Palestine Polytechnic University

College of Administrative Science and Informatics
Information Technology Department

E-Business

(Applied on Shawer Company)

Project Team

Amjad Shawer **Basel Shawer**

Ibrahim ismail

Project Supervisor

Dr. Muhammad Dasht.

This project is submitted in partial fulfillment of the requirements for the degree of B.Sc. in Information Technology in Palestine Polytechnic University

June 06

E-Business

(Applied on Shawer Company)

Project Team:

Amjad Shawer **Basel Shawer**

Ibrahim ismail

Project Supervisor

Dr. Muhammad Dasht.

Graduation Project

Submitted to Information Technology Department in the College of Administrative Science & Informatics

Palestine Polytechnic University

Approved by chairperson of supervisory community ----------

Date ----------

Dedication

To our Parents.....

To our Families.....

To our Friends.....

To our supervisor.....

Amjad

Basel

Ibrahem

Acknowledgment

We would like to give our thanks to every one participated in the success of this project, whether by an advice or encouragement or useful material related to the project.

First of all, we are so grateful to our supervisor **Dr. Mohammad Dasht**, who gave us a lot of his precious time to put us on the right track during our work in the project, and he is never being hesitated or late to give us from his knowledge and experience.

Also, we are grateful to: **Baha'a Amro, , Mohammad al jabary,**

Abstract

The project present an E business system for Shawer Company, the project is to be build web based system to run over internet.

The main goal of the project is facilitate the communication process between company departments (finance, manufacture, and marketing), and facilitate the communication between company customer, its employee;
The main idea of the project is to be the customer to browsing company product to choice any of them then he can submit his order to process it from administration

The system presents other services to customer, employee, and guest
Such as (E payment for bills, report) for customer, (browsing company advertisement, report) for employee, (browsing company product) for guest and other services

Table of contents

:

Chapter three: Software Requirements Specifications.

Chapter four: System Design.

Chapter five: System Implementation.

List of Tables

List of Figures

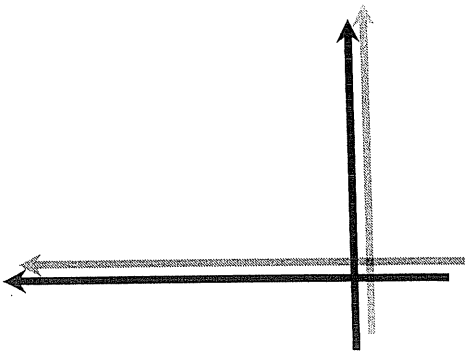
Chapter two

1

Introduction

Chapter one

- *What is E-business*
- *The difference between E-commerce and E-business*
- *The benefit of E-business*



1. Introduction

1.1 What is E Business?

E business presents a broader dimension of eCommerce as it represents the use of electronic technology, especially web and other network technology, for business.

E business represents:

1. A range of online tools and processes that transform a company's value proposition by using sophisticated data warehousing to target customers more effectively, networks to link companies with suppliers, distributors and other business partners; and flexible organizational infrastructures that adapt to change.
2. A way for companies to participate as part of a larger networked community of providers, each bringing specialized skills and new levels of performance to an e-market place.

1.2 The Difference between E-Commerce and E-Business

Electronic commerce, B2C, or e-commerce refers to online sales. Electronic business or e-business, on the other hand, refers to more than just selling online.

E-business is about utilizing Internet technologies – such as simple email, online banking solutions, websites, and more sophisticated applications such as web-based customer relationship management solutions – to provide superior customer service, streamline business processes, increase sales and reduce costs.

1.3 The Benefits of E-Business

The Internet and related technologies can change the way you develop and conduct your business processes, making them more time and cost efficient. They can diversify your marketing channels and, ultimately, help you increase your business revenue.

The Internet levels the playing field for small businesses. That is, it allows small business operators to compete on equal footing with larger businesses in the same industry.

Through the Internet, the small business can distribute information online to a global audience, immediately, with little out of pocket expense. This means you'll reach more clients or customers in a shorter period of time.

It gives you the ability to interact with your clients and customers in new ways, putting power in the hands of the buyer, giving clients or customers more choice than they've ever had before.

And finally, the Internet gives you, the seller, and the ability to readily assess your online business practices and modify them on the fly to ensure they meet the needs of customers.

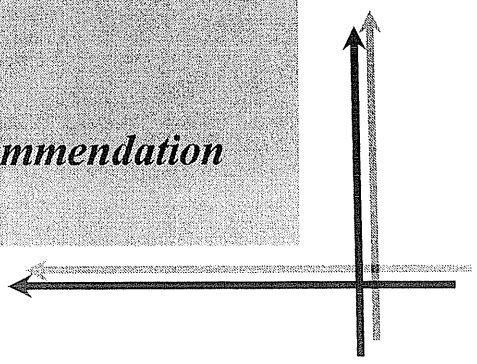
Chapter two

2

System Specification

Chapter two

- Introduction
- Project objective
- Project benefits
- Functional requirements
- Non functional requirements
- Process requirements
- Extra requirements
- Constraints and validation
- Allocation and trade offs
- Feasibility study
- Resource and cost
- Schedule
- Summary and recommendation



2.1 Introduction

After several meeting with the company administrator and understanding the specification requirement, that visits relived the problem being studied and analyzed, this chapter includes:

- ✓ Project objective.
- ✓ Benefit to the company, benefit to the development team and benefit to society.
- ✓ Functional and non-functional requirement.
- ✓ Constraint and validate.
- ✓ Development requirement.
- ✓ Feasibility study.
- ✓ Risk evaluation.
- ✓ Project plan and schedule of process.
- ✓ Summary and recommendation system.

2.2 Project Objective

These are main objective of the project:-

- ✓ Enable customer to with view product by web site.
- ✓ Enable customer to send order.
- ✓ Enable customer to make payment online.
- ✓ Enable administrator to generate balance report for each customer and employee.
- ✓ Add new item and material to company stores.
- ✓ The system allows the customer to refer up of his bill and payment.
- ✓ Reduce cost for customer and administrator.
- ✓ The system proved customer sending e-mail to company administrator any comment or problem.
- ✓ The system allows generating inventory report.
- ✓ The system provides administrator of best product sales by using system recommendation.
- ✓ The system allows he employee to view their account (Number of items was he produced and price for it).

2.3 Project Benefits

2.3.1 Benefits for company:-

- ✓ Reducing company expenses (paper, pencil).
- ✓ Increased reliability, effectiveness.
- ✓ Increase sales.
- ✓ Reduce number of employee (sales man).
- ✓ Facilitate the communication process between company, administrator, and customer.
- ✓ Facilitate the communication process between company, administrator, and employee.
- ✓ Increase customer satisfaction.

2.3.2 Benefit for Development Team:-

- ✓ *fill full the graduating requirement*
- ✓ Increase scientific experience.
- ✓ Increase our works opportunity.
- ✓ Increase our knowledge.
- ✓ Improve skills in team work.

2.3.3 Benefit for Society:-

- ✓ Increase investment on computer and internet.
- ✓ Giving opportunity to more customers to purchase product from company.
- ✓ Transfer society to advance society.
- ✓ Organize society work in company.

2.4 Functional Requirements

This system provides the following functions:

2.4.1 Public Customer (Guest) Requirements:

- **Viewing company product**:

  This function allows the public user to view all company product on the company website (can see products picture but can't make order) for all visitors.

- **Sending email:**

  This function provides guest sending e-mail to administrator about any thing he needed.

- **Guest registration:**

  This function allows the public user to register on company website and wait acceptance to register as a customer.

2.4.2 Customer requirements:

- **Browsing item**:

  The customer should be able to display item by using website.

- **Generate order**:

  The customer should be able to generate order.

- **add select item:**

  This function enables a customer to save his selected items (products) to his order.

- **Update order after generated**.

  This function allows the customer to view report for all details about order he was generated and update quantity of item was entered.

- **Browsing and send order after generated**.

  This function allows the customer to view report for all selected item in order and send it to administrator

- **Generate Customer report:**

  The customer should be able to view his report that includes (customer bills, total cost of bills, paid out cost, and remainder cost).

- **Get Feedback**:

  The customer should be able to report comment about products and feedbacks.

- **Billing:**

  The customer should be able to view and print his bills.

- **Online payment using bank web service**:

  The customer should be able to pay his bills by bank web service.

2.4.3 Employee functional requirements:

- **Viewing Employee report:**

  The employee should be able to view his report

- **Get Feedback**:

  The employee should be able to report comment about salary and request holidays.

2.4.4 Administrator functional requirements:

- **add new category**:

This function allows adding new category.

- **Add new item**:

  The administrator should be able to add new items with image.

- **update item information**:

  This function allows administrator to update item sales.

- **Add new materials**

  This function allow administrator to add new materials in store.

- **Inventory entry for materials**:

  This function allows the purchase manager to add new row materials in the store.

- **Manufacturing item**:

  The administrator should be able to enter quantity of item was produced

- **Store report**:

  This system provides administrator report that includes (item number, item quantity, material number, and material quantity)

- **User profile maintenance**:

  This function allows the administrator to add, delete, or search about any user.

- **Generate bill:**

  The system provides administrator to generate and print bill for customer's order.

- **Administrator help**:

  This function provides the administrator to how use the system.

- **Recommendation**

  This function provides administrator notification for the best item sales in sessions and decreasing quantity of item.

- **Generate Employee reports**:

  The system provides administrator reports that include (employee information payment salary for each employee, number of item was production for each employee).

- **Employee working**

  This function allows the administrator to enter information of employee working

2.4.5 Transparent functional requirements:

1. The bank should generate XML transaction to company.
2. Company should read the XML files and update its database

2.5 Non functional requirements:

Non-functional description defines properties of the system and constraints on the services offered by the system. These properties include the following:

1. Reliability: the system is to be reliable and can be used with no problem and prevent errors occurrence before they causing large problems.
2. Usability: the system must provides an administrator friendly interface to make it easy , and the system is to be easy to use since the software will be user by different culture of people.
3. Speediness and performance: the system must be very fast and acceptable , that means the users can input and retrieve information quickly.
4. Portability: the system is to access every where and to be portal to any place and environments.
5. Robustness: the system must provide high level of robustness, and work the longest time before coming down.
6. Integrity: the system must integrate with the existing systems in the company.
7. Security: means for ensuring privacy and protection for both customers and employees.

2.6 Process Requirements

1. The system and documentation must be delivered at the end of semester.
2. The database in the system is to be built using SQL sever 2000 and should be isolated in a disconnected secure environment.
3. The system is to be built with asp.net because it the most modern developing for the web.

2.7 External requirements

1. Ethical requirement: the system must prevent the unethical transaction to save user data that use the system.
2. Legislative requirements: the system is to be under the rules of the company and the lows of the country.
3. Users should be allowed freely to view and navigate contents of the site without time, cost, or load limitations.
4. The system will be in the internet server, so it must be easy to use the as the users do not have high background in system.

2.8 Constraint and validation

The system must be including constraints:

1. All data must be validated before sending to data base.
2. All products must have barcode number and must be inputted using barcode machine.
3. All inputs must be validating.
4. All users can't login the system without username and password.
5. The data base between bank and Shawer Company must separate in real.
6. The software must be applied in Shawer Company.
7. The password must be encrypted.
8. Password must be more than six characters.
9. The user must have bank ID, password and insufficient account

2.9 Allocation and trade offs

2.9.1 Functional requirement:-

Functional requirement	SW/HW
Viewing company product	SW
guest registration	SW
Sending e-mail.	SW
user login/logout	SW
Change password	SW
Browsing item	SW
Generate order	SW
add select item	SW
Update order after generated.	SW
Browsing and send order after generated.	SW
Generate customer report.	SW
Get feed back. From customer	SW
Billing.	SW
online payment using Bank web service	SW
Viewing Employee report:	SW
Get Feedback from employee	SW
add new category	SW
Add new item	SW
update item information	SW
Add new materials	SW
Inventory entry for materials	SW
Store report	SW
User profile maintenance	SW
Generate bill	SW

Administrator help	SW
Manufacturing item	SW
Generate Employee reports	SW
Recommendation	SW
Employee working	SW
Bank login	SW
Bank confirmation	SW
Generate XML transaction	SW
Reading XML files	SW

Table (2.1) Allocation (functions requirement)

2.9.2 Non-Functional requirement

Non-Functional requirement	**SW/HW**
Reliability	SW
Usability	SW
Speediness and performance	SW
Portability	SW
Robustness	SW
Integrity	SW
Security	SW

Table (2.2) Allocation (non functions requirement)

2.10 Feasibility study:

For a system to be developed from scratch, the most important issue is to evaluate its benefits against its cost, doing so requires a feasibility study that shows whether the system is cost effective or not. In both cases, a feasibility study should justify the purpose for developing such a system.

In this section, we show system alternatives that could be adapted; a Cost-Benefit analysis is conducted to justify the decision for developing the system, and an evaluation of the risks that may face the system and the development process.

2.10.1 Alternatives:

Which methodology is the best for developing our system? What are the available ones? What technologies are available? And which of them is the most appropriate? These questionnaires and others are covered and explained through this section.

2.10.1.1 Methodology:

We can develop our system by working on two different methodologies:

- Implementing WAN technologies; the idea of our system could be implemented developing distributed software applications that run on a network comprised of several computers and connected by a network connectivity technology (wired or wireless connections).

  Advantages:

  - High level of security and reliability could be achieved.

  Disadvantages:

  - Work on a limited geographical area.
  - More expansive.
  - The existence of networking problems.
  - More time is needed to install and configure the system.

- Implementing a web-based client server methodology: the computer running the site will act as a web server on the internet; user can view the system site on by browsing it using client machines connected to the web.

  Advantages:

  - Cover the entire world without geographical restrictions.

- Faster in registering new user.
- Less technical or equipments configurations needed.
- Less expensive.
- No networking problems.
- More scalable.

Disadvantages:

- Security issue is the major challenge.

Conclusion

By comparing the two available methodologies and putting the need to globalize the scope of the system under light, we have chosen the second methodology (web-based client server) and we worked on it.

2.10.1.2 Environment:-

As described in choosing the methodology, the aim of our system is to make it available for users on the internet.

There are tow language support that:

1. CGI (Common Gateway Interface)

Advantage:

- Runs on many different platforms which will ensure its continued popularity.
- CGI is still very popular with many big web sites, particularly those running on UNIX operating systems. It also runs on many different platforms.

Disadvantage:

- CGI requires a lot of server resources, especially in a multi-user situation.
- It is not easy for a beginner to learn how to program such modules.

2. Active Server Pages (ASP)

Advantage:

- A new and powerful server-side technology for creating dynamic web pages.
- Create faster, more reliable dynamic web pages with any of the programming languages supported by the .NET Framework (VB.net).
- With Microsoft .NET, developing Web applications is much easier.
- An infrastructure that is already built.

- ASP.NET allows you to use a far greater selection of full programming languages

Disadvantage:

- You had a lot of related technologies from HTML, Script, Data access, etc.
- The tools were terrible.
- Expensive cost.

***Conclusion*:**

By comparing the two Environments available and putting the need to globalize the scope of the system under light, we have chosen the second language (ASP.NET) and we worked on it.

2.10.2 Cost benefit analysis

There are tangible and non-tangible benefits for the system

2.10.2.1 Tangible benefits:

There are no tangible benefits in this project.

2.10.2.2 Non-tangible benefits:

After we understanding and analysis this system, when implemented the system the company can be gets many benefits such as improve the efficiency of it work and give user more benefits:

1. Reduce time of reading orders to give agreement.
2. Improve the efficiency of the administrator work.
3. Improve customer relationship and loyalty.
4. Reduce duplication in data.
5. Increase facilities dealing with customer.
6. Reduce the time and effort for viewing catalogs.
7. Easy to up date product information.
8. Ease of connecting order processing, inventory processing, and payment processing to the system.

2.10.3 Evaluation Risks

In this section will be discus several risks that may be face our system and give proposed solution.

2.1.3.1 Risks

1. Fear of customers to purchase electronically because increasing amount of fraud on internet
2. The lack of universally accepted standards for quality, security, and reliability.
3. Some customers like to feel and touch products
4. many legal and public policy issues , including taxation, are as unresolved
5. special web server are needed in additional to the networks servers (added cost)
6. security and privacy concerns deter customers from buying
7. Insufficient telecommunication data rate (network speed).

2.10.3.2 Proposed solution:

1. Implementing reliable methods for financial transactions to some good degree will gain the trust of companies and customers. These methods will overcome the risks of the security issue and fear from dealing with internet for financial transactions.
2. Increase the quality of telecommunication services, increase the data rate speed.
3. Encouraging the public class to use the E-commerce application using media.
4. Study all the requirements and identify the importance of each one.

2.11 Resources and Cost:

Hardware cost:

Item	Number of units	Cost of unit(NIS)	Total cost(NIS)
Computer p4 2400 40G HD,512 RAM	2	3850	7700
Printer	1	1000	1000
Camera	1	600	600
scanner	1	500	500
Total=9800			

Table (2.3) Hardware cost

Software cost:

Item	Number of unit	Unit cost (NIS)
ASP.NET 2003	1	450
Acrobat reader	1	450
Office XP	1	450
SOL server2003	1	600
Macromedia flash	1	120
Total=2070		

Table (2.4) Software cost

Humans:

No	Team Role	Number	Hours/week	Cost\hour	Total
1	IT student	3	25	150	11250
2	Interface Design GUI	1	25	100	2500
Total Human Cost [15 week] 13750					

Table (2.5) human cost

Books:

Book Name	Number	Unit cost(NIS)	Total cost(NIS)
ASP.NET	1	60	60
Electronic commerce	1	100	100
Total=160			

Table (2.6) book cost

Others:

There are other 500NIS costs such as (papers, transportation, ink…)

The following tables include the total costs

Cost type	Cost (NIS)
HW costs	11000
SW costs	2320
Human costs	13750
Books costs	160
Others	500
Total=	27730

Table (2.7) Total cost of project

2.12 Schedule:

Activities(task)	symbol	dependences
Determining project subject and writing the introduction	T1	-
System specification	T2	-
Software requirement specification	T3	T2
Design	T4	T3,T2
Coding and implementation	T5	T4,T3
Testing	T6	T5
Maintenance	T7	T6
Writing the document	T8	T2,T3,T4,T5,T6,T7,T8

Table (2.8) Projects Activates and Their Dependents

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1																
T2																
T3																
T4																
T5																
T6																
T7																
T8																

Table (2.9) Project Scheduling

2.13 Summary and Recommendation

According to the mentioned system specification which is approved by the company and the supervisor of the project, the work team decided to continue with the software requirements specification phase.

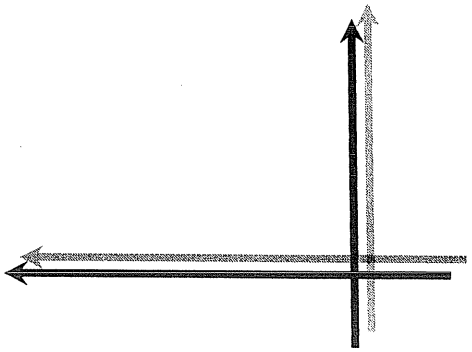
Chapter three

3

Software requirement Specification

Chapter three

- Introduction
- Functional description
- Validation criteria
- Information description
- Summary and recommendation



3.1 Introduction

In this chapter we will to discuses software requirement specification and will be describing functionality for the system software, this chapter will cover three topics:

1. Functional description
2. Validation Criteria
3. Information Description

3.2 Functional Description

This section will cover major functional description of our project (public customer requirements, customer requirements, administrator requirements, and transparent function requirements) that contains:

- ✓ Function Name.
- ✓ Description.
- ✓ Input.
- ✓ Source
- ✓ Output
- ✓ Destination
- ✓ Require
- ✓ Pre-condition
- ✓ Post-condition
- ✓ Validation

3.2.1 Public user requirement

Name: Viewing company product

Description: this function allows the public user to view all company product on the company website (can see products picture but can't make order) for all visitors

Input: click on button. "منتجاتنا"

Source: web form.

Output: advertisement page that include (item picture, item name).

Destination: screen.

Require: none

Precondition: enter company website

Post condition: none

Procedure: this function allows the public user to view all company products without details of product on the company website for all people in the world by enter company website (www.shawer.com) and click on button "منتجات الشركة".

Validation: none

Figure (3.1) viewing company product

Name: guest registration

Description: this function allows the public user to register on company website and wait acceptance to register as a customer.

Input: name, company name, address, tell, fax, city, e mail

Source: web form.

Output: none

Destination: New database records for the new guest account Information.

Require: none

Precondition: guest must visit home page

Post condition: wait acceptance from administrator

Procedure: this function allows guest request to register as customer in company system and must enter all information about him by click on button "زبون جديد."

Validation: All inputs must be filled correctly.

Figure (3.2) quest registration

Name: sending e-mail.

Description: this function provides guest sending e-mail to administrator about any thing he needed.

Input: click a button "e-mail".

Source: web form.

Output: email sends to administrator email account..

Destination: screen.

Require: none.

Precondition: none.

Post condition: none.

Procedure: this function allows the client to click on the button "e-mail" for sending it to the administrator.

Validation: none.

Figure (3.3) send e mail

3.2.2 Customer requirement

Name: customer login

Description: this function allows the customer to login the system by putting username and password.

Input: valid customer ID and password.

Source: web form.

Output: customer home page

Destination: SQL database server.

Require: valid login and valid customer ID and password.

Precondition: the user must visit shawer website

Post condition: the customer can access to special web form

Procedure: this function will request customer ID and password from customer to login the system; if customer ID and password was wrong the system appear error massage else the user can enter the system.

Validation:
1. All fields should be filled with data.
2. Consumer ID and password must be meet database record.

Figure (3.4) customer login

Name: Change password

Description: this function allows the customer to change his password

Input: old password, new password, password confirm

Source: web form.

Output: New customer password.

Destination: system database.

Require: the customer must login the company system (the customer must have old password).

Precondition: the customer must login to the company system through old password.

Post condition: the customer have new password to login later and must logout.

Procedure: this function will request username, password, new password, and password confirmation then an expression validation will be applied on the password, if the validation succeeds the new password will be sent to the database else an error message will be displayed.

Validation: the customer must be entering new password at least six character, doesn't include special character and must not equal the user ID and new password equal password confirmation.

Figure (3.5) customer change password

Name: customer logout

Description: this function allows the customer to logout the system.

Input: Click on button" تسجيل خروج".

Source: web form.

Output: main web site.

Destination: screen.

Require: none.

Precondition: customer must have been logged in company system.

Post condition: Return to main page in the web site.

Procedure: This function should be erasing the value that assigned to the session when the customers click on button ("logout").

Validation: the customer must have already logged in.

Figure (3.6) customer logout

Name: Browsing item.

Description: this function allows the customer to displaying item picture and knows all details of this item (product name, price, description).

Input: Click on button "عرض البضائع".

Source: web form.

Output: web form that contains item details (picture, price, description, product name)

Destination: screen.

Require: quantity of item and item must be available.

Precondition: The customer must have customer ID and password to login the system

Post condition: browsing all items.

Procedure: This function allows the customer to display all items that available in company stores, by click on button "عرض البضائع" to select quantity of item

Validation: the customer must have privilege to enter the company system and must be generate order.

Figure (3.7) browsing item

Name: Generate order

Description: this function allows the customer to purchase item by generate order.

Input: order date, order number, customer ID

Source: web form.

Output: create new record (generate new order)

Destination: system database.

Require: quantity of item must be stabile.

Precondition: The customer must have user ID, password and must log in the system

Post condition: customer has new order.

Procedure: This function allows the customer to generate new order by enter all input in filled then click on button "إنشاء طلبيه".

Validation: the customer must have privilege to enter the company system.

Figure (3.8) generate order

Name: add select item

Description: this function enables a customer to save his selected items (products) to his order.
Input: item name, quantity.

Source: web form.

Output: new product is added to the order.

Destination: system database.

Require: customer session ID.

Precondition: The customer must have user ID, password and must log in the system

Post condition: new record in the database.

Procedure: This function allows the customer to select new item and determine quantity of it and save his order in database by click button"تخزين"

Validation: the customer must enter quantity of item grater than 12 pes.

Figure (3.9) select item

Name: update order after generated.

Description: this function allows the customer to view report for all details about order he was generated and update quantity of item was entered.

Input: Click on button that name "تعديل".

Source: system database and web form.

Output: update record (updates quantity of item.)

Destination: system database

Require: customer must generate order.

Precondition: the customer must select item and enter quantity of it.

Post condition: update record in database.

Procedure: This function allows the customer to update quantity of item by select button "تعديل" then enter new quantity of item.

Validation: the quantity must be grater than zero.

Figure (3.10) update order

Name: Browsing and send order after generated.

Description: this function allows the customer to view report for all selected item in order and send it to administrator

Input: Click on button" عرض الأصناف المختارة

Source: system database.

Output: web form that contains report about order details (items name, quantity, unit sales, description, and subtotal).

Destination: screen, printer, database.

Require: customer login and must generate order.

Precondition: generate order to confirm it.

Post condition: viewing order details and send the order to administrator.

Procedure: This function allows the customer to display report about order after confirm it by click on ("عرض البضائع المختارة") then send order to administrator and finally click on "إرسال".

Validation: none.

Figure (3.11) browsing and send order

Name: Generate customer report.

Description: this function allows the customer to display and print his report that includes (customer bills, total cost of bills, states of bill (paid or not paid). Total cost of bill was not paid, total reminder

Input: Click on button "تقرير"

Source: system database.

Output: web form that contains customer report (customer bills, total cost of bills, states of bill (paid or not paid) Total cost of bill was not paid, total reminder.

Destination: screen, printer

Require: customer login, customer payment through bank website , customer bills and generate order. .

Precondition: the customer must generate order and pay bill through bank website.

Post condition: viewing and printing customer report.

Procedure: This function allows the customer to access the DB (can read his information only) without edit his in function.

Validation: the system appear error massage if doesn't have order.

Figure (3.12) generate customer report

Name: Get feed back from customer

Description: this function allows the customer to send his comments and suggestions about product and anything to administrator.

Input: comments and suggestion and his email

Source: web form

Output: create new record (new comments in database).

Destination: system database

Require: customer must be register in company system and login.

Precondition: the customer must login the company system

Post condition: adding comment into system database

Procedure: this function allows the customer to write comment about product and enter email finally click on button "إرسال"

Validation: must enter email syntax correctly and comment text not empty

Figure (3.13) get feed back from customer

Name: Billing.

Description: this function allows the customer to view, save and print only his bill (bill details).

Input: Selecting button

Source: system database.

Output: bill details (item name, price, subtotal, quantity, total cost

(Printed or save).

Destination: screen, printer

Require: the customer must login the company system (DB).

Precondition: the customer must be generate order

Post condition: viewing and printing customer bill.

Procedure: This function allows the customer to display his bill after generate order by click on invoice.

Validation: the system appear error massage if doesn't have order.

Figure (3.14) billing

Name: online payment using Bank web service

Description: this function allows the consumer to pay his bill online using Bank website.

Input: account number, bill number, customer ID, and amount of money to be paid

Source: web form.

Output: payment confirmation.

Destination: screen.

Require: The registered must have been logged in Shawar company website and had a bank account.

Precondition: the customer must login Bank system.

Post condition The customer should be able to log in the bank website.

Procedure: This function allows the customer paid his bill online by using bank website when the customer click on bill that is not paid then the customer translates on bank website

Validation: all inputs must be filled in and the customer account must greater or equal to total value of bill.

Figure (3.15) payement using web service

3.2.3 Employee requirement

Name: Viewing Employee report:

Description: This function Enable the employee to view his report that includes (number of hour he work, cost of hour, total paid, total reminder).

Input: Enter company website

Source: system database

Output: employee report

Destination: Screen, printer

Require: employee must be register in company (has employee number, password) and still working in company.

Precondition: employee must login the company website.

Post condition: printing employee report.

Procedure: this function allow employee to access the database system to read employee information (hour employee working ,and cost of hour) then can display it on to the screen as report.

Validation: none

Figure (3.16) employee report

Name: Get Feedback from employee

Description: this function allow s employee to report comment to administrator about salary and request holidays.
Input: .employee comments and email

Source: employee.

Output: comments

Destination: system database

Require: employee must be register in company and work in it (has number, password)

Precondition: employee must login the system.

Post condition: adding in to system database

Procedure: this function allows the employee to write comment about product and enter email finally click on button "إرسال!"

Validation: must enter email syntax correctly and comment text not empty

Figure (3.17)get feed back from employee

3.2.4 Administrator requirement

Name: add new category.

Description: This function allow administrator to adding new category with its items.

Input: category no, category name.

Source: web form.

Output: new record in DB (new category)

Destination: database.

Require: login page.

Precondition: can not add new category.

Post condition: can add new category.

Procedure: this function allows adding new category from textbox, this category will be saved in DB.

Validation: none.

Figure (3.18) add new category

Name: Add new item

Description: this function allows administrator to add new item store.

Input: item name, item no, picture item, description, unit sales, unit cost, item quantity material.

Source: web form.

Output: new record in DB (new item) .

Destination: system database.

Require: production manager must be login. .

Precondition: up load picture of new item
Post condition: adding into data base system

Procedure: this function allows administrator adding new item from textbox and will be saved in database

Validation:
1-All inputs are filled in and must be correctly.
2- Sales price must greater than cost price.

Figure (3.19) add new item

Name: update item information

Description: this function allows administrator to update item sales.

Input: new item price sales, item no.

Source: web form.

Output: update item sales price in database.

Destination: system database.

Require: item must be available.

Precondition: none

Post condition: update database record.

Procedure: this function allows the administrator to update sales price of item by select item name from drop dawn list then click on button "تعديل"

Validation: all inputs are filled in.

Figure (3.20) update item information

Name: Manufacturing item

Description: the administrator should be able to enter quantity of item was produced

Input: item no, material no, quantity, category no.

Source: web form.

Output: new record in database.

Destination: system database

Require: the administrator must login the system.

Precondition: the administrator must define the item.

Post condition: adding to system database.

Procedure:

Validation: must be item name, category name are available and quantity grater than 0.

Figure (3.21) manufacturing item

Name: Add new materials

Description: this function allow administrator to add new materials in store.
Input: material name, material number, description.

Source: web form.

Output: adding new material in database.

Destination: system database.

Require: administrator must be login. .

Precondition: none.
Post condition: adding into data base system

Procedure: this function allows adding new material from textbox and will be saved in database

Validation: all inputs are filled in.

Figure (3.22) add new material

Name: Inventory entry for materials

Description: this function allows the administrator to add new row materials in the store.
Input: quantity, purchase price, width, number of roll, length, category number.

Source: web form.

Output: add quantity of materials in database.

Destination: system database.

Require: fabric (material) name must be available.

Precondition: select category name and material name.
Post condition: adding new fabric quantity into database.

Procedure: this function allows manager to select fabric name from database by using dropdown for each and after that add quantity into database

Validation: fabric name, quantity width, roll width must be available.

Figure (3.23) Inventory entry for material

Name: Store report:

Description: this function allows administrator to display reports for each stores that contains information about material and item available in store such as (item number, item quantity, material number, material quantity)

Input: click on button " "

Source: web form.

Output: inventory report for each store.

Destination: screen, printer.

Require: administrator login, item and material are available.

Precondition: None

Post condition: display store reports

Procedure: this function that allows administrator to click on button " تقرير المخزن" then can show all details of item and materials are existed in store

Validation: none.

Figure (3.24) store report

Name: User profile maintenance

Description: this function allows administrator to add, delete and search about any users (customer, administrator, and employee)
Input: information for each user (customer, administrator, employee).

Source: web form

Output: add new user delete any user (customer, administrator, employee)

Destination: system database

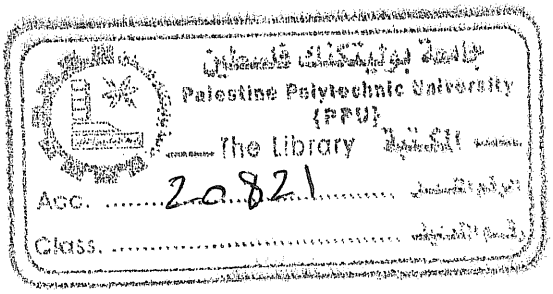
Require: new user to add or old user to delete or search.

Precondition: select user name.
Post condition: adding new user to data base

Procedure: this function allow administrator to enter the user information then click on button "إضافة" to add new user to database, and can delete any user by click on the button" حذف" .

Validation: user not available in database when add, and available when delete or search.

Figure (3.25) User profile maintenance



Name: Generate bill

Description: this function allows the administrator to generate bill for each order was fill full

Input: customer ID, bill date, bill no, order no.

Source: web form.

Output: customer bill

Destination: screen, printer.

Require: order must be generate.

Precondition: select order number for customer who was made order

Post condition: generate bill

Procedure: this function that display all of order number for customer, and after that select the order number which the administrator need to generate then display the bill.

Validation: must select customer name and order number.

Figure (3.26) generate bill

Name: - Administrator help

Description: this function provide user to view help for using web site component

Input: .click button help.

Source: web form

Output: help page.

Destination: screen

Require: none.

Precondition: none

Post condition:

Procedure: this function allow user to click on help button to display help menu screen that help the user how to use this system.

Validation: none.

Figure (3.27) administrator help

Name: Viewing the sent Comments.

Description: The administrator should be able to view comments That sent by customer, employee.

Input: None

Source: system database.

Output: Comments list and details.

Destination: Screen

Require: none

Precondition: The administrator should have been logged in the system and select to view Comments.

Post condition: Write back to customer, employee comments.

Validation: None

Figure (3.28) view comments

Name: Generate Employee reports

Description: this function allows administrator to generate employee reports

Input: employee no.

Source: web form.

Output: generate employee report

Destination: system database

Require: .employee profile, total money was paid to employee, works employee

Precondition: none
Post condition: printing employee report.

Procedure: this function request to click on button to access the database and the choose employee name from dropdown to display all information of employee (name, start working, end working, number of hour was employee working, salary, total money was paid to employee, total remainder) on screen or printer.

Validation: must be employee profile and employee working, total money was paid to employee.

Figure (3.29) generate employee report

Name: Recommendation

Description: this function provide administrator notification for new orders was mad from customer and more item sales in sessions
Input: none.

Source: system database.

Output: new orders from customer, best item.

Destination: web form

Require: customer must generate order, item was selling.

Precondition: administrator must login

Post condition: notification

Procedure: this function that display automatically when the administrator login the system then that display the new order and the best item was sales finally when the administrator click on this alarm that open new page has new order

Validation: none.

Figure (3.30) Recommendation

Name: Employee work record

Description: this function allows the administrator to enter information of employee working (start time, end time, price hour) daily.

Input: start time, end time employee ID.

Source: web form, database.

Output: new hour work for employee

Destination: system database

Require: employee must be working in company that day.

Precondition: must add employee profile

Post condition: add to database.

Procedure: this function allows administrator to register daily for each employee work after select employee name from dropdown them enter start time, end time

Validation: The system must have employee profile and end time must grater than stat time.

Figure (3.31) Employee work record

Name: Bank login

Description: the customer should be able to login the bank web site

Input: user Id and password.

Source: customer.

Output: bank e payment screen

Destination: bank system database

Require: the customer must have user ID and password to logged in shawer company system.

Precondition: the customer must have user ID and password to logged in shawer company system and select to pay by bank and must have ID and password which provided manually by bank.

Post condition: the customer should be able to pay his bill.

Validation: the customer must be entered user ID and password in valid.

Figure (3.32) bank log in

Name: Bank confirmation

Description: the customer should be confirmed by bank that the bill was paid.

Input: none.

Source: bank database.

Output: page that display all bill was paid.

Destination: screen, printer.

Require: the customer must be paid his bill by bank.

Precondition: the customer must be login the shawer company system.

Post condition: print or none.

Validation: none.

Figure (3.33) bank confiramtion

Name: Generate XML transaction

Description: the bank should generate XML transactions to Shawer Company.

Input: bank transactions.

Source: bank database.

Output: XML file.

Destination: bank hard disk.

Require: none

Precondition: not reports.

Post condition: reported

Validation: none.

Figure (3.34) generate XML transaction

Name: Reading XML files.

.

Description: shawer company should read XML files and update its.

Input: XML files.

Source: bank hard disk.

Output: confirmation.

Destination: company system database.

Require: none

Precondition: the bank should have sent XML transaction to Shawer Company.

Post condition: none

Validation: none.

Figure (3.35) read XML file

3.3 Validation Criteria:

As the system has to meet all the requirements that were previously defined and specified, we have included here the criteria by which we ensure requirements validation:

- Passwords have to be six characters at least for all user accounts.
- Passwords have not contained any space.
- No special characters would be allowed within a password string such as (hyphen, semicolon, etc).
- Login names shouldn't have numbers (0-9).
- Login name should not contain spaces.
- Login names should not contain special characters (hyphen or semicolon).
- Each user input should be checked against all constraints.
- The expression of each user input should be checked before it is accepted for further processing.
- Passwords may be the same for users but user ID should be different.

3.4 Information Description:

3.4.1 dataflow diagram :

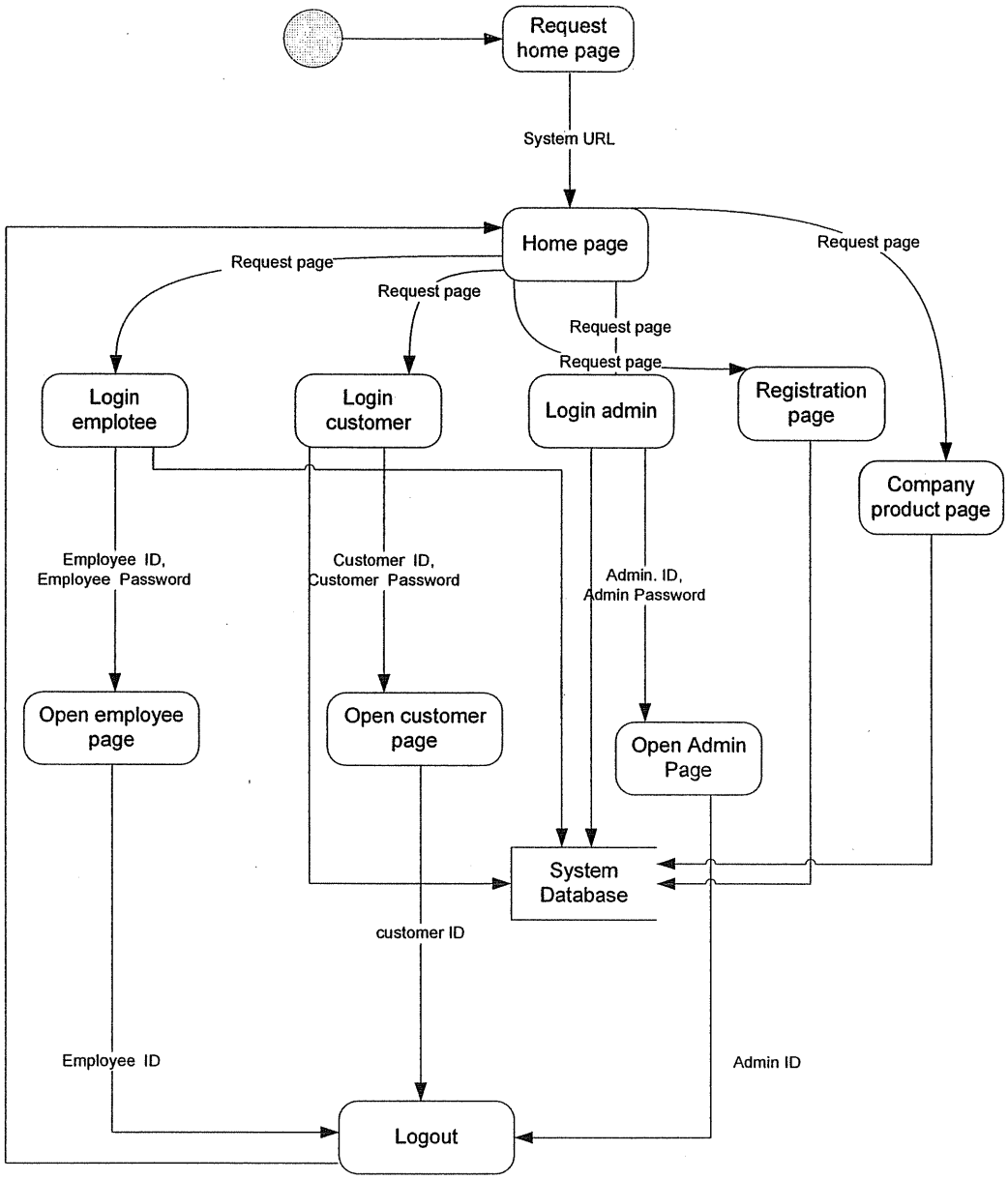


Figure (3.36) Dataflow Diagram level 0.

3.4.2 Data dictionary

Entity name	Type	Description
Viewing company product	Function	This function allows the public user to view all company product on the company website (can see products picture but can't make order) for all visitors
Guest registration	Function	This function allows the public user to register on company website and wait acceptance to register as a customer..
Sending Email	Function	This function provides guest sending e-mail to administrator about any thing he needed.
Customer login	Function	This function allows the customer to login the system by putting username and password.
Change password	Function	This function allows the customer to change his password.
Customer logout	Function	This function allows the customer to out the system.
Browsing item	Function	This function allows the customer to displaying item

		picture and knows all details of this item (product name, price, description).
enerate order	Function	This function allows the customer to purchase item by generate order.
dd select item	Function	This function enables a customer to save his selected items (products) to his order.
order after generated	Function	This function allows the customer to view report for all details about order he was generated and update quantity of item was entered.
vsing and send order after generated.	Function	This function allows the customer to view report for all selected item in order and send it to administrator.
nerate customer report	Function	This function allows the customer to display and print his report that includes (customer bills, total cost of bills, states of bill (paid or not paid). Total cost of bill was not paid, total reminder
Get feed back. From customer:	Function	This function allows the customer to send his comments and suggestions about product and anything to administrator.
Billing	Function	This function allows the

		customer to view, save and print only his bill (bill details).
Online payment using Bank website	Function	This function allows the consumer to pay his bill online using Bank website.
Viewing Employee report:	Function	This function Enable the employee to view his report that includes (number of hour he work, cost of hour, total paid, total reminder)
Get Feedback from employee	Function	This function allows employee to report comment to administrator about salary and request holidays.
Add new category	Function	Function allows adding new category with its items.
Add new item	Function	this function allows administrator to add new item store., administrator , employee)
Update item information	Function	This function allows administrator to update item sales.
Add new materials	Function	This function allow administrator to add new materials in store.
Inventory entry for materials	Function	This function allows the purchase manager to add new row materials in the store.
Store report:	Function	This function allows

		administrator to display reports for each stores that contains information about material and item available in store such as (item number, item quantity, material number, material quantity)
User profile maintenance	Function	This function allows administrator to add, delete and search about any users (customer, administrator, and employee)
Generate bill	Function	This function allows the administrator to generate bill for each order was fill fulfillment
Manufacturing item	Function	the administrator should be able to enter quantity of item was produced
Generate Employee reports	Function	This function allows administrator to generate employee reports
Administrator help	Function	this function provide user to view help for using web site component
Recommendation	Function	This function provide administrator notification for new orders was mad from customer and more item sales in sessions
Employee working	Function	This function allows the

		administrator to enter information of employee working (start time, end time, price hour) daily.
Bank login	Function	the customer should be able to login the bank web site
Bank help.	Function	The customer should get a help from bank web site.
Bank confirmation	Function	The customer should be confirmed by bank that the bills were paid.
Generate XML transaction	Function	The bank should generate XML transaction to shawer company
Reading XML files	Function	Shawer company should read the XML files and up date its
XML	Abbreviation	Extensible Markup language
SH company	Abbreviation	Shawer company

Table (3.1) data dictionary

3.4.3 System Interface description

The interface between our system components, functions, modules, and subsystems are to be reliable and integrated, that is, all functions and other components are designed and implemented in a way that ensures the correctness of collaboration between them.

Doing so requires a clear definition of the exchanged parameters and their types and orders, a reliable methodology for dealing with shared memory along with the input/output resources, and other issues that could appear during system design and system testing phases.

According to the user interface, the System will have an interface with three user categories; we have previously classified them as customers, employee, and administrators.
In this section we describe the system interface with each user category:

a) Customer's Interface :

- Input:

  The interface between customer and the system is to enable Registered Customer to Purchase a full Demand and it is controlled to ensure that he entered parameters of the same type as they was identified during system development by using validation techniques available, the customer will be permitted to input just valid parameters type within an allowed range of values and he can be write comment or suggestion to administrators .

- Output:

  The output for customers must give them a sense of what is happening in each process, as they choose a certain product; they have to see this product added to their purchasing order which will accompany them in their logging tour in the system.

  Further more customers should be provided with a final report after completing purchasing, this report shows all products that were purchased by them with their quantities, prices, product name, and total value of bill.

- GUI

The graphical user interface must provide some what guidance marks that guide customers to the next step to be done and must be easy and attractive for users.

All what we can say here, that GUI must be developed to facilitate customers to investigate easily and purchase securely.

b) Employee

- Input:

The system must have an easy and reliable interface with Employees, so that they should be able to ask for a vocation, ask for increase in their salaries …etc

Providing areas of input with validation techniques that runs on both Employees and the web server will make the interface reliable.

- Output:

Employees must be notified with Special Reports or a page for the company announcement.

- GUI:

The screens that Employees use are to be smooth and meaningful.

c) Administrator's Interface:

- Input:

There are more than one type of administrators , an administrator for Employees who record the Entrance and leave hours for the Employees , and the Merchandising Administrator who is Responsible for entering the raw materials and report the Purchases , and the Products Administrator who is responsible for Entering the Items and the New Products , and there is The General Administrator who is responsible for all tasks and have the previous administrators authorities and can access any page and anywhere and to any point and he is the one who can Acknowledge the Demand to be agreed or not agreed.

All administrators' pages must be secure, the administrator will have the accessibility to the data in database tables, and this will be done through input tools to facilitate him to modify records easily and smoothly.

Validation techniques are needed to ensure that predefined input types and ranges are not violated.

- Output:

The output for all administrator transactions will be reflected on the system database tables thereby site data contents, so no specified output is to be produced for administrators except reports which should be available.

- GUI :

An administrator should be provided with a login page and others to enable him to modify site database records and site data contents; the interface design is described precisely in (System Design).

3.4.4 Database requirement

System database will include the following table's fields:

1) Employee.

- Employee ID: A unique number of the employee.
- Password: A unique characters for each employee, these characters should encrypt in some encryption methods to increase the level of security.
- Employee name: The name of the employee.
- Hour Price: price of hour for each employee.
- Email: The email of the employee.
- Tel: The phone number of the employee.
- City No employee city number.
- ID: Employee identification card number (haweyah)..
- Address: The employee address.
- Dept No: A unique number of the department.
- Job no: job descriptions

2) Employee salary

- Employee Sal No: sequential number.
- Employee ID: A unique number of the employee.
- Total Paid total money paid for employee.
- Payment Date: date of payment.

3) Employee work.

- Employee ID: A unique number of the employee
- Current Day: the day of working employee.
- Start time: the time when employee start to working in day.
- End time: the time when employee stop working in day.
- No Hours: total of hour for each employee working daily

4) Employee comment:

- Comment No: the number of the comment
- Comment content: the contents of the comment
- Employee ID: A unique number of the employee
- Employee Email: The e-mail of the employee who reported the comment

5) Department

- Dept No: the number of the department (sequential number)
- Dept Name: The name of the department

6) City

- City No: city number.
- City Name: **:** name of city

7) Guest

- Guest ID: A unique number of the guest.
- Guest Name: The name of the guest
- Company Name: the name of the guest company
- City No: guest city number
- Tel: The phone number of the guest
- Fax: guest or company fax
- Email: The email of the guest
- Address: The guest address

8) Customer

- Customer ID: A unique number of the customer
- Password: A unique characters for each customer, these characters should encrypt in some encryption methods to increase the level of security.
- Customer Name: The name of the customer
- Company Name: the name of the customer company
- City no: customer city number
- Tel: The phone number of the customer
- Fax: customer or company fax
- Email: The email of the customer
- Address: The customer address

9) Customer comment

- Comment No: the number of the comment
- Comment content: the contents of the comment
- Customer ID: A unique number of the customer
- customer Email: The e-mail of the customer who reported the comment

10) Order

- Order No: the number of order.
- Customer ID:A unique number of the customer
- Order Date: the date of is generated
- Confirm: the number that detriment the order condition

11) Orders.

- Order No: the number of order
- Item No: number of item

- Quantity: quantity of item in order.
- Orders Cost: total cost of each item
- Description: item Description.
- Unit Sails: price of one unit.
- Status: the number that detriment quantity of item are exist in company store or no.

12) Bill

- Bill No: The sequence number of the generated bill, each bill has a unique number.
- Customer ID: A unique number of the customer
- Order No: the number of order
- Total cost: total cost of bill.
- Gen date: generating bill date.

13) Status.

- Status No: A unique No for each status.
- Status Name: A unique name for each status such as paid or unpaid.

14) Manufacture Item.

- Manufacture Item No: sequential number
- Item No: number of item
- Quantity: total quantity for each item.
- Material No: number of material.

15) Item.

- Item No : number of item
- Category No: number of category that is available in the store.
- Item Name: name of item.
- Total Quantity: the total quantity for each item in the store.

- Picture: image path of item.
- Item Quantity Materials: quantity of material that consumed to produce one item
- Unit cost: production cost for each item.
- Unit sales: price sales of one item.
- Description: item Description

16) Material

- Material No: number of each material.
- Material Name: name of material.
- Category No: number of category that is available in the store.
- Total Material: the total quantity for each material. in the store

17) Add Material.

- Number: sequential number
- Material No: : number of each material
- Total Roll: total roll that input to store
- Length: the length of each roll.
- Width: the width of each roll
- Description: roll Description
- Enter Date: date of input quantity roll to store.

18) Category.

- Category No: number of category that is available in the store.
- Category Name: name of category.

19) Advertisement.

- Advertisement No: A unique number for each advertisement.
- Advertisement Title: The title of the advertisement

- Description: Contents of the advertisement
- Expiration Date: The date after which the advertisement mustn't be viewed
- Viewing Date: The date on which the advertisement must be viewed

22 Bank's database Tables

1. Bank Login:

- Customer bank ID: ID that identifies the customer which is given by bank
- Password: customer password which is given by bank (unique for each customer)

2. Bank accounts:

- Customer bank ID: ID that identifies the customer which is given by bank
- Bank account number: the number of the bank account for the customer
- Name: the name of the bank agent
- Account value: the amount of money that the customer has in his account.

3. Bank transaction

- Transaction Number: each transaction has a unique number.
- Customer ID: A unique number for each customer in shawer company
- Bill number: number of the invoice
- Payment date: the date when the customer paid the bill.
- Money paid: amount of money that is paid.
- Bank account number: the number of the bank account for the cust

3.5.5 System Context Model:

This diagram explains relationship in abstract between the system and other environment systems without details.

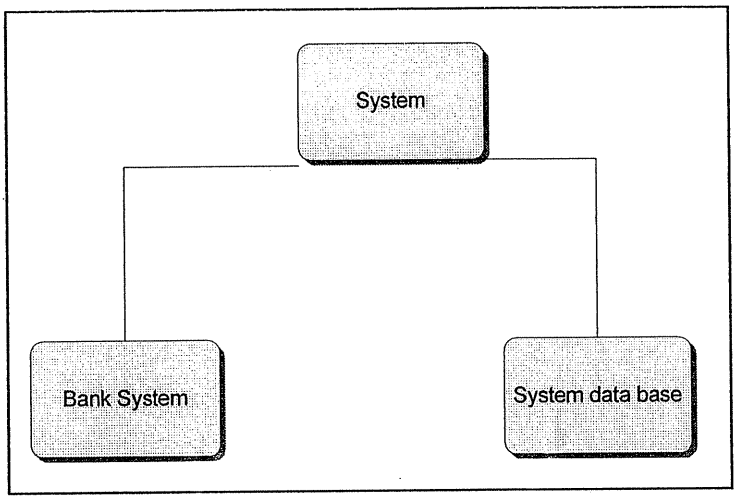


Figure (3.37) System Contexts Diagram

3.5 Summary and Recommendation

After understanding functional and database requirements in details which is approved by the supervisor the team will continue working in the next phase which is system design

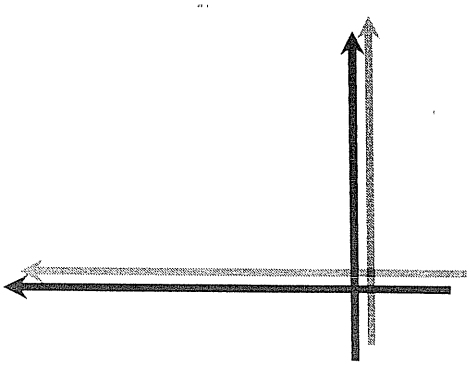
Chapter four

4

Design

Chapter four

- Introduction
- Functional design
- Input/output design
- Database dictionary
- Database model
- Test plan model
- Programming language and coding
- Summary and recommendation



4.1 Introduction

This chapter describes the system design, which including the following topics:

1. Functional design.
2. I/O Design.
3. Database Design.
4. Test plan.
5. Programming Language and Coding.

4.2 Functional design

This section describes the functional design for each module in the software system, accordingly with the description of the interface, the constraints, and the user interface design in which we use means of diagramming to help us understand it:

1) Viewing company product:

a) Description: allows the public user to view advertisement on the company website (can see products picture but can't make order) for any visitor.

b) Interface:

- Inputs: click on button "منتجاتنا"
- Outputs: advertisement page that include (item picture, item name)

c) Constraints:

- None.

d) User interface design:

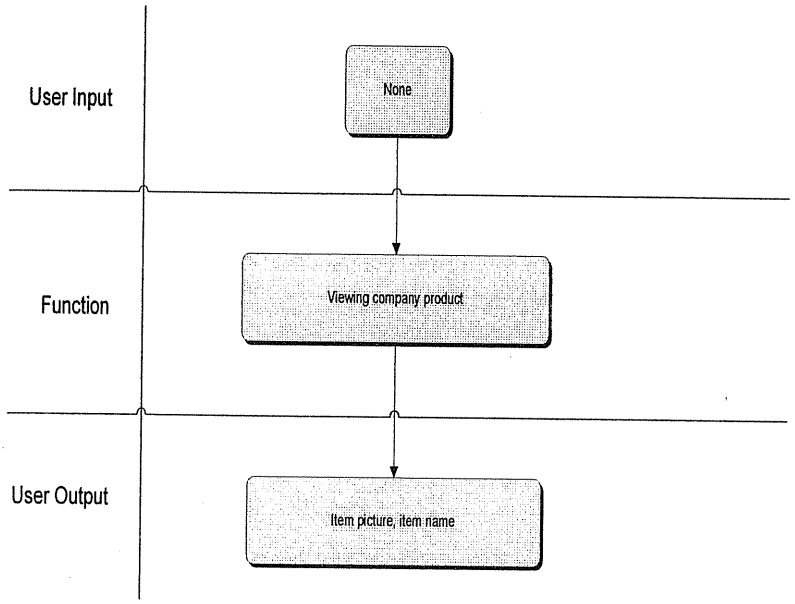


Figure (4.1) viewing company product interface design

e) Flowchart:

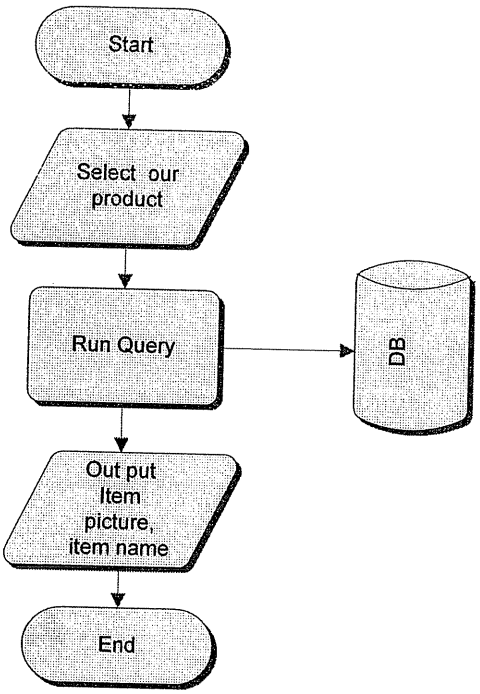


Figure (4.2) viewing company product operation.

