

# Palestine Polytechnic University

College of Administrative Science and Informatics  
Information Technology Department

## E-Payment

(Applied on Hebron Municipality)

### Project Team

Nour Eden Qadi

Fadel Ubu Eisheh

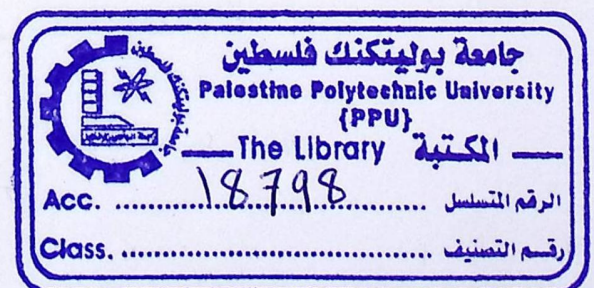
Wisam Haniny

### Project Supervisor

Mr. Mousa Rfaaieh.

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

Jan 06



# E-Payment

(Applied on Hebron Municipality)



**Project Team:**

**Nour Eden Qadi**

**Fadel Ubu Eisheh**

**Wisam Haniny**

**Project Supervisor**

**Mr. Mousa Rfaaieh.**

**Graduation Project**

**Submitted to Information Technology Department in the College of  
Administrative Science & Informatics  
Palestine Polytechnic University**

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

-----

-----

-----

**Date -----**

## Abstract

The main goal of the project is to build web-based system Hebron Municipality. This project offers Internet-based paying system facility through which the consumer can view and pay his bills via Hebron Municipality web site.

In addition, he can view his consumption report. To enable consumer to use this e-Services, he must register firstly in Hebron Municipality web-based system and have sufficient Bank Account balance from Alaraby Bank in order to get benefits from this service.

The system does not provide the paying service only but also gives consumer the ability to report their comments and other essential e-Services.

## Dedication

*To my beloved mother.....*

*To my great father.....*

*To my brothers.....*

*To my Friends.....*

*Nour Eden*

*To my parents*

*To my Brothers & Sisters*

*To my friends, teachers*

*To every one granted me the encouragement and support.*

*Fadel*

*To my lovely Parents, Sister.....*

*To every mother in Palestine*

*Wisam*

# 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 **Mr. Mousa Arfaeah**, 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 not forgetting our grateful to **Mohammad Sonjok**.

## TABLE OF CONTENTS

Abstract.....	I
Dedication.....	II
Acknowledgment.....	III
Table of Contents.....	IV
List of Tables.....	VII
List of Figures.....	VIII

### Chapter One (Introduction)

1.1 Initiation.....	2
1.2 Importance of E-Bill and E-Payment System .....	2
1.3 Paying Bills Online Using Bank Account.....	3
1.4 Hebron Municipality .....	4

### Chapter two (System Specification)

2.1 Introduction.....	6
2.2 System Objectives.....	7
2.3 Functional Requirements.....	7
2.4 Requirements Specifications.....	8
2.5 Non-Functional Requirement .....	9
2.6 Allocation of Roles of System Developers.....	9
2.7 Constraints.....	10
2.8 Feasibility Study.....	10
2.8.1 Alternatives.....	10
2.8.1.1 Environment.....	10
2.8.1.2 Topology.....	13
2.8.2 Cost-Benefit Analysis.....	14
2.8.3 Risk Evaluation.....	15
2.8.4 Economical Study.....	16
2.8.5 Time Feasibility.....	18
2.8.6 Technical Feasibility.....	19
2.8.7 Legal Feasibility.....	19

### Chapter three (Software Requirements Specification)

3.1 Introduction .....	21
3.2 Functional description .....	22
3.3 Functional Specification.....	25
3.3.1 Consumer Function Requirement.....	25
3.3.2 Administration Function Requirement.....	33
3.4 System Contexts (Relations with Other Systems).....	36

3.5 Information Description.....	36
3.5.1 System Data Flow Diagrams.....	37
3.5.2 Data Dictionary .....	38
3.5.3 System Interface Description .....	41
3.5.4 Database Requirements.....	42
3.5.4.1 Hebron Municipality Database Tables.....	42
3.5.4.2 Bank Database Tables.....	44

## Chapter Four (System Design)

4.1 Introduction.....	46
4.2 Functional Design .....	47
4.3 Input/output Design.....	83
4.3.1 Input Screens Design.....	84
4.3.1.1 Consumer Input Form.....	84
4.3.1.2 Bank Account Input Form.....	90
4.3.2 Output Screens Design.....	91
4.3.2.1 Consumer Output Form .....	91
4.4 Database Design.....	96
4.4.1 Database Dictionary.....	96
4.4.1.1 Hebron Municipality DB Dictionary.....	96
4.4.1.3 Bank DB Dictionary.....	99
4.4.2 Database Model.....	100
4.5 Test Plan.....	101

## Chapter Five (Implementation and Coding)

5.1 Introduction .....	103
5.2 Coding Programming Language.....	103
5.3 Establishment of Development Environment.....	103

## Chapter Six (System Testing)

6.1 Introduction.....	117
6.2 Unit and Module Testing.....	118
6.3 Integration Testing.....	122
6.4 System Testing.....	122
6.5 Acceptance Testing.....	122
6.6 Sample Snapshots.....	122
6.6.1 Registration Process Snapshots.....	123
6.6.1.1 Login Page .....	123
6.6.1.2 Registration Process .....	124
6.6.1.3 Consumer Home Page .....	125
6.6.1.4 Consumer Table Before Changes .....	126
6.6.1.5 Consumer Table After Changes .....	126
6.6.1.6 Consumer Answer Table Before Changes .....	127

6.6.1.7 Consumer Answer Table After Changes .....	127
6.6.2 Payment Process Snapshots.....	128
6.6.2.1 Login Page .....	128
6.6.2.2 Consumer Home Page .....	129
6.6.2.3 Dept and Collection Page .....	130
6.6.2.4 Bill Details Page .....	131
6.6.2.5 Disclaimer Page .....	132
6.6.2.6 Bank Account Information Page .....	133
6.6.2.7 Agreement Pay Page .....	134
6.6.2.8 Consumption Receipt Report.....	135
6.6.2.9 Bill Details before Changes .....	136
6.6.2.10 Bill Details after Changes .....	136
6.6.2.11 Bank Account Table Before Changes....	137
6.6.2.12 Bank Account Table After Changes....	137
6.6.2.13 Bank Transaction Table Before Changes.....	138
6.6.2.14 Bank Transaction Table After Changes.....	138

## Chapter Seven ( Maintenance)

7.1 Introduction.....	140
7.2 Migration.....	140
7.3 Maintaining The System.....	141

## Chapter Eight (Conclusion and Recommendations)

8.1 Introduction .....	144
8.2 Conclusion.....	144
8.3 Recommendations .....	144
References.....	145

## List of Tables

Table	Page
Table 2.1 Function- Objectives Matching Table.....	7
Table 2.2 Hardware Costs.....	16
Table 2.3 Development and Implementation Software.....	16
Table 2.4 Human Cost.....	17
Table 2.5 Total Cost.....	17
Table 2.6 Time Schedule.....	18
Table 3.1 Functional Description.....	24
Table 3.2 Data Dictionary.....	40
Table 4.1 Consumer Information DB Table.....	96
Table 4.2 Bill Details DB Table.....	96
Table 4.3 Comments and Suggestions DB Table.....	97
Table 4.4 Consumer Answer DB Table.....	98
Table 4.5 Consumer Question DB Table.....	98
Table 4.6 Area DB Table .....	98
Table 4.7 Bank Account Table .....	99
Table 4.8 Bank Transaction Table .....	99
Table 6.1 Login Consumer Test Cases .....	120
Table 6.2 Check Bank Account Sufficiency Test Cases.....	121

## List of Figures

Figures	Page
Figure 1.1 E-bill /e-payment process .....	3
Figure 2.1 Gantt chart.....	18
Figure 3.1 System Contexts Diagram .....	36
Figure 3.2 Consumer Data Flow Diagram .....	37
Figure 4.1 Consumer Registrations User interface design.....	47
Figure 4.2 Consumer Registrations Flowchart.....	48
Figure 4.3 Consumer Login User interface design.....	49
Figure 4.4 Consumer login Flowchart.....	50
Figure 4.5 Consumer Logout User interface design.....	51
Figure 4.6 Consumer Logout Flowchart.....	52
Figure 4.7 Change Login Password User interface design.....	53
Figure 4.8 Change Login Password Flowchart.....	54
Figure 4.9 Debt and Collection User interface design.....	55
Figure 4.10 Debt and Collection Flowchart.....	56
Figure 4.11 Viewing Bill Details User interface design.....	57
Figure 4.12 Viewing Bill Details Flowchart.....	57
Figure 4.13 Viewing Helps User interface design.....	58
Figure 4.14 Viewing Helps User Flowchart.....	59
Figure 4.15 Viewing FAQ User interface design.....	60
Figure 4.16 Viewing FAQ Flowchart.....	60
Figure 4.17 Acquiring Comments and Suggestion from Consumer User interface design.....	61
Figure 4.18 Acquiring Comments and Suggestion from Consumer Flowchart.....	62
Figure 4.19 Get Bank Account Information User interface design.....	63
Figure 4.20 Get Bank Account Information Flowchart.....	63
Figure 4.21 Check Bank Account sufficiency User interface design.....	64
Figure 4.22 Check Bank Account sufficiency Flowchart.....	65
Figure 4.23 Update Bank Account User interface design.....	66
Figure 4.24 Update Bank Account Flowchart .....	67
Figure 4.25 Generate Bill Consumption Report User interface design.....	68
Figure 4.26 Generate Bill Consumption Report Flowchart .....	69
Figure 4.27 Update Consumer Information User interface design.....	70
Figure 4.28 Update Consumer Information Flowchart .....	71
Figure 4.29 Forget Password User interface design.....	72
Figure 4.30 Forget Password Flowchart .....	73
Figure 4.31 Unsubscribe Consumer Count User interface design.....	74
Figure 4.32 Unsubscribe Consumer Count Flowchart .....	75
Figure 4.33 Administrator Login User interface design.....	76
Figure 4.34 Administrator Login Flowchart .....	76
Figure 4.35 Administrator Logout User interface design.....	77
Figure 4.36 Administrator Logout Flowchart .....	77
Figure 4.37 Viewing Sent Comments User interface design.....	78

Figure 4.38 Viewing Sent Comments Flowchart .....	79
Figure 4.39 FAQ Updating and Maintaining User interface design.....	80
Figure 4.40 FAQ Updating and Maintaining Flowchart .....	80
Figure 4.41 Generate Bills (add or update bills) User interface design.....	81
Figure 4.42 Generate Bills (add or update bills) Flowchart.....	82
Figure 4.43 Consumer Login Form.....	84
Figure 4.44 New Consumer Registration Form .....	85
Figure 4.45 Forget Password Form .....	86
Figure 4.46 Update / View Personal Consumer Information Form .....	87
Figure 4.47 Change Password Form .....	88
Figure 4.48 Comments Form .....	88
Figure 4.49 Agreement Terms and Condition Form.....	89
Figure 4.50 Disclaimer Form .....	90
Figure 4.51 Bank Account Information Form .....	91
Figure 4.52 Debts and Collections Form.....	92
Figure 4.53 Old Debts and Collections Form .....	93
Figure 4.54 Bill Details Form .....	94
Figure 4.55 Bill Consumption Report Form.....	95
Figure 4.56 Database Model .....	100
Figure 5.1 Integrated Development Environment in Visual Studio.NET.....	105
Figure 5.2 Installing IIS (Step One).....	108
Figure 5.3 Installing IIS (Step Two).....	109
Figure 5.4 Install Visual Studio .NET 2003.....	110
Figure 5.5 Mixed Mode Authentications (Step One).....	112
Figure 5.6 Mixed Mode Authentications (Step Two).....	113
Figure 5.7 Exchange Server 2003 Installation Wizards (Step One).....	114
Figure 5.8 Exchange Server 2003 Installation Wizards (Step Two).....	115
Figure 6.1 Testing Schedule.....	117
Figure 6.2 Login Consumer Execution Paths.....	119
Figure 6.3 Check Bank Account Sufficiency Execution Paths .....	121
Figure 6.4 Login Page Snapshot.....	123
Figure 6.5 Registration Process Snapshot .....	124
Figure 6.6 Consumer Home Page Snapshot .....	125
Figure 6.7 Consumer Table Before Changes Snapshot .....	126
Figure 6.8 Consumer Table After Changes Snapshot .....	126
Figure 6.9 Consumer Answer Table before changes Snapshot .....	127
Figure 6.10 Consumer Answer Table After Changes Snapshot.....	127
Figure 6.11 Login Page Snapshot .....	128
Figure 6.12 Consumer Home Page Snapshot .....	129
Figure 6.13 Debt and Collection Page Snapshot.....	130
Figure 6.14 Bill Details Information Page Snapshot.....	131
Figure 6.15 Disclaimer Page Snapshot .....	132
Figure 6.16 Bank Account Information Page Snapshot .....	133
Figure 6.17 Agreement Pay Page Snapshot .....	134
Figure 6.18 Consumption Receipt Report Page Snapshot .....	135
Figure 6.19 Bill Details Table Before Changes Snapshot .....	136

Figure 6.20 Bill Details Table After Changes Snapshot .....	136
Figure 6.21 Bank Account Table Before Changes Snapshot.....	137
Figure 6.22 Bank Account Table After Changes Snapshot .....	137
Figure 6.23 Bank Transaction Table Before Changes Snapshot.....	138
Figure 6.24 Bank Transactions After Changes Snapshot.....	138

# Introduction

## The Context:

### 1.1 Initiation

### 1.2 Importance of E-Bill and E-Payment System

### 1.3 Paying Bills Online Using Bank Account

### 1.4 Hebron Municipality

## Chapter

## 1

# Introduction

The Content:

1.1 Initiation.

1.2 Importance of E-Bill and E-Payment System.

1.3 Paying Bills Online Using Bank Account.

1.4 Hebron Municipality.

## 1.2 Importance of E-Bill and E-Payment System

E-Bill and E-Payment System is the easy, smart way for consumers to securely receive, manage and pay their bills on the Internet. Using this system to get their services.

- The system can automate many processes that managers and their subordinates must accomplish.
- Time and money can be saved by using this system to do what previously could be done only manually.
- High level software let managers develop their own tools for daily monitoring and control of their operation.
- The system supports more and more tasks thereby to end any job in the high quality and little time.

## 1.1 Initiation

The consumers can pay their bills through the internet by this system, through using this system we will minimize the time and cost for both consumer and service provider "Hebron Municipality".

E-Payment is a secure online billing service. Nowadays, receiving and paying bills online (E-Payment) is becoming a necessity in our society. It is expected that E-Payment systems can provide fast, easy and secure system. Just by a click or hit on a button the user can make payment so easy.

Electronic billing and payment are efficient and cost-effective ways to provide online presentation of virtually any kind of document; from traditional bills, insurance notices and complex invoices, to flexible software solution that addresses your requirements for speed to market, resource efficiency, low cost and simplicity

Large size companies are looking for ways to satisfy the consumer needs. This presents a great opportunity for existing mail service providers, payment service providers, or new billing service providers to expand business and create new revenue streams.

## 1.2 Importance of E-Bill and E-Payment System

E-Bill and E-Payment System is the easy, smart way for consumers to securely receive, manage and pay their bills on the Internet. Using this system to get their services:

- The system can automate many processes that managers and their subordinates must accomplish.
- Time and money can be saved by using this system to do what previously could be done only manually.
- High level software let managers develop their own tools for daily monitoring and control of their operation.
- The system supports more and more task benefits to end any job at the high quality and little time.

### 1.3 Paying Bills Online Using Bank Account

Such payments can be made into any Bank account. Many people pay their monthly rent and other bills directly into the payee's Bank accounts. See the following figure that represents the main items that interact together in E-service cycle.

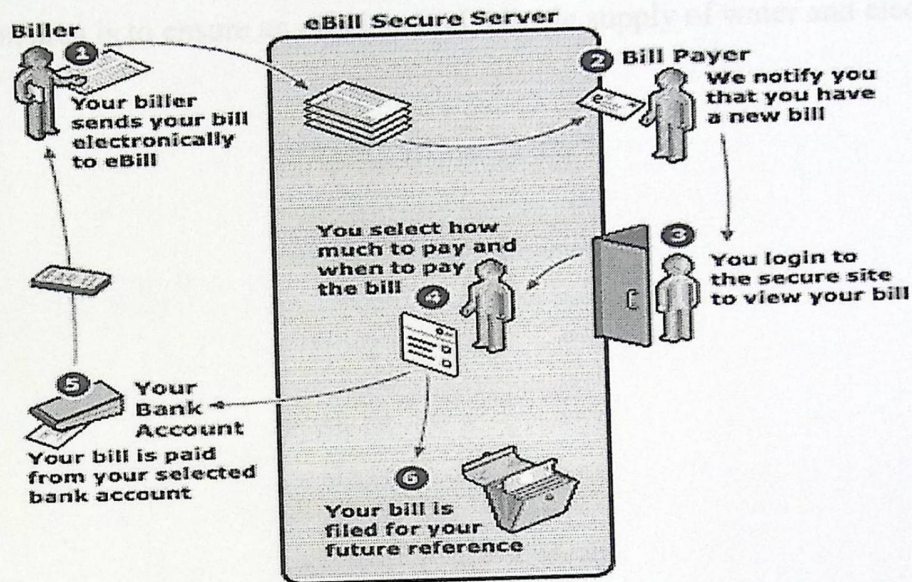


Figure [1.1] E-bill /e-payment process

Under this model figure [1.1], a biller like Hebron Municipality posts bills on its Web site, where consumer can view and pay them in any places in the world. This means that the consumer is required go to service provider Web site to pay all their bills.

E-service could be the solution, because of its huge number of benefits that could be gained, some of them are:

1. Reduce the time and effort during paying process.
2. Staying in touch with wide variety of service provider offers.
3. Serving consumers more easily.

By considering all these issues, we have decided to construct a system that serves as an intermediary web-based solution between the consumer and service provider(Hebron Municipality).

## 1.4 Hebron Municipality

Hebron Municipality is a public firm it is responsible for allot of services one of them distributing water in the Hebron.

The primary concern of Hebron Municipality in delivering water to its consumers in its concession area is to ensure an efficient and reliable supply of water and electricity.

# System Specification

The Content

2.1 Introduction

2.2 System Objectives

2.3 Functional Definition

2.4 Requirements Specification

2.5 Non-Functional Description

2.6 Allocation of roles of system developers

2.7 System Constraints

2.8 Feasibility Study

## 2.1 Introduction

After several meetings with the firm administration, we understand the specifications

**Chapter****2**

# System Specification

The Content:

- 2.1 Introduction.
- 2.2 System Objectives.
- 2.3 Functional Definition.
- 2.4 Requirements Specification.
- 2.5 Non-Functional Description.
- 2.6 Allocation of roles of system developers.
- 2.7 System Constraints.
- 2.8 Feasibility Study.

2.1 Introduction

After several meetings with the firm administration, we understand the specifications of the system requirements, here we demonstrate that specifications which include in this chapter.

This chapter includes the following:

- The System objectives.
- The Functional requirements.
- The Requirements Specification
- The Non-Functional requirements.
- Allocation of roles of system developers.
- The Project's Constraints.
- The Feasibility study
  - ❖ Alternatives
  - ❖ Cost-Benefit analysis
  - ❖ Risk evaluation
  - ❖ Economical Study
  - ❖ Time Feasibility
  - ❖ Technical feasibility
  - ❖ Legal feasibility


2.2 System Objectives

This system aims to serve as an electronic government services that plays as the intermediary between Hebron Municipality and consumer, to reach this goal, certain objectives should be achieved:

1. Building system database that contains consumers, and Hebron Municipality files.
2. Enable consumer to view and pay the Hebron Municipality bills online.
3. Enable consumer to get detailed Hebron Municipality consumption reports.
4. Reducing the time and cost for both consumers and Hebron Municipality.
5. Increasing the efficiency of service.
6. Having reliable and secure data storage.

2.3 Functional Requirements

The following functions and services should be provided by the system:

1. Browsing bills information.
2. Online payment bill process.
3. Provide consumer with authenticated registration via specific authorization process.
4. Collect the needed information about each consumer.
5. Provide consumer with help tools and facilities.
6. Provide managerial and administration tools and facilities.

	Objective1	Objective2	Objective3	Objective4	Objective5	Objective6
Function1		X	X		X	X
Function2		X	X	X	X	X
Function3					X	X
Function4	X					
Function5					X	
Function6	X					X

Table [2.1] Function- Objectives Matching Table

## 2.4 Requirements Specification

This part contains the specification of the functional requirements as follow:

- 1) Browsing bills information.
  - 1.1 *Viewing debts and collections.*
  - 1.2 *Viewing bill details.*
- 2) Online payment bill process.
  - 2.1 *Get bank account information.*
  - 2.2 *Check bank account.*
  - 2.3 *Update bank account.*
  - 2.4 *Generating bill consumption report.*
- 3) Provide consumer with authenticated registration via specific authorization process.
  - 3.1 *Consumer registration.*
  - 3.2 *Consumer login.*
  - 3.3 *Consumer logout.*
  - 3.4 *Unsubscribe Consumer account.*
- 4) Collect the needed information about each consumer.
  - 4.1 *Update login password.*
  - 4.2 *Update consumer information.*
  - 4.3 *Forget Password.*
- 5) Providing consumer with help tools and facilities.
  - 5.1 *Viewing help.*
  - 5.2 *Viewing FAQ.*
  - 5.3 *Acquiring comments and suggestion from consumer.*
- 6) Provide managerial and administration tools and facilities (locally).
  - 6.1 *Administrator login.*
  - 6.2 *Administrator logout.*

6.3 Viewing the sent comment.

6.4 Update and maintain FAQ.

6.5 Bill generation.

## 2.5 Non-Functional Requirement

The following functions appear in this system:

1. **Security:** means for ensuring privacy and protection for consumer's information and the specific authentication should be implemented used to permit accessibility.
2. **Operability:** simultaneous operations are allowed to take place in the site and are managed to not allow interruption or what ever type of corruption or errors and consumer should be allowed freely to view and navigate contents of the site without time, cost, or load limitations.
3. **Speediness and performance:** the system response to inputs of consumer with speedily and high performance.
4. **Usability:** easy and simple for consumer to make agreements with the web site to pay through it.
5. **Compatibility:** all types of browsers (clients) can access and brows the site.
6. **Appearance:** attractive and competitive methodology for displaying web interface.

## 2.6 Allocation of roles of system developers:

1. **Software engineer:** responsible for the documentation and tracing of the development stages of the software.
2. **Leader:** responsible of planning, scheduling and controlling flow of system development processes.
3. **Interface designer:** responsible of the web design and user interface of the system (GUI).
4. **Programmer:** responsible of the system programming, implementation and testing so he must have enough experiences in the ASP.NET development environment.

## 2.7 Constraints:

In this part we say about the constraints and problems that faced us when we making the Hebron municipality project.

1. The system is to be developed within a fixed time interval; just fifteen weeks are available for developing the system.
2. Since E-payment is a new idea, consumers will not really trust with this idea, exactly in Palestine.
3. Lack of information about using internet regarding to the consumers.
4. The weak of telecommunication infrastructure in Palestine (especially the internet).

## 2.8 Feasibility Study

The most important issue when any body make a system is to evaluate its benefits against its cost.

In this section, we show the 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.

### 2.8.1 Alternatives:

In this section we choosing which methodology is the best for developing our system? , What is the available methodology? , What technologies are available? And which of them is the most appropriate? These questions and others are covered and explained through this section.

#### 2.8.1.1 Environment

##### Server-Side Technologies for Providing Dynamic Content

Each of these technologies relies on a modular attachment added onto the web server rather than the browser. Consequently, only HTML, and any client-side script, is sent back to the browser by the web server. In other words, none of the server-side code is sent back.

Server-side technologies have a more consistent look than client-side ones and it doesn't take that much extra learning to move between some of the server-side technologies excepting CGI.

### **Active Server Pages (ASP)**

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.

#### **ASP.NET Advantage:**

- ASP.NET make web developing much easier as developing windows applications.
- If you have any experience in visual basic or access you will see how familiar in the ASP.NET

#### **ASP.NET Disadvantages:**

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

## Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a mechanism for creating scripts on the server, which can then be used to create dynamic web applications. CGI is a module that is added to the web server. It has been around for quite a bit longer than even ASP, and right now, a large proportion of dynamically created web pages are created using CGI and a scripting language. However, it's incorrect to assume that CGI does the same job as ASP.NET or ASP. Rather, CGI allows the user to invoke another program (such as a Perl script) on the web server to create the dynamic web page, and the role of CGI is to pass the user supplied data to this program for processing. However, it does provide the same end result – a dynamic web application.

### CGI Advantage:

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

### CGI Disadvantages:

- It is not easy for a beginner to learn how to program such modules.
- CGI requires a lot of server resources, especially in a multi-user situation.
- It adds an extra step to our server-side model of creating dynamic content: namely, it's necessary to run a CGI program to create the dynamic page, before the page is processed on the server.

What's more, the format in which CGI receives and transmits data means that the data is not easily manipulated by many programming languages, so you need one with good facilities for manipulating text and communicating with other software. The most able programming languages that can work on any operating system for doing this are C, C++ and Perl.

While they can adequately do the job for us, they are some of the more complex languages to learn. Visual Basic doesn't offer adequate text handling facilities, and is therefore rarely used with CGI.

### 2.8.1.2 Topology

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

#### **Advantages:**

1. Cover the entire world without geographical restrictions.
2. Faster in registering new consumer.
3. No networking problems, by developing new reliable communication protocols.
4. More scalable.

#### **Disadvantages:**

1. Security issue is the major challenge, there is no web based system until now achieve full secure environment.

#### **Conclusion:**

By comparing the server- side environments technologies for providing dynamic web pages we as a team decided to choose the ASP technology as a development environment to build our web based system because the .NET match our programming capabilities, more than other developments environment.

### 2.8.2 Cost-benefit analysis

The impact of E-commerce on the economics will improve economic efficiency; several economic benefits are gained when an E-commerce system takes place in world business.

According to our feasibility study on developing our system, we conclude that it is a great step for minimizing the costs of trade over the world.

#### ***Service Provider benefits:***

- Decrease the managerial time and efforts works.
- The service is not restricted with time and distance; it's providing 24 hour /day.

- Increase the efficiency by decreasing the error rate during payment processes.

2.2.2 Risk evaluation

**Consumer benefits:**

- Reduce the time and effort too.
- Make the payment more easily more enjoyable.

According to our feasibility study on developing our system, we conclude that it is a great step for minimizing the costs and efforts.

- 1- The E-payment that is well integrated from companies
- 2- The need for systems that are using the internet and the network services
- 3- Inefficient telephone services
- 4- The fear of vulnerability
- 5- The security risks
- 6- The security risks

Proposed solution:

1. Increase the security
2. Encouraging the public
3. Implementing
4. Increase the security

### 2.8.3 Risk evaluation

There is an opportunity for several risks that may face our system; this section covers them accordingly with the proposed solutions.

#### Risks List:

- 1- The resistance for change by managers and employees in companies when there is a possibility of changing their traditional way of marketing.
- 2- The E-Payment idea is still not accepted from consumer.
- 3- The need for special Web servers and other infrastructures, in addition to the network servers (additional cost).
- 4- Insufficient telecommunication data rate (network speed).
- 5- The fear of customers to deal financially by electronic means such as the internet.
- 6- The security issue and the lack of reliability on computer networks that may be invaded illegally.

#### Proposed solution:

1. Educate the managers and employees with new modern marketing means to cope with latest technology.
2. Encouraging the public class to use the E-commerce application using media.
3. Implementing reliable methods for financial transactions for companies and consumer.
4. Increase the quality of telecommunication services, increase the data rate speed.

These methods will overcome the risks of the security issue and fear from dealing with internet for financial transactions.

### 2.8.4 Economical Study

In this section, resources and cost are described for both the development and implementation requirements.

▪ Hardware Costs For System Development:

No.	Item	Quantity	Specifications	Cost
1	Microsoft compatible PC	1	Pentium 4, 2800 MHz 1 MB cache memory RAM 512 MB Hard Disk drive 80 GHz CD-ROM 52X Floppy drive 1.44 Keyboard and mouse Monitor 17"	\$950
2	Printer	1	HP 1010	\$70
3	Network Adapter	2	3com (100Mbps)	\$30
4	Cables	1	Twisted pair cables (RJ45)	\$20
<b>Total</b>				<b>\$1070</b>

Table [2.2] Hardware Costs

▪ Development and Implementation Software:

Table [2.3] shows the software products required and their costs for the system development and implementation.

No.	Software	Cost
1	Windows XP professional (Recommended in client side)	\$180
2	Windows 2003 server (Recommended in servers)	\$200
3	SQL server 2000	\$110
4	Multimedia design tools	\$230

5	Microsoft Visual Studio.Net V1.0 2003	\$480
6	Microsoft Office2003	\$450
<b>Total</b>		<b>\$1650</b>

Table [2.3] Development and Implementation Software

▪ work cost:

As shown in Table [2.4] the team costs are estimated as the market prices for software developers.

Number	Team Role	Hours/week	Cost\hour	Total
1	Programming	20	\$25	\$500
2	Software Engineering	20	\$25	\$500
3	Interface Design GUI	20	\$25	\$500
<b>Total cost/week</b>				<b>\$1500</b>
<b>Total work Cost [15 week]</b>				<b>22500</b>

Table [2.4] Work Cost.

▪ Other accessories cost:

Other costs such as books, papers, pens, flat (rent, water, elactresity), and transportations are estimated to be \$50/ week.

▪ Total Cost:

The table fellow represent the total cost of all system cost:

Number	System Cost	Total
1	Hardware cost	1070\$
2	Software cost	1650\$
3	work cost	22500\$
4	Other cost	750\$
<b>Total cost</b>		<b>25970\$</b>

Table [2.5] Total Cost

2.8.5 Time Feasibility

In this section we show how we have allocated the given period over the development stages, the time interval that was available to develop the system was 15 weeks, we have distributed this interval over all of the development process. Table [2.6] shows the time schedule for all development tasks.

▪ Time Schedule:

As shown below in Table [2.6], all of the system development tasks are distributed over the available fifteen weeks. Some of these tasks were taken place in parallel.

Task	Work	Time in weeks
T1	Information gathering and System specification.	2
T2	Software requirement specification.	2
T3	System Design.	6
T4	Coding and implementation.	6
T5	System Testing.	3
T6	System Maintenance.	2
T7	Documentation.	15

Table [2.6] Time Schedule.

▪ Gantt chart for time schedule:

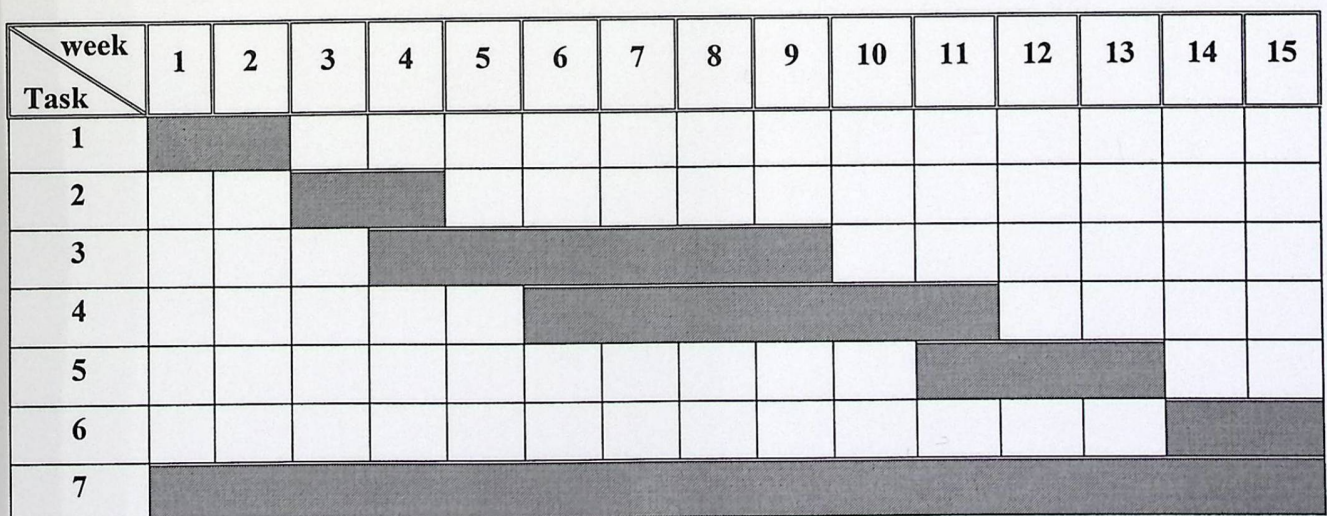


Figure [2.1] Gantt Chart

### **2.8.6 Technical feasibility**

This project requires a programming experience, and requires an experience in designing web pages and in dealing with database systems (building, interacting with database tables considering security issues, availability, etc). Team work members have an experience to some content in doing such application; they have experience in many programming languages such as C, Visual Basic, Java, and others. We can gain help from other expert persons in the university or by asking for it on the web.

