



The study named "Bookstore Inventory Program" which was prepared by Ulku Esra Koroğlu has been accepted by our jury as a Graduate Project in Computer Engineering.

Committee Members:

Chairman:

.....
Prof. Dr. Rahib ABIYEV
Department of Computer Engineering

Member:

.....
Asst. Prof. Dr. Kaan UYAR
Department of Computer Engineering

Member:

.....
Asst. Prof. Dr. Ümit İLHAN (Supervisor)
Department of Computer Engineering

I hereby confirm that the signatures above belong to the mentioned Faculty members.

.....
Prof. Dr. Rhib Abiyev
Chairman
Computer Engineering Department

Date: 14/06/2011



NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

BOOKSTORE INVENTORY PROGRAM

**Master Graduation Project
COM-590**

Student: Ülkü Esra Köroğlu

Supervisor: Dr. Ümit İlhan

Nicosia – 2011

ACKNOWLEDGMENT

Firstly, I would like to thank my supervisor Dr. Ümit İlhan, the vice dean, for his meaningful, subtle advice and for his patience in correcting both my stylistic and scientific errors.

I thank Asst. Doc. Dr. Kaan Uyar, the vice chairman of computer engineering, I started this project with his motivation and invaluable guidance.

Third, I thank my husband who is Ali Erdiñç K  rođlu, for his unique encouragement and technical support during the preparation of this project.

I thank my brother who is Erhan Okuyan, for his endless support, when I faced with setbacks in this project or of course real life, he always on my side and gave me the confidence.

Finally, I would also like to thank all my friend for their advice and support, especially Batur Kaçamak, Ođuz Yarımtepe, Fatih Mert.

ABSTRACT

Management of the bookstore implementing revenue management has many decisions to make regarding the details of the implementation. We study to find an effective stocking policy for bookstore.

The aim of this program is a safe way to keep all the inventory in digital center which is computer, and to provide an easier way while employees sales to book. Obtaining the correct balance is not easy for the bookstore and the stock control department should work closely with the purchasing and marketing departments.

The basic building blocks for the Inventory Management system and Inventory Control activities are: Sales Forecasting or Demand Management, Sales and Operations Planning, Production Planning, Material Requirements Planning, Inventory Reduction.

The emphases on each area will vary depending on the company and how it operates, and what requirements are placed on it due to market demands. Each of the areas above will need to be addressed in some form or another to have a successful program of Inventory Management and Inventory Control. This inventory program performed by using php-mysql-apache.

TABLE OF CONTENTS

ACKNOWLEDGMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
LIST OF ABBREVIATIONS	vi
INTRODUCTION	1
 CHAPTER ONE : INVENTORY CONTROL	 2
1.1. Introduction	2
1.2. What is Inventory Control?	2
1.3. What is Inventory Management?	4
1.3.1 Mismanaged Inventories	4
1.3.2 Booming Inventory Management	4
1.3.3 Ordering and Pricing	4
1.3.4 POS Software	5
1.3.5 Technological Advances in Managing Inventories	5
1.4. How to Manage Inventory	6
1.4.1. Purchase the right products	6
1.4.2. Buying Correct Quantities	6
1.4.3. This knowledge is acquired by analysis and experience	7
1.4.4. Managing Inventory	7
1.5. Safety stock	7
1.5.1. Reasons for safety stock	8
1.5.2. Reducing safety stock	8
1.5.3. Inventory policy	9
1.6. Inventory types	10
1.7. Value of inventory	10
1.8. Stock control methods	10
1.8.1. Just In Time (JIT)	10
1.8.2. Re-order lead time	10
1.8.3. Economic Order Quantity (EOQ)	11
1.8.4. Batch control	11
1.8.5. First in, first out	11
1.9. Inventory control administration	11
1.9.1. Health and safety	12
1.10. Purpose of Inventory Management	12
1.11. Summary	14

CHAPTER TWO : DATABASE MANAGEMENT SYSTEM	15
2.1. Introduction	15
2.2. Overview	17
2.3. History	18
2.3.1. 1960s Navigational DBMS	18
2.3.2. 1970s Relational DBMS	19
2.4. End 1970s SQL DBMS	21
2.5. 1980s Object Oriented Databases	21
2.6. DBMS building blocks	22
2.6.1. Components of DBMS	22
2.6.2. Modeling language	25
2.7. Data structure	25
2.8. Database Query Language	25
2.9. Transaction mechanism	25
2.9.1. ACID	28
2.10. DBMS topics	28
2.10.1. External, Logical and Internal view	29
2.10.2. DBMS features and capabilities	29
2.10.3. DBMS simple definition	31
2.10.4. Meta-data repository	31
2.10.5. Current trends	32
2.11. Summary	32
CHAPTER THREE : PHP, MYSQL AND APACHE	33
3.1. Introduction	33
3.2. PHP Programming Language	34
3.2.1 PHP scripting	35
3.2.2. Reasons for using PHP	35
3.2.3. Structure Of PHP Script	36
3.2.4. PHP Database Manipulation	38
3.2.5. The Benefits of PHP Includes	40
3.2.6. PHP on Linux Versus PHP on Windows	41
3.2.7. PHP vs Java vs Ruby	42
3.3. What is MySQL	44
3.3.1 How do PHP and MySQL work together?	45
3.3.2. Running PHP and MySQL	45
3.4. Phpmyadmin	46
3.4.1. Features	47
3.4.2. Ability of Phpmyadmin	47
3.5. Apache HTTP Server	48
3.5.1. Features	49
3.5.2. Use	50
3.5.3. Performance	50

CHAPTER FOUR : USER INTERFACE AND DATABASE STRUCTURE OF BOOKSTORE PROGRAM	51
4.1. Introduction	51
4.2. Bookstore Login Pages	52
4.3. Main Pages of Bookstore	53
4.4. Password Operation Page	53
4.5. Critical Inventory Alert Pages of Bookstore	54
4.6. Book Registration Page	55
4.7. Invoice Entry Process Page	56
4.8. Book Sales Page	57
4.9. Stock Information Page	60
4.10 Deposit Book Page	61
4.11. Listing Process Page	62
4.12. Barcode Transactions Page	64
4.13. BookStore Database Structure	66
4.13.1. Bs_barcode Table	67
4.13.2. Bs_depo Table	67
4.13.3. Bs_deposit Table	68
4.13.4. Bs_hareket Table	68
4.13.5. Bs_payment Table	69
4.13.6. Bs_sales Table	69
4.13.7. Bs_salesinfo Table	70
4.13.8. Bs_stok Table	70
4.13.9. Bs_stokbilgi Table	71
4.13.10. Bs_users Table	71
4.13.11. Remove-Tables	72
CONCLUSION	73
REFERENCES	74
APPENDIX A	75

LIST OF ABBREVIATIONS

SQL	Structured Query Language
DBMS	Database Management System
HTML	HyperText Markup Language
XML	Extensible Markup Language
ASP	Active Server Pages
PHP	HyperText Preprocessor
GUI	Graphical User Interface
MVC	Model-view-controller

INTRODUCTION

Inventory Management system provides information to efficiently manage the flow of materials, effectively utilize people and equipment, coordinate internal activities and communicate with customers. Inventory Management does not make decisions or manage operations.

The Inventory Management system and the Inventory Control Process provides information to efficiently manage the flow of materials, effectively utilize people and equipment, coordinate internal activities, and communicate with customers.

Inventory control and management are most important for bookstore administration. This work was initiated for that reason. We designed the program for intranet because secure access also major point in bookstore. We should control who can access the program, Of course we could make also with internet this situation but intranet keep minimum all security problem and the primary goal of the this project is to improve staff efficiency. Unnecessary content deters from productivity.

The objective of this project is to perform book sales by tracking inventory in a safe manner. Project was developed in PHP language, the web server has been selected as apache and database as MySQL. The project consist of introduction, three chapter and conclusion.

Chapter One describes the inventory control for bookstore. Also this chapter presents the general representation how to manage inventory and technological advances in managing inventories.

Chapter Two describes the database management system and presents to database management model, according to year. Many DBMS (Database Management System) packages provide Fourth-generation programming language (4GLs) and other application development features. This subject also described in this chapter.

Chapter Three is devoted to the PHP language, MySQL Database Server and Apache HTTP server. In this chapter described their definitions, their relationships with each other and, the application areas are explained.

CHAPTER ONE: INVENTORY CONTROL

1.1. Introduction

Inventory control means keeping the overall costs associated with having inventory as low as possible without creating problems. This is also sometimes called stock control. It is an important part of any business that must have a stock of products or items on hand. Correctly managing inventory control is a delicate balance at all times between having too much and too little in order to maximize profits. The costs associated with holding stock, running out of stock, and placing orders must all be looked at and compared in order to find the right formula for a particular business. Inventory control, is also used to show how much stock you have at any one time, and how you keep track of it.

It applies to every item you use to produce a product or service, from raw materials to finished goods. It covers stock at every stage of the production process, from purchase and delivery to using and re-ordering the stock.

Efficient stock control allows you to have the right amount of stock in the right place at the right time. It ensures that capital is not tied up unnecessarily, and protects production if problems arise with the supply chain.

This guide explains different stock control methods, shows you how to set one up and tells you where to find more information.

The inventory control problem is the problem faced by a firm that must decide how much to order in each time period to meet demand for its products. The problem can be modeled using mathematical techniques of optimal control, dynamic programming and network optimization. The study of such models is part of inventory theory.

Inventory theory (or more formally the mathematical theory of inventory and production) is the sub-specialty within operations research that is concerned with the design of production/inventory systems to minimize costs. It studies the decisions faced by firms and the military in connection with manufacturing, warehousing, supply chains, spare part allocation and so on; it provides the mathematical foundation for logistics.

1.2. What is Inventory Control?

Inventory control is the delicate balance of the costs versus profits associated with having stock on hand.

Inventory control means keeping the overall costs associated with having inventory as low as possible without creating problems. This is also sometimes called stock control. It is an important part of any business that must have a stock of products or items on hand. Correctly managing inventory control is a delicate balance at all times between having too much and too little in order to maximize profits. The costs associated with holding stock, running out of stock, and placing orders must all be looked at and compared in order to find the right formula for a particular business.

It is impossible to have an unlimited supply on hand, for a number of different reasons. Many businesses simply don't have enough money to keep excessively large inventories. There are costs associated with purchasing the items as well as storing them, and having too many products leads to further losses when they don't move off of the shelves.

At the same time, there are issues with inventory control when there isn't enough stock on hand. One common problem is running out of inventory, which is caused by trying to reduce inventory costs too much. This is something that no business wants to have happen, but it happens to virtually all of them at some point. Even the largest stores run out of certain products from time to time when they sell or use more than they expected. This can cause financial losses when inventory is not available for customers to purchase. Part of inventory control is trying to minimize shortages so these are rare occurrences. Most businesses expect they will have shortages on occasion and they have calculated that the small loss is worth the money saved by not having an overstock.

Another important element of inventory control is called reorder point. Businesses need to think ahead and calculate the best time for reordering products. Doing so too soon may cause financial difficulties or running out of space. On the other hand, waiting too long to reorder will result in a shortage and running out of inventory before the next shipment arrives. When figuring out a reorder point, it's necessary to calculate how long it will take the shipment to arrive and the amount of demand for a particular item. The overhead costs, fees, and shipping expenses of ordering large versus small quantities should also be looked at.

Inventory control is an ongoing process that is rarely, if ever, executed perfectly. Experience, expertise, and practice help people to make the best decisions regarding stock, but there are always unknown circumstances and variables. Stores can make good estimates about how many of a specific product they will sell, but they get things wrong

from time to time. This is unavoidable. Inventory control can break a business if it is executed poorly, because either expenses will be too high or customers will get tired of dealing with shortages and find another place to spend their money.

1.3. What is Inventory Management?

Effective inventory management is a crucial aspect of a successful business practice.

Inventory management is an integral part of a successful business. Inventories typically consist of goods, raw materials and finished products. Each of these elements translates into money for the business owner. The key to profitability is a carefully balanced inventory.

Balanced inventories are important because many businesses rely on its stock of items to make a profit. Stockpiles that never move from the shelves do little good for the company. A proper balance is of the utmost importance.

1.3.1. Mismanaged Inventories

Inventory mismanagement can be detrimental to a business, especially considering the weight these items carry. Inventories that run out of control can lead to significant losses that the company may not be able to recoup. Considerable investment is required to develop adequate stock. Poorly managed supplies lead to profit loss.

1.3.2. Booming Inventory Management

Properly managing supplies requires the ability to create a balance. Part of the balancing approach should include aspects of inventories that many business owners fail to recognize. Issues that may be underestimated include:

- Storage cost
- Insurance
- Taxes
- Ordering dilemmas
- Pricing

Storage costs, insurance and taxes are important aspects of stocking shelves and keeping necessary supplies at hand. These costs should be figured into the purchasing budget for the stock. The upfront purchasing costs are complicated with ordering dilemmas and pricing considerations.

1.3.3. Ordering and Pricing

Managing inventories can be complicated, but some considerations can make the process much easier. Management may be concerned primarily with having a balanced

stock while keeping supplies readily available without overstocking the shelves. Other considerations are important as well.

- Balanced assortment of items
- Quick, efficient turnover
- Maintaining service quality
- Stocking up-to-date items
- High volume purchases
- Cost control

Successful inventory management may seem as if it requires psychic abilities, and while a peek into the future can help, managers can fare pretty well by addressing managerial performance. Creating realistic goals backed by evaluations can be beneficial. The data collected in evaluations can give managers insight into the best approach for purchasing stock in the future.

1.3.4. POS Software

One of the most valuable tools available to business managers is POS software. Point of Sale software offers an array of features that provide the guidance necessary for proper inventory and business management. A good program offers more than simply tracking inventories and sales.

Different programs offer different features, so it is important to consider the company's specific needs. The right software system can save a business considerable time and money in a number of ways. The benefits of a Point of Sale software program range from inventory control to accounts receivable, depending on the program.

1.3.5. Technological Advances in Managing Inventories

The latest technologies for businesses offer a wide range of benefits that make managerial tasks much easier. The intricacies of balancing supplies can be as simple as reviewing a tracking summary. Collected data may include the latest purchasing trends and higher demands for specific services.

Technological advances in the realm of inventories provide the necessary tools to help managers make reliable decisions according to current trends in their industries. Software products have a significant impact on the efficiency of a company's performance. Well-balanced inventories translate into profits and effective inventory management can be very lucrative.

1.4. How to Manage Inventory

Techniques to purchase, stock and manage inventory efficiently. Learning to manage inventory efficiently and correctly will allow a company to stock a sufficient quantity of product to meet customer needs. Not enough inventory causes delays and is bad for business. Excessive inventory costs a business money. A retail business only makes money when its stock is sold, maintaining the point inventory until it has sold incurs certain expenses. These expenses include:

- Interest on any loans used to purchase goods.
- Cost of storage space.
- Cost of moving stock and arranging it for display.
- Cost of time spent managing the inventory.
- Any cost of damaged or stolen inventory.
- payroll

A large amount of a business's money may be tied up in inventory. It is possible to manage inventory so that profits are maximized and costs are minimized.

1.4.1. Purchase the right products

- Use market research to identify the proper products for each target market.
- Analyze sales from previous years.
- Look for new products for each target market.
- Study the economy forecast for the coming period and determine how target market purchasing power might be affected.
- Compare the effects competitors sales have on your sales.
- Learn from experience what products to buy and in what quantities to buy them.

1.4.2. Buying Correct Quantities

It is necessary to maintain the proper variety and quantity of inventory to satisfy the target market. At the same time it is important to not get caught with an overstock of obsolete items. To do this you must:

- Know what product to order
- Know how much product to order
- Know when to order the product
- Know when to expect the products to arrive
- Know what quantities of product should be in stock during the business cycle
- Know when reorders should no longer be submitted

- Know when products should no longer be in stock

1.4.3. This knowledge is acquired by analysis and experience:

- Make sure suppliers are reputable and will provide products in the quantities required.
- Study lead times for supply and delivery to ensure ordering at the correct time, or determining the delivery time for the customer if additional products have to be ordered.
- Don't over buy just to take advantage of special supplier deals.
- Review buying plans weekly and adjust as needed. If inventory is selling or being used faster than expected, look to source more product as soon as the trend is spotted.
- Establish stock levels (minimum, maximum and at what point to re-order).

1.4.4. Managing Inventory

The easiest way to manage inventory is with a computer inventory management system. The time required to accurately manage inventory will be minimized by using one of these systems:

- Point-of-sale terminals-automatically update computerized inventory levels
- Job costing and inventory systems-automatically update computerized inventory levels
- Barcodes and barcode readers-allow inventory to be input and stock takes to be completed quickly
- Electronic Supplier product catalogs-allow inventory details to be loaded automatically either via the internet or CD/DVD

Once you ensure the stock management system and its reports accurately allow you to manage inventory:

- Remove all goods from the system as soon as they are sold.
- Check regularly to make sure the stock system is accurate against physical stock quantities with spot checks and stock takes.
- Review stock reports weekly and identify products that are not selling so that appropriate steps may be taken.

1.5. Safety stock

Safety stock is a term used by inventory specialists to describe a level of extra stock that is maintained below the cycle stock to buffer against stockouts. Safety stock

(also called buffer stock) exists to counter uncertainties in supply and demand. Safety stock is defined as extra units of inventory carried as protection against possible stockouts (shortfall in raw material or packaging). By having an adequate amount of safety stock on hand, a company can meet a sales demand which exceeds the demand they forecasted without altering their production plan. It is held when an organization cannot accurately predict demand and/or lead time for the product. It serves as an insurance against stockouts.

With a new product, safety stock can be utilized as a strategic tool until the company can judge how accurate their forecast is after the first few years, especially when used with a material requirements planning worksheet. The less accurate the forecast, the more safety stock is required. With a material requirements planning (MRP) worksheet a company can judge how much they will need to produce to meet their forecasted sales demand without relying on safety stock. However, a common strategy is to try and reduce the level of safety stock to help keep inventory costs low once the product demand becomes more predictable. This can be extremely important for companies with a smaller financial cushion or those trying to run on lean manufacturing, which is aimed towards eliminating waste throughout the production process.

The amount of safety stock an organization chooses to keep on hand can dramatically affect their business. Too much safety stock can result in high holding costs of inventory. In addition, products which are stored for too long a time can spoil, expire, or break during the warehousing process. Too little safety stock can result in lost sales and, thus, a higher rate of customer turnover. As a result, finding the right balance between too much and too little safety stock is essential.

1.5.1. Reasons for safety stock

Safety stocks enable organizations to satisfy customer demand in the event of these possibilities:

- Supplier may deliver their product late or not at all
- The warehouse may be on strike
- A number of items at the warehouse may be of poor quality and replacements are still on order
- A competitor may be sold out on a product, which is increasing the demand for your products
- Random demand (in reality, random events occur)

- Machinery breakdown
- Unexpected increase in demand

1.5.2. Reducing safety stock

Safety stock is used as a buffer to protect organizations from stockouts caused by inaccurate planning or poor schedule adherence by suppliers. As such, its cost (in both material and management) is often seen as a drain on financial resources which results in reduction initiatives. In addition, time sensitive goods such as food, drink, and other perishable items could spoil and go to waste if held as safety stock for too long. Various methods exist to reduce safety stock, these include better use of technology, increased collaboration with suppliers, and more accurate forecasting. In a lean supply environment, lead times are reduced which can help minimize safety stock levels thus reducing the likelihood and impact of stockouts. Due to the cost of safety stock, many organizations opt for a service level led safety stock calculation; for example, a 95% service level could result in stockouts, but is at a level which is satisfactory to the company. The lower the service level, the lower the requirement for safety stock.

An Enterprise Resource Planning system (ERP system) can also help an organization reduce its level of safety stock. Most ERP systems provide a type of Production Planning module. An ERP module such as this can help a company develop highly accurate and dynamic sales forecasts and sales and operations plans. By creating more accurate and dynamic forecasts, a company reduces their chance of producing insufficient inventory for a given period and, thus, should be able to reduce the amount of safety stock which they require. In addition, ERP systems use established formulas to help calculate appropriate levels of safety stock based on the previously developed production plans. While an ERP system aids an organization in estimating a reasonable amount of safety stock, the ERP module must be set up to plan requirements effectively.