2) Guest registration.

a) Description: this function allows the public user to register on company website and wait acceptance to register as a customer.

b) Interface:

- Inputs: name, company name, address, tell, fax, city, e mail
- Outputs: none

c) Constraints:

- None.

d) User interface design:

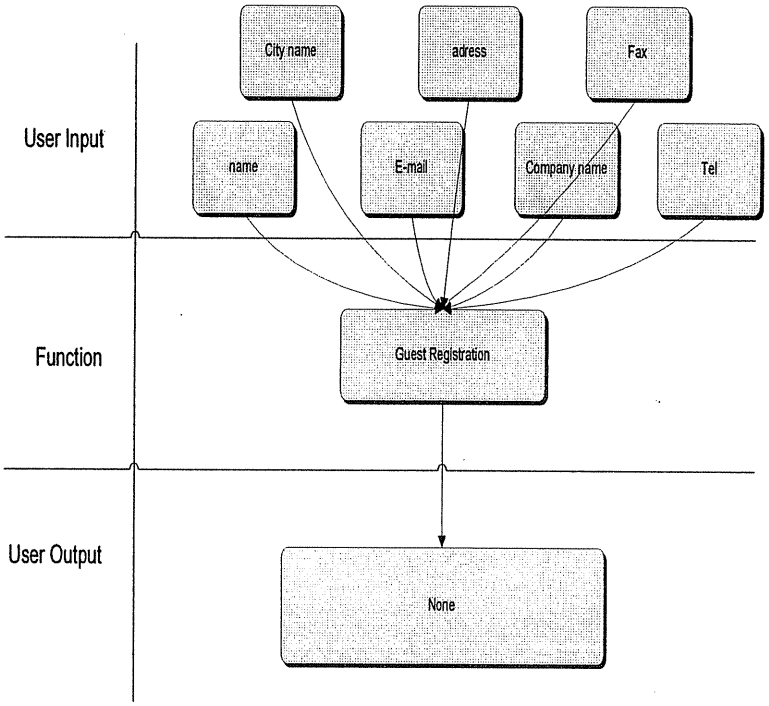


Figure (4.3) guest registration interface design

e) Flowchart:

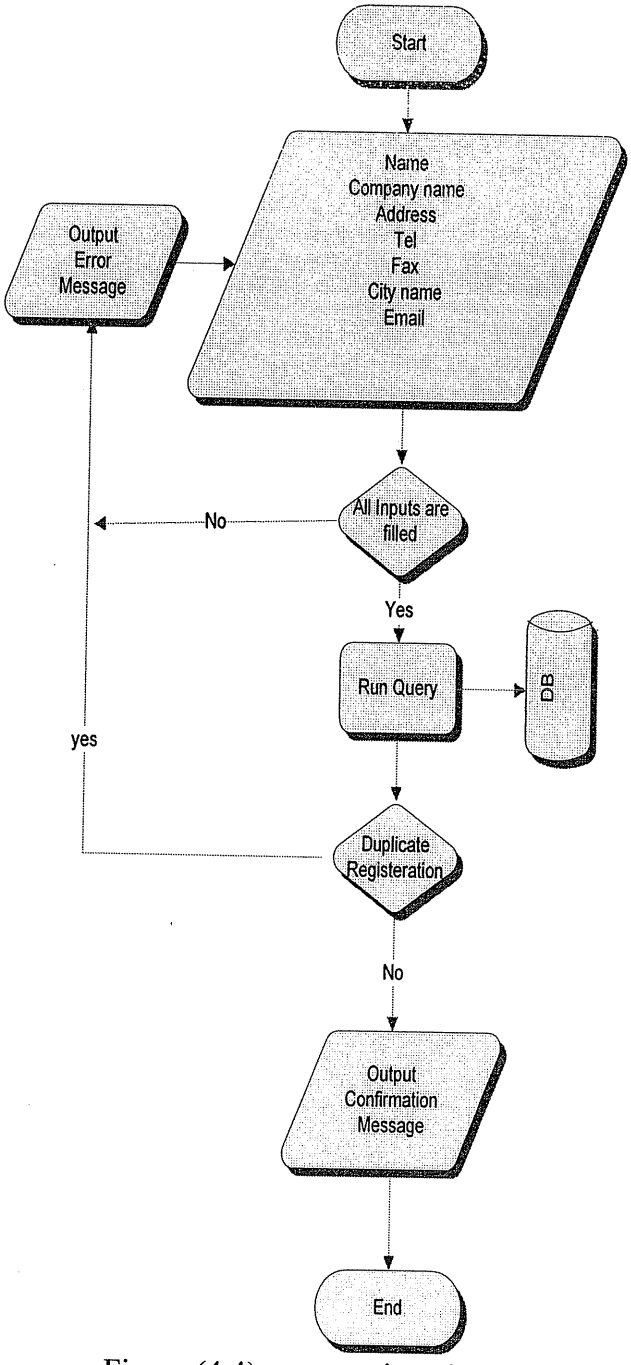


Figure (4.4) guest registration operation

3) Sending Email.

a) Description: this function provides guest sending e-mail to administrator about any thing he needed.

b) Interface:

- Inputs: Click a button to "e-mail".
- Outputs: Emails send to administrator email account.

c) Constraints:

- None

d) User interface design:

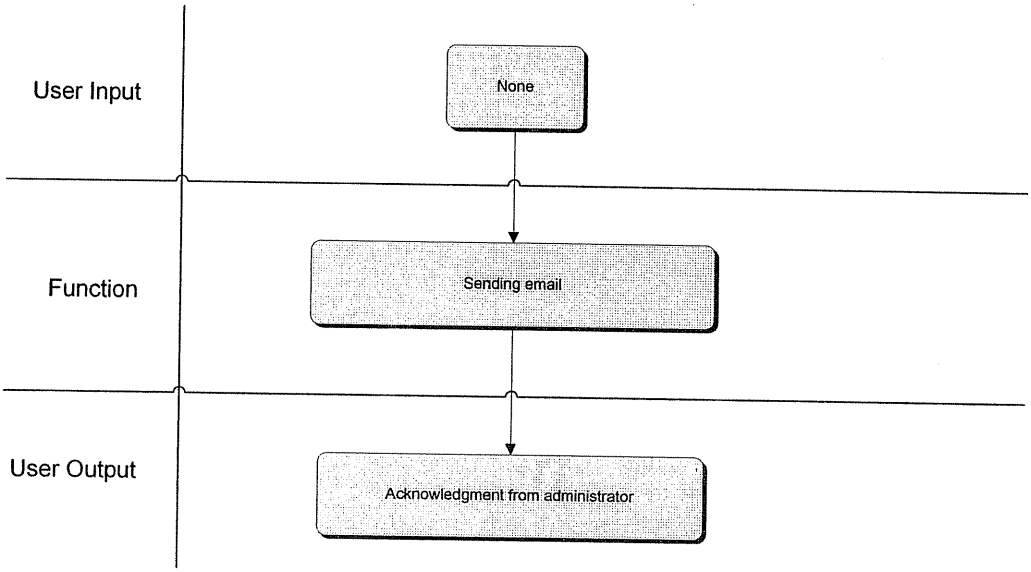


Figure (4.5) sending email interface design

e) Flowchart:

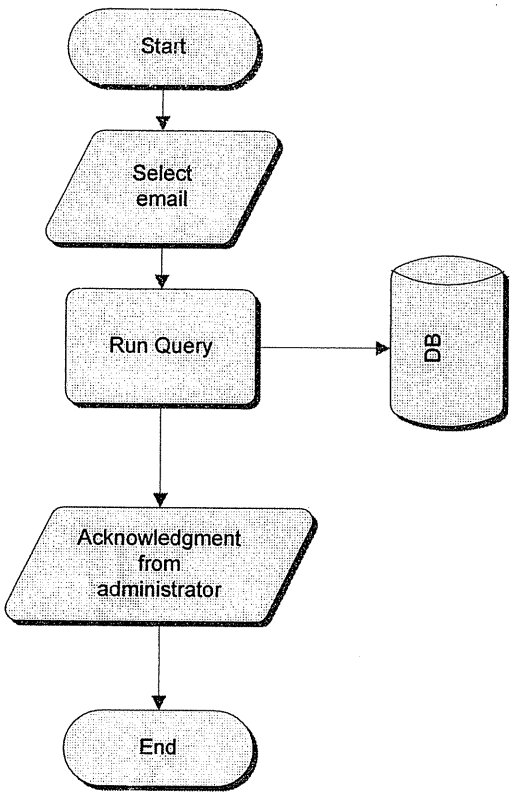


Figure (4.6) sending email operation

4) Customer login

a) Description: this function allows the customer to login the system by putting username and password...

b) Interface:

- Inputs: customer Id, password.
- Outputs: customer home page.

c) Constraints:

- User ID and password must be filled in.
- User ID and password must been registered in SH company database

d) User interface design:

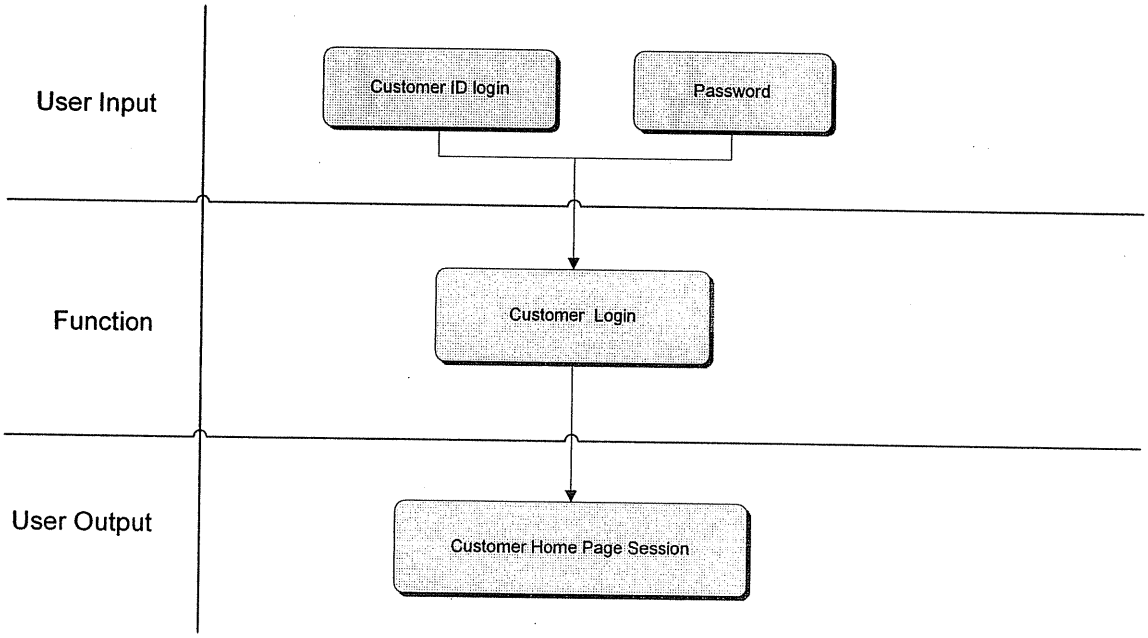


Figure (4.7) customer login interface design

e) Flowchart:

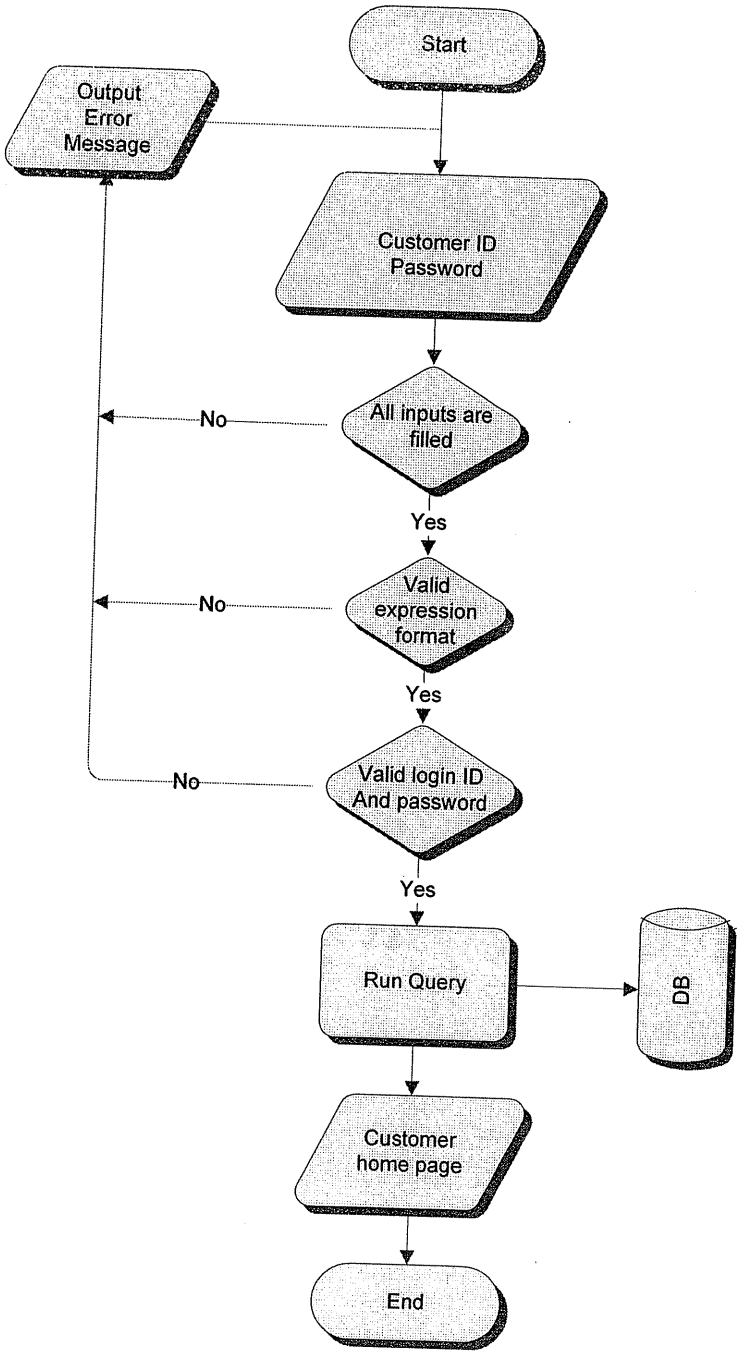


Figure (4.8) customer login operation

5) Change password

a) Description: this function allows the customer to change his password

b) Interface:

- Inputs: old password, new password, password confirm
- Outputs: : New customer password.

c) Constraints:

- New password and its confirmation must match.
- The new password will take place at the next login and after insert old password

d) User interface design:

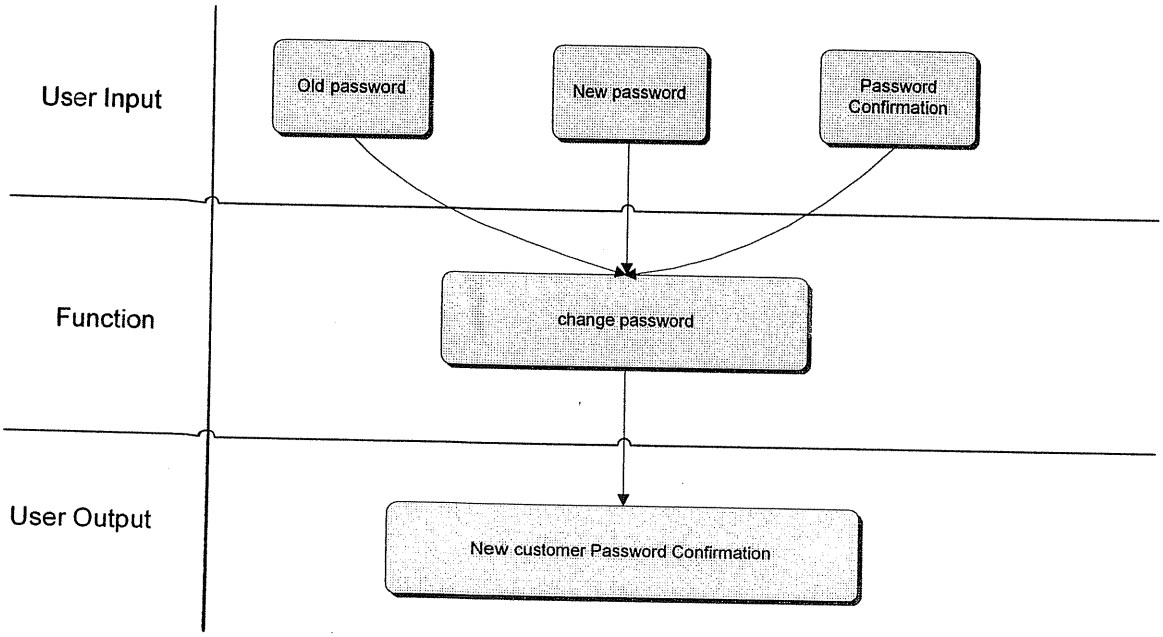
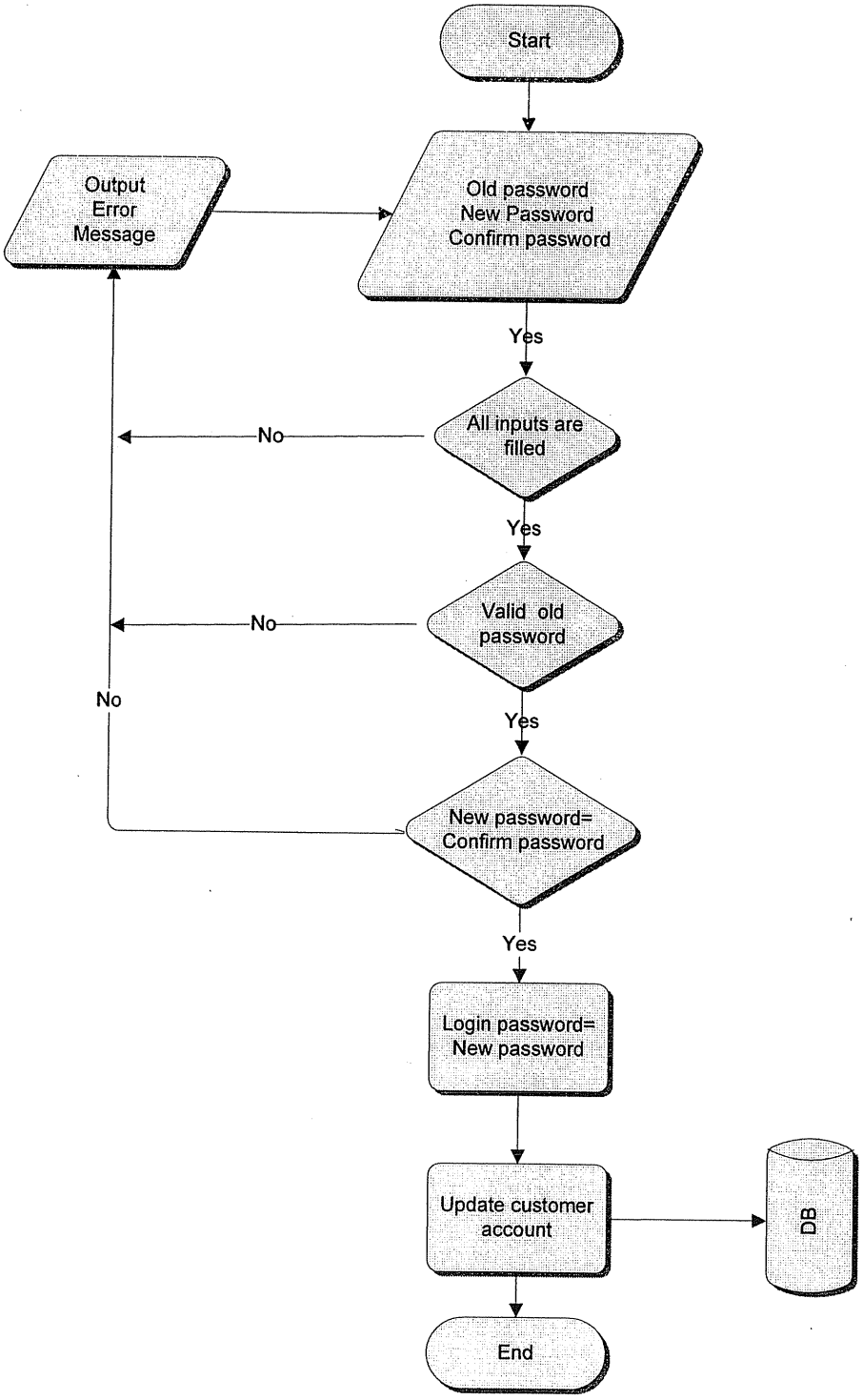


Figure (4.9) change password interface design

e) Flowchart:



Figure(4.10) change password operation

6) Customer logout

a) Description: this function allows the customer to out the system.

b) Interface:

- Inputs: click on button "تسجيل خروج"
- Outputs: Home page.

c) Constraints:

- none

d) User interface design:

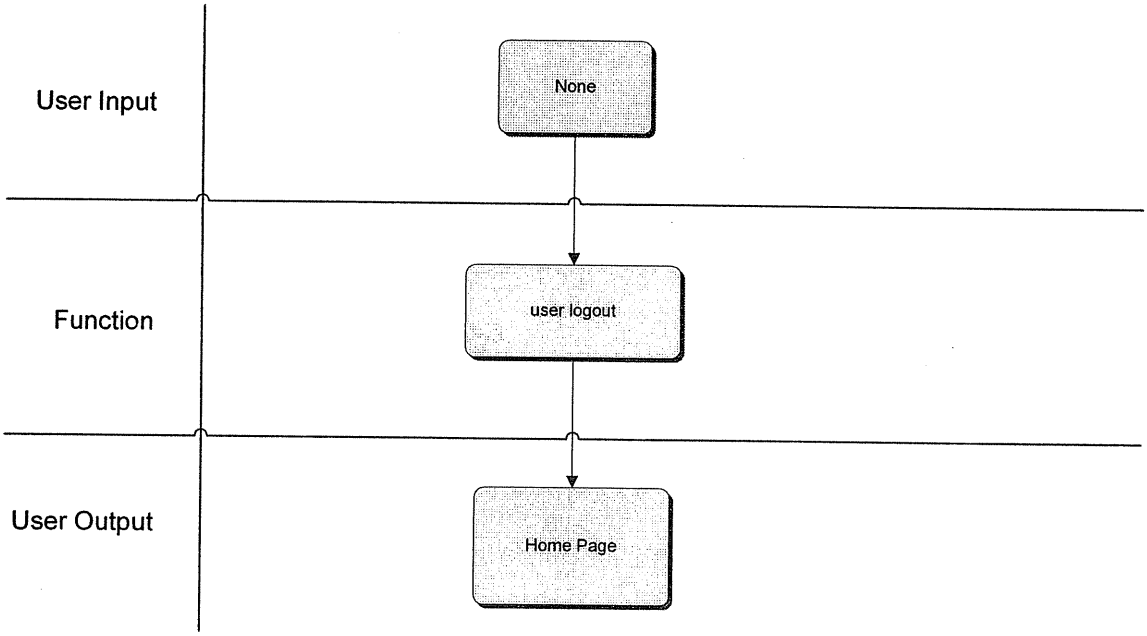


Figure (4.11) customer logout interface design

e) flowchart

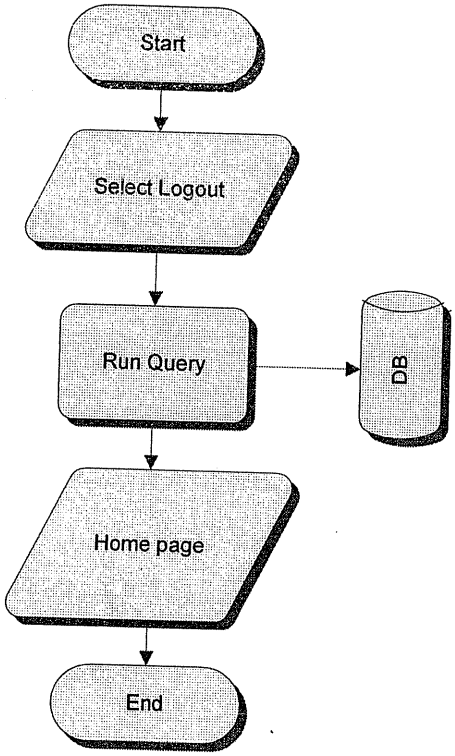


Figure (4.12)) customer logout operation

7) Browsing item.

a) Description: this function allows the customer to brows item information.

b) Interface:

- Inputs: Click on button "عرض البضائع"
- Outputs: item picture, item name, price, description.

c) Constraints:

- The customer must be logged to the system and generate order.

d) User interface design:

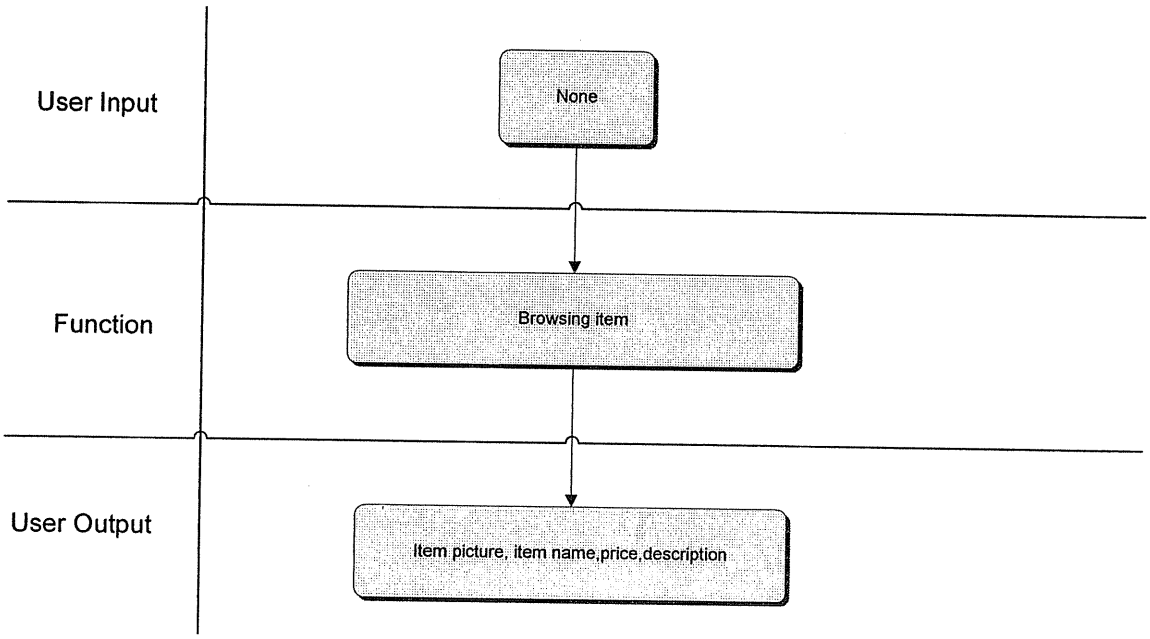


Figure (4.13) browsing item interface design

e) Flowchart:

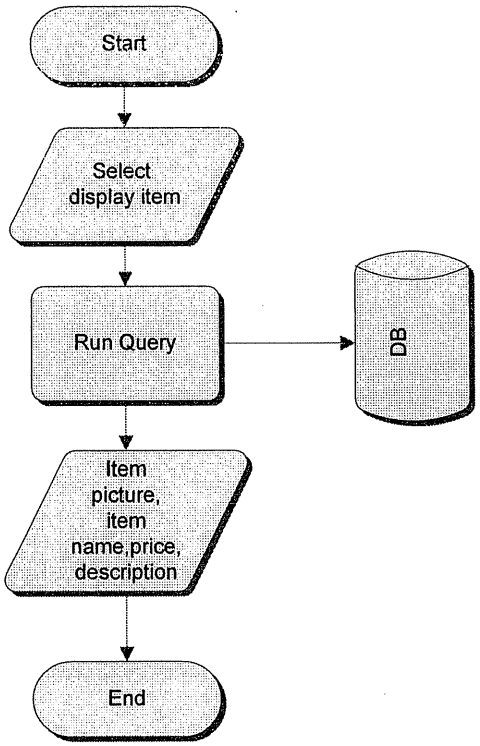


Figure (4.14) browsing item operation

8) Generate order

a) Description: this function allows the customer to purchase item by generate order.

b) Interface:

- Inputs: order date, order number, customer ID
- Outputs: create new record in database.

c) Constraints:

- The customer must be log in the system

d) User interface design:

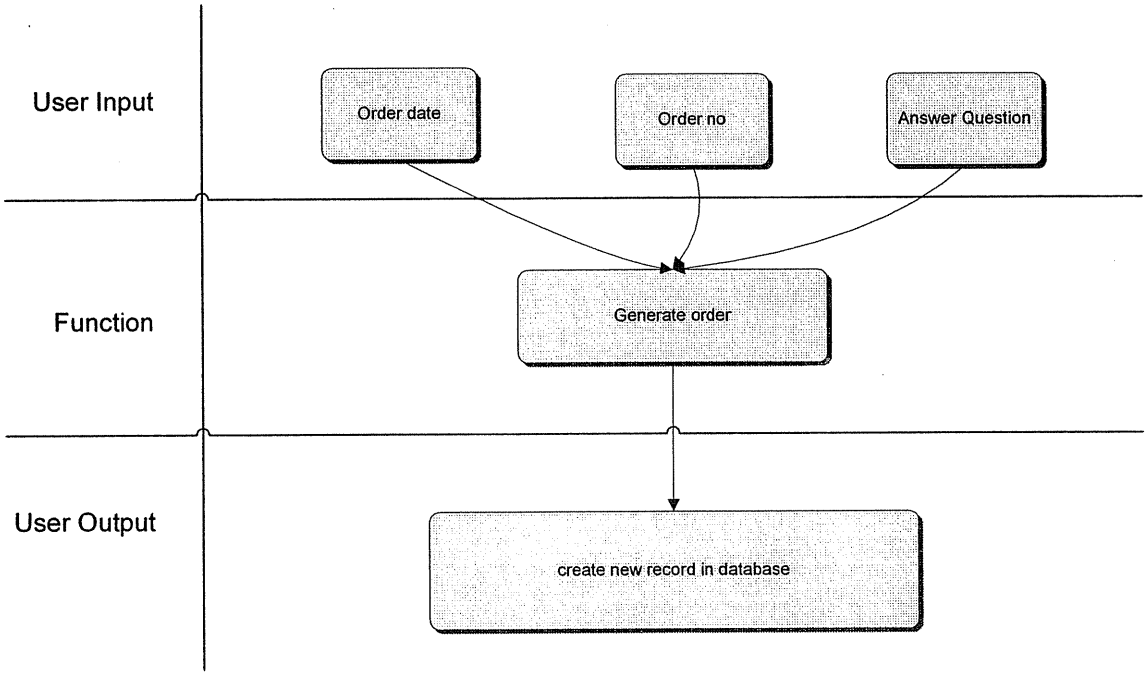
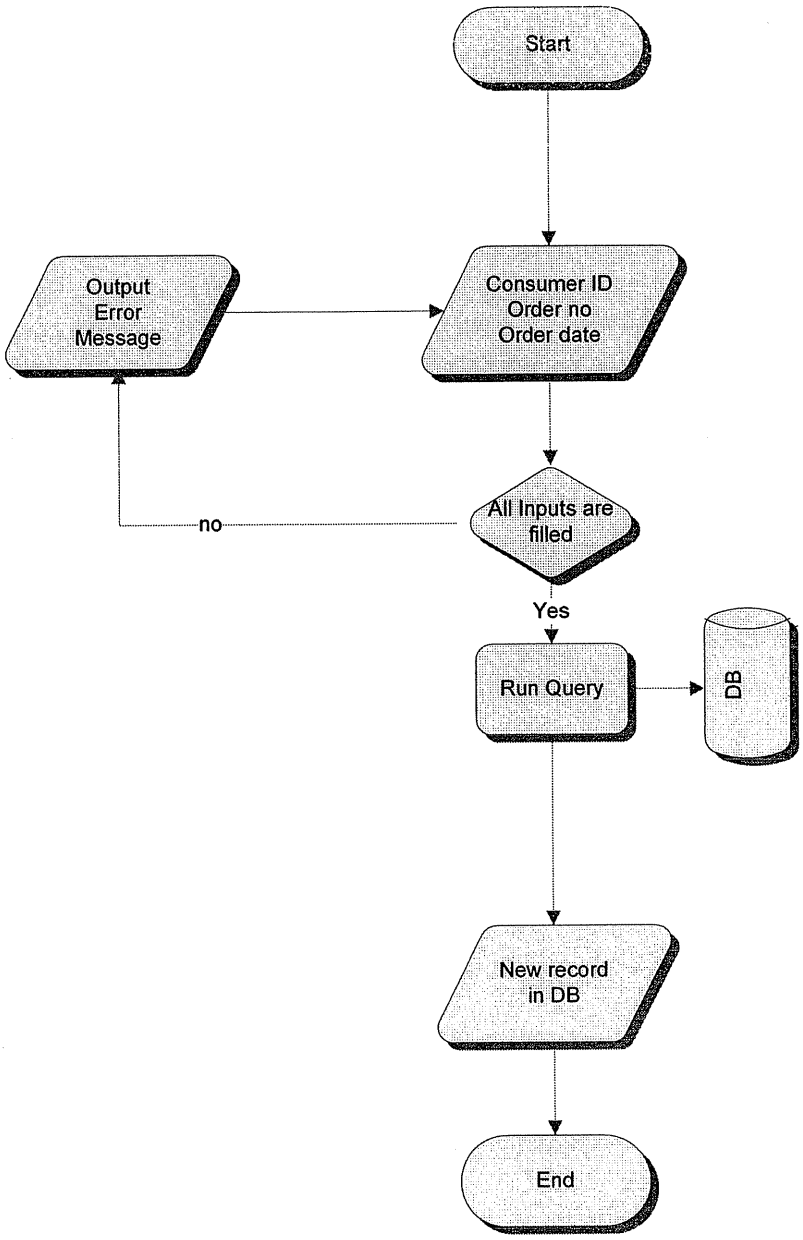


Figure (4.15) generate order interface design

e) Flowchart:



Figure(4.16) generate order operation

9) add select item

a) Description: this function enables a customer to save his selected items (products) to his order.

b) Interface:

- Inputs: item name, quantity.
- Outputs: new product is added to the order.

c) Constraints:

- The customer must be log in the system

d) User interface design:

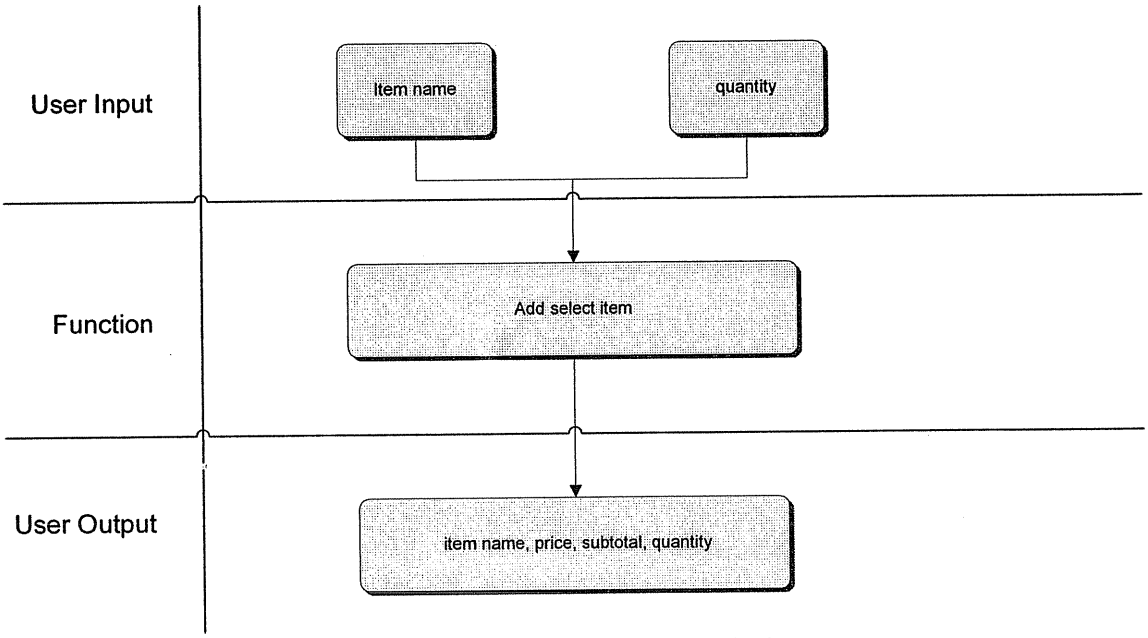
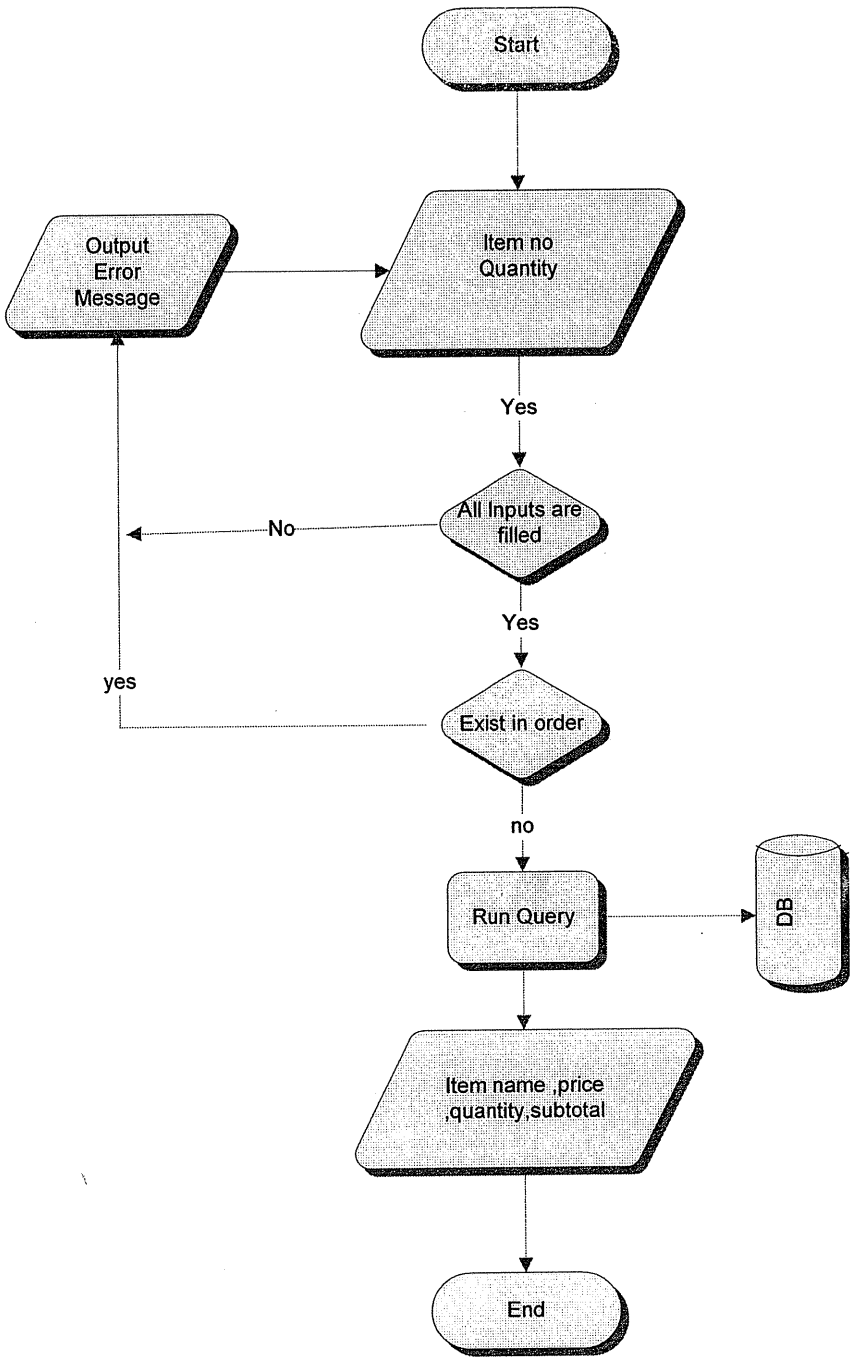


Figure (4.17) add select item interface design.

e) Flowchart:



Figure(4.18) add select item operation

10) Update order after generated.

a) Description: this function allows the customer to view report for all details about order he was generated and update quantity of item was entered.

b) Interface:

- Inputs: Click on button that name "تعديل"
- Outputs: update record (update quantity of item.).

c) Constraints:

- The customer must be log in the system

d) User interface design:

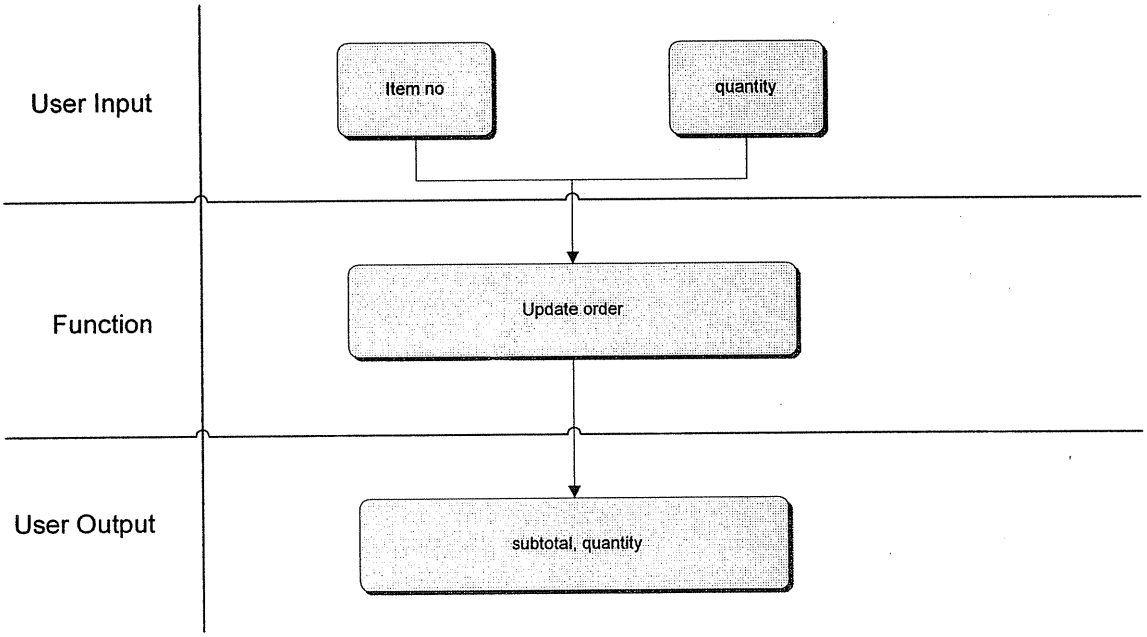


Figure (4.19) update order after generated interface design

f) Flowchart:

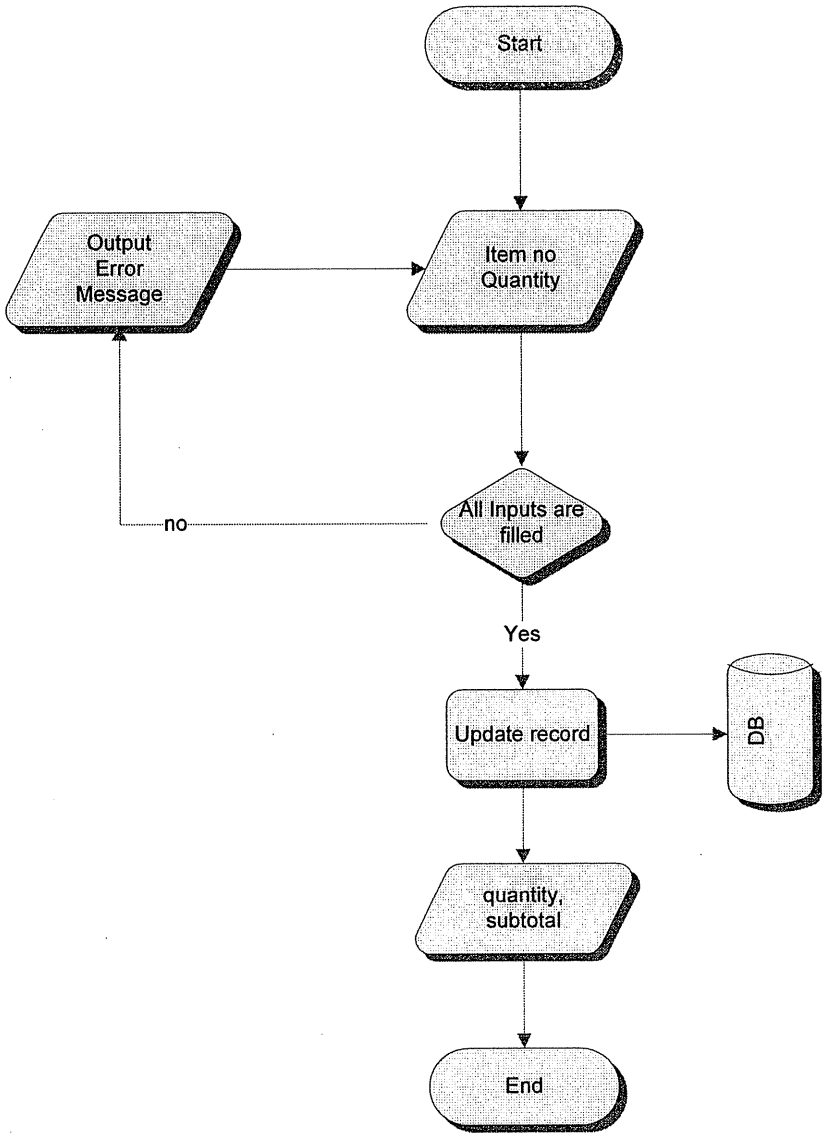


Figure (4.20) update order after generated operation

11) Browsing and send order after generated.

a) Description: this function allows the customer to view report for all selected item in order and send it to administrator

b) Interface:

- Inputs: Click on button عرض البضائع المختارة

- Outputs: items name, quantity, unit sales, description, subtotal.

c) Constraints:

- The order must be generated and the customer must select item

d) User interface design:

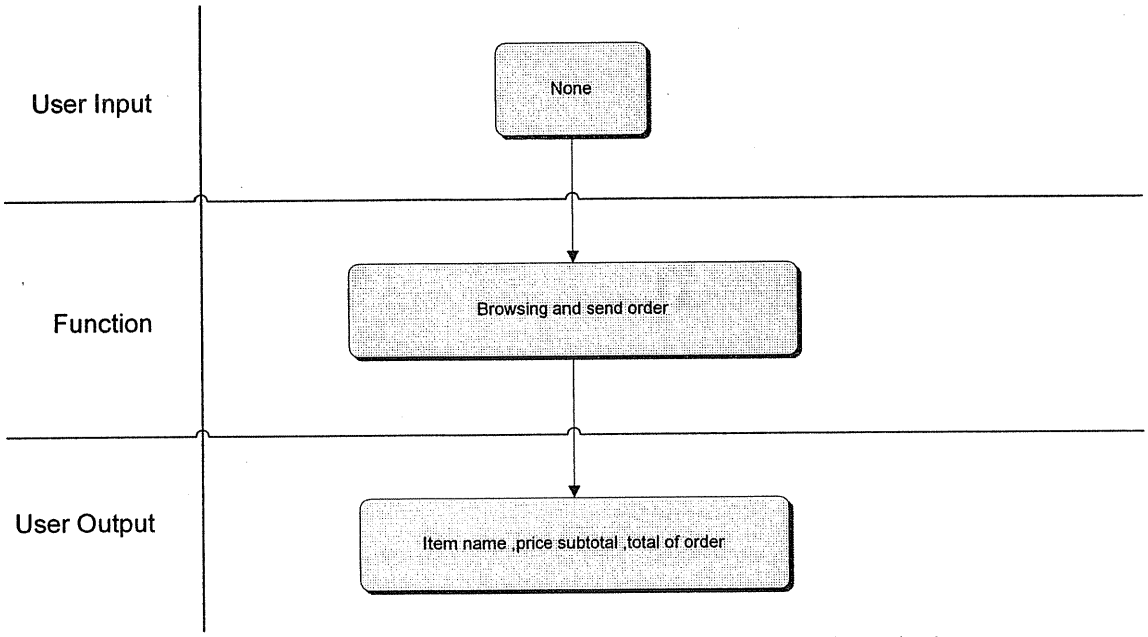


Figure (4.21) browsing and send order after generated interface design

e) Flowchart:

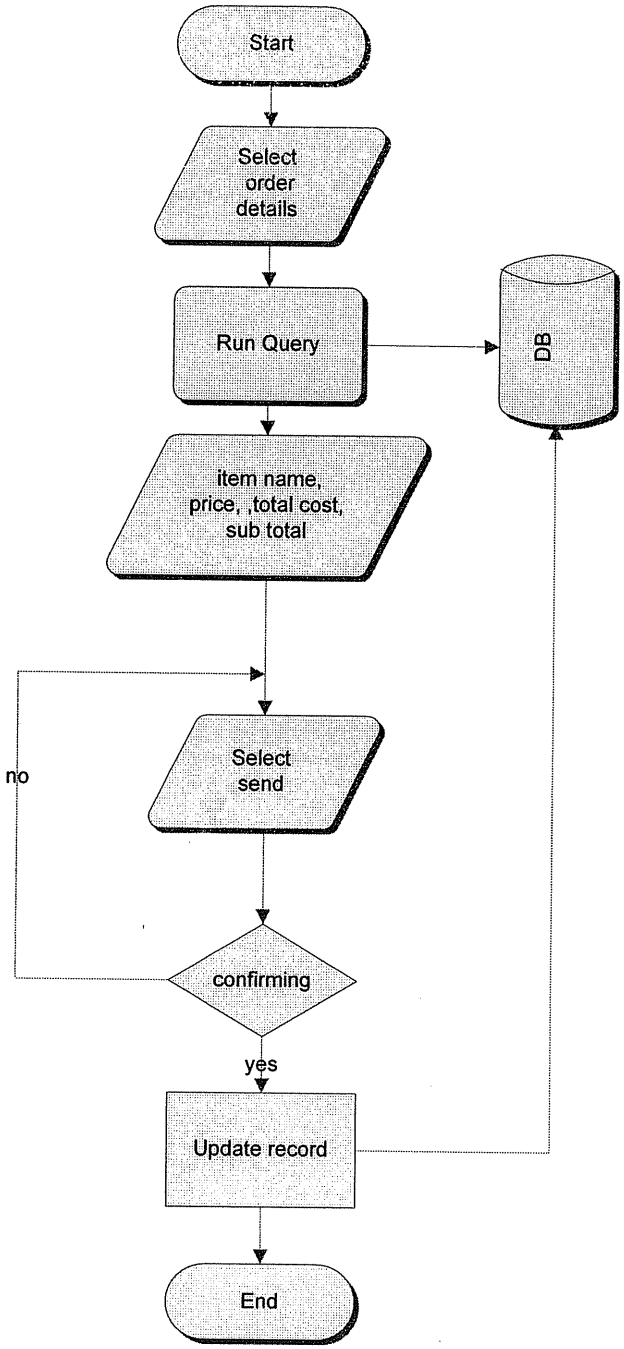


Figure (4.22) browsing and send order after generated operation

12) Generate customer report.

a) Description: this function allows the customer to display and print his report that includes (customer bills, total cost of bills, states of bill (paid or not paid). Total cost of bill was not paid, total reminder

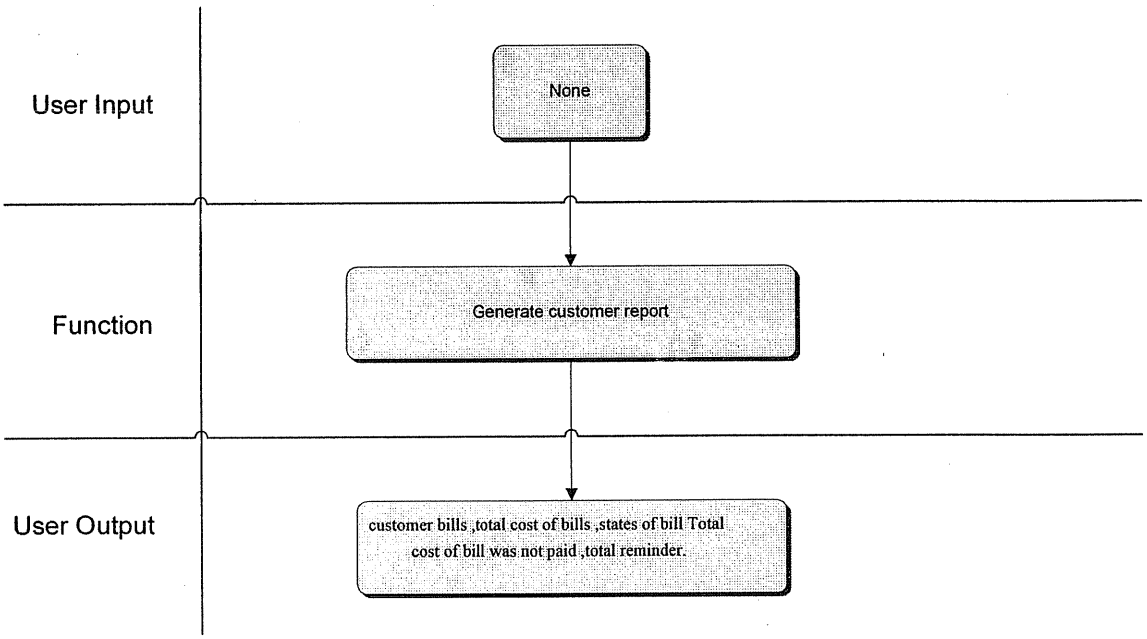
b) Interface:

- Inputs: Click on button "تقرير"
- Outputs: customer bill, total cost of bills, states of bill Total cost of bill was not paid, total reminder.

c) Constraints:

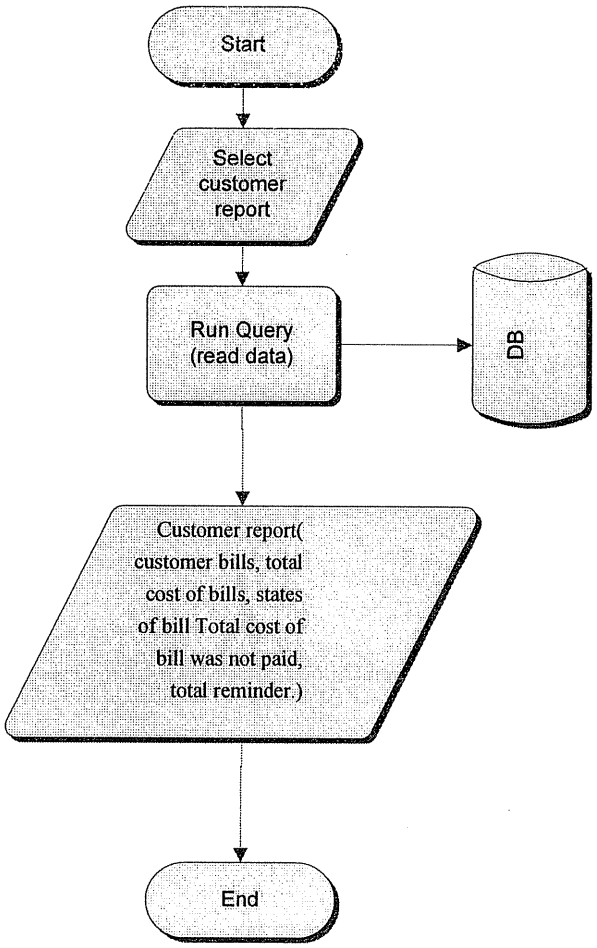
- There must be customer profile, and customer bills.

d) User interface design:



Figure(4.23) generate customer report interface design

e) Flowchart:



Figure(4.24) generate customer report operation

13) Get feedbacks

a) Description: this function allow public user to report comment about product and feedback..

