

Chapter 2

A Software Architecture for Inventory Management System

Taner Arsan, Emrah Başkan, Emrah Ar and Zeki Bozkuş

Abstract Inventory Management is one of the basic problems in almost every company. Before computer age and integration, paper tables and paperwork solutions were being used as inventory management tools. These were very far from being a solution, took so much time, even needed employees just for this section of organization. There was no efficient solution available in the many companies during these days. Every process was based on paperwork, human fault rate was high, the process and the tracing the inventory losses were not possible, and there was no efficient logging systems. After the computer age, every process is started to be integrated into electronic environment. And now we have qualified technology to implement new solutions to these problems. Software based systems bring the advantages of having the most efficient control with less effort and employees. These developments provide new solutions for also inventory management systems in this context. In this paper, a new solution for Inventory Management System (IMS) is designed and implemented. Most importantly, this system is designed for Kadir Has University and used as Inventory Management System.

2.1 Introduction

Inventory Management is one of the basic problems for a company. It may cause a lot of paperwork, if there is no automated system available. Implementing such a system is possible but there are a lot of preliminary works such as determination of

T. Arsan (✉) · E. Başkan · E. Ar · Z. Bozkuş
Department of Computer Engineering, Kadir Has University, Cibali,
Istanbul, 34230 Turkey
e-mail: arsan@khas.edu.tr

the requirements, system structure decision—software requirements, barcode system selection and determination of the software tools.

2.1.1 Determination of the Requirements

Inventory Management System (IMS) is generally used by IT Office/Department or Accounting Office of a company or a university. Therefore, searching the basic needs for implementation is the first step of IMS design. Several meetings with IT Office and Accounting Office are arranged. Accounting Office needs detailed reporting tools, detailed categorization and declaration of specifications on each item, purchasing and billing info. The Information Technologies Office needs another module except the requirements of Accounting Office. The module is about the interior maintenance and exterior product service flow. For interior maintenance flow, there will be a section. This section will be available for all users. Basically, a maintenance request will be created by the users, and the IT Office will respond to these requests. Finally, it is necessary to consider the end user's needs that are also important part of the IMS software design. This led us to use barcode based system.

2.1.2 Software Requirements

After gathering all the requirements and information, we achieve next state which is quite important for the project. If the decision is not well organized, it may cause re-organization and programming the algorithms at the beginning.

At the beginning, first thing to decide is the general structure. We decide to build the system browser based. The system is going to be used by a large number of users, so it should have been designed to be reachable for every single user, instead of having a desktop application for every user. Also if we consider that Kadir Has University already has a similar system—Information Management System, it would be easy to integrate. By using the user database of the existing systems, the login parameters are ready to use for IMS. So, we would provide the portability. The next step is to decide the language to implement.

There are several parameters to decide in realization period. First is the compatibility, and then came the time which is related with ease of implementation. In the light of these redirections, we decided to use PHP as the language to code the system. It is compatible with the other software and hardware, and it is easy to integrate everything. PHP is an open source environment and there are a large range of resources to get help. Information Management System is already developed by using PHP. We also decide to use AJAX to obtain a faster system. In AJAX, only the needed data is being requested from the server, this method speeds up the loading time. While thinking how to code with PHP, we decide to use

Model View Controller based structure. This structure separates the visual design. We also use Code Igniter Framework for decreasing the amount of code lines. It gives ability to program faster with less effort. Code Igniter also support MVC structure. So we decide to PHP, MySQL and APACHE trio. We used MySQL as database. It is the best fit for PHP and it has the open source advantages. We can also show the comments of ORACLE as a reference to our right choice [1]. We used innodb engine, because it is transaction based, which is necessary for the project. To get the user data, we have to communicate with the database of University, which is programmed by ORACLE. We need to have the login information to have a fully integrated system, and also several details such as e-mail addresses and so on.