1.5.3. Inventory policy

The size of the safety stock depends on the type of inventory policy that is in effect. An inventory node is supplied from a "source" which fulfills orders for the considered product after a certain replenishment lead time. In a "periodic review" inventory policy the inventory level is checked periodically (such as once a month) and an order is placed at that time if necessary; in this case the risk period is equal to the time until the next review plus the replenishment lead time. On the other hand, if the inventory policy is a "continuous review" policy (such as an Order point-Order Quantity policy or an Order Point-Order Up To policy) the inventory level is being check

continuously and orders can be placed immediately, so the risk period is just the replenishment lead time. Therefore "continuous review" inventory policies can make do with a smaller safety stock.

1.6. Inventory types

Everything you use to make your products, provide your services and to run your business is part of your stock.

There are four main types of stock:

- raw materials and components - ready to use in production
- work in progress - stocks of unfinished goods in production
- finished goods ready for sale
- consumables - for example, fuel and stationery

1.7. Value of inventory

You can categorise stock further, according to its value. For example, you could put items into low, medium and high value categories. If your stock levels are limited by capital, this will help you to plan expenditure on new and replacement stock.

You may choose to concentrate resources on the areas of greatest value. However, low-cost items can be crucial to your production process and should not be overlooked.

1.8. Stock control methods

There are several methods for controlling stock, all designed to provide an efficient system for deciding what, when and how much to order.

You may opt for one method or a mixture of two or more if you have various types of stock. For further information, see the page in this guide on types of stock.

- **Minimum stock level** - you identify a minimum stock level, and re-order when stock reaches that level. This is known as the Re-order Level.
- **Stock review** - you have regular reviews of stock. At every review you place an order to return stocks to a predetermined level.

1.8.1. Just In Time (JIT)

This aims to reduce costs by cutting stock to a minimum - see our guide on how to avoid the problems of overtrading. Items are delivered when they are needed and used immediately. There is a risk of running out of stock, so you need to be confident that your suppliers can deliver on demand. Find out more about JIT production on the Manufacturing Advisory Service (MAS) website (registration required) - Opens in a new window.

These methods can be used alongside other processes to refine the stock control system.

For example:

1.8.2. Re-order lead time

Allows for the time between placing an order and receiving it.

1.8.3. Economic Order Quantity (EOQ)

A standard formula used to arrive at a balance between holding too much or too little stock. It's quite a complex calculation, so you may find it easier to use stock control software. Find out about the EOQ formula on the InventoryOps website - Opens in a new window.

1.8.4. Batch control

Managing the production of goods in batches. You need to make sure that you have the right number of components to cover your needs until the next batch.

If your needs are predictable, you may order a fixed quantity of stock every time you place an order, or order at a fixed interval - say every week or month. In effect, you're placing a standing order, so you need to keep the quantities and prices under review.

1.8.5. First in, first out

A system to ensure that perishable stock is used efficiently so that it doesn't deteriorate. Stock is identified by date received and moves on through each stage of production in strict order.

1.9. Inventory control administration

There are many administrative tasks associated with stock control. Depending on the size and complexity of your business, they may be done as part of an administrator's duties, or by a dedicated stock controller.

For security reasons, it's good practice to have different staff responsible for finance than those responsible for stock.

Typical paperwork to be processed includes:

- delivery and supplier notes for incoming goods
- purchase orders, receipts and credit notes
- returns notes
- requisitions and issue notes for outgoing goods

Stock can tie up a large slice of your business capital, so accurate information about stock levels and values is essential for your company's accounting.

Figures should be checked systematically, either through a regular audit of stock - **stocktaking** - or an ongoing programme of checking stock - **rolling stocktake**.

If the figures don't add up, you need to investigate as there could be stock security problems or a failure in the system.

1.9.1. Health and safety

Health and safety aspects of stock control are related to the nature of the stock itself. Issues such as where and how items are stored, how they are moved and who moves them might be significant - depending on what they are.

You might have hazardous materials on your premises, goods that deteriorate with time or items that are very heavy or awkward to move.

1.10. Purpose of Inventory Management

Inventory management must tie together the following objectives ,to ensure that there is continuity between functions :

- Company's Strategic Goals
- Sales Forecasting
- Sales & Operations Planning
- Production & Materials Requirement Planning.

Inventory Management must be designed to meet the dictates of market place and support the company's Strategic Plan . The many changes in the market demand , new opportunities due to worldwide marketing , global sourcing of materials and new manufacturing technology means many companies need to change their Inventory Management approach and change the process for Inventory Control.

Inventory is defined as the blocked Working Capital of an organization in the form of materials . As this is the blocked Working Capital of organization, ideally it should be zero. But we are maintaining Inventory . This Inventory is maintained to take care of fluctuations in demand and lead time. In some cases it is maintained to take care of increasing price tendency of commodities or rebate in bulk buying.

Traditional Supply Chain solutions such as Materials Requirement Planning, Inventory Control, typically focuses on implementing more rapid and efficient systems to reduce the cost of communicating information between and across the Inventory links in the SCM.COM focuses in optimizing the total investment of materials cost and workload for every Inventory item throughout the chain from procurement of raw materials to finished goods Inventory . Optimization means providing a balance of

supply to meet the demand at a minimum total cost, Inventory level and workload to meet customers service goal for each items in the link of Inventory Chain.

It is strategic in the sense that top management sets goals. These include deployment strategies (Push versus Pull), control policies, the determination of the optimal levels of order quantities and reorder points and setting safety stock levels. These levels are critical, since they are primary determinants of customer service levels.

Keeping in view all concerns, the latest concept of Vendor Managed Inventory is used to optimize the Inventory. We are entering into Vendor Managed Inventory, Annual Rate Contracts with manufacturers or their authorized dealers, who maintain Inventory on our behalf and supply the items as and when required VMI reduces stock-outs and optimize inventory in supply chain. Some features of VMI include :

- Shortening of Supply Chain
- Centralized Forecasting
- Frequent communication of inventory, stock-outs and planned promotions
- Trucks are filled in a prioritized order, e.g. items that are expected to stock out have top priority then items that are furthest below targeted stock levels then advance shipments of promotional items

Despite the many changes that companies go through, the basic principles of Inventory Management and Inventory Control remain the same. Some of the new approaches and techniques are wrapped in new terminology, but the underlying principles for accomplishing good Inventory Management and Inventory activities have not changed.

The Inventory Management system and the Inventory Control Process provides information to efficiently manage the flow of materials, effectively utilize people and equipment, coordinate internal activities, and communicate with customers. Inventory Management and the activities of Inventory Control do not make decisions or manage operations; they provide the information to Managers who make more accurate and timely decisions to manage their operations.

The basic building blocks for the Inventory Management system and Inventory Control activities are:

- Sales Forecasting or Demand Management
- Sales and Operations Planning
- Production Planning
- Material Requirements Planning

Inventory Reduction

The emphases on each area will vary depending on the company and how it operates, and what requirements are placed on it due to market demands. Each of the areas above will need to be addressed in some form or another to have a successful program of Inventory Management and Inventory Control.

Inventory is usually a distributor's largest asset. But many distributors aren't satisfied with the contribution inventory makes towards the overall success of their business:

- The wrong quantities of the wrong items are often found on warehouse shelves. Even though there maybe a lot of surplus inventory and dead stock in their warehouse(s), backorders and customer lost sales are common. The material a distributor has committed to stock isn't available when customers request it.

- Computer inventory records are not accurate. Inventory balance information in the distributor's expensive computer system does not accurately reflect what is available for sale in the warehouse.

- The return on investment is not satisfactory. The company's profits, considering its substantial investment in inventory, is far less than what could be earned if the money were invested elsewhere.

1.11. Summary

Efficiently controlling the inventory of a retail store is critical to the long term profitability and survival of the store. Retailers make a sizeable investment in the inventory they keep every day. If this investment is not protected, the retailer may face financial losses that cannot be overcome.

There are many areas of inventory control that need constant attention. From the daily delivery of merchandise to the annual stocktake, each aspect of the process is important and should not be overlooked

CHAPTER TWO: DATABASE MANAGEMENT SYSTEM

2.1. Introduction

Database Management Systems (DBMS) are important. They underpin all the activities of a library management system by providing the basic storage and retrieval technology. The library application software sends data to and receives data from the DBMS which, if it is working properly, is hardly noticed at all. Yet great claims are made for different types of database and their particular offerings. You should at least be able to understand the basics to understand what you might be getting - or missing when you choose a Library Management System (LMS).

A Database Management System (DBMS) is a set of computer programs that controls the creation, maintenance, and the use of a database. It allows organizations to place control of database development in the hands of database administrators (DBAs) and other specialists. A DBMS is a system software package that helps the use of integrated collection of data records and files known as databases. It allows different user application programs to easily access the same database. DBMSs may use any of a variety of database models (Figure 2.1.), such as the network model (Figure 2.2.) or relational model (Figure 2.3.) . In large systems, a DBMS allows users and other software to store and retrieve data in a structured way. Instead of having to write computer programs to extract information, user can ask simple questions in a query language. Thus, many DBMS packages provide Fourth-generation programming language (4GLs) and other application development features. It helps to specify the logical organization for a database and access and use the information within a database. It provides facilities for controlling data access, enforcing data integrity, managing concurrency, and restoring the database from backups. A DBMS also provides the ability to logically present database information to users.

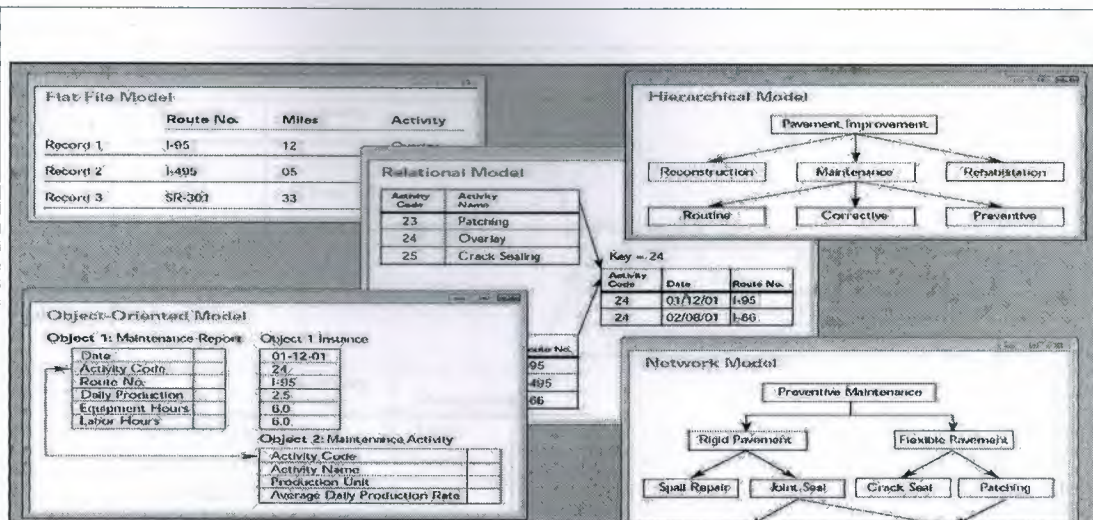


Figure 2.1. A database model or database schema is the structure or format of a database, described in a formal language supported by the database management system, In other words, a "database model" is the application of a data model when used in conjunction with a database management system.

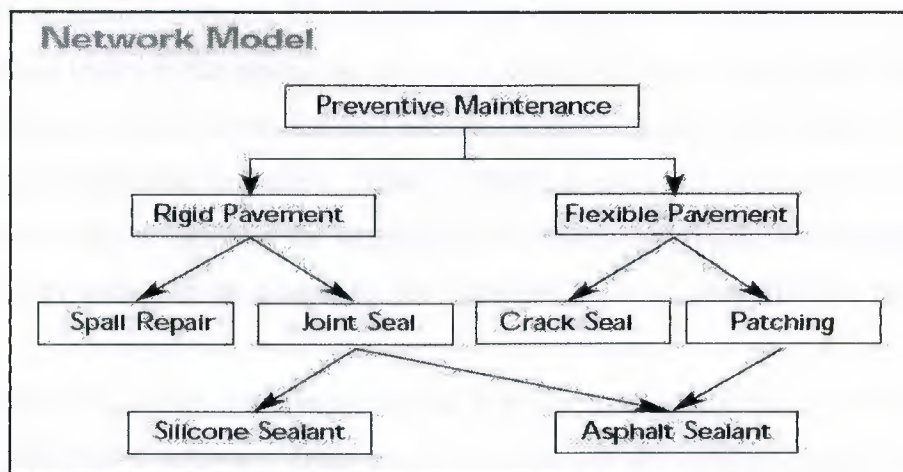


Figure 2.2. The network model is a database model conceived as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is notrestricted to being a hierarchy or lattice.

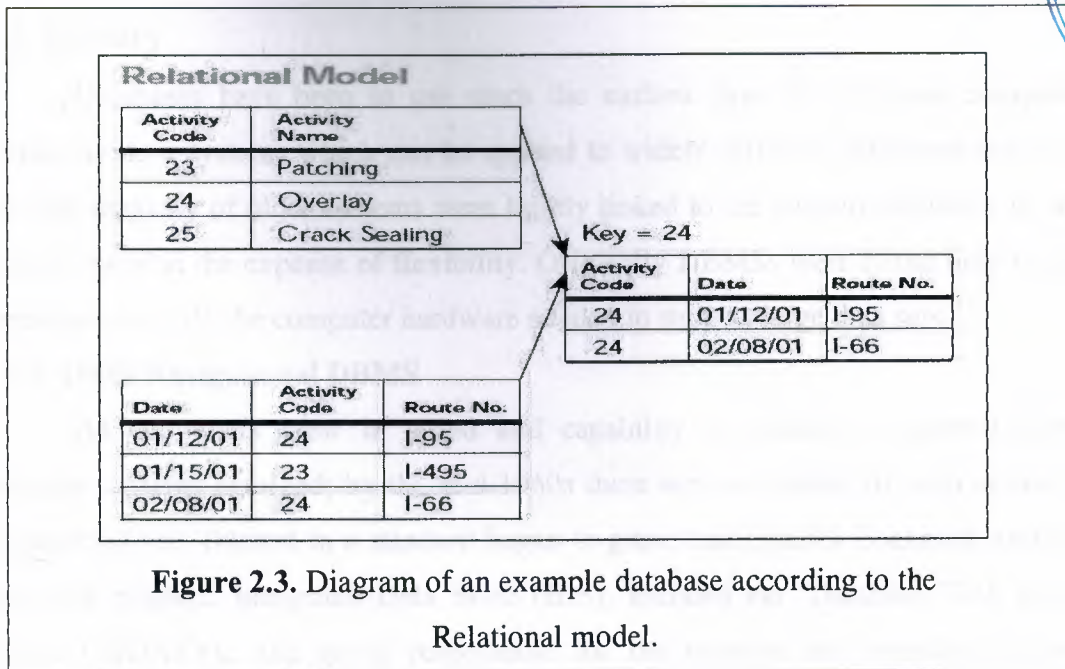


Figure 2.3. Diagram of an example database according to the Relational model.

2.2. Overview

A DBMS is a set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMSs are categorized according to their data structures or types. The DBMS accepts requests for data from an application program and instructs the operating system to transfer the appropriate data. The queries and responses must be submitted and received according to a format that conforms to one or more applicable protocols. When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Database servers are computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions.

2.3. History

Databases have been in use since the earliest days of electronic computing. Unlike modern systems which can be applied to widely different databases and needs, the vast majority of older systems were tightly linked to the custom databases in order to gain speed at the expense of flexibility. Originally DBMSs were found only in large organizations with the computer hardware needed to support large data sets.

2.3.1. 1960s Navigational DBMS

As computers grew in speed and capability, a number of general-purpose database systems emerged; by the mid-1960s there were a number of such systems in commercial use. Interest in a standard began to grow, and Charles Bachman, author of one such product, Integrated Data Store (IDS), founded the "Database Task Group" within CODASYL, the group responsible for the creation and standardization of COBOL. In 1971 they delivered their standard, which generally became known as the "Codasyl approach", and soon there were a number of commercial products based on it available.

The Codasyl approach was based on the "manual" navigation of a linked data set which was formed into a large network. When the database was first opened, the program was handed back a link to the first record in the database, which also contained pointers to other pieces of data. To find any particular record the programmer had to step through these pointers one at a time until the required record was returned. Simple queries like "find all the people in India" required the program to walk the entire data set and collect the matching results. There was, essentially, no concept of "find" or "search". This might sound like a serious limitation today, but in an era when the data was most often stored on magnetic tape such operations were too expensive to contemplate anyway.

IBM also had their own DBMS system in 1968, known as IMS. IMS was a development of software written for the Apollo program on the System/360. IMS was generally similar in concept to Codasyl, but used a strict hierarchy for its model of data navigation instead of Codasyl's network model. Both concepts later became known as navigational databases due to the way data was accessed, and Bachman's 1973 Turing Award award presentation was The Programmer as Navigator. IMS is classified as a hierarchical database. IMS and IDMS, both CODASYL databases, as well as CINCOMs TOTAL database are classified as network databases.

2.3.2. 1970s Relational DBMS

Edgar Codd worked at IBM in San Jose, California, in one of their offshoot offices that was primarily involved in the development of hard disk systems. He was unhappy with the navigational model of the Codasyl approach, notably the lack of a "search" facility which was becoming increasingly useful. In 1970, he wrote a number of papers that outlined a new approach to database construction that eventually culminated in the groundbreaking A Relational Model of Data for Large Shared Data Banks.

In this paper, he described a new system for storing and working with large databases. Instead of records being stored in some sort of linked list of free-form records as in Codasyl, Codd's idea was to use a "table" of fixed-length records (Figure 2.4.). A linked-list system would be very inefficient when storing "sparse" databases where some of the data for any one record could be left empty. The relational model solved this by splitting the data into a series of normalized tables, with optional elements being moved out of the main table to where they would take up room only if needed.

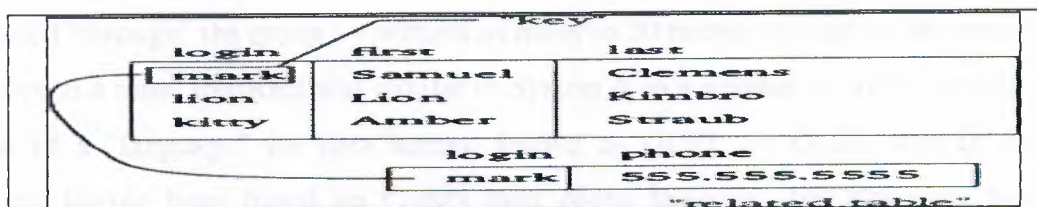


Figure 2.4. Relational Model for databases, highlighting the Key. The image is of two tables, one storing "login," "first name," and "last name." Another table stores "login" and "phone number." The image illustrates using a unique key to link a phone number to a first and last name.

For instance, a common use of a database system is to track information about users, their name, login information, various addresses and phone numbers. In the navigational approach all of these data would be placed in a single record, and unused items would simply not be placed in the database. In the relational approach, the data would be normalized into a user table, an address table and a phone number table (for instance). Records would be created in these optional tables only if the address or phone numbers were actually provided.

Linking the information back together is the key to this system. In the relational model, some bit of information was used as a "key", uniquely defining a particular record. When information was being collected about a user, information stored in the

optional (or related) tables would be found by searching for this key. For instance, if the login name of a user is unique, addresses and phone numbers for that user would be recorded with the login name as its key. This "re-linking" of related data back into a single collection is something that traditional computer languages are not designed for.

Just as the navigational approach would require programs to loop in order to collect records, the relational approach would require loops to collect information about any one record. Codd's solution to the necessary looping was a set-oriented language, a suggestion that would later spawn the ubiquitous SQL. Using a branch of mathematics known as tuple calculus, he demonstrated that such a system could support all the operations of normal databases (inserting, updating etc.) as well as providing a simple system for finding and returning sets of data in a single operation.

