



Development of a SaaS Inventory Management System

Antonis Loizides

Bachelor's thesis of the Degree Programme in Business Information Technology

Bachelor of Business Administration

TORNIO 2013

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Degree Programme: Business Information Technology
Writer: Antonis Loizides
Thesis title: Development of a SaaS Inventory Management System
Pages (of which appendices): 63 (15)
Date: 20.04.2013
Thesis instructor: Aalto Teppo
<p>The objective of this research is to develop a Software as a Service web application and more specifically an Inventory Management System for Caterpro Ltd. The aim of this web application is to provide the basic tool for tracking as well as monitoring sales and inventory to individuals and small businesses who cannot afford the investment of a complete dedicated Inventory Management System. Moreover, this research will used for various purposes all the necessary components as well as steps required for the proper implementation of Software as a Service in a web-based environment.</p> <p>The importance of this research will be the thorough description and analysis of the basic and necessary functionalities of an Inventory Management System web application. In addition, the cornerstone of the research will be the development of the web application and implementation of different techniques and functions for basic security and encryption.</p> <p>The selected methodology of this research will be the constructive method, the most suitable for IT industry and especially for software engineering researches. This method will be used as the primary research methodology for finding theoretically and practically solutions to an existing problem.</p> <p>The result and expected output of this research is the development of an Online Inventory Management System with the basic functionalities. The basic functions will be a secure registration and login system, inventory's items, users, orders, suppliers, customers, history-logs and company configuration modules. In additions to that, there will be more advanced functions such as automatic email notification, Order's pdf generation, export MySQL data to Excel file and a lot more.</p> <p>Keywords: HTML, CSS, PHP, MySQL, Linux, Apache, Security, Encryption</p>

ABBREVIATIONS

HTML	Hypertext markup language
CSS	Cascade style sheets
JS	JavaScript
jQuery	JavaScript libraries
AJAX	Asynchronous JavaScript and XML
PHP	PHP Hypertext Preprocessor
MySQL	Relational Database Management System
Linux	Open Source Unix-like Operating System
Apache	Linux HTTP Server
LAMP	Linux, Apache, MySQL and PHP.
UML	Unified Modeling Language
SSL	Secure Socket Layer
IMS	Inventory Management System
MD5	Message-Digest Algorithm 5
Crypt	Unix-like encryption algorithm
SaaS	Software as a Service
SSH	Secure Shell
HTTP	Hypertext Transport Protocol
cPanel	Unix-based Graphical User Interface Control Panel
RDBMS	Relational Database Management System
Logiwan IMS	Web Application Brand Name
MS Excel	Microsoft Spreadsheet Application
GPL License	General Public license
PDF	Portable Document Format

FIGURES

Figure 1. Elements of constructive research	12
Figure 2. Web application structure	17
Figure 3. Login - backend administration	19
Figure 4. Login - web application	21
Figure 5. Inventory modules	23
Figure 6. Administer and monitor users.....	25
Figure 7. User architecture of the web application	26
Figure 8. MySQL ERD	29
Figure 9. MySQL ERD description.....	30
Figure 10. MySQL ERD relationships.....	31
Figure 11. System settings	32
Figure 12. phpDesigner SFTP setup	33
Figure 13. Backend administrator cPanel	34
Figure 14. PHP version	35
Figure 15. Register globals.....	36
Figure 16. Log errors.....	36
Figure 17. Root directory pointer.....	37
Figure 18. .htaccess file.....	37
Figure 19. .htpasswd file	38
Figure 20. Main homepage.....	39
Figure 21. Demo site	40
Figure 22. Account page	41
Figure 23. Web application login page	42
Figure 24. Web application login page 2	43
Figure 25. Web application homepage.....	44
Figure 26. Items module.....	44
Figure 27. Input sanitization	45
Figure 28. Input validation	46
Figure 29. Database login query	46
Figure 30. Password hash.....	47

Figure 31. Login script.....	48
Figure 32. Session security	50
Figure 33. Logout function.....	51
Figure 34. Mail function.....	53

CONTENTS

ABSTRACT

ABBREVIATIONS

FIGURES

ABSTRACT	2
ABBREVIATIONS	3
FIGURES	4
CONTENTS	6
1 INTRODUCTION AND MOTIVATION	8
1.1 Background information and motivation	8
1.2 Thesis structure	10
2 Research questions	11
3 METHODOLOGY	12
3.1 Research methodology	12
3.2 Development details	13
4 DESIGN AND MODELING	17
4.1 Purpose	17
4.2 Main functions of the application	18
4.2.1 Back-end administrator login	18
4.2.2 User account login	20
4.2.3 Inventory modules	22
4.2.4 Administer and monitor users	24
4.3 User Architecture of the web application	26
4.4 Database model	28
4.4.1 MySQL ERD	28
4.4.2 MySQL ERD description	30
4.4.3 MySQL ERD relationships	31
5 DEVELOPMENT CONFIGURATIONS	32
5.1 System settings	32
5.2 Basic development configurations	33
6 TECHNICAL SPECIFICATIONS AND FUNCTIONALITIES	39

6.1 User interface	39
6.2 Scripting	45
6.2.1 Input validation and sanitization	45
6.2.2 Password hashing and salting	47
6.2.3 Login system	48
6.2.4 Session management security	49
6.2.5 Email notifications	52
7 CONCLUSION	54
REFERENCES.....	56
APPENDICES	59
APPENDIX 1	59
APPENDIX 2.....	62

1 INTRODUCTION AND MOTIVATION

1.1 Background information and motivation

SaaS stands for “Software as a Service” and as the name clearly defines it is a software distribution model in which software and applications hosted by the provider and are available to customers over the Internet (Margaret Rouse 2010). The concept is the same as people using Google’s free email and storage services. Entire businesses and employees run their critical applications through centralized computing platforms over the Internet. There is no need for customers to purchase any expensive licensed applications because Software as a Service prices are based on an affordable monthly or annual fee. In addition, the customers do not need to invest in additional or expensive hardware because the application is hosted remotely. Moreover, Software as a Service removes the need for organizations to handle the installation, configuration and daily maintenance of the software application and hardware resources. A client application, commonly a web browser, gives the ability to the customer to access any software application resources securely over the Internet through the available broadband connectivity. Software as a Service is related to software delivery models as well as on demand and cloud computing.

The most important aspects of Software as a Service are the availability, security, automatic patch management and global accessibility. Business-critical applications need to be available between 99.5% and 99.9% (Liz Herbert 2011). The applications must be available to the customer at any time, in order to complete a specific task without any delay. Time is money and therefore the provider of the applications and services must provide redundant hardware devices and access links in order to decrease the downtime in case of failure. The Software as a Service platforms must be designed and planned for high availability. Security is the cornerstone of the IT Industry and especially for any service or applications that accesses the Internet. Since Software as a Service applications are hosted on the provider’s datacenter, the customer uses the Internet as the only pathway in order to access them. The application must use only encrypted sessions/tunnels to communicate with the customer’s client software and provide the strongest one-way password hashing and salting algorithms. The customer’s

data stored in the cloud, must be secured and encrypted. Automatic patch management is the process of applying fixes and patches to the applications and monitor the activity and health of the system. This procedure it is done automatically by the provider and the customer can focus only in the business activities without worrying for updates, troubleshooting and maintenance. Global accessibility identifies the access of the Software as a Service hosted applications regardless of the physical location of the user. While the customer travels, the important and mission-critical applications of the company will be available through the Internet.

According to Gartner Research, the Software as a Service industry was increased by 17.9% from 2011 to 2012. Additionally, the market growth will continue and the Software as a Service sales will reach \$22.1 billion in 2015 (Gartner 2012).

The motivation behind this research is due to the personal and professional interests and the topic which is mostly related to my study field. The company named Caterpro Ltd was selected as my case company. Caterpro Ltd wanted to expand its business into the Software as a Service industry and therefore the founder assigned me to develop a Software as a Service Inventory Management System. The web application should be suitable for individuals and small businesses users. The objective of this research is to focus on the development and deployment of a Software as a Service application which will serve as stock control and Inventory Management System for Caterpro Ltd.

IMS (Inventory Management System) is a software or application that tracks and monitors sales and available inventory of a business. The Inventory Management System is usually one of the most important and biggest financial investments a business must make (Rosemary Peavler 2012). An Inventory Management System can provide simple functions such as sales, stock control, as well as advanced functions such as barcode tracking. The most commonly used functions for an Inventory Management System enable the user for example to add, remove and modify inventory items, manage customers, suppliers and users, place and track orders etc. Customers, who provide products and services, must use an Inventory Management System in order to have direct access to the status of the stock and profit and loss.

The most fundamental aspect of this Inventory Management System in question is to provide a fully dynamic web application where the user must be able to change the content of the web application. Usability, on the other hand, is another important factor for keeping the users satisfied, by developing a user friendly interface in order to allow the use to complete any task with as few clicks as possible. The users of the web application must also be able to adapt easily. If the users have difficulties to familiarize themselves with the web application and must use the contact form for support and guidelines, then the usability of the web application has failed.

The procedure of the account setup will be as follows: At the first stage, the administrator of the web application creates an account in the back-end administration panel and a confirmation email is sent to the customer's email. Second, the customer must use the unique link in order to validate and confirm that the provided email address is existing and validate the identity of the user too. Third, the user uses the provided username and password to login to the web application and begun building the inventory.

1.2 Thesis structure

The thesis is divided into six chapters. Chapter 1 includes information about the case company, motivation and general background information. Chapter 2 explains 3 research questions. Chapter 3 contains the methodology of the thesis and the development details. Chapter 4 is based on UML modeling with the basic classes, functions diagrams and Database model. Chapter 5 describes the development configurations such as the configuration of the development environment. Chapter 6 contains the technical specifications and functionalities of the web application including user interface and scripting. Chapter 7 is the final chapter with the conclusion of the thesis, suggestions and future improvement of the web application.

2 RESEARCH QUESTIONS

Based on the objective of this research, there are numerous questions that must be addressed and answered.

1. What are the security features of the Logiwan IMS web application?

The customers must be able to establish a secure and encrypted connection with the web-based Inventory Management System and store an encrypted format of their information, in the database. This research demonstrates a collection of the security measures taken for the encryption and validation processes, such as input sanitization, session management, password hashing and salting, etc.

2. What are the functionalities of the Inventory Management System?

The specifications requirements document of the Inventory Management System was provided by Caterpro Ltd, in order to meet their customer's needs. The Inventory Management System will focus on individuals and small businesses' users. The primary use for the Inventory Management System is to track and monitor sales and available inventory of a business. Moreover, the functionalities needed by Caterpro Ltd are "Items", "Orders", "Suppliers", "Customers", "Users", "History/Log" and "Company Configuration". The functionalities are similar to the modules and categories of a software system. For example, in the "Items" module the user will be able to add, modify and remove any item listed in the inventory. "Item" refers to a product, spare part or a service. Moreover, an Administrator back-end module must be developed for the management of the accounts and web-application.

3. What are the development details of the Inventory Management System?

The development process focuses in Open-Source Web Development. Open source web development defines the usage of free-of-charge programming database languages as well as server side script. The hosting provider and domain registrar is Bluehost and the chosen domain name is www.logiwan.com.

3 METHODOLOGY

3.1 Research methodology

The research methodology that was selected for this research is constructive methodology, because this research is a case study research. Constructive methodology's research objective is to produce novel solutions to practical and theoretical relevant problems. This research methodology is the most suitable option and most commonly used methodology in IT industry and especially in software engineering. Figure 1 below illustrates the elements of the constructive research methodology.

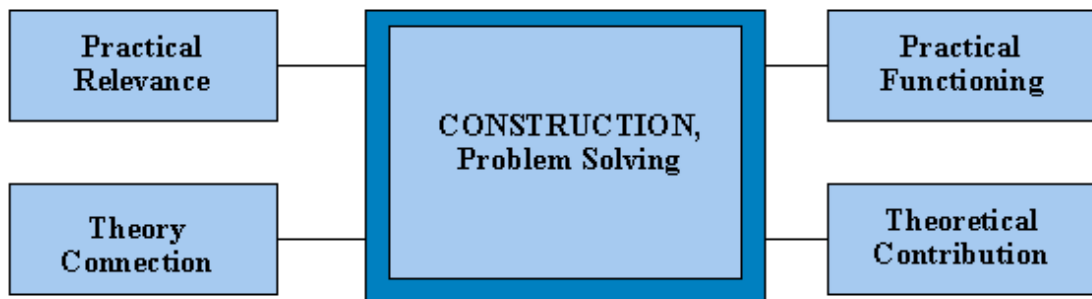


Figure 1. Elements of constructive research (Kasanen & Lukka & Siitonen 1993, 246)

The above figure identifies the different steps and elements in constructive research. As the first step, a practical relevant problem must be identified with possibilities for research. As a second step, it is a major importance to obtain a general and detailed understanding of the selected topic and construct a contemporary solution. After that, the solution must be demonstrated with an optimal functionality. Finally, the theoretical connections and the research contribution of the solution must be shown and constructed in such a way to promote the problem solving method (Kasanen & Lukka & Siitonen 1993, 243-264).

Caterpro Ltd business' expansion in the Software as a Service industry constitutes a practical problem which is intended to be solved with this research. Moreover, the company has many customers, individuals and small businesses, that cannot afford to invest in expensive IT equipment, infrastructure and software licenses for a dedicated Inventory Management System solution. The research product that will solve the problem of Caterpro Ltd is an Online Inventory Management System.