### **2.8.7 Legal feasibility**

While looking to our situation, there are limitations or policies to make our project. When implementing it in the real business environment, governmental regulation, and other policies should be considered because there is a lot of financial transactions.

3.1 Introduction

The software requirements specifications which cover the over all functionality for  
be described, as well that will be classified according to their roles  
include that requirements specifications which relate:

Chapter

3

System Contexts

Information description

Validation

Information description

# Software Requirements Specification

The Content:

- 3.1. Introduction.
- 3.2 Functional Description.
- 3.3. Functional Specifications.
- 3.4. System Contexts.
- 3.5. Information Description.

### 3.1 Introduction

The software requirements specifications which cover the over all functionality for the system software will be described, as well that will be classified according to their roles in the system. We demonstrate here that requirements specifications which include:

- Functional description.
- Functional specifications
- System Contexts
- Information description
- Validation criteria.
- Information description.

- Viewing debts and collections: The registered consumer should be able to view last bills of current year status, does the bill paid or not, old year balance and get him if he has 8 free copy.
- Viewing bill details: The registered consumer should be able to view bills.
- Get bank account information: Checking the validity of the bank account.
- Check bank account: The registered consumer should be confirmed by the bank account ID and check if he has sufficient balance to pay bill.
- Update bank account: This function updates the bank account after payment transaction is completed.
- Generating bill consumption report: The registered consumer should be able to view their consumption bill reports.

### 3.2 Functional description

In this section we describe the major SW functions that meet our system requirements, Table [3.1] shows the functional description for the software system which meet the definition of the requirements which were previously described in chapter two (System Specification):

System requirements	Software Functional Description
<p>1) Browsing bills information</p>	<ul style="list-style-type: none"> <li>• <b>Viewing debts and collections:</b> The registered consumer should be able to view last bills of current year status, does the bill paid or not, old year balance and tell him if he has 8 free cups.</li> <li>• <b>Viewing bill details:</b> The registered consumer should be able to view bills.</li> </ul>
<p>2) Online payment bill process.</p>	<ul style="list-style-type: none"> <li>• <b>Get bank account information:</b> Checking the validity of the bank account.</li> <li>• <b>Check bank account:</b> The registered consumer should be confirmed by the bank account DB and check if he has sufficient balance to pay bill.</li> <li>• <b>Update bank account:</b> This function updates the bank account after payment transaction is completed.</li> <li>• <b>Generating bill consumption report:</b> The registered consumers should be able to view their consumption bill reports.</li> </ul>

<p>3) Provide consumer with authenticated registration via specific authorization process.</p>	<ul style="list-style-type: none"><li>• <b>Consumer registration:</b> Each subscriber in Hebron Municipality able to register in web-based system to enable him to pay his bill online.</li><li>• <b>Consumer login:</b> This function will be the only method for consumer to login to their account login using Consumer No and password.</li><li>• <b>Consumer logout:</b> This function enables consumers to end their sessions.</li><li>• <b>Unsubscribe Consumer Account:</b> This function allow the consumer to deactivate his account from Hebron Municipality database records.</li></ul>
<p>4) Collect the needed information about each consumer.</p>	<ul style="list-style-type: none"><li>• <b>Change login password:</b> The registered consumer should be able to change login password.</li><li>• <b>Update consumer information:</b> This function views or updates the personal information (E-mail) that belongs to specific consumer.</li><li>• <b>Forget Password:</b> This function enable the consumer to retrieve his account if he forget password via consumer E-mail.</li></ul>

<p>5) Providing consumer with help tools and facilities.</p>	<ul style="list-style-type: none"> <li>• <b>Viewing help:</b> The consumer should get help to deal with Hebron Municipality web-based system.</li> <li>• <b>Viewing FAQ:</b> The user should be able to view the frequently asked questions about the system.</li> <li>• <b>Acquiring comments and suggestion:</b> This function enables the consumers to send his comments and suggestions about real services.</li> </ul>
<p>6) Provide managerial and administration tools and facilities.</p>	<ul style="list-style-type: none"> <li>• <b>Administrator login:</b> This function will be the only method for administrator to login to Hebron Municipality database (locally).</li> <li>• <b>Administrator logout:</b> This function enables Administrator to end his session.</li> <li>• <b>Viewing the sent comments:</b> The administrator should be able to view comments that sent by consumer via outlook express.</li> <li>• <b>Update and maintain FAQ:</b> The administrator should be able to add, delete and modify FAQ's.</li> <li>• <b>Generate bills:</b> The administrator should be able generate bills, add, and modify bills.</li> </ul>

Table [3.1] Functional Description.

### 3.3 Functional specifications

In this section we describe the major SW functions that meet our system requirements which were previously described in chapter two (System Specification):

#### 3.3.1 Consumer Functional Requirement.

Function	<i>Viewing Dept And Collection.</i>
<b>Description</b>	The registered consumer should be able to view last bills of current year status, does the bill paid or not, old year balance and tell him if he has 8 free cups.
<b>Inputs</b>	Press on collection link.
<b>Source</b>	Hebron Municipality database.
<b>Outputs</b>	View current year status.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	The registered consumer must have been logged in Hebron Municipality website, and request to view his Bills Status.
<b>Post-condition</b>	None.
<b>Validation</b>	None.

<b>Function</b>	<i>Viewing bills Details.</i>
<b>Description</b>	The registered consumer should be able to view bills.
<b>Inputs</b>	Selecting E-bill details.
<b>Source</b>	Hebron Municipality database.
<b>Outputs</b>	Bill is displayed.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	The consumer must have been logged in Hebron Municipality website consumer, and request to view his bill.
<b>Post-condition</b>	None.
<b>Validation</b>	None.
<b>Function</b>	<i>Get bank account info.</i>
<b>Description</b>	Checking the validity of the bank account.
<b>Inputs</b>	Account No, password.
<b>Source</b>	Consumer bank account.
<b>Outputs</b>	None.
<b>Destination</b>	None.
<b>Pre-condition</b>	Bank account information is available.
<b>Post-condition</b>	Ready or not to perform financial transactions.
<b>Validation</b>	<ol style="list-style-type: none"> <li>1. All fields should be filled with information or making a selection, (change the default setting).</li> <li>2. The bank account No should be entered as digit numbers.</li> </ol>

<b>Function</b>	<i>Check Bank account Sufficiency.</i>
<b>Description</b>	The registered consumer should be confirmed by the bank DB and check if he has sufficient balance to pay bill.
<b>Inputs</b>	None.
<b>Source</b>	Bank account database.
<b>Outputs</b>	Sufficient Bank account balance.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	Consumer should have valid bank account.
<b>Post-condition</b>	Select Hebron Municipality Services such as payments services.
<b>Validation</b>	None.

<b>Function</b>	<i>Update bank account.</i>
<b>Description</b>	This function updates the bank account after payment transaction is finished.
<b>Inputs</b>	None.
<b>Source</b>	Consumer.
<b>Outputs</b>	New bank account balance.
<b>Destination</b>	Bank database.
<b>Pre-condition</b>	Payment process is completed successfully.
<b>Post-condition</b>	Updating the new bank account balance and view consumption bill report.
<b>Validation</b>	None.

Function	<i>Generation Bill Consumption Report.</i>
<b>Description</b>	The registered consumer should be able to view their consumption bill reports.
<b>Inputs</b>	none.
<b>Source</b>	Hebron Municipality database records.
<b>Outputs</b>	Payment bill information.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	Payment process is completed successfully.
<b>Post-condition</b>	None.
<b>Validation</b>	None.
Function	<i>Consumer registration</i>
<b>Description</b>	Each subscriber in Hebron Municipality able to register in web-based system to enable him to pay his bill online.
<b>Inputs</b>	Consumer No, Consumer ID, password, Secret question, Secret answer, E-mail.
<b>Source</b>	Consumer.
<b>Outputs</b>	Login Consumer No, Consumer Name and Consumer Email.
<b>Destination</b>	New database records for the new consumer account information.
<b>Pre-condition</b>	The consumer should visit our home page.
<b>Post-condition</b>	The consumer can login and get e-services.
<b>Validation</b>	<ol style="list-style-type: none"> <li>1. All inputs must be filled correctly.</li> <li>2. The consumer No, consumer ID must match Hebron Municipality database record.</li> <li>3. The password must be at least 6 characters.</li> <li>4. Password doesn't include special character.</li> <li>5. Consumer No duplication also Not allowed.</li> <li>6. Password equal password confirmation.</li> <li>7. Email equal Email confirmation.</li> </ol>

Function	<i>Consumer Login.</i>
Description	This function will be the only method for consumer to login to their account login using Consumer No and password.
Inputs	Consumer No and password.
Source	Consumer.
Outputs	Consumer home page.
Destination	Screen.
Pre-condition	The consumer should have valid account.
Post-condition	Consumer can use any available.
Validation	<ol style="list-style-type: none"> <li>1. All fields should be filled with data.</li> <li>2. Consumer No and password must be meet data base record.</li> </ol>

Function	<i>Consumer Logout.</i>
Description	This function enable consumer to end their session on the home page consumer.
Inputs	Selecting to logout Link.
Source	Consumer.
Outputs	Hebron Municipality Home page.
Destination	Screen.
Pre-condition	An existing consumer session.
Post-condition	Return to the Home page in the website.
Validation	None.

Function	<i>Unsubscribe Consumer account.</i>
Description	This function allows the consumer to deactivate his account from Hebron Municipality database records.
Inputs	Selecting Unsubscribe Link.
Source	Consumer.
Outputs	Delete consumer account registration record from DB.
Destination	Hebron Municipality DB.
Pre-condition	Consumer should have account.
Post-condition	None.
Validation	None.

Function	<i>Change Login Password.</i>
Description	The registered consumer should be able to change login password.
Inputs	Old password, new password and password confirmation.
Source	Consumer.
Outputs	Tell consumer that the password has been changed.
Destination	Hebron Municipality database.
Pre-condition	Already logged in consumer with old password.
Post-condition	New password will take effect on the next login.
Validation	<ol style="list-style-type: none"> <li>1. The new password must be at least 6 characters long.</li> <li>2. New password equal password confirmation.</li> </ol>

Function	<i>Update Consumer Information</i>
<b>Description</b>	This function views or updates the personal information (E-mail) that belongs to specific consumer.
<b>Inputs</b>	Insert new personal information (E-mail).
<b>Source</b>	Hebron Municipality data base Records.
<b>Outputs</b>	Information that belong to consumer.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	The consumer should be login.
<b>Post-condition</b>	Consumer should have account.
<b>Validation</b>	The new e-mail must match the e-mail format.

Function	<i>Forget Password.</i>
<b>Description</b>	This function enables the consumer to retrieve his account if he forget password via E-mail.
<b>Inputs</b>	Consumer No, Secret Answer.
<b>Source</b>	Consumer.
<b>Outputs</b>	Forgetting password.
<b>Destination</b>	Hebron Municipality DB.
<b>Pre-condition</b>	Consumer should have account.
<b>Post-condition</b>	None.
<b>Validation</b>	All fields should be filled with information.

Function	<i>Viewing help.</i>
Description	The consumer should provide with help.
Inputs	Browse help links.
Source	Hebron Municipality database.
Outputs	Help contents.
Destination	Screen.
Pre-condition	The consumer must visit Hebron Municipality e-services website, and request to view help.
Post-condition	None.
Validation	None.

Function	<i>Viewing FAQ.</i>
Description	The user should be able to view the frequently asked questions about the system.
Inputs	Browse FAQ's links.
Source	Hebron Municipality database.
Outputs	FAQ's list.
Destination	Screen.
Pre-condition	The consumer must visit Hebron Municipality website, and request to view FAQ's.
Post-condition	None.
Validation	None.

<b>Function</b>	<i>Acquiring Comments and suggestions from consumer.</i>
<b>Description</b>	This function enables the consumers to send his comments and suggestions about real services.
<b>Inputs</b>	Consumer comments and suggestions.
<b>Source</b>	Consumer.
<b>Outputs</b>	Comments.
<b>Destination</b>	Hebron Municipality database records.
<b>Pre-condition</b>	Consumer should login with his account.
<b>Post-condition</b>	Add records to Hebron Municipality DB and send data to administrator with consumer No.
<b>Validation</b>	None.

### 3.3.2 Administration Functional Description

<b>Function</b>	<i>Administrator login.</i>
<b>Description</b>	This function will be the only method for administrator to login to Hebron Municipality database (locally).
<b>Inputs</b>	User Name, Password.
<b>Source</b>	Hebron Municipality database administrator.
<b>Outputs</b>	Login to his session.
<b>Destination</b>	Changes on administrator related account contents (Log files).
<b>Pre-condition</b>	Must be turn on the web server.
<b>Post-condition</b>	None.
<b>Validation</b>	Password should meet the existence roles in SQL server 2000 or operating system.

<b>Function</b>	<i>Administrator logout.</i>
<b>Description</b>	This function enables Administrator to end his session.
<b>Inputs</b>	None.
<b>Source</b>	Administrator.
<b>Outputs</b>	Session is ended.
<b>Destination</b>	Change during session are considered and saved.
<b>Pre-condition</b>	Administrator should be login to his session.
<b>Post-condition</b>	Changes are saved and session is closed.
<b>Validation</b>	None.

<b>Function</b>	<i>Viewing the sent Comments.</i>
<b>Description</b>	The administrator should be able to view comments that sent by consumer.
<b>Inputs</b>	None.
<b>Source</b>	Consumer.
<b>Outputs</b>	Comments list and details.
<b>Destination</b>	Screen.
<b>Pre-condition</b>	The administrator should have been logged in Hebron Municipality web server locally as administrator and select to view comments after the received alerts from consumers.
<b>Post-condition</b>	Write back to consumer comments.
<b>Validation</b>	None.

Function	<i>FAQ's Updating and maintaining.</i>
Description	The administrator should be able to add, delete and modify FAQ's.
Inputs	Questions and answers details.
Source	Administrator.
Outputs	FAQ list.
Destination	Hebron Municipality database.
Pre-condition	The administrator should have been logged in Hebron Municipality web server locally as administrator.
Post-condition	Publish the new changes, and activate it.
Validation	None.

Function	<i>Generate Bills (add or modify bills).</i>
Description	The administrator should be able generate bills, add or modify bills.
Inputs	Bill's information.
Source	Hebron Municipality database Administrator.
Outputs	New bills.
Destination	Hebron Municipality database records.
Pre-condition	The administrator should have been logged in Hebron Municipality web server locally as administrator.
Post-condition	Publish the new changes and activate it.
Validation	None.

### 3.4 System Contexts (Relations with other systems)

The figure below shows an overview of the system and its relationships with other systems.

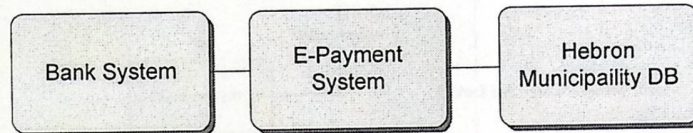


Figure [3.1] System Contexts Diagram

### 3.5 Information description

During this section, an over all description for the system information is discussed, as well a description for the database requirements and database model is available

3.5.1 System Data Flow Diagram

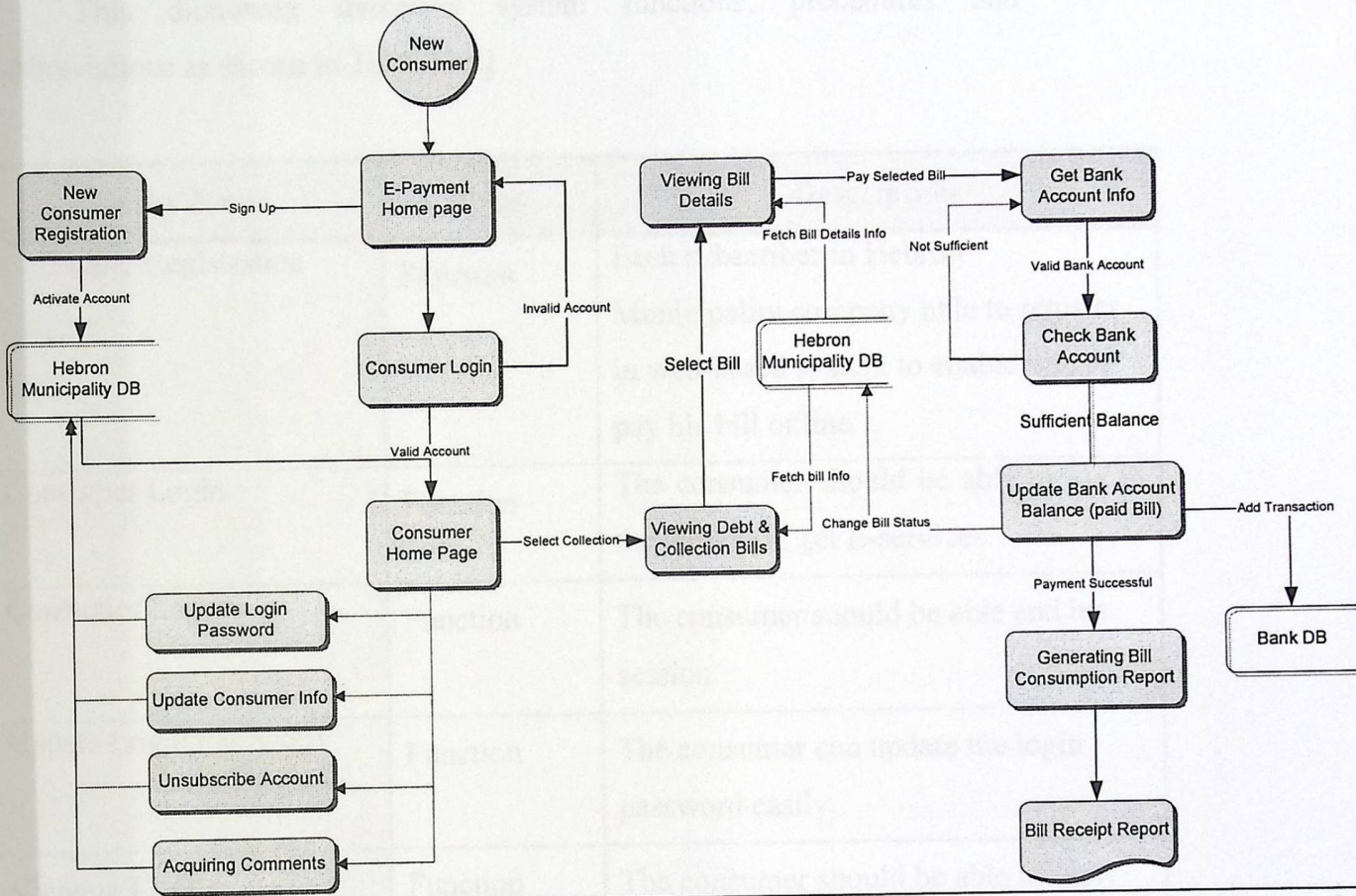


Figure [3.2] System Data Flow Diagram

## 3.5.2 Data Dictionary

This dictionary describes system functions, procedures and abbreviations as shown in Table [3.2]

Entity Name	Type	Description
Consumer Registration	Function	Each subscriber in Hebron Municipality company able to register in web-based system to enable him to pay his bill online.
Consumer Login	Function	The consumer should be able to log in the system to get E-services.
Consumer Logout	Function	The consumer should be able end his session.
Update Login Password	Function	The consumer can update the login password easily.
Viewing FAQ's	Function	The consumer should be able to view the FAQ's easily to understand the unclear subjects by viewing the most problems and its solutions.
Viewing Help	Function	The consumer should get help.
Sending comments	Function	The consumers should be able to send their comments and feedbacks.
Viewing bills status	Function	The consumer should be able to view last 12 bills status paid or not.
Consumer Bill Viewing	Function	The consumers should be able to view, save and print their bills.
Viewing consumption	Function	The consumers should be able to view

report.		their consumption reports after the payment process is finished.
Reporting comments and suggestions	Function	The administrator should be able to view and generate report comments that sent by consumers.
FAQ maintenance	Function	The administrator should be able to add, modify and delete FAQ's (Frequently Asked Questions).
Bills generations	Function	The administrator should be able generate new bills.
Online payment using bank account.	Function	The consumer should be able to pay the bill online using bank account.
Generating Balancing Reports.	Function	The bank account companies should generate balancing transactions to Hebron Municipality database records.
Browsing The New Transaction.	Function	Hebron Municipality should browse the new transaction and update its database.
FAQ	Abbreviation	Frequently Asked Questions.
E-payment	Abbreviation	Electronic payment.
WWW	Abbreviation	World Wide Web.
SID	Abbreviation	Session Identification.
Info	Abbreviation	Information.
Bank account	Object	This is the object that represents a customer's Bank account document that has a unique number which identifies it on the bank to enable him make financial transactions that affects

		his account without the need to make these transactions manually.
Validation	String	This string is the code that is used to ensure that any user of the system has typed the right data types and formats into certain fields.
Consumer	String	A string that represents a person who browses the Hebron Municipality web site.
Services	String	This string identifies a certain entity in the system such as bill detail or bill period.

Table [3.2] Data Dictionary

### 3.5.3 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. (Design and Testing phases are covered in chapter 4 and 6 respectively).

According to the user interface, the System will support only User (Consumer) interface.

#### **Consumer Interface:**

Hebron Municipality web site improved two languages (Arabic and English) to make it easy for both local and foreign consumers.

#### Input:

The interface between consumer and Hebron Municipality system is controlled to ensure that he entered parameters of the same type as they identified later during system development by using validation techniques available, the consumer will be permitted to input just valid parameters type within an allowed range of values to increase the consistency in database.

#### Output:

The output for consumer must give them a sense of what is happening in each process, as they choose a certain bill from bills list, also support the consumer with confirmation and reports to ensure that the process is completely done. This report shows all information and details for each bill to enable the consumer to view the content of the bill.

#### GUI

The graphical user interface must be easy and attractive for users (consumers); it should provide some what guidance marks that guide consumer to the next step to be done.

Also the GUI provides the users (consumers) with Help and FAQ's to enable the user to understand the unclear subjects.

### 3.5.4 Database Requirements

The following tables and fields are for Hebron Municipality System database:

#### 3.5.4.1 Hebron Municipality Databases Tables

##### A- Consumer Information Tables

Consumer No: Each consumer has a unique Consumer No.

Consumer ID: Each consumer has a unique Consumer ID.

Password: A unique characters for each consumer, these characters should encrypt in some encryption methods to increase the level of security.

E-mail: Each consumer has a unique E-mail.

Consumer Name: Contain the information about the full name of each consumer that includes First Name, Last Name, and Middle Name.

Address No: the number of area that consumer lives in.

Free: Boolean No that determine consumer whether if he has free 8 cups or not.

##### B- Consumer Question Table

Question No: A unique no for each question.

Question phrase: The question statement that is selected by consumer.

##### C- Consumer Answer

Question No: A unique no for each question.

Consumer No: A unique number for each registered consumer on the Hebron Municipality web site.

Answer: The secret answer that is selected by consumer.

**D- Area Table**

Address No.: a unique number for each area.

Address Name: a unique name for each area.

**E- Comment and Suggestion Table**

Consumer No.: A unique number for each registered consumer on the Hebron Municipality web site.

Comment No.: A unique number for each comment.

Comments: Set of character less than 60 characters it may be comment or suggestion.

Date: the date of sending the comments or suggestion

**F- Bill Details Table.**

Bill No.: The sequence number of the generated bill, each bill has a unique number.

Consumer No.: A unique number for each registered consumer on the Hebron Municipality web site.

Issue Date: The date that the bill issued at.

Due Date: The date that Hebron Municipality allowed the consumer to paid at.

Status No.: A flag that determine weather the bill is paid or not.

Payment Date: The date that the consumer paid the bill at.

Counter No.: the number of counter as meters cubes.

Previous Reading: the number of previous counter.

Current Reading: the reading of the current counter.

Counter Installation Date: the date of starting work of counter.

Old Total Balance: amount of unpaid bills in previous years.

Bill Value: The end value of bill.

Net Work Maintenance: constant amount every month for maintaining.

Counter Cost: constant amount every month for maintaining.

Price: the price of each cup.

### 3.5.4.2 Bank Database Dictionary

#### A- Bank Account Table

Account No: Each consumer has a unique Account No.

Name: The name of owner of the account.

Account Total Balance: the value of money in the consumer account.

Password: A unique characters for each consumer, these characters should encrypt in some encryption methods to increase the level of security.

#### B- bank Transaction Table

Transaction No: Each consumer has a unique No that is generated by sequence.

Account No: Each consumer has a unique Account No.

Payment Amount: amount of money that is paid by consumer.

Payment Date: The date of the payment.

Beneficiary Account No: account no that received money (Hebron Municipality).

4.1 Introduction

The system design will be discussed in this chapter, but the topics that are covered in

Chapter

4

• Database Design

• Test Plan

# System Design

The Content:

- 4.1 Introductions.
- 4.2 Functional Design.
- 4.3 Input/Output Design.
- 4.4 Database Design.
- 4.5 Test Plan.

### 4.1 Introduction

The system design will describes in this chapter, and the topics that are covered in this chapter:

- Functional Design.
- Input/output Design.
- Database Design.
- Test Plan.

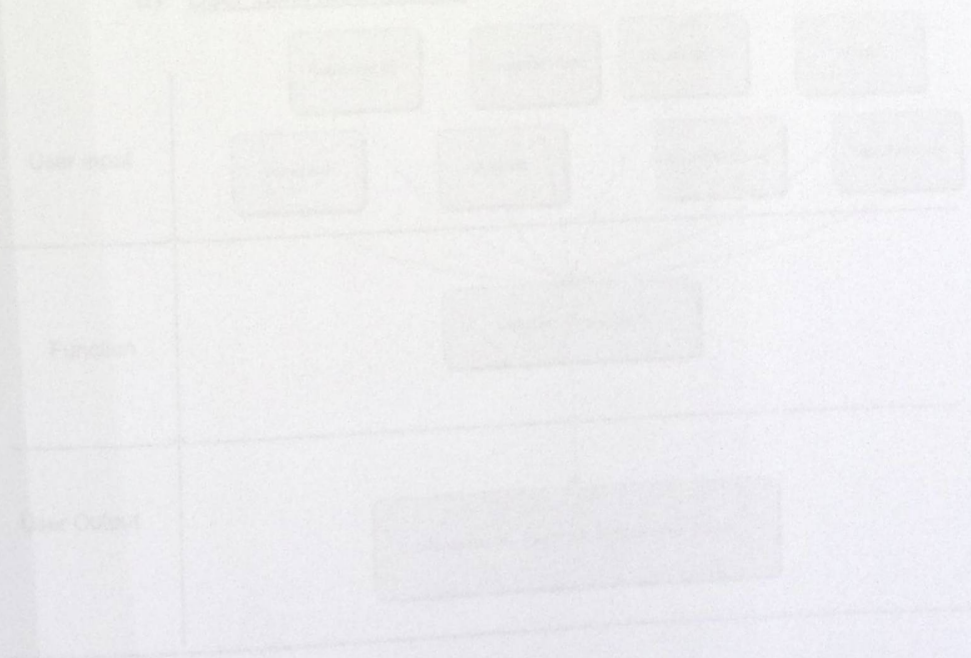


Figure [4.1] Consumer Registration User interface design.

## 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- Consumer Registration

- a) Description: Each subscriber in Hebron Municipality able to register in web-based system to enable him to pay his bill online.
- b) Interface:
- Inputs: Consumer No, Consumer ID, Password, Secret question, Secret answer and E-mail.
  - Outputs: Consumer No, Consumer Name and consumer Email.
- c) Constraints:
- All inputs must be filled correctly.
  - The consumer No must equal Hebron municipality database record.
  - The password must be at least 6 characters.
  - Password doesn't include special character.
  - Consumer No duplication also Not allowed.
- d) User interface design:

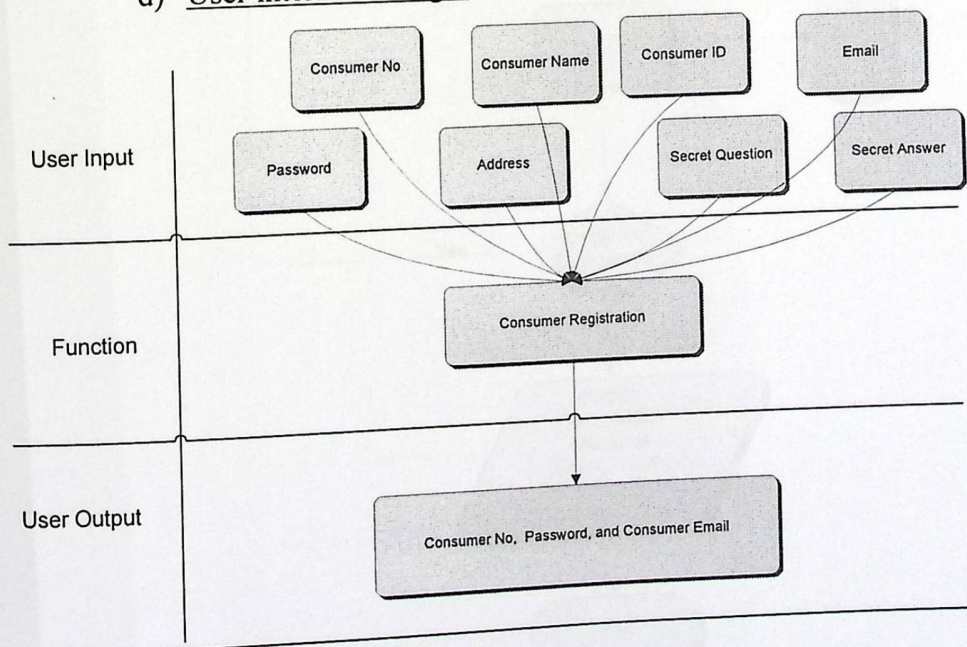


Figure [4.1] Consumer Registrations User interface design

e) Flowchart:

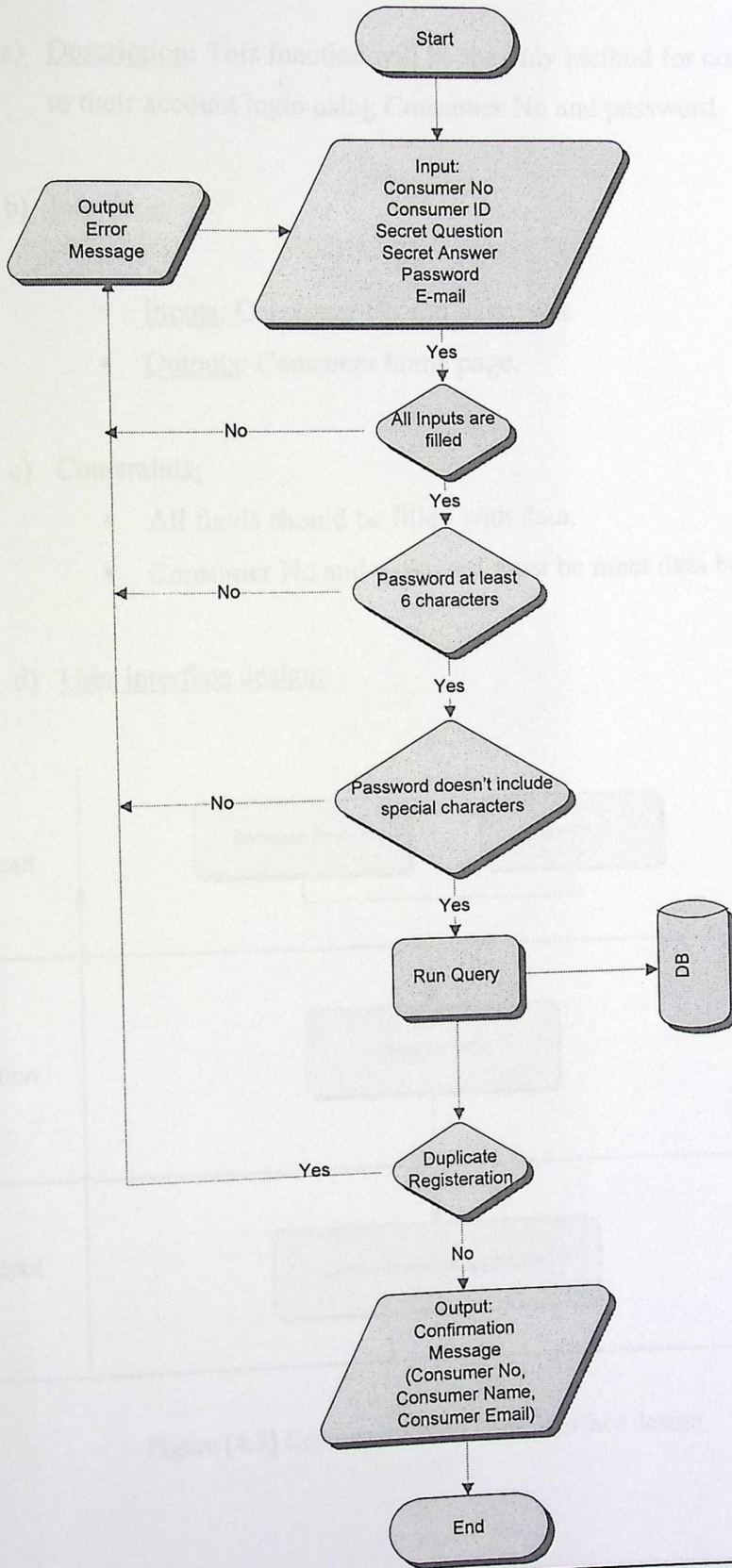


Figure [4.2] Consumer Registrations Flowchart

## 2- Consumer Login

- a) Description: This function will be the only method for consumer to login to their account login using Consumer No and password.
- b) Interface:
- Inputs: Consumer No and password.
  - Outputs: Consumer home page.
- c) Constraints:
- All fields should be filled with data.
  - Consumer No and password must be meet data base record.
- d) User interface design:

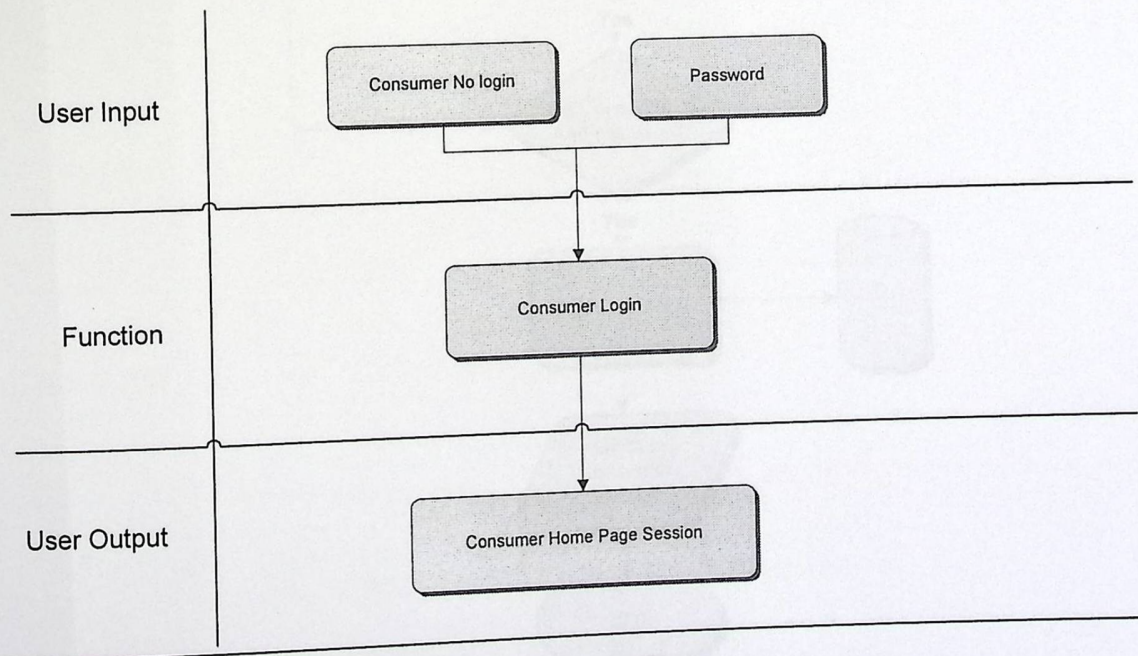
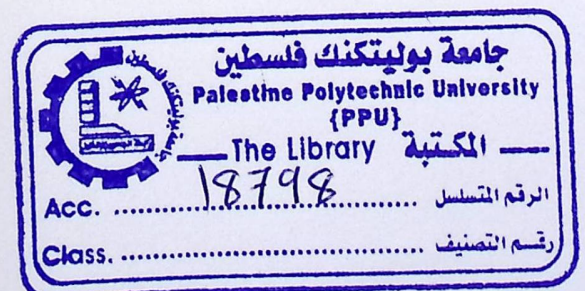


Figure [4.3] Consumer Login User interface design



e) Flowchart:

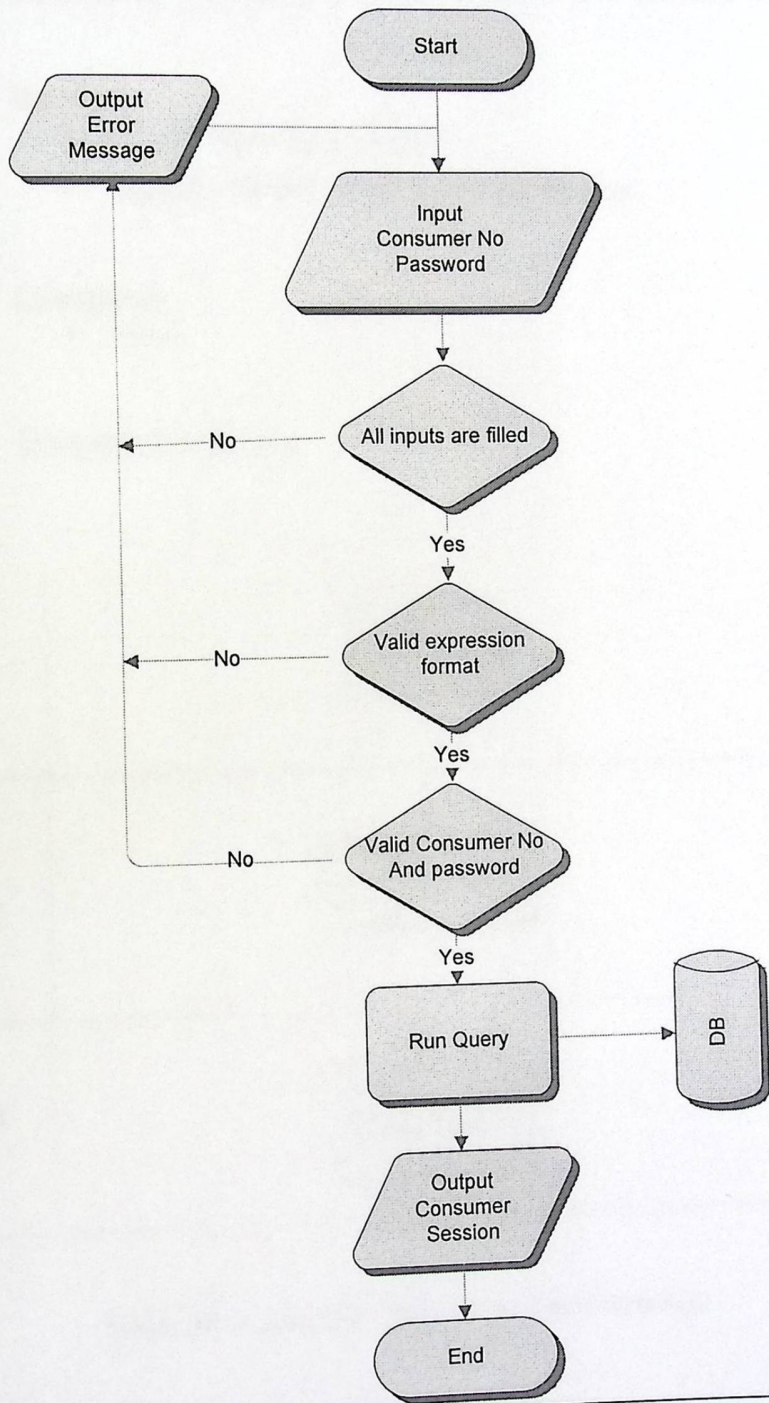


Figure [4.4] Consumer Login Flowchart

## 3- Consumer Logout

- a) Description: This function enables consumers to end their sessions.
- b) Interface:
- Inputs: Selecting to logout.
  - Outputs: Hebron Municipality Home page.
- c) Constraints:
- None.
- d) User interface design:

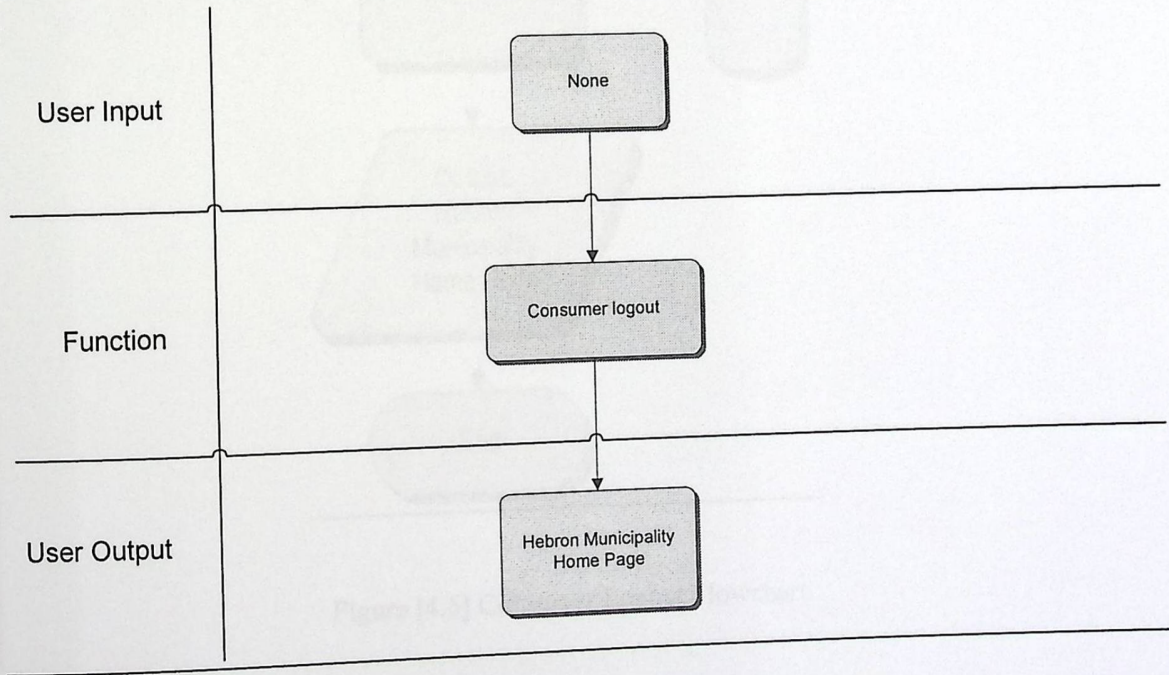


Figure [4.5] Consumer Logout User interface design

e) Flowchart:

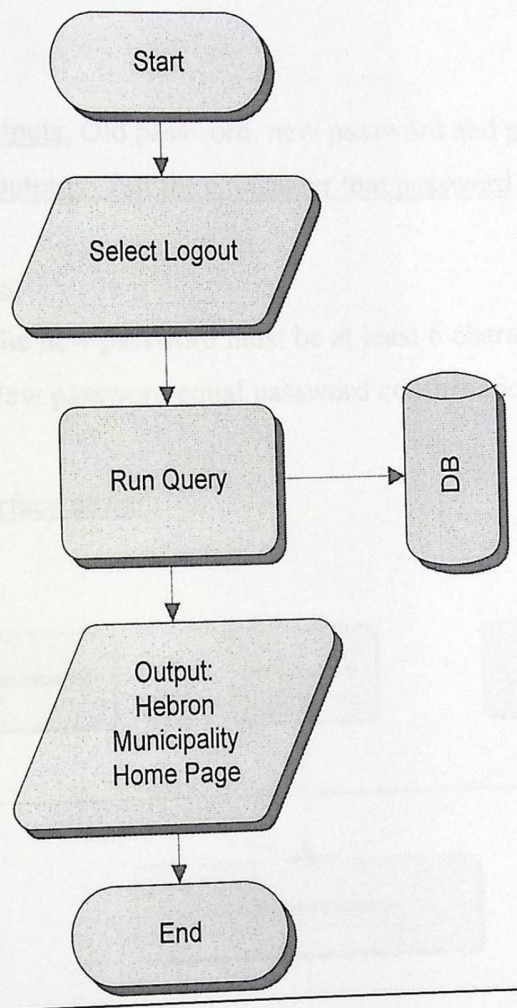


Figure [4.6] Consumer Logout Flowchart

## 4- Change Login Password

- a) Description: The registered consumer should be able to change login password.
- b) Interface:
- Inputs: Old password, new password and password confirmation.
  - Outputs: Tell the Consumer that password has been changed.
- c) Constraints:
- The new password must be at least 6 characters long.
  - New password equal password confirmation.
- d) User interface design:

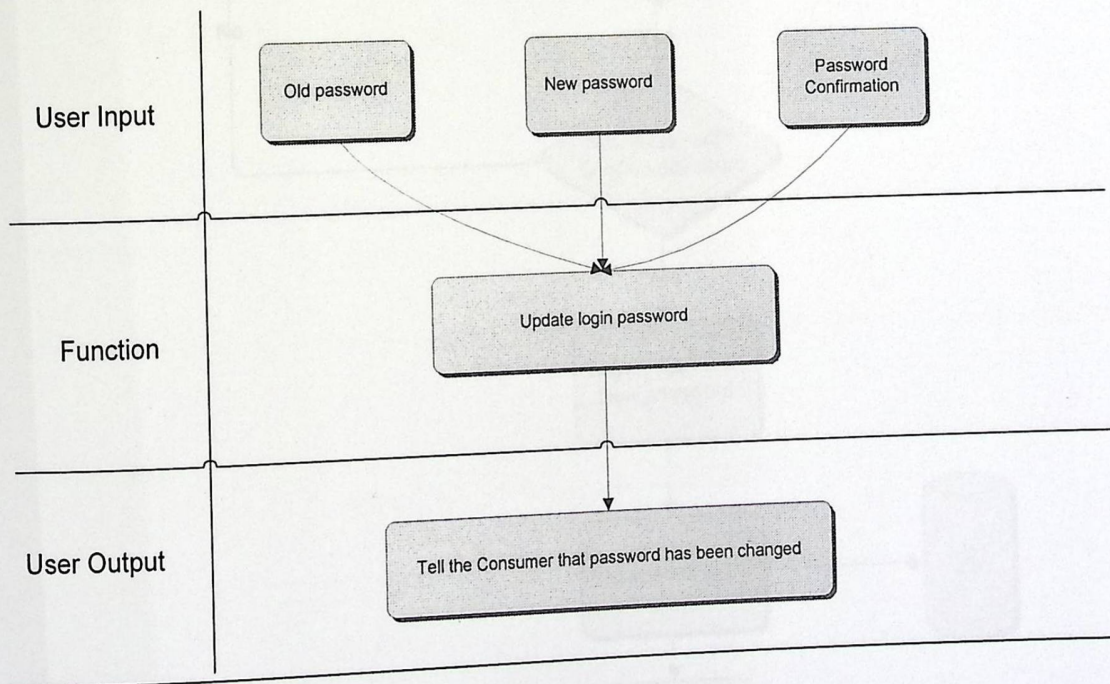


Figure [4.7] Change Login Password User interface design

e) Flowchart:

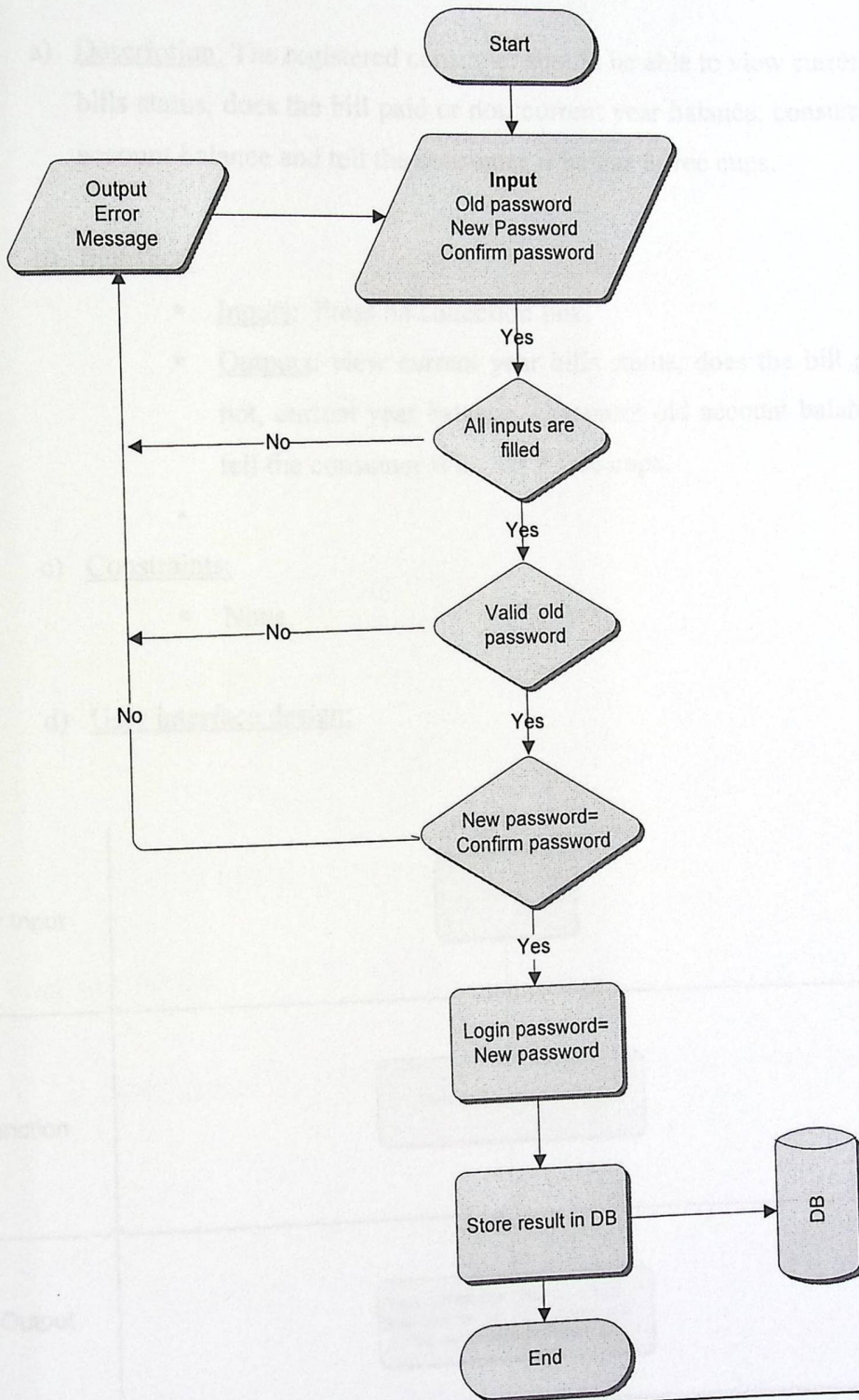
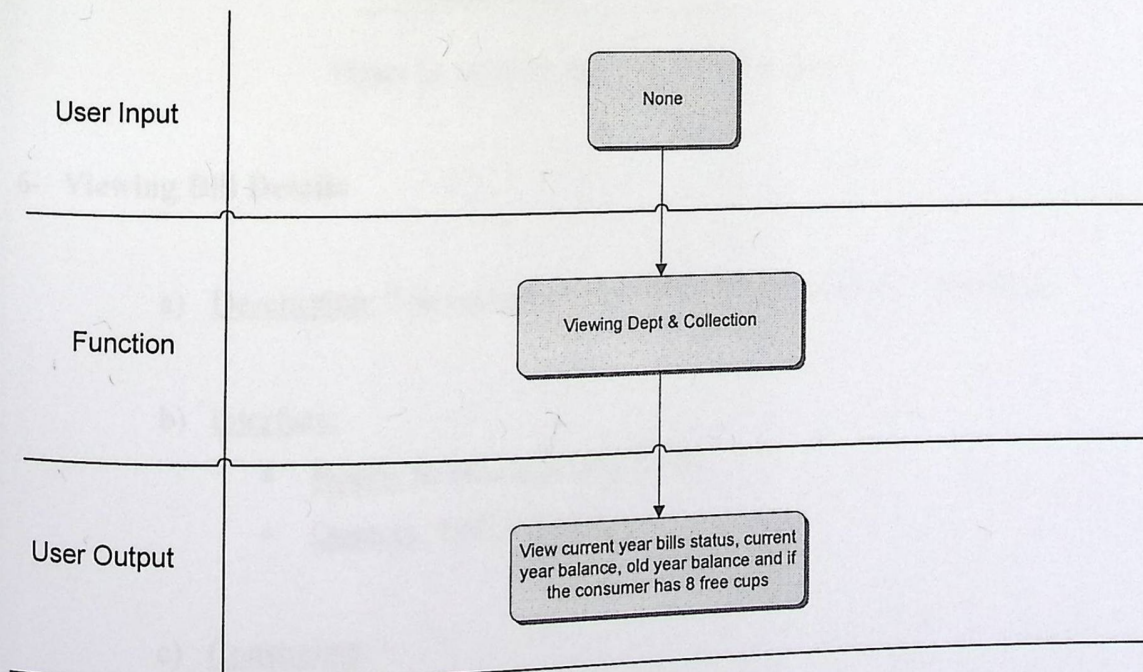


Figure [4.8] Change Login Password Flowchart

## 5- Viewing Dept and Collection

- a) Description: The registered consumer should be able to view current year bills status, does the bill paid or not, current year balance, consumer old account balance and tell the consumer if he has 8 free cups.
- b) Interface:
- Inputs: Press on collection link.
  - Outputs: view current year bills status, does the bill paid or not, current year balance, consumer old account balance and tell the consumer if he has 8 free cups.
  -
- c) Constraints:
- None.
- d) User interface design:



e) Figure [4.9] Debt and Collection User interface design

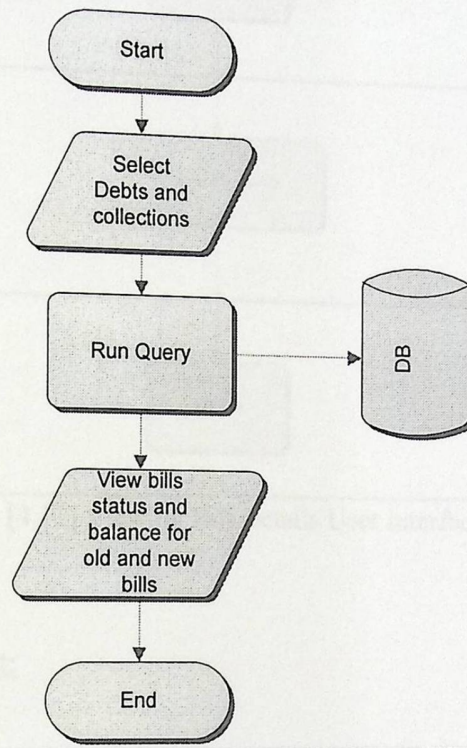
f) Flowchart:

Figure [4.10] Debt and Collection Flowchart

## 6- Viewing Bill Details

- a) Description: The registered consumer should be able to view bills.
- b) Interface:
- Inputs: Selecting details E-bill.
  - Outputs: Bill is displayed.
- c) Constraints:
- None.

d) User interface design:

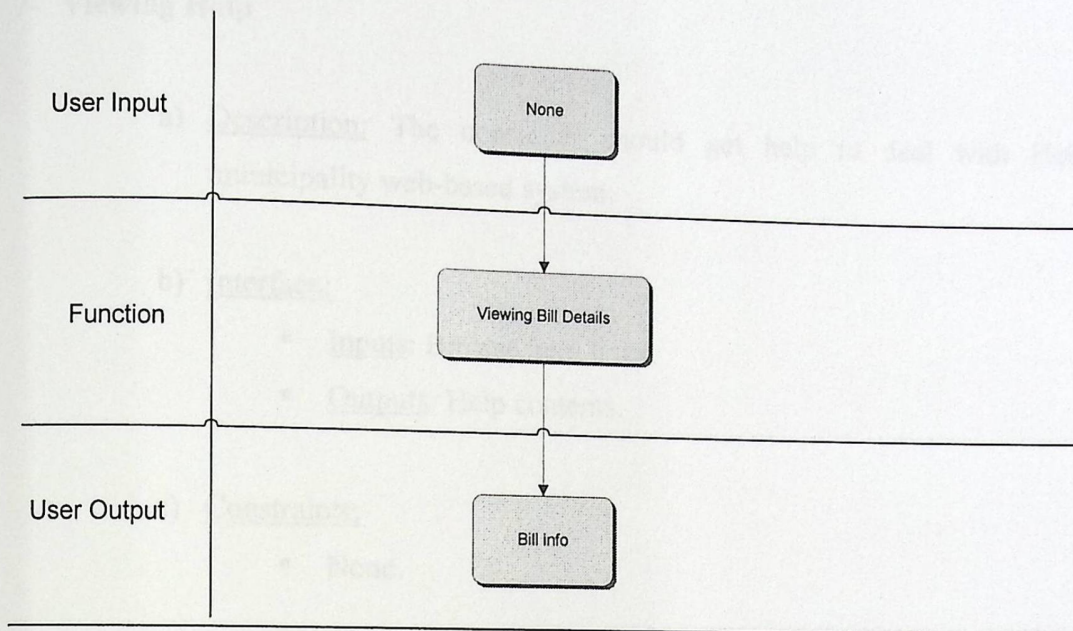


Figure [4.11] Viewing Bill Details User interface design

e) Flowchart:

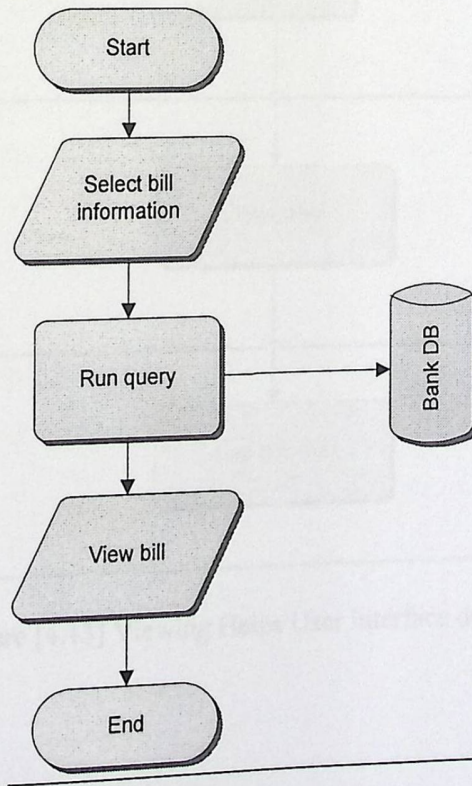


Figure [4.12] Viewing Bill Details Flowchart

## 7- Viewing Help

- a) Description: The consumer should get help to deal with Hebron municipality web-based system.
- b) Interface:
- Inputs: Browse help links.
  - Outputs: Help contents.
- c) Constraints:
- None.
- d) User interface design:

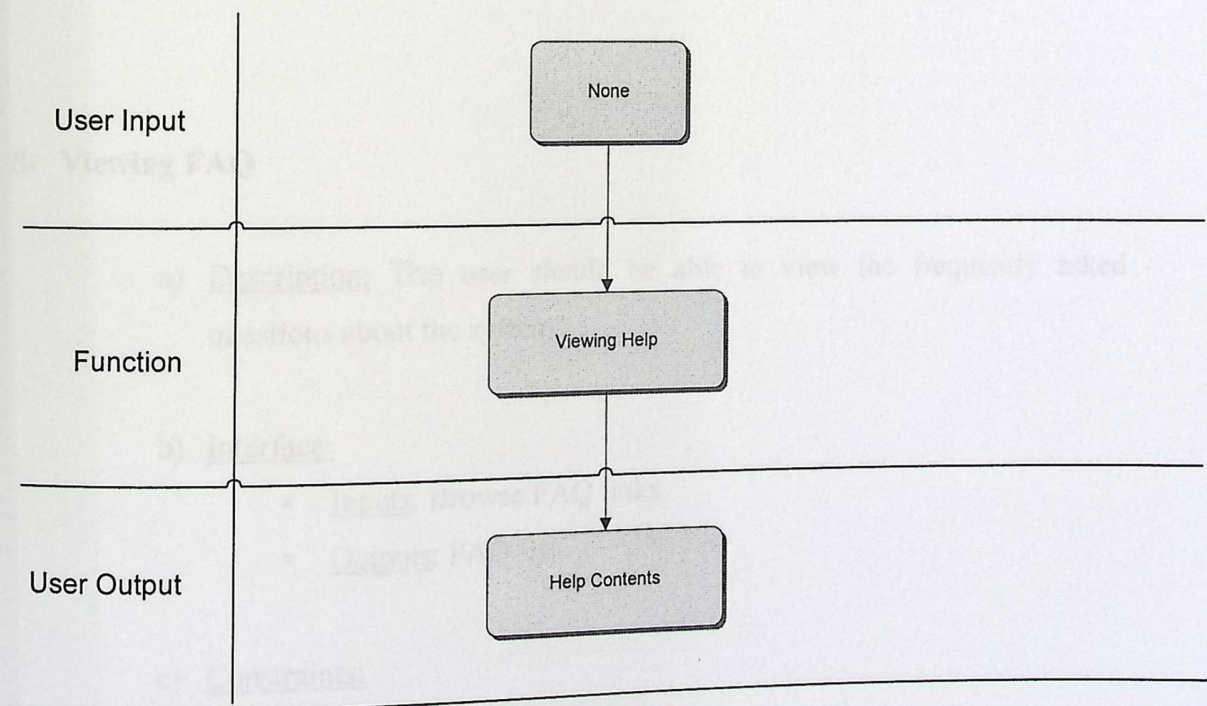


Figure [4.13] Viewing Helps User interface design

e) Flowchart:

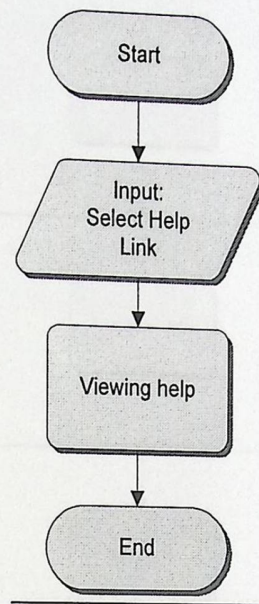


Figure [4.14] Viewing Helps Flowchart

## 8- Viewing FAQ

- a) Description: The user should be able to view the frequently asked questions about the system.
- b) Interface:
- Inputs: Browse FAQ links.
  - Outputs: FAQ list.
- c) Constraints:
- None.

d) User interface design:

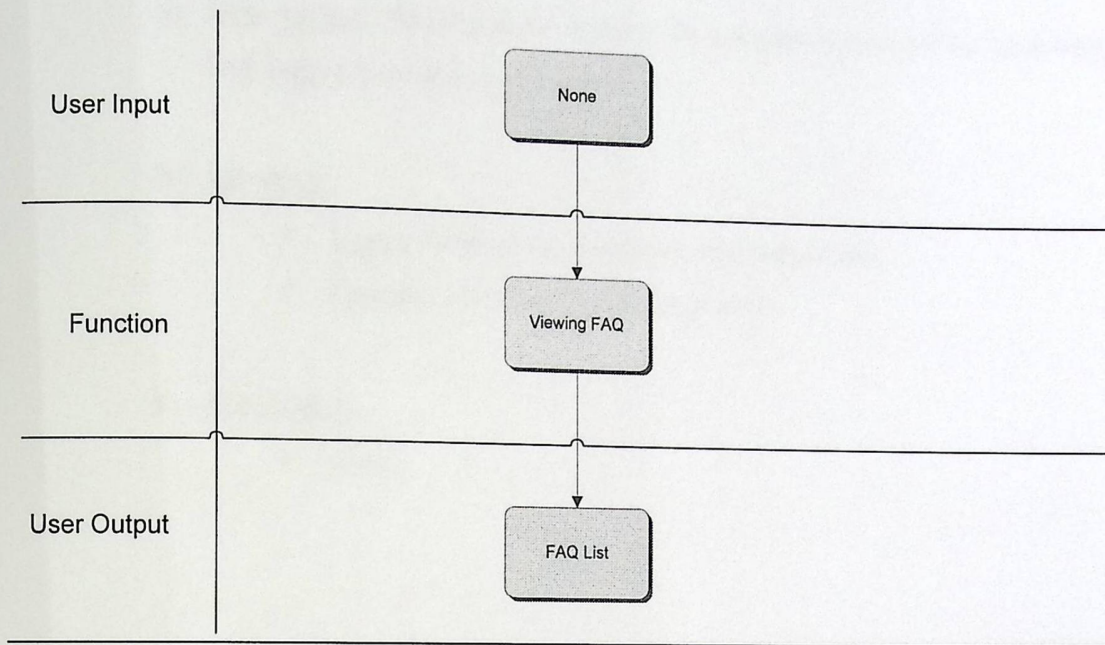


Figure [4.15] Viewing FAQ User interface design

f) Flowchart:

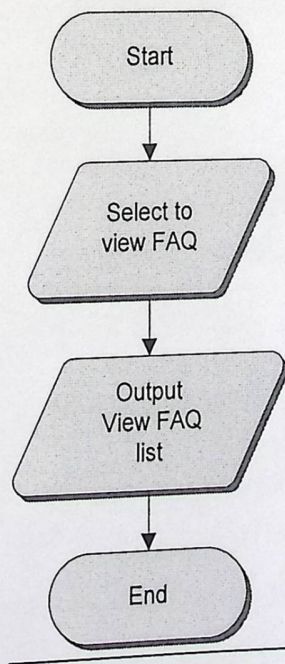


Figure [4.16] Viewing FAQ Flowchart

### 9- Acquiring Comments and Suggestion from Consumer

- a) Description: This function enables the consumers to send his comments and suggestions about real services.
- b) Interface:
- Inputs: Consumer comments and suggestion.
  - Outputs: Comments and suggestions.
- c) Constraints:
- None.
- d) User interface design:

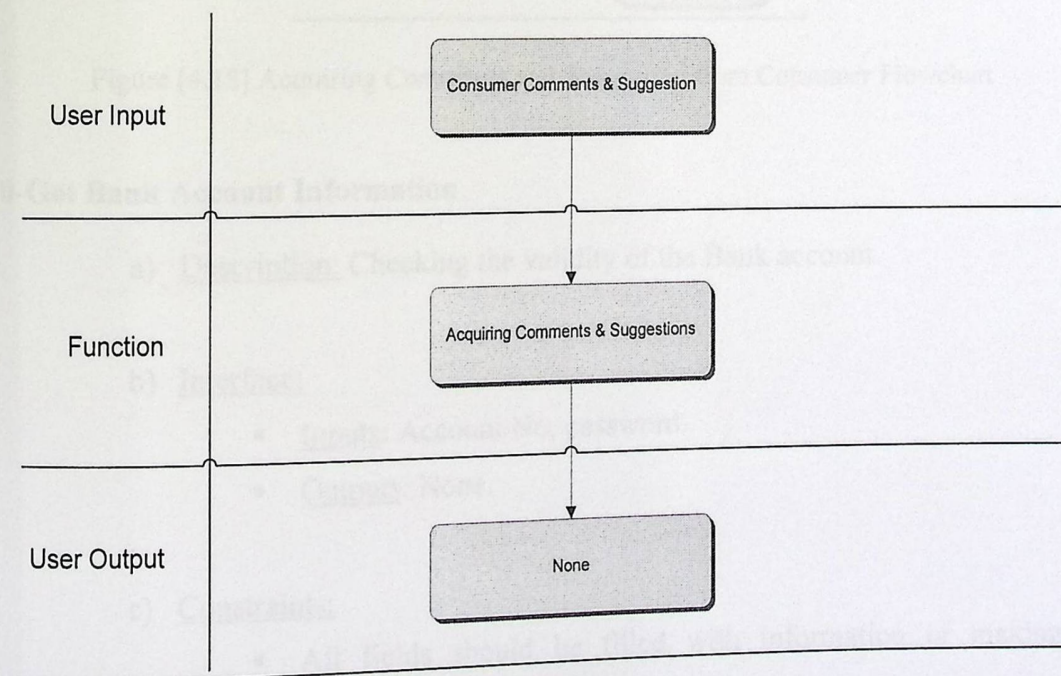


Figure [4.17] Acquiring Comments and Suggestion from Consumer User interface design

e) Flowchart:

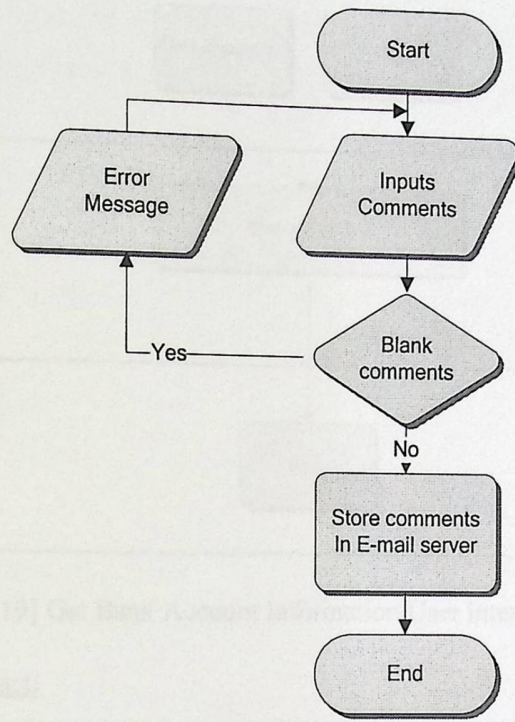


Figure [4.18] Acquiring Comments and Suggestion from Consumer Flowchart

### 10-Get Bank Account Information

- a) Description: Checking the validity of the Bank account.
- b) Interface:
  - Inputs: Account No, password.
  - Outputs: None.
- c) Constraints:
  - All fields should be filled with information or making a selection, (change the default setting).
  - The Bank Account should be entered as digit numbers.

d) User interface design:

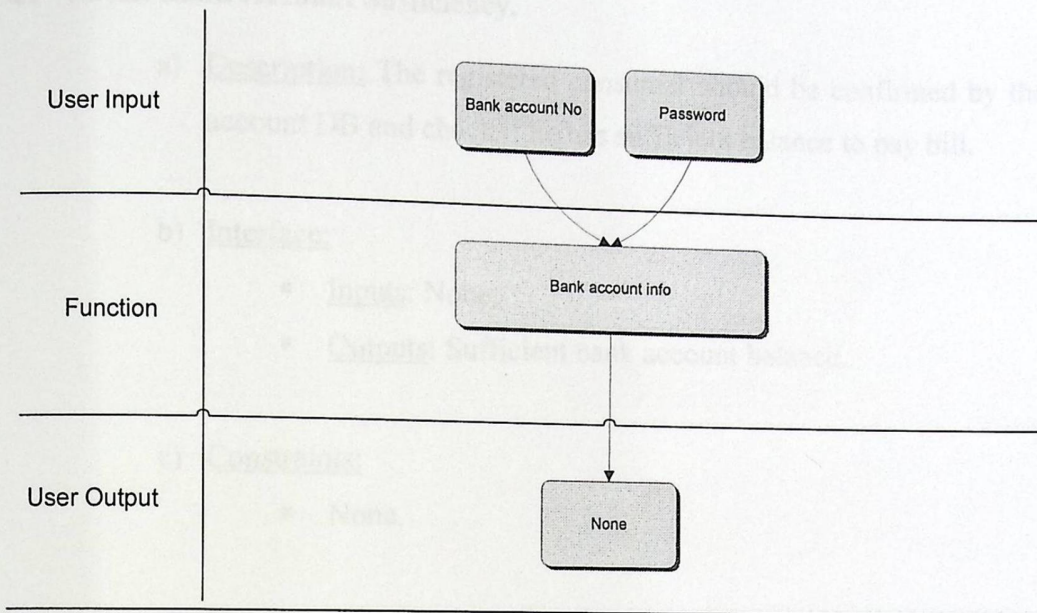


Figure [4.19] Get Bank Account Information User interface design

e) Flowchart:

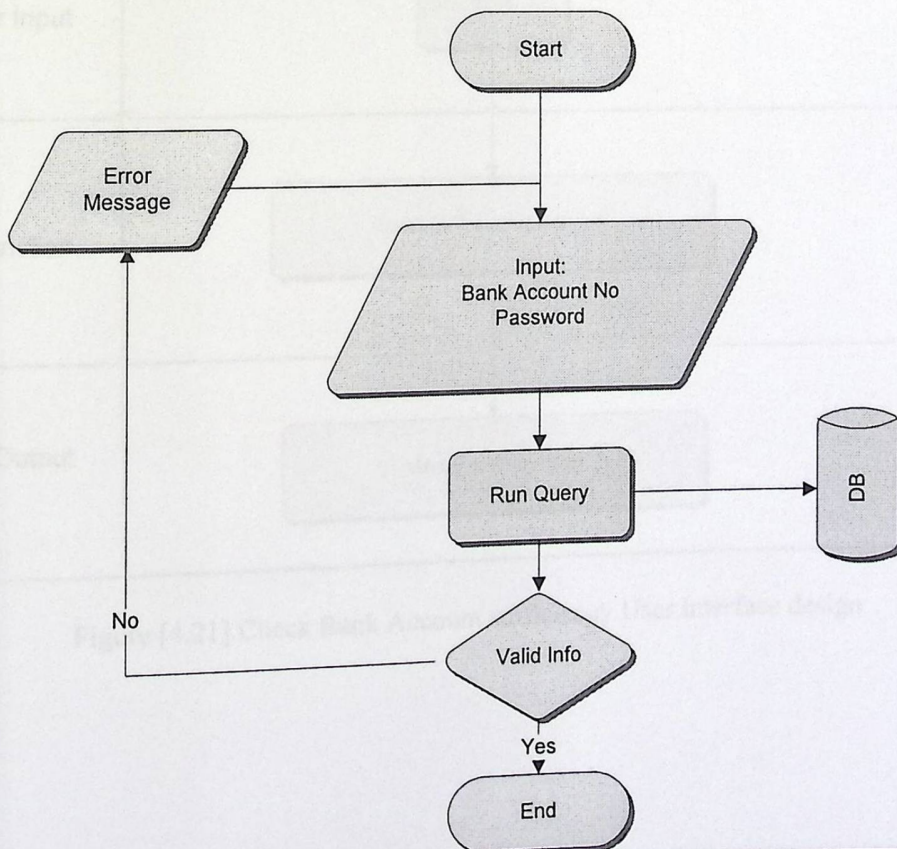


Figure [4.20] Get Bank Account Information Flowchart

### 11- Check Bank Account Sufficiency.

- a) Description: The registered consumer should be confirmed by the bank account DB and check if he has sufficient balance to pay bill.
- b) Interface:
- Inputs: None.
  - Outputs: Sufficient bank account balance.
- c) Constraints:
- None.
- d) User interface design:

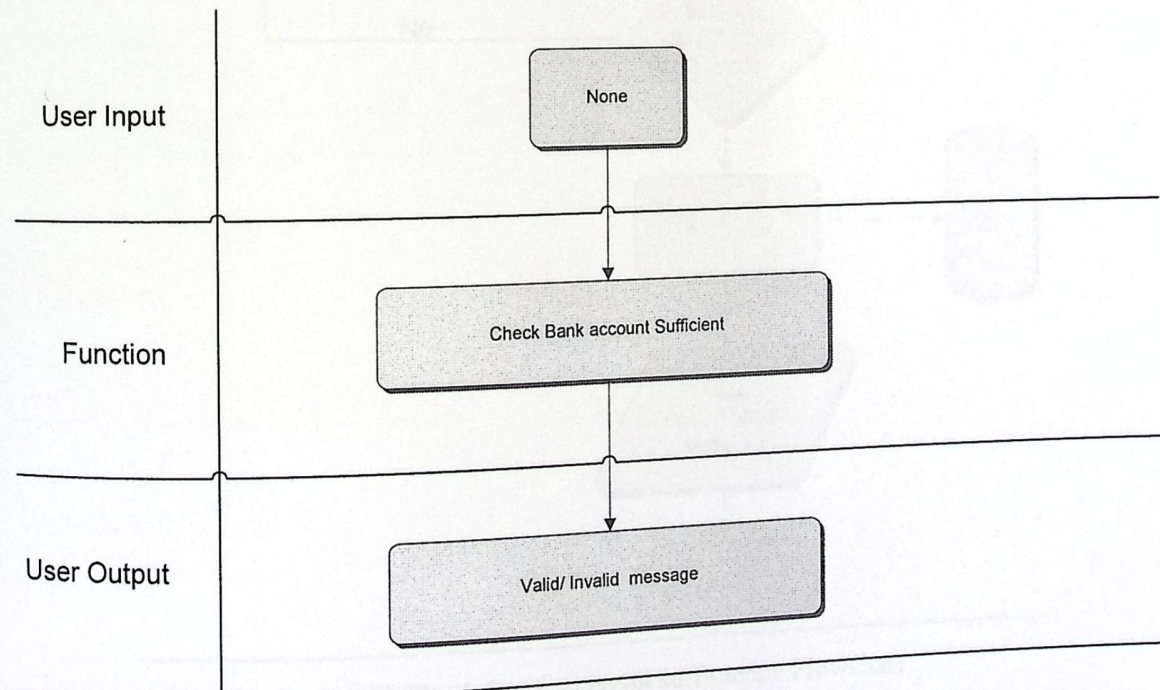


Figure [4.21] Check Bank Account sufficiency User interface design

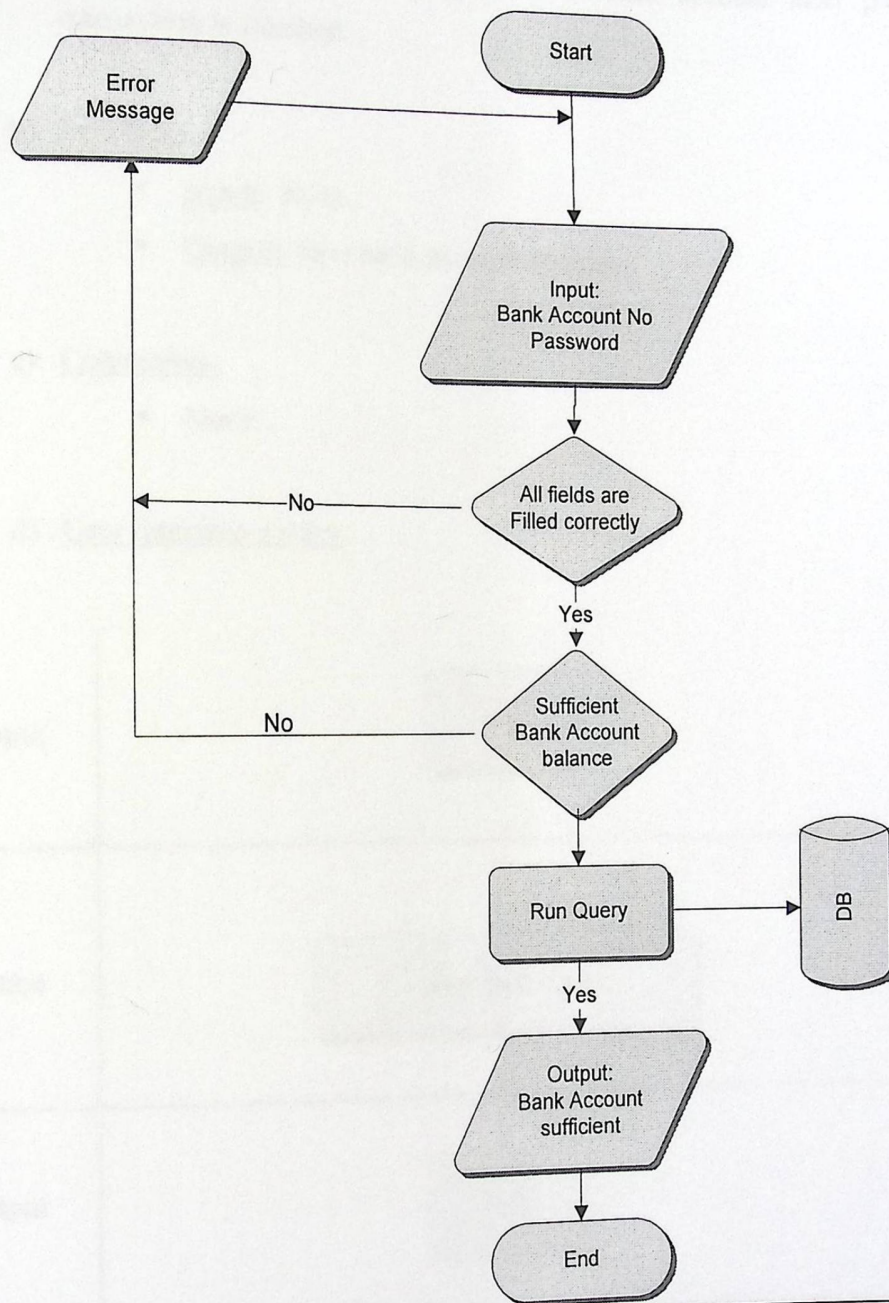
e) Flowchart:

Figure [4.22] Check Bank Account sufficiency Flowchart

## 12- Update Bank Account.

- a) Description: This function updates the bank account after payment transaction is finished.
- b) Interface:
- Inputs: None.
  - Outputs: New bank account balance.
- c) Constraints:
- None.
- d) User interface design:

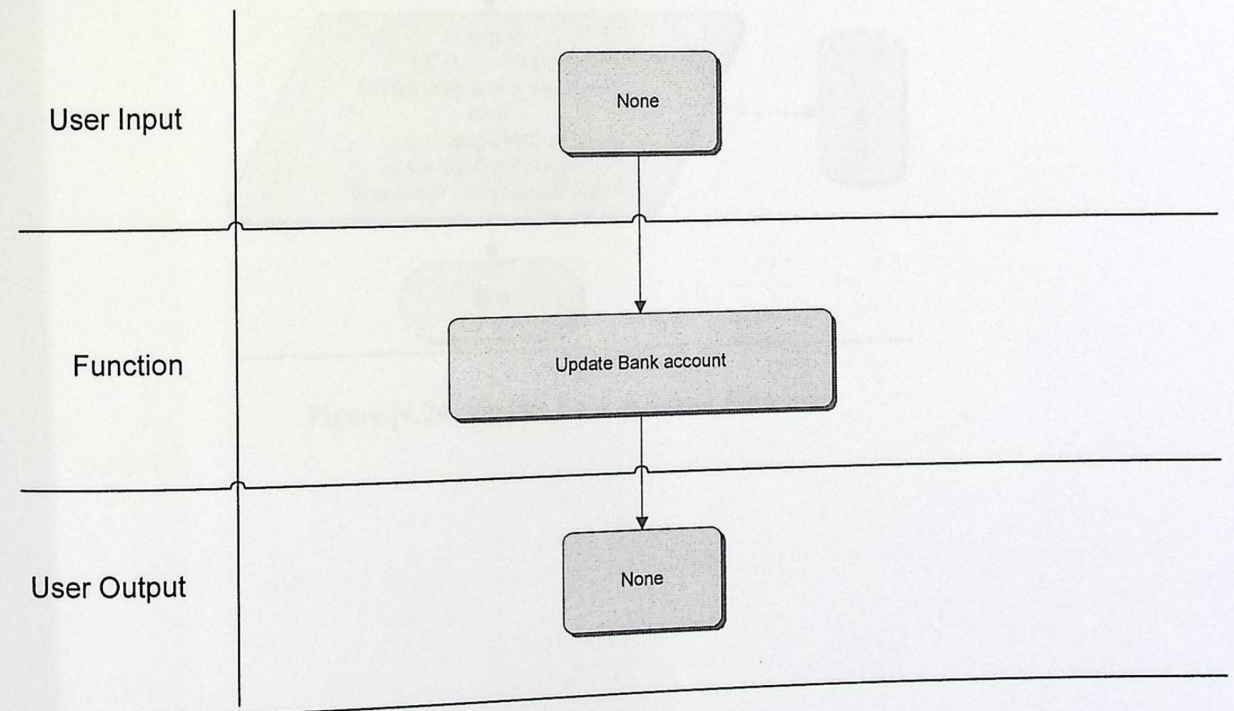


Figure [4.23] Update Bank Account User interface design

## e) Flowchart:

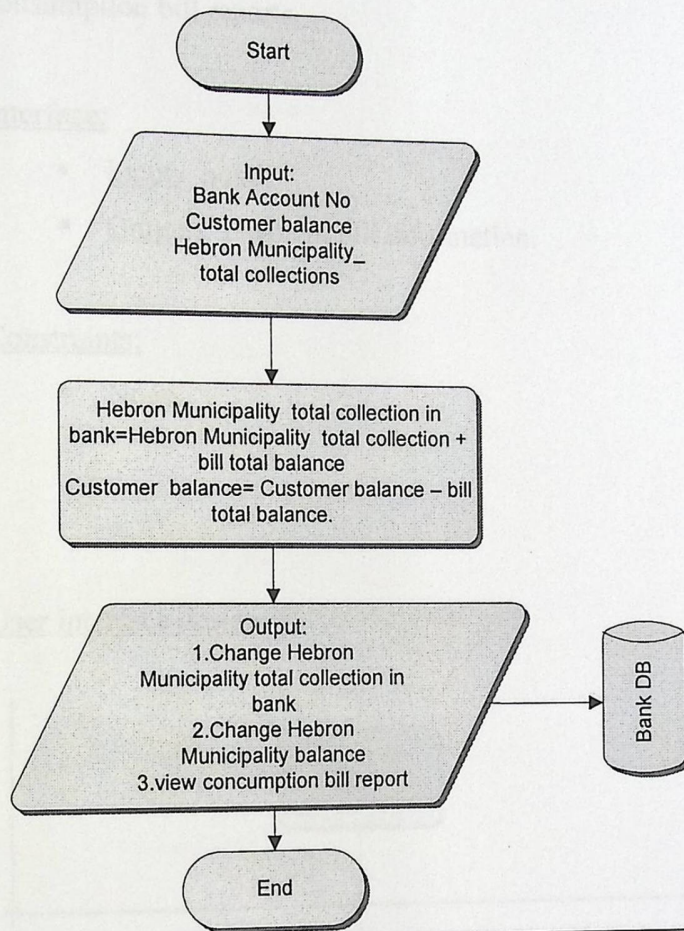


Figure [4.24] Update Bank Account Flowchart

### 13-Generate Bill Consumption Report

- a) Description: The registered consumer should be able to view their consumption bill reports.
- b) Interface:
- Inputs: none.
  - Outputs: Payment bill information.
- c) Constraints:
- None.
- d) User interface design:

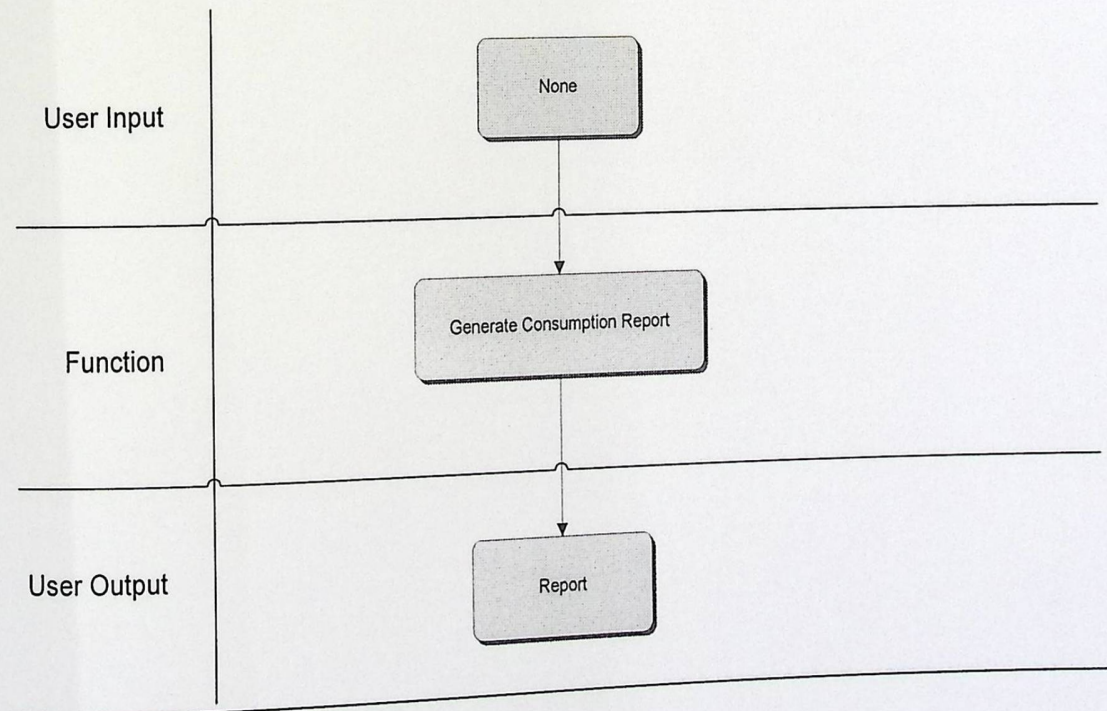


Figure [4.25] Generate Bill Consumption Report User interface design

e) Flowchart

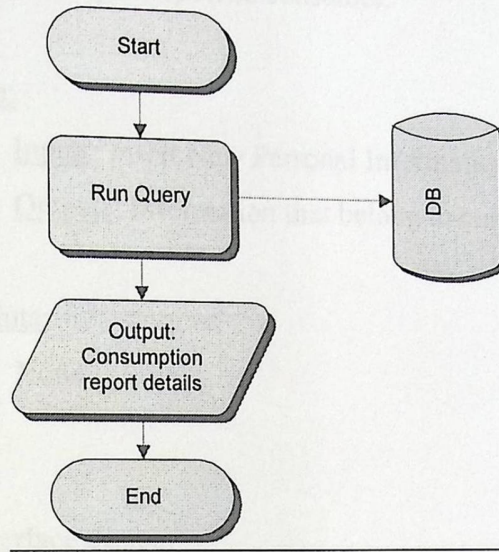


Figure [4.26] Generate Bill Consumption Report Flowchart

## 14- Update Consumer Information

- a) Description: This function views or updates the personal information (E-mail) that belongs to specific consumer.
- b) Interface:
- Inputs: Insert New Personal Information (E-mail).
  - Outputs: Information that belong to consumer.
- c) Constraints:
- None.
- d) User interface design:

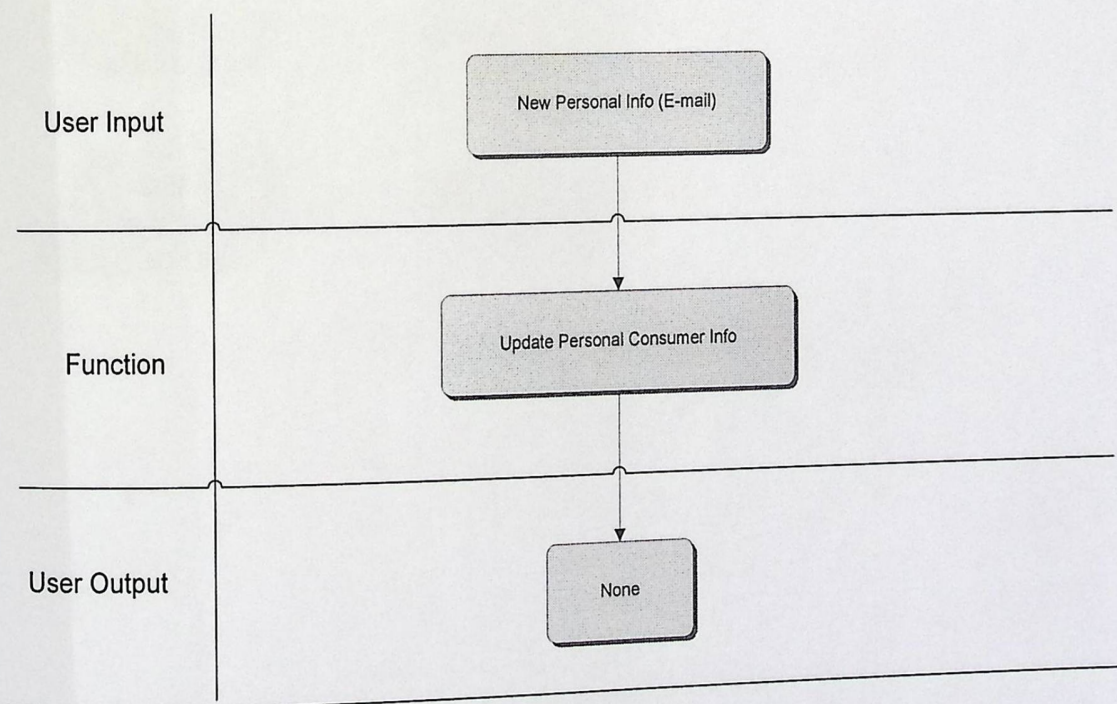


Figure [4.27] Update Consumer Information User interface design

e) Flowchart

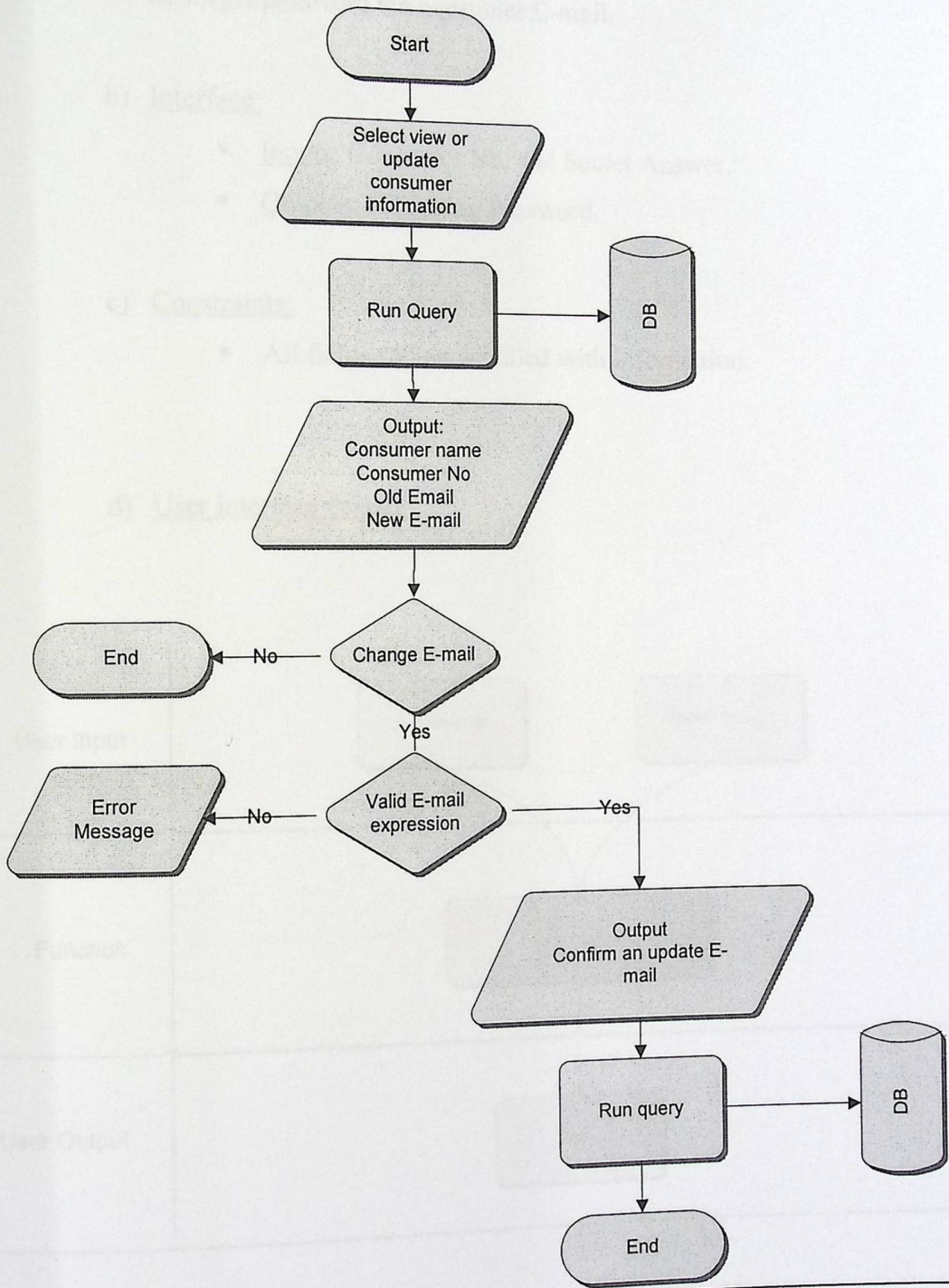


Figure [4.28] Update Consumer Information Flowchart

## 15- Forget Password

- a) Description: This function enables the consumer to retrieve his account if he forget password via consumer E-mail.
- b) Interface:
- Inputs: Consumer No, and Secret Answer.
  - Outputs: Forgetting Password.
- c) Constraints:
- All fields should be filled with information.
- d) User interface design:

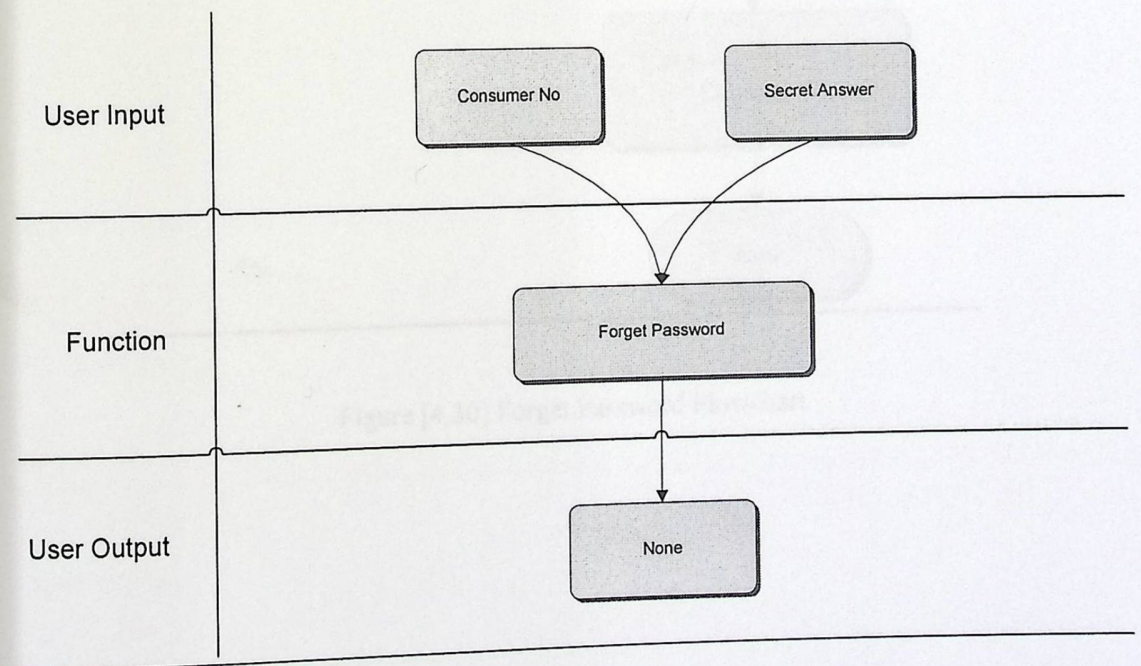


Figure [4.29] Forget Password User interface design

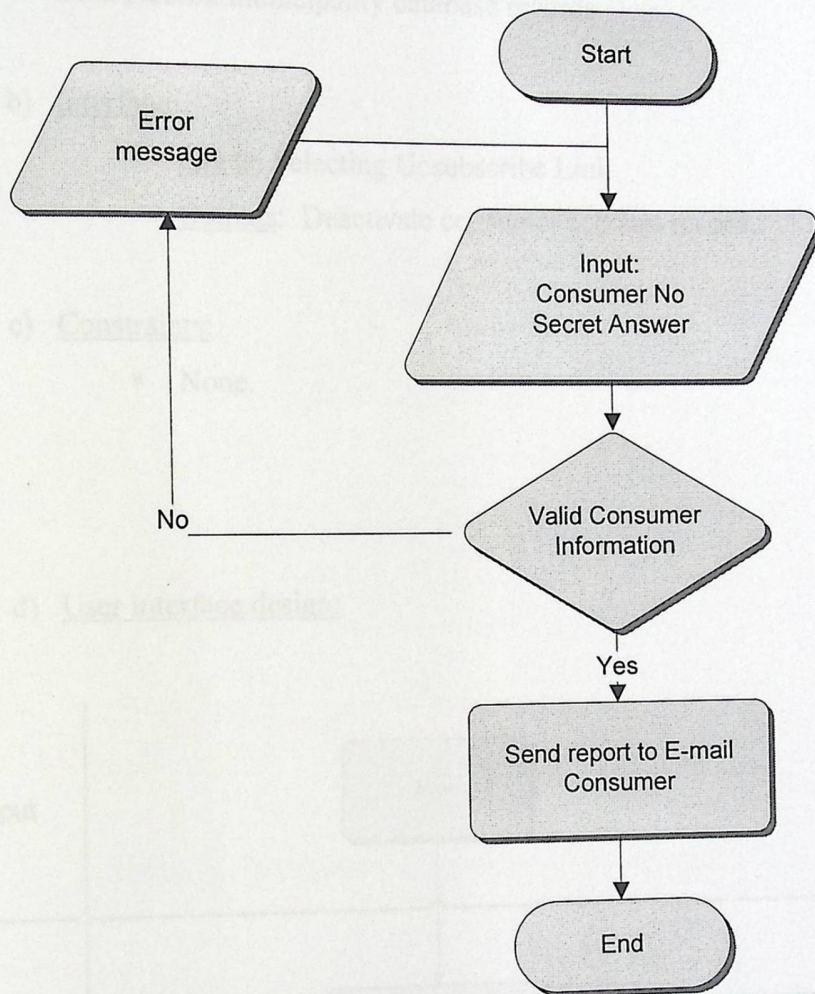
e) Flowchart

Figure [4.30] Forget Password Flowchart

## 16- Unsubscribe Consumer Account

- a) Description: This function allows the consumer to deactivate his account from Hebron municipality database records.
- b) Interface:
- Inputs: Selecting Unsubscribe Link.
  - Outputs: Deactivate consumer account record.
- c) Constraints:
- None.
- d) User interface design:

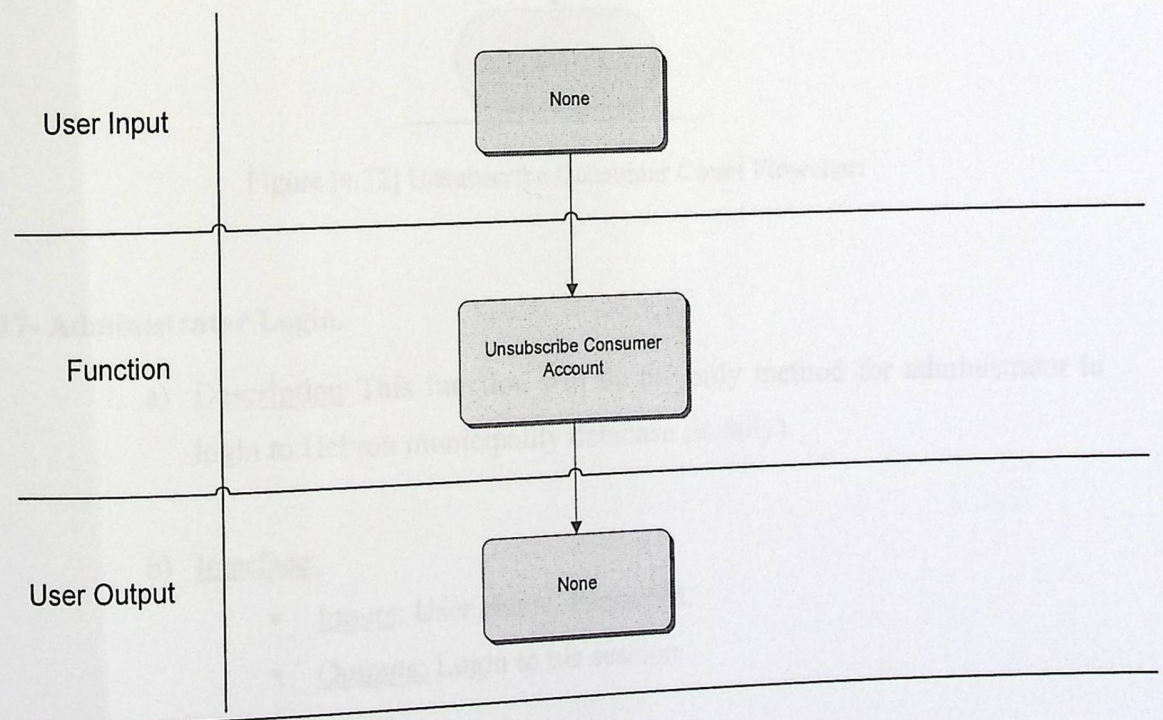


Figure [4.31] Unsubscribe Consumer Count User interface design

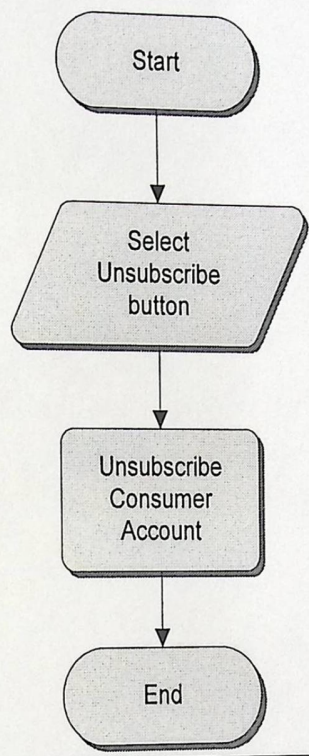
e) Flowchart

Figure [4.32] Unsubscribe Consumer Count Flowchart

**17- Administrator Login.**

- a) Description This function will be the only method for administrator to login to Hebron municipality database (locally).
- b) Interface:
- Inputs: User Name, Password.
  - Outputs: Login to his session.
- c) Constraints:
- Password should meet the existence roles in SQL server 2000 or operating system.

d) User interface design:

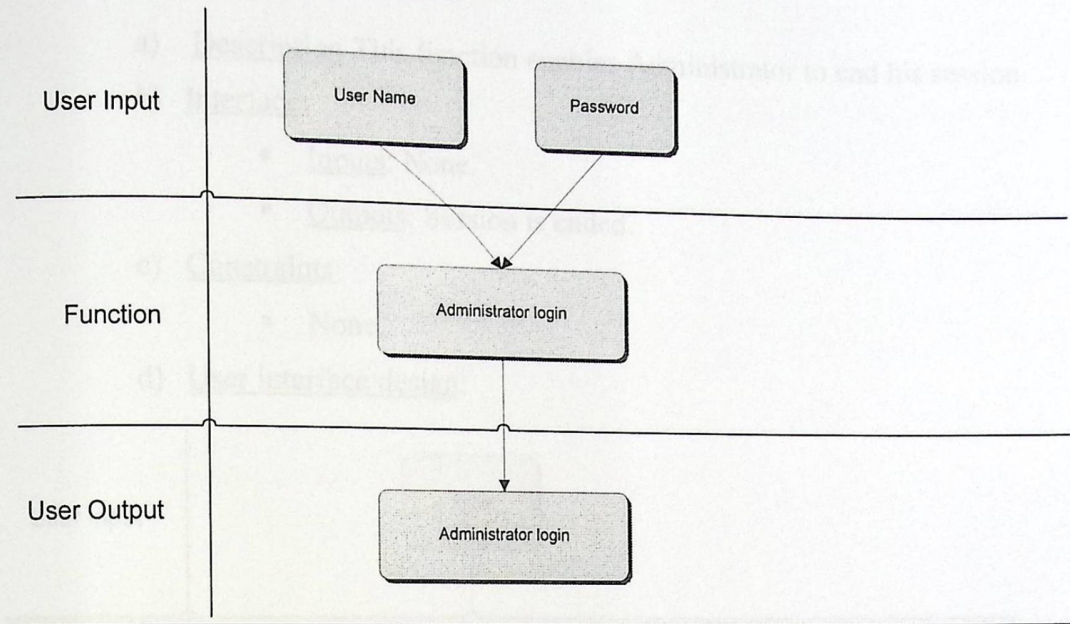


Figure [4.33] Administrator Login User interface design

e) Flowchart

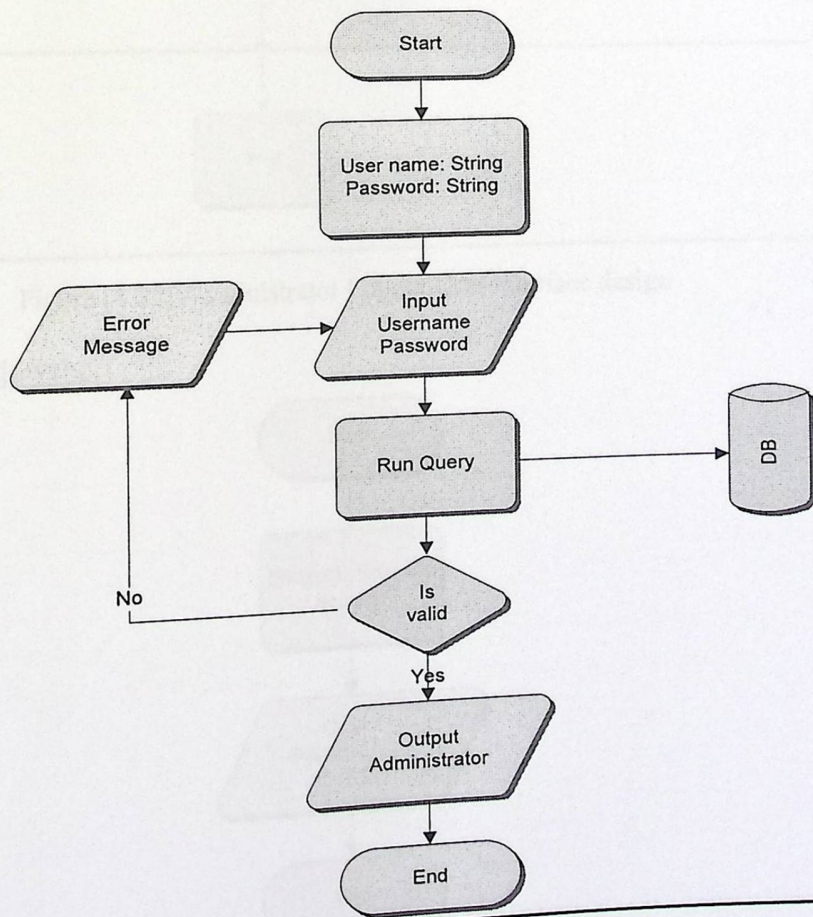


Figure [4.34] Administrator Login Flowchart

18- Administrator logout.

- a) Description This function enables Administrator to end his session.
- b) Interface:
  - Inputs: None.
  - Outputs: Session is ended.
- c) Constraints:
  - None.
- d) User interface design:

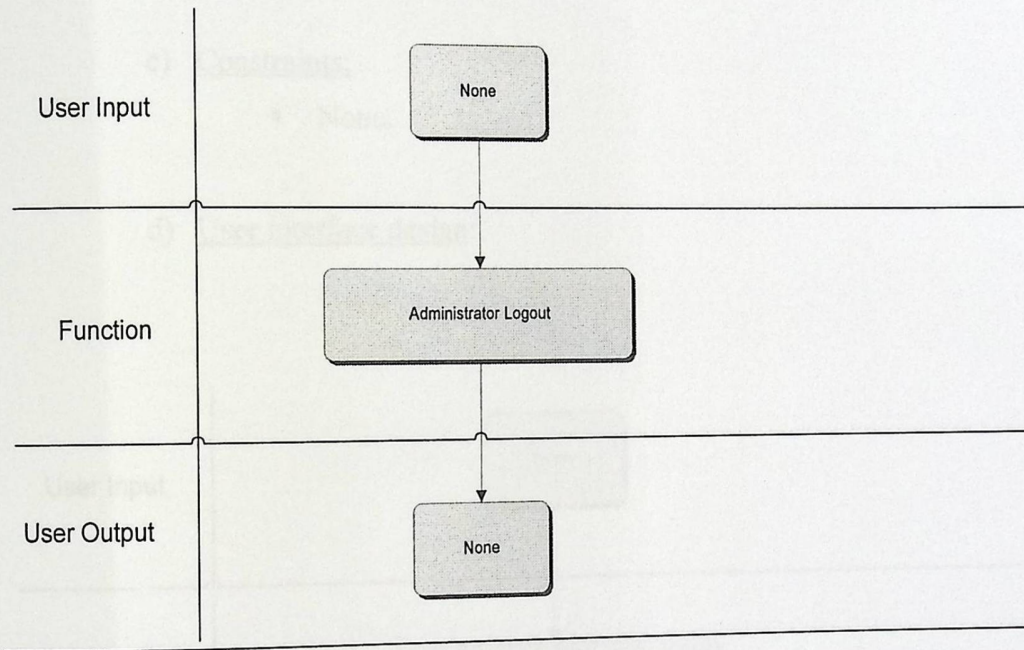


Figure [4.35] Administrator Logout User interface design

e) Flowchart

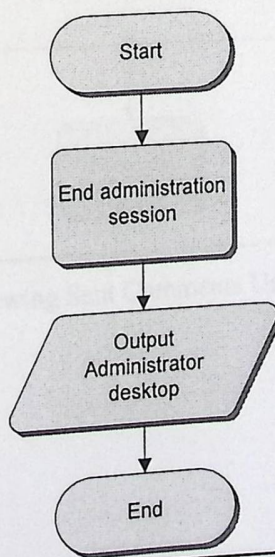


Figure [4.36] Administrator Logout Flowchart

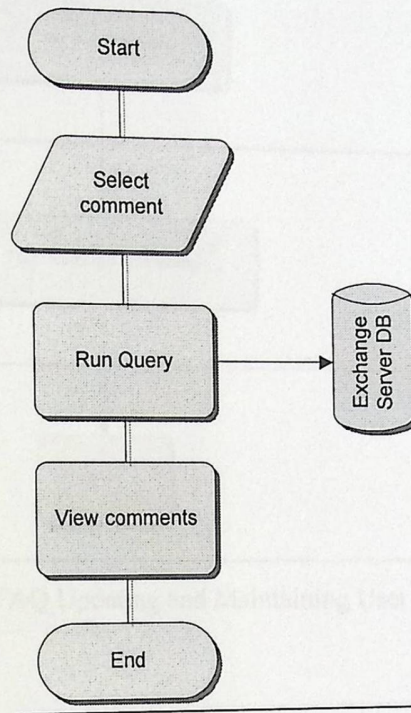
e) Flowchart

Figure [4.38] Viewing Sent Comments Flowchart

**20- FAQ Updating and Maintaining.**

- a) Description The administrator should be able to add, delete and modify FAQ.
- b) Interface:
  - Inputs: Questions and answers details.
  - Outputs FAQ list.
- c) Constraints:
  - None.

d) User interface design:

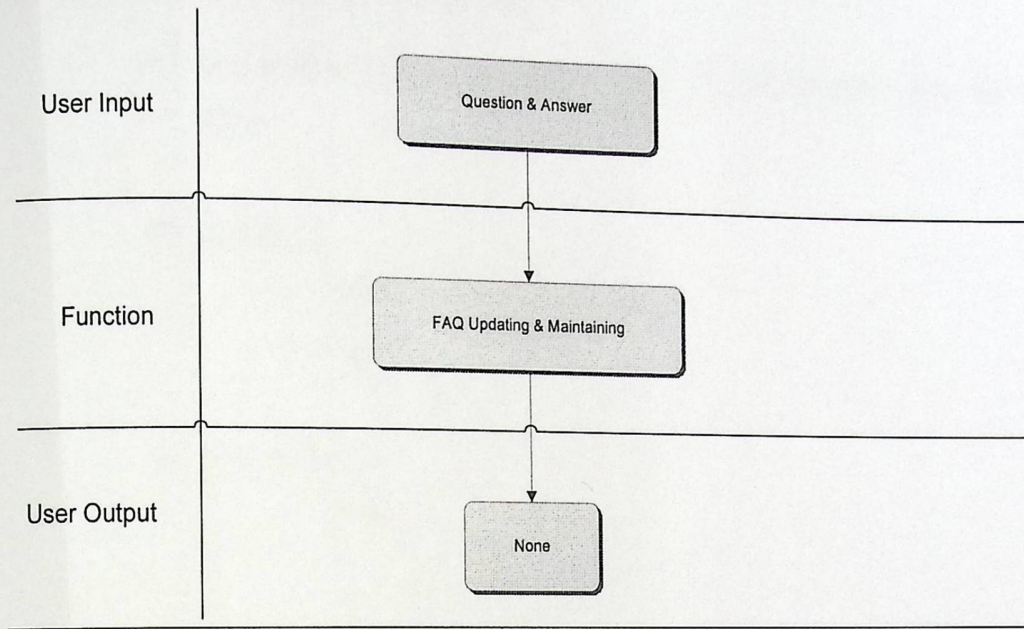


Figure [4.39] FAQ Updating and Maintaining User interface design

e) Flowchart

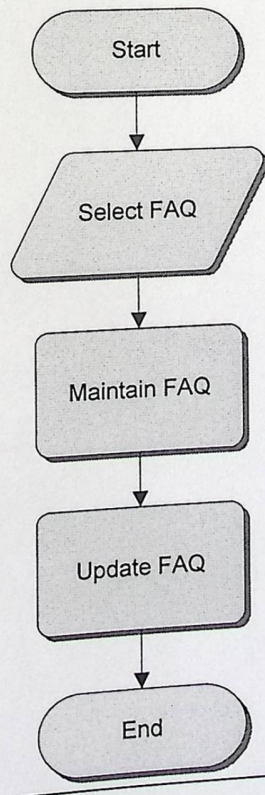


Figure [4.40] FAQ Updating and Maintaining Flowchart

**21- Generate Bills (add or modify bills).**

- a) Description: The administrator should be able generate bills, add modify bills.
- b) Interface:
- Inputs: Bill's information.
  - Outputs: New bills.
- c) Constraints:
- None.
- d) User interface design:

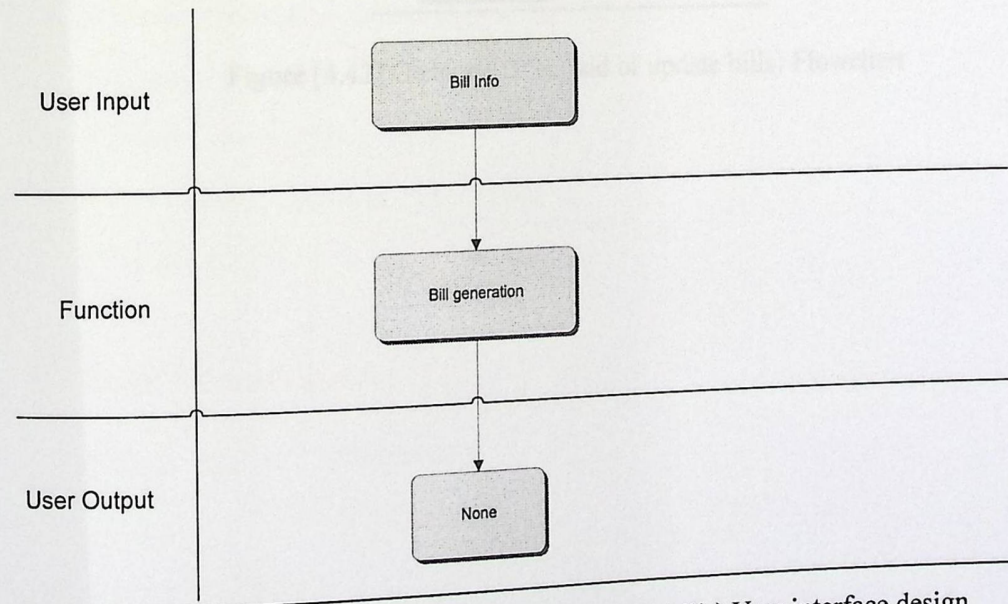


Figure [4.41] Generate Bills (add or update bills) User interface design

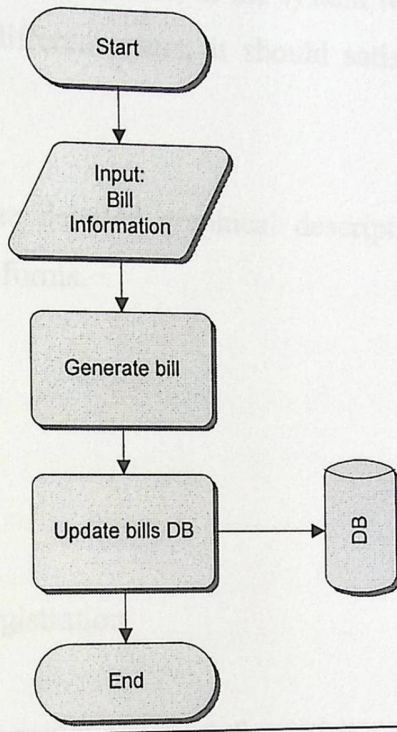
e) Flowchart:

Figure [4.42] Generate Bills (add or update bills) Flowchart

### 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.

I/O design includes the detailed graphical description of all I/O forms for each consumer and Bank Account forms.

#### Input screens design

##### A- Consumer Forms

- Consumer Login
- New Consumer registration
- Forget Password
- View/ Update personal consumer information
- Change Password
- Send Comments
- Agreement Terms & Conditions
- Disclaimer

##### B- Bank Account Forms

- Bank Account Information

#### Output Screen Design

##### A- Consumer Forms

- Debt and Collection
- Bill Details
- Bill Consumption report

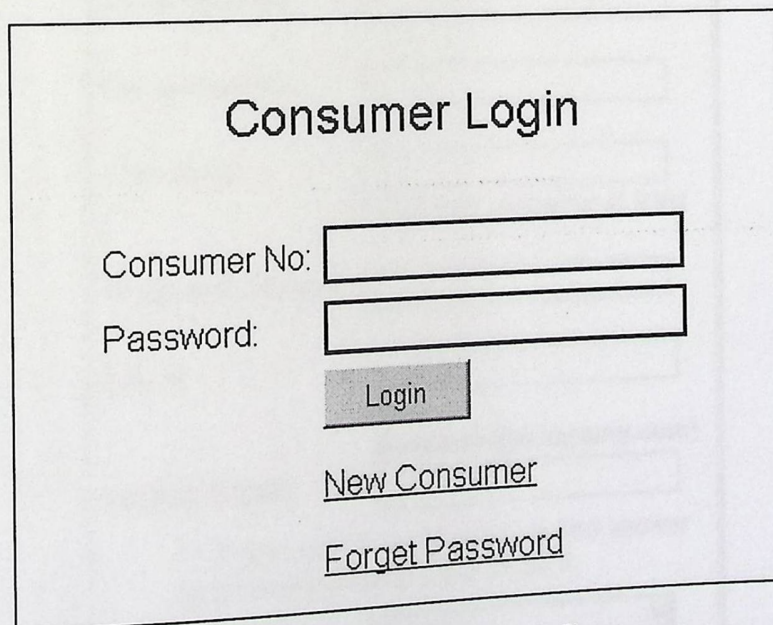
### 4.3.1 Input screens design

The system as specified previously has two forms categories; in this section we describe the input forms design for each of them:

#### 4.3.1.1 Consumer Input Forms

- *Consumer Login*

This screen enables Consumer to login into his account. After meeting the values typed by the Consumer into the two text boxes shown following in figure [4.43], with the consumer's login No and password stored in a record in the database, Consumer types his login No in the first text box and his login password in the second one. (Login) button is clicked to login the account and the new register link is used to make new login account if the consumer don't have an account yet, In addition, forget password link is used to retrieve the password if the consumer forgets it for some reasons.



The figure shows a web form titled "Consumer Login". It contains two text input fields: "Consumer No:" and "Password:". Below the "Password:" field is a "Login" button. At the bottom of the form are two links: "New Consumer" and "Forget Password".

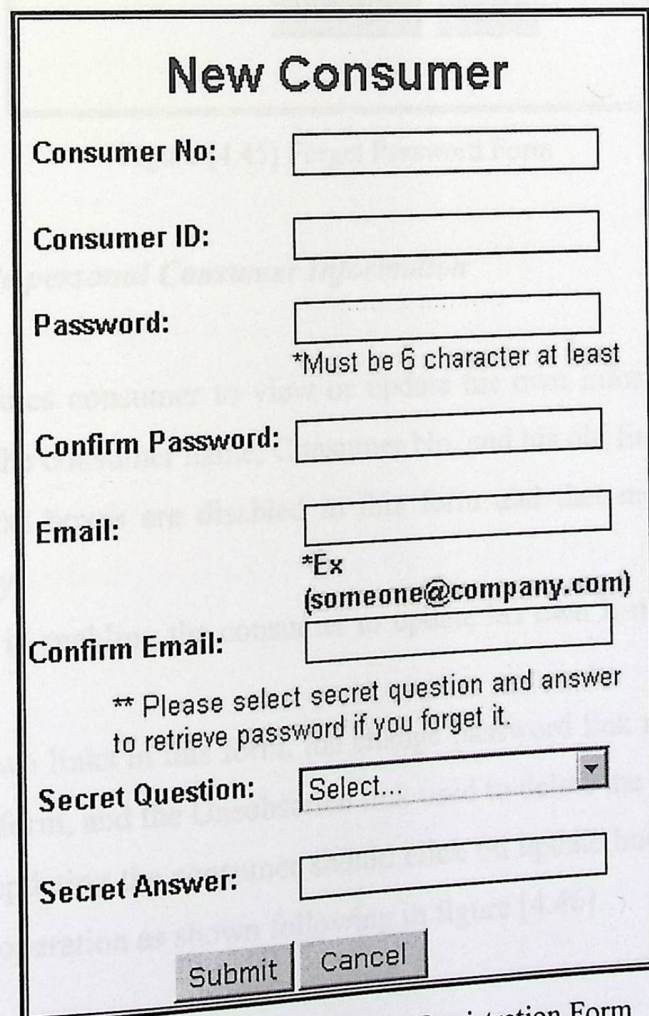
Figure [4.43] Consumer Login Form

- **New Consumer Registration**

In this form there is seven text boxes, one drop down list and two button, the first text box the consumer should input his login No, and in the second text box the consumer should insert his ID, and in the third text box the consumer should insert password, and in the forth text box the consumer should insert confirmation, and in the fifth text box the consumer should insert his Email, and the sixth text box to confirm password.

The second part of this form, the consumer should select secret question (drop down list one) and type secret answer in (text box eighth) to increase the efficiency of our system to enable the consumer to retrieve the password if consumer forgets it.

Then the consumer should click on the Submit button to precede the registration process or cancel button to ignore registration process as shown following in figure [4.44]



**New Consumer**

Consumer No:

Consumer ID:

Password:   
\*Must be 6 character at least

Confirm Password:

Email:   
\*Ex  
(someone@company.com)

Confirm Email:

\*\* Please select secret question and answer to retrieve password if you forget it.

Secret Question:

Secret Answer:

Figure [4.44] New Consumer Registration Form

- **Forget Password**

This form enables the consumer to retrieve the consumer password; the form contains two text boxes and two buttons the consumer should type this controls carefully, that mean the entered values should match the values that entered in registration process, otherwise the consumer can't retrieve his password as shown in figure [4.45]

**Forget Password**

Consumer No:

Secret Answer:

Figure [4.45] Forget Password Form

- **View / Update personal Consumer Information**

This form enables consumer to view or update his own information, the first three text boxes are view the consumer name, Consumer No, and his old Email.

Note: these three text boxes are disabled in this form and that mean they are used for viewing purpose only.

The fourth text box is enabling the consumer to update his own E-mail if he wishes to do that.

Also there is two links in this form, the change password link redirects the consumer to change password form, and the Unsubscribe link used to delete the consumer account. To activate E-mail updating the consumer should click on update button or click on cancel button to cancel the operation as shown following in figure [4.46].

**Update/View Personal Consumer Information**

Consumer Name:

Consumer No:

Old Email

New E-mail:

[Change Password](#)

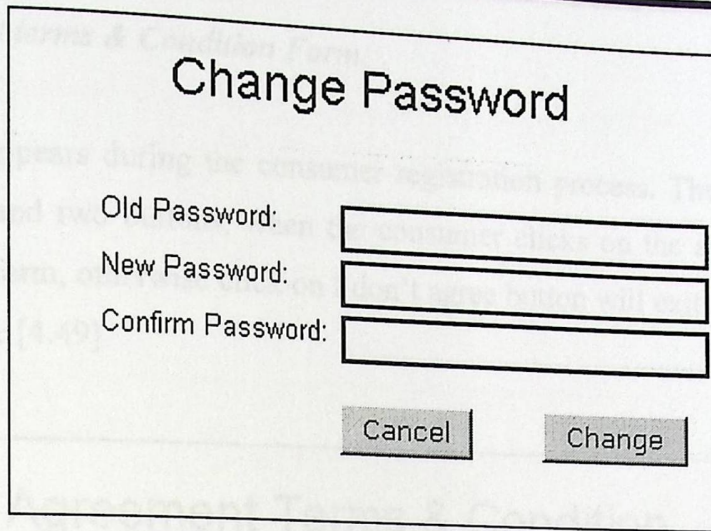
[Unsubscribe](#)

Figure [4.46] Update / View Personal Consumer Information Form

- ***Change Password Form***

This screen enables the consumer to change their login password; a consumer who wishes to change his login password must type his Old password in the first Text box respectively, the new password is to be typed in the second text box and confirmation of the password in the third text box.

