**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Jnana Sangama, Belagavi – 590 014**

 **A Database Management System Project Report On**

**“ASSET MANAGEMENT”**

**Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of**

**Bachelor of Engineering**

**In**

**Computer Science & Engineering**

**ByALI NAVEED (4MW16CS007)**

**KEERTHAN ACHARYA (4MW16CS028)**

**Under the guidance of**

**Mr. NAGARAJ BHATAssistant Professor**

****

**Department of Computer Science and Engineering**  **SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**Vishwothama Nagar, BANTAKAL – 574 115, Udupi DistrictNOVEMBER, 2018**

**Shri Madhwa Vadiraja Institute of Technology and Management**

(A Unit of Shri Sode Vadiraja Mutt Education Trust ®, Udupi)

**Vishwothama Nagar, BANTAKAL – 574 115, Udupi District, Karnataka, INDIA**

**Department of Computer Science and Engineering**

***Certificate***

Certified that the Database Management System Project Work titled **‘ASSET MANAGEMENT’** has been carried out by **Mr.** **KEERTHAN ACHARYA** (**4MW16CS028) and Mr. ALI NAVEED (4MW16CS007),** who are the bonafide students ofShri Madhwa Vadiraja Institute of Technology and Management, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering ofVisvesvaraya Technological University, Belagaviduring the year 2018-19.This Database Management System Project Report has been approved as it satisfies the academic requirements with respect to the project work guidelines prescribed for the said Degree.

**Mr. NAGARAJ BHAT Dr. VASUDEVA**

Professor & Project Guide Professor and Head

Dept. of CSE Dept. of CSE

**External Viva**

**Name of the Examiners: Signature with Date**

1.

2.

**Acknowledgements**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

It is our pleasure to express our heartfelt thanks to Mr. Nagaraj Bhat, Assistant Professor, Department of Computer Science and Engineering, for his supervision and guidance which enabled us to understand and develop this project.

We are indebted to Prof. Dr. Thirumaleshwara Bhat, Principal and Prof. Dr. Vasudeva, Head of the Department, for their advice and suggestions at various stages of the work.

Special thanks go to the Management of Shri Madhwa Vadiraja Institute of Technology and Management, Bantakal, Udupi for providing us with a good study environment and laboratories facilities. Besides, we appreciate the support and help rendered by the teaching and non-teaching staff of Computer Science and Engineering.

Lastly, we take this opportunity to offer our regards to all of those who have supported us directly or indirectly in the successful completion of this project work.

Ali Naveed

Keerthan Acharya

**ABSTRACT**

Today, Digital Asset Management is an integral organizational component that companies must

adopt if they are to compete in the marketplace. However, many companies find it difficult to implement technological strategies to meet the marketplace challenges of increased competition, increased customer demands, and the demands of Internet revolution.

To implement Digital Asset Management solutions, traditional companies must be able to adapt. Organizational success in technological initiatives hinges on whether or not companies can make the transformation from managing tangible products to storing intangible data as a service. As asset management becomes a mounting concern, total communication across the organization will become increasingly relevant. A company’s Digital Asset Management strategy has to revolve around the optimal creation, use, reuse, and repurposing of its assets.