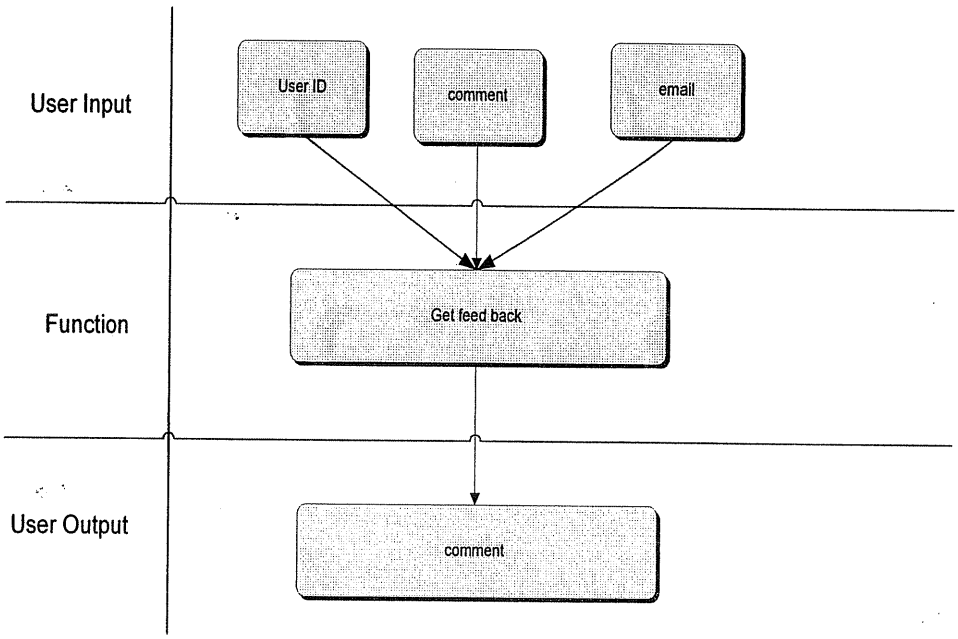
b) Interface:

- Inputs: comment, email.
- Outputs: create new record in DB.

c) Constraints:

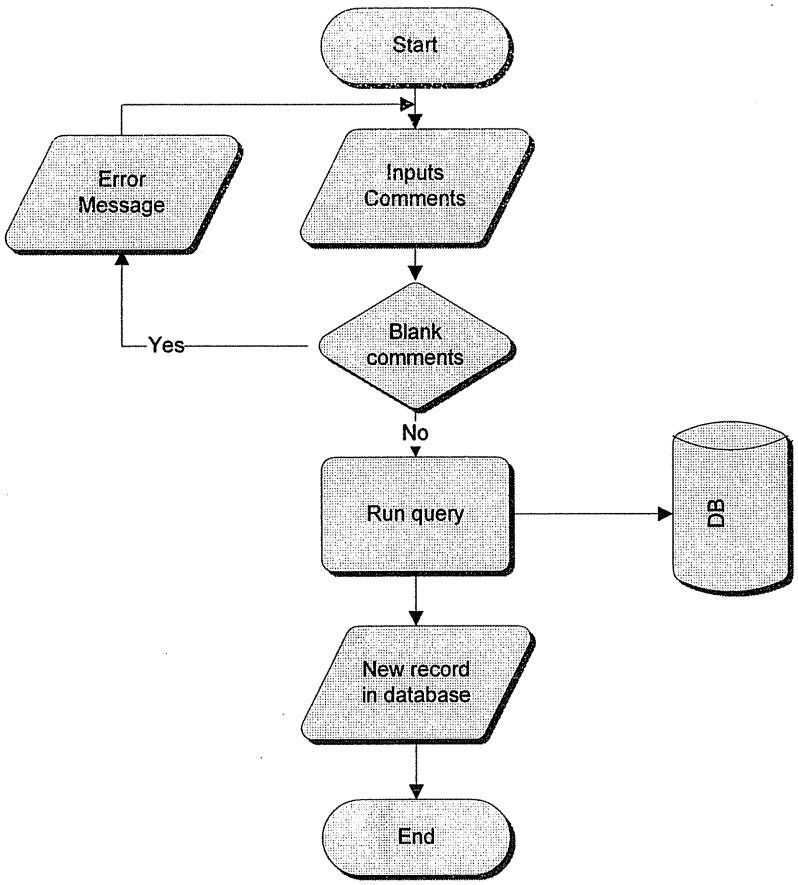
- message mustn't be blank The new
- public user must provide his email.

d) User interface design:



Figure(4.25) get feedback interface design

e) Flowchart:



Figure(4.26) get feedback operation

14) billing

a) Description: this function allows the consumer to view, save and print of bill

b) Interface:

- Inputs: click on button "تفاصيل"
- Outputs: bill details (item name, price, subtotal, quantity, total cost)

c) Constraints:

- Bill must be generated by administrator

d) User interface design:

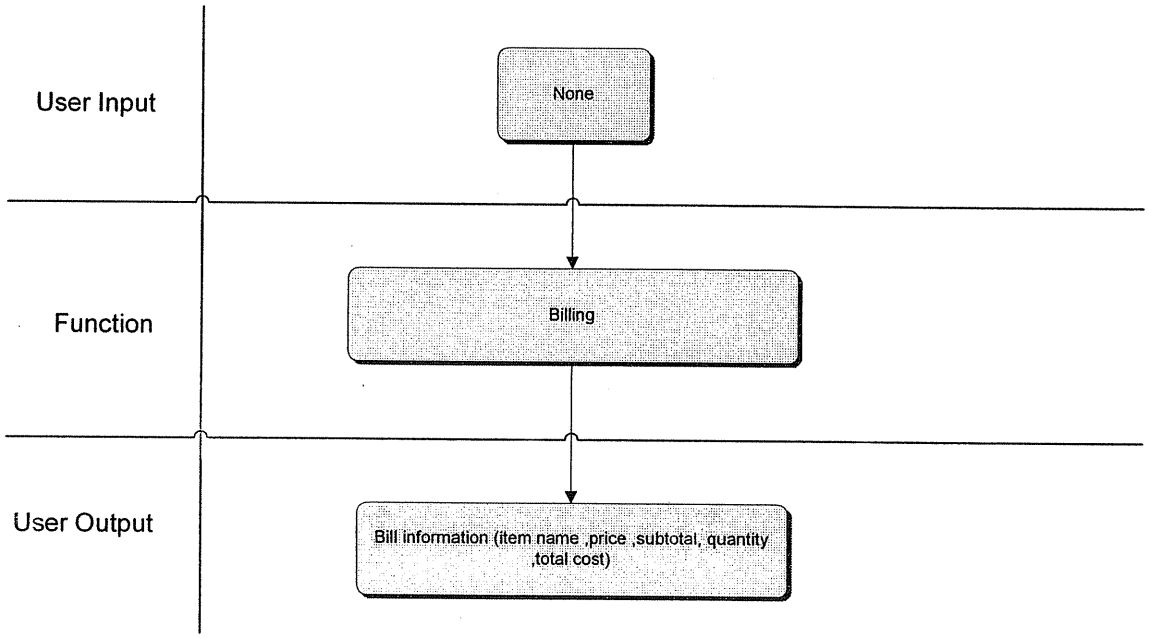


Figure (4.27) billing interface design

e) Flowchart:

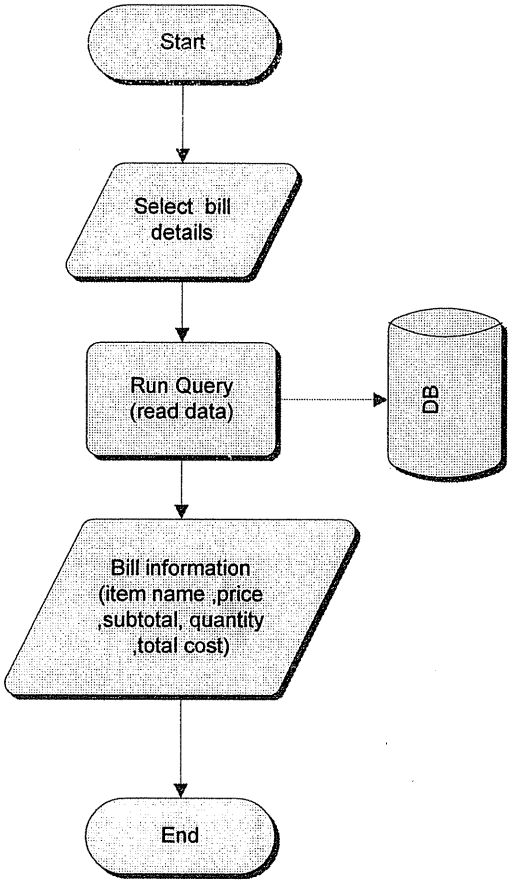


Figure (4.28) billing operation

15) Online payment using Bank web service

a) Description: this function allows the consumer to pay his bill online using Bank web service.

b) Interface:

- Inputs: account_no, bill_no, customer_ID, amount of money to be paid.
- Outputs: payment confirmation.

c) Constraints:

All inputs must be filled in and the registered user(customer) must have an account grater than or equal to bill value.

d) User interface design:

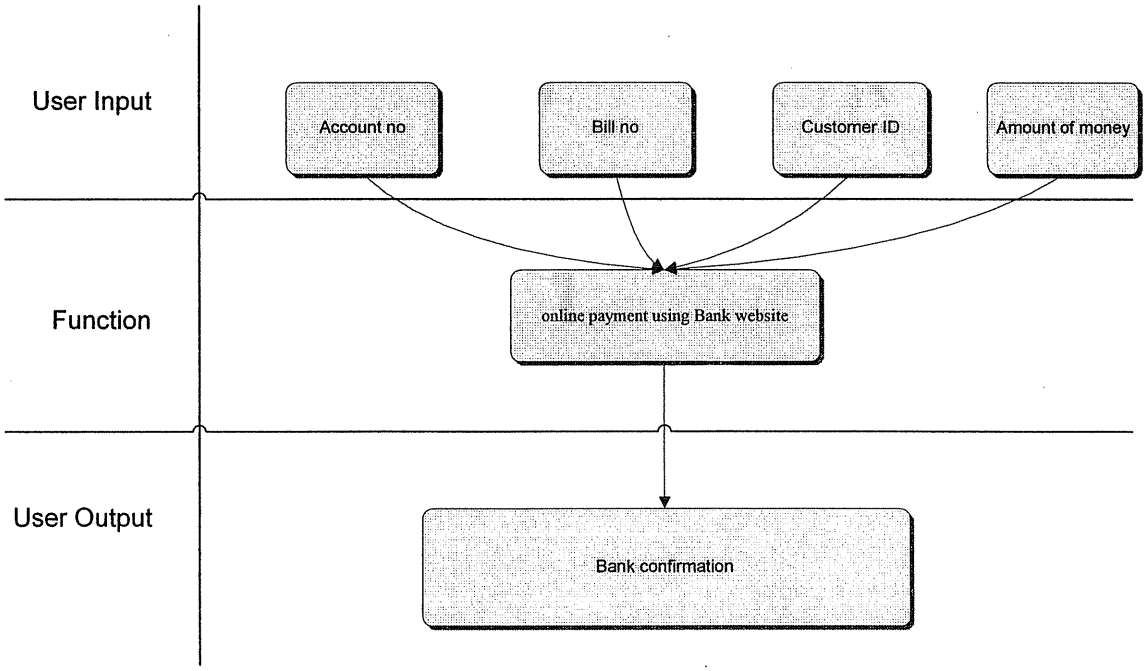


Figure (4.29) online payment using bank web service interface design

e) Flowchart:

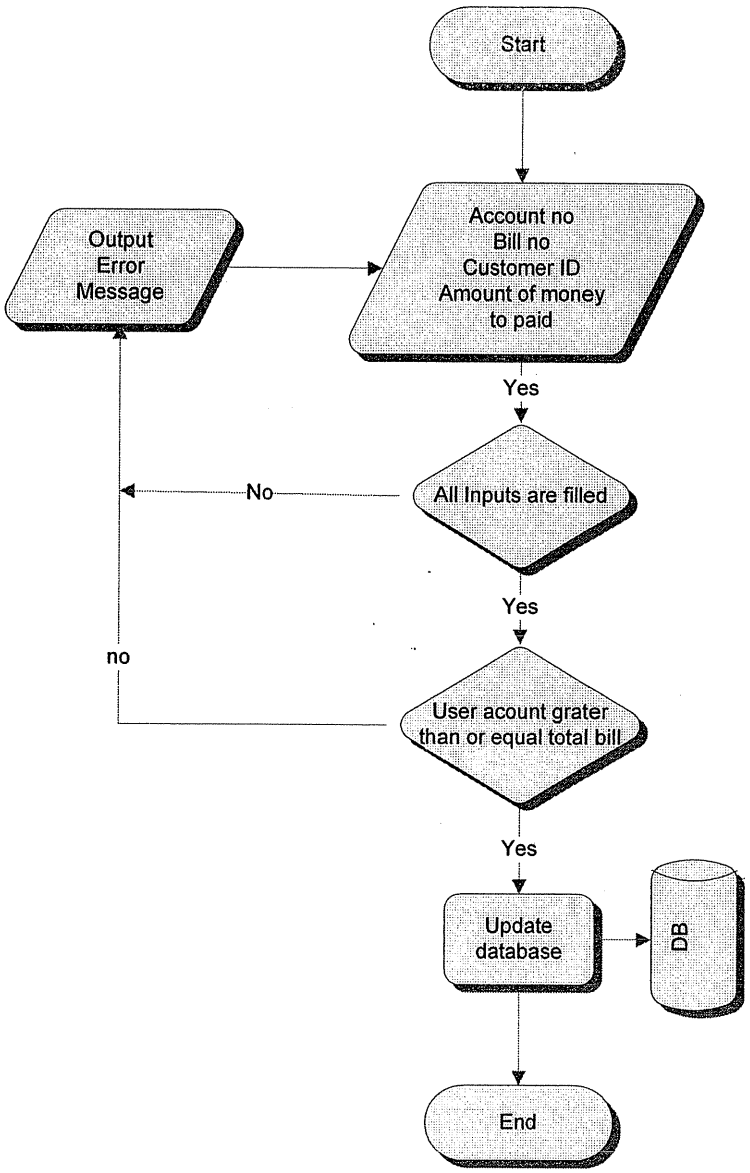


Figure (4.30) payment online operation

16) Add new category.

a) Description: This function allow administrator to adding new category with its items.

b) Interface:

- Inputs: category no, category name.
- Outputs: new record in DB.

c) Constraints:

- All Inputs are filled.

d) User interface design:

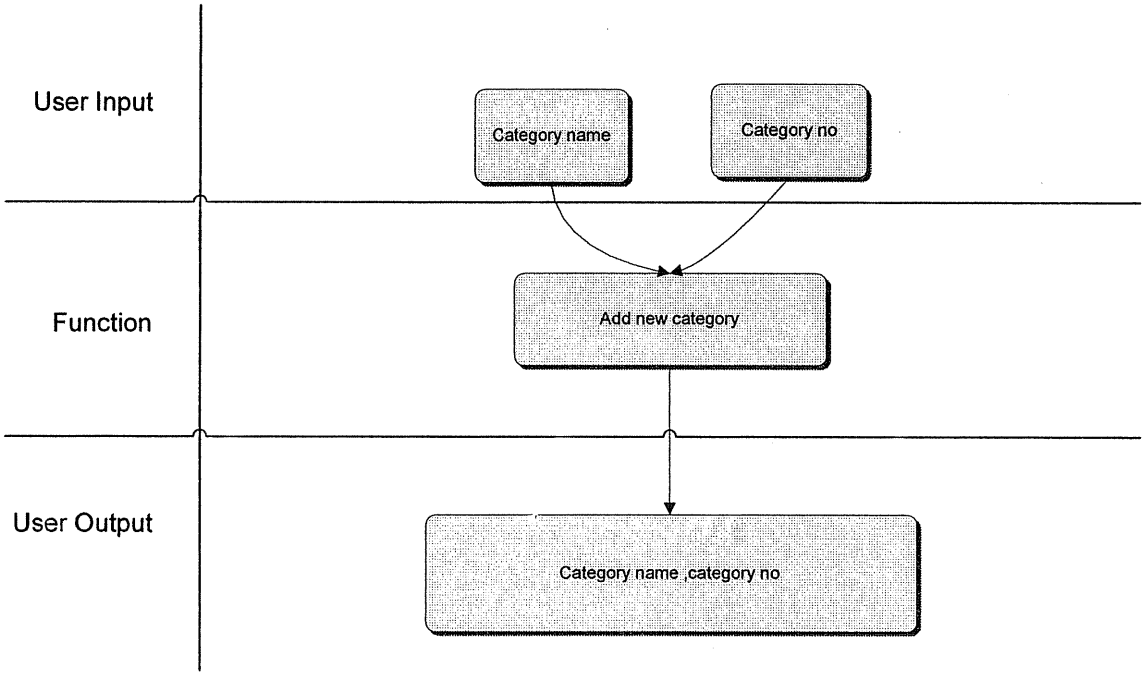
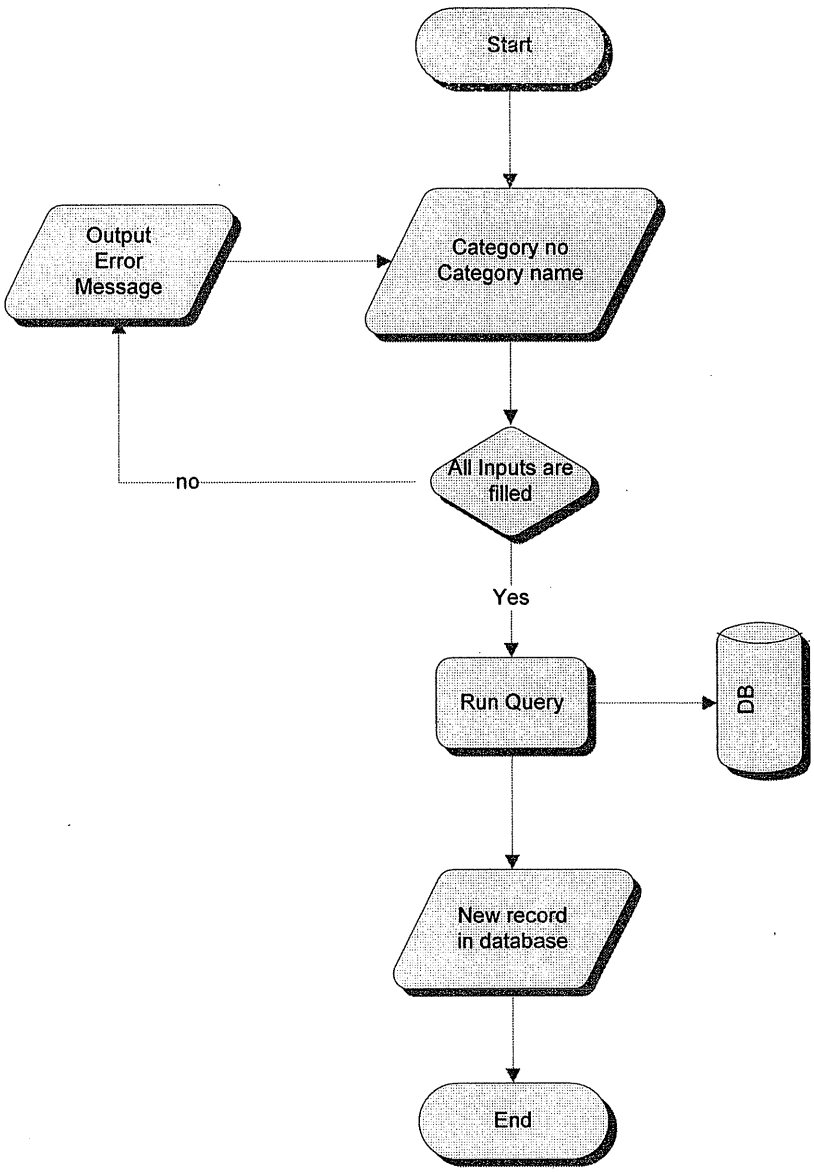


Figure (4.31) add new category interface design

e) Flowchart:



Figure(4.32) add new category operation

17) Add new item

a) Description: this function allows administrator to add new item.

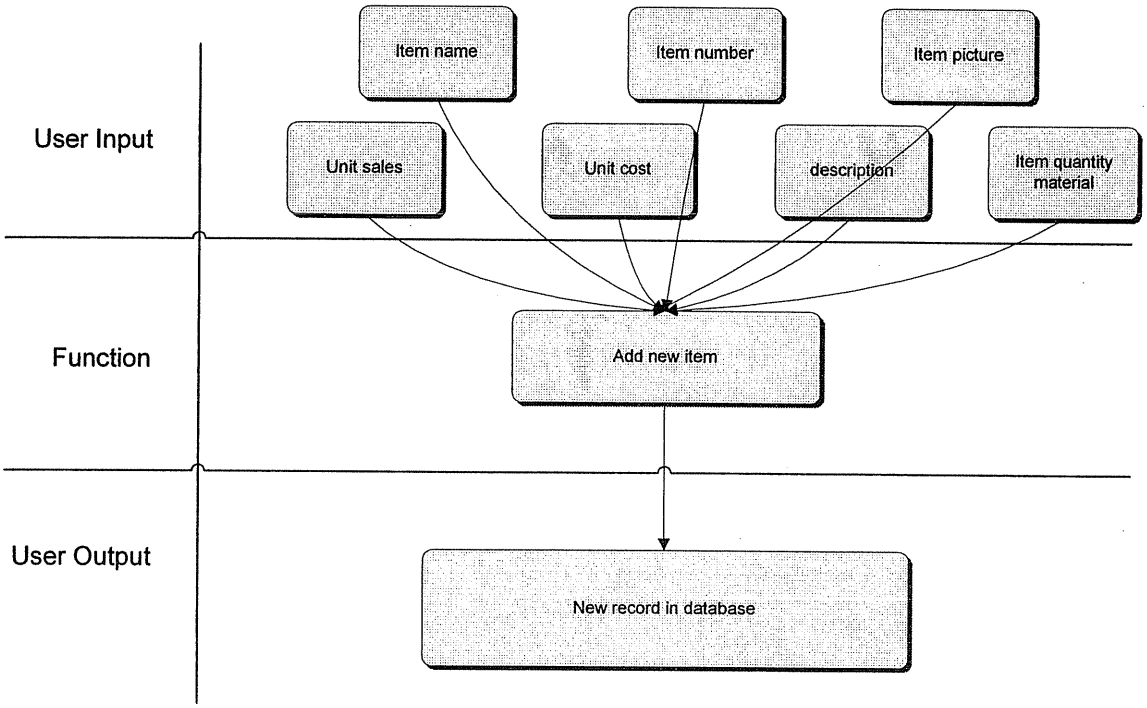
b) Interface:

- Inputs: item name, picture item, description, unit sales, unit cost, item quantity material.
- Outputs: new record in DB.

c) Constraints:

- All Inputs are filled.
- Unit sails grater than unit cost.

d) User interface design:



Figure(4.33) add new item interface design

e) Flowchart:

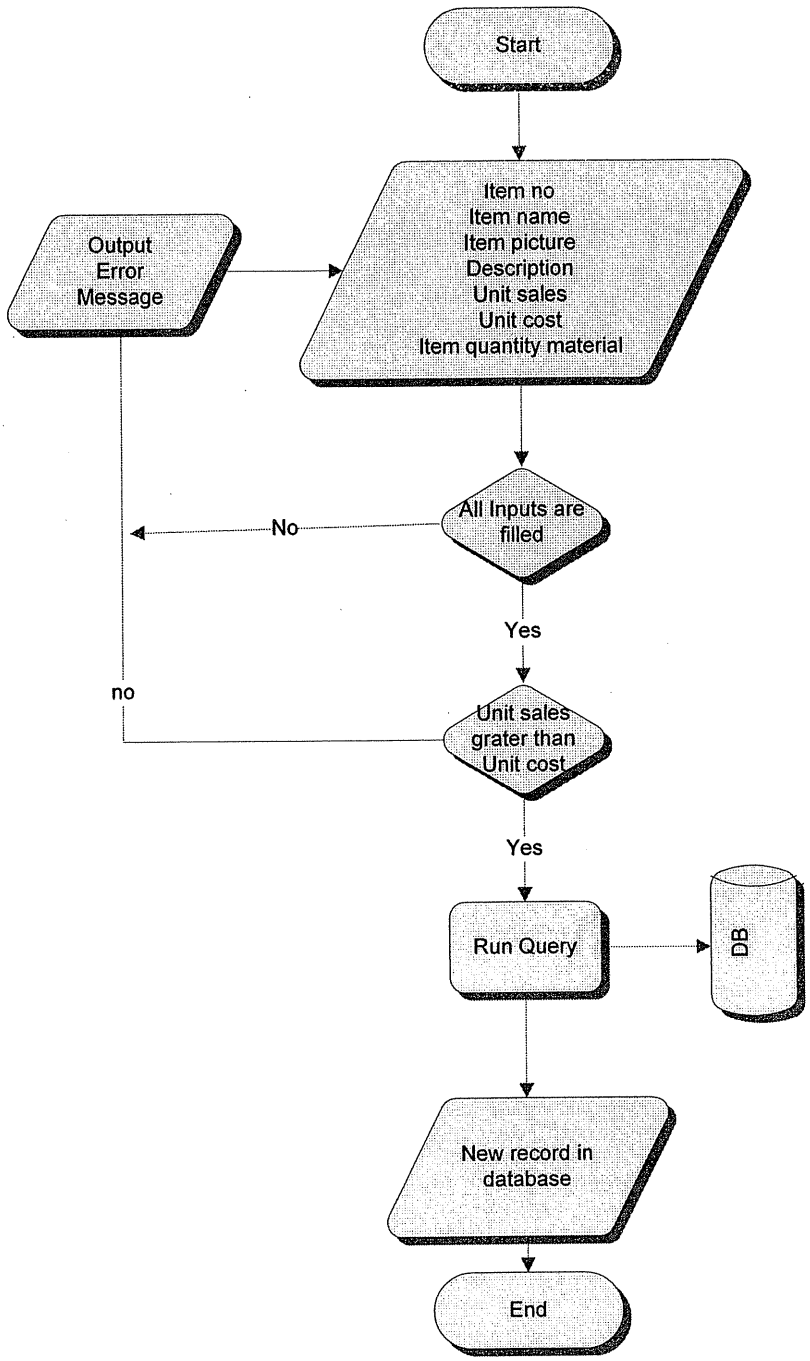


Figure (4.34) add new item operation

18) Update item information

a) Description: this function allows administrator to update item sales.

b) Interface:

- Inputs: new item price sales, item no, item quantity material.
- Outputs: update item sales price in database.

c) Constraints:

- All Inputs are filled.
- Unit sails grater than unit cost

d) User interface design:

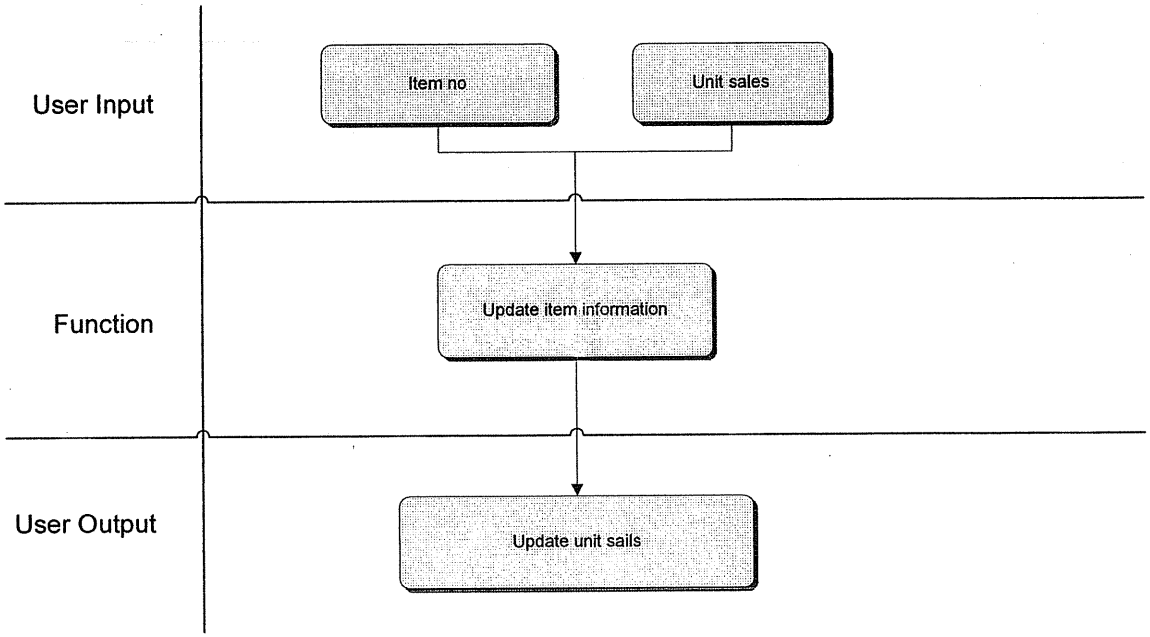
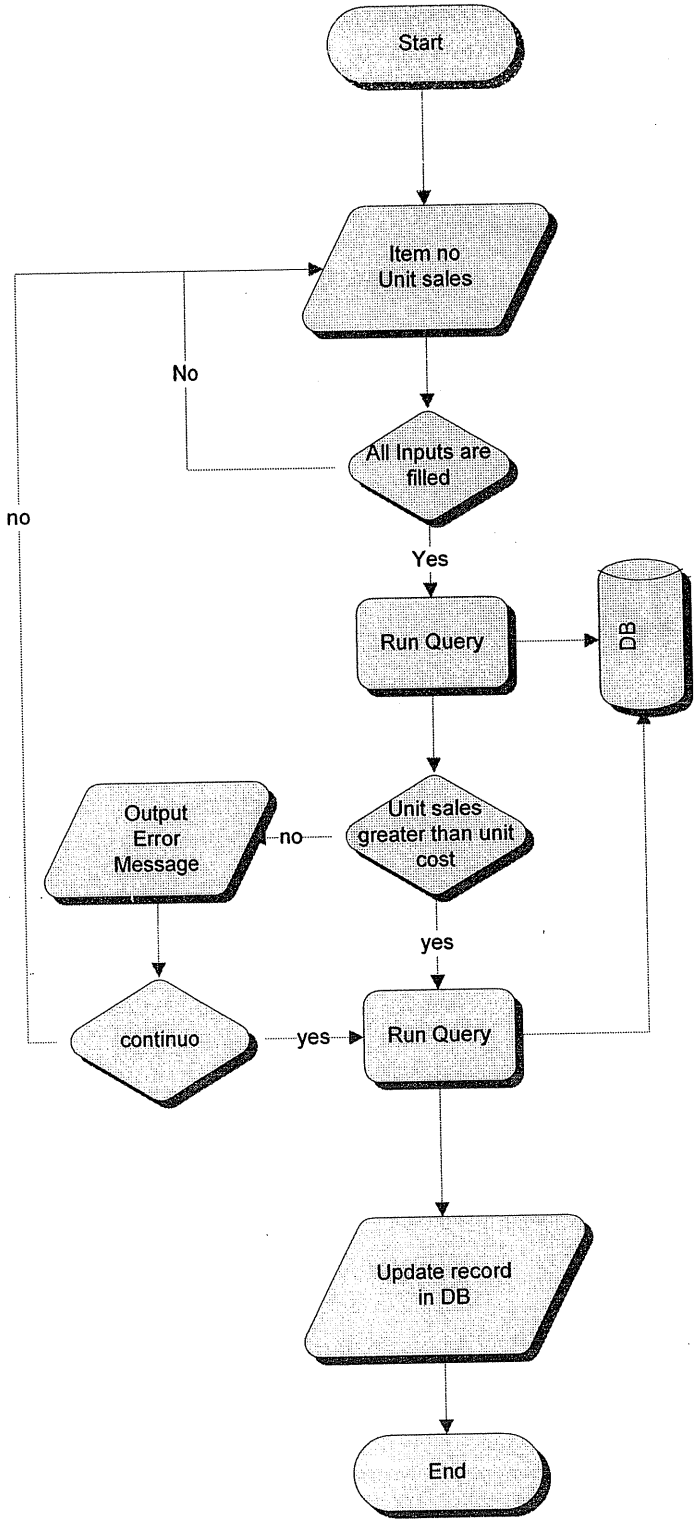


Figure (4.35) update item information interface design

e) Flowchart:



Figure(4.36) update item information operation

19) Manufacturing item

a) Description: the administrator should be able to enter quantity of item was produced

b) Interface:

- Inputs: item no, material no, quantity, category no.
- Outputs: new record in database

c) Constraints:

- All Inputs are filled.
- Quantity must be grater than zero.

d) User interface design:

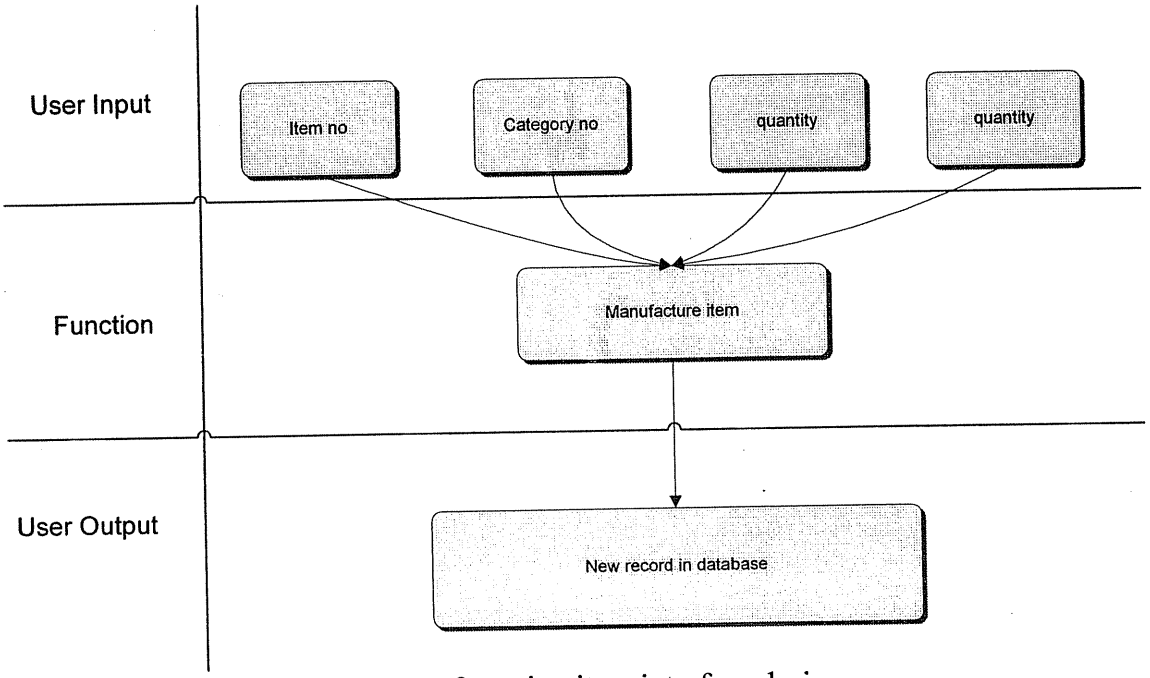


Figure (4.37) manufacturing item interface design

e) Flowchart:

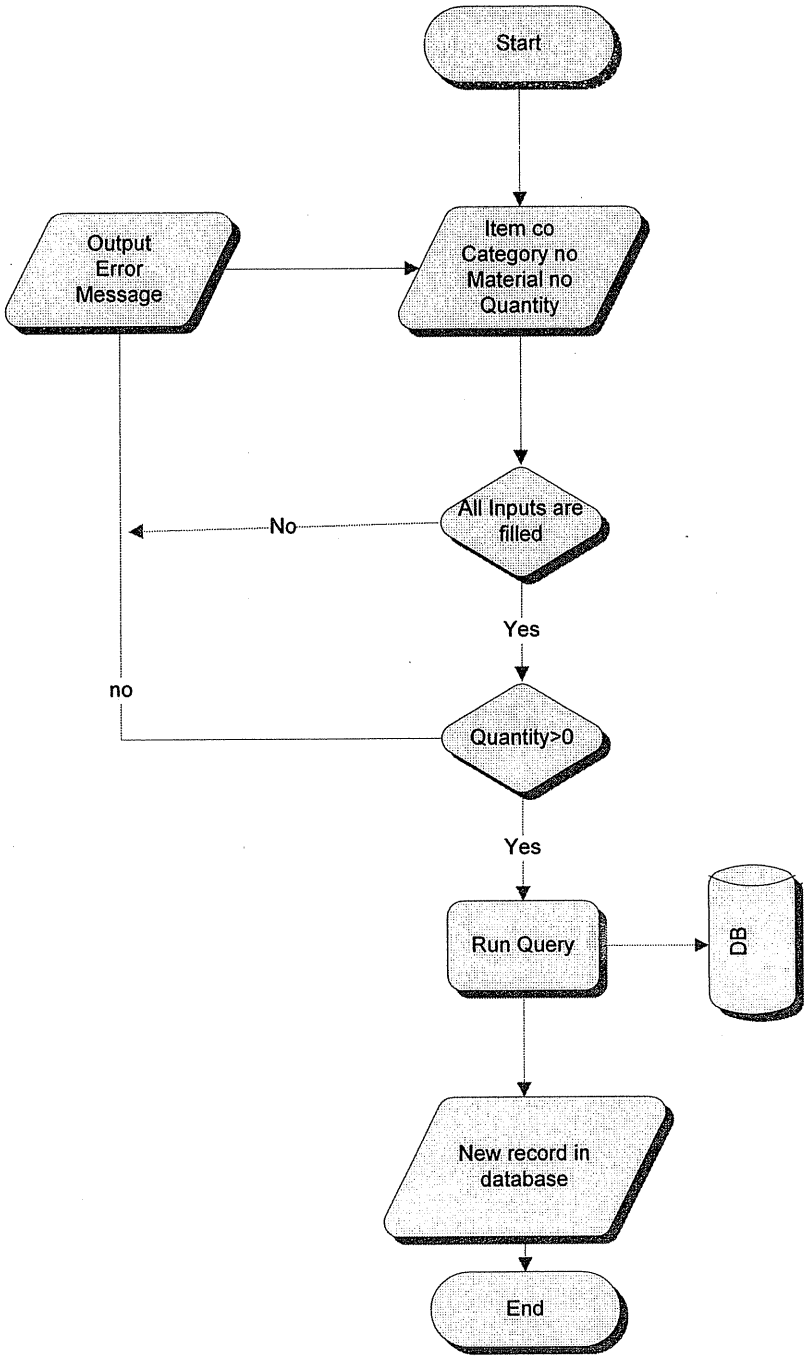


Figure (4.38) manufacturing item operation

20) Add new material

a) Description: this function administrator to add new materials in store.

b) Interface:

- Inputs: material name, material number, description.
- Outputs: new record in DB.

c) Constraints:

- All input must be filled.

d) User interface design:

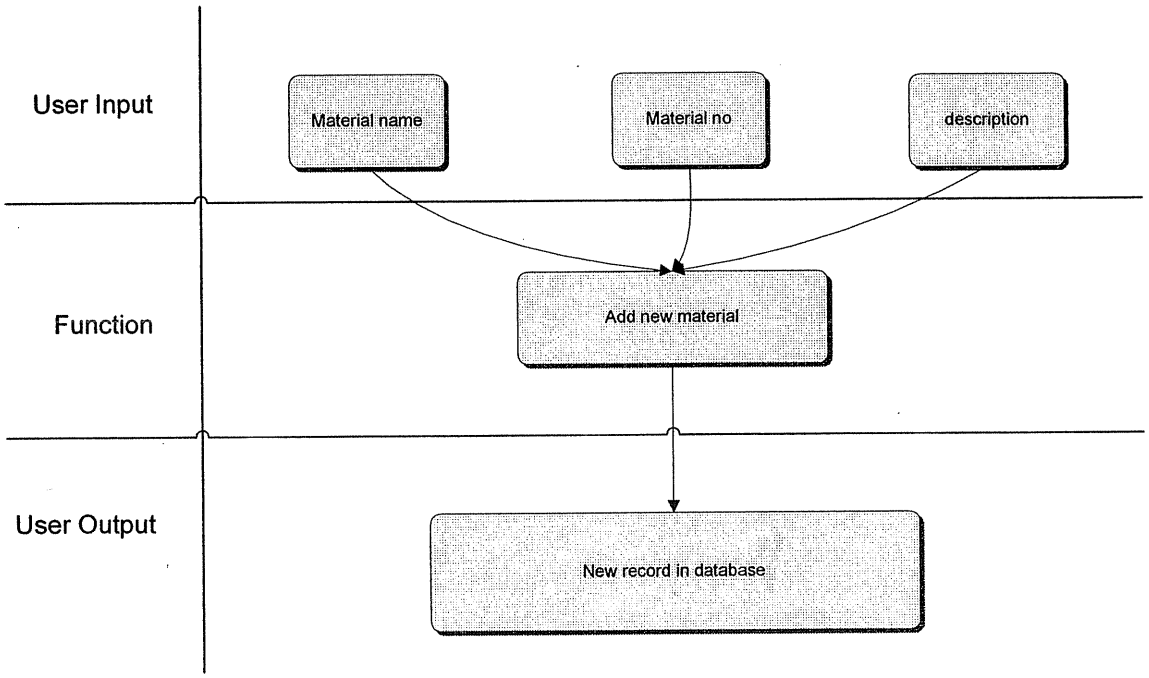


Figure (4.39) add new material interface design

e) Flowchart:

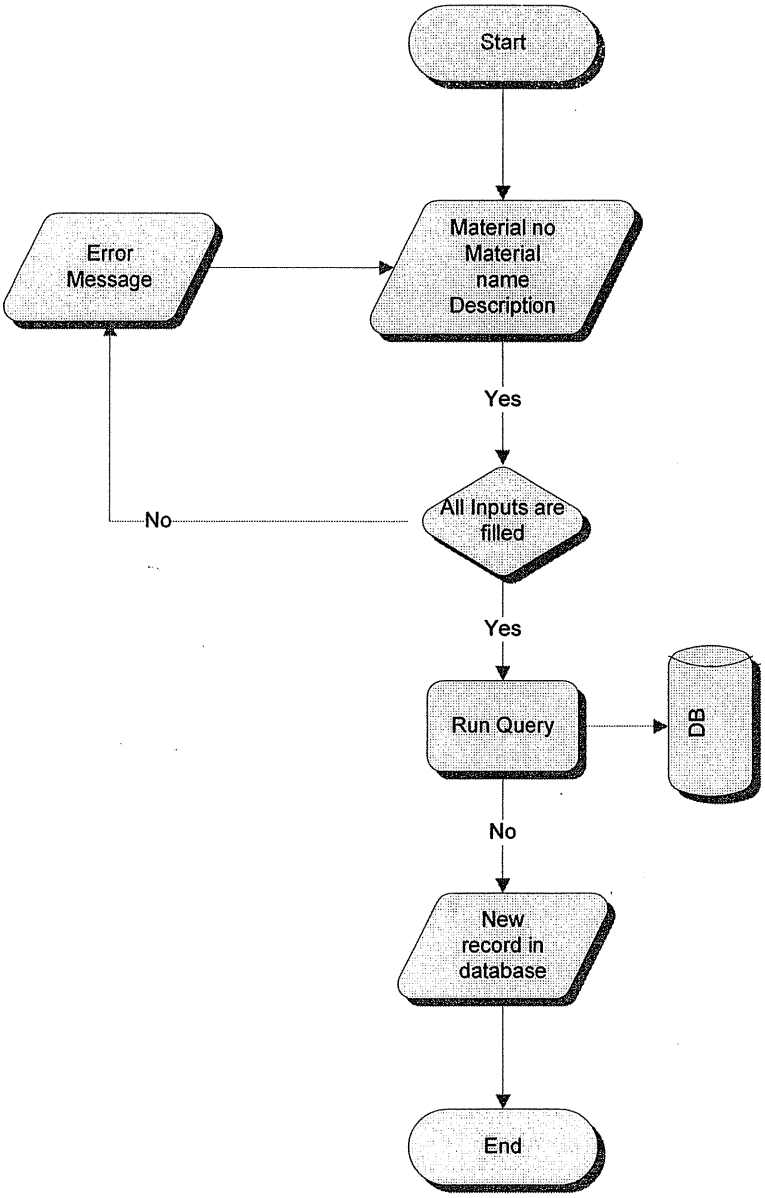


Figure (4.40) add new material operation

21) Inventory entry for materials.

a) Description: this function allows the administrator to add new row materials in the store.

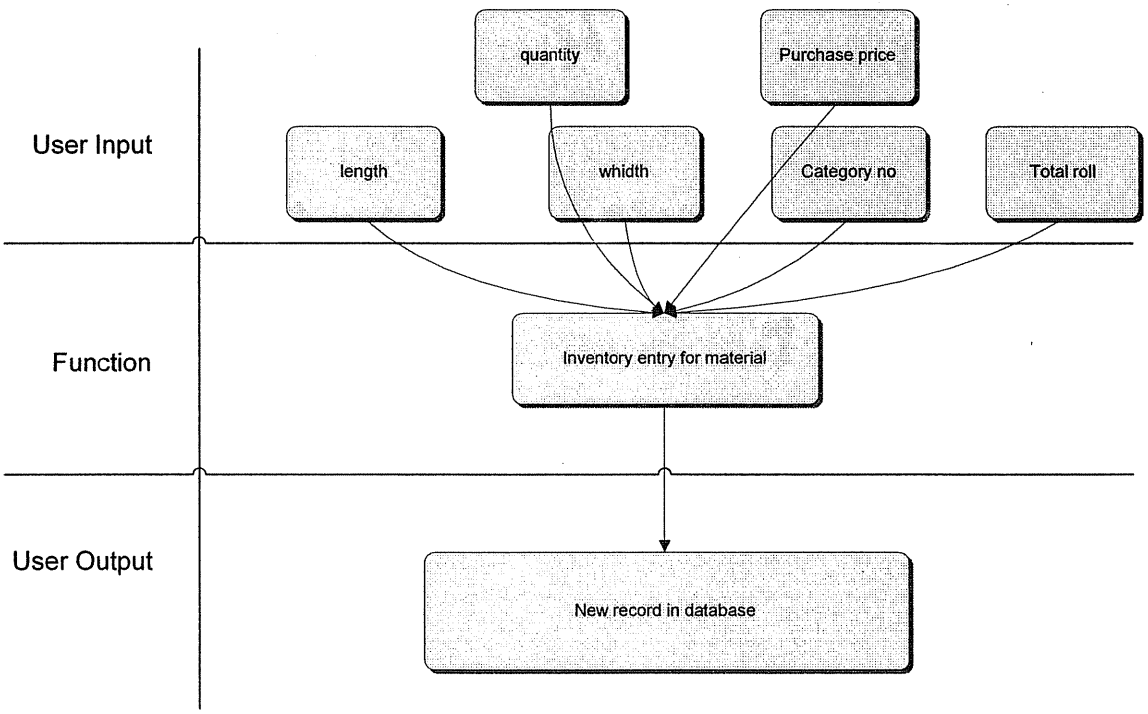
b) Interface:

- Inputs: quantity, purchase price, width, number of roll, length, category number.
- Outputs: new record in DB (new material quantity).

c) Constraints:

- Material name, category name and quantity must be available.

d) User interface design:



Figure(4.41) inventory entry for material interface design

g) Flowchart:

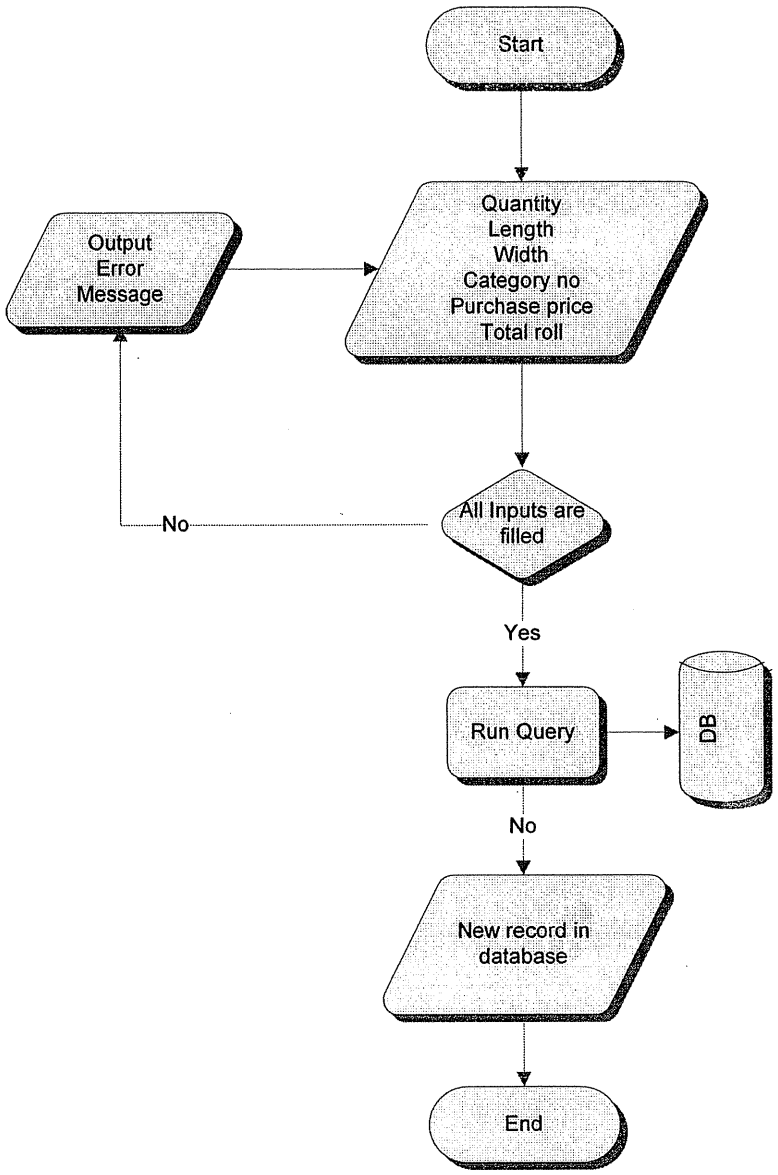


Figure (4.42) inventory entry for material operation

22) Store report.

a) Description: this function allows administrator to display reports for each stores that contains information about material and item available in store.

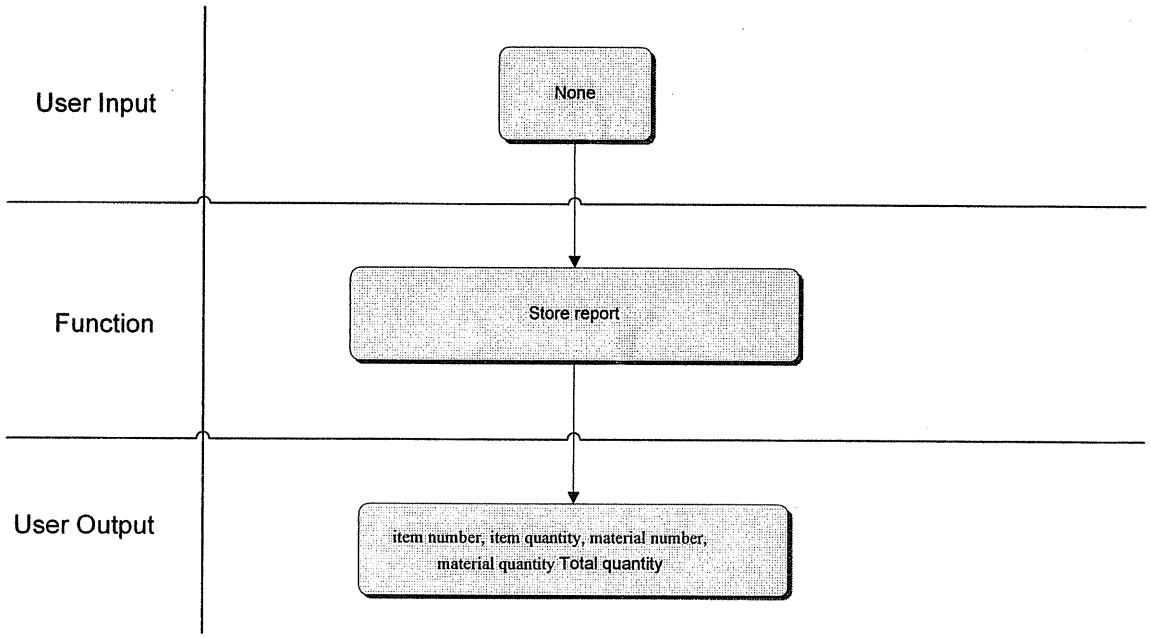
b) Interface:

- Inputs: click on button "تقرير المخزن".
- Outputs: item number, item quantity, material number, material quantity Total quantity.

c) Constraints:

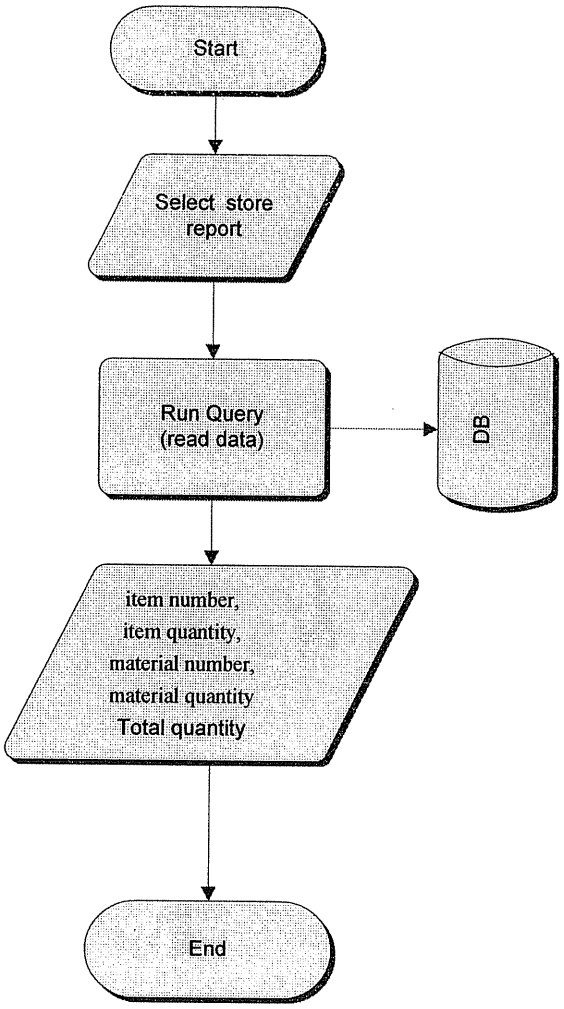
- None.

d) User interface design:



Figure(4.43) store report interface design

e) Flowchart:



Figure(4.44) store report operation

23) User profile maintenance.

a) Description: this function allows administrator to add, delete ,search about any users (customer, administrator and employee).