2.1.3 Barcode System

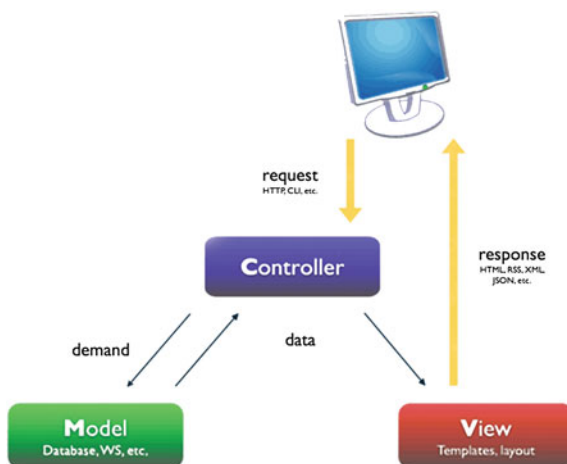
A barcode is an optical machine readable representation of data, which shows certain data on certain products. There are different types of barcodes, some of them have to be world wide unique and need license. We chose “code39” because first of all, code39 doesnot need license, have the possibility to print for unlimited products, supports full ASCII, number, letter and special character usage (capacity of 26 capital letters, 10 numbers, 7 special characters). And also it was the easiest to read and was the best fit for the barcode generators for PHP. As the barcode generator, we found one compatible with PHP. It is using PHP’s GD Graphics Library. We got the script and modified it to be able to print serial barcodes. It prints barcodes in increasing order (in multiple prints) on the screen, and sends to the printer. The script is distributed under GNU License. And the hardware part of the system is out of a label printer and a barcode scanner. Barcode scanner is a USB connected model. It scans the barcode, writes to the field where the cursor is, and presses enter. This function eases the usage. So the only extra hardware to be used by the staff will be a barcode scanner after the system is implemented to the enterprise.

2.1.4 Software Tools

2.1.4.1 PHP

As a derivation of Perl, PHP, is a server side, user interactive, programming language, works nearly in on all platforms. We can say that it is a general purpose scripting language. It can be embedded into html. It can use various databases such as MySQL, SQL, Oracle, MS SQL etc. Also contains many server interfaces. Open source is one of the best specifications of PHP. Among several frameworks, the most popular one is zen [2].

Fig. 2.1 Internal structure of MCV architecture



2.1.4.2 JavaScript

JavaScript is a scripting language used to enable programmatic access to computational objects within a host environment. Generally used to implement dynamic web sites. For more information <http://en.wikipedia.org/wiki/JavaScript>.

2.1.4.3 Asynchronous Javascript and XML

Asynchronous Javascript and XML (AJAX) is being used to build applications which are using JavaScript and XMLHttpRequest. It is mostly used in avoiding to load the whole page, just to load the needed part. By using XMLHttpRequest, it is possible to do more than one independent process.

2.1.4.4 Model–View–Controller

Model–View–Controller (MVC) is a software architecture which provides to implement the visual, data and processing code parts independent. For example the Model unit is a collection of classes in communication with the other parts. It is the process unit, processes the task ordered by the Control Unit. View unit is the place to deal with the presentation of the data to the end user. It can get the data from both Model and Controller unit. Also it can send interactive data to both units. Controller part is the main part of the structure. If we want to concretize the architecture, C is the brain, M is the nerves and V is the move. Internal structure of MCV Architecture is given in Fig. 2.1.

2.1.4.5 Code Igniter Framework

CodeIgniter is a framework with a small footprint, requires nearly zero configuration, does not require the use of command line and provides to use large scale of libraries. CodeIgniter also provided us object oriented programming [3].

2.1.4.6 Oracle

Oracle RDBMS is a relational database management system. It is one of the strongest software in this section and also one of the largest IT companies on the world.

2.1.4.7 MySQL

MySQL is the most popular open source database software. It is easy to use, fast and reliable. Also it is a good match with PHP [4].

2.1.4.8 phpMyAdmin

It is a software, coded with PHP. The main function of this software is to manage MySQL database through Internet. It can create databases, add/edit/delete tables, run SQL queries, manage user authorization and manage field keys are some of the features [5].

2.1.4.9 InnoDB

InnoDB is a standard database engine in all the packages distributed by MySQL. The main difference of this engine is it is compatible with ACID, and the important ones for our project, it is transaction based and supports foreign key [6].

2.1.4.10 Apache

Apache is a GNU licenced open source web server program. It can run on platforms such as Unix, Linux, Solaris, Mac OS X, Microsoft Windows. Even the usage of Apache decreases, still it is the most popular web server program on web [7].

