SECURE E-COMMERCE FOR IBN ALHAYTHAM MEDICAL EQUIPMENT

By

Samer Qadan
Murad Al-Kubabjy
Abdulla Al-Natsheh

Supervisor

Eng.Murad Abu Subaieh

Submitted to the Collage of Administrative Science and Information System as partial fulfillment of the requirements for the degree of B.Sc. in Information System

Palestine Polytechnic University
Collage of Administrative Science and Information System

2003/2004

Acknowledgment

We would like to thank our families and our friends for their support and encouragement through our study.

Also, we would like to thank our supervisor Eng. Murad abu subaieh for his efforts and notes.

Samer

Murad

Abdulla

Dedication

To our parents …

To our families …

To our country …

To our supervisor Eng. Murad abu subaieh …

To our friends and any one who loves us …

Samer

Murad

Abdulla

Abstract

This system aims to computerizing the remote purchasing process in Ibn_ALhaytham medical equipment company. This system allows the customer to shopping remotely through IHME site. The system also allows the administrator to make changes, view reports and update the site through the Internet.

نظام التسوق عن بعد لدى شركة ابن الهيثم للمعدات الطبية يهدف الى تمكين الزبائن من اجراء عملية التسوق للمنتجات باستخدام الانترنت عبر طريقتين للدفع، تسهيلا على زبائنهم و ايفاءً لرغباتهم.

كما يوفر النظام صفحة خاصة للمشرفين الذين بامكانهم عرض التقارير او اضافة منتج جديد او حتى التعديل على المنتجات السابقة بسهولة ويسر.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter One

Introduction

1.1 Introduction

Electronic commerce (the production, distribution, marketing, sale, or delivery of goods and services by electronic means) is not a new form of trade but rather a new medium mode for conducting trade in goods and services. All forms of electronic commerce, as traditional commerce, can be categorized into either goods or services, therefore, specific agreements governing the trading in goods and services using electronic transactions.

Electronic transactions involve transferring of information, products, services or payments via electronic networks. This includes the use of electronic communication as the medium through which goods and services of economic value are designed, produced, advertised, catalogued, inventoried, purchased or delivered.

Today, e-commerce is transforming the international trade landscape. Experts predict that e-commerce will generate worldwide more than $4 trillion in sales in the year 2003, because this development has such potential to benefit economic growth and living standards around the world.

It is important to recognize the need for global cooperation by both business and Governments to facilitate electronic commerce. Otherwise, there is a risk that a large segment of the world's population may not be able to participate in the economic and social benefits that can arise from electronic commerce. Information technology and electronic networks provide a new communications infrastructure which will give more people access to scientific and technical information worldwide, not only in developed countries but also, increasingly, developing countries.

As technological innovations rapidly advance, the medical products needed are more diverse and sophisticated. The secure e-commerce developed for Ibn Al-hytham to ensure that its customers receive the finest in medical products, information, and services in a reliable and timely manner. Ibn Al-hytham focusing on its relationship with the medical community. Ibn Al-haytham was founded in December 1986, in Hebron, Palestine.

1.2 Economic and Social Benefits

The AGB (Alliance for Global Business) views electronic commerce as an innovative approach to ensure future sustainable economic growth. The impact of electronic commerce on the economies will improve economic efficiency, competitiveness and profitability. Within such an environment, countries in all stages of development will have the opportunity to benefit by:

1. Increasing internal organizational and management efficiency of enterprises[1].
2. Increasing transaction efficiency and reducing transaction costs for both suppliers and buyers.
3. Extending market reach of suppliers and increasing choice for both suppliers and consumers.
4. Providing accurate information to improve service delivery such as in health provision or the provision of information to consumers.
5. Shrinking the production and distribution chain by improving the efficiency of intermediation and maximizing the value it adds to the entire process.
6. Providing virtual shopping facilities that will change concepts of retailing for a number of goods and services and enhancing the ability of customers to browse and choose new products and services irrespective of location.
7. Increasing market competition as costs for consumers are reduced and as market entry costs for suppliers are lowered.
8. Raising productivity growth and the development of new products or services will lead to new job creation.

Electronic commerce facilitates established Business-to-Business commercial relations, on witch sales made by companies to other companies, other forms of electronic commerce is Business-to-Consumer commercial relation on witch sales made by company to consumers.

[1] E-commerce Planning and Management, 3rd edition, 2001.

1.3 Problem

After the development team has made an exploration visit to Ibn_Alhaytham medical equipment company, they noticed that all of its operations are done manually in traditional ways without using the computer for any thing that go beyond the financial operations. Thus, the development team proposes the idea of applying the concept of e-commerce in the company in order to face:

1. Difficulties in marketing and promotional techniques that reduce sales.
2. The lack of the customer numbers who's only from a local areas.
3. The costs of entering new market place for medical equipment in different area.
4. Effort and time consuming in the purchasing process which increase the cost for both the company and customer.

1.4 Proposed solution:

Through out the study of the company situation and activities, it was found that applying the concept of e-commerce through a web site on which all information about the company products and services are promoted and advertised can reduce many of the company problems and drastically improve its situation. In addition, it is a good step toward the company strategic development plan. Here are some of the advantages that are expected to be gained:

1. More effective marketing that allows displaying all company's products 24 hours a day.
2. Having global customers around the world, and to have effective communication with them.
3. Reducing the cost of promotional efforts and facilitate the purchasing process.
4. Save money by reduce the number of employees in marketing department.

1.5 Report overview:

The rest of this report is organized as: In chapter two the system risk analysis, restrictions and feasibility study are included. Chapter three includes the requirement analysis, while chapter four discuses system analysis. Chapter five presents system design and interface, input, output, data base design is presented. In chapter six the steps of implementation are included. The last chapter is concerned with system testing, unit and integration testing is included.

Chapter Two

System Planning

2.1 Objectives

This system aims to give the company competitive advantage in its management, marketing and financial department in many different ways for the company and the customer in different ways by:

1. Building effective database to handle product reservation and administration.
2. Reducing the cost of the company current system.
3. Having secured and dependable data storage.
4. Improving the marketing process.
5. Increasing operation efficiency.
6. Remotely control and updating system processes by the administration department.

2.2 Risk analysis:

1. The fear of the employees when applying the new technology and resisting the change.
2. Problems in applying the system especially in the local market because they are not familiar with Internet financial operations.
3. Security and hackers risks on financial issues.

- Risk solution: this group of risk can over come by a good training for employees, increase the awareness for this system and its benefit in the local area because there is a lack of experience in dealing with system. The most important point is the security issue, so the more secure tools will choose for both design and implementation phase.

2.3 Restrictions

- Building and implementing of the system during 12 weeks.
- Total cost should not increase more than $10221, as explained in feasibility study.

2.4 System planning

2.4.1 Feasibility study:

2.4.1.1 Economical study:

- **Hardware for design phase:**

1	Two Microsoft compatible PC.III Pentiums (Pentium 4 recommended) Pentium III, 1000 MHZ, cache 256KB. RAM 128 MB Hard Disk Drive 20 G.B Floppy Drive 1.44 CD-ROM 52X Monitor 15" Keyboard & Mouse Modem Network adapter Cables (twisted pair RJ45) Printer (HP 3420)	$1800 $10 $100
	Total	$1910

Table (2.1) Hardware cost

- **Operating software:**

1	Windows2000 server	$200
2	Windows 200 professional	$200
3	PHP and MYSQL software	$200
4	Microsoft Word 2000	$255
5	Flash MX	$300
6	Adobe Photoshop 7.0	$615
	Total	$1764

Table (2.2) Software cost

- **Implementation software:**

1	Dreamweaver MX	$300
	Total	$300

Table (2.3) Implementation software cost

- **Team costs:**

No.	Name	Number of hours/weeks	Cost /hour	Total
1	Samer Qa'dan	35	$15	$525
2	Murad kubabjy	35	$15	$525
3	Abdulah saed	35	$15	$525
			Total/week	$1575

Table (2.4) Team cost

2.4.1.2 Time schedule:

The schedule is estimated depending on the estimation of the project, Steps and phases. In this project the task are distributed among 12 weeks as:

T1	System definition	2week
T2	Requirement analysis	2 weeks
T3	System design	2 weeks
T4	Implementation	3 week
T5	Testing	3 weeks
T6	Documentation	All the time
	Total	12 weeks

Table (2.5) Time scheduling

2.4.2 Schedule table:

Time (Weeks) / Task	1	2	3	4	5	6	7	8	9	10	11	12
T1												
T2												
T3												
T4												
T5												
T6												

Chapter Three

Requirements Analysis

3.1 Functional requirements:

3.1.1 Requirement definition:

The system should be able to:

1. Provide information about the products through the website.
2. Allow reservation of products.
3. The ability to rollback.
4. Allow the administrator to update the database remotely.
5. Allow customers to register in the system.
6. Support the payment by electronic way (credit card) or by demand bill.
7. Accept comments from customers.
8. Display reports about orders, products and customer details.

3.1.1.1 Provide information about the products through the website.

The system should be able to provide information about the products, prices and features by allowing the user to browse the web site to see all product categories and particularly in a simple way that going with the price changing.

3.1.1.2 Allow reservation of products.

After the user explore the website and decide what products he/she wants, he/she has the ability to make reservation for the products through the reservation form on which all products features and prices are clearly specified before the submission process when the user selects the products and fills the reservation form, he/she must press the submit button in order to emphasis the reservation.

3.1.1.3 The ability to rollback

After the reservation form is completed the user has the ability to rollback before submitting the reservation form. By rolling back, the user has more choices to add some products or remove others from the reservation form or to cancel the whole process.

3.1.1.4 Allow the system administrator to update the database remotely

Allow system administrator to make changes in the database, this includes information, prices, features and type of products. The administrator also can add, remove products and update the prices using authentication and authorization (by using user name and password to login in the system) methods, he also can change his password at any time for more security.

3.1.1.5 Allow customers to register in the system

The system allows the new customer to register in order to capture his information once, by applying this concept the customer will avoid entering his information every time he wants to purchase products. The registration process gives the customer user name and password. Also, the customer has the ability to change his password.

3.1.1.6 Support the payment by electronic way (credit card) or by demand bill

When the reservation is submitted, the user can choose the payment way; he can choose the e-payment method in which he must fill his credit card information. On the other hand, the user can choose the demand bill method in which he must fill his information in order to send the bill for him.

3.1.1.7 Accepting comments from the customers

The system allows customer to add there notifications or comments to the company site and any questions about products or delivery methods in the company, this gives the company a good feedback by which it can maintain a high level of customer satisfaction.

3.1.1.8 Display reports about orders, products and customer details

The system must provide the administrator with different kind of reports, like reports about orders, products and customer details report.

3.1.2 Requirement specifications

3.1.2.1 Provide information about the products through the website.

Software project/ibn alhaytham e-commerce /3.1.2.1

Function: Provide information.

Description: provide all information about product; features, prices and types through the web.

Inputs: exploring the website.

Source: database tables.

Outputs: information required.

Destination: web server that has the information database.

Requires: Internet Explorer installed on the client side.

Pre-condition: website account.

Post –condition: information must be correct and updated continuously.

Side effects: None.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.1

3.1.2.2 Making reservation for the products:

Software project/ibn alhaytham e-commerce /3.1.2.2

Function: Making reservation for the products.

Description: enable the user to make reservation of products through the reservation form on which all products features and prices are clearly specified

Inputs: product types and features.

Source: user choices.

Outputs: reservation form.

Destination: web server that has the information database.

Requires: filling the reservation form, checking quantities.

Pre-condition: none

Post –condition: none

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.2

3.1.2.3 The ability to rollback:

Software project/ibn alhaytham e-commerce /3.1.2.3

Function: The ability to rollback.

Description: the user must have the ability to rollback before submitting the reservation form to add or remove products.

Input: add new product or remove some from the reservation form or cancel the reservation.

Source: user choices.

Output: reservation form.

Destination: web server that has the information database.

Requires: change in the reservation form.

Pre-condition: filled reservation form.

Post-condition: none.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.3

3.1.2.4 Make updating on the database:

Software project/ibn alhaytham e-commerce /3.1.2.4

Function: Make changes on the database.

Description: the supervisor can update the database using authentication and authorization.

Input: administrator name, password.

Source: information of the supervisor from the database.

Output: updated database.

Destination: web server that has the information database.

Requires: administrator name and password.

Pre-condition: changes in the products and prices.

Post-condition: none.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.4

3.1.2.5 Allow customers to register in the system

Software project/ibn alhaytham e-commerce /3.1.2.5

Function: customer registration.

Description: The new customers can register to the system by filling a sing up form with his personal information.

Inputs: the customer must fill his personal information; user name, first name, last name, Email, password, telephone number and address.

Source: customer.

Output: complete registration process.

Requires: valid customer information.

Pre-condition: correctly fill the registration form, without using reserved user name or Email.

Post-condition: none.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.5

3.1.2.6 Support the payment by electronic way (credit card) or by demand bill:

Software project/ibn alhaytham e-commerce /3.1.2.6

Function: Support the payment by electronic way (credit card) or by demand bill.

Description: The system has to give the user the choices for payment method either by using demand bill or by credit card.

Inputs: the customer must fill his order information and according to the chosen payment method he must fill the required data.

Source: customer and bank system.

Output: the completeness of order reservation process.

Requires: valid credit card information or valid bill demand information.

Pre-condition: customer must be login, connection to bank system.

Post-condition: make sure that the payment for the product in the company account then order will be reserved.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.6

3.1.2.7 Accepting comments from the customers

Software project/ibn alhaytham e-commerce /3.1.2.7

Function: Support the feedback between customer and company

Description: The system has to give the customer the ability to submit his comments and questions.

Inputs: the customer can write his comments on product or the chosen payment method.

Source: customer.

Output: comments.

Requires: login to the system.

Pre-condition: the customer must be registered.

Post-condition: none.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.7

3.1.2.8 Display reports about orders, products and customer details

Software project/ibn alhaytham e-commerce /3.1.2.8

Function: system must provide administrator with required reports.

Description: The system has to give the administrator reports about customers, orders and products.

Inputs: the administrator must choose the report.

Source: the system database.

Output: reports.

Requires: the administrator must choose type of reports.

Pre-condition: the administrator must login.

Post-condition: none.

Side effects: none.

Definition: Software project/ibn alhaytham e-commerce /3.1.2.8

3.2 Non-functional requirements:

The non-functional requirements is responsible for the requirements of the user that does not concerned with the need of the software or any thing related with the functionality of the software program, like the requirement concerned with time, team properties and so on .A classification done for the non-functional requirements into three classifications (product requirements, process requirements and external requirements).

3.2.1 Product requirements:

1) Easy to access and login.
2) High powerful and speed display.
3) Easy to use.
4) The display of information must be clearly recognized and sorted in a good way.
5) The security on both the database and the payment method must be very high

3.2.2 Process requirement:

1) The software should be available at the end of this spring semester of the year (2003).
2) The system must be available to make registration process on the web.

3.2.3 External requirements:

1) Any user can visit the company web site.
2) The system should be able to adapt other big systems as a delivery and inventory system.

Chapter Four

System Analysis

4.1 Introduction:

This chapter analyzes the system (inputs, processes, outputs) and the surrounding environment. Also, it includes some description figures and general system structure.

4.2 System Contexts :(Relation with other systems)

The figure below (4.1) shows an overview of the system and its relationships with other systems. This diagram shows only the systems names without details about the relationships.

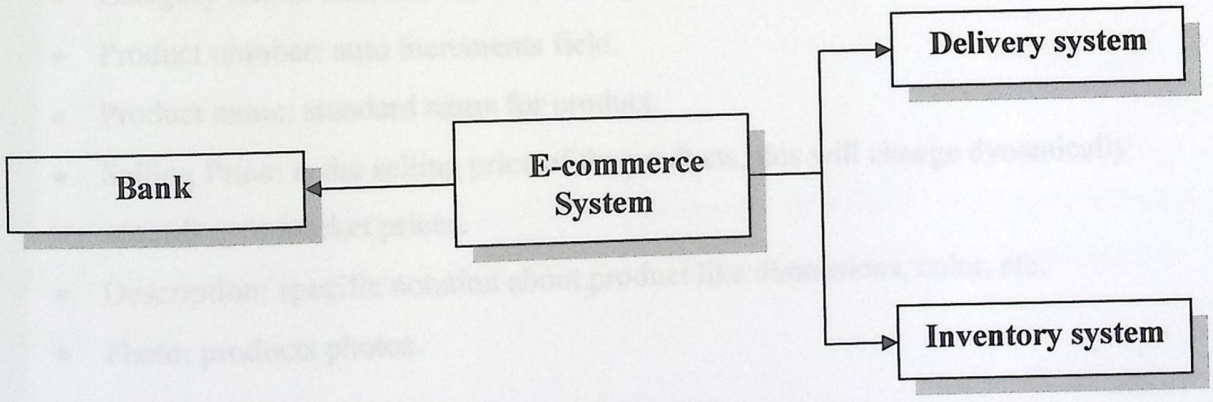


Figure (4.1) Context diagram

4.3 Input data

Input data is classified according to the source into two types:

4.3.1 Administration input:

In this process, the system allows authorized administrators to access the database for updating (add, delete, or modify products). This includes:

4.3.1.1 Product and category data

- Category number: This will be filed automatically (auto increments) by the system.
- Category name: standard name for category.
- Product number: auto increments field.
- Product name: standard name for product.
- Selling Price: is the selling price of the products, this will change dynamically according to market prices.
- Description: specific notation about product like dimensions, color, etc.
- Photo: products photos.

4.3.1.2 Authentications data

The administrator must have the ability to change his user name and password form the website to keep more security level in the system. Figure 4.2 shows the control model of administration input data.

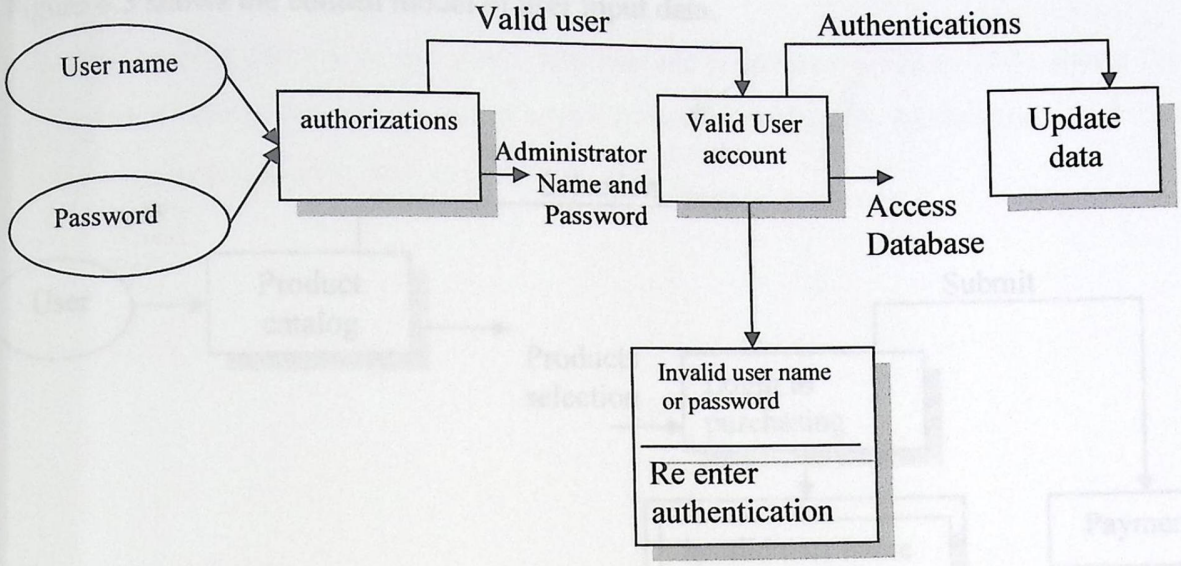


Figure (4.2) Control model for administration data

4.3.2 User inputs:

The user is allowed to browse the website and register in order to purchase products, after he/she complete the reservation form by selecting the products, the following data must be filled:

- User name
- First name.
- Last name.
- Password
- Email
- Phone
- Address

Figure 4.3 shows the control model of user input data.

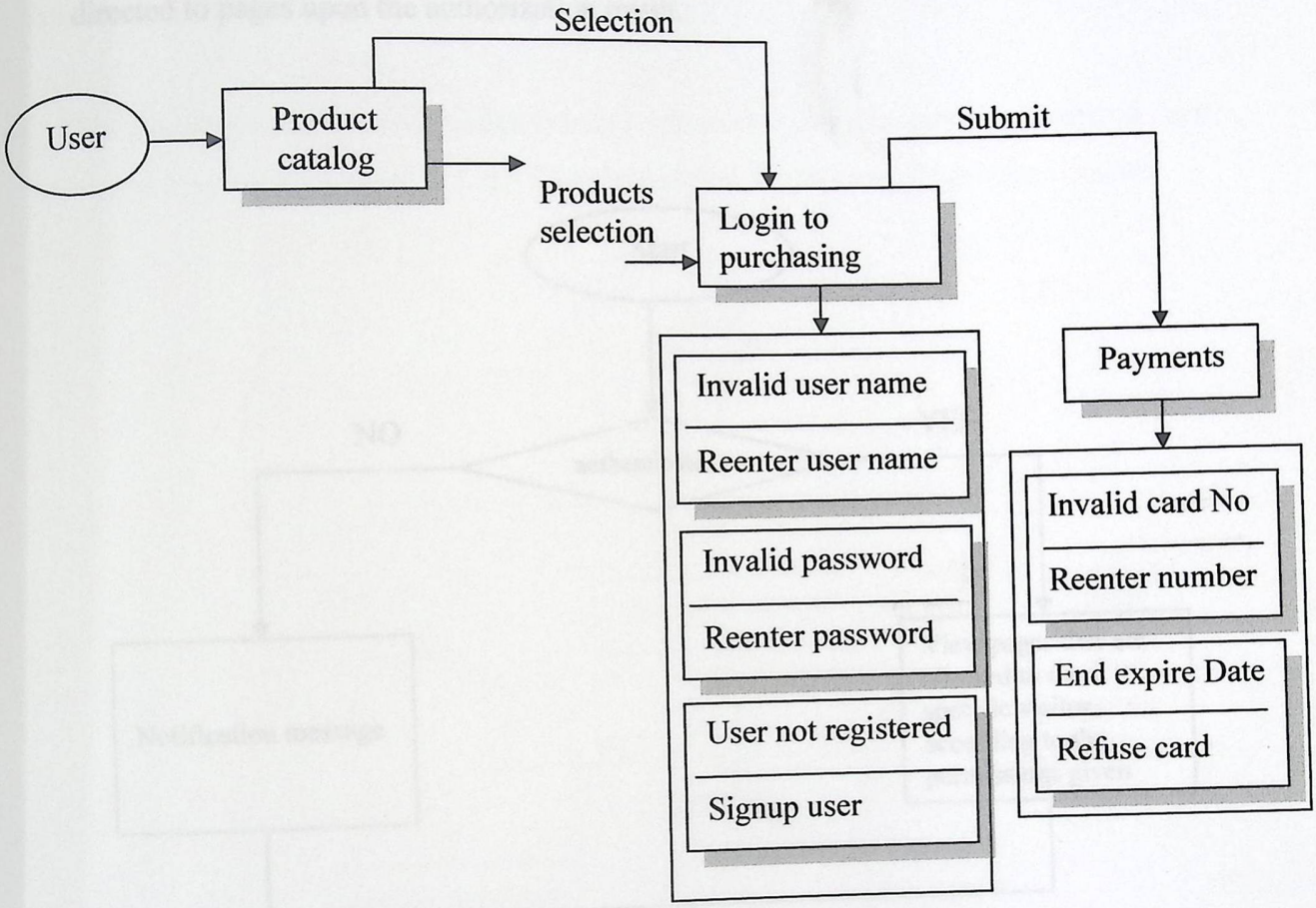


Figure (4.3) Control model of user input data

4.4 Processes:

The system process is classified into two main processes, Figure (4.4) shows the flowchart of the general process in which a user is checked for authorization, and then directed to pages upon the authorization result.