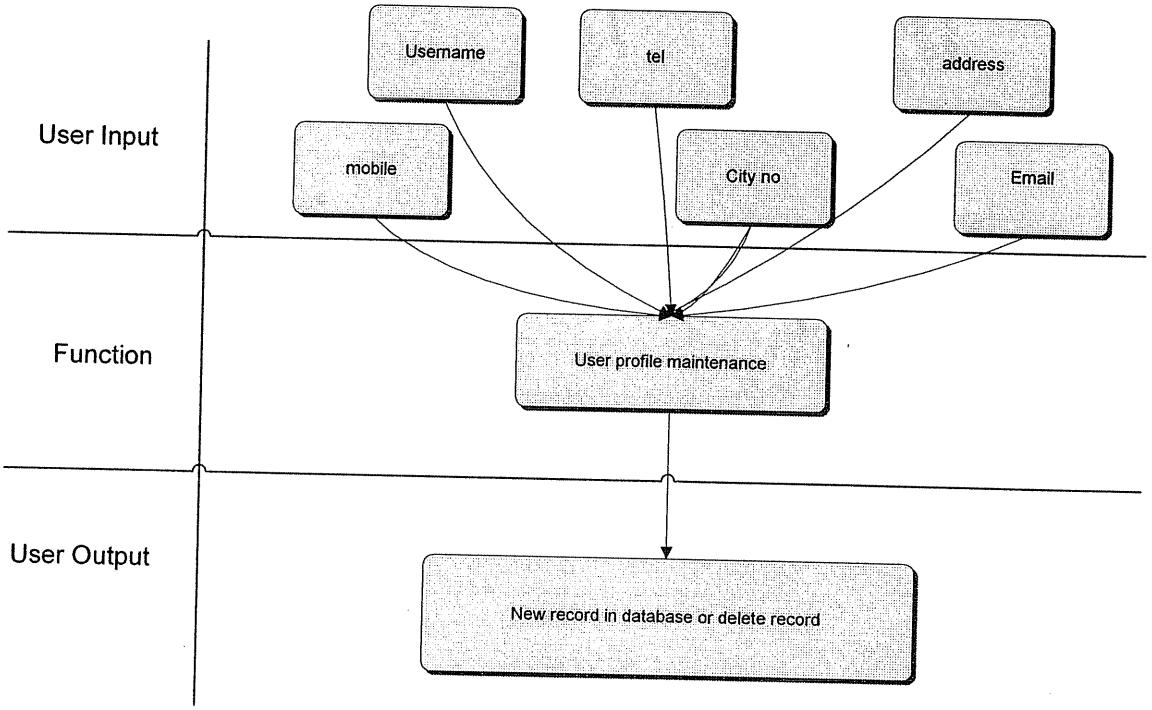
b) Interface:

- Inputs: information for each user (customer, administrator, employee) such as (user name, city, address, tell, email…act).
- Outputs: create new record in DB or delete old record from DB.

c) Constraints:

- User not available in DB when add, user available when delete or search.

d) User interface design:



Figure(4.45) user profile maintenance interface design

e) Flowchart:

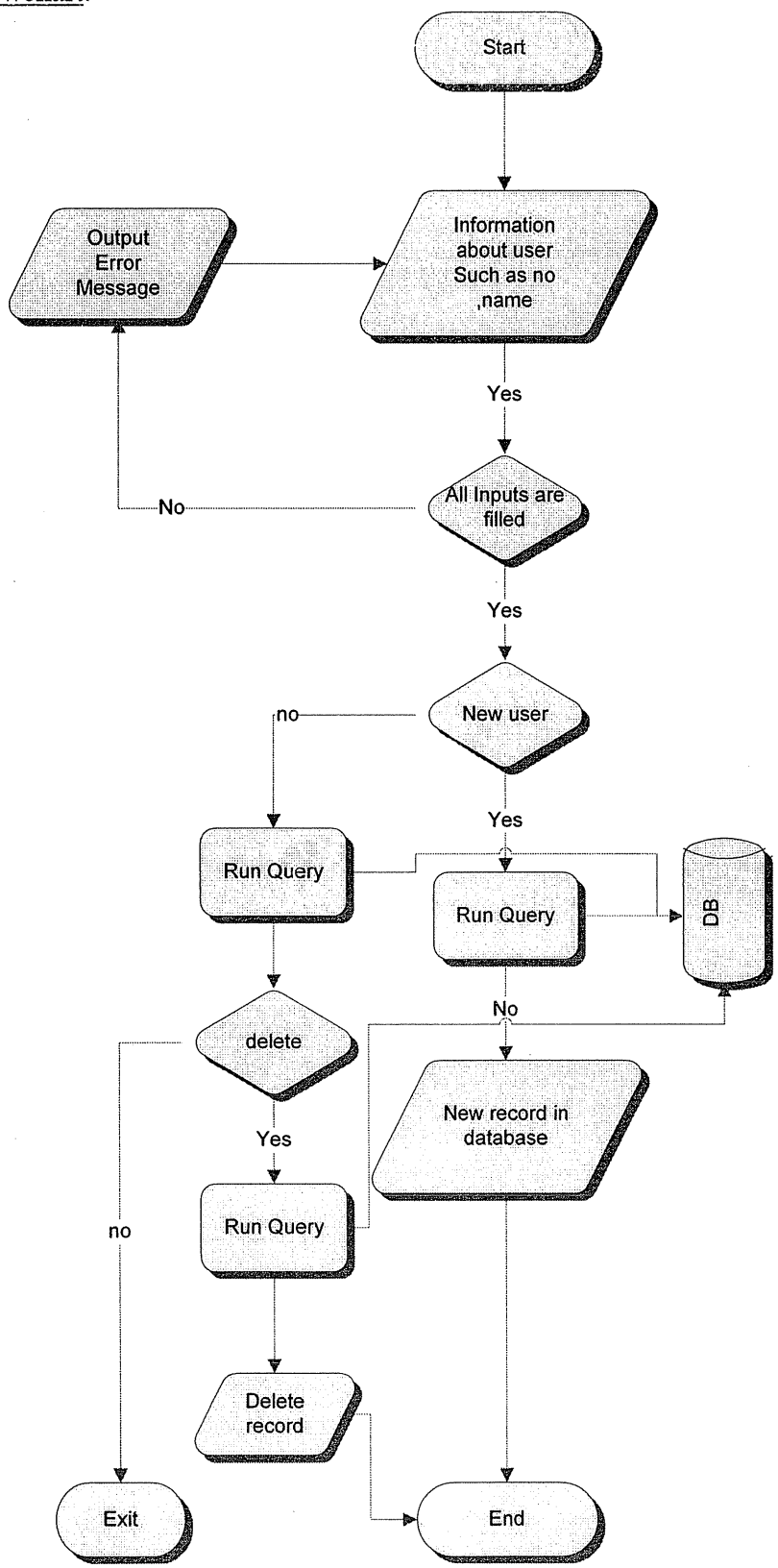


Figure (4.46) user profile maintenance operation

24) Generate bill.

a) Description: this function allows the administrator to generate bill for each order was confirmed by customer.

b) Interface:

- Inputs: customer ID, bill date, bill no, order no.
- Outputs: create new record in DB.

c) Constraints:

- None

d) . User interface design:

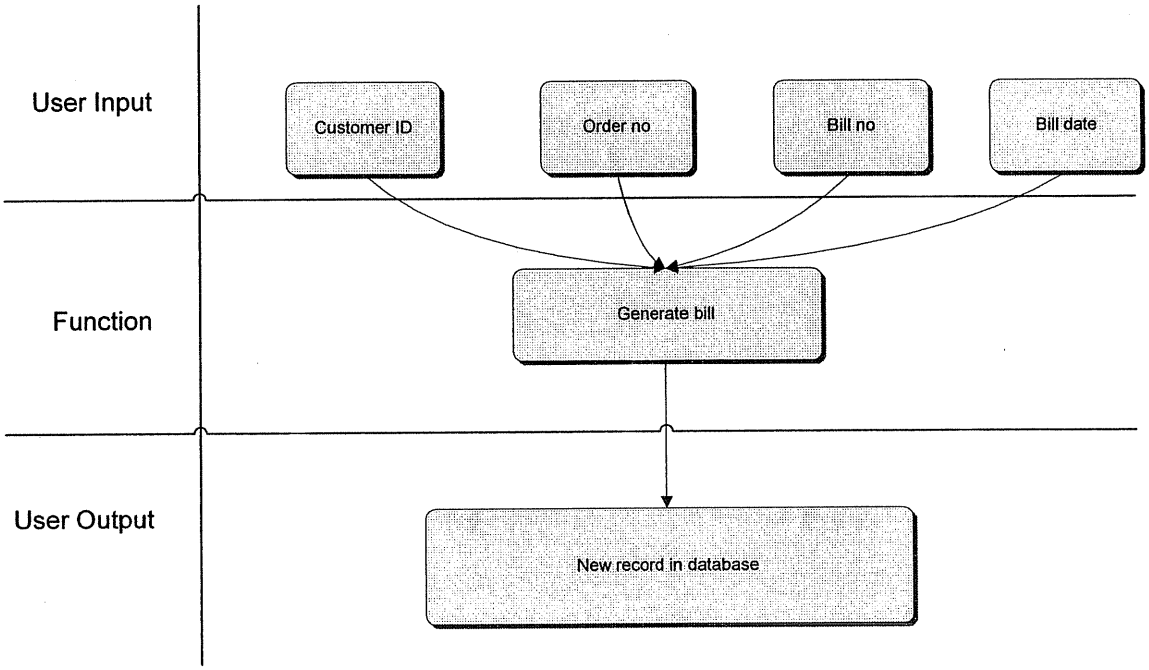


Figure (4.47) generate bill interface design

25) User help.

a) Description: this function provide user to view help for using web site component or the system.

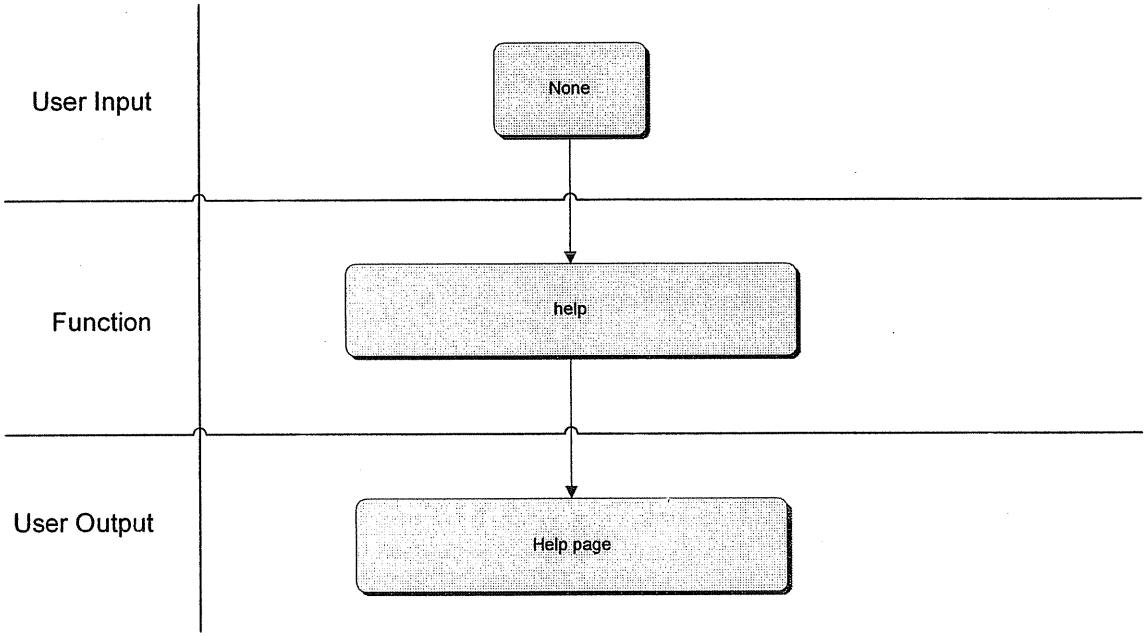
b) Interface:

- Inputs: click on button"مساعدة".
- Outputs: help page.

c) Constraints:

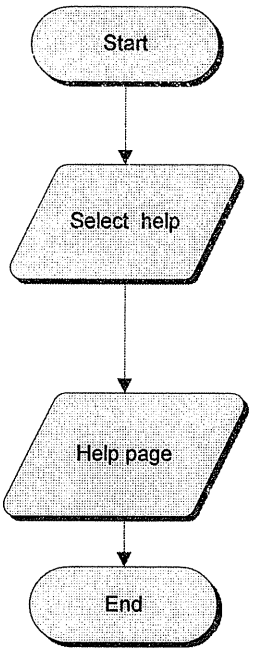
- None.

d) User interface design:



Figure(4.49) user help interface design

e) Flowchart:



Figure(4.50) user interface operation

26) Viewing the sent Comments.

a) Description: The administrator should be able to view comments That sent by customer, employee.

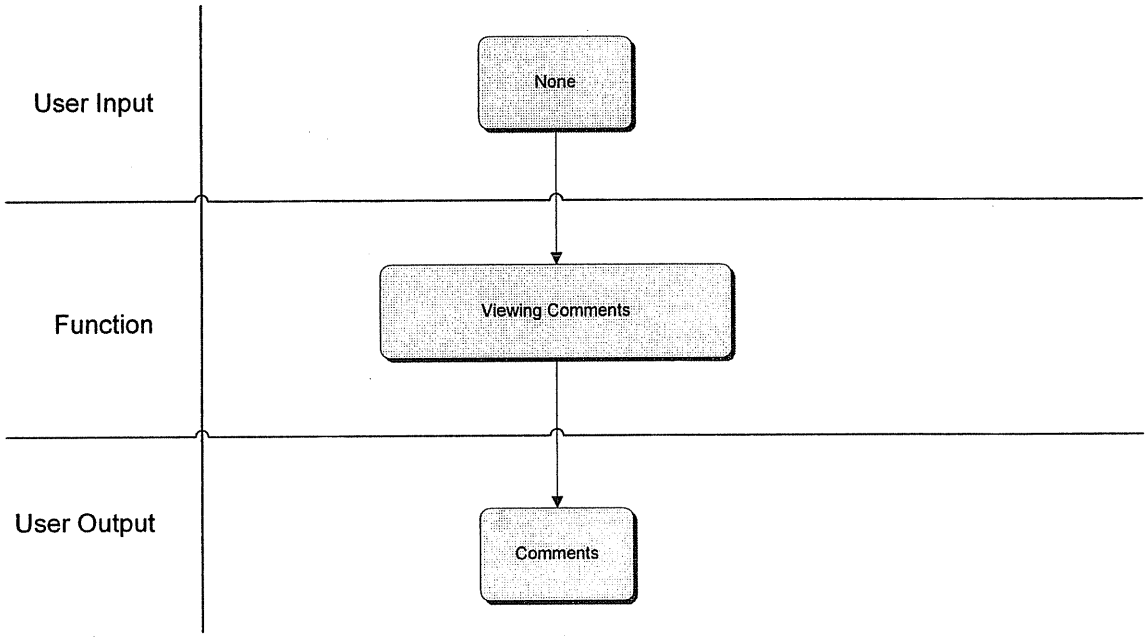
b) Interface:

- Inputs: None.
- Outputs: Comments list and details.

c) Constraints:

- None.

d) User interface design:



Figure(4.51) viewing sent comment interface design

e) Flowchart:

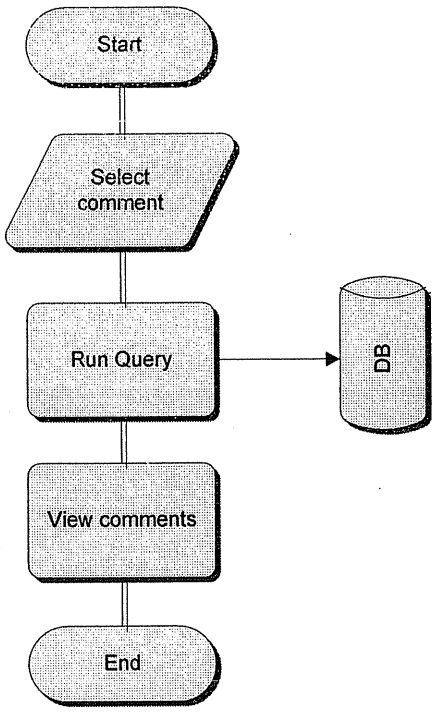


Figure (4.52) viewing sent comment operation

27) Employee work record.

a) Description: this function allows the administrator to enter information of employee working (start time, end time, employee name) daily.

b) Interface:

- Inputs: start time, end time employee name, day.
- Outputs: create new record in DB.

c) Constraints:

- All input must be filled.
- End time must be grater than start time.

d) User interface design:

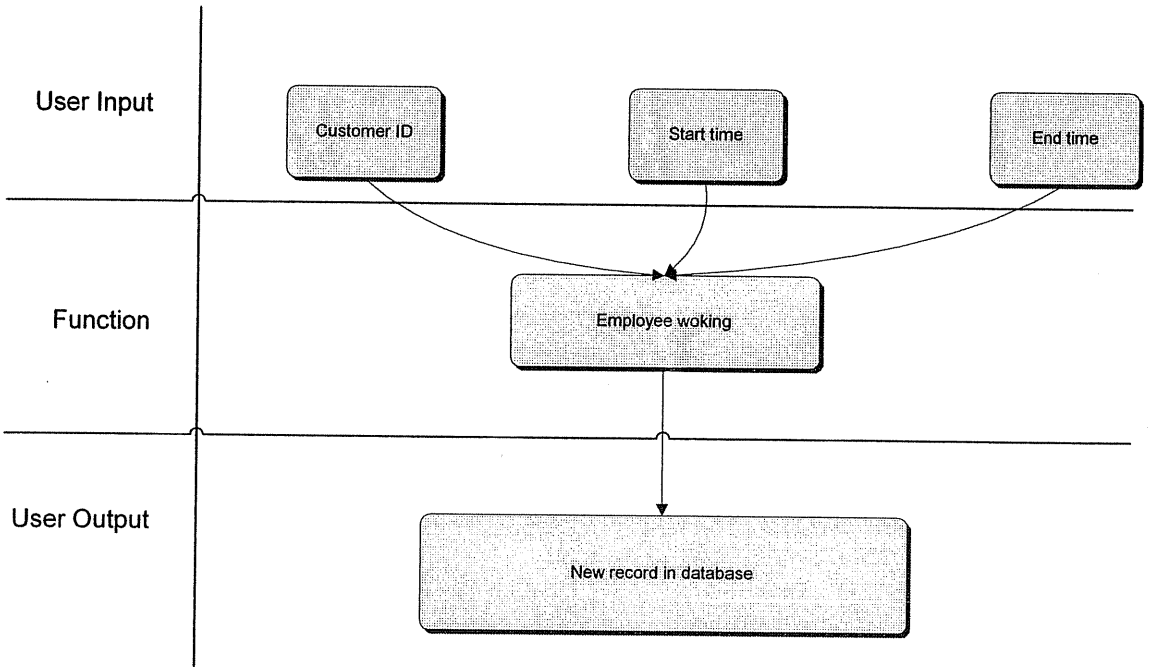


Figure (4.53) employee work record interface design

e) Flowchart:

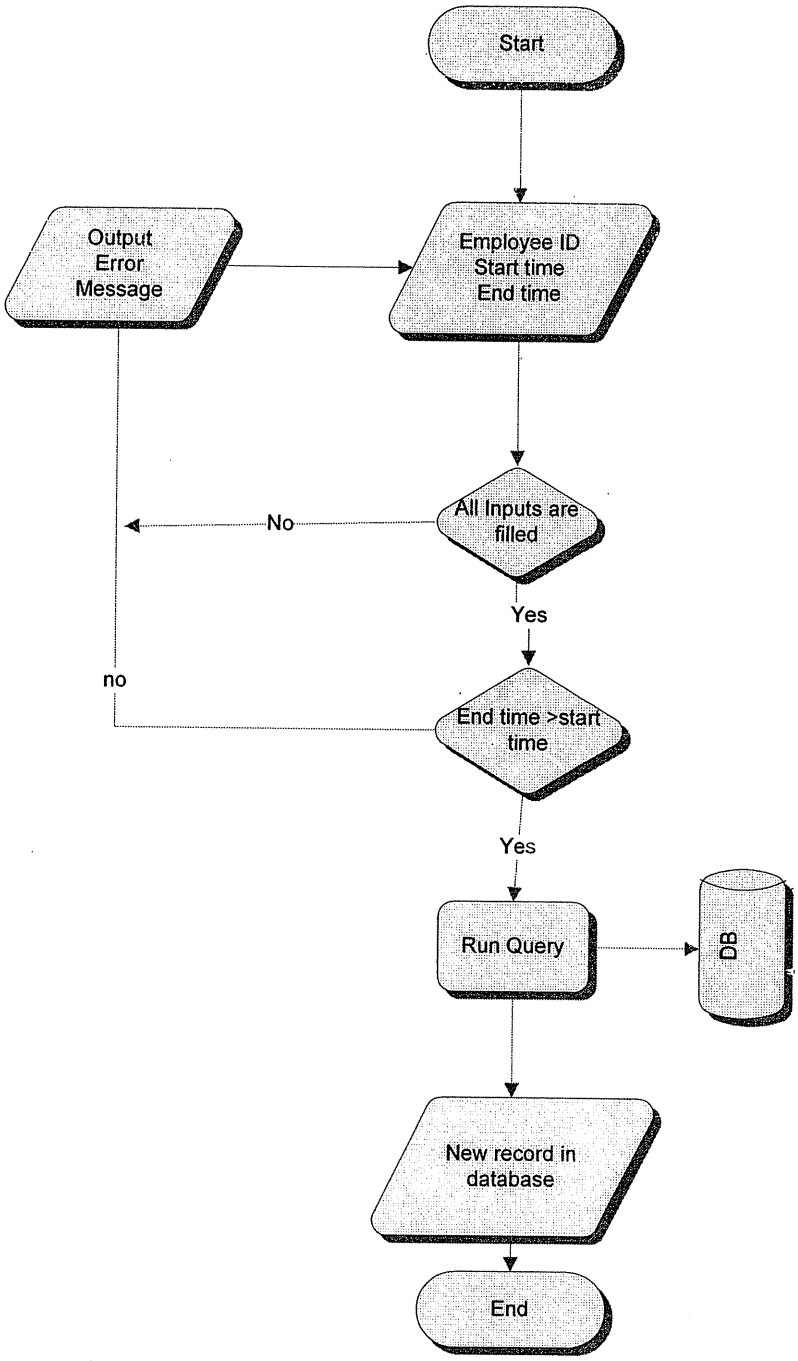


Figure (4.54) employee work record operation

28) Generate Employee reports.

a) Description: this function allows administrator to generate employee reports

b) Interface:

- Inputs: employee no.
- Outputs: number of hour he work, cost of hour, total paid, total reminder.

c) . Constraints:

- None

d) User interface design:

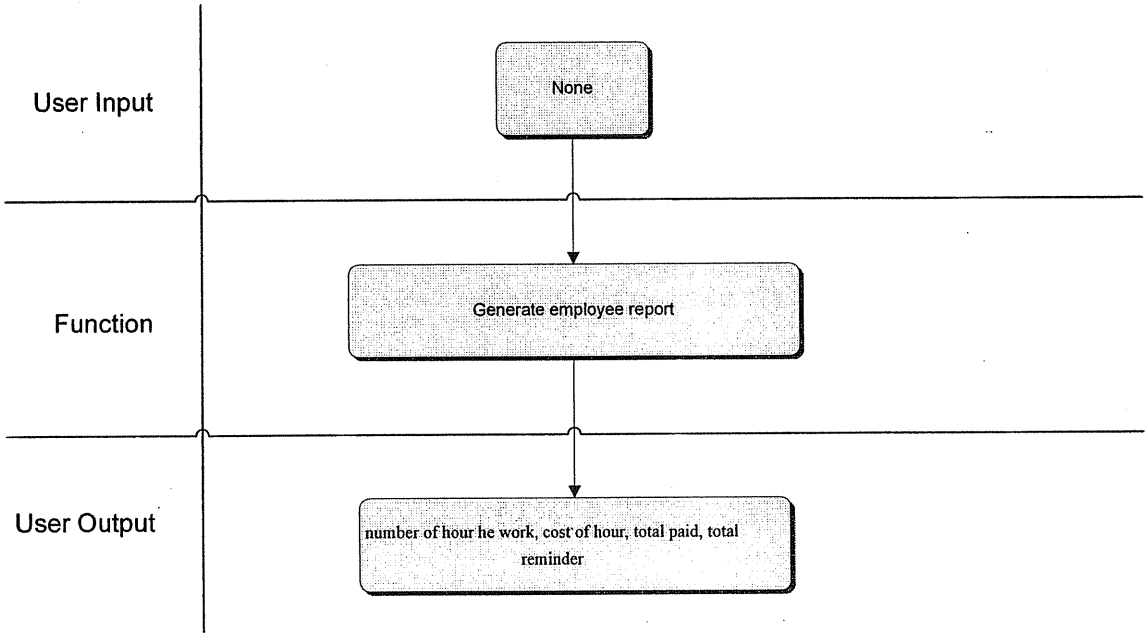


Figure (4.55) generate employee report interface design

e) Flowchart:

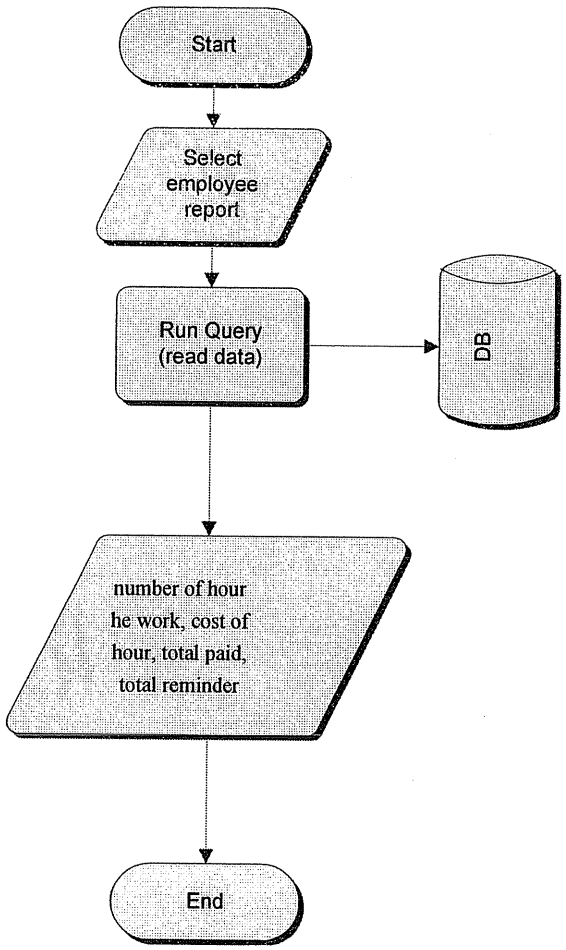


Figure (4.56) generate employee report operation

29) Recommendations.

a) Description: this function provide administrator notification for new orders was mad from customer and more item sales in sessions

b) Interface:

- Inputs: none.
- Outputs: Max sales item or item no, item name

c) Constraints:

- None

d) User interface design:

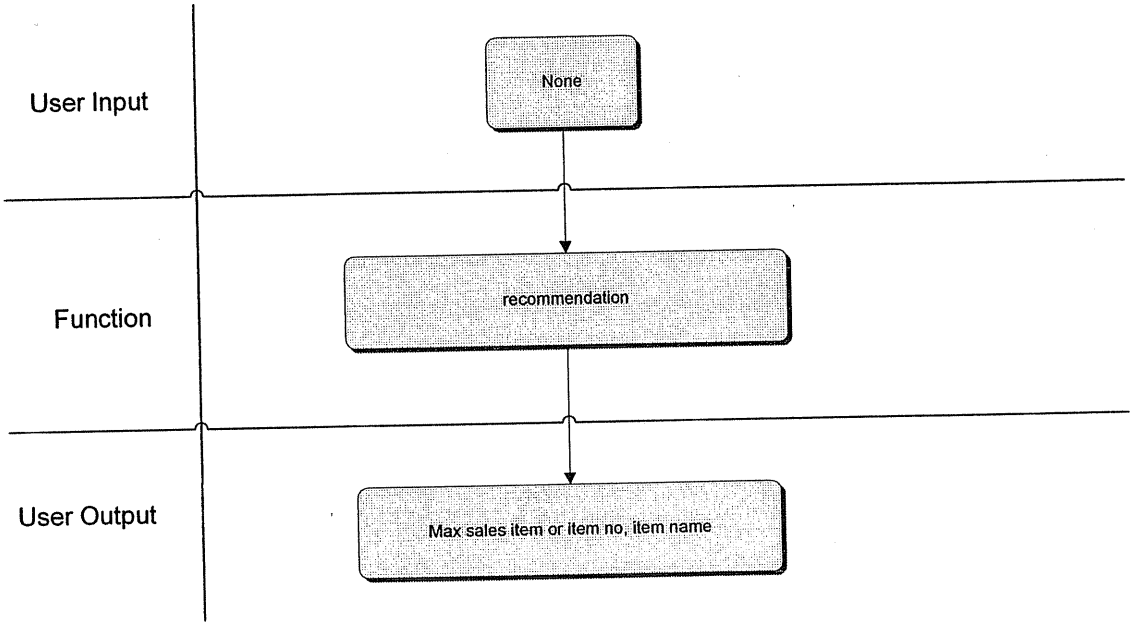


Figure (4.57) recommendation interface design

Flowchart:

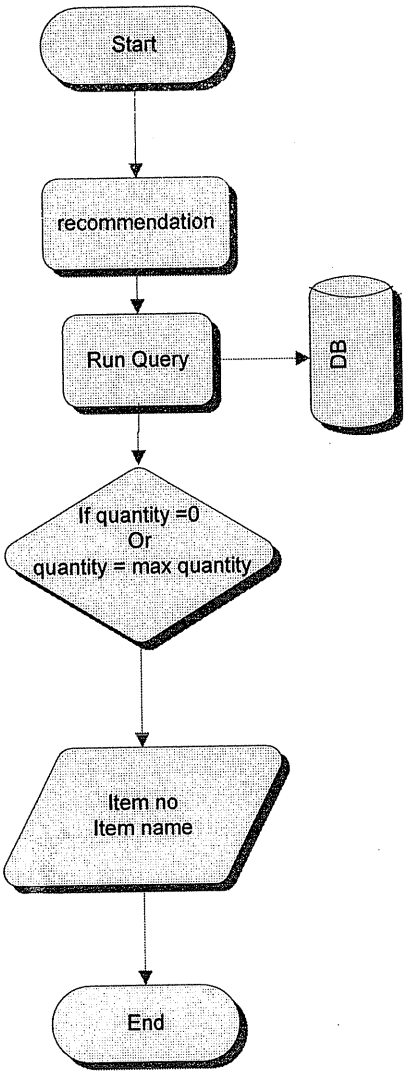


Figure (4.58) recommendation operation

30) Bank login

a) Description: the customer should be able to login the bank web site

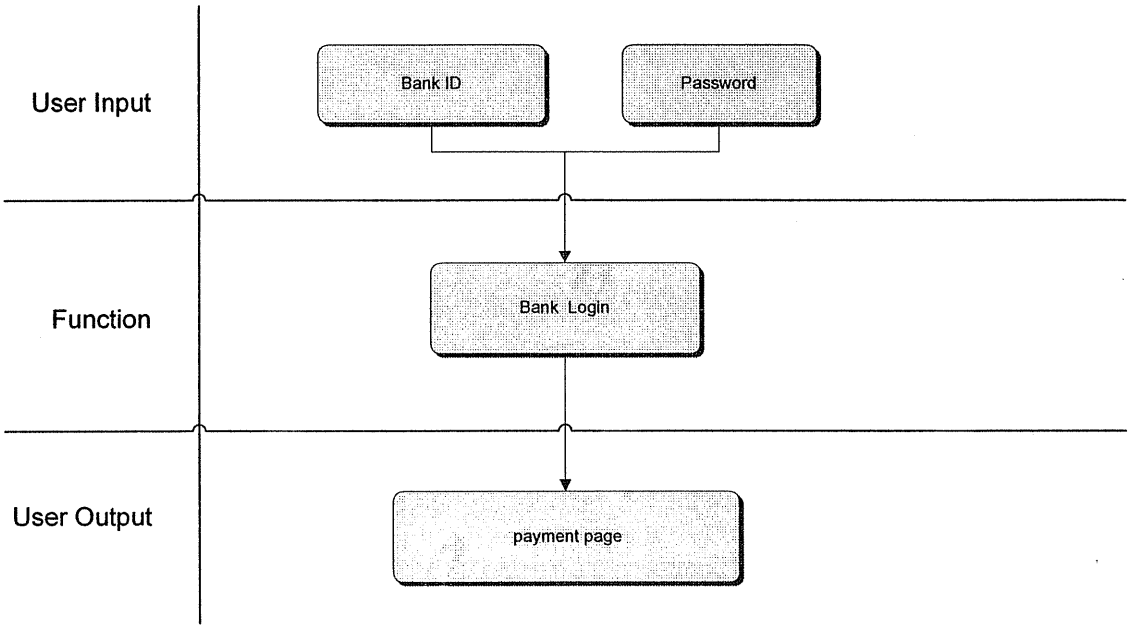
b) Interface:

- Inputs: user Id and password
- Outputs: bank E payment page.

c) Constraints:

- ID and password must be provided and must be valid

d) User interface design:



Figure(4.59) bank login interface design

e) Flowchart:

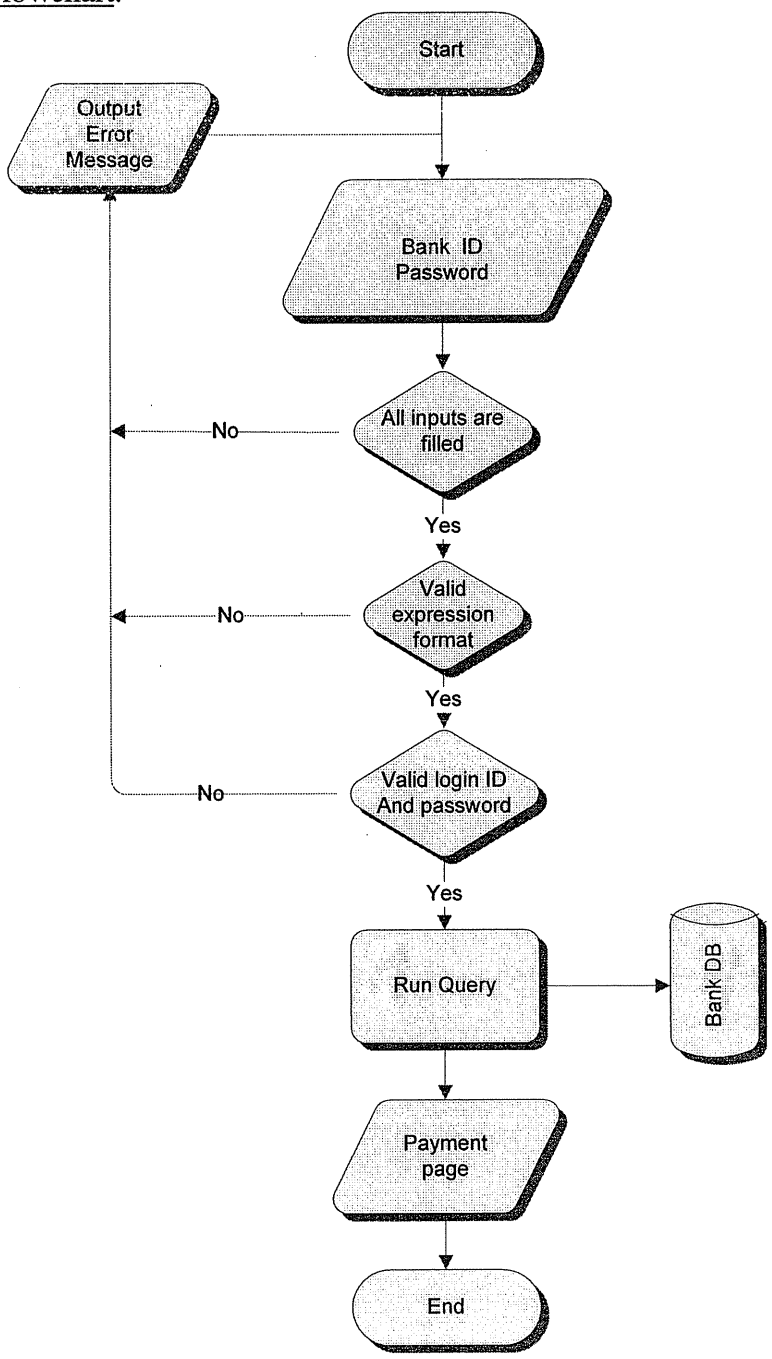


Figure (4.60) bank login operation

31) Bank confirmations

a) Description: the customer should be confirmed by bank that the bill was paid

b) Interface:

- Inputs: : none
- Outputs: Page content (bill no amount payment, payment date)

c) Constraints:

- None.

d) User interface design:

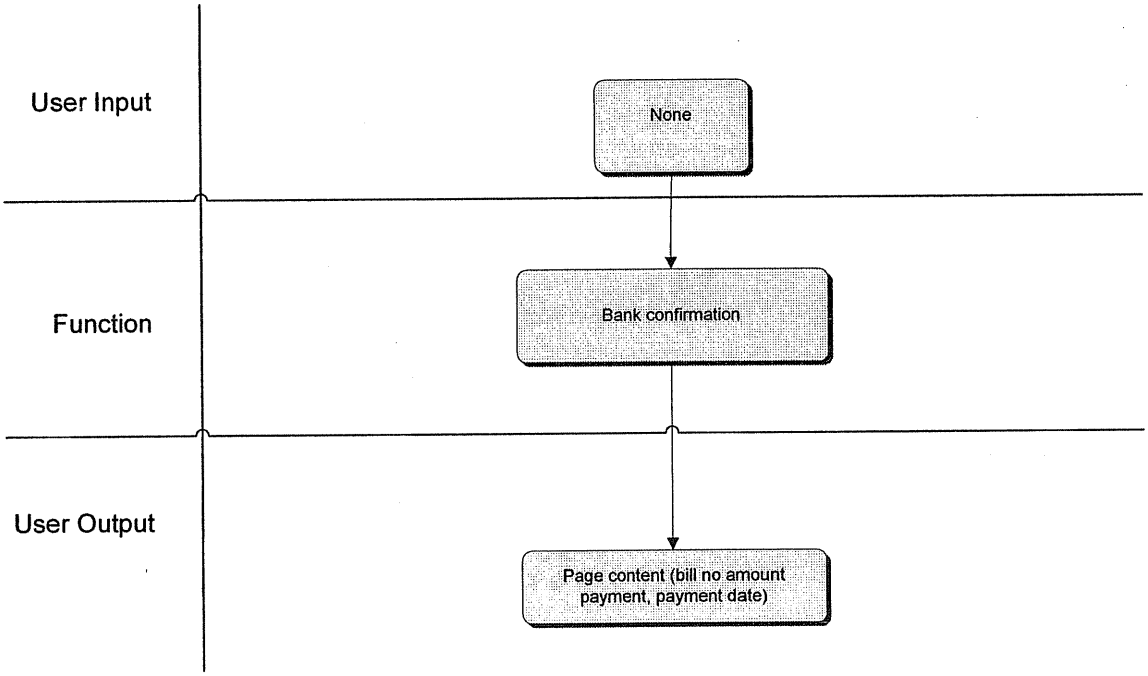


Figure (4.61) bank confirmation interface design

e) Flowchart:

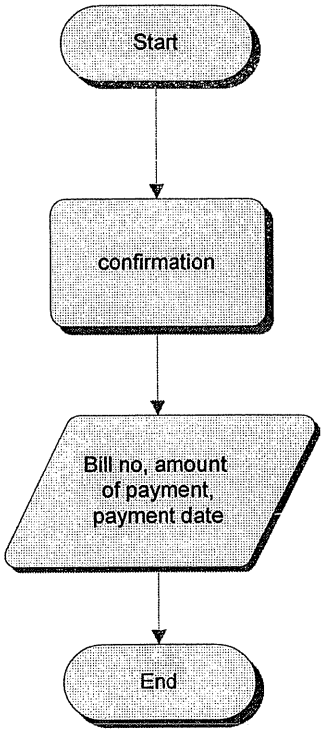


Figure (4.62) bank confirmation operation

32) Generate XML transaction

a) Description: the bank should generate XML transactions to Shawer Company.

b) Interface:

- Inputs: : Bill no, Amount money to be paid, Customer ID
- Outputs: XML file.

c) Constraints:

- None.

d) User interface design:

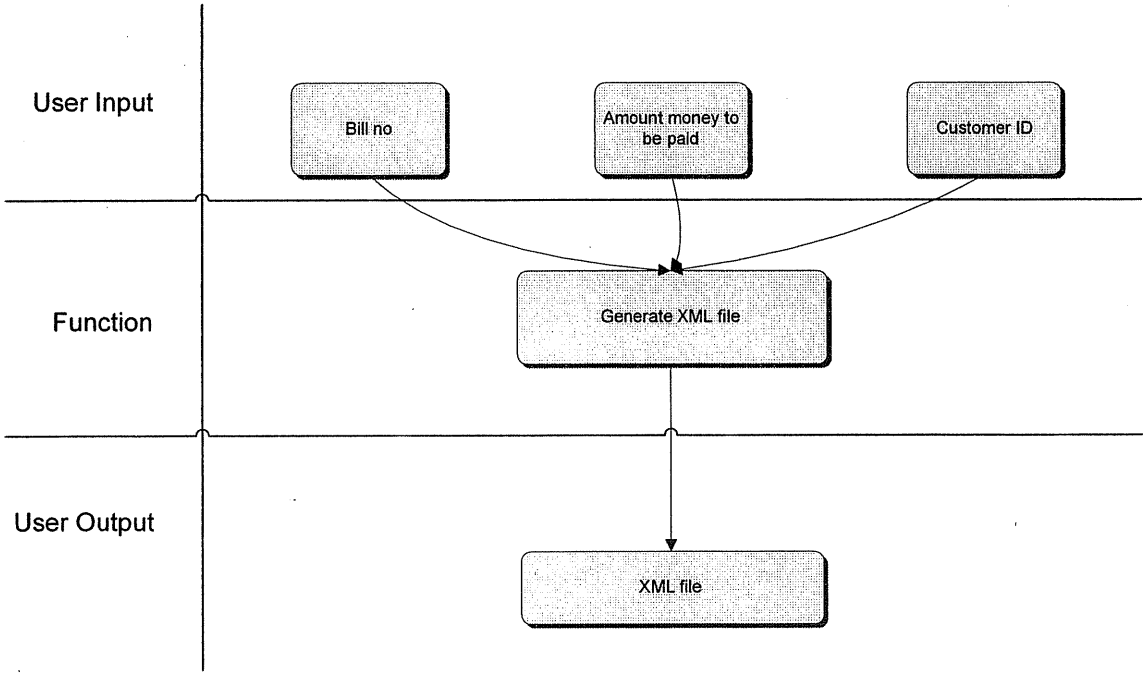


Figure (4.63) generate XML transaction interface design

e) Flowchart:

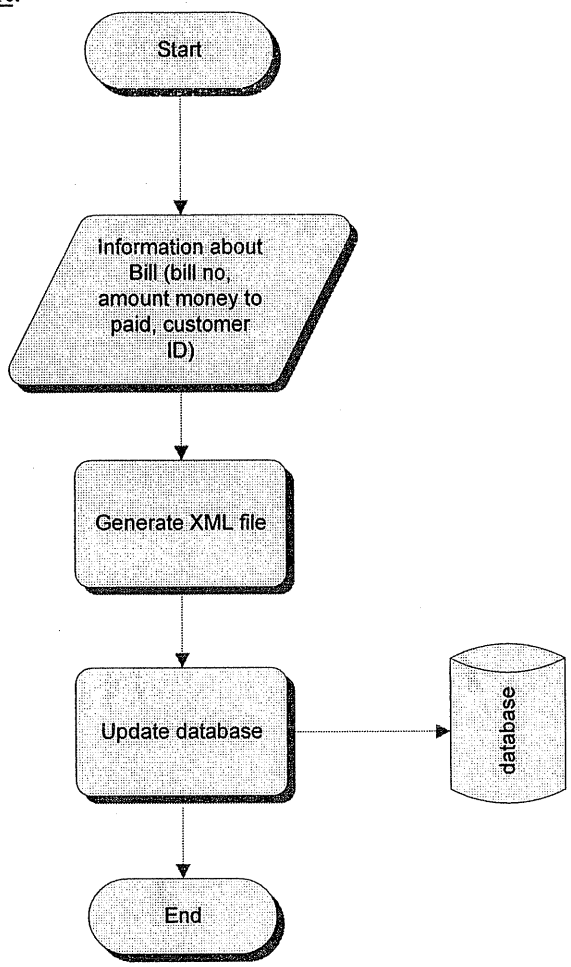


Figure (4.64) generate XML transaction operation

33) 21 Reading XML files.

a) Description: shawer company should read XML files and update its

b) Interface:

- Inputs: : XML file
- Outputs: confirmation.

c) .Constraints:

- None.

d) User interface design:

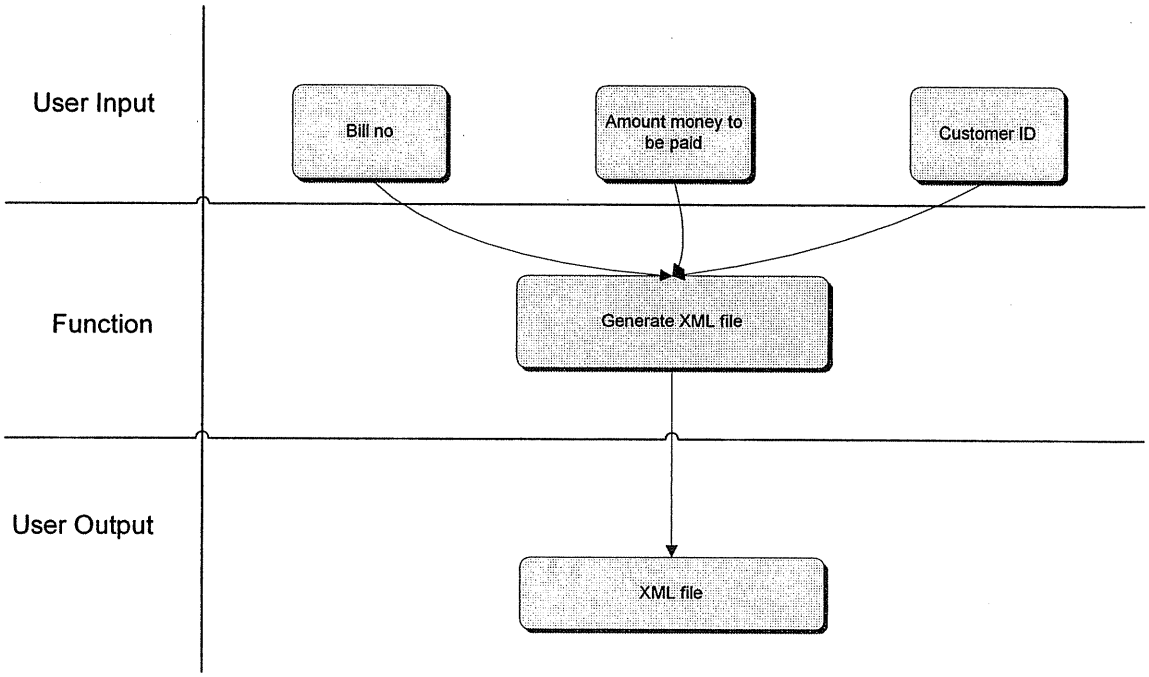


Figure (4.65) reading XML files interface design

e) Flowchart:

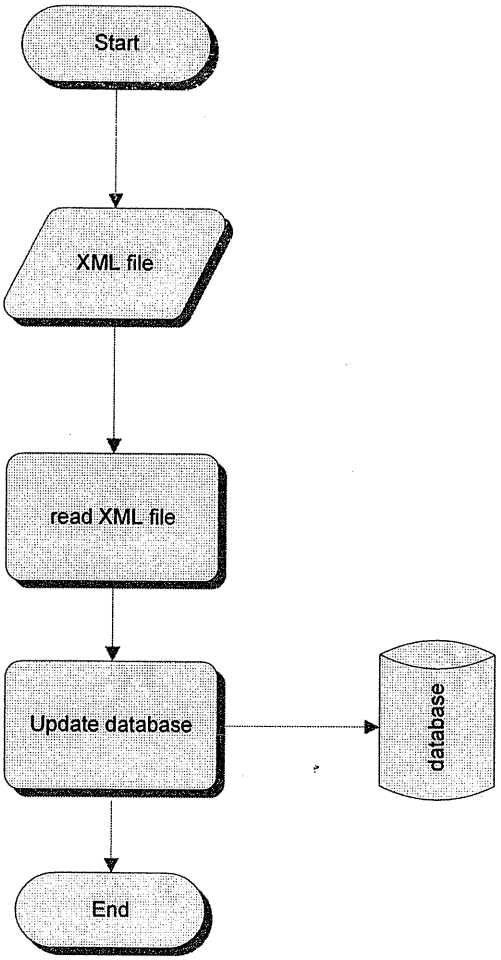


Figure (4.66) reading XML operation

4.3 Input/output Design

Designing the user interface forms is an important complementary issue when developing a powerful commercial system, as the system will operate within a context that contains huge numbers of different users; it should satisfy their needs towards system appearance and interface.

In this chapter we design I/O user interface design, including customer I/O design, Administrator I/O design and employee I/O design.

4.3.1 System interface design

4.3.1.1 Visitor interface design:

a) *Visitor Input design:*

- Registration Form:

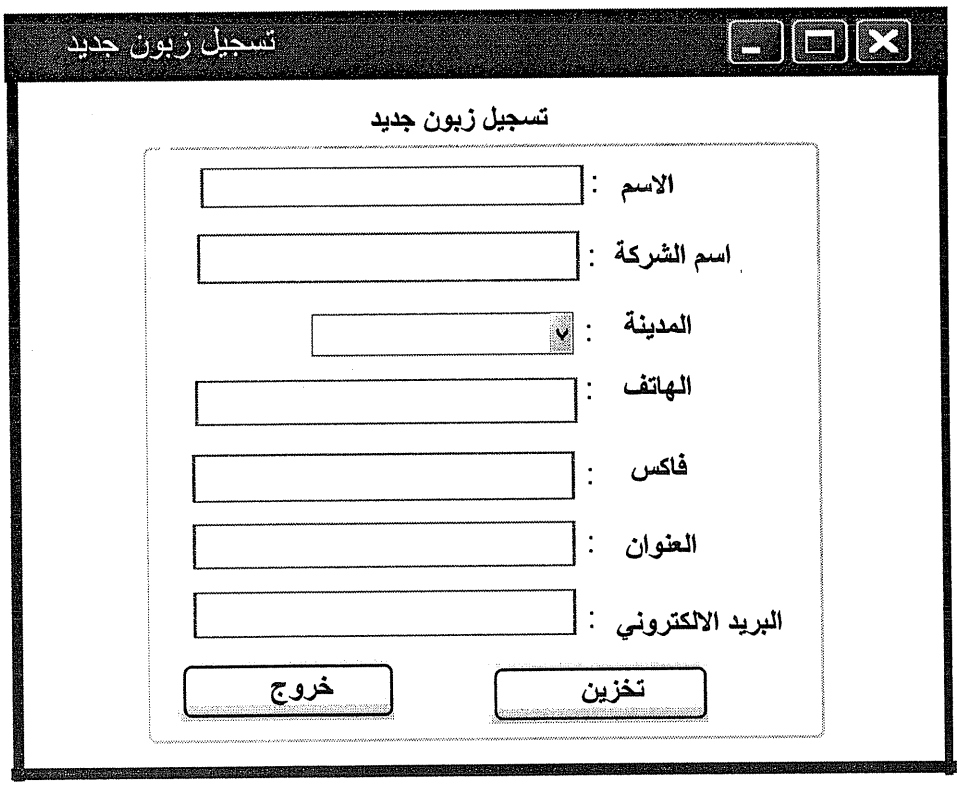


Figure (4.67) registration screen.

b) visitor Output design:

- Displaying Company Product Form:

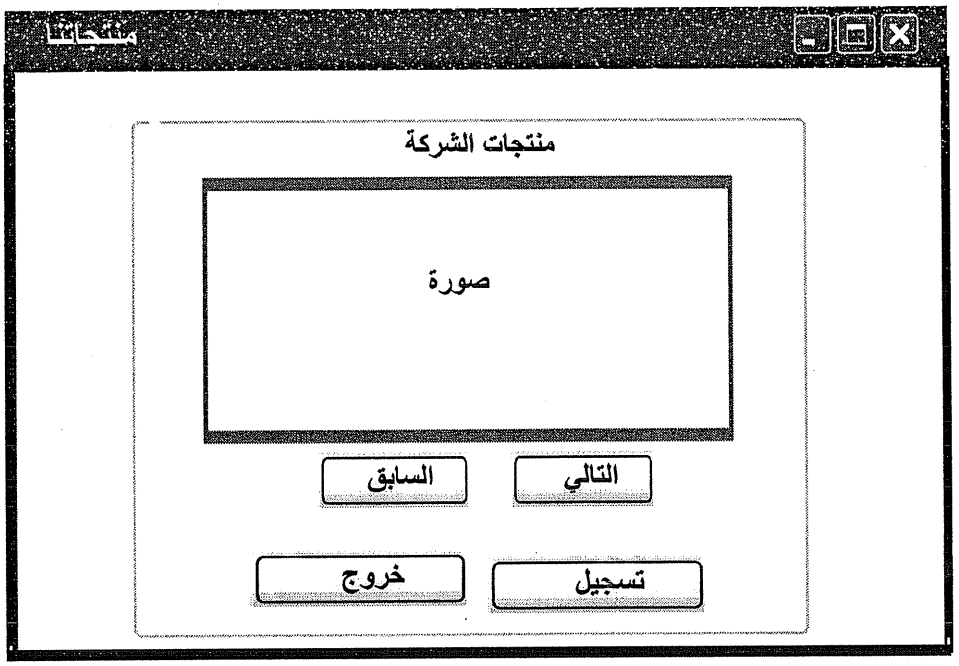


Figure (4.68) displaying company product screen.