2.2 Implementation

In this section, there is a closer look to implementation steps, work flow and software architecture. All the data and process flow will get clear. Even it does not seem so complex on the interface level, there are many process running behind to provide efficient function of the system [8].

2.2.1 Database Design

The most important component of the system is the database design. It can be easily realized, if we consider the amount and the scale of the inventories. To have a strong and flexible system, the design must be well studied.

After some studies including investigation, workflow scenarios, traces and design, we obtain the following tables.

1. Attributes
2. Categories
3. ci Sessions
4. Content
5. Debits
6. Inventory
7. Inventory Attribute Values
8. Jobs
9. Locations
10. Providers
11. Services
12. Service Records
13. Tech Users

We start with categories and locations tables. We needed the ability of listing unlimited category and location of items and setting unlimited hierarchy. So we started to search for sufficient algorithms. The research resulted with Nested Sets. Nested Sets is a model used to represent a set of data organized into a hierarchy is useful in a computer database management context. One way hierarchical data has been commonly represented within a database is with the Adjacency List Model, which takes a set of data nodes and recursively attributes parent nodes with their respective child nodes. The Nested Set Model takes the Adjacency List Model a step further by modeling subordination. This is done by assigning each node a beginning and ending node hierarchy number based on the total amount of data in the hierarchical tree.

Two important structures of the system is the database and the Graphic User Interface (GUI). In between, JavaScript and PHP provides the connection and interaction. GUI is located in the view section. With XHTML and CSS, the visual

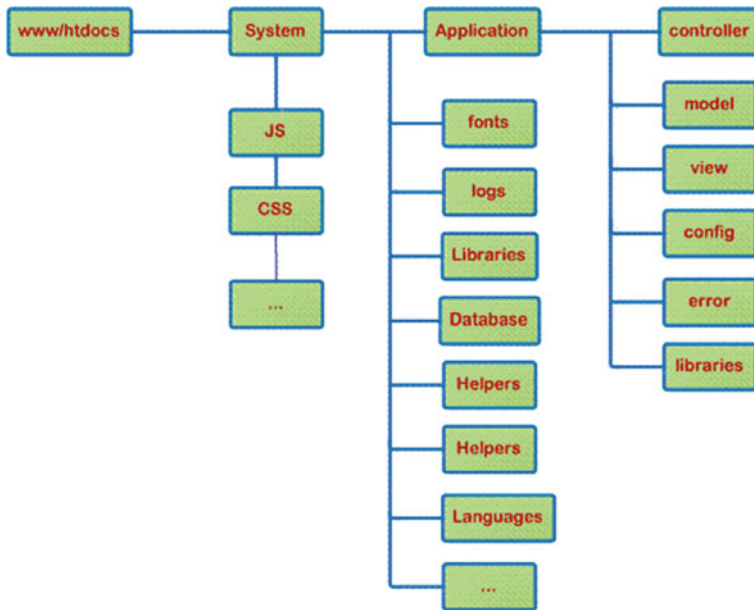


Fig. 2.2 Basic directory codeIgniter framework

part is generated. Javascript updates the HTML by the data sent by other layers and connects the GUI with working PHP structure. We used an Admin Panel called “CompleteLiquid admin control panel”. It was suitable for our MVC structure. With CSS and other related modifications, we could arrange it to our system. In the controller layer, dataflow decision codes take place which are Javascript and PHP. And in the Model section, PHP logic base code is placed.

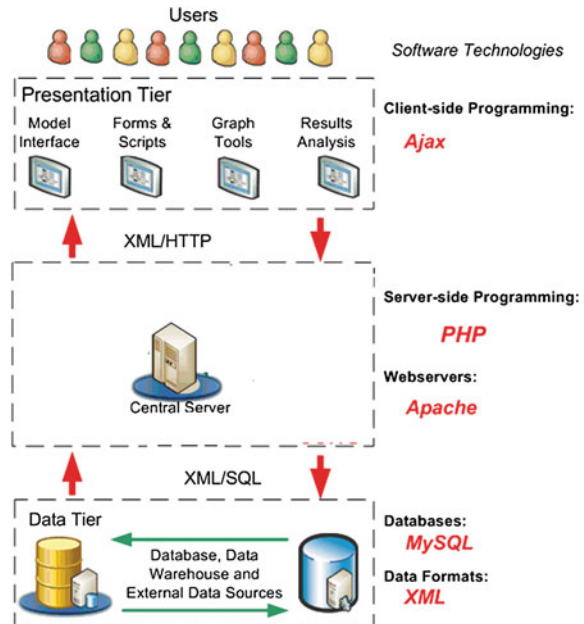
All codes are produced with CodeIgniter framework. The basic directory of the framework can be shown in Fig. 2.2, and a general look of IMS architecture is given in Fig. 2.3.