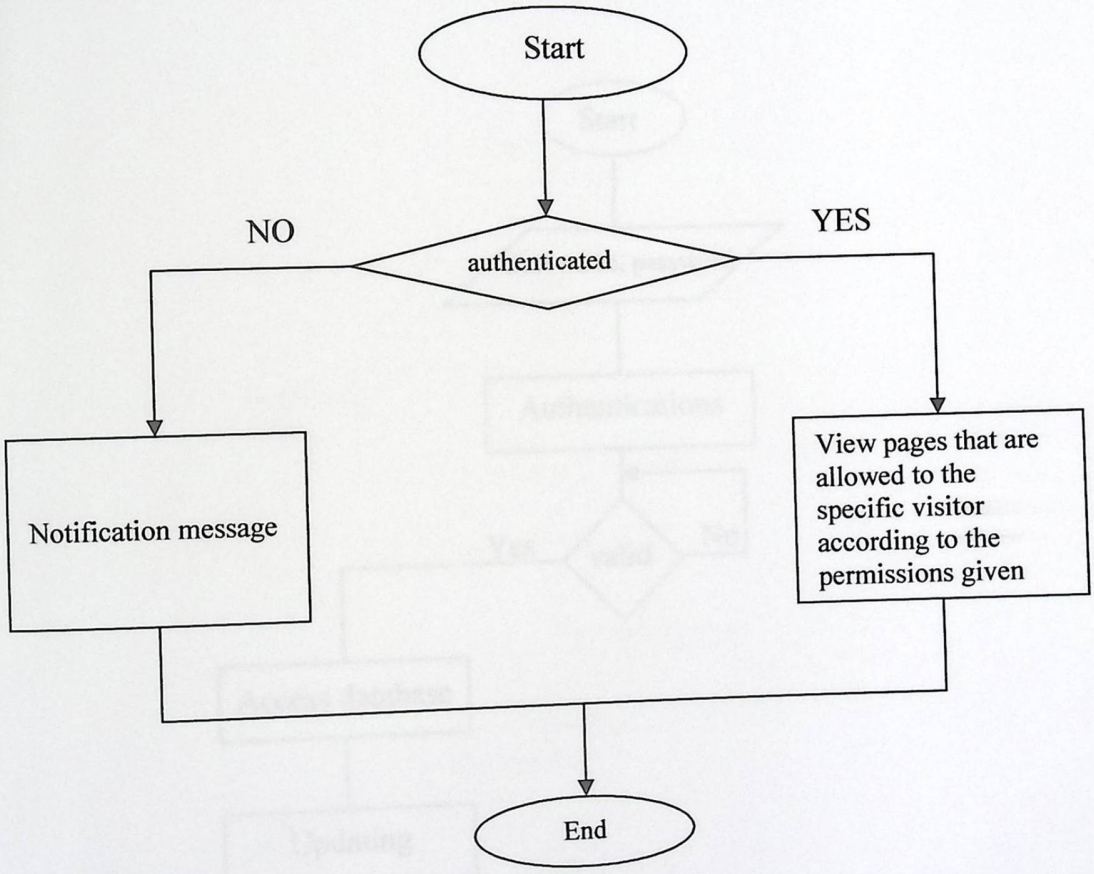


Figure (4.4) General Flowcharts

4.4.1 Updating database:

In this process, the administrator is allowed to make changes in the database, these changes includes updating the price of a product, add a new product, remove a product or modify its specifications.

This process is done under authentication where the administrator must enter a user name and password, in figure 4.5, the flowchart shows the process of updating database.

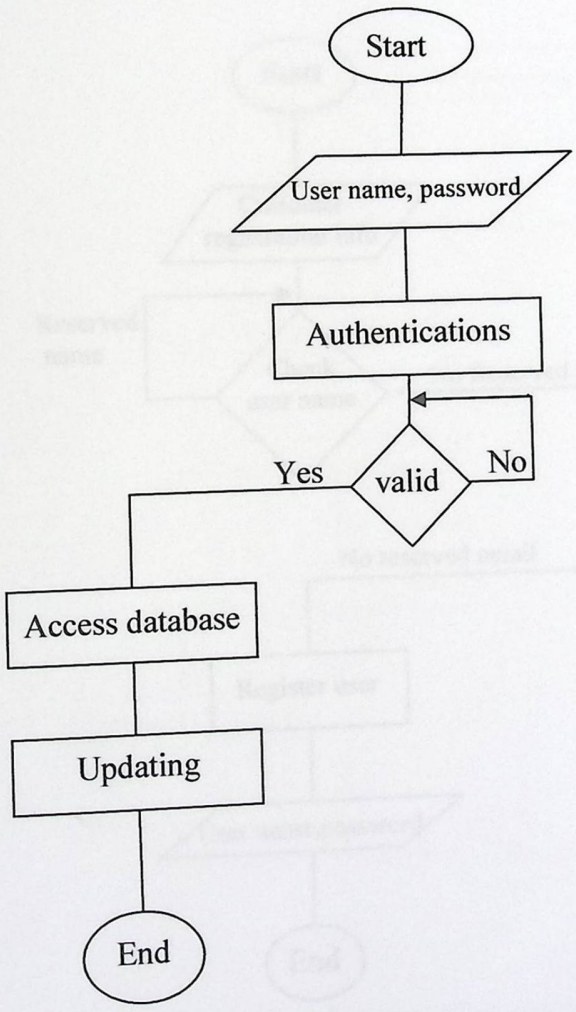


Figure (4.5) Flowchart for updating data base

4.4.2 Customer registration:

The system allows new customer to register in order to capture his information once, by applying this concept, the customer will avoid entering his information every time he wants to purchase a product, the registration process gives the customer user name and password to be used when he wants to login and also he has the ability to change his password. Figure (4.6) shows the flowchart of registration process.

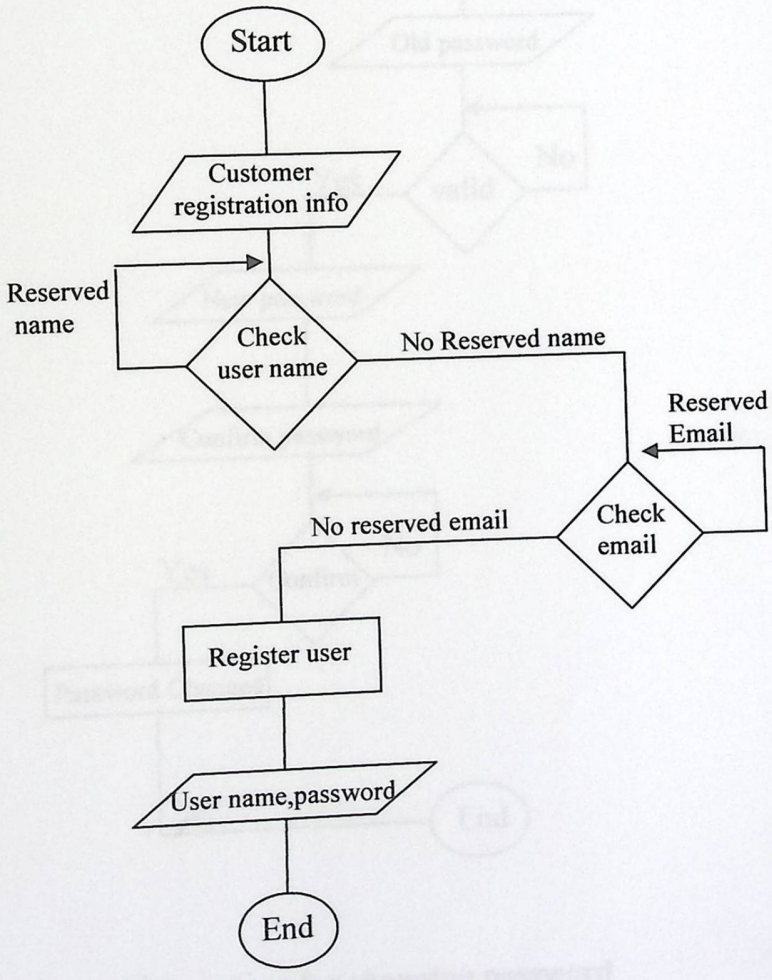


Figure (4.6) registration flowcharts

4.4.3 Changing password

The system gives the administrator and customer the ability to change there passwords Figure (4.7) shows the flowchart of changing password.

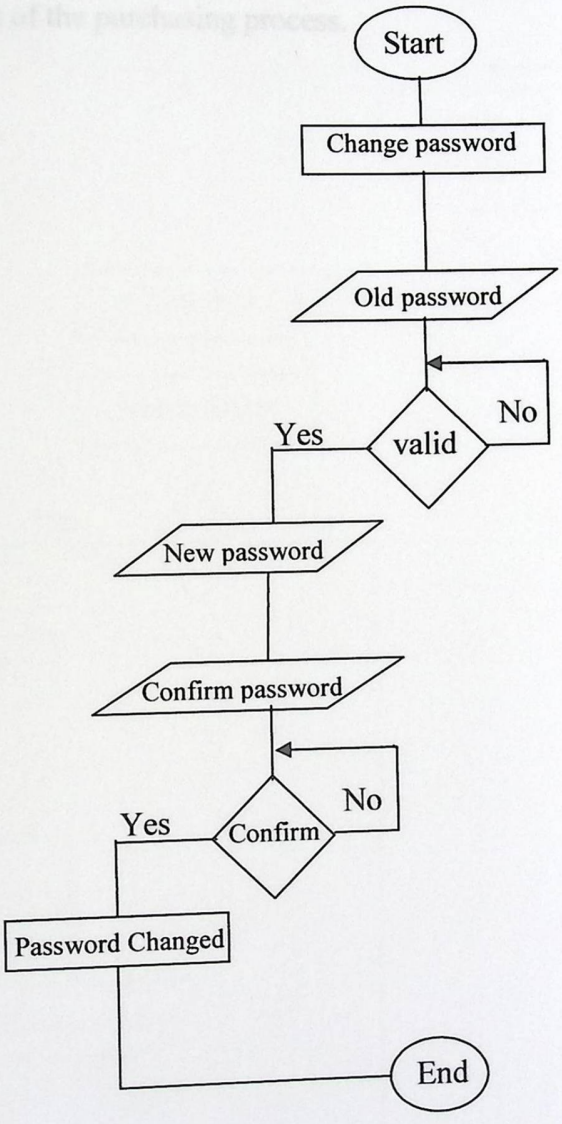


Figure 4.7 Flowchart for changing password

4.4.4 Purchasing

This process consists of three major steps: users are allowed to browse the website and choose the products to purchase, after that, when the user submits the chosen products, the reservation of products complete. Finally, the user must choose the payment way. Figure 4.8 shows flowchart of the purchasing process.

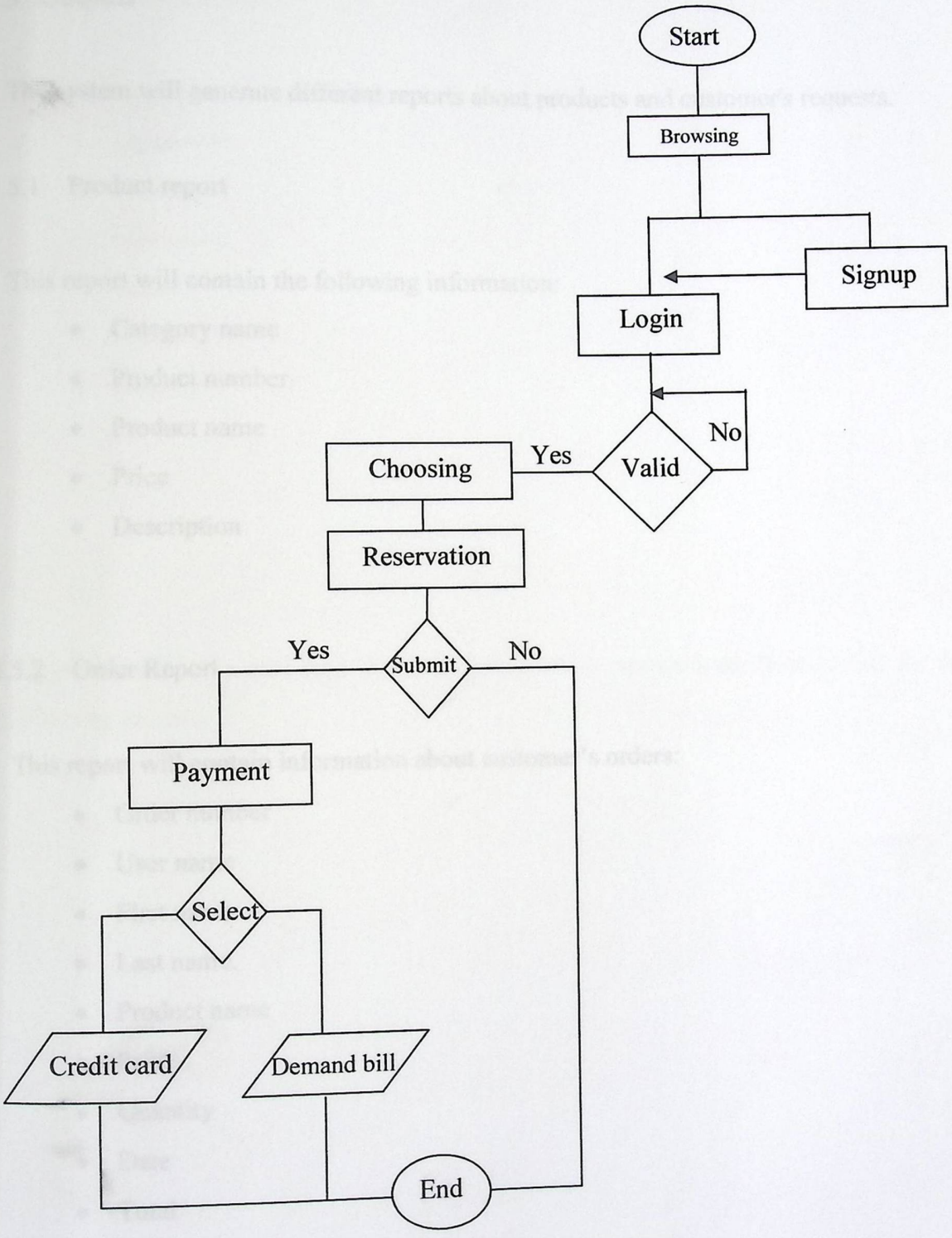


Figure 4.8 Purchasing flowcharts

4.5 Outputs

The system will generate different reports about products and customer's requests.

4.5.1 Product report

This report will contain the following information:

- Category name
- Product number
- Product name
- Price
- Description

4.5.2 Order Report

This report will contain information about customer's orders:

- Order number
- User name
- First name
- Last name.
- Product name
- Price
- Quantity
- Date
- Total
- Status Details

4.5.3 Customer Report

This report will contain information about customers

- User name
- First name
- Last name
- Phone
- Email
- Address

Figure 4.9 show the data flow diagram (DFD) which represent the flow of data for the purchasing process.

