

Palestine Polytechnic University



College of Engineering & Technology
Electrical & Computer Engineering Department

Software Report Project
**Computerized System for AL-IHSAN Charitable
Society**

Submitted to the Department of Electrical and Computer Engineering as
Software Graduates Project

Prepared By:

SAMI SALAMIN
WASEEM AZZAM
MUNEER ATAWNEH

Supervised By:

Dr. NABIL ARMAN

Feb - 2005

د. نائل عرمان
[Handwritten signatures and initials]



Abstract

This system is to be presented for the AL-IHSAN charitable society as a human work for this society that this society has many problems and difficulties in using the manual mode.

This system will be a Database system for this society to help them in there work.

هذا النظام سيقدم لجمعية الاحسان الخيرية كعمل انساني للجمعية حيث ان هذه الجمعية تعاني الكثير من المشاكل و المصاعب باستخدام النظام اليدوي و هذا النظام (اليدوي) مستخدم من بدايات الجمعية الى يومنا الحالي.

هذا النظام عبارة عن نظام محوسب لقواعد البيانات المستخدمة في الجمعية لمساعدتهم في اعمال الادارة للجمعية مع العلم اننا سيقوم بدراسة وتحليل النظام كاملا ولكن ما سيتم برمجته و تمامه كاملا هو قسم واحد.

Table of Contents

Chapter One: Introduction

1.1	OVERVIEW	1
1.2	AL-IHSAN SOCIETY.....	1
1.3	OVERVIEW OF VB.NET TECHNOLOGY.....	2
1.4	SQL SERVER.....	3
1.5	OVERVIEW OF THE SYSTEM.....	3
1.5.1	SYSTEM OBJECTIVES	4

Chapter Two: System Planning

2.1	INTRODUCTION.....	5
2.2	PROJECT ORGANIZATION.....	5
2.2.1	HUMAN RECOURSES.....	5
2.2.2	ALLOCATION OF ROLES OF SYSTEM DEVELOPERS	5
2.3	RISK ANALYSIS.....	6
2.4	FEASIBILITY STUDY.....	7
2.4.1	TECHNICAL FEASIBILITY	7
2.4.2	ECONOMICAL FEASIBILITY.....	8
2.4.2.1	Development Requirements.....	8
2.4.2.1.1	Hardware Development Resources.....	8
2.4.2.1.2	Software Development Resources.....	9
2.4.2.1.3	Human Development Resources	9
2.4.2.1.4	Other Requirements.....	10
2.4.2.2	Operational Requirements	10
2.4.2.2.1	Hardware Operational Resources.....	10
2.4.2.2.2	Human Development Resources	11
2.4.3	COST ESTIMATION.....	11
2.4.3.1	Development Costs	11
2.4.3.2	Implementation (Operational) Costs	12
2.4	PROJECT SCHEDULING.....	14

Chapter Three: Requirements Analysis

3.1	INTRODUCTION.....	16
3.2	REQUIREMENT DEFINITION.....	16
3.2.1	FUNCTIONAL REQUIREMENTS.....	16

3.2.1.1 System Requirements.....	16
3.2.1.2 User Requirements	17
3.2.2 NON FUNCTIONAL REQUIREMENTS.....	17
3.2.2.1 System Non Functional Requirements	17
3.2.2.2 User Non Functional Requirements.....	18
3.2.2.3 Development and Maintenance Nonfunctional Requirement	18
3.3 REQUIREMENTS SPECIFICATIONS	19
3.3.1 FUNCTIONAL REQUIREMENTS	19
3.3.1.1 System Requirements.....	19
3.3.1.2 User Requirements	20
3.3.2 NON FUNCTIONAL REQUIREMENTS.....	21
3.3.2.1 System Non Functional Requirements	21
3.3.2.2 User non functional requirement.....	21
3.3.2.3 Development and maintenance nonfunctional requirement	22

Chapter Four: System Analysis

4.1 INTRODUCTION	23
4.2 SYSTEM CONTEXT	23
4.3 INPUT DATA	24
4.3.1 THE RESIDENT SUB SYSTEM.....	25
4.3.1.1 User Name and Password	25
4.3.1.2 Adding New Resident	26
4.3.1.3 Adding New Sponsor	29
4.3.2 THE PUBLIC RELATIONS SUB SYSTEM	29
4.3.2.1 User Name and Password	29
4.3.2.2 Adding New Project	30
4.3.2.3 Adding external supported institution.....	30
4.3.2.4 Adding local institution	31
4.3.2.5 Adding Character	31
4.3.3 EMPLOYEES AFFAIRS SUBSYSTEM	32
4.3.3.1 User Name and Password	32
4.3.3.2 Adding New Employee.....	32
4.4 OUTPUT DATA	33
4.4.1 THE QUERY OUTPUT.....	33
4.4.2 THE DATA STORED IN THE DATABASE.....	33
4.5 ER DIAGRAM.....	33
4.5.1 RESIDENT ER DIAGRAM	33
4.5.2 THE PUBLIC RELATIONS DIAGRAM.....	35
4.5.3 THE EMPLOYEE AFFAIRS DIAGRAM	36

Chapter Five: System Design

5.1 INTRODUCTION	37
5.2 OVERVIEW	37

4.4 PROCESSES.....	38
4.4.1 RESIDENT SUBSYSTEM PROCESSES	38
4.4.1.1 Add New Resident Flow Chart.....	38
4.4.1.2 Delete, Update Resident	40
4.4.1.3 Add, Delete, Update Sponsor.....	42
4.4.1.4 Query about resident or sponsor.....	43
4.4.1.5 Change Password.....	44
4.4.2 EMPLOYEES AFFAIRS SUBSYSTEM PROCESSES.....	44
4.4.2.1 Add New Employee Flow Chart.....	45
4.4.2.2 Delete, Update Employee	46
4.4.2.3 Query about Employee.....	47
5.3 DATABASE DESIGN	48
5.3.1 MAPPING	48
5.3.1.1 Relational Mapping.....	48
5.3.1.2 Normalization	50
5.3.2 DATA DICTIONARY.....	55
5.3.3 TABLE'S DESCRIPTION.....	56
5.3.4 TABLES	57
5.3.5 THE UML CONCEPTUAL MODEL FOR THE "AL-IHSAN" DATABASE.....	68
5.4 THE DESIGN OF THE SYSTEM MODULES.....	74
5.4.1 SYSTEM AND SUBSYSTEM ARCHITECTURE	74
5.4.2 OBJECT IDENTIFICATION.....	74
5.6 THE USER INTERFACE DESIGN	78

Chapter Six: Implementation

6.1 INTRODUCTION	87
6.2 SETTING UP THE REQUIRED SOFTWARE AND HARDWARE	87
6.2.1 SETTING UP THE HARDWARE AND OPERATING SYSTEMS	87
6.2.2 SETTING UP THE SOFTWARE FOR THE SYSTEM:.....	88
6.3 DATABASE IMPLEMENTATION	88
6.3.1 DATABASE RELATIONSHIPS DIAGRAM.....	88
6.4 DATABASE AND WINDOWS FORMS INTERFACING	92
6.4.1 THE INTERFACE BETWEEN THE DATABASE AND SYSTEM MODULES DESIGN	92
6.4.1.1 The Interface Objects.....	92
6.4.1.2 The Interface Objects Attributes and Methods.....	93
6.4.2 IMPLEMENTATION EXAMPLE.....	94
6.4.2.1 ADD new resident example.....	94
6.4.2.2 The family telephone insert example	100
6.4.2.3 The Resident and Family update example.....	101
6.4.2.4 The Resident Delete Example	104
6.4.5 THE LOGIN EXAMPLE	106

Acknowledgement

It is one of the greatest experiences of our lives to see how much support and help we would find in a work like this; we would like to thank first our supervisor Dr. Nabil Arman . We would like to individually thank everyone who helped us to make this work come to life. We would not forget to thank Palestine Polytechnic University with its helpful staff .

Last but not least we would like to thank all the individuals who contributed to this system with their ideas and advice.

Chapter Seven: Testing

7.1 INTRODUCTION	109
7.2 TESTING THE INDIVIDUAL OPERATIONS.....	109
7.2.1 PASSWORD OPERATION FOR THE USER AND DEVELOPER.....	109
7.2.2 RESIDENT OPERATION.....	111
7.2.3 SPONSOR OPERATION.....	118
7.2.3 VISIT OPERATION	123
7.2.4 GENERAL INFORMATION	128
7.2.4 THE REPORTS.....	130

Chapter Eight: Installation

8.1 INTRODUCTION.....	ERROR! BOOKMARK NOT DEFINED.
8.2 PREREQUISITE.....	ERROR! BOOKMARK NOT DEFINED.
8.3 APPLICATION INSTALLATION.....	133

Chapter nine: System Maintenance

9.1 INTRODUCTION.....	137
9.2 SQL SERVER MAINTENANCE.....	137
9.3 THE .NET FRAMEWORK MAINTENANCE.....	138
9.4 THE SYSTEM APPLICATION.....	138

Chapter ten: Conclusions and Recommendations

10.1 INTRODUCTION.....	139
10.2 CONCLUSIONS	139
10.3 RECOMMENDATIONS.....	139
APPENDIX A.....	140
REFERENCES	141

Table of Tables

<i>Table (2.1) shows the risk analysis of system</i>	7
<i>Table (2.2) technical feasibility of the system compared to other alternatives</i>	8
<i>Table (2.3): shows Development hardware costs</i>	11
<i>Table (2.4): shows the Development software costs for the system</i>	11
<i>Table (2.5): shows the human costs for the system</i>	12
<i>Table (2.6): Total development costs</i>	12
<i>Table (2.7): Operational hardware costs</i>	12
<i>Table (2.8): Human Operational costs</i>	13
<i>Table (2.9): Total Operational costs</i>	13
<i>Table (2.10): Total system Operational costs</i>	13
<i>Table (2.11) the system development time scheduling</i>	14
<i>Table (5.1) The table's description for resident subsystem</i>	56
<i>Table (5.2) The table's description for Public Relations Subsystem</i>	57
<i>Table (5.3) The table's description for Employee's Affairs subsystem</i>	57
<i>Table (5.4) Resident table description</i>	58
<i>Table (5.5) Sponsors table description</i>	59
<i>Table (5.6) Part table description</i>	59
<i>Table (5.7) Family table description</i>	60
<i>Table (5.8) Telephone table description</i>	60
<i>Table (5.9) Visits table description</i>	60
<i>Table (5.10) Pledge table description</i>	61
<i>Table (5.11) Visit table description</i>	61
<i>Table (5.12) Project table description</i>	62
<i>Table (5.13) Characters table description</i>	62
<i>Table (5.14) External supported institution table description</i>	63
<i>Table (5.15) Local institution table description</i>	64
<i>Table (5.16) Informing table description</i>	64
<i>Table (5.17) Visits table description</i>	65
<i>Table (5.18) Employees table description</i>	66
<i>Table (5.19) Additional work table description</i>	66
<i>Table (5.20) The discounting table description</i>	67
<i>Table (5.21) Vacations table description</i>	68
<i>Table (5.22) The Vacations Type Table description</i>	68
<i>Table (6.1) Interfaces objects of the ADO.NET</i>	93

Table of Figures

Figure (2.1) time planning	15
Figure 4.1 The main sub system	23
Figure (4.2) System Context Diagram	24
Figure (4.3) Login control and dataflow	25
Figure (4.4) Add new resident (step1) dataflow	26
Figure (4.5) Add new resident (step2) dataflow	27
Figure (4.6) Add new resident (step3) control and dataflow	28
Figure (4.7) Add new resident (step4) dataflow	28
Figure (4.8) Add new sponsor dataflow	29
Figure (4.9) Add new project control and dataflow	30
Figure (4.10) Adding an external supported institution control and dataflow	30
Figure (4.11) Adding a local institution control and dataflow	31
Figure (4.12) Adding new character control and dataflow	31
Figure (4.13) The data flow of adding new employee	32
Figure (4.14) Resident ER diagram	34
Figure (4.15) Public relations ER diagram	35
Figure (4.16) Employee affairs ER diagram	36
Figure (5.1) Add new resident flow chart	39
Figure (5.2) Delete new resident flow chart	41
Figure (5.3) Add new sponsor flow chart	42
Figure (5.4) Query flow chart	43
Figure (5.5) Change password for resident	44
Figure (5.6) Add new employee flow chart	45
Figure (5.7) Delete/Update the employee flow chart	46
Figure (5.8) The query about employee	47
Figure (5.9) conceptual model for the resident sub system	69
Figure (5.10) Conceptual model for the public relations subsystem	71
Figure (5.11) Conceptual model for employee's affairs subsystem	73
Figure (5.12) AL-IHSAN society sub system	74
Figure (5.13) The object identification for resident subsystem	75
Figure (5.14) The object identification for public relations subsystem	76
Figure (5.15) The object identification for employee affairs subsystem	77
Figure (5.16) Add New Resident Interface Screen	78
Figure (5.17) Delete Resident Interface Screen	78
Figure (5.18) Update Resident Interface Screen	79
Figure (5.19) Query Resident Interface Screen	79
Figure (5.20) Add Sponsor Interface Screen	79
Figure (5.21) Delete Sponsor Interface Screen	80
Figure (5.22) Add Sponsor Interface Screen	80
Figure (5.23) Query Sponsor interface Screen	80
Figure (5.24) Report Generator Interface Screen	81
Figure (5.25) Add Visit Interface Screen	81
Figure (5.26) General Information Interface Screen	81
Figure (5.27) The informing department	82
Figure (5.28) The external supported institution department	83

Figure (5.29) The project department	83
Figure (5.30) The Local Institution Department.....	83
Figure (5.31) The administrator tasks screen.....	84
Figure (5.32) The Add Employee Screen	84
Figure (5.33) The Update Employee Screen	85
Figure (5.34) The Delete Employee Screen	85
Figure (5.35) The Search Employee Screen.....	86
Figure (6.1) The data relations diagram for resident subsystem.....	89
Figure (6.2) The data relations diagram for the public relations subsystem.....	90
Figure (6.3) The data relations diagram for employee's affairs subsystem.....	91
Figure (7.1) Invalid password.....	110
Figure (7.2) Invalid Old Password	110
Figure (7.3) Mismatch the two new passwords.....	110
Figure (7.4) Valid old and two new passwords.....	111
Figure (7.5) Not filling all required data.....	112
Figure (7.6) Invalid date	113
Figure (7.7) All required data are filling und valid	113
Figure (7.8) After adding is successful you add the valid phone #.....	114
Figure (7.9) Enter the existing resident number	114
Figure (7.10) No number is entered.....	115
Figure (7.11) Valid number.....	115
Figure (7.12) Invalid resident number.....	116
Figure (7.13) Resident exists	116
Figure (7.14) No number is entered.....	117
Figure (7.15) Invalid number (Resident not exists).....	117
Figure (7.16) Valid number (Resident is exists).....	118
Figure (7.17) All data are entered is correct	119
Figure (7.18) The sponsor number is exists	120
Figure (7.19) All required data not entered.....	120
Figure (7.20) Valid sponsor number	121
Figure (7.21) Invalid update	121
Figure (7.22) Invalid update	122
Figure (7.23) Invalid search.....	122
Figure (7.24) valid search.....	123
Figure (7.25) All data are entered is correct	124
Figure (7.26) The number is invalid.....	125
Figure (7.27) Valid number and data.....	125
Figure (7.28) Valid number.....	126
Figure (7.29) The data is not enough	126
Figure (7.30) Correct complete visit.....	127
Figure (7.31) Incorrect complete visit.....	127
Figure (7.32) General information form.....	128
Figure (7.33) Resident in the children department	128
Figure (7.34) Resident in the men department.....	129
Figure (7.35) Resident in the girl department.....	129
Figure (7.36) Sponsors information.....	129
Figure (7.37) Residents how have sponsor	129
Figure (7.38) Residents how haven't sponsor	130
Figure (7.39) The report main form.....	130
Figure (7.40) The resident with family report.....	131

<i>Figure (7.41) The resident with sponsor report.....</i>	<i>131</i>
<i>Figure (8.1) Install the application.....</i>	<i>134</i>
<i>Figure (8.2) Attaching the Database file to SQL Server.....</i>	<i>135</i>
<i>Figure (8.3) Configure the connection string.....</i>	<i>136</i>

Chapter One

Introduction

Overview

AL_IHSAN Society

Overview of VB.NET Technology

SQL Server

Overview of the system

1.1 Overview

By entering the new millennium we have witnessed many changes in the world, perhaps the most noticed is the tendency and huge increase of using the computer system in work.

One of the most important activities that may need the computerized system is for storing and retrieving data in fast and economical fashion.

The database system is one of the most widespread systems that used to store, manipulate and retrieve data in little time and less effort.

Our humane work is to be granted to AL-IHSAN society for handicap, after visiting this society and studying its need, we found that this society need a database system.

We will use the SQL server as a DBMS, for creating the database, and VB.NET to develop the front end application.

1.2 AL-IHSAN Society

This society is one of the charitable societies in our country that takes care, treats and helps handicap children and olds.

After several visits to this society and meets the society manager; we understood the system and manage to specify its requirements.

1.3 Overview of VB.NET Technology

✓ *What is the .NET Technology?*

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications.

✓ *Why choosing the VB.NET Technology?*

Streaming from the flexibility and the ease to program with the .NET Framework as a software development environment and the powerfulness of its database interactive system based applications; the work team chose this technology to develop the system which is a database access by a distributed system.

✓ *How does it work?*

This is an object oriented technology that contains different classes of developing frameworks such as web applications and Windows applications. The windows applications consist of windows forms that support the connection with databases from different resources such as MS SQL server or MS Access database management system. This database system is built on the connection between the windows forms and a database built by using the MS SQL Server 2000.

✓ The .NET Framework aims:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

1.4 SQL Server

Our system can be classified as a mid-level system. Our team chooses the Microsoft SQL server for database implementation, because VB.NET and .NET FRAME WORK can be integrated easily with SQL server, SQL server supports multiple users it is also scalable, efficient, and secure DBMS.

1.5 Overview of The System

This database system is designed and implemented as a human work for AL-IHSAN society.

This system is used to save data, manipulate and retrieve data about the handicap patient, workers, nursemaid etc...

The previous system that the system used is contain a lot of separate files that used to stored the data and many office that may visited many times by the employees to do the needed work

This system will be designed in a single-user mode and we may extend this system to be used as a multi-user mode, because the society has a local area that is to be setup in the near future.

1.5.1 System Objectives

Our system should perform the following primary needs:

- provide storing data
- provide retrieve data
- provide data search
- provide updating data
- provide deleting data
- provide report generating and printing
- provide data backup

Chapter Two

System Planning

Introduction
Project Organization
Risk Analysis
Feasibility Study
Project Scheduling

2.1 Introduction

This chapter demonstrates the system planning and resources required to create it. The planning phase is an early stage that includes estimation and scheduling of the system resources and forward coming processes. Since it is a process of estimation it is debatable and subject to changes such as changes in cost or time schedules for some system development phases. This chapter includes the following topics:

2.2 Project Organization

Explain the team work and the relations between team members

2.2.1 Human Recourses

System development work personnel and their supervisor:

1. SAMI SALAMIN
2. MUNEER ATAWNEH
3. WASEEM AZZAM

2.2.2 Allocation of Roles of System Developers

There is no constant situation for any team member job, because we will do the whole work together. But in general we put the allocation of roles of system developers as follows:

1. **Leader:** responsible of planning, scheduling and controlling flow of system development processes.

2. **Programmer:** responsible of the system programming, implementation testing so he must have enough experiences in the VB.NET development environment.
3. **Software engineer:** responsible for the documentation and tracing of the development stages of the software.
4. **Interface designer:** responsible of the form design and user interface of the system.
5. **Technical administrator:** responsible for analysis, monitoring, testing and maintaining the system.

2.3 Risk analysis

<i>Objectives:</i>	<ul style="list-style-type: none"> • A Database System will be used to store, retrieve and manipulate data.
<i>Constraints:</i>	<ul style="list-style-type: none"> • Delivering the system in three months. • Maintain the flexibility of the system. • Maintain the highest level of security. <p>Acceptance by the audience.</p>
<i>Alternatives:</i>	<ul style="list-style-type: none"> • Acquisition of a generic project from the market. • Creation and development the project by helping professional people.
<i>Risks:</i>	<p>Nonfunctional risk:</p> <ul style="list-style-type: none"> • Required time more than the estimated time. • New cost may appear. • Technology, which is used in system, may be changed. <p>Operational risk:</p> <ul style="list-style-type: none"> • Incompatibility between systems with environment. <p><i>Functional risks:</i></p> <ul style="list-style-type: none"> • Such as requirements changes. <p>Operational risk:</p>

	<ul style="list-style-type: none"> • Some applications don't perform as estimated such as the system interface is not comfortable or acceptable by users. • Appearance of new requirement after or during development stage.
<i>Risk resolution:</i>	<ul style="list-style-type: none"> • The system must work in environments that have fewer specifications. • Study the planning process and take right steps to deliver the system in the desired state. • Study all the requirements and identify the importance of each one. • Training the employees to operate this system. • Apply the scheduling table
Results:	<ul style="list-style-type: none"> • Creation of a system by using VB.NET and SQL server 2000.

Table (2.1) shows the risk analysis of system

2.4 Feasibility study

This section is an estimation of the economical and technical costs of the system compared with other alternatives such as traditional handy work, and computerized based system.

2.4.1 Technical feasibility

Here we will provide a comparison between the handy work system that the society uses and the computerized system that we will make it:

Operation	Manual mode	Computerized mode
Add new resident	Difficult	Easy
Update data of existence resident	Difficult	Easy
Delete resident	Easy	Easy
Generate report	Difficult	Easy
Search data	Difficult	Easy

Table (2.2) technical feasibility of the system compared to other alternatives.

2.4.2 Economical Feasibility

This section describes the cost estimation of the system development and implementation phases, including the hardware costs, software costs and human resources costs. The costs are categorized in costs of development and costs of operation.

2.4.2.1 Development Requirements

The development requirements include the hardware resources, software resources, human requirements and other requirements such as books that are used for the development phase.

2.4.2.1.1 Hardware Development Resources

It predicts minimum hardware requirements needed to develop this system. The hardware requirements are fully satisfied for our project since we have the following available characteristics of the development work station:

1. This project requires a PC that can run Windows XP professional. A PC with a Pentium 3 or Pentium 4 processor can play this role. But to get faster execution and operational speed, we select to work on a PC with Pentium 4 processor.

A typical PC specification can be:

- Pentium 4 2800 MHz clock speed
 - 512 MB RAM
 - 40 GB hard disk
 - 17" LG monitor
 - Keyboard + Mouse
2. Laser printer
3. UPS 1k VA.

2.4.2.1.2 Software Development Resources

The PC should have the following operating system and software:

- Microsoft XP professional
- Microsoft Visual Studio.Net® Including ADO.NET
- DBMS (Microsoft SQL Server2000)

2.4.2.1.3 Human Development Resources

We anticipate that we will need three months of work and four hours of work in each day and four US dollars for each hour so:

Total human resources cost= (# of month)*(# of weeks)*(# of days)*(# of hour in a day)*(hour cost) =

$3 * 4 * 7 * 4 * 4 = 1344\$$ for each team member

Total human resources cost for the team= $1344 * 3 = 4032\$$ for the whole team

2.4.2.1.4 Other Requirements

We need to obtain some books and references to complete this system. These books are related to the subjects of VB.NET technology, ADO.NET, SQL server and object-oriented design. These books are mentioned in the references list.

2.4.2.2 Operational Requirements

The implementation requirements are divided into the hardware requirements, software requirements, human requirements and additional requirements.

2.4.2.2.1 Hardware Operational Resources

We will consider the system operational requirement as following:

1. computer: Any computer can used to run this system starting from Pentium with 233MHZ speed as minimum; we suggests the following computer specification
 - Pentium 4 2800 MHz clock speed
 - 512 MB RAM
 - 40 GB hard disk
 - 17" LG monitor
 - Keyboard + Mouse

2. UPS 1k VA

2.4.2.2.2 Human Development Resources

- Administrator of this system
- User

2.4.3 Cost Estimation

This section lists the estimated costs for this system including the development costs and the implementation costs.

2.4.3.1 Development Costs

Hardware Costs

The following table shows the hardware cost

Hardware Component	Cost
A PC with Pentium 4- 2800 MHz	700 \$
Printer HP laser	500 \$
UPS 1k VA	700 \$
Total	1,900 \$

Table (2.3): shows Development hardware costs

Software Costs

The following table shows the software cost.

Software Component	Cost
Windows XP professional	300 \$
Microsoft Visual Studio.net	800 \$
DBMS (MS SQL Server 2000)	700\$
Total	1800\$

*Table (2.4): shows the Development software costs for the system***Human Resources Costs**

The following table shows the human resources cost.

NO.	Work Hours/developer	Cost /hour	Total human cost
3 developers	(28 hours)*(4 weeks)*(3 months)= 336 hours	4 \$	336 hours*(3 developers)*(4 \$)= 4,032 \$

*Table (2.5): shows the human costs for the system***Total Development Costs**

The following table shows the total cost of the system.

HW. Dev. Costs	SW. Dev. Costs	Human Dev. Costs	Other Dev. Req.	Total
1,900 \$	1800\$	4,032 \$	0.0\$	SW+Human=5832\$

*Table (2.6): Total development costs***2.4.3.2 Implementation (Operational) Costs****Hardware Operational Costs**

The following table shows the hardware operational costs.

Hardware Component	Cost
PC	700 \$
UPS 1k VA	700 \$
Total	1,400 \$

Table (2.7): Operational hardware costs

Human Operational Costs

The system needs one administrator (user) to run and maintain this system.

Human resource	Cost/ year
One administrator	(500 \$ /month)*(12 months)* 6,000 \$

Table (2.8): Human Operational costs

Total Operational Cost

The following table shows the total operational costs.

HW. Op. Costs	Human Op. Costs	Other Op. Req. Costs	Total Op. Cost
1,400 \$	6,000 \$	600\$	8,000 \$

Table (2.9): Total Operational costs

Total System Cost

The following table shows the total system cost.