3.2 Development details

For compatibility and easier troubleshooting purposes, it is proposed and recommend that the web application must be developed in the actual production environment that will be used for deploying and running the web application. Different environments have different requirements and versions of Apache HTTP server, PHP Interpreter and MySQL. Also, the environment varies from different platforms and operating systems, especially in Windows operating systems and the functions or configurations may not exist in newer or older versions. The production environment is hosted on a Linux server which is provided by Bluehost, under the domain name www.logiwan.com. The web server is an Apache HTTP server, version 2.2.23, with PHP version 5.4.7 and MySQL version 5.1.66. During the development process, the access to the web applications will be restricted to the public and will be accessible only to certain IP addresses. Moreover, phpDesigner 8 will be the IDE connected to the remote Linux server over the Internet with the use of SSH encrypted tunnels. phpDesigner 8 is the development software for developing HTML, CSS, JS, PHP and .htaccess files for the web application. MySQL Workbench version 5.2 will be used for the design, development and administration of the MySQL database.

The following information describes the technologies, techniques and scripting languages used for the development of the web application.

HTML

HTML is defined as Hypertext Markup Language and is the main markup language for displaying web pages and other information than can be displayed in a web browser (Kyrnin 2013b). HTML is commonly used for the content of web pages and can include images, super links, sounds and videos. Internet browser is the client that converts the syntax of HTML elements into viewable objects.

CSS

CSS is defined as Cascading Style Sheets and is a style sheet used for describing the presentation semantics, layout and formatting of a document written in a markup language (Kyrnin 2013a). CSS is the style sheet language that can be used to align the position and format at any HTML element.

JavaScript

JavaScript is a scripting language commonly implemented as part of a web browser in order to create enhanced user interfaces and dynamic websites (Stephen Chapman 2013). JavaScript commonly refers to a client-side script and enables interactive functions to be added to the web pages.

jQuery

jQuery is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML (Sheo Narayan 2013). jQuery can be used in order to handle events, navigate HTML documents and perform different AJAX programming functions.

AJAX

AJAX is defined as Asynchronous JavaScript & XML and is a group of interrelated techniques used on the client-side to create asynchronous web-applications (Kyrnin 2013c). Furthermore, AJAX is executing like a remote scripting, which allows web application to run different functions behind the scenes and update the web pages immediately and automatically.

PHP

PHP is defined as PHP Hypertext Preprocessor and is an open source general-purpose server-side scripting language originally designed for web development to produce dynamic web pages (Angela Bradley 2013). Moreover, it was the first server-side scripting language that was specifically designed to be embedded into HTML elements for server-side processing and execution.

MySQL

MySQL is the most popular and commonly used open source relational database management system, which operates as a server providing multi-user access to a number of databases (MySQL.com 2013). Most of the applications using MySQL are written in PHP.

SSL Certificate

SSL Certificate defines as Secure Socket Layer Certificate and is a cryptographic protocol that provides communication security over the Internet. SSL Certificate encrypts the segments of network connections at the Application Layer for the Transport Layer (Instantssl.com 2013). SSL Certificate is used for establishing a secure and encrypted connection between a server and a client; using the HTTPS protocol and by default, the port 443. Also, SSL Certificate validates the identity of the web page.

LAMP

LAMP acronym stands for Linux operating system, Apache HTTP server, MySQL RDBMS and PHP server-side script which provides a software solution stack of free open source software (Martin Brown 2013). They are the necessary components and common sets of system software and scripts used to build a Linux web-based server. LAMP environments can be used for production as well as for development processes.

phpDesigner 8

phpDesigner is an IDE (Integrated Development Environment), a fully-featured HTML, CSS, JS, MySQL and PHP editor. phpDesigner can also analyze and debug PHP 5+ and MySQL code and syntax. phpDesigner 8 is offered under a commercial license.

cPanel

cPanel is a Unix-based web hosting control panel that provides a graphical user interface and automation tools to simplify the process of hosting and managing a web site (cPanel.com 2013).

4 DESIGN AND MODELING

4.1 Purpose

In order to complete the web application design and modelling process, technical specifications and characteristics must be provided. The technical specifications documentation is considered crucial and provides a solid foundation for the development process. The analysis of the system must be made in a way that allows for incorporating all the necessary configurations and coding for the development process.

As previously discussed in Chapter 3, the necessary technologies needed for the web application design, content and presentation semantics are HTML and CSS. JavaScript and AJAX technologies are used for the client-side scripts. The server-side script is PHP and the Relational Database Management System is MySQL. All the components and technologies are running on an Apache Linux HTTP Server. The below figure is a graphical representation of the web application structure.

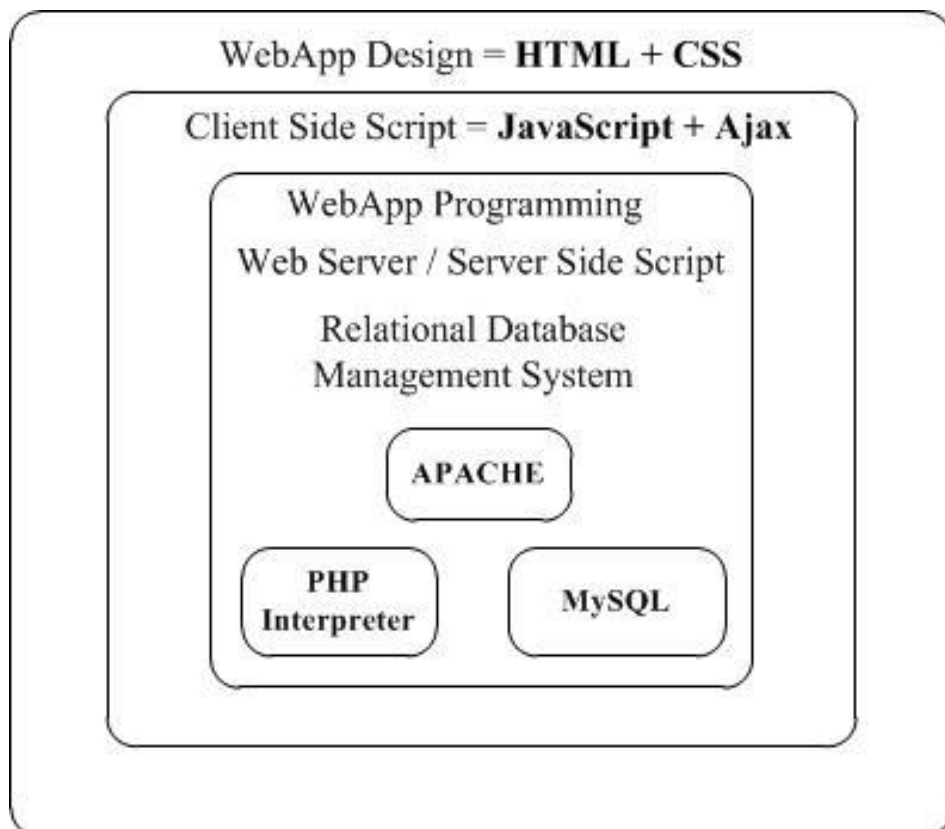


Figure 2. Web application structure

4.2 Main functions of the application

In order to present virtually the procedural flow of actions, several activity diagrams must be designed. Activity Diagrams are graphical representations of activities and actions that show the continuous usage of different actors occur in the system. The use cases of the Activity Diagrams presented are for Login, Items, Orders, Customers, Suppliers, Users and History of Logiwan IMS web application. All the above use cases are called modules, and constitute the components of the web application and represent the different functionalities of the system.

The purpose of the use cases modules is to give a graphical representation of the actions that will be occurring between the system, the database and the user. The messages of representation facilitate the processes of understanding all actions, i.e. add, modify, delete and convert them into functions and server-side scripting.

4.2.1 Back-end administrator login

The purpose for the below activity diagram is to provide authorization and verification to the back-end Administrator of the web application, via the Apache HTTP authentication module and proceed to the back-end administration login page. Apache HTTP Authentication is an additional solid security feature of the Apache web server that prevents unauthorized access to specific content and directories. Username and password will be shared among Caterpro Ltd employees.

As part of the authorization and verification procedure, the user's browser sends a request to the domain name of the web application at <https://logiwan.com/admin-cpanel>, and the user receives the "Authentication Required" window. If the user entered an invalid username or password, the web server returns a "401 Authorization Required" error and prevents the user to access the content in the specific directory. On the other hand, if the user entered a valid username and password, the web server validates that the user is an authorized person of Caterpro Ltd and allows the access to the content of the directory.

At this point, it is important to note that the user will be prompted to provide his private credentials in order to be able to access the back-end administration panel. This procedure identifies each employee with the username, full name, last login and every action registered on the specific account. Figure 3 illustrates the process of the backend administration panel.

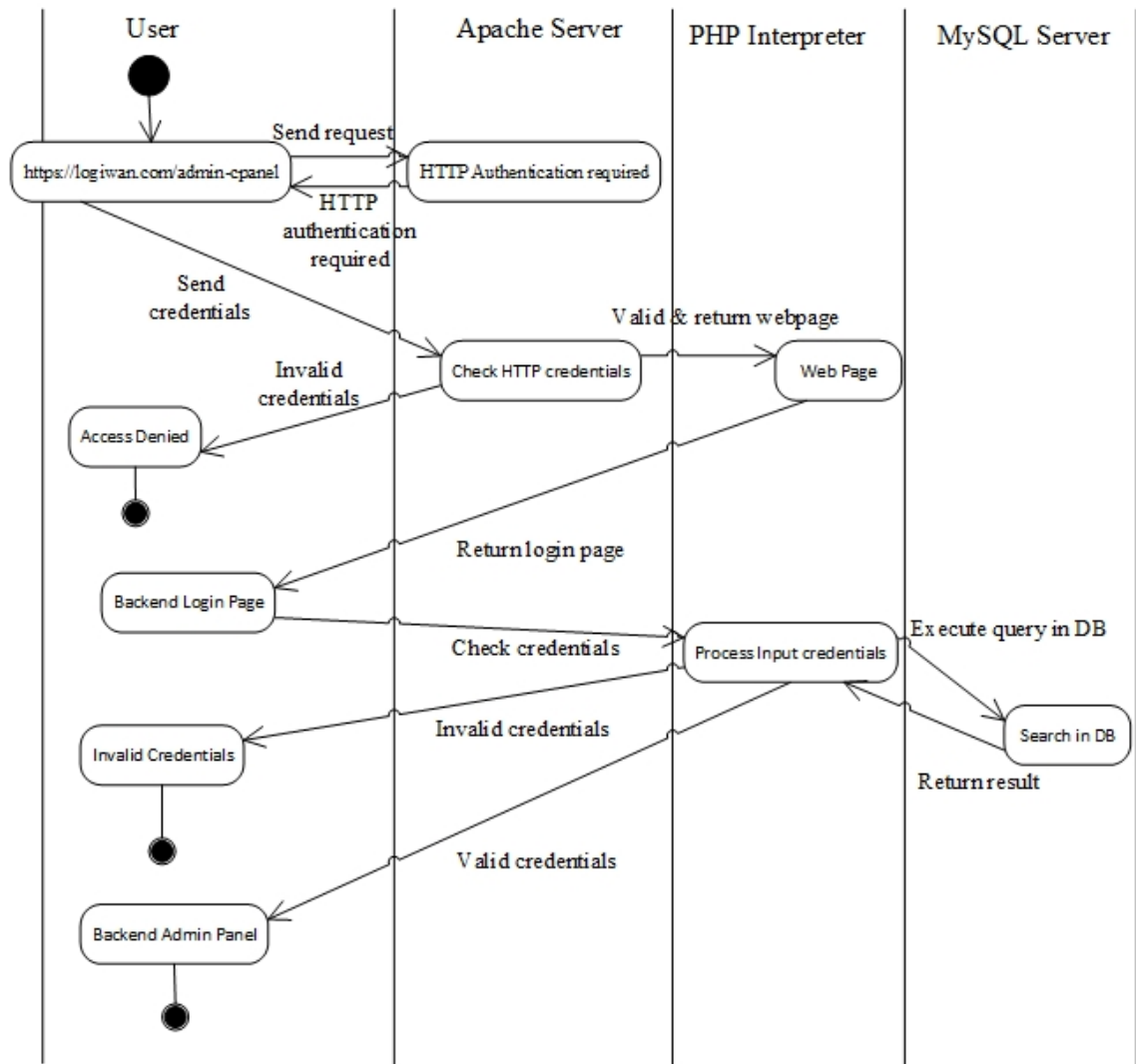


Figure 3. Login - backend administration panel

4.2.2 User account login

The purpose of the below Activity Diagram, is to authorize the user to log in to the inventory. The email of each account provides a unique identification field in the database in order for the user to login to his company's inventory.

In the login page there are three fields; account's email, username and password. Account's email is the field that receives the account's email registered in the web application and identifies the specific inventory. Username and password are the fields that identify each user/employee of the company's account; these fields are disabled before any valid account's email is provided. After a valid email is provided, username and password fields are enabled automatically and the users have to enter their own private username and password credentials. Once a valid credentials set is provided, the system query the database to find if the company's account has any user with the specific credentials associated with it. Then the user is redirected to the homepage of the inventory and a session is generated and track the user until the log out process. Further, the login system recognize if the user account has administrator or user rights of the inventory and load the recommended menu. Figure 4 presents the process of the login system at the inventory.