**Table of Contents**

**Page No.**

**Acknowledgements** i

**Abstract** ii

**Table of Contents** iii

**List of Figures** iv

**Chapter 1 Introduction**

1.1 Digital Asset Management 1

1.2 Classification 2

1.3 Modules 3

1.3.1 Administration 3

1.3.2 Asset Management 3

1.3.3 Registration 3

1.3.4 Reports 3 1.3.5 Roles 3

**Chapter 2 Literature Review 4**

2.1 General 4

2.2 Enterprise DAM Systems 4

2.3 Pay-As-You-Go DAM Systems 4

2.4 Mid-Range DAM Systems 5

2.5 Desktop or Workgroup DAM Systems 5

**Chapter 3 System Environment 6**

3.1 Hardware Configuration 6

3.2 System Configuration 6

3.3 Software Features 6

3.3.1 PHP 7

3.3.2 Security 7

3.3.3 Syntax 8

3.3.4 MySQL 9

3.3.5 PHPMyAdmin 10

3.3.6 Apache Web server 11

**Chapter 4 System Design 13**

4.1 Input Design 13 4.1.1 Administrator 13

4.1.2 Users 14

4.2 Data Flow Design 14

4.3 Database Design 15

4.4 Output Design 16

**Chapter 5 Conclusion 18**

5.1 General Conclusion 18

**References**  19

**List of Figures**

|  |  |
| --- | --- |
|  | Page No. |
| Figure 3.1 | XAMMP Control Panel v3.2.2 7 |
| Figure 3.2 | MySQL Tables 10 |
| Figure 3.3  Figure 3.4  Figure 3.5  Figure 3.6  Figure 3.7 | PHPMyAdmin 10    Admin Table 13  Users Tables 14  Dataflow Design 15  Database Design 16 |
|  |  |

**Chapter 1**

**Introduction**

* 1. **Digital Asset Management**

Digital asset management (DAM) has been referred to by a variety of names that often are used interchangeably, such as enterprise content management (ECM), digital content management (DCM), digital media management (DMM), media asset management (MAM), and rich media asset management (RMAM). From its earliest days, DAM started out as the most rudimentary digital storage system used to store large amounts of data. The term has evolved to describe the process by which an organization manages its digital media

A DAM system is built upon a central repository that facilitates the storage, organization, retrieval, utilization and reuse of digital files. Such a system is a “filing cabinet” containing individual files that are stored with detailed information or metadata about a digital asset. Metadata can be wrapped around information as a sort of digital data container. The data container is a set of categories, such as creation data, creator, additional versions, related files, and copyrights. It exists in addition to or after the digital asset

Two types of digital media management system have evolved to address different customer and industry needs: DAM and content management (CM). The two differ mainly in their underlying data models. DAM systems are built upon a one-to-one relationship with a file or database object that is pushed into or pulled from the central repository. CM systems tend to be more robust and address a many-to-many relationship with database objects. CM systems also address other issues, such as file or object relationships, usage, and grouping of digital assets for publication.

* 1. **Classification**

This Asset Management System helps in creating and managing a data repository of the inventory pertaining to the hardware and software of IT resources in an organization.The IT inventory list basically consists of computers, monitors, software, network devices, printers, plotters, scanners, cartridges, etc. Asset Management, which has two major components, an application to maintain the inventory data pertaining to the installed hardware and software of a computer and its associated peripherals, and second component named which helps in the management of these inventories. The Second module makes use of the data captured by the First and the financial and commercial data pertaining to the inventory. The financial details include data on the invoice, warranty, AMC and the commercial details include data on suppliers, contacts, contracts etc. The financial and commercial data have to be entered manually into the System. The data captured could be imported into System; also manual entry of inventory data is possible in the Software.

* 1. **Modules**

The various modules implemented in Digital Management Systems are:

* + 1. **Administration**

This module is accessible to admin only. Admin can add all assets that are maintained in an organization. He will add company info first. Then will add branches and then related assets and all related information. Will have access to related modules to add the necessary info.

* + 1. **Asset Management**

This module is for adding different assets and its related info to the application. It will have grid view also to see the added info.  It will have add / edit and delete option for any asset held in an organization. It will have the option to add AMC info to and upload attachments related to the same for related assets. It will have the option to update the status of an asset like a computer if it is in repair wherever applicable.

**1.3.3 Registration**

Employees has to register themselves and their login will be approved to access the application. Employees can raise a complaint against hardware issues for the assets they are using.

* + 1. **Reports**

Will be used by admin and management officials to see the status of assets used in the organization as a snapshot.

**1.3.5 Roles**

This module will be used to grant roles to different employees who have registered and access will be granted to them to different modules as per the assigned role.

The users include Administrator / Asset Manager / Employees.

