no" size="40" /></td>

</tr>

<tr>

        <td>phone</td>

        <td><input name="phone" type="text" id="phone" size="40" /></td>

</tr>

<tr>

        <td>email</td>

        <td><input name="email" type="text" id="email" size="40" /></td>

</tr>

<tr>

        <td>course</td>

        <td><input name="course" type="text" id="course" size="40" /></td>

</tr>

<tr>

        <td>comments</td>

        <td><input name="comments" type="text" id="comments" size="40" /></td>

</tr>

</table>

<p>

<label>

<input type="submit" name="Submit" value="Submit" />

```

```

</label>

<input type="reset" name="Submit2" value="Reset" />

</p>

</form>

<a href="about us.html">click</a> <a href="cul.php">back</a><a href="/bit/about us.html">/bit/about
us.html</a>

</body>

</html>

<?php virtual('/bit/Connections/yap.php'); ?>

<?php
function GetSQLValueString($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
{
    $theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;

    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
            break;
        case "double":
            $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
            break;
        case "date":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";

```

```

        break;

    case "defined":

        $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;

        break;

    }

    return $theValue;

}

$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {

    $insertSQL = sprintf("INSERT INTO logon (log_id, f_name, l_name, u_name, password, email) VALUES
    (%s, %s, %s, %s, %s, %s)",

        GetSQLValueString($_POST['log_id'], "text"),
        GetSQLValueString($_POST['f_name'], "text"),
        GetSQLValueString($_POST['l_name'], "text"),
        GetSQLValueString($_POST['u_name'], "text"),
        GetSQLValueString($_POST['password'], "text"),
        GetSQLValueString($_POST['email'], "text"));

    mysql_select_db($database_yap, $yap);
    $Result1 = mysql_query($insertSQL, $yap) or die(mysql_error());
}

```

```
?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Untitled Document</title>
```

```
<style type="text/css">
```

```
<!--
```

```
body {
```

```
    background-color: #FFFFFF;
```

```
    background-image: url(/bit/P2210114.JPG);
```

```
}
```

```
-->
```

```
</style></head>
```

```
<body>
```

```
<form id="form1" name="form1" method="POST" action="<?php echo $editFormAction; ?>">
```

```
<table width="508" border="1" bgcolor="#FFFF66">
```

```
<tr>
```

```
<th width="252" align="left" scope="col">log_id</th>
```

```
<th width="240" scope="col"><label>
```

```
<input name="log_id2" type="text" id="log_id2" size="50" />
```

```
</label></th>
```

```
</tr>
```

```
<tr>
```

```
<td>f_name</td>
```

```
<td><input name="f_name2" type="text" id="f_name2" size="50" /></td>
```

```

</tr>

<tr>

<td>l_name</td>

<td><input name="l_name2" type="text" id="l_name2" size="50" /></td>

</tr>

<tr>

<td height="38">u_name</td>

<td><input name="u_name2" type="text" id="u_name2" size="50" /></td>

</tr>

<tr>

<td>password</td>

<td><input name="password2" type="text" id="password2" size="50" /></td>

</tr>

<tr>

<td>email</td>

<td><input name="email2" type="text" id="email2" size="50" /></td>

</tr>

</table>

<p>

<label>

<input type="submit" name="Submit" value="Submit" />

</label>

<input type="reset" name="Submit2" value="Reset" />

</p>

</form>

<p><a href="/bit/about us.html">click</a><a href="head.php"></a></p>

```

```

</body>

</html>

<html>

<head>

<title>Untitled Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

</head>

<body bgcolor="#CC0099" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0">

<table width="100%" border="0" cellspacing="0" cellpadding="4">

<tr>

<td rowspan="2" bgcolor="#666666"></td>

<td width="100%" height="151" bgcolor="#FFFFFF"><div align="center"><strong><font
color="#003300" size="6"></font></strong>

<table width="100%" border="0" cellspacing="1" cellpadding="2">

<tr>

<td width="20%" align="center" bgcolor="#003300"><a href="DreamTemplate_com.htm"><font
color="#00FF00">Home</font></a></td>

<td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a href="about
us.html">About us</a></font></td>

<td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="services.html">Services</a></font></td>

<td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="faculties.html">Faculties</a></font></td>

<td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="contacts.html">Contacts</a></font></td>

</tr>

</table>

</div></td>

```



```

</tr>

<tr>

  <td bgcolor="#006600">&nbsp;&nbsp;&nbsp;</td>

</tr>

</table>

<table width="100%" border="0" cellspacing="0" cellpadding="4">

  <tr>

    <td bgcolor="#FFFFFF">&nbsp;&nbsp;&nbsp;</td>

    <td width="100%" bgcolor="#006600"><strong><font color="#FFFFFF">ABOUT US
</font></strong></td>