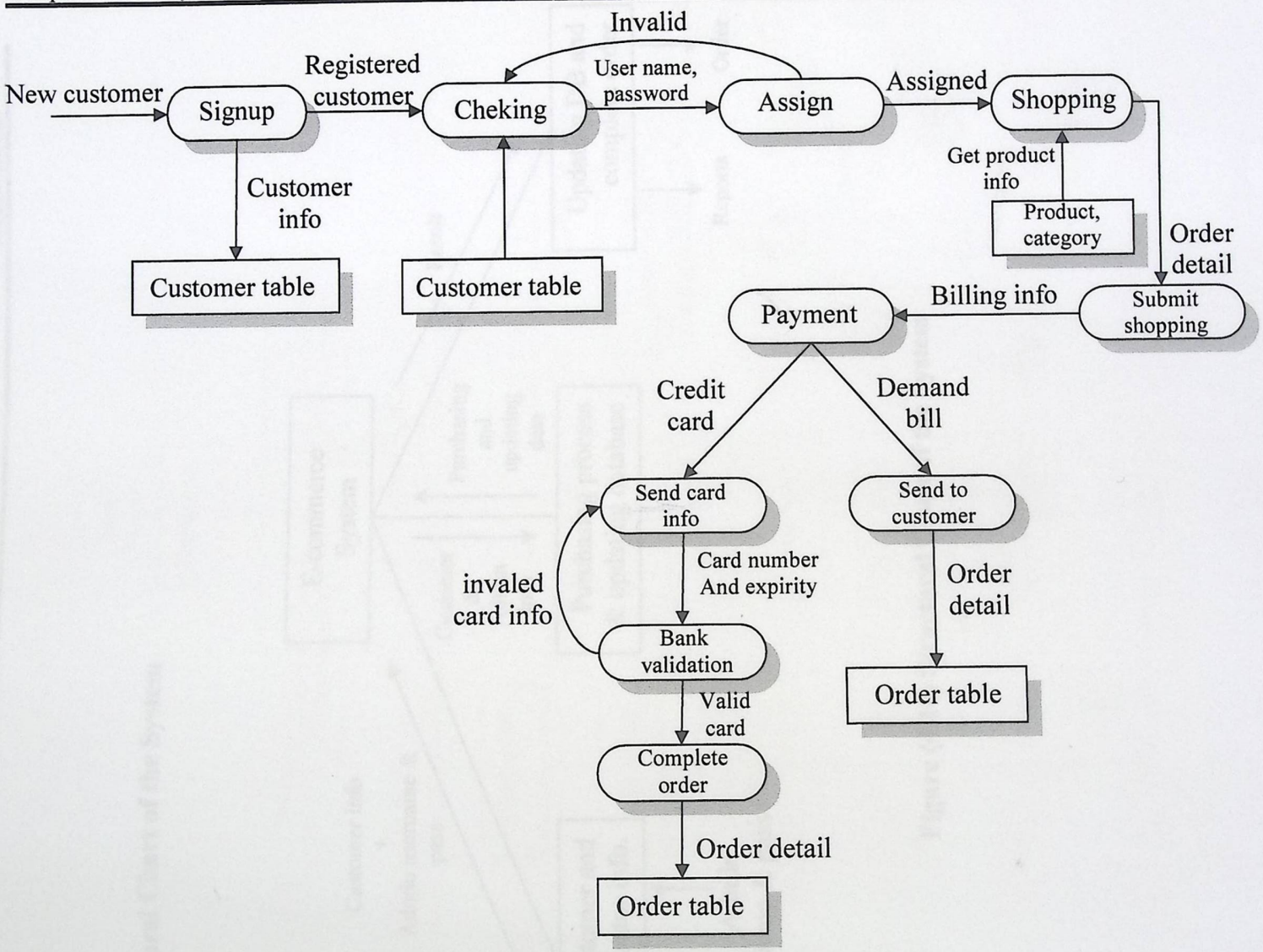


Figure (4.9) Data flow for purchasing process

Structural Chart of the System

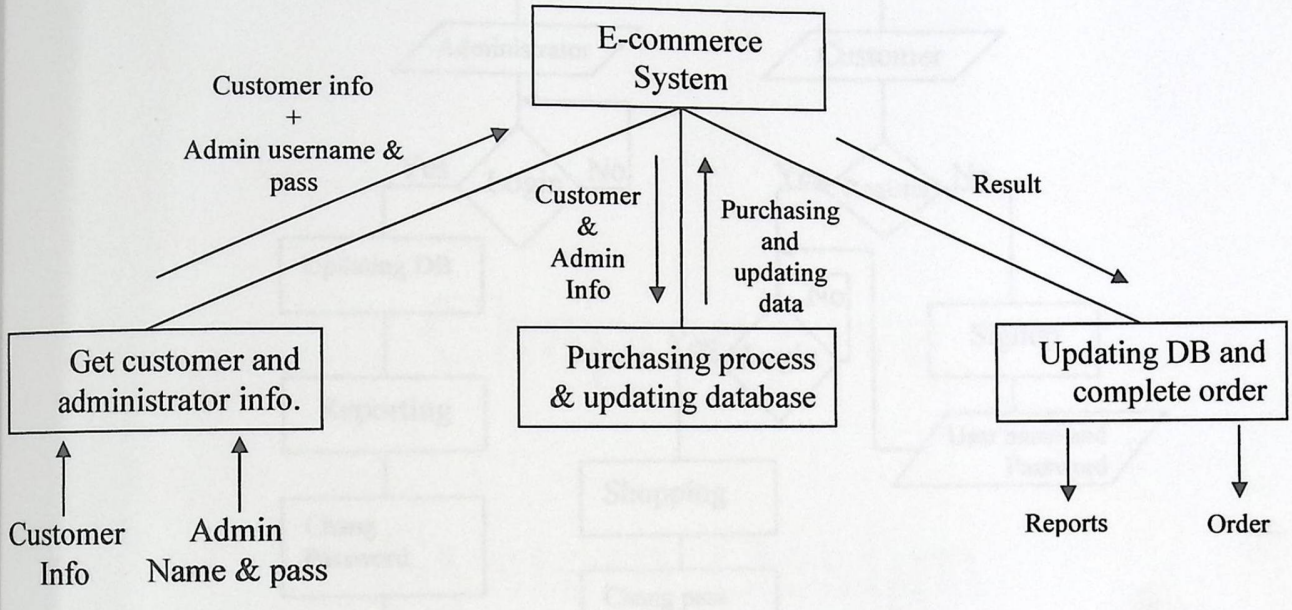


Figure (4.10) Structural Chart of the System

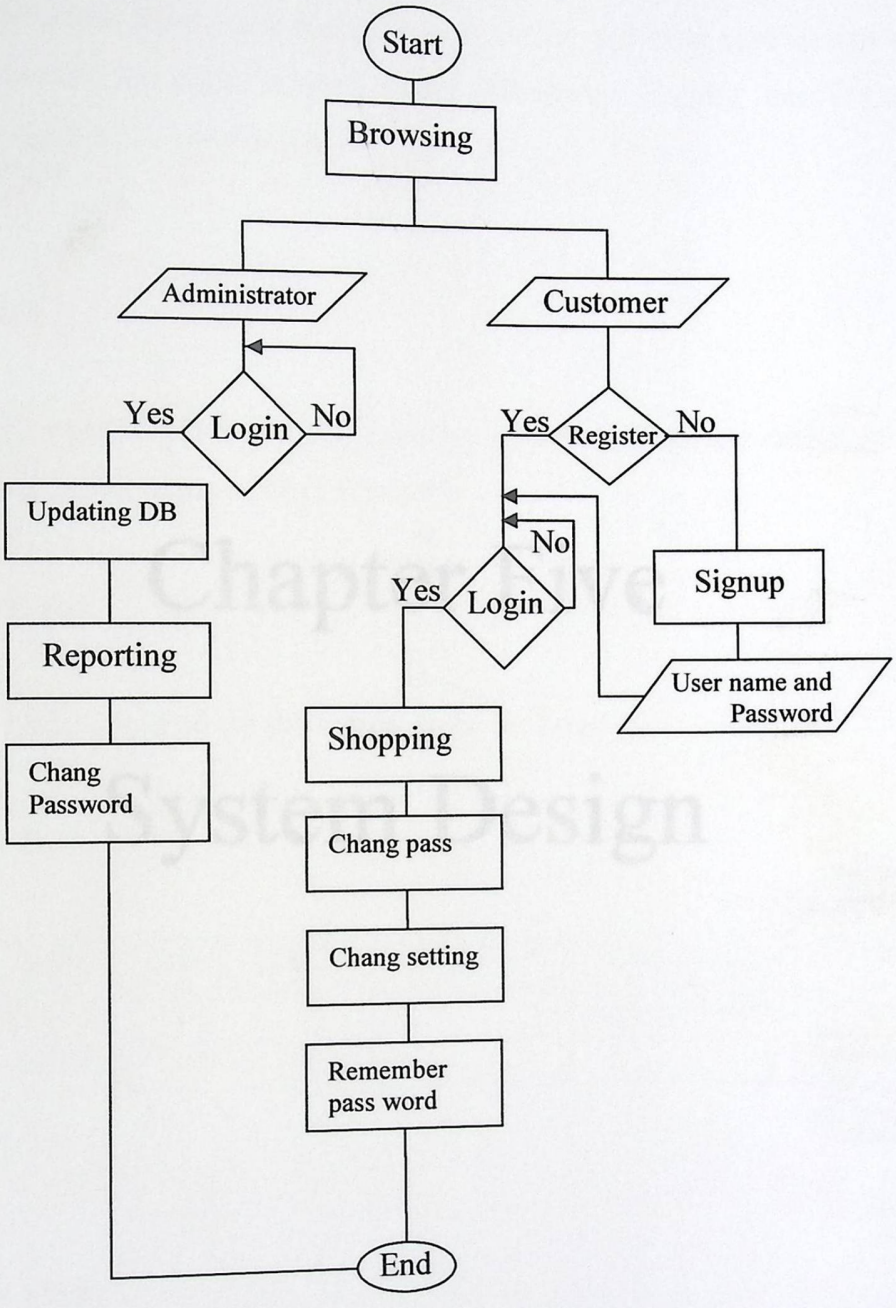


Figure (4.11) System Flowcharts

Chapter Five

System Design

5.1 Introduction

This chapter describes the system design, user interface and Data base design which include data dictionary and entity relation model (ER model). Finally, interface design (input, output design) will be explained.

5.2 Database design

Database design describes the physical structure of tables. Database design includes data dictionary and presentation of the ER model.

5.2.1 Data dictionary

Below are the tables required for the system to be designed

1. Category Table

Field	Type	Null	Key	References	Extra	Description
category_no	int(10)	NO	PK		auto_increment	Category Number
category_name	varchar(50)	NO				Category Name

2. Products Table

Field	Type	Null	Key	References	Extra	Description
product_no	int(10)	NO	PK		auto_increment	Product Number
Category_no	int(10)	NO	PK , FK	Category		Category Number
product_name	varchar(50)	NO				Product Name
description	varchar(50)	YES				Description
Price	float(10,0)	NO				Selling Price
Photo	mediumblob	NO				Photo

3. Orders Table

Field	Type	Null	Key	References	Extra	Description
order_no	int(10)	NO	PK		auto_increment	Order Number
Username	varchar(50)	NO	FK	Customers		User Name
Comments	varchar(100)	YES				Comments
status_details	varchar(50)	NO				Status Details
Date	Date	NO				Date

4. Order Item Table

Field	Type	Null	Key	References	Extra	Description
order_no	int(10)	NO	PK, FK	Orders		Order Number
Category_no	int(10)	NO	PK, FK	Products		Product Number
product_no	int(10)	NO	PK, FK	Category		Category Number
Price	float(10,0)	NO				Purchase Price
Qty	int(10)	NO				Quantity

5. Customers Table

Field	Type	Null	Key	References	Extra	Description
Username	varchar(50)	NO	PK			User Name
Firstname	varchar(50)	NO				First Name
Lastname	varchar(50)	NO				Last Name
Email	varchar(50)	NO				E-mail
User_password	varchar(50)	NO				User Password
Phone	text	YES				User Phone
Address	varchar(100)	YES				User Address

5.2.2 Entity relation model

Figure (5.1) represents the E-R model that describes the relationship between tables.

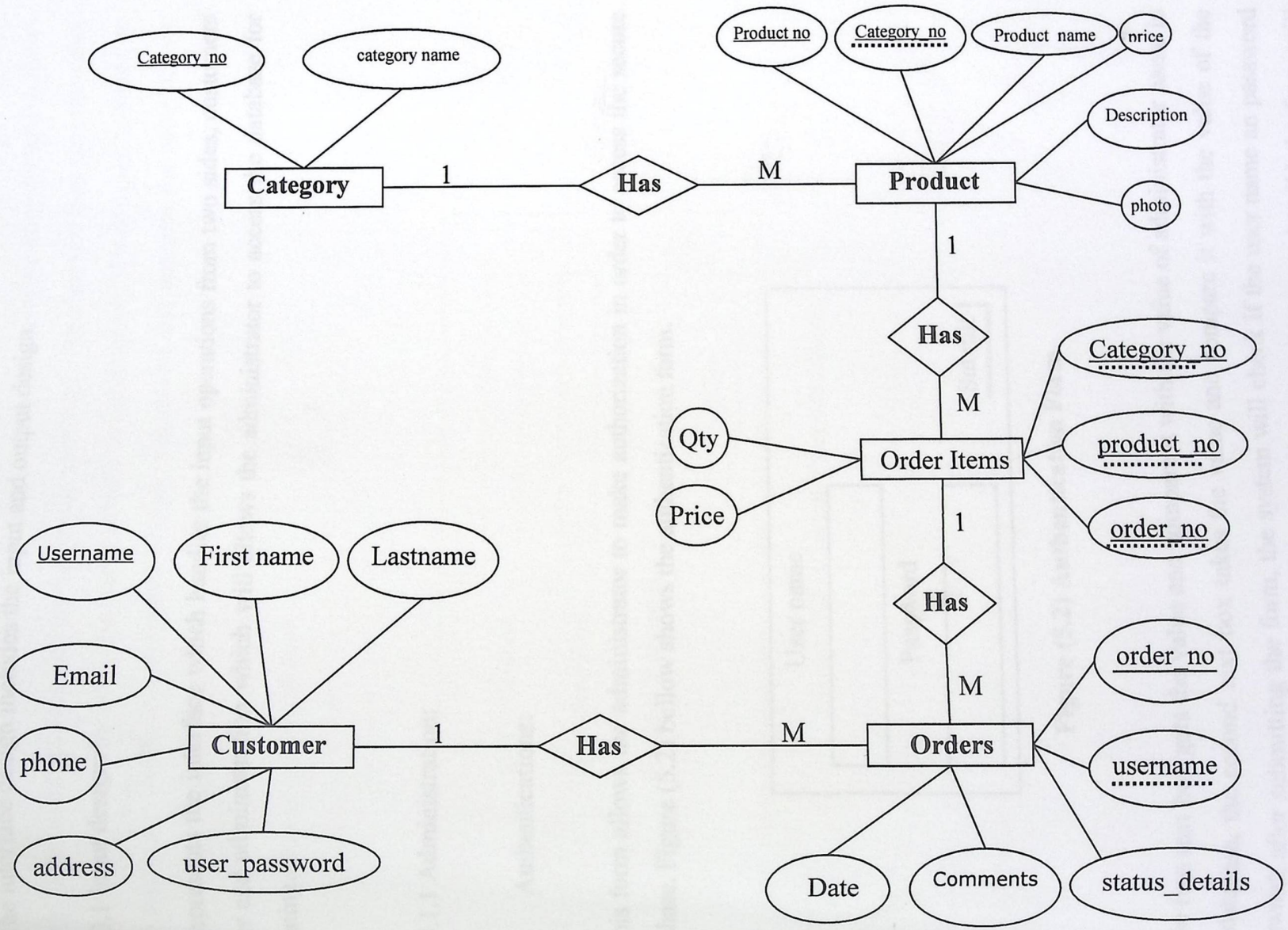


Figure (5.1) ER model

5.3 Interface design

The interface design includes the input and output design.

5.3.1 Input design:

Represents the interface which handles the input operations from two sides, customers order and administration which will allows the administrator to access the database for updating.

5.3.1.1 Administration:

- Authentications:

This form allows the administrator to make authorization in order to access the secure database. Figure (5.2) bellow shows the authentication form.

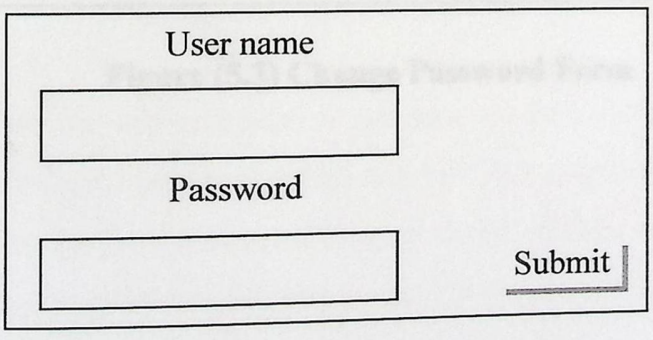


Figure (5.2) Authentication Form

The first text box gets the value and compare it with the value of administrator name in the database, the second text box takes the value and compare it with the value of the password, after submitting the form, the system will check if the user name an password are valid, then the administrator can proceed, if these values are invalid the form will enable him to login again.

- Change password form:

This form allows the administrator to change authorization information by having direct access to the database in order to change the password. The first text box gets the value and compare it with the administrator password in the database, if it dose not matched an error message will ask him to reenter it again, the second text box gets the value of new password and compare it with the value in the third text box, if the old password matches and the new password confirmed, then the password will change, figure (5.3) bellow shows the change password form.

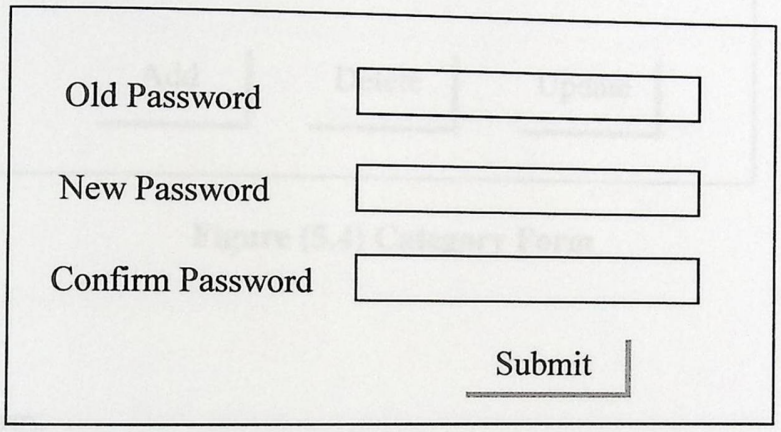


Figure (5.3) Change Password Form

- Category form:

This form allows the administrator to add new product categories; the first box is an auto increment field which will be filled automatically by the system, the second text box (Category name) will accept the new category name and add it to category table in the database. Figure (5.4) bellow shows the category form.

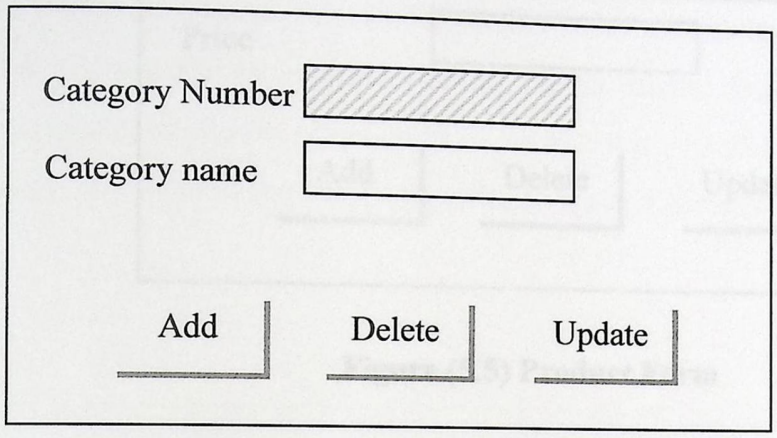


Figure (5.4) Category Form

- Product form:

This form allows the administrator to add new product in the products table, the first text box is an auto increment field which will be filled automatically, the second text box will accept the new product name and insert it in the product table. The category number list box is a foreign key referencing the category number in category table, the description field is optional.

The three updating buttons allow the administrator to up date the product table. Figure (5.5) bellow shows the product form.

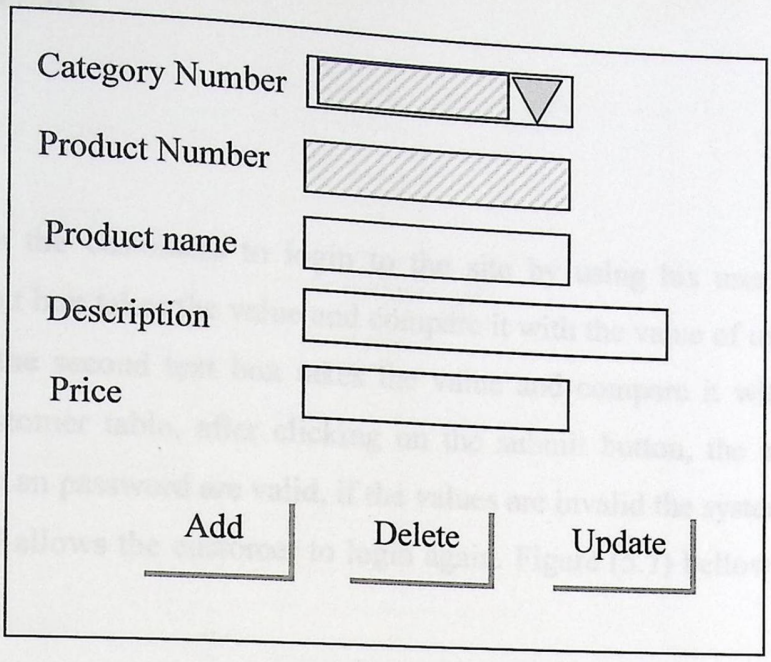


Figure (5.5) Product Form

- Updating status details form:

The Status details filed allows the administrator to fix that certain orders have been paid. Figure (5.6) shows the Updating status details form.

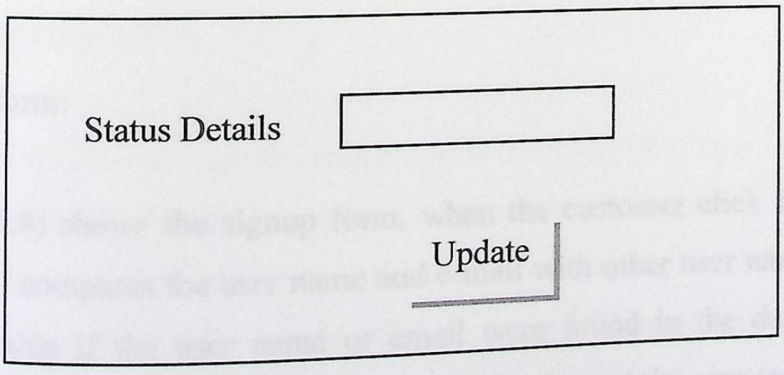


Figure (5.6) Updating status details Form

5.3.1.2 Customer interface

- Login form

This form allows the customers to login to the site by using his user name and password , the first text box takes the value and compare it with the value of user name in the customers table, the second text box takes the value and compare it with value of password filed in customer table, after clicking on the submit button, the system will check if the user name an password are valid, if the values are invalid the system will give an error message, and allows the customer to login again. Figure (5.7) bellow shows the login form.

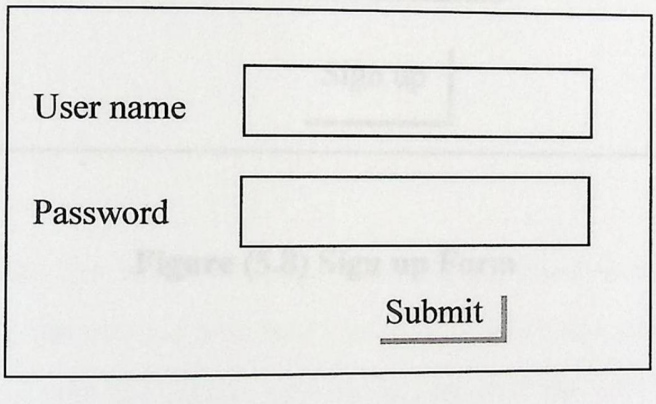


Figure (5.7) Login Form

- Sign up form:

Figure (5.8) shows the signup form, when the customer click the sign up button the system will compares the user name and e-mail with other user names and e-mails in the customer table if the user name or email were found in the database it will show a message that the user name or e-mail are reserved , and the user must chose another user name or email , if there is no duplication in user name or e-mail it will register and the system will show message that the user have successfully register and it will show the new user his user name and password, the figure (5.8) bellow shows the sign up form.

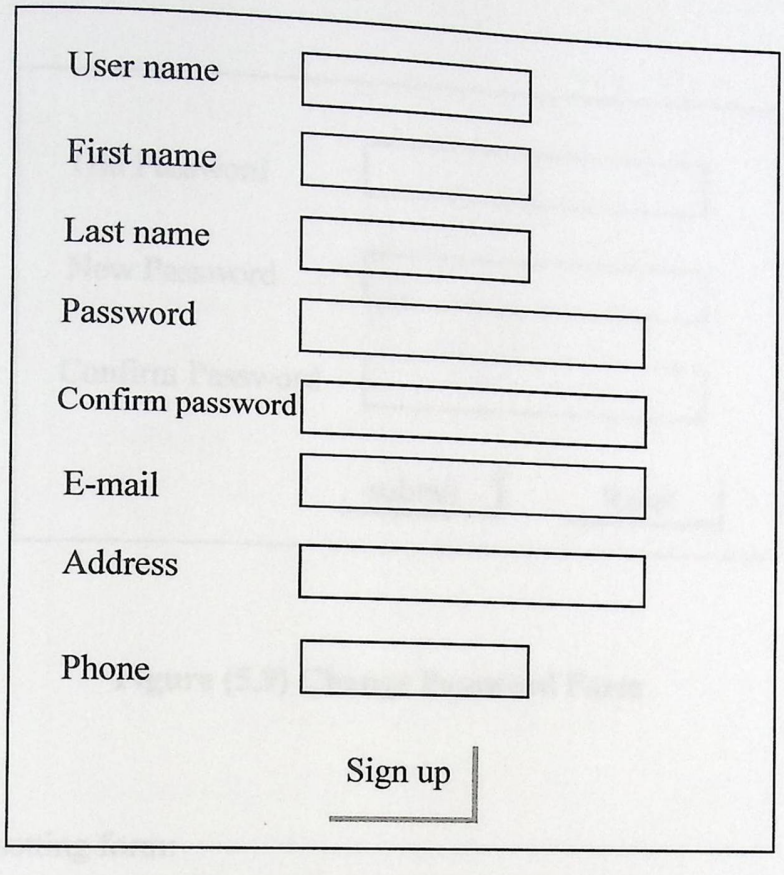


Figure (5.8) Sign up Form

- Change password form:

This form allows the user to change authorization information by having direct access to the database. The first text box takes the value and compare it with the user password in customers table, if it doesn't match, an error message will ask him to reenter it again, the second text box takes the value of new password and compares it with value in the third text box, if the old password matches and the new password confirmed then the password will be changed. Figure (5.9) bellow shows the change password form.

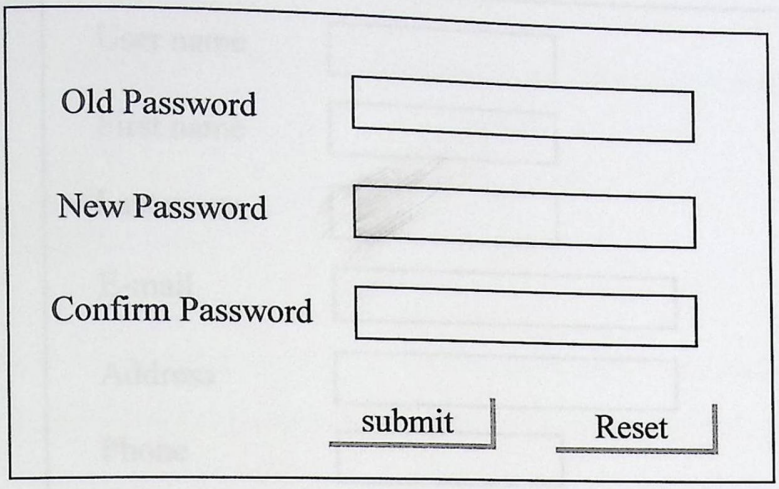


Figure (5.9) Change Password Form

- Change setting form:

This form allows the user to change his settings, the first text box (email) takes the new email value, the second text box (address) takes the new address value , the third text box (phone) take the phone new value , by clicking the (change settings) button it will compare the e-mail with other e-mail in the database, if the email is found in the database, it will show a message that the e-mail address is reserved , and the user must choose another email, if there is no duplication in the e-mail address it will change settings and the system will show message that the user have successfully change settings and it will show the new email , new address and new phone. Figure (5.10) shows the change settings form.