**Chapter 2**

**Literature Review**

**2.1 General**

Traditionally, DAM was used mostly as an archival solution for storing and accessing photographic images for use in high-resolution printing applications. Today, corporations are realizing that DAM is much more than storage and are finding new uses outside the printing and publishing industries. DAM is being used for such applications as remote access, security (retinal scanning, fingerprints), employee photos, and cancelled checks for online banking customers, just to mention a few. Corporate customers also are realizing the necessity of integrating DAM systems with enterprise resource planning (ERP) systems as part of their core strategy and allowing remote access via an Internet browser. Most systems can archive and catalog graphic files in virtually any format, as well as audio and video files. Production users can perform archive searches using simple database queries that can be as broad or specific as needed. DAM systems also provide thumbnail views of images.

**2.2 Enterprise DAM Systems**

Enterprise systems are typically more robust than mid-range or workgroup systems. This type of system comprises a centralized repository or database for the entire organization and handles all of the organization’s digital assets. The information is accessible via a Web browser. With this system, the reuse and repurposing of assets becomes more efficient and much less time-consuming. Enterprise systems are becoming workflow solutions. The key is to share integrated workflow environments, creative spaces, and back-office applications. Enterprise systems require high-performance servers, fast networks, and significant online storage capacity. They also must assure 24x7 operation and be available to the entire organization. Some enterprise system vendors are Artesia, Banta, and Canto.

**2.3 Pay-As-You-Go DAM Systems**

Pay-as-you-go or application service provider (ASP)/outsourcing systems provide users with access to current technology, all system maintenance, and security—all with a lower cost model. The ASP customer pays a monthly maintenance fee and the ASP vendor provides the hardware, software, system maintenance, and security along with sufficient staff and expertise to operate the system. The ASP vendor may also contract for functional training needs. Some vendors of ASP solutions are AGT, Banta, Lazer, and WAM!Base. After showing early promise, ASP models have not caught on as analysts predicted. The attitudes of customers towards outsourcing varies depending on the size of the organization, internal technical expertise, and the corporate mind-set. Large customers, in particular, prefer to have control of their own digital assets and resist adoption of an ASP model.

**2.4 Mid-Range DAM Systems**

Mid-range DAM systems can be deployed across wide area networks (WANs) and utilize client-server technology. They manage all assets used in production jobs, including layouts, images, and fonts, and store files on the server accessible only via the DAM system. They provide limited security and version control using check-in/check-out procedures. All assets of a job are stored with links to each other for simple retrieval. Some vendors of these solutions are Canto, Extensis, and Imation.

Many high-end (enterprise) vendors utilize mid-range systems to give their customers online access to digital assets without giving them access to all of the enterprise system capabilities.

**2.5** **Desktop or Workgroup DAM Systems**

These systems reside at the desktop or workgroup level. They typically organize small collections of assets and provide multiple ways to access them. These systems sometimes are referred to as media catalogs and run on Mac as well as PC platforms. Some vendors at this level are Adobe, Canto, Extensis, Imation, and Jintek (Fraser, 2002). Categories According to Market Focus Another way to categorize DAM systems is by their market focus. The major markets are: publishing/brand management, broadcast/media/entertainment, and animation. The systems differ in functionality and focus, depending on the market they serve.

**Chapter 3**

**System Environment**

**3.1 Hardware Configuration**

1.Pentium IV Processor

2. 512 MB RAM

3. 40GB HDD

4. 1024 \* 768 Resolution Color Monitor

**3.2 System Configuration**

1.OS: Windows 7 and Above

2.XAMP-VM (PHP, MySQL, Apache and PHPMyAdmin)

**3.3 Software Features**

XAMP-VM installs a complete working PHP/MySQL server environment on Windows platforms. Installs PHP, MySQL, Apache, and PHPMyAdmin.

