University of Baghdad
College of sciences
Computer sciences
Department

# DATABASE DESIGN AND IMPLEMENTATION USING VISUAL BASIC

**By:**

*Mohannad Taha & Saad fadhil*

**Supervised by:**

*AmEnah Dahim Abood*

# Acknowledgement

The praise for Allah. Then I would like to express my deepest gratitude to my supervisor, Aminah Dahim , for her continuous support &guidance throughout my academic study. Lastly but not least,

I thank my unforgettable friends for their encouragement. To them I express my love & affection.

# Context

**Chapter 1 (**Database**)** ....................................... 1

**1.1 Introduction** ................................................ 1

**1.2 Database Management System (DBMS)** ................ 2

**1.3 DBMS Functions** ........................................... 3

**1.4 Database Models** ........................................... 4

**1.4.1 Conceptual Models** ............................................... 4

**1.4.2 Implementation Models** ............................... ......................................5

**Chapter 2(**Database System**)** ............................ 6

**2.1 Database system** ......................................... ..........................6

**2.2 Features of DBMS** .......................................... 7

**2.3 Properties of Transaction**..................................... 9

**Chapter 3(**Design and implementation**)** ................. 10

**3.1 Design of the Database** ...................................... 10

**3.2 How communicate with the Database (D.B.) through the Visual Basic** ............................................... 16

**Chapter 4(System Testing)** ................................ 20

**4.1Testing the program** ........................................ 21

**4.2 Administrator** ......................................... ....... 24

**4.3 Creating user** .......................................... 25

**4.4 User Modification** ...............................................................26

**4.5 Log in** ......................................... .............................................29

**4.6 Adding(books)** .............................................. 30

**4.7 Editing(books)** ............................................... 31

**4.8 Searching** .......................................... 32

**4.9 Borrowing** ............................. 33

**4.10 Retriving** ............................. 34

**4.11 About the program** ............................. 35

References .......................................................... 36

# Chapter1

# Database

## 1.1 Introduction:

In the recent past, a manger of almost any small organization was able to keep track of necessary data by using a manual file system. Such a file system was traditionally composed of a collection of file folders; each properly tagged and kept in a filing cabinet [1].

Organization of the data within the file folders was determined by the data's expected use [1].

Ideally, the content of each file folders where logically related. For example, a file folder in a doctor's office might contain patient data, one folder for each patient. All the data in that file older described only that particular patient's medical history [1].

As long as a data collection relatively small and an organization's mangers had few reporting requirements, the manual system served its role well as a data repository. However, as organizations grew and as reporting requirements became more complex, keeping track of data in a manual file system became more difficult. In fact, finding and using data in growing collection of file folders became such a time-consuming and cumbersome task that it became less and less likely that such data would ever generate useful information [1].

Unfortunately, report generation from a manual file system can be slow and cumbersome. In fact, some business mangers faced government-imposed reporting requirements that require weeks of intensive effort each quarter, even when a well-designed manual system was used. Consequently, the pressure built design a computer based system that would track and produce required reports [1].

The problems inherent in file systems make using a database system very desirably. Unlikely the file system, with its many separate an unrelated files, the database consists of logically related data stored in a single data repository. Therefore; the database represents a change in the way end user data are stored, accessed, and managed. The database management system (DBMS) provides numerous advantages over file system management, by making it possible to eliminate most of the file system's data inconsistency, data dependency, and structural dependency problems [1].

Remember that the DBMS is just one of several crucial components of a database system [1].

Perhaps it is even appropriate to refer to the DBMS as the database system's heart. However, just as it takes more than a heart alone to make a human being function, it takes more than a DBMS to make a database system function [1].

**Database** is a collection of data or information that can be stored, sorted, organized and retrieved. Your local telephone book, your Rolodex file, and the card catalog at your local library are all examples of a database.

Traditionally databases are organized by fields, records, and files tables [1].

A is single piece of information; a record is a one complete set of fields; a file is a collection of records. For example, a telephone book is analogous to a file. It contains a list of records, each of which consists of three fields: name, address, and telephone number [1].

To access information form a database, you need a database management system (DBMS). This is a collection of programs that enables you to enter, organize, and select data in a database [1].

## 1.2 Database management system (DBMS):

DBMS is a collection of programs that manages the database structure a controls access to the data stored in the database. The DBMS makes its possible to share the data in the database among multiple application or users. Because the data are the crucial raw material from which information is derived, there are many reasons why DBMSs are important in our information-based [1].

The goal of a DBMS is to provide an environment that is both convenient and efficient to use in [1]:

- Retrieving information from the database.
- Storing information into database.

In effect, the DBMS serves as the intermediary between the user and the database by translating user requests into the complex code required to fulfill those requests. The DBMS hides much of the database's internal complexity from the application programs that use the database. The application program might be written by a programmer, using a programming language such as COBOL, Microsoft Access, and Oracle or it might be created through a DBMS utility program [1].

## 1.3 DBMS Functions:

A DBMS perform several important functions guarantee the integrity consistency of the data in the database. Most of these functions are transparent to end users. These functions include [1]:

- **Data Dictionary Management:** the DBMS requires that definitions of the data elements and their relationship (metadata: data about data) be stored in a data dictionary.
- **Data Storage Management:** the DBMS creates the complex structure required for the data storage.
- **Data Transformation and presentation:** the DBMS transforms entered to conform the data structures are required to store the data.
- **Security Management:** the DBMS creates a security system that enforces user security and data privacy within the database. Security rules determine which users can access the database, which data item each user may access, and which data operations that user may perform.
- **Multi-User Access Control:** the DBMS creates the complex structure that allows multi-user access to the data. In order to provide data integrity and data consistency, the DBMS uses sophist iced algorithms to ensure that multiple user can access to the database concurrently and still guarantee the integrity of the database.
- **Back and Recovery Management:** the DBMS provides backup and data recovery procedures to ensure data and integrity.
- **Data Integrate Management:** the DBMS prompts and enforce integrity rules to eliminate data integrate problems, thus minimizing data redundancy and maximizing data consistency.

- **Database Access Language (DLL and DML) and application programming interface:** the DBMS provides data access via a query language. A query language is a nonprocedural language; which contains two components: a Data Definition Language and a Data Manipulation Language (DML).
- **Database Communication Interfaces:** Current-generation DBMSs provide special communications routines designed to allow the database to accept end user request within a computer network environment.

## 1.4 Database Models:

**A database model** is a collection of logical construct used to represent the data structure and the data relationships found within the database. Database models can be grouped into two categories: **Conceptual Models** and **implementation models** [1]**.**

### 1.4.1 Conceptual Models:

Use three types of relationship to describe association among data: 1: M, M: N, and 1:1 for them, respectively. (Although the M: N notation is a standard label for the many-to-many relationship, the label M: M may also be used.). The following examples illustrate the distinction among the three [1]:

- **One-to-many relationship:** Each record in a table is related to many records in another table. For example: a painter paints many different painting, but each one of them is painted by only that painter. Thus the painter (the "one") is related to the paintings (the "many"). therefore; database designers label the relationship"PIANTER paints PAINTING" as 1: M .similarly, a costumer account (the "one") might contain many invoices, but those invoices (the "many") are related to only a single costumer account. The" CUSTOMER generates INVOICE" relationship would also label 1: M.
- **Many-to-many relationship**: many records in table are related to many records in another table. for example, Student can take many Course, and each course can be taken by many student, thus yielding

the M:N relationship label for the relationship expressed by”STUDENT takes COURSE.”

- **One-to-one relationship:** each record in a table is related to one record in another table .Dean of the collage us related with collage in 1:1 relationship , each collage has one Dean and each Dean is managed one Collage.

  As seen this relation benefit, but in this project found that making this relationship between tables will add over head to the system .therefore; this relation ship not used this project.

### 1.4.2 implementation module:

Places the emphasis on how the data are represented in the database or on how the data structures are implemented to represent what is modeled. Implementation models include the hierarchical database model, the network database model, the relational database model, and the object-oriented database model [1].

In this project the module used is the relational database model, because this model provides the following basic Structure [1]:

- The relational database model is implemented through a very sophisticated relational database management system (RDMBS). The RDMBS performs the same basic function provided by the hierarchical and network DBMS system plus a host of other function that make the relational database module easier to understand and implemented.
- The data and relationship are represented by a collection of tables.
- Relational model does not use pointer or links but relates records by the value that contains (P.K., F.K.)
- Each table in matrix consisting of a series of row/column intersection. Tables, also called relations, are related to each other by sharing a common entry characteristic.
- Although the tables are completely independent of one another, we can easily connect the data between tables. The relational model thus provides a minimum level of controlled redundancy to eliminate most of the redundancies commonly found in file systems.

# Chapter 2

# Database System

## 2.1 Database system:

Is basically a communized record keeping system that is a system whose overall purpose is to maintain information and to take that information available on demand [2].

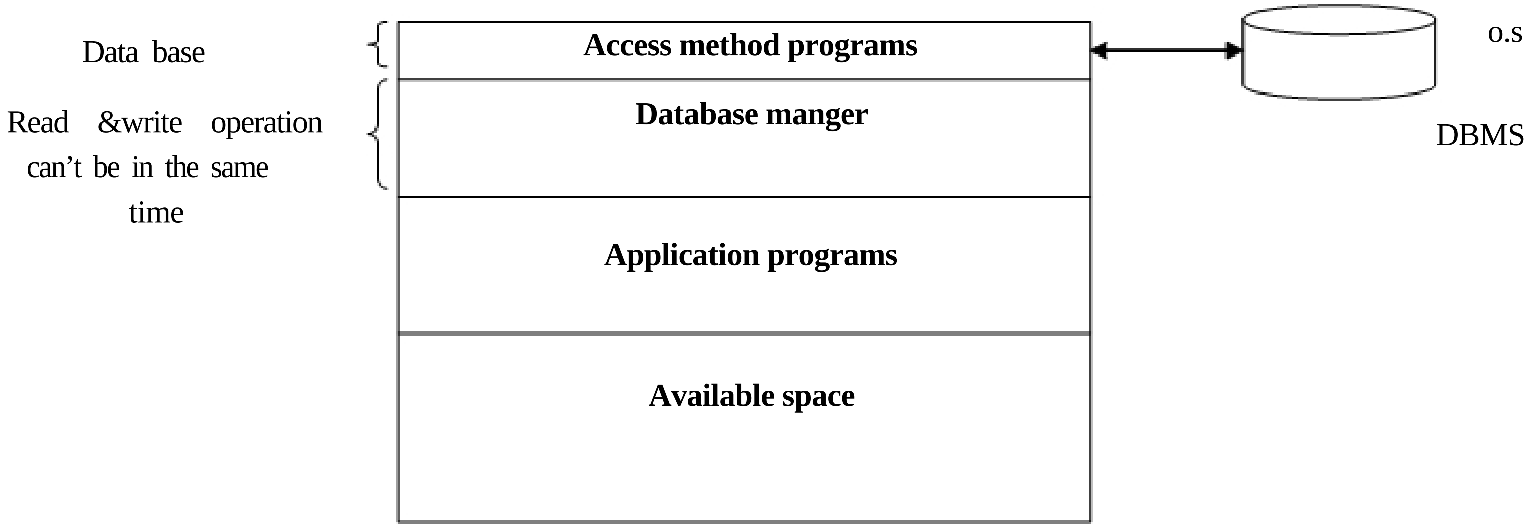
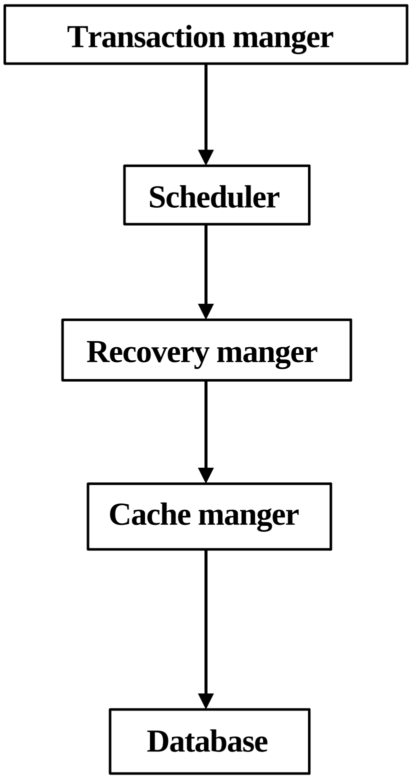


**Fig. (2.1) simplified structure of database**

The benefit of the database system manger (DBMS) appear when multitask processing exist [2].

## 2.2 Features of DBMS [2]:-

- Store, retrieve, and update data.
- Provide integrity services to enforce database control.
- Provide a user accessible catalog of data description.
- Control concurrent processing.
- Support logical transaction.
- Recover from failure.
- Provide security facilities.
- Interface with communication control programs.
- Provide utility service.



**Flowchart (2.1) the internal structure of a DBS.**

In order to study a concurrent control and recovery, we need to suggest a model of the internal structure of database system as shown above.

