**DESIGN AND IMPLEMENTATION OF ASSETS MANAGEMENT SYSTEM**

**For the Application, Visit** [**https://softwarehub.ng/softwares/asset.zip**](https://softwarehub.ng/softwares/asset.zip)

**ABSTRACT**

Asset management is crucial in order for it to realize its goals and objectives. There are challenges facing many company such as uncoordinated asset declaration processes that lead to duplication of records and data redundancy, delays in serving the company, too much paper work that consumes a lot of space, poor decision making on restocking assets declared, and inability to track assets already declared. The goal of this study is to address the above problems by designing and implementing a prototype Asset management system capable of asset registration, track asset declared, and asset stock control.

There is a need to computerize the asset declaration processes for companies in order to minimize corruption to the lowest level, realize that companies meet with their goals, control and monitor assets, keep track of assets holders and their assets.

The research achieved almost all the objectives set and the prototype designed could further be developed into a full working system that could be used by the company or any other business in their asset management. The strengths, weaknesses, opportunities and threats identified in this study could also be used to build on the company’s competence, boost its presence in the market, attract more customers, and thus realize the company goals and objectives.

**CHAPTER ONE**

**INTRODUCTION**

Asset management is defined as the systematic and coordinated activities and practices through which an organization optimally and sustainably manages its assets and asset systems, their associated performance, risks and expenditures over their life cycles for the purpose of achieving its organizational strategic plan where an organizational strategic plan is defined as the overall long-term plan for the organization that is derived from, and embodies, its vision, mission, values, business policies, stakeholder requirements, objectives and the management of its risks. Effective implementation of asset management requires a disciplined approach which enables an organization to maximize value and deliver its strategic objectives through managing its assets over their whole life cycles. This includes determination of appropriate assets to acquire or create in the first place, how best to operate and maintain them, and the adoption of optimal renewal, decommissioning and/or disposal options.

Asset management provides a set of principles that guide an agency in improving how it conducts business, how it reaches decisions, and how it processes, uses, and communicates information related to the management of its infrastructure. At its core, asset management focuses on an agency’s allocation and utilization of resources – funding, people and skills, and information. It provides an integrated framework that establishes common approaches across asset classes in planning, program development, and program delivery. It encourages a number of best practices in these processes: e.g., consideration of the full range of alternatives at each stage of decision, adoption of a long-term view in economic analysis of projects, evaluation of tradeoffs across programs, monitoring of program delivery and system performance, and effective use of management and information systems throughout the infrastructure management cycle. Asset management is results oriented, driven by policy goals and objectives with clear measures of system performance and accountability (Marcow & Racosky, 2001).

To effectively manage asset, there are specialized software systems that keep proper record of relevant information pertaining the different assets owned by the company. Database applications are designed to capture relevant information pertaining the assets such as: type of asset, value of asset, maintenance cost, date of last maintenance, state of asset, etc and this information can be updated whenever any activity is performed with a given asset. This has made the concept of asset management systems a reality.

**1.1 Background of the Study**

With the ever increasing applications of computer programmers, organizations are now able to acquire systems to enable them manage their assets. An integrated asset management system is vital for organizations that are heavily dependent upon physical assets in the creation or delivery of their services or products. Large numbers of assets, or diversity characteristics of assets and asset systems, particularly in an environment of conflicting stakeholder expectations, further increase the importance of having a systematic approach to managing the asset portfolio. There are different levels at which asset units can be identified and managed – ranging from discrete equipment items or components to complex functional systems, networks, sites or diverse portfolios. Many organizations identify assets as equipment units (sometimes referred to as “maintenance significant items” – the unit at which maintenance tasks or work orders are directed), whereas others use the term to describe functional systems or even integrated business units. It does not matter at what such level an asset unit is identified, provided that:

The organization’s goals and strategic priorities are directly reflected in the asset management plan(s);

The asset life cycle costs, risks and performance are considered and optimized. (This will usually require definition of clear asset boundaries for measuring performance, life cycle expenditures and attributing associated risks.);

The aggregations of assets (through integrated asset systems) and contributions of value (as part of the organization’s portfolio) are managed in a coordinated and consistent manner;

All parts of the organization understand and use the same terminology in relation to the assets, their components and their asset system groupings or aggregations.

**1.2 Statement of the Problem**

Many organizations are not properly able to review the appropriateness of their asset management strategy in the light of changes in the operating, regulatory and financial environment. They lack the necessary asset management policy, strategy and plan to ensure that they manage their assets in a sustainable way. More importantly, they do not know what (existing) assets they have, where they are, what condition they are in, what function they perform and their contribution to value. The absence of an asset management system is the reason why the management of most organizations are unable to properly account for their assets.

**1.3 Aim and Objectives of the Study**

The aim of the study is to develop an asset management system. The objectives of the study are:

To create a database application that can be used to keep record of assets information.

To design a system that can be used to update information pertaining to any asset that has been registered.

To capture relevant information that will help in monitoring the condition of assets.

**1.4 Significance of the Study**

The significance of the study is that it will enable the management of Arsan Water Company to properly manage information pertaining to their assets. It will enable them to present needed reports pertaining to assets which is a very vital aspect of accountability in every organization. The study will also serve as a useful reference material for other researchers seeking information on the subject.

**1.5 Scope of the Study**

This study covers design and implementation of assets management system and it is limited to physical assets such as machineries, transport vehicles, etc.

**1.6 Organization of the Research**

This research work is organized into five chapters. Chapter one is concerned with the introduction of the research study and it presents the preliminaries, theoretical background, statement of the problem, aim and objectives of the study, significance of the study, scope of the study, organization of the research and definition of terms.

Chapter two focuses on the literature review, the contributions of other scholars on the subject matter is discussed.

Chapter three is concerned with the system analysis and design. It analyzes the present system to identify the problems and provides information on the advantages and disadvantages of the proposed system. The system design is also presented in this chapter.

Chapter four presents the system implementation and documentation. The choice of programming language, analysis of modules, choice of programming language and system requirements for implementation.

Chapter five focuses on the summary, conclusion and recommendations are provided in this chapter based on the study carried out.

**1.7 Definition of Terms**

Asset(s): plant, machinery, property, buildings, vehicles and other items that have a distinct value to the organization

Asset management: systematic and coordinated activities and practices through which an organization optimally and sustainably manages its assets and asset systems, their associated performance, risks and expenditures over their life cycles for the purpose of achieving its organizational strategic plan

Asset management information: meaningful data relating to assets and asset management

Asset management information system: system for the storage, processing and transmission of asset management information

Asset management plan: Document specifying activities and resources, responsibilities and timescales for implementing the asset management

strategy and delivering the asset management objectives

Asset management policy: principles and mandated requirements derived from, and consistent with, the organizational strategic plan, providing a framework for the development and implementation of the asset management strategy and the setting of the asset management objectives

Asset management strategy: long-term optimized approach to management of the assets, derived from, and consistent with, the organizational strategic plan and the asset management policy.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Asset**

**Asset** can be defined as anything which has value to an enterprise [8]. It can be furniture,documentation, hardware, software etc.

**2.2 Asset Classification**

Assets can be distinguished in two ways. First one is **Primary Assets** and a second one is

**Supporting Assets** [8].

Primary assets are the assets that are imported first. It acts as the default scope to import other assets. Primary assets can be classified into business process and information [9]. Business process basically means the core processes of the organization. If the processes modified, then it will greatly affect the goal of the organization [4]. A business process may be classified as:

* Development,
* Manufacturing,
* Query (read),
* Request (read and update),
* Transaction, and
* Alert, etc.

On the other hand Information is a vital part in the exercise of the organization’s mission. Information is required to achieve the final objective determined by the organization. Some important general properties of information may include:

* Receiving time of the information
* Information outsourcing
* Usability of the information
* Information recovery
* Information Source
* Transmission of information
* Size
* Format
* Encrypted or not
* Information storage
* backup

Information can be classified in many ways like Organizational records, Personal information, Strategic information, and Other valuable information  [[4]](#page52).

1. *Organizational records*

Records refer to documents that state the results achieved by an organization. They also provide evidence that business activities have been performed. Specifically, organizational records comprise of the following:

* + System Documentation
  + User manuals
  + Training materials
  + Audit, Logs and Trails
  + Archived information
  + Strategic, Tactical and Operational records
  + Publicly available information

1. *Personnel information*

Information related to employees, customers, and third parties belong to this category. These include data concerning privacy issues and non-disclosure agreements.

1. *Strategic information*

Strategy refers to formal expression of an organization’s guiding principles []. It determines the direction and development needed in order to benefit from the issues at stake, and of the major changes that an organization is planning. These include information pertaining to

* Contracts and agreements
* Operational and Support Procedure
* Business Continuity Plan
* Fallback arrangements

1. *Other valuable information*

In addition to the above types of information assets, the following are also important for an organization’s business processes:

* DB files, raw data files and Spreadsheet
* Research Information
* Source code
* Temporary Information
* Customer Data

Business process and information basically depend on each other. Using the relevant information only, a business process can be successfully exercised and a business process can also produce some valuable information to achieve the goal of the organization.

Supporting assets are just like secondary assets. They mainly support the primary assets. They are imported after primary assets. So, primary assets are treated as the superset of supporting assets [9]. Supporting assets can be hardware, software, network, site, personnel, and organizational structure.

Hardware Asset mainly means the physical elements of the supporting processes. It may be several kinds of equipments. Such as data processing equipment, transport equipments, fixed equipments, processing equipments, data medium, electronic medium, other medium. Hardware assets, with all of their sub-categories and components, are enumerated in Table 2.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Hardware** |  | **Hardware Name** | **Use** |  |
|  | **Sub-Category** |  |  |  |  |
|  | Data |  | Super-Computer, Mainframe, | These equipments are |  |
|  | Processing |  |  |
|  |  | Minicomputer, Server, | used on the |  |
|  | Equipment |  |  |
|  |  | Workstation | organization’s premises. |  |
|  | (Fixed) |  |  |
|  |  |  |  |  |
|  | Data |  |  | These equipments are |  |
|  | Processing |  |  |  |
|  |  | Mobile computers | used as portable |  |
|  | Equipment |  |  |
|  |  |  | equipments. |  |
|  | (Portable) |  |  |  |
|  |  |  |  |  |
|  |  |  | Semiconductor (Flash | It can be connected to a |  |
|  |  |  | memory), Magnetic (Floppy, |  |
|  |  |  | computer or network for |  |
|  | Data Medium |  | Removable hard disc, |  |
|  |  | data storage purpose. |  |
|  | (Electronic) |  | Magnetic Tape, Back-up |  |
|  |  | They can be used with |  |
|  |  |  | cartridge), Optical (CD-ROM, |  |
|  |  |  | standard equipments. |  |
|  |  |  | DVD, Blue Ray) |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Hardware** |  |  | **Hardware Name** | **Use** |  |
|  | **Sub-Category** | |  |  |  |  |
|  | Data Medium |  |  | Paper, Slide, Transparency, | It can be used for data |  |
|  | (Others) |  |  | Documentation, Fax | storage purpose. |  |
|  |  |  |  |  | These are the equipments |  |
|  | Peripheral |  |  |  | that connected to a |  |
|  |  |  |  | computer via some cable |  |
|  | Equipment |  |  | Input (), Output (), Storage () |  |
|  | (Fixed) |  |  |  | or port for transmitting |  |
|  |  |  |  |  | data. |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  | These are the equipments |  |
|  | Peripheral |  |  |  | that connected to a |  |
|  |  |  |  | computer via some cable |  |
|  | Equipment |  |  | Input (), Output (), Storage () |  |
|  | (Portable) |  |  |  | or port for transmitting |  |
|  |  |  |  |  | data. |  |
|  |  |  | |  |  |  |
|  |  | **Table 2.1 Classification of Hardware Assets** | | | |  |

Software asset consists of the programmes used to do some operation of a data processing set. Software asset can also be classified in many ways like Operating System, Service, maintenance or administration software, Package software or standard software. Details of all types of software assets are given in Table 2.2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Sub-Category** | | | **Software Name** | |  | **Use** | |  |  |
|  | | |  | |  |  |  | |  |
| System S/w (Operating | | | Server OS, Desktop OS, | | These | are | the | | main |
| System) |  |  | Embedded | OS, | operational | | programmes | | |
|  |  |  | Firmware |  | that are used to run other | | | | |
|  |  |  |  |  | programmes. | | |  |  |
| System | S/w | (Language | Interpreter, Compiler | | These are used for high | | | | |
| Translator) | |  |  |  | level language | | | to | low |
|  |  |  |  |  | level |  |  | language | |
|  |  |  |  |  | transformation. | | |  |  |
| Application | | S/w | Asset management s/w | | These are usually used to | | | | |
| (Service, | Maintenance, | |  |  | perform | some | | specific | |
| Administrative software) | | |  |  | operation of the system. | | | | |
| Application | | S/w | DBMS, | Electronic | These | are | | common | |
| (Packaged s/w, Standard | | | messaging | s/w, | products | classified | | | such |
| s/w) |  |  | Groupware, | Directory | as released, medium and | | | | |
|  |  |  | s/w, Web server s/w | | maintenance | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Sub-Category** | | **Software Name** | | |  |  | **Use** | |  |
|  |  |  | | | |  |  |  |  |
| Application | S/w | Accounts s/w, Machine | | | | These | are | used | by |
| (Standard | business | tool | control | | s/w, | business |  | users | to |
| applications) |  | Customer | | care | s/w, | perform various business | | | |
|  |  | Personnel | | competency | | functions. | |  |  |
|  |  | management s/w, Admin | | | |  |  |  |  |
|  |  | s/w |  |  |  |  |  |  |  |
| Application | S/w | Invoice | management of | | | These | are | used | by |
| (Specific | business | telecom |  | operators’ | | business |  | users | to |
| applications) |  | customers, | | Real | time | perform various business | | | |
|  |  | monitoring | | application | | functions. | |  |  |
|  |  | for rocket launching | | |  |  |  |  |  |

**Table 2.2 Classification of Software Assets**

Network asset basically consists of all communication devices. These devices are used to interconnect several remote computers or some other elements of an information system. Network assets can be classified either in physical or logical terms. Physically, there are three distinct categories into which these assets belong:

* Medium and Support,
* Active or Passive Relay, and
* Communication Interface.