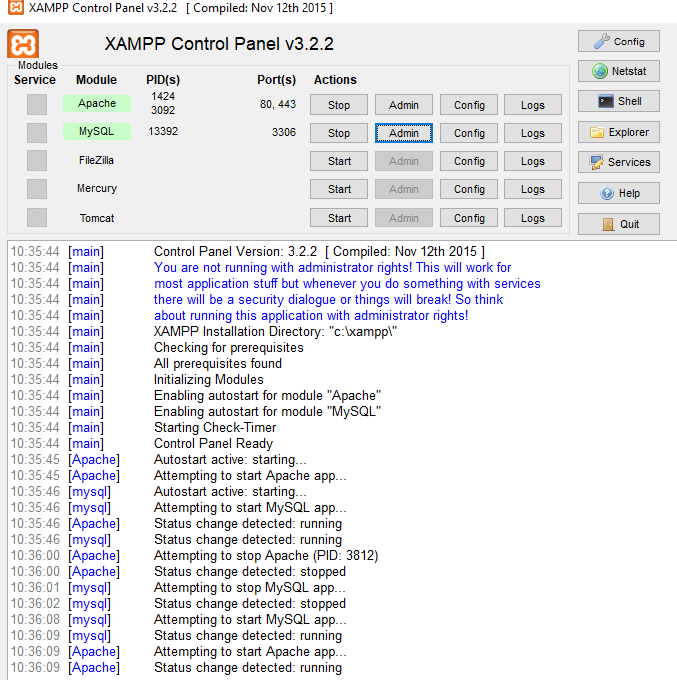


Figure 3.1

**3.3.1 PHP**

**PHP** designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications. While PHP was originally created by Rasmus Lerdorf in 1995, the main implementation of PHP is now produced by **ThePHP Group**

and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License, is a scripting language originally however it is in the GNU General Public License. It is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can be deployed on most web servers and on almost every operating system and platform free of charge. PHP is installed on more than 20 million websites and 1 million web servers.

**3.3.2 Security**

The National Vulnerability Database stores all vulnerabilities found in computer software. The overall proportion of PHP-related vulnerabilities on the database amounted to: 12% in 2003, 20% in 2004, 28% in 2005, 43% in 2006, 36% in 2007, and 35% in 2008. Most of these PHP-related vulnerabilities can be exploited remotely: they allow hackers to steal or destroy data from data sources linked to the web server (such as an SQL database), send spam or contribute to DOS attacks using malware, which itself can be installed on the vulnerable servers. These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent. Recognizing that programmers cannot be trusted, some languages include taint checking to detect automatically the lack of input validation which induces many issues. However, such a feature is being developed for PHP Hosting PHP applications on a server requires a careful and constant attention to deal with these security risks. There are advanced protection patches such as Suhosin and Hardening-Patch, especially designed for web hosting environments. Installing PHP as a CGI binary rather than as an Apache module is the preferred method for added security. With respect to securing the code itself, PHP code can be obfuscated to make it difficult to read while remaining functional.

**3.3.3 Syntax**

<html>

<head>

<title>PHP Test </title>

</head>

<body>

<?php

echo “<p> Hello World </p>”;

?>

</body>

</html>

PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not parsed by PHP. The most common delimiters are <?php and ?>, which are open and close delimiters respectively. <script language=”php”> and </script> delimiters are also available. Short tags can be used to start PHP code, <? or <?= (which is used to echo back a string or variable) and the tag to end PHP code, ?>.These tags are commonly used, but like ASP-style tags (<% or <%= and %>), they are less portable as they can be disabled in the PHP configuration. For this reason, the use of short tags and ASP-style tags is discouraged. The purpose of these delimiters is to separate PHP code from non-PHP code, including HTML. Variables are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted (“”) and here docstrings allow the ability to embed a variable’s value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: /\* \*/ serves as block comments, and // as well as # are used for inline comments. The echo statement is one of several facilities PHP provides to output text (e.g. to a web browser).In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. If conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

**3.3.4 MySQL**

A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents. MySQL is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQLAB now a subsidiary of Sun Microsystem, which holds the copyright to most of the codebase. The project’s source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is a database. The data in MySQL is stored in database objects called tables. A table is a collections of related data entries and it consists of columns and rows. Databases are useful when storing information categorically. A company may have a database with the following tables: “Employees”, “Products”, “Customers” and “Orders”.