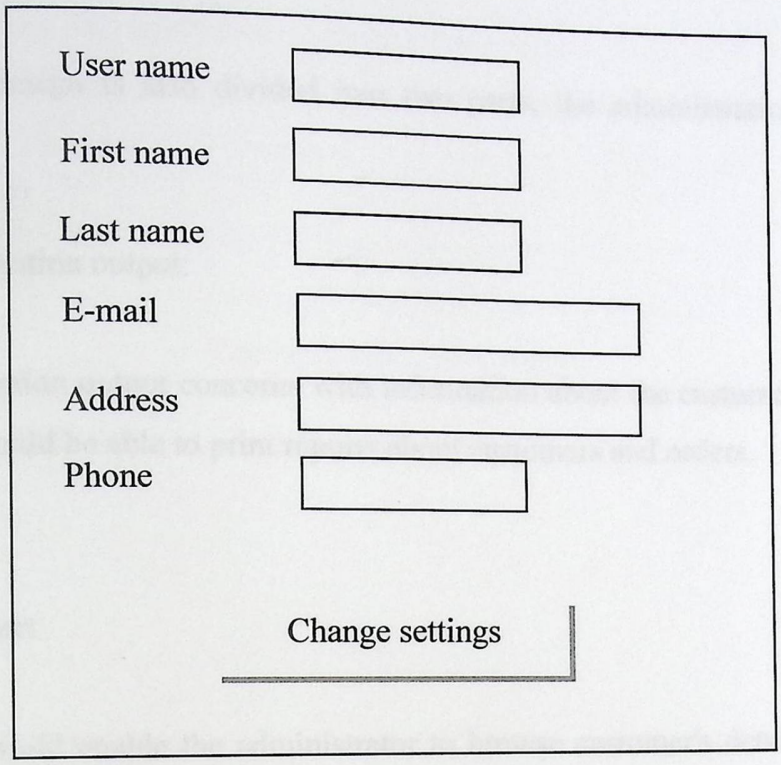


Figure (5.10) Change setting Form

- Forget password form:

Figure (5.11) shows the forget password form, by clicking on the submit button, the system will take the value of the email and then the system will inform the administrator, and then the administrator will send an email to the user containing his password.

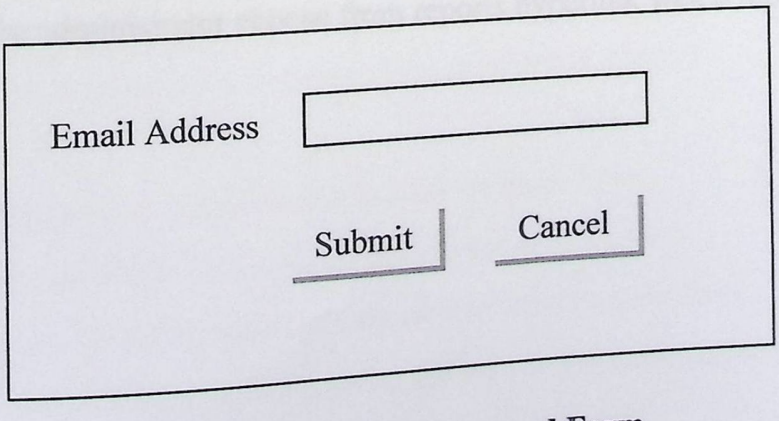


Figure (5.11) Forget Password Form

5.3.2 Output design

The output design is also divided into two parts, the administration part and the customer part.

5.3.2.1 Administration output:

The administration output concerns with information about the customer and orders, in administrator should be able to print reports about customers and orders.

- Customers report

The system should enable the administrator to browse customer's details, these details comes from the customers table and contains the following information:

User name	First name	Last name	Email	Phone	Address

Figure (5.12) Customer report

The administrator an access this report after login by using administrator name and password, then the administrator choose from reports hyperlink which reports he want to see.

- Orders report

This report allows the administrator to see the orders information, this information comes from orders table. This report contains the following information:

Order number	User name	Comments	Amount	Status detail	date

Figure (5.13) Order report

The user should be able to print the generated report.

- Product report

This report provide the administrator with information about the available products, this information comes from the products table.

Category name	Product number	Product name	Price	Description

Figure (5.14) Product report

5.3.2.1 Customers output

- The main product page:

This page allows the customer to proceed the purchasing steps:

- Login hyperlink: connects the user to the login form.
- Signup hyperlink: connects the user to the sign up form.
- Shopping hyperlink: connects the user to the shopping form.
- Shopping cart hyperlink: connects the user to the shopping cart.

Figure (5.15) bellow shows the main product page.

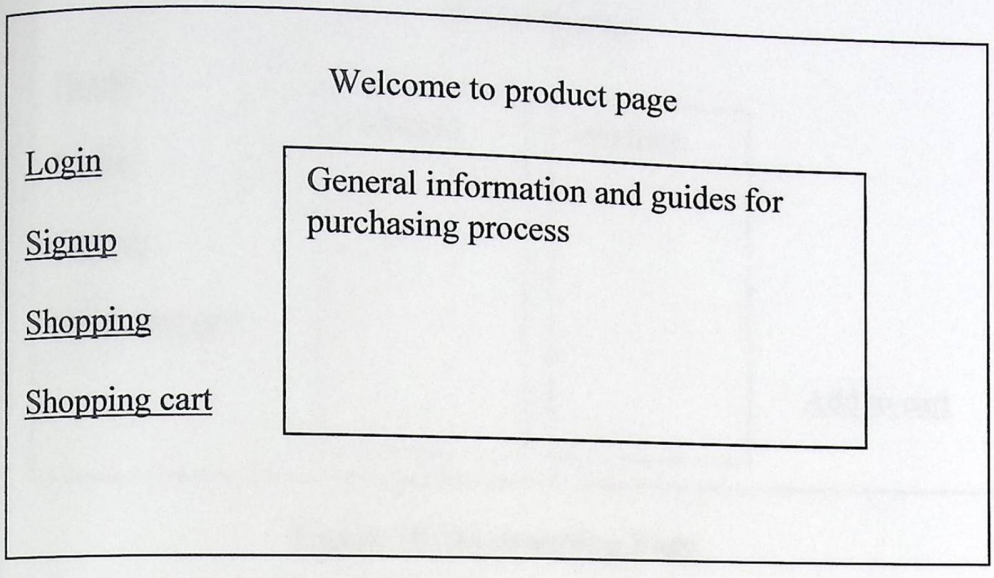


Figure (5.15) Main Product Page

- Shopping page:

This page shows two lists which are product categories and the available products from that category, when the customer selects the category item, the product list of this category will be displayed, when the customer selects a product, full details about the product information is displayed. When the customer wants to purchase a product, he can simply add it to this shopping cart by clicking (add to cart) hyper link. Figure (5.16) shows the shopping page.

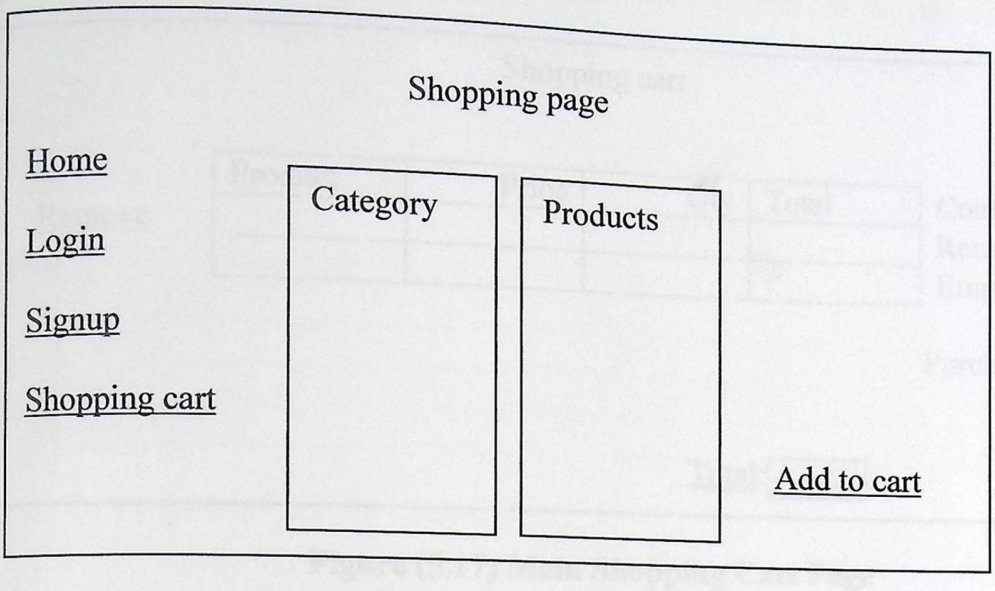


Figure (5.16) shopping Page

- Shopping cart page:

This page contains all the chosen products by the customer with their prices, quantity and the total price for the order, it also contains four links:

- Continue Shopping: this link enables the user to go back to the shopping page in order to select more products.
- Recalculate cart: the customer can change the product quantity and by clicking this link, the total price should be recalculated.
- Empty the cart: empties the shopping cart.
- Purchase now: this link connects the user to the payment method. Figure (5.17) bellow shows the shopping cart page.

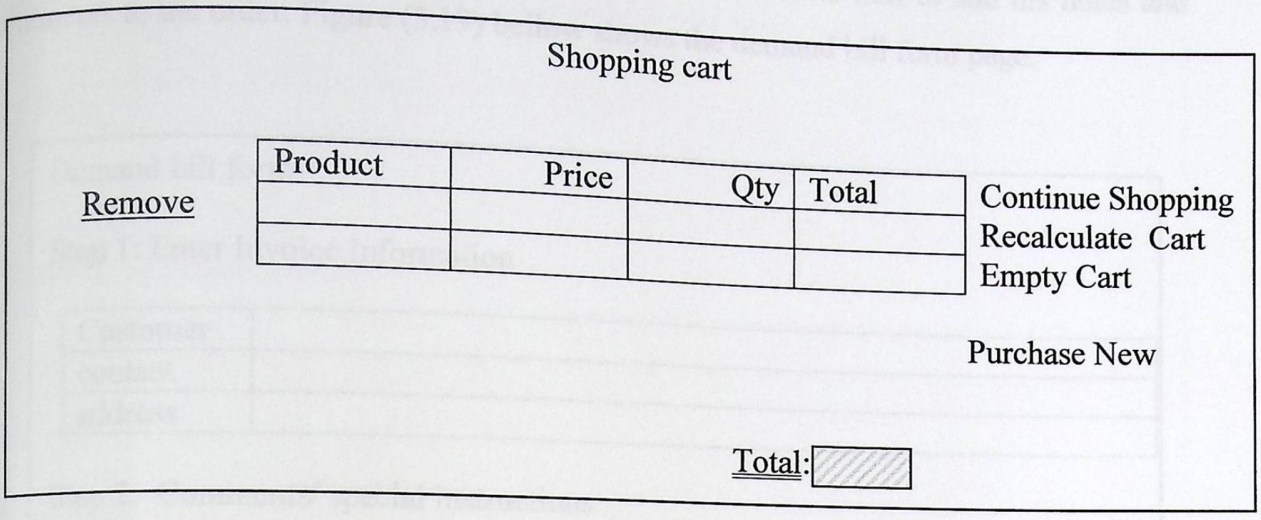


Figure (5.17) Main Shopping Cart Page

- Payment method page:

This page allows the customer to choose the payment method on which the user can chose one of the two payment methods (credit card or demand bill). Figure (5.18) bellow shows the payment method page.

When the customer chooses the appropriate payment method, the customer login form is displayed (see figure (5.7)), this operation facilitate the purchasing process by getting all the user information from the system database which avoid the user to insert his personal information every time he want to purchase on line.

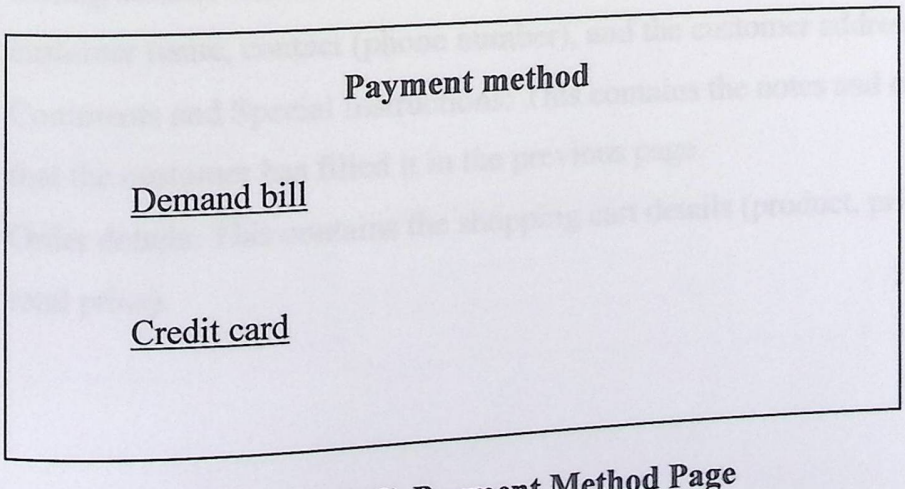


Figure (5.18) Payment Method Page

When the user chooses the demand bill payment method and login, then the demand bill information will be filed from the customers table and allows him to add his notes and comments to the order. Figure (5.19) bellow shows the demand bill form page.

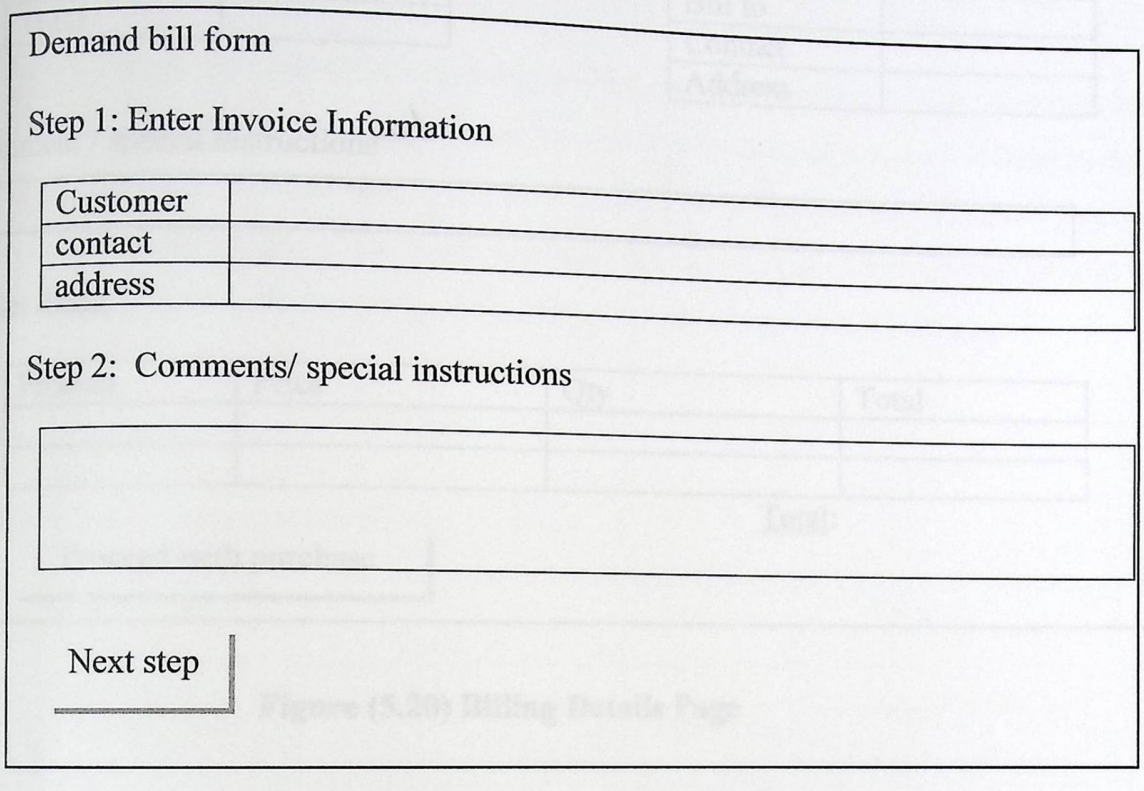


Figure (5.19) Demand Bill Form Page

When the customer clicks on the (Next step) button the information will be saved to the database and the billing confirmation page will appear which contains three parts:

- Billing details: which contains the date of purchase, total price of the order, customer name, contact (phone number), and the customer address.
- Comments and Special Instructions: This contains the notes and comments that the customer has filled it in the previous page.
- Order details: This contains the shopping cart details (product, price, quantity, total price).

Figure (5.20) bellow shows the billing details page

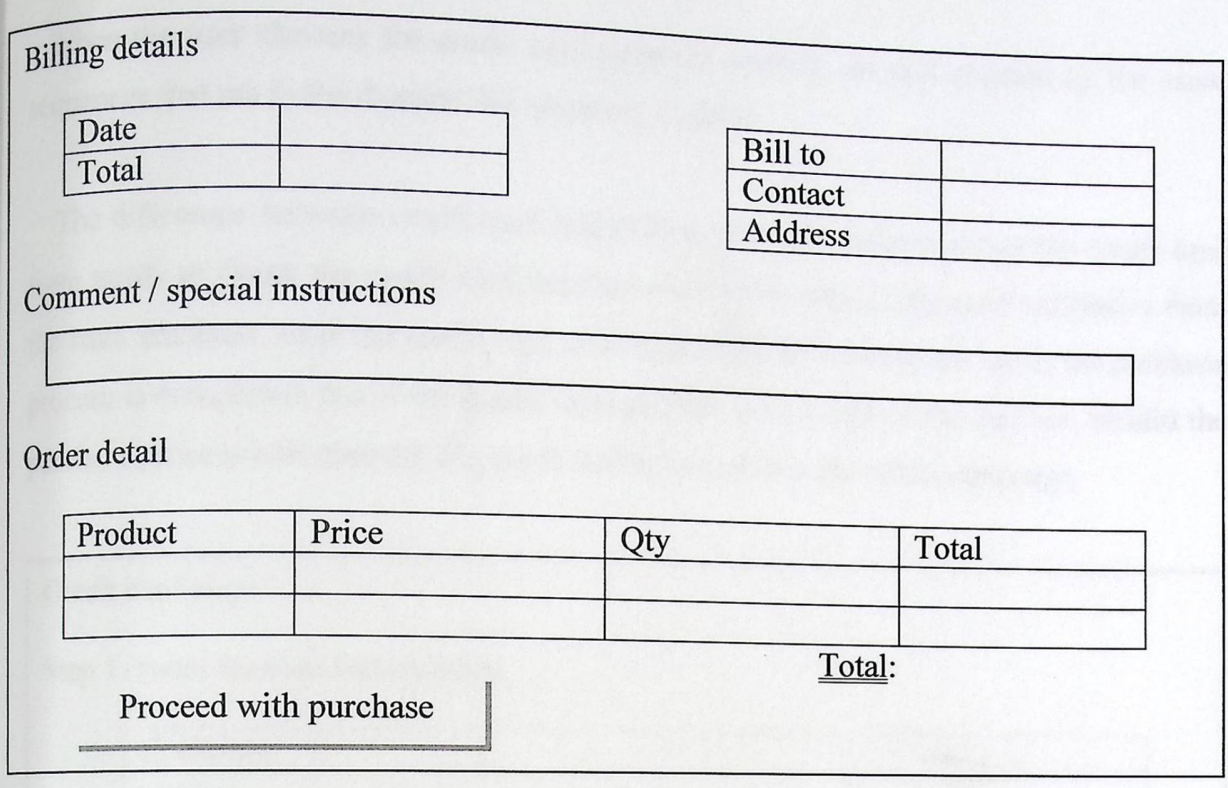


Figure (5.20) Billing Details Page

When the user presses the button (proceed with invoice) all the information of the order will be inserted into the database, then the system will go to the final page which shows that the customer has successfully complete the purchasing process and it will show also the order number and the total price of his order. Figure (5.21) bellow shows the completed order page.

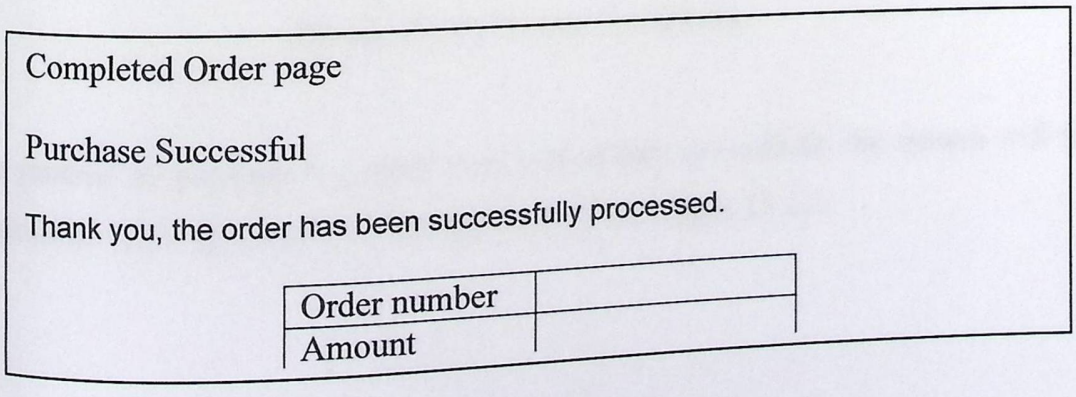


Figure (5.21) Completed Order Page

- Credit card

When the user chooses the credit card payment method, he will proceed by the same sequences that are in the demand bill payment method.

The difference between credit card and billing payment method is that the credit card form needs to check the credit card number and expire date of the card validation from the bank database, so if the credit card and expire date of the card are valid, the purchase process is completed, but if the credit card and the expire date of the card are invalid the purchase process will aborted. Figure (5.22) bellow shows the credit card page.