Codd's paper was picked up by two people at the Berkeley, Eugene Wong and Michael Stonebraker. They started a project known as INGRES using funding that had already been allocated for a geographical database project, using student programmers to produce code. Beginning in 1973, INGRES delivered its first test products which were generally ready for widespread use in 1979. During this time, a number of people had moved "through" the group — perhaps as many as 30 people worked on the project, about five at a time. INGRES was similar to System R in a number of ways, including the use of a "language" for data access, known as QUEL — QUEL was in fact relational, having been based on Codd's own Alpha language, but has since been corrupted to follow SQL, thus violating much the same concepts of the relational model as SQL itself.

IBM itself did one test implementation of the relational model, PRTV, and a production one, Business System 12, both now discontinued. Honeywell did MRDS for Multics, and now there are two new implementations: Alphora Dataphor and Rel. All other DBMS implementations usually called relational are actually SQL DBMSs. In 1968, the University of Michigan began development of the Micro DBMS relational database management system. It was used to manage very large data sets by the US Department of Labor, the Environmental Protection Agency and researchers from University of Alberta, the University of Michigan and Wayne State University. It ran on mainframe computers using Michigan Terminal System. The system remained in production until 1996.

2.4. End 1970s SQL DBMS

IBM started working on a prototype system loosely based on Codd's concepts as System R in the early 1970s. The first version was ready in 1974/5, and work then started on multi-table systems in which the data could be split so that all of the data for a record (much of which is often optional) did not have to be stored in a single large "chunk". Subsequent multi-user versions were tested by customers in 1978 and 1979, by which time a standardized query language, SQL (Structured Query Language), had been added. Codd's ideas were establishing themselves as both workable and superior to Codasyl, pushing IBM to develop a true production version of System R, known as SQL/DS, and, later, Database 2 (DB2).

Many of the people involved with INGRES became convinced of the future commercial success of such systems, and formed their own companies to commercialize the work but with an SQL interface. Sybase, Informix, NonStop SQL and eventually Ingres itself were all being sold as offshoots to the original INGRES product in the 1980s. Even Microsoft SQL Server is actually a re-built version of Sybase, and thus, INGRES. Only Larry Ellison's Oracle started from a different chain, based on IBM's papers on System R, and beat IBM to market when the first version was released in 1978.

Stonebraker went on to apply the lessons from INGRES to develop a new database, Postgres, which is now known as PostgreSQL. PostgreSQL is often used for global mission critical applications (the .org and .info domain name registries use it as their primary data store, as do many large companies and financial institutions).

In Sweden, Codd's paper was also read and Mimer SQL was developed from the mid-70s at Uppsala University. In 1984, this project was consolidated into an independent enterprise. In the early 1980s, Mimer introduced transaction handling for high robustness in applications, an idea that was subsequently implemented on most other DBMS.

2.5. 1980s Object Oriented Databases

The 1980s, along with a rise in object oriented programming, saw a growth in how data in various databases were handled. Programmers and designers began to treat the data in their databases as objects. That is to say that if a person's data were in a database, that person's attributes, such as their address, phone number, and age, were now considered to belong to that person instead of being extraneous data. This allows

for relationships between data to be relation to objects and their attributes and not to individual fields.

Another big game changer for databases in the 1980s was the focus on increasing reliability and access speeds. In 1989, two professors from the University of Michigan at Madison, published an article at an ACM associated conference outlining their methods on increasing database performance. The idea was to replicate specific important, and often queried information, and store it in a smaller temporary database that linked these key features back to the main database. This meant that a query could search the smaller database much quicker, rather than search the entire dataset. This eventually leads to the practice of indexing, which is used by almost every operating system from Windows to the system that operates Apple iPod devices.

2.6. DBMS building blocks

A DBMS includes four main parts: modeling language, data structure, database query language, and transaction mechanisms:

2.6.1. Components of DBMS

- DBMS Engine accepts logical request from the various other DBMS subsystems, converts them into physical equivalents, and actually accesses the database and data dictionary as they exist on a storage device.
- Data Definition Subsystem helps user to create and maintain the data dictionary and define the structure of the files in a database.
- Data Manipulation Subsystem helps user to add, change, and delete information in a database and query it for valuable information. Software tools within the data manipulation subsystem are most often the primary interface between user and the information contained in a database. It allows user to specify its logical information requirements.
- Application Generation Subsystem contains facilities to help users to develop transaction-intensive applications. It usually requires that user perform a detailed series of tasks to process a transaction. It facilitates easy-to-use data entry screens, programming languages, and interfaces.
- Data Administration Subsystem helps users to manage the overall database environment by providing facilities for backup and recovery, security management, query optimization, concurrency control, and change management.

2.6.2. Modeling language

A data modeling language to define the schema of each database hosted in the DBMS, according to the DBMS database model. The four most common types of models are the:

- Hierarchical model (Figure 2.5.),
- Network model (Figure 2.2.),
- Relational model (Figure 2.3.), and
- Object model.

Inverted lists and other methods are also used. A given database management system may provide one or more of the four models. The optimal structure depends on the natural organization of the application's data, and on the application's requirements (which include transaction rate (speed), reliability, maintainability, scalability, and cost).

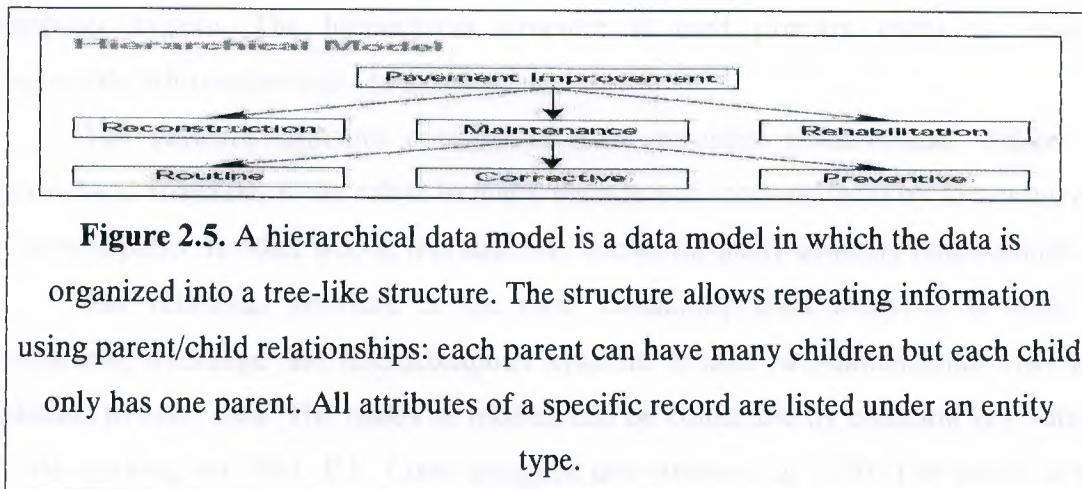


Figure 2.5. A hierarchical data model is a data model in which the data is organized into a tree-like structure. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent. All attributes of a specific record are listed under an entity type.

The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.

Before the database management approach, organizations relied on file processing systems to organize, store, and process data files. End users became aggravated with file processing because data is stored in many different files and each organized in a different way. Each file was specialized to be used with a specific application. Needless to say, file processing was bulky, costly and nonflexible when it came to supplying needed data accurately and promptly. Data redundancy is an issue with the file processing system because the independent data files produce duplicate data so when updates were needed each separate file would need to be updated. Another

issue is the lack of data integration. The data is dependent on other data to organize and store it. Lastly, there was not any consistency or standardization of the data in a file processing system which makes maintenance difficult. For all these reasons, the database management approach was produced. Database management systems (DBMS) are designed to use one of five database structures to provide simplistic access to information stored in databases. The five database structures are hierarchical, network, relational, multidimensional and object-oriented models.

The hierarchical structure was used in early mainframe DBMS. Records' relationships form a treelike model. This structure is simple but nonflexible because the relationship is confined to a one-to-many relationship. IBM's IMS system and the RDM Mobile are examples of a hierarchical database system with multiple hierarchies over the same data. RDM Mobile is a newly designed embedded database for a mobile computer system. The hierarchical structure is used primary today for storing geographic information and file systems.

The network structure consists of more complex relationships. Unlike the hierarchical structure, it can relate to many records and accesses them by following one of several paths. In other words, this structure allows for many-to-many relationships.

The relational structure is the most commonly used today. It is used by mainframe, midrange and microcomputer systems. It uses two-dimensional rows and columns to store data. The tables of records can be connected by common key values. While working for IBM, E.F. Codd designed this structure in 1970. The model is not easy for the end user to run queries with because it may require a complex combination of many tables.

The multidimensional structure is similar to the relational model. The dimensions of the cube looking model have data relating to elements in each cell. This structure gives a spreadsheet like view of data. This structure is easy to maintain because records are stored as fundamental attributes, the same way they're viewed and the structure is easy to understand. Its high performance has made it the most popular database structure when it comes to enabling online analytical processing (OLAP).

The object oriented structure has the ability to handle graphics, pictures, voice and text, types of data, without difficulty unlike the other database structures. This structure is popular for multimedia Web-based applications. It was designed to work with object-oriented programming languages such as Java.

2.7. Data structure

Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).

2.8. Database Query Language

A database query language and report writer allows users to interactively interrogate the database, analyze its data and update it according to the users privileges on data. It also controls the security of the database. Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called subschemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.

If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

2.9. Transaction mechanism

A database transaction mechanism ideally guarantees ACID (atomicity, consistency, isolation, durability) properties in order to ensure data integrity despite concurrent user accesses (concurrency control), and faults (fault tolerance). It also maintains the integrity of the data in the database. The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database.

2.9.1. ACID

In computer science, ACID (atomicity, consistency, isolation, durability) is a set of properties that guarantee database transactions are processed reliably. The concept of ACID is to evaluate databases and application architecture. In the context of databases, a single logical operation on the data is called a transaction. For example, a transfer of

funds from one bank account to another, even though that might involve multiple changes (such as debiting one account and crediting another), is a single transaction.

- **Atomicity**

Atomicity requires that database modifications must follow an all or nothing rule. Each transaction is said to be atomic if when one part of the transaction fails, the entire transaction fails and database state is left unchanged. It is critical that the database management system maintains the atomic nature of transactions in spite of any application, DBMS, operating system or hardware failure.

An atomic transaction cannot be subdivided, and must be processed in its entirety or not at all. Atomicity means that users do not have to worry about the effect of incomplete transactions.

Transactions can fail for several kinds of reasons:

- Hardware failure: A disk drive fails, preventing some of the transaction's database changes from taking effect
- System failure: The user loses their connection to the application before providing all necessary information
- Database failure: E.g., the database runs out of room to hold additional data
- Application failure: The application attempts to post data that violates a rule that the database itself enforces, such as attempting to create a new account without supplying an account number

- **Consistency**

The consistency property ensures that the database remains in a consistent state; more precisely, it says that any transaction will take the database from one consistent state to another consistent state.

The consistency property does not say how the DBMS should handle an inconsistency other than ensure the database is clean at the end of the transaction. If, for some reason, a transaction is executed that violates the database's consistency rules, the entire transaction could be rolled back to the pre-transactional state - or it would be equally valid for the DBMS to take some patch-up action to get the database in a consistent state. Thus, if the database schema says that a particular field is for holding integer numbers, the DBMS could decide to reject attempts to put fractional values there, or it could round the supplied values to the nearest whole number: both options maintain consistency.

The consistency rule applies only to integrity rules that are within its scope. Thus, if a DBMS allows fields of a record to act as references to another record, then consistency implies the DBMS must enforce referential integrity: by the time any transaction ends, each and every reference in the database must be valid. If a transaction consisted of an attempt to delete a record referenced by another, each of the following mechanisms would maintain consistency:

- abort the transaction, rolling back to the consistent, prior state;
- delete all records that reference the deleted record (this is known as cascade delete); or,
- nullify the relevant fields in all records that point to the deleted record.

These are examples of Propagation constraints; some database systems allow the database designer to specify which option to choose when setting up the schema for a database.

Application developers are responsible for ensuring application level consistency, over and above that offered by the DBMS. Thus, if a user withdraws funds from an account and the new balance is lower than the account's minimum balance threshold, as far as the DBMS is concerned, the database is in a consistent state even though this rule (unknown to the DBMS) has been violated.

- **Isolation**

Isolation refers to the requirement that other operations cannot access or see data that has been modified during a transaction that has not yet completed. Each transaction must remain unaware of other concurrently executing transactions, except that one transaction may be forced to wait for the completion of another transaction that has modified data that the waiting transaction requires.

- **Durability**

Durability is the DBMS's guarantee that once the user has been notified of a transaction's success, the transaction will not be lost. The transaction's data changes will survive system failure, and that all integrity constraints have been satisfied, so the DBMS won't need to reverse the transaction. Many DBMSs implement durability by writing transactions into a transaction log that can be reprocessed to recreate the system state right before any later failure. A transaction is deemed committed only after it is entered in the log.

Durability does not imply a permanent state of the database. A subsequent transaction may modify data changed by a prior transaction without violating the durability principle.

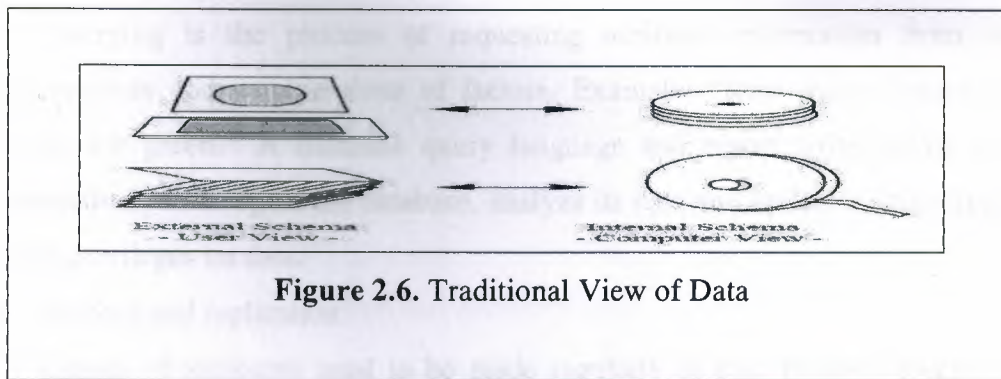
2.10. DBMS topics

2.10.1. External, Logical and Internal view

A database management system provides the ability for many different users to share data and process resources. But as there can be many different users, there are many different database needs. The question now is: How can a single, unified database meet the differing requirement of so many users?

A DBMS minimizes these problems by providing two views (Figure 2.6.) of the database data: an external view (or User view), logical view (or conceptual view) and physical (or internal) view. The user's view, of a database program represents data in a format that is meaningful to a user and to the software programs that process those data. That is, the logical view tells the user, in user terms, what is in the database. The physical view deals with the actual, physical arrangement and location of data in the direct access storage devices (DASDs). Database specialists use the physical view to make efficient use of storage and processing resources. With the logical view users can see data differently from how they are stored, and they do not want to know all the technical details of physical storage. After all, a business user is primarily interested in using the information, not in how it is stored.

One strength of a DBMS is that while there is typically only one conceptual (or logical) and physical (or Internal) view of the data, there can be an endless number of different External views. This feature allows users to see database information in a more business-related way rather than from a technical, processing viewpoint. Thus the logical view refers to the way user views data, and the physical view to the way the data are physically stored and processed...



2.10.2. DBMS features and capabilities

Alternatively, and especially in connection with the relational model of database management, the relation between attributes drawn from a specified set of domains can be seen as being primary. For instance, the database might indicate that a car that was originally "red" might fade to "pink" in time, provided it was of some particular "make" with an inferior paint job. Such higher arity relationships provide information on all of the underlying domains at the same time, with none of them being privileged above the others.

2.10.3. DBMS simple definition

Data base management system is the system in which related data is stored in an "efficient" and "compact" manner. Efficient means that the data which is stored in the DBMS is accessed in very quick time and compact means that the data which is stored in DBMS covers very less space in computer's memory. In above definition the phrase "related data" is used which means that the data which is stored in DBMS is about some particular topic.

Throughout recent history specialized databases have existed for scientific, geospatial, imaging, document storage and like uses. Functionality drawn from such applications has lately begun appearing in mainstream DBMSs as well. However, the main focus there, at least when aimed at the commercial data processing market, is still on descriptive attributes on repetitive record structures.

Thus, the DBMSs of today roll together frequently needed services or features of attribute management. By externalizing such functionality to the DBMS, applications effectively share code with each other and are relieved of much internal complexity. Features commonly offered by database management systems include:

- Query ability

Querying is the process of requesting attribute information from various perspectives and combinations of factors. Example: "How many 2-door cars in Texas are green?" A database query language and report writer allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.

- Backup and replication

Copies of attributes need to be made regularly in case primary disks or other equipment fails. A periodic copy of attributes may also be created for a distant organization that cannot readily access the original. DBMS usually provide utilities to facilitate the process of extracting and disseminating attribute sets. When data is replicated between database servers, so that the information remains consistent throughout the database system and users cannot tell or even know which server in the DBMS they are using, the system is said to exhibit replication transparency.

- Rule enforcement

Often one wants to apply rules to attributes so that the attributes are clean and reliable. For example, we may have a rule that says each car can have only one engine associated with it (identified by Engine Number). If somebody tries to associate a second engine with a given car, we want the DBMS to deny such a request and display an error message. However, with changes in the model specification such as, in this example, hybrid gas-electric cars, rules may need to change. Ideally such rules should be able to be added and removed as needed without significant data layout redesign.

- Security

Often it is desirable to limit who can see or change which attributes or groups of attributes. This may be managed directly by individual, or by the assignment of individuals and privileges to groups, or (in the most elaborate models) through the assignment of individuals and groups to roles which are then granted entitlements.

- Computation

There are common computations requested on attributes such as counting, summing, averaging, sorting, grouping, cross-referencing, etc. Rather than have each computer application implement these from scratch, they can rely on the DBMS to supply such calculations.

- Change and access logging

Often one wants to know who accessed what attributes, what was changed, and when it was changed. Logging services allow this by keeping a record of access occurrences and changes.

- Automated optimization

If there are frequently occurring usage patterns or requests, some DBMS can adjust themselves to improve the speed of those interactions. In some cases the DBMS will merely provide tools to monitor performance, allowing a human expert to make the necessary adjustments after reviewing the statistics collected.

2.10.4. Meta-data repository

Metadata is data describing data. For example, a listing that describes what attributes are allowed to be in data sets is called "meta-information". The meta-data is also known as data about data.

2.10.5. Current trends

In 1998, database management was in need of new style databases to solve current database management problems. Researchers realized that the old trends of database management were becoming too complex and there was a need for automated configuration and management. Surajit Chaudhuri, Gerhard Weikum and Michael Stonebraker, were the pioneers that dramatically affected the thought of database management systems. They believed that database management needed a more modular approach and that there are so many specifications needs for various users. Since this new development process of database management we currently have endless possibilities. Database management is no longer limited to "monolithic entities". Many solutions have developed to satisfy individual needs of users. Development of numerous database options has created flexible solutions in database management.

Today there are several ways database management has affected the technology world as we know it. Organizations demand for directory services has become an extreme necessity as organizations grow. Businesses are now able to use directory services that provided prompt searches for their company information. Mobile devices are not only able to store contact information of users but have grown to bigger capabilities. Mobile technology is able to cache large information that is used for computers and is able to display it on smaller devices. Web searches have even been affected with database management. Search engine queries are able to locate data within the World Wide Web. Retailers have also benefited from the developments with data warehousing. These companies are able to record customer transactions made within

their business. Online transactions have become tremendously popular with the e-business world. Consumers and businesses are able to make payments securely on company websites. None of these current developments would have been possible without the evolution of database management. Even with all the progress and current trends of database management, there will always be a need for new development as specifications and needs grow.

As the speeds of consumer internet connectivity increase, and as data availability and computing become more ubiquitous, database are seeing migration to web services. Web based languages such as XML and PHP are being used to process databases over web based services. These languages allow databases to live in "the cloud." As with many other products, such as Google's GMail, Microsoft's Office 2010, and Carbonite's online backup services, many services are beginning to move to web based services due to increasing internet reliability, data storage efficiency, and the lack of a need for dedicated IT staff to manage the hardware. Faculty at Rochester Institute of Technology published a paper regarding the use of databases in the cloud and state that their school plans to add cloud based database computing to their curriculum to "keep [their] information technology (IT) curriculum at the forefront of technology.