Dev. Costs	Op. Costs	Total system cost
5,832 \$	8,000 \$	13,832\$

Table (2.10): Total system Operational costs

2.4 Project Scheduling

Activities(task)	symbol	Duration in week	Dependences
Get the project idea and discuss it with the supervisor	T1	1	-
Team planning and put team roles	T2	1	-
Feasibility study and Project planning	T3	1	-
Requirements definition	T4	1	T3
Requirement analysis	T5	1	T4
Requirements specification	T6	1	T4,T5
Design	T7	2	T5, T6
Programming	T8	3	T6
Modify system interface design	T9	1	T7
Implementation and testing	T10	3	T6, T7, T8
Documentation	T11	14	T3, T4, T5, T6, T7, T8, T9, T10
Finishing to release the system	T12	1	T11
Total time in weeks		16	

Table (2.11) the system development time scheduling.

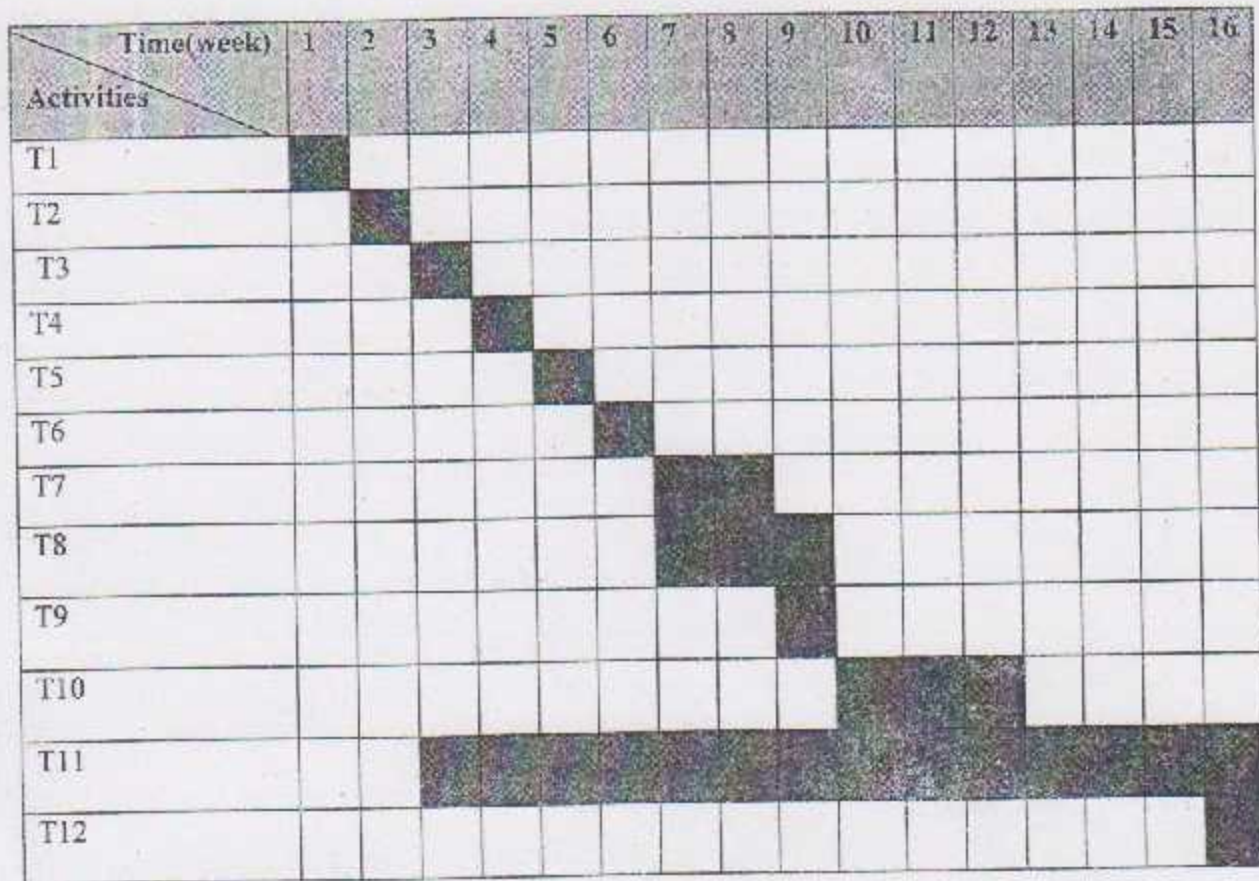


Figure (2.1) time planning

Chapter Three

Requirement Analysis

Introduction

Requirement Definition

Requirements Specifications

3.1 Introduction

This chapter describes the requirements of this system. These requirements are divided into two main parts: functional requirements and non-functional requirements. This is the process of requirements specification.

3.2 Requirement Definition

In this section we will talk about the definition of the whole system requirement as follows:

3.2.1 Functional Requirements

These are statements of services the system should provide, how the system should react to the user inputs and how system should behave in particular situation.

3.2.1.1 System Requirements

This system should be able to provide these requirements:

- The ability to Input and store data
- The ability to Retrieve and change data
- The ability to Delete data
- The ability to Generate reports

- The ability to Print data

3.2.1.2 User Requirements

The system should be able to provide the following user requirements:

- The system should be easy to use
- The system should be user friendly
- The system should be clear and one world!
- The system should have a GUI interface
- The system should use help messages

3.2.2 Non Functional Requirements

The nonfunctional requirement can be classified into:

3.2.2.1 System Non Functional Requirements

The system must provide some requirement that is not a functional requirement such as:

- The system should be Secure to prevent unauthorized access
- The system should be Reliable

- The system should be efficient

3.2.2.2 User Non Functional Requirements

The nonfunctional user requirement can be as following:

- The system should be easy to use
- The system should provide a good user interface

3.2.2.3 Development and Maintenance Nonfunctional Requirement

The system must provide some of maintenance and development nonfunctional requirement as following:

- The system should be easy to maintain
- The system should be easy to develop
- The system should provide external information for maintenance and development

3.3 Requirements Specifications

In this section we will take the requirement with some details as the following:

3.3.1 Functional Requirements

3.3.1.1 System Requirements

➤ *The ability to input and store data:*

The main objective for this system is to solve the hard work that the worker found in the paper-based system that was used to store the data into files, this operation is time consuming and need good knowledge of storing and registration into this system.

For this our system will try to solve this problem as follows:

- The system must be able to add new resident to the resident part
- The system must be able to add new employee to the society
- The system must be able to add new sponsor
- The system must be able to store information about the project that the society works on
- The system must be able to store information about the people and other societies that support the project

➤ *The ability to Retrieve and change data*

The computerized system that we will created must provide the ability to retrieve the stored data using query and the system must provide authorized change and update for the stored data in this system.

➤ *The ability to Generate reports*

It's one of the most important facilities that the modern computerized systems must include. The system must generate reports about any resident or worker in the system. The system must provide direct reports about stored data and the system must give the capability to print this data.

3.3.1.2 User Requirements

➤ *The system should be easy to use*

The system must be very easy to use.

➤ *The system should be clear and unambiguous*

The system must use clear names and clear language. It should use a natural language for the user. It may not have any difficulty for users to work with. The system must provide a graphical user interface (GUI) to facility the interaction between the user and the application.

➤ *The system should use help messages*

The system must provide help messages to provide information for the user about the steps or the work in this system, like giving alerts when the user try to delete data from the system.

3.3.2 Non Functional Requirements

3.3.2.1 System Non Functional Requirements

➤ *The system should be Secure*

The system is a stand-alone system and for this the security that we suggest is not high level security but we will use a login user name and password and we will give the developer another login as password for fix some properties such as change the path of the database table or make a pack up for the Database.

➤ *The system should be Reliable*

The system must be reliable for the society and the user if not the system will not be helpful and they will continue in the old fashion

➤ *The system should be Out of conflict*

The system may use more than one table at the same time, and may change the data in any table. The system must maintain the database consistency and should be out of conflict in the stored data.

3.3.2.2 User non functional requirement

➤ *The system should be comfortable to use*

The user interface should be simple that the user can react in an effective way with the system.

3.3.2.3 Development and maintenance nonfunctional requirement

- *The system should be easy to maintain*

As any program or system the system may have a problem that can't be discovered during development and may not appear during testing. The system must be easy to maintain.

- *The system should be easy to developed*

- *The system should provide external information for maintenance*

If any error happened during runtime, the application must generate a message that can used to maintain the system.

Chapter Four

System Analysis

Introduction
System Context
Input Data
Processes
Output data
ER Diagram

4.1 Introduction

This chapter describes the approach that is used in the system analysis, based on the system requirement specifications that will be used in the system design.

4.2 System Context

In this section, we explain the sub systems that the internal parts have, and the main sub systems that can be changed from the manual way to be computerized, and we will show the system context.

The AL-IHSAN society has many subsystems, after we study the system and meet the manager we found that only three subsystems can be implemented to be computerized.

The subsystems that can be taken to be computerized are shown in the figure below.

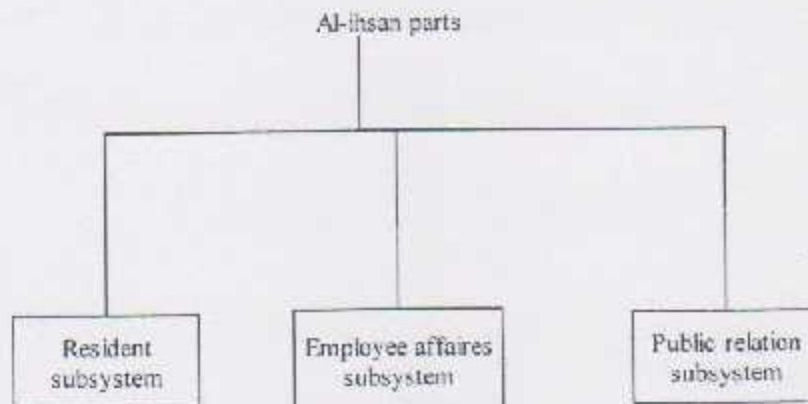


Figure 4.1 The main sub system

The figure below figure(4.2) this diagram shows an overview of the system and its relationships with other systems. This diagram shows only the systems names without details about the relationships.



Figure (4.2) System Context Diagram

4.3 Input Data

This section will describe the input data movement for the system. Only users of the system can input data to be stored, or delete or retrieve data and no unauthorized access can be found. The data movement can be separated by each subsystem as following:

4.3.1 The Resident Sub System

This subsystem is for resident data and related data include sponsor and family and visits... etc.

The data flow that this subsystem may include is defined as followed:

4.3.1.1 User Name and Password

One of the requirements of the system is to prevent the unauthorized access to the system; this functionality can be achieved by using the login process to the system

The login to the system will be classified into two modes:

- 1- The user login
- 2- The administrator/developer login

The login control and data flow process will be as the following:

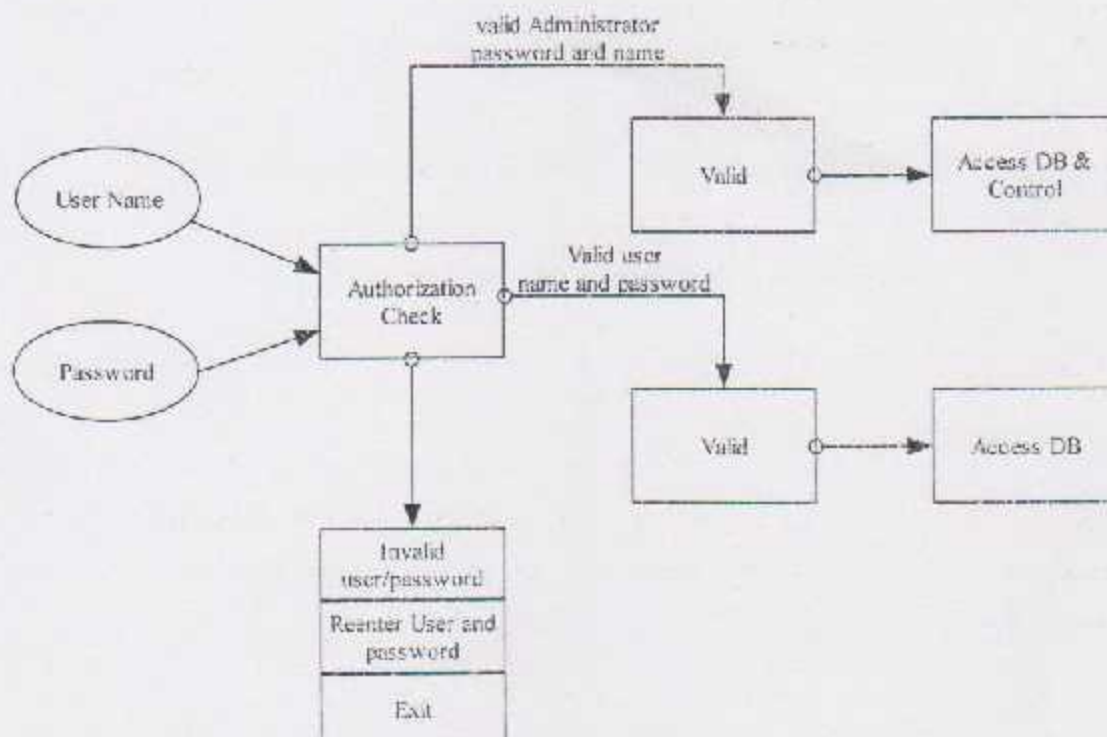


Figure (4.3) Login control and dataflow

4.3.1.2 Adding New Resident

To add a new resident to the system there are many fields that should be completed correctly to perform this operation correctly, this operation will take many steps and any error in these steps will stop the next steps.

The adding resident control and data flow process will be as the following:

Step 1:

This is the first step to add a new resident to the system, this step we will enter the resident themselves information like name, sex, birth of date and ...etc.

This step we will take check on the required field to be completed, check for data type and ranges if error in data is found this step we will ask to reenter the error field, if no error found the data will be stored into the Data Base.

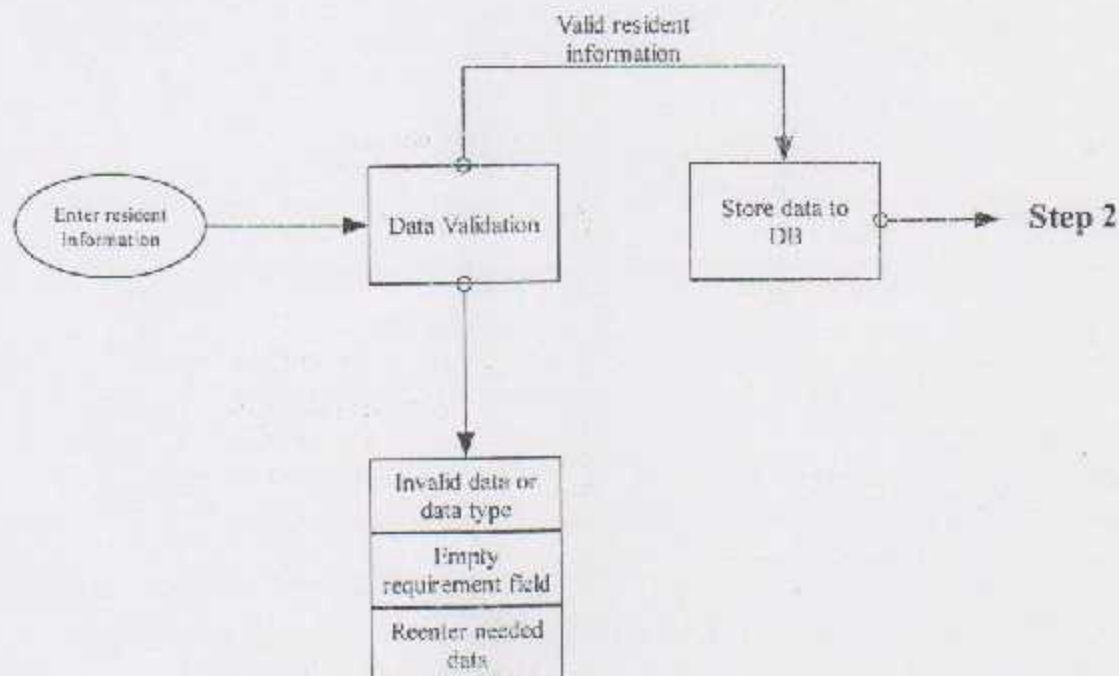


Figure (4.4) Add new resident (step1) dataflow

Step2:

This step will continue to enter the resident information; but this step will take the family information.

This step checks the required field to be completed, check for data type and ranges if error in data is found this step will ask to reenter the error field, if no error found the data will be stored into the Data Base.

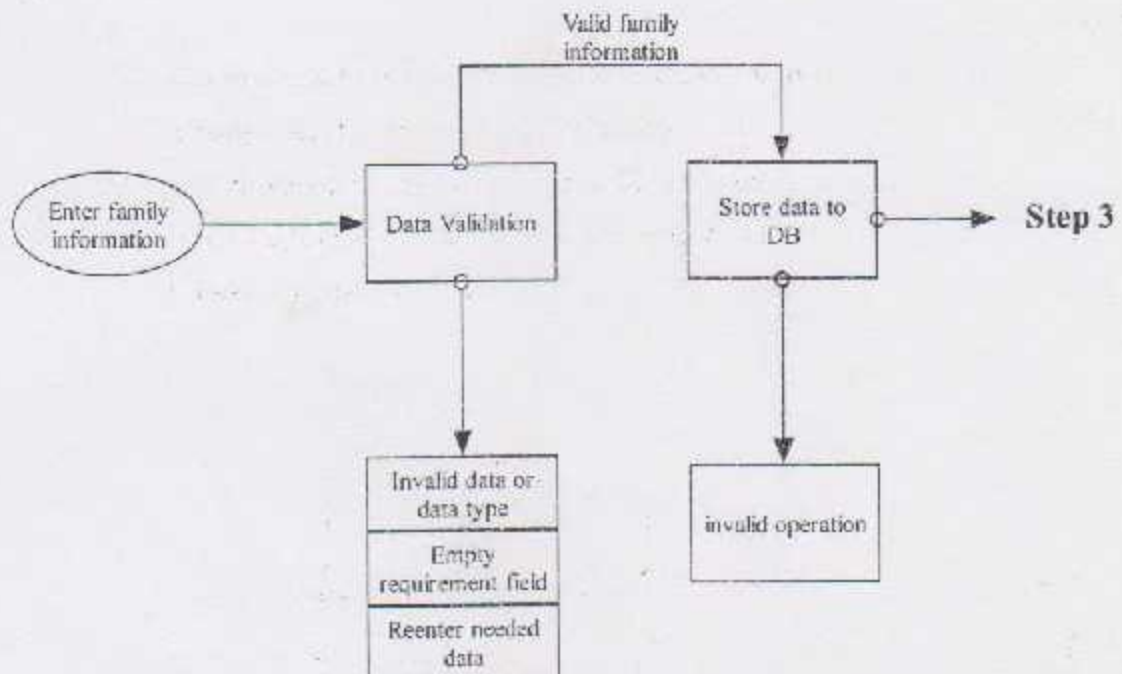


Figure (4.5) Add new resident (step2) dataflow

Step3:

This step is store the family telephone number, this operation will accept at least two mismatch telephone numbers, and up to three telephone number in any format with any separator character.

This operation can be skipped and can be updated in the update operation.

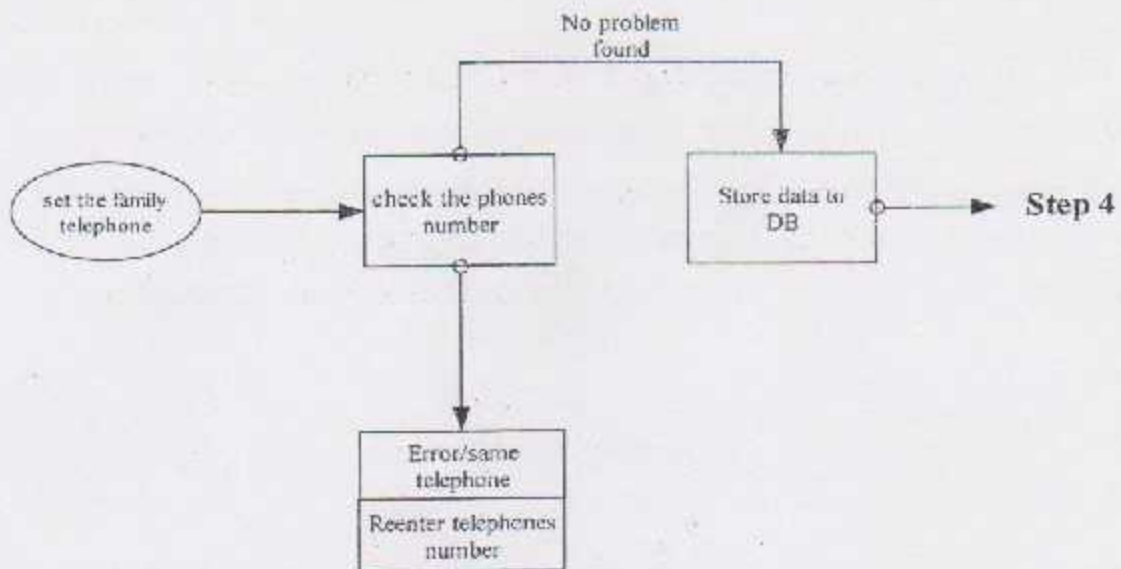


Figure (4.5) Add new resident (step3) control and dataflow

Step4:

This step is for choose sponsor (if needed); the society makes a social scanning for the new resident to find the financial situation for the resident family, and then the society find the total payment and determined if the resident needed to have sponsor and determined the pledge money.

This operation can be skipped and can be updated in the update operation

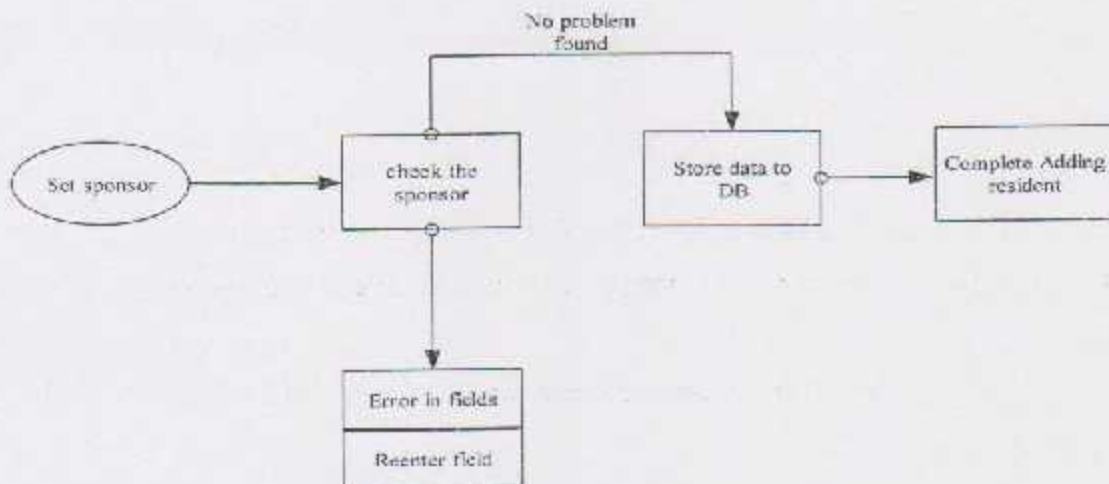


Figure (4.7) Add new resident (step4) dataflow

4.3.1.3 Adding New Sponsor

To add new sponsor a few data should be entered as followed:

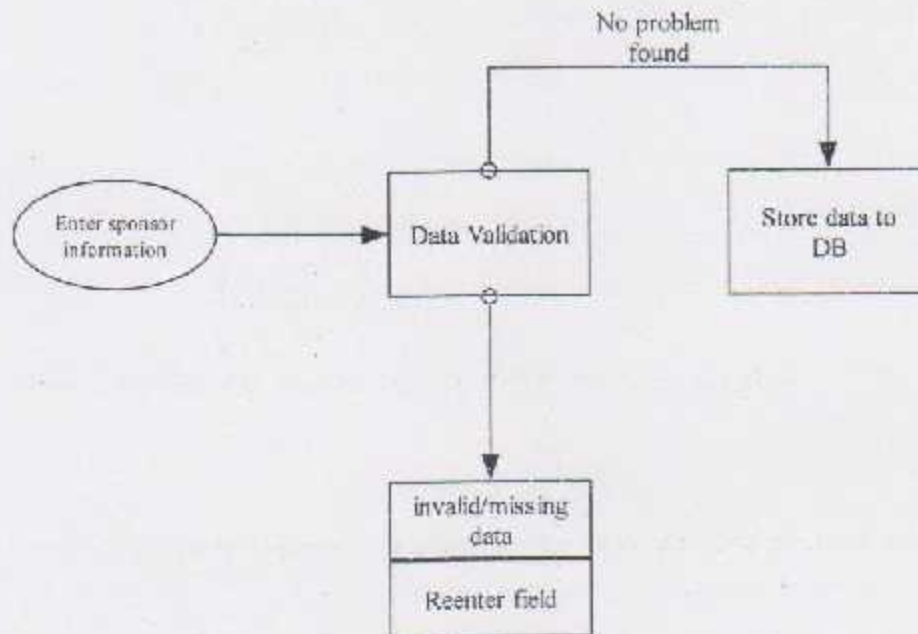


Figure (4.8) Add new sponsor dataflow

4.3.2 The Public Relations Sub System

This subsystem is for projects data and related data include external supported institution local institution, Characters, and informing...etc.