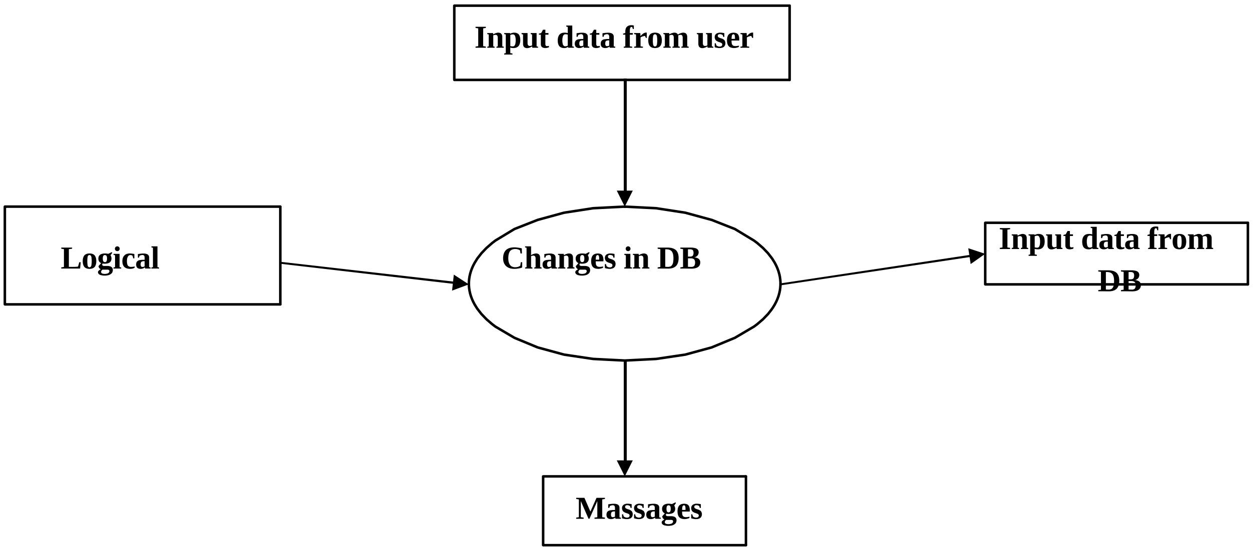
Our model, a database system DBS consist of modules [2]:

- Transaction manger(TM): it receives transaction operations and forwards them to the scheduler. The TM may also perform other functions, easily for a distributed database system.
- Scheduler: it is a program or a collection of programs that controls a current execution of transactions.
- Recovery manger: it is responsible of detecting failures and restore to the database to a state that that existed prior to the occurrence of the failure.
- Cache manger: it is operates directly on the database and manages the portion of the database that is kept in volatile storage.
- Transactions: is a program UNIT that access and possible updates various database items. The execution of a transaction preserves the consistence of the database.

In order to preserve the consistency, a transaction must be atomic.

Therefore the goal of concurrency control and recovery is to ensure that transactions execute atomically, which means that [2]:

- Each transaction access a shared data without interfering without interfering with other transactions.
- If a transaction terminates normally then all of it’s effect are made permanent , otherwise ,it has no effect at all , therefore; there are two type s of transaction:
  _aborted transaction: it’s effect on ram only.
  _committed: the changes save In DB (on hard disc).



Flowchart (2.2)

## 2.3 Properties of transaction [2]:

- Atomicity: the end result of running transaction must be the same as when each transaction is an atomic action. An action is said to be atomic if it is either done completely or not at all. Thus if transaction specific several step, the end result must be as if either all or none of these steps are carried out.
- Consistency: the transaction transforms the database from one consistence state to another. A consistence state is one that result from the execution of some given sequence of transactions
- Isolation: the actions of a transaction are not visible to any other transaction until and unless that transaction has committed.
- Durability: the actions of a transaction on database (upon committed) are permanent.

Collectively, these are called the ACID properties.

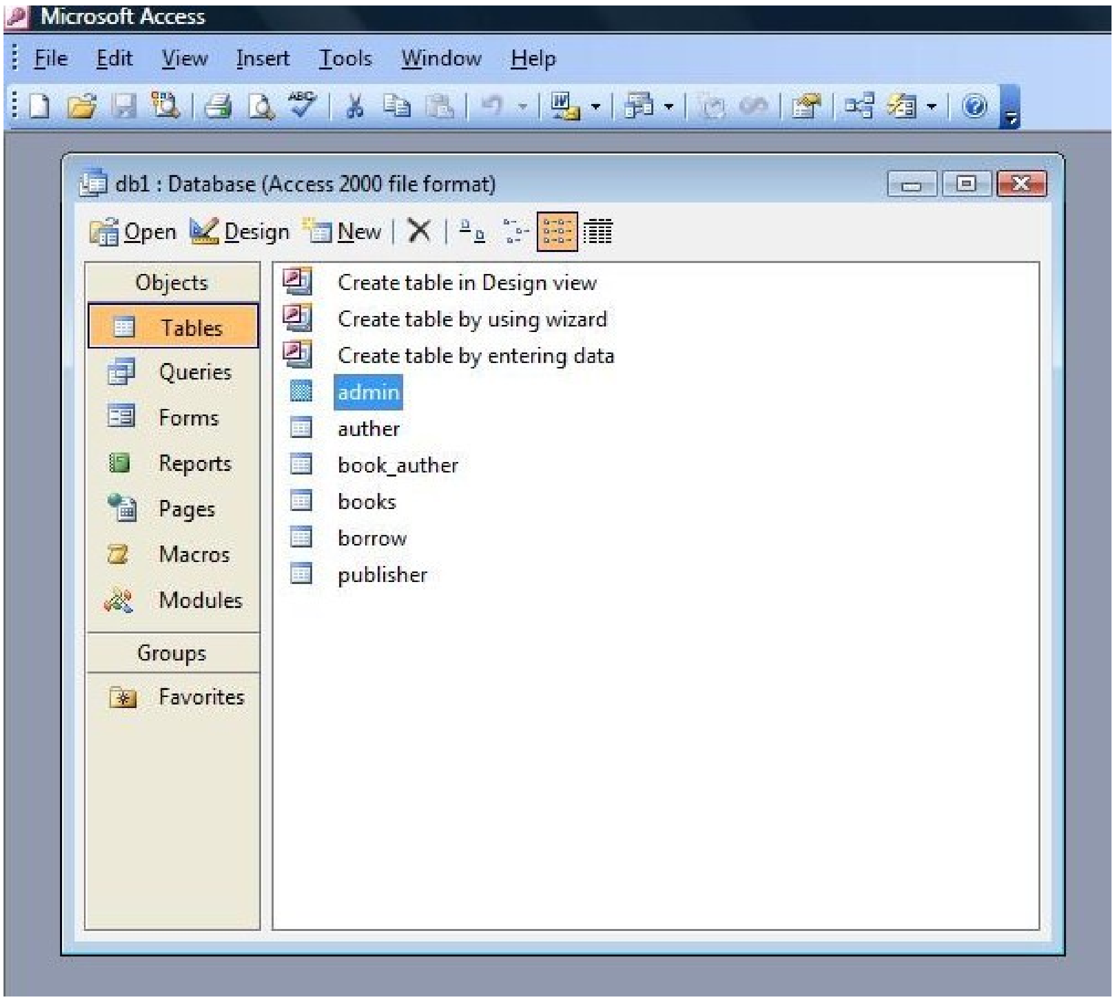
# Chapter 3
# Design and implementation

## 3.1 Design of the Database:

This project dealing with library management in this project database includes many tables, these are:

1. Four tables _to save books information, this table is [books, auther, publisher, and book_auther].
2. One table_ to save the users information and there authorize. this table is[admin].
3. One table_ to save the  information about borrower. this table is[borrow].

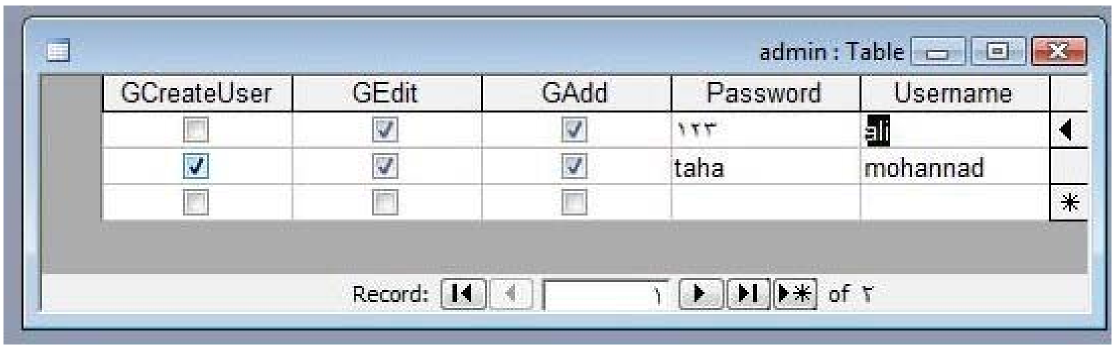
**Fig. (1.1)  showing this tabels**



Create table in Design view
Create table by using wizard
Create table by entering data
admin
auther
book_auther
books
borrow
publisher

**Fig. (1.1)** All Tables

Now admin table contains five fields as we show in **Fig. (1.2)**



admin : Table

| GCreateUser | GEdit | GAdd | Password | Username | |
|---|---|---|---|---|---|
| ☐ | ☑ | ☑ | ١٢٣ | ali | ◄ |
| ☑ | ☑ | ☑ | taha | mohannad | |
| ☐ | ☐ | ☐ | | | * |

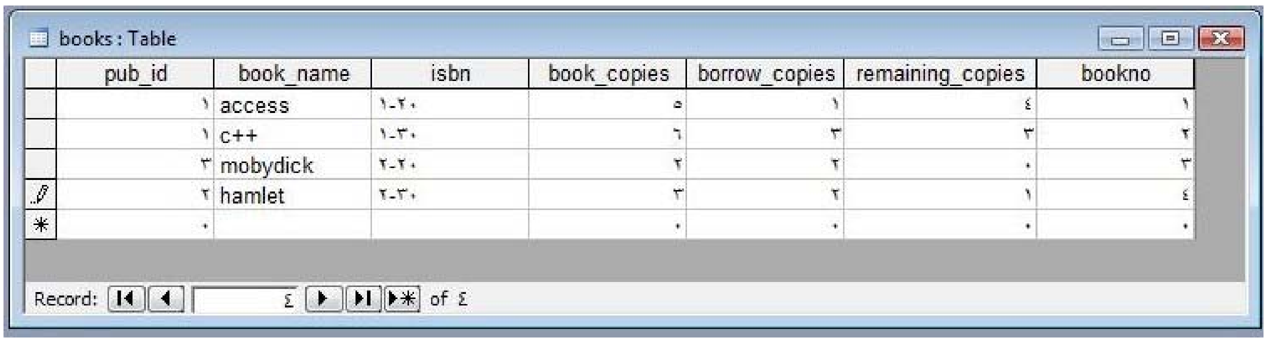
Record: ١ of ٢

**Fig. (1.2)**

Note that :

GCreateUser ,Gedit and Gadd to add and delete authorize to users

Also **Fig. (1.3) a,(1,3)b** showing the books and book_auther tables respectively whichs contains details about books

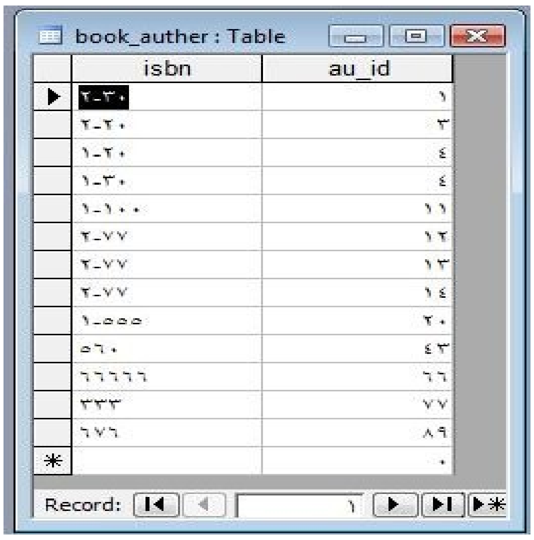


books : Table

| | pub_id | book_name | isbn | book_copies | borrow_copies | remaining_copies | bookno |
|---|---|---|---|---|---|---|---|
| | ١ | access | ١-٢٠ | ٥ | ١ | ٤ | ١ |
| | ١ | c++ | ١-٣٠ | ٦ | ٣ | ٣ | ٢ |
| | ٣ | mobydick | ٢-٢٠ | ٢ | ٢ | ٠ | ٣ |
| ✎ | ٢ | hamlet | ٢-٣٠ | ٣ | ٢ | ١ | ٤ |
| * | ٠ | | | ٠ | ٠ | ٠ | ٠ |

Record: ٤ of ٤

**Fig. (1.3) a**



book_auther : Table

| | isbn | au_id |
|---|---|---|
| ▶ | ٢-٣٠ | ١ |
| | ٢-٢٠ | ٣ |
| | ١-٢٠ | ٤ |
| | ١-٣٠ | ٤ |
| | ١-١٠٠ | ١١ |
| | ٢-٧٧ | ١٢ |
| | ٢-٧٧ | ١٣ |
| | ٢-٧٧ | ١٤ |
| | ١-٥٥٥ | ٢٠ |
| | ٥٦٠ | ٤٣ |
| | ٦٦٦٦٦ | ٦٦ |
| | ٣٣٣ | ٧٧ |
| | ٦٧٦ | ٨٩ |
| * | | ٠ |

Record: ١

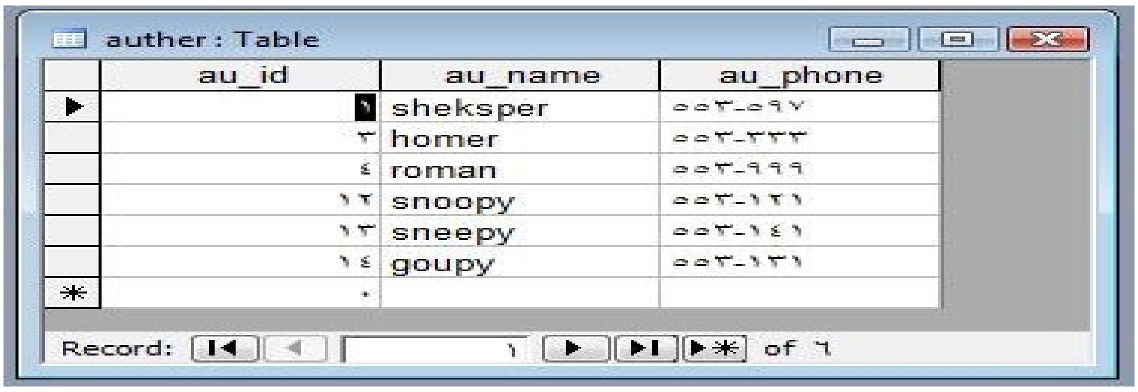
**Fig (1,3)b**

<span style="color:red">**Note that**</span> **:**

**This table show the authors' how authoring the book(s) according to ISBN**

The two tables publisher and author contains information about each like Id, name , and phone

Show **Fig (1,4)** and **Fig(1,5)** .