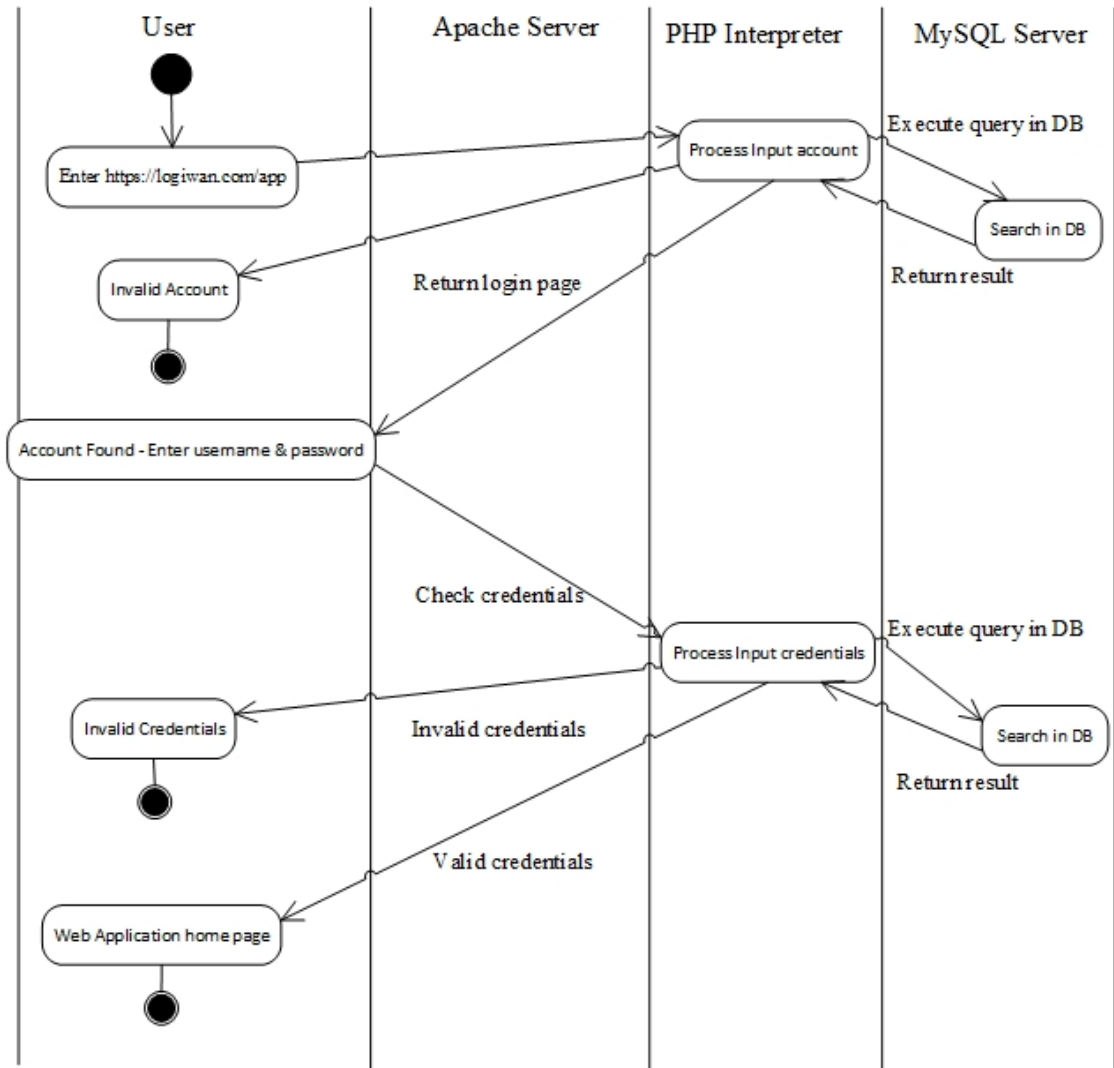


Figure 4. Login - web application

4.2.3 Inventory modules

The purpose of the inventory modules use case is to demonstrate that the user is able to Add, Modify and Delete any entry of Items, Orders, Customers, Suppliers and Users associated with the company's inventory.

The procedure for the modules described above, remains the same with every module; the first action is "Insert" and the user enters the information for a new entry in the form of any module. As a second step, a custom PHP function is called in order to check for any possible empty fields. In case a field is empty, an error message is generated and returned. When all the fields are filled-in, another custom PHP function is called to validate the information for the recommended format. For example, in the "quantity" field the user can only type in an integer, if the user types in any alphabetical character the validation fails and returns an error message.

The second action is "Modify". A table with a MySQL query fetch all the non-deleted entries of the selected module. A small pencil icon on every row provides the link to trigger the function that loads all the data of the selected entry in the form for modifications. After submitting the modified data, the PHP validation function validate all the fields.

The third action is "Delete". In the loaded entries of the table there is a small "x" icon which provides the link to trigger the function for deletion of the selected entry. Upon the deletion of any entry, a pop-up window is generated, containing a confirmation message which prompts the user to confirm the action. Figure 5 represents the functionality of the inventory modules.

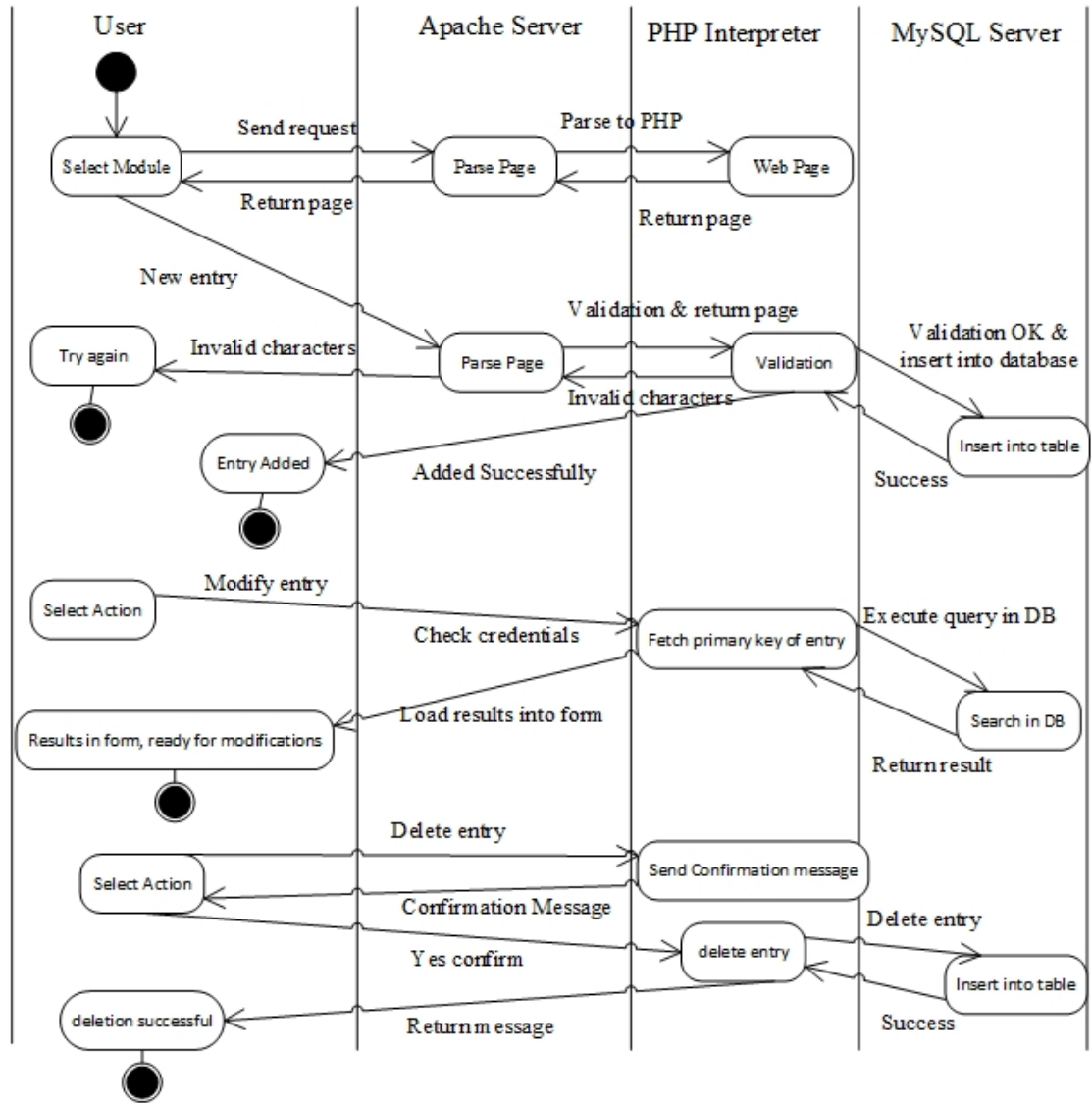


Figure 5. Inventory modules

4.2.4 Administer and monitor users

The administrator and user accounts will have the same functionality modules with the only difference that administrators will have the privilege to modify the user accounts that are associated with the company's inventory. Additionally, the administrator has access to monitor all the actions taken to any of the inventory modules through "History" module.

The "Users" module has the same functionality as the Items, Orders, Customers and Suppliers modules. The only difference is that only the administrator can access it. The administrator executes the same actions as insert, modify or delete but there are some limitations. For example, the username of the account must be only letters and numbers without any spaces. Also, the administrator cannot delete the last administrator of the Inventory and of course cannot delete the user account that is currently logged in.

"History" module are simple tables with database queries that store every action taken in any of the modules. "History" module functionality is the same as the log files in every software or operating system because it records every action taken in any of the modules. For example, when a user inserts a new item in the "Items" module, a function insert duplicate data in "items_history" table. The action is listed in the Items History with five different values. Those values identifies the user that executed the action, the item number, the quantity, date time and action's description. Moreover, the action column can take three different values; "New" for a new entry, "Modified" for an existing entry that has been modified and "Deleted" for an existing entry that has been deleted. The data stored in the "History" module cannot be deleted by the Administrator of the Inventory but only from the account holder at www.logiwan.com/account. Figure 6 illustrates the process of administer and monitor users among with the different tools of the inventory application.

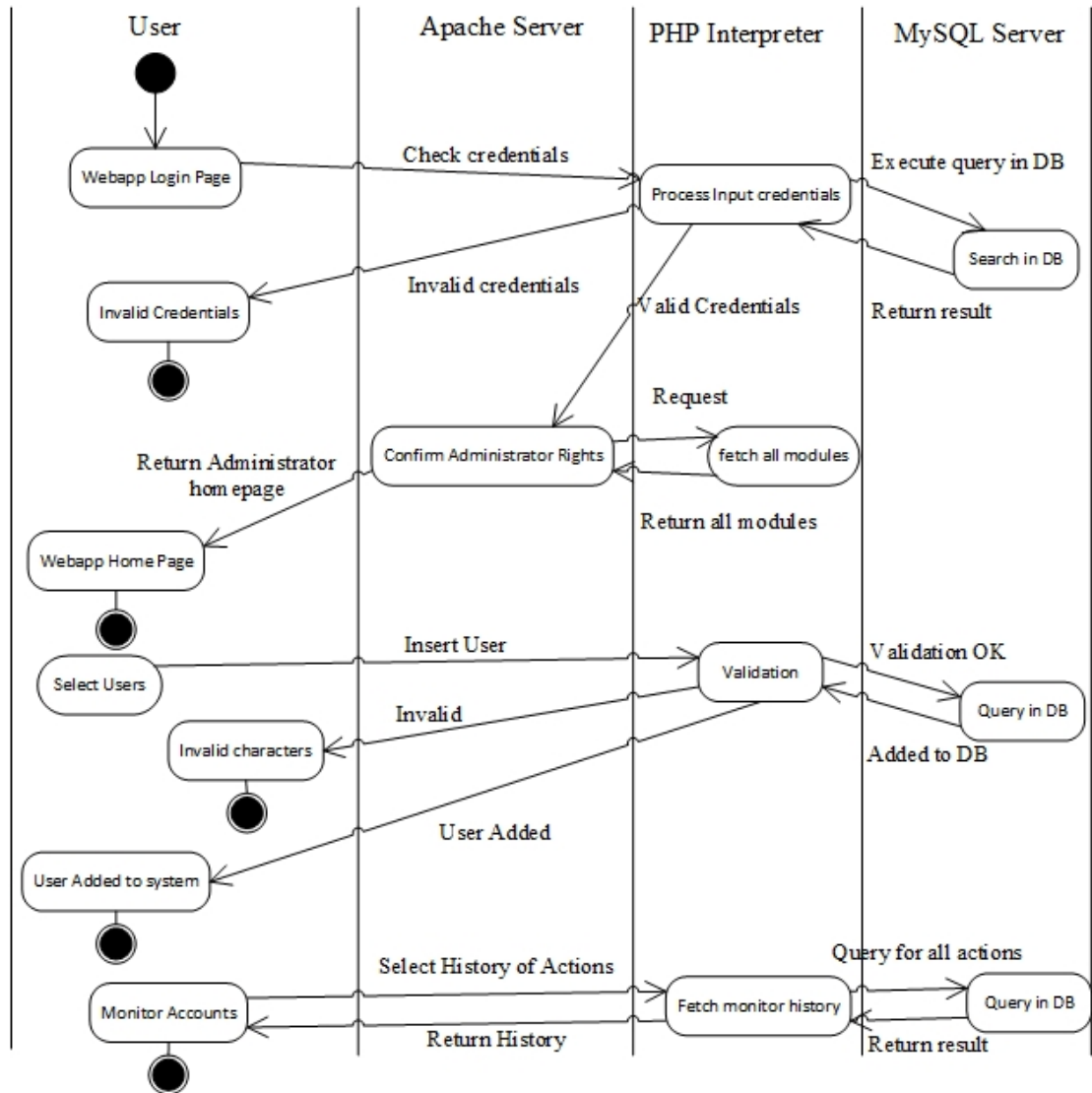


Figure 6. Administer and monitor users

4.3 User Architecture of the web application

In the figure below, the use case diagram represents the functionality provided by the web application. The main purpose of the use case diagram is to help in the development procedure to visualize the functional requirements of the web application. Moreover, the use case figure is an essential graphical model because it defines the interaction between the system; web application, and the actor; user, in order to achieve a specific goal.