The types of assets in each of these categories are listed in Table 3.

Logical categorization of network assets is done by grouping them into Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Network (WAN), and Internet.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Network Asset Sub-** | **Network Asset Name** |  | **Use** |  |  |
| **Category** |  |  |  |  |  |
| Medium and Support | PSTN, Gigabit Ethernet, | These are mainly | | | used |
|  | ADSL, Internet, Wi-Fi | for | communication | | |
|  | 802.11, Bluetooth, Voice | purpose. | These | | are |
|  | over IP service | characterized | | by | the |
|  |  | physical | and | technical | |
|  |  | characteristics | | and | some |
|  |  | communication | |  |  |
|  |  | protocols. |  |  |  |
| Active or Passive Relay | Router, Bridge, Hub, | These | are | | the |
|  | Switch, Wireless Access | intermediate | | or | relay |
|  | Point, Transceivers | devices. | They include | | |
|  |  | some | routing | | and |
|  |  | filtering | functions | | and |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Network Asset Sub-** | **Network Asset Name** | | |  | **Use** |  |  |
| **Category** |  |  |  |  |  |  |  |
|  |  |  |  | services. | They | can | be |
|  |  |  |  | administrated | | remotely | |
|  |  |  |  | and are also capable of | | | |
|  |  |  |  | generating logs. | |  |  |
| Communication | GPRS, |  | Network | These are | connected | | to |
| Interface | Interface | Card | (NIC), | the processing units. But, | | | |
|  | LAN Adapter, | | ISDN | they are characterized by | | | |
|  | terminal | adapters, | | the media and supported | | | |
|  | Dongle |  |  | protocols, | by | | the |
|  |  |  |  | installed filtering, log | | | or |
|  |  |  |  | warning | generation | | |
|  |  |  |  | functions. |  |  |  |

**Table 2.3 Classification of Network Assets**

All the above assets are basically IT assets. There are also some different kind of assets are present like personnel, site and organization.

Personnel type consists of all the groups of people involved in the information system. Decision maker, users, operation/maintenance staff, developers all are the part of personnel [4].

The site type comprises all the places containing the scope or part of the scope and the physical means required for it to operate [4].

Organizational structure mainly consists of various branches of the organization including cross functional activities. It generally goes under the control of its management.

**Asset Life Cycle**

Every organization has a significant amount of asset; the management of these assets is also an important task. Through asset management, we can control the asset, determine the true values of assets and also have some idea for improvements of the organization by decreasing the loss.

**Asset Management** basically means the comprehensive management of asset demand,planning, procurement, operation, maintenance, rehabilitation, disposal and replacement. It is used to maximize the return on investment at the required standard of service to current and future generations.

Broadly, asset management refers to any system where the values of an entity or group are monitored, maintained, upgrade and also disposed in a systematic cost effective way. Basically by the asset management process, the built systems of facilities are monitored and maintained. It has some objective and it also provides the best possible service to the user. So, this type of management maintains these IT assets with respect to control, risk and cost.

**IT asset management** generally means the set of business practices that join financial,contractual and inventory functions to support life cycle management and strategic decision making for the IT environment [13].

According to ISO/IEC 19770 some basic processes are there to manage the IT asset in a systematic way. They are **Organizational Management Process, Core Process and** **Primary Process Interfaces** [1].

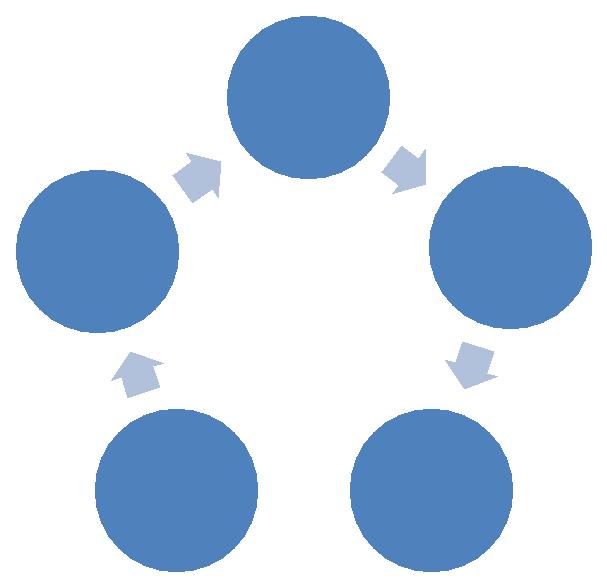
**Organizational Management Process** mainly establishes and maintains the system withsome policies and procedures. In this process the roles and responsibilities for related assets are clearly defined.

**Core Process** is the main inventory process for asset management. It has mainly three parts.They are: **Asset Identification, Asset Inventory Management and Asset Control**.

Firstly, the Asset identification is mainly used to select and grouped the necessary classes of assets. So, the items to be managed are chosen using some selection criteria. Next, the Asset

11

Inventory Management Process ensures that the physical instances of the assets are properly stored. So, it is very useful to protect the assets from some unauthorized access, change and corruption. It also provides the effective role for asset recovery. Then the Asset Control Process provides the control mechanism over the assets and changes to software and related assets while maintaining a record of changes to the status and approvals. So, some policies and procedures are also developed, approved and issued before the release of software. After that, the Primary Process Interfaces are generally consists of some life cycle processes that are required to maintain and manage several types of assets of a system.



Procurement

Disposal Deploy

Support  Maintain

**Figure 4.1 General Asset Life Cycle**

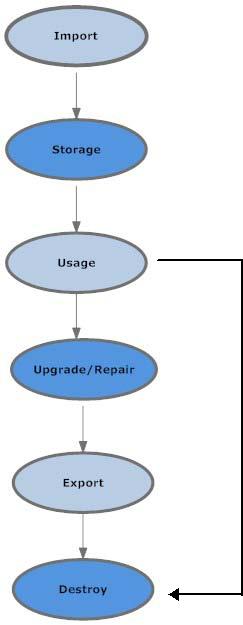
In this tool, all the hardware, software, network and information asset follow the asset life cycle significantly. To achieve the life cycle phases, this asset management tool follows some operations. User can insert the asset information which signifies the procurement phase of asset life cycle. Then the inserted asset information is stored according to its category in a suitable asset table. An organized database is working behind the tool to store and manage all the asset information. Many database tables are maintained in this database to handle the four categories of asset. Then the asset can persist in the system in active or inactive state s per the requirement. Some specific data attributes are designed to store the vital information on each asset. The asset can be upgraded or update with the progress of time. So, the modification phase is also present in the tool. Some assets are performing in the system for a particular period of time. So, after their life period, they need to remove from the system. To perform those tasks, uninstall and delete asset information detail operations are also present in this tool. In this way, this newly developed asset management tool can manage the entire

asset information of any enterprise following the asset life cycle. Here, the asset life cycle of each category of asset is described below-

**2.3 Hardware Asset Life Cycle**

Here, the hardware life cycle model has been designed. The main phases of this cycle are Import, Storage, Usage, Upgrade/Repair, Export and Destroy. In this case, a product of hardware will be imported. Then it will be stored in a safe and appropriate area. After that it will be used for a specific purpose, then it will be upgraded or repaired (if possible), then it will be exported and after passing through all phases it will be destroyed. In the newly introduced IT asset management tool, the entire hardware assets are following the hardware life cycle. All the hardware assets are having some relevant information to manage and maintain the hardware asset. This information is described in Annexure Table A.1.

Suppose a printer is manufactured. Then it will be connected to the set of computers and will also be used. If it will get damaged, then it may be repaired otherwise the damaged printer will be replaced.

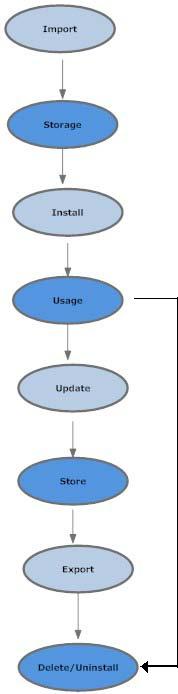


**Figure 4.2 Hardware Asset Life Cycle**

**2.4 Software Life Cycle**

Here, the software life cycle model has been designed. The main phases of this cycle are Import, Storage, Install, Usage, Update, Store, Export and Delete/Uninstall. In this case, firstly a software product will import. Then it will be stored in a safe area. After that the software product will be installed. After installation, the software asset will use for a specific task. Then it will be updated (if possible). After that this product will be exported and after passing through all these phases, it will be safely deleted or uninstalled. In the newly introduced IT asset management tool, the entire software assets are following the software life cycle. All the software assets are having some relevant information to manage and maintain the software asset. This information is described in Annexure Table A.2.

Suppose an asset management software is purchased. Then it will be installed in a computer and manage all the asset of the network Next, the asset management software will update if any updated version is available. Then the updated software will stored. After that, if the asset management software no more needed for the system, then it will be deleted from the computer.

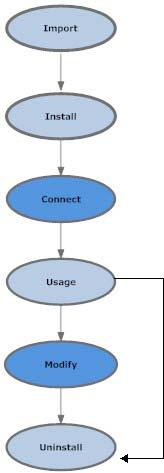


**Figure 4.3 Software Asset Life Cycle**

**2.5 Network Asset Life Cycle**

Here, the network life cycle model has been designed. The main phases of this cycle are Import, Install, Connect, Usage, Modify, and Uninstall. In this case, a network asset will import. Then it will be installed. It may consist of two or more nodes. Then the nodes of the network will be connected via media equipments. Then the network will be used for some purposes. At this time, it will also be modified as per the requirement. After passing through all these phases, finally, the network will be uninstalled. In the newly introduced IT asset management tool, the entire network assets are following the network life cycle. All the network assets are having some relevant information to manage and maintain the network asset. This information is described in Annexure Table A.3.

Suppose a router will purchase and install on a network. All PCs of the network will get connected to this router using LAN. Then the router will choose the best route for the data packet and deliver it to the computer. If any up gradation will available, then the router will upgraded. If the router will get faulty or not able to give proper service, then it will be uninstalled from the system.

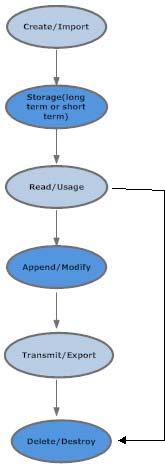


**Figure 4.4 Network Asset Life Cycle**

**2.6 Information Asset Life Cycle**

In information life cycle, some meaningful information will be created or import. Then the information will be stored for a long or short time in a proper safe area as per the requirement. Then the information will be read or be used. After that more information will be added to it or the existing information will be modified if necessary. Then, the information will be exported and after passing through all these phases, it will safely be deleted or be destroyed. In the newly introduced IT asset management tool, the entire information assets are following the information life cycle. All the information assets are having some relevant information to manage and maintain the information asset. This information is described in Annexure Table A4.

Suppose, a database file will be attached to the system. Then it will be used for some data management purpose. Any data modification can be done during this phase. After some time, if the database files will no more needed, it will be deleted from the system.



**Figure 4.5 Information Asset Life Cycle**

***References:***

1. ISO – IEC 19770-1 Information technology — Software asset management - International standard. First Edition, May, 2006
2. ISO – IEC 27001:2008 Information technology – Security techniques – Security techniques- Information security management system- Requirements,

International standard, 2008

1. ISO/IEC 27002:2005, Information technology – Security techniques - Code of

practice for information security management. Edition 1. Switzerland.

1. ISO – IEC 27005:2008 Information technology – Security techniques – Security techniques- Information security management system- Requirements,

International standard, 2008

1. IEEE recommended practice for Software requirements specification.
2. Guidelines for Implementing Total Management Planning, Asset Management
3. J. Kreutzkamp, L. Hagge, E. Deffur, A. Gellrich, B. Schulz, Experience with an IT Asset Management System, 24-28 March 2003, La Jolla, California
4. Wilco van Ginkel, A Business Process Approach to Assets and Risks, Verizon Business, White Paper, Security Solutions
5. Primary and Secondary asset,  [http://www.symantec.com/business/support/index?page=content&id=HOWTO409](http://www.symantec.com/business/support/index?page=content&id=HOWTO40975)7  [5](http://www.symantec.com/business/support/index?page=content&id=HOWTO40975) , Accessed in November, 2012
6. IT Asset Life Cycle Management, Asset Explorer Software,  [http://www.manageengine.com/products/asset-explorer/it-asset-life-cycle](http://www.manageengine.com/products/asset-explorer/it-asset-life-cycle-management.html)- [management.htm](http://www.manageengine.com/products/asset-explorer/it-asset-life-cycle-management.html)l, Accessed in December, 2012
7. Asset Management Software  [http://www.sysaid.com/free-asset-management-software.ht](http://www.sysaid.com/free-asset-management-software.htm)m, Accessed in March, 2013
8. Asset Manager tool,

<http://www.quest.com/asset-manager/> , Accessed in April, 2103

1. Details of IT Asset Management,  [www.wikipedia.o**r**](http://www.wikipedia.org/)g, Accessed in October, 2012

**CHAPTER THREE**

**SYSTEM DESIGN METHODOLOGY**

**3.1.0 INTRODUCTION**

This chapter focuses on the system design methodology and covers areas such as requirement specification, which is aimed at stating the system requirements meant for the proposed system.

System design will also be looked at, it covers areas like activity diagram, class diagram and program specification. At the end of this chapter the structure of the database design will be shown.

**3.2.0 SYSTEM REQUIREMENT SPECIFICATION**

In order to design and implement an efficient system, there are some basic requirements that are crucial and must be considered; this system requirement includes the following:

* The system should have security access control that enforces users to sign in before accessing any function or feature of the system.
* For any member to access the system he/she should be dully registered as a staff on this platform.
* The system should have a staff management page through which the system admin can dully administrate the entire system process.
* The system should have a list of all asset store
* The system should provide an easy search and sort window for ease of use.
* The system should also allow users to maintain an online profile.

.

**3.3.0 SYSTEM DESIGN**

Designing a project management system entails translating the requirements specification into a physical form which requires using different patterns to realize the intended system.