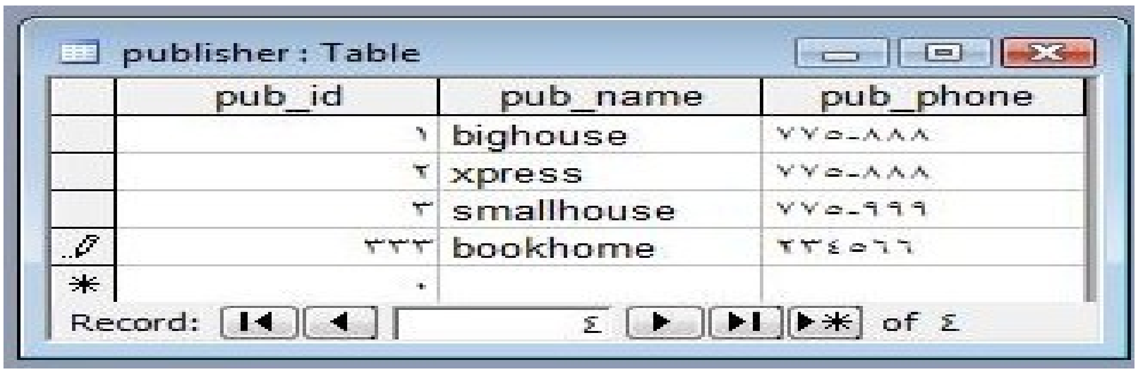


auther : Table

| | au_id | au_name | au_phone |
|---|---|---|---|
| ▶ | ١ | sheksper | ٥٥٣-٥٩٧ |
| | ٣ | homer | ٥٥٣-٣٣٣ |
| | ٤ | roman | ٥٥٣-٩٩٩ |
| | ١٢ | snoopy | ٥٥٣-١٢١ |
| | ١٣ | sneepy | ٥٥٣-١٤١ |
| | ١٤ | goupy | ٥٥٣-١٣١ |
| * | ٠ | | |

Record: ١ of ٦

**Fig (1,4)**



publisher : Table

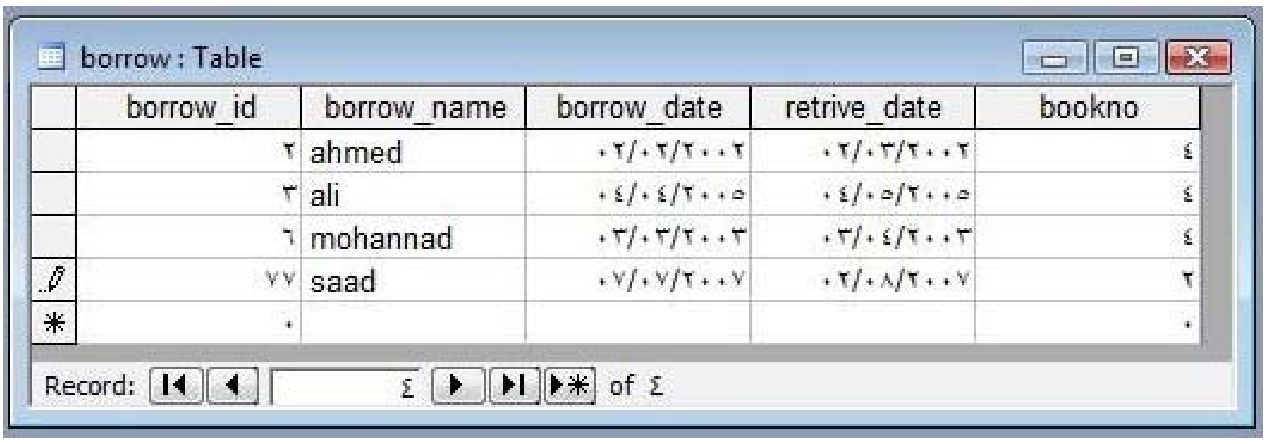
| | pub_id | pub_name | pub_phone |
|---|---|---|---|
| | ١ | bighouse | ٧٧٥-٨٨٨ |
| | ٢ | xpress | ٧٧٥-٨٨٨ |
| | ٣ | smallhouse | ٧٧٥-٩٩٩ |
| ✎ | ٣٣٣ | bookhome | ٢٣٤٥٦٦ |
| * | ٠ | | |

Record: |◀ ◀ ٤ ▶ ▶| ▶* of ٤

**Fig(1,5)**

Finally

we have borrow table which contains information about borrower id , name , (borrow and retrieve date) , and bookno .
show **Fig(1,6)**



borrow : Table

| | borrow_id | borrow_name | borrow_date | retrive_date | bookno |
|---|---|---|---|---|---|
| | ٢ | ahmed | ٠٢/٠٢/٢٠٠٢ | ٠٢/٠٣/٢٠٠٢ | ٤ |
| | ٣ | ali | ٠٤/٠٤/٢٠٠٥ | ٠٤/٠٥/٢٠٠٥ | ٤ |
| | ٦ | mohannad | ٠٣/٠٣/٢٠٠٣ | ٠٣/٠٤/٢٠٠٣ | ٤ |
| ✎ | ٧٧ | saad | ٠٧/٠٧/٢٠٠٧ | ٠٢/٠٨/٢٠٠٧ | ٢ |
| * | ٠ | | | | ٠ |

Record: |◀ ◀ ٤ ▶ ▶| ▶* of ٤

**Fig(1,6)**

## 3.2 How communicate with the Database (D.B.) through the Visual Basic:

There is different type of technique used to make connection between the visual basic and database (Access), like:

1. DAO
2. ADO
3. RDO
4. OO4O

Where the data base project used is (Microsoft Access) version2003

In this project the technique used is the ADO technique because it is the best. And the ADO term mean (Active X Data Object).

From this technique the user can access to the data base and control it like (add, delete, modify …).

So, if the user wants to communicate with data base using the ADO technique, the following steps must be done (after run the visual basic program):

- From the (project) list choose the (References).as the shown in the fig. below:

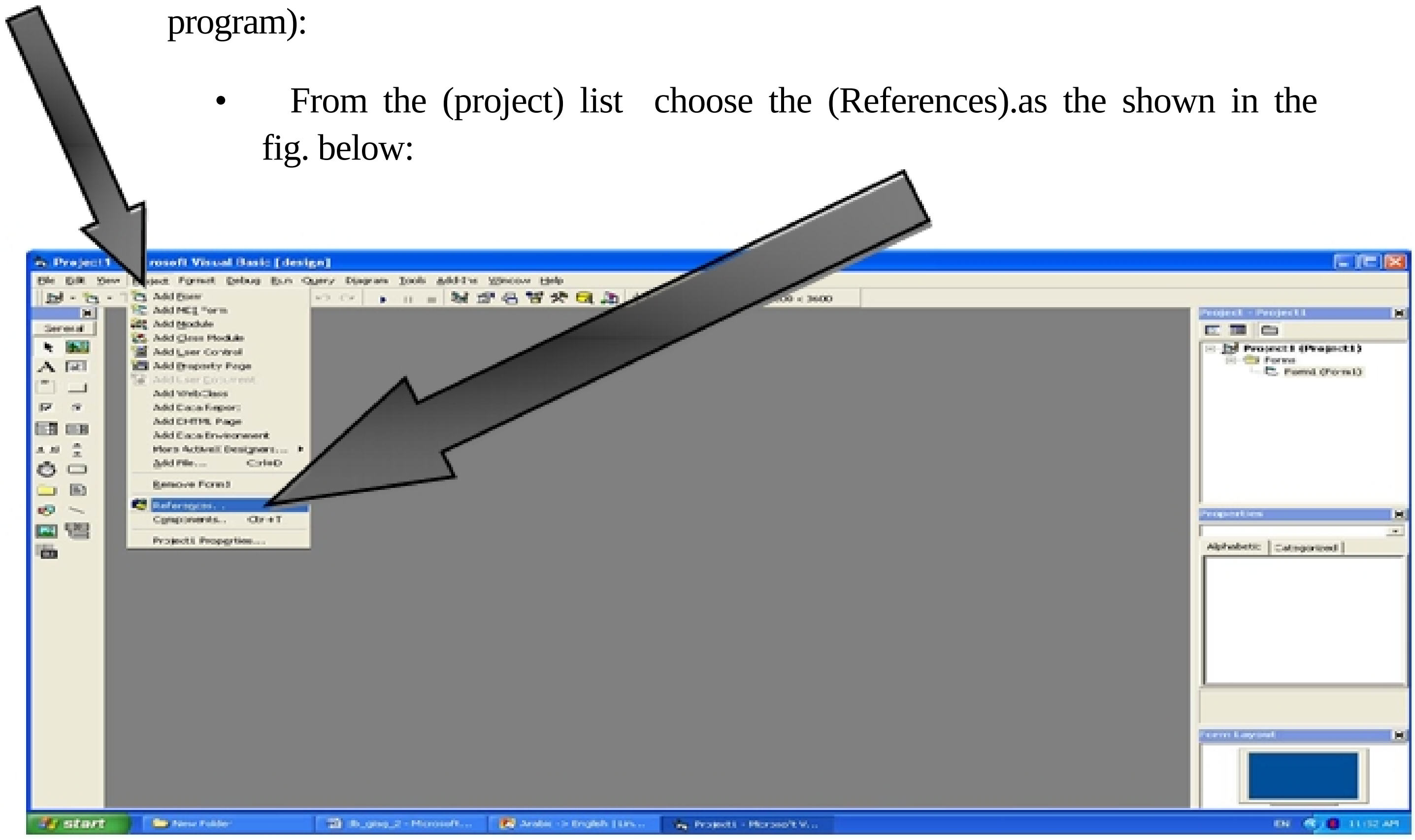
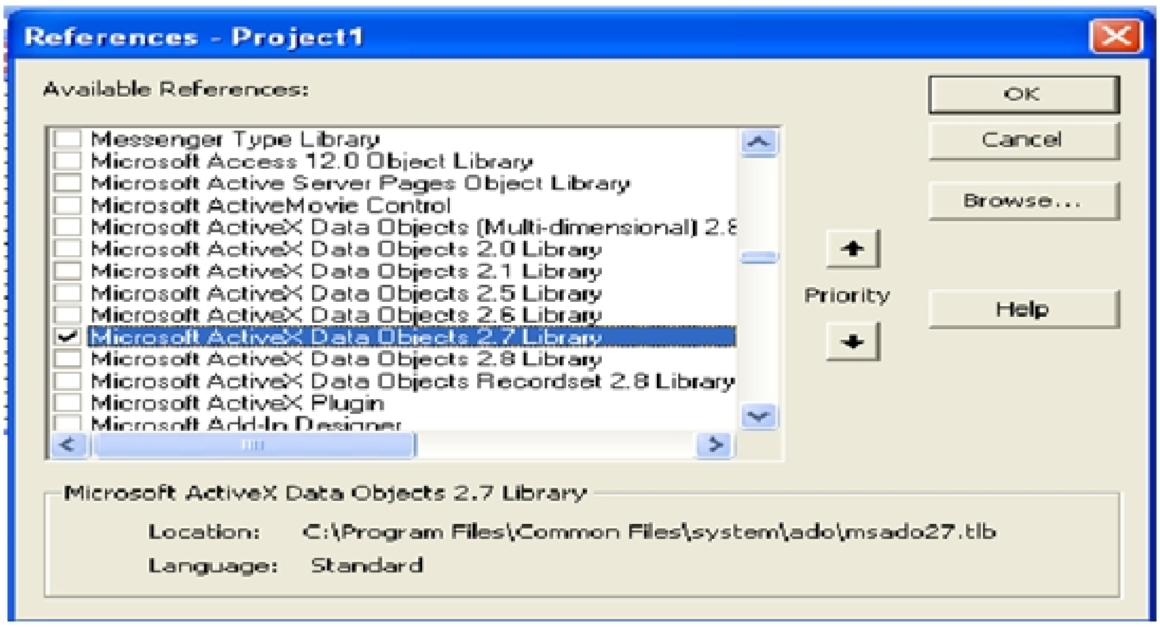


**Fig. (3.6)**

- The following fig. will appear, and then the user must choose the (Microsoft Active X Data Objects 2.7), the number not important but the (2.7) prefer.