  </tr>

</table>

<br>

<table width="90%" border="0" cellspacing="2" cellpadding="4" align="center">

  <tr bgcolor="#333333">

    <td>&nbsp;&nbsp;&nbsp;</td>

  </tr>

</table>

<table width="90%" border="0" cellspacing="4" cellpadding="4" align="center">

  <tr>

    <td width="70%" valign="top" bordercolor="#FF0000" bgcolor="#33CCFF">
<p><strong><em>INTRODUCTION</em></strong></p>

    <hr>

    <p>KIU is a private University in Uganda which has build a name in Uganda 's education sector with
remakable resilience. The University is chatered by the Governmenet of Uganda as a erecognised and
acknowledgement of its ability to and competence to offer relevant Education and is a member of the pre-
eminet bodies which is commonwealth Universities , Association of African University and Interner
University Council of East Africa.</p>

    <p><strong><em>LOCATION</em></strong></p>

```

Kampala International University (main campus) is located along Ggaba road in Kansanga, a rapidly growing suburb of Kampala, 3 km from the city centre and 2 km from the shores of Lake Victoria, the second largest fresh water lake in the world. The location of the university, near Uganda's commercial capital, provides students with opportunities for social and cultural enrichment. Kampala International University (western campus) is located in Ishaka, Bushenyi district. The beauty of the country side environment of the western campus provides an ambience for comfort and concentration.

Kampala International University (KIU), commenced operation as a private University in October 2001. The university offers both undergraduate and post-graduate programs on full-time as well as part-time basis.

All KIU academic programs are conducted on a semester basis except the Access Program, Health Sciences, In-service Education courses, and the MBA (Executive), which follow separate schedules.

	Western campus
	Admission
	Research and Evaluation
	Post Graduate
	logon form click to log

```

</table></td>

</tr>

</table>

<br>

<table width="100%" border="0" cellspacing="0" cellpadding="2" bgcolor="#CCCCCC">

<tr>

<td width="2%" bgcolor="#003300">&nbsp;</td>

<td width="98%" align="right" bgcolor="#003300"></td>

</tr>

</table>

</body>

</html>

<html>

<head>

<title>Untitled Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

</head>

<body bgcolor="#CC9966" text="#003300" leftmargin="0" topmargin="0" marginwidth="0"
marginheight="0">

<table width="100%" border="0" cellspacing="0" cellpadding="4">

<tr>

<td rowspan="2" bgcolor="#666666"></td>

<td width="100%" height="154" bgcolor="#FFFFFF"><div align="center"><strong><font
color="#003300" size="6"></font></strong></div></td>

</tr>

```

```

<tr>

<td bgcolor="#006600">    <table width="100%" border="0" cellspacing="1" cellpadding="2">

    <tr>

        <td width="20%" align="center" bgcolor="#003300"><a href="DreamTemplate_com.htm"><font
color="#00FF00">Home</font></a></td>

        <td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a href="about
us.html">About us </a></font></td>

        <td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="services.html">Services</a></font></td>

        <td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="faculties.html">Faculties</a></font></td>

        <td width="20%" align="center" bgcolor="#003300"><font color="#00FF00"><a
href="contacts.html">Contacts</a></font></td>

    </tr>

</table>

</td>

</tr>

</table>

<br>

<table width="90%" border="0" cellspacing="2" cellpadding="4" align="center">

<tr bgcolor="#333333">

<td> <table width="100%" border="0" cellspacing="0" cellpadding="4">

    <tr>

        <td bgcolor="#FFFFFF">&nbsp;</td>

        <td width="100%" bgcolor="#006600"> <font color="#FFFFFF"><strong>RESEARCH AND
EVALUATION </strong> </font></td>

    </tr>

</table></td>

</tr>

```


`<p align="justify">Implement a dynamic research program, which can benefit the students, staff and faculty. </p>`

`<h2>Objectives- </h2>`

`<h2>To: </h2>`

``

`Update the students, staff and faculty on research and evaluation through continuing education and training.`

`Coordinate research and evaluation activities with the various colleges/departments.`

`Develop interest and confidence among the students, staff and faculty in conducting experimental or non-experimental studies employing the qualitative or quantitative types of research.`

`Improve the delivery of instruction, management and services through empirical investigation.`

`Establish local, national and international linkages to highlight KIU's research capabilities through publications/exchange materials and participation in research paper contests, conventions and conferences.`

`Generate income to sustain and maintain Research and Evaluation Centre operations through external research assistance.`

`Establish a well – equipped research laboratory where experimental research studies can be conducted `

``

`<p></p> </td>`

`<td width="30%" align="right" valign="top"> <table width="227" height="334" border="1">`

`<tr>`

`<td bgcolor="#003300">Western Campus</td>`

`</tr>`

`<tr>`

`<td bgcolor="#003300">Admission</td>`

`</tr>`

`<tr>`