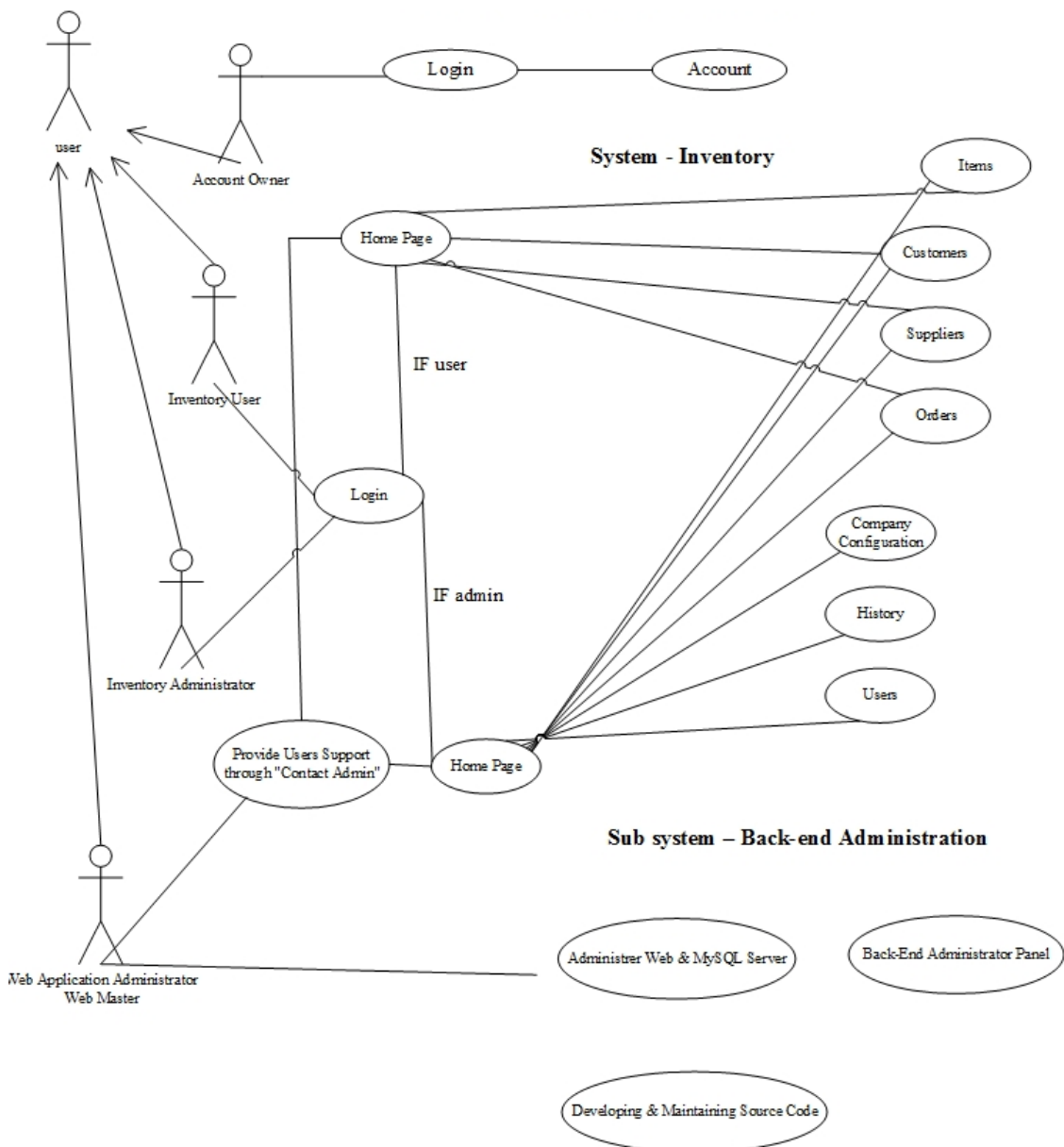


Figure 7. User Architecture of the web application

A more detailed description of the users among their functionalities is below.

Webmaster

The person who is responsible for support and maintenance for functionality, services and technical programming aspects of the web application. The application's functionalities are defined as the procedure for developing and maintaining the source code and the administration of the web and database server. The webmaster is also responsible for the back-end administration of the Inventory Management System application as well as for the creation, monitoring, suspension and deletion of the accounts.

Account Holder

The account holder is a company or an individual who has an account in the Inventory Management System application. The account can be only accessed through www.logiwan.com/account and allows the account holder to modify the company name, email, account holder name, clear the "History" logs data as well as close the account.

Inventory Administrator

As the name clearly defines, the inventory administrator is responsible for the administration of the company's inventory which can be accessed through www.logiwan.com/app . The administrator has access to all the modules of the inventory of the web application.

Inventory User

The inventory user is in control of the inventory which can be accessed through www.logiwan.com/app . The user cannot access the "User", "History" and "Company" modules due to the lack of administrative rights.

4.4 Database model

MySQL is one of the most popular RDBMS for both web and embedded systems and is a central component of the LAMP open source web application software pack.

Relational Database refers to the set of separate files, called tables and combines data elements from the files for queries and reports when required (Andy Opperl 2005).

Moreover, it has the flexibility to “join” two or more tables by comparing their primary keys fields such as “ID” and generating a new table fields from records that meet the matching criteria. One of the most important functions of Relational Database Query is the indexes. In practical related task, the use of a non-optimal database query, can be very slow. However, for speeding up the process, indexes are created “on the fly”; during the process, when the data are requested.

4.4.1 MySQL ERD

ERD defines an Entity Relationship Diagram and describes the entities, attributes and relationships between them. In an ERD model, a table filled with attributes is called an entity. The creation of the ERD is a graphical representation of the database structure. MySQL Workbench is a professional graphical user interface tool that is required to create the model of the MySQL database as well as administer the databases and develop SQL code (Mike Chapple 2012). Figure 8 represents the ERD of the web application database.

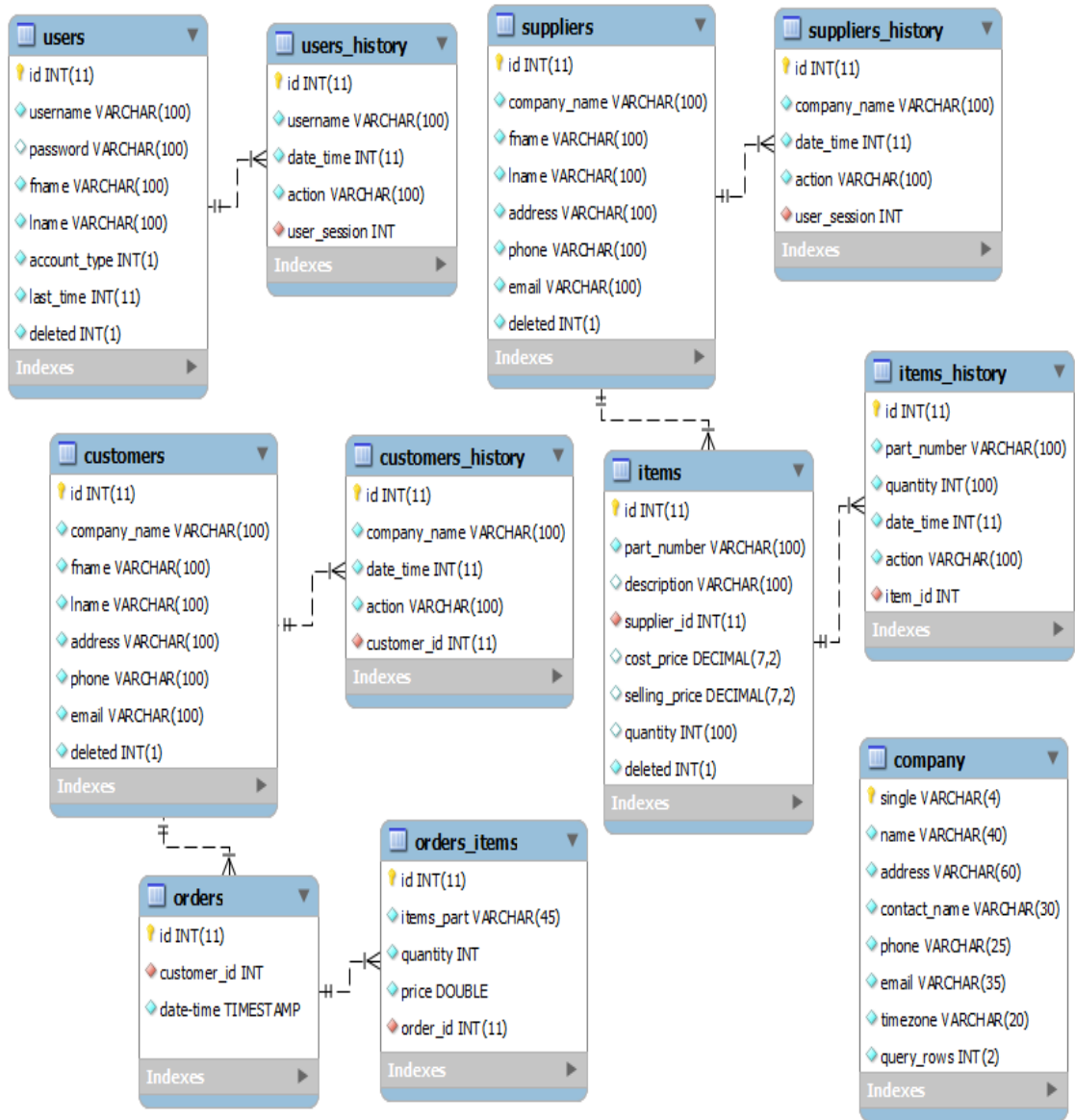


Figure 8. MySQL ERD

4.4.2 MySQL ERD description

The following ERD Model depicted in Figure 9 is a sample of the description of the modeling and relationships between the database tables.

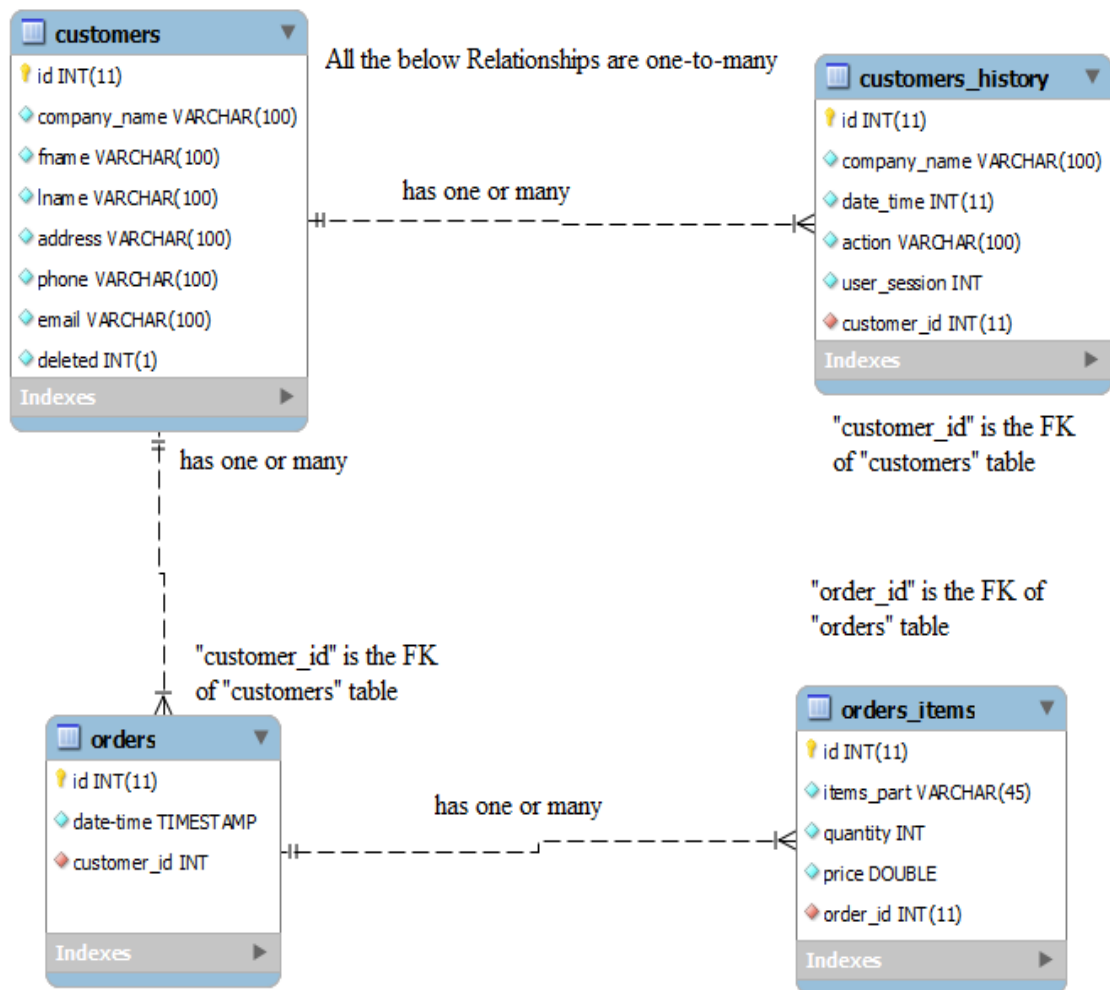


Figure 9. MySQL ERD description

4.4.3 MySQL ERD relationships

One-to-many is the most common relationship between database's tables. For instance a row in customers table has many matching rows in the table, but a row in orders table can have only a single matching row in customers table. In the Figure below, the "orders" and "orders_items" tables have one-to-many relationship: each order consists of many different order items, but each order item belongs only to one order. The below figure explains further the relationship, one-to-many, of the above MySQL ERD model. The "id" of the "customers" table refers to the "orders" table as "customer_id"; foreign key. The "id" of "orders" tables refers to the "id" of "orders_items" table as "order_id"; foreign key.