The data flow that this subsystem may include is defined as follows:

4.3.2.1 User Name and Password

This operation is to prevent the unauthorized access to the system. The same operation as section (4.3.1.1)

4.3.2.2 Adding New Project

The society stores the project information in sequence that shown next

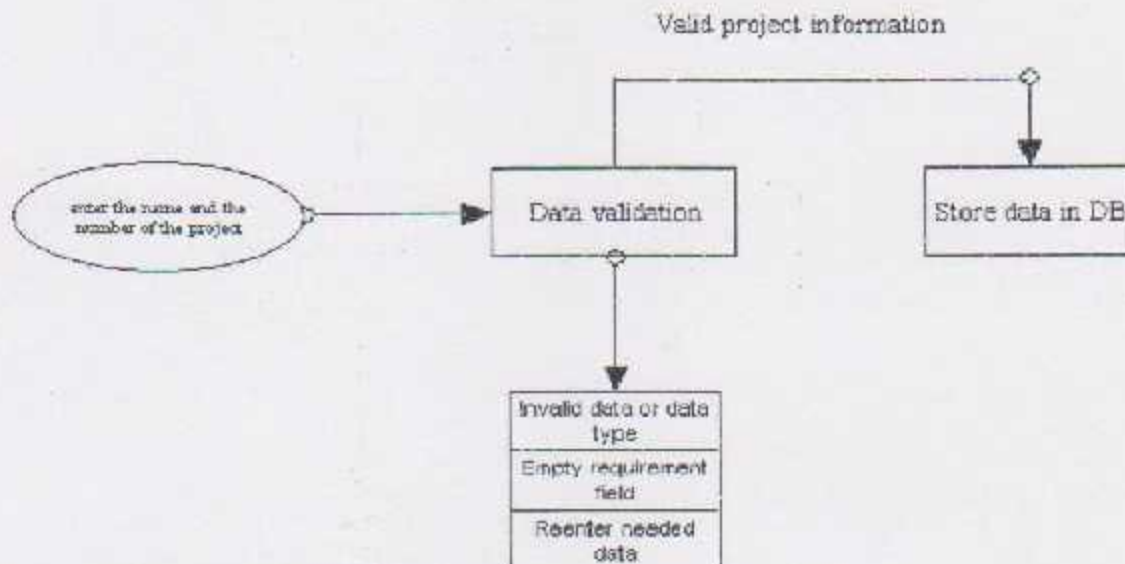


Figure (4.9) Add new project control and dataflow

4.3.2.3 Adding external supported institution

In this section we will shown the adding external supported institution control and dataflow as follows.

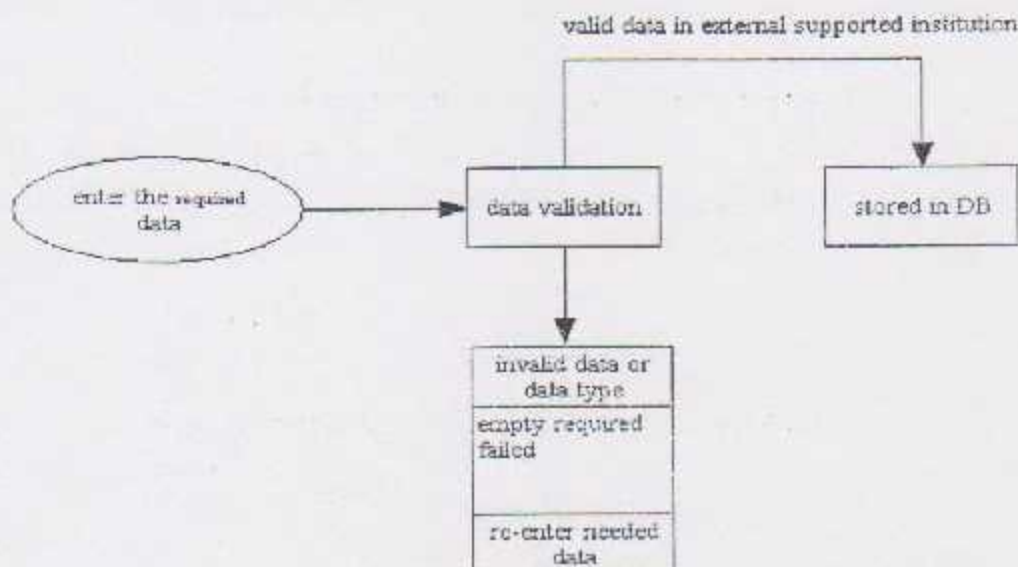


Figure (4.10) Adding an external supported institution control and dataflow

4.3.2.4 Adding local institution

In this section we will shown the adding local institution control and dataflow as follows.

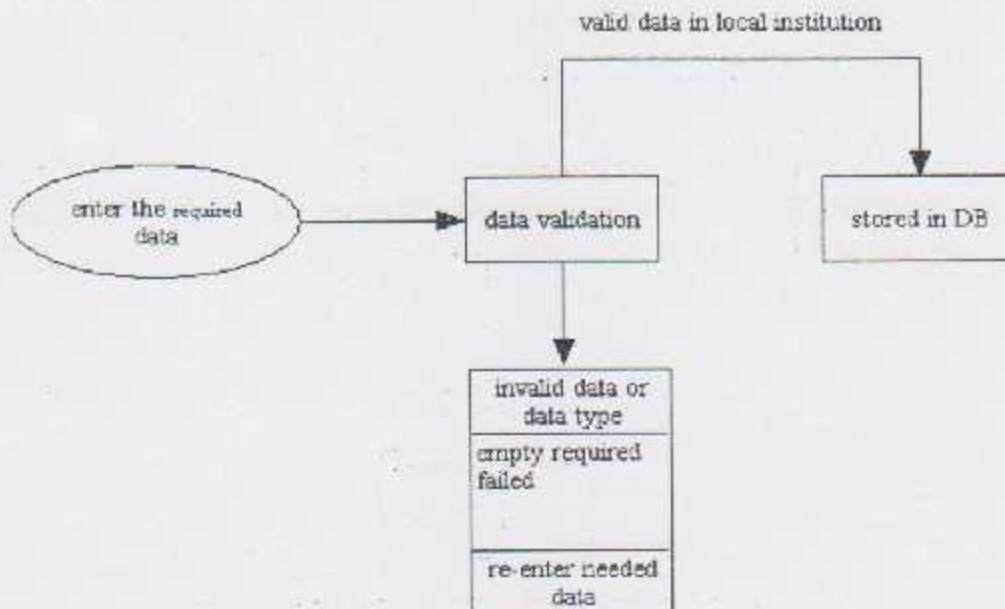


Figure (4.11) Adding a local institution control and dataflow

4.3.2.5 Adding Character (helper person)

In this section we will shown the adding character control and dataflow as follows

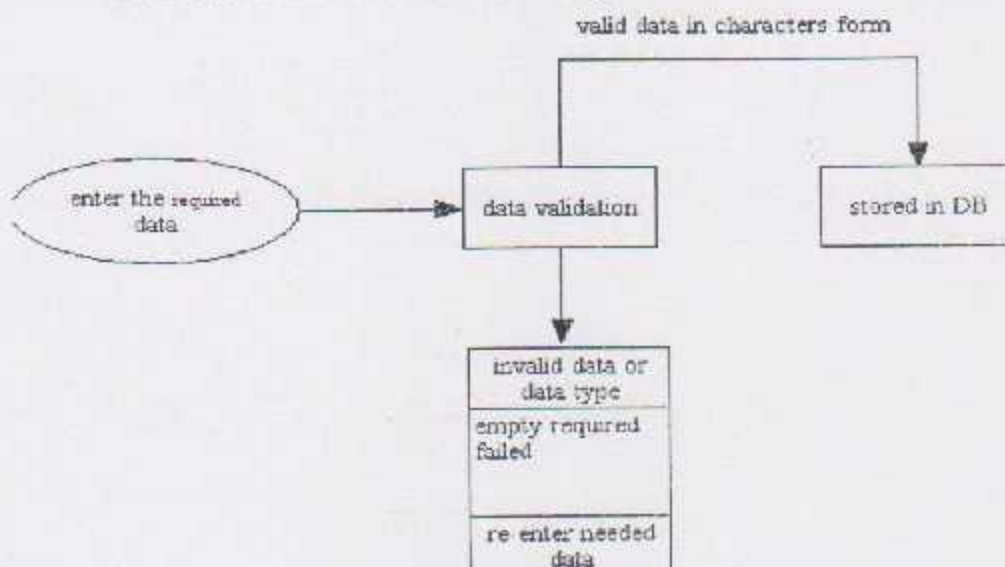


Figure (4.12) Adding new character control and dataflow

4.3.3 Employees Affairs Subsystem

This subsystem is for Employee data and related data include additional work, discounting, and vacations.

The data flow that this sub system may include is defined as followed:

4.3.3.1 User Name and Password

This operation is to prevent the unauthorized access to the system. The same operation as section (4.3.1.1)

4.3.3.2 Adding New Employee

To add a new employee to the system there are many number of fields that should be completed correctly to perform this operation correctly.

The adding resident control and data flow process will be as the following:

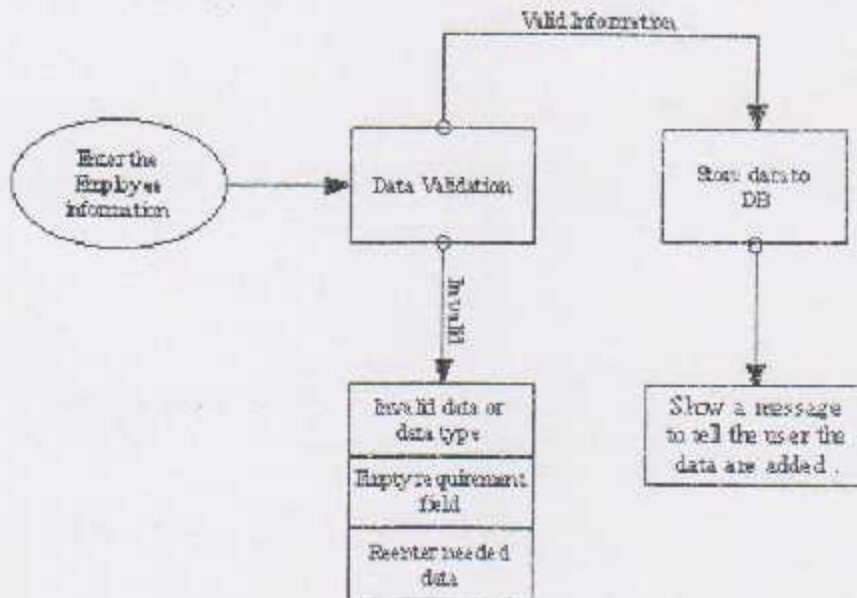


Figure (4.13) The data flow of adding new employee

4.4 Output data

The output data is the product of the system and the value returned by entering the input data and performing the processes through out the system. The output in this system consists of two main outputs; the query output, the data stored in the database.

4.4.1 The Query Output

The output of the query process is the information about the resident, sponsor, employee, project...etc, that the AL-IHSAN society needed to store there data and retrieve the data about those in fast way and include the ability to print this data to make a hard copy or formal paper for other using.

4.4.2 The data stored in the database

The output of the successful add, delete, update is to be updated to the database base, It represents data that is available to be shown in the query process.

4.5 ER Diagram

This section will include the ER model of each subsystem.

4.5.1 Resident ER Diagram

This section include the relational diagram for the resident subsystem as followed

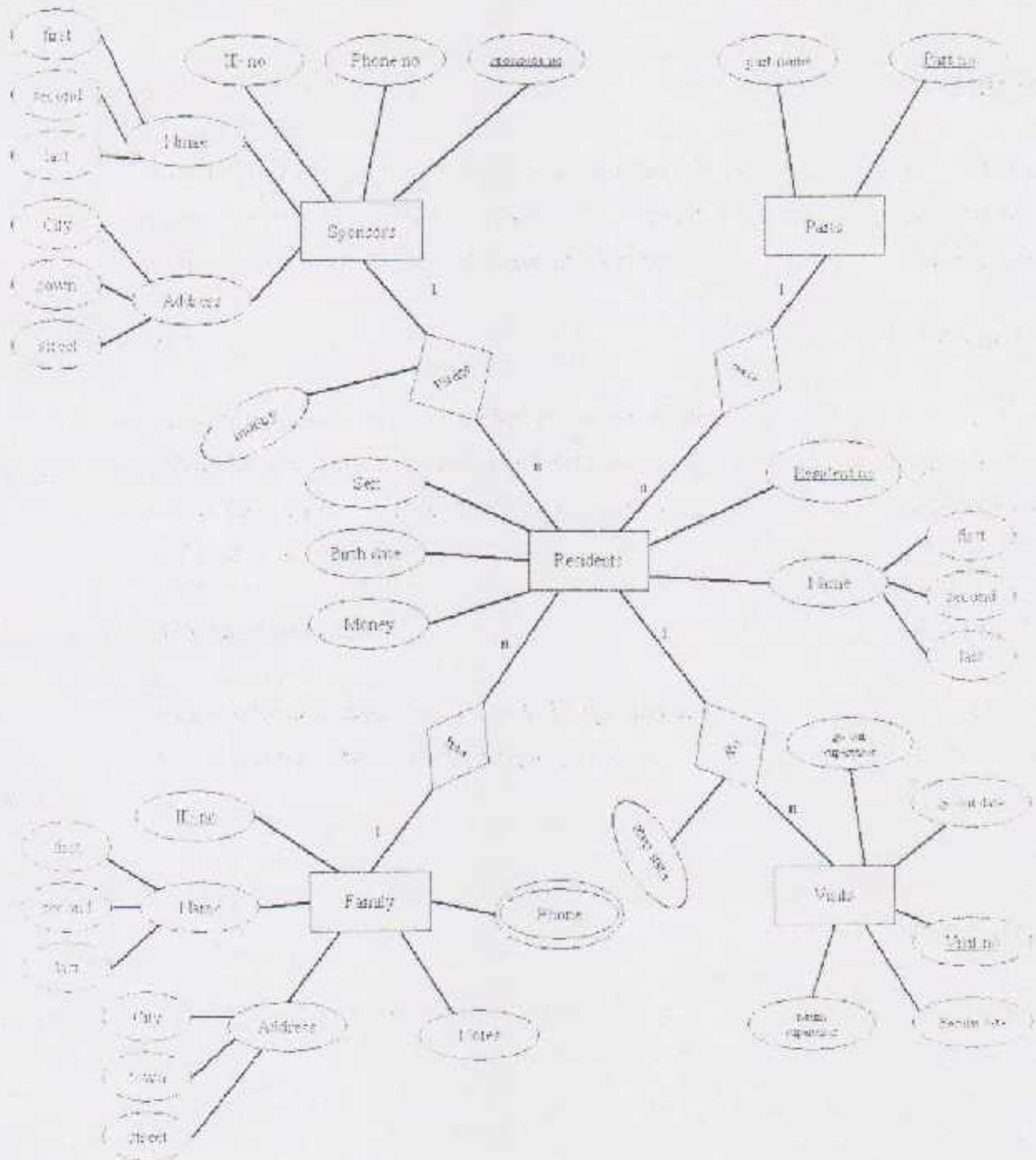


Figure (4.14) Resident ER diagram

4.5.2 The Public Relations Diagram

This section includes the relational diagram for the public sub system as followed.

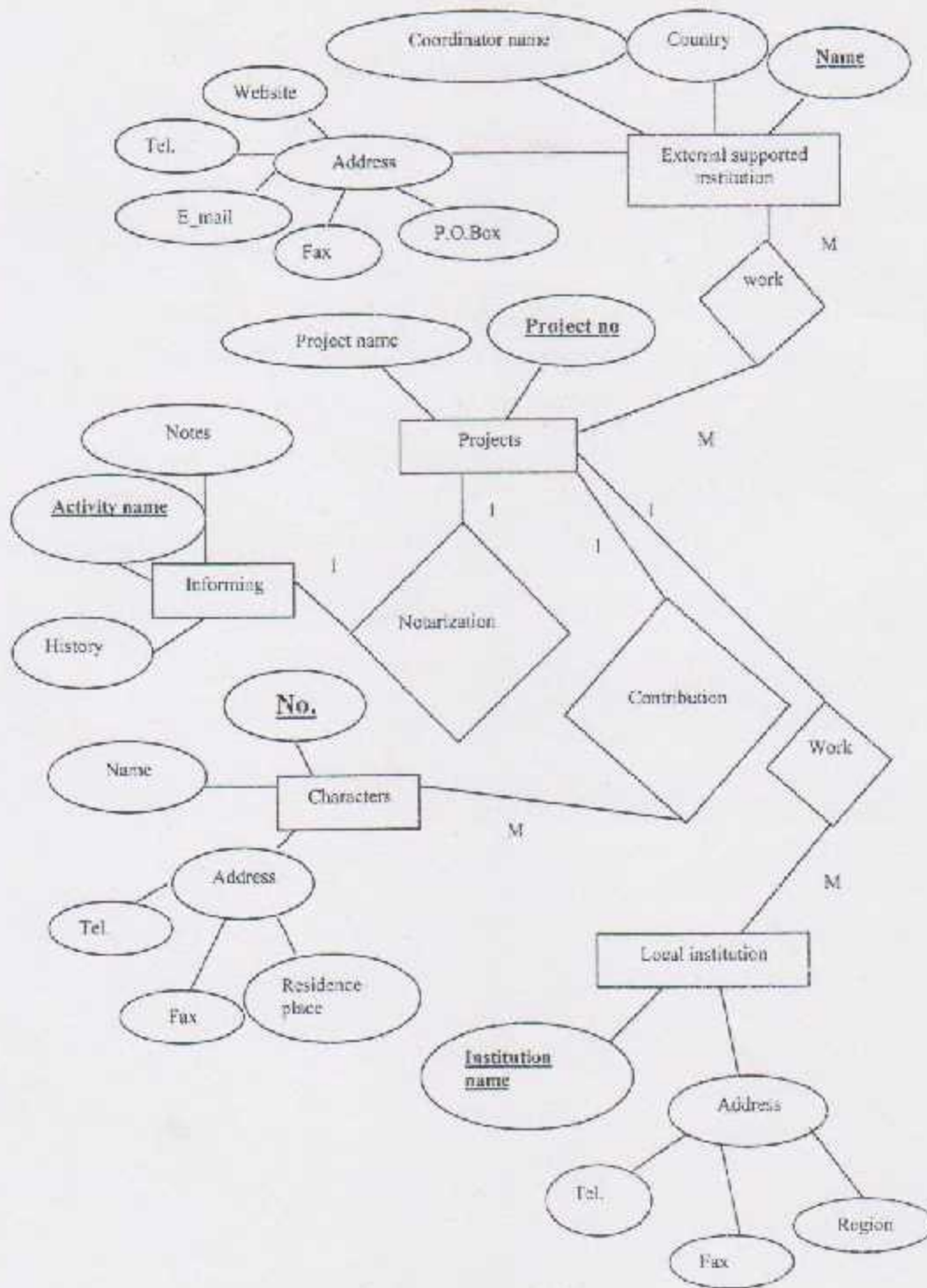


Figure (4.15) Public relations ER diagram

4.5.3 The Employee Affairs Diagram

This section includes the relational diagram for the employees affairs sub system as followed.

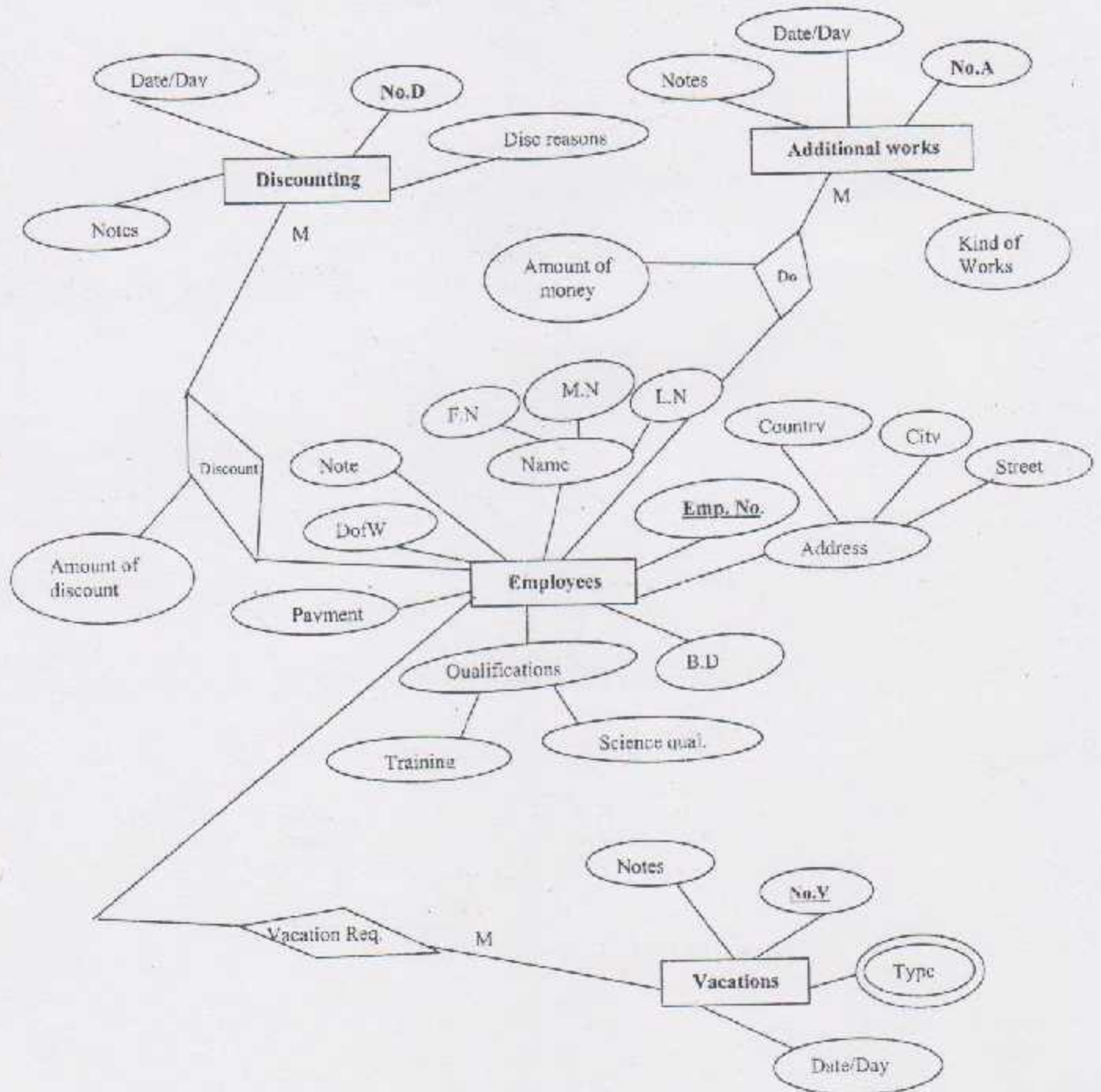


Figure (4.16) Employee affairs ER diagram

Chapter Five

System Design

Introduction

Overview

Database Design

The design of the system modules

The user interface design

5.1 Introduction

This chapter is a description of the system overall design, it includes the database design, the system objects and object models, the interface between system modules (administrator's module, client's module and visitor's module) with the database, and the user interface design.

Since the system is built in an object-oriented development environment (VB.NET framework) the work team foresees that its design process should be represented using the UML notations.

5.2 Overview

The system is composed from the following subsystems:

- Resident Subsystem.
- Public Relations Subsystem.
- Employees Affairs Subsystem.

Each subsystem is based on the department that uses the system:

The Resident subsystem is the system that used in the resident department and used to store and manage all the needed process that may needed to be taken at the resident data and other related data include the family of the resident and the sponsors (if any) and other stored data.

In the public relations there is a management between the project and the external supported institution, Local institution, and Characters that supported the "AL-IHSAN society" after that this department making an informing for the work that do it.

In the Employees Affairs subsystem the system used to manipulate, store, and manage the needed process for the data in employees and another related data.

5.3 Processes

This section will describe the entire process and conditions that will take place in the system, the processes are different from each department and it will be categorized by department as followed.

5.3.1 Resident Subsystem Processes

This part of section will show the entire operation for the resident sub system. This subsystem contain five different process that include add resident, delete/modify resident, add/delete/modify sponsor, change password and query resident or sponsor information as show in the flowing flow chart diagrams.

5.3.1.1 Add New Resident Flow Chart

This process start when the user needs to register a new resident to the system, the process will take the required field data as minimum then the system will validate the input data as followed.

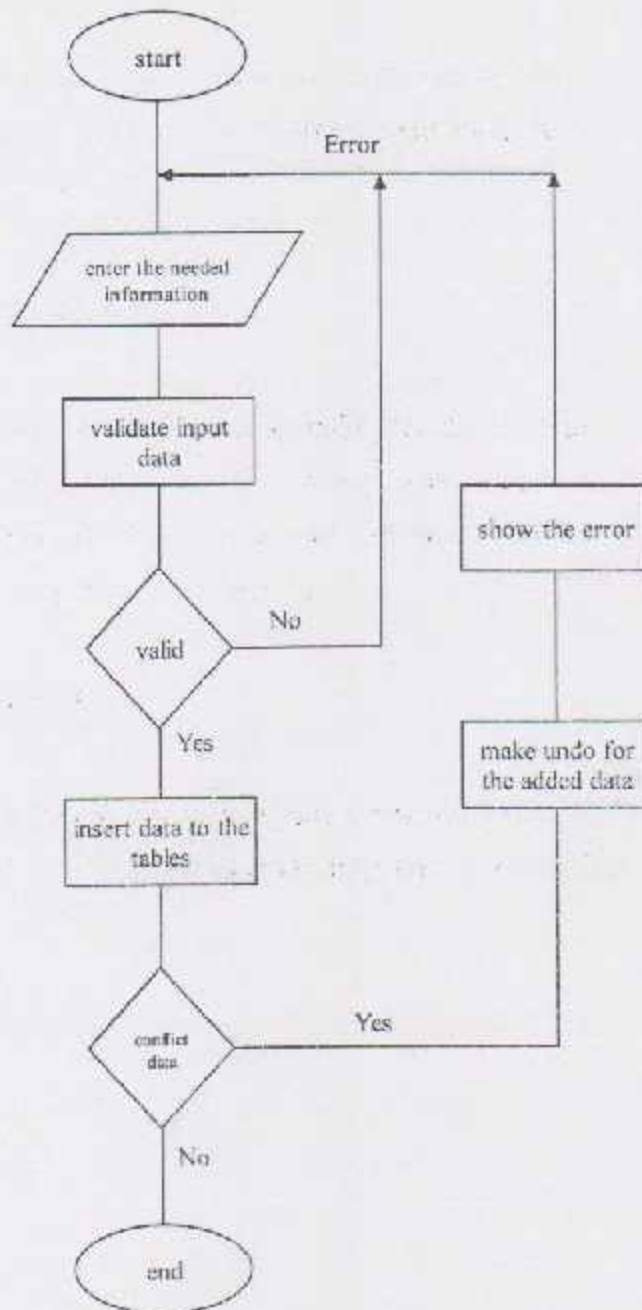


Figure (5.1) Add new resident flow chart

5.3.1.2 Delete, Update Resident

These processes will take place when the user of the system tries to change or delete resident information from the system as followed.