2.11. Summary

The database that is proffered with your library application is important. It is important that it works well (reliably, efficiently and flexibly), can respond to the upcoming changes in the computer and information handling world and is commercially viable. Adherence to industry standards like SQL, ODBC etc will ensure that you can use third party tools like Access against the database to get data to the desktop.

So when looking at a library system consider the advantages and disadvantages of the various types of DBMS that underlie the system and assess whether your needs are likely to be limited by the technologies offered.

If you need lower hardware and administration costs then the greater efficiency of Nested Relational DBMS or OpenM could be useful. If you must retain corporate standards and these are Relational DBMS like Oracle, then you can take advantage of the many tools and skills that may be available from the corporate IT department.

CHAPTER THREE: PHP, MySQL AND APACHE

3.1. Introduction

PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

PHP is a server-side, HTML-embedded scripting language that may be used to create dynamic Web pages. It is available for most operating systems and Web servers, and can access most common databases, including MySQL. PHP may be run as a separate program or compiled as a module for use with the Apache Web server.

MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

The Apache HTTP Server, commonly referred to as Apache, is web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache was the first viable alternative to the Netscape Communications Corporation web server (currently known as Sun Java System Web Server), and has since evolved to rival other Unix-based web servers in terms of functionality and performance. The majority of web servers using Apache run a Unix-like operating system

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP.

3.2. PHP Programming Language

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. What does that mean? An example (Figure 3.1.):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hi, I'm a PHP script!";
      print "Hello Web!";
    ?>
  </body>
</html>
```

Figure 3.1. Basic example of PHP language.

Instead of lots of commands to output HTML, PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions `<?php` and `?>` that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your

HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

PHP generally runs on a web server, taking PHP code as its input and creating Web pages as output, however it is also popular for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers and on almost every OS platform free of charge. The PHP Group also provides the complete source code for users to build, customize and extend for their own use.

3.2.1. PHP scripting

PHP is now officially known as "PHP: HyperText Preprocessor". It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities are endless. The output from PHP code is combined with the HTML in the script and the result sent to the user's web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because all the browser sees is HTML.

PHP's support for Apache and MySQL further increases its popularity. Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it. The combination of Apache, MySQL and PHP is all but unbeatable. That doesn't mean that PHP cannot work in other environments or with other tools. In fact, PHP supports an extensive list of databases and web-servers. The rise in popularity of PHP has coincided with a change of approach in web-publishing. While in the mid-1990s it was ok to build sites, even relatively large sites, with hundreds of individual hard-coded HTML pages, today's webmasters are making the most of the power of databases to manage their content more effectively and to personalize their sites according to individual user preferences.

3.2.2. Reasons for using PHP

There are some indisputable great reasons to work with PHP. As an open source product, PHP is well supported by a talented production team and a committed user community. Furthermore, PHP can be run on all the major operating systems with most servers.

The speed of development is also important. Because PHP allows you to separate HTML code from scripted elements, you will notice a significant decrease in development time on many projects. In many instances, you will be able to separate the coding stage of a project from the design and build stages. Not only can this make life easier for you as a programmer, but it also can remove obstacles that stand in the way of effective and flexible design.

Well-maintained open source projects offer users additional benefits. You benefit from an accessible and committed community who offer a wealth of experience in the subject, as fast and as cheap as possible. Chances are that any problem you encounter in your coding can be answered swiftly and easily with a little research. If that fails, a question sent to a mailing list or forum can have an intelligent, authoritative response. You also can be sure that bugs will be addressed as they are found, and that new features will be made available as the need is defined. You will not have to wait for the next commercial release before taking advantage of improvements, and there is no hidden interest in a particular server product or operating system. You are free to make choices that suit your needs or those of your clients and incorporate whatever components you want.

3.2.3. Structure Of PHP Script

Let's get back to the script we used before:

```
<?php  
print "Hello Web!";  
?>
```

In this simple script you can see some of the most used components of a PHP script. First off, PHP tags are used to separate the actual PHP content from the rest of the file. You can inform the interpreter that you want it to execute your commands by adding a pair of these: standard tags "<?php ?>"; short tags "<? ?>"; ASP tags "<% %>"; script tags "<SCRIPT LANGUAGE='php'> </SCRIPT>". The standard and the script tags are guaranteed to work under any configuration, the other two need to be enabled in your "php.ini"

Now that you know how to define a block of PHP code, take a closer look at the code above. The "print" function is used to output data, so anything output by "print()" ends up in the HTML file. Therefore, you can say that a function is a command that performs an action. Usually, you send some data to the function, and the function uses that data to come up with a result. There are a lot functions in PHP, and almost each one performs a different action. Data sent to a function is almost always placed in parentheses after the function name; there are some exceptions where parentheses are optional, and the "print()" function is one of them.

After that first line of code, you can see a semicolon. This semicolon informs the interpreter you have completed a statement – a statement is to PHP what a sentence is to the English language. It represents an instruction to the interpreter, and some additional data. If PHP doesn't find a semicolon at the end of your statement, then it will continue parsing the file until it finds one, ignoring any white-spaces or empty lines. So, your one statement doesn't necessarily have to use only one line of code. There can be two or more statements on a single line, but, on the other hand, a statement could use two or more lines. PHP also ignores white spaces, so you can have as many blanks as you want between the statements, and between the statements' parameters. You should know that you don't have to use a semicolon with the last statement in your script (just before the closing tag). So the following scripts are equivalent:

```
<?php
```

```
print "This is a test" ;
```

```
?>
```

```
<?php print "This is a test" ?>
```

Commenting your PHP code can be very helpful. If some code seems to be very clear at the time of writing, the same code can look like a black hole a few weeks later, when you want to modify it. So adding comments to your code can save you time later on, and make it easier for other people to work with your code. But, wait, what is a comment? A comment is a text in a script that is ignored by the interpreter. So you can write anything you want in it, from copyright notices to detailed information about your code. PHP recognizes a comment by checking out if it contains two forward slashes ("//") or a single hash sign ("#"). The text beginning from either of these marks until the end of the line is ignored. You can also use multi-line comments. They begin with a forward slash followed by an asterisk ("/*") and end with an asterisk followed by a forward slash ("*/"):

```

<?php
/*
This is a demo script.
All it does is output Hello Web! in your browser.
*/
print "Hello Web!"; //outputs a message.
?>

```

3.2.4. PHP Database Manipulation

One of the defining features of PHP is the ease with which you can connect to and manipulate databases. PHP implements functions for connecting to a wide range of databases systems: MySQL, Oracle, MSSQL, Interbase, dBase, and many more. While there are many commercial database systems which cost thousands of dollars and provide thousands of components, the Internet community provides solutions for low-budget users as well. MySQL, for example, doesn't require you to purchase a license if you don't use it for commercial activities.

Most of the manipulating of any database is done while using SQL, which stands for Structured Query Language. This provides standard syntaxes by which different types of database can be queried. While there are a lot of extensions to SQL implemented by database systems, and some queries could work on a system but fail on another one, all of them incorporate some standard manipulating statements.

The way PHP talks to the database is simple. First you need to connect to the database system, which runs as a daemon (service for Windows) in the background. In order to do this, you must have appropriate permissions to connect to the database server from its owner. While you can always install and configure a database system on your own computer for starters, when it comes to hosting a web-site you must talk to a hosting company. They will open an account for you, and give you the information on how to connect to the database server.

Many hosting companies use MySQL, so we'll focus on using this database system. First, you must use the "mysql_connect()" function along with the server's IP, the username and the password required to connect to the server. After you connect, you can start running SQL statements as you like. Don't forget that to close the connection using "mysql_close()". You either must do this everytime you want to manipulate the database, or you use a persistent connection. This kind of connection is implemented when using the Apache web-server, and tells Apache not to terminate the connection

after your script stops executing even if you call "mysql_close()". Instead, the connection is left active, waiting for another process to call mysql_pconnect(). So, using this function, you will skip the time required for your computer to connect to the database server.

```
if(!mysql_connect("localhost", "bob", "secret")) //tries to connect to the database
server
die("Unable to connect to the database server!"); //terminates the script, and outputs
the error
```

Depending on the hosting services, you can have access to one or more databases. But before you start changing data, you must also connect to the database of your choice. Selecting a database to work with is also accomplished using a simple function:

```
mysql_select_db("test_db") or die("Error when selecting the database!");
```

While we can treat the errors using the "die()" function, and stop the script if something bad happens, sometimes we might want to output more detailed information on the error. Whenever an operation fails, MySQL sets an error number and an error string, so we just have to read it:

```
if(!mysql_connect($db_ip, $db_user, $db_pass))
die("MySQL Error: " . mysql_error()); //stops the script if unable to connect to the
database, and outputs the MySQL error message
```

All of the database manipulation is done thru SQL statements (also known as queries), which range from basic statements used to insert, modify or delete data in a table to complex statements used to retrieve some particular data using specified filters. You can find a lot of documentation on SQL on the Internet. Here are some of the most important statements:

```
//creates a table using the specified fields
```

```
CREATE TABLE users
```

```
(id INT NOT NULL AUTO_INCREMENT,
```

```
PRIMARY KEY(id),
```

```
username VARCHAR(20),
```

```
password VARCHAR(12),
```

```
email VARCHAR(20)
```

```
)
```

```
//adds a new record into the database
```

```
INSERT INTO my_table(id, username, password, email) VALUES (NULL, "bob",  
"secret", "bob@softwareprojects.org")
```

```
//modifies the password of the user "bob"
```

```
UPDATE my_table SET password = "very secret" WHERE username = "bobo"
```

```
//retrieves the email address of the user "bob"
```

```
SELECT email FROM users WHERE username = "bob"
```

To execute these queries, you must call the "mysql_query()" function. The result of this function is a resource that will be used with other functions, in order to access the information returned by the query.

```
$sql_result = mysql_query("SELECT password FROM users WHERE  
username='Bob'")
```