Available References:

□ Messenger Type Library
□ Microsoft Access 12.0 Object Library
□ Microsoft Active Server Pages Object Library
□ Microsoft ActiveMovie Control
□ Microsoft ActiveX Data Objects (Multi-dimensional) 2.8
□ Microsoft ActiveX Data Objects 2.0 Library
□ Microsoft ActiveX Data Objects 2.1 Library
□ Microsoft ActiveX Data Objects 2.5 Library
□ Microsoft ActiveX Data Objects 2.6 Library
☑ Microsoft ActiveX Data Objects 2.7 Library
□ Microsoft ActiveX Data Objects 2.8 Library
□ Microsoft ActiveX Data Objects Recordset 2.8 Library
□ Microsoft ActiveX Plugin
□ Microsoft Add-In Designer

Microsoft ActiveX Data Objects 2.7 Library

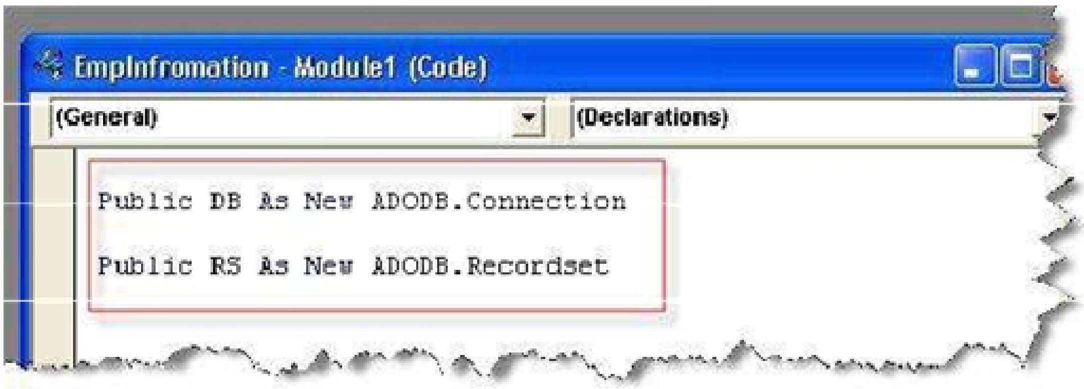
Location: C:\Program Files\Common Files\system\ado\msado27.tlb

Language: Standard

**Fig. (3.7)**

Then, when the user writes the code of the frame that wants to communicate with data base. The user must define a public variable. The benefit of this variable is to make connection with specific data base table. And give to this variable any name, in this project this variable’s name is (DB).

Then the user must define another variable, the benefit of this variable is (open table, add record, delete, modify…). And give to this variable any name, in this project this variable’s name is (RS).



```
Public DB As New ADODB.Connection

Public RS As New ADODB.Recordset
```

**Fig. (3.8)**

Then , in the frame that want to communicate with data base. The user must write the following code in the load of this frame:

```
If DB.State = adStateOpen Then DB.Close
DB.Provider = "Microsoft.JET.OLEDB.4.0";
DB.Open App.Path & "\Data.mdb"
```

```
If DB.State = adStateOpen Then DB.Close
```

If the base DB then close it, this to avoid any problem.

```
DB.Provider = "Microsoft.JET.OLEDB.4.0";
```

Here define the type of the communicate with the base DB.Provider
If the user want to communicate with data base the following code must be used:
Microsoft.JET.OLEDB.4.0
And the provider that can make communicate with data base through it, is the following:

```
DB.Open App.Path & "\Data.mdb"
```

Here detect the path where the tables exist, the user can change this path to detect the path where the tables exist.

Also, in this project another method used to make communication with data base. And this method accomplished through the (Adodc) tool. This tool can be added by doing the following steps:

1.

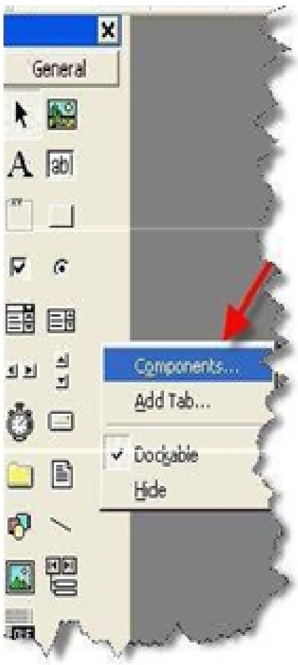
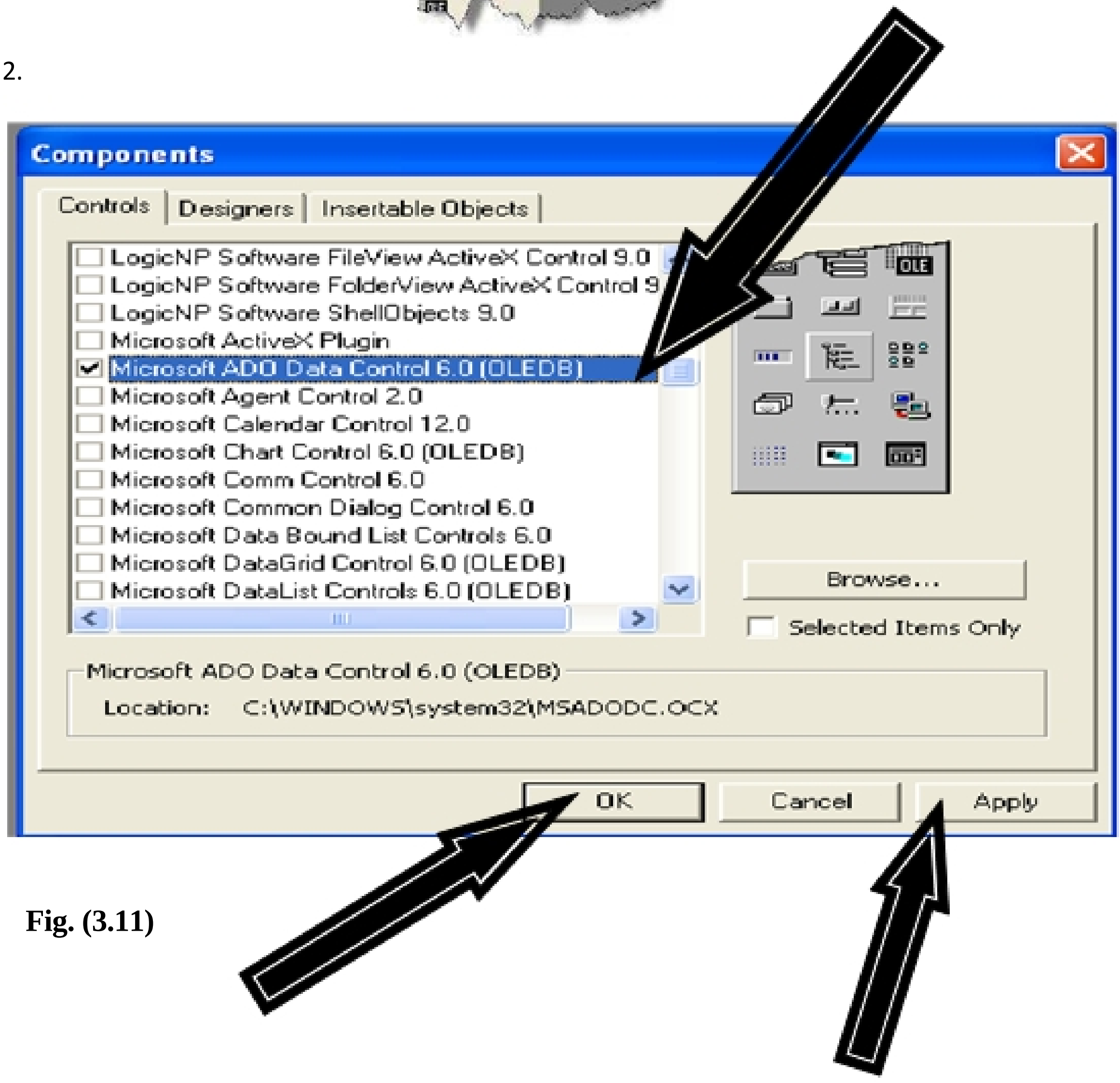


Fig. (3.9)

2.



□ LogicNP Software FileView ActiveX Control 9.0
□ LogicNP Software FolderView ActiveX Control 9
□ LogicNP Software ShellObjects 9.0
□ Microsoft ActiveX Plugin
☑ Microsoft ADO Data Control 6.0 (OLEDB)
□ Microsoft Agent Control 2.0
□ Microsoft Calendar Control 12.0
□ Microsoft Chart Control 6.0 (OLEDB)
□ Microsoft Comm Control 6.0
□ Microsoft Common Dialog Control 6.0
□ Microsoft Data Bound List Controls 6.0
□ Microsoft DataGrid Control 6.0 (OLEDB)
□ Microsoft DataList Controls 6.0 (OLEDB)

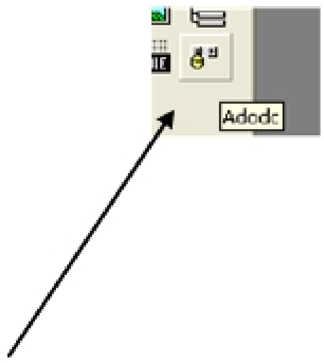
□ Selected Items Only

Microsoft ADO Data Control 6.0 (OLEDB)

Location: C:\WINDOWS\system32\MSADODC.OCX

Fig. (3.11)

3.



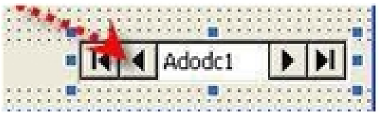


**Fig. (3.12)**

4. the user must type the following code in [load] of the frame which want to communicate with data base:

```
Adodc1.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
App.Path +"\Data.mdb;Persist Security Info=False"
```

Adodc1.ConnectionString

Means the source of the data that will bring from.

Note that the ADODC tools will the information that brings to it, to use this information in another situation.