4.3.1.2 Customer interface design:

a) *customer Input design:*

- Customer Login Form:

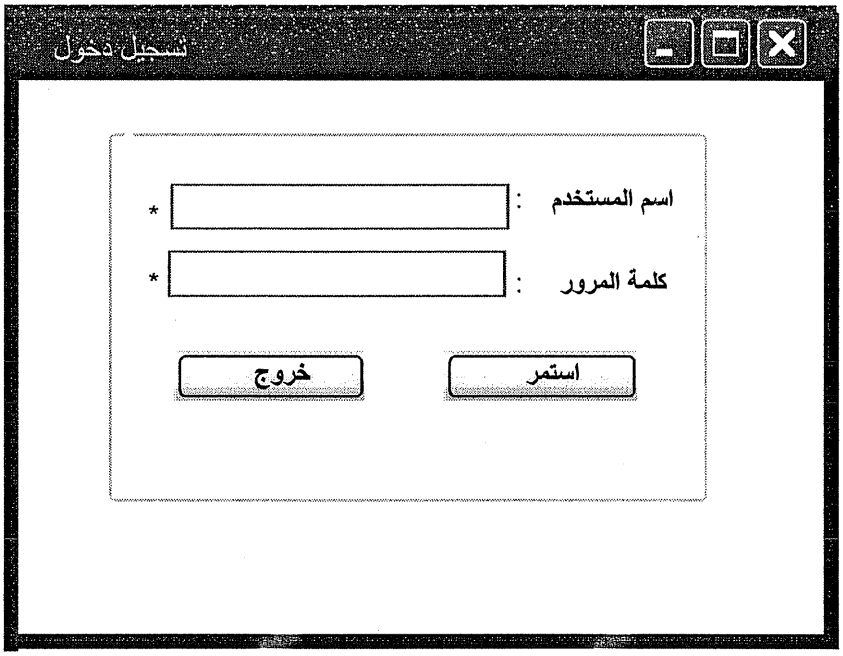


Figure (4.69) customer login screen.

➢ Chang Customer Password Form :

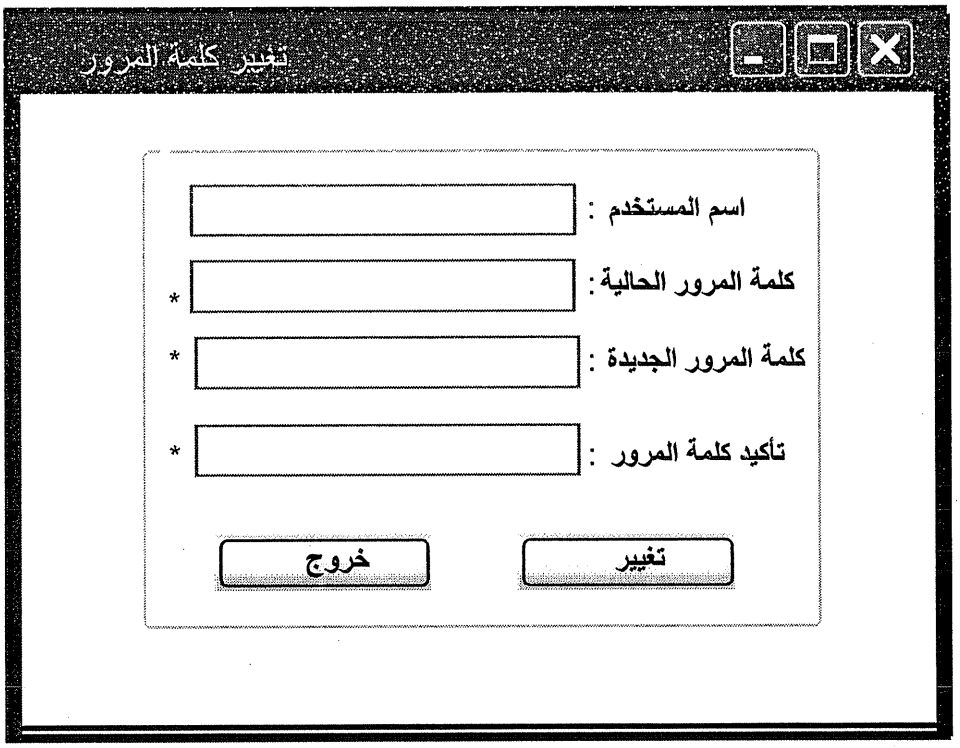


Figure (4.70) change customer password screen.

➢ Customer Comment Form:

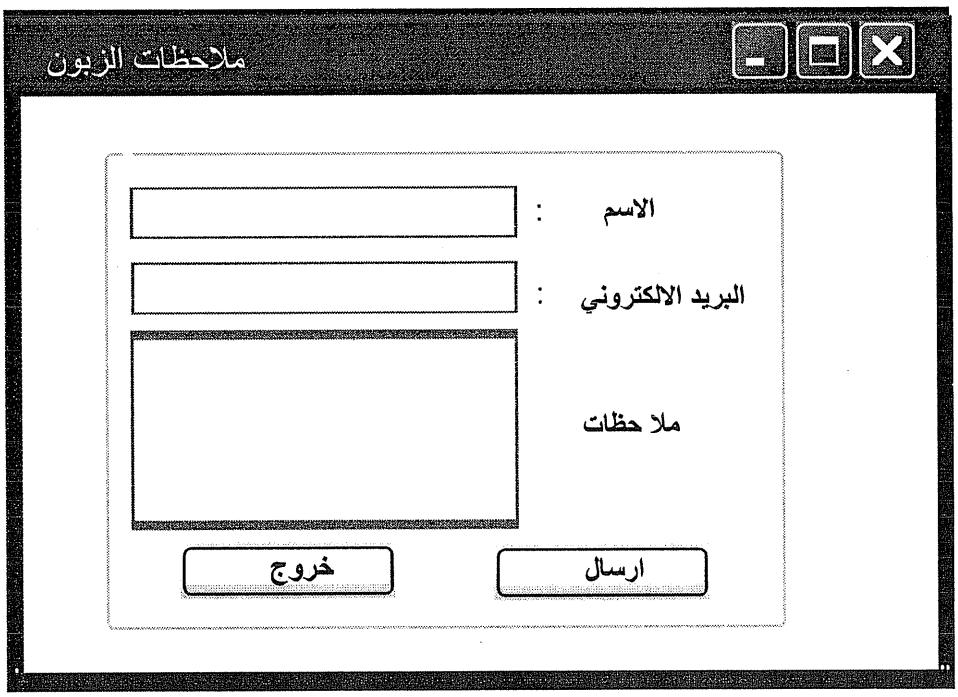


Figure **(4.71)** customer comment screen.

- Generate Order form:

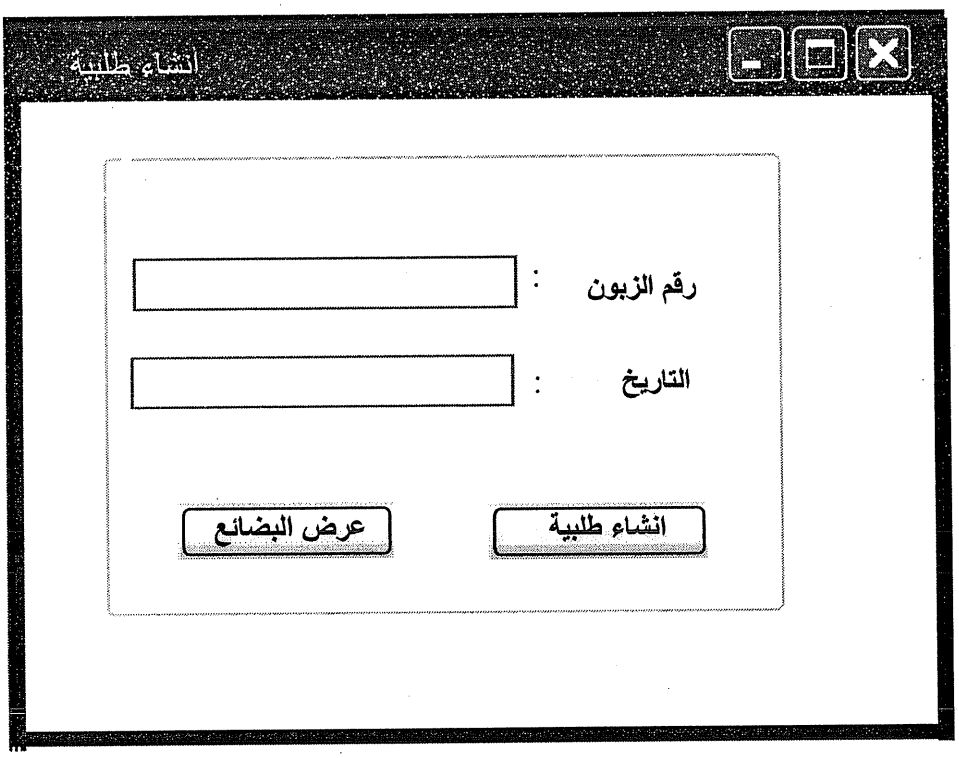


Figure (4.72) generate order screen.

- Select Item to Order.

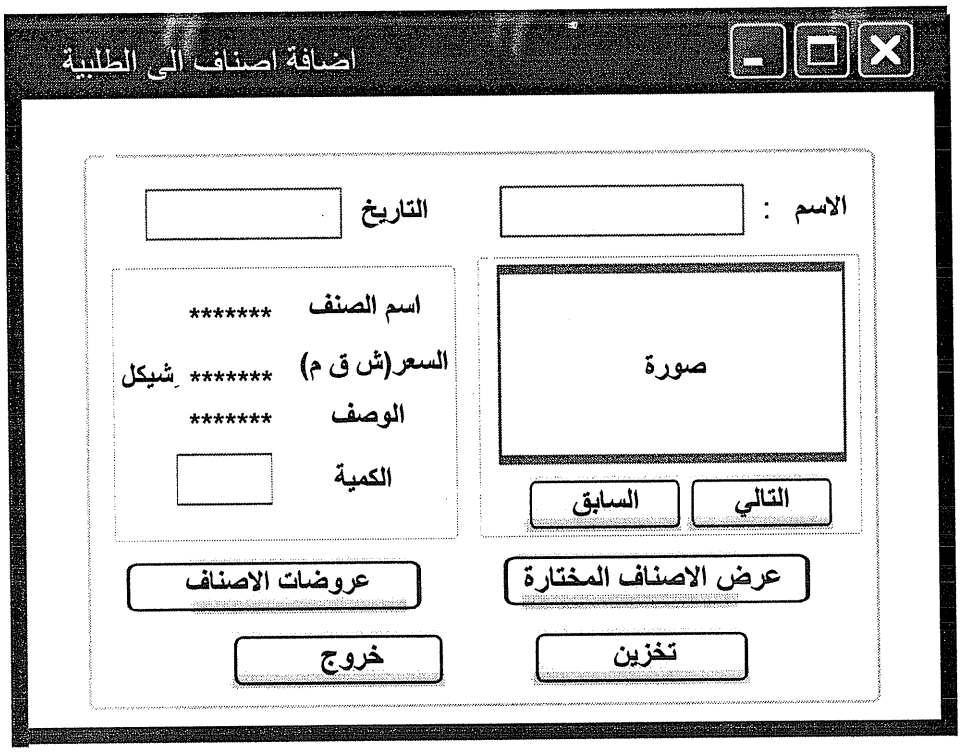


Figure (4.73) select item screen.

➢ Display and update choice item :

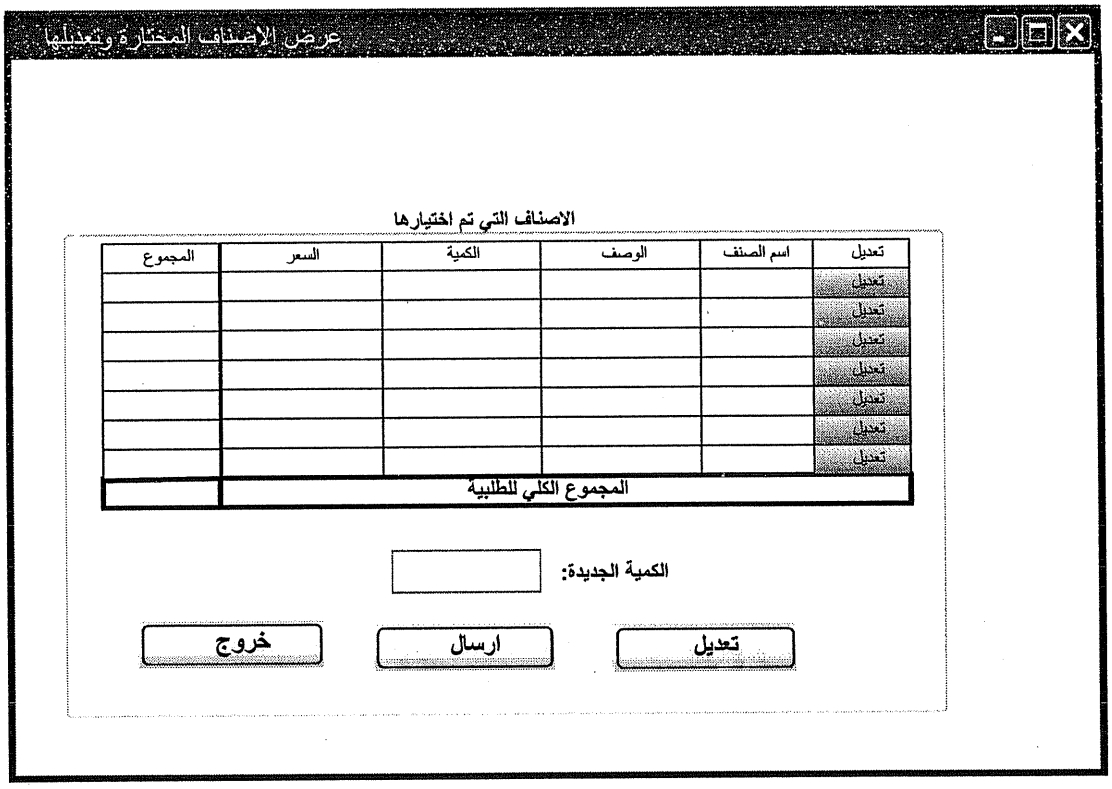


Figure (4.74) Display and update choice item screen.

b) *customer Output design :*

- Customer Report Form

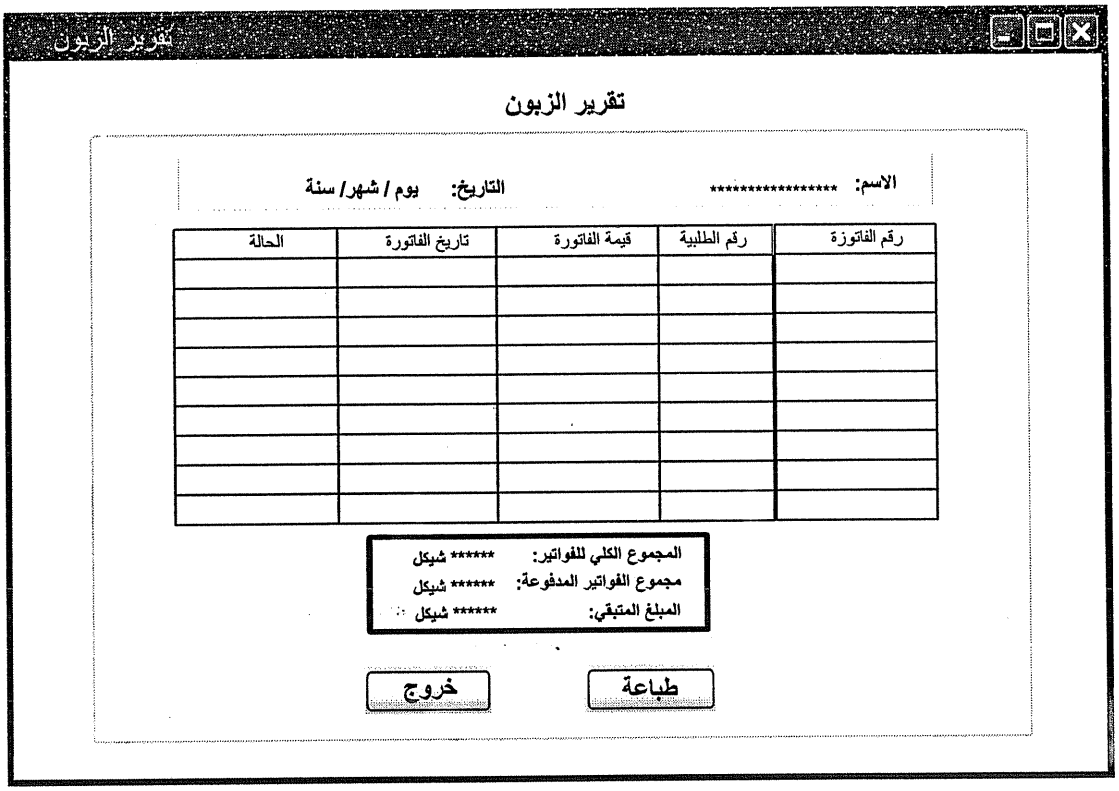


Figure (4.75) customer report screen.

- Invoice for paid form:

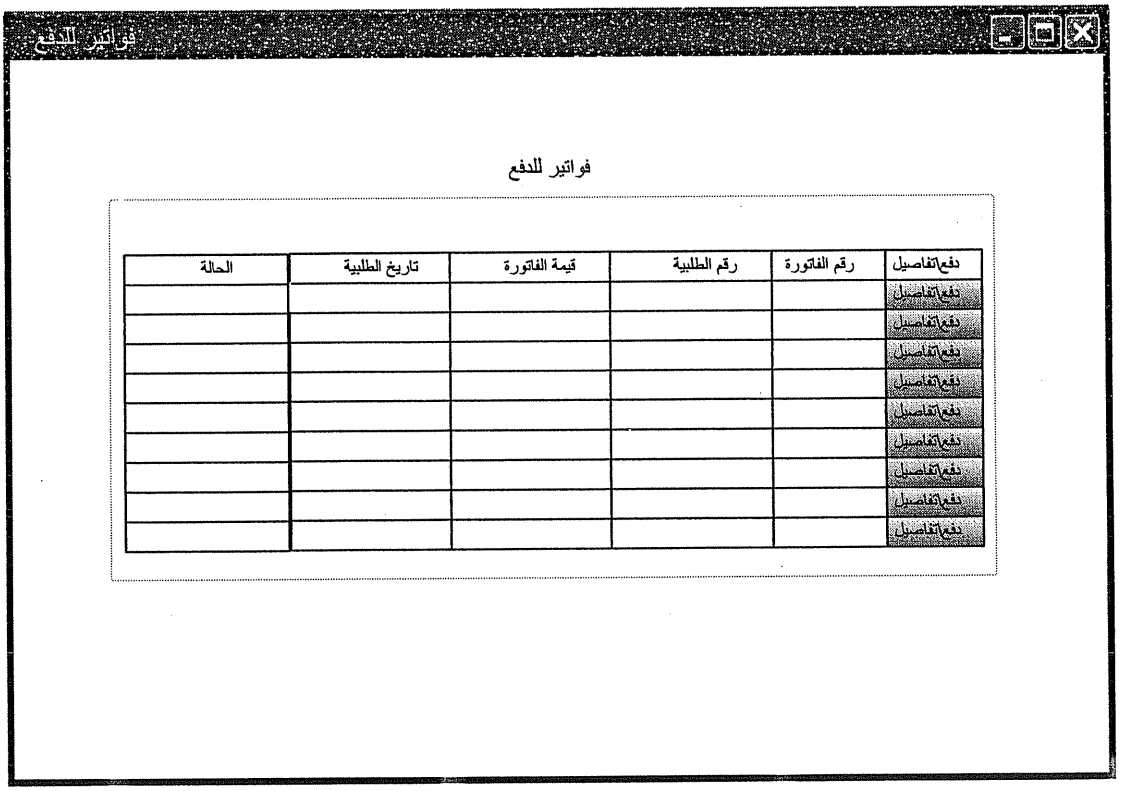


Figure (4.76) Invoice for paid screen

- Invoice

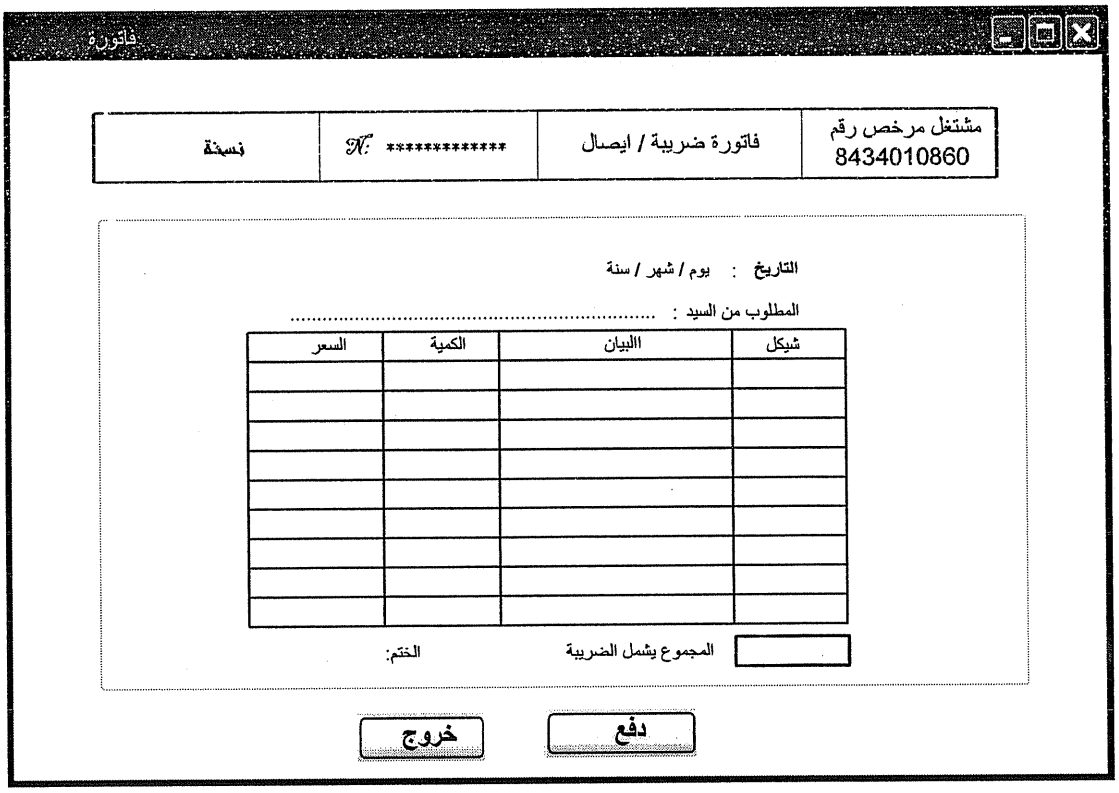


Figure (4.77) invoice screen.

- Billing form

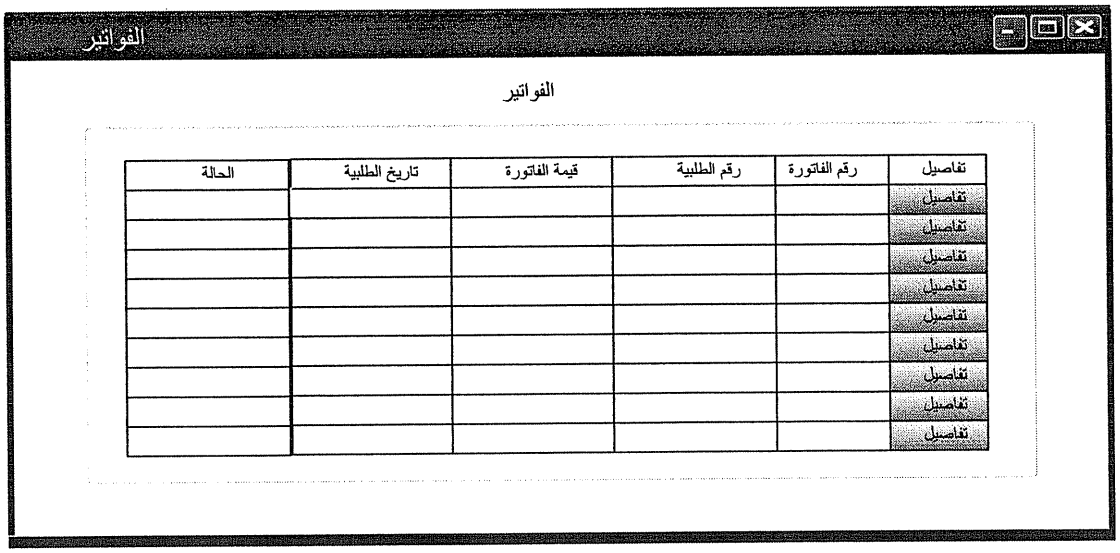


Figure (4.78) billing screen.

4.3.1.3 Administrator interface design:

a) Administrator Input / output design:

- Login administrator :

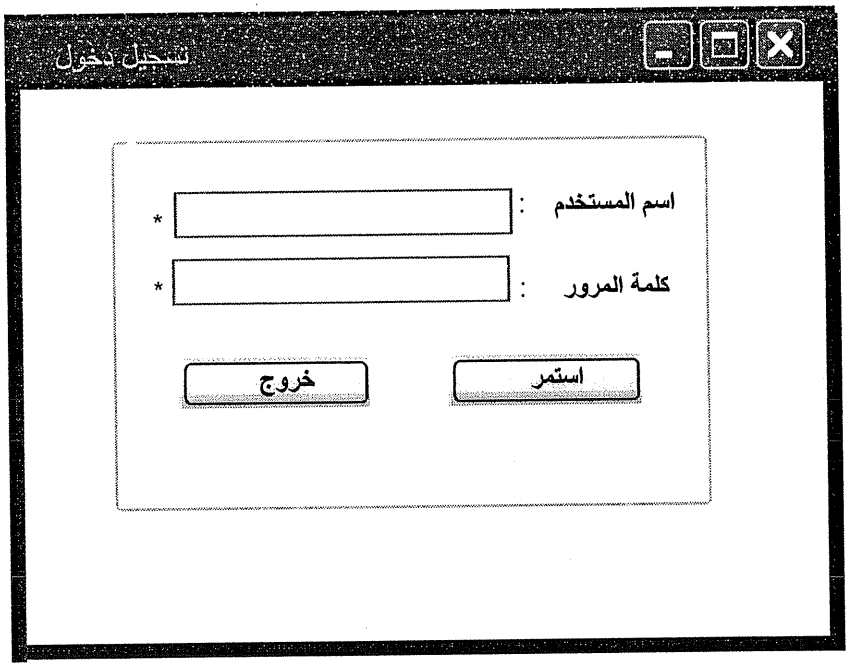


Figure (4.79) administrator login screen.

- Chang Administrator Password Form

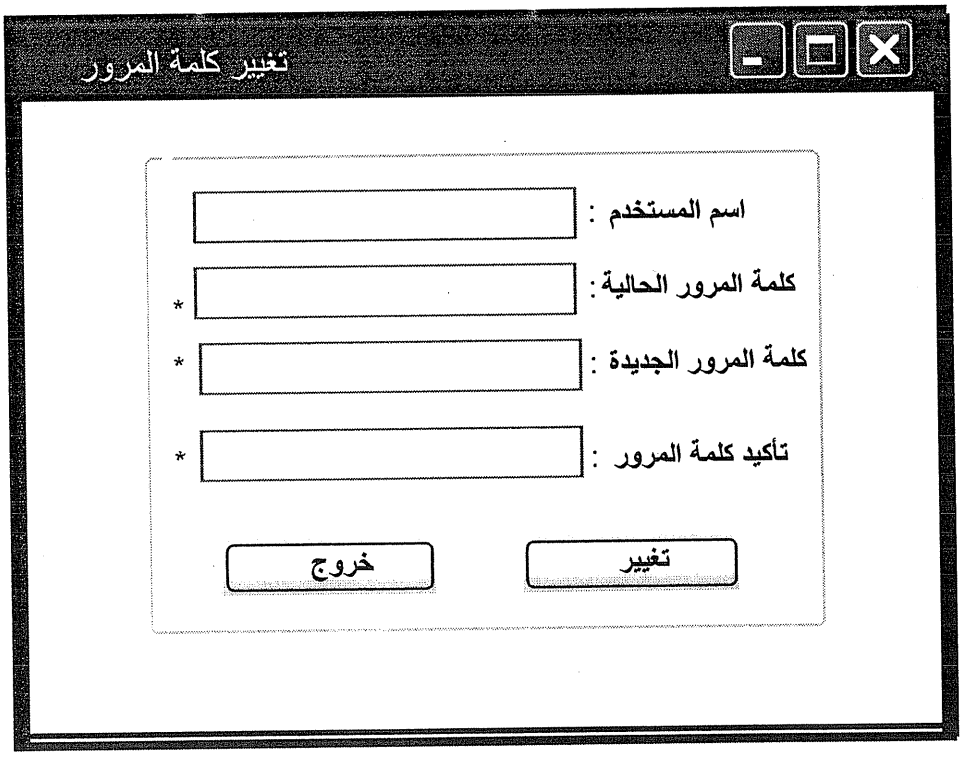


Figure (4.80) change password for Administrator:

- Add New Category Form

Figure (4.81) add new category screen:

- Add New Item Form.

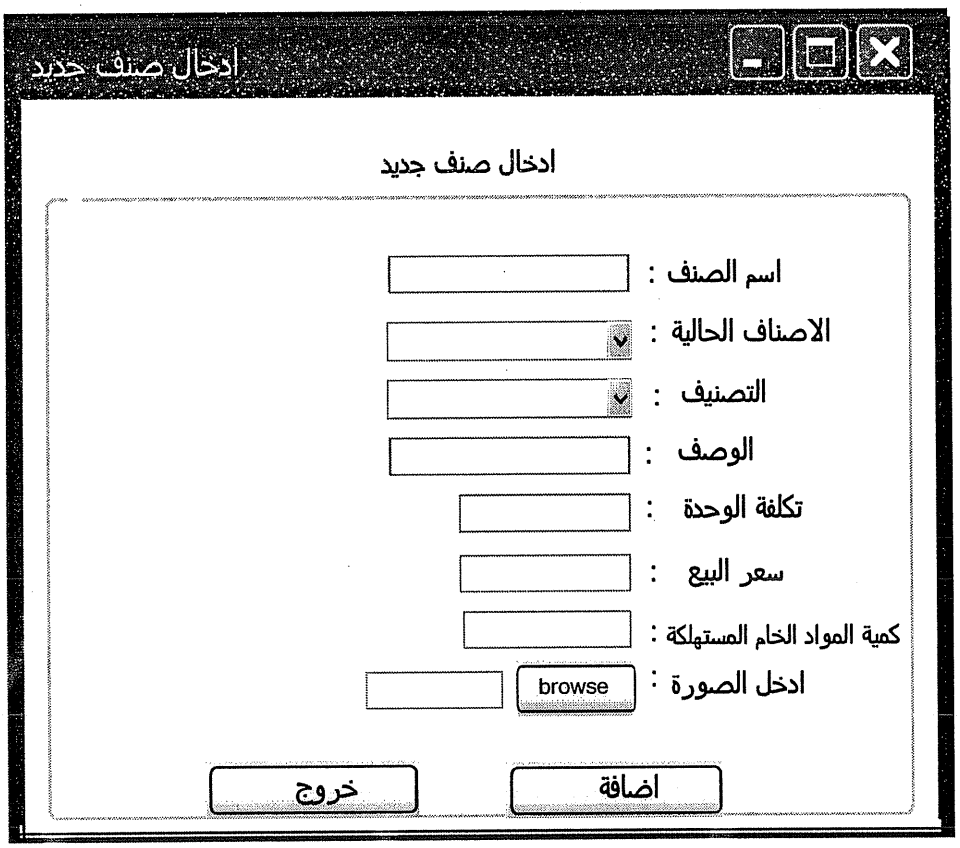


Figure (4.82) add new item screen:

- Add New Material Form

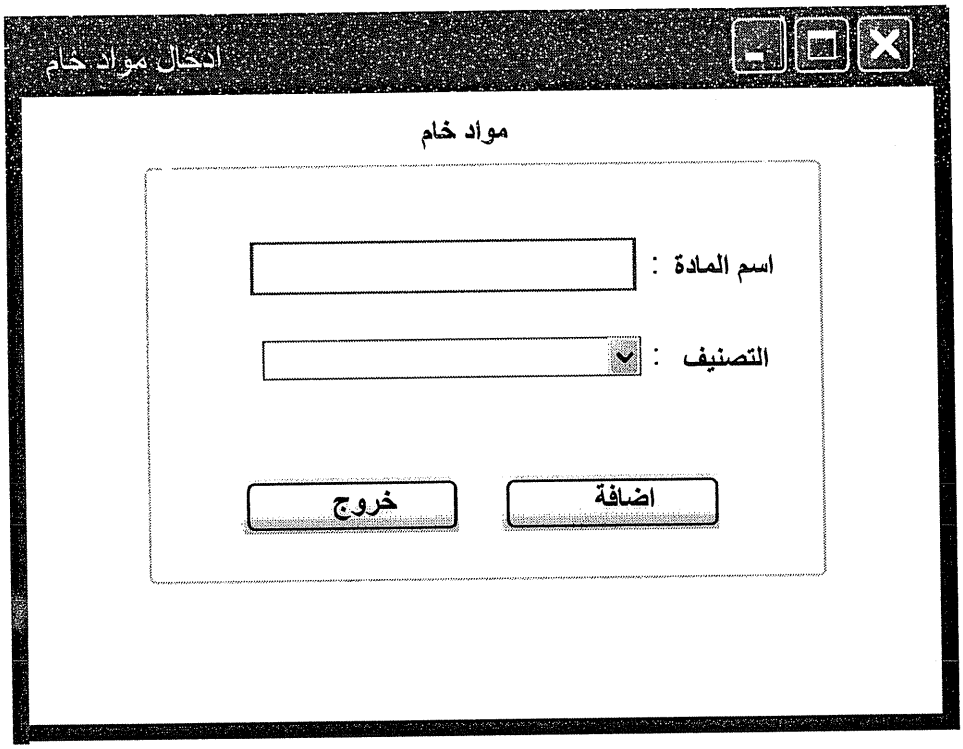


Figure (4.83) add new material screen

- Add New City Form

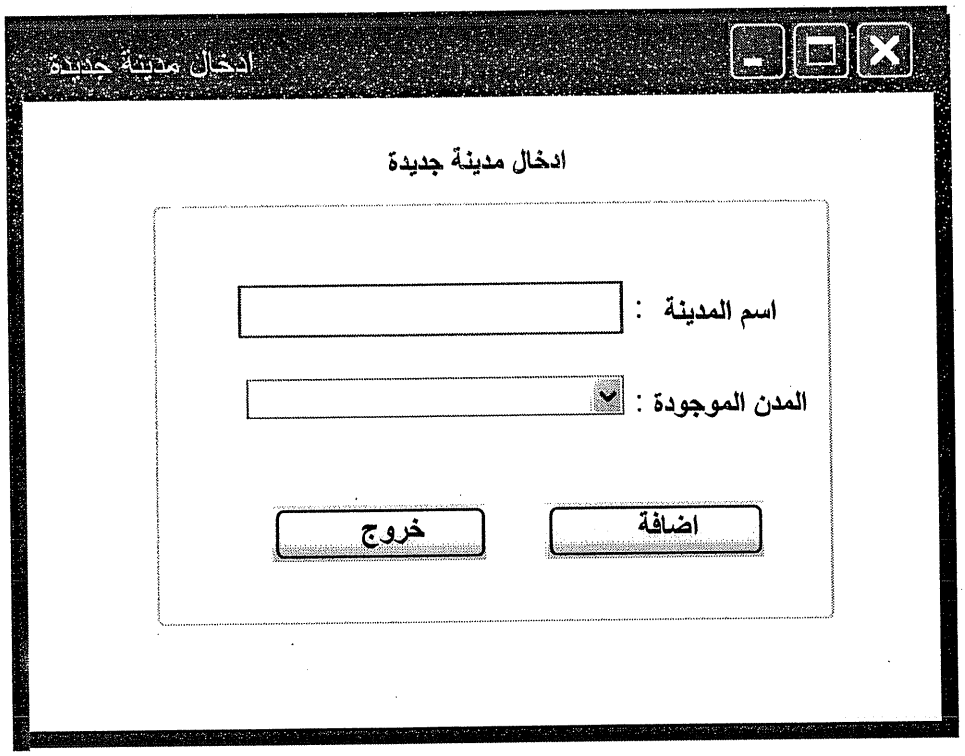


Figure (4.84) add new city screen

- Add New Department Form

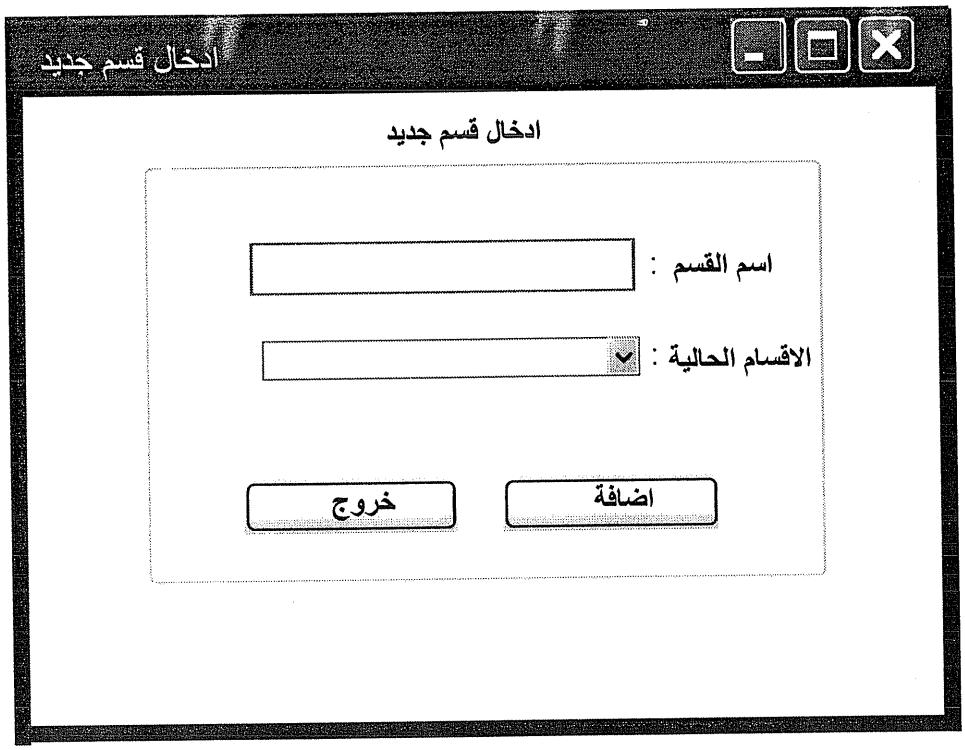


Figure (4.85) add department screen

➢ Add New Quantity Material Form

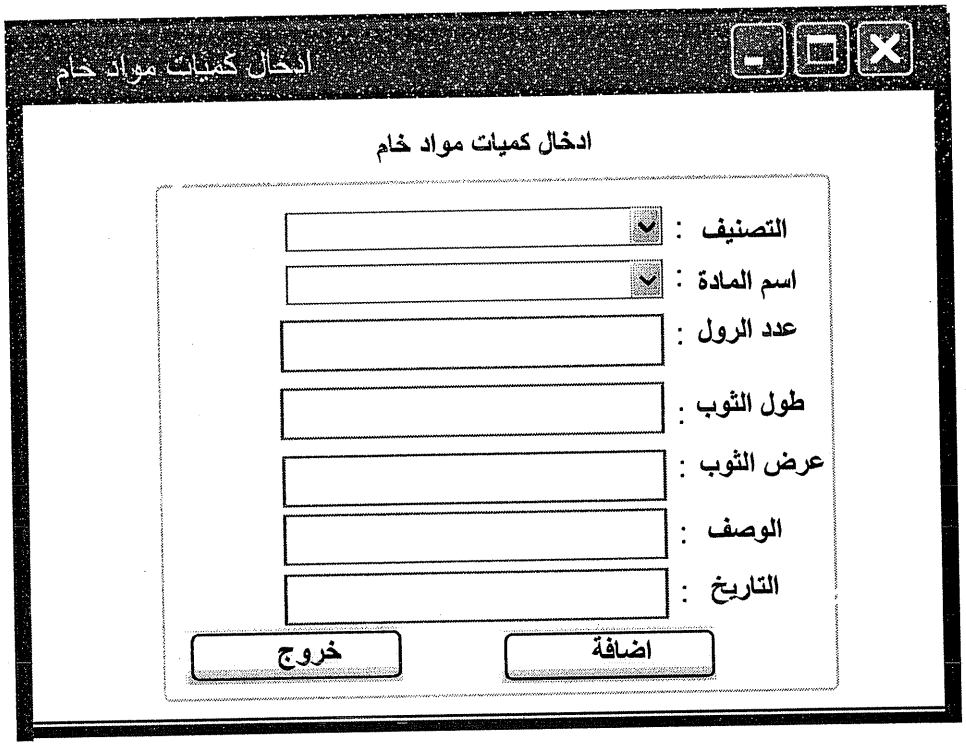


Figure (4.86) add new quantity material screen

➢ Manufacturing Items Form

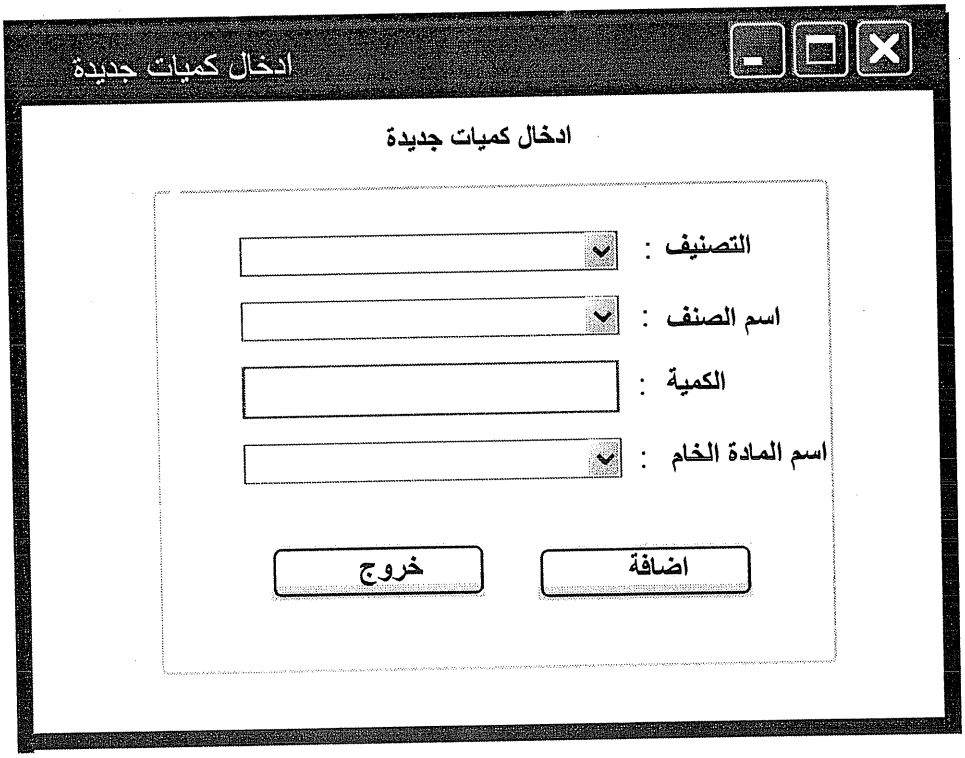


Figure (4.87) Manufacturing items screen

➢ Add New Employee Form

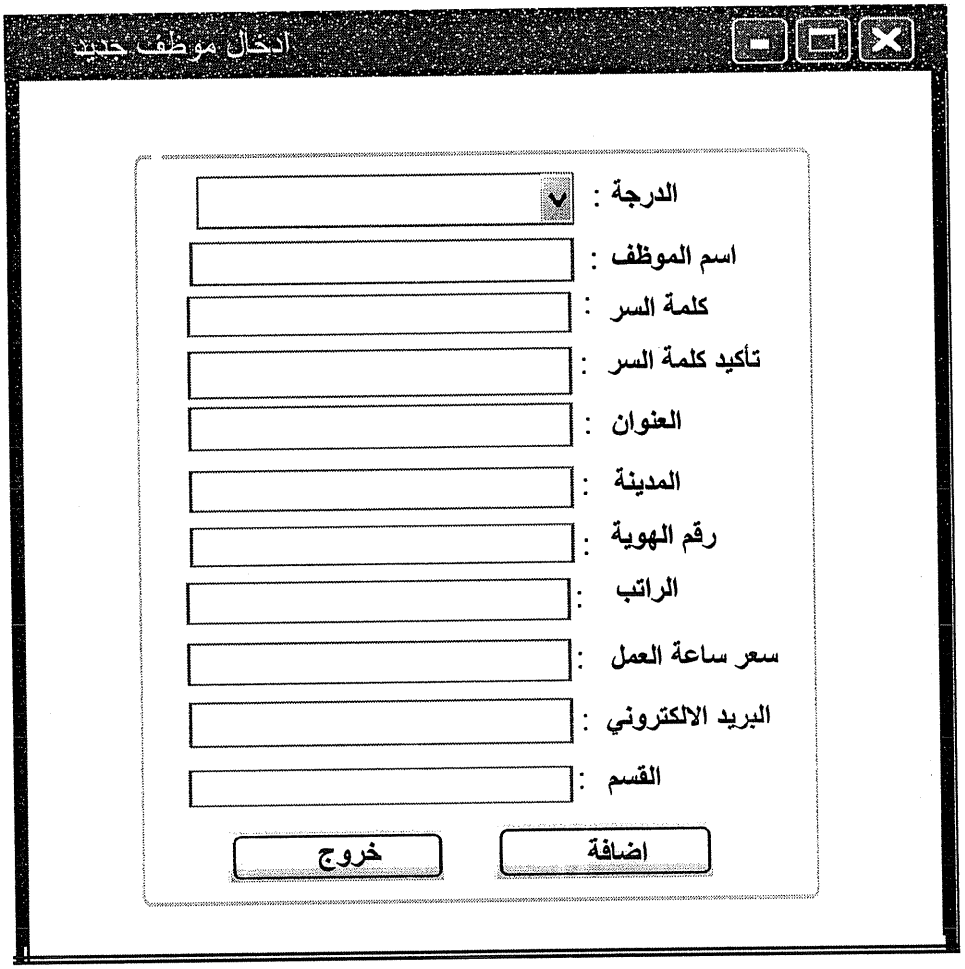


Figure (4.88) add new employee screen

- Add new Advertisement Form.