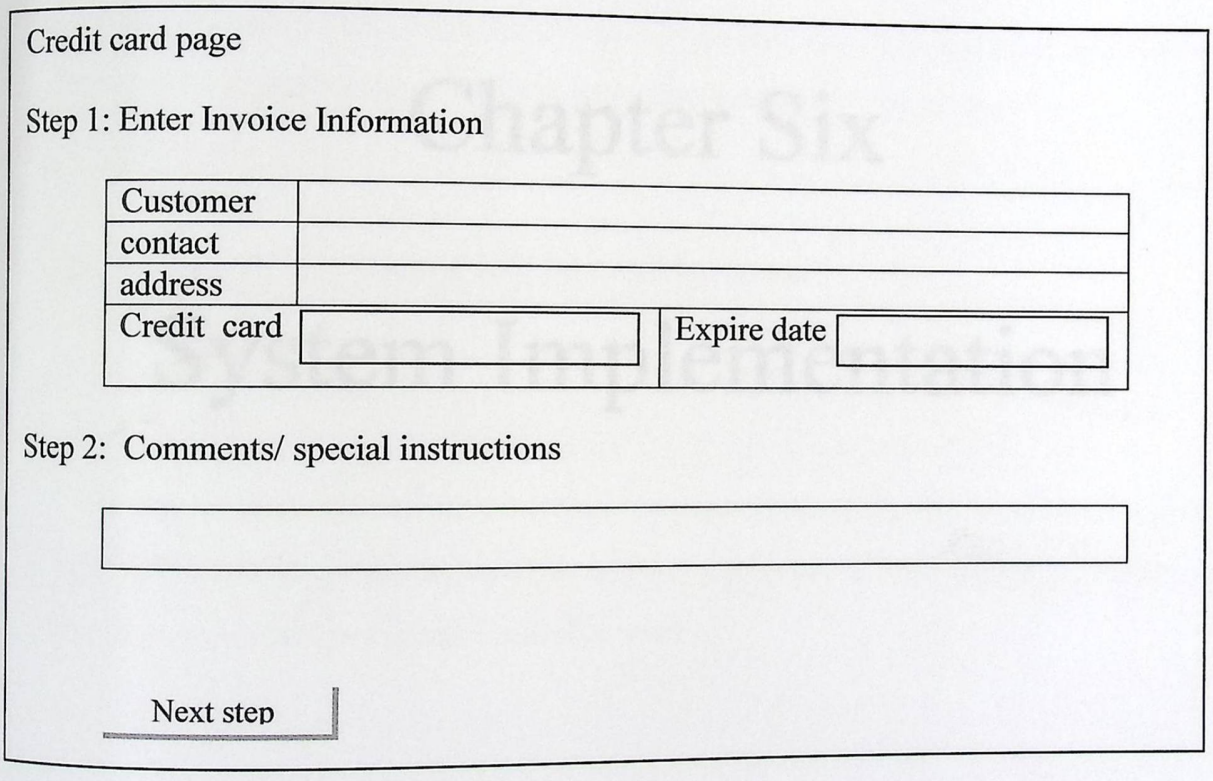


Figure (5.22) Credit Card Page

If the process of payment by credit card completed successfully the system will inform the customer by congratulation message as shown in figure (5.21).

Chapter Six

System Implementation

6.1 Introduction:

The internet is growing fast and the number of its users is increasing rapidly day after day, it becomes the most common, simple, and fast way of communication.

As a result, many software packages are available now for designing internet web sites, companies compete to produce packages for internet programmers that allows them to design, manage, present data in a simple, dynamic ways. Also many tools are available now that allows the users to manage there information, database remotely.

In this chapter system implementation is presented, the steps that must be followed to implement the functions that have been discussed before are presented.

6.2 Setting up the required software and hardware

6.2.1 Network setting:

The system must operate on a network consists of a server, client, cables (Twisted Pair RJ45) and software required to operate the network and system as shown in chapter two (system planning/ feasibility study).

6.2.2 Setting hardware and operating system:

The operating system required for the server side is Microsoft windows 2000 server, and on the client side Microsoft windows 2000 professional. Windows 2000 server is a powerful operating system. It supports many features needed in this project. Hardware requirements are three PC's as shown in chapter tow (feasibility study).

6.2.3 Installing PHP 4.3.1 and MYSQL:

Building the database for the system by using MYSQL which is a database management system, PHP (stander for Hyper Text Processor), which installed to connect the database with website (for more details see appendix).

PHP is a server side and it is a standard for hyper text processor, it is the open source alternative to ASP that runs on multiple operating systems, including Linux and Windows. There are six reasons that make PHP better than ASP:

1) Speed

ASP is built on COM-based architecture, when an ASP programmer uses VBScript; he is running a COM object. When he writes to the client, he's calling the Response COM object's Write method. When he accesses a database, he uses another COM object to do so. When he accesses the file system, another COM object is called. All this COM overhead adds up and slows things down.

In PHP modules, everything runs in PHP's memory space. This means that PHP code will run faster because there is no overhead of communicating with different COM objects in different processes. Here is a research study done by expert to show the speed of PHP and ASP:

The research team where execute a Select statement 40 times on Microsoft SQL Server 7 using PHP's MSSQL7 extension, PHP's ODBC extension and COM , the results where as follows :

PHP Querying MSSQL7	**Seconds** (lower is better)
Using MSSQL extension	01.88
Using ODBC extension	09.54
Using ODBC via COM (ADO)	17.28
Using OLEDB via COM	06.19

Figure (6.1) PHP speed

When the research team access the database using PHP's ODBC extension, it's 9.54 secs. Using the COM interface to connect to ODBC adds a 80% overhead (17.28 secs) to ODBC.

OLEDB is Microsoft's high speed COM technology for accessing databases. It is faster than ODBC, but when the research team uses the PHP MSSQL extension the research team gets a 200% increase in performance.

2) Price

PHP instillations are definitely cheaper on Linux which is free, on the other hand ASP runs on the IIS Server (Internet Information Server) which need's Windows N.T/2000/XP.

Apart from that ASP mostly uses MS-SQL Server as the back end which again is expensive, where as PHP programmers mostly use MySQL which is FREE.

3) Superior Memory Management

In ASP's model (in IIS 4), if an ASP file *header.ASP* is included into 20 web pages, then 20 compiled copies of that *header.ASP* are maintained in memory. IIS 5 has implemented an improved memory management model, but only programmers who are using Windows 2000 can upgrade because it is not backward compatible with IIS 4, and Windows NT 4.0 cannot run IIS 5. This means that most IIS web servers are still stuck with the inferior memory management model. This is unlike PHP, which only loads include files that are required to generate the web pages.

4) No Hidden Costs with PHP

One of the things that the buyer hates when buying a product is the hidden costs. This problem found when buying ASP, so when the buyer need encryption, he must buy ASPEncrypt, if the buyer need email management he must buy Server Object's QMail, and so on.This hidden costs not found in PHP because its built on it for free.

5) Integration with MySQL

PHP has a lot of tools that manage and maintain MySQL databases, and that makes MySQL databases very powerful when dealing with PHP, for example PHP has very useful functions like mysql_insert_id and mysql_affected_rows which are not available for other database products.

ASP and PHP are both very good solutions for mid-range web sites. What makes PHP stand out is the tight integration with MySQL. MySQL is also tuned for mid-range web-sites, where selecting and pumping loads of data is more important than transaction support. This transaction support allows the programmer to synchronies updates on multiple tables; which the majority of web sites do not require.

6) No Show Stopper Bugs

Have you ever asked Microsoft to fix a bug in ASP? If you aren't a large corporation like Boeing, the chances of getting that fix quickly are pretty low. And if it is a show-stopper of a bug, where nothing will work if this bug is not fixed, what are you going to do?

With PHP, even if you don't have the expertise in-house to fix the bug, you can definitely hire the expertise. There are no show-stoppers in PHP. Your investment is protected by the Open Source nature of PHP.

6.2.4 Support tools:

- SQLyog:

Is a very fast, compact and simple to use GUI tool to manage MySQL. This software is primarily for the users who work with MySQL during the development process.

- PHPmyadmin used to manage MySQL Database.

- Macromedia Dreamweaver MX: This is the new version of Dreamwaver, this is the operational software for system implementation.

6.2.5 Building database:

The database is to be built using mysql database management system, MySQL is an open source, Enterprise-level, multi-threaded, relational database management system. MySQL was developed by a consulting firm in Sweden called TcX. They were in need of a database system that was extremely fast and flexible. Unfortunately (or fortunately, depending on your point of view), they could not find anything on the market that could do what they wanted. So, they created MySQL. The product they created is fast, reliable, and extremely flexible. It is used in many places throughout the world. Universities, Internet service providers and nonprofit organizations are the main users of MySQL, mainly because of its price (it is mostly free). Lately, however, it has begun to permeate the business world as a reliable and fast database system. The reason for the growth of MySQL's popularity is the advent of the Open Source Movement in the computer industry. The Open Source Movement, in case you haven't heard about it, is the result of several computer software vendors providing not only a product but the source code as well. This allows consumers to see how their program operates and modify it where they see fit. This, and the popularity of Linux, has given rise the use of open source products in the business world. Because of Linux's skyrocketing popularity, users are looking for products that will run on this platform. MySQL is one of those products. MySQL is often confused with SQL, the structured query language developed by IBM. It is not a form of this language but a database

system that uses SQL to manipulate, create, and show data. MySQL is a program that manages databases, much like Microsoft's Excel manages spreadsheets. SQL is a programming language that is used by MySQL to accomplish tasks within a database, just as Excel uses VBA (Visual Basic for Applications) to handle tasks with spreadsheets and workbooks. Other programs that manage databases include Microsoft's SQL Server, Sybase Adaptive Server, and DB2.

MySQL is more than just a database. It is a system that manages databases. It controls who can use them and how they are manipulated. It logs actions and runs continuously in the background. The important thing to remember is that MySQL is large enough and quick enough to function in almost any situation.

6.2.5 Implementing input and output design:

The input interfaces, forms and screen were built by using Macromedia Dreamweaver MX, these screens was made to accept and process the purchasing operations online.

6.3 Operating the system:

In order to operate the system and make it functional the following steps must be done

- Setting the network and give the IP addresses to the computers
- Setting the infrastructure of the system (PHP 4.3.1 and MYSQL) as shown in the appendix.
- Building the database of the system
- Setup the system on the server

Chapter Seven

System Testing

7.1 Introduction

System testing is important to ensure that the designed and implemented system meets the required specifications. Also, we can find out mistakes and problems.

7.2 Testing schedule:

Estimated time / Testing process	The first week	The 2nd week	The 3rd week
Unit testing			
Modular testing			
Sub-system testing			
System testing			
Acceptance testing			

Table (7.1) Testing scheduling

7.3 Unit testing:

At this stage, the individual components are tested to ensure that they operate correctly. Each component should be tested independently. Each unit or task is tested by following the steps and forms. It was found that all units operate properly.

7.3.1 Add new product:

This form page allows the administrator to add new product, figure (7.1) shows this operation.

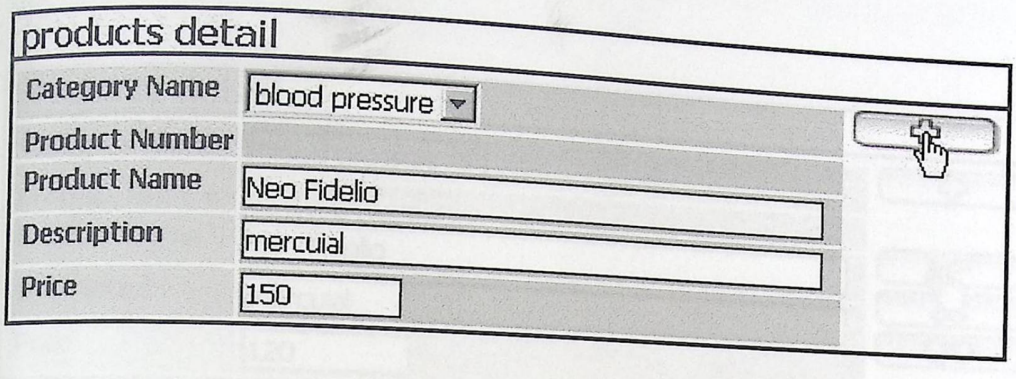
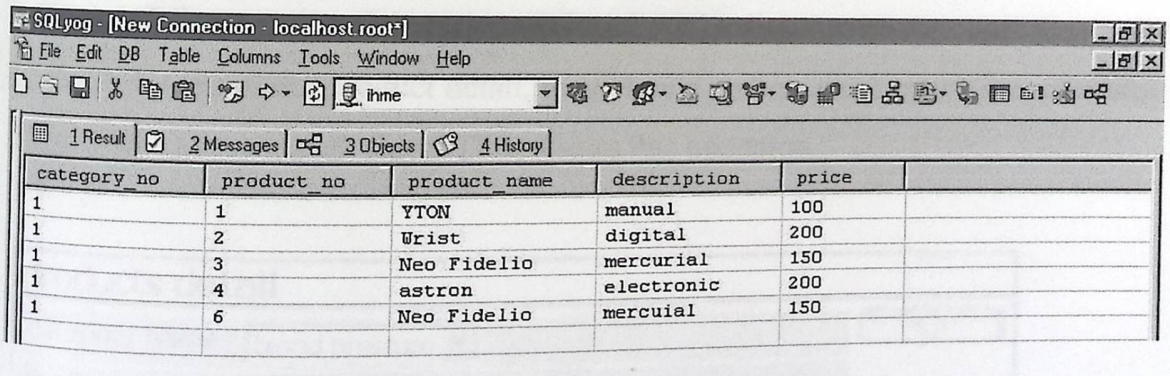


Figure (7.1) Add New Product

After adding the new product, the database tables are scanned to see if the item is added or not. Figure (7.2) shows this operation.



category_no	product_no	product_name	description	price
1	1	YTON	manual	100
1	2	Wrist	digital	200
1	3	Neo Fidelio	mercurial	150
1	4	astron	electronic	200
1	6	Neo Fidelio	mercuial	150

Figure (7.2) Products Table

7.3.2 Update products table:

The administrator can update the product name by selecting that product and then change its name, description and price in the product detail, figure (7.3) shows this operation.

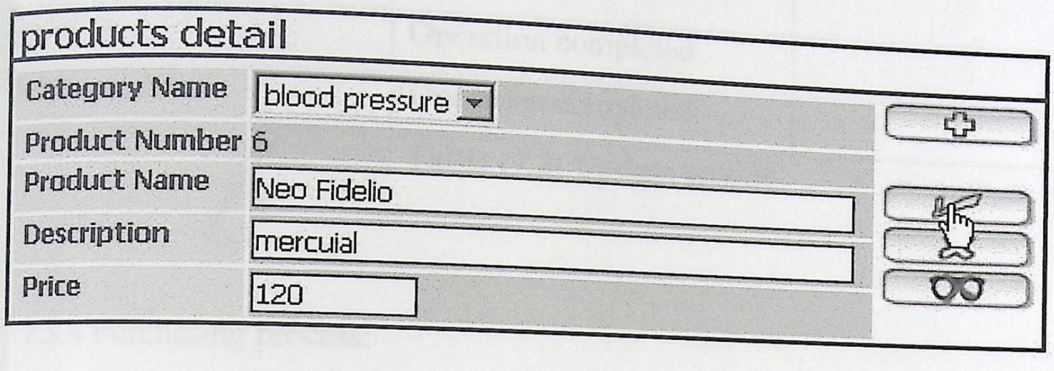


Figure (7.3) Update product

7.3.3 Delete product:

The administrator can delete the product by selecting that product and then press the delete button in the product detail, the deleting process must ensure the referential integrity role as we tested, figure (7.4) shows this operation.

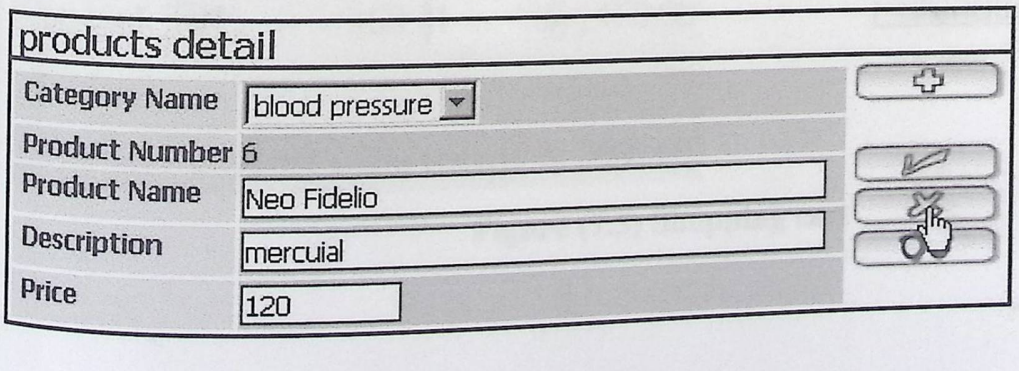


Figure (7.4) Delete Category

All of the above operations (add, delete and update) were tested for all tables in the same way. Table (7.1) summaries the previous operations and the testing results.

Process	Result	Note
Add new product	Operation completed	
Add an existing product	Operation not completed successfully	No repeat in primary key
Updating	Operation completed	
Delete product	Operation completed	

Table (7.2) Update testing

7.3.4 Purchasing process:

After the customer choose the products, the shopping cart displays the customer order information on which he can continue with shopping, recalculate the cart, empty it, or purchase now, this cart information shown in figure (7.5)

Your Shopping Cart Contains:

	Product	Price	Qty		Total
[Remove]	YTON	100	5	$	500.00
[Remove]	Wrist	200	10	$	2000.00
[Remove]	Neo Fidelio	150	1	$	150.00
			Total Price:	$	2650.00

- Continue Shopping
- Recalculate Cart
- Empty the Cart

Purchase Now

Figure (7.5) Shopping cart

This process were tested and confirmed that it done successfully, table (7.3) shows the testing result.

Choice	Data Displayed	Results	Note
Continue with shopping	Product catalogue	Customer can buy another products	Done successfully
Recalculate cart	Calculated shopping cart	New qty and the total recalculated	Done successfully
Empty the cart	No data in cart	Empty cart	Done successfully
Purchase new	the Payment method form	Customer choose the way to payment	demand bill or credit card

Table (7.3) Testing shopping cart

After the customer chooses (purchase now) hyper link, he has two choices of the payment method as shown in figure (7.6)

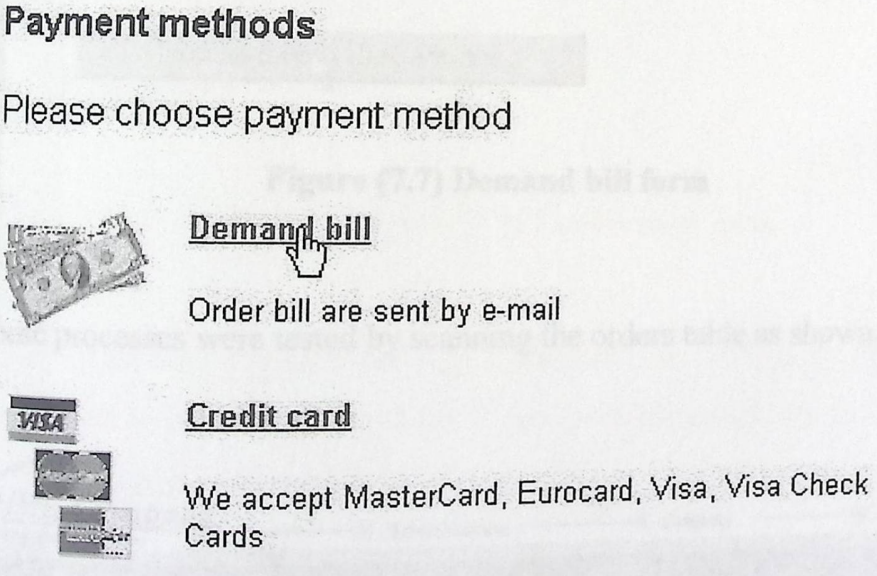


Figure (7.6) Payment method

The two links shown in figure (7.6) were tested successfully, when the customer pays by demand bill, the form of demand bill will appear as shown in figure (7.7).

IHME Company for Medical Equipments

Demand Bill Details

Date:	2003-07-13	**Bill To:**	ahmad alnatsheh
Total:	$ 1250.00	**Contact:**	2217689
		Address:	ein sara / hebron /palestine

Comments / Special Instructions

None

Order Details

Product	Price	Qty		Total
YTON	100	1	$	100.00
Wrist	200	5	$	1000.00
Neo Fidelio	150	1	$	150.00
		Total Price :	$	1250.00

Proceed with demand bill

Figure (7.7) Demand bill form

These processes were tested by scanning the orders table as shown in figure (7.8)

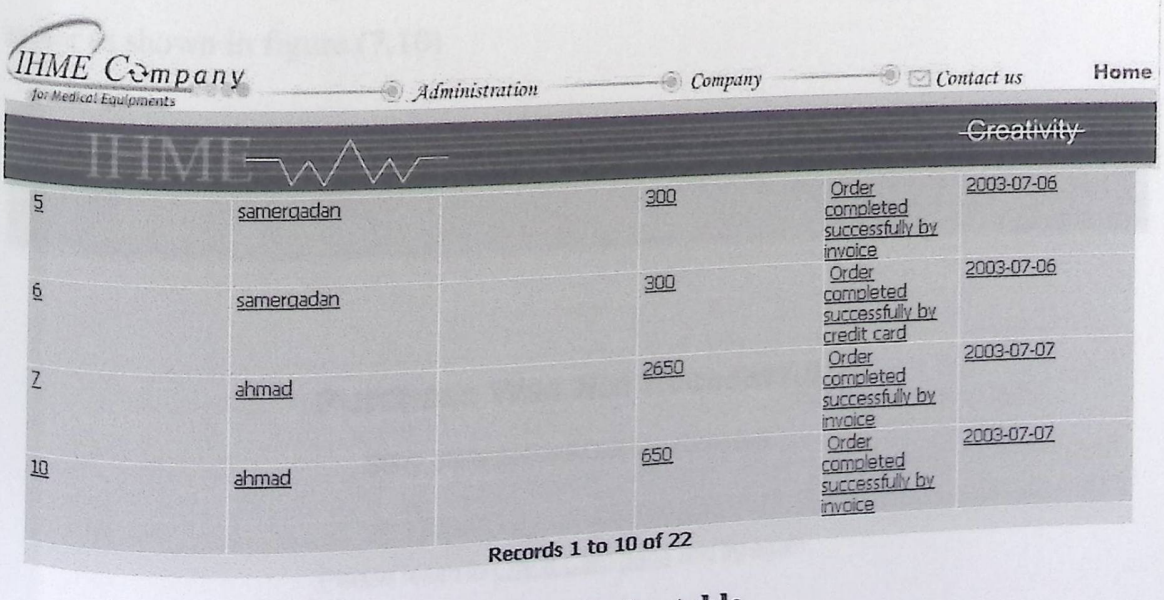


Figure (7.8) Order table

On the other hand, when the user pays by credit card, he must fill the form shown in figure (7.9) with the correct credit card number and expire date.