2.2.2 Modules

IMS is designed as several modules, separated by their specific roles and functions. In this part, the function of each module will be explained step by step. There are six modules in the system.

1. Definitions/Setup
2. Inventory Management
3. Service Management
4. Maintenance Management

Fig. 2.3 General IMS architecture



5. Debit Management
6. Deposit Management

2.2.2.1 Definitions/Setup

This module is the most important part of the system. Because if the definition of the enterprise is not implemented correctly to the system, it won't work efficiently. To have a successful, operational system, introducing the enterprise to the system is a must. And this is the first step of implementing the system to the enterprise.

In this module, there are four sections, which are mentioned below.

- User Setup
- Location Hierarchy
- Categorization
- Suppliers and Maintainers

(a) User Setup

In User Setup Menu, System Admin can create, edit or delete a user from the system. The table below is the user creation form. System admin can choose the role of the user from the picklist on the bottom.

(b) Location Hierarchy

The Location Hierarchy section is the place where you set the ownership hierarchy of the enterprise. System Admin can create one sub unit under another by choosing a unit.

(c) Categorization

In this section, a categorization of the inventories in the enterprise is being created. First step is the main categorization. The second step is related titles, and the last one is the specific keyword. User can add subcategories under each category after choosing it.

(d) Suppliers and Maintainers

In this section, System Admin can define inventory suppliers and the maintainers of each supplier. To define a maintainer, user can choose related supplier, and add maintainers under it.

2.2.2.2 Inventory Management

This Module is the center for all processes about incoming inventories. Module has two sections:

- New Registry
- Instant Registry

(a) New Registry

If the inventory is ready to record, this is the section to use. First step is to print a new barcode for the new inventory. The system pops out an alert. After this alert, there comes the next page, where the user has a button to print to confirm the barcode by scanning it into the field.

After confirming the barcode, the next page is an information form. There are Category, Supplier, barcode number (auto entered), serial number, price, receipt date and extra info fields to be filled. Then, the next step is the product specifications part. The first part has the title group of related inventory category. The user can switch between titles and choose related specifications from the second list. Then the chosen specifications are added to the last field as related keywords of the inventory. And after clicking the button, the entry is on the database, and in ready state for other actions on the system.

(b) Instant Registry

This section allows user to print several barcodes when the inventory is not ready to record. For example, it may not be possible to bring label printer everywhere. So the user can print as much as needed, the system creates empty fields in the database to be filled in inventory registration process. When these barcodes are scanned, register page comes to complete the register, and the normal registry flow continues.