```
or die("Error or running query: " . $mysql_error())
```

```
$row = mysql_fetch_array($sql_result, MYSQL_NUM);
```

While it may seem that "mysql_fetch_array()" retrieves all the results from the query, you should know that it only returns one row at a time. After a call to this function, the internal index of \$sql_result will be incremented by one, so the next time the next row will be retrieved, if there is any.

3.2.5. The Benefits of PHP Includes

- If you have only a small website of seven or eight pages, it will be easy enough to just put your navigation list on every page manually. However, if you need to change something, you'll have to go back to every page and make your changes. This can be very time consuming even with a small site, but with a larger one of 20 or more pages, it can take hours – especially if you want to make big changes to the design of the navigation list, for example.
- If you use GUI web design software such as FrontPage or Dreamweaver, you will probably have a page template which just duplicates on every page. This is a good solution insofar as you can make changes to your whole site without having to repeat them on every page. However, the problem is that if you don't have your GUI web design package available on the computer on which you need to make your changes, you're stuck – you will have to change every page.
- At this point, you might cave in and decide to build a framed website, with the navigational links in a side frame and your content in the large centre frame. If you know anything about web design, you'll know that framed websites are, almost without exception, a bad idea – the most obvious reasons being

inconsistent browser rendering of framed sites, and the fact that internal bookmarking of a site becomes more complicated to implement.

- If your server is PHP-enabled, a far better solution than all of these is a PHP include. Basically, a PHP include inserted into the html of a web page tells the browser to retrieve an html file from the server and include it when it renders the web page. This means that you only need to have one file with your navigation links in it stored on your server, and every one of your web pages can be rendered including this same file. The benefits are clear: if you need to make a change, you need only change your single file and the change will show immediately wherever you have put a PHP include for that file.

3.2.6. PHP on Linux Versus PHP on Windows

Some things are faster on Windows, other are faster on Linux.

- RAW statement execution seems faster on Windows.
- Function Calls were faster on Windows
- Object Creation / Access was faster on Linux with PHP 5.2 but faster on Windows with 5.3
- Library calls were faster on Linux. (Example: Encryption 3-5 times faster on Ubuntu.)
- File Access is faster on Linux by a small percentage, except for file copy operations which was as much as 60% slower on Windows probably due to the ACL advanced security.
- MySQL access with Linux is faster by more than a little and on Windows, MySQL access deteriorates in version 5.3 (This seems to be poor implementation, see PostgreSQL below.)
- PostgreSQL performance is very close on both platforms (within 6/100 of a second for 1000 Operations) – It's faster on Windows and faster still on Windows with PHP 5.3
- MS SQL Server access from PHP 5.2 on Windows is marginally slower than MySQL access on Linux. (PHP 5.3 not yet supported at the time of this writing.)

So what does all that mean ?

- We can probably say that in terms of raw execution – performance on Linux versus Windows is probably a wash (more or less equivalent) so that the performance of PHP itself becomes a moot factor in choosing Linux or Windows for PHP application deployment.

- If you are building an application, or running an application that supports it, PostgreSQL might be a better database choice since it performs pretty much the same on Windows and Linux.
- If you are running an application that locks you in to Sun Microsystems' MySQL and you want to run it on Windows, you should do scale planning. (My personal guess is that it's unlikely that Sun will markedly improve MySQL performance on Windows.)
- Version 1 of the PHP Driver for SQL Server (V2 is in the works) is somewhat slower than MySQL or PostgreSQL but probably not enough to discourage use where diverse developer access is desired. (v2 of the driver will improve performance.)

3.2.7. PHP vs Java vs Ruby

With Sun GPLv2'ing its Java software this week, a lot of focus is on the other open frameworks that have risen during the same era, and their comparative value in a post open source Java world. One such take is from well-known Sun insider, Tim Bray, who spoke at last week's International PHP Conference in Germany.

Bray, who is Director of Web Technologies at Sun Microsystems, co-creator of the XML 1.0 and XML Namespace standards, co-founder of Open-Text, and contributor to the Atom Publishing standard spoke in a presentation entitled "How to combine PHP technology with Java based on Enterprise Systems".

In the slide below, Bray outlines the basic advantages of each of these three platforms (or rather the two languages and one framework if you are scoring at home). In perhaps in a nod to his new employer, Sun, Bray totally omits any mention of .NET or smaller products like Adobe's Flex/ColdFusion platform.

His idea, if you are able to take it on the surface only, is to rate your priorities of (Figure 3.2.)

1) Scalability, 2) Dev Speed, 3) Dev Tools, and 4) Maintainability. In this highly generalized approach, you could theoretically pick your web application building language/framework based on only these priorities.

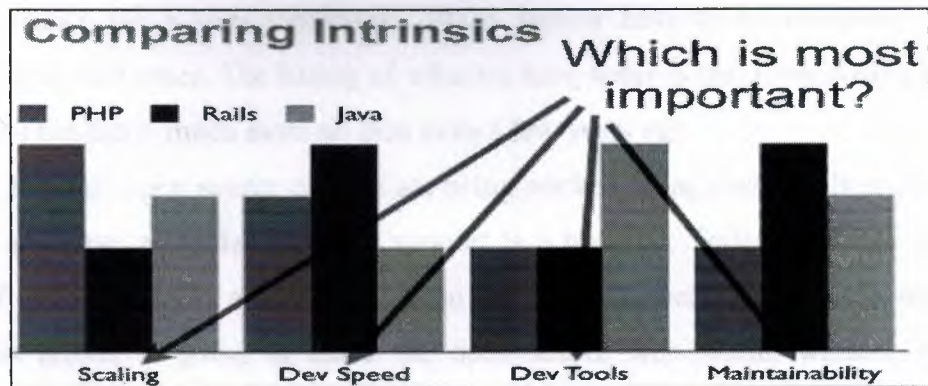


Figure 3.2. Comparing Intrinsic

Tim weighs-in heavy on the maintainability side, and we'd tend to agree with him. To quote: "Good apps, once built, tend to be in production for an astonishingly long time. Which means that they have to be maintained for an astonishingly long time. Which means that maintainability is important. There are a lot of things that go into maintainability, but I suggest that the biggies are object-orientation, MVC architecture, code readability, and code size."

On the maintainability side Tim places Ruby on Rails out front. Quoting him again, Tim describes why RoR is in the lead here.

"Mostly because there's less code. The fact that Ruby forces MVC on you helps too, also the fact that the templating and [Object Relational Mapping (ORM)] and testing and application code are all so tightly integrated helps. Bear in mind that we still don't know how well Rails is going to work outside of the CRUD-centered greenfield app space where it shines."

Tim's metrics don't really exist on a quantitative level nor are they based on any published statistics — just a very informed intellect and a privileged position. Obviously, additional parameters such as total cost of ownership and native security (PHP, we are looking at you) should also play a big role the decision making process. As you can probably guess, in response to this presentation, the proponents of Java, PHP and Ruby are lighting the Internet aflame in defense of their platform of choice and at the same time nay-saying their fellow players. While very entertaining and sometimes educational to follow, not much is going to be gained from this bickering. Bray concludes the presentation with the Rodney King-esque statement "One way or another we all have to integrate."

The reality is that there are a great number of ways to hone-in on the language or framework choice for a web application. Many factors have to be considered and weighed against each other. The beauty of what we have today is that there ARE a great many choices out there, much more-so than even a few years ago.

The fact that open source options are being stacked up against the likes of Java and .NET, and have a similar levels of support is a blessing. Today, with the open-sourcing of Java, we have a new entry onto the field. Rather than exclude one or another, this arrival is going to make the open source web world, warmer, more sophisticated, and for better or worse, a little sunnier.

If you want to download PHP language in pardus, write this code in console; "sudo pisi it mod_php".

3.3. What is MySQL

MySQL is a relational database system that is used to store information. MySQL can store many types of data from something as tiny as a single character to as large as complete files or graphics. Although it can be accessed by most programming languages, it is often coupled with PHP because they work together with ease.

Information stored in a MySQL database hosted on a web server can be accessed from anywhere in the world with a computer. This makes it a good way to store information that needs the ability to change over time, but also needs to be accessed over the net. Some examples that can utilize MySQL are a web message board or a customer's shipping status.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Sun Microsystems, a subsidiary of Oracle Corporation.

Members of the MySQL community have created several forks such as Drizzle and MariaDB. Both forks were in progress before the Oracle acquisition (Drizzle was announced 8 months before the Sun acquisition).

Free-software projects that require a full-featured database management system often use MySQL. Such projects include (for example) WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products including Wikipedia, Google and

Facebook. MySQL runs on virtually all platforms, including Linux, Unix, and Windows. It is fully multi-threaded using kernel threads, and provides application program interfaces (APIs) for many programming languages, including C, C++, Eiffel, Java, Perl, PHP, Python, and Tcl.

MySQL is used in a wide range of applications, including data warehousing, e-commerce, Web databases, logging applications and distributed applications. It is also increasingly embedded in third-party software and other technologies. According to MySQL AB, their flagship product has over six million active MySQL installations worldwide.

3.3.1 How do PHP and MySQL work together?

PHP and MySQL compliment each other to do with neither can do alone. PHP can collect data, and MySQL can in turn store the information. PHP can create dynamic calculations, and MySQL can provide it with the variables it uses. PHP can create a shopping cart for your web store, but MySQL can then keep the data in a format PHP can use to create receipts on demand, show current order status, or even suggest other related products.

Although PHP and MySQL can each be used independently, when you put them together it opens up countless possibilities for your site. As the internet progresses, it becomes more and more necessary to deliver dynamic content to keep up with the demands of web surfers and their desire to have information instantly delivered to them online. By learning to use PHP and MySQL you can deliver this information to them on demand.

3.3.2. Running PHP and MySQL

In order to do some of our tutorials (or program with PHP and MySQL in general) you need to be able to run PHP and MySQL. Although most free hosts do not have PHP and MySQL support there are some that do, and there are also many low cost hosting options available. If you already have hosting and are unsure if you have PHP and MySQL support you should contact your host directly. If you are shopping for hosting be sure that they support PHP and that your package includes at least one MySQL database. Another option is to install PHP and MySQL directly onto your Windows computer. If you are a Mac user you already have the capability and just need to activate PHP and MySQL.