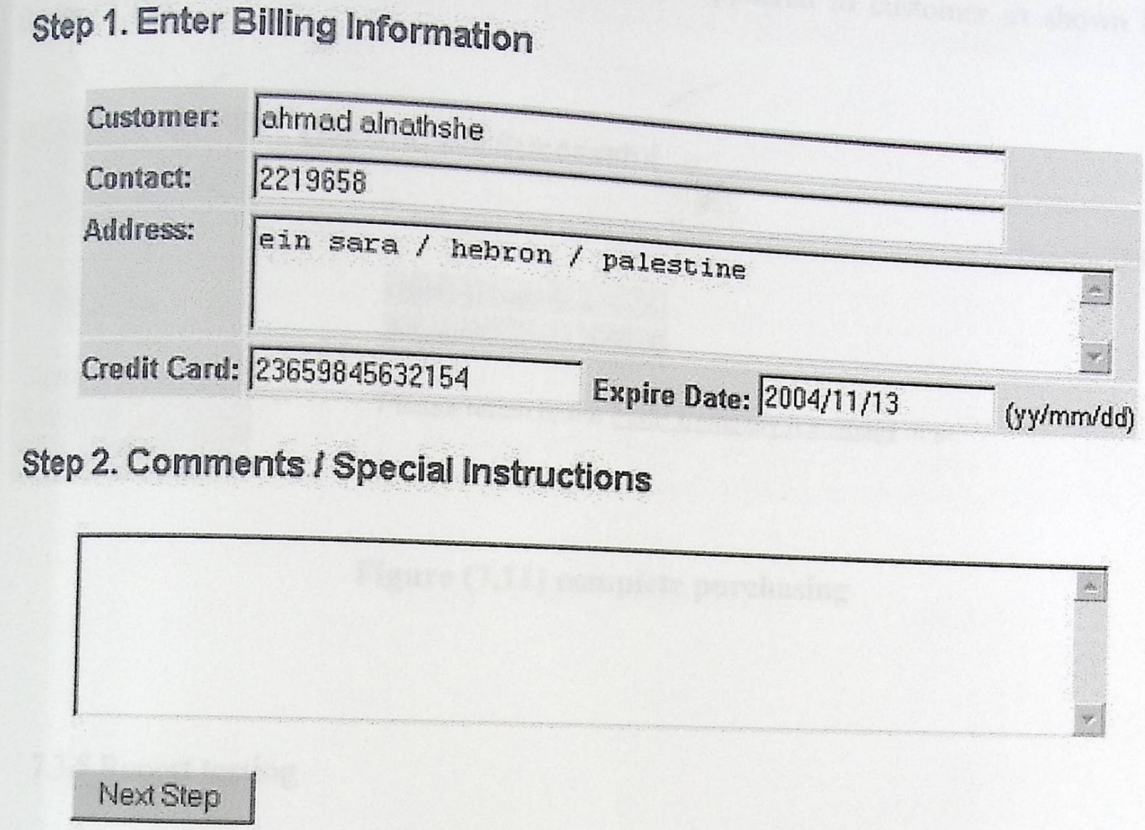


Figure (7.9) Credit card form

The credit card number and expire date will be checked by the connection to the bank server to see if they are valid, if any error happened, the following massage will appear as shown in figure (7.10)

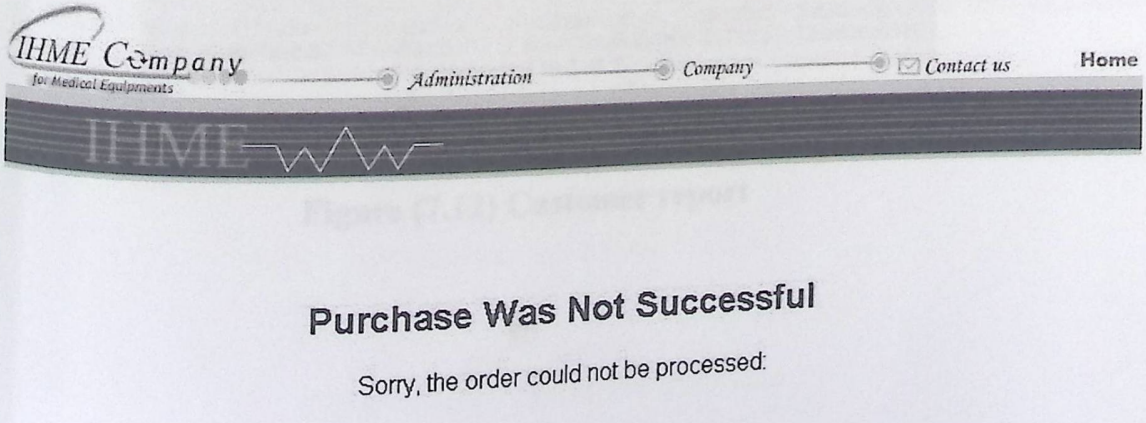


Figure (7.10) Credit card error Message

This process (testing the validation of the credit card) was done by a simulation bank system created, when the purchasing process completed by one of the two payment methods, the congratulations message appeared to customer as shown in figure (7.11)

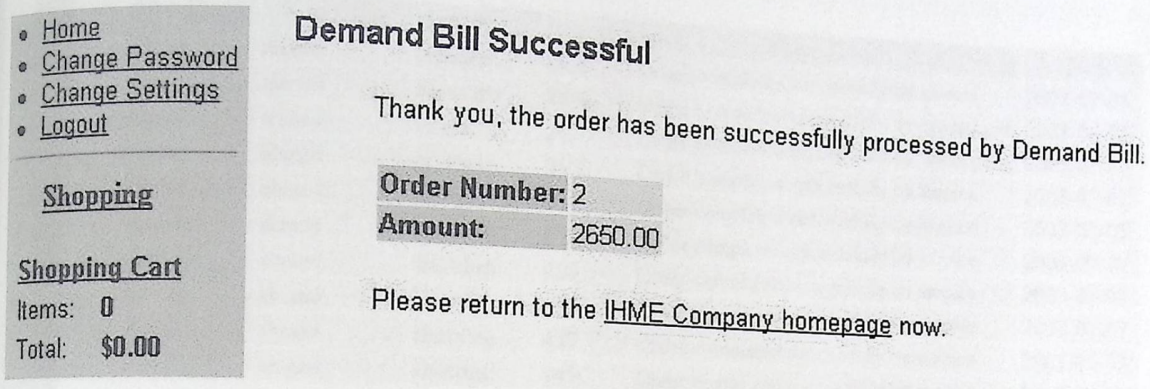


Figure (7.11) complete purchasing

7.3.5 Report testing

Figures (7.12) and (7.13) show some of the generated reports.

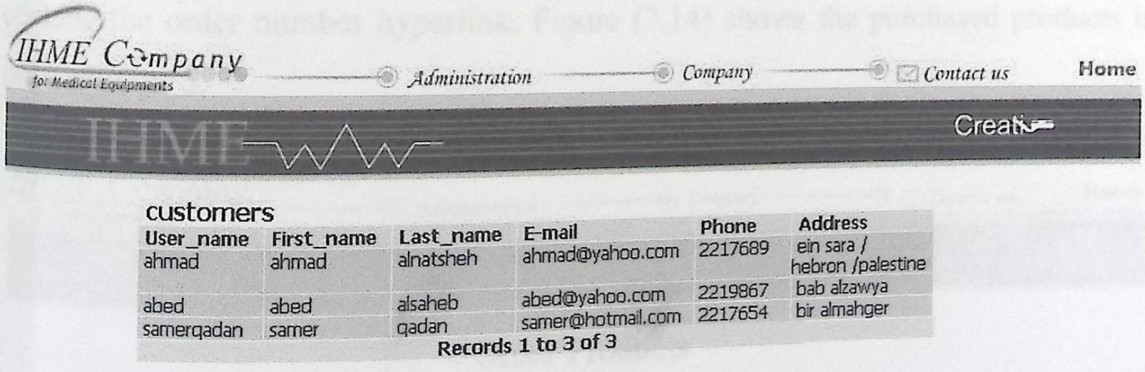
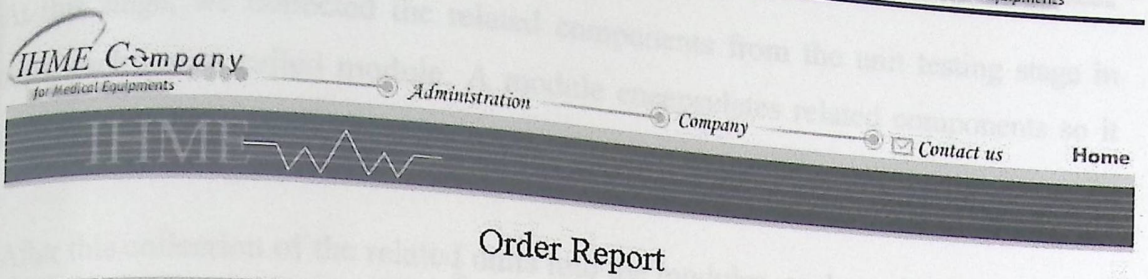


Figure (7.12) Customer report



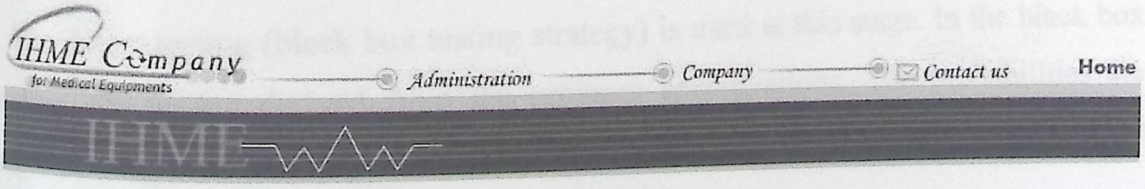
Order Report

Order No	User Name	First Name	Last Name	Total	Status Details	Date
2	ahmad	ahmad	alnatsheh	2650	Order completed successfully by invoice	2003-07-05
3	ahmad	ahmad	alnatsheh	2650	Order completed successfully by credit card	2003-07-05
4	ahmad	ahmad	alnatsheh	2650	Order completed successfully by credit card	2003-07-05
5	ahmad	ahmad	alnatsheh	650	Order completed successfully by invoice	2003-07-07
10	ahmad	ahmad	alnatsheh	650	Order completed successfully by invoice	2003-07-07
11	ahmad	ahmad	alnatsheh	450	Order completed successfully by invoice	2003-07-07
12	ahmad	ahmad	alnatsheh	450	Order completed successfully by invoice	2003-07-07
13	ahmad	ahmad	alnatsheh	450	Order completed successfully by invoice	2003-07-07
14	ahmad	ahmad	alnatsheh	1450	Order completed successfully by invoice	2003-07-07
15	ahmad	ahmad	alnatsheh			

Figure (7.13) Order report

7.4 Module testing

This report allows the administrator to see the purchased products for every order by clicking the order number hyperlink. Figure (7.14) shows the purchased products in order number 5.



Order Products

Category Name	Product Name	Price	Qty
blood pressure	YTON	100	5
blood pressure	Wrist	200	10
blood pressure	Neo Fidelio	150	1

Figure (7.14) Order Product Report

At this stage, we collected the related components from the unit testing stage in groups, each group called module. A module encapsulates related components so it can be tested without other components.

After this collection of the related units into the modules, each module is tested as a whole. For each module, a number of parameters entered to its units and checked. This stage should give the expected results from the modules, i.e. all the components of each module and all their interactions operate correctly together.

7.5 Sub-systems testing

Here, the testing performed over a collection of modules which have been integrated into sub-systems.

The following sub-systems are tested:

- Administrator
- User (customer)
- Card validation

The defect testing (black box testing strategy) is used at this stage. In the black box testing, the testing derived from the program specifications. The probabilities are taken into account and then a comparison made between the expected values and the actual values.

Administrator login:

The administrator can access the database through login form by using user name and password. Figure (7.15) shows this form

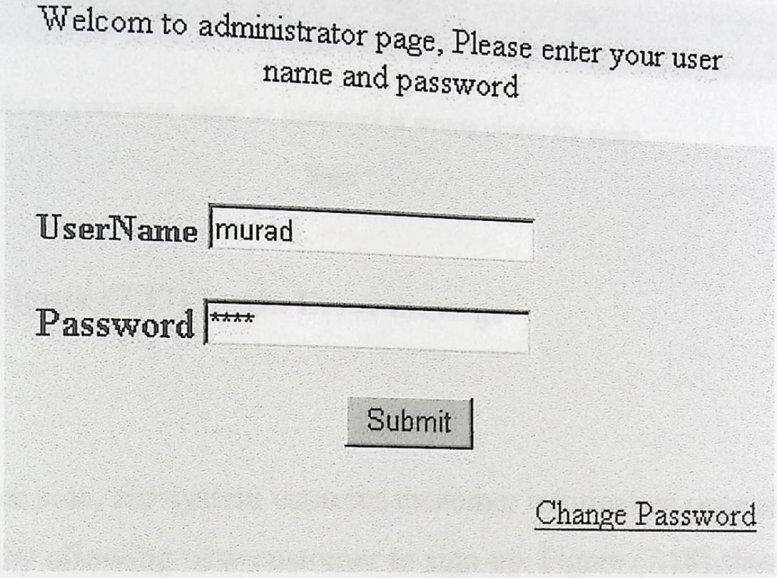


Figure (7.15) Login Form

If the administrator log in successfully, the list of database tables is displayed. Figure (7.16) show the tables and report page.

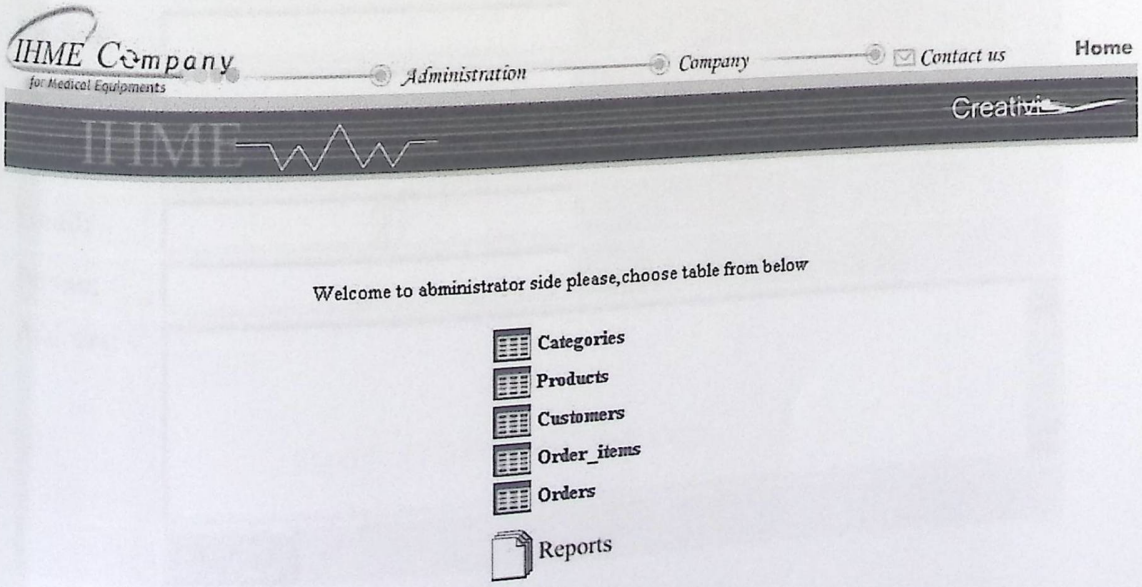


Figure (7.16) Tables and Report Page

Otherwise the access will be dined and an error message appeared. Figure (7.17) shows the error message.

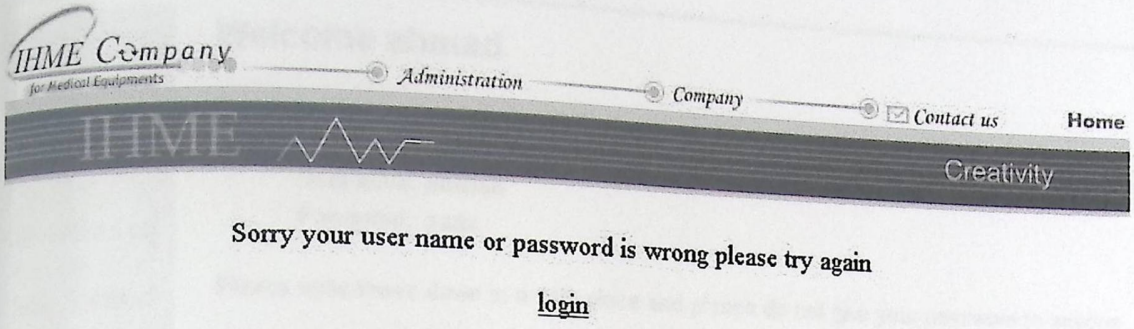


Figure (7.17) Login Error message

From the customer side, the system captures customer information once and keeps it in customers table by allowing new customer to sign up. Figure (7.18) shows the sign up form.

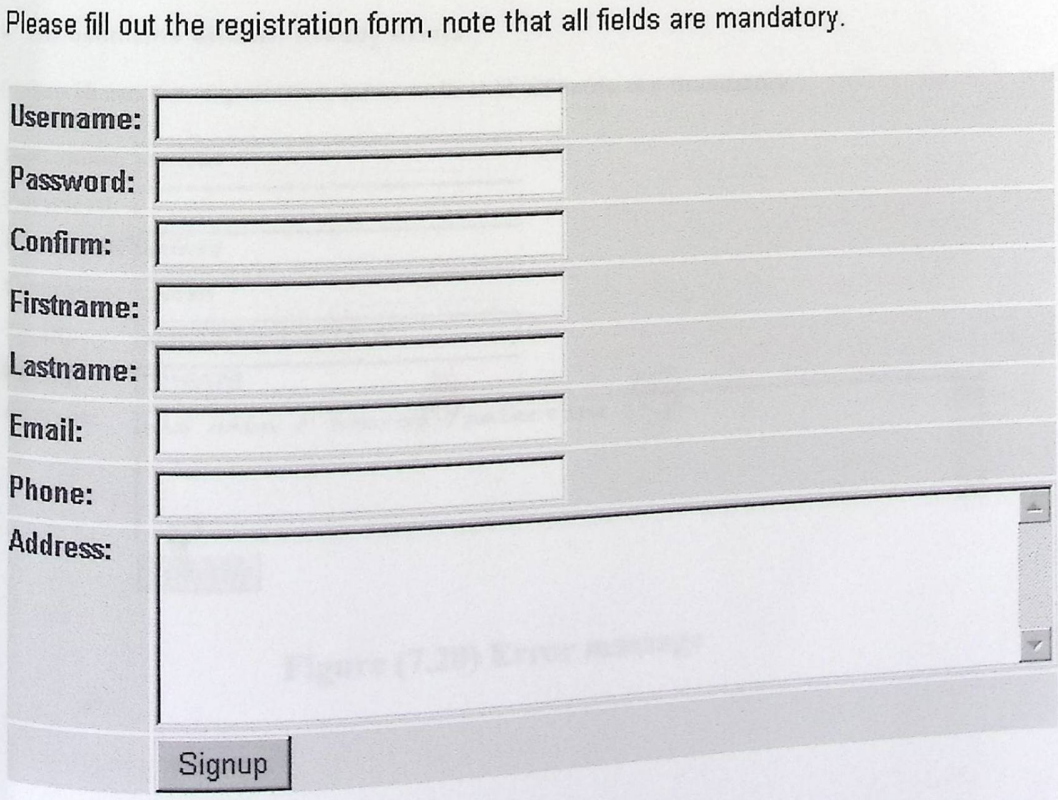


Figure (7.18) sign up form

After a new customer signed up, a registration message tells him that this operation done successfully; figure (7.19) shows this message.