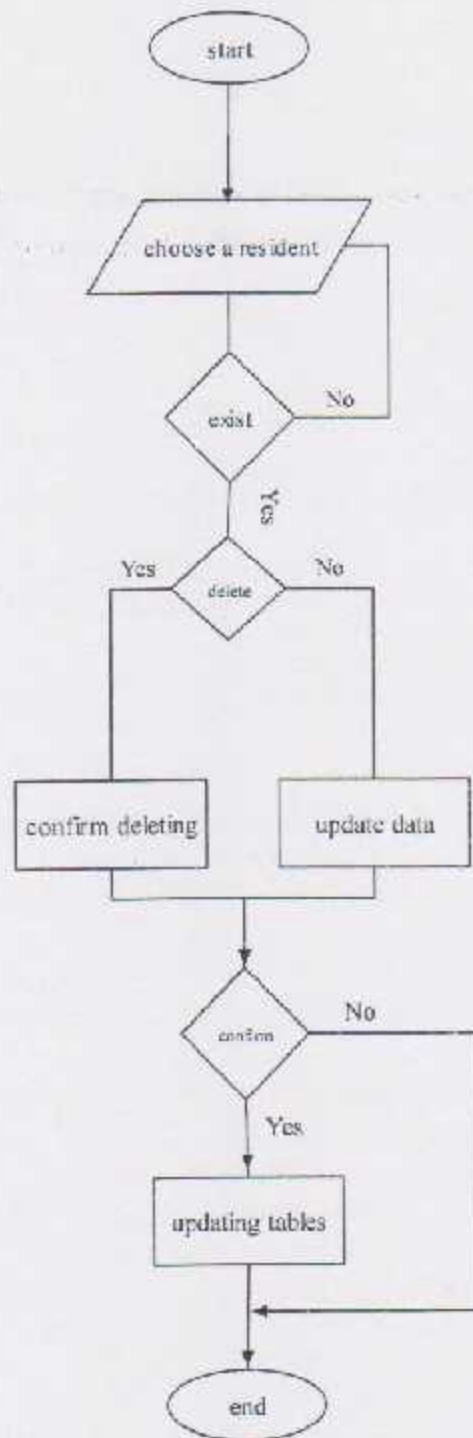


Figure (5.2) Delete new resident flow chart

5.3.1.3 Add, Delete, Update Sponsor

This process will take place when the user of the system deal with the sponsor information, the process is contain multi process that can be taken as a group together, the process start by enter the sponsor ID number as a minimum requirement then the system well check for the existing of the sponsor as followed.

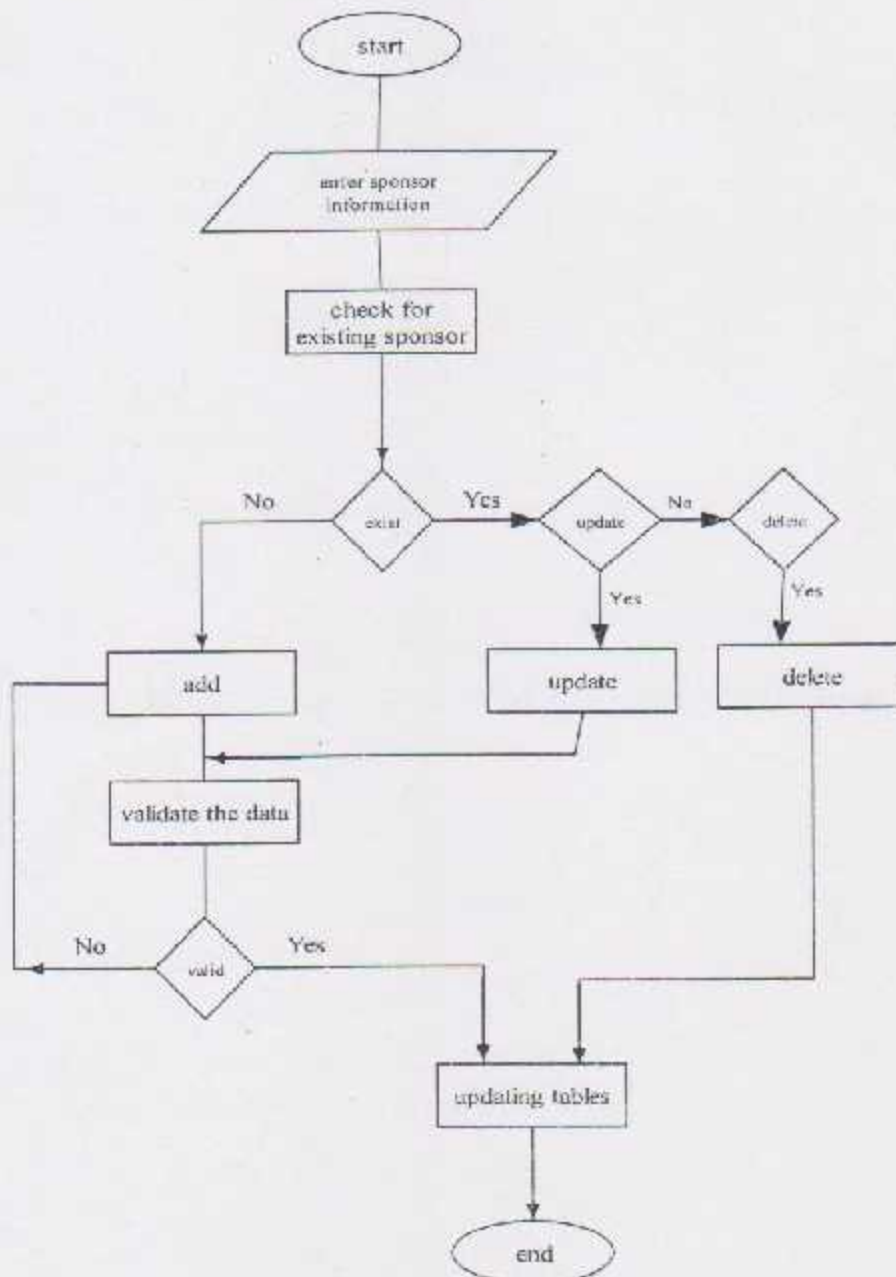


Figure (5.3) Add new sponsor flow chart

5.3.1.4 Query about resident or sponsor

This process is to get information directly from the DB including the search operation, by choosing the subject that includes resident or sponsor from the user of the system as followed.

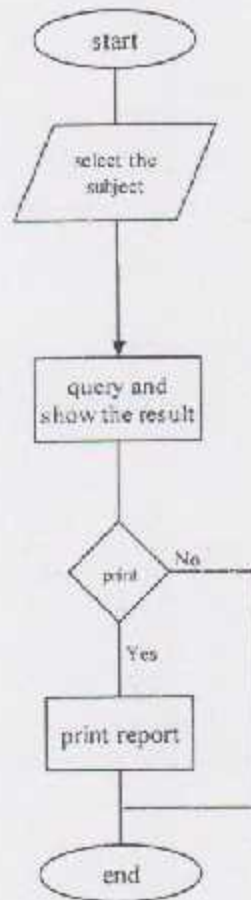


Figure (5.4) Query flow chart

5.3.1.5 Change Password

This process will take place when the user or administrator tries to change the login password, this process required to enter the old password of the domain name and new password as followed.

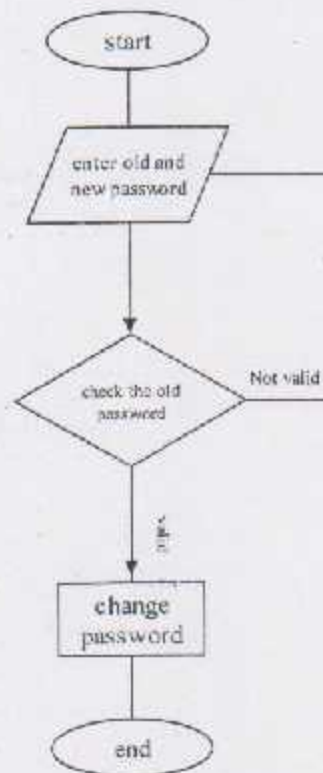


Figure (5.5) Change password for resident

5.3.2 Employees Affairs Subsystem Processes

This part of section will show the entire operation for the employee affairs subsystem. This sub system contain four different process that include add employee, delete/modify employee, change password of administrator, and query employee information as show in the flowing flow chart diagrams.

5.3.2.1 Add New Employee Flow Chart

This process start when the administrator needs to register a new Employee to the system, the process will take the required field data as minimum then the system will validate the input data as followed.

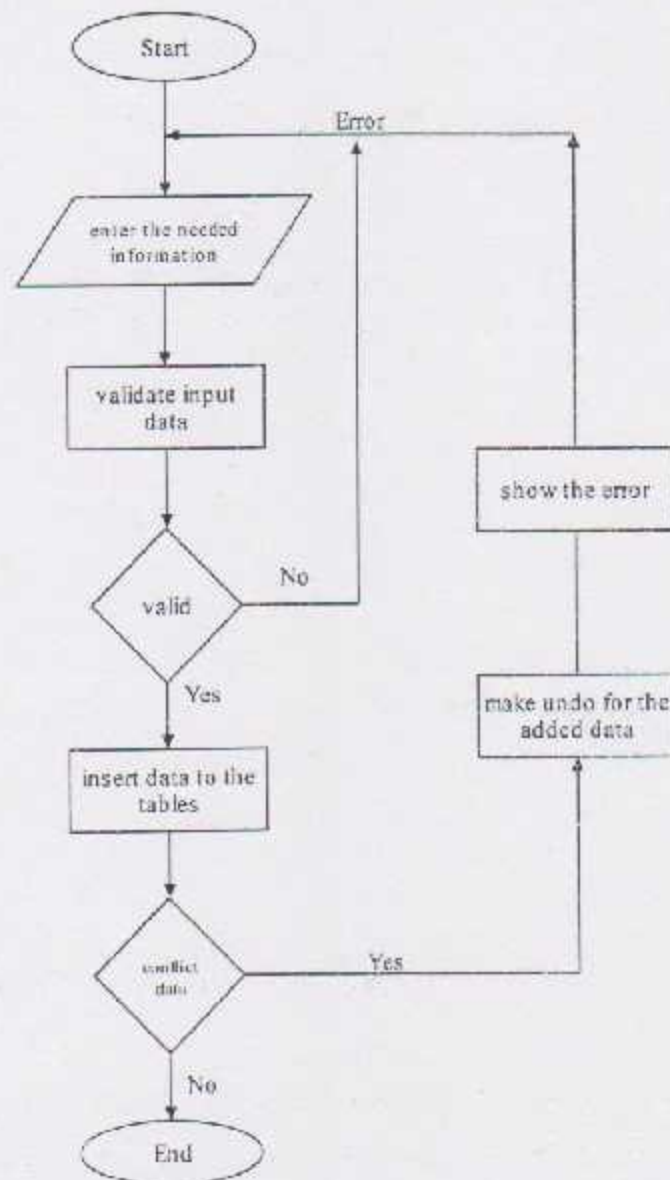


Figure (5.6) Add new employee flow chart

5.3.2.2 Delete, Update Employee

These processes will take place when the administrator needs to change or delete the employee's information from the system as followed.

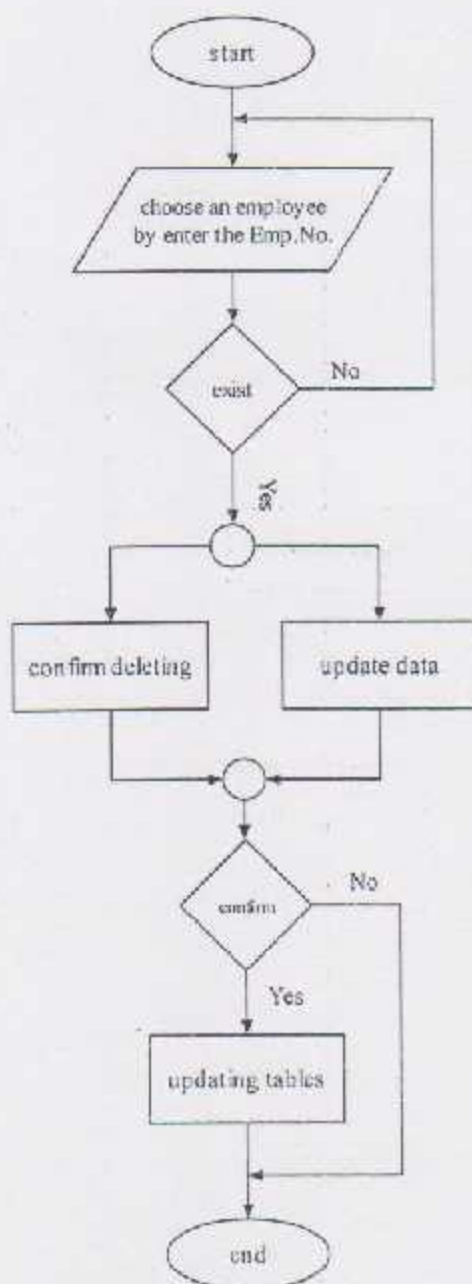


Figure (5.7) Delete/Update the employee flow chart

5.3.2.3 Query about Employee

This process is to get information directly from the DB including the search operation, by writing the appropriate query from the administrator then sees the data that needed and allows printing the information.

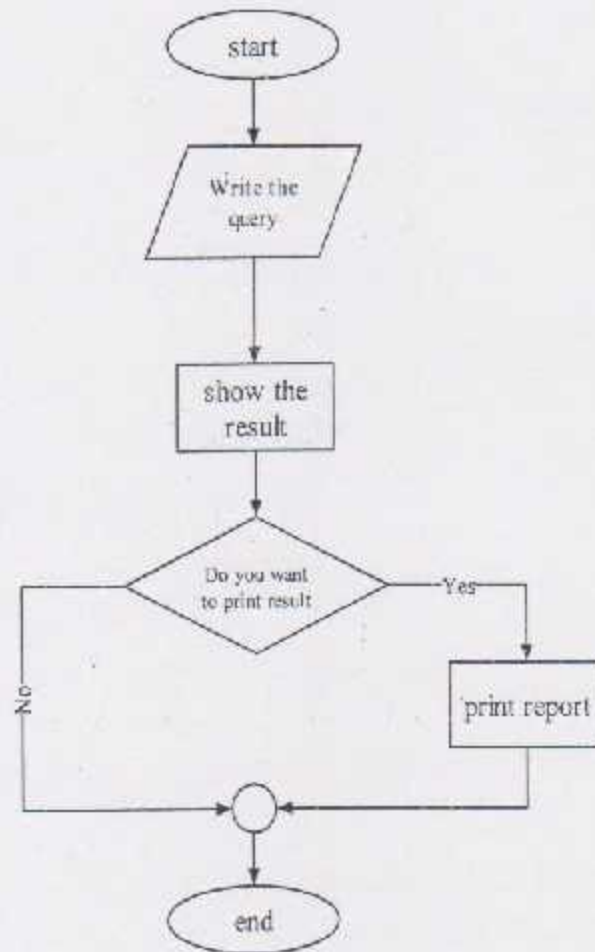


Figure (5.8) The query about employee

5.4 Database Design

This section includes the database design which is described in the data dictionary (next), database relations diagram and the UML conceptual model for the "AL-IHSAN society" database.

5.4.1 Mapping

In this section we will show the mapping process as followed:

5.4.1.1 Relational Mapping

1. *The Resident Subsystem*

- Sponsors (sponsor number, phone number, first name, second name, last name, city, town, street, ID number)
- Parts (part number, part name)
- Family (ID number, first name, second name, last name, city, town, street, notes)
- Telephone (TEL number, ID number FK refers to family)
- Residents (resident number, first name, second name, last name, birth date, sex, the needed money, sponsor number FK refers to sponsors, part number FK refers to parts, ID number FK refers to family)
- Visits (visit number, go out date, return date, go out supervisor name, return supervisor name, resident number FK refers to resident)
- Pledge (sponsor number, resident number, the pledge money)
- Visit (visit number, resident number, visit date)

2. The Public Relations Subsystem

- Informing (Activity name, History, Notes, Project no. *FK refers to the Projects*)
- Projects (Project no., Project name, Activity name *FK refers to the Informing*)
- Characters (No., Name, Fax, Tel., Residence Place, Project no *FK refers to the Projects*)
- Local Institution (Institution name, Region, Fax, Tel, Project no *FK refers to the Project*)
- External supported Institution (Name, Country, Coordinator name, E_mail, Web_site, P.O.Box, Tel, Fax)
- Supported (External Supported Institution Name, Project no)

3. The Employees Affairs Subsystem

- Employees (Emp.No. , F.Name , M.Name , L.Name , Street ,City, Country, date of work, payment, Training , Science qualification , B.D)
- Additional works (No.A , Date/Day , Kindofworks , Notes, Amount of money , Emp No. from Employees relation (F.K))
- Discounting (No.D ,Date/Day , Discount reason , Notes, Amount of discount , Emp.No. from employees relation (F.K))



- Vacations (No.V , Date/Day ,Notes Emp.No. from Employee relation (F.K))
- Vacations_Type (Type , No.V from vacation relation (F.K))

5.4.1.2 Normalization

We assume that the relational database schematic is normalized because the way that used to analysis and design is prevent to get non stable or non normalized system, so we will make the two basic normalize form (the first and second normal form) as the following.

1. The Resident Subsystem

Sponsors:

<u>sponsor number</u>	phone number	first name	second name	last name	city	town	street	ID number
-----------------------	--------------	------------	-------------	-----------	------	------	--------	-----------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (sponsor number) is primary key, every non-prime attribute is fully functionally dependent on the primary key.

Parts:

<u>part number</u>	part name
--------------------	-----------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (part number) is primary key; every non-prime attribute is fully functionally dependent on the primary key.

Family:

<u>ID number</u>	first name	second name	last name	city	town	street	notes
------------------	------------	-------------	-----------	------	------	--------	-------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (ID number) is primary key; every non-prime attribute is fully functionally dependent on the primary key.

Telephone:

<u>TEL number</u>	<u>ID number</u> FK refers to family
-------------------	--------------------------------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (TEL number) is primary key; every non-prime attribute is fully functionally dependent on the primary key.

Residents:

<u>resident number</u>	first name	second name	last name	birth date	sex	the needed money	<u>sponsor number</u> FK	<u>part number</u> FK	<u>ID number</u> FK
------------------------	------------	-------------	-----------	------------	-----	------------------	--------------------------	-----------------------	---------------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (resident number) is primary key; every non-prime attribute is fully functionally dependent on the primary key.

Visits:

<u>visit number</u>	go out date	return date	go out supervisor name	return supervisor name	<u>resident number</u> FK
---------------------	-------------	-------------	------------------------	------------------------	---------------------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (visit number) is primary key; every non-prime attribute is fully functionally dependent on the primary key.

Pledge:

<u>sponsor number</u>	<u>resident number</u>	the pledge money
-----------------------	------------------------	------------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (sponsor number, resident number) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Visit:

<u>visit number</u>	<u>resident number</u>	visit date
---------------------	------------------------	------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (visit number, resident number) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

2. The Public Relations Subsystem

Informing:

<u>Activity name</u>	History	Notes	Project no. F.K. refers to the Projects
----------------------	---------	-------	---

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (Activity name) is primary key, every non-prime attribute is fully functionality dependent on the primary key.

Projects:

<u>Project no.</u>	project name	Activity name F.K. refers to the
--------------------	--------------	----------------------------------

		Informing
--	--	-----------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (Project no) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Characters:

<u>No.</u>	Name	Residence Place	Fax	Tel	Project no. F.k.. refers to the Projects
------------	------	-----------------	-----	-----	--

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (No. of character) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Local Institution:

<u>Institution name</u>	Region	Fax	Tel	Project no FK refers to the Project
-------------------------	--------	-----	-----	-------------------------------------

First normal form: It's in the first normal form because its part of relation definition

Second normal form: (Institution name) is primary key; every non-prime attribute is fully functionality dependent on the primary key

External Supported Institution:

<u>name</u>	Country	Coordinator name	E_mail	Web_site	P.O.Box	Tel	Fax
-------------	---------	------------------	--------	----------	---------	-----	-----

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (name) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Supported:

<u>External supported institution</u>	<u>Project no</u>
---------------------------------------	-------------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (External supported institution, Project no) is primary key; every and there is no non-prime attribute depend on it.

3. The Employees Affairs Subsystem

Employees:

<u>Employee No</u>	Birth of date	first name	second name	last name	city	town	street	Date of work	payment
training	Science qualifications		Notes						

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (Employee No.) is primary key, every non-prime attribute is fully functionality dependent on the primary key.

Additional Work's:

<u>No.A</u>	Date/Day	Kind of work	Amount of money	notes
-------------	----------	--------------	-----------------	-------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (No.A) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Discounting:

<u>No.D</u>	Date/Day	Discount reason	Amount of discount	notes
-------------	----------	--------------------	-----------------------	-------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (No.D) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Vacations:

<u>No.V</u>	Date/Day	notes
-------------	----------	-------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (No.V) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

Vacation type:

<u>TypeNo</u>	<u>type</u>
---------------	-------------

First normal form: It's in the first normal form because its part of relation definition.

Second normal form: (TypeNo.,Type) is primary key; every non-prime attribute is fully functionality dependent on the primary key.

5.4.2 Data Dictionary

This part is a description of the database and tables used in the system; it contains the overall description of tables and the specification of each table and its fields.

5.4.3 Table's Description

After analyzing the requirements of the database of the system and depending on security and integrity rules, the tables of the database are shown in the following table:

1. The Resident Subsystem

Table name	Description
Resident	It contains the names of the resident in the society and information about them
Sponsors	It contains the name of the people or societies that paid money for pledge resident
Parts	It contains the name of parts the resident may exist in.
Family	It contains the information about the family of the existing resident
Telephone	It contains the telephone number for the resident family
Visits	It contains the information about the resident visits that taken
Pledge	It contains the amount of money that paid by the sponsor for a resident.

Table (5.1) The table's description for resident subsystem

2. The Public Relations Subsystem

Table name	Description
Project	It contains the names of the projects that done by the society and information about them
Characters	It contains the name of the people that paid money for this society and some data for them

External supported institution	It contains the name of the External supported institution that built the project and some data for them.
Local institution	It contains the information about the local institution
Informing	It contains the name of the project and the notarization for it
Supported	It contains the information about the project number and the external supported institution that built it

Table (5.2) The table's description for Public Relations Subsystem

3. The Employees Affairs Subsystem

Table Name	Description
Employee	It contains names of employee, payment, date of work, and information.
Additional work's	It contains the information about additional work's for employees, the date and day.
Discounting	It contains discount reason, date and day, and information.
Vacations	It contains the vacations that requested from the employee, record the date and day, and type of vacations.
Vactions_type	It contains the types of vacations.

Table (5.3) The table's description for Employee's Affairs subsystem

5.4.4 Tables

This is a description of each table and its fields

1. The Resident Subsystem

- 1- Resident table: this table is the main table for this department, this table will take the most information about the resident and this table will be the key for many table as following:

Field Name	Data Type	Null	Key	References	Length	Description
rnumber	Integer	No	PK		9	Resident unique

						number
fname	text	No			50	Resident first name
sname	text	No			50	Resident second name
lname	text	No			50	Resident last name
bdate	text	No			10	Resident birth date
sex	text	No			4	The resident sex
Tmoney	Integer	No			9	The total money that should paid
Spnumber	Integer	Yes	FK	sponsor	9	Sponsor number
pnumber	Integer	No	FK	part	9	Part number
ldnumber	Integer	No	FK	Family	9	Family id number

Table (5.4) Resident table description

2-sponsors table: this table is created to contain the information about the sponsors that they donate for the resident in the society and it contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
<u>snumber</u>	Integer	No	PK		9	sponsor unique number
Spnumber	text	No			50	Sponsor phone number
Fname	text	No			50	Sponsor first name
Sname	text	Yes			50	Sponsor second

						name
Lname	text	No			50	Sponsor last name
City	text	No			50	Sponsor city
Town	text	No			50	Sponsor town
Street	text	No			50	Sponsor street
IDnumber	Integer	Yes			9	Sponsor ID card number

Table (5.5) Sponsors table description

3-part table: this table is used to store the part that the resident department contains and it will as followed:

Field Name	Data Type	Null	Key	References	Length	Description
<u><i>pnumber</i></u>	Integer	No	PK		9	part unique number
<u><i>Pname</i></u>	text	No			50	Part name

Table (5.6) Part table description

4-family table: this table is used to store the information about the resident family and it contains:

Field Name	Data Type	Null	Key	References	Length	Description
<u><i>IDnumber</i></u>	Integer	No	PK		9	part unique number
<u><i>fname</i></u>	text	No			50	first name
<u><i>Sname</i></u>	text	No			50	Second name
<u><i>Lname</i></u>	text	No			50	Last name

<u>City</u>	text	No			50	Family city
<u>Town</u>	text	No			50	Family town
<u>Street</u>	text	No			50	Family street
<u>Notes</u>	text	No			50	

Table (5.7) Family table description

5- Telephone table: this table is created to solve the multi-value for the family telephone number and it contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
<u>Tnumber</u>	text	No	PK		50	telephone unique number
<u>Idnumber</u>	Integer	No	FK	family	9	The family ID number

Table (5.8) Telephone table description

6-Visits table: this table is created to store the visits data, the resident can take a visit for his family and there many data that we most get from this process and the table contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
<u>vnumber</u>	Integer	No	PK		9	visit unique number
<u>gdate</u>	text	No			10	Go out date
<u>rdate</u>	text	No			10	Return date
<u>gsuper</u>	text	No			50	Go out supervisor
<u>rsuper</u>	text	No			50	Return supervisor

<u>rnumber</u>	integer	No	FK	resident	9	Resident name
----------------	---------	----	----	----------	---	---------------

Table (5.9) Visits table description

7-pledge table: this table is created to store the amount of money that the sponsor is donating for a resident and it contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
<u>spnumber</u>	Integer	No	PK		9	Sponsor number
<u>rnumber</u>	Integer	No	PK		9	Resident number
<u>pmoney</u>	Integer	No			9	Pledge money

Table (5.10) Pledge table description

8-visit table: this table is designed to store the agreed date for the visit between the family and the society and it contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
<u>vnumber</u>	Integer	No	PK		9	visit number
<u>rnumber1</u>	Integer	No	PK		9	Resident number
<u>vdate</u>	text	No			10	Visit date

Table (5.11) Visit table description

2. The Public Relations Subsystem

1- Project table: this table is the main table for this department, this table will take the information about the project and this table will be the key for many tables as following:

Field Name	Data Type	Null	Key	References	Length	Description
Project no	Integer	No	PK		9	Project unique

						number
Project name	text	No			50	project name
Activity name	text	No	FK	Fk refers to informing table	50	Activity name for the project

Table (5.12) Project table description

2- Characters table: this table is created to contain the information about the characters that they grant the money to the society and it contains the following:

Field Name	Data Type	Null	Key	References	Length	Description
Number	Integer	No	PK		9	Characters unique number
Name	text	No			50	Characters name
Fax	Integer	YES			9	Fax # for the characters
Tel	Integer	NO			9	Tel # for the characters
Residence name	text	No			50	Residence name for the characters
Project no	Integer	No	FK	FK refers to the project table	9	Project number have money from this characters

Table (5.13) Characters table description

3-External supported institution: this table is used to store the name and some data for the external supported institution that support the project in this society as followed:

Field Name	Data Type	Null	Key	References	Length	Description
name	text	No	PK		50	unique name
country	text	No			50	Name of the country for the institution
Coordinator name	text	No			50	Name of the coordinators
E-mail	text	No			50	e-mail for the institution
Web-site	text	No			50	Web-site for this institution
P.O.Box	integer	No			9	P.O.Box for the institution
Tel	integer	No			9	Tel for the institution
Fax	integer	No			9	Fax for the institution

Table (5.14) External supported institution table description

4-local institution table: this table is used to store the information about the local institution and it contains:

Field Name	Data Type	Null	Key	References	Length	Description
Institution name	text	No	PK		50	institution unique name
Region	text	No			50	Name of the region
Fax	integer	Yes			9	Fax number for the institution

Tel	integer	No			9	Tel number for the institution
Project no	integer	No	FK	FK refers to the project table	9	Number for the project that supported by this institution

Table (5.15) Local institution table description

5- Informing table: this table is used to store information for each project and notarization for it by that date.