**3.3.1.0 LOGICAL DESIGN**

The logical design converts the system requirements specification into system model, by implementing the major features of the system. This design provides a means through which a user can create a new project, add members to the project, develop project milestones and work through using the activities section.

User login

User login

Name:

Password:

Login

Home screen

Register

Users

Search

stocks

Software

Details

Hardware

Details

Menu Welcome to Asset management system

Dashboard

Assignment

Vendor list

Category list

Search

Logout

Menu

Add New User

Menu User management

Dashboard

Assignment

Vendor list

Category list

Search

Logout

User E-mail Laboratory Delete/Edit

- Delete/Edit

Search form

Search Now

V

V

Search form

Asset Name:

Asset type:

Vendor Name:

Category:

Start Date:

End Date:

Cancel

Add New Software (+)

V

V

Add New Software

Software:

Serial key:

Vendor Name:

Date of purchase:

Date of expiry:

Unit price:

Category:

Stock

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Following are the list od Avail | | | | | |
| Hardware | Qty/price | vendor | category | D.O.P | Delete |
| \_\_\_\_\_\_\_\_\_\_\_\_  ------  \_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_  \_\_\_\_\_\_  Add New  Hardware | Pic  Pic  pic | ---  ----  ---- | ------  --------  ------- | ------  --------  ------ |

**3.6 ACTIVITY DIAGRAM**

**LOGIN**

**CORRECT**

**RETURN TO LOGIN PAGE**

**SELECT ACTION**

**CLIENT REGISTRATION**

**VIEW PROJECT HISTORY**

**ADD TEAM MEMBERS**

**VERIFY STAFF DATA**

**CREATE AND MANAGE MILESTONES**

**MODIFY CLIENT**

**INFORMATION**

**CREATE PROJECT**

**EVALUATE PROJECTS   
TASK AND FILESHARING**

**ASYNCHRONOUS MESSAGES**

**REGISTER STAFF**

**LOGIN**

**LOGOUT**

**Fig 3.6**

**3.3.1.6 SYSTEM CONTROLS**

The input and output is controlled through the use of automatic validation of input password access to the system sessions are used to keep track of user and data from one page to another.

**3.3.2.0 STRUCTURE OF DATABASE**

The structure of relational database shows the different tables that make up the database and links among the fields, the database consists of seven tables which are:

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION**

* 1. **INTRODUCTION**

This chapter focuses on the implementation of the system. The features of the implementation languages used in this research- PHP and MYSQL will be discussed extensively. The system testing strategies, the target computer requirements as well as the software maintenance issues that would arise in the system would be discussed also.

* 1. **FEATURES OF IMPLEMENTATION LANGUAGES**

The programming languages used in the implementation of this project are PHP (Hypertext Preprocessor) and MYSQL programming languages. PHP is a general purpose server side scripting language originally designed for web development to produce dynamic web pages. It has also evolved to include a command line interface capability and can be used in stand-alone graphical applications.

The following features make PHP a preferred implementation language for this project:

1. PHP has its root in C and C++. PHP syntax is most similar to C and C++ language syntax, so programmers find it easy to learn and manipulate.
2. PHP can run on both UNIX and windows. Hence it is compatible across various operating systems.
3. PHP has powerful output buffering that further increases over the output flow. PHP internally rearranges the buffer so that the header comes before the content.
4. PHP is platform independent: this is because it is parsed by the web browser hence compatibility issues do not arise when code written in PHP is ported to a different platform.
5. PHP can be used with a large number of relational database management systems, runs on all of the most popular web servers and is available to many different operating systems.
6. PHP is fully an object oriented programming language and its platform independence and speed on LINUX servers help to build large and complex web applications.
7. PHP has also attracted the development of many frameworks that provide building blocks and design structure to promote Rapid Application Development (RAD). Some of these include cake PHP, code igniter, Yii framework and Zend framework.
8. PHP IDS add security to any PHP application to defend against intrusion. PHPIDS detects cross-site scripting (XSS), SQL injection, header injection, directory traversal, remote file execution, local file execution and Denial of Service (DOS).

MYSQL is a relational database management system written in C and C++, that runs as a server providing multi user access to a number of databases. MYSQL is used basically to create a relational database structure on a server in order to store data orautomate procedures. The following features make MYSQL also a preferred implementation language in this research:

1. MYSQL is written in C and C++ and tested with a broad range of different compilers. It also functions on different platforms.
2. It uses multi-layered server design with independent modules.
3. It is designed to be fully multi-threaded using kernel threads to easily use multiple CPUs if they are available.
4. It is a server/client system. The database server (MYSQL) and the arbitrary many clients (application programs) which communicates with the server to query data and save changes.
5. MYSQL is designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
6. It provides transactional and non-transactional storage engines, uses very fast B-tree disk tables with index compression and a fast thread-base memory allocation system.
7. It executes very fast joins using an optimized nested loop join; implements in-memory hash tables which are used as temporary tables.
8. It implements SQL functions using a highly optimized class library that should be as fast as possible.
9. It provides the server as a separate program for use in a client/server networked environment and as a library that can be embedded (linked) into stand-alone applications. Such applications can be used in isolation or in environments where no network is available.
   1. **SYSTEM TESTING STRATEGIES**

This section is concerned with testing and debugging of the programs and general processes involved in achieving the objectives of the system requirement. System testing is conducted on a complete integrated system to evaluate the system’s compliance with its specified requirements. System testing falls within the scope of black box testing and as such should require no knowledge of the inner design of the code or logic. During system testing, the focus is on the software design, behavior and even the believed expectations of the customer. So we can also refer to the system testing phase as investigatory testing phase of the software development life cycle. The system testing strategies used in this system include the unit test and integration test.

* + 1. **UNIT TEST**

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code and determine whether it behaves exactly as it is expected to behave. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

The most common approach to unit testing requires drivers and stubs to be written. The driver simulates a calling unit and the stub simulates a called unit. The investment of developer time in this activity sometimes results in demoting unit testing to a lower level of priority and that is almost always a mistake. Even though the drivers and stubs cost time money, unit testing provides some undeniable advantages. It allows for automation of the testing process, reduces difficulties in discovering errors contained in complex pieces of the application. During the unit testing of the application, errors uncovered by the researcher were rectified and the result was satisfactory.

* + 1. **INTEGRATION TESTING**

Integration testing is a logical extension of unit testing. In its simplest form, the units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combination of pieces and eventually expand the process to test your modules with those of other groups. Integration testing can be done in a variety of ways which include top-down approach, bottom-up approach and the umbrella approach.

In the integration testing of the software, satisfactory results were obtained from the test using the bottom-up approach.

* 1. **TARGET COMPUTER SYSTEM REQUIREMENTS**

This section considers the requirements that must be met by the target system to enable the developed software application function as required. The target computer system requirement will be discussed in the area of software and hardware requirements.

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| Operating system | Windows 2000, XP, Vista |
| Memory | 128MB or higher |
| Database | MySQL 5 |
| Web server | WAMP server |

**Table 4.1: software requirement for target computer system**

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| RAM | 256MB of RAM |
| Hard disk | 10GB of hard disk space |
| Processor | 333Hz or higher |

**Table 4.2: hardware requirements for target computer system**

* 1. **SOFTWARE MAINTENANCE ISSUES**

This section focuses on software maintenance issues. Software maintenance is the modification of a software product after delivery to correct faults, improve performance or other product attributes or to adapt the product to a new or changing environment. It also serves as an opportunity to improve the performance o the software to suit the needs of the users if it becomes necessary for the user requirements to be improved upon or changed.

Maintenance would be seen in three areas in this research; corrective maintenance, preventive maintenance and adaptive maintenance.

* + 1. **CORRECTIVE MAINTENANCE**

Corrective maintenance is a maintenance task performed to identify, isolate and rectify a fault so that the failed system can be restored to an operational condition within the tolerances or limits established for in-service operations. Necessary corrections in the form of removal, modification or addition of program modules should be permitted by the software to allow for optimal use of the application.

* + 1. **PREVENTIVE MAINTENANCE**

This is a schedule o planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of software before it actually occurs. It is designed to preserve and enhance software reliability by replacing error-prone components before they actually fail. Recent technological advances in tools for inspection and diagnosis have enabled more accurate and effective software maintenance. Measures like regular diagnosis, database backups, creating system mirrors preserve the integrity of information stored in the application. If these are strictly followed, limited instances of such occurrences would be noticed in the use of the software application.

* + 1. **ADAPTIVE MAINTENANCE**

This involves enhancing the system by adding features, capabilities and functions in response to new technology, upgrades, new requirements or new problems. Since the environment in which the application would be running is dynamic, it should be made to suit whatever requirements that may change in the long run.

**CHAPTER FVE**

**SUMMARY AND CONCLUSION**

1. **SUMMARY**

The aim of this research is to create an asset management system which will make asset management in Springlight organization more flexible and reliable. The existing types of asset management were discussed and it was narrowed down to IT asset management which is the main thrust of this project. The history of asset management where highlighted. Principles of asset management were outlined. The system design was made suitable to the system requirements. Embedded in the system design is activity diagram which shows the activities and actions in the system. The implementation of the software was successful.

* 1. **CONCLUSION**

Asset management System improves business activities in large, medium and small enterprise. Hence it is advised that all companies in Nigeria should adopt an asset management system which more reliable, flexible and result oriented.

The resulting software would be of benefit to individuals who which to manage his/her asset. The software has been able to meet its objectives and will make asset management more successful.

* 1. **RECOMMENDATION**

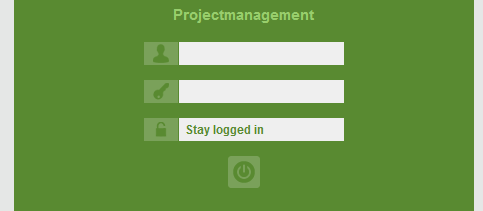
The software developed for the implementation of this research can be used by any asset management company other than Springlight Technology. The following recommendations are proposed:

* Admin should create passwords with long characters so as to make password hacking difficult.
* System users should ensure safekeeping of password since it provides access to the system.
* System client should login from time to time to see the progress on his or her work.