Interacting with a MySQL database is a little weird as you don't have the tried and true WYSIWYG interface that something as easy as Microsoft Access affords.

When creating tables, you'll either have to create them by using SQL Statements, or by using another open-source tool available online called PHPMysqlAdmin. PHPMysqlAdmin gives you an easy-to-use interface that allows you to create tables and run queries by filling in a little bit of information and then having the tables created for you. This is good if you're either lazy, or don't feel like bothering with big and complicated SQL Statements.

MySQL tables also have the luxury of using "real" data types – not those little fluffy ones you see in Microsoft Access. Need a text field that can hold over four billion characters? Not a problem, just use the LongText data type. Want the field to hold that many characters and be case-sensitive? Easy, just use the LongBlob data type. Need to store numbers from 0 to 18,446,744,073,709,551,615 (for those of you who are curious, that would be over 18 Quintillion), then use the BigInt data type. Indeed, 18 quintillion is a big integer.

As you can see, the limitations of Access fall well within the capabilities of MySQL. This is one of the many reasons that huge webpages use MySQL to handle their millions of daily hits. Well, that, and it's free!

If you want to download MySQL server in pardus, write this code in console;
"sudo pisi it mysql-server".

Then MySQL server needed to run the following command to create the databases.

"mysql_install_db".

Finally, If you want to run MySQL as a service [with the opening of the operating system to automatically start work] must run the following command.

"service mysql_server on".

3.4. Phpmyadmin

PhpMyAdmin is a free software tool written in PHP intended to handle the administration of MySQL over the World Wide Web. PhpMyAdmin supports a wide range of operations with MySQL. The most frequently used operations are supported by the user interface (managing databases, tables, fields, relations, indexes, users, permissions, etc), while you still have the ability to directly execute any SQL statement.

PhpMyAdmin can manage a whole MySQL server (needs a super-user) as well as a single database. To accomplish the latter you'll need a properly set up MySQL user who can read/write only the desired database.

Many people have difficulty understanding the concept of user management with regards to phpMyAdmin. When a user logs in to phpMyAdmin, that username and password are passed directly to MySQL. phpMyAdmin does no account management on its own (other than allowing one to manipulate the MySQL user account information); all users must be valid MySQL users.

If you use phpMyAdmin in Pardus, this package in the store as well as additional contributions (contrib) can be found. If you use PHP from the console on php-cli package available in the store.

3.4.1. Features

- Intuitive web interface
- Support for most MySQL features:
 - browse and drop databases, tables, views, fields and indexes
 - create, copy, drop, rename and alter databases, tables, fields and indexes
 - maintenance server, databases and tables, with proposals on server configuration
 - execute, edit and bookmark any SQL-statement, even batch-queries
 - manage MySQL users and privileges
 - manage stored procedures and triggers
- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, L^AT_EX and others
- Administering multiple servers
- Creating PDF graphics of your database layout
- Creating complex queries using Query-by-example (QBE)
- Searching globally in a database or a subset of it
- Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
- And much more...

3.4.2. Ability of Phpmyadmin

PhpMyAdmin can;

- browse and drop databases, tables, views, fields and indexes
- create, copy, drop, rename and alter databases, tables, fields and indexes
- maintenance server, databases and tables, with proposals on server configuration

- execute, edit and bookmark any SQL-statement, even batch-queries
- load text files into tables
- create and read dumps of tables
- export data to various formats: CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel and LATEX formats
- import data and MySQL structures from Microsoft Excel and OpenDocument spreadsheets, as well as XML, CSV, and SQL files
- administer multiple servers
- manage MySQL users and privileges
- check referential integrity in MyISAM tables
- using Query-by-example (QBE), create complex queries automatically connecting required tables
- create PDF graphics of your Database layout
- search globally in a database or a subset of it
- transform stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
- track changes on databases, tables and views
- support InnoDB tables and foreign keys
- support mysqli, the improved MySQL extension
- communicate in 58 different languages
- synchronize two databases residing on the same as well as remote servers

PhpMyAdmin is Open Source, so you're invited to contribute to it. Many great features have been written by other people and you too can help to make phpMyAdmin a useful tool.

3.5. Apache HTTP Server

Often referred to as simply Apache, a public-domain open source Web server developed by a loosely-knit group of programmers. The first version of Apache, based on the NCSA httpd Web server, was developed in 1995.

Core development of the Apache Web server is performed by a group of about 20 volunteer programmers, called the Apache Group. However, because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. In many respects, development of Apache is similar to development of the Linux operating system.

The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms.

The name is a tribute to the Native American Apache Indian tribe, a tribe well known for its endurance and skill in warfare. A common misunderstanding is that it was called Apache because it was developed from existing NCSA code plus various patches, hence the name a patchy server, or Apache server.

Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The application is available for a wide variety of operating systems, including Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF, and eComStation. Released under the Apache License, Apache is characterized as open source software. Since April 1996 Apache has been the most popular HTTP server software in use. As of February 2010 Apache served over 54.46% of all websites and over 66% of the million busiest.

3.5.1. Features

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Popular authentication modules include `mod_access`, `mod_auth`, `mod_digest`, and `mod_auth_digest`, the successor to `mod_digest`. A sample of other features include SSL and TLS support (`mod_ssl`), a proxy module (`mod_proxy`), a URL rewriter (also known as a rewrite engine, implemented under `mod_rewrite`), custom log files (`mod_log_config`), and filtering support (`mod_include` and `mod_ext_filter`).

Popular compression methods on Apache include the external extension module, `mod_gzip`, implemented to help with reduction of the size (weight) of web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for web applications. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different actual websites. For example, one machine with one Apache installation could simultaneously serve `www.example.com`, `www.test.com`, `test47.test-server.test.com`, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs).

3.5.2. Use

Apache is primarily used to serve both static content and dynamic Web pages on the World Wide Web. Many web applications are designed expecting the environment and features that Apache provides.

Apache is redistributed as part of various proprietary software packages including the Oracle Database and the IBM WebSphere application server. Mac OS X integrates Apache as its built-in web server and as support for its WebObjects application server. It is also supported in some way by Borland in the Kylix and Delphi development tools. Apache is included with Novell NetWare 6.5, where it is the default web server. Apache is included with many Linux distributions.

Apache is used for many other tasks where content needs to be made available in a secure and reliable way. One example is sharing files from a personal computer over the Internet. A user who has Apache installed on their desktop can put arbitrary files in Apache's document root which can then be shared.

Programmers developing web applications often use a locally installed version of Apache in order to preview and test code as it is being developed.

3.5.3. Performance

Although the main design goal of Apache is not to be the "fastest" web server, Apache does have performance comparable to other "high-performance" web servers. Instead of implementing a single architecture, Apache provides a variety of MultiProcessing Modules (MPMs) which allow Apache to run in a process-based, hybrid (process and thread) or event-hybrid mode, to better match the demands of each particular infrastructure. This implies that the choice of correct MPM and the correct configuration is important. Where compromises in performance need to be made, the design of Apache is to reduce latency and increase throughput, relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

At the end; If you want to download Apache server in pardus, write this code in console;

“sudo pisi install apache”.

Then, If you want to run apache as a service [with the opening of the operating system to automatically start work] must run the following command.

“service apache on”.

CHAPTER FOUR: USER INTERFACE AND DATABASE

STRUCTURE OF BOOKSTORE PROGRAM

4.1. Introduction

Bookstore program briefly to explain, library easily can keep the book records, especially keep records of sales revenues as a report and can view of that books.

This program was designed as platform independent and performed in vim editor at pardus which is one of unix operation system. We did not use any ready template so every line has a necessary function and code, but disadvantage of this situation takes a long time to write the codes, organize to tables and visual design. Database structure was created in phpmyadmin, phpmyadmin is a free software tool written in PHP intended to handle the administration of MySQL over the World Wide Web.

There are two interface; User interface (Figure 4.1.) and admin (Figure 4.2.) interface.

For user interface, User can;

- Registration new book.
- Search by
 - ISBN
 - Book Title
 - Autor of the book
 - Publisher
 - Categories
 - Course Code and so on..
- Do limited option.
- See limited pages (Figure 4.1.).
- See just own sales report.

For Admin interface (Figure 4.2.), Admin can;

- Registration new book.
- Search by
 - ISBN
 - Book Title
 - Autor of the book
 - Publisher
 - Categories

- Course Code and so on..
- Delete, edit operation
- View every user sales report.
- See every pages.
- Do barcode printing.

If necessary more page or more operation for program ofcourse it is improvable program.

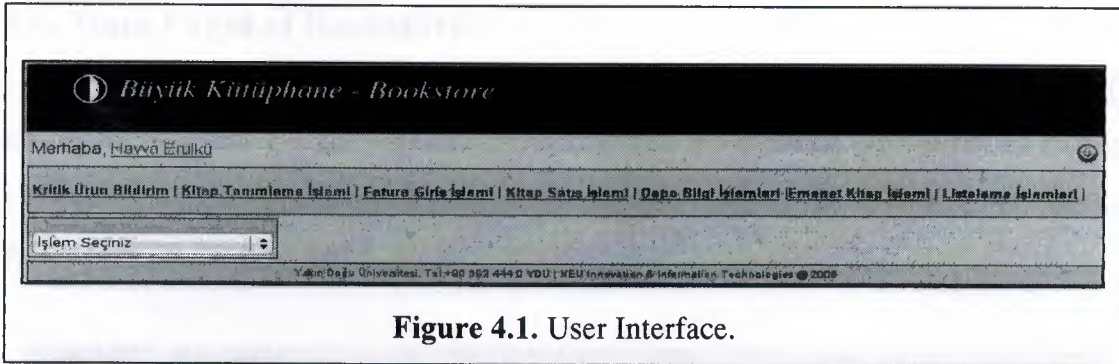


Figure 4.1. User Interface.

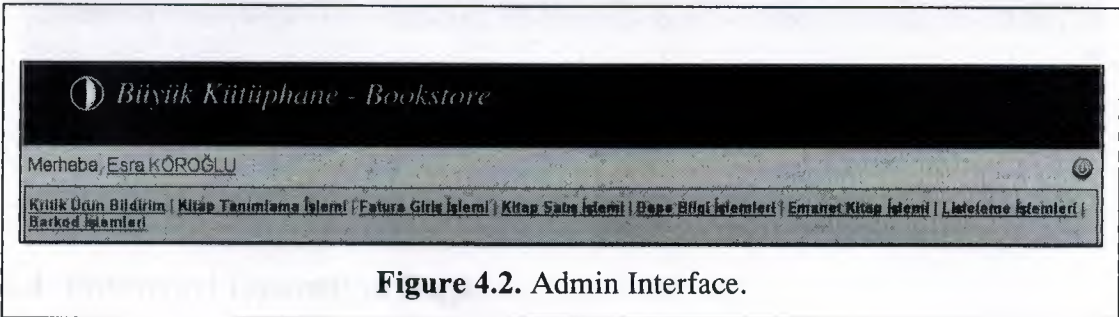


Figure 4.2. Admin Interface.

4.2. Bookstore Login Pages