Field Name	Data Type	Null	Key	References	Length	Description
Activity name	text	No	PK		50	telephone unique number
history	date	No			8	The date for the project
notes	text	Yes			16	To write the notes on the project
Project no	integer	No	FK	FK refers to the project table	9	Project number for this informing

Table (5.16) Informing table description

6-Supported table: this table is created to store the project number and the external supported institution name, and it is created to solve the relation (M: M) between 2 tables (project and the external supported institution).

Field Name	Data Type	Null	Key	References	Length	Description
Project no	Integer	No	PK		9	Project unique

						number
external supported institution name	text	No	PK		.50	Name of this institution

Table (5.17) Visits table description

3. The Employees Affairs Subsystem

1. Employees table "Emp": this table is created to add, delete, and update the employee information and it contains the following fields.

Field Name	Data Type	Null	Key	References	Length	Description
<u>EmpNo</u>	Integer	No	PK		9	Employee unique number
Fname	text	No			50	First name of Emp.
Mname	text	No			50	Second name of Emp.
Lname	text	No			50	Last name of Emp.
Street	text	Yes			50	Part of address.
City	text	Yes			50	Part of address.
Country	text	Yes			50	Part of address.
DofW	date	No			8	The date of start work.

Payment	Integer	No			9	The payment of Emp.
Training	text	Yes			50	The training field.
Science qual.	text	Yes			50	The science qualification
Notes	text	Yes			50	Additional information.

Table (5.18) Employees table description

2. Additional works table "Add.W": this table provides the information for additional work's that accomplished by different employees and the following fields are contains.

Field Name	Data Type	Null	Key	References	Length	Description
Ano	Integer	No	PK		9	This Auto number.
KofW	text	No			50	The kind of work.
EmpNo	Integer	No	FK	Employee	9	The Emp. number that exists in the Employee table.
W_Date	text	No			8	The date for each additional work.
AofM	Integer	No			50	Amount of the money.
Notes	text	Yes			50	Information.

Table (5.19) Additional work table description

3. The Discounting Table "disc": this table is used to record the reason of discounting therefore it discount from the origin amount of money and it will contains the following fields.

Field Name	Data Type	Null	Key	References	Length	Description
Dno	Integer	No	PK		9	The auto number of each discounting employee.
Dreason	text	Yes			50	Used to record the causes of discounting.
EmpNo	Integer	No	FK	Employee	9	The Emp. number that exists in the Employee table.
Dmoney	Integer	No			9	The value of discounting money.
Notes	text	No			50	Information.

Table (5.20) The discounting table description

4. The Vacations table "vac": this tables is used to store information about the vacations that takes from the employee request and determines it allowed or not for takes the agreements and it will contains the following there fields.

Field Name	Data Type	Null	Key	References	Length	Description
Vno	Integer	No	PK		9	A unique number of the vacations.
Date/Day	Date	N			8	The date and day
Emp.No	Integer	No	FK	Employee	9	The Emp.

						number that exists in the Employee table.
Notes	text	Yes			50	Information.

Table (5.21) Vacations table description

5. The Vacation_type table "Vac_Type": this table it contains different types of vacations, it relation by vacation table, and it will contains the following fields.

Field Name	Data Type	Null	Key	References	Length	Description
<u>Type</u>	text	No	PK		50	The type of vacations.
<u>Vno</u>	Integer	No	FK		9	Refers to vacations table.

Table (5.22) The Vacations Type Table description

5.4.5 The UML Conceptual Model for the "AL-IHSAN" Database

1. The Resident Subsystem:

This diagram is a conceptual model that represents all the database tables with their relations, primary and foreign keys. The relations in this system are built between the "resident" table and the "family", "sponsor", "part" and "telephone" and "visits" tables in order to facilitate the access to the database.

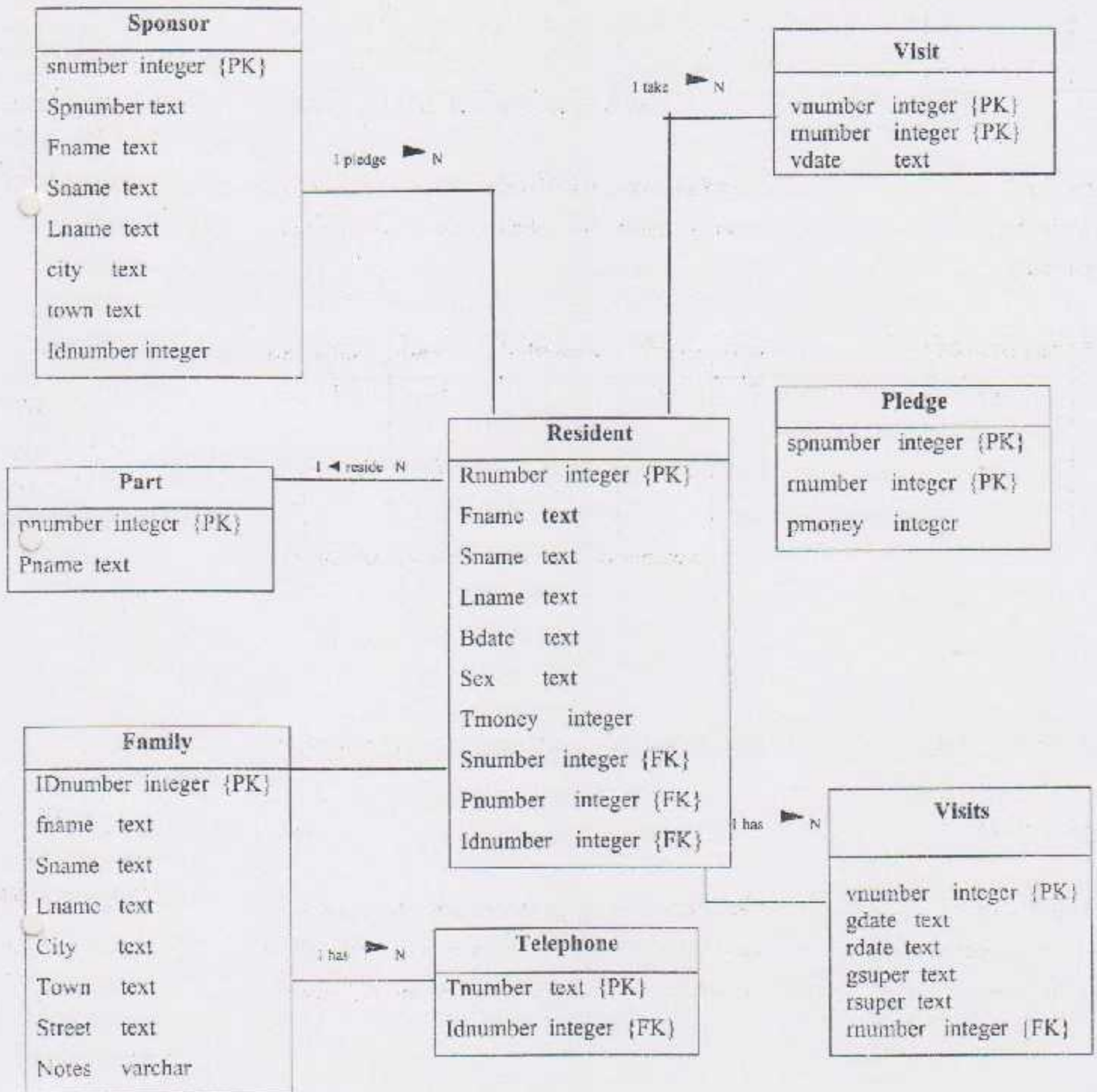


Figure (5.9) Conceptual model for the resident sub system

2. The Public Relations Subsystem:

This diagram is a conceptual model that represents all the database tables with their relations, primary and foreign keys. The relations in this system are built between the "project" table and the "characters", "supported", "local institution" and "Informing" and "External supported institution" tables in order to facilitate the access to the database.

3. The Employees Affairs Subsystem:

This diagram is a conceptual model that represents all the database tables with their relations, primary and foreign keys. The relations in this system are built between the "Employee", "Additional work", "discounting", and "vacations" tables in order to facilitate the access to the database.

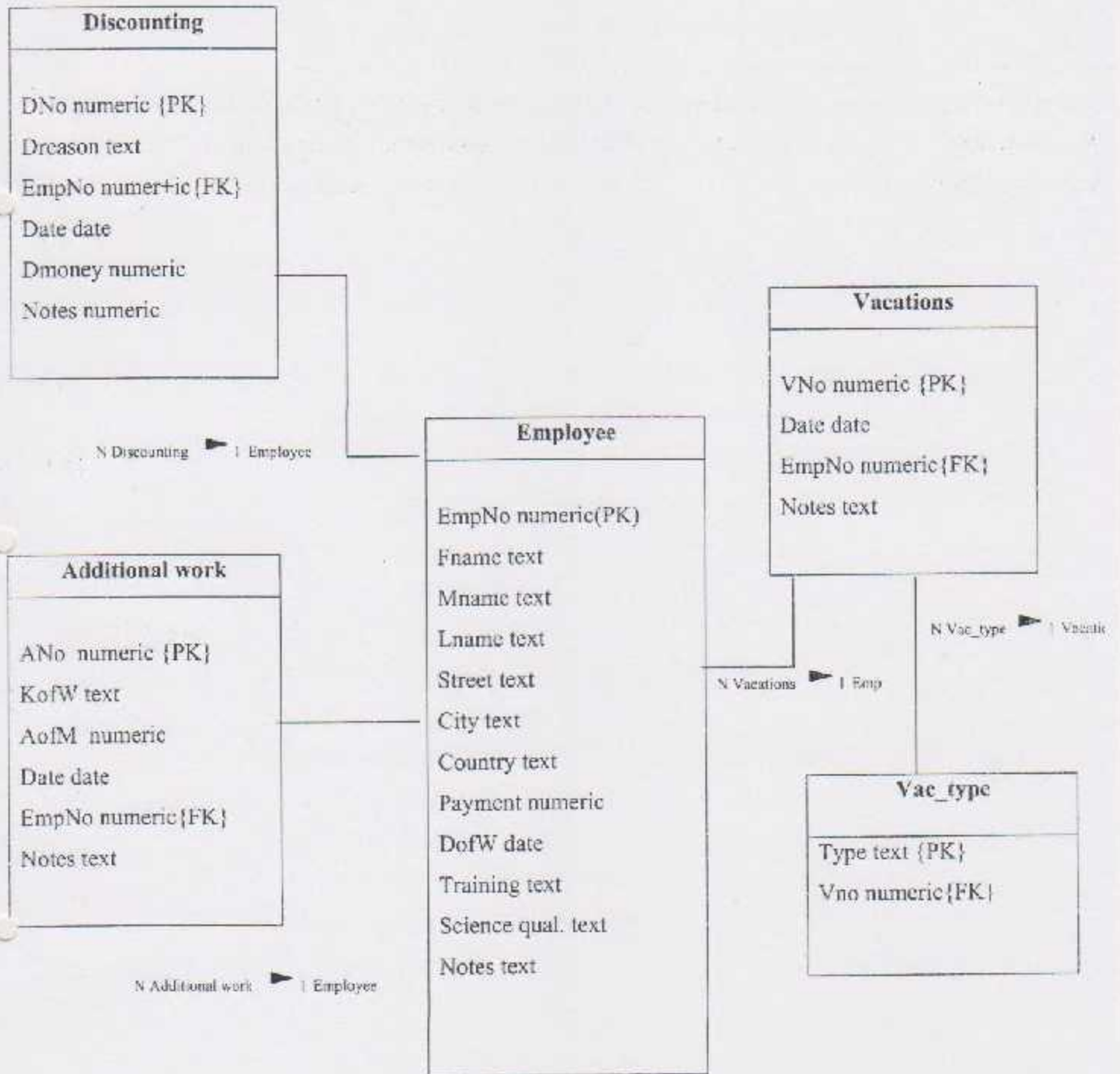


Figure (5.11) Conceptual model for employee's affairs subsystem

5.5 The Design of the System Modules

The design process of the system is built on is an object oriented approach as following.

5.5.1 System and Subsystem Architecture

Al-IHSAN society consists of three main sub system as following:

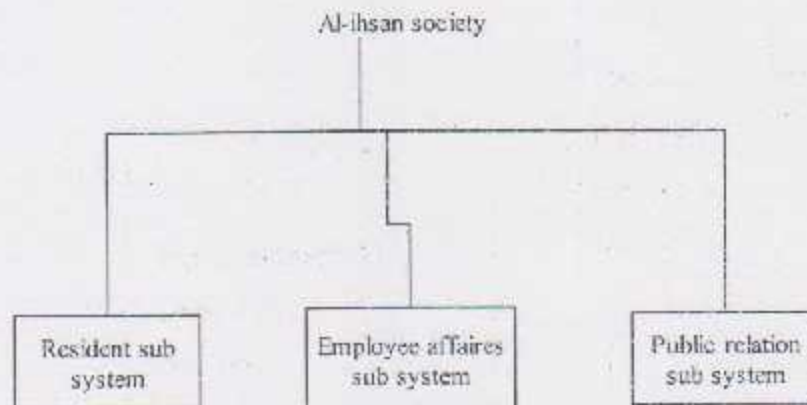


Figure (5.12) AL-IHSAN society sub system

5.5.2 Object Identification

This section includes the main and active objects of the system, their attributes and methods according to the UML documentation criteria. These figures show the principle objects of the system.

1. The Resident Subsystem

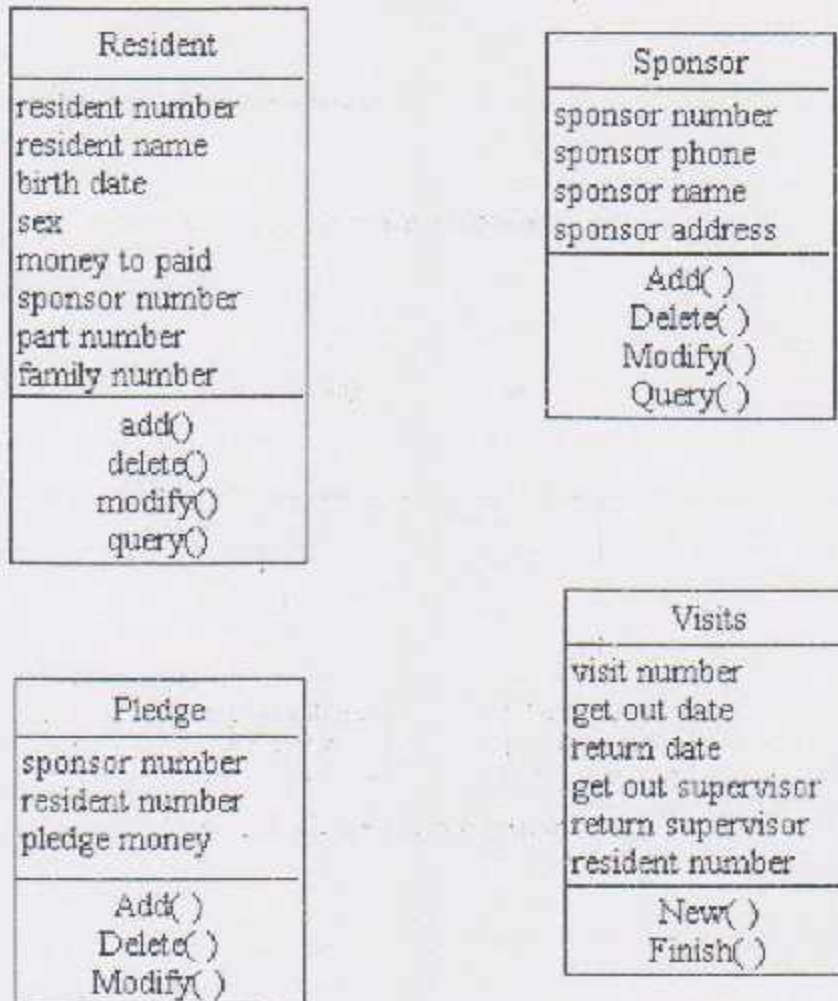


Figure (5.13) The object identification for resident subsystem

2. The Public Relations Subsystem

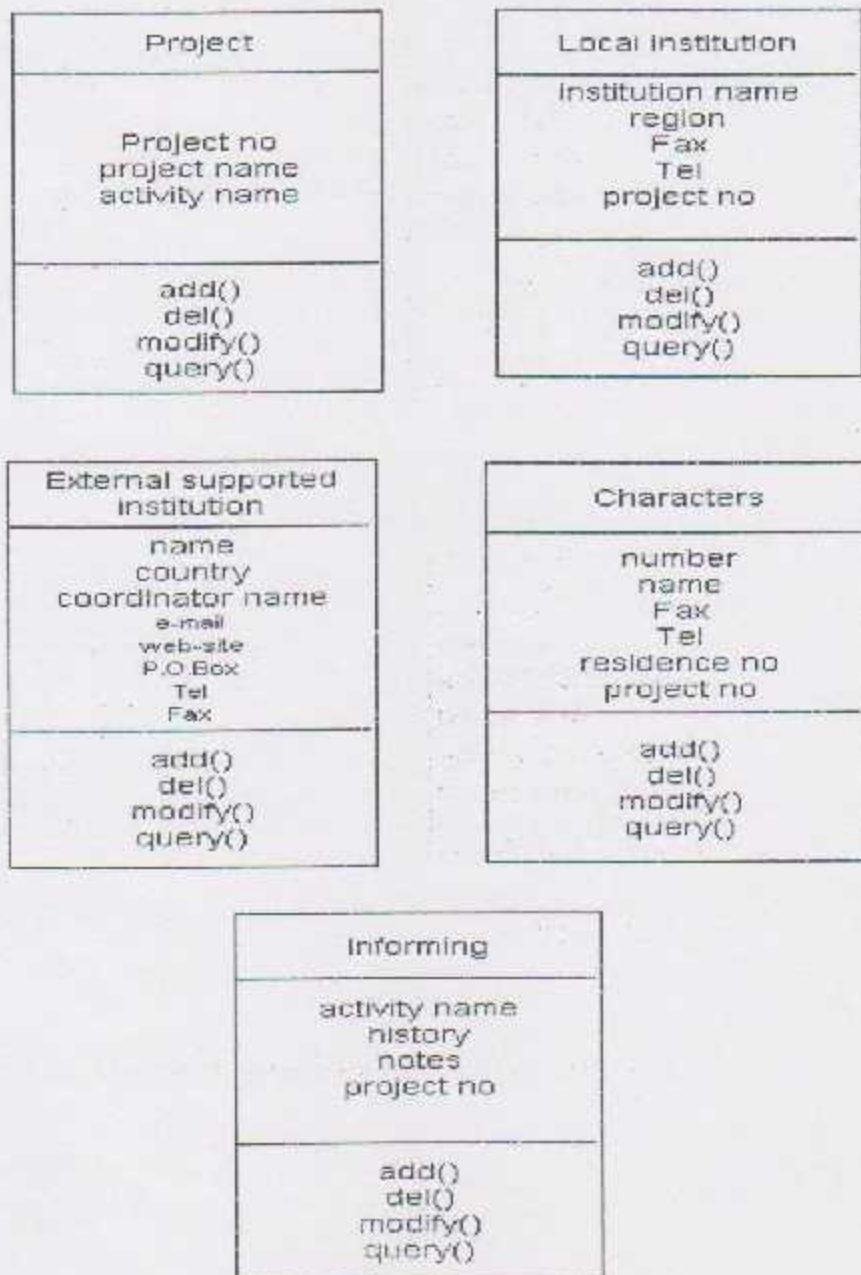


Figure (5.14) The object identification for public relations subsystem

3. The Employee Affairs Subsystem

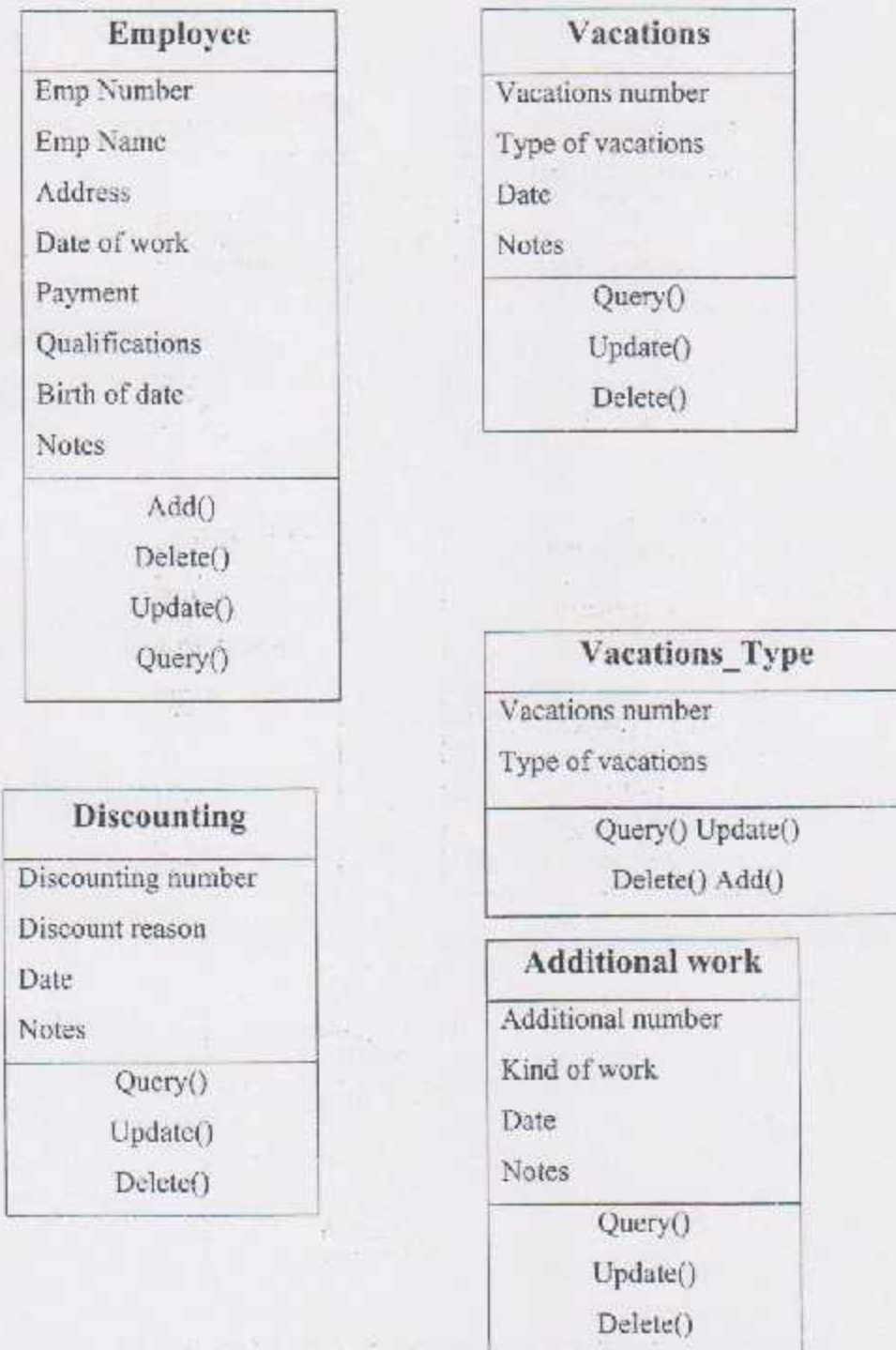


Figure (5.15) The object identification for employee affairs subsystem

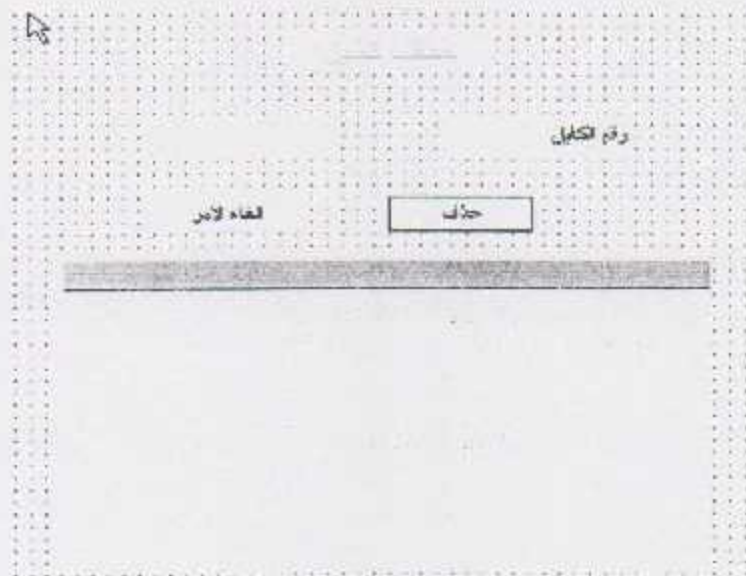


Figure (5. 21) Delete Sponsor Interface Screen



Figure (5.22) Add Sponsor Interface Screen

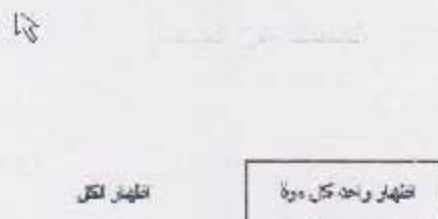


Figure (5.23) Query Sponsor interface Screen



Figure (5.24) Report Generator Interface Screen



Figure (5.25) Add Visit Interface Screen

عدد الزلا	11	موزعين حسب
الاصفال	10	اشهار
الشباب	1	اشهار
النشابات	0	اشهار
عدد الكلاء	5	اشهار
مبلغ التبرع	108775	
متبقي	188475	
عدد الزلا	2	اشهار