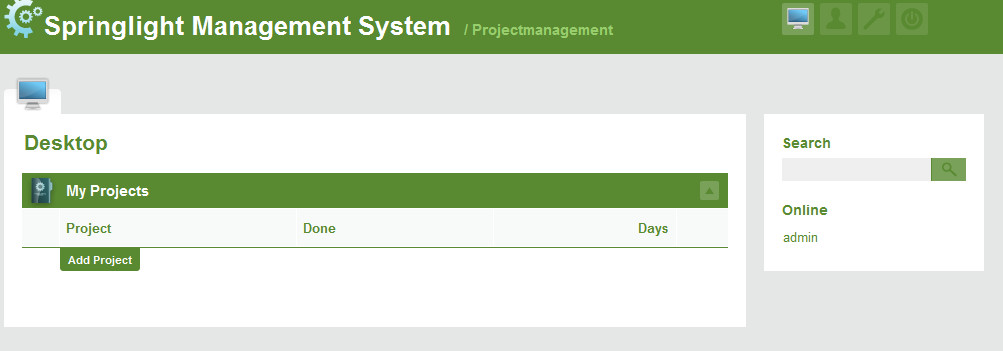
**APPENDIX**

**APPENDIX A: SCREEN WINDOW**

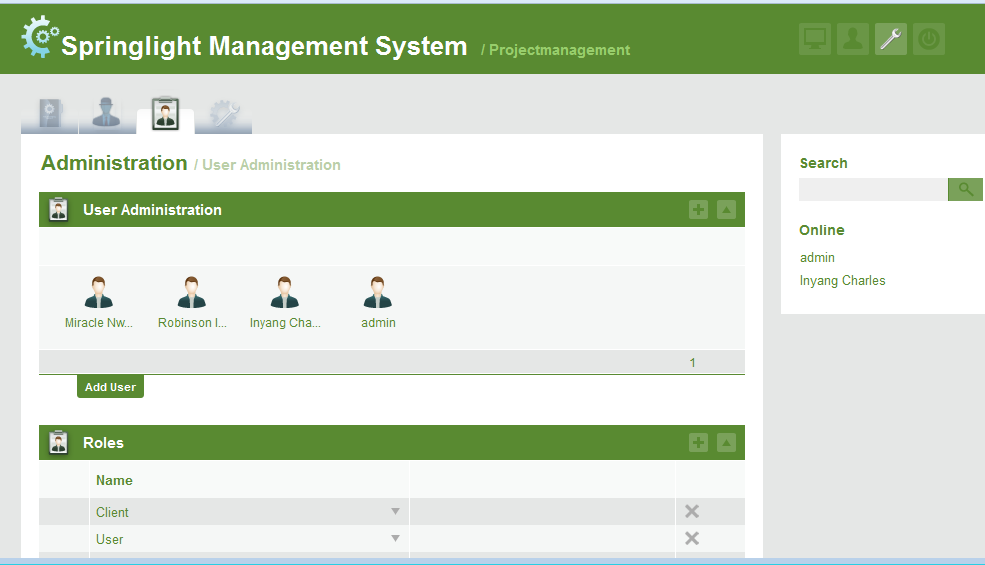
**LOGIN WINDOW**

****

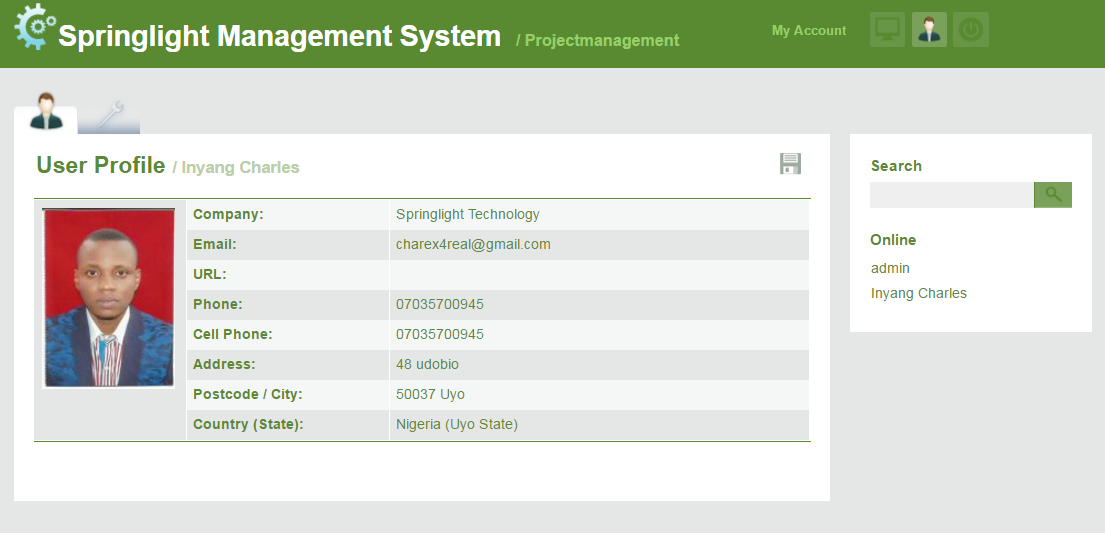
**ADMIN WINDOW**

****

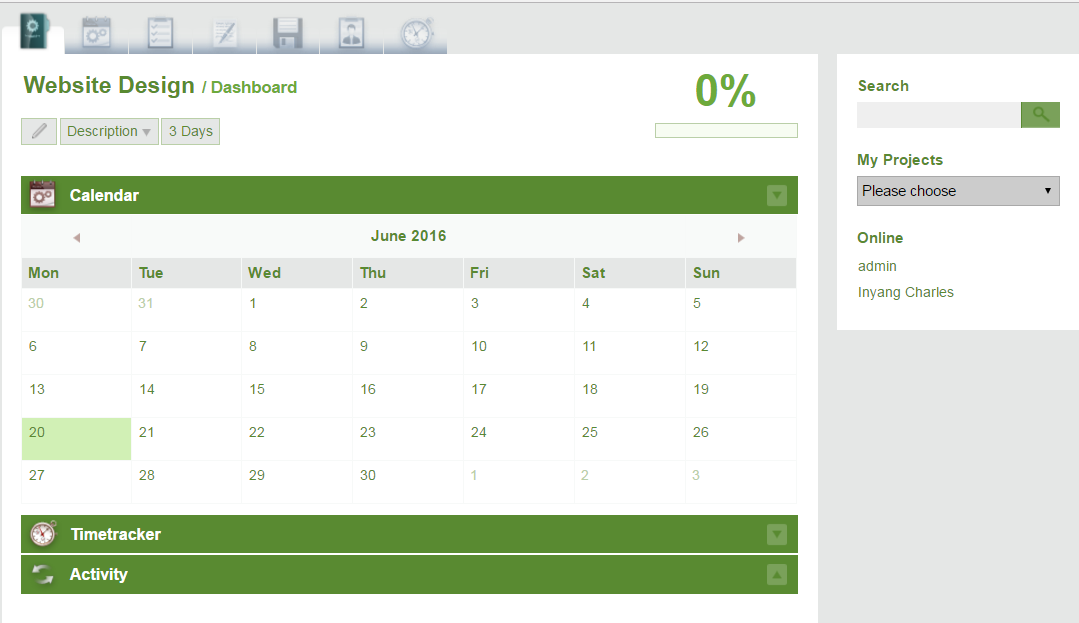
**USER ADMINISTRATION**

****

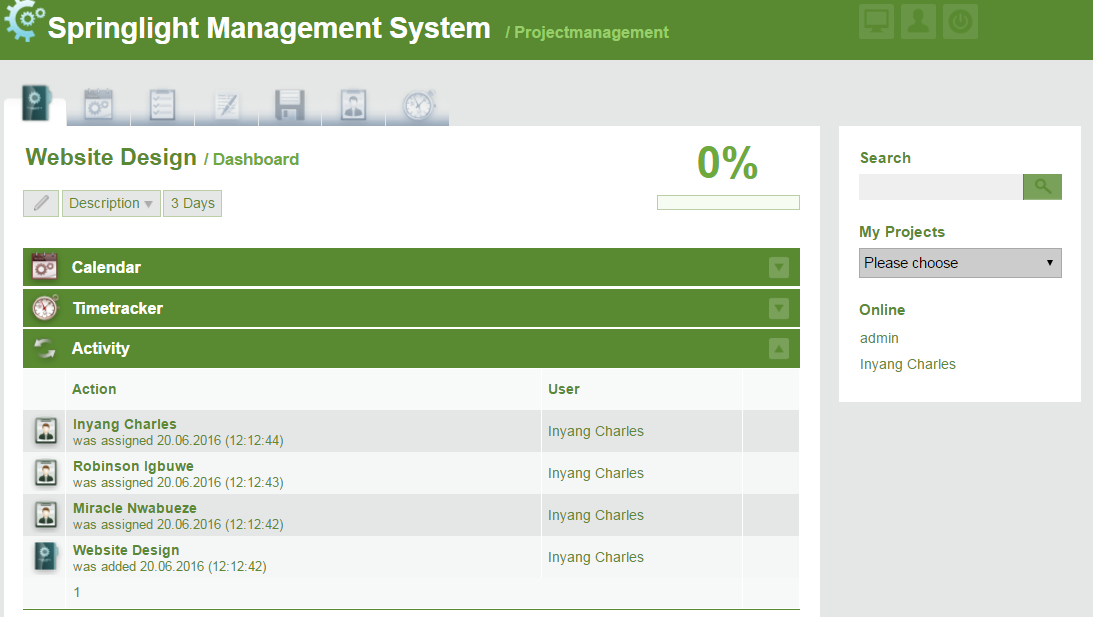
**PROFILE PIX**

****

**Project create view**

****

**PROJECT ACTIVITIES WINDOW**

****

**APPENDIX B: SOURCE CODE**

**INDEX.PHP**

<?php

require("./init.php");

//check if the user is loged in

if (!isset($\_SESSION["userid"])) {

$template->assign("loginerror", 0);

$mode = getArrayVal($\_GET, "mode");

$template->assign("mode", $mode);

$template->display("login.tpl");

die();

}

// collabtive doesn't seem to be installed properly , redirect to installer

if (empty($db\_name) or empty($db\_user)) {

if($db\_driver == "mysql")

{

$loc = $url . "install.php";

header("Location: " . $loc);

}

}

// Set the desktop icon in the top icon menue

$mainclasses = array("desktop" => "active",

"profil" => "",

"admin" => ""

);

$template->assign("mainclasses", $mainclasses);

// create objects

$project = new project();

$customer = new company();

$milestone = new milestone();

$mtask = new task();

$msg = new message();

//create arrays to hold data

$messages = array();

$milestones = array();

$tasks = array();

//create a counter for the foreach loop

$cou = 0;

// If user has projects, loop through them and get the messages and tasks belonging to those projects

if (!empty($myOpenProjects)) {

foreach($myOpenProjects as $proj) {

//get all the tasks in this project that are assigned to the current user

$task = $mtask->getAllMyProjectTasks($proj["ID"], 100);

//get all messages in the project

$msgs = $msg->getProjectMessages($proj["ID"]);

//write those to arrays

if (!empty($msgs)) {

array\_push($messages, $msgs);

}

if (!empty($task)) {

array\_push($tasks, $task);

}

$cou = $cou + 1;

}

}

$myClosedProjects = $project->getMyProjects($userid,0);

// If the user is allowed to add projects, also get all users to assign to those projects

if ($userpermissions["projects"]["add"]) {

$user = new user();

$users = $user->getAllUsers(1000000);

$template->assign("users", $users);

$company = new company();

$companies = $company->getAllCompanies();

$template->assign("customers", $companies);

}

// by default the arrays have a level for each project, whcih contains arrays for each message/task . reduce array flattens this to have all messages/tasks of all projects in one structure

if (!empty($messages)) {

$messages = reduceArray($messages);

}

$etasks = reduceArray($tasks);

// Create sort array for multisort by adding the daysleft value to a sort array

$sort = array();

foreach($etasks as $etask) {

array\_push($sort, $etask["daysleft"]);

}

// Sort using array\_multisort

array\_multisort($sort, SORT\_NUMERIC, SORT\_ASC, $etasks);

// On Admin Login check for updates

$mode = getArrayVal($\_GET, "mode");

if ($mode == "login") {

$chkLim = 0;

// only check if an admin logs in

if ($userpermissions["admin"]["add"]) {

// only check 1/2 of the times an admin logs in, to reduce server load

$chkLim = mt\_rand(1, 2);

if ($chkLim == 1) {

$updateChk = getUpdateNotify();

if (!empty($updateChk)) {

if ($updateChk->pubDate > CL\_PUBDATE) {

$template->assign("isUpdated", true);

$template->assign("updateNotify", $updateChk);

}

}

}

}

}

// Get todays date and count tasks, projects and messages for display

$today = date("d");

$tasknum = count($etasks);

$projectnum = count($myOpenProjects);

$oldProjectnum = count($myClosedProjects[0]);

$msgnum = count($messages);

$title = $langfile["desktop"];

// Assign everything to the template engine

$template->assign("title", $title);

$template->assign("today", $today);

$template->assign("myprojects", $myOpenProjects);

$template->assign("oldprojects", $myClosedProjects);

$template->assign("projectnum", $projectnum);

$template->assign("closedProjectnum", $oldProjectnum);

$template->assign("projectov", "yes");

$template->assign("mode", $mode);

$template->assign("tasks", $etasks);

$template->assign("tasknum", $tasknum);

$template->assign("messages", $messages);

$template->assign("msgnum", $msgnum);

$template->display("index.tpl");

?>

INI.PHP

<?php

ini\_set("arg\_separator.output", "&amp;");

ini\_set('default\_charset', 'utf-8');

// Set content security policy header. This instructs the browser to block various unsafe behaviours.

header("Content-Security-Policy:default-src 'self'; style-src 'self' 'unsafe-inline'; script-src 'self' 'unsafe-inline' 'unsafe-eval';frame-src 'self'");

// Start output buffering with gzip compression and start the session

ob\_start('ob\_gzhandler');

session\_start();

// get full path to collabtive

define("CL\_ROOT", realpath(dirname(\_\_FILE\_\_)));

// configuration to load

define("CL\_CONFIG", "standard");

// collabtive version and release date

define("CL\_VERSION", 2.0);

define("CL\_PUBDATE", "1426201200");

// uncomment next line for debugging

// error\_reporting(E\_ALL || E\_STRICT);

// include config file , pagination and global functions

require(CL\_ROOT . "/config/" . CL\_CONFIG . "/config.php");

require(CL\_ROOT . "/include/SmartyPaginate.class.php");

// require html purifier

require(CL\_ROOT . "/include/HTMLPurifier.standalone.php");

// load init functions

require(CL\_ROOT . "/include/initfunctions.php");

//assume mysql as the default db

if(!isset($db\_driver))

{

$db\_driver = "mysql";

}

// Start database connection

// Depending on the DB driver, instantiate a PDO object with the necessary credentials.

switch ($db\_driver) {

case "mysql":

if (!empty($db\_name) and !empty($db\_user)) {

$conn = new PDO("mysql:host=$db\_host;dbname=$db\_name;charset=utf8", $db\_user, $db\_pass);

break;

}

case "sqlite":

$conn = new PDO("sqlite:" . CL\_ROOT . "/files/collabtive.sdb");

break;

}

// Start template engine

$template = new Smarty();

// STOP smarty from spewing notices all over the html code

$template->error\_reporting = E\_ALL &~E\_NOTICE;

// get the available languages

$languages = getAvailableLanguages();

// get URL to collabtive

$url = getMyUrl();

$template->assign("url", $url);

$template->assign("languages", $languages);

// set the version number for display

$template->assign("myversion", "2.1");

$template->assign("cl\_config", CL\_CONFIG);

// Assign globals to all templates

if (isset($\_SESSION["userid"])) {

// unique ID of the user

$userid = $\_SESSION["userid"];

// name of the user

$username = $\_SESSION["username"];

// timestamp of last login

$lastlogin = $\_SESSION["lastlogin"];

// selected locale

$locale = $\_SESSION["userlocale"];

// gender

$gender = $\_SESSION["usergender"];

// what the user may or may not do

$userpermissions = $\_SESSION["userpermissions"];

// update user lastlogin for the onlinelist

$mynow = time();

$upd = $conn->exec("UPDATE user SET lastlogin='$mynow' WHERE ID = $userid");

// assign it all to the templates

$template->assign("userid", $userid);

$template->assign("username", $username);

$template->assign("lastlogin", $lastlogin);

$template->assign("usergender", $gender);

$template->assign("userpermissions", $userpermissions);

$template->assign("loggedin", 1);

}else {

$template->assign("loggedin", 0);

}

// get system settings

if (isset($conn)) {

// Set PDO options

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_WARNING);

//$conn->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

// create a global mylog object for loging system events

$mylog = new mylog();

// get a settings object, and fetch an array containing the system settings

$set = (object) new settings();

$settings = $set->getSettings();

// define a constant that holds the default dateformat

define("CL\_DATEFORMAT", $settings["dateformat"]);

// set the default TZ for date etc

date\_default\_timezone\_set($settings["timezone"]);

$template->assign("settings", $settings);

}

// Set template directory

// If no directory is set in the system settings, default to the standard theme

if (isset($settings['template'])) {

$template->template\_dir = CL\_ROOT . "/templates/$settings[template]/";

// $template->tname = $settings["template"];

}else {

$template->template\_dir = CL\_ROOT . "/templates/standard/";

// $template->tname = "standard";

}

// If no locale is set, get the settings locale or default to english

if (!isset($locale)) {

if (isset($settings["locale"])) {

$locale = $settings['locale'];

}else {

$locale = "en";

}

$\_SESSION['userlocale'] = $locale;

}

// if detected locale doesnt have a corresponding langfile , use system default locale

// if, for whatever reason, no system default language is set, default to english as a last resort

if (!file\_exists(CL\_ROOT . "/language/$locale/lng.conf")) {

$locale = $settings['locale'];

$\_SESSION['userlocale'] = $locale;

}

// Set locale directory

$template->config\_dir = CL\_ROOT . "/language/$locale/";

// Smarty 3 seems to have a problem with re-compiling the config if the user config is different than the system config.