When the consumer decides to change his password, he should click on the update button, or cancel this operation by clicking on the cancel button as shown following in figure [4.47]



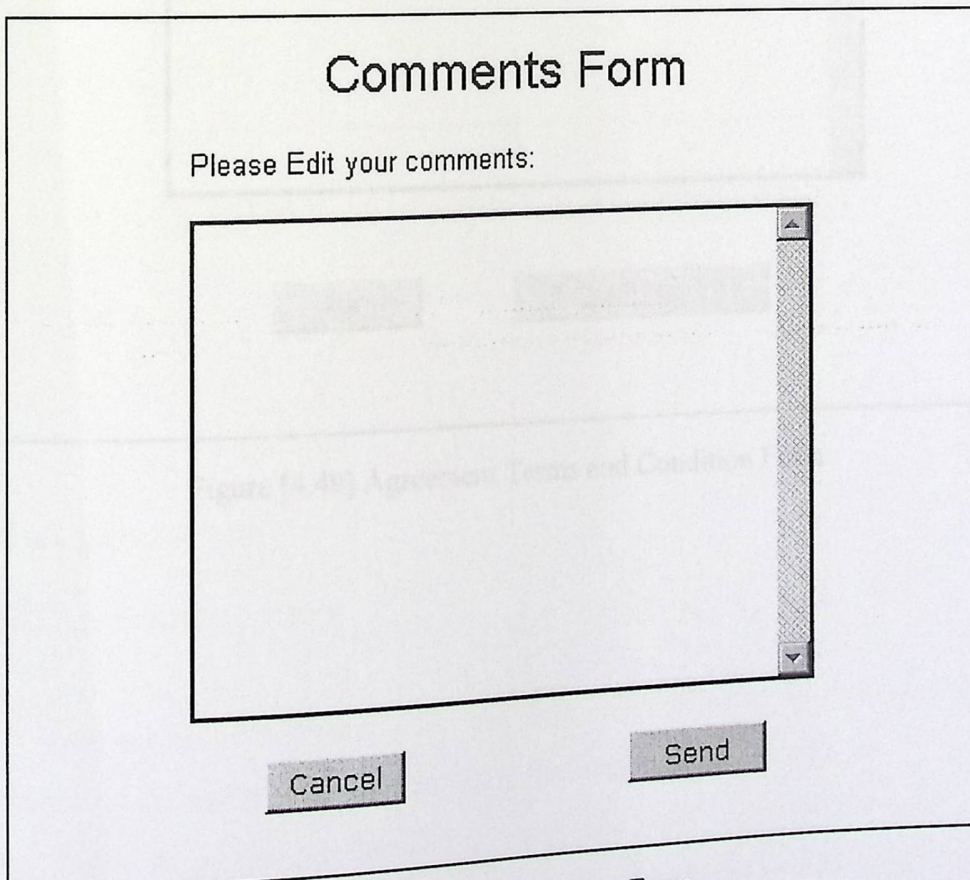
The image shows a 'Change Password' form. At the top, the title 'Change Password' is centered. Below the title, there are three input fields: 'Old Password:', 'New Password:', and 'Confirm Password:'. Each label is followed by a rectangular text box. At the bottom of the form, there are two buttons: 'Cancel' on the left and 'Change' on the right.

Figure [4.47] Change Password Form

▪ **Comments Form:**

This screen contains text box that enable consumer to type comments and suggestion about the E-payment system that used.

If consumer decides to send these comments he should click on Send button, otherwise click on cancel button to exit from this form as shown following in figure [4.48]



The image shows a 'Comments Form'. At the top, the title 'Comments Form' is centered. Below the title, the text 'Please Edit your comments:' is displayed. Underneath this text is a large rectangular text area with a vertical scrollbar on the right side. At the bottom of the form, there are two buttons: 'Cancel' on the left and 'Send' on the right.

Figure [4.48] Comments Form

- *Agreement terms & Condition Form:*

This form appears during the consumer registration process. The form contains the agreement script and two buttons, when the consumer clicks on the agree button, he can move to the next form, otherwise click on I don't agree button will exit this form, as shown following in figure [4.49]

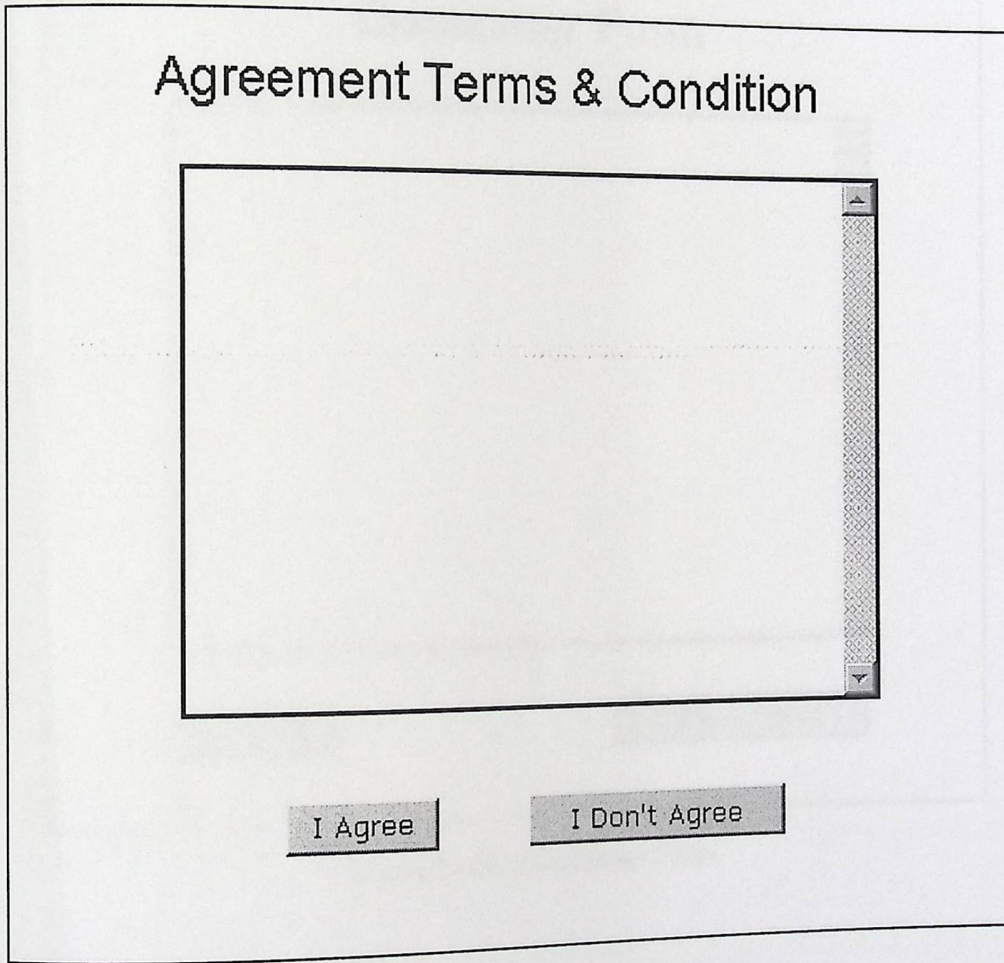
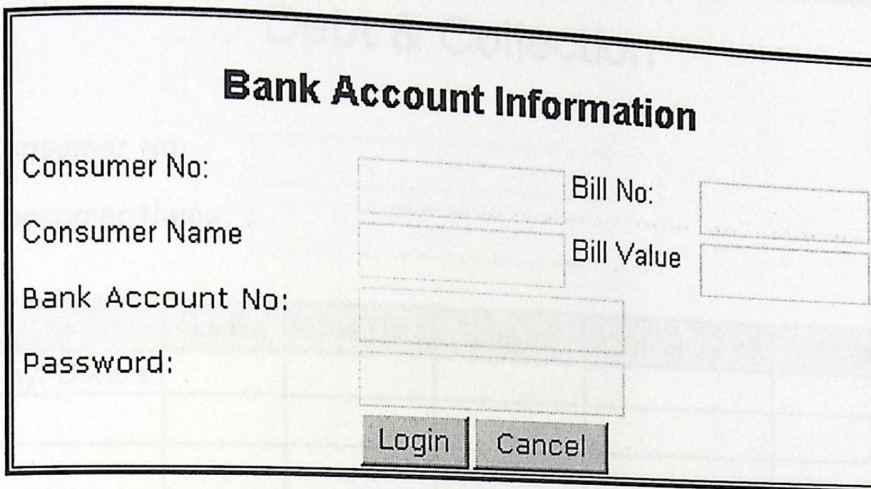


Figure [4.49] Agreement Terms and Condition Form



**Bank Account Information**

Consumer No:  Bill No:

Consumer Name:  Bill Value:

Bank Account No:

Password:

Figure [4.51] Bank Account Information Form

### 4.3.2 Output Screens Design

The system as specified previously three forms categories; in this section we describe the output forms design for each of them:

#### 4.3.2.1 Consumer Output Forms

- **Debts and collections**

This screen display thee current years status, current year balance, old year balance and tell the consumer if he has 8 free cups .Which contains basic information about each bill such as Bill No, Issue Date, Total Bill, payment date, and Status.

Also this form contain two text boxes used for shown purpose include consumer name and Consumer No and two buttons, the first to view all bills and the second to go into consumer home page.

If the consumer decides to view the bill details or pay the selected bill he should click on the Pay/ Details link as shown following in figure [4.52]





### Bill Details

Consumer No:  Bill No:

Consumer Name:  Address:

Issue Date:  Counter No:

Previous Reading:	Current Reading:	No of Consumed Cups:
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Consumption Value:**

Counter Cost:

Network Maintenance:

Due Date:

**Total Bill Value:**

Tell The Consumer if he has  
free 8 cups

Figure [4.54] Bill Details Form

▪ **Bill Consumption report**

This form contain three canvases the first part contain consumer information, the second part contain the information about the paid bill, and the third part contain the information of payment method. As shown in figure [4.55]

## Bill Consumption Report

### Consumer Information

Consumer No:

Consumer Name:

Currency:

### Paid Bill

Bill No	Issue Data	Total Bill Value
<input type="text"/>	<input type="text"/>	<input type="text"/>

### Payment Method

Bank

Please keep the receipt to proof payment process.

Payment Date:

The bill was paid



[Home Page](#)

Figure [4.55] Bill Consumption Report Form

## 4.4 Database design

This section describes the design of the system database and the database model.

### 4.4.1 Database Dictionary

#### 4.4.1.1 Hebron municipality DB Dictionary

##### Consumer Information Table

Field	Data type	Required	Key	Extra	description
Consumer_No	Numeric (6)	Yes	<b>PK</b>		A unique number for each registered consumer on the Hebron municipality web site.
Consumer_ID	Numeric (9)	Yes			A unique number for each consumer.
Password	Nvarchar (15)	Yes			Unique characters for each consumer.
Consumer_Name	Nvarchar (50)	Yes			Consumer Name
E-mail	Nvarchar (30)	Yes			Each consumer has a unique E-mail account.
Address	Nvarchar (60)	Yes			Consumer full address.
Free	Int(2)	yes			Boolean No that determine consumer whether if he has free 8 cups or not.

Table [4.1] Consumer Information DB Table

##### Bill Details Table

Field	Data type	Required	Key	Extra	description
Bill_No	Numeric (6)	Yes	<b>PK</b>	Auto Increment	The sequence number of the generated bill, each bill has a unique number.
Consumer_No	Numeric (6)	Yes	<b>FK</b>		A unique number for each registered consumer on the Hebron municipality web site.
Issue_Date	DateTime (8)	Yes			The date that the bill issued at.
Due_Date	DateTime (8)	Yes			The date that Hebron municipality allowed the consumer to pay at.
Status_No	Integer (4)	Yes	<b>FK</b>		A flag that determine weather the bill is paid or not.

Payment_Date	DateTime (8)	Yes				
Counter_No	Numeric (4)	Yes				The date that the consumer paid the bill at.
Previous_reading	Numeric (7)	Yes				A unique number of counters.
Current_reading	Numeric (7)	Yes				The number of previous counter.
Counter_install_date	DateTime (8)	Yes				The reading of the current counter.
Old_total_balance	Numeric (6)	Yes				The date of starting work of counter.
Network_maintenance	Integer (4)	Yes				Amount of unpaid bills in previous years.
Counter_cost	Integer (4)	Yes				Constant amount every month for maintaining.
Price	Integer (2)	Yes				Constant amount every month for maintaining.
Bill_value	Integer (4)	Yes				The price of each cup.
						The end value of bill.

Table [4.2] Bill Details DB Table

**Comments and Suggestions Table**

Field	Data type	Required	Key	Extra	description
Consumer_No	Numeric (6)	Yes	PK, FK		A unique number for each registered consumer on the Hebron municipality web site.
Comment No	Numeric (3)	Yes	PK		A unique number for each comment.
Comment	Nvarchar (150)	Yes			Set of character less than 60 characters it may be comment or suggestion.
Ddate	DateTime (8)	Yes			the date of sending the comments or suggestion

Table [4.3] Comments and Suggestions DB Table

## Consumer Answer Table

Field	Data type	Required	Key	Extra	description
Consumer_No	Numeric (6)	Yes	<b>PK, FK</b>		A unique number for each registered consumer on the Hebron municipality web site.
Question_No	Integer (3)	Yes	<b>PK, FK</b>		Each Question contain unique no.
Answer	Nvarchar (50)	Yes			The answer of the selected question typed by the consumer.

Table [4.4] Consumer Answer DB Table

## Consumer Question Table

Field	Data type	Required	Key	Extra	description
Question_No	Integer (3)	Yes	<b>PK</b>		Each Question contain unique no.
Question_Phase	Nvarchar (50)	Yes			Each Question has question text.

Table [4.5] Consumer Question DB Table

## Area Table

Field	Data type	Required	Key	Extra	description
Address_No	Integer (2)	Yes	<b>PK</b>		A unique number for each area.
Address_Name	Nvarchar (50)	Yes			A unique name for each area.

Table [4.6] Area DB Table

## 4.4.1.2 Bank DB Dictionary

## Bank Account Table

Field	Data type	Required	Key	Extra	description
Account_No	Numeric (6)	Yes	PK		A unique No for each account.
Account_total_balance	Numeric (12)	Yes			The value of money in the consumer account.
Password	Nvarchar (50)	Yes			Unique characters for each consumer, these characters should encrypt in some encryption methods to increase the level of security.
Name	Nvarchar (50)	Yes			The name of owner of the account.

Table [4.7] Bank Account Table

## Bank Transaction Table

Field	Data type	Required	Key	Extra	description
Transaction_No	Numeric (2)	Yes	PK	Auto Increment	The sequence number of each transaction.
Account_No	Numeric (6)	Yes	FK		A unique No for each account.
Payment_Amount	Float (4)	Yes			The mount of money that is paid via consumer bank account
Payment_Date	DateTime (8)	Yes			The date that the consumer paid the bill at.
Beneficiary_Account_No	Numeric (6)	Yes			Account no that received money (Hebron Municipality).

Table [4.8] Bank Transaction Table

4.4.2 Database Model

We have implemented a relational database model for our system database design and implementation, shown bellow in figure [4.56] UML notations that describes the database model.

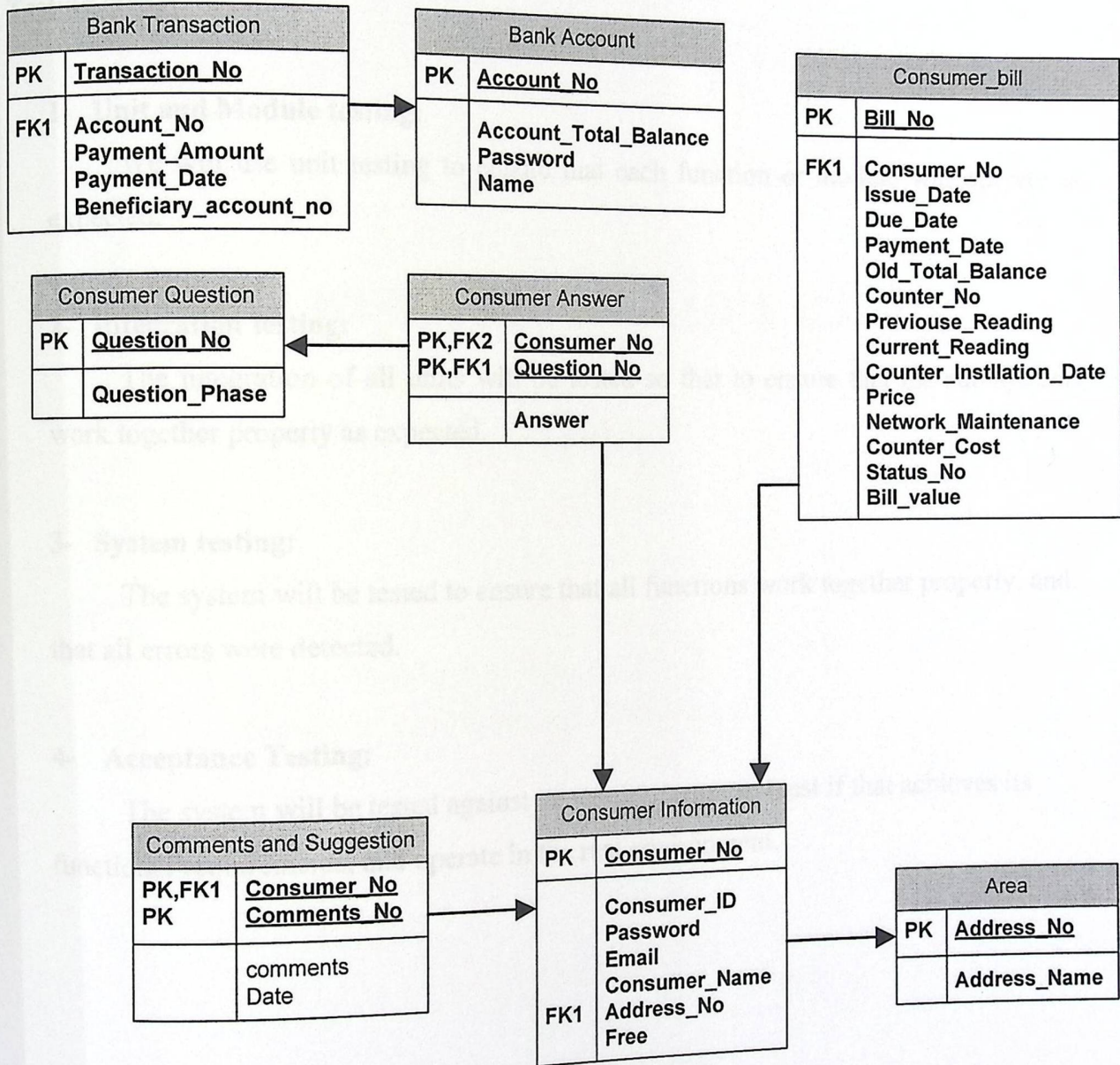


Figure [4.56] Database Model

## 4.5 Test Plan

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

### Testing steps:

#### 1- Unit and Module testing:

We will use unit testing to ensure that each function or module will operate as expected.

#### 2- Integration testing:

The integration of all units will be tested so that to ensure that the sub-systems work together properly as expected.

#### 3- System testing:

The system will be tested to ensure that all functions work together properly, and that all errors were detected.

#### 4- Acceptance Testing:

The system will be tested against its requirements, and test if that achieves its functional requirements, and operate in the real environment.

5.1 Introduction

Chapter

5

# Implementation and Coding

The Content:

- 5.1 Introduction.
- 5.2 Coding Programming Language.
- 5.3 Establishment of Development Environment.

5.2 Coding P

relationship between ASP.NET, VB.NET, SQL server 2000 Exchange server 2003

After creating the design specifications that will be used while creating the web application, create a new project in the Visual Studio.NET and create the interface for Hebrew municipality web application by setting the initial properties for the ASP.NET web forms. Visual Basic.NET is used to write the event procedures that will run when different users are performed on a control or object. Then Hebrew municipality website is built using SQL server 2000.

## 5.1 Introduction

Here we describes the main steps must be followed to start in coding and programming to reach the design that is described in the previous chapters, and mention the programming language is used for this purpose.

This chapter focuses on the coding and implementation of Hebron municipality project.

Coding refers to the process of writing the necessary program, which implements the main procedures and functions of the project. The code of the project is to be written from the scratch using VB.NET language and ASP.NET technology.

The project is implemented as a set of programmed pages; each page carries out a specific functions related to the project, the project is to be programmed under windows server 2003 operating system.

## 5.2 Coding Programming Language

Before starting in coding, it is desirable to study the scopes of languages and to match between the project requirements and what a specific language offers. There are different programming languages and different DB systems to implement the project, but after studying and analyzing all these options, the development team has selected ASP.NET for coding the web-based application, SQL server 2000 for DB, Visual Basic.NET for coding and Exchange server 2003 for E-mail hosting.

### **Relationship between ASP.NET, VB.NET, SQL server2000 Exchange server 2003:**

After creating the design specifications that will be used while creating the web application, create a new project in the Visual Studio.NET and create the interface for Hebron municipality web application by setting the initial properties for the ASP.NET web forms. Visual Basic.NET is used to write the events procedures that will run when different actions are performed on a control or object. Then Hebron municipality database is built using SQL server 2000.

From the visual studio.net a connection to Hebron municipality is created, this connection provides the facilities to access the database (input, output, access) from the web application. Since the desirable system is to be run on the internet and need a database handle its data, ASP.NET is chosen which is good to complete the project.

### **ASP .NET platform:**

There are many languages that can be used to develop a system such ours, but the most two effective languages are CGI language and .NET using Microsoft Visual Studio .NET as the development environment, here we describe the advantages of each, and why our selection was on the ASP.NET:

#### **1- Device Independent:**

ASP.Net are a device independent language, this means that any user can open the page that build using .NET framework environment from any explorer without any additional components or drivers, because the client host will receive only HTML code.

#### **2- Security:**

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

In the ASP.NET technology there are a build in valuator 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.

#### **3- Server side:**

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

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 figure [5.1]

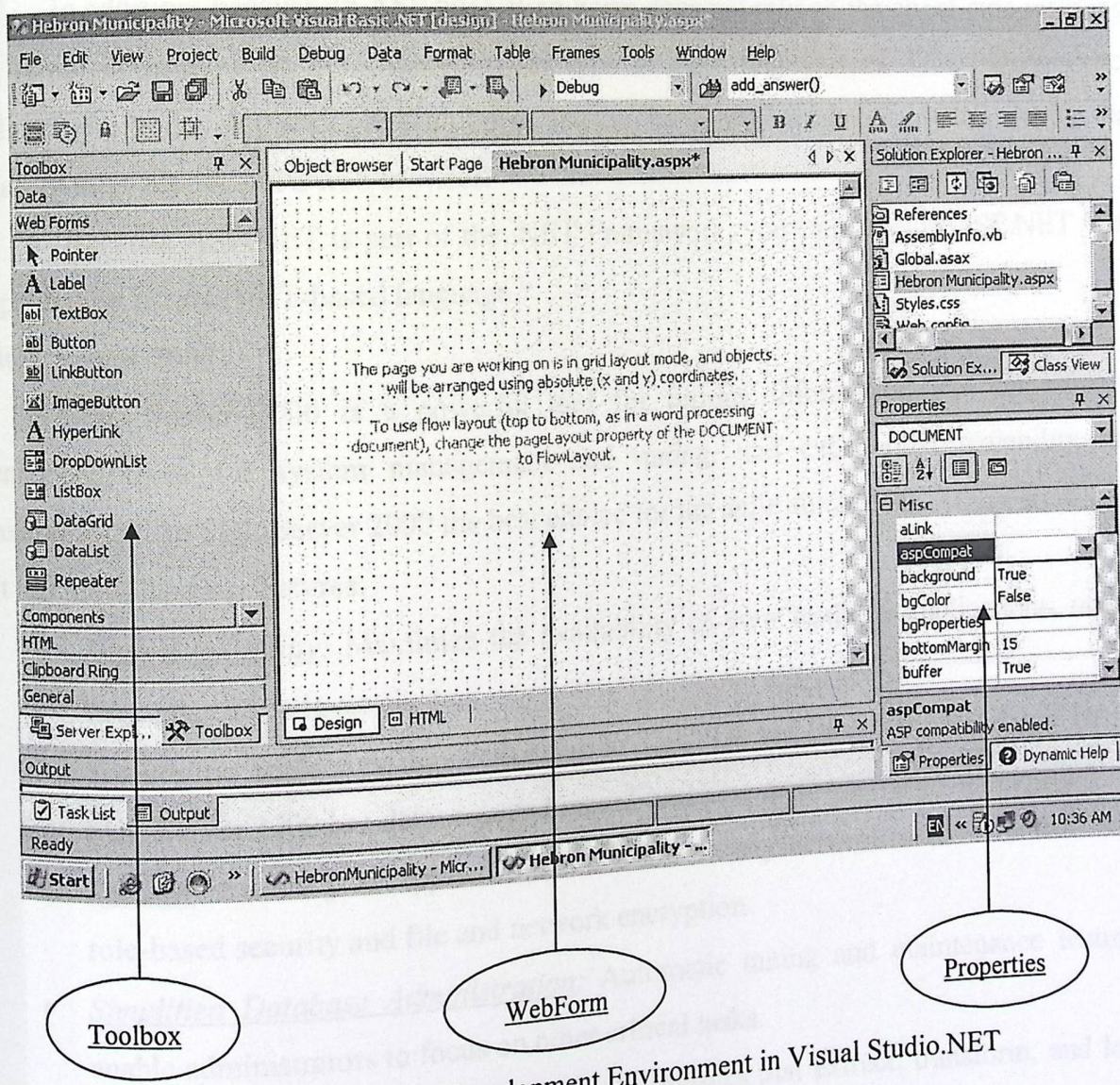


Figure [5.1] Integrated Development Environment in Visual Studio.NET

**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 can view using internet 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.

**SQL server 2000:**

SQL Server 2000 is a powerful tool for turning information into opportunity, 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, 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.
- 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.

**Exchange server 2003:**

E-mail is currently the most widely used collaborative technology. More businesses use Exchange Server for e-mail-based collaboration than any other product. Exchange Server 2003 enables knowledge workers to gain access to critical business communications almost whenever and wherever they need to and it designed to deliver greater security, availability, and reliability.

***Features at a Glance:***

- Database size limited only by your hardware (with a maximum size of 16 terabytes).
- Multiple databases per server.
- Clustering support four nodes on Microsoft Windows 2000 Advanced Server or eight nodes on Microsoft Windows Server 2003 Enterprise Edition.
- The flexibility to be configured as a Microsoft Office Outlook Web Access front-end server.
- Mailbox stores and public folder stores that are each limited to 16-gigabyte (GB) maximum.
- Recovery Storage Group.

### 5.3 Establishment of Development Environment

- Purchase the computers and the software required for developing the system.
- Install windows XP.
- Install the required utilities.
- Install the IIS; from the control panel, choose add remove programs then choose windows components then add the IIS component by choosing its check box then click Next as shown in figure [5.2], [5.3]

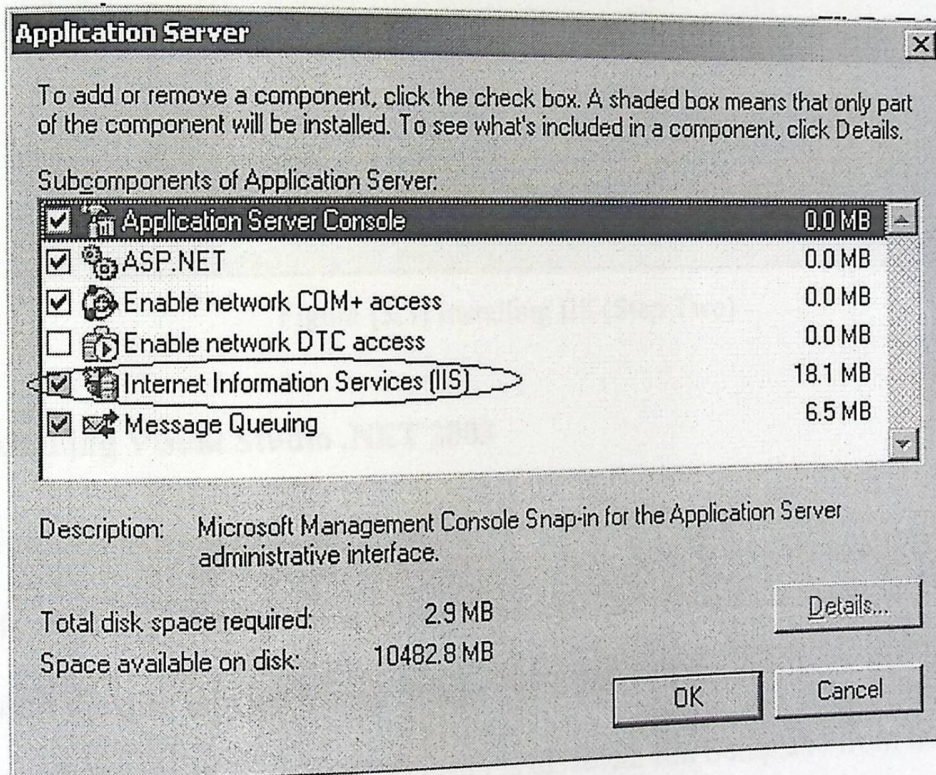


Figure [5.2] Installing IIS (Step One)

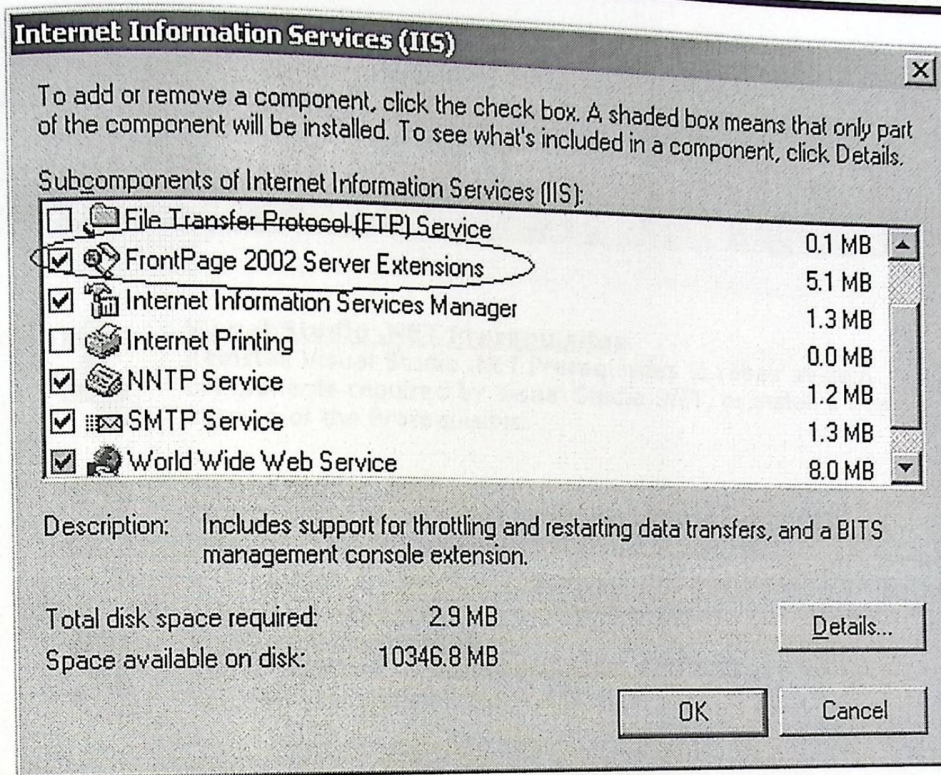


Figure [5.3] Installing IIS (Step Two)

- **Installing Visual Studio .NET 2003**

To install Visual Studio .NET 2003

- Close all open applications to prevent additional reboots during setup.
- Insert the disc labeled Visual Studio .NET 2003 CD1.
- Autorun starts Setup.exe. If Autorun is disabled, run Setup.exe from the root of the installation CD.
- The setup program scans your disk for installed components. If the scan determines that the system requires a component update, Step 1: Install Visual Studio .NET 2003 Prerequisites is available in the Installation dialog box. Select Step 1 to update the system components. If a component update is not required, this option is not available. After updating the system components, the Installation dialog box enables Step 2: Install Visual Studio .NET 2003. Select Step 2 to install Visual Studio .NET 2003. As shown in figure [5.4]

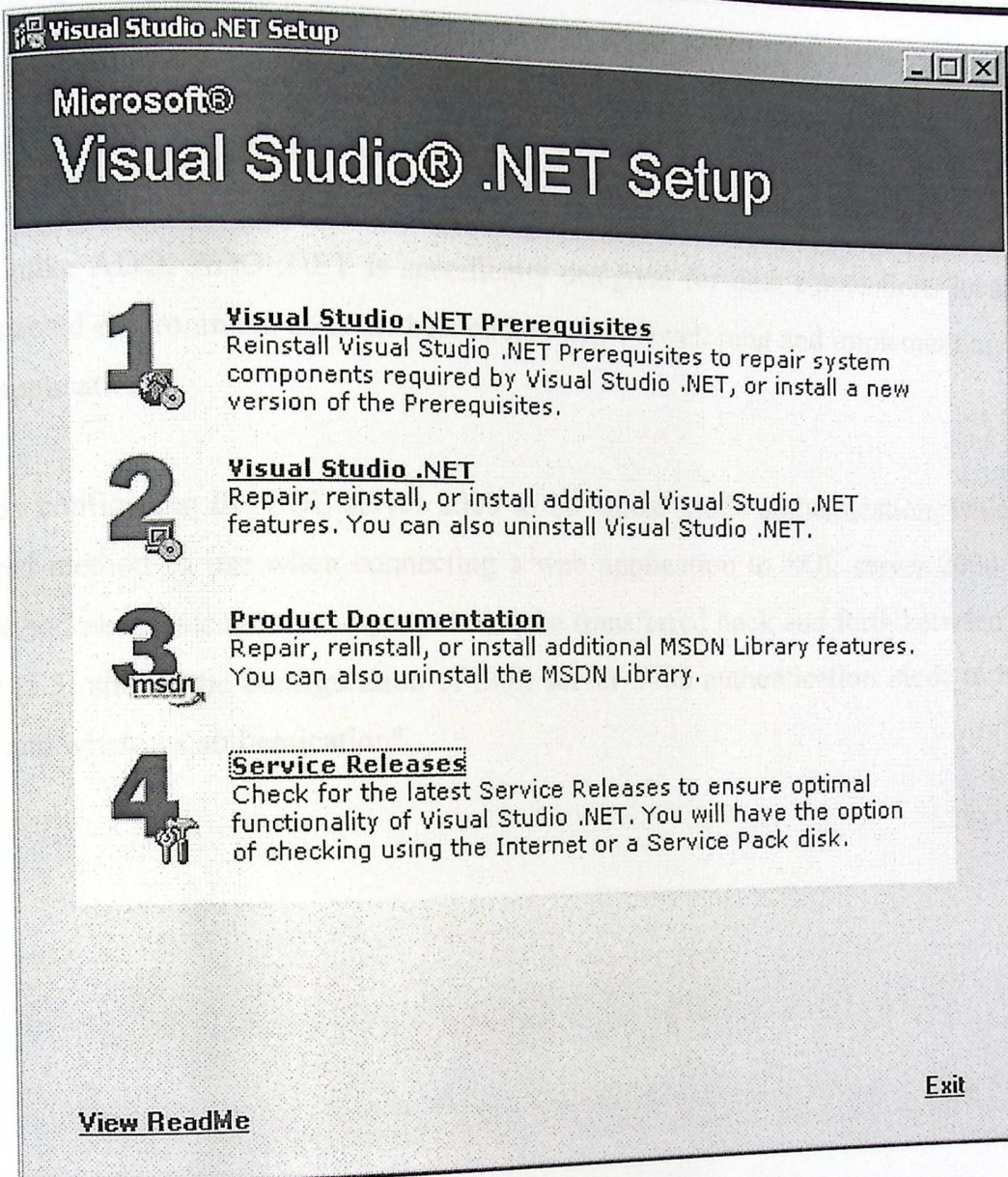


Figure [5.4] Install Visual Studio .NET 2003

- **Install SQL server 2000 software.**

An Enterprise edition of the Microsoft SQL server 2000 meets our system requirement in creating, accessing, and managing the system database.