This pages (Figure 4.3.) also first pages of bookstore program. If person who want to see and use the program, firstly should write own username and password (Figure 4.4.).

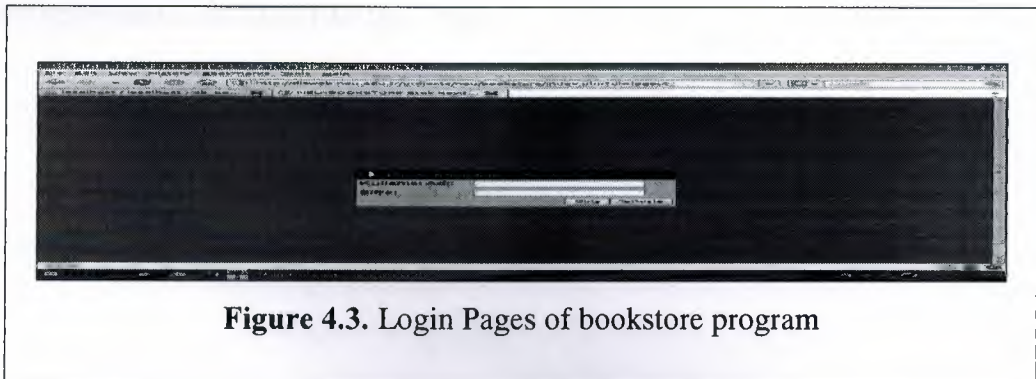


Figure 4.3. Login Pages of bookstore program

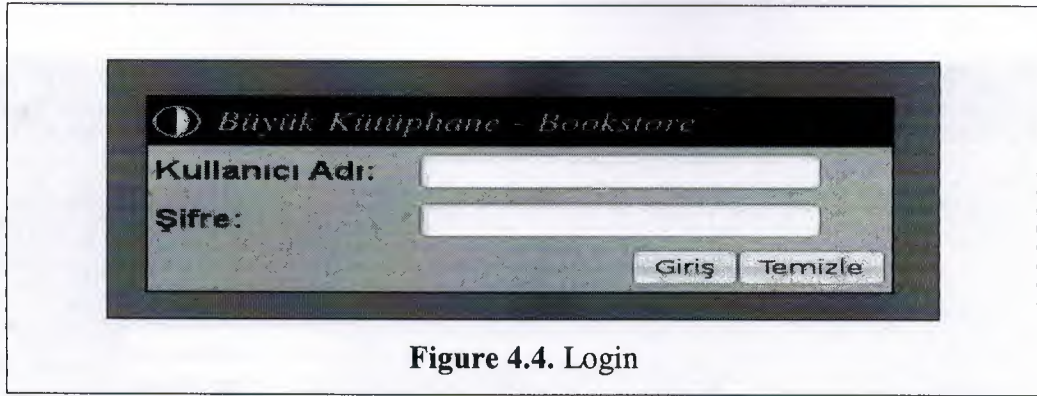


Figure 4.4. Login

4.3. Main Pages of Bookstore

This page shows main menu of bookstore. Main menu (Figure 4.5.) has a eight sub-menu and one hidden sub-menu. Hidden sub-menu about password operations. User can change own password with this page. In this menu the user, can be accessed by clicking on the own name.

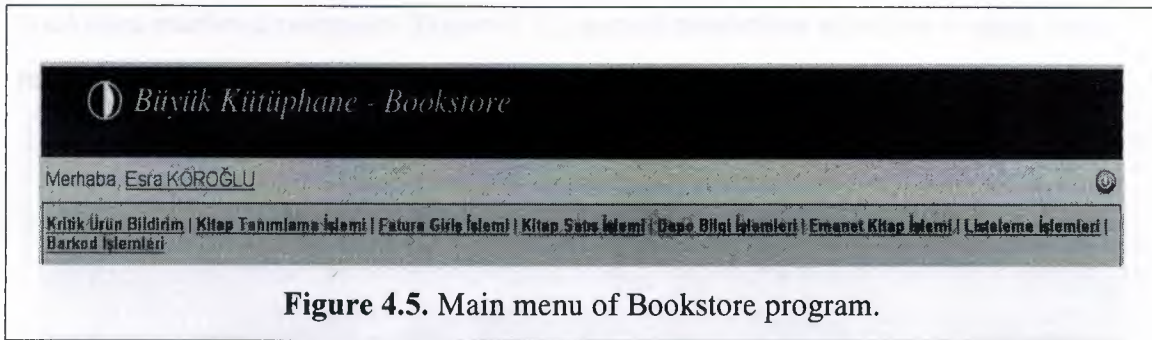


Figure 4.5. Main menu of Bookstore program.

4.4. Password Operation Page

This page is hidden page of bookstore program which can be change password (Figure 4.6.). User can see by clicked own name. When user enter the page, will see three textbox and one button. If user want to change own password, write first of all old password which is currently used, then user should write new password twice for security reason. After every steps user should click the button for finish the work. The password operation is finished like that steps.

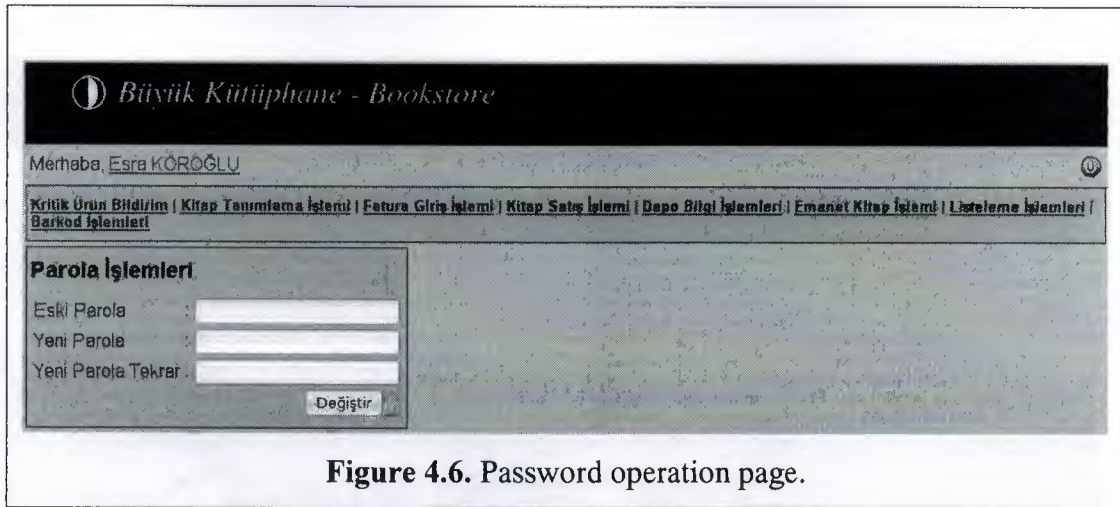


Figure 4.6. Password operation page.

4.5. Critical Inventory Alert Pages of Bookstore

This pages shows us Critical inventory of bookstore sales sides and bookstore storage sides. First combobox selection shows how many book on critical level in bookstore market department (Figure 4.7.), second combobox selection is again same reason but in bookstore storage (Figure 4.8.) department.

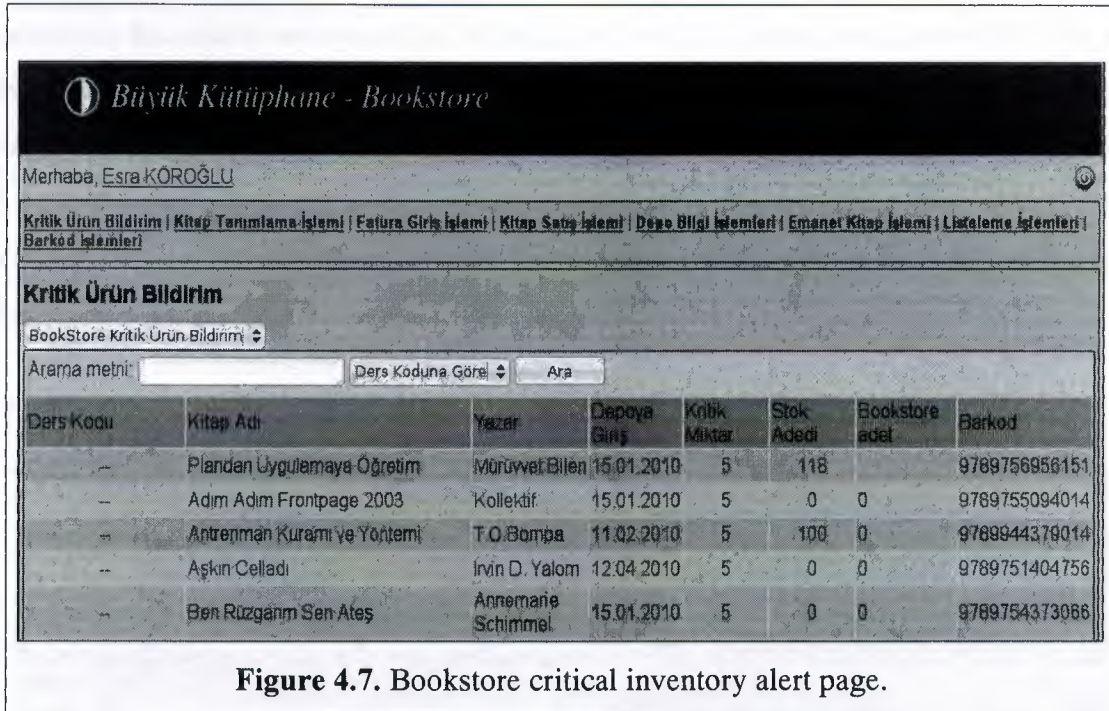
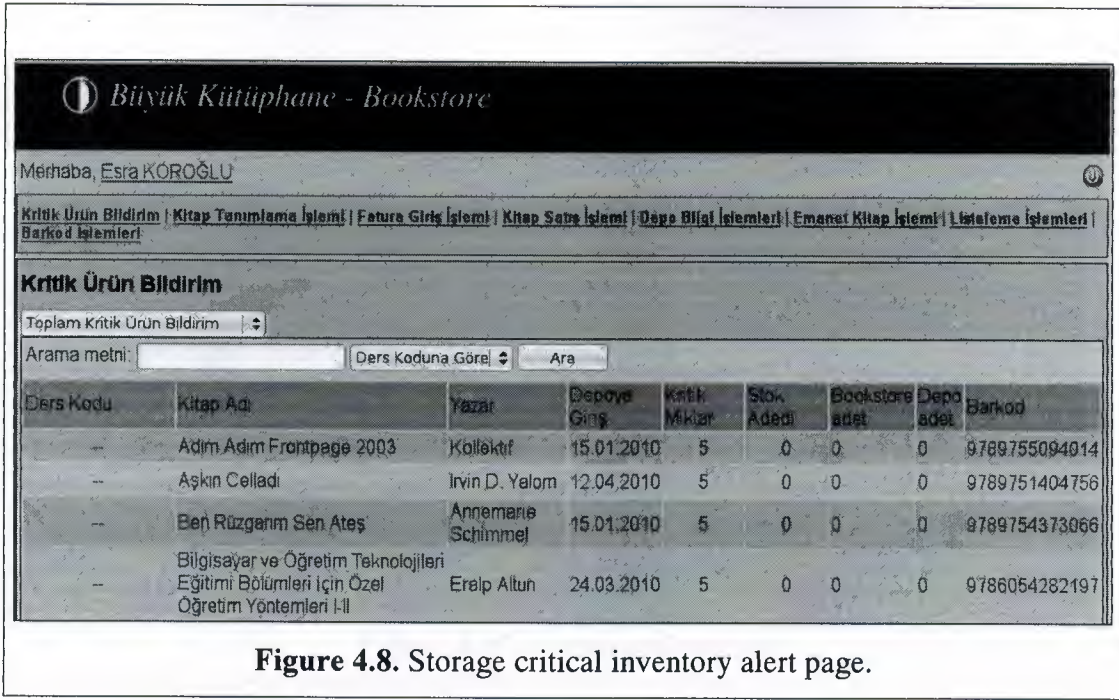


Figure 4.7. Bookstore critical inventory alert page.



4.6. Book Registration Page

This pages (Figure 4.9.) shows us how to register a new book. This operation necessary because every operation like sales the book or invoice operation need to know book information. So user must enter the book data before all operation. User can search book information from this pages and also takes print out from click to excel button but only admin edit and delete book information (Figure 4.10.). If necessary, user can see how many book totaly in stock. But of course neither user nor admin can change sum of that book.

Kritik Ürün Bildirim | Kitap Tanımlama İşlemi | Fatura Giriş İşlemi | Kitap Satış İşlemi | Depo Bilgi İşlemleri | Emanet Kitap İşlemi | Listeleme İşlemleri | Barkod İşlemleri

Kitap Tanımlama İşlemi

Ders Kodu: Kitap Adı: Yazar: Yayıncı Firma: Yayıncı Tarih:

Barkod: ISBN No: Min: Barkod: Ekle:

Tüm Liste

Arama metni: ISBN'ye Göre: Ara: Excel Çıktı:

Ders Kodu	Kitap Adı	Yazar	Yayıncı Firma	Yayıncı Tarih	Barkod	ISBN No	Min Adet	Barkod	İşlemler
--	Çeza Muhakemesi Kararı	Seçkin	Seçkin	2009	28	9750211195	5	19	9789750211195
--	System Analysis and Design Methods	Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittman	McGraw-Hill	2004	6	0074215212	5	5	9780071215213
--	Aşkın, Senal Hellen	Canan Tan	Altın	2010	14	9789752112056	5	9	9789752112056

Figure 4.9. Book registration page

Büyük Kütüphane - Bookstore

Merhaba, **Hayva Erülkü**

Kritik Ürün Bildirim | Kitap Tanımlama İşlemi | Fatura Giriş İşlemi | Kitap Satış İşlemi | Depo Bilgi İşlemleri | Emanet Kitap İşlemi | Listeleme İşlemleri | Barkod İşlemleri

Kitap Tanımlama İşlemi

Ders Kodu	Kitap Adı	Yazar	Yayıncı Firma	Yayın Tarihi
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Baskı	ISBN No	Min	Barkod	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Ekle"/>

İşlem Gerçekleştirmeye Yetkiniz Yoktur.

Tüm Liste

Arama metni: ISBNye Göre

Figure 4.10. Authorization alert.

4.7. Invoice Entry Process Page

This page also important for the book operation because user enter book quantity with this page (Figure 4.11.). If user do not enter, Can not sell that book because there is also alert about inadequate book. So every user should enter quantity information for every book. User can enter book information when enter the barcode of that book, After enter, new div open with book information (Figure 4.12.). If necessary, User can edit invoice (Figure 4.13.).

Kritik Ürün Bildirim | Kitap Tanımlama İşlemi | Fatura Giriş İşlemi | Kitap Satış İşlemi | Depo Bilgi İşlemleri | Emanet Kitap İşlemi | Listeleme İşlemleri | Barkod İşlemleri

Fatura Giriş İşlemi

Barkod Bul

Lütfen barkod girişi yapınız.

Tüm Malzemeler

Arama metni: Ara

Fatura No	Fatura Tarihi	Barkod	Kitap	Firma Adı	Alın Fiyatı	Satış Fiyatı	Miktar	İşlemler
T22.04.2010.	22.04.2010.	000000002463	Atatürk İktisadi ve İnkılap Terhhi Soru Bankası	A.E.Özkul	12	14	50	<input type="button" value="x"/> <input type="button" value="y"/>
seç	18.03.2010	9789750211195	Ceza Muhakemesi Kanunu	Seçkin	0	8	20	<input type="button" value="x"/> <input type="button" value="y"/>

Figure 4.11. Invoice page

Fatura Giriş İşlemi

Barkod Bul

Kitap adı: deneme , Kitap Yazarı: deneme , Yayıncısı: deneme

Fatura No	Fatura Tarihi	Firma Adı	Alış Fiyatı	Satış Fiyatı	Miktar

Ekle

Tüm Malzemeler

Arama metni: Ara

Excel Çıktı

Figure 4.12. Enter barcode of book.

Fatura Giriş İşlemi

Barkod Bul

Kitap adı: Atatürk İlkeleri ve İnkılap Tarihi Soru Bankası, Kitap Yazarı: K.K.T.C. Eğitim Bilimleri, Yayıncısı: K.K.T.C. Eğitim Bilimleri Yayınları

Fatura No	Fatura Tarihi	Firma Adı	Alış Fiyatı	Satış Fiyatı	Miktar
122.04.2010	22.04.2010	A. E. Özkul	12	14	50

Kaydet

Tüm Malzemeler

Figure 4.13. Edit invoice.

4.8. Book Sales Page

Sales page (Figure 4.14.) has three sub-pages. First page is sales page user insert barcode of the book then press enter, program find that book if exist, by book identify then that book shown left side of the pages (Figure 4.15.). Every enter added quantity of the book but if user want to write exactly number of the book, write quantity section exactly number then give barcode of book, press enter program again do same operation. User, if necessary, can add more then one book that list after added book, then insert esteemed, explanation part and select pay rate that book/s ready to sales with click confirm button (Figure 4.16, and Figure 4.17.).

Kıtap Satışı

Barkod	Kıtap Adı	Adet	Fiyat	Satış İşlemi
				Adet: 1
				Barkod:
				Sayın:
				Açıklama:
				Ödeme Şekli: Nakit
				Onayla Temizle Sil
				Fatura Listeleri
				Toplam:

Figure 4.14. Sales page

Kıtap Satışı

Barkod	Kıtap Adı	Adet	Fiyat	Satış İşlemi
0	deneme	1	21,3	Adet: 1
				Barkod: 0
				Sayın:
				Açıklama:
				Ödeme Şekli: Nakit
				Onayla Temizle Sil
				Fatura Listeleri
				Toplam: 21,3

Figure 4.15. Total textfield automatically added total quantity of all book, while user add the book.

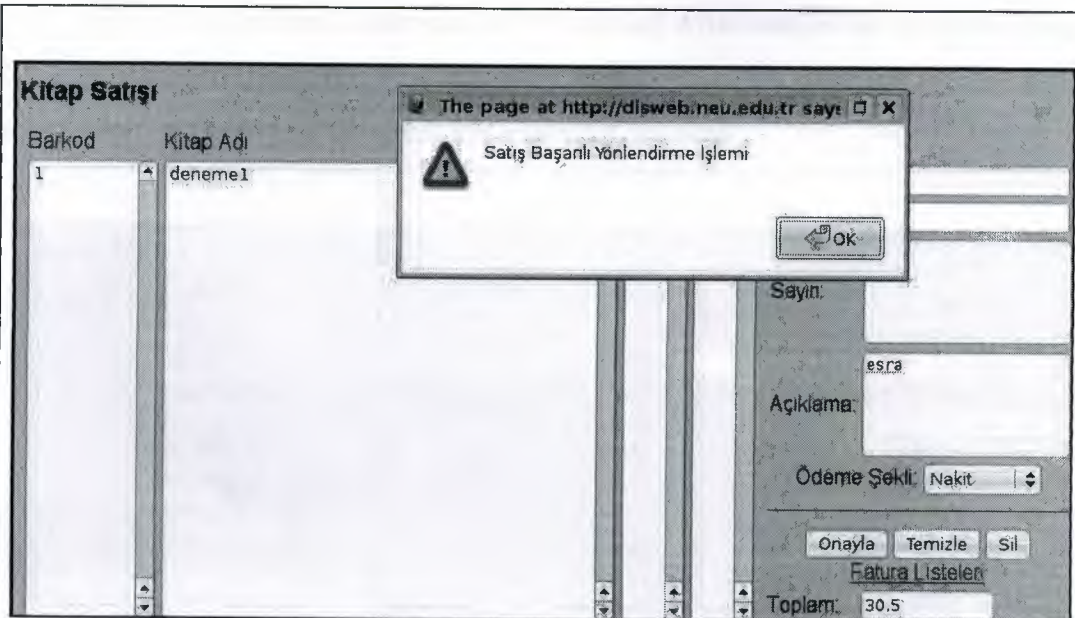


Figure 4.16. When you click confirm button, Alert windows will come.
That is about routing process is successful.

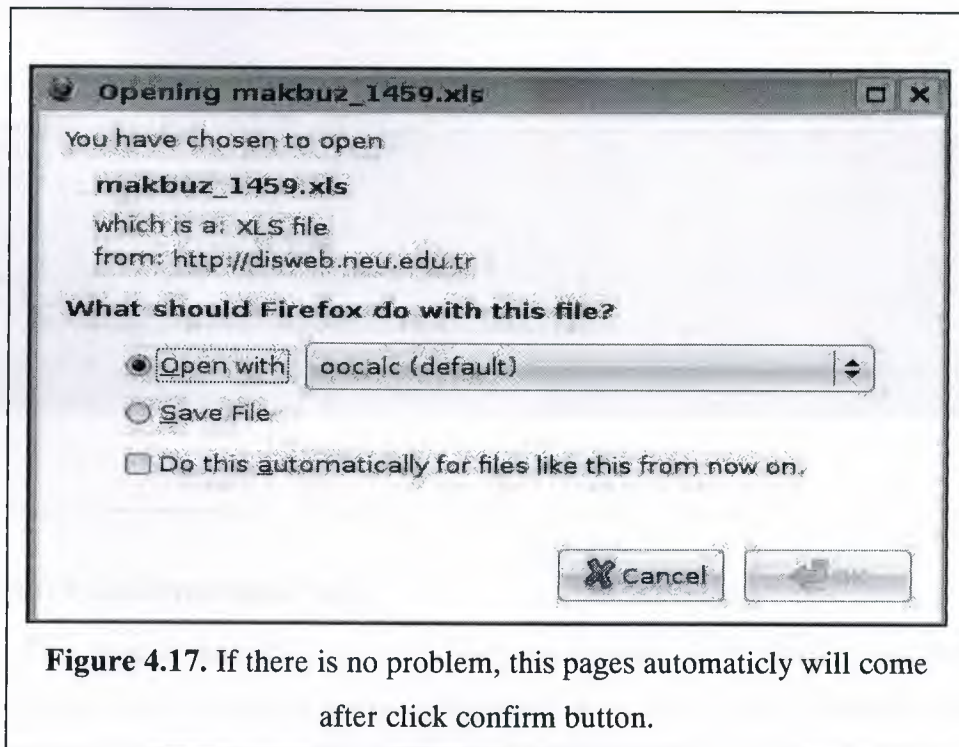


Figure 4.17. If there is no problem, this pages automaticly will come
after click confirm button.

After sales the book, user can see all invoice with invoice list guidance (Figure 4.18.). When user click that, new sub-page will come. This page for information all invoice, user can search all invoices and canceled invoices.

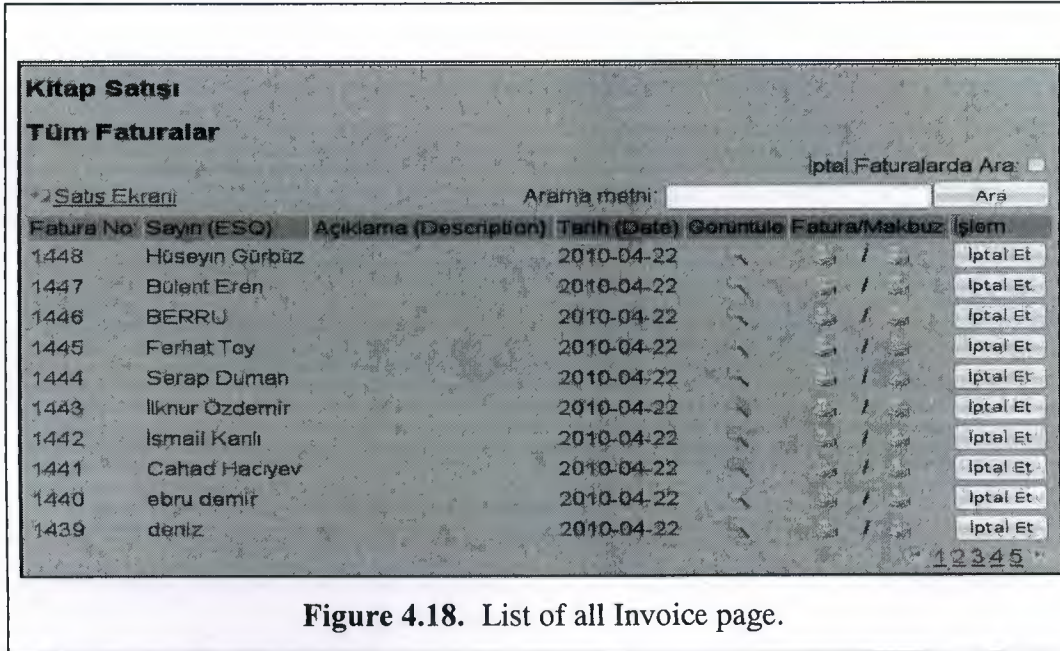


Figure 4.18. List of all Invoice page.

Then If user want to see more information of invoice, click view icon in that invoice line (Figure 4.19.). Then if necessary user can return sales page or list page with return buttons.

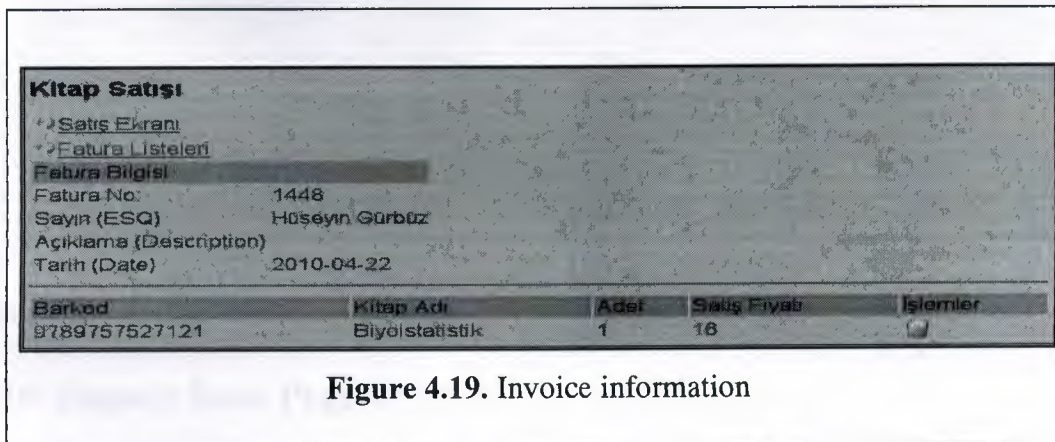
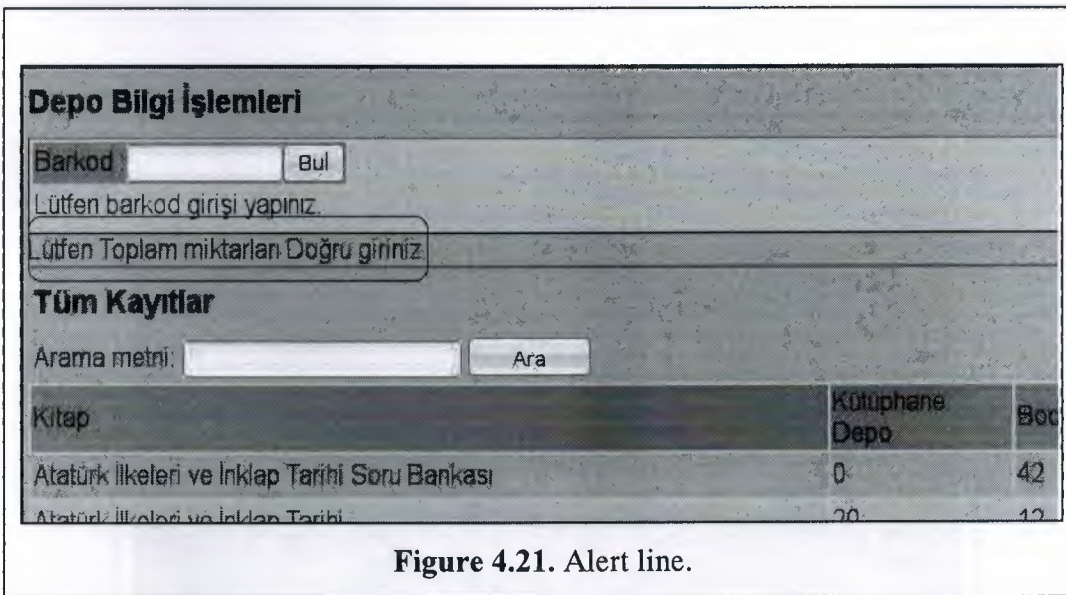
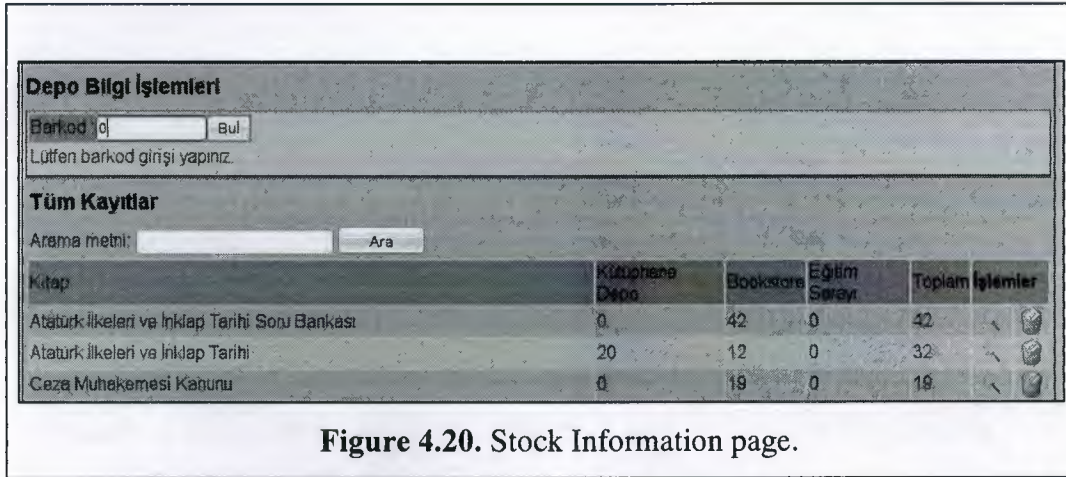


Figure 4.19. Invoice information

4.9. Stock Information Page

This page is necessary for the book sales because if you do not describe which book storage where, program do not understand where the value of stocks will count down. So every book should be register from this page (Figure 4.20.). Sometimes this page is being confused invoice entry process page that page user should be enter total

number of book but this page user should describe how many book is which storage room for example bookstore or stock storage room, and total of that book should be same as other page entry otherwise program should be alert (Figure 4.21.).



4.10. Deposit Book Page

This page (Figure 4.22.) was prepared to keep information of book which that give akademik staff. Why we do not use sales page? because we do not sell the book just give deposited the book but of course when user do this process sum of book number will decrease. User can add when enter barcode of that book and book information will come with some textfield. After fiil in the blank textfield process is

finished (Figure 4.23.) click add button then automatically decrease sum of book from stock.

Figure 4.22. Deposit book page.

Figure 4.23. Added book to the table.

4.11. Listing Process Page

The last page of the user interface side (Figure 4.24.). User can search every thing with this page and also can take to day-end report with this page.

Merhaba, Haywa Erülkü

[Kritik Ürün Bildirim](#) | [Kitap Tanımlama İşlemi](#) | [Fatura Giriş](#) | [Barkod İşlemleri](#)

Listeleme İşlemleri

Raporlama

Ders Kodu:

Kitap Adı:

Alış Fiyatı: ile

Satış Fiyatı: ile

Depodaki Miktar: ile

Yazar:

Fatura No:

İşlem Tarihi: ile

Fatura Girişinde: ☐

Ödeme Şekli:

Figure 4.24. Listing page of user interface side.

Merhaba, Esra KÖROĞLU

[Kritik Ürün Bildirim](#) | [Kitap Tanımlama İşlemi](#) | [Fatura Giriş](#) | [Barkod İşlemleri](#)

Listeleme İşlemleri

Raporlama

Ders Kodu:

Kitap Adı:

Alış Fiyatı: ile

Satış Fiyatı: ile

Depodaki Miktar: ile

Yazar:

Fatura No:

İşlem Tarihi: ile

Kullanıcı Adı:

Fatura Girişinde: ☐

Ödeme Şekli:

Figure 4.25. Listing page of admin interface side.

Admin interface side (Figure 4.25.) of this page has a little difference operation like search by username but user interface side, each user can view and list only own operations but can not see other user operations. After all this information you can search whatever you want by click the search button (Figure 4.26.), and If necessary, user can transfer to excel form of this list (Figure 4.27.).

Listeleme İşlemleri									
Sonuçlar									
Geri									Excel Çıktı
Ders Kodu	Kitap Adı	Alış Fiyatı	Satış Fiyatı	Stok Adet	Yazar	Fatura No	İşlem Zamanı	Ödeme Şekli	İşlem Yapan
silmeğin.	deneme	12.00	21.00	32232	deneme 1		2010-03-15 11:53:52	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 2		2010-03-15 11:54:18	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 3		2010-03-15 11:59:11	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 4		2010-03-15 11:59:24	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 5		2010-03-15 12:00:03	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 6		2010-03-15 12:00:38	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 7		2010-03-15 12:07:51	Nakit	Esra KÖROĞLU
silmeğin.	deneme	12.00	21.00	32232	deneme 8		2010-03-15 12:08:08	Nakit	Esra KÖROĞLU

Figure 4.26. Listing process

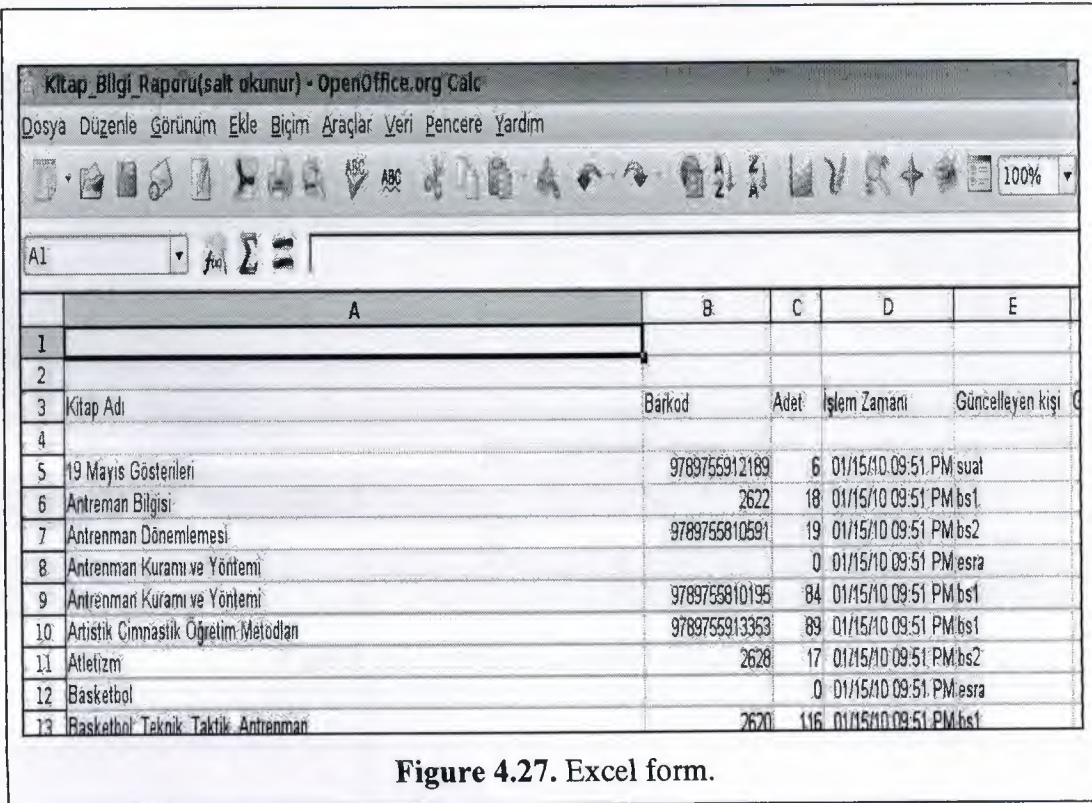



Figure 4.27. Excel form.

4.12. Barcode Transactions Page

This page (Figure 4.28.) is only for admin interface. Admin can printing barcode with this page. Admin can print with three options;

- With same barkod
- With sequential barcode
- With giving the exactly barcode number (Figure 4.31.).

First line is automatic line, admin give number (Figure 4.29.) barcode how many want to printing then click the button new windows will come as you see below figure 4.30. If Admin click 'same barcode' checkbox area then program print same barcode (Figure 4.31.) which how many barcode write that field. Second line for spesific barcode number like 123456789012 (Figure 4.32.). admin write 12-character barkod (See for alert Figure 4.33.).