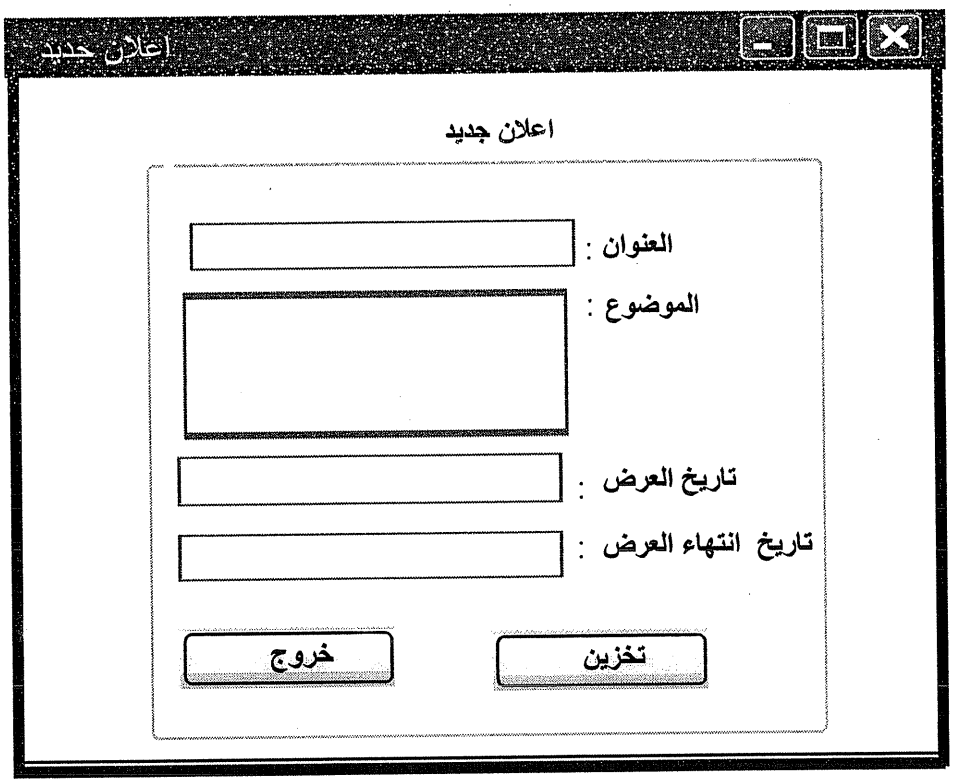


Figure (4.89) add new advertisement screen

- Register new customer Form

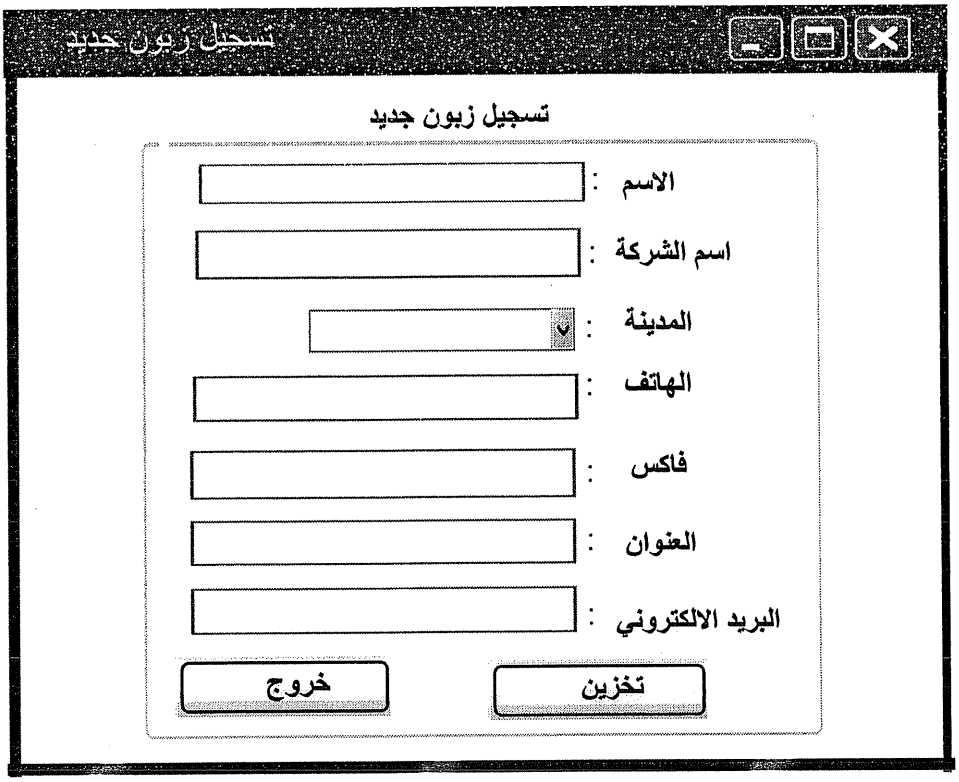


Figure (4.90) Register new customer screen

- Update Item Price Form

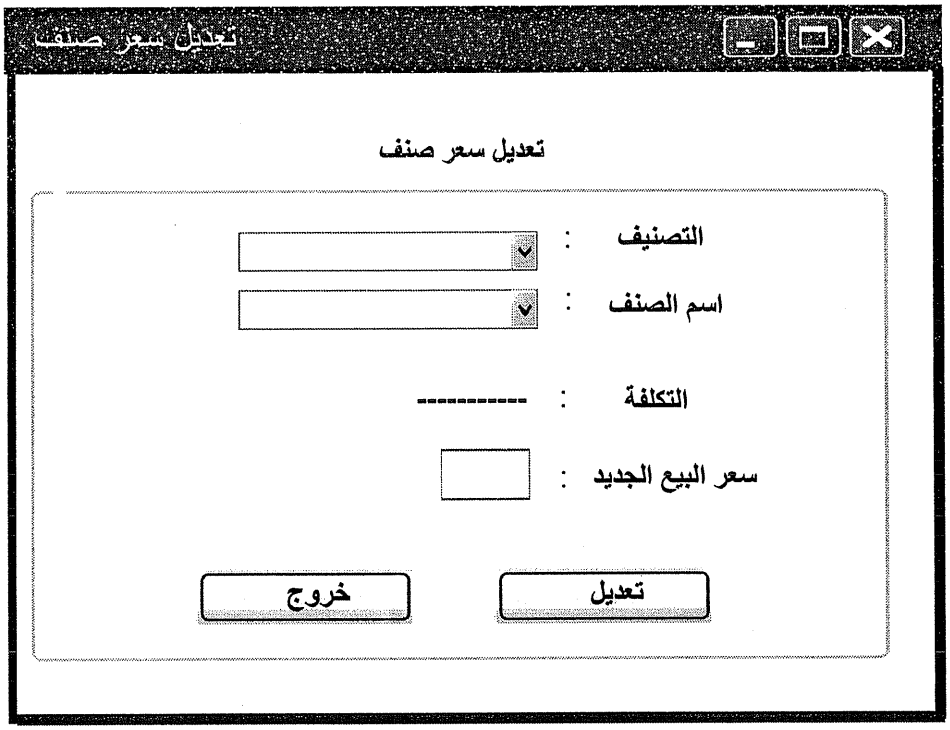


Figure (4.91) update item price screen

- Customer's orders Form

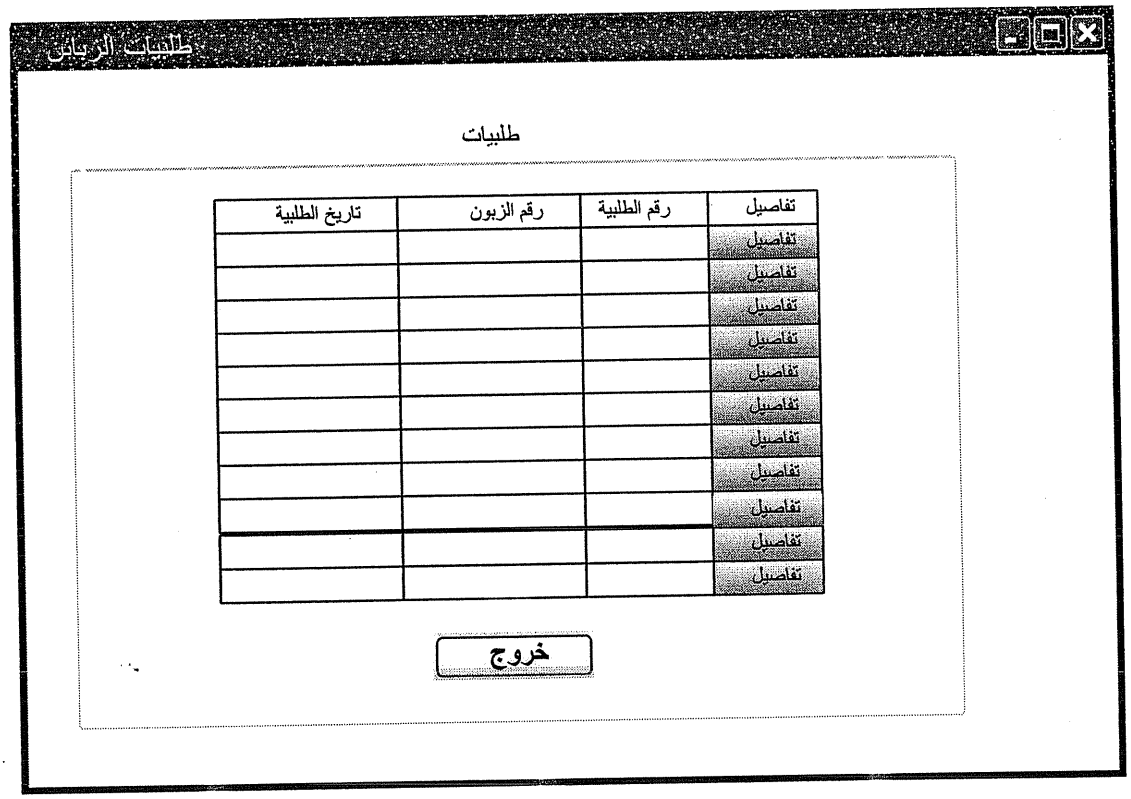


Figure (4.92) customer's orders screen

- Displaying comment

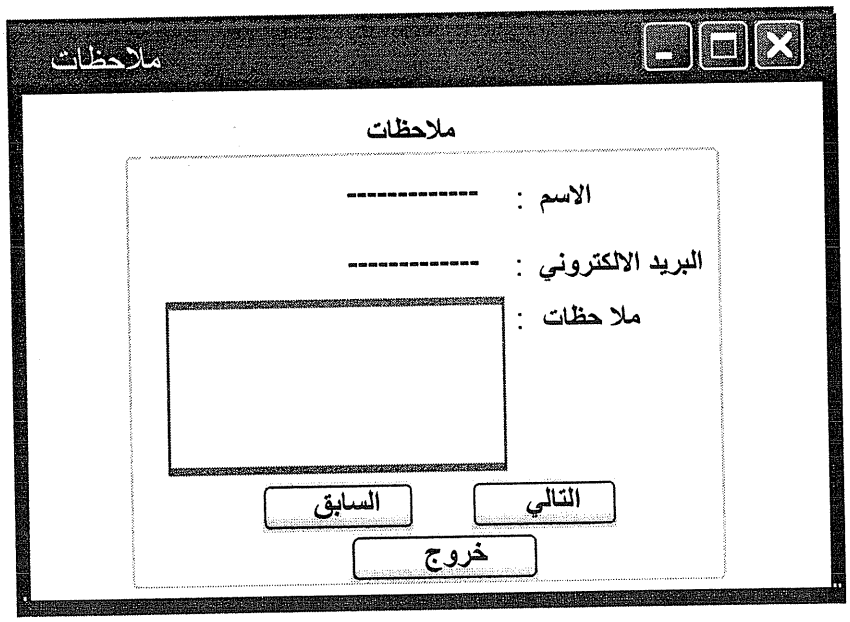


Figure (4.93) displaying comment screen

- Employee Work Record
  - Start Time

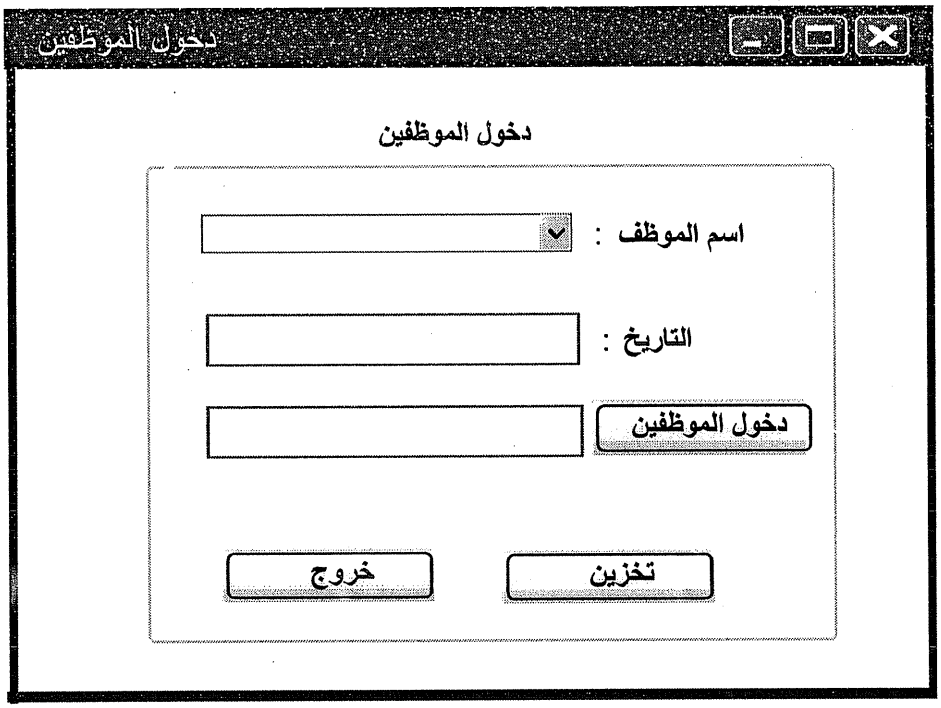


Figure (4.94) start time screen

- End Time

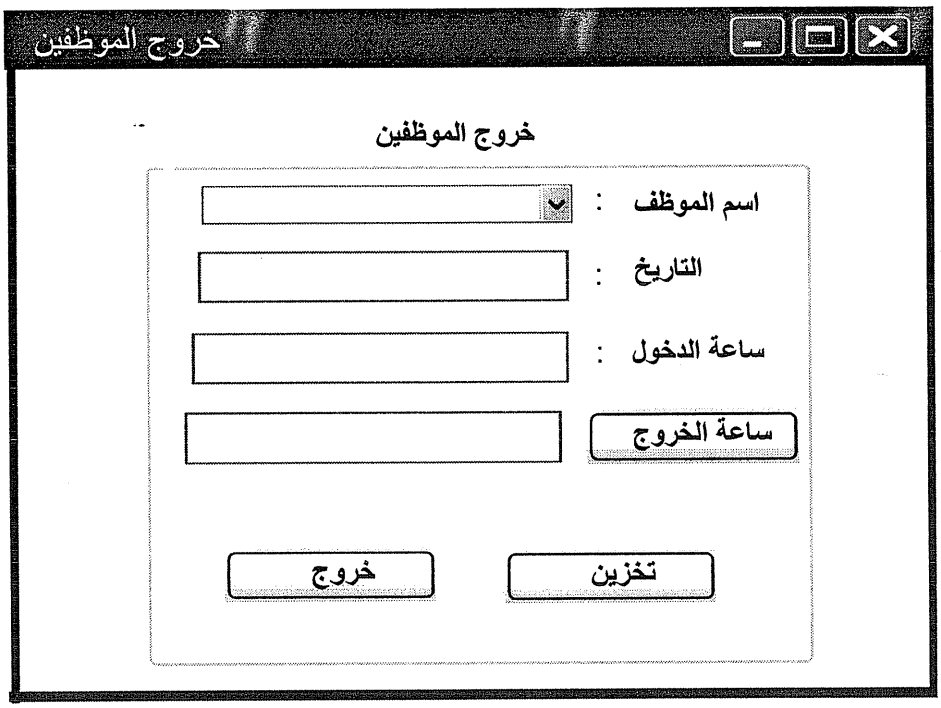


Figure (4.95) end time screen

- Browsing Orders Form

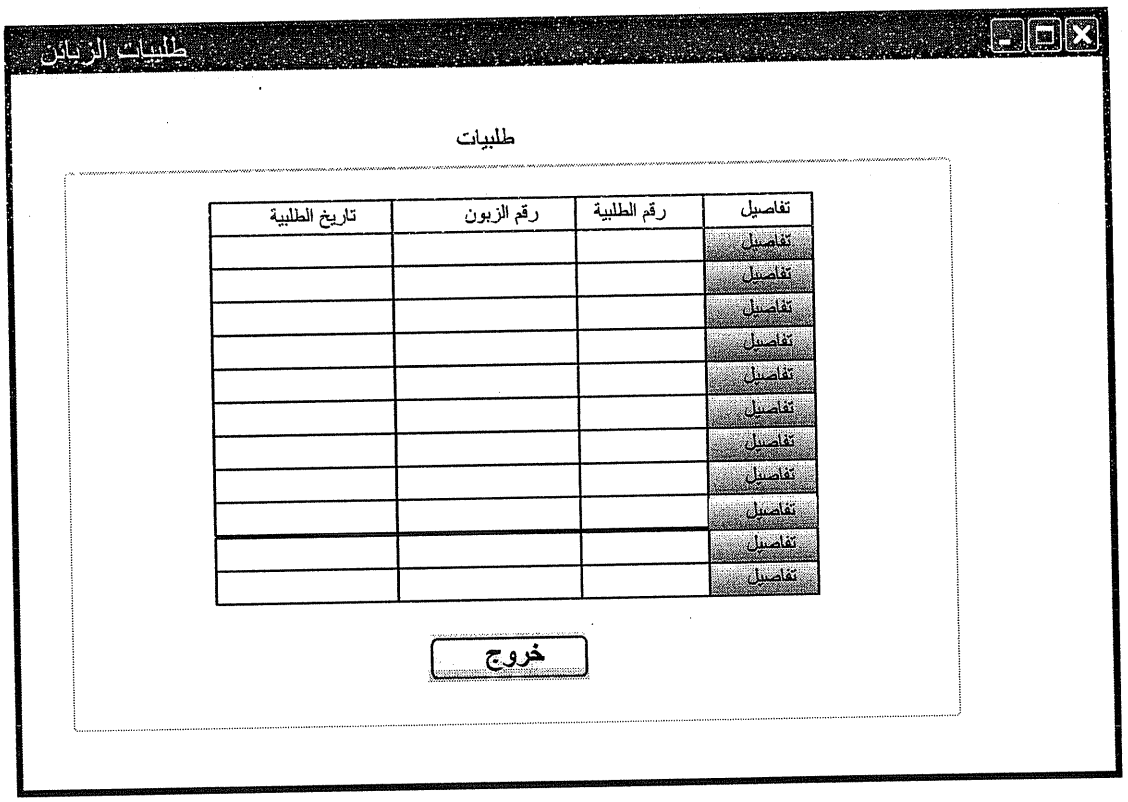


Figure (4.96) browsing order

- Accept Order Form

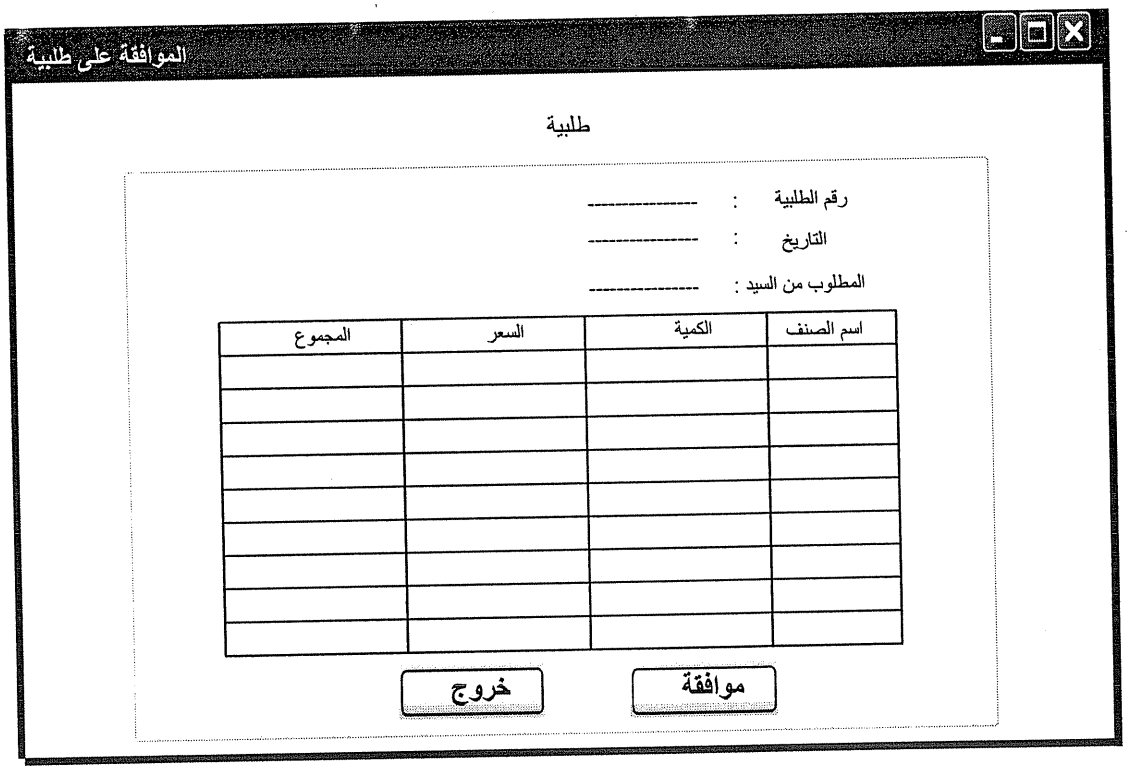


Figure (4.97) accept order screen

➢ Preparing Order Form

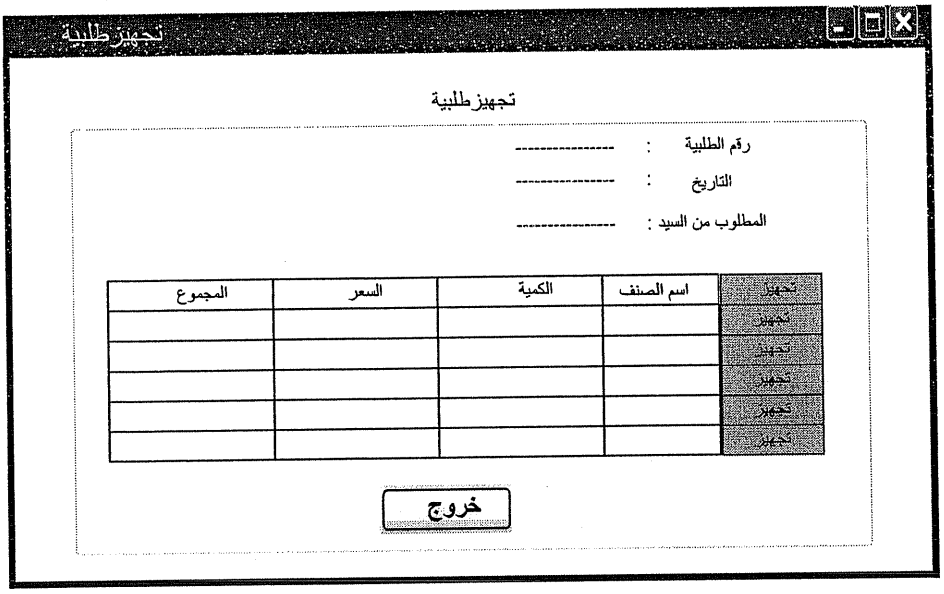


Figure (4.98) Preparing Order screen

- Payment For Employee Form

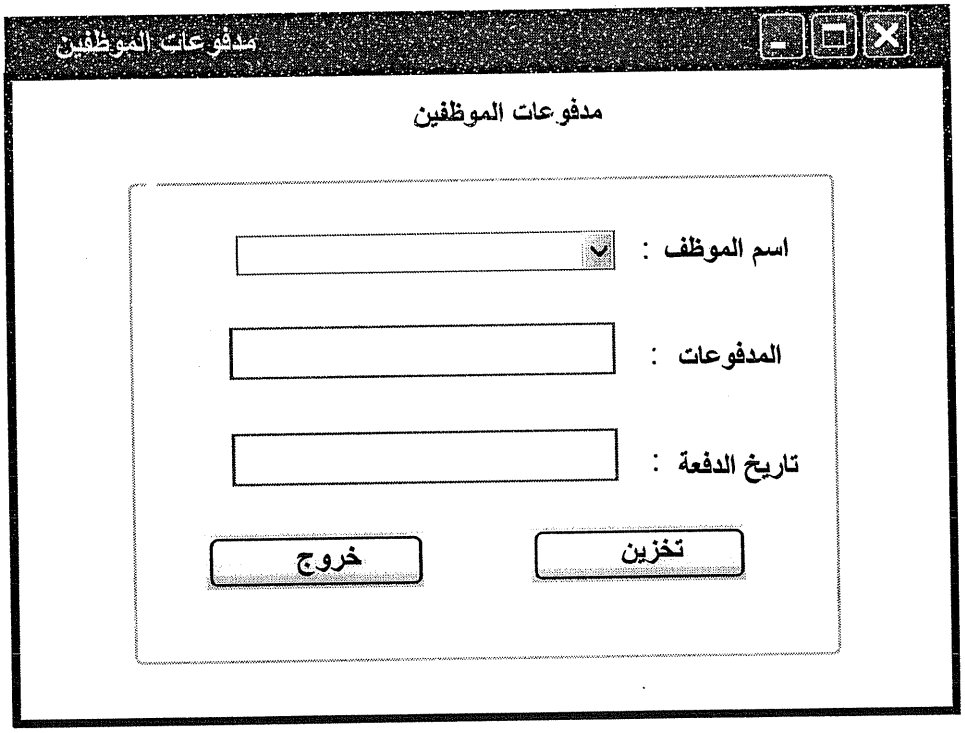


Figure (4.99) payment for employee screen

- Accept Customer Form

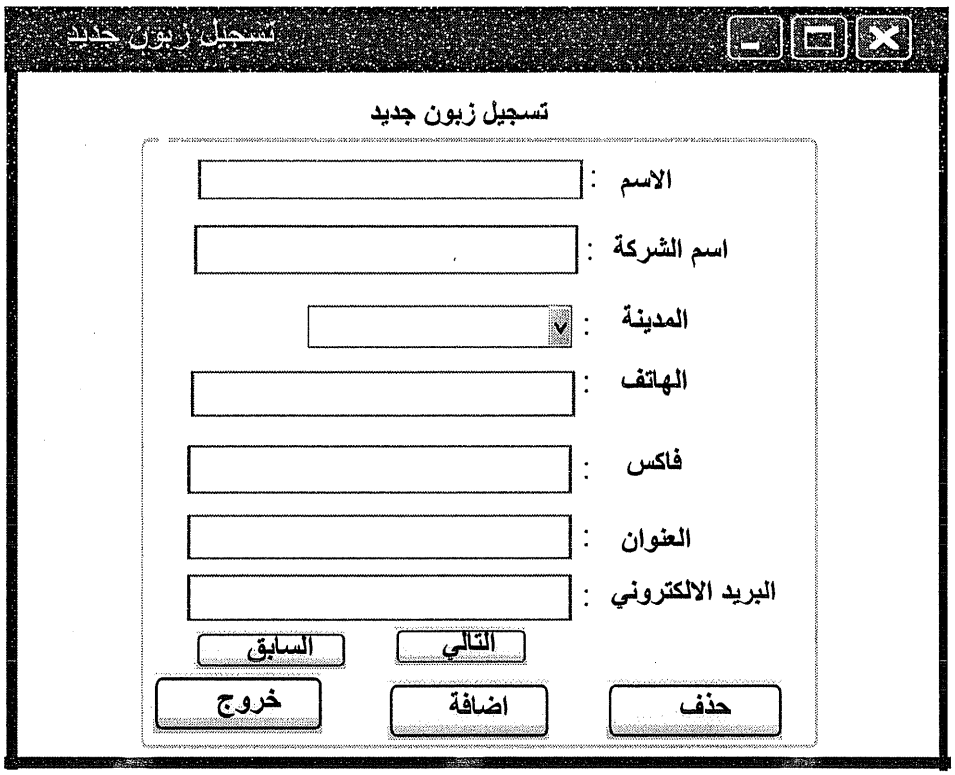


Figure (4.100) accept customer screen

- Store Report

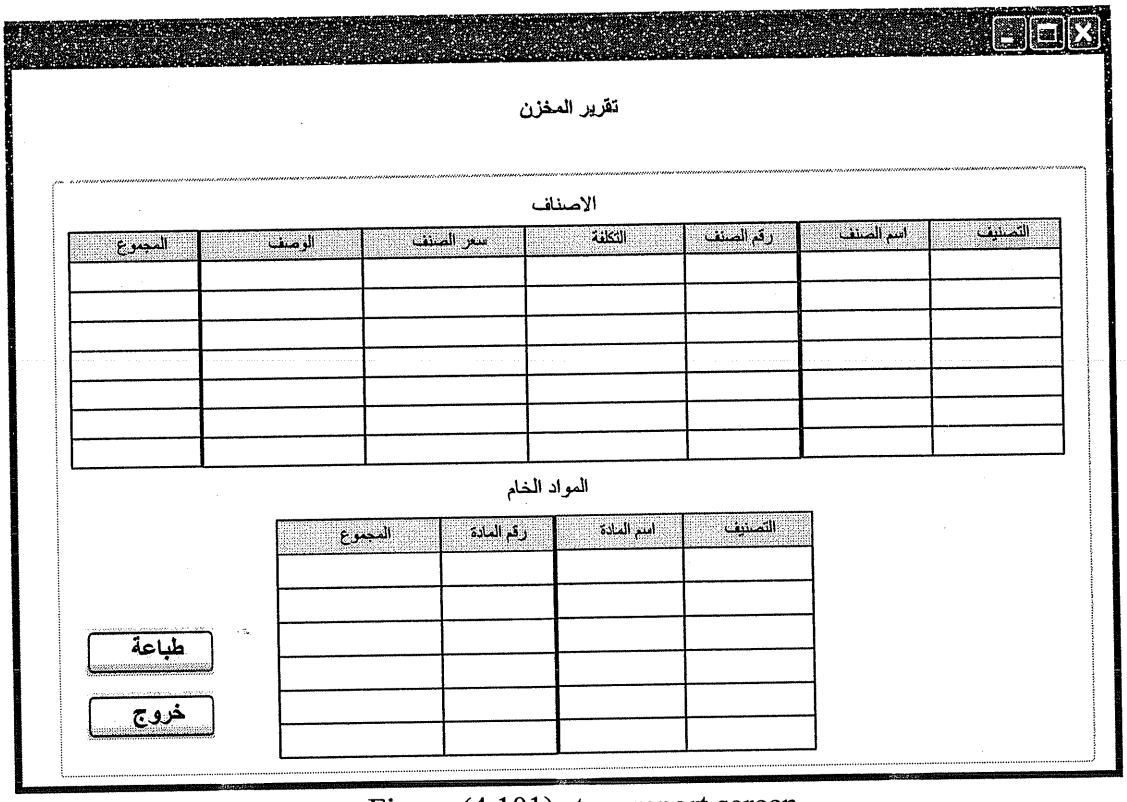


Figure (4.101) store report screen

- Customer Report Form

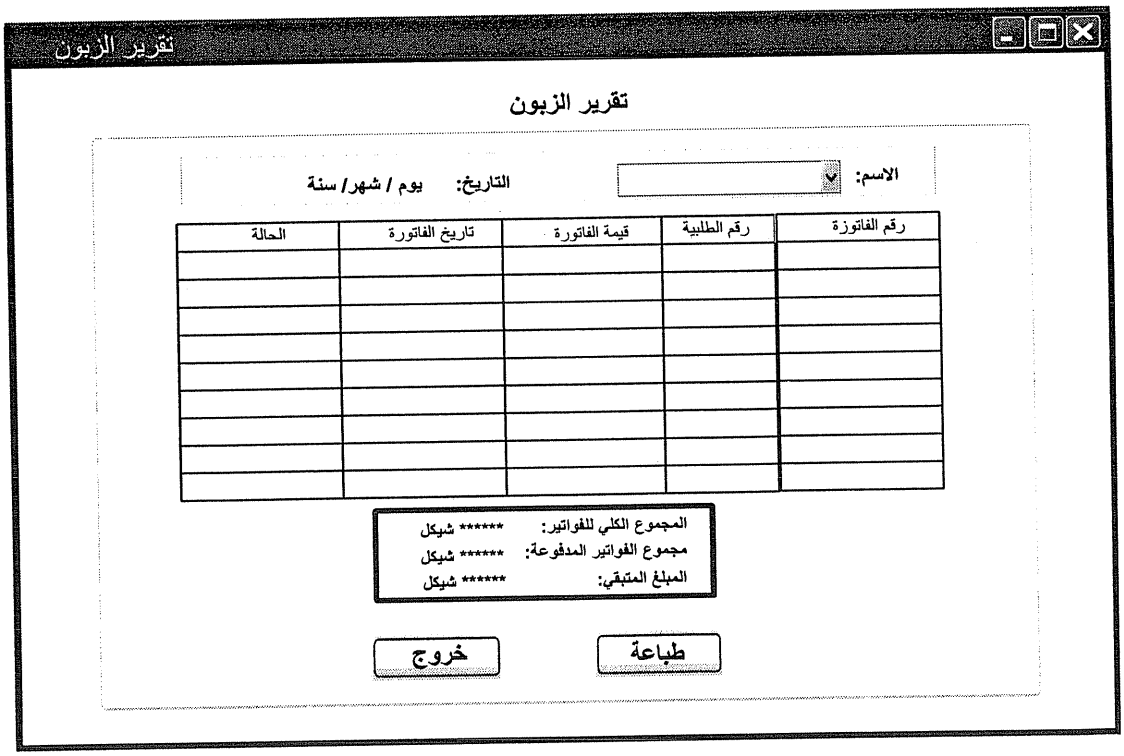


Figure (4.102) customer report screen

- Employee report Form

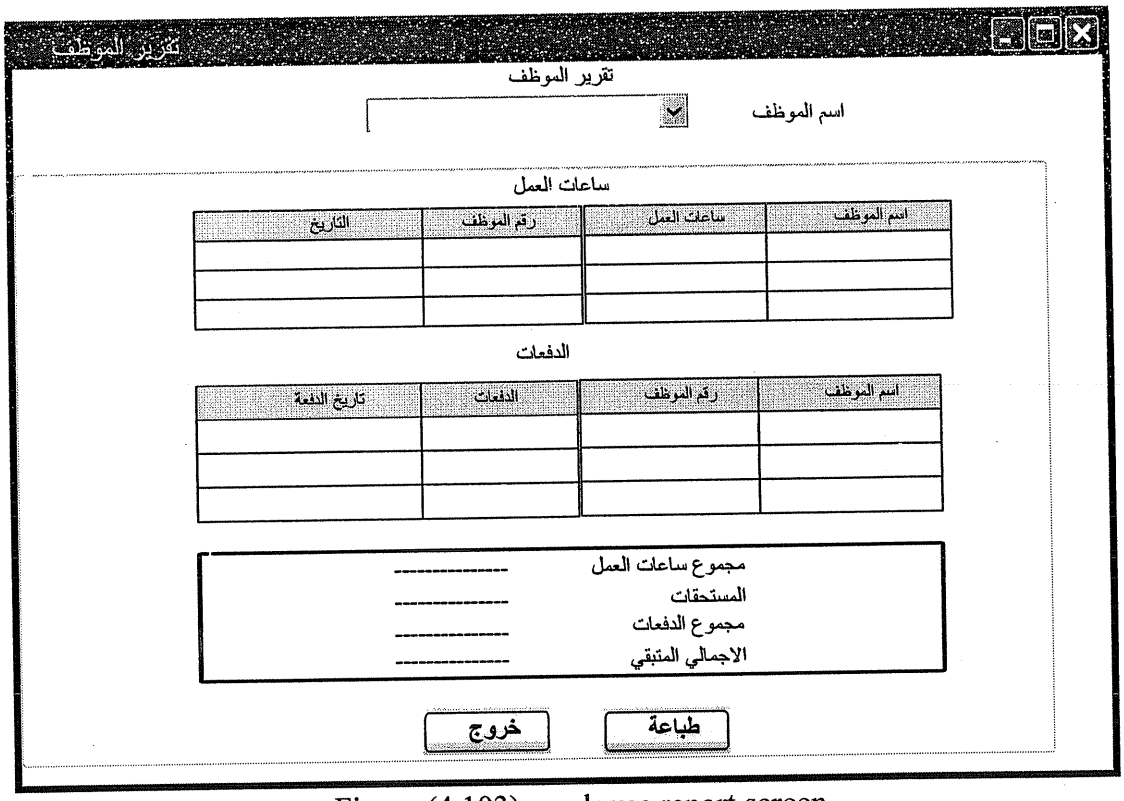


Figure (4.103) employee report screen

4.3.1.4 Employee interface design:

a) employee Input design:

- Login employee Form:

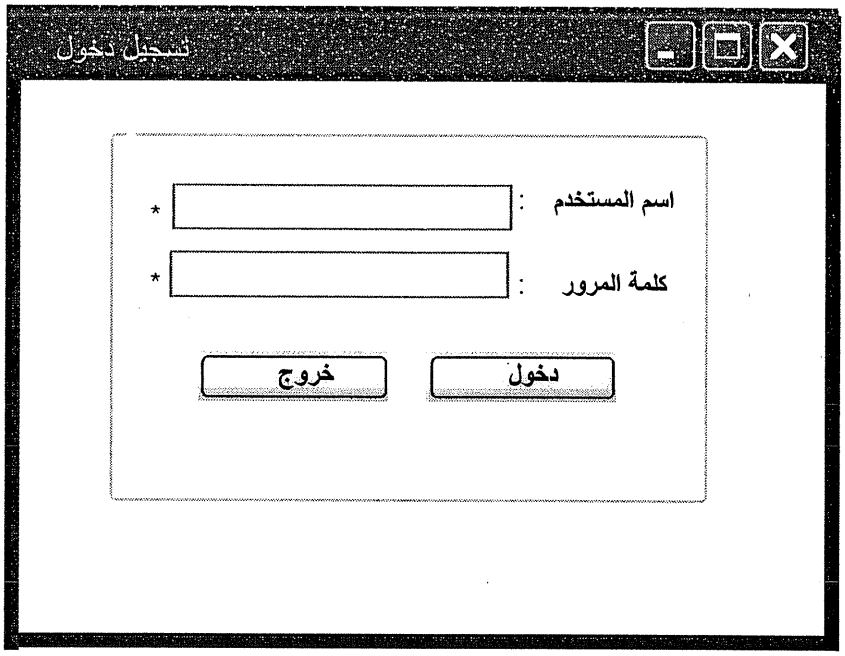


Figure (4.104) employee login screen.

- Chang employee Password

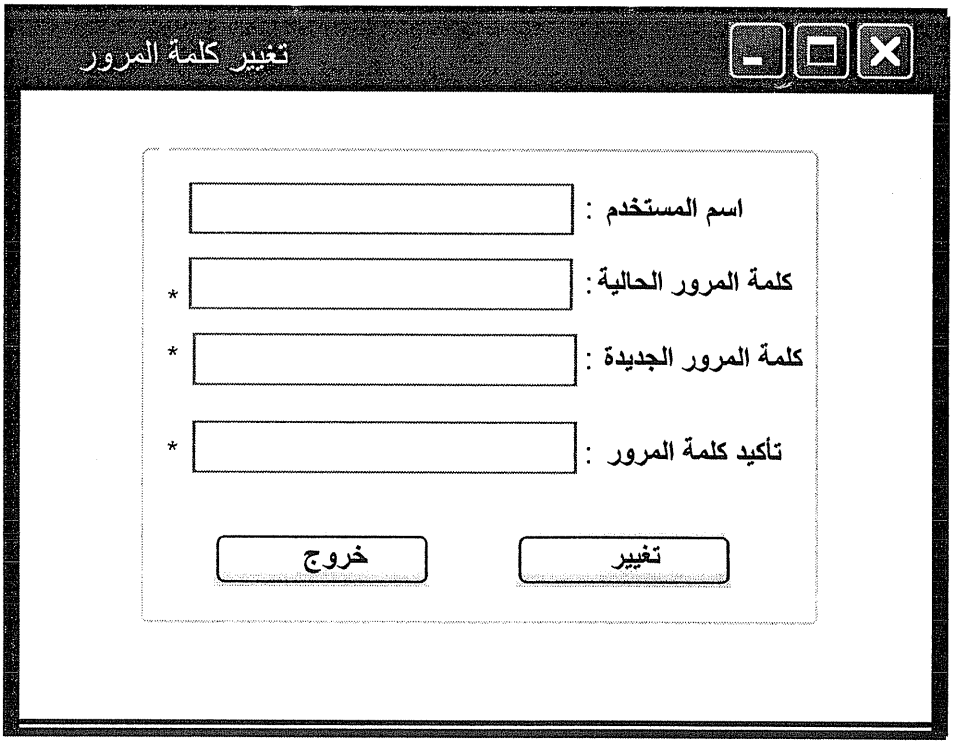


Figure (4.105) change password for employee:

- Employee comment Form

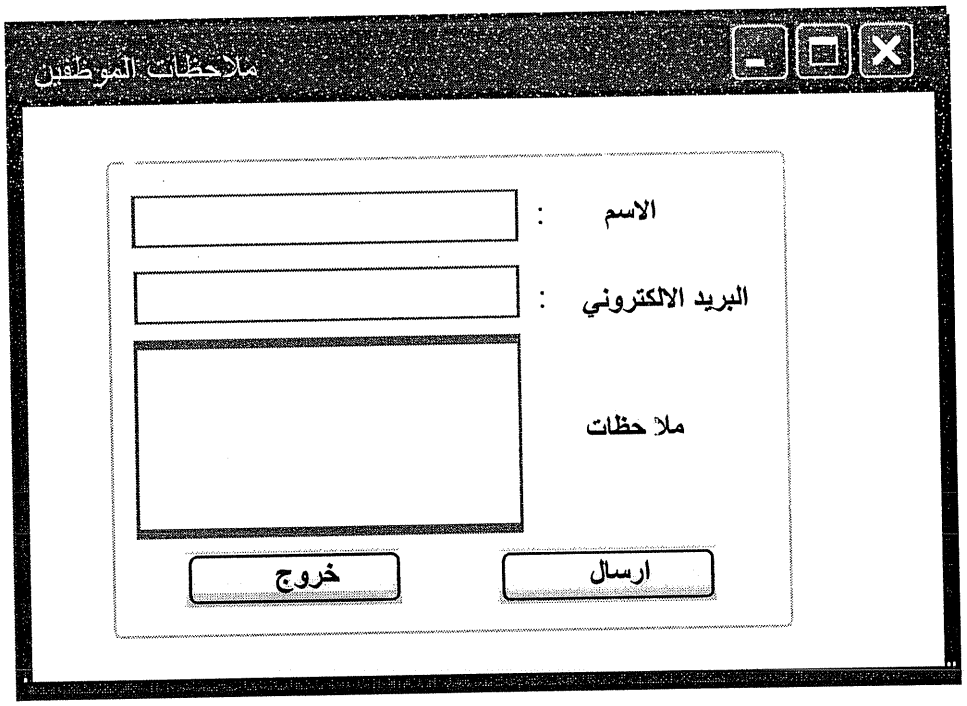


Figure (4.106) employee comment screen

b) employee output design:

- employee report Form:

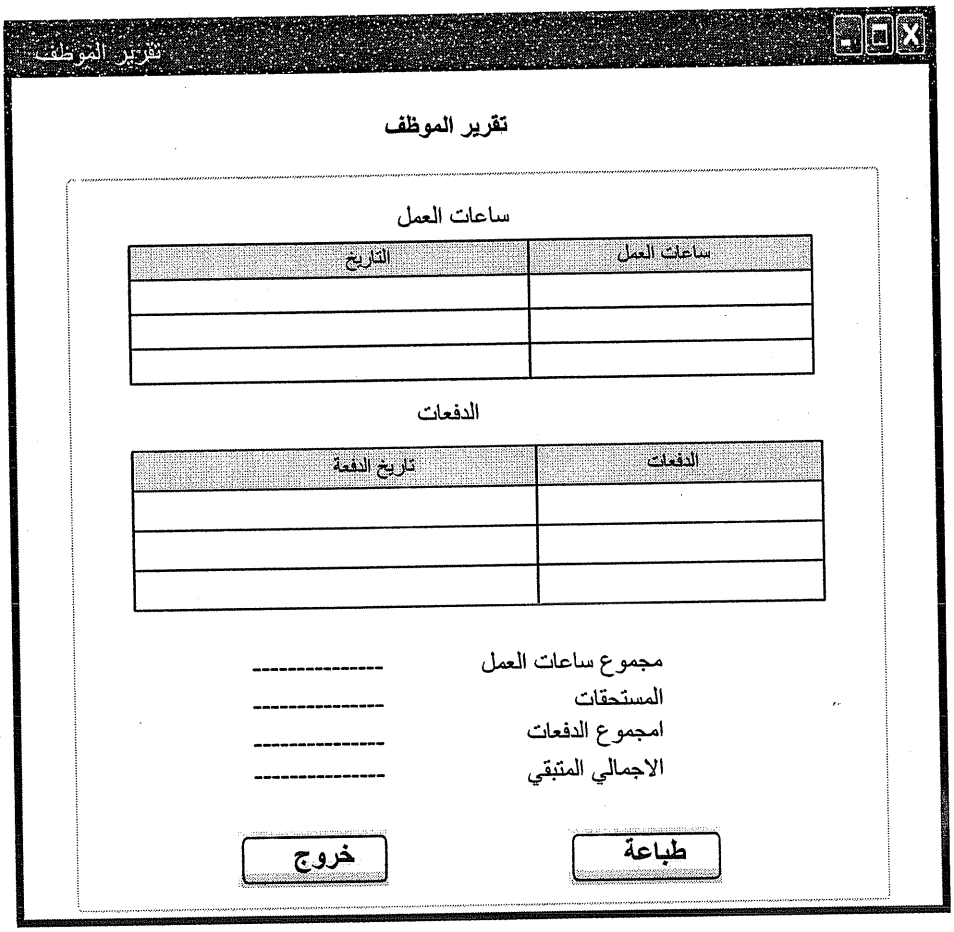


Figure (4.107) employee report screen

- advertisement Form

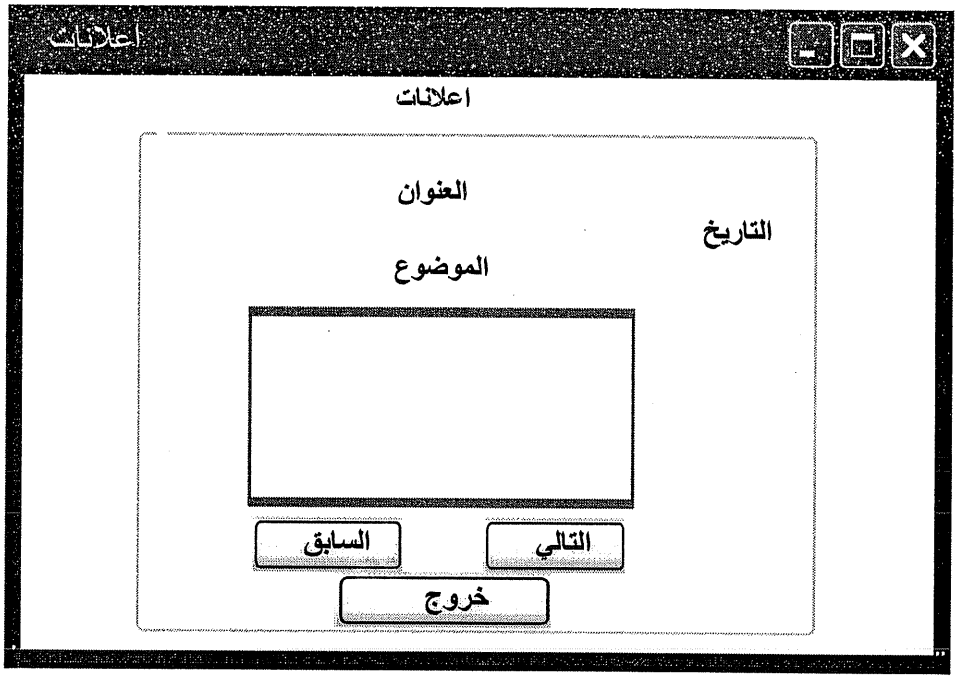


Figure (4.108) advertisement screen

4.3.2 Bank interface design:

a) Bank Input design:

- Login Form:

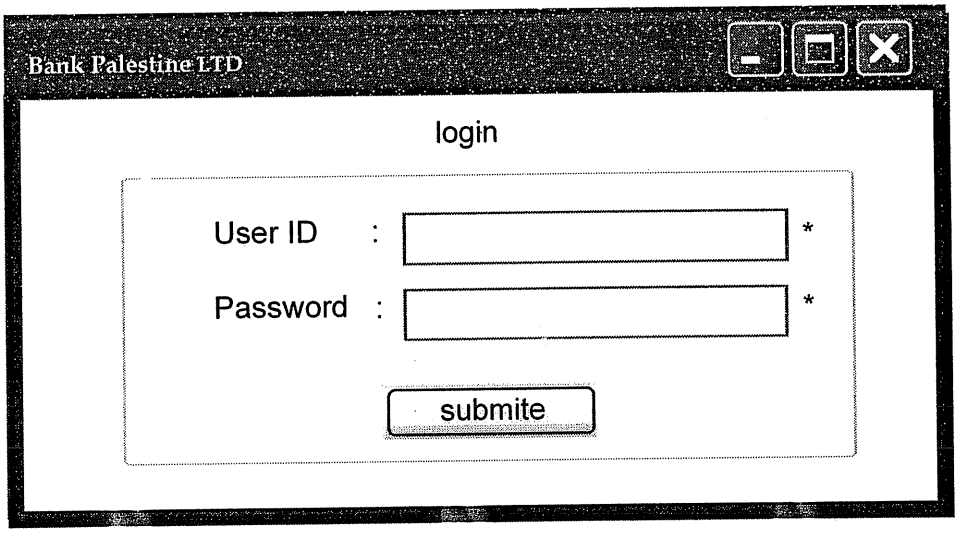


Figure (4.109) bank login screen.

- Enter bank account Form

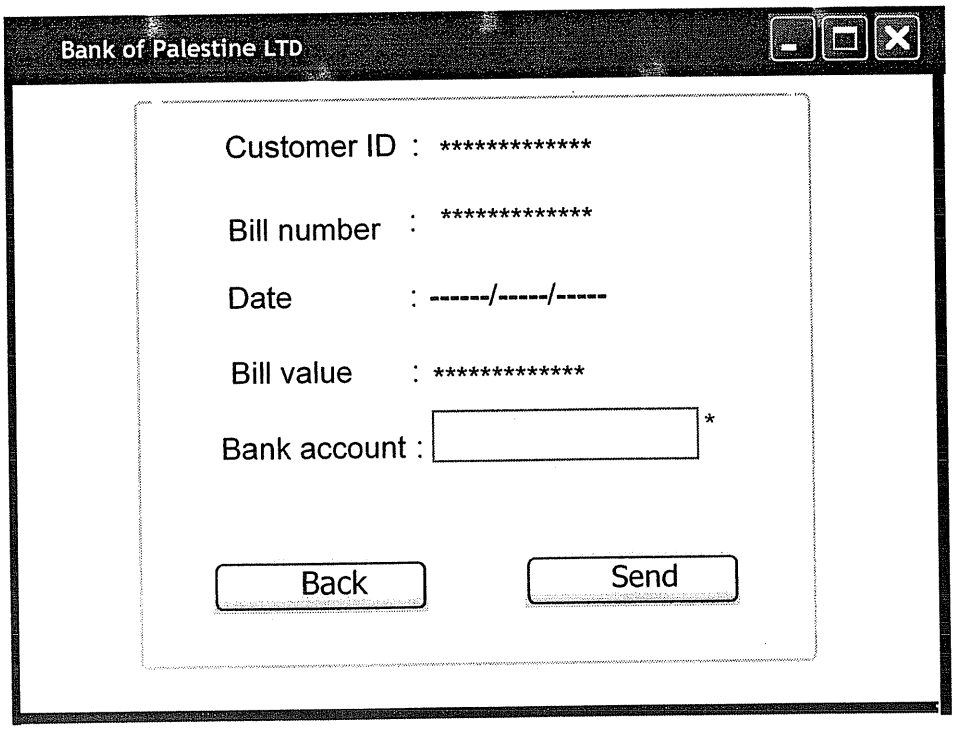


Figure (4.110) enter bank account screen

b) Bank output design:

- Bank confirmation Form

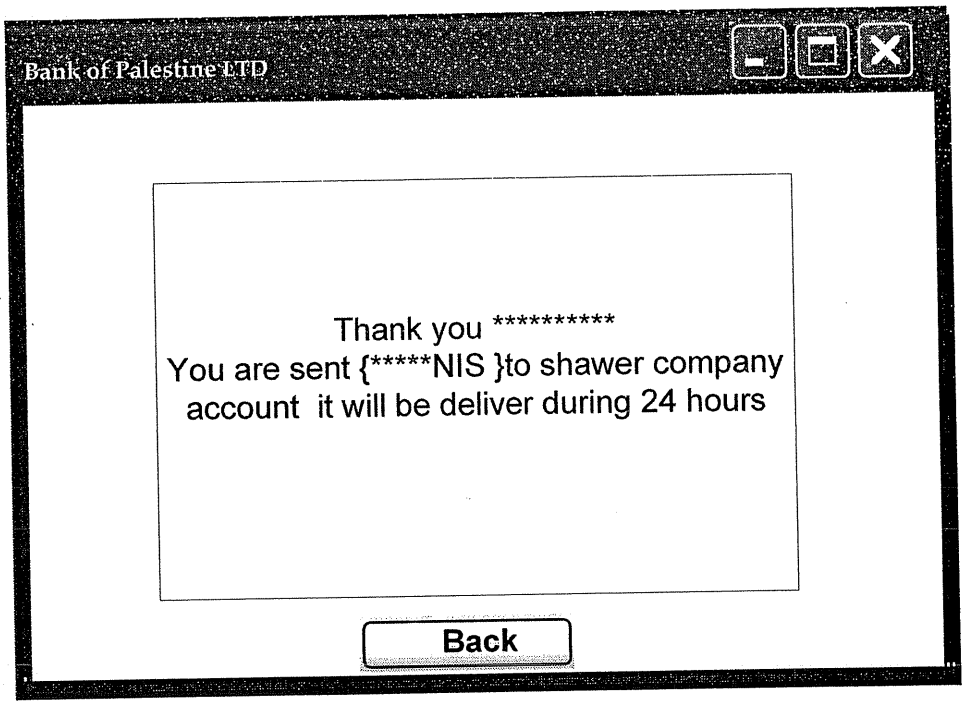


Figure (4.111) Bank confirmation screen

4.4 Data base dictionary

4.4.1 System database Tables

1. Employee.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
employee_ID	Integer	4	PK		Employee ID
Password	Nvarchar	20			Employee password.
Employee_name	Nvarchar	50			Employee Name
Hour_price	Integer	4			price of hour work
Email	Nvarchar	50			Employee Email
tell	Nvarchar	9			employee phone number
City_no	integer	4	FK	City(city_no)	city number
ID	integer	4			Employee identification
Birth_date	date	8			birth day
address	Nvarchar	50			Address
dept_no	integer	4	FK	Department(dept_no)	Department number
Job_no	boleen	1	FK	Job(job_no)	Job description

Table (4.1) employee table

2. Employee salary

Field	*Data Type*	*length*	*Key*	*References*	*Description*
emp_sal_no	Integer	4	PK		Sequential number
employee_ID	Integer	4	FK	Employee(Employee_ID)	Employee ID
Total_paid	Mony	8			total money paid
Payment_date	date	8			Date of payment.

Table (4.2) employee salary table

3. Employee work.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
employee_ID	Integer	4	PK,FK	Employee(Employee_ID)	Employee ID
Current_day	date	4	PK	Employee(Employee_ID)	Date of day.
Start_time	date time	8			Start Time
End_time	date time	8			End Time.
No_hour	flout	8			Total of Hour

Table (4.3) employee table work table

4. Employee comment:

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Comment_no	Integer	4	PK		Sequential number
comment_contant	Nvarchar	50			Comments
employee_ID	Integer	4	FK	Employee(Employee_ID)	Employee ID
employee_e_mail	Nvarchar	50			E_mail

Table (4.4) employee comment table

5. Department

Field	*Data Type*	*length*	*Key*	*References*	*Description*
dept_no	Integer	4	PK		Sequential number
dept_name	Nvarchar	50			dept Name

Table (4.5) department table

6. City

Field	*Data Type*	*length*	*Key*	*References*	*Description*
City_no	Integer	4	PK		Sequential number
City_name	Nvarchar	50			City Name

Table (4.6) city table

7.Guest

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Guest_ID	Integer	4	PK		Guest ID
Guest_name	Nvarchar	20			Guest Name
Company_name	Nvarchar	50			Company Name
City_no	integer	4	FK	City(city_no)	City number
Tell	Nvarchar	9			Guest phone number
Fax	Nvarchar	9			Guest fax number
Email	Nvarchar	50			Email
address	Nvarchar	50			Address

Table (4.7) gust table

8. Customer

Field	*Data Type*	*length*	*Key*	*References*	*Description*
customer_ID	Integer	4	PK		Customer ID
Password	Nvarchar	20			Customer password.
Customer _name	Nvarchar	20			Customer Name
Company_name	Nvarchar	50			Company Name
City_no	integer	4	FK	City(city_no)	City number
Tell	Nvarchar	9			Customer phone number
Fax	Nvarchar	9			Customer fax number
Email	Nvarchar	50			Email
Address	Nvarchar	50			Address

Table (4.8) customer table

9. Customer comment

Field	Data Type	length	Key	References	Description
comment no	Integer	4	PK		Sequential number
comment_contant	Nvarchar	50			Comments
Customer _ID	Integer	4	FK	Customer (Customer _ID)	Customer ID
Customer _e_mail	Nvarchar	50			E_mail

Table (4.9) customer comment

10. Order

Field	Data Type	length	Key	References	Description
Order_no	Integer	4	PK		Sequential number
Customer_ID	Integer	4	FK	Customer (Customer_ID)	Customer ID
order_date	date	8			Date of generate order .
Confirm	Integer	4			Order status

Table (4.10) order table

11. orders detail.

Field	Data Type	length	Key	References	Description
order no	Integer	4	PK,FK	Order (Order_no)	Order number
item no	Integer	4	PK,FK	Item(item_no)	Item number.
quantity	Integer	4			Quantity of item
Orders_cost	Float	9			Total value of order
Description	Nvarchar	50			Description of item
unit sales	Float	9			Item price
status	Boolean	1			Item fill full or no

Table (4.11) orders detail table

12. bill

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Bill_no	Integer	4	PK		Sequential number
Customer_ID	Integer	4	FK	Customer(customer_ID)	Customer number.
Order_no	Integer	4	FK	Order(order_no)	Order number
Totaal_val_bill	Float	9			Total value of bill
Gen_dat	Date	8			Date of bill
status	Boolean	1		Status(status_no)	Paid or not paid

Table (4.12) bill table

13. status

Field	*Data Type*	*length*	*Key*	*References*	*Description*
status_no	Boolean	1	PK		0 or 1
Status_name	Nvarchar	50			Paid or not paid

Table (4.13) status table

14. Manufacture Item.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Manf_item_no	Integer	4	PK		Sequential number
item no	Integer	4	FK	Item(item_no)	Item number.
Quantity	Integer	4			Quantity of item produced
Material_no	Integer	4	FK	Material(material_no)	Material number

Table (4.14) manufacture item table

15. Item.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Item_no	Integer	4	PK		Sequential number
Item_name	Nvarchar	20			Item Name
Category_no	Integer	4	FK	Category(category_no)	Category number
Total_quantity	Integer	4			Total of Item
Item_quantit_mate	Float	9			Amount of consume material
Unit_cost	Float	8			Cost of item
Unit sails	Float	8			Price of item
Description	Nvarchar	50			Description

Table (4.15) item table

16. material

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Material_no	Integer	4	PK		Sequential number
Material_name	Nvarchar	20			Material Name
Category_no	Integer	4	FK	Category(category_no)	Category number
Total_Material	Integer	4			Total of material

Table (4.16) material table

17. Add Material.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Number	Integer	4	PK		Sequential number
Material_no	Integer	4	FK	Material(material_no)	Material number
Total_roll	Integer	4			Total of roll entered
Length	Float	8			Length of roll
Width	Float	8			Width of roll
Description	Nvarchar	50			Description of roll
Date	Date	8			Date

Table (4.17) add material

18. category.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Category_no	Integer	4	PK		Sequential number
Category_name	Nvarchar	50			Category Name

Table (4.18) category table

19. Advertisement.

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Adv_no	Integer	4	PK		Advertisement number
Title	Nvarchar	20			Title of Advertisement
Description	Nvarchar	50			Description
Viewing_date	Date	8			Date of viewing advertisement
Expiration_date	Date	8			Date of expiration

Table (4.19) advertisement table

4.4.2 bank's database Tables

1. Bank Login

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Customer bank ID	Integer	4	PK		Bank login
Password	Nvarchar	50			password

Table (4.20) bank login table

2. Bank accounts

Field	*Data Type*	*length*	*Key*	*References*	*Description*
Bank_account_no	Nvarchar	50	PK		Bank account number
Name	Nvarchar	50			password
Customer bank ID	Integer	4	FK	Bank Login (Customer bank ID)	Bank login
Account_val	Float	8			Amount of money in customer's account

Table (4.21) bank account table

3. Bank transaction

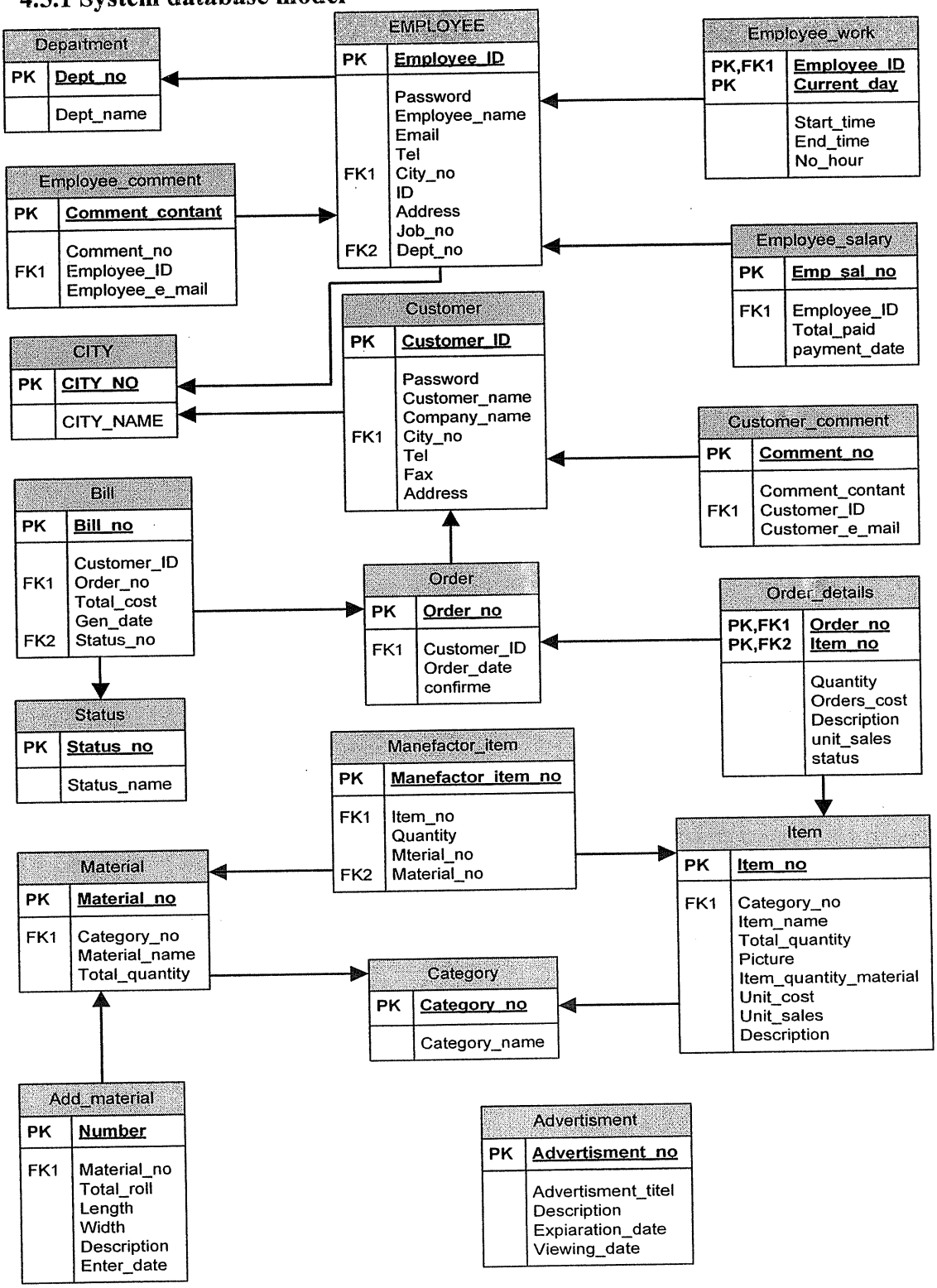
Field	*Data Type*	*length*	*Key*	*References*	*Description*
Ttans_no	Integer	4	PK		Sequential no
Customer_ID	Integer	4			Customer ID in shawer company
Bill no	Integer	4			Bill number
Pay_Date	Date	8			Date
Money_paid	money	8			Amount of monry was paid(bill value)
Bank_account_no	Nvarchar	50	FK	Bank account(Bank_ account_no)	Bank account number

Table (4.22) bank transaction

4.5 Database Model

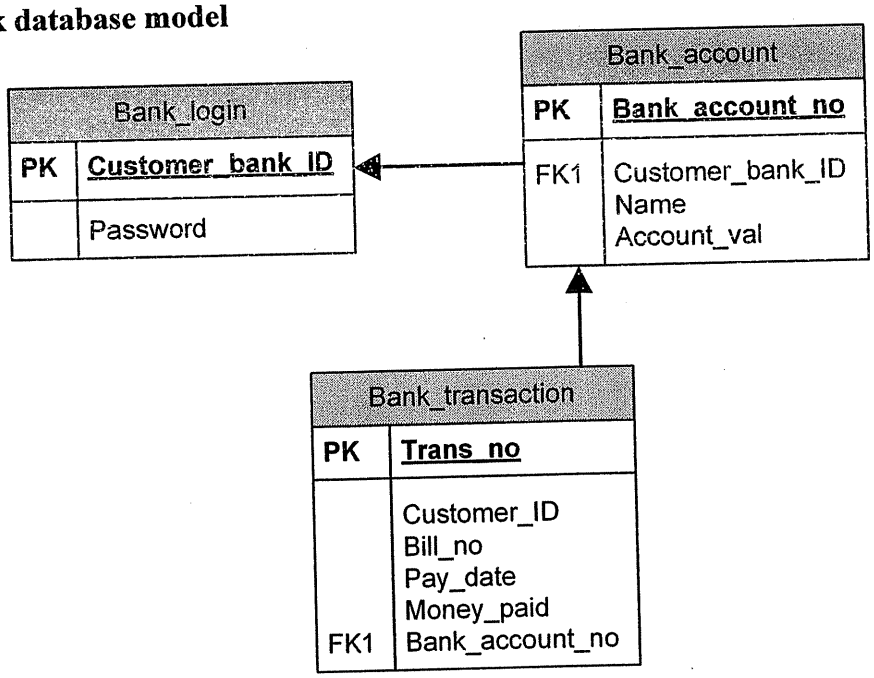
We have implemented a relational database model for our system and bank database design and implementation, by using UML notations to describe the database model.