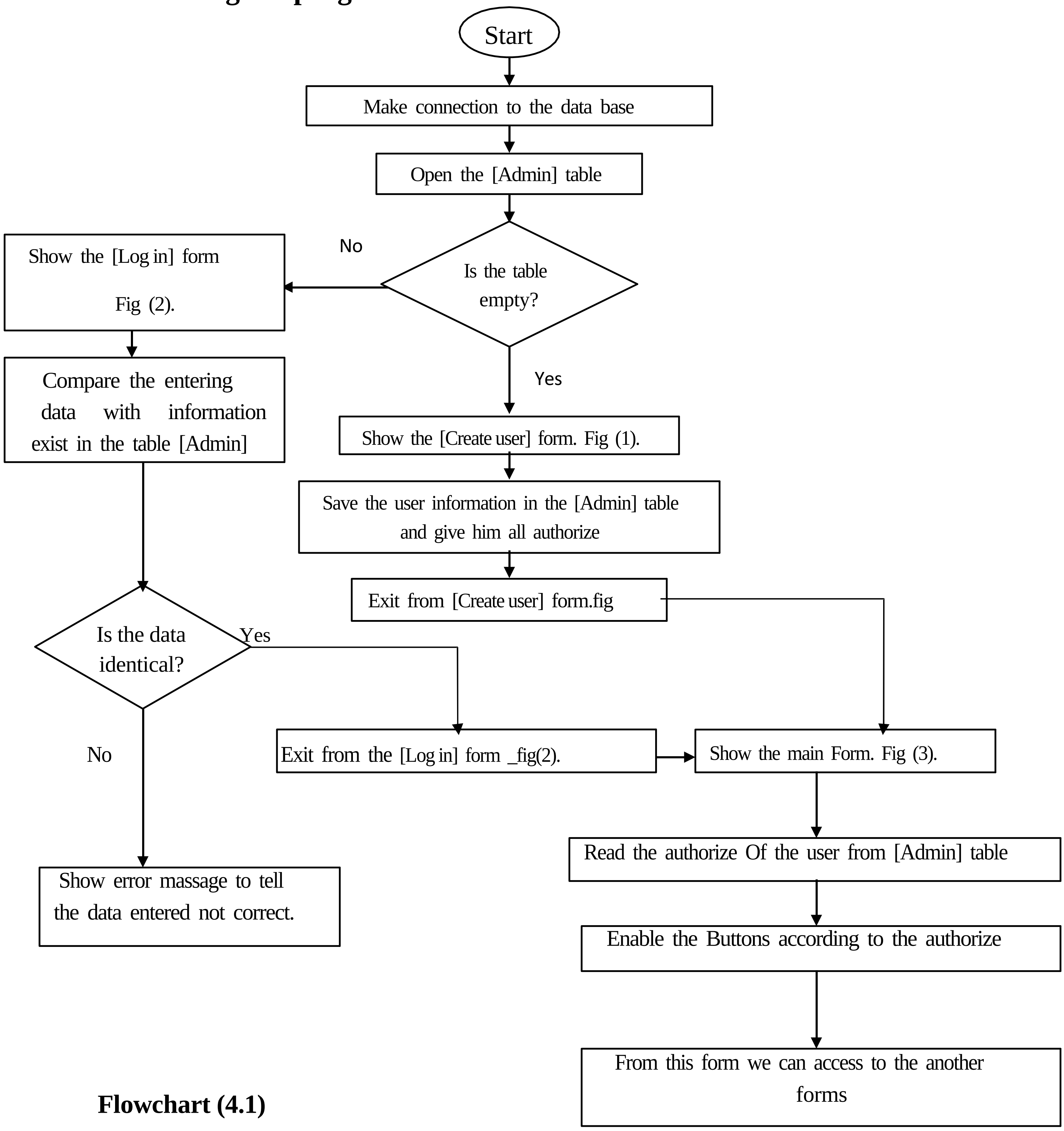
# Chapter 4

# System Testing

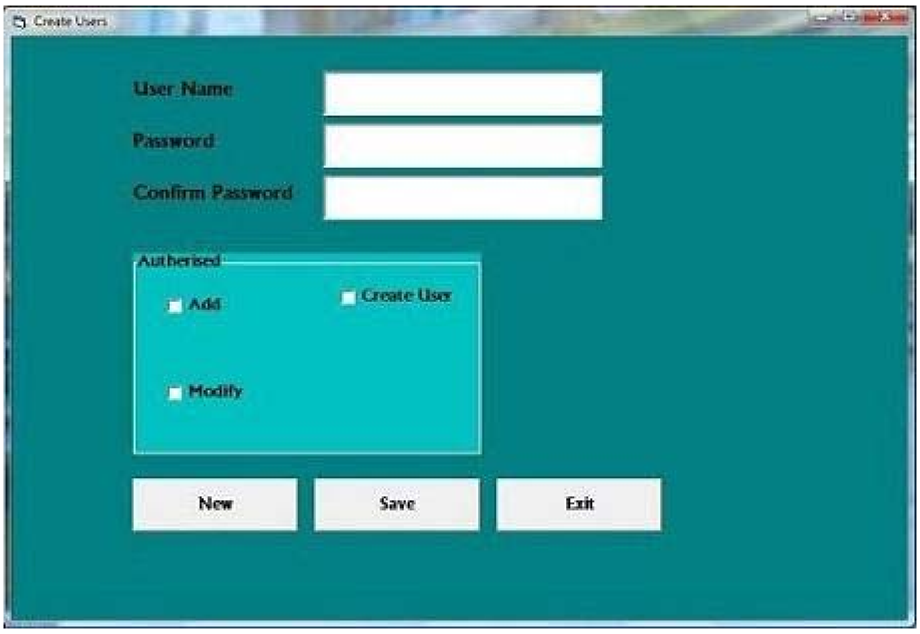
The program design is one of the important things in the technique and information world, therefore; it is impossible to do without programming. Language to design a special programs or public programs. So, the (Microsoft Visual Basic 6.0) is used in this project. And must detect the type of the database that must be used to save the desire information needed. And the type of the data base depends on the size of the program. if there is a normal company requires using database, then the data base used here is (Microsoft Access).

.

## 4.1 Testing the program:


Start
Make connection to the data base
Open the [Admin] table
Is the table empty?
No
Yes
Show the [Log in] form Fig (2).
Compare the entering data with information exist in the table [Admin]
Show the [Create user] form. Fig (1).
Save the user information in the [Admin] table and give him all authorize
Exit from [Create user] form.fig
Is the data identical?
Yes
No
Exit from the [Log in] form _fig(2).
Show the main Form. Fig (3).
Read the authorize Of the user from [Admin] table
Show error massage to tell the data entered not correct.
Enable the Buttons according to the authorize
From this form we can access to the another forms

Flowchart (4.1)



User Name

Password

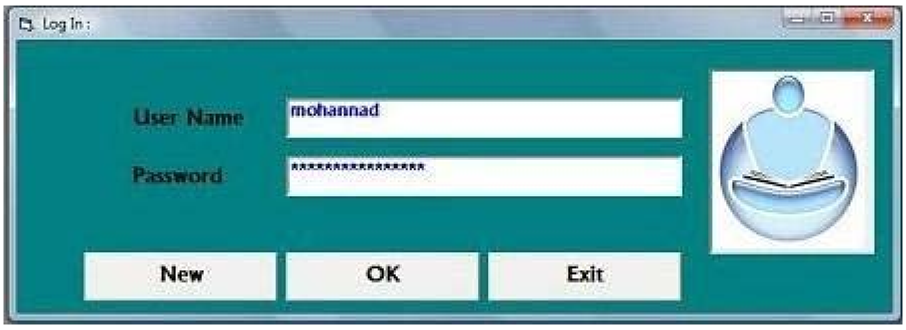
Confirm Password

□ Add

□ Create User

□ Modify

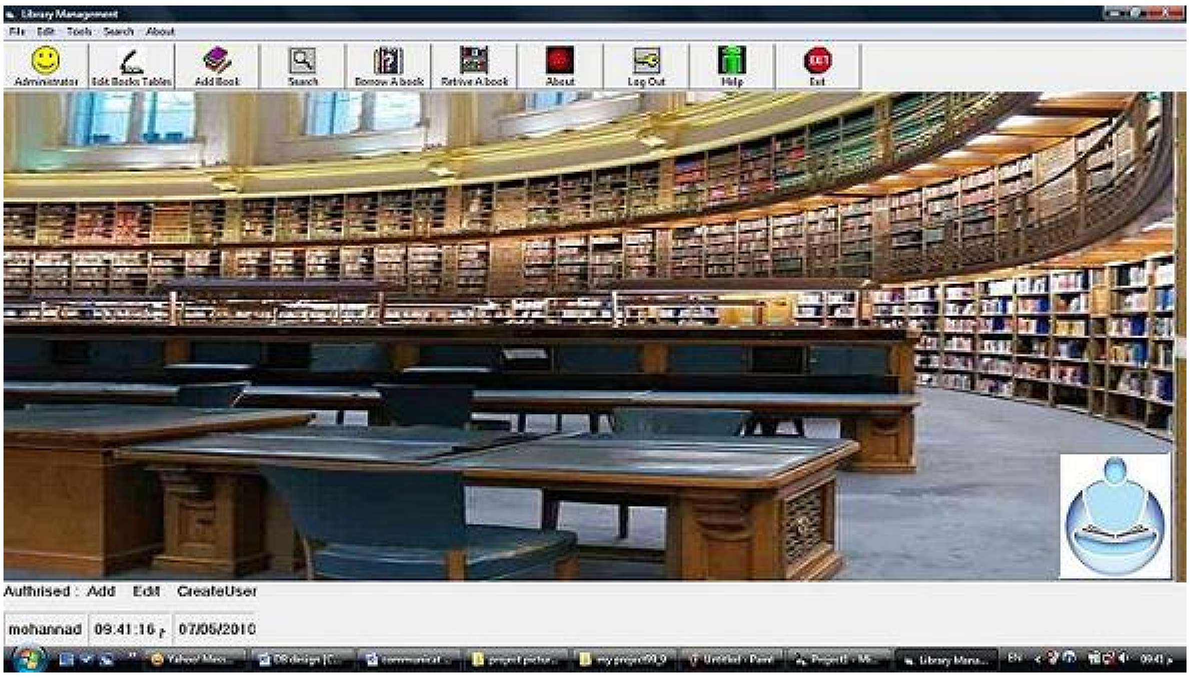
Fig (1).



User Name mohannad

Password ****************

fig(2).



Authrised : Add Edit CreateUser

mohannad 09:41:16 م 07/05/2010

Fig(3) [ Main form}

When the program run, the first form will work is the [create user] form_ Fig (1). this form make access to the data base to read the content of the [User]table .the read operation will not read the content of this table, but the reading operation represent by return the number of the records exist in this table .

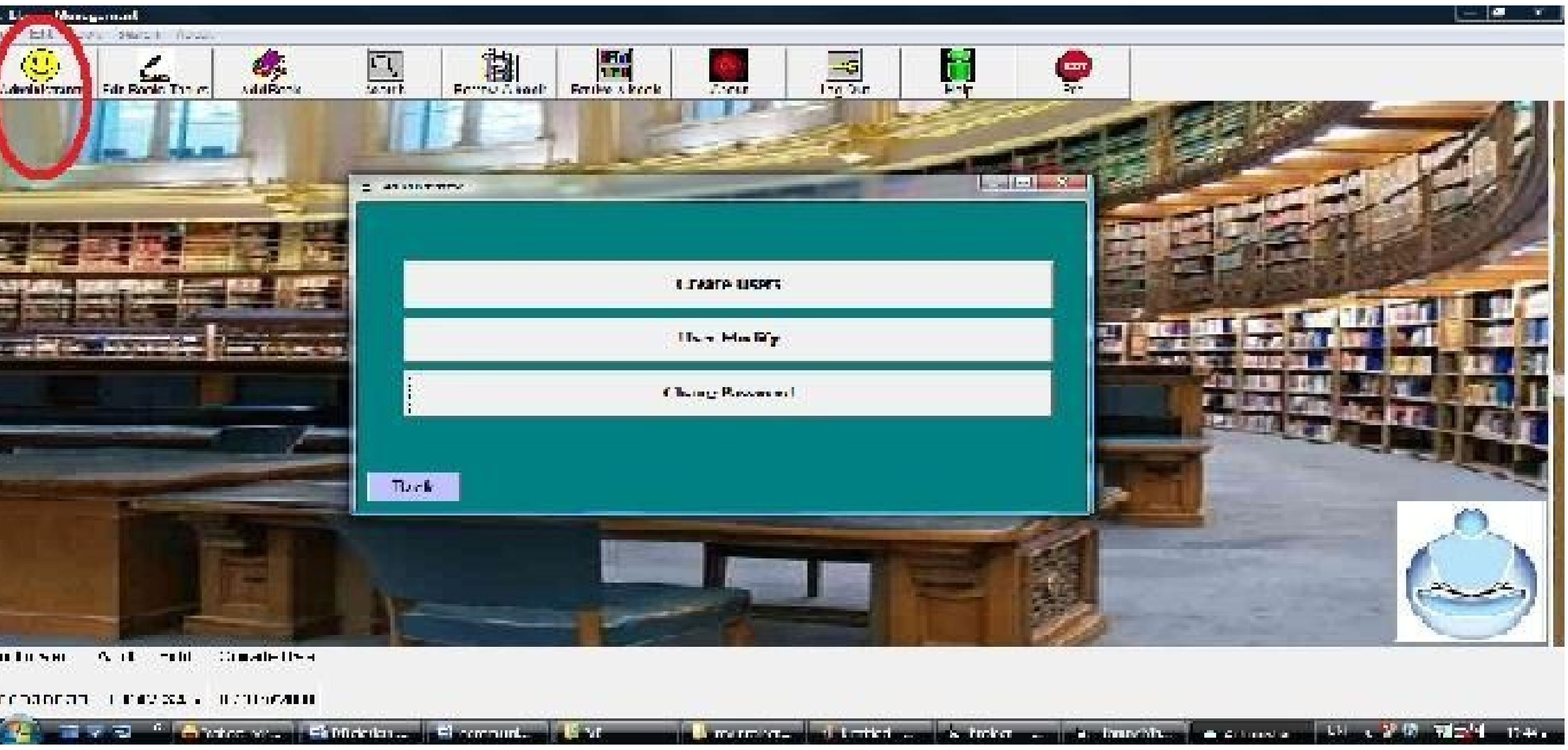
If this number equal to zero: that mean the current user is the first user run this program. So, this user will access to this program as a manger of the system. So when the current user enters his information, this information will save will save in the [user] table and give this user all authorize {the authorize here is: creating another user, modifying student information, add student and printing}. After this , the main Form_ Fig(3).will appear ,and since the current user have all authorize, so all buttons in the main Form_ Fig(3). Will be active.

If the number of records not equal to zero: that mean the current user is not the first user run this program. So, the [create user] form_ Fig (1).will make [Login] form _fig(2).running instead of it.

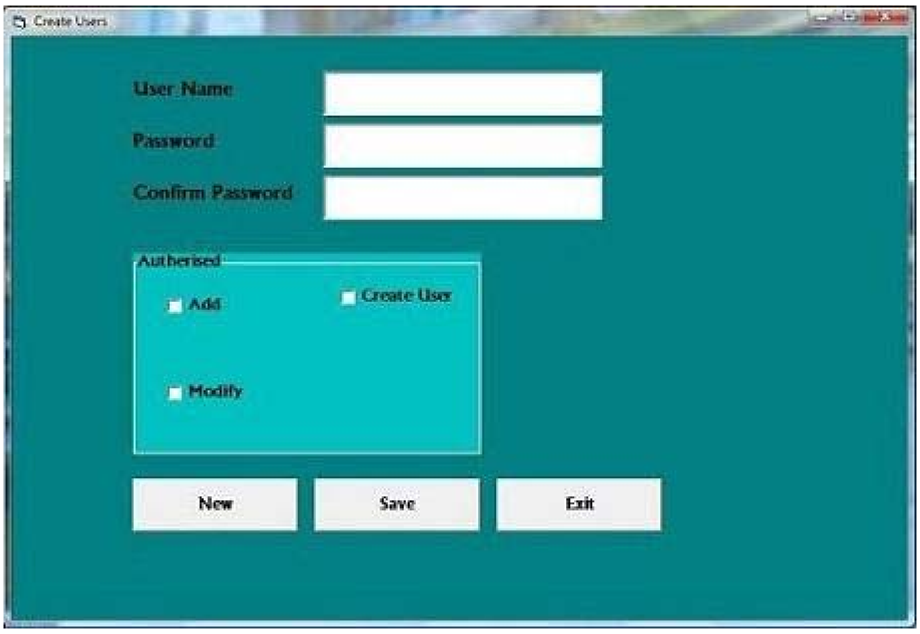
When [Log in] form _fig(2).run it will ask the user to enter user name and password. so, when this data enter from the user, access operation to the data base will done, to read the content of the [user]table(the read operation here is to retrive data, not to return the record number exist in this table). Then, compare operation will done between the entering data and the data existing in the [User]table, if matching exist,then read operation on the user autherize will done,and then main Form_ Fig(3) will run .according to the user authorize, the buttons in the main Form_ Fig(3) will be active.