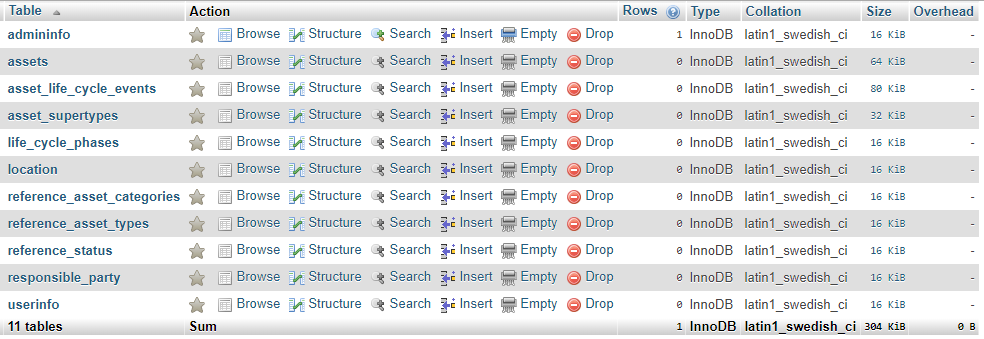


Figure 3.2

**3.3.5 PHPMyAdmin**

phpMyAdmin is an open source tool written in PHP intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage users and permissions, and manage keys on fields. while you still have the ability to directly execute any SQL statement. phpMyAdmin can manage a whole MySQL server (needs a super-user) as well as a single database. To accomplish the latter you’ll need a properly set up MySQL user who can read/write only the desired database. It’s up to you to look up the appropriate part in the MySQL manual.