// this forces a compile of the config.

// uncomment this if you have issues with language switching

// $template->compileAllConfig('.config',true);

// read language file into PHP array

$langfile = readLangfile($locale);

$template->assign("langfile", $langfile);

$template->assign("locale", $locale);

// css classes for headmenue

// this indicates which of the 3 main stages the user is on

$mainclasses = array("desktop" => "desktop",

"profil" => "profil",

"admin" => "admin"

);

$template->assign("mainclasses", $mainclasses);

// get current year and month

$they = date("Y");

$them = date("n");

$template->assign("theM", $them);

$template->assign("theY", $they);

// Get the user's projects for the quickfinder in the sidebar

if (isset($userid)) {

$project = new project();

$myOpenProjects = $project->getMyProjects($userid);

$template->assign("openProjects", $myOpenProjects);

}

// clear session data for pagination

SmartyPaginate::disconnect();

?>

INSTALL.PHP

<?php

error\_reporting(0);

// Check if directory templates\_c exists and is writable

if (!file\_exists("./templates\_c") or !is\_writable("./templates\_c")) {

die("Required folder templates\_c does not exist or is not writable. <br>Please create the folder or make it writable in order to proceed.");

}

// check if the settings table / object is present. if yes, assume collabtive is already installed and abort

if (!empty($settings)) {

die("Collabtive seems to be already installed.<br />If this is an error, please clear your database.");

}

session\_start();

session\_destroy();

session\_unset();

setcookie("PHPSESSID", "");

date\_default\_timezone\_set("Europe/Berlin");

require("./init.php");

error\_reporting(0);

$action = getArrayVal($\_GET, "action");

$locale = getArrayVal($\_GET, "locale");

if (!empty($locale)) {

$\_SESSION['userlocale'] = $locale;

} else {

$locale = $\_SESSION['userlocale'];

}

if (empty($locale)) {

$locale = "en";

}

$title = $langfile['installcollabtive'];

$template->config\_dir = "./language/$locale/";

$template->template\_dir = "./templates/standard/";

$installSettings["template"] = "standard";

$installSettings["theme"] = "standard";

$template->assign("locale", $locale);

$template->assign("title", $title);

$template->assign("settings", $installSettings);

if (!$action) {

// check if required directories are writable

$configfilechk = is\_writable(CL\_ROOT . "/config/" . CL\_CONFIG . "/config.php");

$filesdir = is\_writable(CL\_ROOT . "/files/");

$templatesdir = is\_writable(CL\_ROOT . "/templates\_c/");

$phpver = phpversion();

$is\_mbstring\_enabled = extension\_loaded('mbstring');

$template->assign("phpver", $phpver);

$template->assign("configfile", $configfilechk);

$template->assign("filesdir", $filesdir);

$template->assign("templatesdir", $templatesdir);

$template->assign("is\_mbstring\_enabled", $is\_mbstring\_enabled);

$template->display("install1.tpl");

} elseif ($action == "step2") {

function randomPassword()

{

$alphabet = "abcdefghijklmnopqrstuwxyzABCDEFGHIJKLMNOPQRSTUWXYZ0123456789";

$pass = array(); //remember to declare $pass as an array

$alphaLength = strlen($alphabet) - 1; //put the length -1 in cache

for ($i = 0; $i < 16; $i++) {

$n = rand(0, $alphaLength);

$pass[] = $alphabet[$n];

}

return implode($pass); //turn the array into a string

}

//create a random password to encrypt files with

$filePass = randomPassword();

// check if the settings table / object is present. if yes, assume collabtive is already installed and abort

if (!empty($settings)) {

die("Collabtive seems to be already installed.<br />If this is an error, please clear your database.");

}

$db\_host = $\_POST['db\_host'];

$db\_name = $\_POST['db\_name'];

$db\_user = $\_POST['db\_user'];

$db\_pass = $\_POST['db\_pass'];

$db\_driver = $\_POST['db\_driver'];

// write db login data to config file

$file = fopen(CL\_ROOT . "/config/" . CL\_CONFIG . "/config.php", "w+");

$str = "<?php

\$db\_host = '$db\_host';\n

\$db\_name = '$db\_name';\n

\$db\_user = '$db\_user';\n

\$db\_pass = '$db\_pass';\n

\$db\_driver = '$db\_driver';\n

?>";

$put = fwrite($file, "$str");

if ($put) {

@chmod(CL\_ROOT . "/config/" . CL\_CONFIG . "/config.php", 0755);

}

//this will be checked in install\_mysql to make sure it can only be run from the installer

$installer\_include = "yes";

// connect database.

switch ($db\_driver) {

case "mysql":

require\_once("install\_mysql.php");

break;

case "sqlite":

$conn = new PDO("sqlite:" . CL\_ROOT . "/files/collabtive.sdb");

break;

}

// Get the servers default timezone

$timezone = date\_default\_timezone\_get();

// insert default settings

$defSets = array("name" => "Collabtive", "subtitle" => "Projectmanagement", "locale" => $locale, "timezone" => $timezone, "dateformat" => "d.m.Y", "template" => "standard", "mailnotify" => 1, "mailfrom" => "collabtive@localhost", "mailfromname" => "", "mailmethod" => "mail", "mailhost" => "", "mailuser" => "", "mailpass" => "", "rssuser" => "", "rsspass" => "", "theme" => "standard", "filePass" => $filePass);

foreach($defSets as $setKey => $setVal) {

$ins = $conn->query("INSERT INTO settings (`settingsKey`,`settingsValue`) VALUES ('$setKey','$setVal')");

}

if (!$ins) {

$template->assign("errortext", "Error: Failed to create initial settings.");

$template->display("error.tpl");

die();

}

$template->display("install2.tpl");

} elseif ($action == "step3") {

//create required folders

mkdir(CL\_ROOT . "/files/" . CL\_CONFIG . "/");

mkdir(CL\_ROOT . "/files/" . CL\_CONFIG . "/avatar/", 0777);

mkdir(CL\_ROOT . "/files/" . CL\_CONFIG . "/ics/", 0777);

//include the database cnfig

require(CL\_ROOT . "/config/" . CL\_CONFIG . "/config.php");

// Start database connection

switch ($db\_driver) {

case "mysql":

$conn = new PDO("mysql:host=$db\_host;dbname=$db\_name;charset=utf8", $db\_user, $db\_pass);

break;

case "sqlite":

$conn = new PDO("sqlite:" . CL\_ROOT . "/files/collabtive.sdb");

break;

}

$user = $\_POST["name"];

$pass = $\_POST["pass"];

// create the first user

$usr = new user();

//check if there are existing users, in which case collabtive is already installed

$installChk = $usr->getAllUsers();

if ($installChk) {

// There already are users. abort install.

die("Collabtive seems to be already installed.<br />If this is an error, please clear your database.");

}

//add the first user

$usrid = $usr->add($user, "", 0, $pass);

if (!$usrid) {

$template->assign("errortext", "Error: Failed to create first user.");

$template->display("error.tpl");

die();

}

// insert default roles

$rolesobj = new roles();

$adminrid = $rolesobj->add("Admin", array("add" => 1, "edit" => 1, "del" => 1 , "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "read" => 1, "view" => 1), array("add" => 1), array("add" => 1));

$userrid = $rolesobj->add("User", array("add" => 1, "edit" => 1, "del" => 0, "close" => 0, "view" => 1), array("add" => 1, "edit" => 1, "del" => 0, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "close" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "view" => 1), array("add" => 1, "edit" => 1, "del" => 1, "read" => 0, "view" => 1), array("add" => 1), array("add" => 0));

$clientrid = $rolesobj->add("Client", array("add" => 0, "edit" => 0, "del" => 0, "close" => 0), array("add" => 0, "edit" => 0, "del" => 0, "close" => 0), array("add" => 0, "edit" => 0, "del" => 0, "close" => 0), array("add" => 0, "edit" => 0, "del" => 0, "close" => 0), array("add" => 0, "edit" => 0, "del" => 0), array("add" => 0, "edit" => 0, "del" => 0, "read" => 0), array("add" => 0), array("add" => 0));

if (!$adminrid or !$userrid or !$clientrid) {

$template->assign("errortext", "Error: Failed to create initial roles.");

$template->display("error.tpl");

die();

}

$rolesobj->assign($adminrid, $usrid);

$template->display("install3.tpl");

}

?>

**MANAGEAJAX.PHP**

<?php

require("init.php");

/\*

if (!isset($\_SESSION["userid"]))

{

$template->assign("loginerror", 0);

$mode = getArrayVal($\_GET, "mode");

$template->assign("mode", $mode);

$template->display("login.tpl");

die();

}

\*/

$milestone = new milestone();

$action = getArrayVal($\_GET, "action");

$mid = getArrayVal($\_GET, "mid");

$mode = getArrayVal($\_GET, "mode");

$template->assign("mode", $mode);

$id = getArrayVal($\_GET, "id");

$start = getArrayVal($\_GET, "start");

$end = getArrayVal($\_GET, "end");

$project = array('ID' => $id);

$template->assign("project", $project);

$pro = new project();

if (!$id) {

$id = 0;

}

$template->assign("id", $id);

if ($action == "makeinputs") {

$num = getArrayVal($\_GET, "num");

$file = $langfile["file"] . ":";

$title = $langfile["title"] . ":";

$tags = $langfile["tags"] . ":";

for($i = 1;$i <= $num;$i++) {

echo " <div class=\"row\"><label for = \"title$i\">$title </label><input type = \"text\" name = \"userfile$i-title\" id=\"title$i\" /></div>

<div class=\"row\"><label for = \"tags$i\">$tags </label><input type = \"text\" name = \"userfile$i-tags\" id=\"tags$i\" /></div>

<div class=\"row\"><label for = \"userfile$i\">$file </label><input type=\"file\" id = \"userfile$i\" name=\"userfile$i\" /><div style=\"clear:left\"></div>";

}

}

//This is used to add the search functionality to firefoxs seachbar

elseif ($action == "addfx-all") {

$templ = $url . "managesearch.php?action=search&amp;query={searchTerms}";

$templ2 = $url . "managesearch.php?action=searchjson&amp;query={searchTerms}";

$fav = $url . "templates/standard/images/favicon.ico";

$strsearch = $langfile["search"];

$sysname = $settings["name"];

echo "

<OpenSearchDescription xmlns=\"http://a9.com/-/spec/opensearch/1.1/\">

<ShortName>$sysname $strsearch</ShortName>

<Description>Search all Collabtive</Description>

<Tags></Tags>

<Image height=\"16\" width=\"16\" type=\"image/x-icon\">$fav</Image>

<Url type=\"text/html\" method=\"get\" template=\"$templ\"/>

<Url type=\"application/x-suggestions+json\" method=\"get\" template=\"$templ2\"/>

<InputEncoding>UTF-8</InputEncoding>

<OutputEncoding>UTF-8</OutputEncoding>

<AdultContent>false</AdultContent>

</OpenSearchDescription>";

}

//This is used to add the search functionality to firefoxs seachbar

elseif ($action == "addfx-project") {

$templ = $url . "managesearch.php?action=projectsearch&amp;project=$project&amp;query={searchTerms}";

$templ2 = $url . "managesearch.php?action=searchjson-project&amp;project=$project&amp;query={searchTerms}";

$fav = $url . "templates/standard/images/favicon.ico";

$project = $\_GET['project'];

$strsearch = $langfile["search"];

$pro = new project();

$pname = $pro->getProject($project);

$pname = $pname["name"];

echo "

<OpenSearchDescription xmlns=\"http://a9.com/-/spec/opensearch/1.1/\">

<ShortName>$pname $strsearch</ShortName>

<Description>Search project $pname</Description>

<Tags></Tags>

<Image height=\"16\" width=\"16\" type=\"image/x-icon\">$fav</Image>

<Url type=\"text/html\" method=\"GET\" template=\"$templ\"/>

<Url type=\"application/x-suggestions+json\" method=\"get\" template=\"$templ2\"/>

<InputEncoding>UTF-8</InputEncoding>

<OutputEncoding>UTF-8</OutputEncoding>

<AdultContent>false</AdultContent>

</OpenSearchDescription>";

}

//This is used to put file lists into tinymce for selection

elseif ($action == "jsonfiles") {

if (!chkproject($userid, $id)) {

$errtxt = $langfile["notyourproject"];

$noperm = $langfile["accessdenied"];

$template->assign("errortext", "$errtxt<br>$noperm");

$template->display("error.tpl");

die();

}

$myfile = new datei();

$ordner = $myfile->getAllProjectFiles($id);

if (!empty($ordner)) {

$json = "var tinyMCEImageList = new Array(\n";

foreach($ordner as $file) {

if ($file["imgfile"] == 1) {

$json .= "[\"$file[datei]\", \"managefile.php?action=downloadfile&id=$file[project]&file=$file[ID]\"],\n";

}

}

$json = substr($json, 0, strlen($json)-2);

$json .= ");";

} else {

$json = "";

}

echo $json;

}

//this is used to display the project files

elseif ($action == "fileview") {

if (!chkproject($userid, $id)) {

$errtxt = $langfile["notyourproject"];

$noperm = $langfile["accessdenied"];

$template->assign("errortext", "$errtxt<br>$noperm");

$template->display("error.tpl");

die();

}

$POST\_MAX\_SIZE = ini\_get('post\_max\_size');

$POST\_MAX\_SIZE = $POST\_MAX\_SIZE . "B";

$folder = getArrayVal($\_GET, "folder");

$myfile = new datei();

$ordner = $myfile->getProjectFiles($id, 1000000, $folder);

$finfiles = array();

if (!empty($ordner)) {

foreach($ordner as $file) {

array\_push($finfiles, $file);

}

}

$filenum = count($finfiles);

if (empty($finfiles)) {

$filenum = 0;

}

if ($folder == 0) {

$folders = $myfile->getProjectFolders($id);

$foldername = "";

$thefolder = array("parent" => 0);

} else {

$folders = $myfile->getProjectFolders($id, $folder);

$thefolder = $myfile->getFolder($folder);

$foldername = $thefolder["abspath"];

}

$finfolders = $folders;