4.5.1 System database model



Figure(4.112) system database model

4.5.2 Bank database model



Figure(4.113) bank database model

4.6 Test Plan:

Here we describe briefly the methodology that we have adapted to test the system, steps that will be followed in the system testing are described bellow:

4.6.1 Testing steps:

1- Unit and Module testing:

We will use the white and black test Box testing to ensure that each function or module will operate as expected, by inserting valid and invalid inputs to show how the system will handle it.

2- Sub-system testing:

In this stage we will test each sub-system individually to ensure that each sub-system is operates as expected and meet its requirements.

3- Integration testing:

The integration of all sub-systems will be tested so that to ensure that the subsystems work together properly as expected, and meets its requirements.

4- System testing:

The system with all subsystems and functions will be tested to ensure that it functions properly, it meets its specification, and show if there defects throw system running.

4.7 Programming Language and Coding

There are many languages that can be used to develop a system such ours, but the most two effective languages are JAVA language and ASP.NET using Microsoft Visual Studio .NET as the development tool, here we describe the advantages of each, and why our selection was on the ASP.NET (we describe the ASP.NET technology in details in chapter 1):

***1- Device Independent*:**

Both Java and ASP.Net are a device independent languages, this means that any user can open the page that written using theses two languages from any explorer without any additional components or drivers, because he will just receive an HTML code.

2- Security

Java and ASP.Net have a high level of security during transmitting data; they provide us with many algorithms and techniques.

In the ASP.NET there are a build in valuators that ensure the user's input before any generation on the server, so if there any unusual input the webpage it self will not return to the server.

In Java the programmer must do all algorithms and validations manually.

3- Server side

One of the most powerful advantages of the ASP.NET technology is that it do not need to make any efforts on the client side, all operations and functions will work on the server.

4.8 Summery and Recommendations

According to the design covered in this chapter the work team will continue to the following stage of development which is coding and implementation

Chapter five

5

Coding and implementation

Chapter five

Introduction

Establishment of development environment

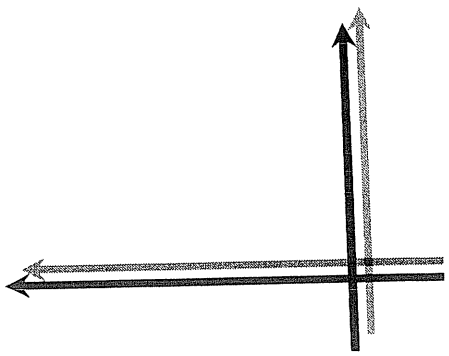
Database implementation

Supporting software

Operating the system

coding

development process implementation



5.1 Introduction

This system is an internet application that depends on a number of technologies that need to be installed maintained and updated continuously. There exist a large number of software development packages that belong to different companies such as Microsoft.

This system is built basically on a group of Microsoft technologies such as the MS SQL server 2003 and ASP.NET 2003. We also included some other applications that serve for the user interface design such as Macromedia Flash and InfoPath.

This chapter describes the packages of software and hardware used in the system for the operation phase and how they are installed and prepared for work, the database and web interfacing, and the user interface implementation.

5.2 Establishment of development environment

5.2.1 Hardware Environment

For the system development we use the following hardware:

1. A Server characterized computer.
2. One printer.
3. Scanner.
4. Networking equipment such as cabling, network adapters, and hubs
5. An internet connection (Leased line).

5.2.2 Software Environment

For the system development environment we use the following software:

1. Microsoft Windows XP Professional edition with IIS web server and server extensions.

   We use this platform or operating system because our system is built on Microsoft visual studio.NET technology, and this type of operating system is the best platform for this technology we can use.

2. Microsoft visual studio.NET 2003 with ADO.NET.

   From this package we use ASP.NET technology which is a new powerful technology designed to create web applications, making complete error handling, and provide data access tools.

- ***Why Visual Studio.NET?***

Visual Studio.NET simplifies the development of powerful, reliable enterprise Web solutions and increases developer efficiency by providing a familiar, shared development environment. Pre-build components, programming wizards and the ability to reuse components that are written in any language can reduce development time significantly. It provides Integrated Development Environment (IDE) that provides a consistent look and feel, regardless of programming language being used. IDE provides the tools that are used in programming as shown in figure5.1.

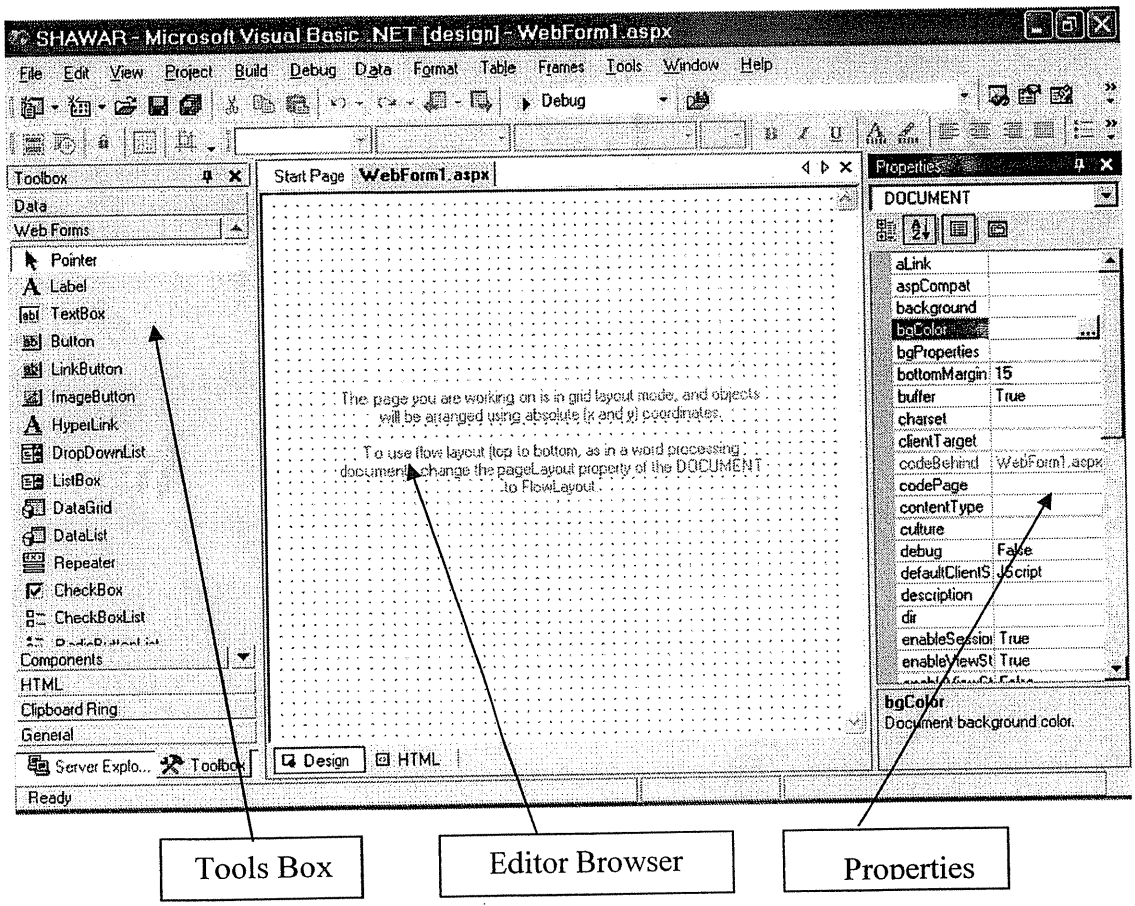


Figure (5.1) Integrated Development Environment in visual studio.NET

- ***What is ASP.NET?***

Developing ASP.NET web applications in the .NET framework are similar to developing windows applications. The fundamental component of ASP.NET is the web form. The web form is the web page that users view in browsers. An ASP.NET web application comprises of one or more web forms. A web form is dynamic page that can access server resources.

For example, a traditional web page can run script on the client to perform basic tasks. An ASP.NET web form conversely, can also run server side code to access a database, to generate additional Web Forms or to take advantage of built-in security in the server.

In addition, because an ASP.NET Web Form does not rely on the client-side scripting, it is not dependent on the client browser type or operating system. This independence allows you to develop a single Web Form that can be viewed on practically any device that has Internet access and a web browser

Because ASP.NET is part of the .NET Framework, you can develop ASP.NET Web application in any .NET-based language

The ASP.NET technology also supports XML service. XML web services are distributed applications that use XML for transferring information between client's application and another XML web service.

- ***Why XML?***

XML is the universal format that is used for describing and exchanging structured documents and data on the Internet. XML is defined by the World Wide Web Consortium (W3C), thereby ensuring that the structured data will be uniform and independent of web applications.

XML define the structure of data in an open and self-describing manner. The open and self describing manner allows data to be easily transferred over a network and to be consistently processed by the receiver. XML document describes how the data is structured, not how it should be displayed or used, it contains tags that assign meaning to the content of the document. The stages allows programmers to find the data that they need in the XML document

3. Microsoft SQL server 2000.

SQL Server 2000 is a powerful tool for turning information into opportunity. Industry-leading support for XML, enhanced tools for system management and tuning, and exceptional scalability and reliability make SQL Server 2000 the best choice for the agile enterprise.

It has the following features:

- High Availability: Maximize the availability of your business applications with log shipping, online backups, and failover clusters.
- Scalability: Scale your applications up to 32 CPUs and 64 gigabytes (GB) of RAM. SQL Server 2000 has demonstrated record-breaking performance that you can leverage
- Security: Ensure your applications are secure in any networked environment, with role-based security and file and network encryption.
- Simplified Database Administration: Automatic tuning and maintenance features enable administrators to focus on other critical tasks.
- Data Transformation Services: Automate routines that extract, transform, and load data from heterogeneous sources.
- Rich XML Support: Simplify the integration of your back-end systems and data transfer across firewalls using XML.

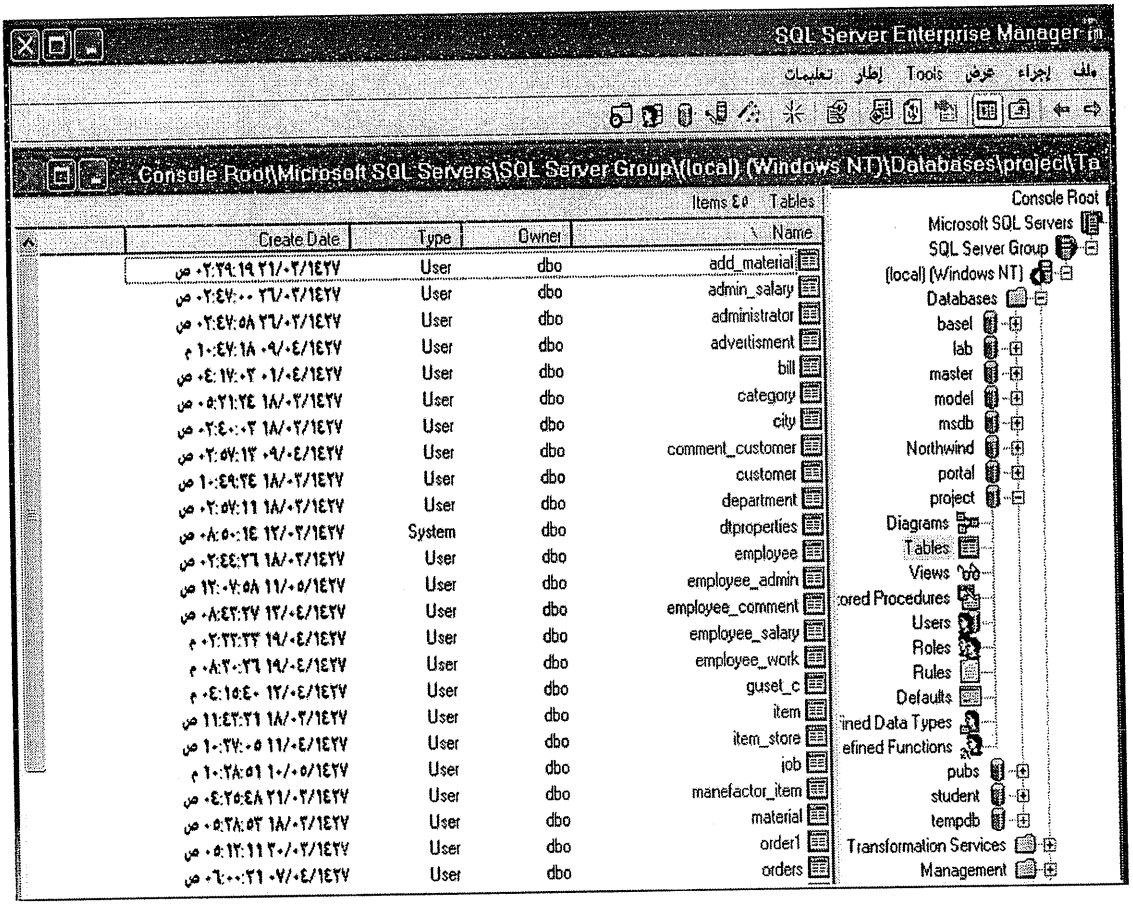


Figure (5.2) Enterprise Manager Window.

✓ **Configuration of SQL server 2000:**

When using .NET technology ADO.NET is used for database connections, ADO.net which is a new technology that is based on the usefulness of Microsoft ActiveX Data Object (ADO), however. It is a new technology for manipulating data, it contains numerous improvements over the previous version of ADO, and it is greatly simplifies the process of connecting a web application to a database.

Unlike ADO, ADO .NET is specifically designed for data connections located in a disconnected environment, so it the best choice when developing and implementing internet based applications.

By configuring the SQL server 2000 to be mixed mode authentication, which is the preferred method to use when connecting a web application to SQL server 2000 DBMS, this method need user name and password to be transferred back and forth between servers. Figure [5.3] shows the configuration of SQL server 2000 authentication mode

to be "SQL server and windows authentication".

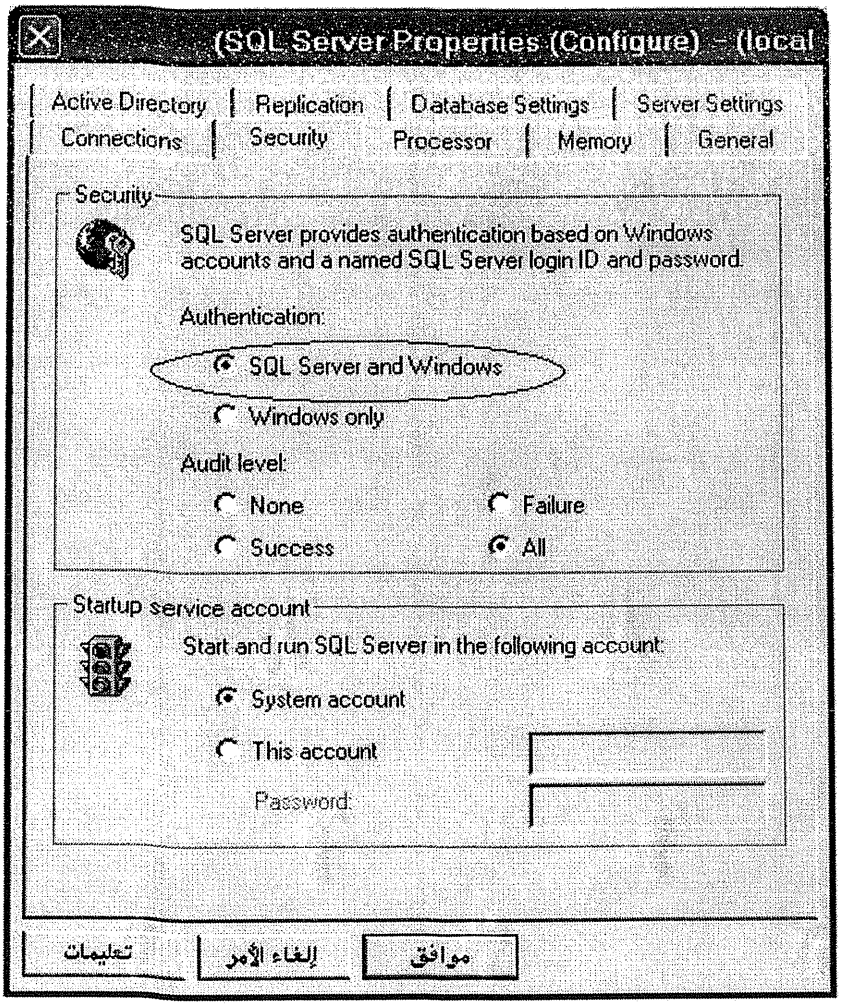


Figure (5.3)Mixed Mode Authentication (Step One).

The last configuration of the SQL server 2000 is to work effectively and ensuring that the integration more comfortable is to add a new account in the login group of the SQL server. This account (sa) is created by the SQL 2000 during installation process and it should be added to the login group of the SQL server. Figure [5.4] shows how to add this account to the logins group of the SQL Server.

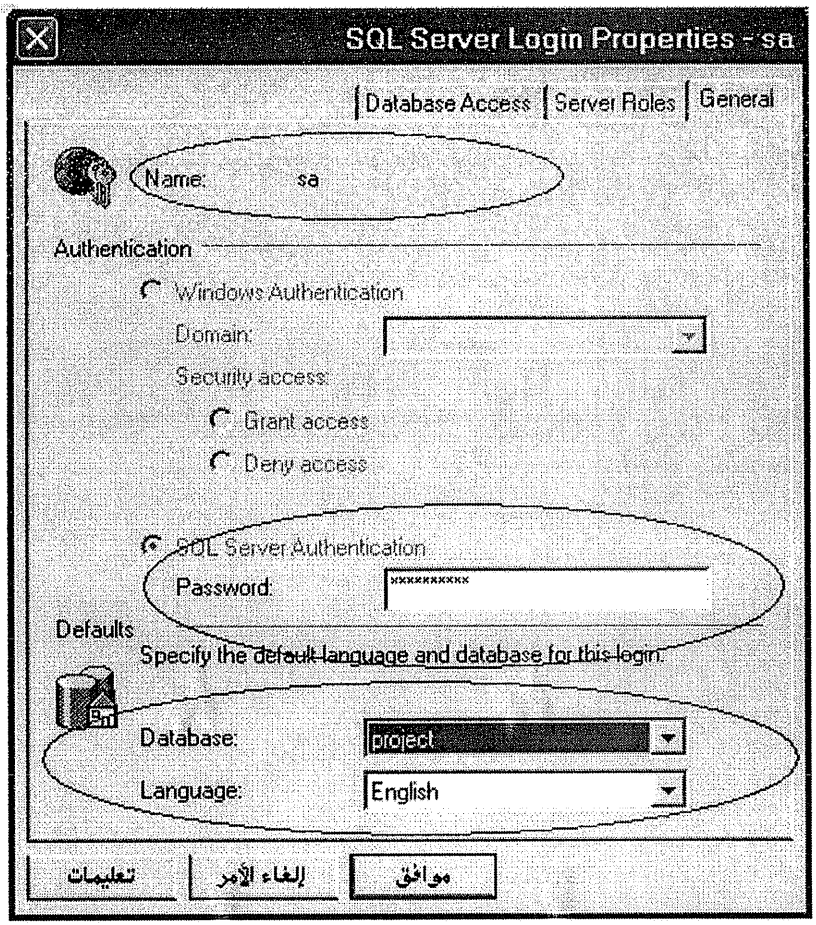


Figure (5.4) Mixed Mode Authentication (Step Two).

5.3 Database implementation

The system database is implemented using the SQL server 2000 enterprise manager with the following properties:

A. SQL server 2000 database name: portal.
B. Database normalization, it normalized to insure that all tables and relations are not contained redundancy data and other database problems , so that it make it correct.
C. Database creation : all keys which include, primary and foreign are created to ensure database consistency and correct relations.

D. Database connection

At this stage, the path to the data stored in the database could be opened and used as a two channel path.

In fact, to configure a connection, it is better to do it programmatically i.e. by coding, but using wizards that create connections is preferred at the starting point of the database connection establishment process.

The core here is whether a connection secure or not, but in configuring the SQL server 2000 we have chosen the windows only authentication, now in this stage we have to integrate the connection security with the SQL server 2000 security mode, thereby making the connection secure, this could done by wizard, as it appears in figure (5.5) shown bellow, we select the integrated security option to be used when creating a connection to the database using Visual Studio.NET

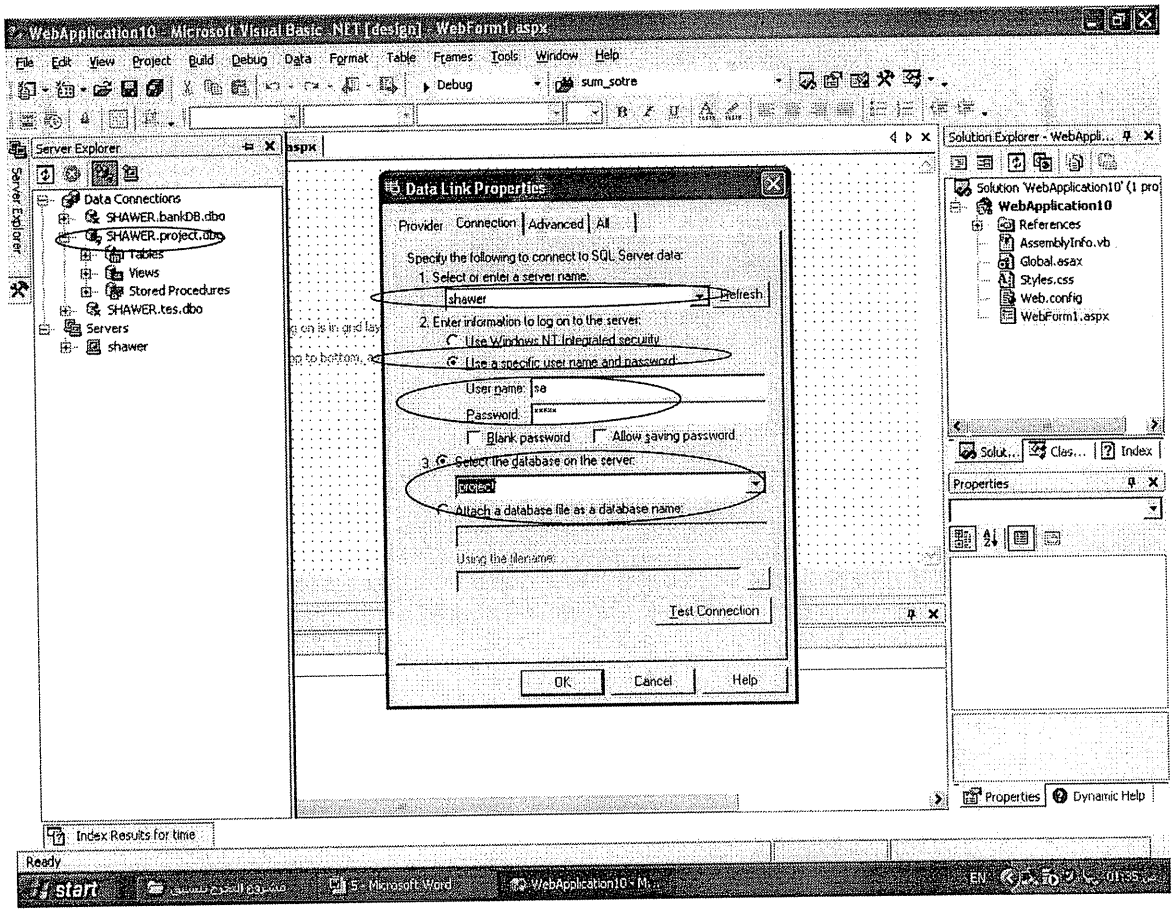


Figure (5.5) Creating Integrated widows security data link

After connect with database we can add, delete, update, and select data from database.

5.4 Supporting software:

Many other software tools where required to improve our system, these are used to support the appearance of GUI of our web application. So we used macromedia Flash 6.0, Microsoft front page XP, a dope Photoshop, and Photo Impact 8.0.

5.5 Operating the system:

To operate the system properly. Many steps should be executed before the system operates as expected in its environment:

- Configuring the network (physical connection).
- Assigning appropriate IP addresses to the computer in the network.
- Setting up the .NET framework.
- Building the system Database.
- Creating the Database connection with security options.
- Setting up the system on the server (publishing through IIS).

5.6 *Coding:*

In our case a web-based application we used visual Studio.NET, which simplifies the development of powerful, reliable software application by providing familiar and shared development environment. It contain pre-built component, programming wizard, as well the ability to use component built using various language. In Visual Studio.Net there is a single integrated development environment (IDE), which provides a sense of what you see is what you get (the visual programming environment).

The usage of this tool for the purpose of programming and coding reduce the time and efforts and thereby increasing the performance.

When using Visual Studio.Net as a programming environment we gain the benefits of the separation between writing the logical code (the program functionality) from one side and the design of the appearance and graphical user interface (GUI) from the other side.

5.7 Development process implementation.

The system was first created by going to Microsoft visual studio.NET development environment and named as G_project..

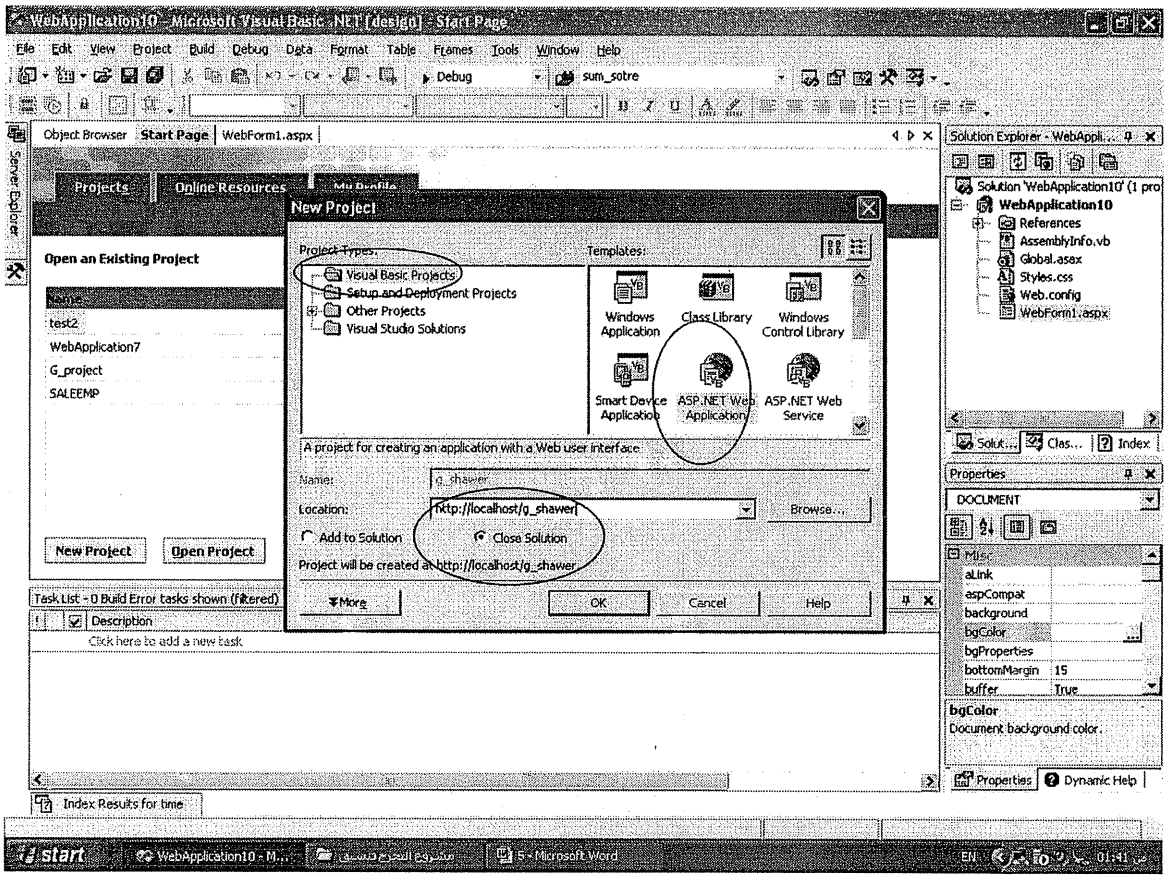


Figure (5.6) Development Process (create new project)

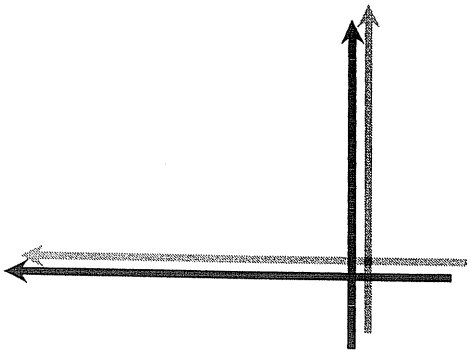
Chapter six

6

Testing

Chapter six

- Introduction
- Unit and module
- Integration testing
- System testing
- Acceptance testing
- Sample snapshots



6.1 Introduction

Testing the system to ensure that it meets its specifications is one of the most important stages in the software system development.

For the purpose of delivering a system that works properly as expected, certain testing procedures should be performed on system and its components; accordingly with an acceptance testing that may be stated as a result for the success of the testing process.

This chapter covers the testing for:

- System units and module testing.
- Subsystems testing
- Integration testing.
- System testing.
- Acceptance testing.

Testing will take place in a time space that was assigned for the testing process. Table (6.1) shows the testing schedule:

Time in week / Testing Process	1st Week	2nd Week
Unit and module testing		
Subsystem testing		
Integration testing		
Acceptance testing.		

Table (6.1) Testing Schedule

6.2 Unit and module testing

We have tested the units and modules by using the path testing on each function in the system. In this section we describe some of these testing procedures on a number of selected functions that are classified as units and modules, these testing procedures are described here according with a certain snapshots that were captured from the real operating system interface.

1- **Tested Function:** " User Login":

Method: path testing.

Test cases: each test case covers the set of input values in a certain execution path as shown in the function flowchart figure [6.1].

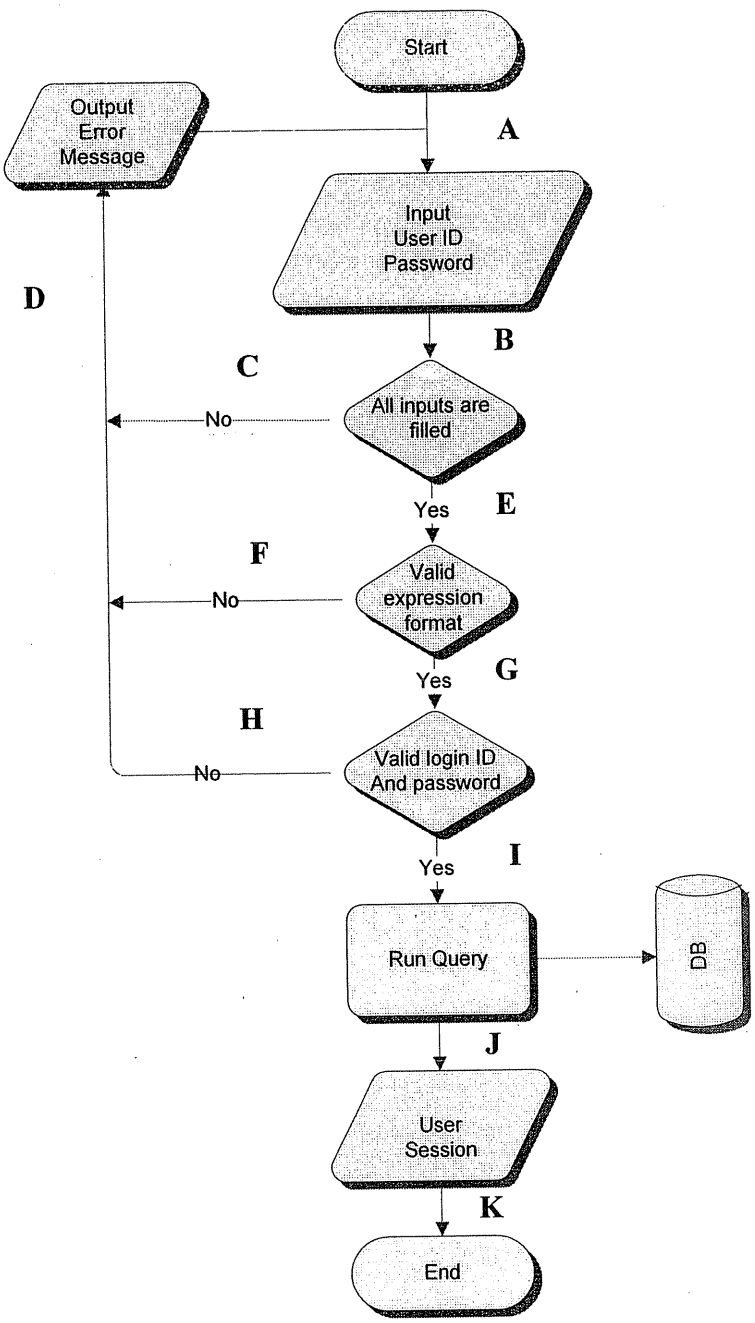


Figure [6.1] Login customer Execution Paths.

Test Cases Path	Test Data		Expected Output	Actual Output
	Login ID	Password		
A-B-E-G-I-J-K	5000	1234567	Valid login ID and Password (Accepted User).	Valid login ID and Password (Accepted User).
A-B-C-D	———	———	User ID and Password is not filled in.	User ID and Password is not filled in.
A-B-E-F-D	shawer	12\232	Invalid login ID format.	Invalid login ID format.
A-B-E-G-H-D	10163829	134567890	Invalid login ID or Password.	Invalid login ID or Password.

Table [6.2] Login User Test Cases.

Conclusion: function confirms to its specifications.

2- **Tested Function:** " Change password":

Method: path testing.

Test cases: each test case covers the set of input values in a certain execution path as shown in the function flowchart figure [6.2].

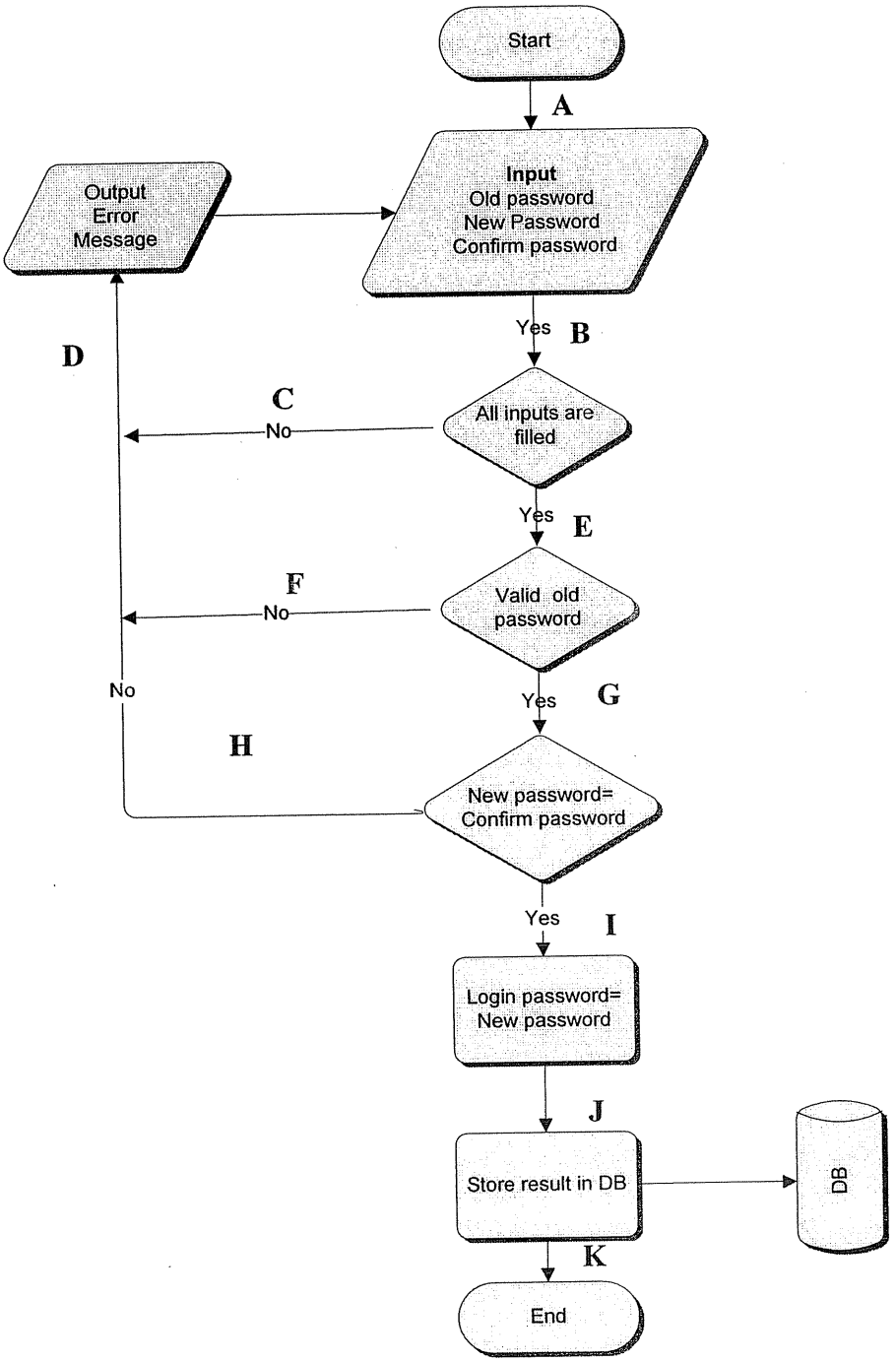


Figure [6.2] Change password Execution Paths.

Test Cases Path	Test Data				Expected Output	Actual Output
	user ID	Old password	New Password	Confirm		
A-B-E-G-I-J-K	Session (username)	1234567	1357234	1357234	Password updated successfully	Password updated successfully
A-B-C-D	Session (username)	1234567	1-123s\1	1-123s\1	New Password is not filled in.	New Password is filled in.
A-B-E-F-D	Session (username)	1232451	7654321	7654321	Invalid old password.	Invalid old password.
A-B-E-G-H-D	Session (username)	1234567	3214567	4567321	Invalid confirmed Password.	Invalid confirmed Password.

Table [6.3] Change Password Test Cases.

Conclusion: function confirms to its specifications.

3- **Tested Function:** "add new item":

Method: path testing.

Test cases: each test case covers the set of input values in a certain execution path as shown in the function flowchart figure [6.2].

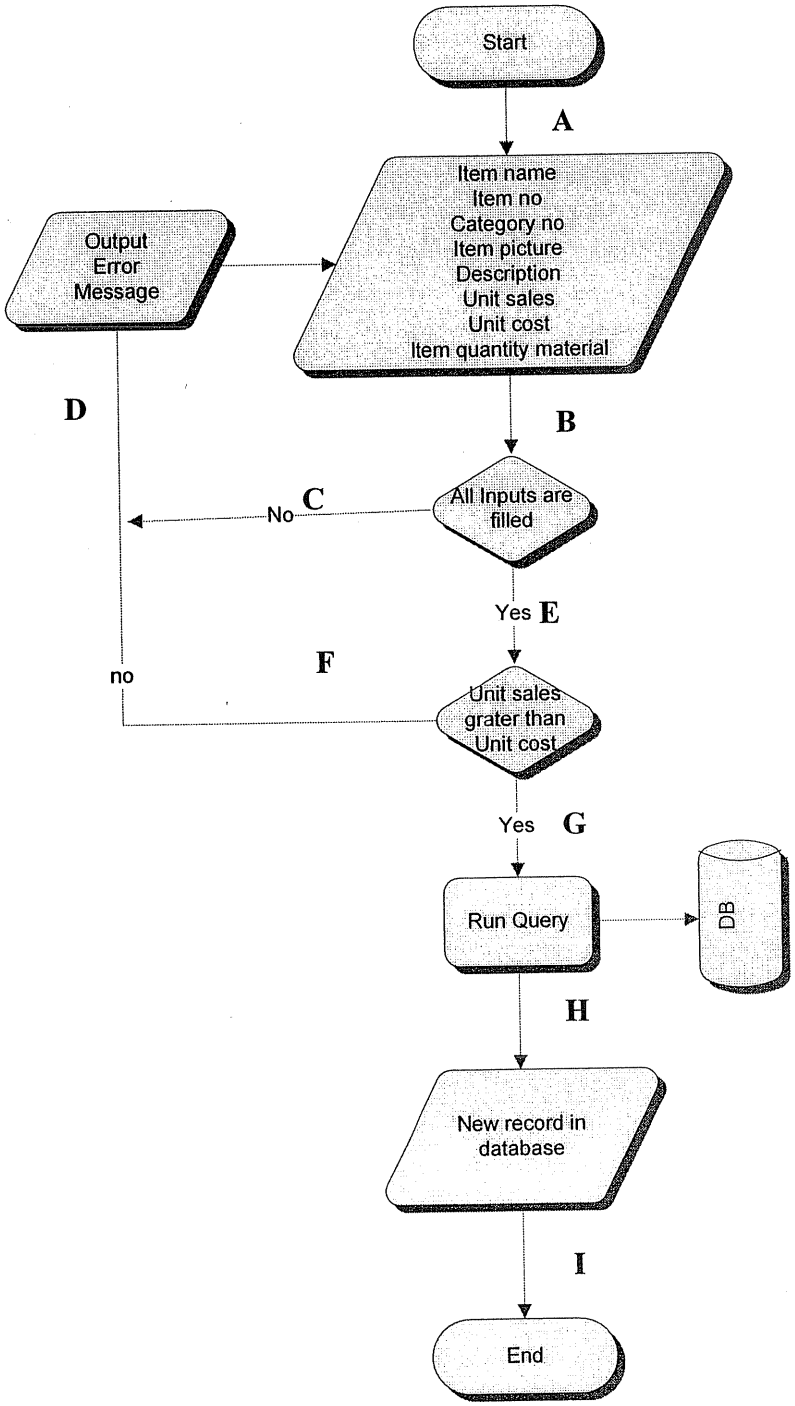


Figure [6.3] Add new item Execution Paths.

Test Cases Path	Test Data					Expected Output	Actual Output
	Item no	Item picture	Unit sales	Unit cost	Item Quantity material		
A-B-E-G-H-I	بجامة	بجامة.jpg	80	40	1.6	Add item successful	Add item successful
A-B-C-D	بلبيل	بجامة.jpg	77	71	111	Item quantity material is not formatted.	Item quantity material is not formatted.
A-B-E-F-D	روب	روب.jpg	20	100	1.2	Unit sales must greater than unit cost	Unit sales must greater than unit cost

Tables [6.4] add new Item Test Cases.

Conclusion: function confirms to its specifications.

6.3 Integration testing

After the integration of all modules was tested to ensure that the whole system performs as expected, we have tested the integration using scenario based testing approach was used so that the system integrated based on the use scenarios. Testing here demonstrates on the interfaces between all modules

Test results: the testing of the system integration indicated that the system performs as expected.

6.4 System Testing

The system was tested under several conditions, some errors were detected, and upon these results, we have solved these problems and we imposed the system another time to testing techniques to ensure that it disposed all types of defects and problems.

6.5 Acceptance Testing

The system was tested against its requirements, we conclude that it achieves its functional requirements, and could operate soon in the real environment.