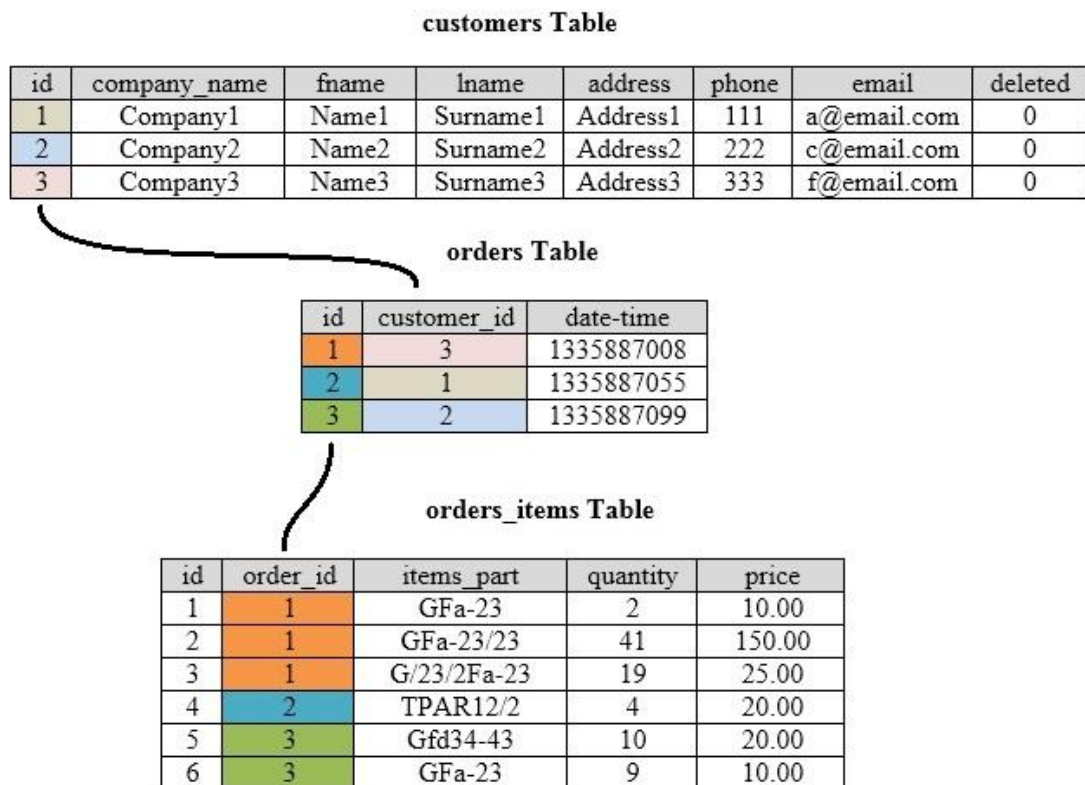


Figure 10. MySQL ERD relationships

5 DEVELOPMENT CONFIGURATIONS

5.1 System settings

The hosting environment provided by Bluehost is shared, and therefore it is not possible to make advanced system changes, modify any firewall or open any ports. The security infrastructure and firewalls are administered by Bluehost personnel and it is considered more secure than being administered by individuals. In the table below, the available ports, URLs and protocols of the hosting infrastructure are listed.

Protocol	Port	Description	URL
http	80	Web Access	http://logiwan.com
https	443	Secure Web Access	https://logiwan.com
ftp	21	File Transfer Protocol	ftp://logiwan.com
sftp	2222	Secure File Transfer Protocol	sftp user@logiwan.com:22
ssh	2222	Secure Shell	ssh user@logiwan.com:22
pop	110	Secure Incoming Mail Server	pop.logiwan.com
smtp	587	Secure Outgoing Mail Server	smtp.logiwan.com
https	443	cPanel	https://logiwan.com/cpanel
https	443	Backend Administrator Panel	https://logiwan.com/admin-cpanel
https	443	Web Application (IMS)	https://logiwan.com/app
MySQL	3306	MySQL Remote Access	<boxname>.Bluehost.com

Figure 11. System settings

5.2 Basic development configurations

The development environment is located in the domain hosting but the access is restricted to the public. The HTTP Authentication from the Apache Web Server is blocking any unauthorized action to the entire domain name at www.logiwan.com. phpDesigner 8 is connected directly to the root directory of the remote web server. An SFTP (Secure File Transfer Protocol) encrypted connection is established between the host computer and the FTP server of the domain hosting. SFTP is a secure connection that provides file management and transmission of encrypted data between the server and the client (Bradley Mitchell 2013).

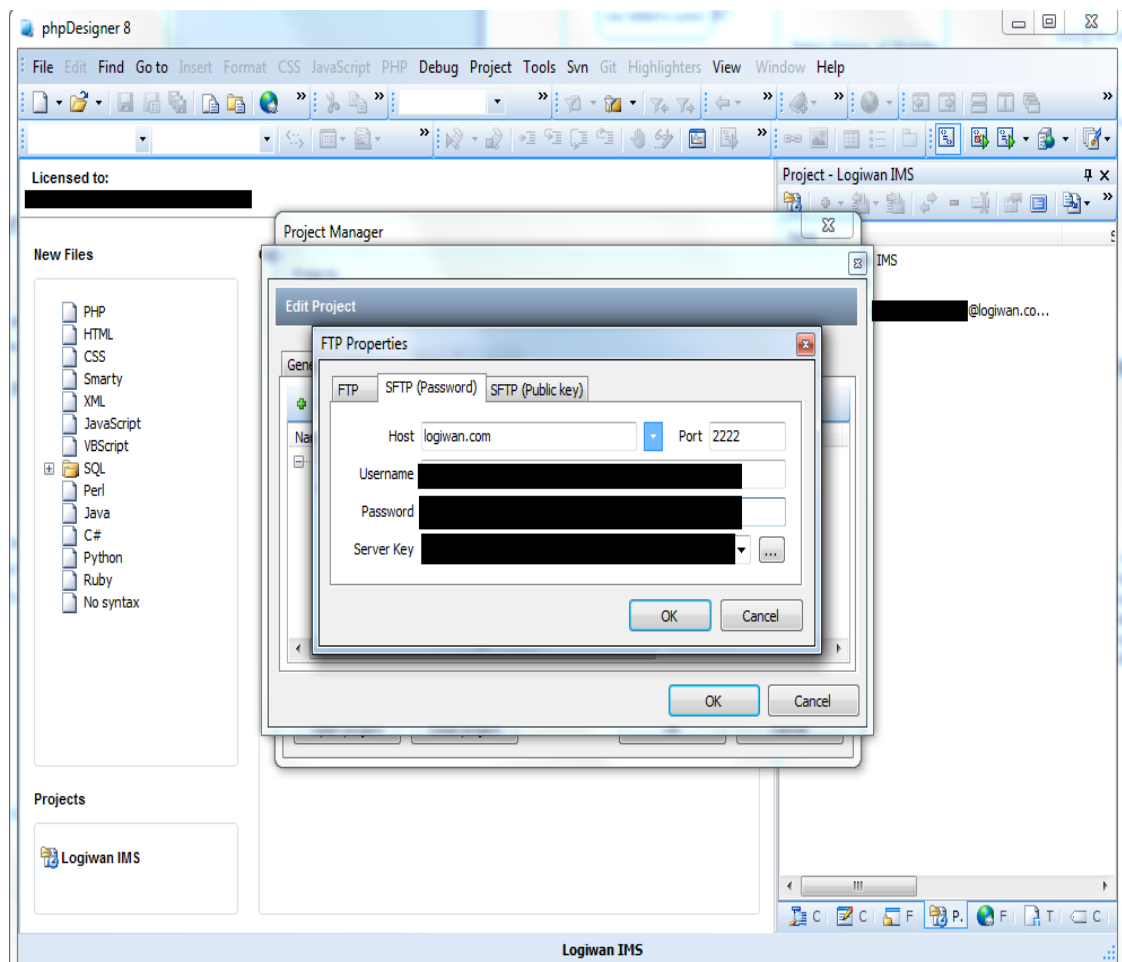


Figure 12. phpDesigner SFTP setup

Before the beginning of the development, there are a number of settings that must be configured in order for the specifications to meet the needs of the web application. To access the back-end administrator cPanel for the hosting configurations, a login is required at “www.logiwan.com/cpanel”. The below figure represents the homepage after as successful login preceded at www.logiwan.com/cpanel.

The screenshot shows the Bluehost cPanel interface. The top navigation bar includes links for cPanel, Domain Manager, Site Builders, Upgrades, Postini, Dedicated IP, SSL Certificates, Profile/Billing, Checkout, Feedback, and HELP. The main content area is organized into several sections:

- Stats:** A sidebar on the left showing account details for logiwan.com, including file count (2165), email accounts (3/∞), subdomains (2/∞), parked domains (0/∞), add-on domains (1/∞), FTP accounts (0/∞), mailing lists (0/∞), all SQL databases (1/∞), MySQL databases (1), PostgreSQL databases (0), account expiration, hosting package (Platinum Pak), hostname, cPanel version (11.32.5), theme (bluehost), Apache version (2.2.23), PHP version (5.4.8), MySQL version (5.5.29-log), architecture (x86_64), operating system (linux), and dedicated IP address (67.20.88.177).
- Promotional:** A section for promotional offers.
- Partners:** A section for partner links.
- Preferences:** A section for user preferences.
- Logs:** A section containing tools like Latest Visitors, Bandwidth, Webalizer, Webalizer FTP, Raw Access Logs, Choose Log Programs, Error Logs, Awstats, and CPU Throttling.
- Files:** A section containing tools like Site Backup & Restore, Legacy File Manager, File Manager, File Count, Web Disk, Disk Space Usage, FTP Accounts, FTP Session Control, Website Movers, and Unlimited FTP.
- Domains:** A section containing tools like Register Domain, Transfer Domain, Subdomains, Addon Domains, Parked Domains, Redirects, DNS Zone Editor, CloudFlare, and Domain Manager.
- Security:** A section containing tools like Password Protect Directories, IP Deny Manager, SSL/TLS Manager, SSH/Shell Access, HotLink Protection, Leech Protect, and GnuPG Keys.

Figure 13. Backend administrator cPanel

First of all, some important modifications have been done in the “php.ini” file. This file is a configuration file that is used to customize the runtime of the PHP interpreter.

Moreover, it enables easy administration in the way you administer Apache web server using configuration files. Some of the functions that can be modified are applied to upload directory, log errors, display errors, max file size for upload, register global variables and a lot of other configuration settings.

The first configuration is to select the version of PHP that will be run with “.php” extension files. The latest, most secure and reliable PHP version is 5.4. In the Figure below, all the available PHP versions are listed.

- PHP 5.2**
*All files with the extension .php will be handled by the PHP 5.2 engine.
Legacy PHP with security updates. Compatible with most environments.*
- PHP 5.2 (Single php.ini)**
Same as PHP 5.2, but all subdirectories will use ~/public_html/php.ini
- PHP 5.2 (FastCGI)**
*All files with the extension .php will be handled by PHP 5.2 FastCGI processes.
FastCGI for PHP makes all your PHP applications run through mod_fastcgi instead of mod_suphp.
This eliminates the overhead of loading the PHP interpreter on every hit. Since it is always in memory ready for the next hit, the responses will be generated faster.*
- PHP 5.3**
*All files with the extension .php will be handled by the PHP 5.3 engine.
Most reliable and compatible version of PHP.*
- PHP 5.3 (Single php.ini)**
Same as PHP 5.3, but all subdirectories will use ~/public_html/php.ini
- PHP 5.4**
*All files with the extension .php will be handled by the PHP 5.4 engine.
Latest version of PHP.
Note: Zend Guard Loader currently not supported.*
- PHP 5.4 (Single php.ini)**
Same as PHP 5.4, but all subdirectories will use ~/public_html/php.ini

Figure 14. PHP version (Bluehost.com cPanel Page)

Global variables can be enabled and disabled through “php.ini” file. The configuration name of global variable is “register_globals” and must be disabled in order to prevent and close any security holes. If “register_globals” is enabled, a malicious user can pass a parameter through the URL and bypass any authentication or inject malicious code. Figure 15 presents the configuration command for Global variables.

```
; You should do your best to write your scripts so that they do not require  
; register_globals to be on; Using form variables as globals can easily lead  
; to possible security problems, if the code is not very well thought of.  
register_globals = off
```

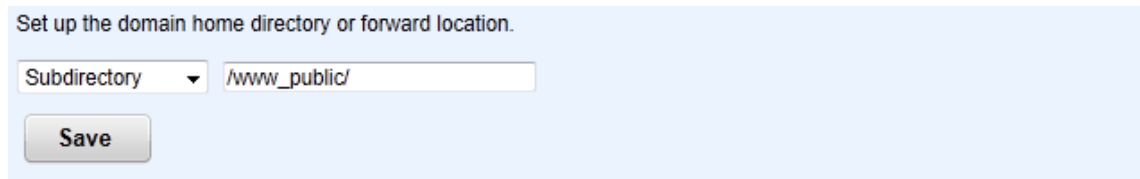
Figure 15. Register globals

PHP offers an effective solution to log all errors to a log file. Log errors must be enabled in development and production environments because errors and failures of the code are very important for troubleshooting. Figure 16 represents the configuration command for error logs.

```
; Log errors into a log file (server-specific log, stderr, or error_log (below))  
; As stated above, you're strongly advised to use error logging in place of  
; error displaying on production web sites.  
log_errors = on
```

Figure 16. Log errors

Moreover, a subdirectory has been created in the root directory of the Linux server and point to the root of the web server. In case the root directory of the web server has been compromised, the attacker will not have the actual root directory of the server but only a subdirectory. Figure 17 illustrates the creation of the web server root directory.