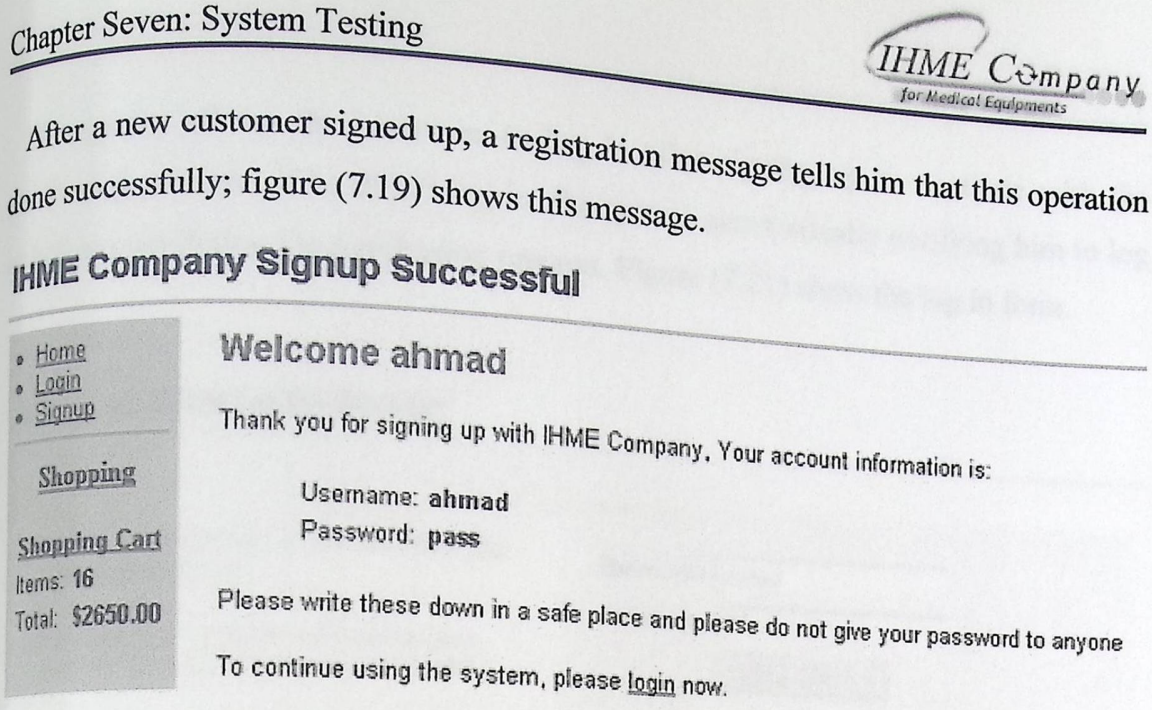


Figure (7.19) sign up successfully

If the user name was reserved an error message notifies him as shown in figure (7.20)

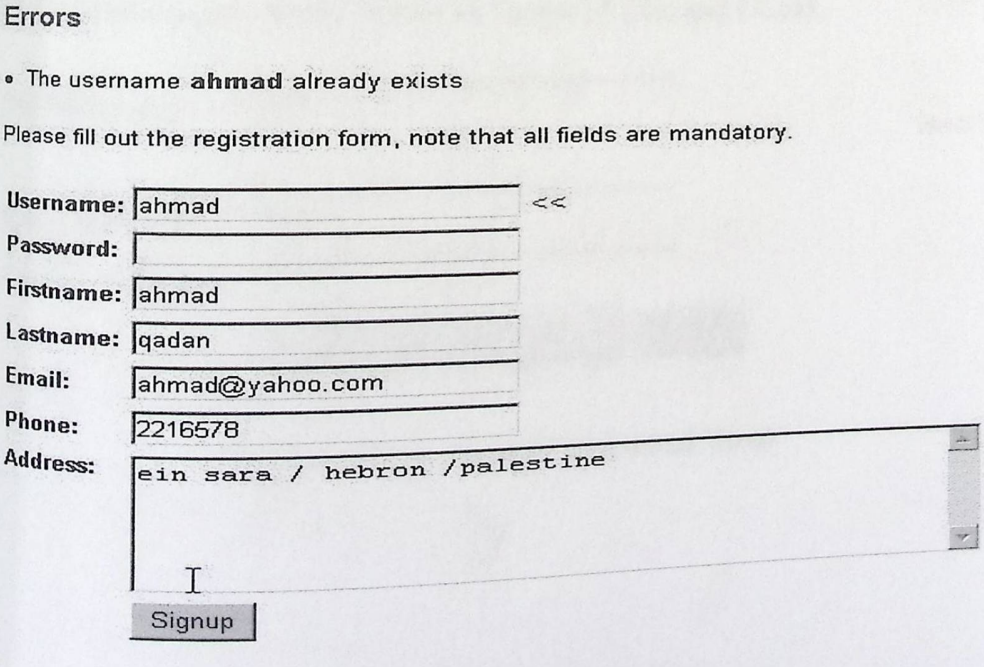


Figure (7.20) Error massage

The system allows the customer to log in after signing up to continue with the purchasing process , if he didn't log in, the system automatically notifying him to log in before completing the purchasing process. Figure (7.21) show the log in form.

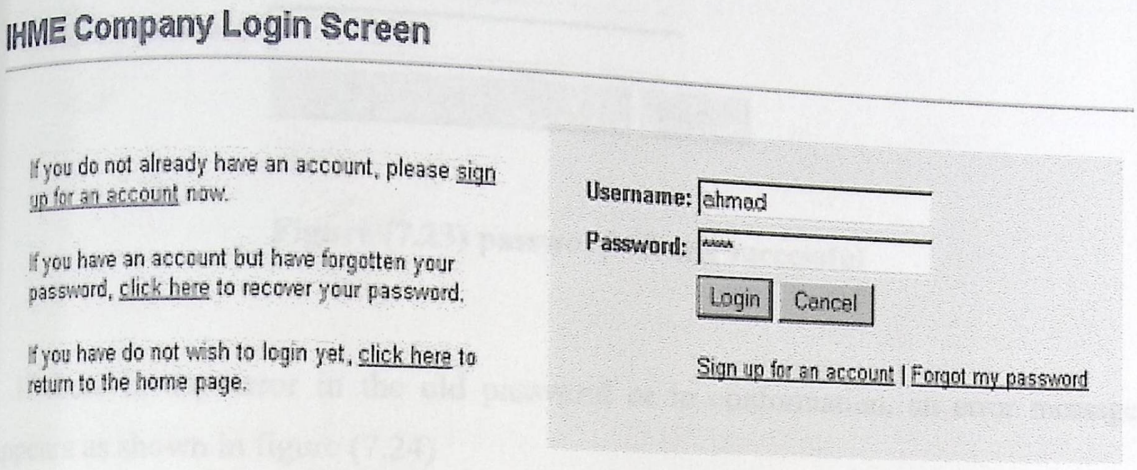


Figure (7.21) User login form

The system gives the customer the ability to change his password, changing password forms are shown below in figure (7.22), and (7.23).

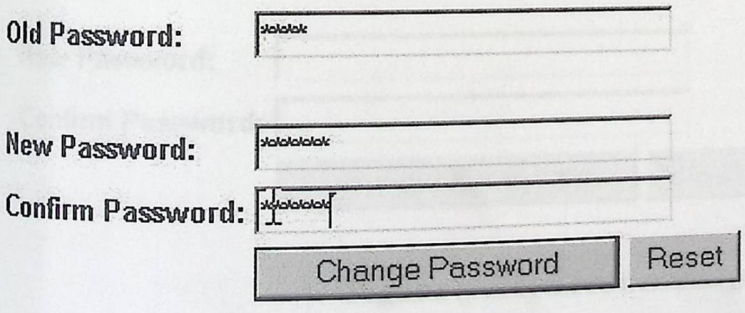


Figure (7.22) Change password form

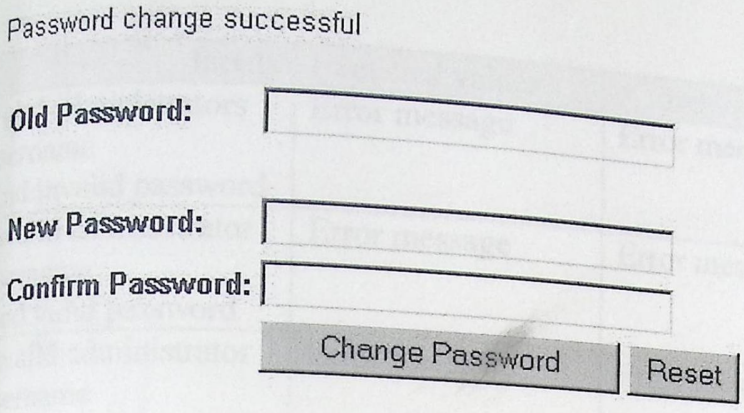


Figure (7.23) password Chang successful

If there is any error in the old password or in conformation, an error message appears as shown in figure (7.24)

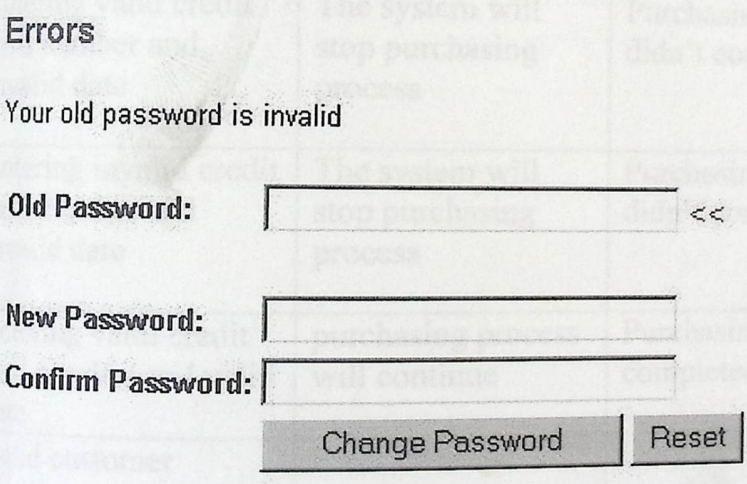


Figure (7.24) Error message

7.5.1 Black box testing:

This type of testing depending on the work of suggesting specific problems and states for the system and a proceeding answer for the behavior of the system and then apply these states on the system, the development team suggested some problems to be tested as shown in table (7.4).

Inputs	Expected values	Actual values	Notes
Valid administrators username And invalid password	Error message	Error message	Match
Invalid administrator username And valid password	Error message	Error message	Match
Invalid administrator username And invalid password	Error message	Error message	Match
Valid administrator user name And password	Open administrator page	Open administrator page	Match
Entering invalid credit card number and valid date	The system will stop purchasing process	Purchasing process didn't complete	The system ask the customer to reenter credit card information again
Entering valid credit card number and invalid date	The system will stop purchasing process	Purchasing process didn't complete	The system ask the customer to reenter credit card information again
Entering invalid credit card number and invalid date	The system will stop purchasing process	Purchasing process didn't complete	The system ask the customer to reenter credit card information again
Entering valid credit card number and valid date	purchasing process will continue	Purchasing process completed	Match
Valid customer username And invalid password	Error message	Error message	Match
Invalid customer username And valid password	Error message	Error message	Match
Invalid customer username And invalid password	Error message	Error message	Match
Valid customer username And valid password	Customer login and purchasing process will continue	Customer login and purchasing process will continue	Match

Table (7.4) Black box testing strategy

7.6 System testing

After finishing the testing of the sub-systems, the sub-systems are integrated to makeup a system. Testing here is concerned with errors that generated from anticipated interaction between sub-systems and system components.

7.6.1 White box testing

This testing will be done according to the critical paths on the flowchart figure (7.25) that we will suspect the value and then see the real value of that critical path, table (7.6) summaries the white box testing result

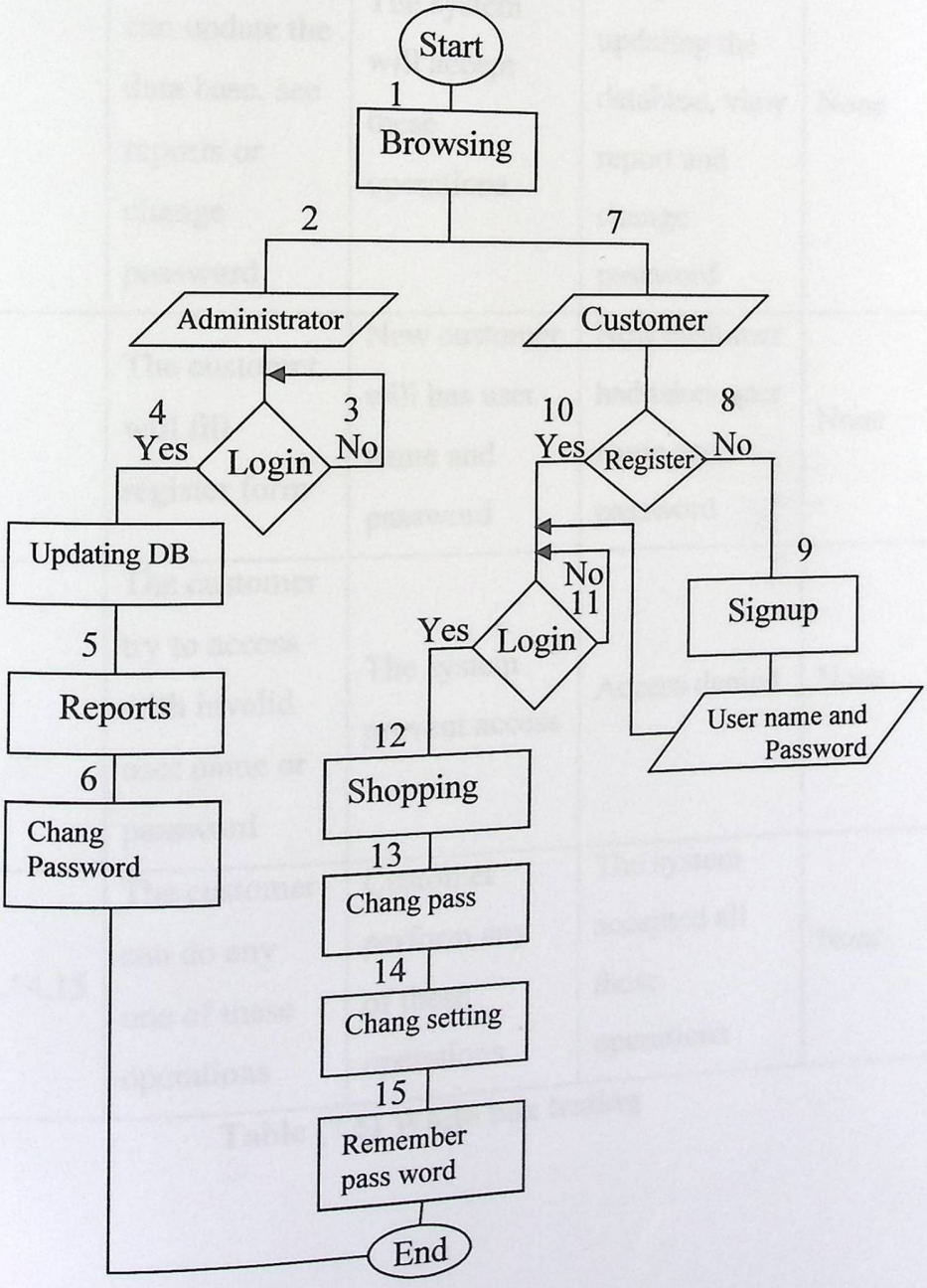


Figure (7.25) System flowchart

Branch	Input value	Expected value	Actual value	Notes
1,2,3	The administrator try to access data base with invalid user name or password	The system prevent access to database	Access denied	None
1,2,4,5,6	The administrator can update the data base, see reports or change password	The system will accept these operations	The system accepted updating the database, view report and change password	None
1,7,8,9	The customer will fill register form	New customer will has user name and password	New customer had taken user name and password	None
1,7,10,11	The customer try to access with invalid user name or password	The system prevent access	Access denied	None
1,7,10,12,13,14,15	The customer can do any one of these operations	Customer perform any of these operations	The system accepted all these operations	None

Table (7.5) White box testing

7.7 Acceptance testing

Acceptance testing is used to ensure that the system design is consistent with the system requirements. It was found that the designed, implemented system is consistent with the stated requirements

Chapter Eight

Conclusion

Conclusion

In this report, an electronic commerce system for Ibn Al-Haytham for medical equipments is analyzed, designed, implemented and tested.

The system offer the services for different levels of users including customers and administrator, it allows the administrator to update and control the system remotely. The implementations of all requirements were performed successfully, and a high level of the security is also achieved.

Here are a set of points can be identified:

1- Working in security issues gives us the ability to understand carefully the need for applications like PHP and MYSQL
2- Dealing with E-commerce is built on trust and governed by the international law of commerce.
3- Good customer service and feedback not only guarantee having new customer but also keeping them, a good feedback means not putting an email address on a web page and neglecting to check or answer that email promptly.
4- The core of this system is to facilitate E-commerce, so any customers can deal with it without having an expert knowledge in E-commerce.

Future Work

This system is an introductory step in the remote administration for E-commerce systems. It can be considered as a step for large and more important E-commerce systems in the future.

For the future work the following cases are recommended to be worked on:

a- Integration with inventory system.
b- Integration with delivery system.
c- Connection with a bank database server, for card number, user name and checking.

References

[1] Luke Welling., and Laura Thomson. "PHP and MySQL Web Developments" SAMS, 1999.

[2] David Dwyer," Web Application Development with PHP 4.0" New Riders Publishing, First Edition, 2000.

[3] Mark Maslakowski. "Sam's Teach Yourself MySQL in 21 Days," Sams, 2000.

[4] Saker Enezy ,R., "Get Familiar with PHP and MySQL," 3rd edition, Harcourt Brace Collage Publisher,2001.

[5] PHP Developer., <http:// www.phpdeveloper.org >, 2003.

[6] PHP Group., <http:// www.php.net/PHP mysql_close - Manual.htm >, 2003.

[7] R. Neassm Charles, E-commerce Planning and Management, 3rd edition, Harcourt Brace Collage Publisher, 2001.

Appendix

*******/ Create Database and Tables /******

Create database

```
Create database ihme ;
```

Category table

```
CREATE TABLE categories (
 category_no int(10) NOT NULL auto_increment,
 category_name varchar(50) NOT NULL,
 PRIMARY KEY  (category_no));
```

Products table

```
CREATE TABLE products (
 category_no int(10) NOT NULL,
 product_no int(10) NOT NULL auto_increment,
 product_name varchar(50) NOT NULL ,
 description varchar(50) NULL,
 price float(10,0) NOT NULL ,
 photo mediumblob ,
 PRIMARY KEY  (product_no,category_no),
 Foreign KEY cat category_no references categories (category_no)
);
```

Orders table

```
CREATE TABLE orders (
 order_no int(10) NOT NULL auto_increment,
 username varchar(50) NOT NULL,
 comments varchar(100) ,
 status_details varchar(50) ,
 date date NOT NULL,
 PRIMARY KEY (order_no),
 Foreign KEY cust username references customers (username)

);
```

Order items table

```
CREATE TABLE order_items (
 order_no int(10) NOT NULL ,
 product_no int(10) NOT NULL,
 category_no int(10) NOT NULL,
 price float(10,0) NOT NULL ,
```

```
qty int(10) NOT NULL,
PRIMARY KEY  (order_no, product_no, category_no )
Foreign KEY ord order_no references orders (order_no),
Foreign KEY pro product_no references product (product_no),
Foreign KEY cat2 category_no references categories (category_no)
);
```

Customers table

```
CREATE TABLE customers (
 username varchar(50) NOT NULL ,
 firstname varchar(50) NOT NULL ,
 lastname varchar(50) NOT NULL ,
 email varchar(50) NOT NULL ,
 user_password varchar(50) NOT NULL ,
 phone text ,
 address varchar(100) NOT NULL,
 PRIMARY KEY (username));
```

**************/ interface code /**************

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

<body>

<table width="784" border="0" cellpadding="0" cellspacing="0">
 <!--DWLayoutTable-->
 <tr>
  <td width="173" rowspan="2" valign="top"><img src="final%20logo.jpg"
width="170" height="59"></td>
 <td width="94" height="40"> </td>
 <td width="1"> </td>
 <td width="87"> </td>
 <td width="99"> </td>
 <td width="53"> </td>
```

```
    <td width="98"> </td>
    <td width="116"> </td>
    <td width="16"> </td>
    <td width="47"> </td>
    <td width="1"> </td>
  </tr>
  <tr>
    <td height="19" valign="top"><img src="head_atomgroup.gif" width="94"
height="14"></td>
    <td> </td>
    <td valign="top"><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
version=6,0,29,0" width="85" height="17">
      <param name="BGCOLOR" value="">
      <param name="BASE" value=".">
      <param name="movie" value="Text9.swf">
      <param name="quality" value="high">
      <embed src="Text9.swf" width="85" height="17" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" base="."></embed></object></td>
    <td valign="top"><img src="head_atomgroup.gif" width="94" height="14"></td>
    <td valign="top"><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
version=6,0,29,0" width="52" height="18">
      <param name="BGCOLOR" value="">
      <param name="BASE" value=".">
      <param name="movie" value="text12.swf">
      <param name="quality" value="high">
      <embed src="text12.swf" width="52" height="18" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" base="."></embed></object></td>
    <td valign="top"><img src="head_atomgroup.gif" width="94" height="14"></td>
    <td valign="top"><img src="but_letter.gif" width="20" height="19"> <object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
version=6,0,29,0" width="63" height="16">
      <param name="BASE" value=".">
      <param name="BGCOLOR" value="">
      <param name="movie" value="Text13.swf">
      <param name="quality" value="high">
      <embed src="Text13.swf" width="63" height="16" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" base="."></embed></object></td>
    <td> </td>
    <td colspan="2" valign="top"><object classid="clsid:D27CDB6E-AE6D-11cf-
96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
version=6,0,29,0" width="44" height="16">
```

```
        <param name="movie" value="Text14.swf">
        <param name="quality" value="high">
        <embed src="Text14.swf" quality="high"
    pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
    shockwave-flash" width="44" height="16"></embed></object></td>
     </tr>
     <tr>
      <td height="58" colspan="10" valign="top"><object classid="clsid:D27CDB6E-
    AE6D-11cf-96B8-444553540000"
    codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#
    version=6,0,29,0" width="785" height="55">
        <param name="movie" value="top%202.swf">
        <param name="quality" value="high">
        <embed src="top%202.swf" quality="high"
    pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
    shockwave-flash" width="785" height="55"></embed></object></td>
      <td> </td>
     </tr>
    </table>
    </body>
    </html>
```

Administrator side codes

Connection to database

```
?>php


function dbconn()
}
  global $dbhost,$dbuser,$dbpassword,$dbname;
$  dbconn=@mysql_connect($dbhost,$dbuser,$dbpassword);(
  if(!$dbconn(
}

    echo("Unable to connect to the database server;("!
    exit;()
{
  else
}

    if(!@mysql_select_db($dbname,$dbconn((
}
```

```
        echo("No database found!("!
      exit!()
{
{
```

log in code

```
!>php

$hostname_login = "localhost!"
$database_login = "login!"
$username_login = "root!"
$password_login!"" =
$login = mysql_pconnect($hostname_login, $username_login, $password_login) or
die(mysql_error!(()
<!
!>php
mysql_select_db($database_login, $login!(
$query_Recordset1 = "SELECT * FROM login!"
$Recordset1 = mysql_query($query_Recordset1, $login) or die(mysql_error!(()
$row_Recordset1 = mysql_fetch_assoc($Recordset1!(
$totalRows_Recordset1 = mysql_num_rows($Recordset1!(

//form has been submitted and username/password combination is correct
if ((count($HTTP_POST_VARS)>0) AND $totalRows_Recordset1 == 1(

}

//set session variable to value of the username textbox
      session_start!()
      session_register("$username!("
      $$username = $name!