اشهار انزلا = اثنين بدون كلاء

موافق طاعة

Figure (5. 26) General Information Interface Screen

2. The Public Relations Sub System

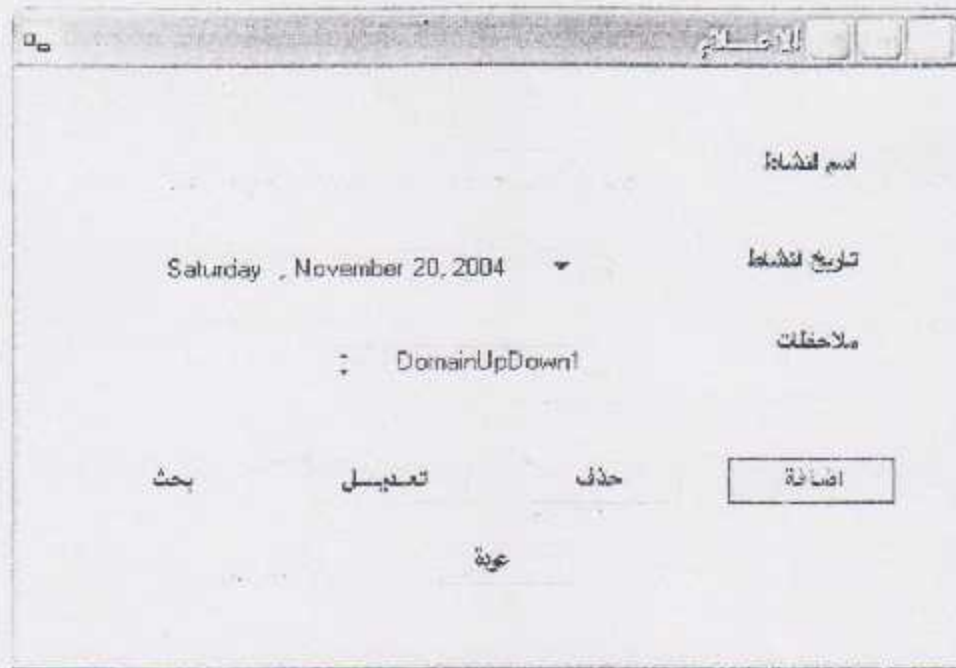


Figure (5.27) The informing department



Figure (5.28) The external supported institution department

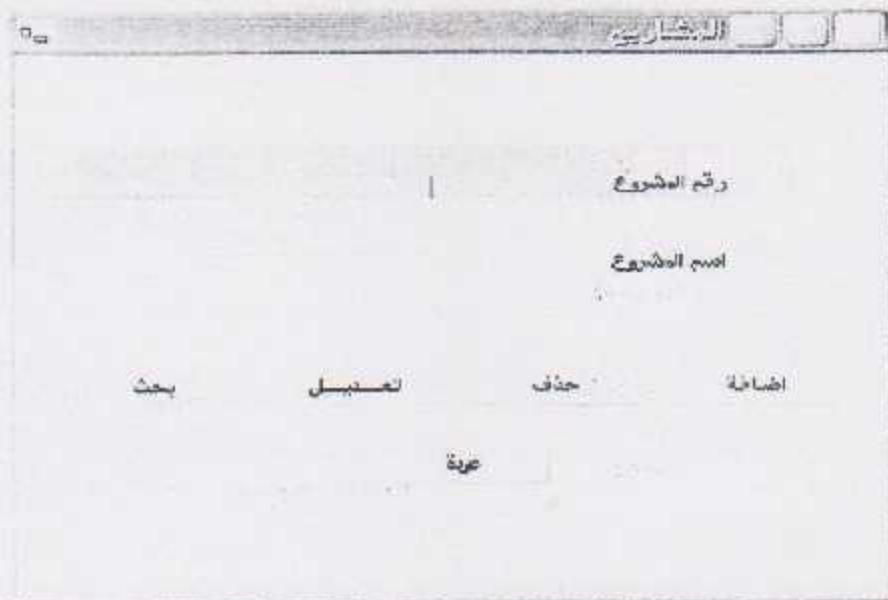


Figure (5.29) The project department

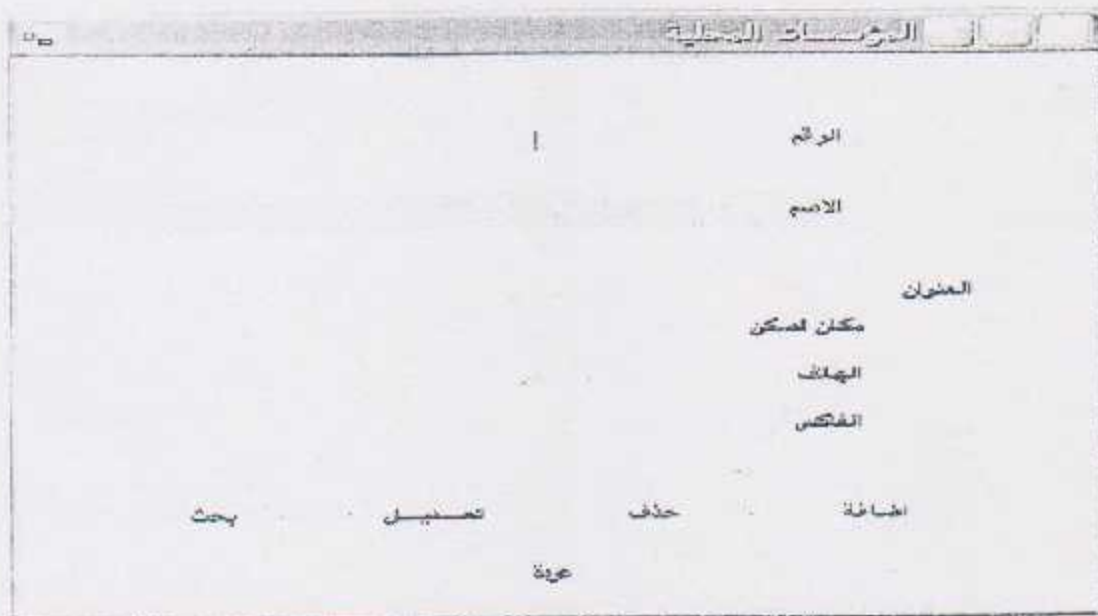


Figure (5.30) The Local Institution Department

3. The Employees Affairs Sub System

This part consists of the screens that the administrator can view and access through the system.

These screens are:

- Task selection screen



Figure (5.31) The administrator tasks screen

- Employee Add screen



Figure (5.32) The Add Employee Screen

- Employee update screen



Figure (5.33) The Update Employee Screen

- Employee delete screen

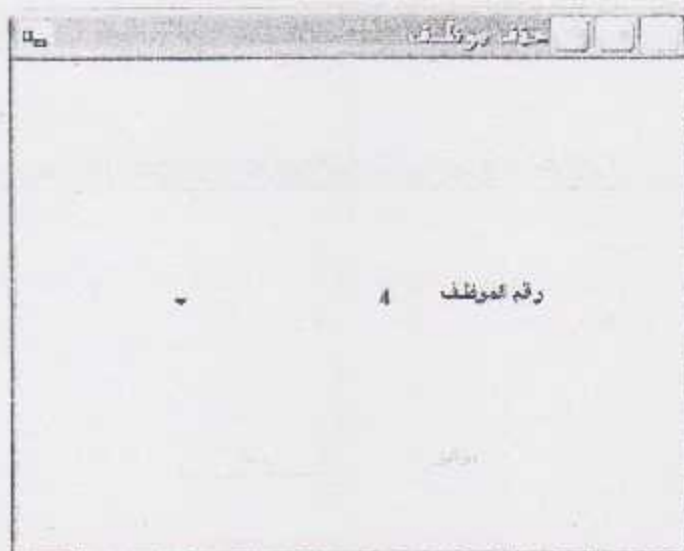


Figure (5.34) The Delete Employee Screen

- Employee Search Screen

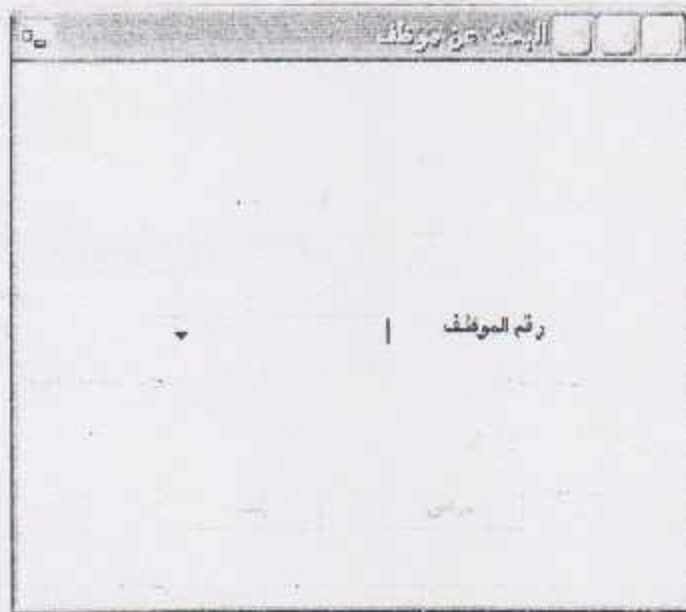


Figure (5.35) The Search Employee Screen

Introduction
Setting up the Required Software and Hardware
Database implementation
Database and windows forms interfacing

System Implementation

Chapter Six

6.1 Introduction

This system is a windows application that depends on a number of technologies that needs to be installed; there exist two main types of software that our system depends on, the software packages belong to the same company that is Microsoft.

This system is built basically on a group of Microsoft technologies such as the MS SQL server 2000 and VS.NET 2003, no other software package will be needed during development

This chapter describes the packages of software used in the system for the operation phase and how they are installed and prepared for works.

6.2 Setting up the Required Software and Hardware

6.2.1 Setting up the Hardware and Operating Systems

The hardware equipments needed for the operation of this system are:

- A Pentium I, II, III or IV.

For distributed system

- A Data Base Server.
- Networking equipment such as cabling, network adapters, and hubs.

More information see chapter 3 Requirement

The operating system required for the server in the distributed mode is Microsoft windows 2003 server or windows XP, and on the development PC we recommended Microsoft windows XP professional.

This diagram shows the relation between the tables in the database. It is a general graphical diagram that represents the fields and the relations they reference.

6.3.1 Database Relationships Diagram

The database is implemented using the SQL server and by checking the integrity and security rules of the system. The full description of the database is previously mentioned in the design chapter where the design was hierally applied in the implementation of the database. For more details on the database see Chapter 5 (system Design).

6.3 Database Implementation

The development of the system began with building the database using the SQL server, then building the front-end application using the Visual Basic.NET, and using ADO.NET with stored procedures and usual SQL command as the interface between the database and the windows forms of the system.

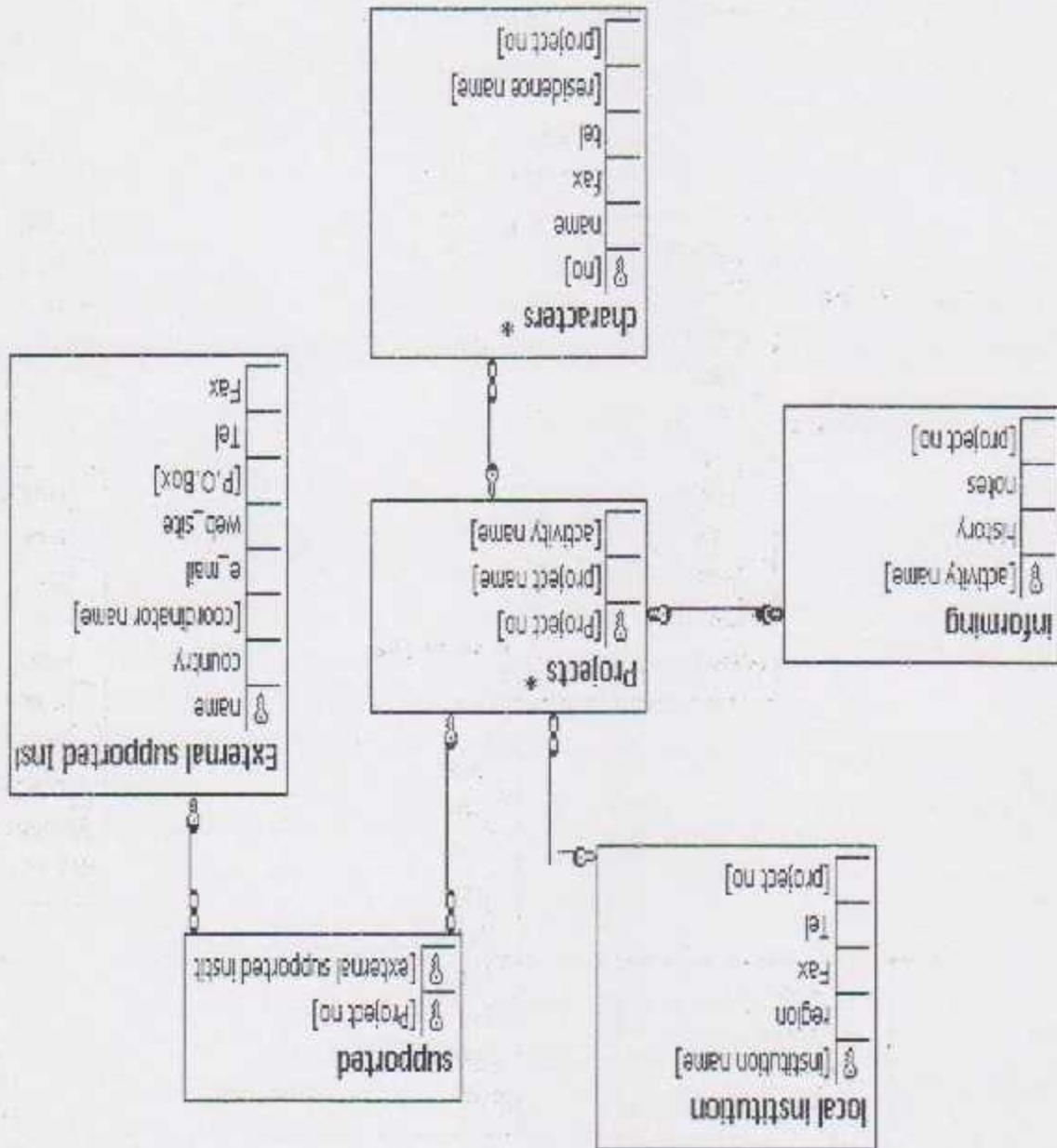
The .NET Framework work is a very important software package that must be installed, because it is the infrastructure of the .NET technology and it's needed to run all the .NET windows application.

- The .NET Framework Work
- Microsoft SQL server 2000 developer edition.
- Microsoft Windows XP professional or Home edition.

The software packages that this system is built on are:

6.2.2 Setting up the Software for the system:

Figure (6.2) The data relations diagram for the public relations subsystem



• The Public Relations Subsystem Relationship Diagram:

• *The Employee's Affairs Subsystem Relationship Diagram:*

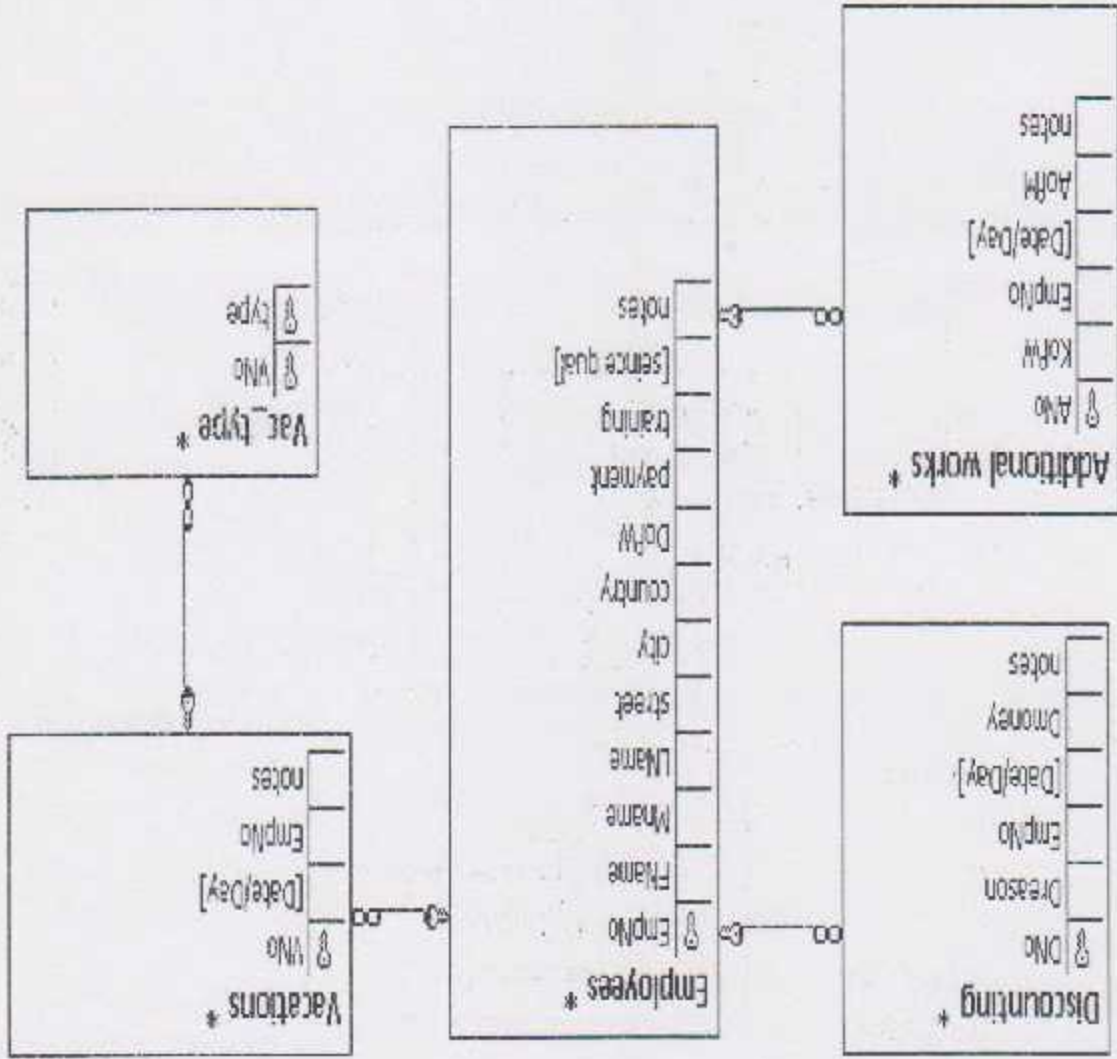


Figure (6.3) The data relations diagram for employee's affairs subsystem

6.4 Database and windows forms interfacing

The interfacing in this system is done through the ADO.NET technology using the stored procedures in the SQL server and a usual SQL command that can be executed through the ADO.NET directly. The windows forms developed in the Visual Basic.NET are linked to the database using SQL connections and SQL commands that can be mapped to stored procedures in the database of AL-IHSAN Society.

6.4.1 The interface between the database and system modules design

This section is a description of the objects that establish the interface medium between the system and the database. These objects are created using the ADO.NET technology to import the information from the database resources. They are used to access the database, read data, insert data, delete data, or update already existing data.

Following in this chapter is the list of these objects and the illustration of each object with its attributes and methods that are necessary to establish the database access.

6.4.1.1 The Interface Objects

The objects that are used in the AL-IHSAN system as interface objects between the database and the system modules are:

- The SqlConnection: this object establishes a connection with the database. It is necessary to access the database and used by the other objects in the interface between the database and the system modules such as the SqlDataReader.
- The SqlDataAdapter: this object acts as an agent between other objects of the ASP.NET and the database; it opens an already existing connection, reads, inserts, updates and deletes data from the database using built-in SQL commands or

defined commands. It can carry data from only one data source (a table in the database).

- The DataSet: an object that represents an in-memory cache of data brought by the SqlDataAdapter object from the database.
- The SqlCommand: this object executes an SQL command that is used for selecting, inserting, deleting and updating data in the database.
- The SqlDataReader: this object is used for reading data from the data source only; it is used to interact with only one control in the web application form such as a label.

6.4.1.2 The Interface Objects Attributes and Methods

The ADO.NET objects described above have some attributes and methods that contribute to their work of the database access, these methods and attributes are described in the following table:

Object name	Attributes	Methods
SqlConnection	<ul style="list-style-type: none"> • Connection string 	<ul style="list-style-type: none"> • Open() • Close()
SqlDataAdapter	<ul style="list-style-type: none"> • Selectcommand 	<ul style="list-style-type: none"> • Fill()
DataSet	<ul style="list-style-type: none"> • Tables 	None
SqlCommand	<ul style="list-style-type: none"> • CommandText • CommandType • Connection • Parameters 	<ul style="list-style-type: none"> • ExecyteReader() • ExecuteNonQuery() • ExecuteScalar()
SqlDataReader	<ul style="list-style-type: none"> • Item 	<ul style="list-style-type: none"> • Read() • Open() • Close()

Table (6.1) Interface objects of the ADO.NET

SQL commands are created in the VB.NET windows forms (Code behind page) and related to a certain connection out implementation will use a common connection that all the Data Base shared for any needed operation with the Data Base; they are mapped to a stored procedure in order to approve security, speed and development changes, the usage of parameters is optional, but in the following examples there are parameters used. The command then is assigned to a SqlDataReader or a SqlDataAdapter. These adapters and readers are used to access the database and read and manipulate data. Following are two examples of using the SqlCommand with stored procedures and the stored procedures for each one.

6.4.2 Implementation Example

In this section we will include some programmatically code for the implementation operation as followed.

6.4.2.1 ADD new resident example

The following procedure is to add new resident to the system, the procedure first check if all the needed information is entered then check if the family information is previously added then added the resident information, and the family information if needed then show the telephone form to insert the family telephone number.