#### **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 [5.5] shows the configuration of SQL server 2000 authentication mode to be "SQL server and windows authentication".

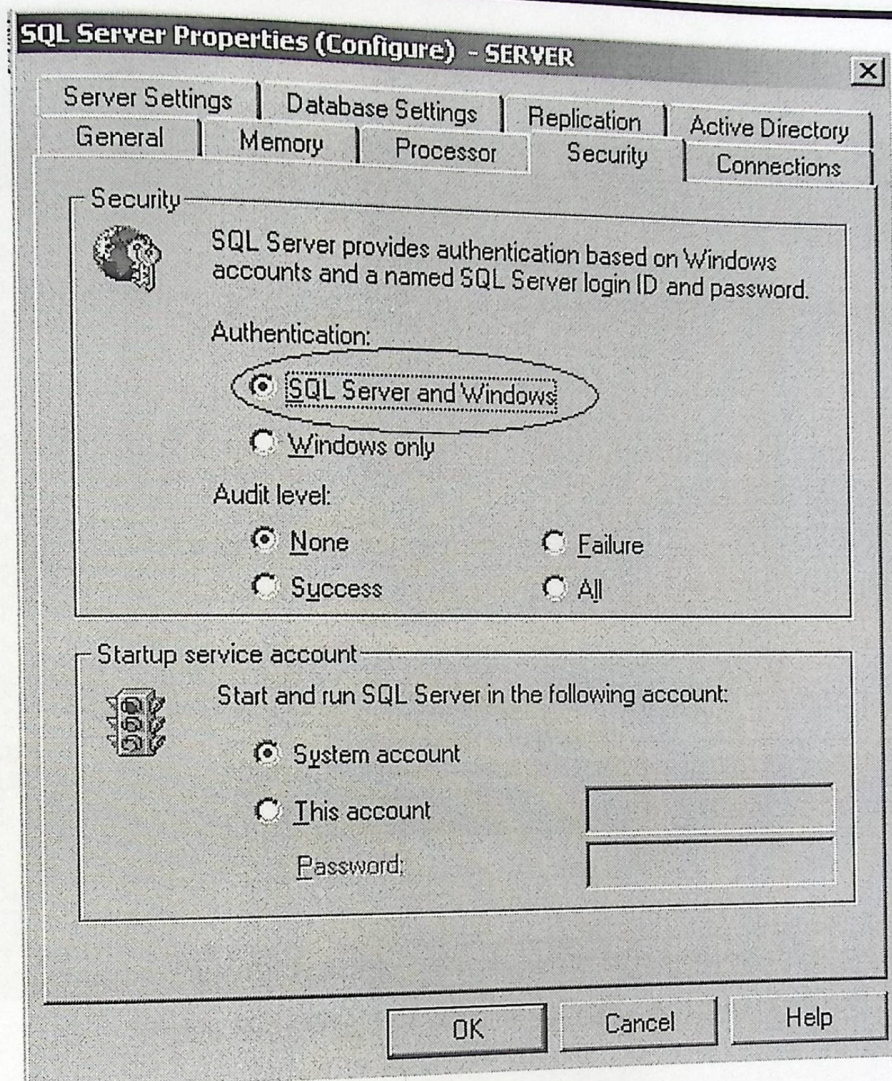


Figure [5.5] Mixed Mode Authentication (Step One).

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 [5.6] shows how to add this account to the logins group of the SQL Server.

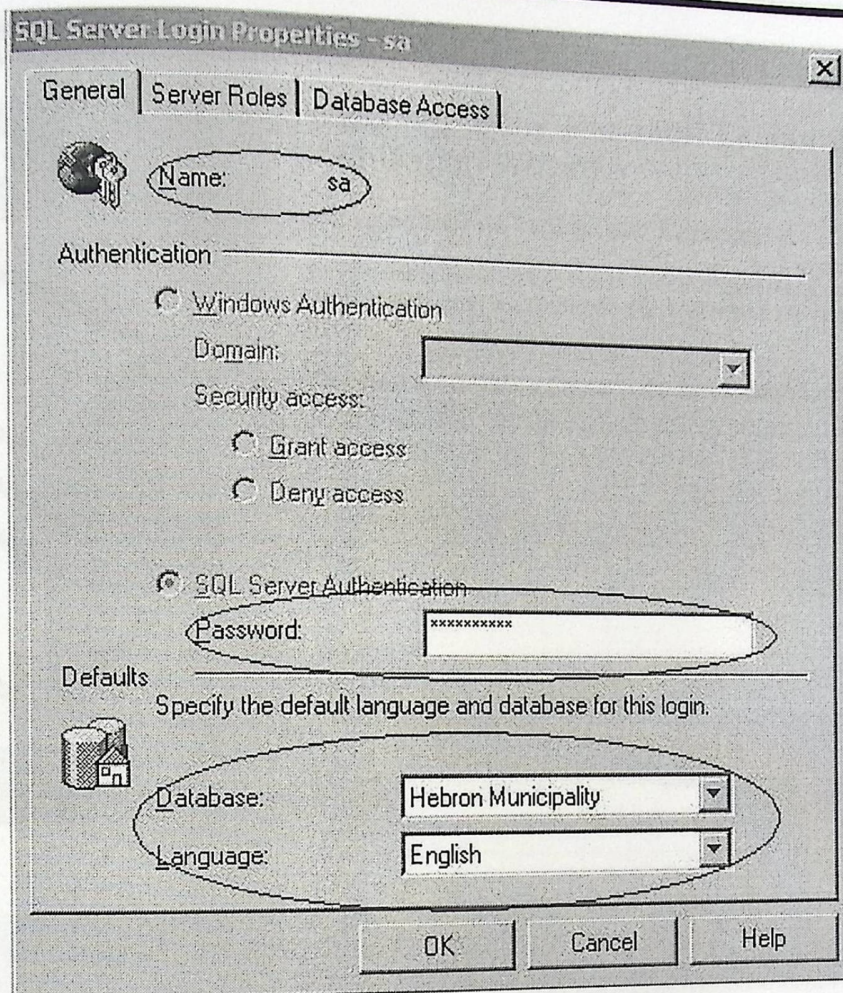


Figure [5.6] Mixed Mode Authentication (Step Two).

- **Install Exchange server 2003.**

- On the server, install windows 2000 server SP3 or later, windows 2000 advanced server SP3 or later, or windows server 2003.
- Verify that the NNTP, SMTP, and World Wide Web services are installed and enabled on the server, if you are running windows server 2003, verify that ASP.NET is installed.
- Run forestprep to extend the active directory schema for exchange server 2003, to run forestprep your account should have the administrator permission.
- Run domainprep to prepare the domain for exchange 2003.
- Install exchange server 2003 on the new server by running exchange setup.

As shown in figure [5.7]

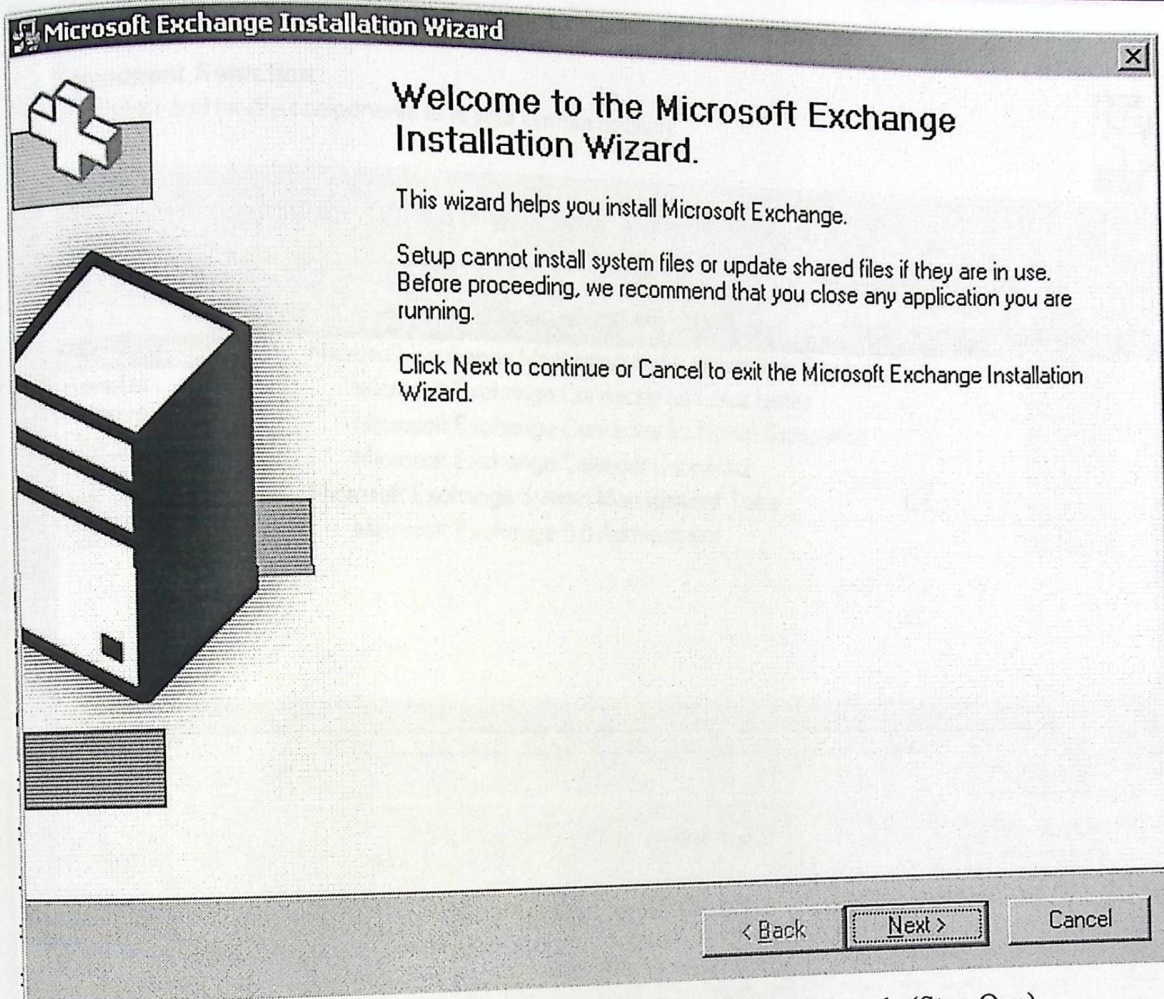


Figure [5.7] Exchange Server 2003 Installation Wizards (Step One).

After the installation complete go to the control panel > Microsoft exchange > Change/ remove button, you will get the window that shown bellow, and from this window you can perform some action on exchange server like Remove, Reinstall, figure [5.8] represented the action menu.

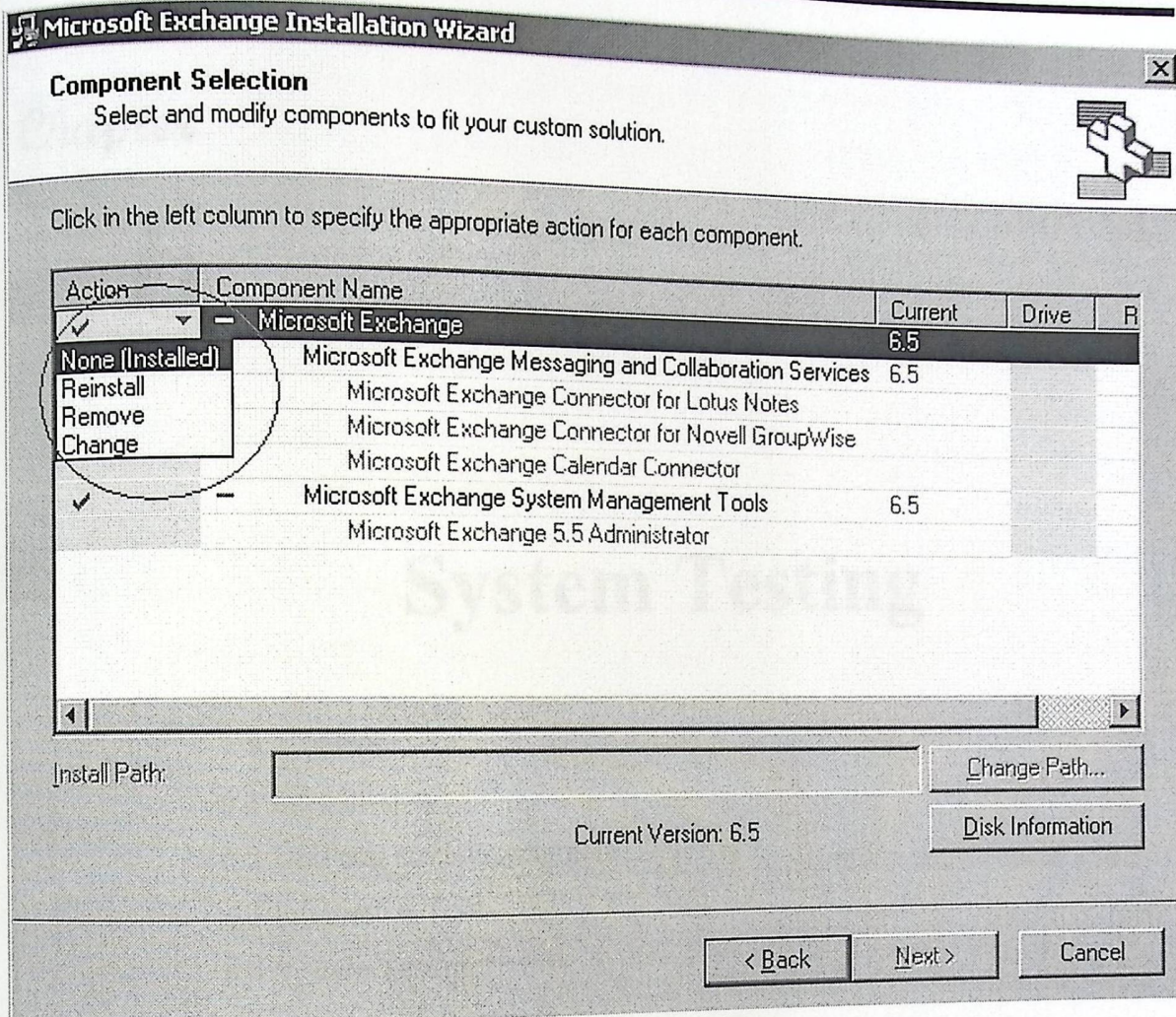


Figure [5.8] Exchange Server 2003 Installation Wizards (Step Two).

*Chapter*

6

# System Testing

The Content:

- 6.1 Introduction.
- 6.2 Unit and module testing.
- 6.3 Integration Test.
- 6.4 System Testing.
- 6.5 Acceptance Testing.
- 6.6 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.

Figure (6.1) shows the testing schedule:

Time in week \ Testing process	1st week	2 <sup>nd</sup> week	3 <sup>rd</sup> week
Unit and module testing.			
Sub-system testing.			
Integration testing.			
Acceptance testing.			

Figure (6.1) Testing Schedule.

## 6.2 Unit and module testing

We have tested the units and modules using the white box testing method, 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: " Consumer 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 [6.2].

Figure 6.2) Login Consumer Execution Path

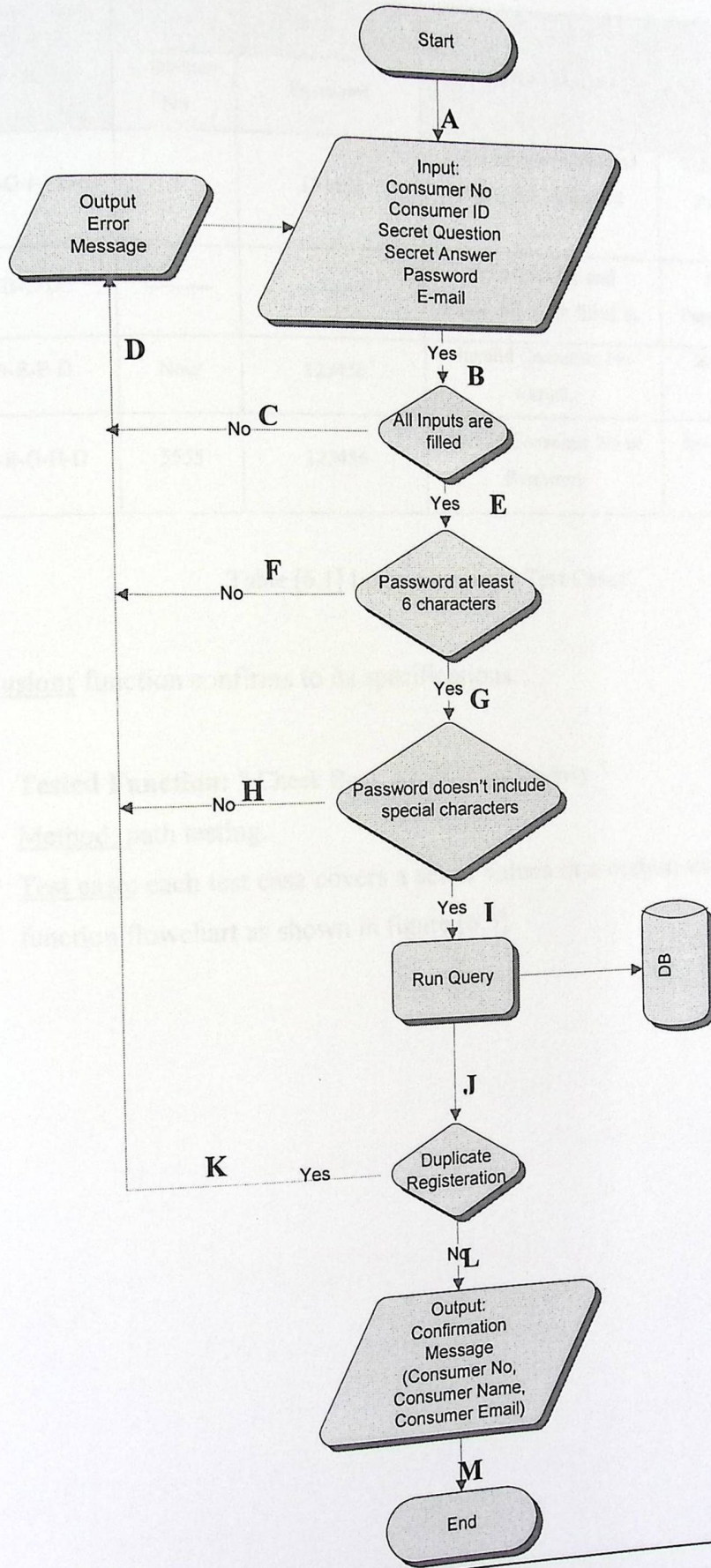


Figure [6.2] Login Consumer Execution Paths.

Test Cases Path	Test Data		Expected Output	Actual Output
	Consumer No	Password		
A-B-E-G-I-J-L-M	1	123456	Valid Consumer No and Password (Accepted User).	Valid Consumer No and Password (Accepted User).
A-B-C-D	_____	_____	Consumer No and Password is not filled in.	Consumer No and Password is not filled in.
A-B-E-F-D	Nour	123456	Invalid Consumer No format.	Invalid Consumer No format.
A-B-E-G-H-D	5555	123456	Invalid Consumer No or Password.	Invalid Consumer No or Password.

Table [6.1] Login Consumer Test Cases.

**Conclusion:** function confirms to its specifications.

2. **Tested Function:** " Check Bank Account Sufficiency "

Method: path testing.

Test case: each test case covers a set of values in a certain execution path in the function flowchart as shown in figure [6.3].

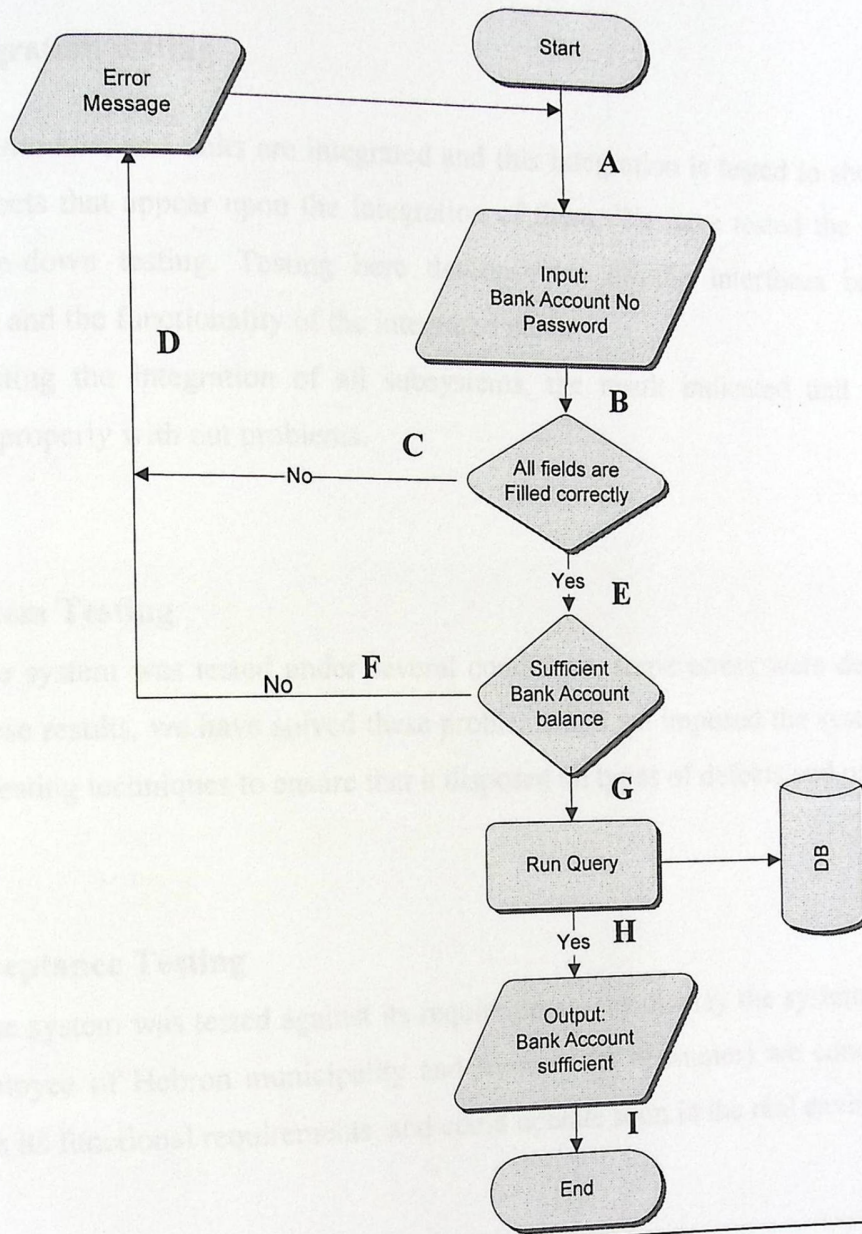


Figure [6.3] Check Bank Account Sufficiency Execution Paths.

Test Cases path	Test Data		Expected Output	Actual Output
	Account No	Password		
A-B-E-F-D	1	123456	Not sufficient account balance	Not sufficient account balance
A-B-C-D		123321	All field not filled correctly	All field not filled correctly
A-B-E-G-H-I	3	112233	sufficient bank account balance	sufficient bank account balance

Table [6.2] Check Bank Account Sufficiency Test Cases.

**Conclusion:** function confirms to its specifications.

### 6.3 Integration testing

All module, and units are integrated and this integration is tested to show if there were defects that appear upon the integration of them. We have tested the integration using top-down testing. Testing here demonstrates on the interfaces between all modules, and the functionality of the integrated parts.

After testing the integration of all subsystems, the result indicated that they work together properly with out problems.

### 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, (by display the system to some of the employee of Hebron municipality and some of its consumer) 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 and these snapshots are describe the main functions of our web based system as shown bellow:

### 6.6.1 Registration Process Snapshots

To enable the consumer to get our e-services he /she must register first in Hebron Municipality web site, by clicking on New User Icon on the left panel the consumer will start the registration process as shown bellow:

#### 6.6.1.1 Login Page.

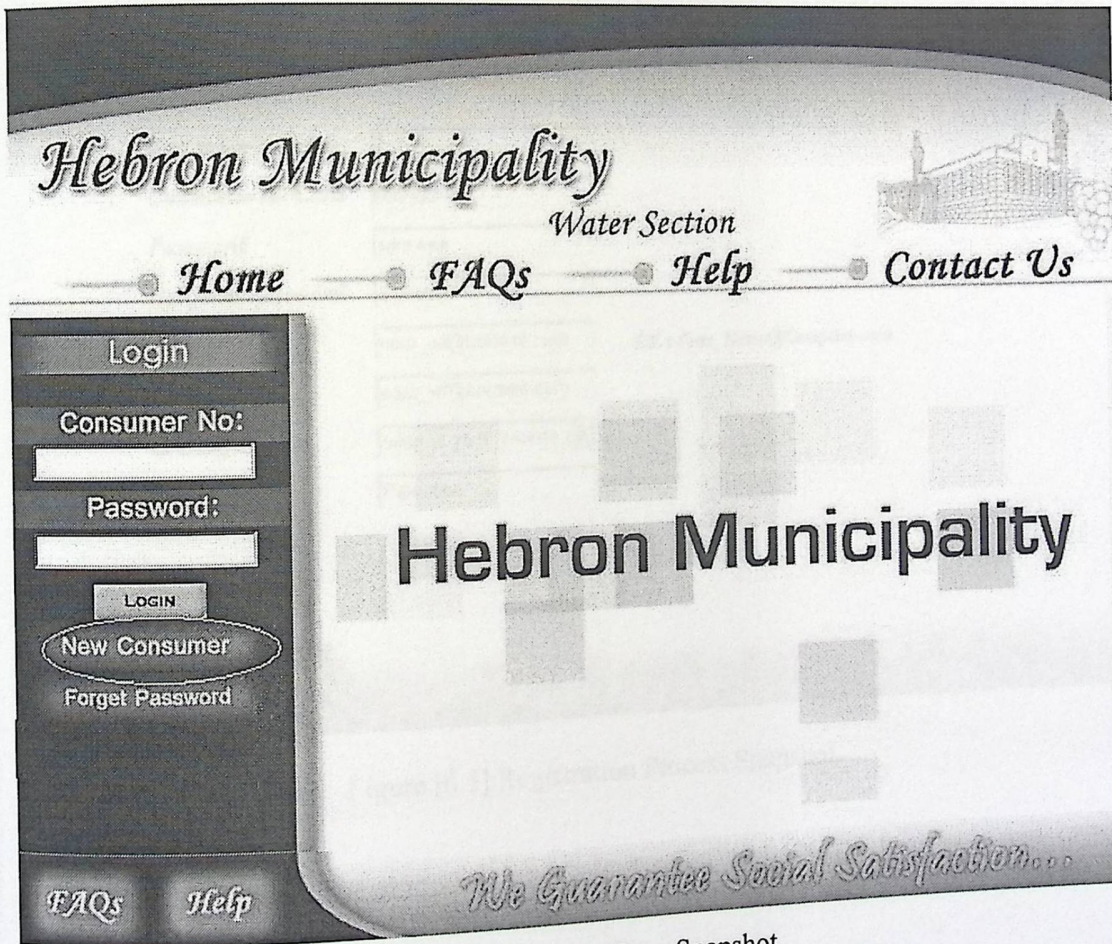


Figure [6.4] Login Page Snapshot.

## 6.6.1.2 Registration process.

After the successful login registration the registration process is shown.

**Hebron Municipality**  
Water Section

Home    FAQs    Help    Contact Us

## Registration

Consumer number

Consumer ID Card

Password

Confirm Password

E-Mail  EX : Your\_Name@Company.com

Confirm E-Mail

Secret Question

Secret Answer

FAQs    Help    *We Guarantee Social Satisfaction...*

Figure [6.5] Registration Process Snapshot.

### 6.6.1.3 Consumer Home page

After the consumer finish successfully the registration process the Hebron Municipality system redirect the consumer to his home page as confirmation step that he completes the registration process successfully as shown bellow:

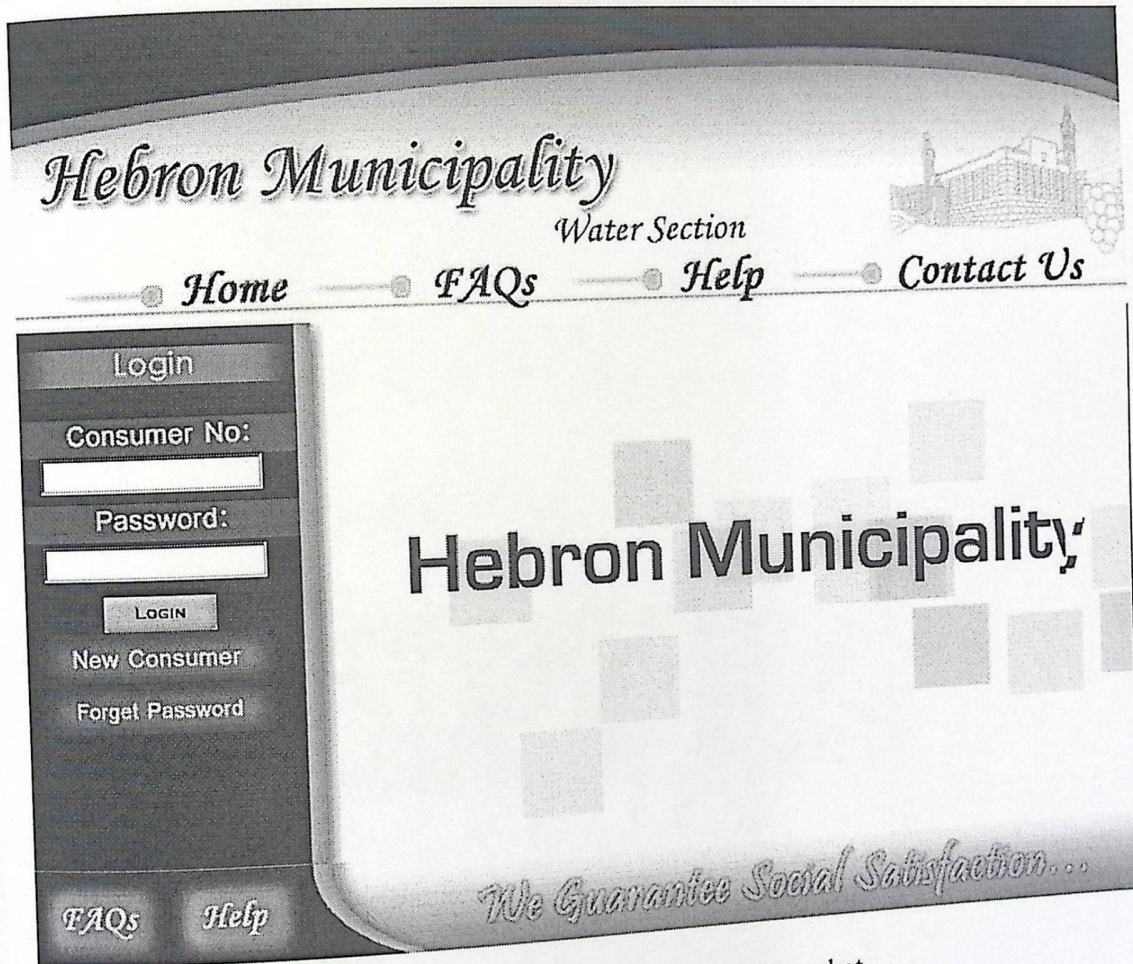


Figure [6.6] Consumer Home Page Snapshot.