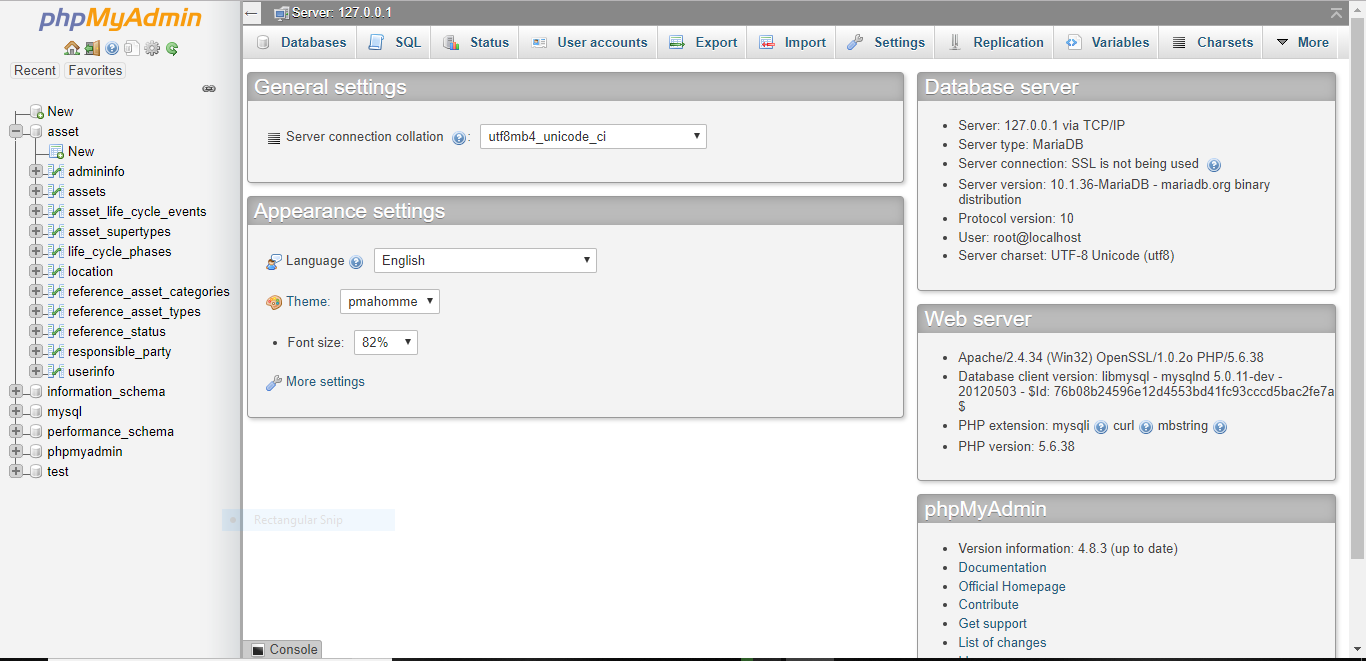
****

Figure 3.3

phpMyAdmin can:

·browse and drop databases, tables, views, fields and indexes

·create, copy, drop, rename and alter databases, tables, fields and indexes

·maintenance server, databases and tables, with proposals on server configuration

·execute, edit and bookmark any SQL-statement, even batch-queries

·load text files into tables

·create and read dumps of tables

·export data to various formats: CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel and LATEX formats

·administer multiple servers

·manage MySQL users and privileges

·check referential integrity in MyISAM tables

·using Query-by-example (QBE), create complex queries automatically connecting required tables

·create PDF graphics of your Database layout

·search globally in a database or a subset of it

·transform stored data into any format using a set of predefined functions, like displaying BLOB- data as image or download-link

·support InnoDB tables and foreign keys

·support mysqli, the improved MySQL extension

**3.3.6 Apache Web server**

Often referred to as simply Apache, a public-domain open source Web server developed by a loosely-knit group of programmers. The first version of Apache, based on the NCSA httpd Web server, was developed in 1995.Core development of the Apache Web server is performed by a group of about 20 volunteer programmers called the Apache Group.

However, because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. In many respects, development of Apache is similar to development of the Linux operating system. The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms. The name is a tribute to the Native American Apache Indian tribe, a tribe well known for its endurance and skill in warfare. A common misunderstanding is that it was called Apache because it was developed from existing NCSA code plus various patches, hence the name a patchy server, or Apache server. Apache consistently rates as the world’s most popular Web server according to analyst surveys. Apache has attracted so much interest because it is full-featured, reliable, and free. Originally developed for UNIX™ operating systems, Apache has been updated to run on Windows, OS/2, and other platforms. One aspect of Apache that some site administrators find confusing — especially those unfamiliar with UNIX-style software — is its configuration scheme. Instead of using a point-and-click graphic user interface (GUI) or Windows Registry keys as most other modern software packages, Apache generally relies on simple text files for its configuration settings.

**Chapter 4**

**System Design**

**4.1 Input Design**

The system design is divided into two portions. The Administrator section and the User(student’s)section.

**4.1.1 Administrator**

1. The Administrator can add or remove users.

2.He can update or modify the data.

3.He can control the status of the database.

4.He can edit the details of the assets. He can change their type, edit and delete the asset records.

A process of converting user originated inputs to a computer-based format. Input design is an important part of development process since inaccurate input data are the most common cause of errors in data processing. Erroneous entries can be controlled by input design. It consists of developing specifications and procedures for entering data into a system and must be in simple format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. In input data design, we design the source document that capture the data and then select the media used to enter them into the computer.

There are two major approaches for entering data in to the computer. They are

·Menus.

·Dialog Boxes.

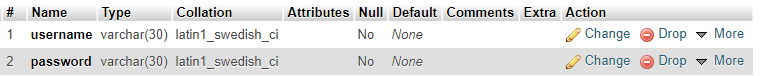


Figure 3.4

**Menus**

A menu is a selection list that simplifies computer data access or entry. Instead of remembering what, the user chooses from a list of options. A menu limits a user choice of response but reduce the chances for error in data entry.

**4.1.2 Users**

1.A user can only view contents of database.

2.A user can create an account and access only the tables that has been given permission to be viewed by the admin.