 *Büyük Kütüphane - Bookstore*

Merhaba, Esra KOROĞLU

Kritik Ürün Bildirim | Kitap Tanımlama İşlemi | Fatura Giriş İşlemi
Barkod İşlemleri

Barkod İşlemleri

Barkod Sayısı: ☐ Aynı Barkod

Sabit Barkod: Adet:

Yakın Doğu Üniversitesi, Tel: +90 392

Figure 4.28. Barcode Transaction page

Barkod İşlemleri

Barkod Sayısı: ☐ Aynı Barkod

Sabit Barkod: Adet:

Figure 4.29. Write how many barcode want to print.

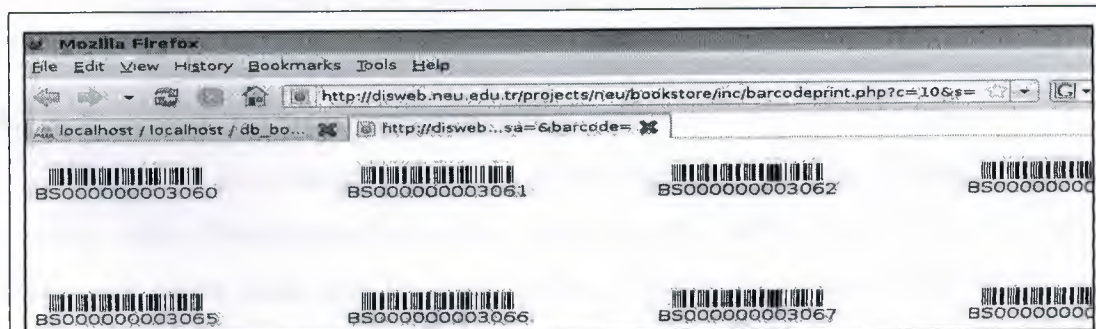


Figure 4.30. Barcodes.

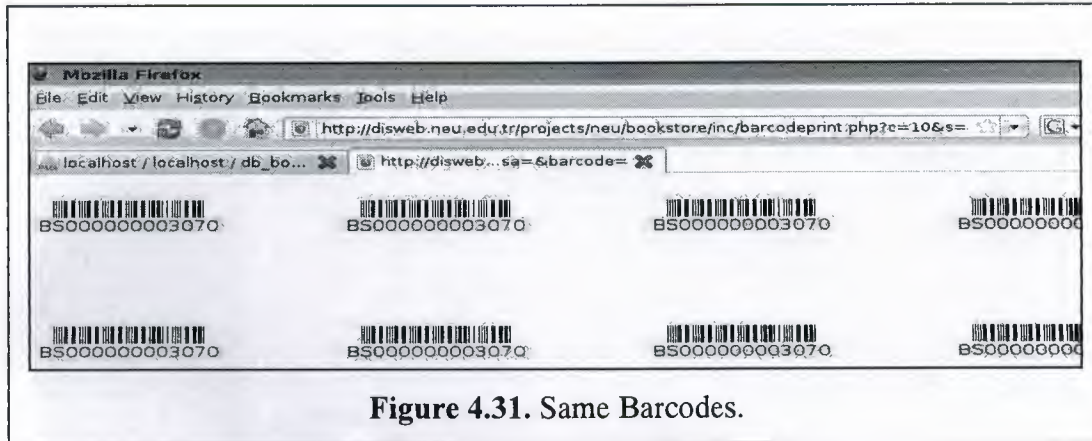


Figure 4.31. Same Barcodes.

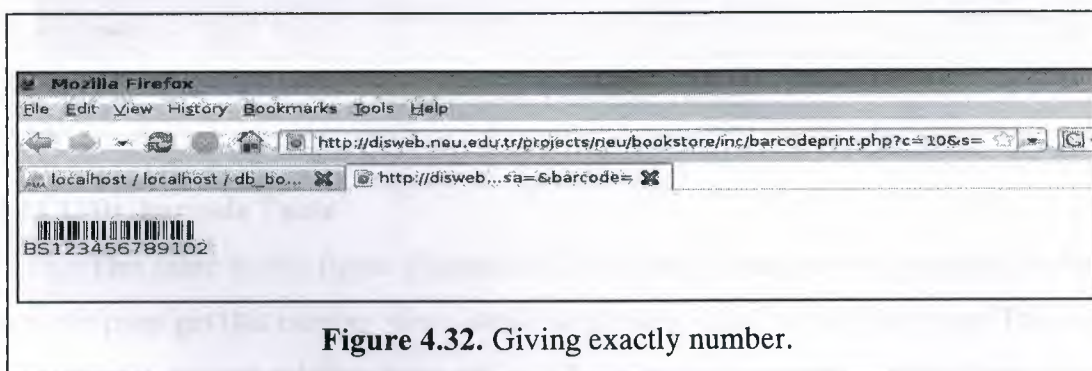


Figure 4.32. Giving exactly number.

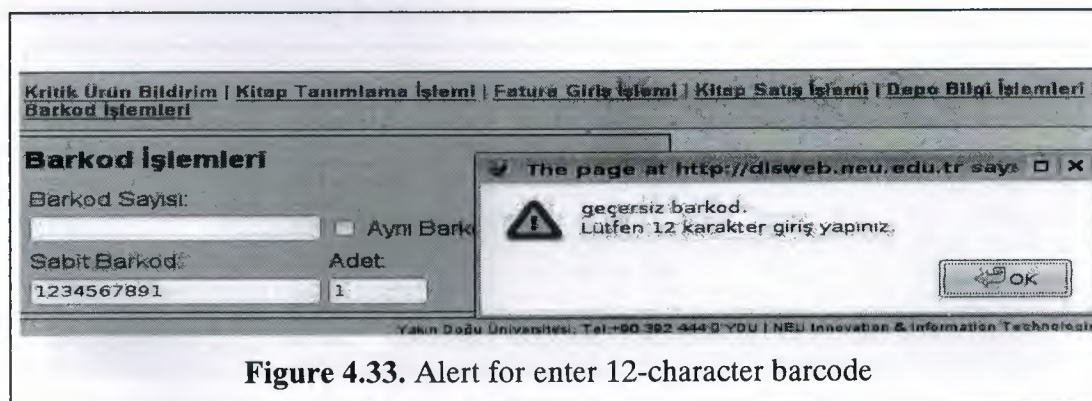


Figure 4.33. Alert for enter 12-character barcode

4.13. BookStore Database Structure

Database structure (Figure 4.34.) of bookstore program has seventeen relational database table. Fundamentally twelve tables but five table for removal transaction. When user delete some data for some reason, actually insert these tables at the same time.

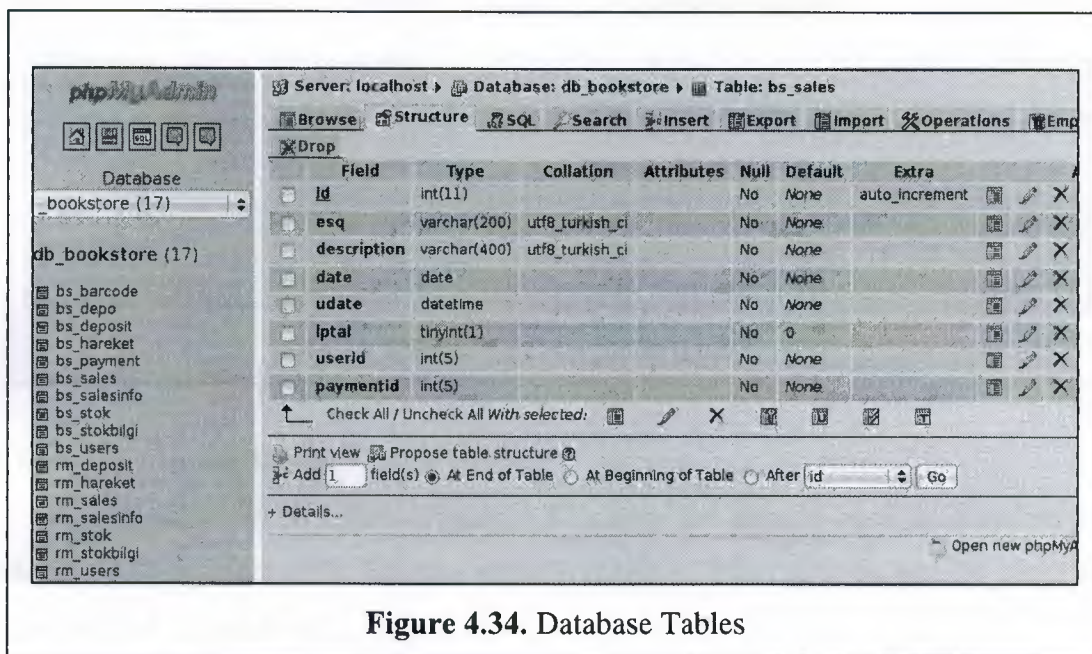


Figure 4.34. Database Tables

4.13.1. Bs_barcode Table

This table in the figure (Figure 4.35.) for keep latest barcode number. Printing barcode page get this number when admin select sequential part of that page. This table in the figure not any relation other tables. The table has just one field that type integer for keep number how many barcode printing.

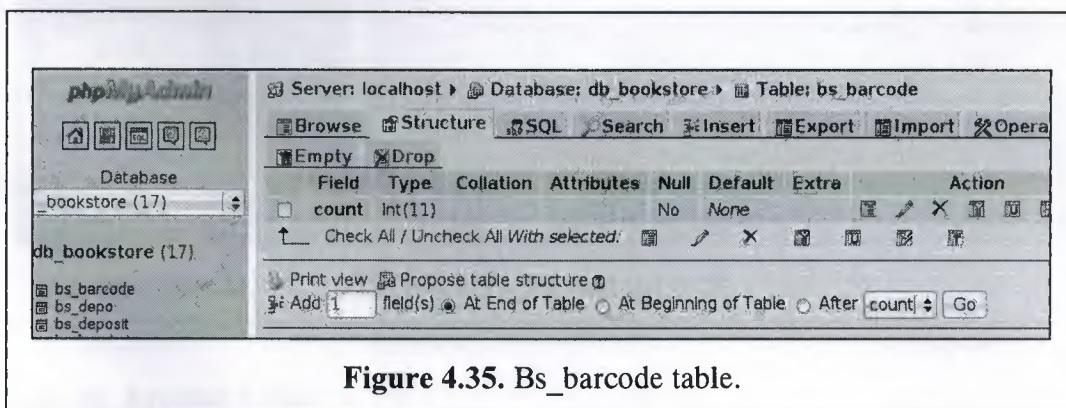
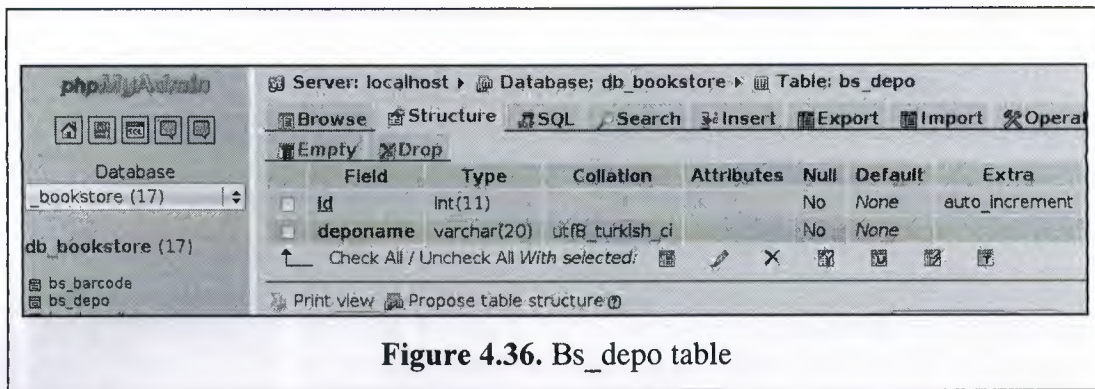


Figure 4.35. Bs_barcode table.

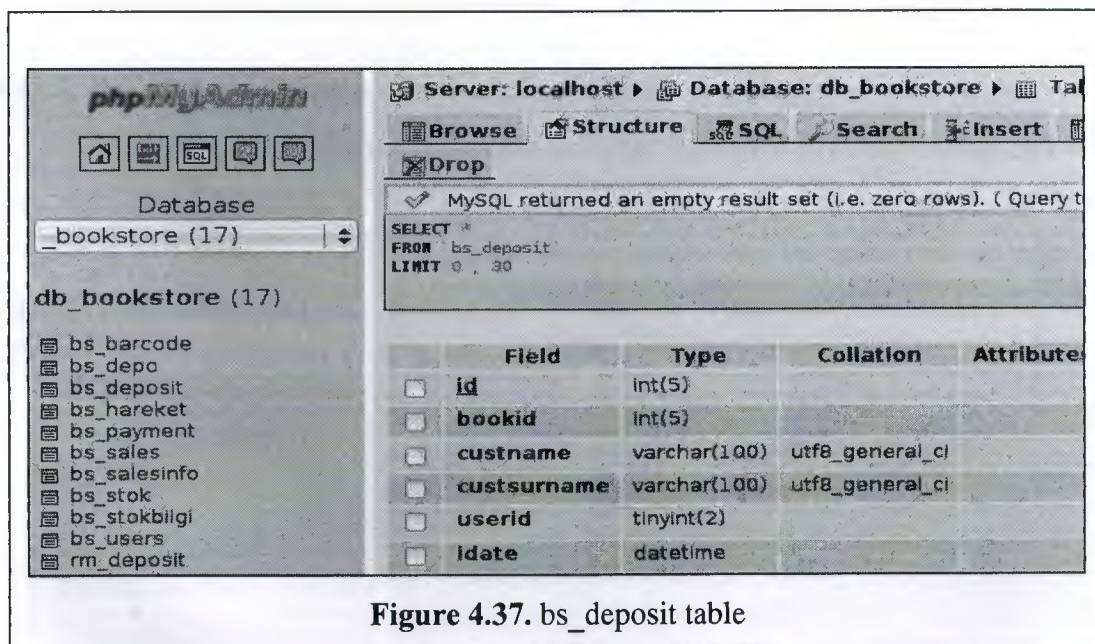
4.13.2. Bs_depo Table

This table in the figure (Figure 4.36.) for keep name of storage place. The table has two field first one for auto-increment identification, second one is for keep storage name. This table in the figure relation with bs_users table, relation name is id-depoid relation. This relation necessary for which user include which storage place.



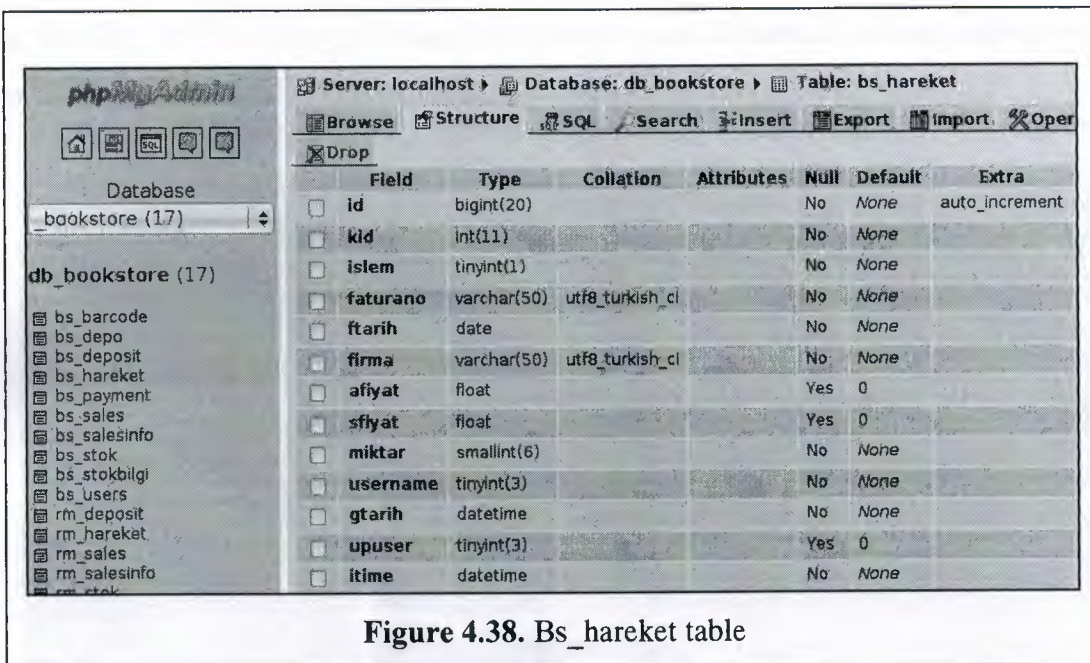
4.13.3. Bs_deposit Table

This table in the figure (Figure 4.37.) keep the deposit book information and relation with bs_users table. Relation name is userid-id, other relation with bs_stock and relation field is bookid-id.



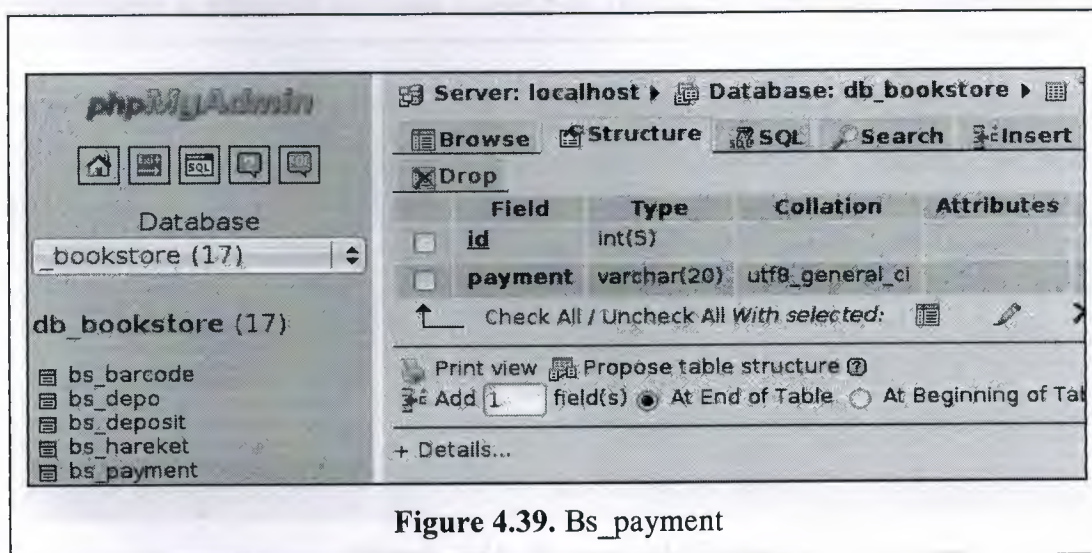
4.13.4. Bs_hareket Table

This table in the figure for invoice page. The table (Figure 4.38.) keep price of books. This table in the figure relation with bs_users table and bs_stok table. Relation name with bs_user is username,upuser-id and relation name with bs_stok is kid-id. We keep userid twice because username for who is record invoice firstly. Upuser for who was changed this invoice after first record. Kid field is book identification and come from stock table.



4.13.5. Bs_payment Table

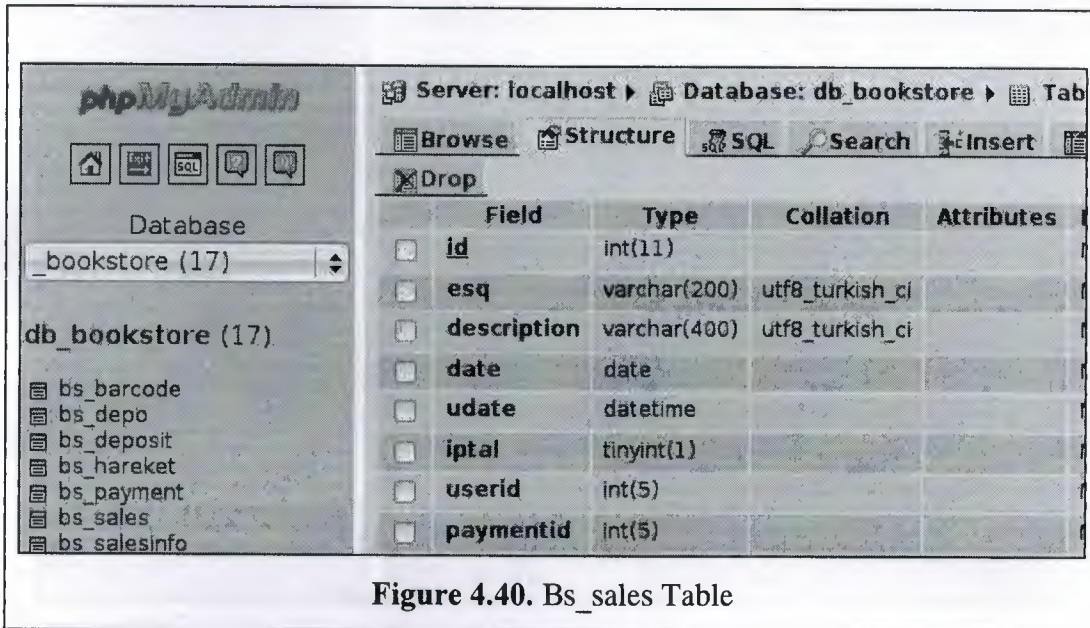
This table in the figure for book sales page invoice and receipt needs payment procedure and This table in the figure keeps payment name. This table in the figure (Figure 4.39.) relation with two tables which names are bs_sales and bs_salesinfo relation names id-paymentid.



4.13.6. Bs_sales Table

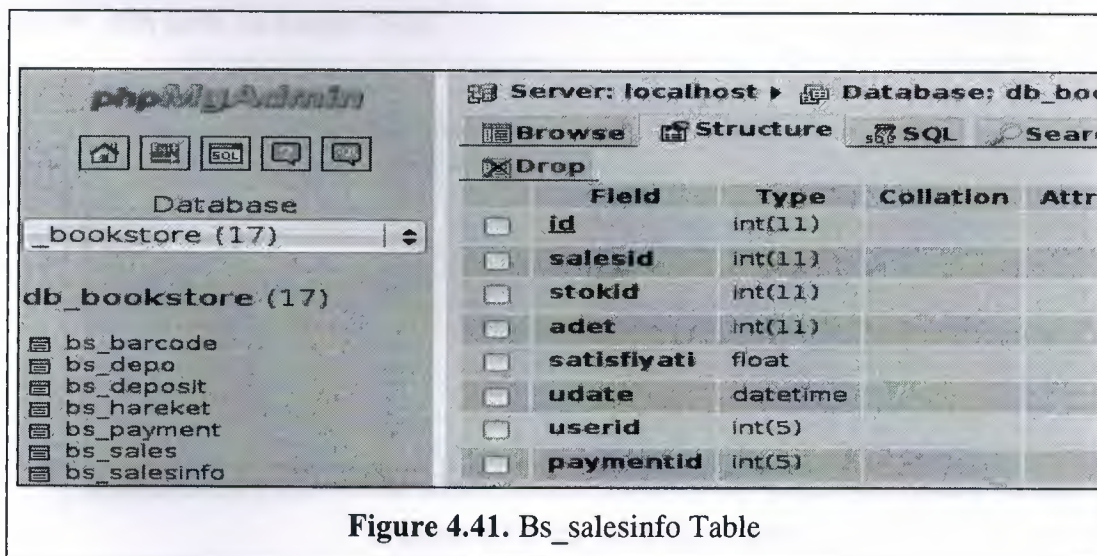
This table in the figure (Figure 4.40.) for book sales page and has eight fields. General book sales information keep This table in the figure and information of that invoice keep bs_salesinfo tables. So This table in the figure relation with three tables,

bs_salesinfo, bs_users, bs_payment. First relation with id-salesid, second relation with userid-id, last relation is paymentid-id.



4.13.7. Bs_salesinfo Table

This table in the figure (Figure 4.41.) detail of invoice. Relation tables are; bs_sales, bs_stok, bs_users, bs_payment. First relation name is salesid-id, second relation name is stokid-id third one is userid-id, the last relational field is paymentid-id.



4.13.8. Bs_stok Table

This table in the figure (Figure 4.42.) for keep book information like autor, publisher, total amount of book. Book register page use This table in the figure for insert data. Relation with bs_users table on id-username and id-upuser.

Server: localhost Database: db_bookstore

Database: bookstore (17)

db_bookstore (17)

- bs_barcode
- bs_depo
- bs_deposit
- bs_hareket
- bs_payment
- bs_sales
- bs_salesinfo
- bs_stok
- bs_stokbilgi
- bs_users
- rm_deposit
- rm_hareket
- rm_sales
- rm_salesinfo
- rm_stok
- rm_stokbilgi
- rm_users

Field	Type	Collation	Attributes	Null	Default
<input type="checkbox"/> id	int(11)			No	None
<input type="checkbox"/> ciscode	varchar(200)	utf8_turkish_ci		No	None
<input type="checkbox"/> title	varchar(100)	utf8_turkish_ci		No	None
<input type="checkbox"/> author	varchar(100)	utf8_turkish_ci		No	None
<input type="checkbox"/> publisher	varchar(100)	utf8_turkish_ci		No	None
<input type="checkbox"/> publishdate	smallint(6)			No	None
<input type="checkbox"/> edition	int(11)			No	None
<input type="checkbox"/> isbn	varchar(100)	utf8_turkish_ci		No	None
<input type="checkbox"/> total	smallint(6)			No	None
<input type="checkbox"/> barcode	varchar(20)	utf8_turkish_ci		No	None
<input type="checkbox"/> min	smallint(6)			No	None
<input type="checkbox"/> gdate	datetime			No	None
<input type="checkbox"/> username	tinyint(3)			No	None
<input type="checkbox"/> upuser	tinyint(3)			No	None
<input type="checkbox"/> ltime	datetime			No	None

Figure 4.42. Bs_stok Table.

4.13.9. Bs_stokbilgi Table

This table in the figure (Figure 4.43.) keep book information like which book storage where. The table relation with bs_users on userid and upuserid, and relation with bs_stok table on kitapid field.

Server: localhost Database: db_bookstore Table: bs_stokbilgi

Database: bookstore (17)

db_bookstore (17)

- bs_barcode
- bs_depo
- bs_deposit
- bs_hareket
- bs_payment
- bs_sales
- bs_salesinfo
- bs_stok
- bs_stokbilgi
- bs_users
- rm_deposit
- rm_hareket
- rm_sales
- rm_salesinfo
- rm_stok
- rm_stokbilgi
- rm_users

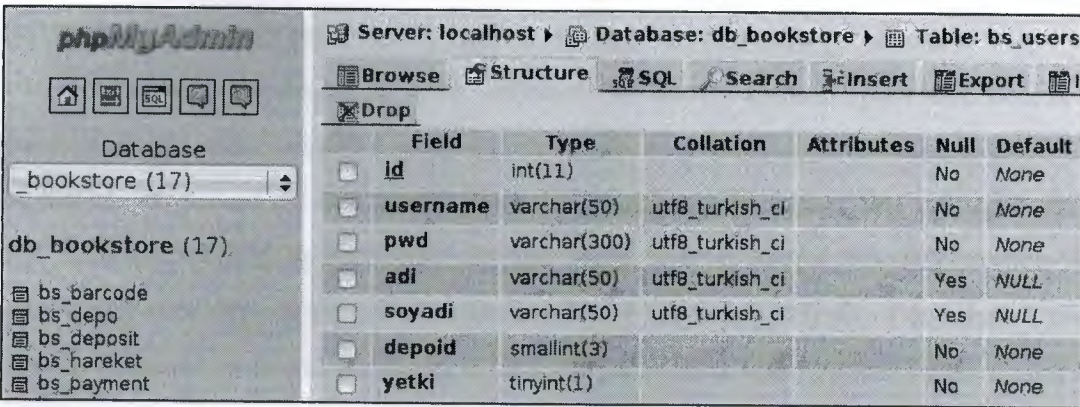
Field	Type	Collation	Attributes	Null	Default
<input type="checkbox"/> id	int(5)			No	None
<input checked="" type="checkbox"/> kitapid	varchar(20)	utf8_turkish_ci		No	None
<input type="checkbox"/> anadepo	varchar(6)	utf8_turkish_ci		No	0
<input type="checkbox"/> bs1	varchar(6)	utf8_turkish_ci		No	0
<input type="checkbox"/> bs2	varchar(6)	utf8_turkish_ci		No	0
<input type="checkbox"/> userid	varchar(6)	utf8_turkish_ci		No	None
<input type="checkbox"/> upuserid	varchar(6)	utf8_turkish_ci		No	None
<input type="checkbox"/> ltime	datetime			No	None
<input type="checkbox"/> barcode	varchar(20)	utf8_turkish_ci		Yes	NULL

Figure 4.43. Bs_stokbilgi Table

4.13.10. Bs_users Table

This table in the figure (Figure 4.44.) is used everywhere in program. Because This table in the figure determined user of the program, and relation with whole tables

of database. Because when users some do some transtaction program automaticly save userid for which user do that transaction.



The screenshot shows the phpMyAdmin interface. On the left, the database '_bookstore (17)' is selected, and the table 'db_bookstore (17)' is expanded, showing several tables including 'bs_barcode', 'bs_depo', 'bs_deposit', 'bs_hareket', and 'bs_payment'. The main panel shows the 'Table: bs_users' structure. The table has the following fields:

Field	Type	Collation	Attributes	Null	Default
<input type="checkbox"/> Id	int(11)			No	None
<input type="checkbox"/> username	varchar(50)	utf8_turkish_ci		No	None
<input type="checkbox"/> pwd	varchar(300)	utf8_turkish_ci		No	None
<input type="checkbox"/> adi	varchar(50)	utf8_turkish_ci		Yes	NULL
<input type="checkbox"/> soyadi	varchar(50)	utf8_turkish_ci		Yes	NULL
<input type="checkbox"/> depoid	smallint(3)			No	None
<input type="checkbox"/> yetki	tinyint(1)			No	None

Figure 4.44. Bs_users Table.

4.13.11. Remove-Tables

There are seven tables for removal transaction. The purpose of these tables, security reasons. When some user delete important data we will know which user was deleted and when, and other important point is to prevent data loss. These tables; rm_deposit, rm_hareket, rm_sales, rm_salesinfo, rm_stok, rm_stokbilgi, rm_users.

CONCLUSION

Bookstore inventory program control the stock price, stock level of bookstore, and program also helps the user in the book sales, User can take report own work with excel button and search procedure which is all page has that to help the users.

This program designed with intranet. In short, intranet provides access to information and other services quickly and easily in a familiar environment since it is like a web site found on the World Wide Web with a home page and related links to other information and applications. At the same time it keeps this information and applications securely off the public Internet. There are many benefit of intranet. Unlike the traditional paper-based documents, intranet provides a wide variety of media types, which include audio, video, and interactive applications. These features allow users to communicate in a dynamic way and to enhance the accuracy and speed of information sharing. An intranet is also well suited for multimedia applications. The intranet also contributes directly to the effectiveness of the company in producing business results. For example faster time to market by better communication to and with affected parties, improved coordination and agreements and rapid information access. Effectiveness also increases by better, faster feedback, adaptation, correction and increased participation.

The program working in intranet but if necessary it is portable to internet, and user can be work with all operation systems like Mac, Windows, Linux. because it was designed independently from the platform.

REFERENCES

- [1] A guide for writing research for microcontroller, Retrieved May 17, 2011 from the World Wide Web:“ <http://www.merchantos.com/articles/inventory>”
- [2] A guide for the first chapter, Retrieved May 17, 2011 from the World Wide Web: “http://en.wikipedia.org/wiki/Safety_stock”
- [3] Reasearch for the first chapter, Retrieved May 19, 2011 from the World Wide Web: “<http://www.businesslink.gov.uk>”
- [4] Reasearch, Retrieved May 19, 2011 from the World Wide Web: “http://www.prenhall.com/rm_student/html/glossary/r_gloss.html”
- [5] A guide for database management, Retrieved May 23, 2011 from the World Wide Web: “http://en.wikipedia.org/wiki/Database_management_system”
- [6] A guide for how to write software projects, Retrieved May 28, 2011 from the World Wide Web:“<http://www.softwareprojects.org/>”
- [7] Reasearch about basics of php language, Retrieved May 29, 2011 from the World Wide Web:“<http://php.about.com/od/phpbasics/>”
- [8] A guide for Apache server, Retrieved May 29, 2011 from the World Wide Web: “http://www.webopedia.com/TERM/A/Apache_Web_server.html”
- [9] Khashman A., Abiyev R., *Graduation project: A complete guide 1st ed.*, Near East University Press, Nicosia, 2001
- [10] Welling L., Thomson L., *PHP ve MySQL Uzmanlar İçin Uygulama ve Geliştirme Kılavuzu 7th ed.* , Alfa Basım Yayım Dağıtım, Istanbul, 2009
- [11] Delisle M., *Mastering phpMyAdmin 3.1 for Effective MySQL Management 4th ed.* , Packt Publishing, 2009
- [12] Meloni J., *Teach Yourself PHP, MySQL and Apache in 24 Hours 4th ed.* , Sams Publishing, 2003

APPENDIX A

Appendix-A included source code and all the source codes of the program implemented in the compact disc (CD).