## 6.6.1.3 Consumer Home page

After the consumer finish successfully the registration process the Hebron Municipality system redirect the consumer to his home page as confirmation step that he completes the registration process successfully as shown bellow:



Figure [6.6] Consumer Home Page Snapshot.

### 6.6.1.3 Consumer Home page

After the consumer finish successfully the registration process completed there some changes occurred in Hebron Municipality system redirect the consumer to his home page as he completes the registration process successfully as shown below.

consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football

Figure [6.7] Consumer Table Before Changes Snapshot.

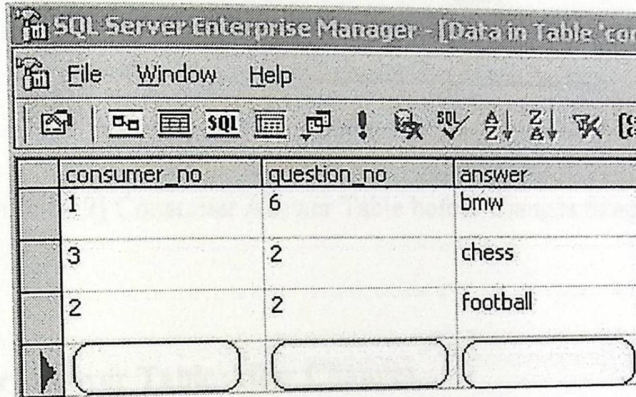
### 6.6.1.5 Consumer Table After Changes

consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football
7	1	cat

Figure [6.8] Consumer Table After Changes Snapshot.

After the registration process completed there some changes occurred in Hebron Municipality Database that include the information about new consumer account as shown bellow:

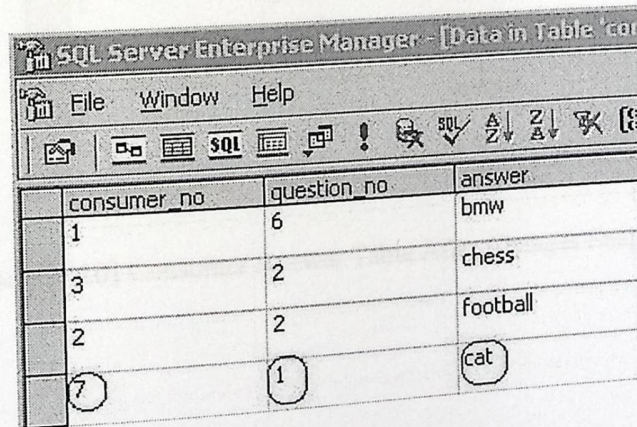
#### 6.6.1.4 Consumer Table Before Changes



consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football

Figure [6.7] Consumer Table Before Changes Snapshot.

#### 6.6.1.5 Consumer Table After Changes

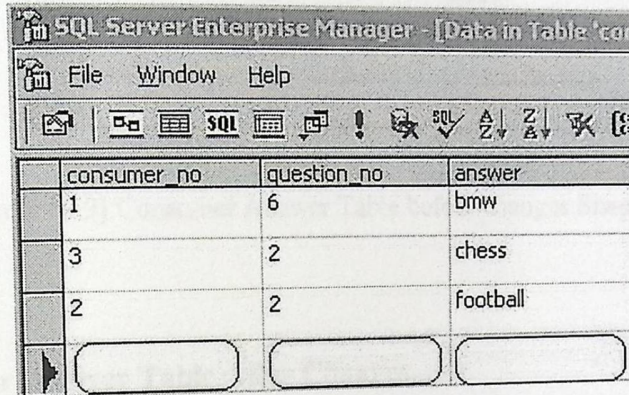


consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football
7	1	cat

Figure [6.8] Consumer Table After Changes Snapshot.

After the registration process completed there some changes occurred in Hebron Municipality Database that include the information about new consumer account as shown bellow:

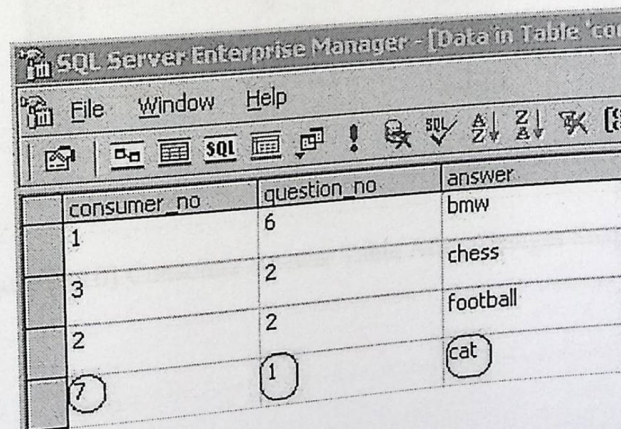
#### 6.6.1.4 Consumer Table Before Changes



consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football

Figure [6.7] Consumer Table Before Changes Snapshot.

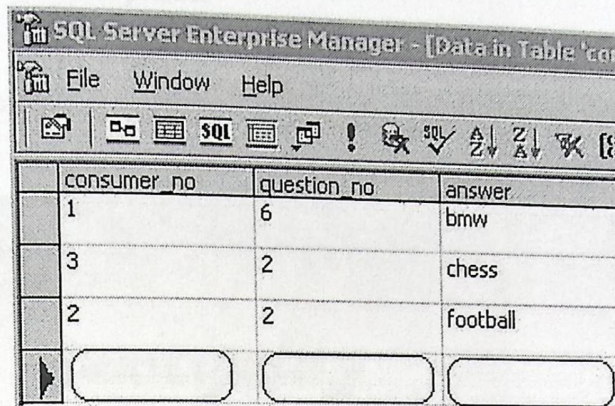
#### 6.6.1.5 Consumer Table After Changes



consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football
7	1	cat

Figure [6.8] Consumer Table After Changes Snapshot.

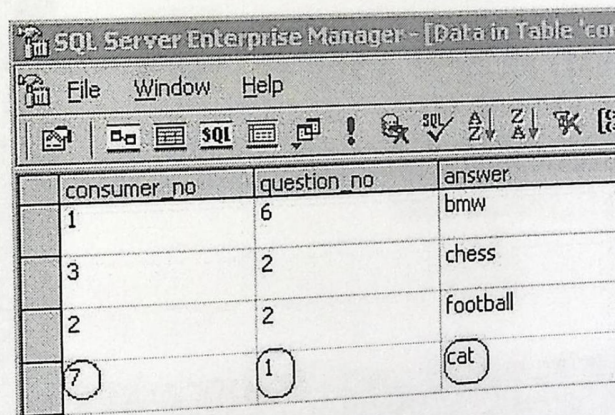
## 6.6.1.6 Consumer Answer Table Before Changes



consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football

Figure [6.9] Consumer Answer Table before changes Snapshot.

## 6.6.1.7 Consumer Answer Table After Changes



consumer_no	question_no	answer
1	6	bmw
3	2	chess
2	2	football
7	1	cat

Figure [6.10] Consumer Answer Table After Changes Snapshot.

6.6.2 Payment Process Snapshots

6.6.2.1 Login Page



Figure [6.11] Login Page Snapshot.

6.6.2.2 Consumer Home Page

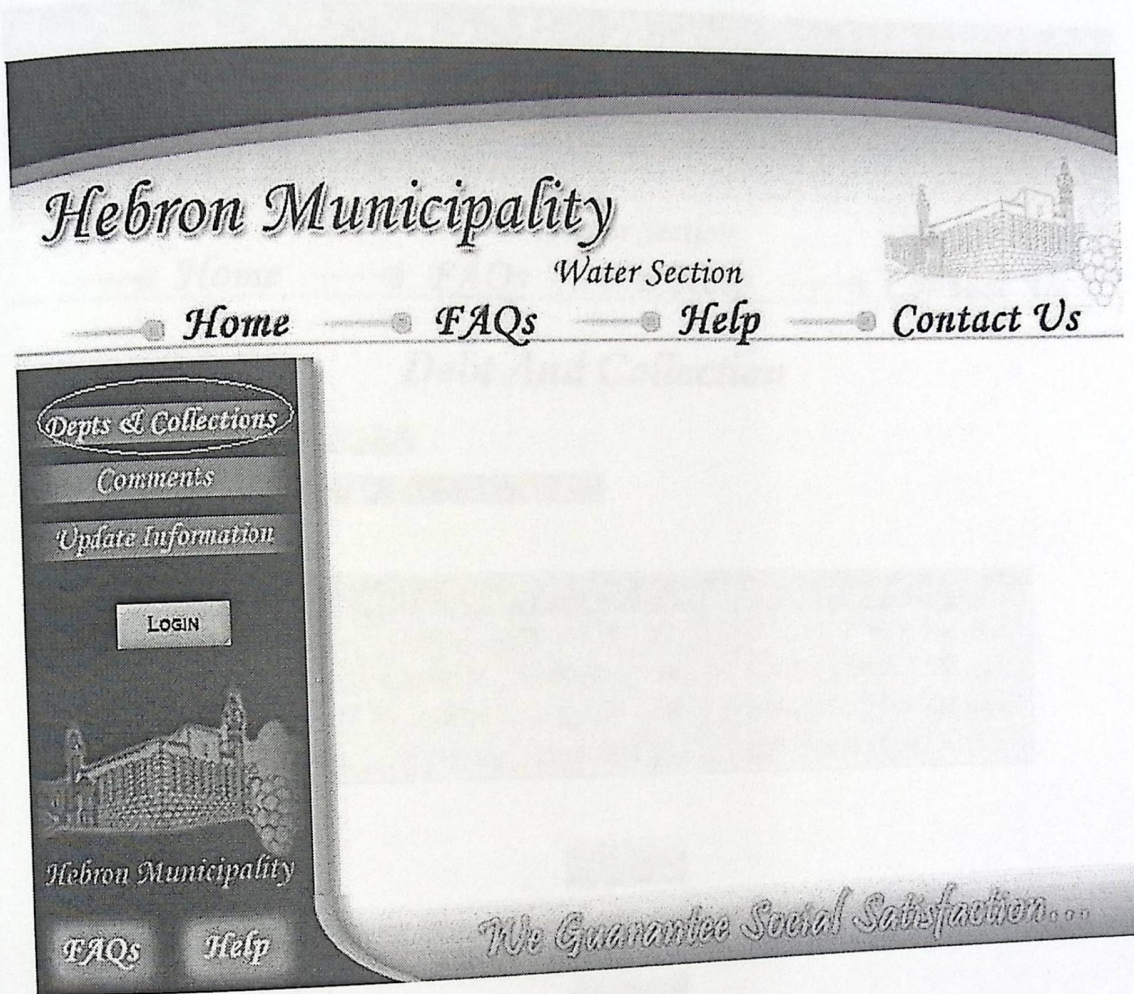


Figure [6.12] Consumer Home Page Snapshot.

## 6.6.2.3 Dept and Collection Page

**Hebron Municipality**  
Water Section

Home    FAQs    Help    Contact Us

### Debt And Collection

consumer no: 7

consumer name: Nour Qadi Tamimi

	Bill No	Issue Date	Total Bill	Payment Date	Status
<a href="#">Pay/Details</a>	4	1/1/2006 12:00:00 AM	56	10/1/2006 12:00:00 AM	unpaid
<a href="#">Pay/Details</a>	6	1/5/2006 12:00:00 AM	506	12/7/2006 12:00:00 AM	paid
<a href="#">Pay/Details</a>	8	6/6/2006 12:00:00 AM	406	12/6/2006 12:00:00 AM	paid
<a href="#">Pay/Details</a>	11	12/12/2006 12:00:00 AM	25	12/7/2006 12:00:00 AM	paid

[VIEW ALL BILLS](#)

current year balance 56      old year balance 568

[HOME PAGE](#)

FAQs    Help    We Guarantee Social Satisfaction...

Figure [6.13] Debt and Collection Page Snapshot.

## 6.6.2.4 Bill Details Page

**Hebron Municipality**  
Water Section

Home    FAQs    Help    Contact Us

### Bill Details

Consumer No: 7      Bill No: 4  
 Consumer Name: Nour Qadi Tamimi      Counter No: 12345  
 Issue Date: 1/1/2006      Address: Raa's Al-Jora

Previous Reading	Current Reading	No Of Consumed Cups
810	820	10

Consumption Value: 50

Counter Cost: 3  
 Network Maintenance: 3  
 Due Date: 10/1/2006  
 Total Bill Value: 56

PAY    BACK

FAQs    Help    We Guarantee Social Satisfaction...

Figure [6.14] Bill Details Information Page Snapshot.

## 6.6.2.5 Disclaimer Page

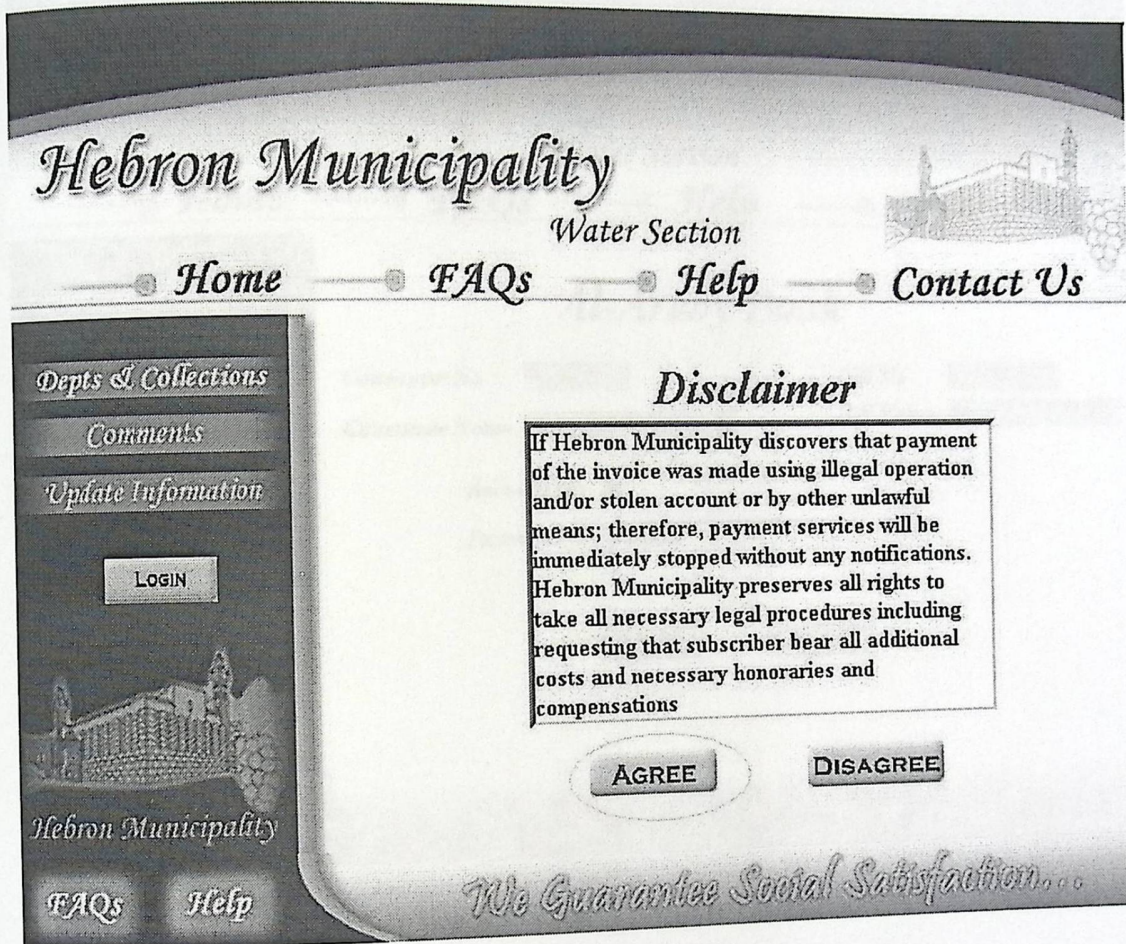


Figure [6.15] Disclaimer Page Snapshot.

## 6.6.2.5 Disclaimer Page

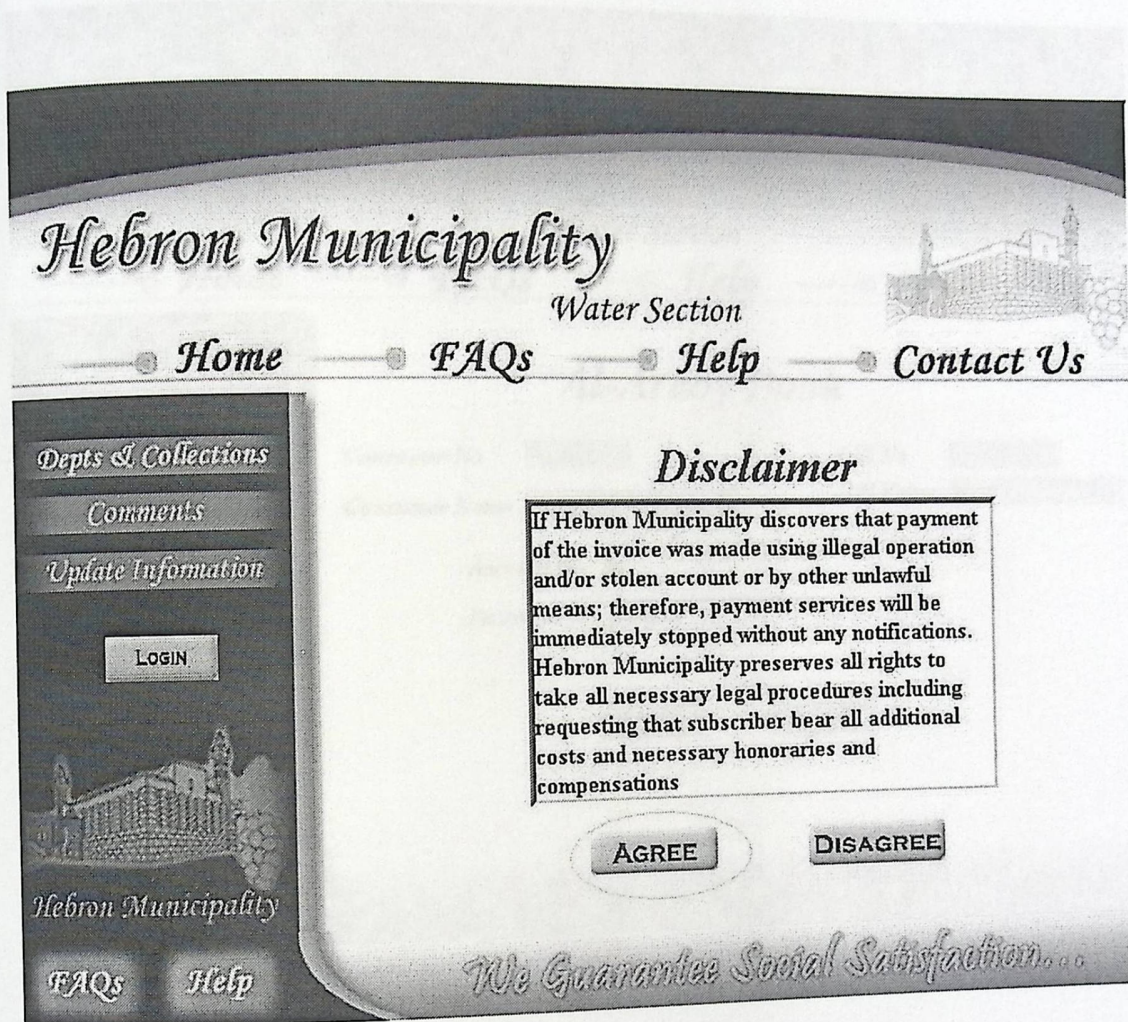


Figure [6.15] Disclaimer Page Snapshot.

6.6.2.6 Bank Account Information Page

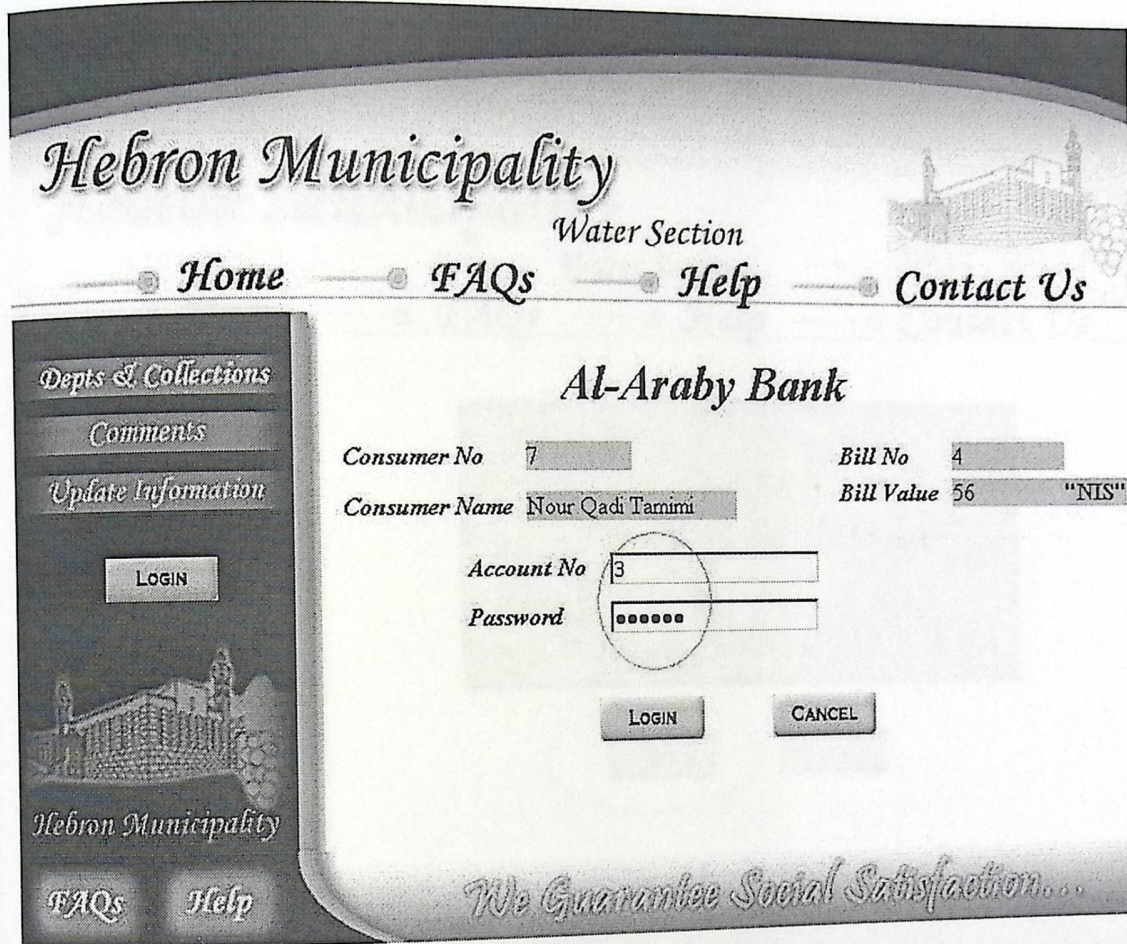


Figure [6.16] Bank Account Information Page Snapshot.

6.6.2.7 Agreement Pay Page

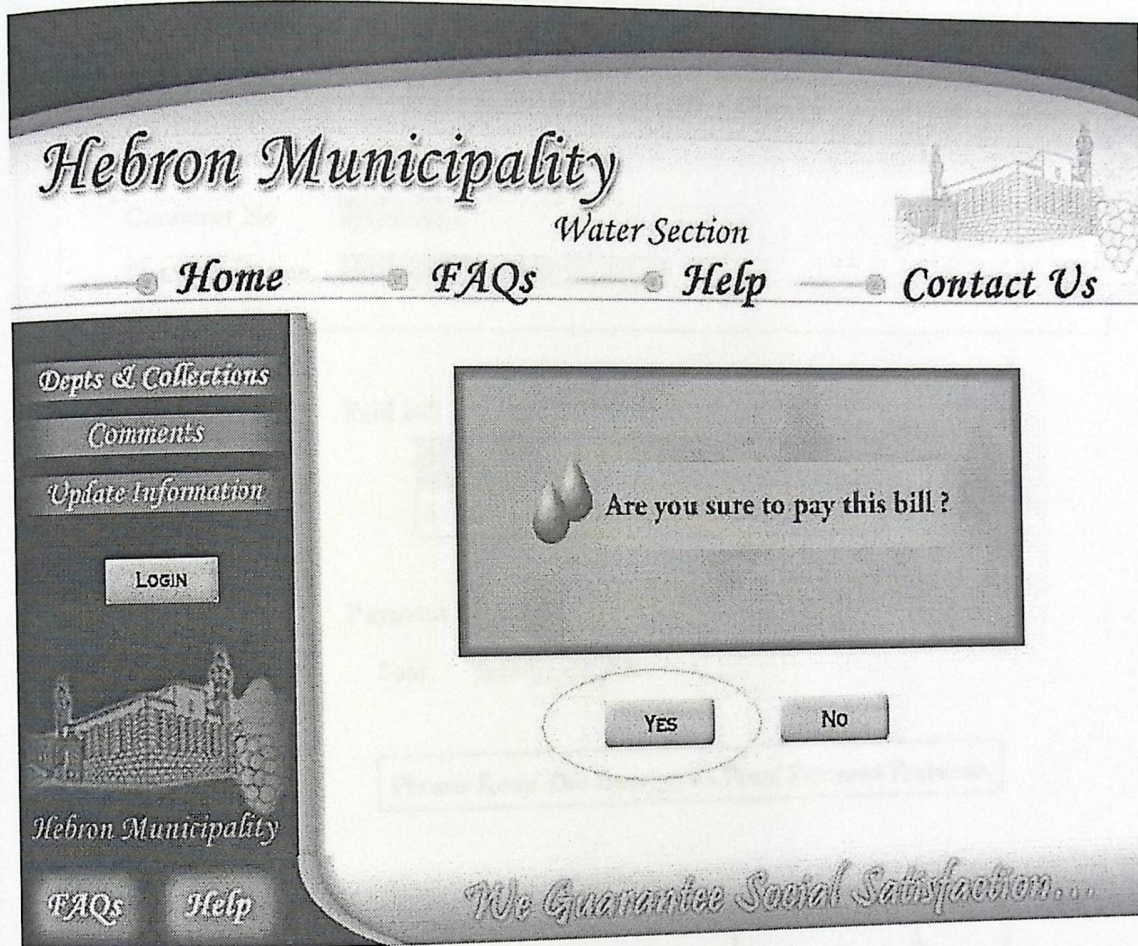


Figure [6.17] Agreement Pay Page Snapshot.



## 6.6.2.9 Bill Details before Changes

bill_no	consumer_no	issue_date	due_date	status	payment_date	counter_no
1	1	1/11/2004	10/11/2004	paid	9/11/2004	12345
2	7	1/11/2004	10/11/2004	unpaid	9/11/2004	12345
3	7	1/12/2004	10/12/2004	unpaid	<NULL>	12345
4	7	1/1/2006	10/1/2006	unpaid		12345
5	1	1/2/2006	10/2/2006	paid	12/6/2006	12345
6	7	1/5/2006	5/15/2006	paid	12/7/2006	12345
7	3	1/5/2006	5/15/2006	paid	12/7/2006	12345
8	7	6/6/2006	6/6/2006	paid	12/6/2006	12345
9	7	1/10/2004	10/10/2004	paid	10/10/2004	12345
10	1	12/12/2006	12/15/2006	paid	1/1/2006	12345
11	7	12/12/2006	12/15/2006	paid	12/7/2006	12345
12	7	10/10/2005	10/25/2005	unpaid	<NULL>	11111

Figure [6.19] Bill Details Table Before Changes Snapshot.

## 6.6.2.10 Bill Details after Changes

bill_no	consumer_no	issue_date	due_date	status	payment_date	counter_no
1	1	1/11/2004	10/11/2004	paid	9/11/2004	12345
2	7	1/11/2004	10/11/2004	unpaid	9/11/2004	12345
3	7	1/12/2004	10/12/2004	unpaid	<NULL>	12345
4	7	1/1/2006	10/1/2006	paid	1/2/2006	12345
5	1	1/2/2006	10/2/2006	paid	12/6/2006	12345
6	7	1/5/2006	5/15/2006	paid	12/7/2006	12345
7	3	1/5/2006	5/15/2006	paid	12/7/2006	12345
8	7	6/6/2006	6/6/2006	paid	12/6/2006	12345
9	7	1/10/2004	10/10/2004	paid	10/10/2004	12345
10	1	12/12/2006	12/15/2006	paid	1/1/2006	12345
11	7	12/12/2006	12/15/2006	paid	12/7/2006	12345
12	7	10/10/2005	10/25/2005	unpaid	<NULL>	11111

Figure [6.20] Bill Details Table After Changes Snapshot.

## 6.6.2.11 Bank Account Table Before Changes

account_no	name	account_total_balance	password
1	wisam	1469	123456
2	fadel	1000	123456
3	nour	4238	123456
4	samer	5000	123456
5	saeb	200000	123456
1000	hebron municipality	7241	123456

Figure [6.21] Bank Account Table Before Changes Snapshot.

## 6.6.2.12 Bank Account Table After Changes

account_no	name	account_total_balance	password
1	wisam	1469	123456
2	fadel	1000	123456
3	nour	4182	123456
4	samer	5000	123456
5	saeb	200000	123456
1000	hebron municipality	7297	123456

Figure [6.22] Bank Account Table After Changes Snapshot.

## 6.6.2.13 Bank Transaction Table Before Changes

transaction_no	account_no	beneficiary_accour	payment amount	payment_date
1	1	1000	406	12/6/2005
2	2	1000	506	12/6/2005
3	1	1000	506	12/6/2005
4	2	1000	506	12/6/2005
5	2	1000	506	12/7/2005
6	2	1000	506	12/7/2005
7	3	1000	506	12/7/2005
8	3	1000	256	12/7/2005
9	2	1000	506	12/7/2005
10	1	1000	506	12/9/2005
11	1	1000	25	1/1/2006

Figure [6.23] Bank Transaction Table Before Changes Snapshot.

## 6.6.2.14 Bank Transaction Table After Changes

transaction_no	account_no	beneficiary_accour	payment amount	payment_date
1	1	1000	406	12/6/2005
2	2	1000	506	12/6/2005
3	1	1000	506	12/6/2005
4	2	1000	506	12/6/2005
5	2	1000	506	12/7/2005
6	2	1000	506	12/7/2005
7	3	1000	506	12/7/2005
8	3	1000	256	12/7/2005
9	2	1000	506	12/7/2005
10	1	1000	506	12/9/2005
11	1	1000	25	1/1/2006
12	3	1000	56	2/1/2006

Figure [6.24] Bank Transaction After Changes Snapshot.

## 7.1 Introduction

## Chapter

## 7

## 7.2 Migration

The deployment of the system must be built by certain steps so that it work properly within its environment; the production environment has to be established, configured, and a deployment plan of migrating the new system. During all constraints and risks in the process of migration of the new system. Toward deploying and migrating to the new system we describe here the steps that must be done.

## The Content:

1. Establish the production environment.
    - 7.1 Introduction.
    - 7.2 Migration.
    - 7.3 Maintaining the system.
- The steps of repairing the system are described in chapter two (system recovery). The 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.

## Deciding to deploy the new system:

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

## 7.1 Introduction

The maintenance chapter describes how to start working with the system; the establishment of the environment that the system will work in, what is the process of deployment, and the maintenance plan.

## 7.2 Migration

The deployment of the system must be built 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 constraints 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:

### 1. 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.

### 2. Deciding to deploy the new system:

A decision of the deployment of the new system must be taken according to the plan of the deployment and 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.

### 3. **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:**

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 2003].

### 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 software, all what it needs is the production environment that was described earlier in this chapter.

## Conclusion and Recommendations

### The Contents:

#### 4.1. Introduction.

#### 4.3. Conclusion.

#### 4.2. Recommendations.

## Chapter

# 8

# Conclusion and Recommendations

The Content:

8.1. Introduction.

8.3. Conclusion.

8.2. Recommendations.

### 8.1 Introduction

In the Conclusion and Recommendations 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 system is work beside the traditional system, and both choices are available to consumer to select the suitable method to pay his bill.
- In the other hand the web-based systems reduced the time and efforts for both consumer and E-Service provider “Hebron Municipality”, and increase the efficiency of service during the payment process.

### 8.3 Recommendations

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

1. Implement this system in the real environment.
2. Integrate our system with real web services like Bank Account system to enable our system to check the Bank Account information validity, and currency rate exchanger.
3. Inform the consumer with new bills via consumer e-mail accounts at time.