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    'check for all textboxes to be filled
    Dim c As Control
    Dim i As Boolean = True
    For Each c In Me.Controls
        If TypeOf c Is TextBox Then
            If c Is TextBox15 Then
                i = i
            Else
                If c.Text = "" Then
                    i = False
                End If
            End If
        End If
    End For
End Sub
```

```

        End If
    End If
Next
'check for sponsor check without get one
If CheckBox1.Checked = True Then
    If TextBox16.Text = "" Or ComboBox2.SelectedIndex < 0 Then
        i = False
    End If
End If
If i = False Or ComboBox1.SelectedIndex < 0 _
    Or ComboBox3.SelectedIndex < 0 Then
    MsgBox("البيودج تعبئة اكمل")
Else

    Dim t As New Date((Now.Year - 2), Now.Month, Now.Day)
    Dim n As New Date(1950)

'check the birthdate and the range
    If IsDate(TextBox5.Text) Then
        If Convert.ToDateTime(TextBox5.Text) < (t) _
            And Convert.ToDateTime(TextBox5.Text) > n Then
            Dim sql3 As String = "select count(*) from family where IDnumber" & _
                & Val(TextBox8.Text)
            Dim cmd3 As New SqlCommand(sql3, res.con)
            Dim ee As Boolean = False
            Try
                res.con.Open()
'check if existing family
            Dim f As Int16 = cmd3.ExecuteScalar()
            If f < 1 Then
'add new resident with new family
                Dim cmd As New SqlCommand("new_res", res.con)
                cmd.CommandType = CommandType.StoredProcedure
                cmd.Parameters.Add(New SqlParameter("@ID", SqlDbType.NVarChar, 50))
                cmd.Parameters.Add(New SqlParameter("@fname", SqlDbType.NVarChar, 50))
                cmd.Parameters.Add(New SqlParameter("@aname", SqlDbType.NVarChar, 50))
                cmd.Parameters.Add(New SqlParameter("@lname", SqlDbType.NVarChar, 50))
                cmd.Parameters.Add(New SqlParameter("@city", SqlDbType.NVarChar, 50))
                cmd.Parameters.Add(New SqlParameter("@town", SqlDbType.NVarChar, 50))
            End If
        End If
    End If
End If

```

```
cmd.Parameters.Add(New SqlParameter("@street", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@notes", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@rosidentnum", SqlDbType.NVarChar,
50))
cmd.Parameters.Add(New SqlParameter("@fn", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@sn", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@ln", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@bdat", SqlDbType.DateTime))
cmd.Parameters.Add(New SqlParameter("@sex", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@money", SqlDbType.Money, 9))
cmd.Parameters.Add(New SqlParameter("@spnumber", SqlDbType.NVarChar,
50))
cmd.Parameters.Add(New SqlParameter("@pnumber", SqlDbType.Int, 8))
    Dim ii As Integer
    If CheckBox1.Checked = True Then
        ii = ComboBox1.SelectedValue
    Else
        ii = 0
    End If
    Try
cmd.Parameters(0).Value = TextBox8.Text
id = TextBox8.Text
cmd.Parameters(1).Value = TextBox9.Text
cmd.Parameters(2).Value = TextBox10.Text
cmd.Parameters(3).Value = TextBox11.Text
cmd.Parameters(4).Value = TextBox12.Text
cmd.Parameters(5).Value = TextBox13.Text
cmd.Parameters(6).Value = TextBox14.Text
cmd.Parameters(7).Value = TextBox15.Text
cmd.Parameters(8).Value = TextBox1.Text
cmd.Parameters(9).Value = TextBox2.Text
cmd.Parameters(10).Value = TextBox3.Text
cmd.Parameters(11).Value = TextBox4.Text
cmd.Parameters(12).Value = TextBox5.Text
cmd.Parameters(13).Value = ComboBox3.SelectedItem
cmd.Parameters(14).Value = CType(TextBox7.Text, Double)
cmd.Parameters(15).Value = ii
cmd.Parameters(16).Value = ComboBox1.SelectedIndex
cmd.ExecuteNonQuery()
```



```

MsgBox("تم اضافة العائلة")
Dim tell As New tell
tell.Show()
Me.Hide()
Catch ex As SqlException
MsgBox(ex.Message)
End Try
Else
'add new resident to existing family
Dim cmd As New SqlCommand("new_res0", res.con)
cmd.CommandType = CommandType.StoredProcedure
cmd.Parameters.Add(New SqlParameter("@residentnum", SqlDbType.NVarChar,
50))
cmd.Parameters.Add(New SqlParameter("@fn", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@sn", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@ln", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@bdat", SqlDbType.DateTime))
cmd.Parameters.Add(New SqlParameter("@sex", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@tmoney", SqlDbType.Money, 9))
cmd.Parameters.Add(New SqlParameter("@spnumber", SqlDbType.NVarChar,
50))
cmd.Parameters.Add(New SqlParameter("@pnumber", SqlDbType.Int, 8))
cmd.Parameters.Add(New SqlParameter("@ID", SqlDbType.NVarChar, 50))
Dim ii As Integer
If CheckBox1.Checked = True Then
ii = ComboBox1.SelectedValue
Else
ii = 0
End If
cmd.Parameters(0).Value = TextBox1.Text
cmd.Parameters(1).Value = TextBox2.Text
cmd.Parameters(2).Value = TextBox3.Text
cmd.Parameters(3).Value = TextBox4.Text
cmd.Parameters(4).Value = TextBox5.Text
cmd.Parameters(5).Value = ComboBox3.SelectedItem
cmd.Parameters(6).Value = CType(TextBox7.Text, Double)
cmd.Parameters(7).Value = ii
cmd.Parameters(8).Value = ComboBox1.SelectedIndex
cmd.Parameters(9).Value = TextBox8.Text

```



```

        Try
            cmd.ExecuteNonQuery()
            MsgBox("موجودة عائلة الي الاضافة تمت")
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End If
    Catch ex As SQLException
        MsgBox(ex.Message)
    End Try
    res.con.Close()
Else
    MsgBox("invalid data")
End If
Else
    MsgBox("invalid data")
End If
End If
End Sub

```

- ❖ The stored procedure that used to add the resident and family information:

CREATE PROCEDURE new_res

```

(@ID nvarchar(50),@fname nvarchar(50),@sname nvarchar(50),
@lname nvarchar(50),@city nvarchar(50),@town nvarchar(50),@street
nvarchar(50),@notes nvarchar(50),@residentnum nvarchar(50),@fn
nvarchar(50),@sn nvarchar(50),@ln nvarchar(50),@bdat datetime,@sex
nvarchar(50),@tmoney numeric(9),@spnumber nvarchar(50),@pnumber
numeric(9))

```

AS

BEGIN TRANSACTION

```

insert into family(IDnumber,fname,Sname,Lname,city,Town,Street,Notes)
values(@ID,@fname,@saame,@lname,@city,@town,@street,@notes)

```

```

insert into
resident(residentnum, fname, sname, lname, bdate, sex, Tmoney, Spnumber, pnumber, Id
number)
values(@residentnum, @fn, @sn, @ln, @bdat, @sex, @tmoney, @spnumber, @pnu
mber, @ID)

```

```

IF @@ERROR <> 0
BEGIN
    ROLLBACK
    RETURN
END
COMMIT TRANSACTION

```

GO

- ❖ The stored procedure that used to add new resident when the family has previously added by registering new resident

CREATE PROCEDURE new_res0

```

(@residentnum nvarchar(50), @fn nvarchar(50), @sn nvarchar(50), @ln
nvarchar(50), @bdat datetime, @sex nvarchar(50), @tmoney
numeric(9), @spnumber nvarchar(50), @pnumber numeric(9), @ID nvarchar(50))

```

AS

```

Begin Transaction
Insert into
resident(residentnum, fname, sname, lname, bdate, sex, Tmoney, Spnumber, pnumber, Id
number)

```

```

values(@residentnum, @fn, @sn, @ln, @bdat, @sex, @tmoney, @spnumber, @pnu
mber, @ID)

```

```
IF @@ERROR <> 0
    BEGIN
        ROLLBACK
        RETURN
    END
COMMIT TRANSACTION
GO
```

6.4.2.2 The family telephone insert example

After insert the resident and family information for the first time, the telephone window will appear to add the telephone numbers of the family; the family can have much telephone number start from nothing up to 3 as followed

❖ The procedure that used for this process as followed:

```
'check for no telephone number is repeated
If TextBox1.Text <> TextBox2.Text And TextBox1.Text <> TextBox3.Text
And TextBox3.Text <> TextBox2.Text Then
Dim cmd As New SqlCommand("new_tel", res.con)
cmd.CommandType = CommandType.StoredProcedure

cmd.Parameters.Add(New SqlParameter("@tel", SqlDbType.NVarChar, 50))
cmd.Parameters.Add(New SqlParameter("@id", SqlDbType.NVarChar, 50))
cmd.Parameters(1).Value = add_res.id

    Try
        res.con.Open()
'add the first number
        If TextBox1.Text <> "" Then
            cmd.Parameters(0).Value = TextBox1.Text
            cmd.ExecuteNonQuery()
        End If
'add the second number

        If TextBox2.Text <> "" Then
```

```
cmd.Parameters(0).Value = TextBox2.Text
cmd.ExecuteNonQuery()
End If
'add the third number

If TextBox3.Text <> "" Then
    cmd.Parameters(0).Value = TextBox3.Text
    cmd.ExecuteNonQuery()
End If
Catch ex As Exception
    MsgBox(ex.Message)
End Try
Else
    MsgBox("القيم من تأكد")
End If
res.con.Close()
End Sub
```

- ❖ The stored procedure that used to add the telephone number is:

```
CREATE PROCEDURE new_tel
```

```
(@tel nvarchar(50),@id nvarchar(50))
AS
insert into telephone(Tnumber,Idnumber)
values(@tel,@id)
```

```
GO
```

6.4.2.3 The Resident and Family update example

The following procedure show the code that used to update the resident and family information, the update process will not take effect on the primary key for both tables:

- ❖ The procedure that used for this process as followed:

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim cmd As New SqlCommand("upd_res", res.con)
    cmd.CommandType = CommandType.StoredProcedure

    cmd.Parameters.Add(New SqlParameter("@ID", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@fname", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@sname", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@lname", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@city", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@town", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@street", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@notes", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@residentnum", SqlDbType.NVarChar,
50))

    cmd.Parameters.Add(New SqlParameter("@fn", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@sn", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@ln", SqlDbType.NVarChar, 50))
    cmd.Parameters.Add(New SqlParameter("@bdat", SqlDbType.DateTime))
    cmd.Parameters.Add(New SqlParameter("@tmoney", SqlDbType.Money, 9))
    cmd.Parameters.Add(New SqlParameter("@spnumber", SqlDbType.NVarChar,
50))

    cmd.Parameters.Add(New SqlParameter("@pnumber", SqlDbType.Int, 8))

    Dim ii As Integer
    If CheckBox1.Checked = True Then
        ii = ComboBox2.SelectedValue
    Else
        If TextBox17.Text = "لا يوجد" Then
            ii = 0
        Else
            ii = Val(TextBox17.Text)
        End If
    End If

    Try
        res.con.Open()
        cmd.Parameters(0).Value = TextBox8.Text
        cmd.Parameters(1).Value = TextBox9.Text
    
```

```

cmd.Parameters(2).Value = TextBox10.Text
cmd.Parameters(3).Value = TextBox11.Text
cmd.Parameters(4).Value = TextBox12.Text
cmd.Parameters(5).Value = TextBox13.Text
cmd.Parameters(6).Value = TextBox14.Text
cmd.Parameters(7).Value = TextBox15.Text
cmd.Parameters(8).Value = TextBox1.Text
cmd.Parameters(9).Value = TextBox2.Text
cmd.Parameters(10).Value = TextBox3.Text
cmd.Parameters(11).Value = TextBox4.Text
cmd.Parameters(12).Value = TextBox5.Text

cmd.Parameters(13).Value = CType(TextBox7.Text, Double)
cmd.Parameters(14).Value = ii
cmd.Parameters(15).Value = ComboBox1.SelectedIndex
cmd.ExecuteNonQuery()
MsgBox("done")
Me.Hide()

Catch ex As Exception
    MsgBox(ex.Message)
End Try
res.con.Close()
End Sub

```

- ❖ The stored procedure that used to update resident and family is:

CREATE PROCEDURE upd_res

```

(@ID nvarchar(50),@fname nvarchar(50),@sname nvarchar(50),
@lname nvarchar(50),@city nvarchar(50),@town nvarchar(50),@street
nvarchar(50),@notes nvarchar(50),

```

```

@residentnum nvarchar(50),@fn nvarchar(50),@sn nvarchar(50),@in
nvarchar(50),

```

```

@bdat datetime,@tmoney numeric(9),@spnumber nvarchar(50),

```

```

@pnumber numeric(9))

```

AS

```
BEGIN TRANSACTION
```

```
update family set
```

```
fname=@fname,Sname=@sname,Lname=@lname,City=@city,Town=@town,Street=@street,Notes=@notes
```

```
where IDnumber=@ID
```

```
update resident
```

```
set
```

```
fname=@fn,sname=@sn,lname=@ln,bdate=@bdat,Tmoney=@tmoney,Spnumber=@spnumber,pnumber=@pnumber
```

```
where residenum=@residentnum
```

```
IF @@ERROR <> 0
```

```
    BEGIN
```

```
        ROLLBACK
```

```
        RETURN
```

```
    END
```

```
COMMIT TRANSACTION
```

```
GO
```

6.4.2.4 The Resident Delete Example

The following procedure shows the code for the delete resident process:

First, check if the resident exists.

Second, check if there other resident belong to the same family and perform the operation as:

- 1- if there other resident to the same family → delete the resident data only
- 2- if there no resident to the same family → delete the resident and family

❖ The procedure that used for this process as followed:

```

Dim cmd As New SqlCommand("check_res", res.con)
    Dim s As String
cmd.CommandType = CommandType.StoredProcedure
cmd.Parameters.Add(New SqlParameter("@residenum", SqlDbType.NVarChar,
50))
cmd.Parameters(0).Value = TextBox1.Text

    Try
        res.con.Open()
        s = cmd.ExecuteScalar
    Catch ex As SqlException
        MsgBox(ex.Message)
    End Try

    If s <> "" Then
Dim sql As String = "select count(*) from resident where Idnumber=" + s
        Dim cmdd As New SqlCommand(sql, res.con)
        Dim i As Integer = cmdd.ExecuteScalar
        MsgBox(i)
        If i > 1 Then
MsgBox("اكمل العائلة حذف لا يمكن, العائلة لنفس تزيد من اكثر يوجد"
, MsgBoxStyle.OKCancel)
            If DialogResult.OK Then
                cmd.CommandText = "del_res"
MsgBox("are you sure to delete the resident information",
MsgBoxStyle.OKCancel)
                If DialogResult.OK Then
                    Try
                        cmd.ExecuteNonQuery()
                        MsgBox("resident deleted")
                        Me.Hide()
                    Catch ex As SqlException
                        MsgBox(ex.Message)
                    End Try
                Else
                    Me.Hide()
                End If
            End If
        End If
    End If

```



```

        End If
    End If
Else
    cmd.CommandText = "del_res"
    Try
        cmd.ExecuteNonQuery()
        ' cmd.CommandText = "del_fam"
        Dim cm As New SqlCommand("del_fam", res.con)
        cm.Parameters(0).Value = s
        cm.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End If
Else
    MsgBox("resident not found")
End If
res.con.Close()
End Sub

```

❖ The stored procedure that used to delete resident and family is:

1. CREATE PROCEDURE del_res

(@residenum nvarchar(50))

AS

delete from resident where residenum=@residenum

GO

2. CREATE PROCEDURE del_fam

(@id nvarchar(50))

AS

delete from family where IDnumber=@id

GO

6.4.5 The login example

This operation is a simple security method that depends on local access to the system; we used to read/write from/to a binary file where the login password is stored there, because we need fast operation and we not have a high level security we just use the operation to not allow any unauthorized access and the operation of changing the connection string needed to access to the system.

There is two type of users to the system

- User: access to the system and only deal with the DB operation
- Developer: access to the system and deal with DB operation, can change the password and can modify the connection string

❖ The procedure that used for this process as followed:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim stream As FileStream
    Dim Binarystream As BinaryReader
    Dim u, d As String
    Dim found As Boolean = False
    'read the stored password from file
    Try
        stream = New FileStream("./asd.dat", FileMode.Open)
        Binarystream = New BinaryReader(stream)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    Try
        'The user password
        u = Binarystream.ReadString
        the developer password
        d = Binarystream.ReadString
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    stream.Close()
```