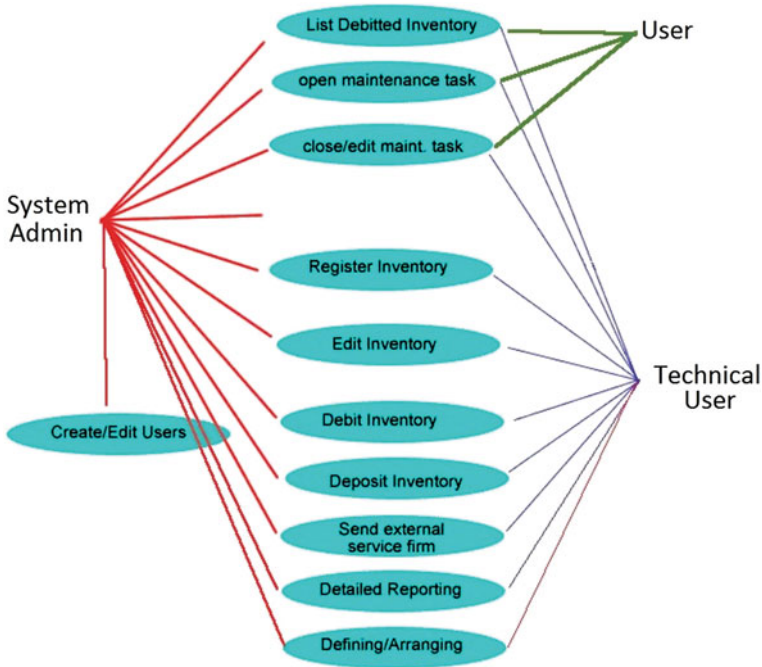


Fig. 2.4 Use case diagram of IMS

2.2.2.3 Service Management

Interior maintenance may not be sufficient, or may cause warranty problems, when inventories have technical problems. In this situation, they may need to be fixed by authorized maintainers. This module is for management of exterior maintenance.

There are three search fields in this section with Barcode No, Serial No parameters to reach the inventory or User Name parameter to reach the inventories of a specific user. After reaching the inventory, the window below comes to take action.

The user can view the service history of the inventory, and open a new service record to send the inventory to service. There is a pick list to select the service firm, and an info box to write a brief explanation. Then, another “close service record” link appears on the inventory information table.

When the inventory is returned from the service firm, user reaches the table below to close the record. User can write the information, process done to the text field and closes the service record, which will be seen in service history from that moment. Also user can use the “User Name” search box to reach a specific user, and list the inventories on users debit. Then, the same program flow can be used to take action.

2.2.2.4 Maintenance Management

This module is for interior inventory maintenance flow. There are two sections: first is to create new maintenance request for device users, the other is maintenance task listing for related worker.

The user will fill the required fields and then the request will be an active task for maintenance staff and will drop in their task table that they have on their interface if a staff member is assigned to the task. Otherwise, the task will drop into task pool. Then staff deals with the problem, and closes the task after writing info about actions taken. If the problem gets fixed before maintenance arrive, the user also can close the task.

2.2.2.5 Debit Management

This module sets the relationship between users and inventories. Basically, assign an inventory to a user, and performs other operations based on debiting.

2.2.2.6 Deposit Management

This module manages the barrowed items. Assign the inventories to desired person. There is also a chance to list deposited inventories in this module. It is a manner of tracking.

2.2.3 Users

Users have an authority hierarchy according to their section, and they are only allowed to use related modules except system admin. System admin has the authority to use every module in the system. In order to prevent improper use, the system logs every act, every edit. Although users can only use their modules, they can also use some other features of the system. There are three user types and use case diagram of the IMS is shown in Fig. 2.4:

1. System Admin
2. Technical Users, Accounting: Has authority on Inventory Management, Deposit Management, Debit Management, Service Management and also reporting features.
3. Basic Users: Basic users are only able to report their debit inventories and request service for them.