```

        <td bgcolor="#003300"><a href="research_&_evaluation.html"><font color="#00FF00">Research and
Evaluation</font></a></td>

    </tr>

    <tr>

        <td bgcolor="#003300"><a href="post_graduate.html"><font color="#00FF00">Post
Graduate</font></a></td>

    </tr>

</table></td>

</tr>

</table>

<br>

<table width="100%" border="0" cellspacing="0" cellpadding="2" bgcolor="#CCCCCC">

    <tr>

        <td bgcolor="#003300">&nbsp;</td>

        <td align="right" bgcolor="#003300">&nbsp;</td>

    </tr>

</table>

</body>

</html>

```

5.7 Documentation

5.7.1 Installation

This system was installed based on a server side application. To install this system, the following steps were followed.

- a) Install Wamp server on the computer where the whole project is stored
- b) After installing wamp server, copy the whole project folder to the www directory found in c:\wamp\www

- c) Start wamp server and locate the localhost which is the local server, the folder is located in the body of the local host page.
- d) Click the folder, the system will be accessed

5.8 User training

The users have not been trained; though this task was supposed to come after testing this was so because the system has not been delivered to its end users.

5.9 System conversion

The fact that the relationship between paper work and computer systems is little, we recommend parallel conversion strategy. This is a conversion strategy where both systems are kept running for some certain period of time until the new system is endorsed as a perfect system. This has an advantage that in case the new system contains some un identified errors; the old system can always be resorted too. This strategy has also been recommended because from the research we carried out, we found out that most of the likely users of this system don't understand it hence the need for them to first get experienced and thereafter they can do away with the old system.

Change over strategy – This will be affected as long as users are comfortable with the new system in place thus a complete resort to the new system.

5.10 Testing Plan

Testing is an important part of the systems design. It involves checking for correctness, which is one of the focuses of the project. The way to test to see if the system is working correctly is to use black box testing (or functional testing). What black box testing does is to feed in inputs and obtain relevant outputs. These inputs and outputs are then studied and checked to make sure that they are the desired result, in order to make sure that the system is behaving properly.

The testing process in systems like the one that we will attempt to build is not an easy job. We will not build a simple application. We will build a whole university system, which consists of sub-systems. We are planning to perform test in parallel with the implementation of the new system and we are going to follow some testing procedures described in the software engineering literature.

We will first build our database and we will insert sample data in it. Note that even if the data that we will put are sample, they are going to be taken from the real university of KIU. Here we have to mention that we will have of course a smaller in physical size database, than in a real environment, but the most important is

the quality of the sample data. And this quality will be the quality of a real environment. The functionality of the database will be tested with sample operations, such as queries, updates, insertions and deletions.

Then we will build the Web-Client's application. We will start from the user-interface in order to identify the components and objects that we will need. Then we will define the appropriate classes and methods. We will test with sample data the operation of each method independently, and inside its class. Then, we will test the functionality of each class with our database. The same testing approach we will follow with the Local-Client subsystem.

5.11 Testing Schedule

In this part we are going to analyze the testing schedule and the techniques that we are planning to use.

5.11.1 Code Walk-through

The programmer who developed it inspects the code for each part of the system. This is seen as far better to find a problem at this low level rather than later in the testing cycle where several components have been joined together.

These should be completed as soon as the developer completes the code for a section of the system. Things to look for in the code are: syntax errors, incorrect function / variable names, inefficient data structures. Once the changes are completed a second walk-through should occur to check for any problems created by changes made to the code.

5.11.2 Component testing

This should occur on the development machine, immediately after the code has been given the 'all clear' in the code walk-through. The code for the individual function is compiled and executed to see if it completes the function that it was designed to. If a section of code does not produce the required outcome then the code should be checked against the requirements specification given for that section, and rewritten as required.

5.11.3 Bottom-up integration test

Integration testing is where all the components will be combined together and checked to see if they work correctly. If they don't, then there are problems with the code and unit testing will have to be reviewed again in order to pinpoint the problem and correct it. According to Sommerville (2001), the integration test should ideally be completed on a machine that was not used during production of any individual sections of code.