```
If RadioButton1.Checked = True Then
    If u = TextBox1.Text Then
        Dim res1 As New res
        Me.Hide()
        res1.Show()
        res1.MenuItem25.Enabled = False
        found = True
    End If
End If
If RadioButton2.Checked = True Then
    If d = TextBox2.Text Then
        Dim res1 As New res
        Me.Hide()
        res1.Show()
        found = True
    End If
End If
If found = False Then
    MsgBox("user not found " & Chr(13) & "re enter the password")
End If
stream.Close()
End Sub
```

Chapter Seven

Testing

Introduction

Testing the individual operations

7.1 Introduction

The testing process of the system is the process of validation and verification of the system components separately and for the system as one unit. We test the system step after step; that after finish any operation we test this operation, for this we will include some examples here.

7.2 Testing the individual operations

The operations that need to be tested are mainly "add" and "delete" operations because those are the operations that affect the system and the database.

7.2.1 Password Operation for the User and Developer

When the user or developer needs to use the system or change the password, the operations that are needed to accomplish that are.

The initialized login password will be:

For user → password = user
 For developer → password = dev

Process	Probability of situation	The Message	Show in Figure
<i>Enter the password.</i>	Invalid password.	Wrong password, retry to enter the correct password.	Figure(7.1)
<i>Change the password.</i>	Invalid old password.	There exists an error in password.	Figure(7.2)
	Mismatch the two new passwords.	Sure for new password.	Figure(7.3)
	Valid old and two new passwords.	Are you sure to change?	Figure(7.4)

Table (7.1) Password operation



Figure (7.1) Invalid password



Figure (7.2) Invalid Old Password



Figure (7.3) Mismatch the two new passwords



Figure (7.4) Valid old and two new passwords

7.2.2 Resident operation

When the user needs to add, delete, modify, and query the database there are some messages that may show up because entering invalid data or other problems.

The next table illustrates some of operations done by the user or developer.

Process	Probability of situation	The Message	Show in Figure
<i>Add new resident</i>	Not filling all required data.	Please enter the all required data.	Figure(7.5)
	Invalid date.	Please sure for the entering valid date.	Figure(7.6)
	All required data are filling and valid.	Adding successfully.	Figure(7.7)
	After adding is successful you add the valid phone #.	The phones are added.	Figure(7.8)
	Enter the existing resident number.	There exist resident with same number.	Figure(7.9)
<i>Delete resident</i>	No number is entered.	Enter the resident number.	Figure(7.10)
	Valid number.	You Sure for delete the resident.	Figure(7.11)

Update resident	Invalid resident number.	There resident not exist.	Figure(7.12)
Update resident	Resident exists.	The resident data are changed.	Figure(7.13)
Query resident	No number is entered.	Enter the resident number.	Figure(7.14)
	Invalid number (Resident not exists).	The resident not found.	Figure(7.15)
	Valid number (Resident is exists)	The resident is found	Figure(7.16)

Table (7.2) The resident operations



Figure (7.5) Not filling all required data

٥٥

إضافة تذوق

989453345	رقم الهوية	5	رقم التذوق
هادي	الإسم الأول	سامر	الإسم الأول
عامر	الإسم الأوسط	هادي	الإسم الأوسط
راضي	الإسم الأخير	راضي	الإسم الأخير
الخليج	العينة	2/22/2006	تاريخ الميلاد
يها			الجنس
الرئيسي			المبلغ المطلوب

تأكد من كتابة التاريخ بشكل صحيح

OK

الأطفال

يوجد كتيل

إلغاء الأمر مسح

Figure (7.6) Invalid date

٥٦

إضافة تذوق

550000745	رقم الهوية	5	رقم التذوق
هادي	الإسم الأول	سامر	الإسم الأول
عامر	الإسم الأوسط	هادي	الإسم الأوسط
راضي	الإسم الأخير	راضي	الإسم الأخير
الخليج	العينة	2/8/2000	تاريخ الميلاد
يها		ذكر	الجنس
الرئيسي		900	المبلغ المطلوب

تمت الاضافة

OK

الأطفال

يوجد كتيل

قيمة تبرع 100

55 قادي عرض

إلغاء الأمر مسح

Figure (7.7) All required data are filling and valid

□

إضافة هاتف

022295684

هاتف 1

052639258



مسافة

إضافة

Figure (7.8) After adding is successful you add the valid phone #

□

إضافة تلميذ

معلومات التلميذ

666666665

رقم الهوية

5

رقم التلميذ

خادي

الإسم الاول

بامل

الإسم الأول

سالم

الإسم لاوسط

خادي

الإسم لاوسط

الغول

الإسم الاخير

الغول

الإسم الاخير

الحليل

الإسم الاخير

3/6/1995

تاريخ الميلاد

واد الهوية

الإسم الاخير

022295684

رقم الهاتف

مغربي العجوري

الإسم الاخير

ذكر

الجنس

الإسم الاخير

600

المدارس

المدارس

يوجد كنيش

قيمة الترخ

99 خادي عوف

إلغاء الأمر

مسح

حفظ

Figure (7.9) Enter the existing resident number

□



Figure (7.10) No number is entered

□



Figure (7.11) Valid number



Figure (7.12) Invalid resident number.

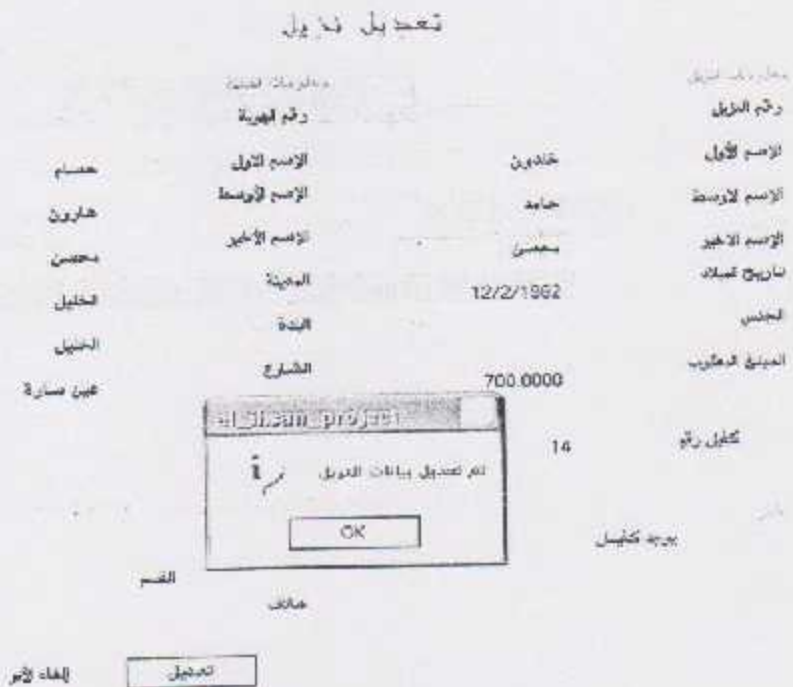


Figure (7.13) Resident exists

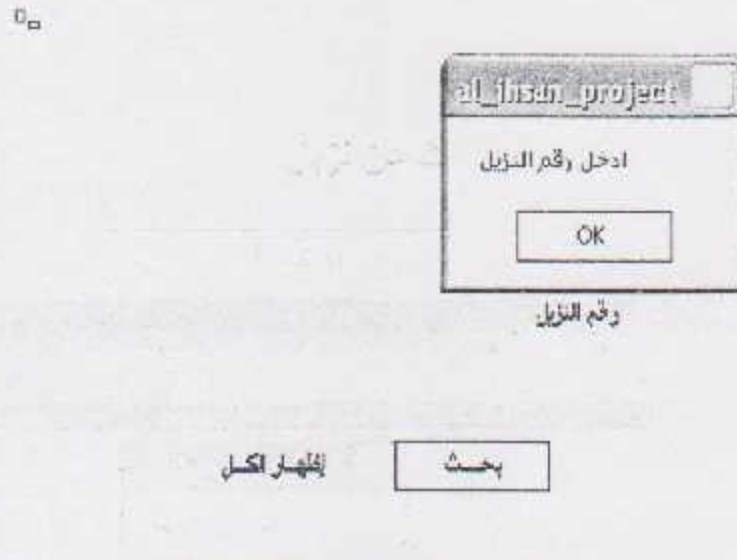


Figure (7.14) No number is entered

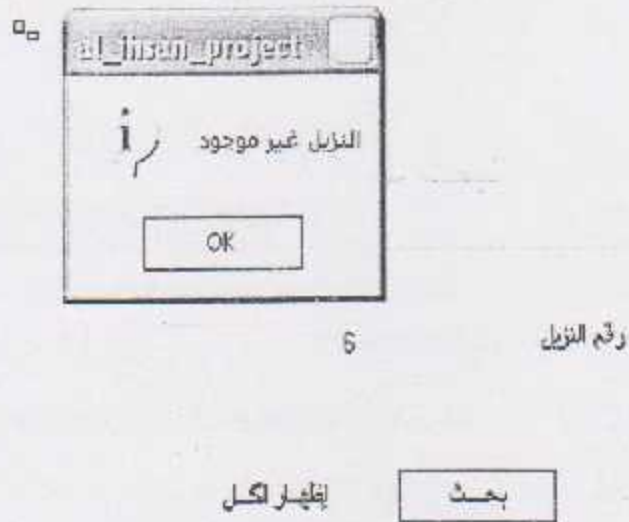


Figure (7.15) Invalid number (Resident not exists)

Process	Probability of situation	The Message	Show in Figure	Delete Sponsor	
				Valid sponsor number.	This process that direct effect on the data. Figure(7.20)
Add new sponsor	The sponsor number is exists.	All data are entered is correct.	The adding is successfully. Figure(7.17)	Valid sponsor number.	
				The sponsor number is exists. Figure(7.18)	
				All required data not entered. Figure(7.19)	
				Complete filling the form. Figure(7.19)	

When the user needs to add, delete, modify, and query the database there are some messages that may show up because entering invalid data or other problems. The next table illustrates some operations done by the user.

7.2.3 Sponsor operation

Figure (7.16) Valid number (Resident is exists)

الاسم	رقم السكن	تاريخ الميلاد	الجنس	رقم الهاتف	البريد الإلكتروني
515485464	1	1/1/2000	ذكر	200.0000	12

Update Sponsor	Invalid update	Please continue in your enter data	Figure(7.21)
	Invalid update	The update is done	Figure(7.22)
Query Sponsor	Invalid search	Please re enter the number	Figure(7.23)
	valid search	Ok searched is done	Figure(7.24)

Table (7.3) The sponsor operations

رقم الترخيل	5
الاسم الاول	قارس
الاسم الاوسط	حمد
الاسم الاخير	حمدان
رقم الهاتف	022563985
المدينة	الخليل
البلدة	يطا
الشارع	الركنيسي
رقم الهوية	983258147
قيمة التبرع	800

إضافة مسح إلغاء

Figure (7.17) All data are entered is correct



Figure (7.18) The sponsor number is exists



Figure (7.19) All required data not entered

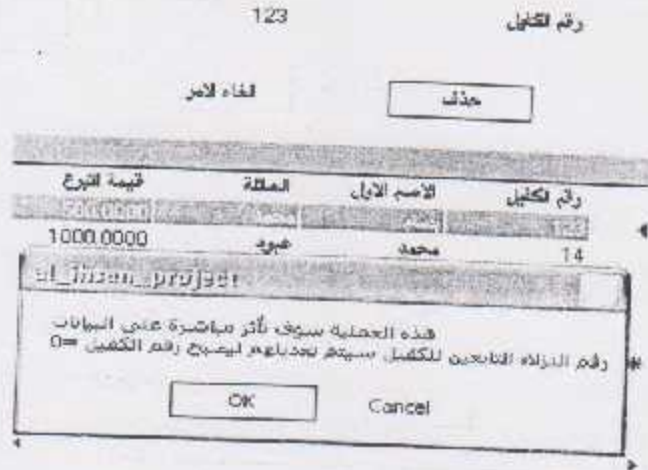


Figure (7.20) Valid sponsor number



Figure (7.21) Invalid update



Figure (7.22) Invalid update



Figure (7.23) Invalid search

الاسم الاول	الاسم الثاني	الاسم الاخير	المنية	البلدة	الشارع	رقم الهوية	قيمة التبرع	لمنتهي	رقم الهاتف
منير	احمد	سليم	الخليج	بيت كلن	الريسي	915137806	500.0000	100.0000	059689413

< 1 >

اختيار 20 رقم الكتيب

Figure (7.24) valid search

7.2.3 Visit operation

When the user needs to add, delete, modify, and query the database there are some of messages that may show up because entering invalid data or other problems.

The next table illustrates some of operations done by the user.

Process	Probability of situation	The Message	Show in Figure
<i>Determine new visit</i>	All data are entered is correct.	The adding is successfully.	Figure(7.25)
	The number is invalid	Re-enter the number for the resident	Figure(7.26)
<i>Special visits</i>	Valid number and data.	Ok you add this visit successfully	Figure(7.27)
	Valid number	Please re-enter the number for the resident	Figure(7.28)
	The data is not enough	Please complete the entered data	Figure(7.29)
	Correct complete visit	You completer the visit successfully	Figure(7.30)
	Incorrect complete visit	You must enter the found number for resident or there is no visits	Figure(7.31)

Table (7.4) The Visits sponsor operations



Figure (7.25) All data are entered is correct



Figure (7.26) The number is invalid



Figure (7.27) Valid number and data



Figure (7.28) Valid number



Figure (7.29) The data is not enough

رقم الزيارة	2
تاريخ الخروج	
اسم مشرف الخروج	
رقم الزيارة	
تاريخ العودة	1/23/2005
اسم مشرف الرجوع	محمد الملايحي
الغاء	<input type="button" value="كامل"/>

تم اتمام الزيارة بنجاح

Figure (7.30) Correct complete visit

رقم الزيارة	23
تاريخ الخروج	
اسم مشرف الخروج	
رقم الزيارة	
تاريخ العودة	
اسم مشرف الرجوع	
الغاء	<input type="button" value="كامل"/>

رقم الزيارة غير صحيح او لا يوجد زيارات

Figure (7.31) Incorrect complete visit

7.2.4 General Information

When the user needs to show all data in the database there are many options for that, the figure below shows the general form that we can use.

The screenshot shows a window titled 'معلومات عامة' (General Information). It contains a table with the following data:

عدد النزلاء	4	موزعين حسب:
الاطفال	2	اطهار
الشباب	1	اطهار
الشابات	1	اطهار
عدد الكفلاء	4	اطهار
مبلغ التبرع	3500	
متبقي	2700	
عدد النزلاء	3	اطهار

Below the table is a text box containing 'اطهار النزلاء الذين يكون كفلاء' (Orphan children who are sponsors), with a button labeled 'طباعة' (Print) to its left. At the bottom right, there is a button labeled 'مواقي' (Queries).

Numbered callouts point to the following elements:

- 1: Arrow pointing to the 'اطهار' field in the first row.
- 2: Arrow pointing to the 'اطهار' field in the second row.
- 3: Arrow pointing to the 'اطهار' field in the third row.
- 4: Arrow pointing to the '3500' value in the 'مبلغ التبرع' row.
- 5: Arrow pointing to the 'اطهار' field in the last row.
- 6: Arrow pointing to the text box containing 'اطهار النزلاء الذين يكون كفلاء'.

Figure (7.32) General information form

1- In this part the data will show as we see below:

residenum	fname	sname	lname	bdate	sex	Tmoney	Spnumber	prnumber	idnum1
	احمد	عبدالحجاز	البيبي	1/1/2000	ذكر	200.0000	12	0	515461
123	محمد	عبد	البيبي	8/8/1999	ذكر	200.0000	12	0	978654

Figure (7.33) Resident in the children department

2- In this part the data will show as we see below:

residnum	lname	sname	lname	bdate	sex	Tmoney	Sprumber	pnumber	ldnumbe
	عبد الخليل	خليل	محمد	2/25/1979	ذكر	300.0000	20	1	9874554

Figure (7.34) Resident in the men department

3- In this part the data will show as we see below:

residnum	lname	sname	lname	bdate	sex	Tmoney	Sprumber	pnumber	ldnumbe
	فخرية	محمد	المعوضي	8/22/1972	انثى	400.0000	0	2	9343530

Figure (7.35) Resident in the girl department

4- In this part the data will show as we see below:

snumber	apnumber	lname	sname	lname	city	town	street	ldnumber	mon
	0	0	0	0	0	0	0	0	0
12	2267980	عبد الصمد	محمد	الغرايدي	الخليل	يظا	المداربي	864256743	1000.000
20	059609413	منير	احمد	سالم	الخليل	بيت كحل	لورينسي	915137806	500.0000
3	227675544	محيي الدين	عبد النبي	حازم	ام الفحم	ام الفحم	لورينسي	9787554543	2000.000

Figure (7.36) Sponsors information

5- In this part the data will show as we see below:

residnum	lname	sname	lname	bdate	sex	Tmoney	Sprumber	pnumber	ldnumbe
	احمد	عبد الجبار	الياسي	1/1/2000	ذكر	200.0000	12	0	5154554
123	محمد	محمد	ابراهيم	8/8/1399	ذكر	200.0000	12	0	9765543
22	عبد الخليل	خليل	محمد	2/25/1979	ذكر	300.0000	20	1	9374554

Figure (7.37) Residents how have sponsor

6- In this part the data will show as we see below:

residantun	fname	sname	lname	bddate	sex	Tmoney	Sprumber	pnumber	ldrname
	فخرية	محمد	أبوغوش	8/22/1972	لثي	400.0000	0	2	\$343630

Figure (7.38) Residents how haven't sponsor

7.2.4 The Reports

In this part, we want to show the report as two parts. The first one is the resident with his family, The second is the resident with his sponsor.



Figure (7.39) The report main form

1- Resident information with there family information:



Figure (7.40) The resident with family report

2- Resident information with there sponsor information

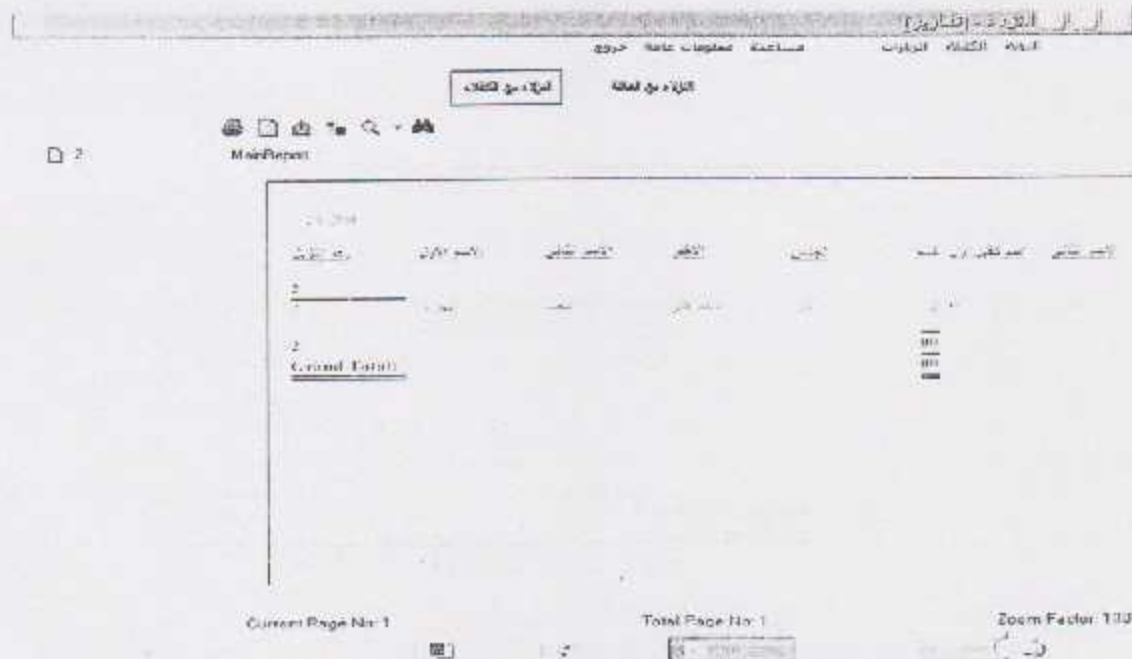


Figure (7.41) The resident with sponsor report

Chapter Eight

Installation

Introduction

Prerequisite

Application Installation

8.1 Introduction

This chapter will summarize the most important steps to setup the developed application to be ready for use.

8.2 Prerequisite

As we maintained in previous chapters there are some software systems that must be installed before the system installation. The software system packages can be summarized as follows:

- 1- The .NET Framework: this is the infrastructure of all the .NET technology, this installation package can be downloaded free from the Microsoft official web site(the package will be included with application installation CD)
- 2- The Microsoft SqlServer 2003 Enterprise Edition(for the server only in the distributed system)
- 3- Windows XP as operating system or windows 9x

8.3 Application Installation

- 1- The application package will be finished to have an executable setup file that will install all the required files on the user machine as shown in the next figure(8.1).



Figure (8.1) a- The package content

The installation wizard will start after launching the executable file and the steps shown in the next figures.

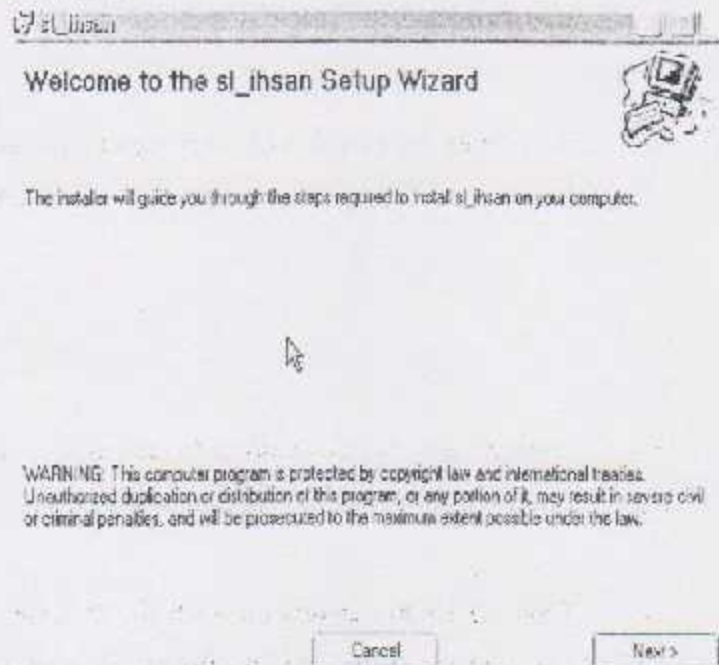


Figure (8.1) b- The first installation package



Figure (8.1) c- Select installation directory

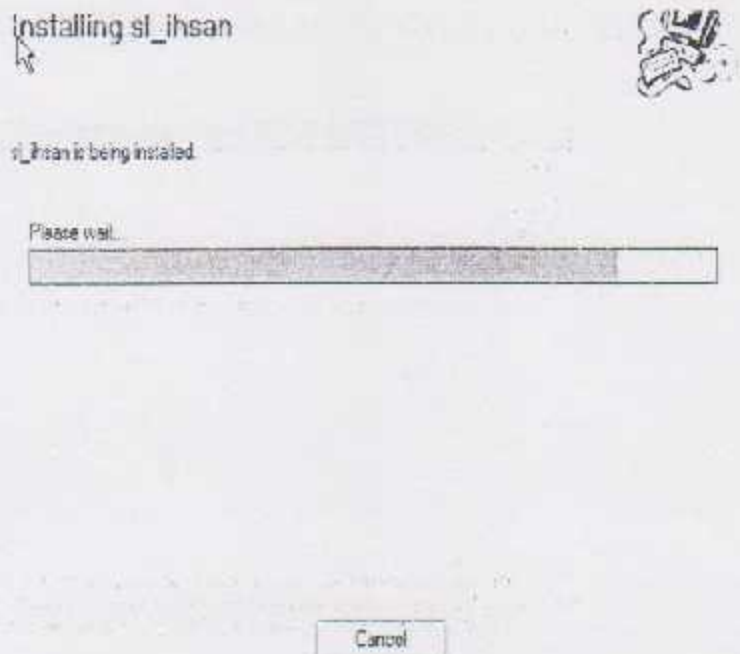


Figure (8.1) d- Installation in progress

Figure (8.1) Install the application

- 2- Attaching the Database to the SqlServer; the Database will be installed on the user machine and need to be attached to the SqlServer as shown in figure(8.2). First go to the Enterprise Manger that can found in the Microsoft SQL Server directory then right click on the Database and to all task then choose attach figure(8.1 a) then choose the location of the Database files figure(8.2 b).

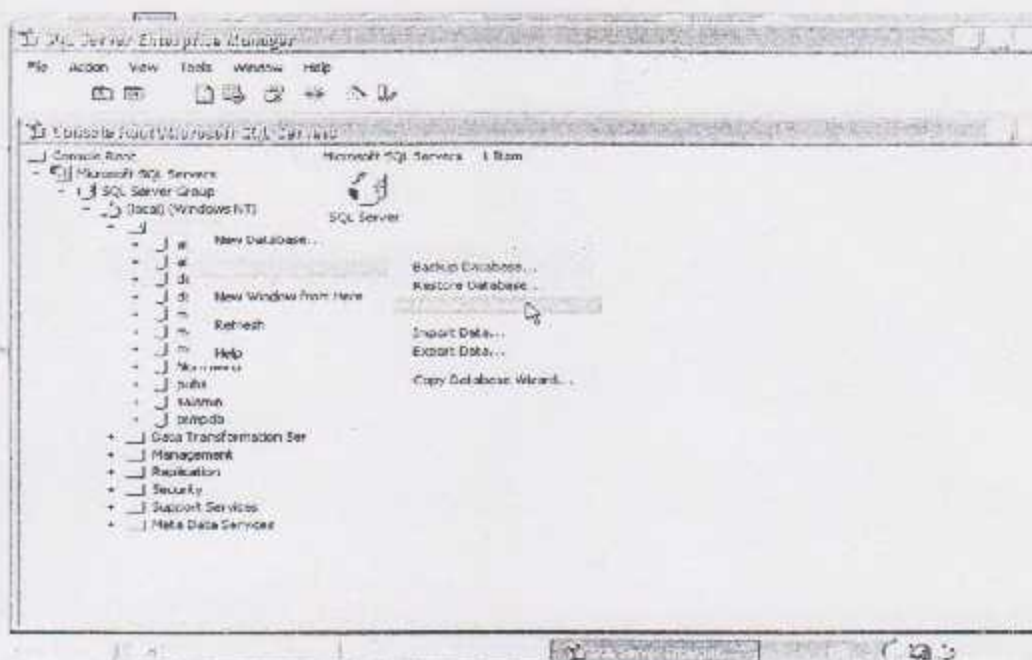


Figure (8.2) a-Enterprise manger

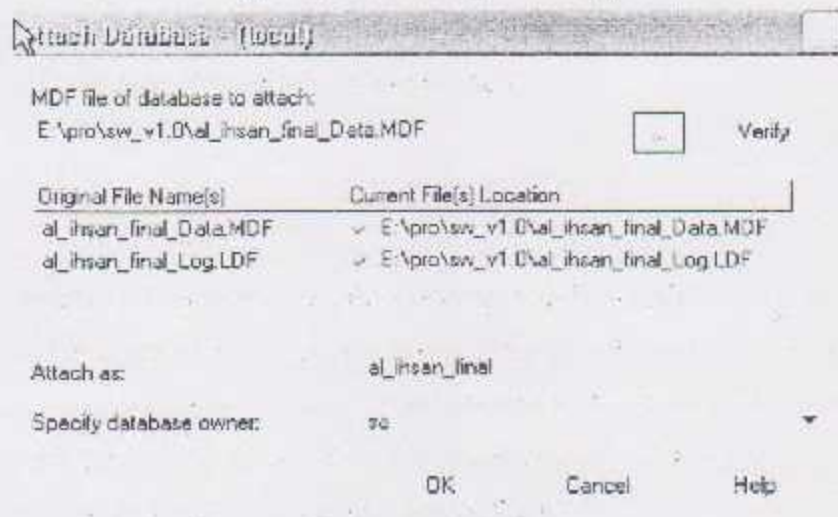


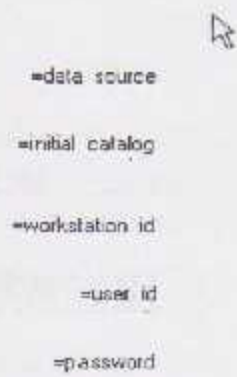
Figure (8.2) b- Specify the Database file location

Figure (8.2) Attaching the Database file to SQL Server

- 3- After installing the application we need to manipulate the connection string to match the installed computer information, this process need to start the application by access to the system using the developer access and changing the connection string as follows:



Figure (8.3) a- Choose to manipulate connection string



Test

Figure (8.3) b- Connection String Data

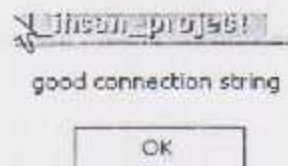


Figure (8.3) c- After Testing a Good Connection String

SQL Server does not exist or access denied.

OK

Figure (8.3) d- After Testing a non Working Connection String

Figure (8.3) Configure the connection string

Chapter Nine

System maintenance

Introduction

SQL Server maintenance

The .NET Framework maintenance

The System application

9.1 Introduction

This chapter provides the guidelines for the system administrator to keep tracing and maintaining the system after running it. The information provided in this chapter is built on the distinctiveness of the system which comes from the characteristics of the SQL server and the .NET Framework.

The areas of maintenance are divided into three integrated areas:

- SQL server maintenance.
- .NET Framework maintenance.
- System Application.

9.2 SQL Server Maintenance

In the console window of the SQL server you can find the objects of the server in the form of a tree. By double-clicking on any item or clicking the (+) sign on its root you can view its components.

The main component in our system is the database of AL-IHSAN; it contains all the tables and the stored procedures used for the system. Another component in the console window is the security branch and logins inside it, it determines the roles over the database and the privileges allowed for users, also you can create any user or login and give it the desired privileges over any database on the system.

Through these components you can control the SQL server and the database locally and trace any errors by viewing the log files stored in the database.

9.3 The .NET Framework Maintenance

If there any trouble with the .NET Framework you can use the .NET Framework installation package to re-install the .NET Framework file

9.4 The System Application

If there any trouble in the AL-HISAN system application that provide the interface between the user of the system and the Database we can use the installation package to repair the application without destroy the previous files

Chapter Ten

Conclusions and Recommendations

Introduction
Conclusions
Recommendations

10.1 Introduction

This system aimed to make a computerized system for AL-IHSAN society, which until now use the manual method in there management, and after the hard work we assume that we success to make computerized system for one subsystem, But there are some considerations that should be taken in grant that caused the system not to be as perfect as required; following are these considerations, conclusions and recommendations

- This system has been the experience of a limited time period and limited resources for the development phase.
- This system depends mainly on the worker in the AL-IHSAN society, but the system is not run yet.

Considering these circumstances the work team decided to take the following conclusion and recommendations.

10.2 Conclusions

The work team has concluded the following:

- The system is subject to upgrading and maintenance according to the users.
- The system is subject to be expanded to the rest subsystems of the society.

10.3 Recommendations

The work team recommends the following actions as the future work for the system:

- To make implementation for the other subsystem
- To fix any problem that we may not face it during the implementation time

Appendices

Appendix A

Appendix

Appendix A

Appendix A



التاريخ: /--- / ٢٠٠١ م

تعليمات بشأن إدخال نزلنا جنتنا إلى الجمعية

السيد /-----/ ولي أمر النزيل

السلام عليكم ورحمة الله وبركاته

نرحب بكم في جمعية الإحسان الخيرية ونذكركم أن من أهم أهدافنا التي نسعى لتحقيقها تقديم رعاية صحية وتأهيلية متميزة للنزلاء والزوار لذا نأمل منكم التعاون معنا والالتزام بما يلي :

- ١- إحضار ٤ صور شخصية للنزيل .
- ٢- إحضار التقارير الطبية .
- ٣- صورة عن شهادة الميلاد .
- ٤- ملابس داخلية وخارجية .
- ٥- صورة عن هوية الأب .
- ٦- صورة عن هوية النزيل (إن وجدت) .
- ٧- الالتزام بدفع الرسوم الشهرية في موعدها المحدد ومقدارها (شكل قابلة للتغيير .

٨- الالتزام بالزيارة الشهرية في موعدها وهو (من كل شهر .

٩- الاستجابة لجميع مطالب الجمعية فيما يتعلق باحتياجات الأطفال .

نشكركم على تعاونكم

توقيع ولي أمر النزيل

المدير الإداري

عادل الضبيدي



اليوم والتاريخ: _____ / ____ / 200 م

تقرير إدخال فرد إلى الجمعية

الاسم:	تاريخ الميلاد:
العنوان:	رقم الهوية =
رقم الهوية:	الجنس:

تقرير طبي:

تقرير التأهيل:

تقرير العلاج الطبيعي:

تقرير العلاج الوظيفي:

تقرير أخصائي النطق:

تقرير الباحثة الاجتماعية:

توصيات رئيس التأهيل:

المدير الإداري:

شؤون البيئة الإدارية:



(للمعاقين المذكور)

التاريخ: / / ٢٠٠٠م

تذكرة والتزام

انا الموقع اسمي ادناه: _____ من _____
ولي امير التفصيل / النائب / المعاق / _____
اتعهد بالالتزام بقوانين وانظمة جمعية الإحسان الخيرية لرعاية المعاقين، واتعهد بدفع الرسوم
الشهريه وقدرها () شيكل / دينار في موعدها دون تأخير، وان اقوم باحضار الملابس اللازمة له.
كما واتعهد باحضار المذكور اعلاه الى بيتي في كل شهر مرة واحدة للاقاسة شهدي خمسة
ايام، بما في ذلك يوم الاسلام ويوم الاعادة. حتى أبقى على الترابط الاسري بيني وبينه.
وفي حالة عدم التزامي بذلك فللجمعية الحق في انهاء خدماتها تجاهه وارساله بالسيارة الى
بيتي على نفقتي الخاصة.

والله الشاهدني اني سواء الضمير

الاسم: _____

التوقيع: _____

رقم الهوية: _____

المدير الاداري للجمعية



نموذج طلب ادخال معاق الى الجمعية

التاريخ: / / 200 م

حضرات السادة رئيس وأعضاء الهيئة الادارية لجمعية الإحسان الخيرية - المعتمدين

الموضوع: / نائب ادخال معاق الى الجمعية

السلام عليكم ورحمة الله وبركاته وبعد:

انا الموقع اسمي ادناه من سكان _____

احمل هوية رقم _____

ارجو السوافة تولى ادخال المعاق الى _____ الى

الجمعية . حيث يبلغ من العمر _____ سنة وصلة قرابتي به _____ وهو بحاجة الى العناية والرعاية .

واتعهد بدفع مبلغ وقدره _____ (دينار / شيكل) شهرياً كمساهمة في النفقات التي تنفق عليه في الجمعية .

وفي حالة عدم وفائي بالمستلزمات المالية للجمعية الحق في توقيف خدماتها له . واعادته الى بيته .

معلومات حول الأسرة:

1- هل الوالدين على قيد الحياة؟ نعم / لا

2- وضع المعاق الاجتماعي: يتيم الأب / يتيم الأم / يتيم الأبوين / الأبوين على قيد الحياة.

الأخوة الذكور:

الرقم	الاسم	العمر	العمل
1.			
2.			
3.			
4.			
5.			

مع التسبب والاختتام

مقدم الطلب

الاسم: _____

التوقيع: _____

تلفون رقم: _____



التاريخ: / / ٢٠٠٠م

من / البحث الاجتماعي
إلى / قسم المحاسبة

بطاقة خروج نزيل

اسم النزيل:	تاريخ الادخال:
تاريخ الميلاد:	الرسوم الشهرية:
مكان الإقامة:	القسم:

سبب الخروج:

وفاة بتاريخ: / / ٢٠٠٠م

الأهل بتاريخ: / / ٢٠٠٠م

الجمعية بتاريخ: / / ٢٠٠٠م

اسم المسؤول:

التوقيع:



جمعية الاحسان الخيرية لرعاية وتأهيل المعاقين
قسم الشباب المعاقين / نموذج استلام وتسليم نزلاء القسم
الدوام : (24) ساعة متواصلة

قسم البحث الاجتماعي

الموظفون المرادون	النزلاء الذين هم في زيارة منزلية	النزلاء الموجودين في القسم	أيام الأسبوع والتاريخ
			السبت 200 / /
			الأحد 200 / /
			الاثنين 200 / /



رقم الطلب :

نموذج طلب عمل

دائرة شؤون الموظفين

تاريخ تقديم الطلب : / / ٢٠٠٠ م

العائلة

اسم الجد

اسم الأب

الاسم الشخصي

العسل المطلوب بالتحديد :

الاسم الكامل :

المعلومات الشخصية

انثى

ذكر : الجنس

١٩ / / م

تاريخ الميلاد :

مكان الولادة

عدد الاولاد

متزوج

اعزب : الحالة العائلية

رقم الهوية :

المؤهلات العلمية والخبرات :-

سنة التخرج :

الجامعة :

التخصص :

الشهادة :

الدورات التي شارك فيها : ١.

٢.

الخبرات العملية :

غير مدخن

مدخن ارجيلة

مدخن جزلي

مدخن كامل

هل أنت مدخن ؟

العنوان

رقم الهاتف :

العنوان الحالي :

مكان الإقامة :

اسماء ثلاثة معروفين : ١.

٢.

٣.

للاستعمال الرسمي فقط

تعلق مسؤول شؤون الموظفين

تعلق المدير الإداري :

قرار الهيئة الإدارية :



السادة رئيس وأعضاء الهيئة الإدارية لجمعية الإحسان الخيرية / المحترمين

بواسطة / حضرة السيد / مسؤول شؤون الموظفين في الهيئة الإدارية / المحترم

بواسطة / حضرة السيد مدير جمعية الإحسان الخيرية / المحترم

صله :

اسم مقدم الطلب :

الموضوع :

السلام عليكم ورحمة الله وبركاته وبعد :

توقيع مقدم الطلب :

التاريخ :

ملاحظات مسؤول قسم شؤون الموظفين الميداني :

تعليق مدير الجمعية :

تعليق مسؤول شؤون الموظفين في الهيئة الإدارية :

قرار الهيئة الإدارية :

رئيس الجمعية

د. عصام التميمي

References

1. Sommerville, Ian, *Software Engineering*, 6th edition, Addison-Wesley, 2001.
2. www.microsoft.com
3. Martin, Robert C., *UML for JAVA Programmers*, Prentice Hall, 2003.
4. Microsoft SQL server 2003 Help Topics.
5. Microsoft .NET Framework SDK v1.1.