Set up the domain home directory or forward location.

Subdirectory

Save

Figure 17. Root directory pointer

Last but not least is the configuration of the HTTP Authentication by the Apache web server. The configuration file responsible for the HTTP Authentication is “.htaccess” file; monitor and rejects the income requests. This file is a directory-level configuration file, which is defined as hypertext access, and is located in the root directory of the web server. The purpose of this file is to alter the default configuration of the Apache HTTP server and enable or disable any functionality and features accordingly. The below “.htaccess” code authenticate every IP address reach “www.logiwan.com” domain. “Allow from” directive is filled with the public IP address that the configuration file will bypass and not perform any authentication; ex: 192.168.10.152. Figure 18 presents the directory password protection script.

```
#DIRECTORY PASSWORD PROTECTION
AuthUserFile /root/dir/public_html/.hidden_file/.htpasswd
AuthType Basic
AuthName "Website is currently under maintenance - Authorized Access Only"
Require valid-user
Order allow,deny
Allow from XXX.XXX.XXX.XXX
satisfy any
```

Figure 18. .htaccess file

Also, Apache HTTP Authentication is required to load “.htpasswd” file. This file is a flat-file used to store username and passwords for the basic authentication of the web server. The passwords are stored in a hash format, encrypted with MD5 or Crypt function of Unix-Like Operating Systems (Bradley Mitchell 2013). Figure 19 presents the credentials’ file.

```
#Username:HashPassword  
username:password_in_hash_format
```

Figure 19. .htpasswd file

6 TECHNICAL SPECIFICATIONS AND FUNCTIONALITIES

6.1 User interface

One of the most important aspects of the web application is the user interface and its usability. Users should be able to simply quickly and intuitive use any web application, like with any tool in life (Oleg Mokhov 2013).

Below, there are five important elements to be considered for user interface usability. First of all, the application design is a simple combination of HTML and CSS. Second, the layout is very important and the content is visible without any scrolling. Third, the navigation is positioned on the top of every page and the current page is highlighted on the menu. Fourth, the custom design template is the same in every page. Lastly, the web application readability must be guaranteed with the avoidance of any color contrasts issues and promote the use of darker color text on lighter background. The most common color used for the web application design is a combination of silver and grey. The following Figure represents the homepage of Logiwan IMS website.

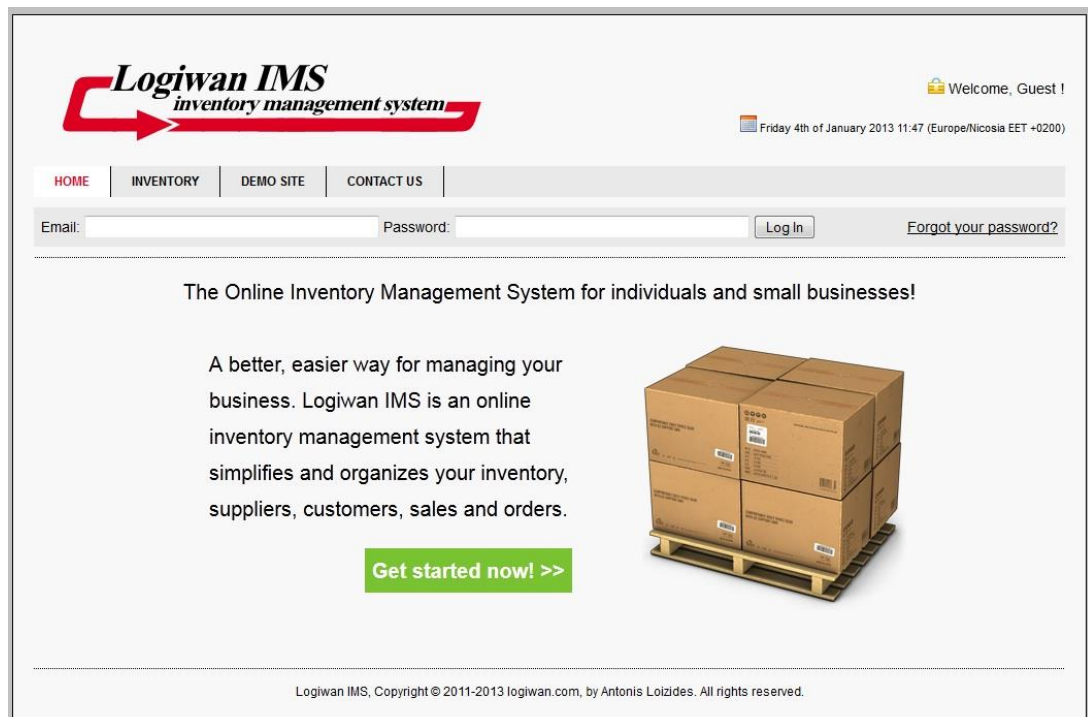


Figure 20. Main homepage

Caterpro Ltd suggested the development of another one page called “Demo Site”. Figure 21 represents the Demo Site, a demonstration purposes site where the user can use the provided credentials to log in to the inventory management system and take a free tour. This site serves potential customers to take the appropriate decision and advertise the web application as well as the below figure presents.

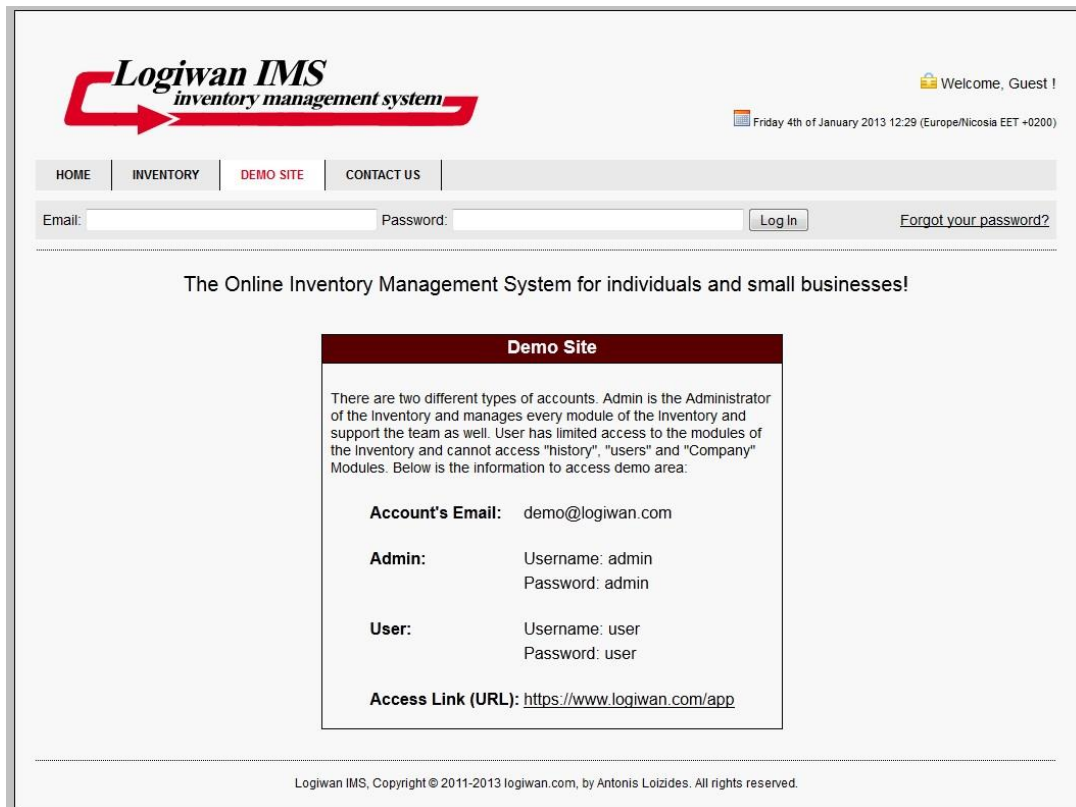


Figure 21. Demo site

In the below Figure is the account page. This page is a different part of the web application and is only available for the account holder. The account holder can perform a few administration actions such as changing name and email, clearing the logs and even closing the account and deleting all the data associated with it. Figure 22 represents the account page.

Clear all history logs', and 'Delete Account: [Close account and delete all data](#)'. At the bottom of the page, there is a copyright notice: 'Logiwan IMS, Copyright © 2011-2013 logiwan.com, by Antonis Loizides. All rights reserved.'" data-bbox="187 118 853 455"/>

Logiwan IMS
inventory management system

Welcome, you are logged in as **Antonis Ltd** | [Logout](#)

Thursday 3rd of January 2013 23:47 (Pacific/Pago_Pago SST -1100)

HOME | INVENTORY | **ACCOUNT** | CONTACT US

Current IP: [redacted] | Last Activity: 2nd of January 2013 22:55

The Online Inventory Management System for individuals and small businesses!

Account

Company: Antonis Ltd
 Email: loizides.antonis@gmail.com
 Owner: Antonis Loizides
 Country: Cyprus
 Timezone: (GMT+02:00) Helsinki
 Currency: EUR - €

Last Activity: 2nd of January 2013 22:55
 Since: 25th of November 2012 21:52
 Clear History Logs: [Clear all history logs](#)
 Delete Account: [Close account and delete all data](#)

Logiwan IMS, Copyright © 2011-2013 logiwan.com, by Antonis Loizides. All rights reserved.

Figure 22. Account page

The actual web application is located at www.logiwan.com/app. The user can visit directly the above URL and login to Inventory Management System application. Figure 23 is the login page window before the user enters the email of the account. The user must provide the unique email of the account holder in order to login to the inventory application. The username and password fields are disabled before any valid email provided. Figure 23 presents the login page of the inventory.

Please enter your account's email below, in order to locate your Company's Inventory!

Account's Email:

Logiwan IMS
inventory management system

Version: 1.1 beta

Inventory Log In

Username:

Password:

Logiwan IMS, Copyright © 2011-2013 logiwan.com, by Antonis Loizides. All rights reserved.

[Return to Account](#)


Recommended Browsers for Logiwan IMS
Internet Explorer: 7 - 9
Firefox: 4 - 10

Figure 23. Web application login page

The below figure is the status of the web application login page after the user enters a valid email that is associated with an account. Username and password are enabled automatically and the user can enter his private credentials that belong to the specific company's account. The message and input field at the top of the page has been replaced with a different message and the account's name of the company. As the figure example shows below, the company's account name is "Antonis Ltd".

Welcome, **Antonis Ltd!** To continue, please enter your username and password below

Version: 1.1 beta



Inventory Log In

Username:

Password:

Logiwan IMS, Copyright © 2011-2013 logiwan.com, by Antonis Loizides. All rights reserved.

[Return to Account](#)

Recommended Browsers for Logiwan IMS

Internet Explorer: 7 - 9
Firefox: 4 - 10

Figure 24. Web application login page 2

Once the user provides a valid set of credentials the login script runs, validates the input data and redirects the user to the home page of the Inventory Management System. Important information is always available and visible throughout the application. At the top-left position the logo “Logiwan IMS” is visible and clickable; redirecting the user back to the homepage. On the top-right position there is a message that welcomes the current user with the username listed and the logout link. Also, below the username the time is shown and adjusted to the specific time zone of the inventory account.

HTML “div” tags are division or sections in an HTML document that are used as a container unit for the encapsulation of other HTML and page elements. The first “div” tag is the menu with the highlighted menu choice of the current page. Below the menu, another “div” tag located on the left, contains the inventory cost and selling price. The user is able to monitor the current cost and selling price of the items listed in the inventory in real time. . On the same line “div” tag but right position, the current IP address of the user and the last activity of the account is listed; for informational

purposes. All of the above HTML elements and “div” tags are part of the master “div” tag the header.

The next master “div” tag is the container. The container holds all the content of the current page positioned in the middle of the page. For instance, in the home page the container “div” tag holds the labels with an icon of the inventory modules. The last master “div” tag is the Footer. This element holds the copyright information along with the name of the web application. The below figure represents the homepage of the inventory application.