$template->assign("filenum", $filenum);

$template->assign("foldername", $foldername);

if (!$thefolder["parent"]) {

$thefolder["parent"] = 0;

}

$template->assign("folders", $finfolders);

$template->assign("folderid", $thefolder["parent"]);

$template->assign("langfile", $langfile);

$template->assign("theAction", "fileview");

SmartyPaginate::assign($template);

$template->assign("files", $finfiles);

$template->assign("postmax", $POST\_MAX\_SIZE);

$template->display("fileview.tpl");

}

//this is used to display the project files

elseif ($action == "fileview\_list") {

if (!chkproject($userid, $id)) {

$errtxt = $langfile["notyourproject"];

$noperm = $langfile["accessdenied"];

$template->assign("errortext", "$errtxt<br>$noperm");

$template->display("error.tpl");

die();

}

$POST\_MAX\_SIZE = ini\_get('post\_max\_size');

$POST\_MAX\_SIZE = $POST\_MAX\_SIZE . "B";

$folder = getArrayVal($\_GET, "folder");

$myfile = new datei();

$ordner = $myfile->getProjectFiles($id, 1000000, $folder);

$finfiles = array();

if (!empty($ordner)) {

foreach($ordner as $file) {

array\_push($finfiles, $file);

}

}

$filenum = count($finfiles);

if (empty($finfiles)) {

$filenum = 0;

}

if ($folder == 0) {

$folders = $myfile->getProjectFolders($id);

$foldername = "";

$thefolder["parent"] = 0;

} else {

$folders = $myfile->getProjectFolders($id, $folder);

$thefolder = $myfile->getFolder($folder);

$foldername = $thefolder["abspath"];

}

$finfolders = $folders;

$template->assign("folders", $finfolders);

$template->assign("filenum", $filenum);

$template->assign("foldername", $foldername);

$template->assign("folderid", $thefolder["parent"]);

$template->assign("langfile", $langfile);

$template->assign("theAction", "fileview\_list");

SmartyPaginate::assign($template);

$template->assign("files", $finfiles);

$template->assign("postmax", $POST\_MAX\_SIZE);

$template->display("fileview\_list.tpl");

} elseif ($action == "folderview") {

if (!chkproject($userid, $id)) {

$errtxt = $langfile["notyourproject"];

$noperm = $langfile["accessdenied"];

$template->assign("errortext", "$errtxt<br>$noperm");

$template->display("error.tpl");

die();

}

$myfile = new datei();

$ordner = $myfile->getProjectFolders($id);

$myproject = new project();

$template->assign("langfile", $langfile);

$template->assign("ordner", $ordner);

$template->display("folderview.tpl");

}

//this is used to display the calendar on the desktop

elseif ($action == "newcal") {

$thisd = date("j");

$thism = date("n");

$thisy = date("Y");

$m = getArrayVal($\_GET, "m");

$y = getArrayVal($\_GET, "y");

if (!$m) {

$m = $thism;

}

if (!$y) {

$y = $thisy;

}

$nm = $m + 1;

$pm = $m -1;

if ($nm > 12) {

$nm = 1;

$ny = $y + 1;

} else {

$ny = $y;

}

if ($pm < 1) {

$pm = 12;

$py = $y-1;

} else {

$py = $y;

}

$today = date("d");

$calobj = new calendar();

$cal = $calobj->getCal($m, $y);

$weeks = $cal->calendar;

// print\_r($weeks);

$mstring = strtolower(date('F', mktime(0, 0, 0, $m, 1, $y)));

$mstring = $langfile[$mstring];

$template->assign("mstring", $mstring);

$template->assign("m", $m);

$template->assign("y", $y);

$template->assign("thism", $thism);

$template->assign("thisd", $thisd);

$template->assign("thisy", $thisy);

$template->assign("nm", $nm);

$template->assign("pm", $pm);

$template->assign("ny", $ny);

$template->assign("py", $py);

$template->assign("weeks", $weeks);

$template->display("calbody.tpl");

}

elseif($action == "chkconn")

{

$dbHost = getArrayVal($\_GET,"dbhost");

$dbUser = getArrayVal($\_GET,"dbuser");

$dbName = getArrayVal($\_GET,"dbname");

$dbPass = getArrayVal($\_GET,"dbpass");

$chk = new PDO("mysql:host=$dbHost;dbname=$dbName;charset=utf8", $dbUser, $dbPass);

echo $chk;

}

**UPDATE.PHP**

<?php

error\_reporting(0);

// Check if directory templates\_c exists and is writable

if (!file\_exists("./templates\_c") or !is\_writable("./templates\_c")) {

die("Required folder templates\_c does not exist or is not writable. <br>Please create the folder or make it writable in order to proceed.");

}

require("./init.php");

// VERSION-DEPENDENT

// VERSION-INDEPENDENT

// Clear templates cache

$handle = opendir($template->compile\_dir);

while (false !== ($file = readdir($handle))) {

if ($file != "." and $file != "..") {

unlink(CL\_ROOT . "/" . $template->compile\_dir . "/" . $file);

}

}

// Optimize tables

$opt1 = $conn->query("OPTIMIZE TABLE `files`");

$opt2 = $conn->query("OPTIMIZE TABLE `files\_attached`");

$opt3 = $conn->query("OPTIMIZE TABLE `log`");

$opt4 = $conn->query("OPTIMIZE TABLE `messages`");

$opt5 = $conn->query("OPTIMIZE TABLE `milestones`");

$opt6 = $conn->query("OPTIMIZE TABLE `milestones\_assigned`");

$opt7 = $conn->query("OPTIMIZE TABLE `projectfolders`");

$opt8 = $conn->query("OPTIMIZE TABLE `projekte`");

$opt9 = $conn->query("OPTIMIZE TABLE `projekte\_assigned`");

$opt10 = $conn->query("OPTIMIZE TABLE `roles`");

$opt11 = $conn->query("OPTIMIZE TABLE `roles\_assigned`");

$opt12 = $conn->query("OPTIMIZE TABLE `settings`");

$opt13 = $conn->query("OPTIMIZE TABLE `tasklist`");

$opt14 = $conn->query("OPTIMIZE TABLE `tasks`");

$opt15 = $conn->query("OPTIMIZE TABLE `tasks\_assigned`");

$opt16 = $conn->query("OPTIMIZE TABLE `timetracker`");

$opt17 = $conn->query("OPTIMIZE TABLE `user`");

$template->display("update.tpl");

?>

**COFIG.PHP**

<?php

$db\_host = 'localhost';

$db\_name = 'collaborate';

$db\_user = 'collaborate';

$db\_pass = 'collaborate';

$db\_driver = 'mysql';

?>

**ADMINUSERINFO.PHP**

{include file="header.tpl" jsload="ajax"}

{include file="tabsmenue-admin.tpl" usertab="active"}

<div id="content-left">

<div id="content-left-in">

<div class="user">

<h1>{#deleteuser#}<span>/ {$user.name}</span></h1>

<div class="block\_in\_wrapper">

<h2>{#deleteform#}</h2>

<form class="main" method="post" action="admin.php?action=deleteuser&amp;id={$user.ID}" enctype="multipart/form-data" {literal} onsubmit="return validateCompleteForm(this);" {/literal} >

<fieldset>

<input type="hidden" name="id" value="{$user.ID}" />

{section name=proj loop=$projects}

<div class="row">

<h3>{$projects[proj].name}</h3>

</div>

<div class="row">

<label for="{$projects[proj].ID}membs">{#assignto#}:</label>

<select name="uprojects[]" id="{$projects[proj].ID}membs">

<option value="{$projects[proj].ID}#0" selected>{#deletetasks#}</option>

{section name=member loop=$projects[proj].members}

{if $projects[proj].members[member].ID != $user.ID}

<option value="{$projects[proj].ID}#{$projects[proj].members[member].ID}" />{$projects[proj].members[member].name}</option>

{/if}

{/section}

</select>

</div>

<div class="content-spacer"></div>

{/section}

<div class="row-butn-bottom">

<label>&nbsp;</label>

<button type="submit"onfocus="this.blur();">{#send#}</button>

<button onclick="javascript:history.back();return false;" onfocus="this.blur();">{#cancel#}</button>