6.6 Sample Snapshots

We have selected some program snapshots to be displayed here to show how the real program behaves when working under certain situations

- **Order table after change**

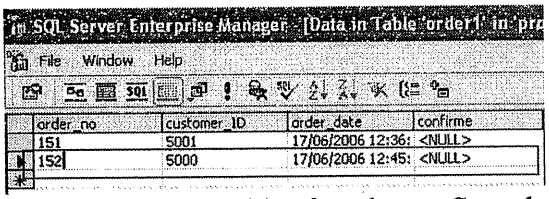


Figure [6.6] Order table after change Snapshot

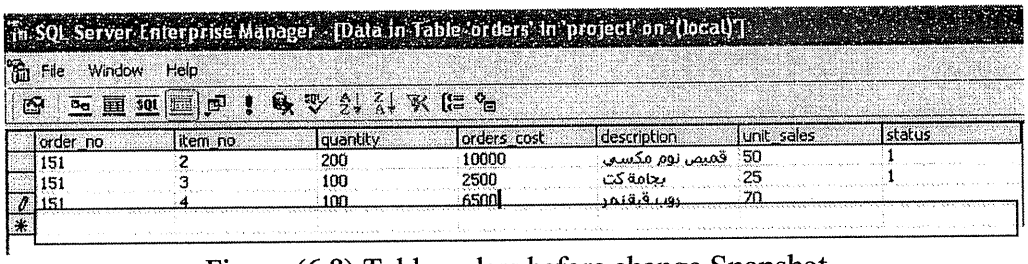
6.1.2 Add select item into order snapshot

To enable customer to add all item he needed to his order was generated must input quantity in textbox "الكمية" after browsing item by using button " التالي" ,"السابق", then must he click on button "تخزين" , customer can be adding more than one item to his order

- **Selecting and adding item page**

Figure [6.7] Selecting and adding item page Snapshot

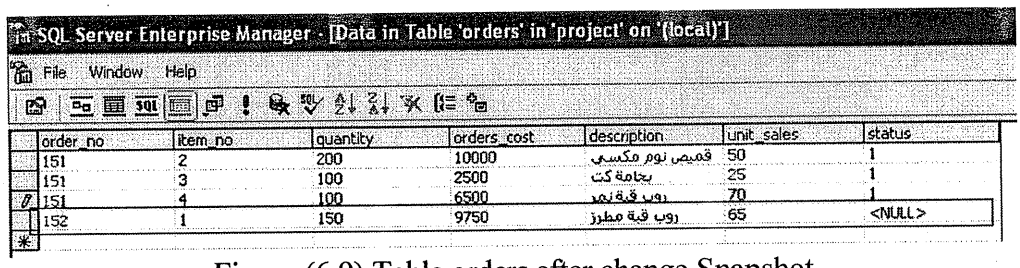
- **Table orders before change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قرقنمر	70	

Figure (6.8) Table orders before change Snapshot

- **Table orders after change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قرقنمر	70	1
152	1	150	9750	روب قرقة مطرز	65	<NULL>

Figure (6.9) Table orders after change Snapshot

6.1.3 Displaying all items was selecting (adding) and sends order snapshot

In this page the customer can change quantity of item he selected by click on button "تعديل" then must enter new quantity he needed, finally click on button "حفظ"
When the customer wants to send order must click on button"إرسال"

- **Display all item page**

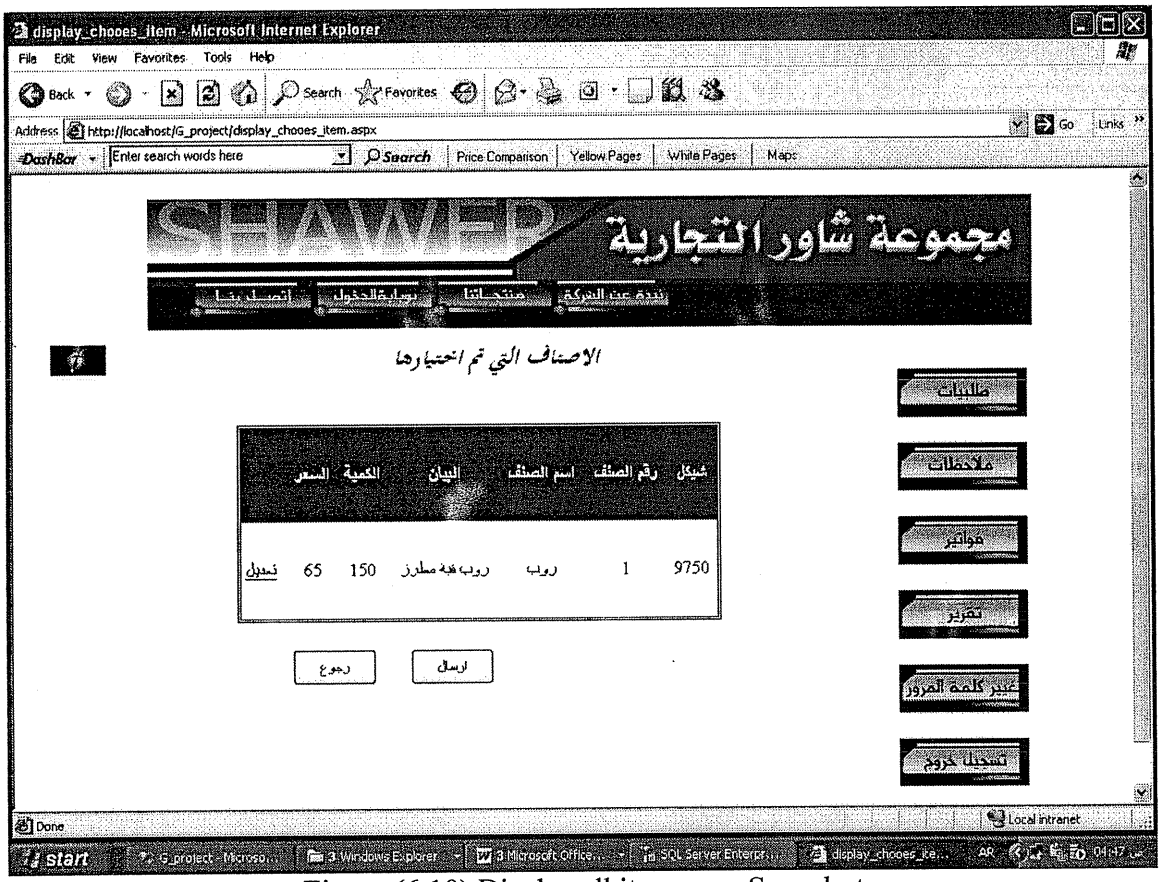
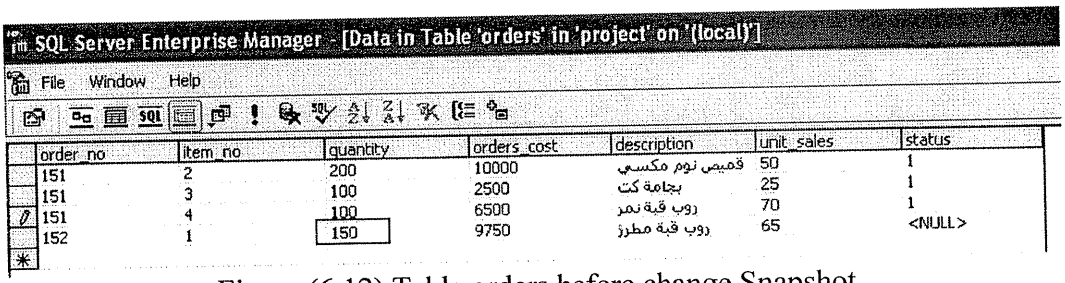


Figure (6.10) Display all item page Snapshot

- **Update quantity of item page**

Figure (6.11) Update quantity of item page Snapshot

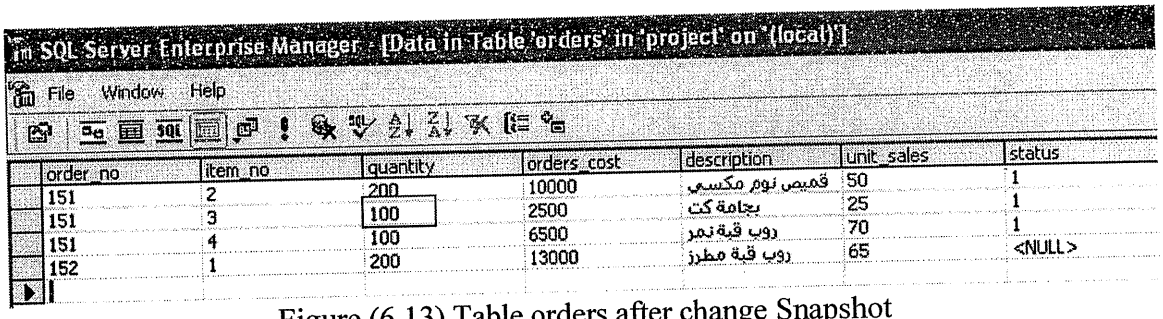
- **Orders table before change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قبة نمر	70	1
152	1	150	9750	روب قبة مطرز	65	<NULL>

Figure (6.12) Table orders before change Snapshot

- **Orders table after change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسيي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قبة نمر	70	1
152	1	200	13000	روب قبة مطرز	65	<NULL>

Figure (6.13) Table orders after change Snapshot

6.1.4 Send order process snapshot

When customer click on button " إرسال"the massage must appear

- **Send order page**

Figure (6.14) send order page Snapshot

- **Order table before change**

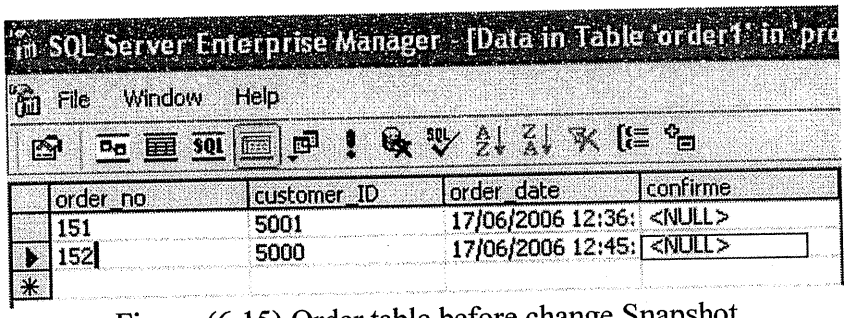


Figure (6.15) Order table before change Snapshot

- **Order table after change**

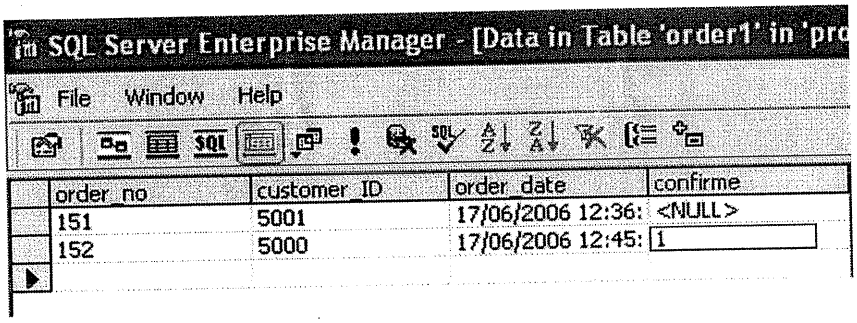


Figure (6.16) Order table before change Snapshot

6.1.5 Administrator (marketing) viewing all customers orders to acceptance it's snapshot

The administrator can viewing order details by click on button "تفاصيل"

- **Customers order page**

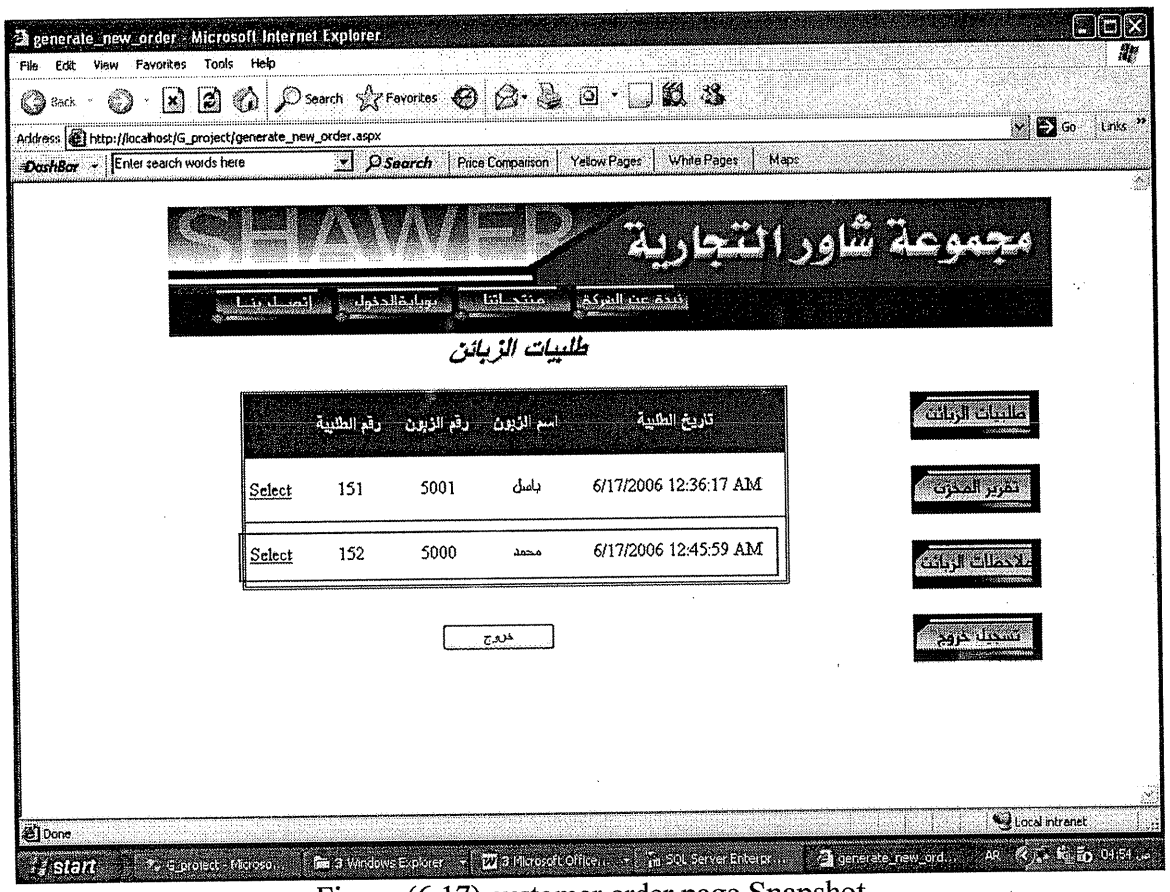


Figure (6.17) customer order page Snapshot

- **Order details and acceptance page**

In this page the administrator can accept order by click on button "موافقة" or accept it later by click on button "رجوع"

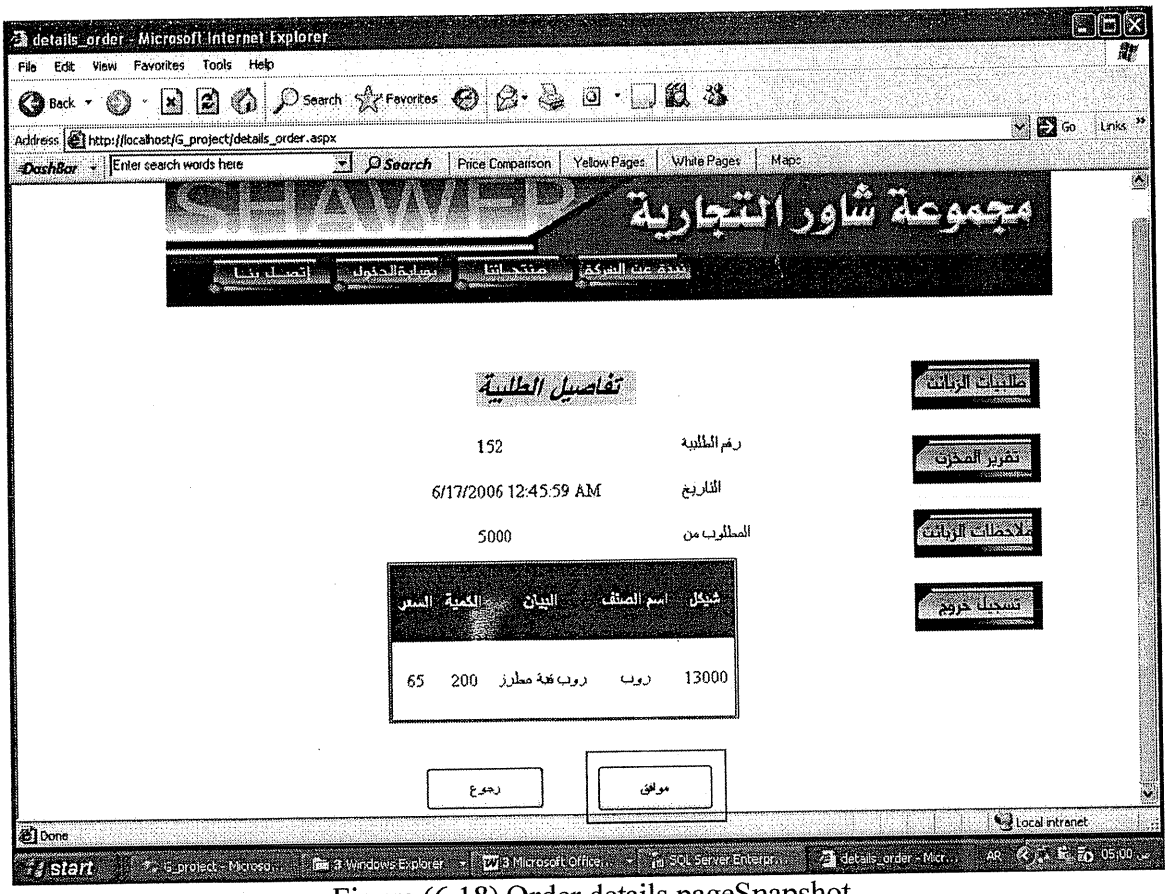


Figure (6.18) Order details pageSnapshot

- **Order table before change**

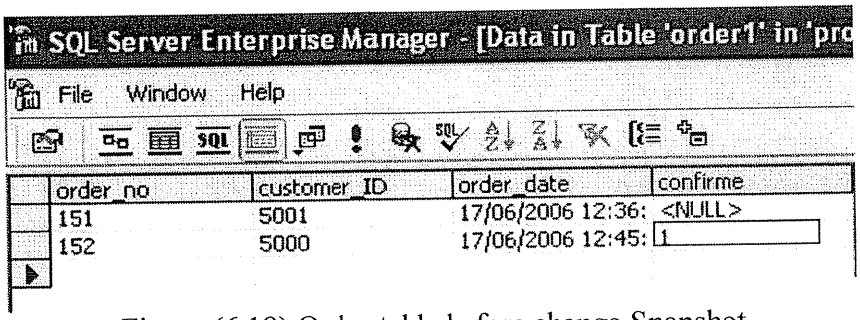


Figure (6.19) Order table before change Snapshot

- **Order table after change**

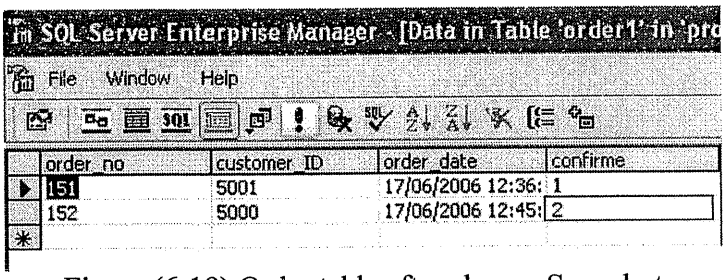


Figure (6.19) Order table after change Snapshot

6.1.6 Discount quantity item process from store snapshot

- **Discount quantity item page**

This page enables the administrator (store) to discounting quantity of its item in specific order by click on button "تجهيز" for all items

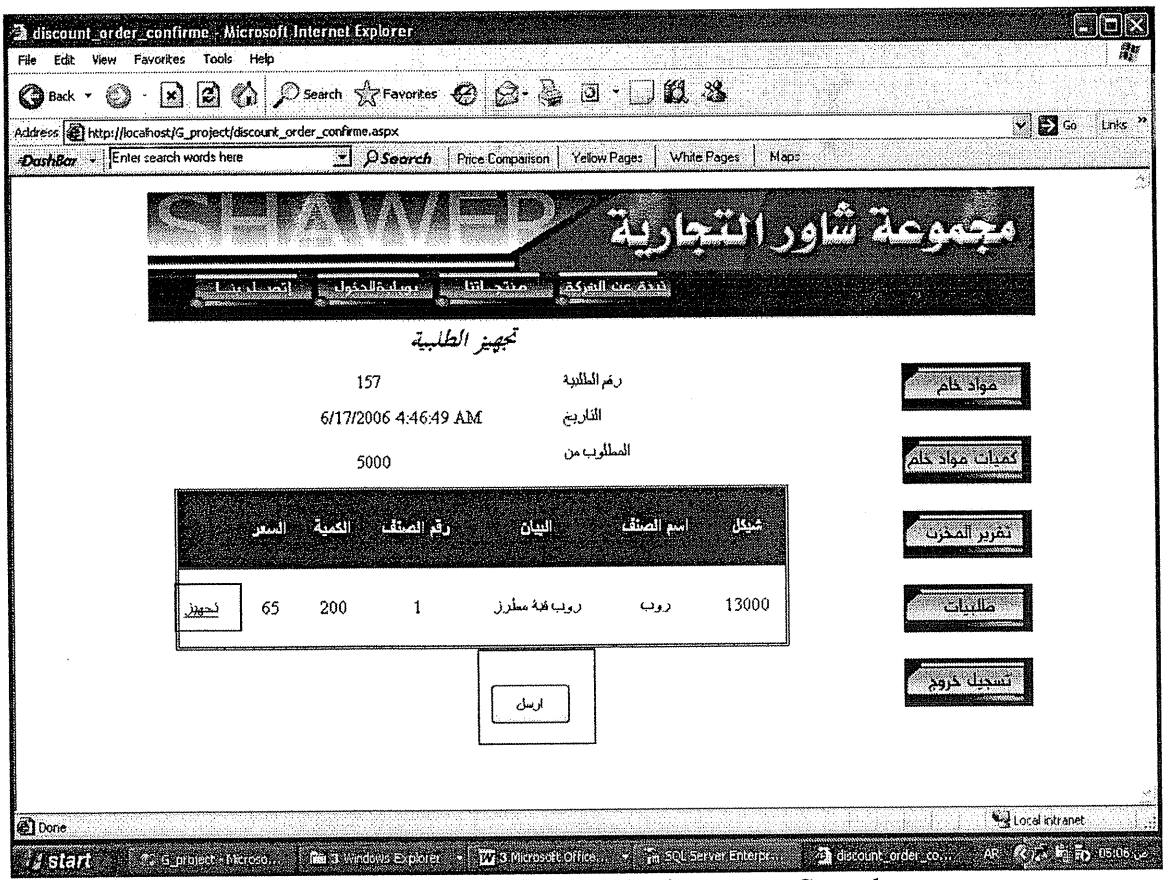
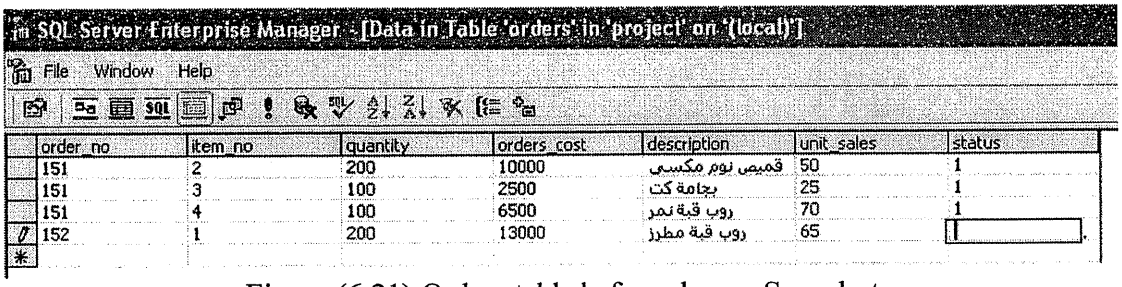


Figure (6.20) Discount quantity item page Snapshot

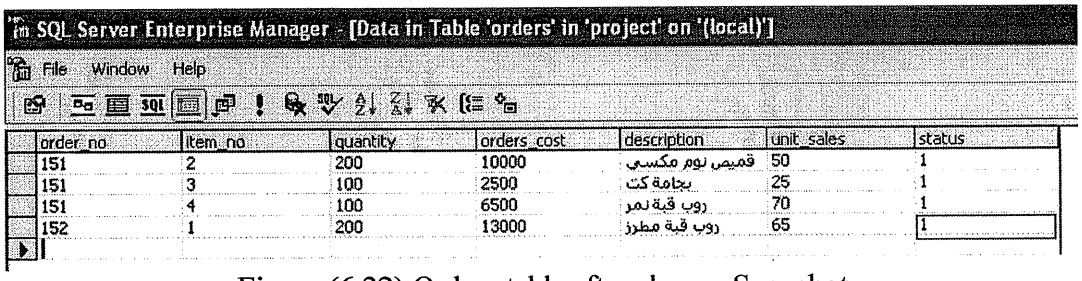
- **Orders table before change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قبة نمر	70	1
152	1	200	13000	روب قبة مطرز	65	

Figure (6.21) Orders table before change Snapshot

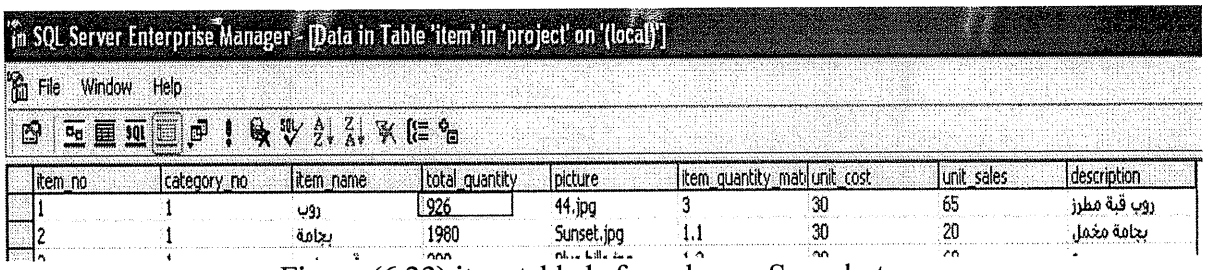
- **Orders table after change**



order_no	item_no	quantity	orders_cost	description	unit_sales	status
151	2	200	10000	قميص نوم مكسي	50	1
151	3	100	2500	بجامة كت	25	1
151	4	100	6500	روب قبة نمر	70	1
152	1	200	13000	روب قبة مطرز	65	1

Figure (6.22) Orders table after change Snapshot

- **Item table before change**



item_no	category_no	item_name	total_quantity	picture	item_quantity_mat	unit_cost	unit_sales	description
1	1	روب	926	44.jpg	3	30	65	روب قبة مطرز
2	1	بجامة	1980	Sunset.jpg	1.1	30	20	بجامة مخمل

Figure (6.23) item table before change Snapshot

- **Item table after change**

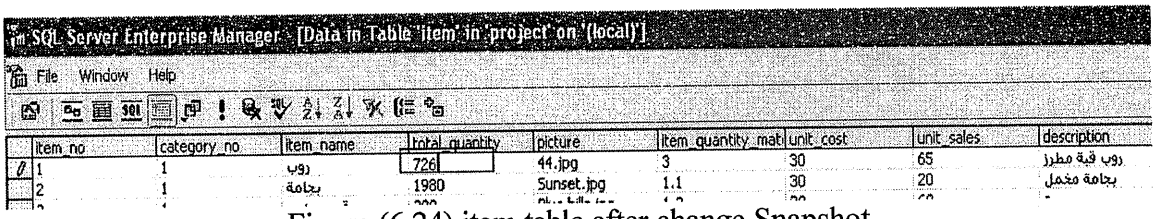


Figure (6.24) item table after change Snapshot

6.1.7 Generate bill process snapshot

- **Generate bill page**

This page enables administrator to generate order by click on "إنشاء"

Figure (6.25) generate bill page Snapshot

- **Bill page**

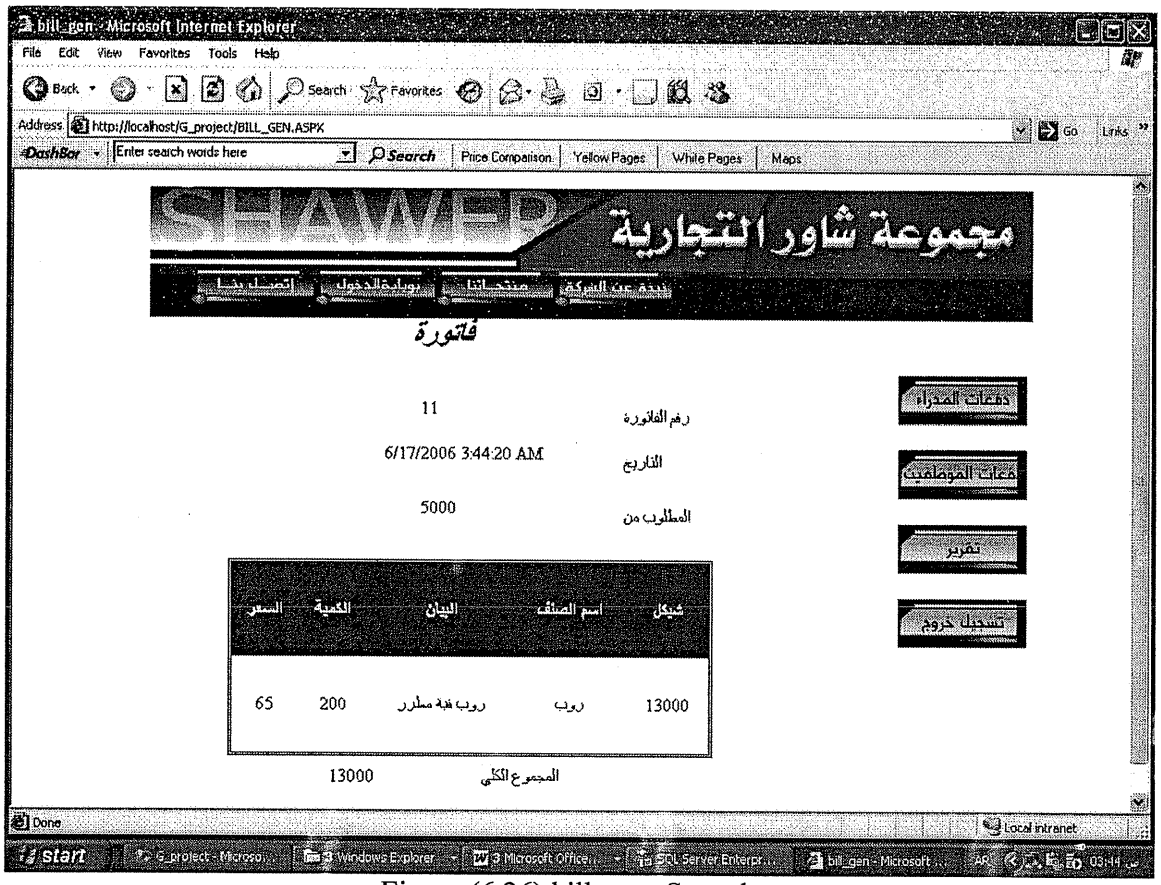


Figure (6.26) bill page Snapshot

- **Bill table before change**

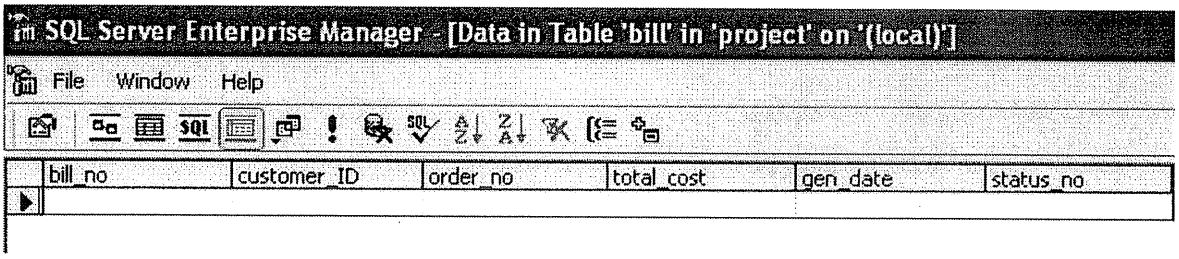


Figure (6.27) bill table before change Snapshot

- **Bill table after change**

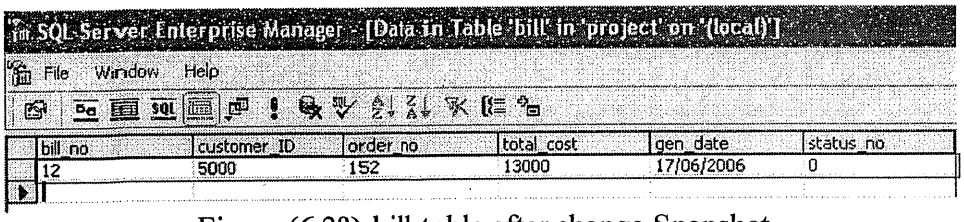


Figure (6.28) bill table after change Snapshot

6.1.8 Billing process snapshot

- **Billing page**

This page enable customer to viewing bill details

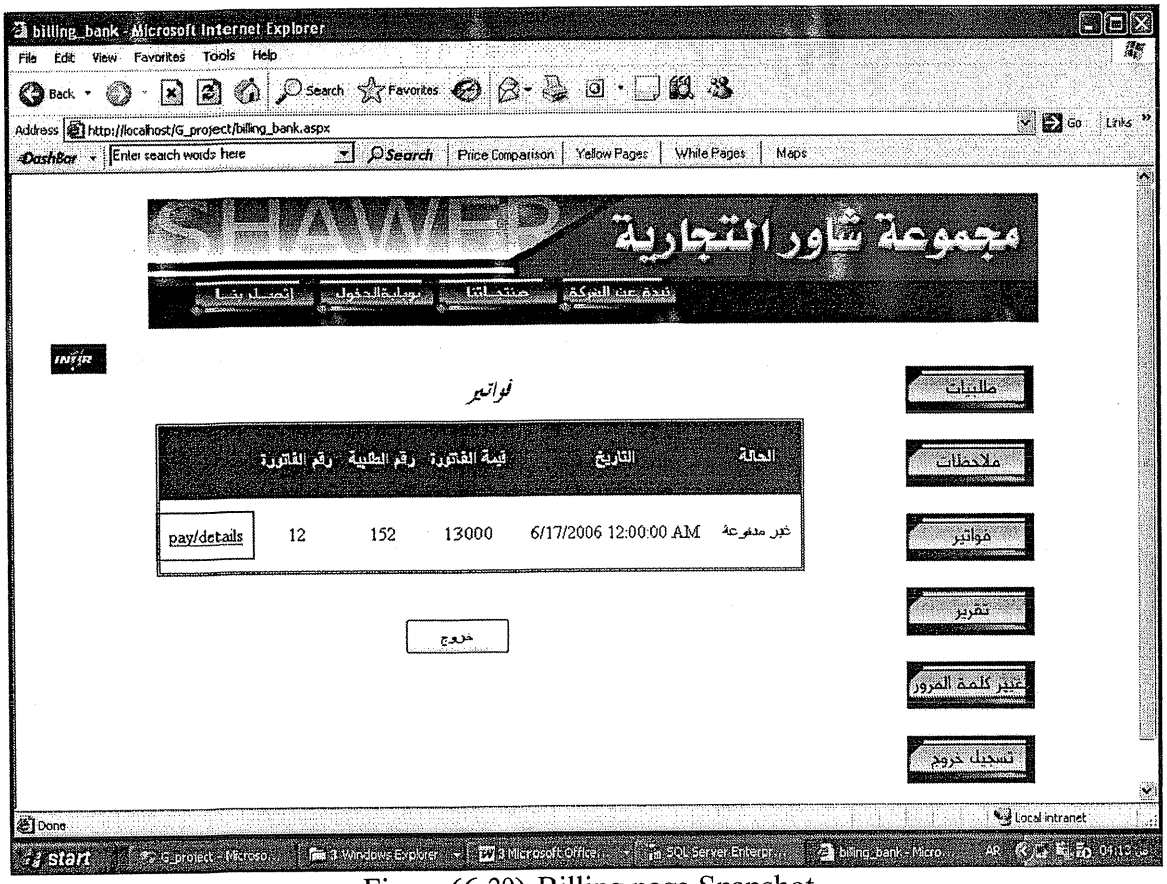


Figure (6.29) Billing page Snapshot

- **Bill details page**

When customer click on button "دفع" the system translate him to bank login page

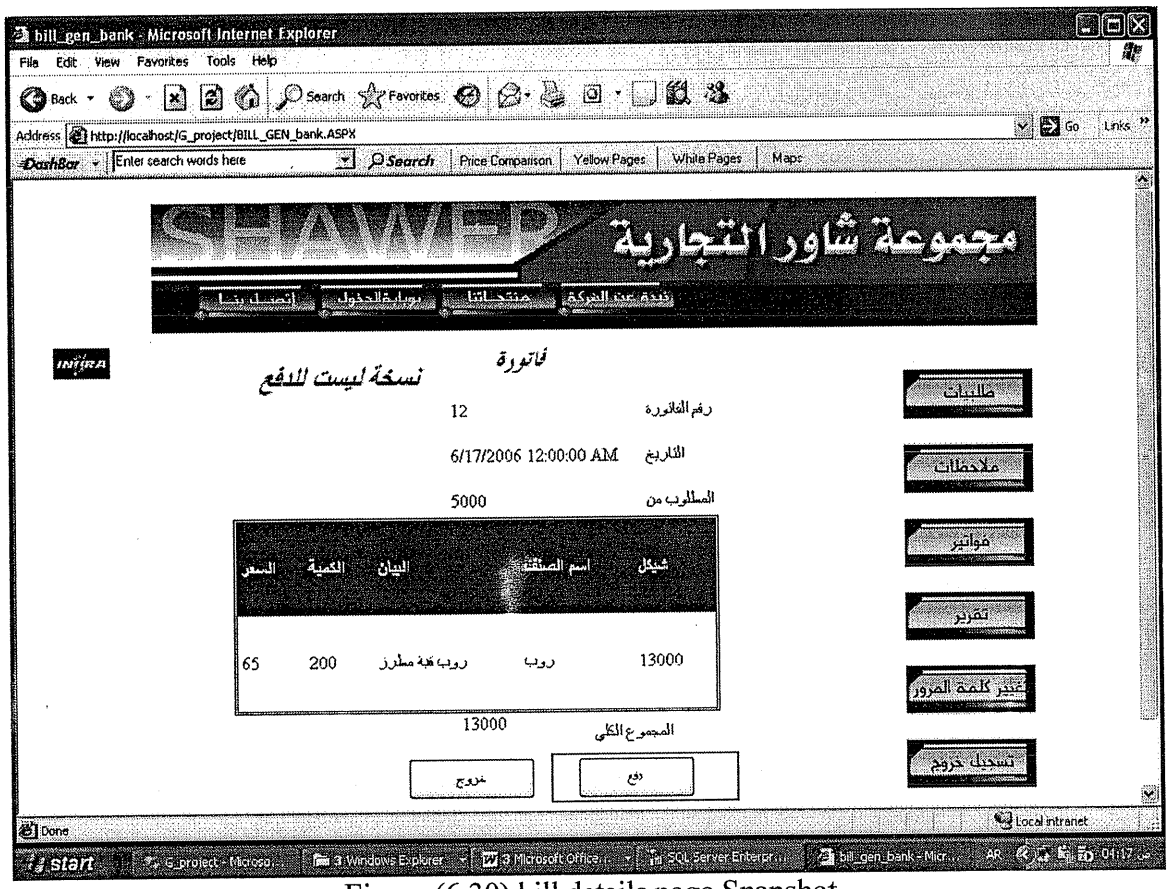


Figure (6.30) bill details page Snapshot

6.1.9 on line payment process using bank web service

- Bank Login page

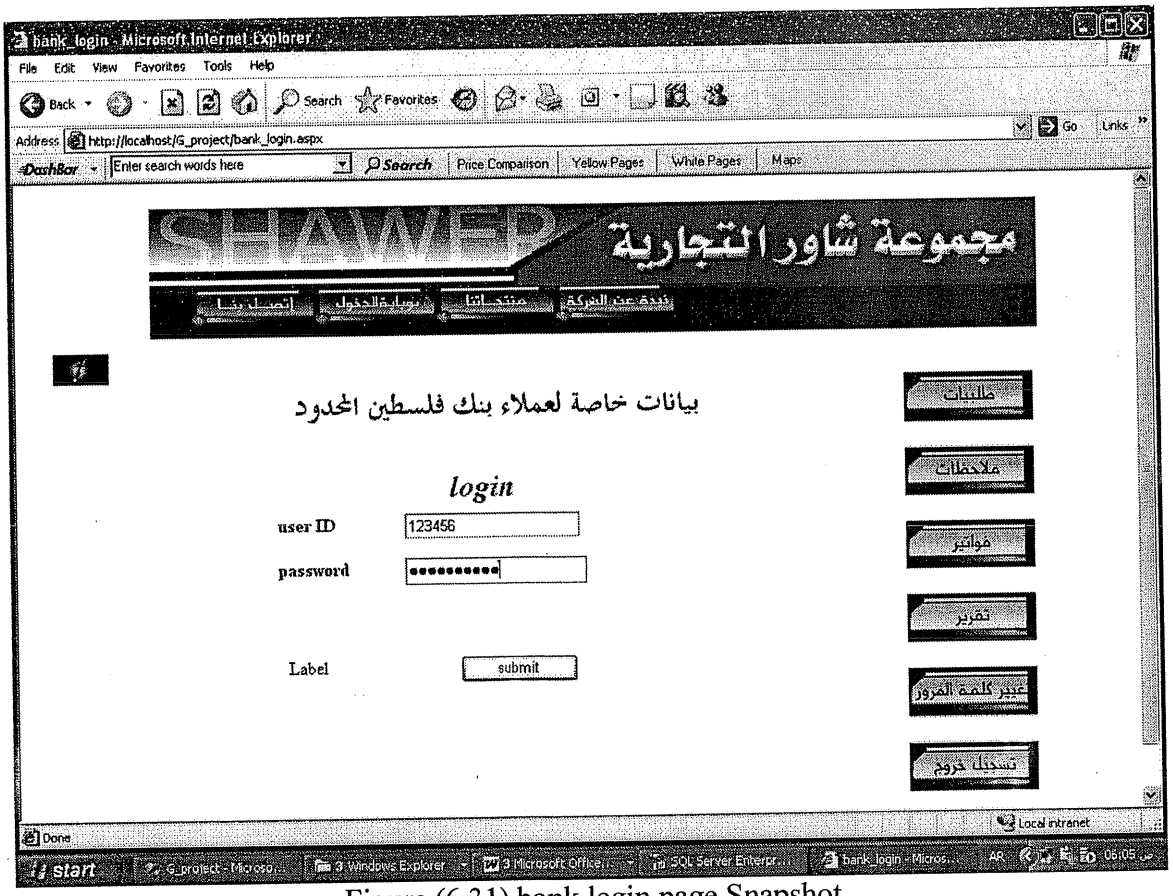


Figure (6.31) bank login page Snapshot

- **Payment page**

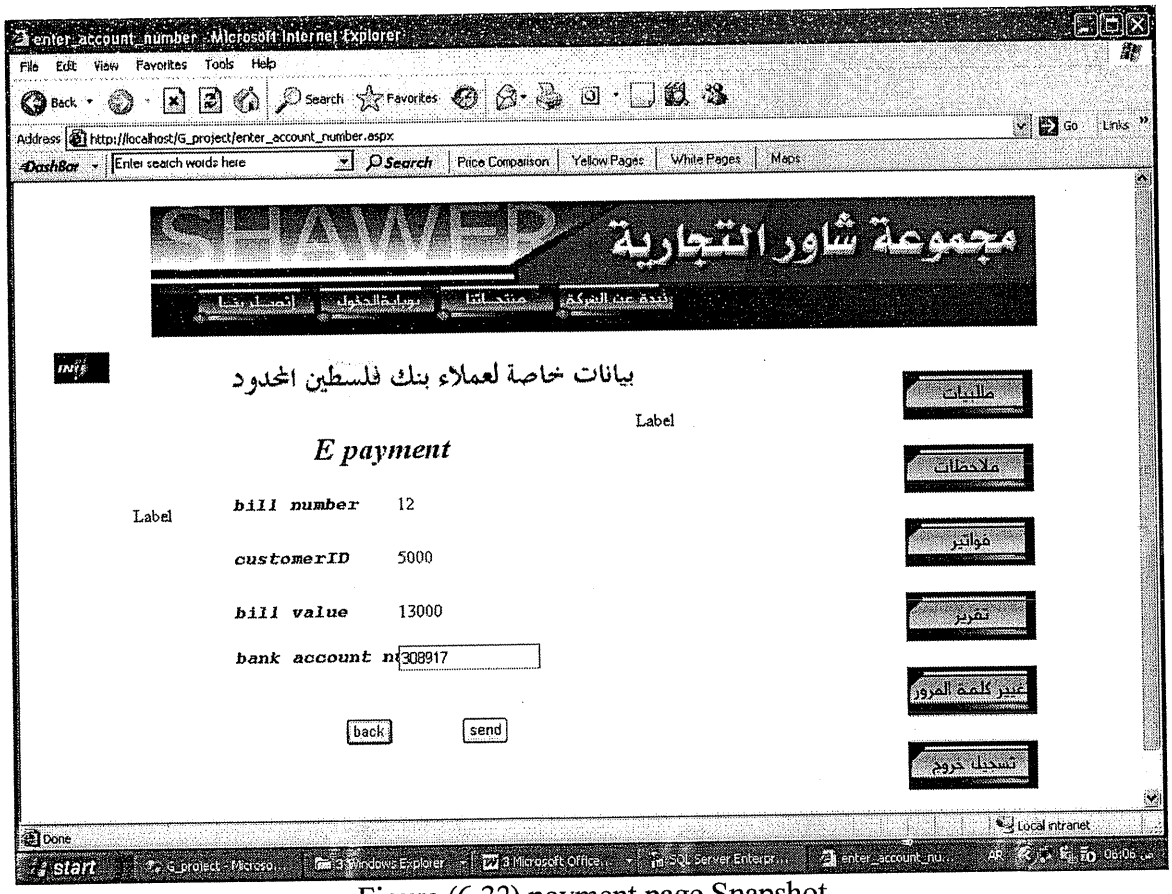


Figure (6.32) payment page Snapshot

- **Confirmation page**

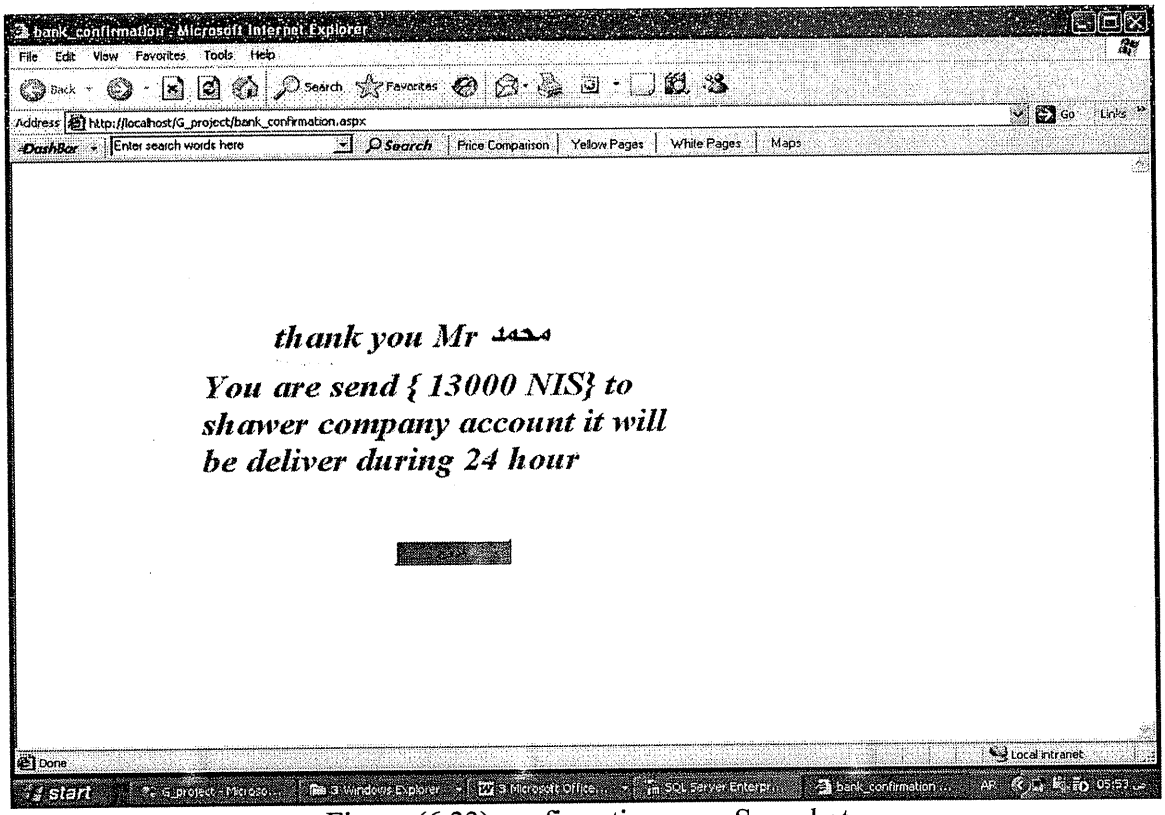


Figure (6.33) confirmation page Snapshot

- **Bill table before change**

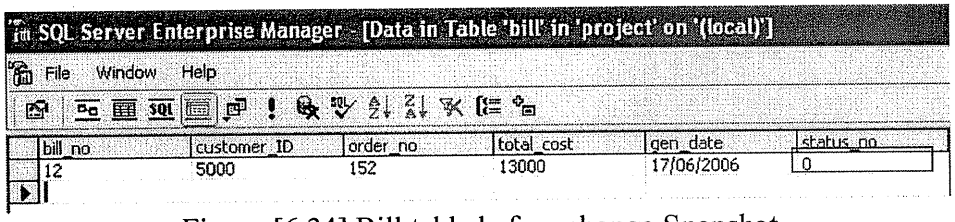


Figure [6.34] Bill table before change Snapshot

- **Bill table after change**

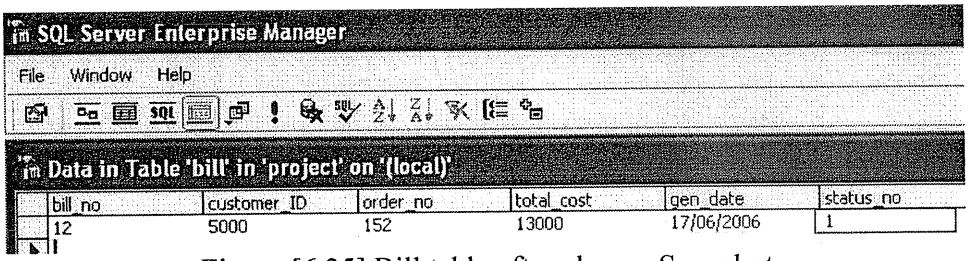


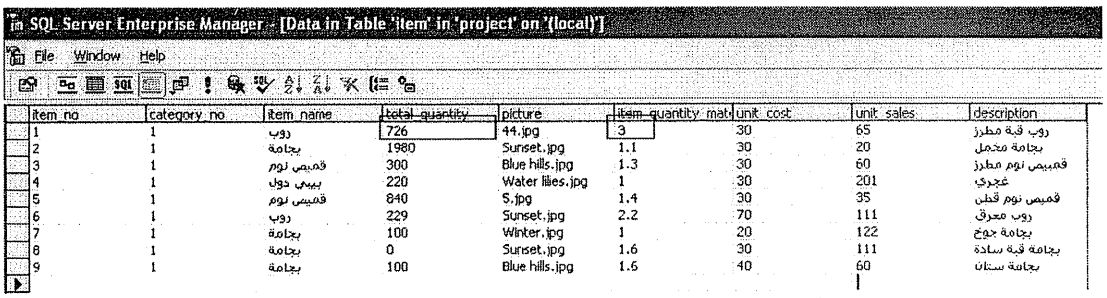
Figure [6.35] Bill table after change Snapshot

6.1.10 Manufacture item process snapshot

Adding item quantity page

Figure [6.36] adding item quantity page Snapshot

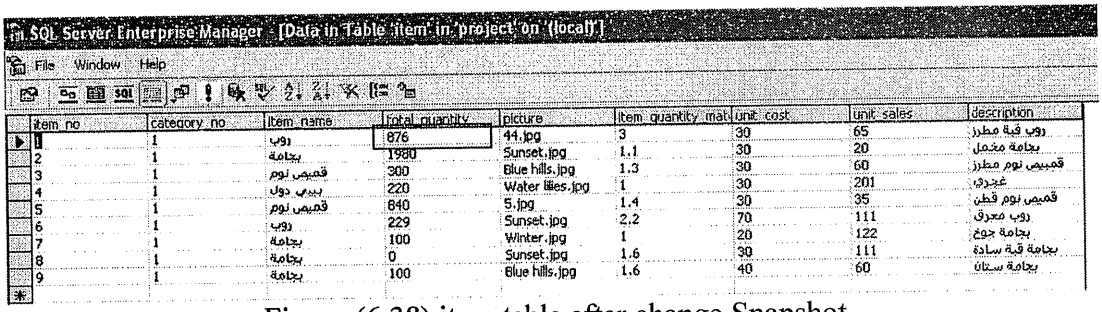
Item table before change



item_no	category_no	item_name	total_quantity	picture	item_quantity_mat	unit_cost	unit_sales	description
1	1	روب	726	44.jpg	3	30	65	روب قبة مطرز
2	1	بجامة	1980	Sunset.jpg	1.1	30	20	بجامة مخمل
3	1	قميص نوم	300	Blue hills.jpg	1.3	30	60	قميص نوم مطرز
4	1	بيبي دول	220	Water lilies.jpg	1	30	201	عجري
5	1	قميص نوم	840	S.jpg	1.4	30	35	قميص نوم قطن
6	1	روب	229	Sunset.jpg	2.2	70	111	روب معرق
7	1	بجامة	100	Winter.jpg	1	20	122	بجامة جوخ
8	1	بجامة	0	Sunset.jpg	1.6	30	111	بجامة قبة سادة
9	1	بجامة	100	Blue hills.jpg	1.6	40	60	بجامة ستان

Figure [6.37] item table before change Snapshot

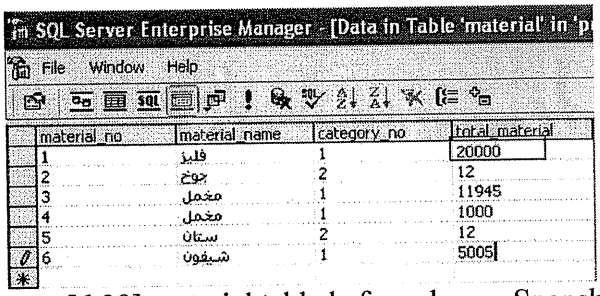
Item table after change



item_no	category_no	item_name	total_quantity	picture	item_quantity_mat	unit_cost	unit_sales	description
1	1	روب	676	44.jpg	3	30	65	روب قبة مطرز
2	1	بجامة	1980	Sunset.jpg	1.1	30	20	بجامة مخمل
3	1	قميص نوم	300	Blue hills.jpg	1.3	30	60	قميص نوم مطرز
4	1	بيجي دول	220	Water lilies.jpg	1	30	201	غجري
5	1	قميص نوم	840	5.jpg	1.4	30	35	قميص نوم قطن
6	1	روب	229	Sunset.jpg	2.2	70	111	روب معرق
7	1	بجامة	100	Winter.jpg	1	20	122	بجامة جوخ
8	1	بجامة	0	Sunset.jpg	1.6	30	111	بجامة قبة سادة
9	1	بجامة	100	Blue hills.jpg	1.6	40	60	بجامة ستان

Figure (6.38) item table after change Snapshot

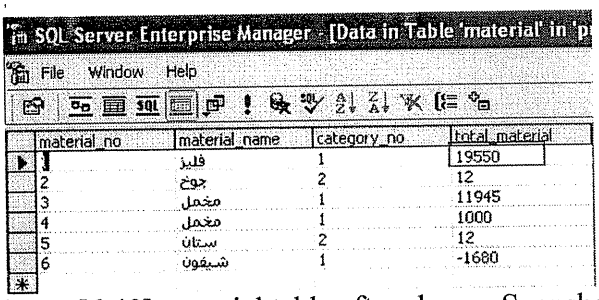
Material table before change



material_no	material_name	category_no	total_material
1	فليز	1	20000
2	جوخ	2	12
3	مخمل	1	11945
4	مخمل	1	1000
5	ستان	2	12
6	شيفون	1	5005

Figure [6.39] material table before change Snapshot

Material table after change



material_no	material_name	category_no	total_material
1	فليز	1	19550
2	جوخ	2	12
3	مخمل	1	11945
4	مخمل	1	1000
5	ستان	2	12
6	شيفون	1	-1680

Figure [6.40] material table after change Snapshot

** Note:

3 * 150 =450
20000-450=19550

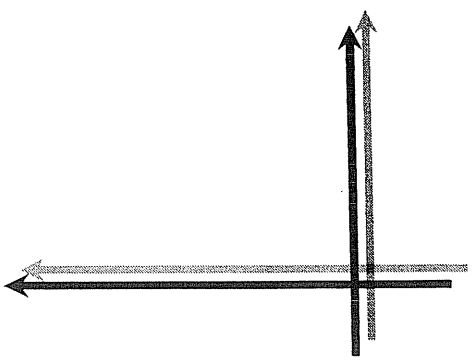
Chapter seven

7

Maintenance

Chapter seven

- Introduction
- Migration
- Maintaining the system



7.1 Introduction

At this chapter we will provide and explain the process and techniques that guidelines the system administrator to keep tracing and maintaining the system after running it.

At this stage, the system which had been developed and tested is needed to take place in the real working environment. However, it is a kind of joke to think about customers as if they were the developers of the system, they are just end users, so those users must be provided by sufficient information and guidance about system deployment and how it could be maintained. In this chapter, we describe the real working environment within which the system will operate, as well how it could be established, how it could be migrated, and how it is maintained.

7.2 Migration

The deployment of the system must be preceded by certain steps so that to work properly within its environment; the production environment has to be established, configured, and a decision of operating on the new system must be taken considering all constrains and risks of the process of migration to the new system. Toward deploying and migrating to the new system we describe here the steps that must be done:

- *Establishment of the production environment:*

  The minimal requirements of deploying the system are described in chapter two (system specification), and the needed configurations for the machine running the system are described in chapter five (Coding and implementation). For example, we say that our software system will not operate on a machine that doesn't have the .NET Framework, so that the company that decides to work on our system must have all of the production environment elements available

- *Deciding to deploy the new system*:

  A decision of the deployment of the new system must be taken accordingly with the plan of the deployment and the migration to this new system, these are managerial issues and managers with their company capabilities are responsible for doing so. Does this system cover our requirements? Do we have the minimal requirements to deploy it? How could we deploy it?

Here we say that our system was tested and we found that it works well as it should be, the system can operate immediately whenever a suitable production environment is created, but we say that our software is an integrated system that must run with its all parts integrated, nevertheless the system may not work properly. Now, the way of migration to it depends on the technicians and managers recommendations in the company that wishes to deploy this new system.

- *Running the system*

The aim of having the new system is to work on it, after it was being designed and tested during the development process, and whenever a company buys it, decides to migrate to it, and deploy it, the system will operate and run.

7.3 Maintaining the system

When running the system there is always a possibility for failure, errors, and other types of problems that may appear. We describe here the maintenance plan that covers some of the actions that is to be taken when certain situations occur:

1. **Backup:**

   Database is the nerve of the system, so it is the most important component of the system, because it contains all tables, views, and stored procedures that contain the data of the system, so we should keep this data from losing and harmful by specific means such as take a backup for this database periodically to enable maintain the system in the future.

   This is an issue where the adaptation of the backup methodology (type and time interval) is determined and implemented by the company working on the system itself. The backup on the system database could be configured by means that are provided by the producer company of the DBMS that we have used in our software system which is Microsoft© on its DBMS product (SQL Server 2000).

2. **Error reporting :**

When errors occur, certain actions are to be taken as the contract agreement describes the conditions and situations of the maintenance and the responsibility allocation on the contract sides.

Our system provides a simple way for reporting certain errors immediately when they occur, means of messaging on the computer screen describing what the problem is are implemented.

3. **Upgrade:**

This process is to be carefully implemented if it is essential and by considering the software producers. Upgrading to the new system doesn't need a previous operating E-commerce software, all what it needs is the production environment that was described earlier in this chapter.

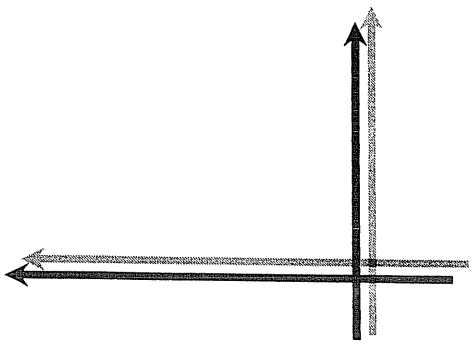
Chapter eight

8

Conclusions & Recommendations

Chapter eight

- *Introduction*
- *Conclusion*
- *Recommendation.*
- *Reference*



8.1 Introduction

In this chapter we will provide recommendations that result during our system development and implementation processes.

8.2 Conclusion

The work team has concluded the following:

- The web-based systems reduced the time and efforts for administrators and customers.
- The web-based systems Increased customer satisfaction

8.3 Recommendations

The project team recommend for the following works and enhancements on the system:

1. We recommend to applied our project in such organization in Hebron
2. We recommend to expand our project to be able interact with supplier
3. We recommend to encourage the IT organization such as FFKITCE to developed such system for other organization in our reign
4. We recommend to developed our portal to connect economic governmental organization that work in the same scope

References:

1. Sommerville, Ian, *Software Engineering*, 7th edition, Addison-Wesley, 2001.
2. Microsoft SQL server 2003 Help Topics.
3. Microsoft .NET Framework SDK v1.1.
4. www.asp.net/forums.
5. MSDN, developing Microsoft ASP.NET web application using visual studio.NET.