5.11.4 System Test

Here the objective is to make sure that the system can perform the way the customer has asked it to and that the system design has been implemented correctly. The two major issues that need to be taken into consideration here are:

- Functional testing
- Performance testing

Functional testing checks that once the system has been integrated together, it works as desired, by checking that the functional requirements have been satisfied.

Performance testing checks that the non-functional requirements have been satisfied. This will include security matters, speed, accuracy, and reliability.

5.12 CONCLUSION

This project will enhance the use of internet for data and information communication through websites. Web site development and Internet Programming is a section of computer science that seeks to enhance communication between computer and telecommunication devices.

Any user with a web browser can access the site, browse or search for information about KIU, and view the details. The details of student include the name, course they want to apply. Such elements as text, bit-mapped images (GIFs, JPEGs) and forms can be placed on the page using HTML/XHTML/XML tags. Displaying more complex media (vector graphics, animations, videos, sounds) requires plug-ins such as Adobe Flash, QuickTime, Java run-time environment, etc. Plug-ins are also embedded into web page by using HTML/XHTML tags. Improvements in browsers' compliance with W3C standards prompted a widespread acceptance and usage of XHTML/XML in conjunction with Cascading Style Sheets (CSS) to position and manipulate web page elements and objects. Latest standards and proposals aim at leading to browsers' ability to deliver a wide variety of content and accessibility options to the client possibly without employing plug-ins. Typically Web pages are classified as static or dynamic:

- Static pages don't change content and layout with every request unless a human (web master/programmer) manually updates the page. A simple HTML page is an example of static content.

- Dynamic pages adapt their content and/or appearance depending on end-user's input/interaction or changes in the computing environment (user, time, database modifications, etc.) Content can be changed on the client side (end-user's computer) by using client-side scripting languages (JavaScript, JScript, Action script, etc.) to alter DOM elements (DHTML). Dynamic content is often compiled on the server utilizing server-side scripting languages (Perl, PHP, ASP, JSP, ColdFusion, etc.). Both approaches are usually used in complex applications. With growing specialization in the information technology field there is a strong tendency to distinguish between web design and Web development. Web design is a kind of graphic design intended for the development and styling of objects of the Internet's information environment to provide them with high-end consumer features and aesthetic qualities. This definition separates Web design from web programming, emphasizing the functional features of a web site, as well as positioning web design as a kind of graphic design.^[1] The process of designing web pages, web sites, web applications or multimedia for the Web may utilize multiple disciplines, such as animation, authoring, communication design, corporate identity, graphic design, human-computer interaction, information architecture, interaction design, marketing, photography, search engine optimization and typography.
- Markup languages (such as HTML, XHTML and XML)
- Style sheet languages (such as CSS and XSL)
- Client-side scripting (such as JavaScript)
- Server-side scripting (such as PHP and ASP)
- Database technologies (such as MySQL and PostgreSQL)
- Multimedia technologies (such as Flash and Silver light)

Web pages and websites can be static pages, or can be programmed to be dynamic pages that automatically adapt content or visual appearance depending on a variety of factors, such as input from the end-user, input from the Webmaster or changes in the computing environment (such as the site's associated database having been modified).

References

1. K. Thompson, G. J. Miller, and R. Wilder. "Wide-area internet traffic patterns and characteristics," IEEE Network, 11(6):10-23, November/December 1997.
2. Committee on Computing, Information, and Communications, "High Performance Computing and Communications: Advancing the Frontiers of Information Technology," National Science and Technology Council, Washington, D.C., 1997.
3. Boothroyd, G. Dewhurst P. and Knight W. (1994). Product Design for Manufacture and Assembly, Marcel Dekker, Inc.,.
4. Denis Borodayev. Web site as a Graphic Design Object. Monograph.
5. Web Content Accessibility Guidelines (WCAG) 2.0. December 11, 2008. <http://www.w3.org/TR/WCAG20/>.
6. "Berners-Lee on the read/write web". London: BBC News. 2005-08-09. <http://news.bbc.co.uk/1/hi/technology/4132752.stm>. Retrieved 2010-03-24.
7. Web warrior series.planning, development and marketing successful websites by Jason miletsky.
8. Web design workshop by John Tollet, Robin Williams, David Rohr
9. Barbara K. Kaye, Norman J.Medoff. The world Wide Web, A mass communication Perspective
10. Lynch P. J and Horton S. Web style Guide-Basic Design principles for Creating Websites
11. Succeeding with Technology Computer system concepts for Real life by Ralphm Stair. Kenneth J. Baldauf.
12. World Wide Web Beyond the Basics by Marc Abrams
13. Sun Microsystems Inc., TCP/IP and Data Communications Administration Guide, part1 number 802-5753-10.
14. Rajiv Khosla and Dillon T. S. Engineering Intelligent Hybrid Multi-agent Systems.