</div>

</fieldset>

</form>

<div class="clear\_both"></div> {\*required ... do not delete this row\*}

</div> {\*block\_in\_wrapper end\*}

<div class="content-spacer"></div>

</div> {\*User END\*}

</div> {\*content-left-in END\*}

</div> {\*Content\_left end\*}

{include file="sidebar-a.tpl"}

{include file="footer.tpl"}

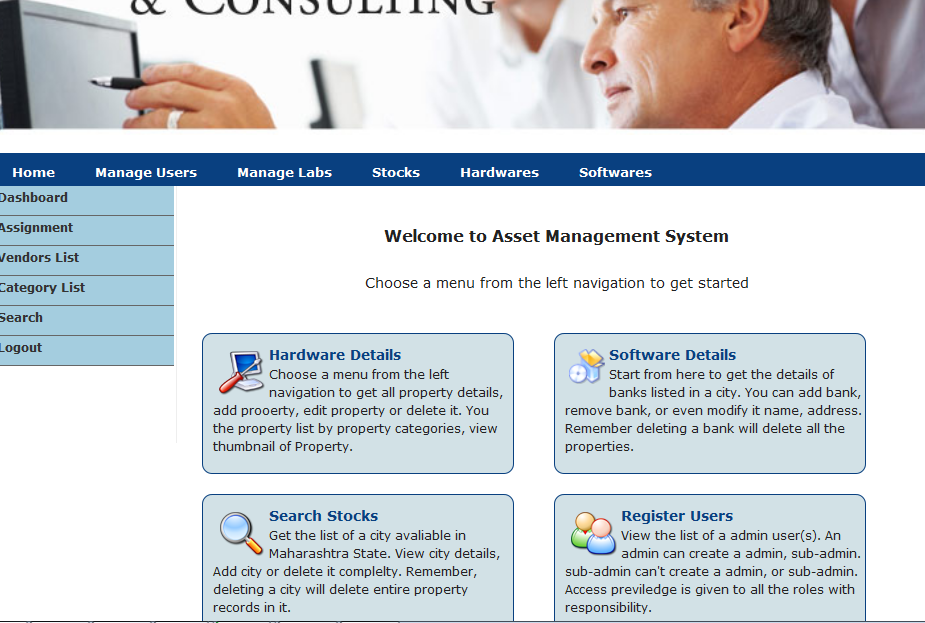
**APPENDIX**

**APPENDIX A: SCREEN WINDOW**

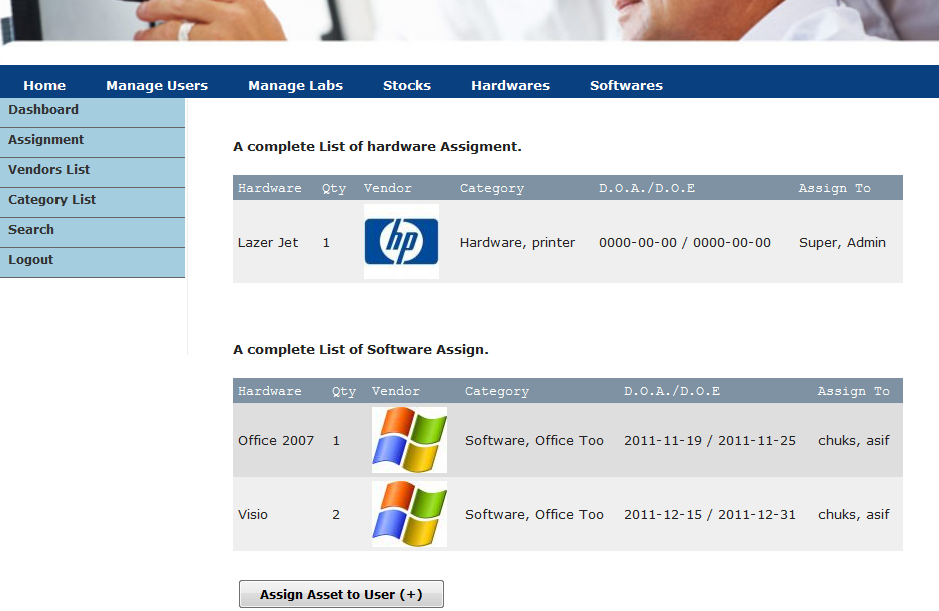
login.php



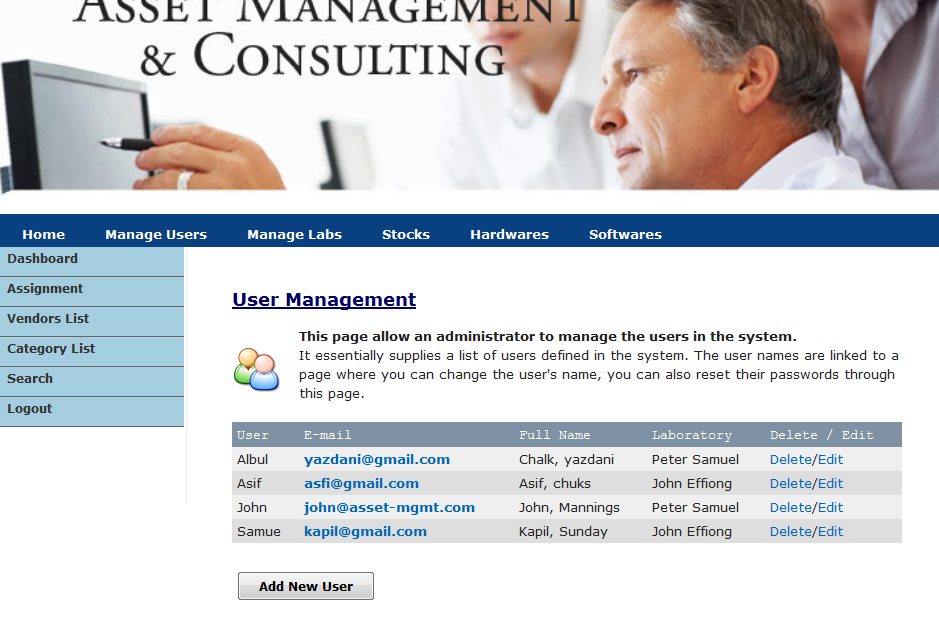
**Index.php**



**ASSIGN HOME.PHP**

****

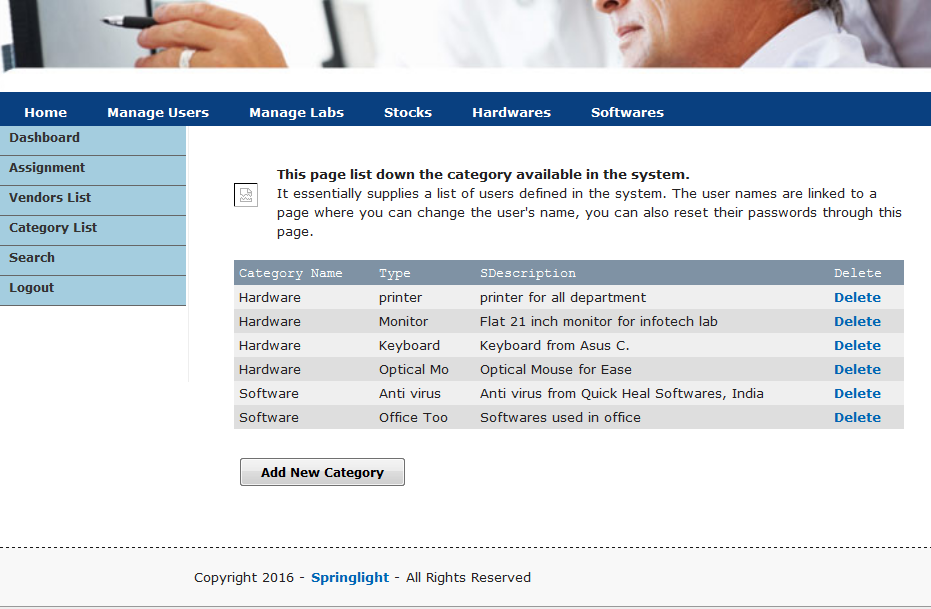
**MENU.PHP**

****

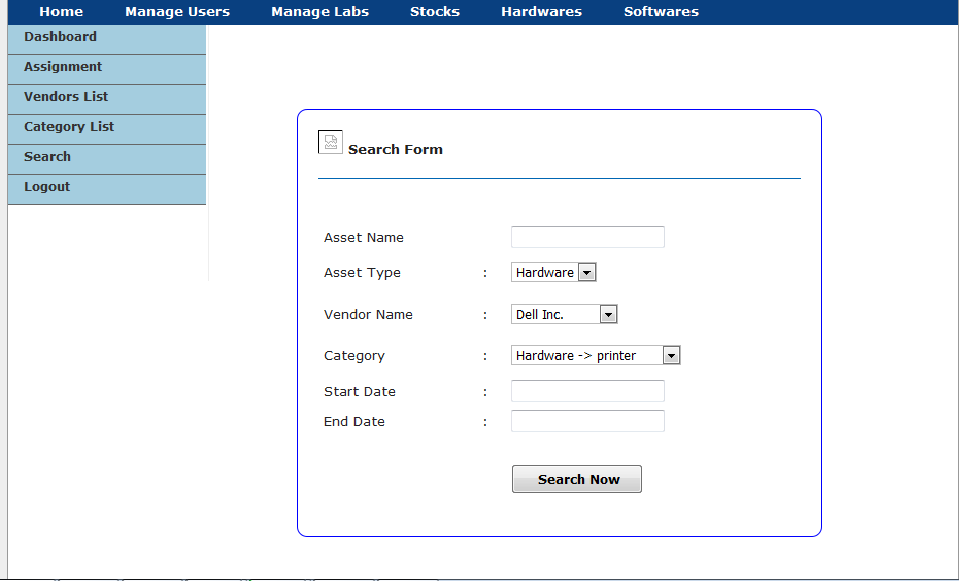
**VIEW.PHP**

****

**CATEGORY FOLDER.PHP**

****

**SEARCH.PHP**

****

**APPENDIX B : SOURCE CODE**

<?php

require\_once 'library/config.php';

require\_once 'library/functions.php';

checkUser();

$content = 'main.php';

$pageTitle = 'Asset Management';

$script = array();

require\_once 'template.php';

?>

**LOGIN.PHP**

<?php

require\_once 'library/config.php';

require\_once 'library/functions.php';

$errorMessage = '&nbsp;';

if (isset($\_POST['txtUserName'])) {

$result = doLogin();

if ($result != '') {

$errorMessage = $result;

}

}

?>

<html>

<head>

<title>Asset Management - Login</title>

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

<link href="css/screen.css" rel="stylesheet" type="text/css">

<link href="css/menu.css" rel="stylesheet" type="text/css">

<link href="include/style.css" rel="stylesheet" type="text/css">

<style>

body{ margin-top:20px; background-color:#EEEEEE;}

a {text-decoration:none;}

</style>

</head>

<body>

<div class="container" style="border: 1px solid #999999;">

<div class="span-24">

<img src="images/asset\_header.jpg" width="950"/>

</div>

<div class="span-6">&nbsp;</div>

<div class="span-12" style="margin:50px 0px;">

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

<tr>

<td>

<img src="images/Vista\_icons\_09.png" /> </td>

<td>

<h2 style="font-size:18px;">User Login</h2>

<form method="post" name="frmLogin" id="frmLogin">

<div class="errorMessage" align="center"><?php echo $errorMessage; ?></div>

<p><strong>Name :&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</strong>

<input name="txtUserName" type="text" id="txtUserName" size="10" maxlength="20">

<br/><br/>

<strong>Password : </strong>

<input name="txtPassword" type="password" id="txtPassword" size="10"><br/>

<br/><input name="btnLogin" type="submit" id="btnLogin" value="Login" class="button">

</p>

</form>

</td>

</tr>

</table>

</div>

<div class="span-6">&nbsp;</div>

<div class="prepend-5 span-15 append-4" style="background-color:#F8F8F8; border-top:dashed 1px #181818;">

<?php include\_once("footer.php"); ?>

</div>

</div>

</body>

</html>

**TEMPLATE.PHP**

<?php

if (!defined('WEB\_ROOT')) {

exit;

}

$self ='index.php';

?>

<html>

<head>

<title><?php echo $pageTitle; ?></title>

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

<link href="../css/screen.css" rel="stylesheet" type="text/css">

<link href="../css/menu.css" rel="stylesheet" type="text/css">

<script language="JavaScript" type="text/javascript" src="../library/common.js"></script>

<script language="JavaScript" type="text/javascript" src="../library/lab.js"></script>

<script language="javascript" type="text/javascript" src="../library/jquery.min.js" ></script>

<script language="JavaScript" type="text/javascript" src="../library/software.js"></script>

<script language="JavaScript" type="text/javascript" src="../library/assignment.js"></script>

<?php

$n = count($script);

for ($i = 0; $i < $n; $i++) {

if ($script[$i] != '') {

echo '<script language="JavaScript" type="text/javascript" src="library/' . $script[$i]. '"></script>';

}

}

?>

<style>

body{margin-top:20px; margin-bottom:20px;background-color:#EEEEEE;}

a {text-decoration:none;}

</style>

</head>

<body>

<div class="container" style="border: 1px solid #999999; margin-bottom:20px;">

<div class="span-24">

<img src="../images/asset\_header.jpg" width="950"/>

</div>

<div class="span-24">

<?php include\_once("mymenu.php"); ?>

</div>

<div class="span-5 border">

<?php include\_once("left.php"); ?>

</div>

<div class="span-19 last">

<?php

require\_once $content;

?>

</div>

<div class="prepend-5 span-15 append-4" style="background-color:#F8F8F8; border-top:dashed 1px #181818;">

<?php include\_once("../footer.php"); ?>

</div>

</div>

</body>

</html>

**FUNCTION.PHP**

<?php

function checkUser()

{

if (!isset($\_SESSION['asset\_user\_id'])) {

header('Location: login.php');

exit;

}

if (isset($\_GET['logout'])) {

doLogout();

}

}

function doLogin()

{

// if we found an error save the error message in this variable

$errorMessage = '';

$userName = $\_POST['txtUserName'];

$password = $\_POST['txtPassword'];

// first, make sure the username & password are not empty

if ($userName == '') {

$errorMessage = 'You must enter your username';

} else if ($password == '') {

$errorMessage = 'You must enter the password';

} else {

// check the database and see if the username and password combo do match

$sql = "SELECT uid, uname, utype

FROM tbl\_users

WHERE uname = '$userName' AND pwd = '$password'";

$result = dbQuery($sql);

if (dbNumRows($result) == 1) {

$row = dbFetchAssoc($result);

$\_SESSION['asset\_user\_id'] = $row['uid'];

$\_SESSION['asset\_user\_name'] = $row['uname'];

$\_SESSION['asset\_user\_type'] = $row['utype'];

// now that the user is verified we move on to the next page

// if the user had been in the admin pages before we move to

// the last page visited

if (isset($\_SESSION['login\_return\_url'])) {

header('Location: ' . $\_SESSION['login\_return\_url']);

exit;

} else {

header('Location: ../index.php');

exit;

}

} else {

$errorMessage = 'Wrong username or password';

}

}

return $errorMessage;

}

/\*

Logout a user

\*/

function doLogout()

{

if (isset($\_SESSION['asset\_user\_id'])) {

unset($\_SESSION['asset\_user\_id']);

//session\_unregister('asset\_user\_id');

}

header('Location: ../login.php');

exit;

}

/\*

Generate combo box options containing the categories we have.

if $catId is set then that category is selected

\*/

function buildCategoryOptions($catId = 0)

{

$sql = "SELECT cat\_id, cat\_parent\_id, cat\_name

FROM tbl\_category

ORDER BY cat\_id";

$result = dbQuery($sql) or die('Cannot get Product. ' . mysql\_error());

$categories = array();

while($row = dbFetchArray($result)) {

list($id, $parentId, $name) = $row;

if ($parentId == 0) {

// we create a new array for each top level categories

$categories[$id] = array('name' => $name, 'children' => array());

} else {

// the child categories are put int the parent category's array

$categories[$parentId]['children'][] = array('id' => $id, 'name' => $name);

}

}

// build combo box options

$list = '';

foreach ($categories as $key => $value) {

$name = $value['name'];

$children = $value['children'];

$list .= "<optgroup label=\"$name\">";

foreach ($children as $child) {

$list .= "<option value=\"{$child['id']}\"";

if ($child['id'] == $catId) {

$list.= " selected";

}

$list .= ">{$child['name']}</option>\r\n";

}

$list .= "</optgroup>";

}

return $list;

}

/\*

If you want to be able to add products to the first level category

replace the above function with the one below

\*/

/\*

function buildCategoryOptions($catId = 0)

{

$sql = "SELECT cat\_id, cat\_parent\_id, cat\_name

FROM tbl\_category

ORDER BY cat\_id";

$result = dbQuery($sql) or die('Cannot get Product. ' . mysql\_error());

$categories = array();

while($row = dbFetchArray($result)) {

list($id, $parentId, $name) = $row;

if ($parentId == 0) {

// we create a new array for each top level categories

$categories[$id] = array('name' => $name, 'children' => array());

} else {

// the child categories are put int the parent category's array

$categories[$parentId]['children'][] = array('id' => $id, 'name' => $name);

}

}

// build combo box options

$list = '';

foreach ($categories as $key => $value) {

$name = $value['name'];

$children = $value['children'];

$list .= "<option value=\"$key\"";

if ($key == $catId) {

$list.= " selected";

}

$list .= ">$name</option>\r\n";

foreach ($children as $child) {

$list .= "<option value=\"{$child['id']}\"";

if ($child['id'] == $catId) {

$list.= " selected";

}

$list .= ">&nbsp;&nbsp;{$child['name']}</option>\r\n";

}

}

return $list;

}

\*/

/\*

Create a thumbnail of $srcFile and save it to $destFile.

The thumbnail will be $width pixels.

\*/

function createThumbnail($srcFile, $destFile, $width, $quality = 75)

{

$thumbnail = '';

if (file\_exists($srcFile) && isset($destFile))

{

$size = getimagesize($srcFile);

$w = number\_format($width, 0, ',', '');

$h = number\_format(($size[1] / $size[0]) \* $width, 0, ',', '');

$thumbnail = copyImage($srcFile, $destFile, $w, $h, $quality);

}

// return the thumbnail file name on sucess or blank on fail

return basename($thumbnail);

}

/\*

Copy an image to a destination file. The destination

image size will be $w X $h pixels

\*/

function copyImage($srcFile, $destFile, $w, $h, $quality = 75)

{

$tmpSrc = pathinfo(strtolower($srcFile));

$tmpDest = pathinfo(strtolower($destFile));

$size = getimagesize($srcFile);

if ($tmpDest['extension'] == "gif" || $tmpDest['extension'] == "jpg")

{

$destFile = substr\_replace($destFile, 'jpg', -3);

$dest = imagecreatetruecolor($w, $h);

imageantialias($dest, TRUE);

} elseif ($tmpDest['extension'] == "png") {

$dest = imagecreatetruecolor($w, $h);

imageantialias($dest, TRUE);

} else {

return false;

}

switch($size[2])

{

case 1: //GIF

$src = imagecreatefromgif($srcFile);

break;

case 2: //JPEG

$src = imagecreatefromjpeg($srcFile);

break;

case 3: //PNG

$src = imagecreatefrompng($srcFile);

break;

default:

return false;

break;

}

imagecopyresampled($dest, $src, 0, 0, 0, 0, $w, $h, $size[0], $size[1]);

switch($size[2])

{

case 1:

case 2:

imagejpeg($dest,$destFile, $quality);

break;

case 3:

imagepng($dest,$destFile);

}

return $destFile;

}

/\*

Create the paging links

\*/

function getPagingNav($sql, $pageNum, $rowsPerPage, $queryString = '')

{

$result = mysql\_query($sql) or die('Error, query failed. ' . mysql\_error());

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$numrows = $row['numrows'];

// how many pages we have when using paging?