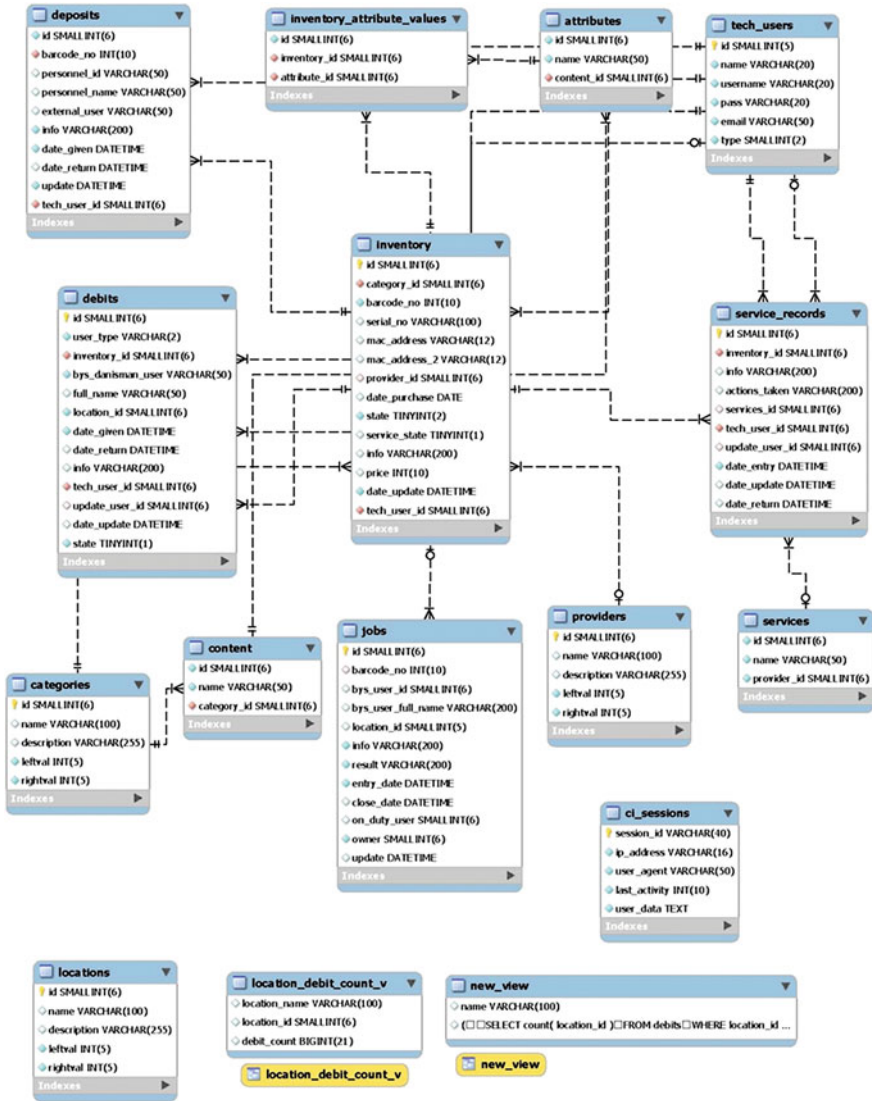


Fig. 2.5 Software architecture of—IMS

2.2.4 Login

The first step is “login” to the system. There are two login alternatives. First one is the system management, the other one is the end users. The login data is gathered from University database, so the users can login the system with their existing username and password.

Complete software architecture of IMS software is given in Fig. 2.5.

2.3 Conclusions

In the light of software engineering methods, we gather requirements, analyze and figure out the workflow, design methods and structures, construct scenarios, make tests, code the software, debug the faults and bugs, and finally we obtain a new IMS software. Our first aim is to develop the IMS software that is able to meet the requirements gathered. During the development process, there are many feedbacks, led us to re-design.

The software has the ability to track, to keep history, to give detailed reporting for each inventory. Also modules of the software manage the services and operations for inventories and users. Briefly, we could satisfy the user's requirements. It is also a scalable and a flexible solution. No matter how fast business's growing, proposed software can adapt to meet or exceed the requirements.

A new, functionally content rich software architecture model of an IMS is developed, presented, implemented and discussed. This level of implementation detail does not commonly appear in the literature. We believe this is a significant contribution.

References

1. Jason Gilmore W (2010) Beginning PHP and MySQL: from novice to professional, 4th edn. Apress, Berkely
2. Quigley E, Gargenta M (2006) PHP and MySQL by Example. Prentice Hall PTR, Upper Saddle River
3. Griffith A (2010) Codeigniter 1.7 Professional Development, Packt Publishing, Birmingham
4. DuBois P, Hinz S, Pedersen C (2005) MySQL 5.0 Certification Study Guide, MySQL Press
5. Zandstra M (2007) PHP Objects, Patterns, and Practice, 2nd edn. Apress, Berkely
6. Schwartz B, Zaitsev P, Tkachenko V, Zawodny J, Lentz A, Derek J. Balling (2008) High performance Mysql, 2nd Edn. O'Reilly, Sebastopol
7. Fultus Corporation (2010) Apache HTTP Server 2.2 Official Documentation—Volume II. Security and Server Programs. Fultus Corporation, Palo Alto
8. Chaffer J, Swedberg K (2007) JQuery reference guide: a comprehensive exploration of the popular javascript library. Packt Publishing, Birmingham