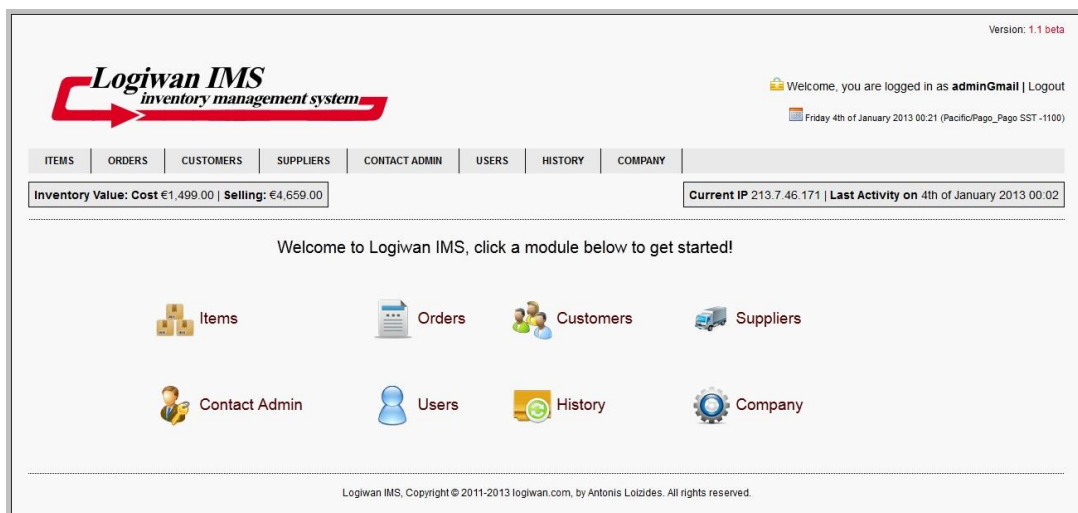


Figure 25. Web application homepage

Another one example of the master container “div” tag, is shown in figure 26 with the items module. The “div” tags belonging to the master container are the left panel input and right panel query. Left panel input of the below figure represents the input form and right panel query the table with the inventory items.



Figure 26. Items module

6.2 Scripting

Once the custom template is created with HTML and CSS, the last and most important part is the design and programming of the functional scripts. The essential functions the web application needed are the input validation, login, session and logout system, password hashing and e-mail notifications as well as export MySQL tables to an MS Excel file.

6.2.1 Input validation and sanitization

Input validation and sanitization is defined as the process of ensuring that a program operates on clean, correct and useful data (Rabin 2012).

A user, especially a web user, will not always submit data that the application will expect. So, the first and the most important principle in web application security is “Don’t trust user input in any way” (OWASP.org 2013). After an extended research in web application security, the development of the above functions proved to be essential. Every input string and variable must be cleared with “sanitizeString” function. The below figure is a function that uses some ready-made PHP functions to sanitize and clear all the variables.

```
function sanitizeString($var)
{
    $var = strip_tags($var);
    $var = htmlentities($var);
    $var = stripslashes($var);
    return mysql_real_escape_string($var);
}

$username = sanitizeString($_POST['username']);
$password = sanitizeString($_POST['password']);
```

Figure 27. Input sanitization

Once the sanitization process is completed and the function cleared the variables another PHP function is called in order to validate the input data and meet the needs of the specific input field. For instance, the input characters available for the username and password are letters and numbers without spaces. The function below, “preg_match” searches the variables username and password in order to match to the regular expression; “/^[a-zA-Z0-9]+\$/i”. If the expression is “false” and equals zero, the function returns an error. Figure 28 illustrates the input validation statement.

```

if (preg_match("/^[a-zA-Z0-9]+$\/i", $username)=== 0)
    {
        $error = "<div class='failed_msg_input'>Username Invalid:
        Use only letters(a-z) and numbers without spaces</div>";
    }
elseif (preg_match("/^[a-zA-Z0-9]+$\/i", $password)=== 0)
    {
        $error = "<div class='failed_msg_input'>Password Invalid:
        Use only letters(a-z) and numbers without spaces</div>";
    }

```

Figure 28. Input validation

If all the statements returns “true”, then the function for the password hashing is called, to convert the password into a hash value; refer to Chapter 6.2.2 for further explanation on hash function, and compare username and password variables with the values in the database of the specific account id. Moreover, the database query excludes those accounts that are not activated or suspended from logging in to the web application.

Figure 29 represents the database login query.

```

$account_id = $_SESSION['account_id'];
hash_password();
$query = queryMysql("SELECT users.username,users.password,accounts.suspend,accounts.activation_url
FROM accounts,users WHERE users.account_id=accounts.id
AND users.username='$username' AND users.password='$password' AND accounts.activation_url='1'
AND accounts.suspend='0' AND accounts.id='$account_id'");

```

Figure 29. Database login query

6.2.2 Password hashing and salting

Password hashing is specified as a cryptographic function algorithm that takes an arbitrary block of data and returns a fixed-size bit string, the cryptographic hash value, such that any change to the data will change the hash value (Antone Consalves 2013). Also, hash functions, called one-way functions because the hash value is extremely difficult to be converted back to its initial state. Password salting provides a random data stream that is used as an additional input to a one-way function in order to increase the security of the password. Password is one of the most important fields that must be always stored in hashed and salted format in order to prevent unauthorized access.

In the below figure, a custom function is developed in order to hash and salt the passwords before storing them in the database. The default PHP hash algorithm has been used in order to calculate the hash value of the password with combination with a salting key. Furthermore, “SHA-512” was selected as the hashing algorithm, which is the maximum supported hash algorithm in the current PHP version. The original function is confidential due to restrictions provided from the management of Caterpro Ltd. However, the figure below shows the basic form of the salting function and process.

```

/** Password Hashing **/
function hash_password()
{
    global $password;
    $salt = "!(*&FYH#(E&)(*^%TRYd#$$^&*(saUJ@#$$^&*Nodisf(*&^%$@$3red";
    $password = hash("sha512", $salt . $password);
}

```

Figure 30. Password hash

6.2.3 Login system

Firstly, the web application's login page requires the account's email in order to locate the unique id number of the account in the database. Then, the query script searches if the provided username and password belong to the specific account id.

The coding is divided into two different files, "index.php" and "login_controller.php". The index file contains HTML code which defines the structure and format of the page and the "login_controller.php" PHP file contains all the functions and the actual PHP scripts.

Following up the input sanitization and validation processes, once the user's input is validated, the user will be identified with the primary account id associated with the company. The below script will query the database and load the required data into the session variables that will be used at a later stage in the web application's functionality. For example, currency and time zone fields will be used later in the company configuration and orders modules. The account type session variable is the value that will provide the necessary access control to administrators and users in the inventory application. At the end of the script, the user is redirected to the homepage of the inventory; www.logiwan.com/app/home. A sample of the login script is displayed in Figure 31. Appendix 1 provides the full code.

```

if (isset($_POST['account_email'])) {
    $email = sanitizeString($_POST['account_email']);
    if ($email=="") {
        $error = "Please enter your account's email.";
    }
    else {
        $result = queryMysql("SELECT id,company,timezone,currency,email,fname,lname
                             FROM accounts WHERE email = '$email'");
        if (mysql_num_rows($result)==1) {
            $rows = mysql_fetch_array($result);
            $fname = $rows['fname'];
            $lname = $rows['lname'];
            $_SESSION['account_id'] = $rows['id'];
            $_SESSION['current_company'] = $rows['company'];
            $_SESSION['timezone'] = $rows['timezone'];
            $_SESSION['currency'] = $rows['currency'];
            $_SESSION['email'] = $rows['email'];
            $_SESSION['fullname-main'] = "$fname $lname";
            $_SESSION['browser'] = sha1($_SERVER['HTTP_USER_AGENT']);
            $_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
        }
        else {
            $error = "Account's Email invalid"; } } }
if (isset($_POST['username']) && !isset($_SESSION['current_company'])) {
    $error = "First, please enter your account's email.";}

```

Figure 31. Login script

6.2.4 Session management security

The web-based session management is the most common method of tracking a user's activity through a web page by assigning a unique session id and having this information transmitted back to the web server with every request (Gunter Ollmann 2013). PHP Session functions generate and assign to every host a unique id. PHP sessions are global variables and store the information on the server, based on this id, where can be accessed in every PHP file. However, each session file that holds all the variables is temporary and is usually deleted after the user has ended the session either by leaving or closing the browser window.

On the other hand, session data flowing are still visible in plain text over the Internet and computer network links and anyone can steal the information with the use of the packet sniffing method. This is the primary reason why SSL certificate is an essential and a required technique to be implemented in the web application. As previously mentioned in Chapter 3.2, SSL Certificate is used to establish and maintain a secure and encrypted connection between the web server and the host; using HTTPS protocol and default port 443. The chosen certificate authority is COMODO CA Limited. The encryption is classified as a High-grade Encryption with the RC4 algorithm and 128 bit keys.

After adding an encryption layer between the web application and the user, a session management script is required to manage the user's data and security. The script in question provides a layer of complexity to prevent the session hijacking vulnerability. This session hijacking vulnerability attack describes the exploitation of the web session control mechanism by sending the id of an authorized session of another user to the web server.

The session variables for IP address and browser identify the combination of two important keys for validating the user's session and preventing any session vulnerabilities. The session file is included in the template of the web application and runs in every user request. The script initially checks if any session authorization variable is declared. Another important factor is the statement which compares the

browser identity stored in the session variable with the current identity and the IP address which is stored in another session variable with the existing IP address. In case any of these statements failed, the else statement calls the function responsible for destroys all of the data associated with the current session id and redirect the user back to the login page. Figure 32 illustrates the script for session security.

```

if (isset($_SESSION['auth-app']) || isset($_SESSION['auth-main'])
    || isset($_SESSION['current_company']))
{
    if ($_SESSION['browser']==sha1($_SERVER['HTTP_USER_AGENT'])
        && $_SESSION['ip']==$_SERVER['REMOTE_ADDR'])
    {
        /** User is validated and verified */
    }

    else
    {
        $_SESSION=array();

        if (session_id() != isset($_COOKIE[session_name()]))
            setcookie(session_name(), '', time()-2592000, '');

        session_destroy();
        session_start();
        $_SESSION['ip_changed'] = "";
        $check_dir = sanitizeString($_SERVER['PHP_SELF']);
        //$page_name = basename($page_name, ".php");
        if (strpos($check_dir, "app") !== false)
        {
            Header("Location:$domain/app");
        }
        else
        {
            Header("Location:$domain/login");
        }
        exit();
    }
}

```

Figure 32. Session security

Every time a session is initialized, a cookie is stored in the user's browser with the generated unique session id. The cookie is set to expire every time the user close the browser or if the user click logout. The script in figure 33 identifies whether the variable "logout" exists in the URL address bar and destroy the session. Also, the script identifies if the user is logged in the inventor web application or in the account page and redirects the user accordingly. Figure 33 represents the logout function.

```
if (isset($_GET['logout']))
{
    $_SESSION=array();

    if (session_id() !=  isset($_COOKIE[session_name()]))
        setcookie(session_name(), '', time()-2592000, '');

    session_destroy();
    if (strpos($_SERVER['REQUEST_URI'], "app")
    {
        Header("Location:$domain/app");
    }
    else
    {
        Header("Location:$domain/login");
    }
    exit();
}
```

Figure 33. Logout function

6.2.5 Email notifications

The email notification system is a fundamental part of the web application which is used when the back-end administrator of the web application creates a new account, for the “contact us” form or when the user wishes to proceed with a password reset. The PHP mail function chosen is phpMailer version 5.2.2, which is a PHP email transport class featuring file attachments, SMTP servers, CCs, BCCs, HTML messages, word wrap as well as other features. In addition, the class is considered source and is published under the GPL License; users can freely use, modify and distribute the source code.

In order to send emails, the source code PHP file class must be included in the script and then called with the use of sendmail, PHP mail, QMail or directly SMTP functions. For the purpose of this script, a custom PHPMailer function was implemented and used. “mail_functions.php” is the file which includes the phpMailer class, the custom mailer function and the three different statements that call the function.

The first statement which calls the function is the register process. When the back-end administrator creates a new account, an automatic email is sent to the account holder’s email with all the default admin credentials of the account and inventory. The second statement is the contact_us process where the user fills in the appropriate information for contacting Logiwan IMS. The last statement is the recovery process where the user fills in a valid account email and an automatic email is sent to the specific email for resetting the password.

An if statement compares the value of the string of “register”, “contact_us” and “recover” functions to a constant variable; “page_name”, that retrieves the current page name and select the appropriate values for the selected script. Moreover, all the declared global variables are executed from the POST variables of the previous script files. Figure 34 shows a sample of the mail function script file. Refer to Appendix 2 for the full code.

```

require_once ('class.phpmailer.php');
$domain = get_domain;
function mailer() {
global $domain,$password_account,$email,$activation_url,
$recovery_url,$send_to_email,$comment,$comment_original,$default_password;
$ip = $_SERVER['REMOTE_ADDR'];
if (page_name == "register"){
    $subject = "Logiwan IMS - Account Activation";
    $comment = "Welcome to Logiwan IMS! Please use the below link to activate your account.<br/><br/>
    $domain/login?activation&url=$activation_url<br/><br/>
    Please use the below credentials to log in at:<br/>
    Account Page (www.logiwan.com/login):<br/>
    Username: $email<br/>
    Password: $password_account
    <br/><br/>
    Inventory (www.logiwan.com/app):<br/>
    Username: admin<br/>
    Password: $default_password<br/><br/>
    *Logiwan IMS recommended that you change your default passwords once you logged in.<br/><br/>
    The request came from the IP: $ip";}
if (page_name == "signup"){
    $subject = "Logiwan IMS - Account Activation";
    $comment = "Welcome to Logiwan IMS! Please use the below link to activate your account.<br/><br/>
    $domain/login?activation&url=$activation_url<br/><br/>
    The request came from the IP: $ip";}

```

Figure 34. Mail function

7 CONCLUSIONS

Deriving from the objectives of this research, the expected outcome of this research was the development of a Software as a Service Inventory Management System web application for Caterpro Ltd. The aim of Caterpro Ltd was to find an online inventory management system web application in order to expand its IT business in the Software as a Service industry. In conclusion, the research was divided into seven steps. At first, the system requirements document was collected from the company in order to identify all the major components to provide a solid foundation of the problem to develop. Secondly, a research for Software as a Service software model and inventory management system applications was carried out. Following the second step, the suitable research methodology was selected with the development tools and different technologies used. Fourthly, an important part of the whole procedure was to design the different diagrams in order to get the logical infrastructure of the web application's functions and the different steps involved for development at a later stage. The fifth step was the basic design of the web application's layout developed with "div" structure HTML elements. Finally, the cornerstone of the web application was the development of the functions and scripts in order to give the required functionality to the web application and meet the pre-defined requirements set from the company.