$maxPage = ceil($numrows/$rowsPerPage);

$self = $\_SERVER['PHP\_SELF'];

// creating 'previous' and 'next' link

// plus 'first page' and 'last page' link

// print 'previous' link only if we're not

// on page one

if ($pageNum > 1)

{

$page = $pageNum - 1;

$prev = " <a href=\"$self?page=$page{$queryString}\">[Prev]</a> ";

$first = " <a href=\"$self?page=1{$queryString}\">[First Page]</a> ";

}

else

{

$prev = ' [Prev] '; // we're on page one, don't enable 'previous' link

$first = ' [First Page] '; // nor 'first page' link

}

// print 'next' link only if we're not

// on the last page

if ($pageNum < $maxPage)

{

$page = $pageNum + 1;

$next = " <a href=\"$self?page=$page{$queryString}\">[Next]</a> ";

$last = " <a href=\"$self?page=$maxPage{$queryString}{$queryString}\">[Last Page]</a> ";

}

else

{

$next = ' [Next] '; // we're on the last page, don't enable 'next' link

$last = ' [Last Page] '; // nor 'last page' link

}

// return the page navigation link

return $first . $prev . " Showing page <strong>$pageNum</strong> of <strong>$maxPage</strong> pages " . $next . $last;

}

?>

**COFIG.PHP**

<?php

require\_once '../../library/config.php';

require\_once '../library/functions.php';

checkUser();

$view = (isset($\_GET['view']) && $\_GET['view'] != '') ? $\_GET['view'] : '';

switch ($view) {

default :

$content = 'main.php';

$pageTitle = 'Shop Admin Control Panel - Shop Configuration';

}

$script = array('shop.js');

require\_once '../include/template.php';

?>

**PROCESSCONFIG.PHP**

<?php

require\_once '../../library/config.php';

require\_once '../library/functions.php';

checkUser();

$action = isset($\_GET['action']) ? $\_GET['action'] : '';

switch ($action) {

case 'modify' :

modifyShopConfig();

break;

default :

// if action is not defined or unknown

// move to main page

header('Location: index.php');

}

function modifyShopConfig()

{

$shopName = $\_POST['txtShopName'];

$address = $\_POST['mtxAddress'];

$phone = $\_POST['txtPhone'];

$email = $\_POST['txtEmail'];

$shipping = $\_POST['txtShippingCost'];

$currency = $\_POST['cboCurrency'];

$sendEmail = $\_POST['optSendEmail'];

$sql = "UPDATE tbl\_shop\_config

SET sc\_name = '$shopName', sc\_address = '$address', sc\_phone = '$phone', sc\_email = '$email',

sc\_shipping\_cost = $shipping, sc\_currency = $currency, sc\_order\_email = '$sendEmail'";

$result = dbQuery($sql);

header("Location: index.php");

}

?>

**INDEX.PHP**

<?php

require\_once '../library/config.php';

require\_once '../library/functions.php';

$\_SESSION['login\_return\_url'] = $\_SERVER['REQUEST\_URI'];

checkUser();

$view = (isset($\_GET['view']) && $\_GET['view'] != '') ? $\_GET['view'] : '';

switch ($view) {

case 'list' :

$content = 'list.php';

$pageTitle = 'Shop Admin Control Panel - View Users';

break;

case 'add' :

$content = 'add.php';

$pageTitle = 'Shop Admin Control Panel - Add Users';

break;

case 'edit' :

$content = 'edit.php';

$pageTitle = 'Asset Management - Edit User';

break;

default :

$content = 'list.php';

$pageTitle = 'Shop Admin Control Panel - View Users';

}

$script = array('user.js');

require\_once 'template.php';

?>

**USER/INDEX.PHP**

<?php

if (!defined('WEB\_ROOT')) {

exit;

}

$sql = "SELECT u.uid, u.uname, u.email, u.fname, u.lname, d.lname AS dname

FROM tbl\_users u,

tbl\_depts d

WHERE u.utype != 'ADMIN' AND u.did = d.id

ORDER BY uname";

$result = dbQuery($sql);

?>

<div class="prepend-1 span-17">

<p>&nbsp;</p>

<h2 class="catHead">User Management</h2>

<p><img src="images/users.png" class="left"/>

<strong>This page allow an administrator to manage the users in the system.</strong>

<br/>

It essentially supplies a list of users defined in the system. The user names are linked to a page where you can change the user's name, you can also reset their passwords through this page.

</p>

<form action="processUser.php?action=addUser" method="post" name="frmListUser" id="frmListUser">

<table border="0" align="center" cellpadding="2" cellspacing="1" class="text">

<tr align="center" id="listTableHeader">

<td>User</td>

<td>E-mail</td>

<td>Full Name</td>

<td>Laboratory</td>

<td>Delete&nbsp;/&nbsp;Edit</td>

</tr>

<?php

while($row = dbFetchAssoc($result)) {

extract($row);

if ($i%2) {

$class = 'row1';

} else {

$class = 'row2';

}

$i += 1;

?>

<tr class="<?php echo $class; ?>">

<td><?php echo ucfirst($uname); ?></td>

<td align="center"><a href="mailto:<?php echo $email; ?>"><?php echo $email; ?></a></td>

<td align="center"><?php echo ucfirst($fname.", ".$lname); ?></td>

<td align="center"><?php echo $dname; ?></td>

<td align="center"><a style="font-weight:normal;" href="javascript:deleteUser(<?php echo $uid; ?>);">Delete</a>/<a style="font-weight:normal;" href="javascript:editUser(<?php echo $uid; ?>);">Edit</a></td>

</tr>

<?php

} // end while

?>

<tr>

<td colspan="5">&nbsp;</td>

</tr>

<tr>

<td colspan="5" align="right"><input name="btnAddUser" type="button" id="btnAddUser" value="Add New User" class="button" onClick="addUser()"></td>

</tr>

</table>

<p>&nbsp;</p>

</form>

</div>

**SEARCH.PHP**

<link href="../css/jquery.ui.datepicker.css" rel="stylesheet" type="text/css" />

<link href="../css/jquery.ui.theme.css" rel="stylesheet" type="text/css" />

<script src="../library/jquery.min.js" language="javascript"></script>

<script src="../library/jquery.ui.core.js" language="javascript"></script>

<script src="../library/jquery.ui.datepicker.js" language="javascript"></script>

<script language="javascript">

$(function() {

$("input#sdate").datepicker({dateFormat: 'yy-mm-dd'});

$("input#edate").datepicker({dateFormat: 'yy-mm-dd'});

});

</script>

<?php

if (!defined('WEB\_ROOT')) {

exit;

}

$errorMessage = (isset($\_GET['error']) && $\_GET['error'] != '') ? $\_GET['error'] : '&nbsp;';

$sql\_v = "SELECT id, vname FROM tbl\_vendors";

$res\_v = dbQuery($sql\_v);

$sql\_c = "SELECT cid, cname, ctype FROM tbl\_categories";

$res\_c = dbQuery($sql\_c);

?>

<div class="prepend-2 span-13 append-4">

<p>&nbsp;</p>

<p class="errorMessage"><?php echo $errorMessage; ?></p>

<form action="process.php?action=search" method="post" name="frmListUser" id="frmListUser" style="border-radius:10px; border:1px solid #0000FF; padding:20px;">

<p><img src="../images/search.png" />

<strong>Search Form</strong>

</p>

<hr/>

<table width="500" border="1" cellpadding="0" cellspacing="0">

<tr>

<td width="33%">&nbsp;</td>

<td width="6%">&nbsp;</td>

<td width="61%">&nbsp;</td>

</tr>

<tr>

<td>Asset Name </td>

<td>&nbsp;</td>

<td><label>

<input name="name" type="text" id="name" />

</label></td>

</tr>

<tr>

<td>Asset Type </td>

<td>:</td>

<td><label>

<select name="type">

<option value="1">Hardware</option>

<option value="2">Software</option>

</select>

</label></td>

</tr>

<tr>

<td>Vendor Name </td>

<td>:</td>

<td>

<select name="vid">

<?php

while($v = dbFetchAssoc($res\_v)){

?>

<option value="<?php echo $v['id']; ?>"><?php echo $v['vname']; ?></option>

<?php

}

?>

</select> </td>

</tr>

<tr>

<td>Category</td>

<td>:</td>

<td>

<select name="vid">

<?php

while($c = dbFetchAssoc($res\_c)){

?>

<option value="<?php echo $c['cid']; ?>"><?php echo $c['cname']. ' -> '. $c['ctype']; ?></option>

<?php

}

?>

</select>

</td>

</tr>

<tr>

<td>Start Date </td>

<td>:</td>

<td><label>

<input type="text" id="sdate" name="sdate" readonly="true" />

</label></td>

</tr>

<tr>

<td>End Date </td>

<td>:</td>

<td><label>

<input type="text" name="edate" id="edate" readonly="true"/>

</label></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td><input type="submit" name="Submit" value=" Search Now" class="button"/></td>

</tr>

</table>

</form>

<p>&nbsp;</p>

<p>&nbsp;</p>

<?php

if(isset($\_SESSION["result\_data"]) && count($\_SESSION["result\_data"]) > 0) {

echo "<pre>";

//print\_r($\_SESSION["result\_data"]);

echo "</pre>";

?>

<table border="0" align="center" cellpadding="2" cellspacing="1" class="text">

<tr align="center" id="listTableHeader">

<td>Hardware</td>

<td>Vendor Name</td>

<td>Vendor</td>

<td>Category</td>

<td>D.O.A</td>

<td>Type</td>

</tr>

<?php

for($i = 0; $i<count($\_SESSION["result\_data"]); $i++) {

extract($\_SESSION["result\_data"][$i]);

if ($i%2) {

$class = 'row1';

} else {

$class = 'row2';

}

if($thumb) {$img = WEB\_ROOT . "images/vendors/".$thumb;}

else {$img = "images/no-image-small.png";}

//$i += 1;

?>

<tr class="<?php echo $class; ?>">

<td><?php echo $name; ?></td>

<td align="center"><?php echo $vname; ?></td>

<td align="center">

<img src="<?php echo $img;?>" title="<?php echo $vname; ?>" /></td>

<td align="center"><?php echo $cname; ?></td>

<td align="center"><?php echo $dop; ?></td>

<td align="center"><?php echo $cname; ?></td>

</tr>

<?php

} // end for

?>

</table>

<?php

}//if

unset($\_SESSION["result\_data"]);

?>

</div>

**DATABASE.PHP**

<?php

require\_once 'config.php';

$dbConn = mysql\_connect ($dbHost, $dbUser, $dbPass) or die ('MySQL connect failed. ' . mysql\_error());

mysql\_select\_db($dbName) or die('Cannot select database. ' . mysql\_error());

function dbQuery($sql)

{

$result = mysql\_query($sql) or die(mysql\_error());

return $result;

}

function dbAffectedRows()

{

global $dbConn;

return mysql\_affected\_rows($dbConn);

}

function dbFetchArray($result, $resultType = MYSQL\_NUM) {

return mysql\_fetch\_array($result, $resultType);

}

function dbFetchAssoc($result)

{

return mysql\_fetch\_assoc($result);

}

function dbFetchRow($result)

{

return mysql\_fetch\_row($result);

}

function dbFreeResult($result)

{

return mysql\_free\_result($result);

}

function dbNumRows($result)

{

return mysql\_num\_rows($result);

}

function dbSelect($dbName)

{

return mysql\_select\_db($dbName);

}

function dbInsertId()

{

return mysql\_insert\_id();

}

?>

**DATABASE/CONFIG.PHP**

<?php

ini\_set('display\_errors', 'On');

//ob\_start("ob\_gzhandler");

error\_reporting(E\_ALL);

// start the session

session\_start();

// database connection config

$dbHost = 'localhost';

$dbUser = 'root';

$dbPass = '';

$dbName = 'db\_myasset';

// setting up the web root and server root for

// this shopping cart application

$thisFile = str\_replace('\\', '/', \_\_FILE\_\_);

$docRoot = $\_SERVER['DOCUMENT\_ROOT'];

$webRoot = str\_replace(array($docRoot, 'library/config.php'), '', $thisFile);

$srvRoot = str\_replace('library/config.php', '', $thisFile);

define('WEB\_ROOT', $webRoot);

define('SRV\_ROOT', $srvRoot);

// these are the directories where we will store all

// category and product images

define('CATEGORY\_IMAGE\_DIR', 'images/category/');

define('PRODUCT\_IMAGE\_DIR', 'images/product/');

// some size limitation for the category

// and product images

// all category image width must not

// exceed 75 pixels

define('MAX\_CATEGORY\_IMAGE\_WIDTH', 75);

// do we need to limit the product image width?

// setting this value to 'true' is recommended

define('LIMIT\_PRODUCT\_WIDTH', true);

// maximum width for all product image

define('MAX\_PRODUCT\_IMAGE\_WIDTH', 300);

// the width for product thumbnail

define('THUMBNAIL\_WIDTH', 75);

if (!get\_magic\_quotes\_gpc()) {

if (isset($\_POST)) {

foreach ($\_POST as $key => $value) {

$\_POST[$key] = trim(addslashes($value));

}

}

if (isset($\_GET)) {

foreach ($\_GET as $key => $value) {

$\_GET[$key] = trim(addslashes($value));

}

}

}

// since all page will require a database access

// and the common library is also used by all

// it's logical to load these library here

require\_once 'database.php';

require\_once 'common.php';

?>

**HARDWARE.PHP**

<?php

require\_once '../library/config.php';

require\_once '../library/functions.php';

$\_SESSION['login\_return\_url'] = $\_SERVER['REQUEST\_URI'];

checkUser();

$view = (isset($\_GET['view']) && $\_GET['view'] != '') ? $\_GET['view'] : '';

switch ($view) {

case 'list' :

$content = 'list.php';

$pageTitle = 'Shop Admin Control Panel - View Users';

break;

case 'add' :

$content = 'add.php';

$pageTitle = 'Shop Admin Control Panel - Add Users';

break;

case 'modify' :

$content = 'modify.php';

$pageTitle = 'Shop Admin Control Panel - Modify Users';

break;

default :

$content = 'list.php';

$pageTitle = 'Shop Admin Control Panel - View Users';

}

$script = array('hardware.js');

require\_once 'template.php';

?>