//and redirect to the login successful page
      header("Location: index.html!("
{

//form has been submitted and username/password combination is wrong
elseif ((count($HTTP_POST_VARS)>0) AND $totalRows_Recordset1 != 1(


}

//redirect to the login failed page
      header("Location: /denid.php!("
{

else
```

```
}
//if form has not yet been submitted show login message

{
<?
```

add , update ,and delete codes for categories

```
function insert()
{
        global $HTTP_SERVER_VARS, $HTTP_GET_VARS,
$HTTP_POST_VARS;

        if(get_magic_quotes_gpc())
        {
                $category_name = $HTTP_POST_VARS["category_name"];
        }
        else
        {
                $category_name =
addslashes($HTTP_POST_VARS["category_name"]);
        }

        sql("insert into categories (category_name) values (" . (($category_name !=
"") ? "'$category_name'" : "NULL") . ")");
        return mysql_insert_id();
}

function delete($selected_id)
{
        // insure referential integrity ...
        // child table: products
        $res = sql("select category_no from categories where
category_no='$selected_id'");
        $category_no = mysql_fetch_row($res);
        $rires = sql("select count(1) from products where
category_no='$category_no[0]'");
        $rirow = mysql_fetch_row($rires);
        if($rirow[0])
        {
                return "Couldn't delete record due to presence of $rirow[0] related
record(s) in table 'products'";
        }

        sql("delete from categories where category_no='$selected_id'");
```

```
}
function update($selected_id)
{
        global $HTTP_SERVER_VARS, $HTTP_GET_VARS,
$HTTP_POST_VARS;

        if(get_magic_quotes_gpc())
        {
                $category_name = $HTTP_POST_VARS["category_name"];
        }
        else
        {
                $category_name =
addslashes($HTTP_POST_VARS["category_name"]);
        }

        sql("update categories set category_name=" . (($category_name != "") ?
"'$category_name'" : "NULL") . " where category_no='$selected_id'");
}
```

add , update ,and delete codes for products

```
?>php

function insert()
}
        global $HTTP_SERVER_VARS, $HTTP_GET_VARS,
$HTTP_POST_VARS;

        if(get_magic_quotes_gpc(()
        }
                $category_no = $HTTP_POST_VARS["category_no;["
                $product_name = $HTTP_POST_VARS["product_name;["
                $description = $HTTP_POST_VARS["description;["
                $price = $HTTP_POST_VARS["price;["
        {
        else
        }
                $category_no = addslashes($HTTP_POST_VARS["category_no;([["
                $product_name =
addslashes($HTTP_POST_VARS["product_name;([["
                $description = addslashes($HTTP_POST_VARS["description;([["
                $price = addslashes($HTTP_POST_VARS["price;([["
        {

        sql("insert into products (category_no, product_name, description, price)
values (" . (($category_no != "") ? "'$category_no'" : "NULL") . ", " .
```

```
((($product_name != "") ? "'$product_name'" : "NULL") . ", " . (($description != "") ?
"'$description'" : "NULL") . ", " . (($price != "") ? "'$price'" : "NULL‡("(" . ("
        return mysql_insert_id‡()
{

function delete($selected_id(
}
        //insure referential integrity...
        //child table: order_items
        $res = sql("select category_no from products where
product_no='$selected_id‡("'
        $category_no = mysql_fetch_row($res‡(
        $rires = sql("select count(1) from order_items where
product_no='$category_no[0‡("'[
        $rirow = mysql_fetch_row($rires‡(
        if($rirow[0([
        }
                return "Couldn't delete record due to presence of $rirow[0] related
record(s) in table 'order_items‡'"
        {

        sql("delete from products where product_no='$selected_id‡("'
{

function update ($selected_id(
}
        global $HTTP_SERVER_VARS, $HTTP_GET_VARS,
$HTTP_POST_VARS‡

        if(get_magic_quotes_gpc()
        }
                $category_no = $HTTP_POST_VARS["category_no‡["
                $product_name = $HTTP_POST_VARS["product_name‡["
                $description = $HTTP_POST_VARS["description‡["
                $price = $HTTP_POST_VARS["price‡["
        {
        else
        }
                $category_no = addslashes($HTTP_POST_VARS["category_no‡(["
                $product_name =
addslashes($HTTP_POST_VARS["product_name‡(["
                $description = addslashes($HTTP_POST_VARS["description‡(["
                $price = addslashes($HTTP_POST_VARS["price‡(["
        {

        sql("update products set category_no=" . (($category_no != "") ?
"'$category_no'" : "NULL") . ", product_name=" . (($product_name != "") ?
"'$product_name'" : "NULL") . ", description=" . (($description != "") ?
"'$description'" : "NULL") . ", price=" . (($price != "") ? "'$price'" : "NULL") . "
where product_no='$selected_id‡("'
```

```
{
//combobox: category_no
        $combo_category_no = new DataCombo;
        $combo_category_no->Query = "select category_no, category_name from
categories order by category_name;"
        $combo_category_no->SelectName = "category_no;"

        if($selected_id(
        }
                $res = sql("select * from products where product_no='$selected_id;('"
                $row = mysql_fetch_object($res;(
                $combo_category_no->SelectedData = $row->category_no;
}
```

delete code for customers

```
Function delete($selected_id)
{
        // insure referential integrity ...
        sql("delete from customers where username='$selected_id'");
}
```

update, and delete codes for orders

```
function delete($selected_id)
{
        // insure referential integrity ...
        sql("delete from orders where order_no='$selected_id'");
}

function update($selected_id)
{
    global $HTTP_SERVER_VARS, $HTTP_GET_VARS,
$HTTP_POST_VARS;

    if(get_magic_quotes_gpc())
    {
            $user_name = $HTTP_POST_VARS["user_name"];
            $comments = $HTTP_POST_VARS["comments"];
            $amount = $HTTP_POST_VARS["amount"];
            $status_details = $HTTP_POST_VARS["status_details"];
            $date = $HTTP_POST_VARS["date"];
    }
    else
    {
```

```
        $user_name = addslashes($HTTP_POST_VARS["user_name"]);
        $comments = addslashes($HTTP_POST_VARS["comments"]);
        $amount = addslashes($HTTP_POST_VARS["amount"]);
        $status_details = addslashes($HTTP_POST_VARS["status_details"]);
        $date = addslashes($HTTP_POST_VARS["date"]);
    }

    sql("update orders set user_name=" . (($user_name != "") ? "'$user_name'" :
"NULL") . ", comments=" . (($comments != "") ? "'$comments'" : "NULL") . ",
amount=" . (($amount != "") ? "'$amount'" : "NULL") . ", status_details=" .
(($status_details != "") ? "'$status_details'" : "NULL") . ", date=" . (($date != "") ?
"'$date'" : "NULL") . " where order_no='$selected_id'");
}
```

Change password code

```
?>php

$hostname_login = "localhost;"
$database_login = "login;"
$username_login = "root;"
$password_login;"" =
$login = mysql_pconnect($hostname_login, $username_login, $password_login) or
die(mysql_error;(()
<?
?>php


 session_start;()
     session_register("$a;("
     $a = $newpass;
          session_start;()
     session_register("$b;("
     $b = $confirm;
     session_start;()
     session_register("$c;("
     $c = $oldpass;
     session_start;()
     session_register("$d;("
     $d = $hid;
     if ($c==$d} (
 if ($a==$b} (


function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue ("" =
}
$ theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;
```

```
switch ($theType} (
case "text:"
$    theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL;"
break    ;
case "long:"
case "int:"
$    theValue = ($theValue != "") ? intval($theValue) : "NULL;"
break;
case "double:"
$    theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL;"
break;
case "date:"
$    theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL;"
break;
case "defined:"
$    theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
break;
{
return $theValue;
{

$editFormAction = $HTTP_SERVER_VARS['PHP_SELF;['
if (isset($HTTP_SERVER_VARS['QUERY_STRING} ((['
$ editFormAction .= "?" . $HTTP_SERVER_VARS['QUERY_STRING;['
{

if ((isset($HTTP_POST_VARS["MM_update"])) &&
($HTTP_POST_VARS["MM_update"] == "form1} ((
$ updateSQL = sprintf("UPDATE login SET password=%s WHERE name=%s'"
            GetSQLValueString($HTTP_POST_VARS['confirm'], "text'("
            GetSQLValueString($HTTP_POST_VARS['name'], "text;(("

mysql_select_db($database_login, $login;(
$ Result1 = mysql_query($updateSQL, $login) or die(mysql_error;(()

$ updateGoTo = "passchanged.php;"
if (isset($HTTP_SERVER_VARS['QUERY_STRING} ((['
  updateGoTo .= (strpos($updateGoTo;"?" : "&" ? (("?' '
  updateGoTo .= $HTTP_SERVER_VARS['QUERY_STRING;['

neader(sprintf("Location: %s", $updateGoTo;((


ysql_select_db($database_login, $login;(
uery_Recordset1 = "SELECT * FROM login;"
ecordset1 = mysql_query($query_Recordset1, $login) or die(mysql_error;(()
w_Recordset1 = mysql_fetch_assoc($Recordset1;(
talRows_Recordset1 = mysql_num_rows($Recordset1;(
```

```
{
elseif ($a!=$b} (
        header("Location: /mismatchpass.php");/* go to the page (unconformd
password/* (
        {
        {
        if ($c!=$d} (
header("Location: /wrongoldpass.php");/* go to the page (wrong password/* (
        {

<?
>html<
>head<
>title>Untitled Document</title<
>meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1<"
/>head<

>body<
>form name="form1" method="POST" action="<?php echo $editFormAction<"<? ؛
> p <
?>
> input name="name" type="hidden" id="name" value="<?php echo
$row_Recordset1['name<"<? ؛['
?>



insert name to text(name) from db


p> <p <
>   input name="o ldpass" type="password" id="oldpass<"
  oldpass </p<
> p <
>   input name="newpass" type="password" id="newpass<"
  newpass </p<
> p <
>   input name="confirm" type="password" id="confirm<"
  confirm </p<
> p <
>   input type="submit" name="Submit" value="Submit<"
/> p<
> input type="hidden" name="MM_update" value="form1<"
?> php /*  **********get value from db to "hid" text<?/**********
> input name="hid" type="hidden" id="hid" value="<?php echo
$row_Recordset1['password<"<? ؛['
?> php /*get value from db to hid text<?/
/>form<
/>body<
/>html<
```

```
?>php
$ScrollUp = floor(($FirstRecord - 1) / $RecordCount * $this->DataHeight؛(

$ScrollBox = floor($ScrollBox؛(
if($ScrollBox < 3) // set minimum scroll box height
        $ScrollBox = 3؛
if( ($ScrollUp + $ScrollBox) > $this->DataHeight(
$ScrollUp = $this->DataHeight - $ScrollBox؛

$ScrollDn = $this->DataHeight - $ScrollBox - $ScrollUp؛

?>
```

User side codes

Sign up new user

```
function insert_user(&$frm} (
 */add the new user into the database/*

        $qid = db_query")
        INSERT INTO customers)
                username, user_password, firstname, lastname, email,    phone,
                address
         (VALUES)
                $' frm[username'[
                 . "'‹md5($frm["password"". ([" 
                $'‹frm[firstname'[
                $'‹frm[lastname'[
                $'‹frm[email'[
                $'‹frm[phone'[
                $'‹frm[address'[
        ؛("(
{
```

Change customer settings

```
Function update_settings(&$frm} (
 */set the user's password to the new one/*

        global $SESSION؛
```

```
        $username = $SESSION["user"]["username؛["

        $qid = db_query")
        UPDATE customers SET
                email = '$frm[email'[
                ‹phone = '$frm[phone'[
                ‹address = '$frm[address'[
        WHERE username = '$username'
        ؛("
{
```

Get product and category form (show product and category in form)

```
function get_sub_categories($category_no} (
 */get the name and  of all the sub-categories under this one/*

        $qid = db_query("SELECT category_no, category_name  FROM categories
؛("
        return $qid؛
{

function get_products($category_no} (
 */get all the products under this category/*

        $qid = db_query")
        SELECT
                product_no
                ‹product_name
                ‹description
                ‹price
                ‹category_no
        FROM
                products

        WHERE  category_no = $category_no
        ؛("

        return $qid؛
{

<؟
```

Add photo function

```
؟>php
    dbconn؛()
$     imgcontent=addslashes(fread(fopen($imgfile,"r"),filesize($imgfile؛(((
$     insert_data="INSERT INTO image)

       imgname,timeline,imgfile,imgfilename,imgfilesize,imgfiletype,brief,profile (

       VALUES('$imgname','$timeline','$imgcontent','$imgfile_name','$imgfile_size'
,'$imgfile_type؛'
              $'brief','$profile؛"('
@      mysql_query($insert_data) or die("Could not Add Man of Science؛("
$     title="Added: You may add one more؛"....
$     header=$title؛

<؟
```

Get the photo from database

```
؟>php

$ get_data="SELECT * FROM image WHERE id ='$product_no؛'"
$  get_result=@mysql_query($get_data) or die("there is no connection؛("
$  result=mysql_fetch_array($get_result؛(

$  imgfile=$result["imgfile؛["
$  imgfiletype=$result["imgfiletype؛["
 header("Content-type: $imgfiletype؛("
 echo($imgfile؛(
<؟
```

Validate the billing information form

```
function validate_form(&$frm, &$errors} (

        $errors = new Object؛
        $msg؛"" =

        if (empty($frm["customer} (([["
                $errors->customer = true؛
                $msg .= "<li>You did not specify your name؛"
```

```
        {elseif (empty($frm["contact} (([" 
                $errors->contact = true؛
                $msg .= "<li>You did not specify a way to contact you؛"

        {elseif (empty($frm["address} (([" 
                $errors->address = true؛
                $msg .= "<li>You did not specify your billing address؛"

        {

        return $msg؛
{
```

Create order

```
function create_order(&$order} (
 */this function saves this order info in the database
 in the orders table for the order, and then save the shopping cart

        global $SESSION؛

         */build the custinfo string/*
        $custinfo=
                "  Customer   : $order->customer\n"
                " .Contact    : $order->contact\n"
                " .Address    : $order->address\n؛"



        $qid = db_query")
        INSERT INTO orders)
                user_name, status, custinfo, comments, amount
         (VALUES)
                $}' SESSION["user"]["username'{["

                1‹
                $'‹custinfo'
                $'‹order->comments'
                $}'‹SESSION["cart"]->total'{
        ؛("(

        $orderid = db_insert_id؛()

add the shopping cart items:

        $qid = get_cart_items؛()
        while ($item = db_fetch_object($qid} ((
```

```
            db_query")
            INSERT INTO order_items)
                order_no, product_no, price, qty
            (VALUES)
                $' orderid'
                $'‹item->product_no'
                $'‹item->price'
                $}'‹SESSION["cart"]->items["$item->product_no'{["
            ;("(
    {

    return $orderid;
{
```

Save order info

```
function create_order(&$order} (
 */this function saves this order info the database, it will make one entry
 * in the orders table for the order, and then save the shopping cart
 * contents into the order_items table/*

    global $SESSION;

     */build the custinfo string/*
    $custinfo=
            "  Customer   : $order->customer\n"
            " .Contact    : $order->contact\n"
            " .Address    : $order->address\n"
            " .Credit Card: " . chop_ccnum($order->creditcard) . " expiry $order-
>expiry\n;"

     */save order information first/*
    $qid = db_query")
    INSERT INTO orders)
            user_name, status, custinfo, comments, amount
     (VALUES)
            $}' SESSION["user"]["username'{["

            1‹
            $'‹custinfo'
            $'‹order->comments'
            $}'‹SESSION["cart"]->total'{
    ;("(

    $orderid = db_insert_id;()

     */now add the shopping cart items into the order_items table/*
```

```
        $qid = get_cart_items؛()
        while ($item = db_fetch_object($qid} ((
                db_query")
                INSERT INTO order_items)
                        order_no, product_no, price, qty
                 (VALUES)
                        $' orderid'
                        $'،item->product_no'
                        $'،item->price'
                        $}'،SESSION["cart"]->items["$item->product_no'{["
                ؛("(
        {

        return $orderid؛
{

function update_orderstatus($orderid, $status, $status_details} (
 */this function is used to update the order status after we find out if
 * the transaction was approved or not/*

        db_query")
        UPDATE orders SET
                 status = '$status'
                ،status_details = '$status_details'
                ،date = now()
         WHERE order_no = '$orderid'
         ؛("
{
```

Error message codes

```
function validate_form(&$frm, &$errors} (
 */validate the billing information form, and return the error messages in a
 * string.  if it empty there is no errors/*

         $errors = new Object؛
         $msg؛"" =

         if (empty($frm["customer} ((["
                 $errors->customer = true؛
                 $msg .= "<li>You did not specify your name؛"

          {elseif (empty($frm["contact} ((["
                 $errors->contact = true؛
                 $msg .= "<li>You did not specify a way to contact you؛"

          {elseif (empty($frm["address} ((["
                 $errors->address = true؛
```

```
            $msg .= "<li>You did not specify your billing address؛"

        {elseif (empty($frm["creditcard} ((["
            $errors->creditcard = true؛
            $msg .= "<li>You did not specify your credit card number؛"

        {elseif (empty($frm["expiry} ((["

            $errors->creditcard = true؛
            $msg .= "<li>You did not specify your credit card expiry date؛"
        {

        return $msg؛
{
```

Checking credit card validation

```
$colname_Recordset1 = "1؛"
if (isset($HTTP_POST_VARS['cardnumber} ((['
$  colname_Recordset1 = (get_magic_quotes_gpc()) ?
$HTTP_POST_VARS['cardnumber'] :
addslashes($HTTP_POST_VARS['cardnumber؛([']

{

$colname2_Recordset1 = "1؛"
if (isset($HTTP_POST_VARS['valdate} ((['
$  colname2_Recordset1 = (get_magic_quotes_gpc()) ?
$HTTP_POST_VARS['valdate'] : addslashes($HTTP_POST_VARS['valdate؛([']

{

mysql_select_db($database_bank, $bank؛(
$query_Recordset1 = sprintf("SELECT * FROM bank WHERE cardnumber = '%s'
AND valdate = '%s'", $colname_Recordset1,$colname2_Recordset1؛(
$Recordset1 = mysql_query($query_Recordset1, $bank) or die(mysql_error؛(()

$row_Recordset1 = mysql_fetch_assoc($Recordset1؛(
$totalRows_Recordset1 = mysql_num_rows($Recordset1؛(


if ((count($HTTP_POST_VARS)>0) AND $totalRows_Recordset1 == 1(

}


        session_start؛()
```

```
        session_register("$cardnumber‹("
        $$cardnumber = $cardnumber‹

{

elseif ((count($HTTP_POST_VARS)>0) AND $totalRows_Recordset1 != 1(

}

        header("Location: /denid.php‹("

{

else

}

        echo ("<p align=\"center\">Please enter your valid card number and expire
date .</p‹("<

{
<‹
```

Preparing the Windows MySQL Environment

Starting with MySQL 3.23.38, the Windows distribution includes both the normal and the MySQL-Max server binaries. Here is a list of the deferent MySQL servers you can use:

Binary Description

- Mysqld: Compiled with full debugging and automatic memory allocation checking, symbolic links, InnoDB, and BDB tables.
- Mysqld-opt: Optimized binary with no support for transactional tables.
- Mysqld-nt: Optimized binary for NT/2000/XP with support for named pipes.
- mysqld-max: Optimized binary with support for symbolic links, InnoDB and BDB tables.
- mysqld-max-nt : Like mysqld-max, but compiled with support for named pipes.

You will need to use an option file to specify your MySQL configuration under the following circumstances:

The installation or data directories are deferent from the default locations ('c:\mysql' and 'c:\mysql\data').

You want to use one of these servers:

mysqld.exe
mysqld-max.exe
mysqld-max-nt.exe
You need to tune the server settings.
Normally you can use the WinMySQLAdmin tool to edit the option file my.ini.

There are two option files with the same function: 'my.cnf' and 'my.ini'. Both files are plain text. The 'my.cnf' file, if used, should be created in the root directory of the C drive.

The 'my.ini' file, if used, should be created in the Windows system directory. (This directory is typically something like 'C:\WINDOWS' or 'C:\WINNT'. You can determine its exact location from the value of the windir environment variable.) MySQL looks first for the my.ini file, then for the 'my.cnf' file.

If your PC uses a boot loader where the C drive isn't the boot drive, your only option is to use the 'my.ini' file. Also note that if you use the WinMySQLAdmin tool, it uses only the 'my.ini' file. The '\mysql\bin' directory contains a help file with instructions for using this tool.

Using notepad.exe, create the option file and edit the [mysqld] section to specify values for the basedir and datadir parameters:

```
[mysqld]
# set basedir to installation path, e.g., c:/mysql
basedir=the_install_path
# set datadir to location of data directory,
# e.g., c:/mysql/data or d:/mydata/data
datadir=the_data_path
```

Note that Windows pathnames should be specified in option files using forward slashes
rather than backslashes. If you do use backslashes, you must double them.
If you would like to use a data directory deferent from the default of 'c:\mysql\data', you must copy the entire contents of the 'c:\mysql\data' directory to the new location.

Installing the Binaries
1. If you are working on an NT/2000/XP server, logon as a user with administrator privileges.

2. If you are doing an upgrade of an earlier MySQL installation, it is necessary to stop the server. If you are running the server as a service, use:
C:\> NET STOP MySQL Otherwise, use:
C:\mysql\bin> mysqladmin -u root shutdown

3. On NT/2000/XP machines, if you want to change the server executable (e.g., -max or-nt), it is also necessary to remove the service:
C:\mysql\bin> mysqld-max-nt --remove

4. Unzip the distribution file to a temporary directory.

5. Run the 'setup.exe' file to begin the installation process. If you want to install into another directory than the default 'c:\mysql', use the Browse button to specify your preferred directory.

6. Finish the install process.

Installing PHP

Download PHP for free from http://www.php.net/downloads.php. You'll want the *Windows Binaries* package, and be sure to grab the version that includes both the CGI binary and the server API versions if you have a choice.

In addition to PHP itself, you will need a Web server such as Internet Information Services (IIS) or Apache. PHP was designed to run as a plug-in for existing Web server software. To test dynamic Web pages with PHP, you'll need to equip your own computer with Web server software, so that PHP has something to plug into. If you have Windows 2000, XP Professional, or .NET Server, then install IIS (if it's not already on your system):

First, *whether you have IIS or not*, complete these steps:

1. Unzip the file you downloaded into a directory of your choice. I recommend C:\PHP and will refer to this directory from here onward, but feel free to Choose another directory if you like.

2. Find the file called php4ts.dll in the PHP folder and copy it to the System32 Subfolder of your Windows folder (e.g. C:\Windows\System32).

3. Find the file called php.ini-dist in the PHP folder and copy it to your Windows folder. Once there, rename it to php.ini.

4. Open the php.ini file in your favorite text editor (use WordPad if Notepad Doesn't display the file properly). It's a large file with a lot of confusing options, But look for a line that begins with extension_dir and set it so that It points to your PHP folder:
extension_dir = C:\PHP