The research has shown that many measures have been taken regarding the security of the web application such as secure session management, input validation and sanitization, secure login and logout systems as well as email notifications. On the other hand, security measures for the user have been taken as well, such as irreversible hashed and salted password's functions.

Due to strict time constraints from Caterpro Ltd, only the basic and essential functions of an inventory management system application were developed. However, there are still more advanced functions needed to be developed such as a unique company domain name/ i.e. company.logiwan.com. In addition, the PDF invoice template must be more professional including the logo and policy of each company etc.

Moreover, the company has an upcoming upgrade project for Logiwan IMS in order to convert the PHP functions in fully functional AJAX environment with the use of asynchronous functions with jQuery.

REFERENCES

Bradley Angela 2013. What is PHP. Downloaded February 20, 2013.

<http://php.about.com/od/phpbasics/qt/what_is_php_used_for.htm>

Bradley Mitchell 2013. SFTP. Downloaded February 20, 2013.

<<http://compnetworking.about.com/od/ftpfiletransfer/g/sftp-definition.htm>>

Bradley Mitchell 2012, Sniffer. Downloaded December 28, 2012.

<http://compnetworking.about.com/od/networksecurityprivacy/g/bldef_sniffer.htm>

Bluehost 2013. Figure 14, the site is not accessible to the public unless a hosting account is purchased from the provider at bluehost.com. Downloaded February 20, 2013.

<<http://www.bluehost.com/>>

Brown Martin 2013. Understanding LAMP. Downloaded February 20, 2013.

<http://www.serverwatch.com/tutorials/article.php/10825_3567741_1>

Chapman Stephen 2013. What is JavaScript. Downloaded February 20, 2013.

<<http://javascript.about.com/od/reference/p/javascript.htm>>

Chapple Mike 2012. Entity-Relationship Diagram. Downloaded December 20, 2012.

<<http://databases.about.com/cs/specificproducts/g/er.htm>>

Charles Torvalds 2013. PHPMailer tutorial. Downloaded January 10, 2013

<<http://www.askapache.com/php/phpfreaks-eric-rosebrocks-phpmailer-tutorial.html#Introduction>>

Consalves Antone 2013. Contest aims to boost state of password encryption. Downloaded February 20, 2013.

<<http://www.csoonline.com/article/728944/contest-aims-to-boost-state-of-password-encryption>>

Cpanel.net 2013. cPanel Official Site. What is cPanel. Downloaded February 20, 2013.

<<http://cpanel.net/cpanel-whm>>

E. K. Lukka Kasanen & A. Siitonen. 1993. The constructive approach. Downloaded December 28, 2012.

<<http://maaw.info/ArticleSummaries/ArtSumKasanenetal93.htm>>

Gartner Research 2012, Gartner says Worldwide Software-as-a-Service Revenue to reach \$14.5 Billion in 2012. Downloaded December 28, 2012.

<<http://www.gartner.com/it/page.jsp?id=1963815>>

Herbert Liz 2011, Buyers Scrutinize SaaS Contracts more in H1 2011, as deal sizes grow. Downloaded December 28, 2012.

<http://blogs.forrester.com/liz_herbert/11-08-26-buyers_scrutinize_saas_contracts_more_in_h1_2011_as_deal_sizes_grow>

Instantssl.com 2013. Comodo SSL Certificate Authority. What is SSL. Downloaded February 20, 2013.

<http://www.instantssl.com/ssl-certificate-products/what_is_ssl.html>

Kyrnin Jennifer 2013a. What is CSS. Downloaded February 20, 2013.

<<http://webdesign.about.com/od/beginningcss/a/aa021607.htm>>

Kyrnin Jennifer 2013b. What are Markup Languages. Downloaded February 20, 2013.

<<http://webdesign.about.com/od/htmlxhtmltutorials/p/what-are-markup-languages.htm>>

Kyrnin Jennifer 2013c. What is Ajax. Downloaded February 20, 2013.

<<http://webdesign.about.com/od/ajax/a/aa101705.htm>>

Mpsoftware.dk 2013. phpDesigner 8 Official Site. Downloaded December 20, 2012.

<<http://www.mpsoftware.dk/phpdesigner.php>>

Mokhov Oleg 2011, 10 Essential Web Application Usability Guidelines. Downloaded December 28, 2012.

<<http://speckyboy.com/2011/03/31/10-essential-web-application-usability-guidelines/>>

Mysql.com 2013. MySQL Official Site. What is MySQL. Downloaded February 20, 2013.

<<http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>>

Narayan Sheo 2013. What is jQuery. Downloaded February 20, 2013.

<<http://www.codeproject.com/Articles/157446/What-is-jQuery-and-How-to-Start-using-jQuery>>

Ollmann Gunter 2007, Web Based Session Management Best Practices in managing HTTP-based client sessions. Downloaded December 28, 2012.

<<http://www.technicalinfo.net/papers/WebBasedSessionManagement.html>>

Oppel Andy 2005. General RDBMS Considerations. Downloaded December 20, 2012.

<<http://www.devshed.com/c/a/MySQL/SQL-Performance-and-Tuning-Considerations/2/>>

Owasp.org 2013. Don't trust user input. Downloaded December 20, 2012.

<https://www.owasp.org/index.php/Don%27t_trust_user_input>

Peavler Rosemary 2012. Inventory Investment and Maximizing Profit. Downloaded December 28, 2012.

<http://bizfinance.about.com/od/inventory/a/Inventory_Investment.htm>

Phpmailer.worxware.com 2013. PHPMailer. Downloaded January 10, 2013.

<<http://phpmailer.worxware.com/>>

Php.net 2013. PHP Official Site. Downloaded January 10, 2013.

<<http://php.net>>

Rabin 2012. Part 1: Data Validation and Sanitization in WordPress. Downloaded February 20, 2013.

<<http://devotepress.com/coding/data-validation-sanitization-wordpress-1/>>

Realmagick.com 2012, Constructive Research. Downloaded December, 28 2012.

<<http://www.realmagick.com/constructive-research/>>

Rouse Margaret 2010, Software as a Service (SaaS). Downloaded December 28, 2012.

<<http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>>

Webopedia.com 2012, SaaS Software as a Service. Downloaded December 28, 2012.

<<http://www.webopedia.com/TERM/S/SaaS.html>>

APPENDICES

LOGIN SCRIPT

Appendix 1

1(3)

```

if (isset($_POST['account_email']))
{
    $email = sanitizeString($_POST['account_email']);
    if ($email=="")
    {
        $error = "Please enter your account's email.";
    }
    else
    {
        $result= queryMysql("SELECT id,company,timezone,currency,email,fname,lname
        FROM accounts WHERE email = '$email'");
        if (mysql_num_rows($result)==1)
        {
            $rows = mysql_fetch_array($result);
            $fname = $rows['fname'];
            $lname = $rows['lname'];
            $_SESSION['account_id'] = $rows['id'];
            $_SESSION['current_company'] = $rows['company'];
            $_SESSION['timezone'] = $rows['timezone'];
            $_SESSION['currency'] = $rows['currency'];
            $_SESSION['email'] = $rows['email'];
            $_SESSION['fullname-main'] = "$fname $lname";
            $_SESSION['browser'] = sha1($_SERVER['HTTP_USER_AGENT']);
            $_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
        }
        else
        {
            $error = "Account's Email invalid";
        }
    }
}
if (isset($_POST['username']) && !isset($_SESSION['current_company']))
{
    $error = "First, please enter your account's email.";
}
if (isset($_POST['username']) && isset($_SESSION['current_company']))
{
    $username = sanitizeString($_POST['username']);
    $password = sanitizeString($_POST['password']);
    $account_id = $_SESSION['account_id'];
    hash_password();
}

```

```

$query=queryMysql("SELECT
    users.username,users.password,accounts.suspend,accounts.activation_url
FROM accounts,users WHERE users.account_id=accounts.id
AND users.username='$username' AND users.password='$password' AND
    accounts.activation_url='1'
AND accounts.suspend='0' AND accounts.id='$account_id'");
$query2 = queryMysql("SELECT activation_url,recovery_url,suspend FROM
    accounts WHERE id='$account_id'");
$rows2 = mysql_fetch_array($query2);

if (mysql_num_rows($query) == 0)
{
$error = "Username/Password invalid";
if ($rows2['activation_url'] != 1)
{
$error = "Your account was not activated yet. Please use the activation url sent to
    your email in order to
    activate your account";
}
if ($rows2['recovery_url'] != 0)
{
$error = "Forgot My Password is pending. Please use the recovery url sent to your
    email in order to
    change your account's password";
}
if ($rows2['suspend'] == 1)
{
$error = "Your account has been suspended. Please contact the system
    administrator";
}
}
else
{
$query=queryMysql("SELECT id,account_type,last_time,fname,lname,card FROM
    users WHERE account_id='$account_id' AND username='$username' ");
$rows = mysql_fetch_array($query);
$account_type_session = $rows['account_type'];
$last_time_session = $rows['last_time'];
$fname = $rows['fname'];
$lname = $rows['lname'];
$user_id = $rows['id'];
$timestamp_now = time();
queryMysql("UPDATE users SET last_time='$timestamp_now' WHERE
    account_id='$account_id' AND username='$username'");
$_SESSION['card']=$rows['card'];

```

```
$_SESSION['user_id'] = $user_id;
$_SESSION['username'] = $username;
$_SESSION['account_type'] = $account_type_session;
$_SESSION['last_time'] = $last_time_session;
$_SESSION['fullname-app'] = "$fname $lname";
$_SESSION['auth-app'] = TRUE;

$result = queryMysql("SELECT id FROM orders_cart WHERE user_id=$user_id");
if(mysql_num_rows($result)>=1)
{
    queryMysql("DELETE FROM orders_cart WHERE user_id=$user_id");
    queryMysql("UPDATE users SET cart='0' WHERE id=$user_id");
}
Header("Location: home");
exit();
}
```

MAIL FUNCTION

Appendix 2
1(3)

```

require_once (rootdir.'/library/phpmail/class.phpmailer.php');
$domain = get_domain;

function mailer() {

global $domain;
global $password_account;
global $email;
global $activation_url;
global $recovery_url;
global $comment;
global $send_to_email;
global $comment_original;
global $default_password;
$ip = $_SERVER['REMOTE_ADDR'];

if (page_name == "register")
{
$subject = "Logiwan IMS - Account Activation";
$comment = "Welcome to Logiwan IMS! Please use the below link to activate your
account.<br/><br/>
$domain/login?activation&url=$activation_url<br/><br/>
Please use the below credentials to log in at:<br/>
Account Page (www.logiwan.com/login):<br/>
Username: $email<br/>
Password: $password_account
<br/><br/>
Inventory (www.logiwan.com/app):<br/>
Username: admin<br/>
Password: $default_password<br/><br/>
*Logiwan IMS recommended that you change your default passwords once you
logged in.<br/><br/>
The request came from the IP: $ip";
}
if (page_name == "signup")
{
$subject = "Logiwan IMS - Account Activation";
$comment = "Welcome to Logiwan IMS! Please use the below link to activate your
account.<br/><br/>
$domain/login?activation&url=$activation_url<br/><br/>
The request came from the IP: $ip";
}
if (page_name == "contact_us")
{
if (isset($send_to_email))

```

```

    {
        $subject = "Logiwan IMS - Message";
        $comment = str_replace('\r\n', "<br/>", $comment_original);
        $comment = "Below is a message from your contact form submitted.<br/><br/>";
        $comment <br/><br/>
        The request came from the IP: $ip";
    }
}
if (page_name == "login")
{
    if (isset($_GET['recovery']))
    {
        $subject = "Logiwan IMS - Account's Password Recovery";
        $comment = "Please use the below link to reset your account's
        password.<br/><br/>";
        $domain/login?recovery&url=$recovery_url<br/><br/>
        The request came from the IP: $ip";
    }
}
if (page_name == "account")
{
    if (isset($_GET['delete']))
    {
        $email_account = $_SESSION['email'];
        $subject = "Logiwan IMS - Account's Deletion";
        $comment = "Account Requested a Deletion:<br/><br/>";
        $email_account<br/><br/>
        The request came from the IP: $ip";
    }
}

$mail = new PHPMailer();
$mail->IsSMTP(); // SMTP
$mail->SMTPAuth = true; // enable SMTP authentication
$mail->SMTPSecure = 'ssl'; //SSL encryption
$mail->Host = "mail.logiwan.com"; // SMTP server
$mail->Port = 465; // SMTP port
$mail->Username = "do-not-reply@logiwan.com"; // SMTP username
$mail->Password = ")A%sPK5z{}H4"; // SMTP password
$mail->SetFrom('do-not-reply@logiwan.com', 'Logiwan IMS');
$mail->Subject = $subject;
$comment = "<html>$comment</html>";
$mail->MsgHTML($comment);
$mail->AddAddress($mail, "");
$mail ->send();

```