## 4.2 Administrator:

When click the Administrator( fig 4 ) button we can create user(s) , user modify and change password As we show in the following figures 5 , 6 ,7 respectively :



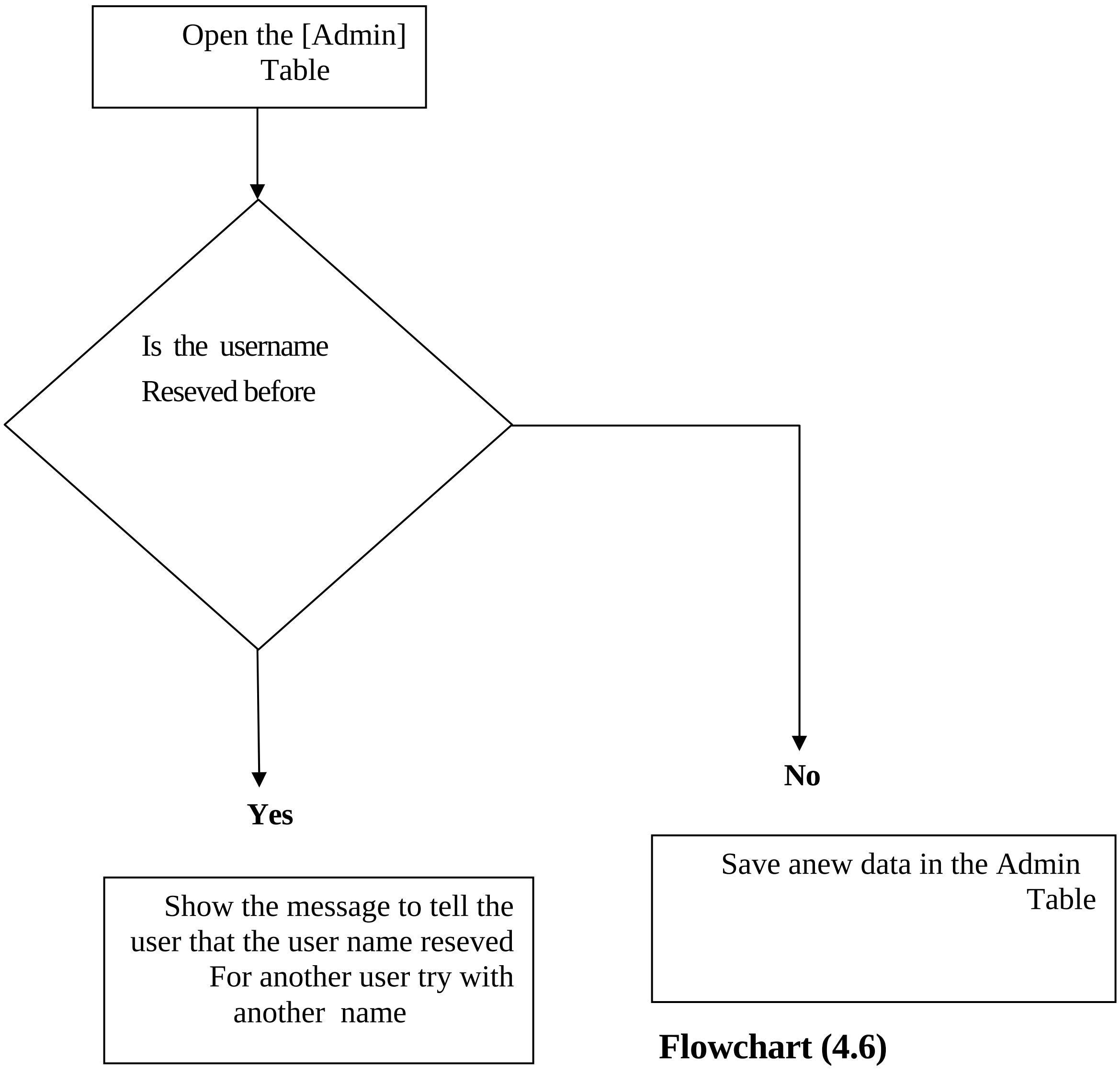
Fig(4)



Fig(5)

According to the following flow chart :

## 4.3 Creating User:

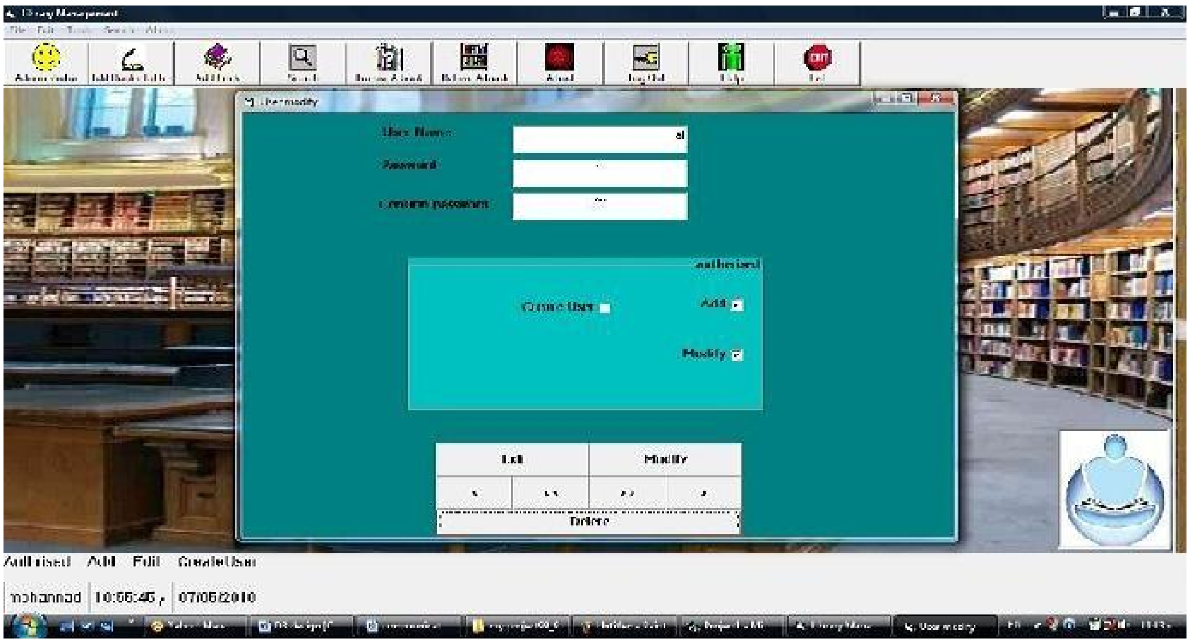

Open the [Admin] Table
Is the username Reseved before
Yes
No
Show the message to tell the user that the user name reseved For another user try with another name
Save anew data in the Admin Table

**Flowchart (4.6)**

Must be notecible that if the user have the creating student autherize, then this user must have all other autherize.

## 4.4 User modify :

We can also modify the authorized(Add or Delete) , name , and password for any user in The [Admin] table : show figure ( Fig 5 )



User Name: al

Password

Confirm password

authorised

Create User □

Add ☑

Modify ☑

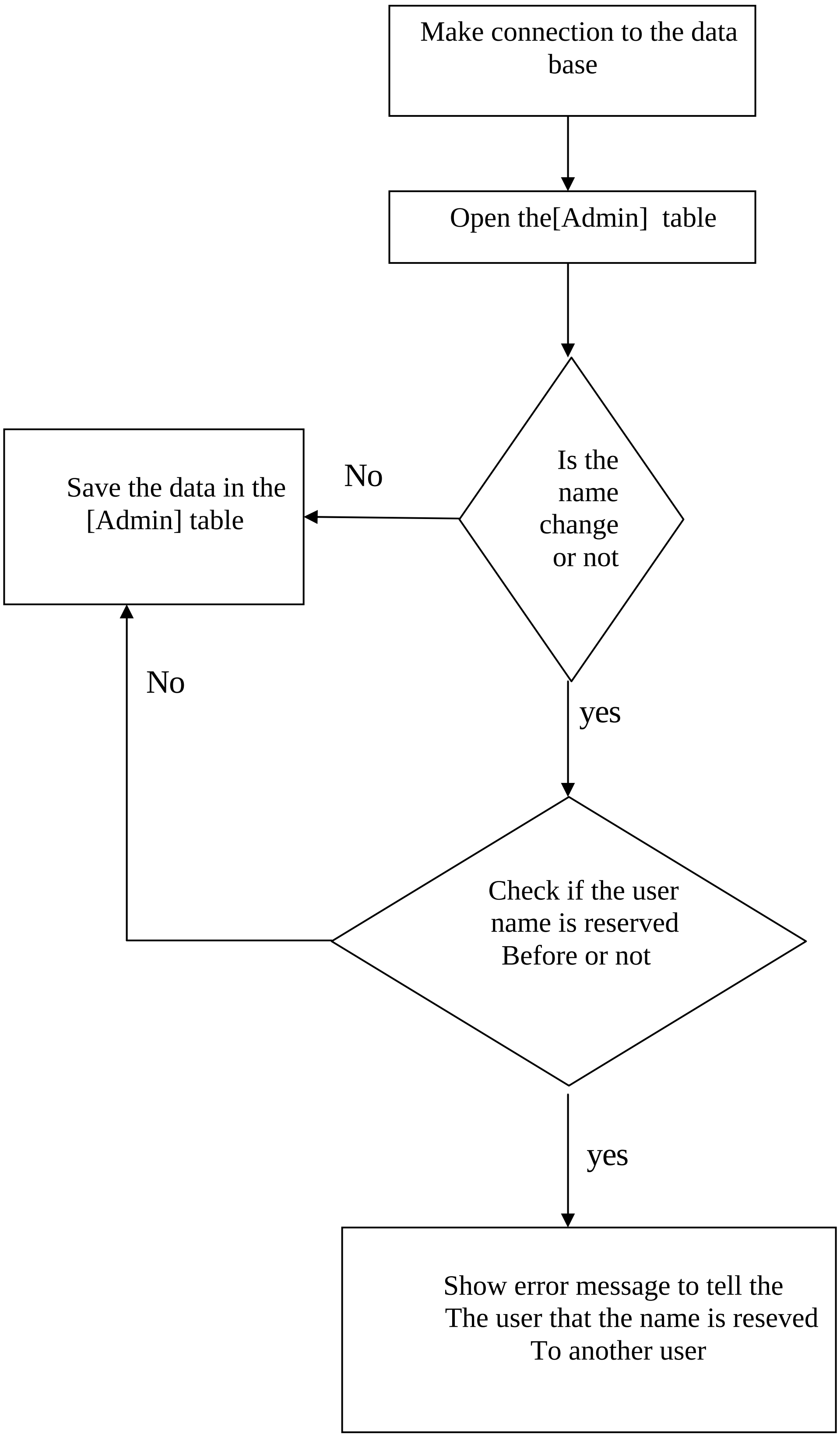
Authorised Add Edit CreateUser

mohannad | 10:56:46 م | 07/05/2010

Fig(5)

**Show Flowchart (4.7)**

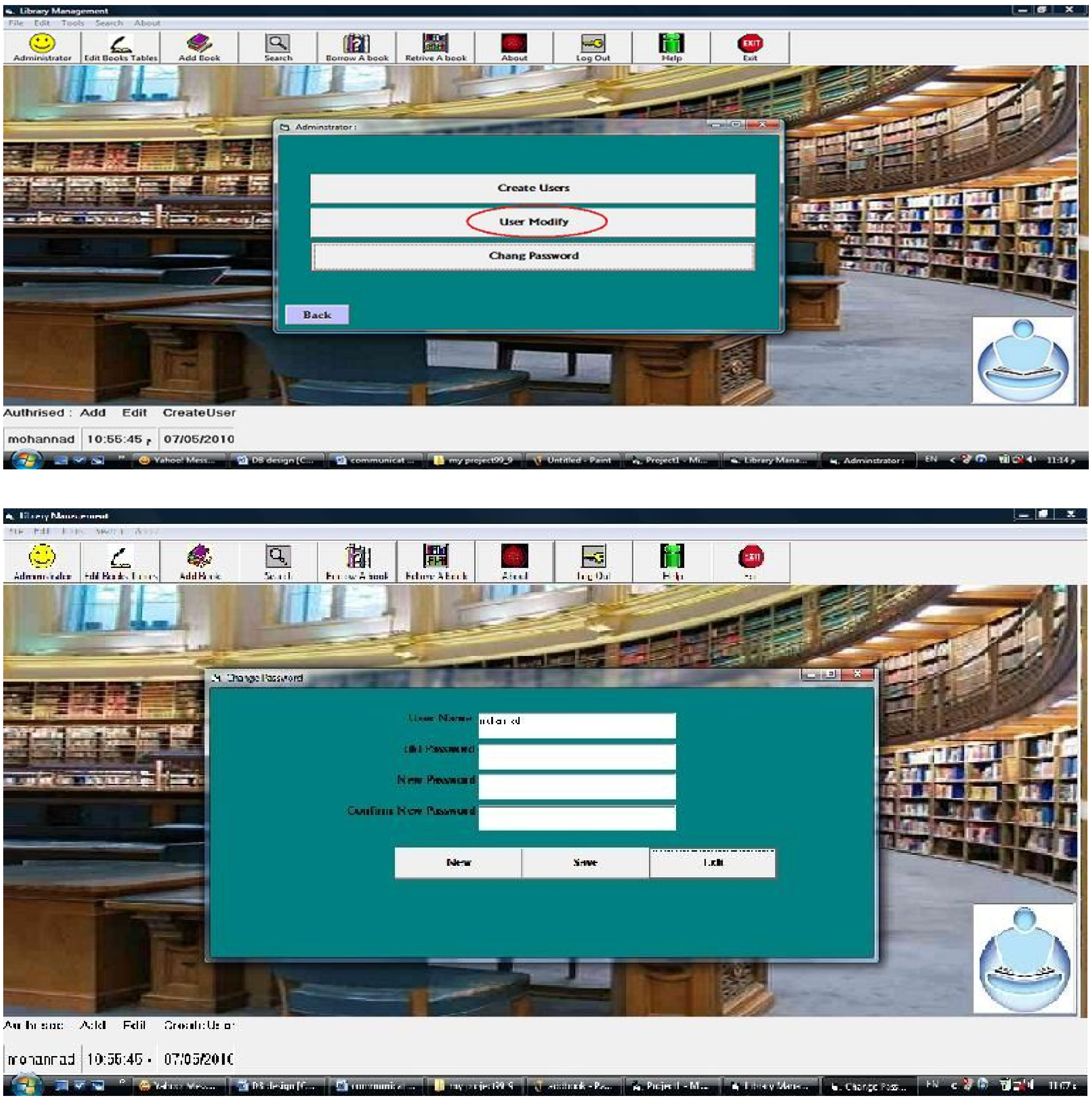
## 4.4 User Modification:



Make connection to the data base
Open the[Admin]  table
Is the name change or not
No
Save the data in the [Admin] table
yes
No
Check if the user name is reserved Before or not
yes
Show error message to tell the The user that the name is reseved To another user

## Flowchart (4.7)

Another option we can change the password for the user see Fig(6)



Authrised : Add Edit CreateUser

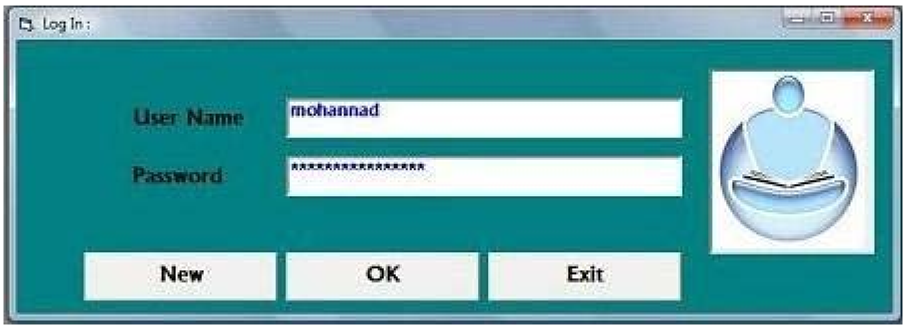
mohannad 10:55:45 07/05/2010

Fig(6)

Notice that the new password is entered in the [Admin] table as update for That user .

## 4.5 log in:

To log in a program( if the admin table not empty) the log in form will appear Fig (7) .

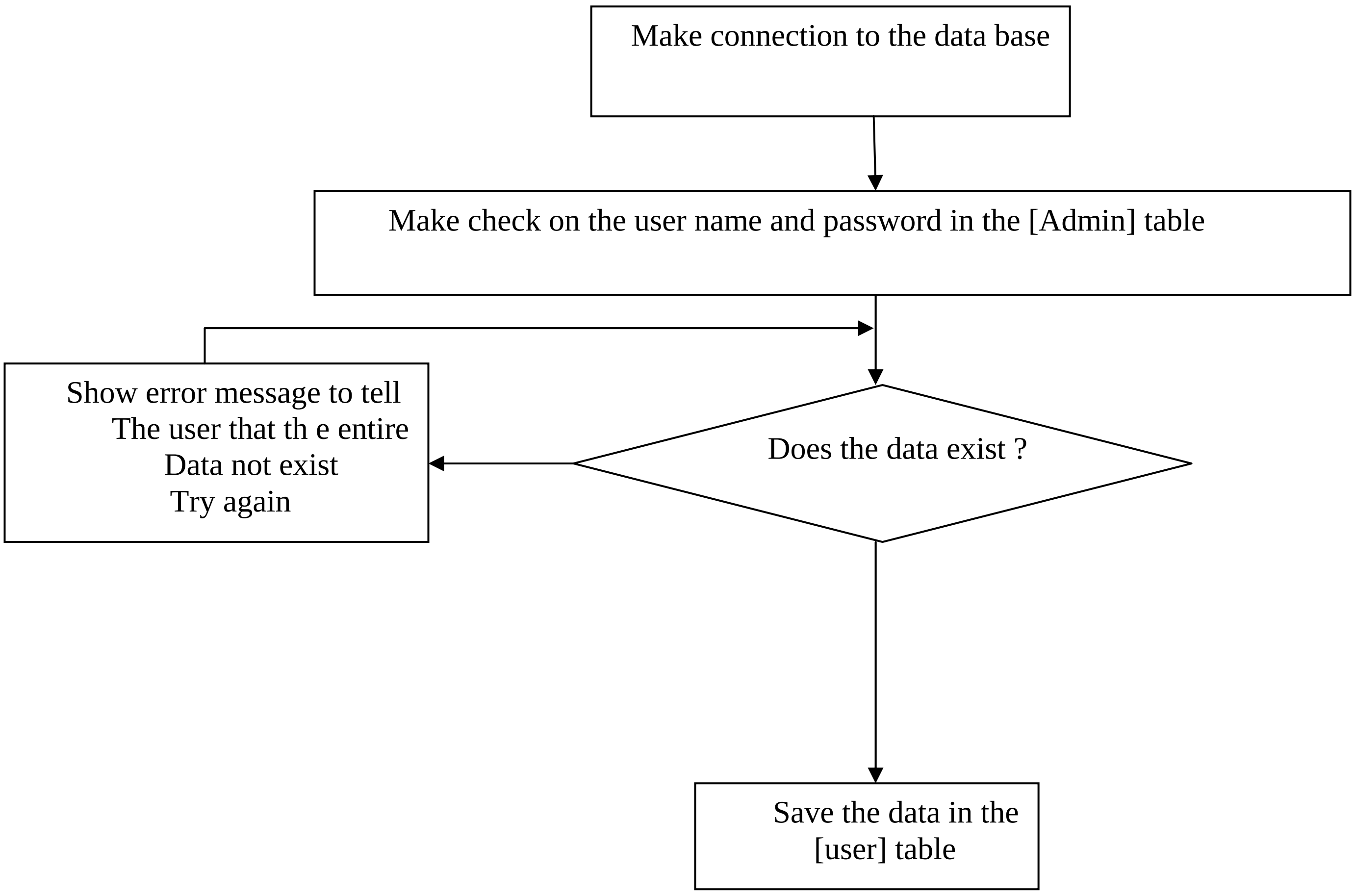


User Name mohannad

Password ****************

Fig (7)

It work according to (4.8) flow chart


Make connection to the data base
Make check on the user name and password in the [Admin] table
Show error message to tell
The user that th e entire
Data not exist
Try again
Does the data exist ?
Save the data in the
[user] table

(4.8) flow chart

## 4.6 Adding(books) :

Again return to the main form if you clicked "Add book " the addbook form will appear see Fig (8),Fig(9)

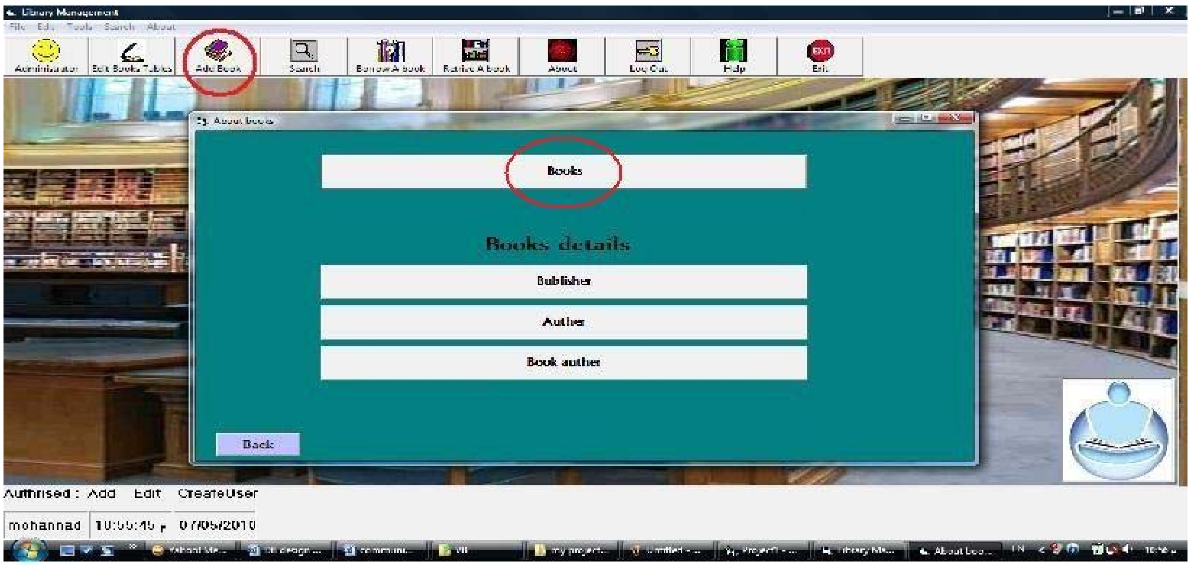


Fig (8)

We can add the book after entered the information about book, some information must entered ,and another its optional

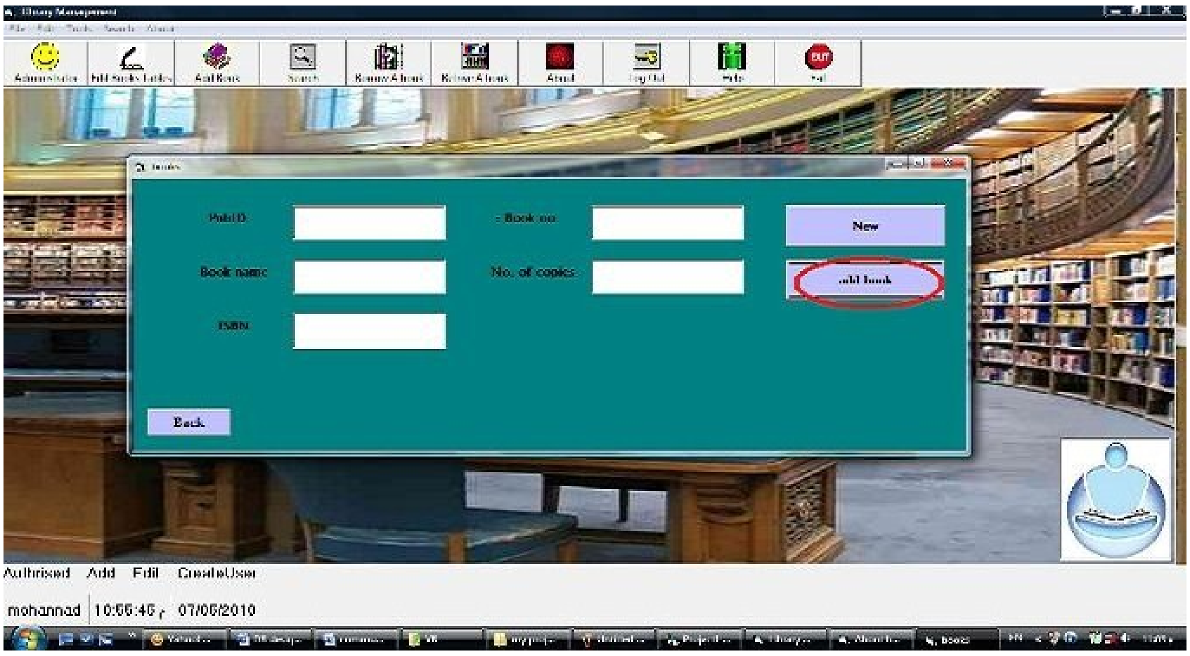
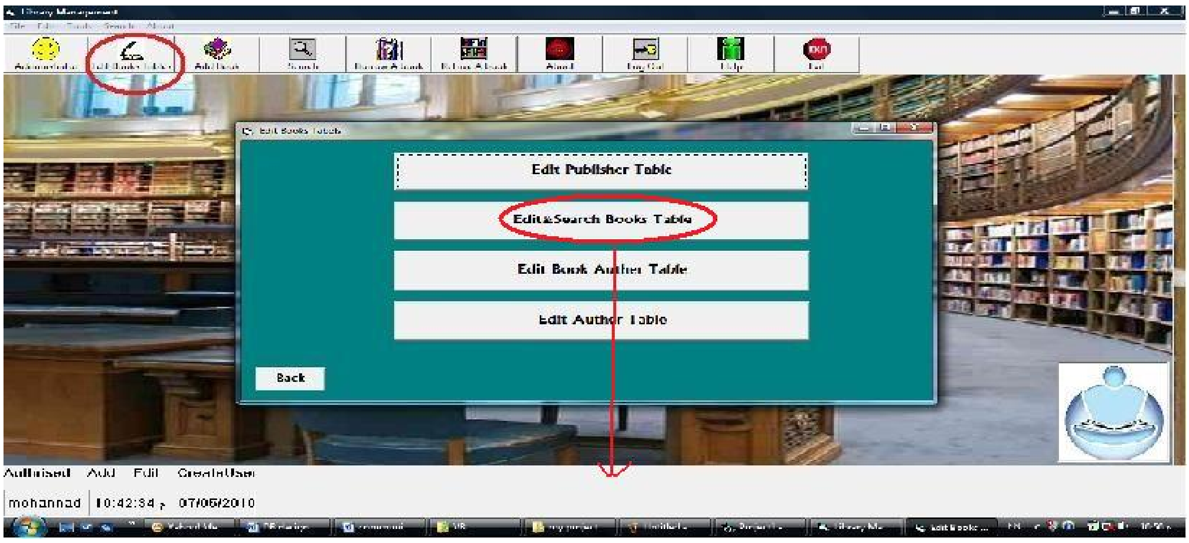