3. The user has read only permission.

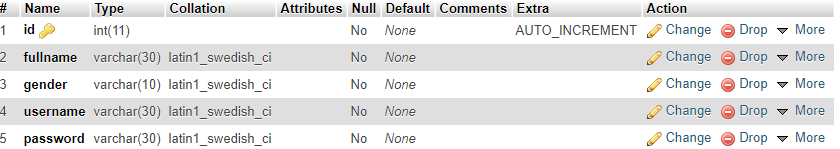


Figure 3.5

**4.2 Data Flow Design**

Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow Diagram and System Flow chart are the tools used for process design. System Flow Chart is a graphical representation of the system showing the overall flow of control in processing at the job level; specifies what activities must be done to convert from a physical to logical model. Data Flow Diagram is the logical representation of the data flow of the project. The DFD is drawn using various symbols. It has a source and a destination. The process is represented using circles and source and destination are represented using squares. The data flow is represented using arrows. One reader can easily get the idea about the project through Data Flow Diagram.

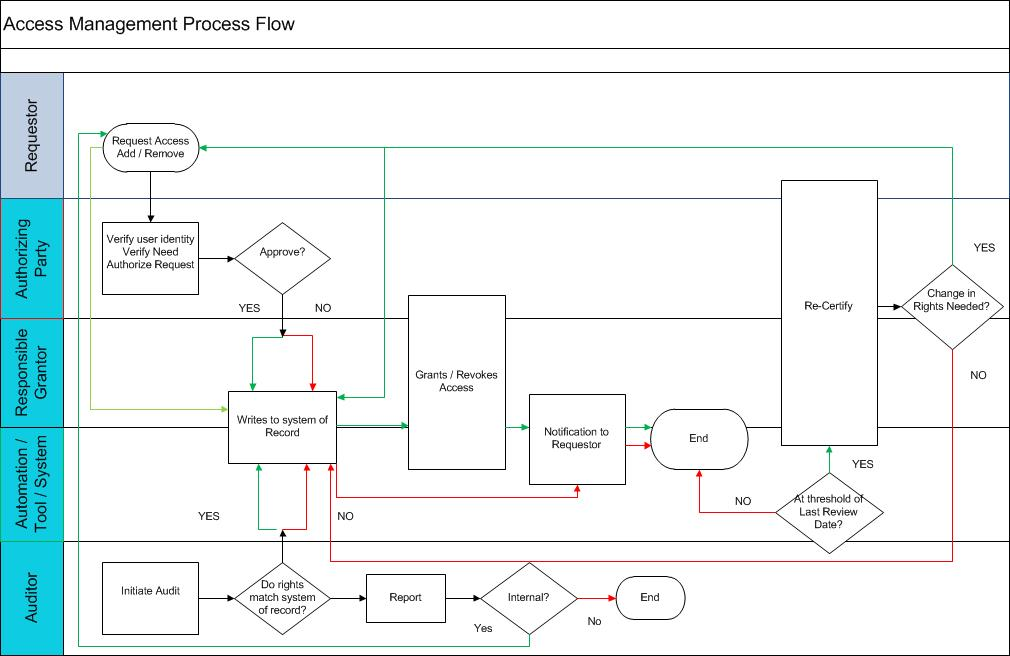


Figure 3.6

**4.3 Database Design**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

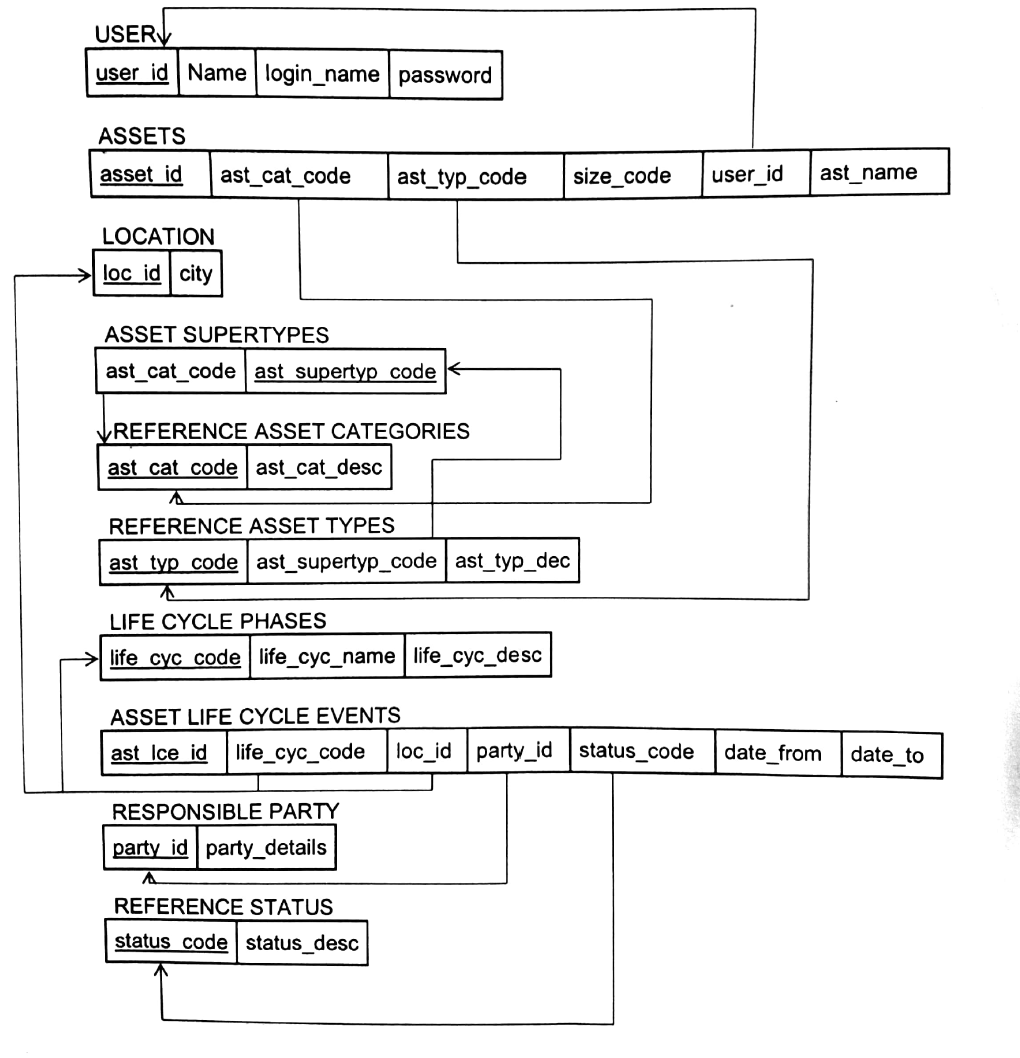


Figure 3.7

**4.4 Output Design**

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design an output we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output. Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the require output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned to the overall performance and the system’s working, as it should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, i.e. maximum user friendly. Proper messages and appropriate directions can control errors committed by users. The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system’s working, as it should. Output design consists of displaying specifications and procedures as data presentation. User never left with the confusion as to what is happening without appropriate error and acknowledges message being received. Even an unknown person can operate the system without knowing anything about the system.

**Chapter 5**

**Conclusion**

**5.1 General Conclusion**

To conclude the description about the project: The project, developed using PHP and MySQL is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. The expanded functionality of today’s software requires an appropriate approach towards software development. This asset management software is designed for people who want to manage various activities in a digital store. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

|  |  |
| --- | --- |
|  |  |

**References**

[1] Bowen, S. (2002, September/October). What (exactly) is DAM? . AIIM E – Doc Magazine, 16(5), 22

[2] Meserve, J. (2003, November 10). Digital asset management becomes a reality.

[3] Smith, D. (2000, December 1). Digital asset management Folio, 29(15), 67.

[4] Beyer, P. (2002, June). Digital content vs. digital assets 17(3), 34-37.

[5] Fraser, B. (2002, July). Digital-asset managers. Macworld, 19(7), 28-29