Fig (9)

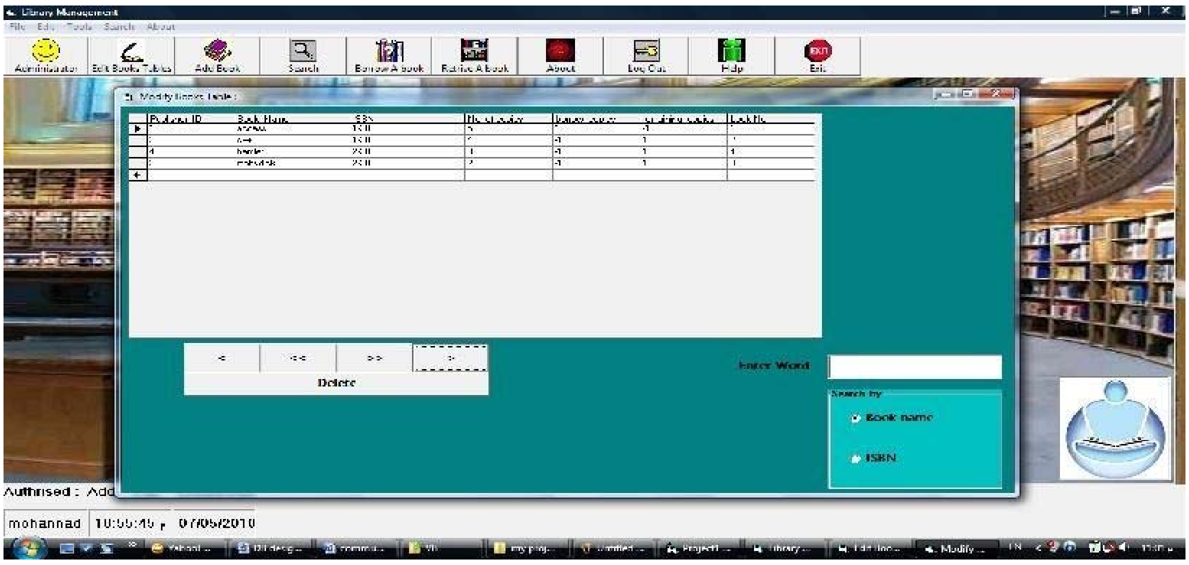
## 4.7 Editing(update,insert,delete) :

In a same manner we can entered information about author and publisher

Also we can insert, update, search and delete from the books tables see Fig(10),Fig(11)



Fig(10)



| Publisher ID | Book Name | ISBN | [illegible] | [illegible] | [illegible] | [illegible] |
|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | 5 | [illegible] | -1 | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | -1 | 1 | [illegible] |
| 4 | hamlet | [illegible] | 1 | -1 | 1 | 1 |
| [illegible] | [illegible] | [illegible] | [illegible] | -1 | 1 | 1 |

Fig(11)

## **4.8** Searching:

- We can also search for all books authoring by a specific author , by type a  author name in the text search
- We can search for all books published by a specific publisher organization by type publisher name in the text search
- We can search for any book by type a name of book in the text search

Show  Fig(12)

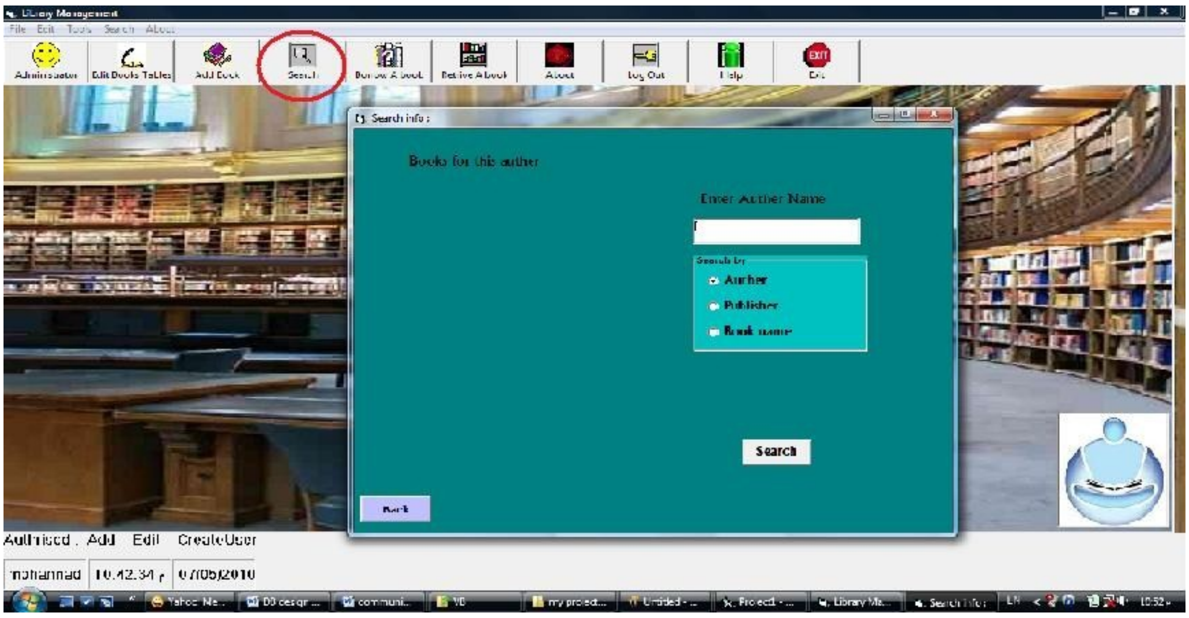
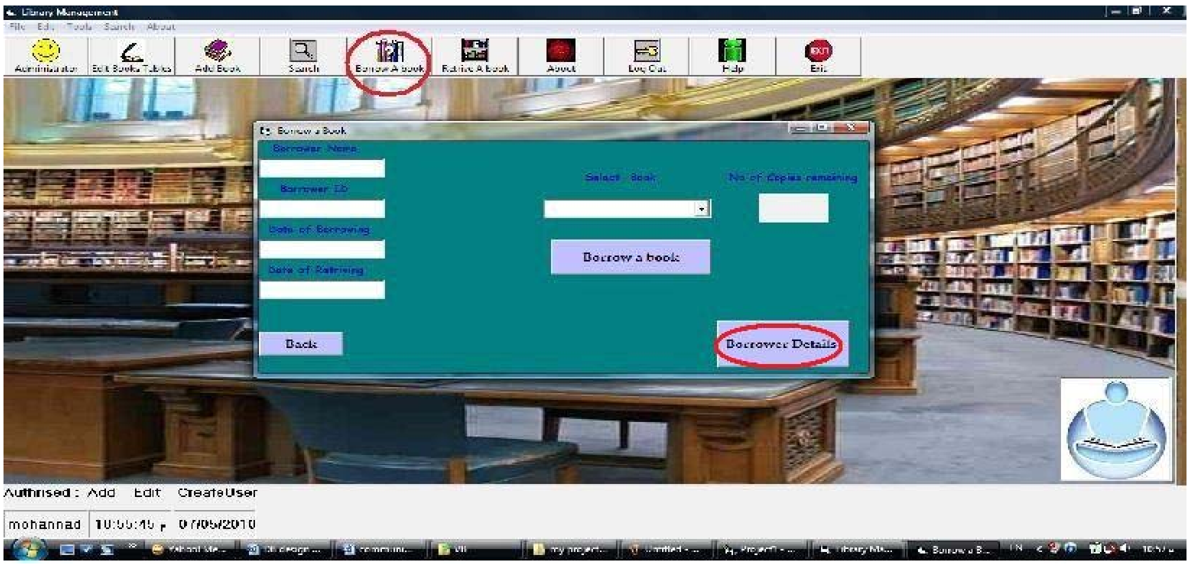


Fig (12)

## **4.9** Borrowing :

when someone  want  to borrow a book from a library we must enter a some  information about him.
Like his name , book name,  date of  borrowing  and date of retrieving the book . the number of book copies will decries
This information will saved in the [borrow] table.
We can see borrower information by click on borrower details command
 see figure (13)



Fig(13)

## **4.10** Retrieving:

When someone want to retrieve a book to the library we must delete him from a borrow table .
The number of book copies will increase .
See figure (14).

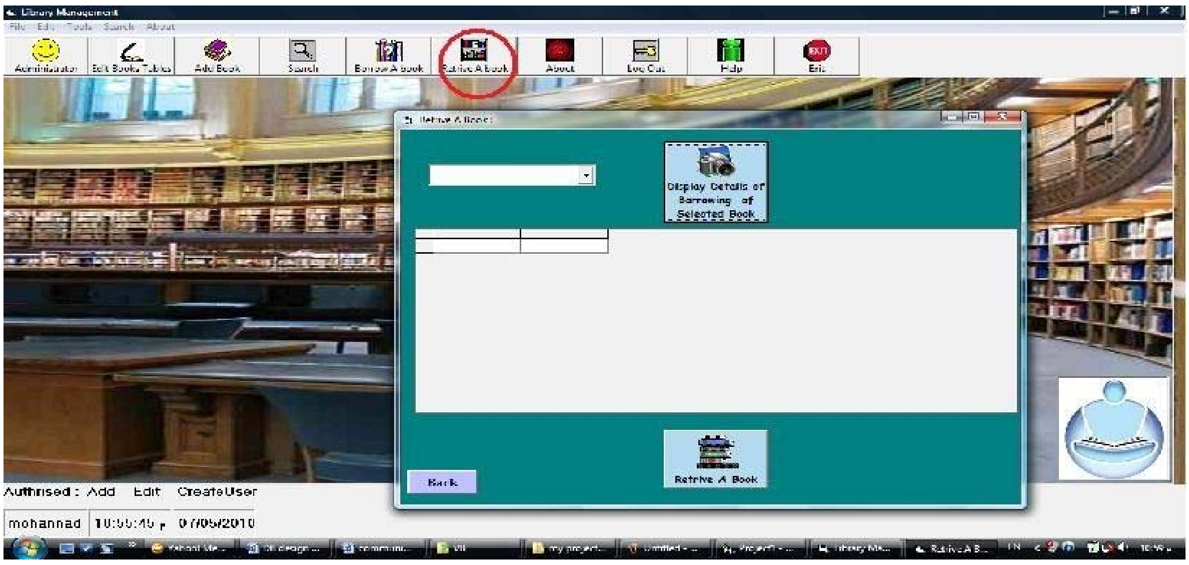
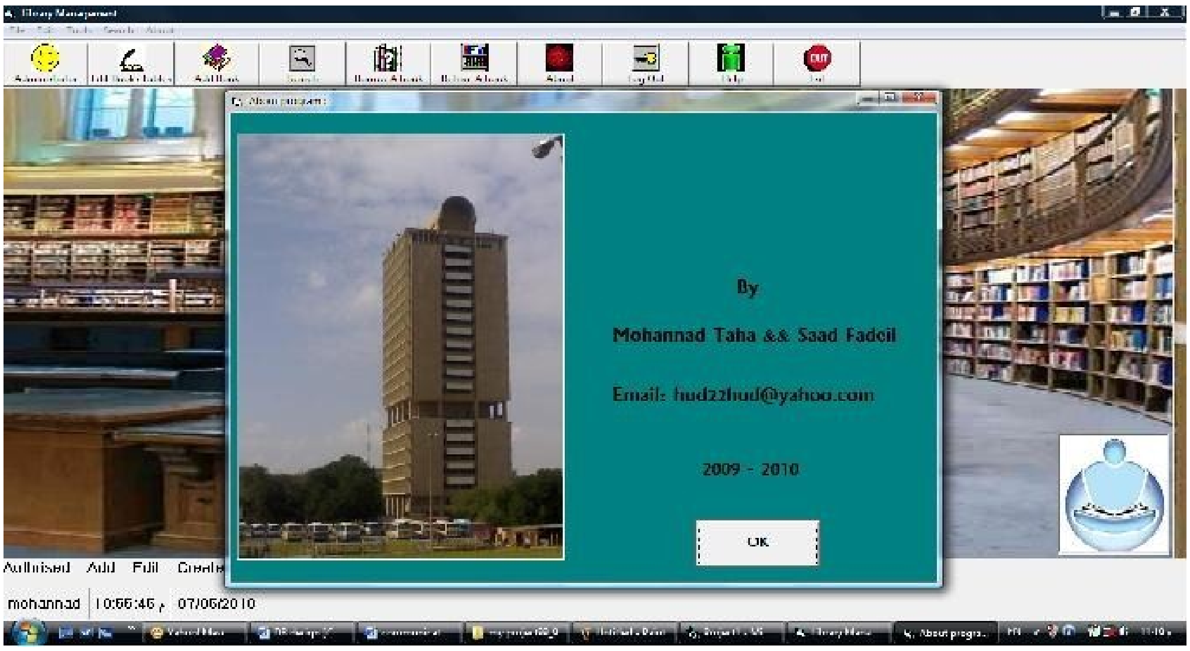


Fig (14)

## **4.11** About the program:

when clicking on the about command , some information appear about creator and super visitor of the program , and information about Baghdad university .



By

Mohannad Taha && Saad Fadeil

Email: hud22hud@yahoo.com

2009 - 2010

Finally

- The log out command will logout from a user without exit from the program , another user can login the program by enter a user name and password.
- The help command will display some help about the program
- The exit command will quit from the program

## 4.12 References :

[1] Nawras Hussein, "database design and implementation using visual basic " college of engineering, 2009.

[2] data base system C.J.Date Forth Edition

[3] John Carter "Data Base Design and programming

`